



THE WITCHER®
— ROLE-PLAYING GAME —

D'JINNI

USER'S MANUAL

Created by Slawomir "TobiAlex" Wasik
version 1.0.0.0 (15.01.2008 / 11:32)

Übersetzt von der Witcher-Community des WoP.

CDPROJEKT:

Final Version: 0.4.9.3
Fertiggestellt am 02.07.08 um 18:00 Uhr.

Inhaltsverzeichnis

Vorwort des Autors	Seite 4
Danksagungen	Seite 4
Der D'jinni Editor	
Einleitung	Seite 5
Ordner-Strukturen und Dateiformate	Seite 5
Editor Einstellungen	Seite 11
Interface	Seite 18
Module	
Einleitung	Seite 28
Erstellen eines Bereiches und eines der Module	Seite 28
Bewegen zwischen Areas	Seite 134
Charaktere	
Einleitung	Seite 138
Templates	Seite 138
Spawn Sets	Seite 179
Spawn Points	Seite 223
Action Points	Seite 233
Conversations	
Einleitung	Seite 243
Quest Conversations	Seite 243
Standard- und Cutscene-Konversations	Seite 266
Entertainment in Conversations	Seite 303

Quests

Einleitung [Seite 315](#)

Quests [Seite 315](#)

Quest Datenbank [Seite 351](#)

Running and Testing the Game

Einleitung [Seite 355](#)

Assigning and Adding Files to a Module [Seite 355](#)

Starten eines Moduls [Seite 365](#)

Testen eines Moduls [Seite 367](#)

Cutscenes

Einleitung [Seite 403](#)

Erstellen einer Cutscene [Seite 405](#)

Einführung ins Scripting

Einleitung [Seite 440](#)

Starten einer Cutscene [Seite 442](#)

Der Hexer und seine Ausrüstung [Seite 453](#)

Testen der Flags [Seite 460](#)

Hilfe beim Scripten [Seite 467](#)

Anhang

Einleitung [Seite 468](#)

Lager [Seite 468](#)

Geschäft [Seite 476](#)

Ruheplatz [Seite 486](#)

Schlusswort [Seite 492](#)

Vorwort des Autors

Willkommen im Handbuch des D'Jinni Editors. Dieses Handbuch, an dem ich über zwei Monate gearbeitet habe, ist für all diejenigen gedacht, die ihre eigenen Witcher-Abenteuer basteln wollen. Das Handbuch ist so verfasst, dass sowohl diejenigen damit etwas anfangen können, die noch nie mit einem Game-Editor gearbeitet haben, als auch jene, die sich bereits damit auskennen. Jene unter Euch, die bereits mit dem Neverwinter Nights Tool gearbeitet haben, werden ein wenig überrascht sein, denn der D'Jinni Editor ist vollkommen anders als das Neverwinter Nights Tool, obwohl The Witcher auf derselben Engine basiert.

Als ich den D'Jinni Editor das erste Mal angeschaut habe, schien mir alles viel zu kompliziert, um da durchzusteigen. Als ich mich allerdings länger damit beschäftigt hatte, erkannte ich, dass D'Jinni einer der besten Editoren ist, den ich bisher gesehen habe. Er ist ziemlich komplex, aber gerade darin liegt das große Potential vom D'Jinni.

Auf den folgenden Seiten werde ich Euch Schritt für Schritt durch den Arbeitsprozess eines Witcher-Moduls führen. Damit ihr einen besseren Überblick erhaltet, wie was funktioniert, werdet ihr vielen Screenshots (über 1.400) begegnen, die euch zeigen, welche Kommandozeilen ihr schreiben, und was ihr anklicken müsst. Einige Beispiele weichen leicht von dem ab, was in eurer Version vom D'Jinni steht. Der Grund ist einfach: Als ich mich daran setzte das Handbuch zu schreiben, wurde D'Jinni kontinuierlich vom CD Projekt RED Team überarbeitet. Ich habe meine Version immer gut auf dem aktuellsten Stand mit dem Entwicklerteam gehalten. Ich kann dennoch Kleinigkeiten übersehen haben.

Das Handbuch ist auf die gleiche Weise verfasst, wie bei der Beschreibung eines Beispiel-Modules, auf das ich eingehen werde. Es beginnt mit den Basics: Beispielsweise mit der Erschaffung eines Areal, Charaktere werden hinzugefügt, dann Dialoge und die Quests. Darum empfehle ich jedem, das gesamte Handbuch von Anfang bis Ende durchzulesen. Das ist eine Menge Material, das meiste davon nehmen die Screenshots ein. Ihr könnt ruhig das eine oder andere ausprobieren, wenn wir durch das Handbuch gehen. Nichts macht mehr Spaß als etwas Eigenes zu erschaffen.

Diesem Editor ist ein Modul beigelegt, das das Witcher-Team geschrieben hat. Es lohnt sich daher, das Modul "Price of Neutrality" zu laden, um einen Einblick zu bekommen, wie die Entwickler dieses Modul gemacht haben.

Danksagungen

Als Erstes möchte ich meiner geliebten Frau Kasia danken, die mich so großartig unterstützte, während ich am Handbuch gearbeitet habe. Bedauerlicherweise bekam sie sehr oft von mir zu hören: "Schatz, tut mir leid, ich kann jetzt nicht, ich muss schreiben." Sie hat mir sehr dabei geholfen, zu entscheiden, ob dies oder jenes wirklich verständlich ist, was ich geschrieben habe. Dabei hat sie natürlich selber mit dem D'Jinni herumprobiert. Ich möchte mich auch bei unserem kleinen elf Monate alten Sohn bedanken, der kaum geschrien hat, als ich am Arbeiten war.

Und zu guter Letzt geht mein Dank an das CD Projekt RED Team, dafür dass sie mir die Gelegenheit gaben das Handbuch zu schreiben. Sowie das Vertrauen, das sie mir entgegengebracht haben, und die technische Unterstützung. Am meisten danke ich ihnen jedoch für das großartige Spiel, das sie mit The Witcher geschaffen haben, und natürlich den D'Jinni-Editor selbst.

Sławomir "TobiAlex" Wąsik

Der D'jinni Editor

Einleitung

Bevor wir mit dem Erstellen der neuen Module für The Witcher beginnen, lohnt es sich ein paar Worte über den Editor selbst zu schreiben. In Wirklichkeit ist dies eine komplette Entwicklungsumgebung, die dafür benutzt wird um eigene Module zu erstellen. Die Entwickler verwendeten es selbst während der Entwicklung des Spiels. Jedermann der The Witcher gespielt oder gesehen hat, wird erkennen wozu der Editor fähig ist.

Seine charakteristischen Eigenschaften sind:

All-In-One-Editor:

Der D'jinni Editor besteht aus so genannten Plug-Ins. Unter der Verwendung des D'jinni, kannst Du Schablonen für Charaktere, Gespräche und Quests gestalten. Obwohl in Wahrheit, jede dieser Aufgaben, auf einen unabhängigen Editor hindeutet.

Multi-Tasking Architektur:

Erlaubt Dir, an mehreren Dateien sofort und simultan zu arbeiten. Du erschaffst einen neuen Charakter, Quest oder ein Gespräch.

Echtzeittests und Bug-Beseitigung:

Du kannst das Modul in Echtzeit testen das Du erschaffen hast, und kannst gleichzeitig überprüfen ob es funktioniert usw. Außerdem hast Du ein umfassendes Testprogramm zur Verfügung,

Flexibilität:

Es erlaubt Dir, das Arbeitsumfeld auf Deine persönliche Bedürfnisse einzustellen.

Dies sind die wichtigsten Gesichtspunkte des D'jinni Editors. Trotz des Aussehens ist der Editor nicht so schwer zu benutzen, und es nimmt gerade mal eine Stunde Zeit in Anspruch um sich mit ihm vertraut zu machen.

Ordner-Strukturen und Dateiformate

In diesem Unterkapitel werde ich die Struktur der Dateien und der Formate beschreiben, die durch den Editor erkannt werden. Diese Formate sind speziell für alle Ordner, die benötigt werden um Dein eigenes Abenteuer zu erstellen und gleichzeitig in denen es gespeichert wird. Es ist also ein sehr praktischer Bereich. Man sollte sich aber immer eines vor Augen halten: Der D'jinni Editor kann Ordner „nicht sehen“. Für ihn gibt es nur eine Folge von Dateien.

Nachdem Du den D'jinni gestartet hast, initialisiert er den Inhalt der Dateiformate. So weiß der Editor welche Dateien verfügbar sind.

Es ist sehr wichtig dies zu beachten, weil es Deine Arbeit auf zwei Weisen beeinflusst bestimmte.

Erstens bedeutet es, dass es keine zwei Dateien mit demselben Namen geben kann. Es ist nicht wichtig ob die Erste in dem Data\2DA-Ordner ist, und die Zweite in dem Data\Dialogues-Ordner. Es *kann keine* zwei Dateien mit demselben Name geben!

Zweitens, wenn der D'jinni bereits geöffnet ist, und Du erstellst einen neuen Ordner im Data-Ordner (z.B. über den Windows Explorer) wird dieser vom Editor nicht erkannt! Du musst dazu den D'jinni erneut starten. Alle Dateien müssen auf den D'jinni Editor erstellt werden.

Bitte beachte diese zwei wichtigen Dinge. Und bedenke auch, dass Namen max. 16 Zeichen und keine Sonderzeichen enthalten dürfen (z.B. auch keine Leerzeichen). Kommen wir jetzt zum eigentlichen Zweck die-

ses Handbuchs.

Dies ist die Struktur des Data-Ordners, zusammen mit den Kurzbeschreibungen von seinem Inhalt.

Laufwerk:\...\The Witcher\Data

\2DA Die .2da Dateien. Diese sind einfache Datenbanken mit verschiedenen Definitionen, die in The Witcher benutzt werden

\alchemy - .2da Ordner in Bezug auf Alchemie

\aremaps - .2da Ordner in Bezug auf die Umgebung

\combat - .2da Ordner in Bezug auf den Kampf

\creature - .2da Ordner in Bezug auf die Kreaturen

\cutscenes - .2da Ordner in Bezug auf Cutscenes

\gameplay - .2da Ordner in Bezug auf das Gameplay

\gfx - .2da Ordner in Bezug auf die Spezialeffekte

\sound - .2da Ordner bezogen für den Ton

\toolset - .2da Ordner in Bezug auf den D’jinni Editor

\Cutscenes alle .cut Ordner. Hier sind alle Dateien für die Cutscenes, die von der Witcher’s Spielengine genutzt werden

\Dialogues Die .dlg Dateien. Das sind die Dateien für die Gespräche der Charaktere

\Djinni Die Dateien die ausschließlich vom D’jinni genutzt werden

\FX Der .fx Ordner. Das sind die Dateien mit den Spezialeffekten

\Alchemy - Spezialeffekte in Bezug auf die Alchemie

\CameraEffects - Spezialeffekte in Bezug auf die Kameraeffekte

\Combat - Spezialeffekte in Bezug auf den Kampf

\Cutscenes - Spezialeffekte in Bezug auf Filmsequenzen

\Deaths - Spezialeffekte in Bezug auf das Sterben

\Decorations - Spezialeffekte in Bezug auf die Dekoration

\Gameplay - Spezialeffekte in Bezug auf das Gameplay

\GUI - Spezialeffekte in Bezug auf die grafische Schnittstelle

\Monsters - Spezialeffekte in Bezug auf die Monster

\Signs - Spezialeffekte in Bezug zu den Hexer-Zeichen

\Spawn - Spezialeffekte in Bezug auf das Wiederauftauchen (respawning)

\Work - Spezialeffekte in Bezug auf die Arbeit

\Items Die .mdl Dateien. Das sind 3D Items und Modelle, hergestellt unter der Benutzung des 3D Studio Max

\Material Die .mat Dateien. Das sind Dateien die Materialdefinitionen für 3D Modelle beinhalten

\Meshes Die .mdl Dateien. Das sind die 3D Mesh Models hergestellt unter der Benutzung des 3D Studio MAX

\Charaktere - 3D Models für Charaktere

\Critters - 3D Models für Monster

\CutsceneSets - 3D Models zur Benutzung für Cutscenes

\GUI - 3D Models zur Benutzung für das grafische Interface

\Items - 3D Models für die Items

\Locations - 3D Models für die Umgebung

\Placeables - 3D Models erstellbar

\Toolset - 3D Models zur Benutzung für den D'jinni Editor

\Module Die .adv Ordner. Dieses ist das Format, das für die Speicherung des Moduls verwendet wird

\Music Die .ogg Dateien. Das sind die Musik Dateien

\Quests Die .qst und .qdb Format Dateien. Die .qst Dateien sind Quests, hingegen die .qdb Dateien sind Datenbanken sie enthalten Sets für die Quests

\Scripts Die .nss und .lua Dateien. Die .nss Dateien und Scripts geschrieben in der Sprache aus Neverwinter Nights, hingegen die .lua Dateien sind Scripts geschrieben für die Lua Script Sprache

\Sounds Die .wav Dateien. Das sind verschiedene Sounddateien

\Creatures - Monstersounds

\Cutscenes - Für Cutscenes verwendete Sounds

\Doors - Türeounds

\Enemies - Gegnersounds

\Fistfight - Sounds verwendet für den Faustkampf

\Hero - Witchesounds

\Impacts - Aufprallsounds

\Inventar - Inventarsounds

\Items - Itemsounds

\Jingles - Klingelsounds

\Locations - Umgebungsspezifische Sounds

\Magic - Magiesounds

\Menu - Menüsounds

\Misc - Andere, verschiedene Sounds

\Movement - Bewegungsounds

\Sequence - Sound verwendet für Kampfsequenzen

\SoundGroups - Gruppensounds

\Special - Spezialsounds

\Voicesets - Stimmensounds

\Weapons - Waffensounds

\Templates Schablonen für verschiedene Dinge für The Witcher

\Actionpoints - Die .uta Dateien. Das sind Schablonen für Aktionspunkte

\CameraSphere - Die .cam Dateien. Das sind Schablonen für die Kamerakugel

\Characters - die .utc Files. Das sind Schablonen für die Charaktere

\Animals - Schablonen für Tiere

\Commoners - Schablonen für einfache Charaktere

\Cutscenes - Schablonen Verwendung für Cutscenes

\Enemies - Schablonen für Gegner

\Monsters - Schablonen für Monster

\NPC - Schablonen für NPCs

\Player - Schablonen für den Spieler

\Triggered - Schablonen für Auslöser

\Unique - einzige Schablonen

\Doors - die .utd Dateien. Dies sind Schablonen für die Türen, die benutzt werden um sich zwischen Bereichen, z.B. von der Straße in ein Gebäude zu bewegen

\Fx - die .wfx Dateien. Hier sind Dateien für die visuellen Effekte

\Gifts - die .uti Dateien. Hier sind Schablonen für Geschenke

\Items - die .uti Dateien. Hier sind Schablonen für Gegenstände

\NPCSpawn - die .npc Dateien. Hier sind Schablonen für NPCs

\Placeables - die .utp Dateien. Hier sind Schablonen für Platzier bares

\Containers - Schablonen für Behälter (Kisten, Fässer etc.)

\Cutscenes - Schablonen für Platzier bares zur Verwendung in Cutscenes

\Decorations - Schablonen für Dekorationen

\Light_Sources - Schablonen für Lichtsprünge

\Special - Spezialschablonen

\Story - Schablonen für Platzier bares in Bezug auf die Story

\Useables - Schablonen für Platzier bares und verwendbare Gegenstände

\Shops - Die .utm Dateien. Das sind Schablonen für den Laden und Warenhäuser

\Sounds - Die .uts Dateien. Das sind Schablonen für Sounds

\Spawnpoints - Die .utx Dateien. Das sind Schablonen für Wieder-Erzeugungspunkte

\Spawnsets - Die .spn Dateien. Das sind Schablonen für Erzeugungssets

\Triggers - Die .utt Dateien. Das sind Schablonen für Auslöser

\Waypoints - Die .utw Dateien. Das sind Schablonen für Wegpunkte

\Voices Sounddateien die von den Charakteren verwendet werden

Wenn Du die Ordner des D'jinni etwas untersuchst, findest Du im D'jinni Editor und im Spiel selbst, verschiedene Typen der Dateien. Für uns als die Schöpfer der neuen Module sind diese Dateien von geringer Bedeutung. Sie sind für den Editor und für das Spiel jedoch sehr wichtig. Dies schließt folgende Dateien mit ein:

Models:

.mdl Diese Dateien sind die mit Hilfe des 3D Studio Max geschaffene 3D-Modelle. Sie enthalten Beide die Modelle der Charaktere, Tiere, Standorte usw. und die Modelle vom grafischen Interface die auch vom Spiel verwendet werden

- **.mdv** Diese Dateien enthalten „virtuelle Animationen“. Sie haben den gleichen Namen, wie die .mdl Datei. Sollte es keine .mdv Dateien für die .mdl Datei geben, bedeutet es, dass dieses Model keine „virtuelle Animationen“ aufweisen kann

- **.mde** Diese Dateien enthalten Ereignisse, z.B. Sounds für die Animationen

- **.mda** Diese Dateien enthalten Animationen

- **.wok** Diese Dateien enthalten Modelle der Steuerung und Zusammenstöße. Sie werden ausschließlich für die Bereiche genutzt

- **.pkw, .dwk** Diese Dateien enthalten Schablonen für die Türenbenutzung

- **.set** Diese Dateien enthalten Bereiche und Informationen über die 3D Modelle die für die Umgebung verwendet werden z.B verschiedene Typen vom Gras

Textures:

- **.tga** Das ist das Grundformat für unkomprimierte Grafikdateien, die durch den D´Jinni Editor und The Witcher benutzt werden

- **.dds** Das ist das Grundformat für unkomprimierte Grafikdateien, die durch den D´Jinni Editor und The Witcher benutzt werden

- **.bmp** Diese Grafikdateien werden nur benutzt, um Bereichslandkarten zu schaffen, (dies sind Landkarten die der Spieler im Journal sehen kann.) Sie müssen aber in 8-Bit Format, d.h. in 256 Farben gespeichert werden

- **.txi** Dies sind keine Grafikdateien, aber sie enthalten die zusätzlichen Daten (z.B. Filter) verwendet für die angegeben Grafikdatei. Sie haben den gleichen Namen, wie angegeben

Sounds:

- **.mp3** .mp3 Sound Dateien (komprimiert)

- **.wav** .wav Sound Dateien (unkomprimiert)

- **.ogg** .ogg Sound Dateien (komprimiert)

- **.say** Dies sind keine Sound Dateien. Sie enthalten Informationen über die verwendeten Lippenbewegungen der Charaktere während der Gespräche. Sie haben denselben Namen wie die angegebene Audiodatei.

Physics:

- **.ka** Diese Dateien enthalten Informationen über Gamephysics. Sie wird vom Karma-System benutzt das auch im Witcher eingesetzt wird.
- **.psk** Diese Dateien enthalten physical Simulationen. Dieses Format wird benutzt bei der Unreal Engine.

Schließlich sollte ich auch die .bif Dateien erwähnen - diese sind Bibliotheksdateien, ähnlich .zip oder .rar Dateien, aber das Format wurde von der Witcher Mannschaft entwickelt. Diese Dateien enthalten alle anderen Dateien. Z.B. werden stichhaltige Dateien in der sounds.bif Datei gespeichert.

Das vom D´Jinni Editor und The Witcher verwendete Prioritätssystem ist es auch mal Wert erwähnt zu werden. Ich zeige Dir, wie es unter Verwendung eines Beispiels funktioniert, welches die sounds.bif Datei mit einbezieht.

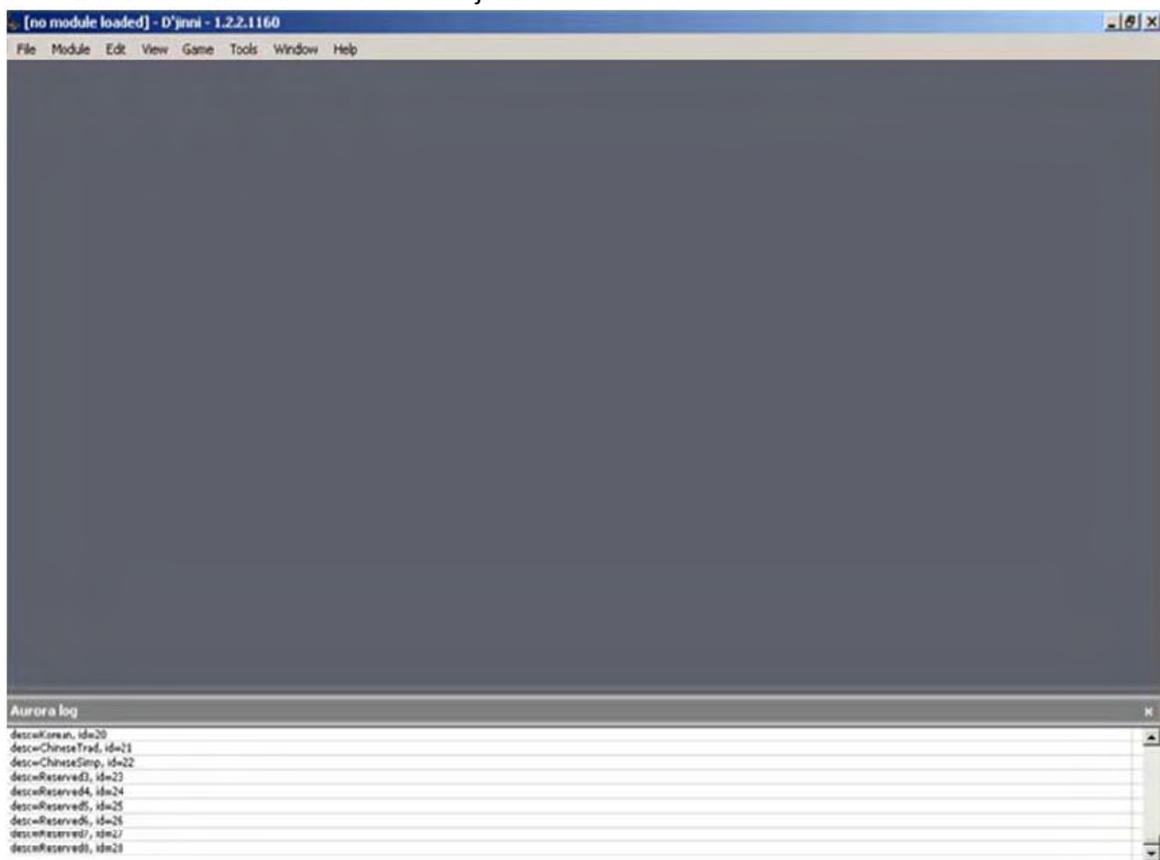
Lass uns die sounds.bif Dateien nehmen, Sie enthält bereits eine shot.wav Datei (eine arrowshot Sound Datei). Wenn Du eine Datei, desselben Namens erstellen d.h. ein shot.wav und es im Ordner des D´Jinni speicherst, überschneiden sich die Namen, schließlich haben wir bereits schon eine Datei die diesen Namen enthält, die in der Sound.bif Datei gespeichert werden.

Allerdings, wegen dem Weg des Prioritätenssystem wird während der Arbeit, nichts geschehen. Die unarchivierte shot.wav Datei hat eine höhere Priorität als die schot.wav Datei die in der Sound.bif Datei gespeichert wurde.

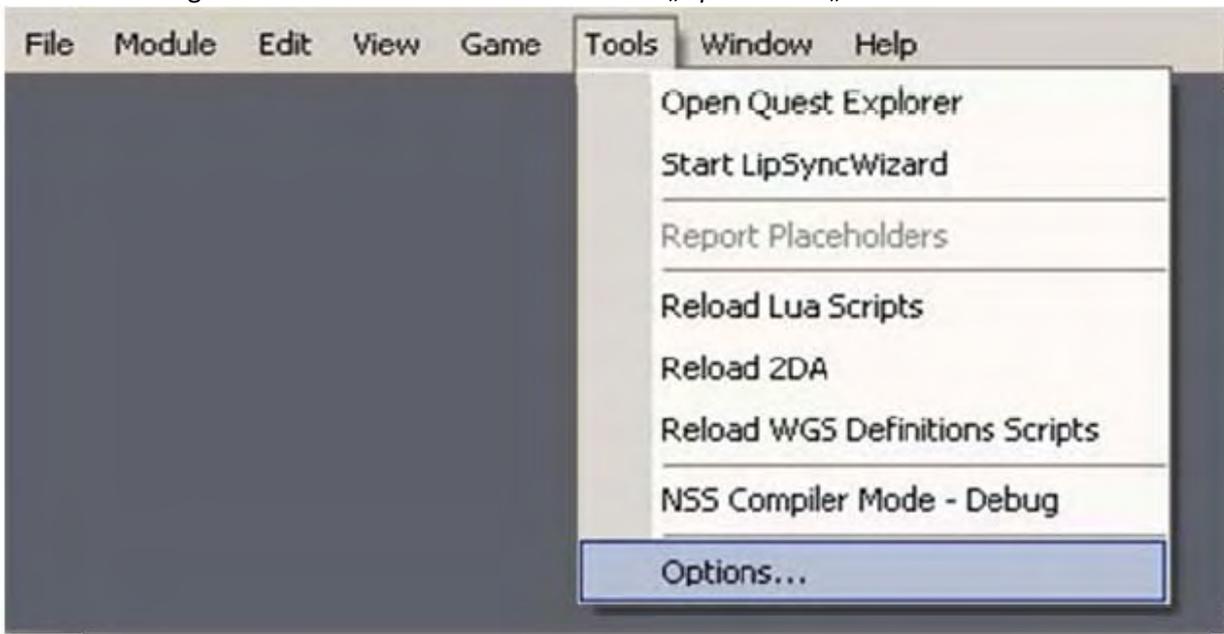
Das ist eine sehr gute Nachricht für uns, so wird unsere selbst erstellte Modifikation eine höhere Prioritätsdatei zugewiesen, und wir bewahren immer noch die identisch benannten Dateien in den. bif Dateien. Auf diese Weise bleiben die Originaldateien erhalten.

Editor Einstellungen

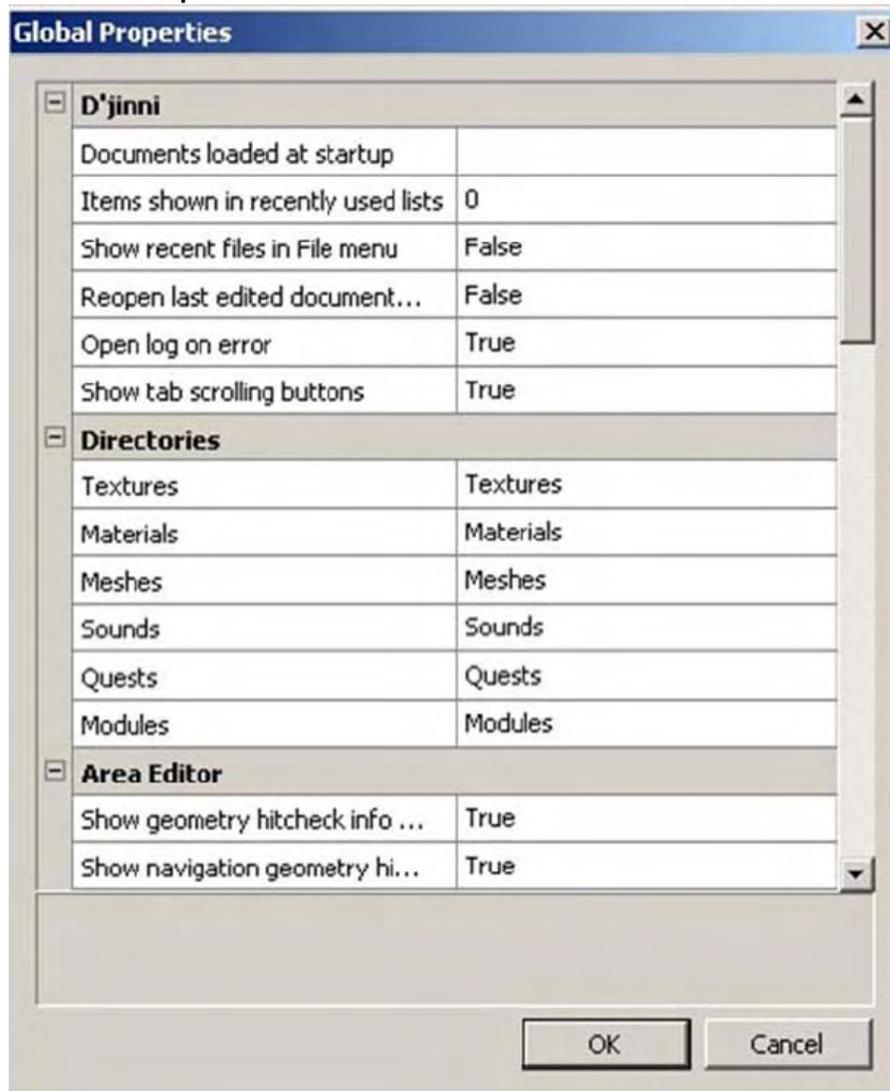
Starte den Editor. Normalerweise müsste es jetzt so aussehen:



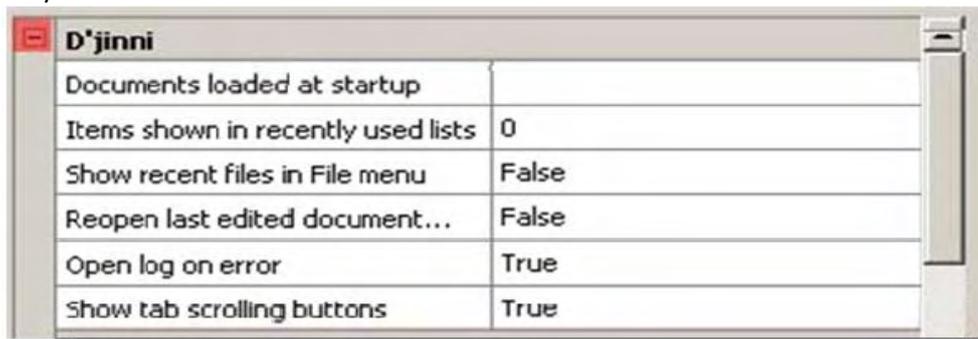
Im Moment hast du nur Zugriff auf das **Hauptmenü** des D'jinni Editors und auf das **Event Register** Fenster im **Aurora Log**. Später hängt die Menge an offenen Fenstern nur von deinen Vorlieben und Einstellungen ab. Doch bevor wir uns die Oberfläche des Editors im Detail anschauen, lass uns erstmal einen Blick auf seine Einstellungen werfen. Um das zu erreichen wähle „Options“ im „Tools“ Menü:



Nun öffnet sich das **Global Properties** Fenster:



Dieses Fenster beinhaltet die individuell einstellbaren Optionen des Editors, Das Fenster ist in folgende Sektionen unterteilt: *D'jinni*, *Directories*, *Area Editor*, *Build Wizard*, *Conversation Editor*, *Cutscene Editor*, *Game Debugging*, *Module Explorer*, *Resources Explorer*, *String Editor*, *Scene Editor Camera Movement Keys* und *String Database*. Um sich leichter zwischen den Sektionen zu bewegen, kannst du sie durch klicken auf ein kleines „-“ Symbol neben dem Sektionsnamen minimieren:



D'jinni	
Documents loaded at startup	
Items shown in recently used lists	0
Show recent files in File menu	False
Reopen last edited document...	False
Open log on error	True
Show tab scrolling buttons	True

Um eine vorher minimierte Sektion wieder zu maximieren, klicke auf das kleine „+“ Symbol neben dem entsprechendem Sektionsnamen:



+ D'jinni	
+ Directories	
+ Area Editor	
+ Build Wizard	
+ Conversation Editor	
+ Cutscene Editor	
+ Game debugging	
+ Module Explorer	
+ Resource Explorer	
+ String Editor	
+ Scene Editor Camera Movement Keys	
+ String Database	

Das Fenster ist in zwei Spalten unterteilt, in der linken Spalte steht der Name einer Einstellung, Und in der rechten der entsprechende Wert für diese Einstellung. Die Liste ist ziemlich lang, so dass du den Scrollbalken auf der rechten Seite zum Hoch- und Runter scrollen benutzen kannst:



Nachfolgend wirst du die entsprechenden Optionen im **Global Properties** Fenster lernen:

D'jinni Sektion:

Diese Sektion beinhaltet die grundlegenden Optionen des D'jinni Editors.

- **Documents loaded at startup.** Beinhaltet die Namen der Dokumente, die geladen werden, wenn der Editor gestartet wird.
- **Items shown in recently used lists.** Gib die Anzahl der angezeigten Einträge aus der kürzlich benutzten Dokumentenliste an. Der Wert 0 gibt an, dass keine der kürzlich benutzten Dokumente angezeigt werden.
- **Show recent files in File menu.** Gibt an, ob die kürzlich benutzten Dateien im File Menü angezeigt werden. Die möglichen Werte für diese Option sind:

- *False* – kürzlich benutzte Dateien werden nicht im File Menü angezeigt.

- *True* – kürzlich benutzte Dateien werden im File Menü angezeigt.

- **Reopen last edited documents after restart.** Gibt an, ob die als letztes bearbeiteten Dokumente beim nächsten Start des Editors geöffnet werden sollen. Die möglichen Werte für diese Option sind:

- *False* – bearbeitete Dokumente werden nicht automatisch beim nächsten Start des Editors geöffnet.

- *True* – bearbeitete Dokumente werden automatisch beim nächsten Start des Editors geöffnet.

- **Open log on error.** Gibt an, ob das **Event Register** Fenster geöffnet werden soll, wenn ein Fehler auftritt. Die möglichen Werte für diese Option sind:

- *False* – das Event Register Fenster öffnet sich nicht automatisch, wenn ein Fehler auftritt.

- *True* – das Event Register Fenster öffnet sich automatisch, wenn ein Fehler auftritt.

Directories Sektion:

Diese Sektion beinhaltet die Verzeichnisnamen zu den gegebenen Elementen im D'jinni Editor.

- **Textures.** Beinhaltet den Namen für das Textures-Verzeichnis.
- **Materials.** Beinhaltet den Namen für das Materials-Verzeichnis.
- **Meshes.** Beinhaltet den Namen für das Meshes-Verzeichnis.
- **Sounds.** Beinhaltet den Namen für das Sounds-Verzeichnis.
- **Quests.** Beinhaltet den Namen für das Quests-Verzeichnis.
- **Modules.** Beinhaltet den Namen für das Modules-Verzeichnis.

Area Editor Sektion:

Diese Sektion beinhaltet Optionen für den Area Editor. Dieser stellt eines der wichtigsten Werkzeuge des D'jinni Editors dar. Mit diesem Werkzeug kannst Du Objekte platzieren und ihre Eigenschaften einstellen.

- **Show geometry hitcheck info in status bar.** Gibt an, ob im Statusbalken die geometrischen Koordinaten des Mauszeigers angezeigt werden sollen. Der Statusbalken befindet sich am Fuße des Editors. Die möglichen Werte für diese Option sind:

- *False* – Koordinaten werden nicht angezeigt,

- *True* – Koordinaten werden angezeigt.

- **Show navigation geometry hitcheck info in status bar.** Gibt an, ob im Statusbalken die Navigationskoordinaten des Mauszeigers angezeigt werden sollen. Die möglichen Werte für diese Option sind:

- *False* – Koordinaten werden nicht angezeigt,

- *True* – Koordinaten werden angezeigt.

- **Use walkmesh material mask in hitcheck.** Gibt an, ob das geometrische Material während der Navigation benutzt werden soll. Die möglichen Werte für diese Option sind:

- *False* – Material wird nicht benutzt,

- *True* – Material wird benutzt.

- **Show geometry hitcheck info under mouse cursor.** Diese Option gibt an, ob die geometrischen Koordinaten unterhalb des Mauszeigers angezeigt werden sollen. Die möglichen Werte für diese Option sind:

- *False* – Koordinaten werden nicht angezeigt,

- *True* – Koordinaten werden angezeigt.

- **Show navigation geometry hitcheck info under mouse cursor.** Diese Option gibt an, ob die Navigationskoordinaten unterhalb des Mauszeigers angezeigt werden sollen. Die möglichen Werte für diese Option sind:
 - *False* – Koordinaten werden nicht angezeigt,
 - *True* – Koordinaten werden angezeigt.
- **Real-time instance placement verification.** Gibt an ob die Verteilung von Spawn Point, Action Points und Placeables in Echtzeit getestet werden soll, z.B. wenn die Vorlage bearbeitet wird. Die möglichen Werte für diese Option sind:
 - *False* – die Verteilung wird nicht getestet,
 - *True* – die Verteilung wird getestet.

Build Wizard Sektion:

Diese Sektion beinhaltet Optionen zur Erstellung verschiedener Versionen des Spiels. Diese interessieren uns nicht wirklich.

- **Compiler Path.** Beinhaltet den Pfad zum Compiler.
- **Log File Path.** Beinhaltet den Pfad zur Event Log-Datei.
- **Log File Enabled.** Gibt an, ob die zusammengetragenen Ereignisse in der Event Log-Datei gespeichert werden sollen. Die möglichen Werte für diese Option sind:
 - *False* – Ereignisse werden nicht in der Event Log-Datei gespeichert,
 - *True* – Ereignisse werden in der Event Log-Datei gespeichert.

Conversation Editor Sektion:

Diese Sektion beinhaltet Optionen für den Conversation Editor (Dialoge). Mit diesem Editor entwirfst du die Dialoge zwischen Charakteren.

- **Show full conversation entries.** Gibt an, ob die kompletten Einträge während dem Entwurf einer Konversation angezeigt werden sollen. Die möglichen Werte für diese Option sind:
 - *False* – die komplette Einträge werden nicht angezeigt,
 - *True* – die kompletten Einträge werden angezeigt.

Cutscene Editor Sektion:

Diese Sektion beinhaltet Optionen für den Movie Editor. Mit diesem Editor entwirfst du Filme mit Hilfe der Spiel-Engine von The Witcher.

- **Autosave enable.** Gibt an, ob der augenblicklich geöffnete Film automatisch gespeichert werden soll. Die möglichen Werte für diese Option sind:
 - *False* – das automatische Speichern ist ausgeschaltet,
 - *True* – das automatische Speichern ist angeschaltet.
- **Autosave interval.** Gibt die Zeit (in Minuten) an, nach der, der augenblicklich geöffnete Film automatisch gespeichert werden soll.

Game Debugging Sektion:

Diese Sektion beinhaltet Optionen zum Testen von Modulen.

- **Show all quest on module start.** Gibt an, ob alle Quests angezeigt werden sollen, wenn das Modul gestartet wird. Die möglichen Werte für diese Option sind:
 - *False* – Quests werden nicht angezeigt,
 - *True* – Quests werden angezeigt.
- **Enable 'BestPerformance' mode on start.** Gibt an, ob der Modus für die beste Leistung automatisch aktiviert werden soll, wenn das Modul gestartet wird. Die möglichen Werte für diese Option sind:
 - *False* – 'BestPerformance' Modus wird nicht aktiviert,
 - *True* – 'BestPerformance' Modus wird aktiviert.

Module Explorer Sektion:

Diese Sektion beinhaltet Optionen für den Module Explorer. Der Module Explorer wird zur Erschaffung und Bearbeitung von Gebieten benötigt.

- **Load last loaded module at startup.** Gibt an, ob das zuletzt benutzte Modul beim Start vom D'jinni Editor wieder geladen werden soll. Die möglichen Werte für diese Option sind:
 - *False* – zuletzt benutztes Modul wird nicht geladen,
 - *True* – zuletzt benutztes Modul wird geladen.
- **Always reload the module on game end.** Gibt an, ob das momentan benutzte Modul neu geladen werden soll, wenn du es im D'jinni Editor getestet hast. Die möglichen Werte für diese Option sind:
 - *False* – Modul wird nicht neu geladen,
 - *True* – Modul wird automatisch neu geladen.

Resources Explorer Sektion:

Diese Sektion beinhaltet Optionen für den Resources Explorer. Dieser Explorer verwaltet die Dateien/Ressourcen, die benötigt werden um ein neues Modul zu erschaffen.

- **TortoiseProc Path.** Gibt den Zugangspfad zu dem Programm an, dass die Kompatibilität kontrolliert.
- **Silent script compilation.** Gibt an, ob Skripts erst kompiliert und dann auf ihre Kompatibilität kontrolliert werden sollen. Die möglichen Werte für diese Option sind:
 - *False* – die Skripts werden vor dem Kompatibilität-Check nicht kompiliert.
 - *True* – die Skripts werden vor dem Kompatibilität-Check kompiliert.

String Editor Sektion:

Diese Sektion beinhaltet Optionen für den String Editor. Dieser Editor wird zur Übersetzung der Texte in verschiedene Sprachen benutzt.

- **Database Path.** Beinhaltet den Zugangspfad zu der String Database.
- **Lock Path.** Beinhaltet den Zugangspfad zu der verschlossenen (abgesicherten) String Database.

Scene Editor Camera Movement Key Sektion:

Diese Sektion beinhaltet die Tasteneinstellungen, die benötigt werden, um sich innerhalb des Area Editors fortzubewegen.

- **Forward.** Bewege die Kamera nach vorne.
- **Backward.** Bewege die Kamera nach hinten.
- **StrafeLeft.** Bewege die Kamera nach links.
- **StrafeRight.** Bewege die Kamera nach rechts.
- **Upward.** Bewege die Kamera nach oben.
- **Downward.** Bewege die Kamera nach unten.
- **PitchUp.** Kamera nach oben neigen.
- **PitchDown.** Kamera nach unten neigen.
- **YawLeft.** Kamera nach links drehen.
- **YawRight.** Kamera nach rechts drehen.

String Database Sektion:

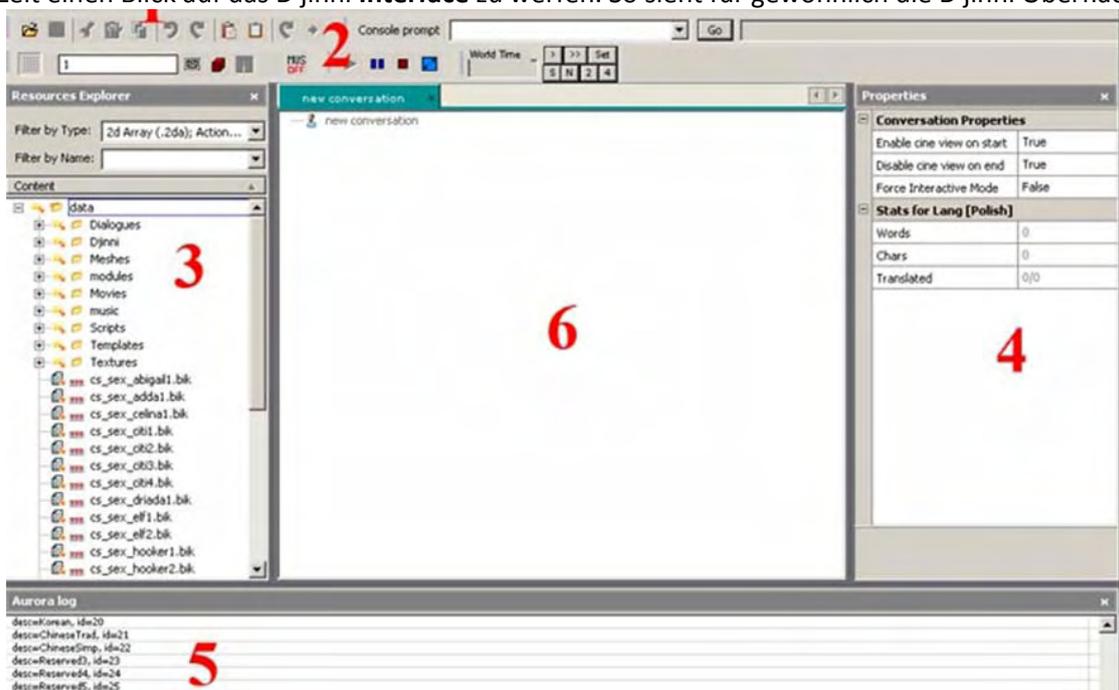
Diese Sektion beinhaltet Optionen für die String Database.

- **String Database Path.** Beinhaltet den Zugangspfad zur String Database.

Wenn du eine bestimmte Einstellung ändern willst, dann klicke auf die OK-Schaltfläche im Global Properties Fenster. Ich empfehle jedoch, die vorgegebenen Einstellungen beizubehalten.

INTERFACE:

Es ist Zeit einen Blick auf das D'jinni **Interface** zu werfen. So sieht für gewöhnlich die D'jinni Oberfläche aus:



1. Das Hauptmenü des Editors.
2. Die Editor Symbolleiste.
3. Der Ressourcen Explorer, Modul Explorer und das Spiel Objekt Prüf Fenster.
4. Die Eigenschaften, Favoriten und das Region Listen Fenster .
5. Das Aurora log und Konsolen Überwachungs Fenster.

6. Der Arbeitsplatz.

Jedes dieser Elemente wird beginnend im Hauptmenü, ausführlicher beschrieben.

Der Zugang der Hauptmenübefehle hängt davon ab, welches Fenster z.Z. auf dem Arbeitsplatz des Editors geöffnet ist.

> **Datei** – dieses Menü enthält Befehle ähnlich dem Dateisystem

> **Neu** – öffnet eine Liste zum Erstellen eines neuen Befehlsordners

> **2DA File (.2da)** – erstellt einen neuen .2da-Ordner

> **Cutscene (.cut)** – erstellt einen neuen Cutscene-Ordner

> **Conversation (.dlg)** – erstellt einen neuen Gesprächs-(dialogue)-Ordner

> **Lua Script (.lua)** – erstellt einen neuen Lua-Sprachen-Script-Ordner

> **Meta Module (.mmd)** – erstellt einen neuen Meta-Module-Ordner

> **Composite model(.mdl)** – erstellt ein neues Objekt-Model auf der Basis des vorherigen Modells

> **Module(.adv)** – erstellt ein neues Modul

> **Neverwinter Console Macro (.ncm)** – erstellt ein neues Console-Macro

> **Neverwinter Script(.nss)** – erstellt ein neues Neverwinter-Script

> **Quest Database(.qdb)** – erstellt eine neue Quest-Datenbank

> **Quest File (.qst)** – erstellt eine neue Quest

> **Spawn Set (.spn)** – erstellt ein neues Spawn-Set

> **Scene (.scn)** – erstellt eine neue Szene

> **Story NPC (.npc)** – erstellt ein neues NPC-Spawn-Set

> **Template**

> **Actionpoint (.uta)** – erstellt eine neue Schablone für die Aktionspunkte

> **Creature (.utc)** – erstellt eine neue Schablone für einen Charakter

> **Door (.utd)** – erstellt eine neue Schablone für eine Tür

> **Item (.uti)** – erstellt eine Schablone für ein Item

> **Placeable (.utp)** – erstellt eine neue Schablone für Platzier bares

> **Shop (.utm)** – erstellt einen neue Schablone für ein Geschäft

> **Sound (.uts)** – erstellt eine neue Schablone für den Ton

> **Spawnpoint (.utx)** – erstellt eine neue Schablone für Spawnpunkte

- > **Trigger (.utt)** – erstellt eine neue Schablone für ein Ereignis
- > **Waypoint (.utw)** – erstellt eine Schablone für einen Wegpunkt
- > **Text Note (.txt)** – erstellt eine Text Datei
- > **WootFX Effect (.wfx)** – erstellt neue Spezial-Effekte
- > **Open...** – öffnet eine Datei
- > **Open recent** – enthält eine Liste der zuletzt bearbeiteten Dateien
- > **Close** – schließt die entsprechende Datei
- > **Save** – Speichert die entsprechende Datei
- > **Save as...** – speichert die entsprechende Datei als ein neues Spiel
- > **save a copy** – speichert die entsprechende Kopie der Datei
- > **Save to Module As...** – speichert die entsprechende Datei direkt in das Modul
- > **Exit** – schließt den D’jinni Editor
- > **Module** – dieses Menü enthält Befehle zum Verändern des Moduls
- > **New module (.adv)** – erstellt ein neues Modul
- > **Open module...** – öffnet ein Modul
- > **Open recent** – enthält eine Liste von vor kurzem geöffneten Modulen
- > **Import Module...** – importiert ein Modul
- > **Save Module** – speichert das Modul
- > **Save Module As...** – speichert das Modul unter einem neuen Namen
- > **Close Module** – schließt das Modul
- > **Edit** – dieses Menü enthält Befehle für veränderbare Dateien
- > **Undo** – macht die letzte Aktion rückgängig
- > **Redo** – wiederholt den letzten Vorgang
- > **Cut** – schneidet das gewählte Objekt aus
- > **Copy** – kopiert das gewählte Objekt
- > **Paste** – fügt das gewählte Objekt ein
- > **Clear** – löscht das gewählte Objekt

- > **Select All** – markiert alle Objekte
- > **View** – diesen Menü enthält Befehle für das Anzeigen von zusätzlichen Utility Tools
- > **Utility Windows** – zeigt oder versteckt die zusätzlichen Utility Tools
- > **Aurora log** – zeigt das Ereignis-Registers-Fenster
- > **Console Watch** – zeigt das Konsolen-Uhrfenster
- > **Favourites** – zeigt die Favoriten im Fenster
- > **Game Object Inspector** – zeigt das Spiele-Objekt-Prüffenster
- > **Lua Watch 1** – zeigt das Lua Script Überwachungsfenster
- > **Lua Watch 2** – zeigt das Sekunden Lua Script-Überwachungsfenster
- > **Module Explorer** – zeigt das Modul-Explorer-Fenster
- > **Properties** – zeigt die Eigenschaften im Fenster
- > **Quest Explorer** – zeigt das Quest-Explorer-Fenster
- > **Region List** – zeigt die Regionen-Liste im Fenster
- > **Resources Explorer** – zeigt das Ressourcen-Explorer-Fenster
- > **VirtualMem Viewer** – zeigt das virtuelle Speicherfenster an
- > **Toolbars** – zeigt an oder versteckt die D’jinni-Editor-Symbolleiste
- > **Area Editor Tools** – zeigt die Bereichs-Editor-Symbolleiste an
- > **Console** – zeigt die Befehlssymbolleiste an
- > **Game** – zeigt die Spieleleiste
- > **Game time** – zeigt die Spieluhrsymbolleiste
- > **Standard** – zeigt die Standardsymbolleiste für den D’jinni-Editor
- > **Language** – stellt die Sprachauswahl für Texte ein
- > **Status bar** – schaltet die Statusleiste des D’jinni-Editor an oder aus
- > **Game** – dieses Menü enthält Befehle zum Starten des Spiels im D’jinni-Editor-Level
- > **Play** – lässt ein Modul auf den D’jinni-Editor-Level laufen
- > **Pause** – pausiert das gegenwärtige Modul
- > **Stop** – stoppt das gegenwärtige Modul
- > **Using MetaModule** – setzt das Meta-Modul vom Modul auf Anfang

- > **Fullscreen** – startet das Modul im Vollbildmodus
- > **Tools** – dieses Menu enthält Befehle für das Öffnen der zusätzlichen Werkzeuge
- > **Open Quest Explorer** – öffnet den Quest-Explorer
- > **Start LipSyncWizard** – startet den LipSyncWizard (wird verwendet, um Lippenbewegungen während der Gespräche zu synchronisieren)

- > **Reload Lua Scripts** – lädt die Lua-Scripts neu

- > **Reload 2DA** – lädt die .2da Datei neu

- > **NSS Compiler Mode** – Debug – startet den Neverwinter-Script-Test

- > **Options...** – Freie Optionen für das Anpassen des D`jinni-Editors
- > **Windows** – diese Menü enthält Befehle, bezogen auf das geöffnete Fenster
- > **Open Windows** – Schalter zwischen den geöffneten Fenstern (wenn Du mehr als eine Datei geöffnet hast)

- > **Close** – schließt das gegenwärtige Fenster
- > **Help** – enthält allgemeine Informationen über den D`jinni-Editor

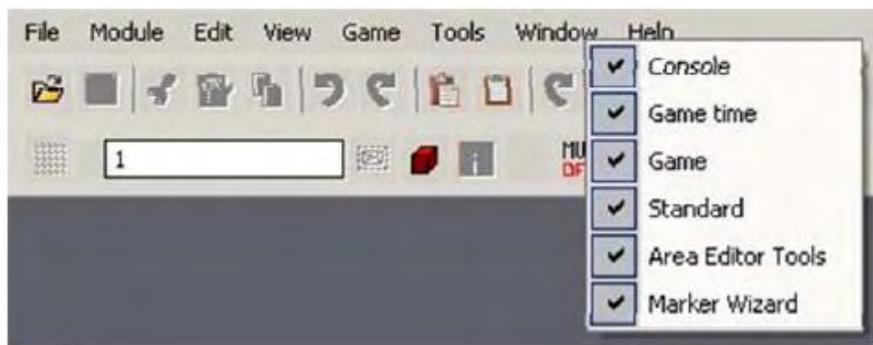
Dies alles umfasst das **Hauptmenü** des D`jinni Editors. Wenn wir zufällig auf ein neues Menü stoßen während unserer Arbeit, werde ich es Dir näher beschreiben. (z.B. dem **Ressource-Explorer**).

Werfen wir jetzt einen näheren Blick auf die **Werkzeugleiste** des D`Jinnis.

Erinnere dich, Du kannst unter Verwendung der Ansicht etwas anzeigen oder verbergen. Auf der Symbolleiste kannst Du durch Rechtsklick im **Hauptmenü** des D`jinni-Editors folgendes Menü öffnen:



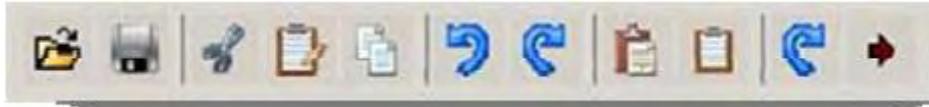
Um eine Werkzeugleiste anzuzeigen, reicht es eine entsprechende Option im Hauptmenü zu wählen. Eine aktive Leiste wird angewählt. Im nächsten Bild kannst Du sehen, dass alle Werkzeugleisten aktiviert wurden.



Um eine der angegebenen Symboleisten zu deaktivieren, klicke sie einfach im Menü an.

Jetzt beschreibe ich dir die Bedeutungen der Symbole auf den Symboleisten:

- **Standard:** Die Standardwerkzeuggestreife des D'Jinni-Editors



1. Öffnen einer Datei
2. Speichern einer Datei
3. Ausschneiden
4. Kopieren
5. Einfügen
6. Zurück
7. Wiederholung
8. Anzeigen des Resource-Explorers
9. Anzeigen des Modul-Explorers

- **Console:** Diese Symboleiste wird für die Konsolenbefehle gebraucht



Trage einen Konsolenbefehl in das Konsolenaufforderungsfeld ein, und klicke danach auf weiter. Dieser Befehl wird jetzt ausgeführt, und das Ergebnis erscheint auf der rechten Seite neben dem Go-Button. Das Konsolenaufforderungsfeld zeigt auch eine Liste der zuletzt bearbeiteten Konsolenbefehle an.

- **Bereichs-Editor-Werkzeug.** Diese Symboleiste wird verwendet um einen Bereich zu Editieren.

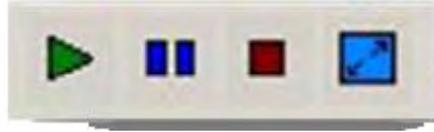


1. Dreht das Rasterfeld und schaltet die Schwerkraft im Editbereich an/aus. Dieses Rasterfeld mit der deaktivierten Schwerkraft erleichtert die Bewegungen der Objekte in diesem Bereich
2. Einstellen der Rasterskala
3. Greifen des Gitters
4. Anzeigen der versteckten Hilfssymbole im Editbereich.
5. Nicht genutzt

6. Lokale Operation

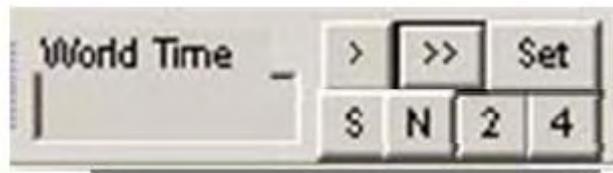
7. Ausschalten der Hintergrundtöne.

- **Game.** Diese Symboleiste wird verwendet, um das Modul im D'jinni Editor Level laufen zu lassen.



1. Startet das Modul im Spiel-Modus
2. Pausieren
3. Stoppen des Moduls
4. Schaltet den Modus auf Vollbildschirm um

- **Game time:** Diese Symboleiste wird verwendet, um die Zeit während des Spiels zu steuern.



1. Das Weltzeitfeld zeigt die korrekte Uhrzeit an (natürlich die Ingame-Zeit, ist doch klar)
2. > - Macht einen einfachen Zeitsprung
3. >> - Macht einen doppelten Zeitsprung
4. set - Erlaubt dir die Zeit selber einzustellen



5. S verlangsamt das Spiel
6. N - Das Spiel läuft in Normalgeschwindigkeit
7. 2 - Das Spiel läuft in 2 facher Geschwindigkeit
8. 4 - Das Spiel läuft in 4 facher Geschwindigkeit

Bevor wir zum praktischen Gebrauch des D’jinni-Editors kommen, möchte ich noch etwas über das Fenster des Editors hinzufügen. Wie zuvor erwähnt, ist der D’jinni ein Multifunktionseditor, der dir erlaubt damit alles zu tun. Ein Beispiel mit verschiedenen geöffneten Fenstern ist weiter unten:



Es gibt vier geöffnete Fenster. Die deaktivierten Fenster sind grau, während die aktiven Fenster eingefärbt werden. Um ein Fenster zu aktivieren, klicke auf seinen Namen. Wenn du z.B. das vierte Fenster aktivieren möchtest, klicke auf das **new spawn set** Fenster:



Das Fenster wird sofort aktiviert:



Wenn du ein Fenster schließen willst, klicke auf das kleine „x“ Zeichen das du auf der rechten Seite des Registers findest:



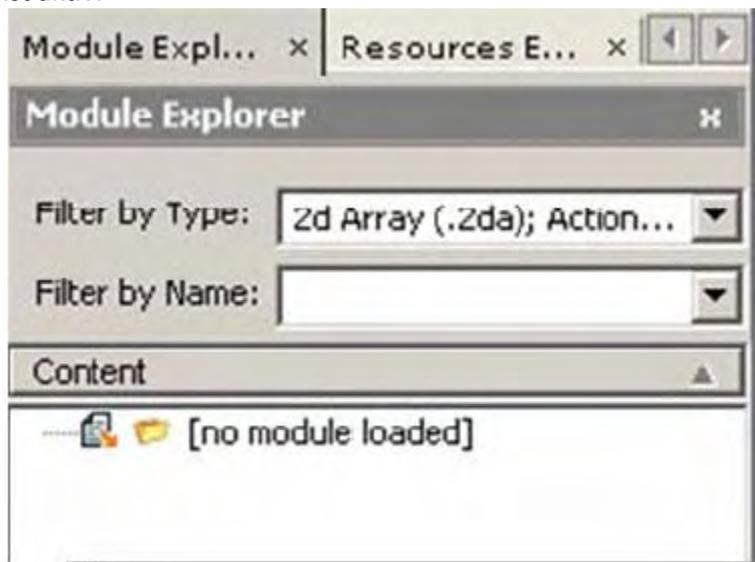
Das Fenster wird geschlossen:



Wenn Du so viele Fenster geöffnet hast, dass sie nicht mehr in den Arbeitsbereich passen, kannst Du mit den Pfeilen, oben rechts im Arbeitsfenster, nach links und rechts scrollen:



Der **Ressource-Explorer** und der **Module-Explorer** arbeiten ähnlich in Fenstern. Sie können gleichzeitig geöffnet werden, aber nur eins kann aktiv sein. Im Beispiel unten werden beide Fenster angezeigt aber nur der **Module-Explorer** ist aktiv:



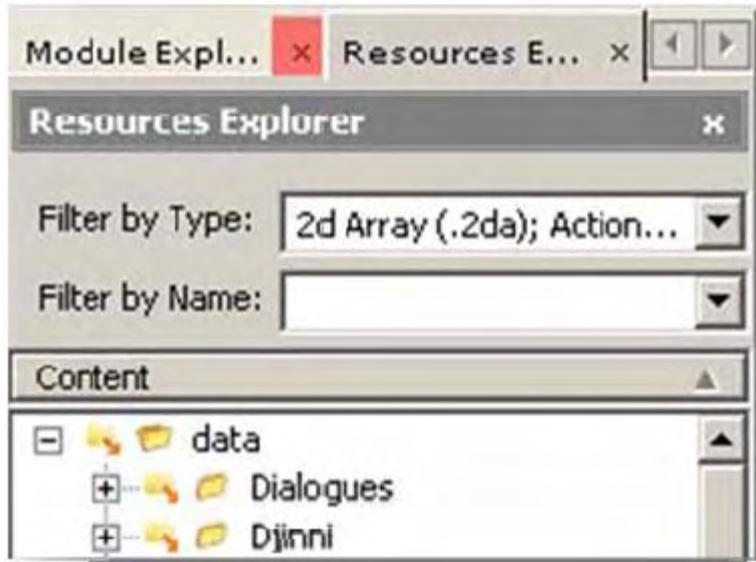
Wenn Du den **Ressources-Explorer** aktivieren möchtest, klicke auf den Namen:



Der **Ressource-Explorer** wird aktiv:



Wenn Du das Fenster schließen willst, klicke auf das „x“ rechts oben in der Registerkarte:



Das Fenster wird geschlossen.



Dies ist das äußere Erscheinungsbild des D`Jinni Editor Interface. Ich denke jetzt können wir zur praktischen Anwendung übergehen.

Module

Einleitung:

Ein Modul setzt eins der wichtigsten Elemente des Witcher fest. Es ist ein Set, der Elemente des ganzen Spiels, seine Bereiche, Charaktere, Schablonen, Spawn sets, Quests usw. das alles wird später im Handbuch beschrieben. Jedes Modul besteht mindestens aus einem Bereich (genannt wird es auch die Position). Wohin wir auch alle anderen Gegenstände wie, Kisten, Tische, Stühle, Charaktere und Spawn Punkte und Aktion Punkte usw. ablegen können.

Was ist ein Bereich? Ein Bereich ist ein einfach Isoliertes Fragment von der Welt des Spiels.

Es kann auch von inneren und äußeren gewaltigen Freiräumen wie Sümpfe oder Wälder bestehen. Das ist der Grund warum die Bereiche in zwei unterschiedlichen Kategorien eingeteilt werden können: Äußere gewaltige Bereiche, und umfassend gewaltige Freiräume(Wälder, Sümpfe, Straßen, Städte usw.): und auch innere Gebäude wie Höhlen, Krypten, Abwasserkanäle dies alles kann es umfassen.

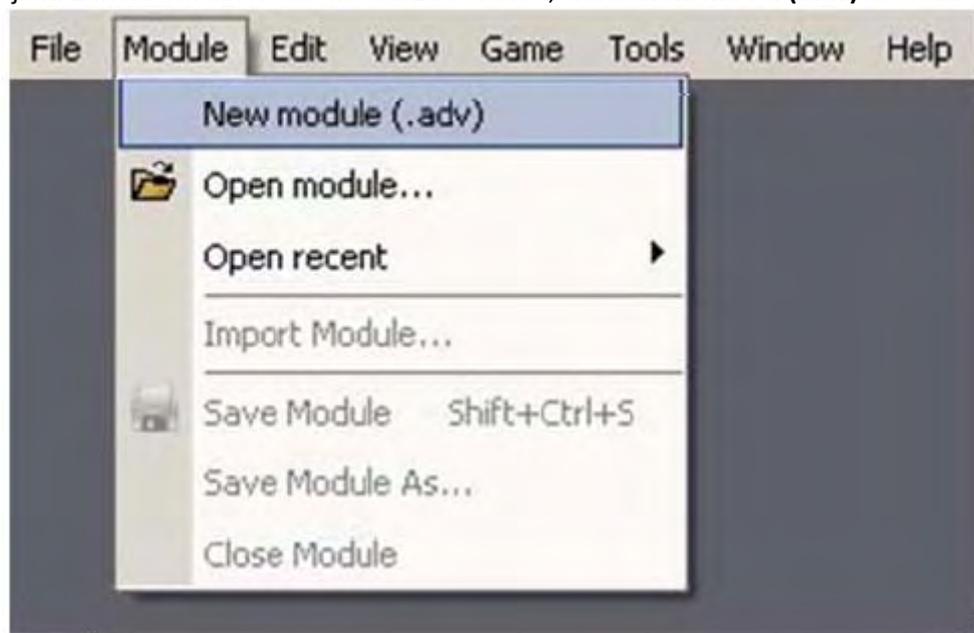
Anders als die anderen Editoren – wie das Neverwinter Aurora Toolset, der Chrom-Editor, oder der Unreal-Editor werden alle Bereiche mit dem 3D Studio Max hergestellt. Zusammen mit dem Editor erhältst du einige unterschiedliche Bereiche und das kannst du in deinem Abenteuer verwenden. Um zu erklären wie man eigene Witcher Abenteuer erstellt, demonstriere ich mit einfachen Quests wie Geralt also der(Spieler) einen Banditenanführer (alias den Professor) beseitigen muss.

Siegfried hat die Aufgabe uns zugeteilt, wir haben vor in der Story etwas mehr zu machen und geben Farbe durch das hinzufügen einer Wache indem wir Jehro zum Schutz herbei holen der Leidenschaftlich würfelt. All die Charaktere sind vertraute Bewohner von The Witcher um Geralt bei Laune zuhalten fügen wir ein Paar Wölfe hinzu. Wir schaffen die ganzen Quest in einem Bereich entweder in der Höhle oder im Wald.

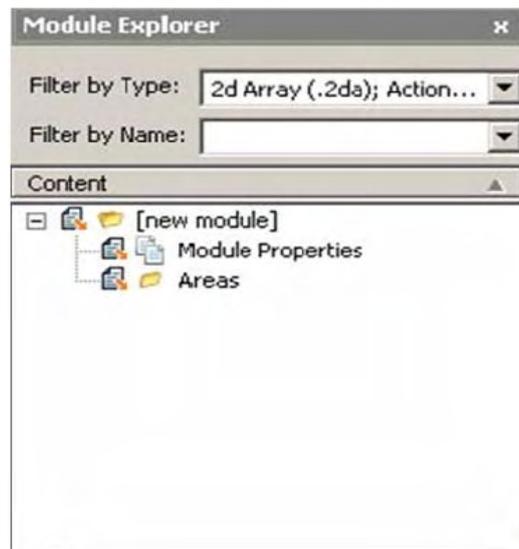
Das liegt natürlich nur bei Dir, und ich empfehle Dir dass du selber ein wenig experimentierst. Lass uns dennoch damit anfangen.

Erstellen eines Bereichs und eines der Module:

Starte den D`jinni Editor. Um ein neues Modul zu schaffen, wähle **new Modul (.adv)** über das Modulsmenü:



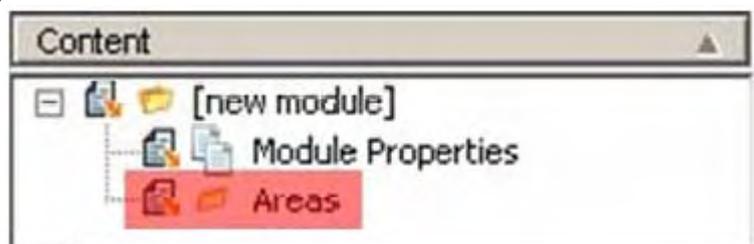
Der **Module-Explorer** geht auf der linken Seite des Bildschirms auf:



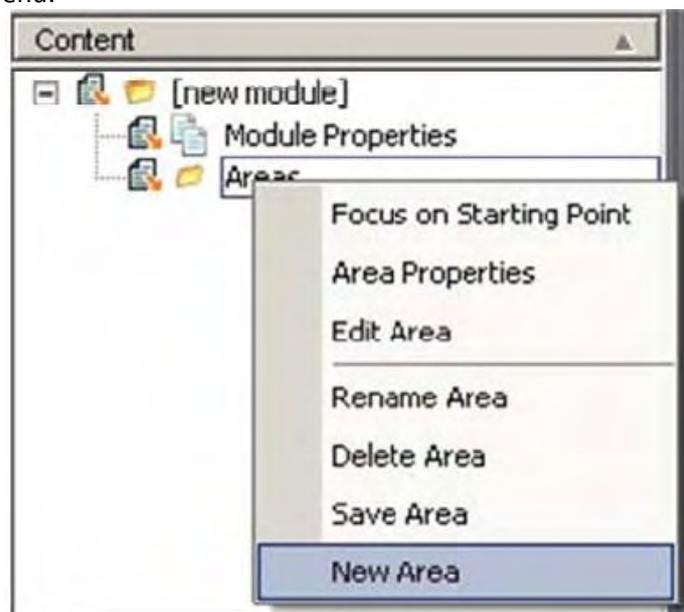
Der Inhalt des Fensters wird als Baum angezeigt. Im Moment ist das alles was du siehst, (ein neues Modul) und 2 Verzweigungen:

- **Module Properties.** Attribute setzen des Moduls.
- **Areas.** Enthält alle listen der Bereiche des Moduls, grade in diesem Augenblick existieren keine Bereiche deshalb ist die Liste leer.

Jetzt haben wir vor, die ersten Bereiche hinzufügen, um das zu können klicke mit der rechten Maustaste auf die Area Verzweigung:



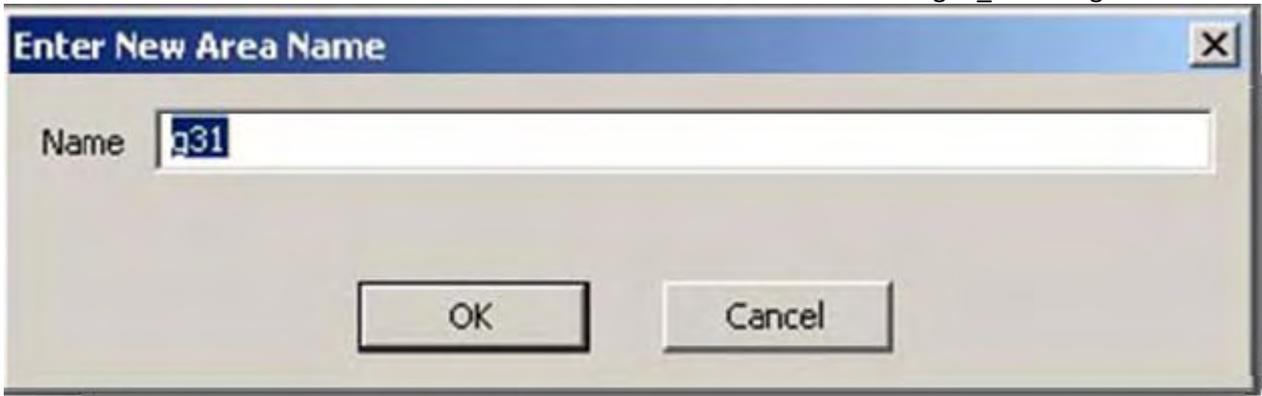
Wähle **new Area** vom Menü:



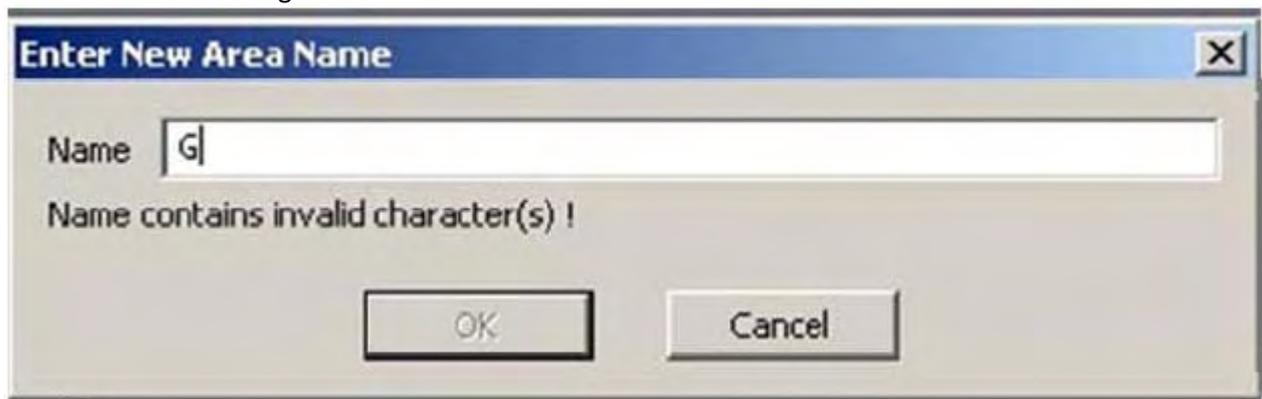
Dies öffnet ein Standard Windows Fenster das für den Ladeordner benutzt wird gehe zum Data\Meshes Ordner. Dieser Ordner enthält alle Bereiche die vom Witcher-Team geschaffen wurde. Jeder Bereich ist in den getrennten Ordnern gespeichert.

Öffne den Ordner der einen der Bereiche enthält, den du ausgewählt hast. Das Beste wär vielleicht die ausfindig gemachte Höhle.

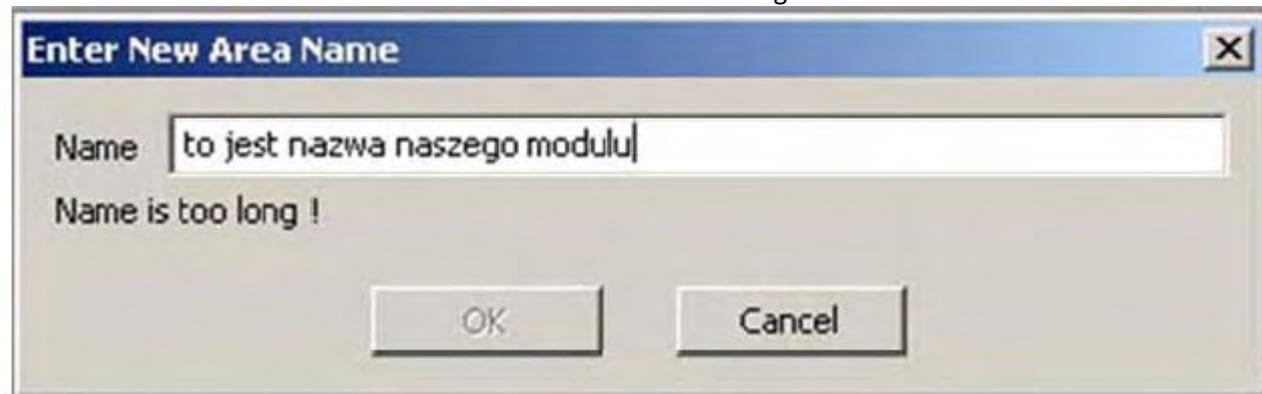
Im Ordner g31. Zu finden Öffne den Ordner, und wähle die g31.set Datei und klicke auf den geöffneten Button. Ein anderes fenster wird erscheinen. diesmal ist es ein Fenster wo du g31_cave eingibst:



Trage einen Namen für den Bereich hier ein, aber beachte: der Name kann nicht mit einem Großbuchstaben beginnen und Du kannst nicht-Englische Buchstaben oder andere Zeichen, wie @ # \$, benutzen! Wenn Du versuchst, einen Großbuchstaben oder irgendein anderes unzulässiges Zeichen einzugeben bekommst du diese Fehlermeldung:



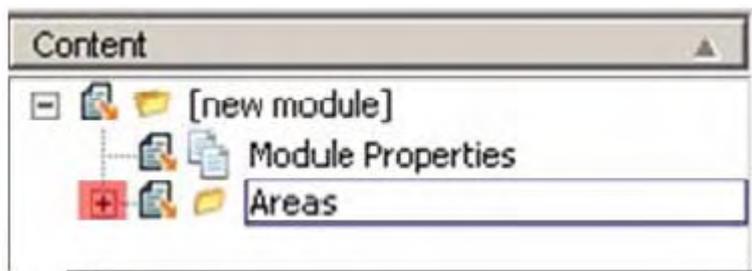
Gibst du mehr als 16 Zeichen ein erhältst du diese Fehlermeldung:



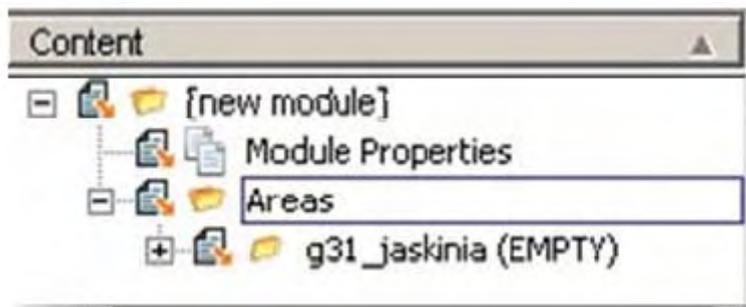
Und der Name muss mindestens eine Zeile erhalten sonst findest du deine Informationen nicht mehr:



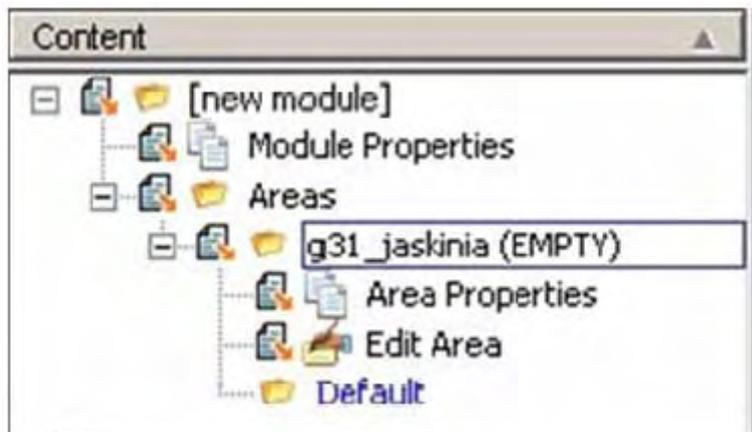
In einigen der obengenannten Fälle wirst du nicht im Stande sein, den OK, Knopf zu drücken, es wird unmöglich sein, ein neues Gebiet zu erstellen. Gib g31_cave als Name ein und Klick auf OK, Das Gebiet wird zu unserem Modul hinzugefügt, aber um es zu sehen, müssen wir den Bereichszweig im **Module-Explorer** öffnen, indem wir auf das kleine '+' Zeichen neben dem Namen klicken:



Du kannst jetzt sehen dass ein neuer Bereich geschaffen geworden ist. Die g31_Cave Verzweigung befindet sich jetzt unter dem Unterzweig des Bereiches:



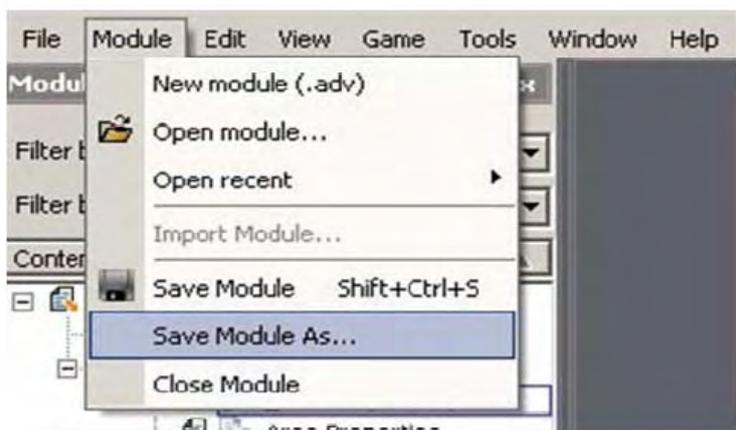
Öffne jetzt den Zweig der durch das klicken von '+' erhältst du einen neuen Bereich, Der nahe am unseren eingegeben Namensfeld ist. Jetzt haben wir eine volle Ansicht des Moduls und des Bereiches:



Im Moment enthält der neue Bereich drei Unterzweige:

- **Area Properties** – Attribute des Set Bereichs (bitte nicht zu verwechseln mit dem Modul Properties, das ist nämlich etwas völlig anderes)
- **Edit Area** – Editiert den betreffenden Bereich,
- **Default** – normale ebene des Bereichs.

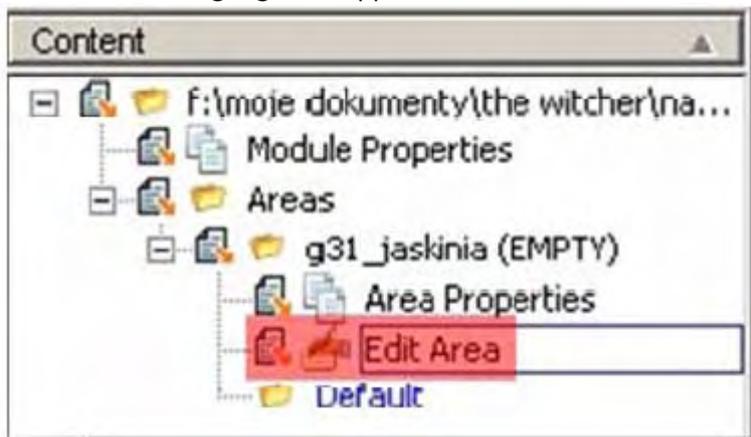
Wir sollten unser Modul jetzt speichern, bevor wir an etwas schwierigerem gehen. Um das tun zu gehen beim Modul unter **Save Module As...** im Menü:



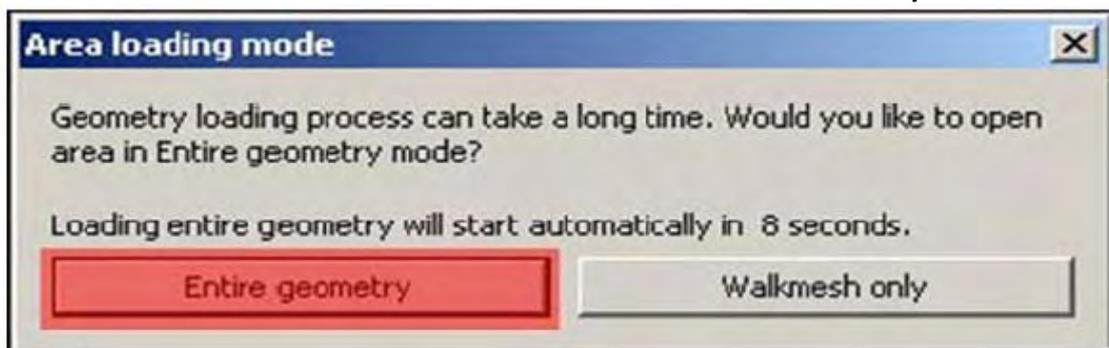
Dies öffnet ein Standard Windows-Fenster, das für den Speicherordner benutzt wird. Gehe zu Dokumenten und zu den Einstellungen \ Username \ meine Dokumente \ The Witcher, in denen die Module gespeichert wurden. Geb ein: **our_adventure**. Wenn du den Namen eingegeben hast, drücke auf Save.

Lass uns beginnen, unseren Bereich zu editieren!

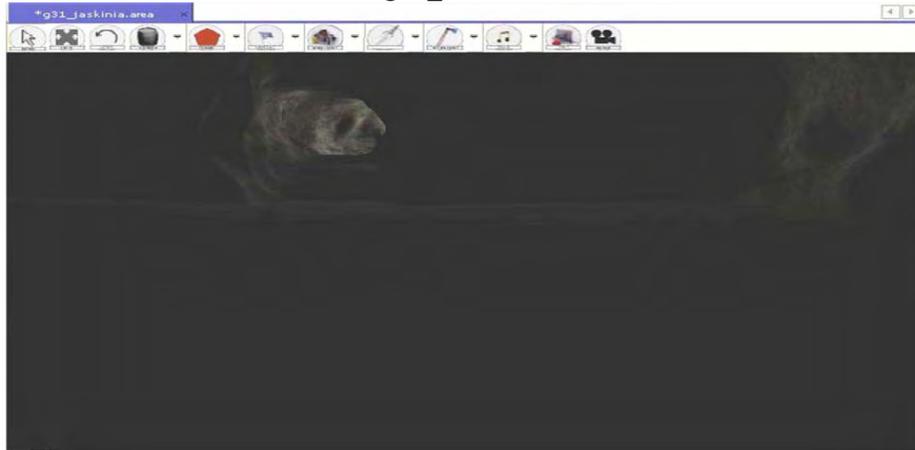
Du kannst einen Bereich in der Verzweigung mit Doppelklick editieren:



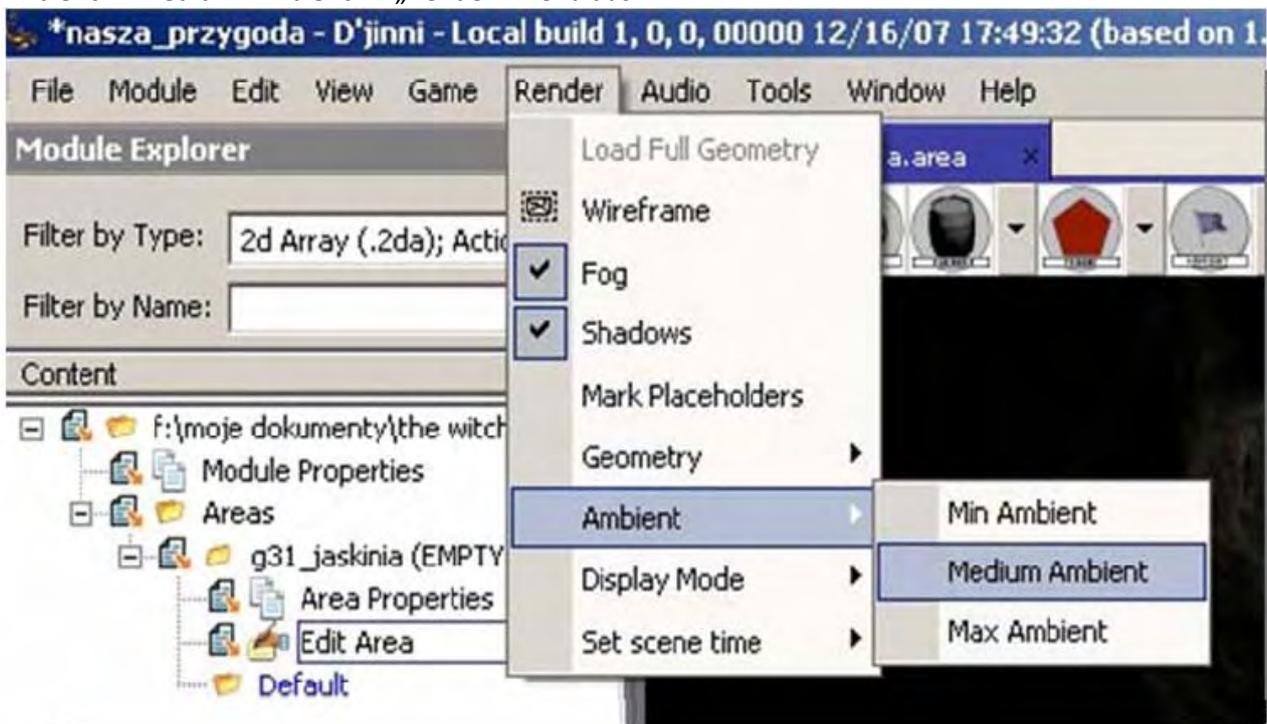
Bereich des Ladefensters erscheint auf dem Schirm. Klicke auf den **Entire Geometry** Button:



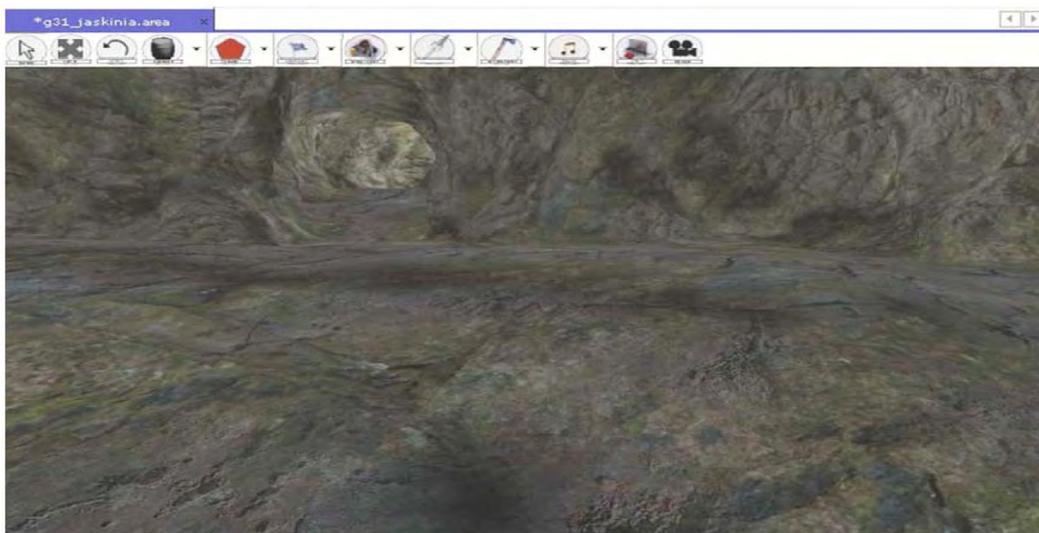
Wenn du 10 Sekunden nichts tust hat es dieselbe Wirkung wie als würdest du die ganze Zeit auf die Geometrie klicken. So oder so, erscheint ein neues g31_cave Fenster. Es enthält unser Gebiet:



Wenn der Bereich zu dunkel ist, wenn es geladen hat, kannst Du ihn erhellen. Um das zu tun, wähle unter **Ambient -> Medium Ambient** im „Render“ Menü aus:



Dies macht den Bereich sofort heller:



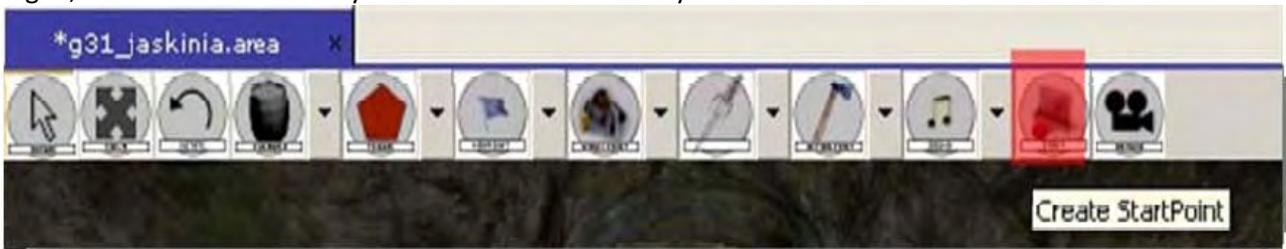
Selbstverständlich änderst Du die Beleuchtung des spielbaren Bereichs auf diese Weise nicht, es ist nützlich für das Editieren des Bereichs. Wenn Du dein Modul jetzt laufen lassen würdest, wär Deine Map in kompletter Dunkelheit getaucht. Wir werden etwas Licht dem Bereich später hinzufügen.

Jetzt Erkläre ich Dir, wie man sich in dem Bereich bewegen kann,

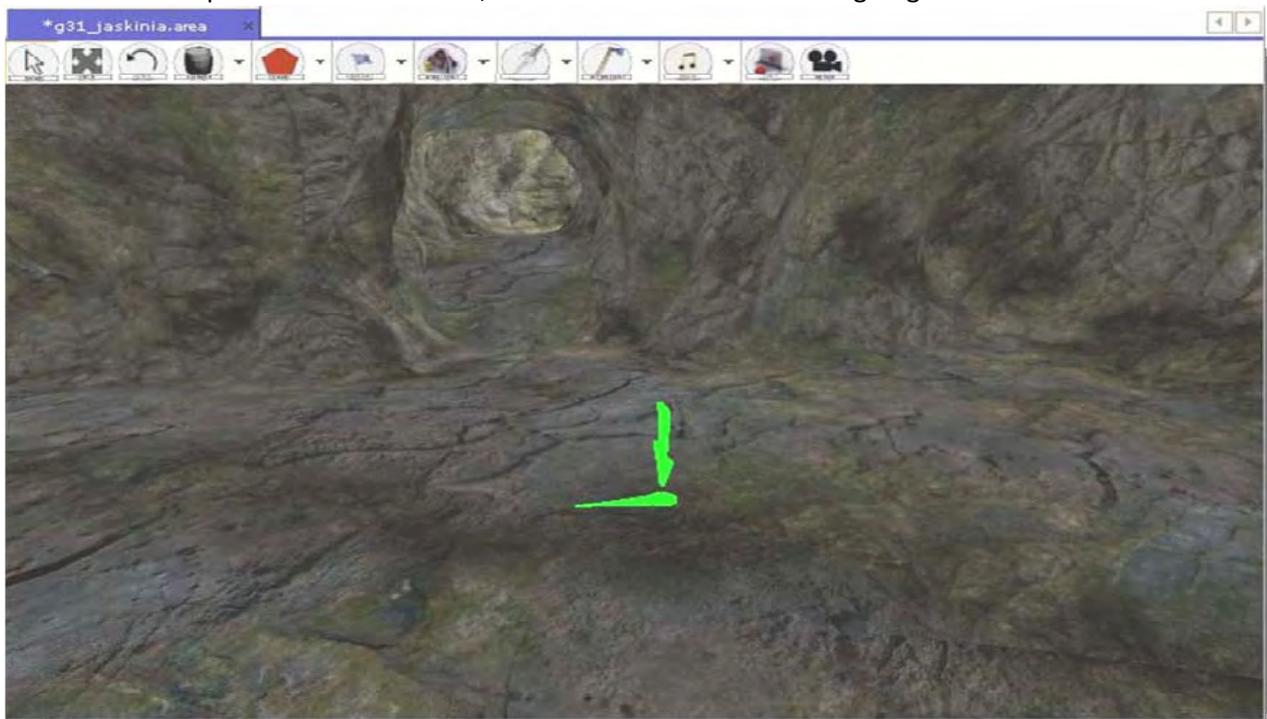
während Du:

- **die Pfeiltasten** verwendest du, um dich durch den Bereich zu bewegen,
- **die W A S D** Tasten verwendest du, um die Kamera zu bewegen,
- **Um sich zu drehen**, Klicke mit der rechten Maustaste, auf den Bereich und halte die Maus gedrückt und bewege die Maus in die gewünschte Richtung.

Jetzt sollten wir einen Anfangspunkt hinzufügen. Dies ist der Platz, wohin Geralt (der Spieler) erscheinen wird. Das heißt, es ist ein Platz in unserem Modul, in dem das Spiel beginnt. Um den Anfangspunkt hinzuzufügen, klicke auf das zweite Symbol von Rechts auf der Symbolleiste über dem editierten Bereich:



Ein Gegenstand, der den Anfangspunkt darstellt, wird am Cursor "gehaftet". Links-Klicke auf dem Punkt, in dem Du den Startpunkt setzen möchtest, Es wird dann im Bereich hinzugefügt:

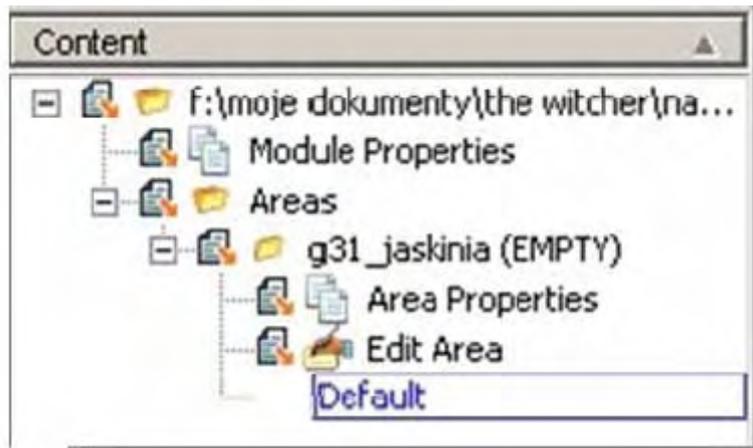


Der Anfangs-Punkt wird immer zur Default-Layer(Schicht) hinzugefügt. Jetzt ist eine gute Zeit dafür um zu erläutern was Layers sind, Stelle Dir ein riesengroßes Gebiet mit Hunderten von verschiedenen Gegenständen vor.

Wenn Du etwas in einem dieser Gegenstände ändern wolltest, kostet das suchen, viel Zeit und viel Anstrengung, wenn wir die Layers nicht hätten.

Wenn Du ein Gebiet erschaffen hast, kannst Du Layers hinzufügen und verschiedene Gegenstände auf verschiedenen Layers legen, z.B. Punkte auf einen Layer und Handlungspunkte auf einem anderen Erzeugen. Das folgende Layer könnte für Placeables verwendet werden (Stühle, Du kannst einen Layer verbergen, so verschwinden die Gegenstände, die darauf gelegt worden sind, aber nur wenn Du das Gebiet editierst. Sie werden noch im Spiel selbst anwesend sein.

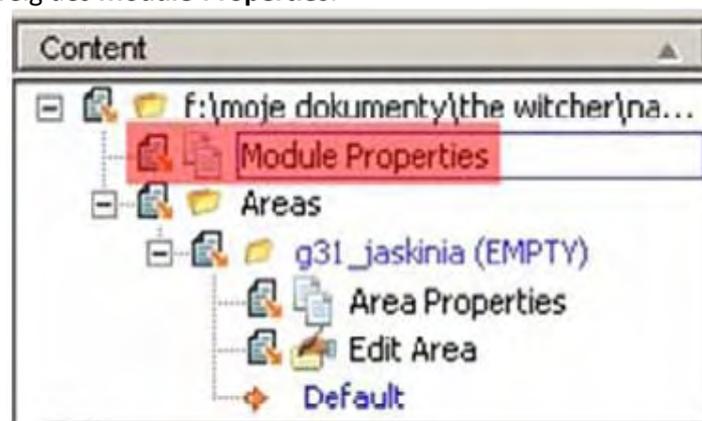
Du kannst auch eine Schicht einfrieren lassen, so dass Gegenstände darauf nicht geändert werden können (Sie werden nicht im Stande sein, sie zu löschen oder zu bewegen, oder ihre Eigenschaften zu ändern). Layer machen unsere Arbeit mit dem Editor etwas leichter. Jedes kürzlich geschaffene Gebiet hat einen Default-Layer:



Das ist genau, wo wir den Start-Punkt hinzugefügt haben. Erinner dich daran, dass es nicht wichtig ist, wie viele Layers es gibt in den Bereichen, oder welche von ihnen aktiv sind, der Start-Punkt wird im Normalfall immer einem Default Layer hinzugefügt.

Normalerweise ist unser Modul jetzt bereit um zu starten, aber das zu tun ist sinnlos- erstens wir haben kein Attributen-Set weder für das Modul noch für das Gebiet. Und zweitens hat auch unser gebiet keine Beleuchtung es wär einfach viel zu dunkel immerhin ist es noch eine Höhle, Bevor wir Beleuchtung hinzufügen, lass uns die Eigenschaften des Moduls einstellen.

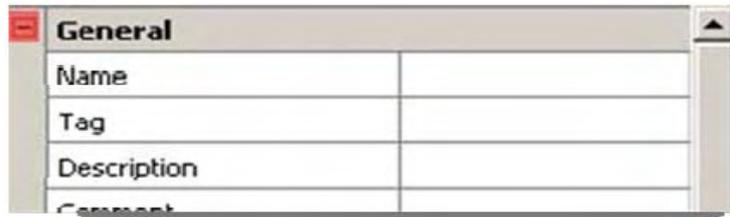
Doppelklicke auf den Zweig des **Module-Properties**:



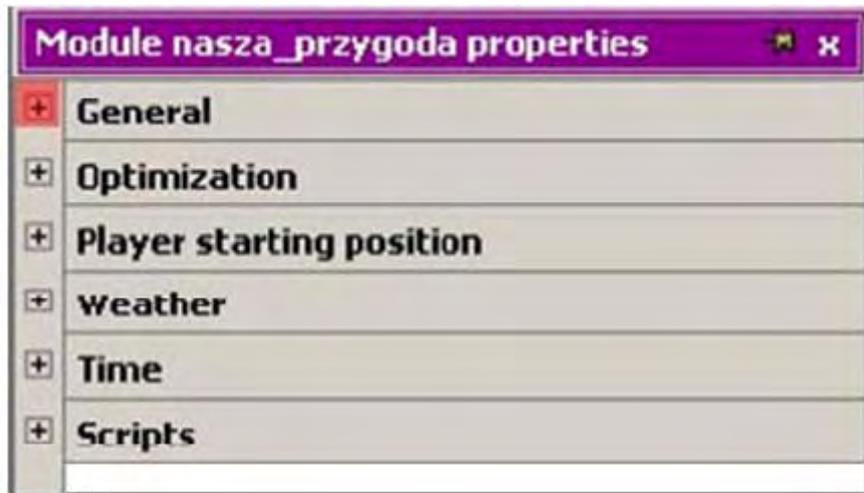
Das Modul für our_adventure Eigenschaften-Fenster wird auf der rechten Seite des Schirms erscheinen:



Das Fenster wird in zwei Spalten geteilt. Die linke enthält die Eigenschaften, die in folgenden Abschnitte unterteilt sind: Ausgangsposition des General, der Optimierung, des Spielers, Wetter, Zeit und des Skripts. Um zwischen die Abschnitte leicht zu schalten, kannst Du auf eines der kleinen '-' Zeichen klicken, die nahe bei den Abschnittsnamen sind:



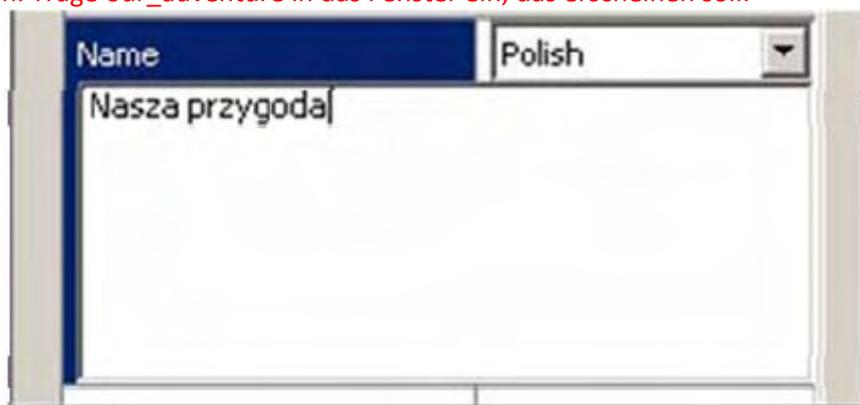
Wenn Du die vorher herabgesetzte Gruppe maximieren möchtest, klicke auf das kleine '+' Zeichen, das nahe bei den Namen dieses Abschnitts vorhanden ist:



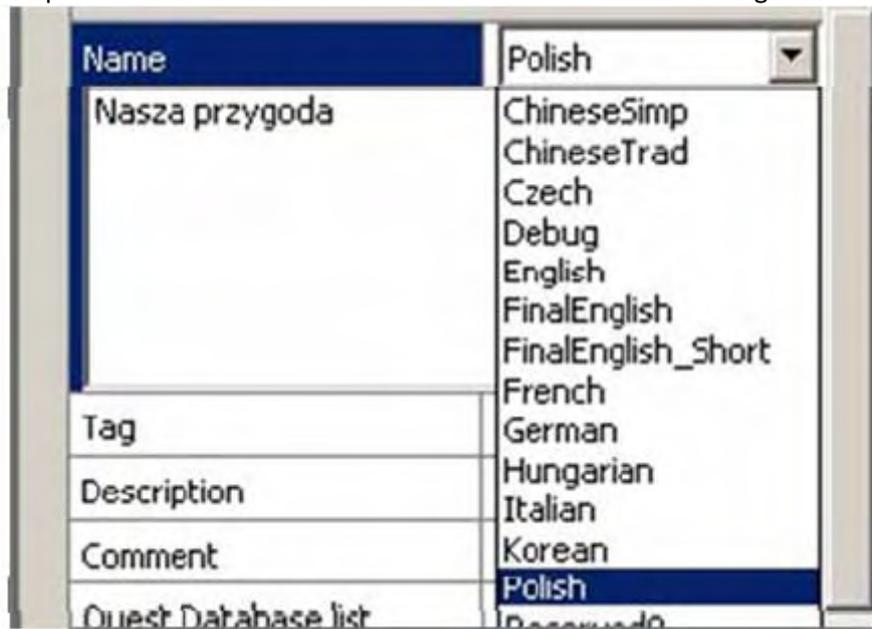
Die rechte Spalte enthält die Werte der jeweiligen Attribute. Die Liste ist ziemlich lang, so kannst Du auf und ab im der leiste scrollen:

Jetzt werden wir die jeweiligen Attribute einstellen. Alles wird nach und nach beschrieben. Allgemeiner Abschnitt:

- **Name.** Enthält den Namen des Moduls. Um den Namen zu speichern, klicke auf der rechte Spalte gegenüber vom Namen. Trage our_adventure in das Fenster ein, das erscheinen soll:



Du kannst auch die Sprachversion für den Namen wählen. Wähle das aus der folgenden Liste:



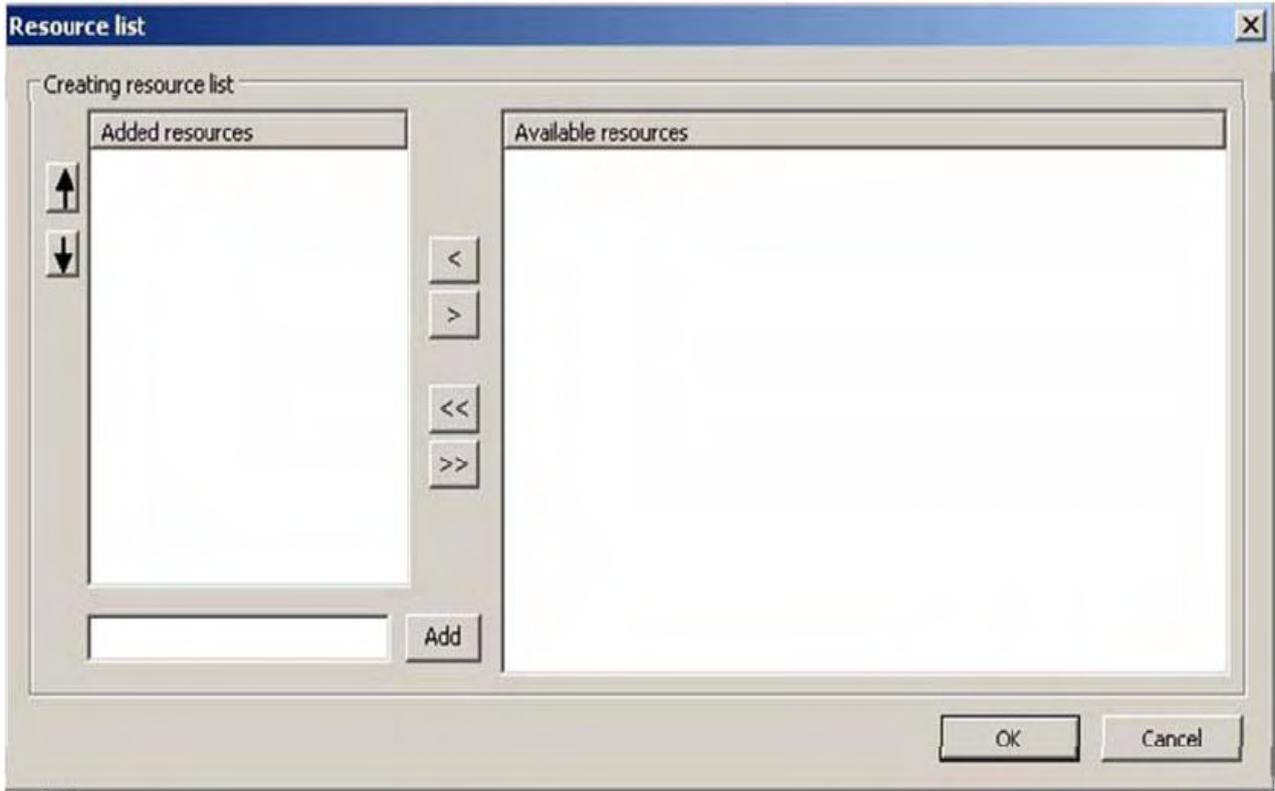
Wähle FinalEnglish_Short.

- **Tag.** kennzeichnet die Markierungsfahnen, die das angegebene Modul beschreibt. Es wird hauptsächlich über in den Script verwendet, die wir später im Handbuch besprechen werden.
Um auf den Text einzugehen, klicke auf die rechte Säule und gebe in Ihr Abenteuer ein.
 - **Description.** Nicht verwendet
 - **Comment.** Enthält einen Raum indem du eine Anmerkung zu schreiben kannst. Das wird im Spiel nicht verwendet, das ist ein Informationsfeld. Du kannst schreiben, dass Du, zum Beispiel Informationen über das Modul willst
 - **Quest Database List.** Teilt die eine Quest-Datenbank Modul zu. Das erstellen solche Datenbanken und das Zuweisen eines Moduls wird in einem anderen Kapitel beschrieben.
 - **Quest List.** Teilt dem Modul eine Quest zu, wie man eine Quest dem Modul zuteilt, wird in einem anderen Kapitel des Handbuches beschrieben
 - **Story NPC List.** Teilen einen Spawn set dem Modul zu. Das erstellen eines Spawn Set und die Zuteilung wird in einem anderen Kapitel des Handbuches beschrieben
 - **Spawnset list.** Teile einen allgemeinen Spawn-Charakter Set dem Modul zu. Das erstellen solcher Spawn sets und die Zuteilung wird in einem anderen Kapitel beschrieben
- Optimierung Abschnitt:**
- **Cache NSS scripts.** Weist die Skripte zu, die im Modul benutzt werden. Diese Skripte werden mit dem Modul geladen (jedoch nicht, dynamisch). Dank diesem läuft unser Spiel schneller, aber das Laden des Moduls kann länger dauern.
Die Skripte werden in einem anderen Kapitel des Handbuches beschrieben.

Um Skripte zuzuweisen, klicken auf das kleine gelben Icon, das auf der rechten Spalte gegenüber von den NSS Skripten platziert ist:



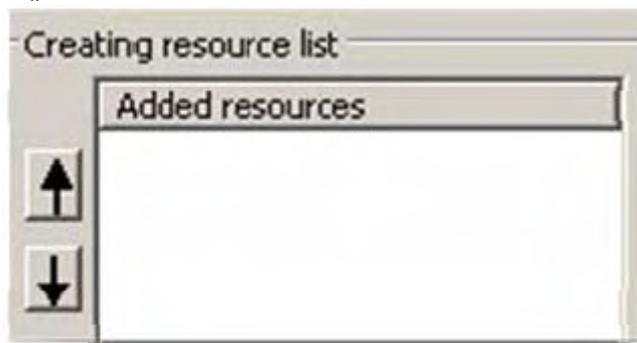
Dies öffnet das Ressourcen list Fenster:



Im Moment sind keine Skripte im "Available resources" Feld, da noch keine erstellt wurden. Aber falls Du ein existierendes Script zum "Cache NSS scripts Attribute" zuweisen willst, klicke auf den Namen im "Available Ressource" Feld und dann auf den "<" Button:

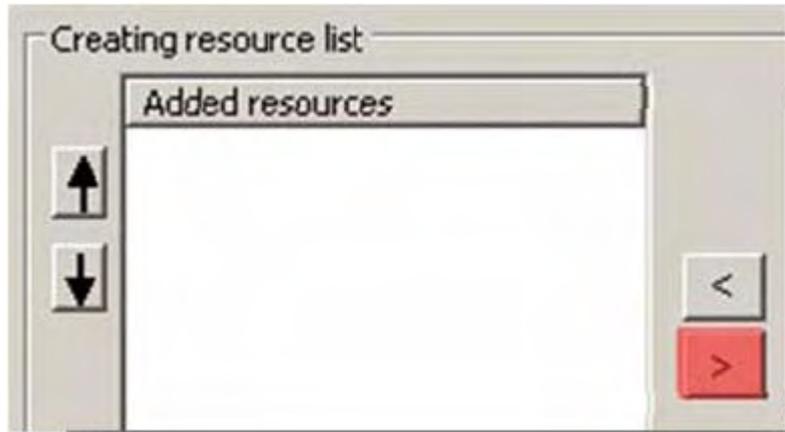


Das Script erscheint dann im „Added Ressource“ Feld:



Eine andere Möglichkeit ein Script hinzuzufügen ist ein Doppel-Klick auf seinen Namen im „Available resources“ Feld zu machen. Das Ergebnis ist das gleiche – das Script wird dem „Added Ressource“ Feld hinzugefügt.

Wenn Du das Skript wieder vom „Added Ressource“ Feld entfernen willst, Doppel-klick einfach darauf, oder benutze den „<“ Button:



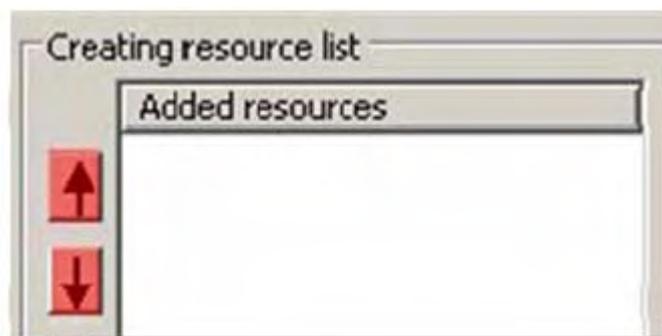
Das Skript wird aus dem „Cached NSS script Attribute“ entfernt. Wenn Du alle Skripte (angenommen da sind mehrere) dem Attribut zuweisen willst, klicke auf den „<<“ Button;



Eine ähnliche Aktion entfernt alle hinzugefügten Skripte. Um diese Aktion auszuführen, klicke einfach den „>>“ Button:



Wenn Du, aus irgendwelchen Gründen auch immer, die Reihenfolge der Skripte im „Added resources“ Feld ändern willst, markiere einfach das besagte Skript und klicke dann auf den „Pfeil nach oben“ oder „Pfeil nach unten“ Button:



Nachdem Du alle Skripte hinzugefügt hast, klicke auf den „OK“ Button im „Resource List“ Fenster.

Player starting position Section:

- **Entry Area.** Enthält die Area (Bereich) mit dem Startpunkt. Dies wird hier immer automatisch aktualisiert, sobald Du den Startpunkt änderst. Du hast hier keine Möglichkeit irgendwas zu ändern.
- **Entry position X, Entry position Y, Entry position Z.** Diese drei Attribute enthalten die X, Y, Z Koordinaten vom zugewiesenen Startpunkt in der Area. Diese Attribute werden ebenfalls automatisch aktualisiert, sobald du den Startpunkt änderst. Hier hast Du keine Möglichkeit etwas zu ändern.

Weather Section:

Dies ist der Wetter Eigenschaften Bereich um das Wetter in den Modulen zu bestimmen.

- **Minimum weather intensity:** Bestimmt die Intensität des Wetters (Regen oder Schnee) in den Modulen. Dies ist der kleinste Wert. Gebe hier die Zahl 0 ein. Um die Einstellung zu ändern, klicke auf die „Minimum weather intensity“ rechts gegenüber liegende Spalte und geb einen Wert ein.
- **Maximum weather intensity:** Bestimmt die Intensität des Wetters (Regen oder Schnee) in den Modulen. Dies ist der größte Wert. Geben hier die Zahl 0 ein. Um die Einstellung zu ändern, klicken auf die „Maximum weather intensity“ rechts gegenüber liegende Spalte und gebe einen Wert ein.
- **Chance of rain:** Bestimmt die Wahrscheinlichkeit vom Regen in den Modulen. 0 bedeutet gar keinen Regen und 100 bedeutet dass es nur noch regnet. Um die Einstellung zu ändern, klicke auf die „Chance of rain“ rechts gegenüber liegende Spalte und geben einen Wert ein.
- **Chance of snow:** Bestimmt die Wahrscheinlichkeit von Schnee in den Modulen. 0 bedeutet gar keinen Schnee und 100 bedeutet dass es nur noch schneit. Um die Einstellung zu ändern, klicken auf die „Chance of snow“ rechts gegenüber liegende Spalte und gebe einen Wert ein.

Time Section:

Dieser Bereich stellt die Zeit in den Modulen ein.

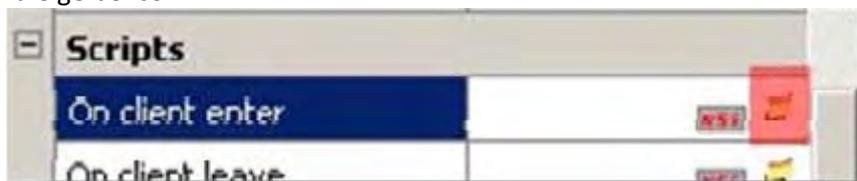
- **Minute per hör:** Bestimmt wie viele Minuten Echtzeit eine Stunde im Spiel betragen. Ist hier eine 4 eingetragen, bedeutet es das 4 Minuten Echtzeit eine Stunde im Spiel beträgt. Zum ändern dieses Wertes klicke in die rechts gegenüberliegende Spalte von „minutes per hour“ und gebe dort einen neuen Wert ein.
- **Start day:** Bestimmt an welchen Tag das Spiel (Modul) starten soll. 1 entspricht Montag und 7 Sonntag. Sie können hier nur 1-7 einstellen. Zum ändern des Wertes gehe auf die rechte gegenüberliegende Spalte und gebe dort einen neuen Wert ein.
- **Start hour:** Bestimmt die Zeit in der das Spiel starten soll. Es können hier nur Werte von 00 bis 23 eingestellt werden.
- **Start year:** Bestimmt das Jahr in dem Spiel. Du kannst hier einen Wert von 0 bis 99 eingeben. Wenn Du zum Beispiel 43 eingibst, bedeutet es das Jahr 1243.
- **Dawn hour:** Bestimme um wie viel Uhr die Sonne aufgehen soll. Du kannst hier einen Wert von 0-23 eingeben. Gebe eine 7 ein, so geht die Sonne um 7 Uhr früh auf.
- **Dusk hour:** Bestimmt wann die Sonne untergehen soll. Du kannst hier einen Wert von 0 bis 23 eingeben. Wenn du beispielsweise 20 eingibst, bedeutet es dass die Sonne um 8 Uhr Abends untergeht.

Script Section:

Diese Einstellungen sind dafür verantwortlich das die Skripte ausgeführt werden wenn ein Ereignis stattfindet. In anderen Kapiteln wird mehr über Scripting erzählt. Im Moment beschreibe ich hier die jeweiligen Einstellungen.

- **On client enter:** Skripte werden geladen wenn der Spieler das Modul betritt.
- **On client leave:** Skripte werden geladen wenn der Spieler das Modul verlässt.
- **On heartbeat:** Skripte werden alle 5 Sekunden geladen
- **On module loaded:** Skripte werden geladen wenn das Modul geladen wird.
- **On module started:** Skripte werden geladen wenn das Modul gestartet wird.
- **On player death:** Skripte werden geladen wenn der Spieler tot ist.
- **On player dying:** Skripte werden geladen wenn der Spieler am sterben ist.
- **On player level-up:** Skripte werden geladen wenn der Spieler einen Level aufsteigt.
- **On player rest:** Skripte werden geladen wenn der Spieler eine Rast (Meditation) einlegt.
- **On player respawn:** Skripte werden geladen wenn der Spieler wieder erscheint.
- **On user defined:** Skripte werden vom User definiert.

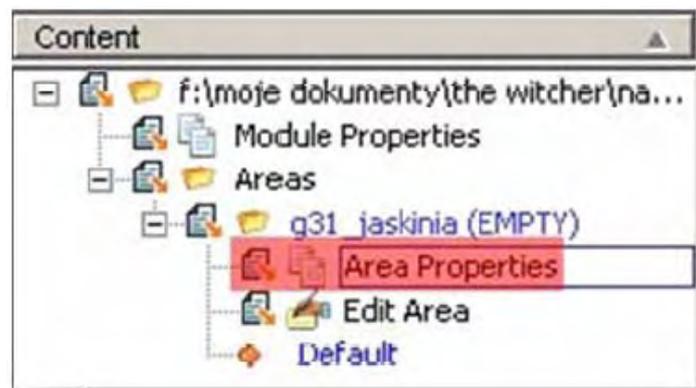
Wenn du ein Script einem Ereignis (Event) z.B. „On client enter“ zuweisen möchtest, klicke auf der rechten Seite auf das schmale gelbe Icon:



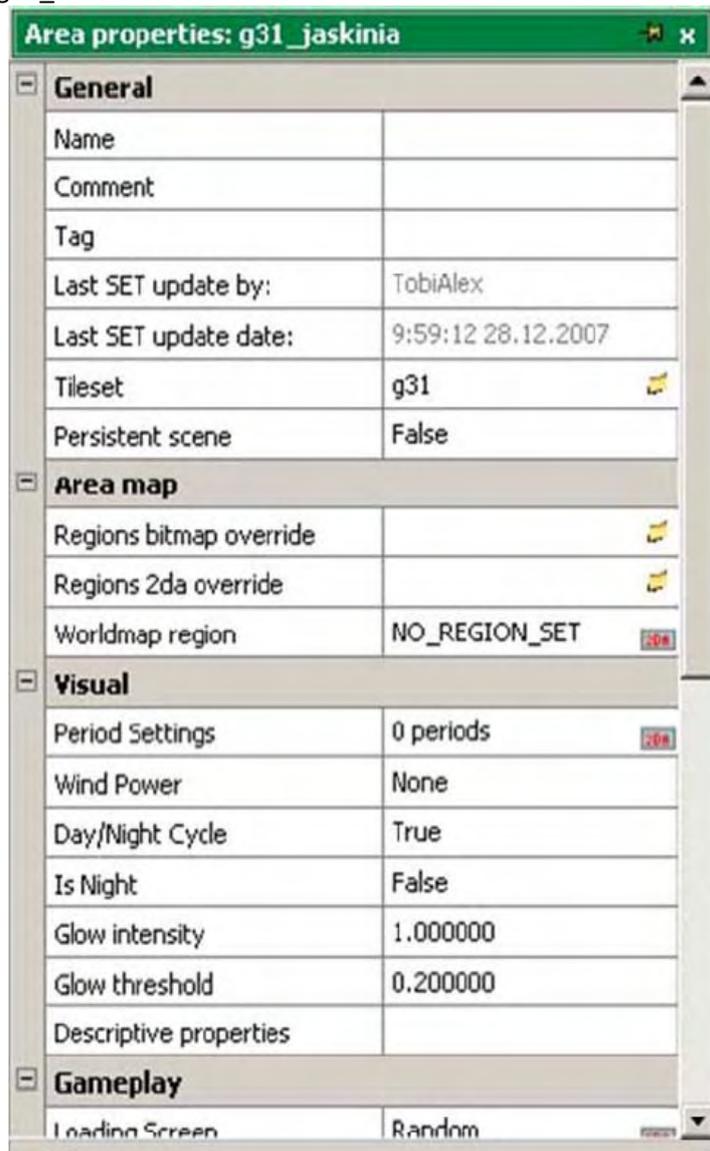
Es öffnet sich ein Windows Fenster zum laden von Dateien. Gehe nun zum Ordner „Data\Scripts“ (das ist der Ordner für sämtliche Skript Dateien). Suche das gewünschte Skript und klicke auf „öffnen“.

Das war es über die Modul Eigenschaften. Bitte denke daran, dass die Modul Eigenschaften für das ganze Modul gelten, während die Area Eigenschaften, wohin wir in ca. 1 Minute kommen ☹, nur für einen bestimmten Bereich (Area) im Modul gelten.

Um die Eigenschaften einer Area einzustellen, Doppel-klicke auf „Area Properties“ auf den Bereich den Du ändern willst:



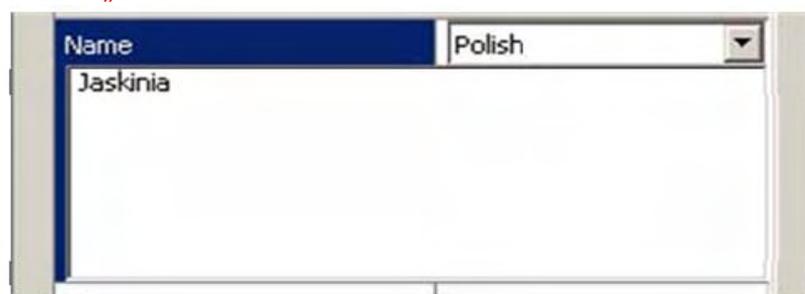
Die „Area properties“: g31_cave erscheint auf der rechten Seite vom Bildschirm in einem Fenster:



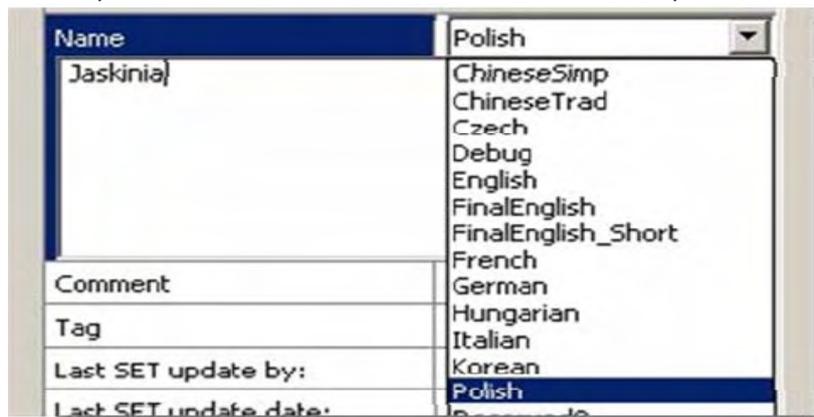
Das Fenster arbeitet genauso wie die vorher beschriebenen Fenster. Der einzige Unterschied besteht in den Bereichen und den Einstellungen die es beinhaltet. Nun werde ich die individuellen Einstellungen näher beschreiben und zeige Dir auch die passenden Einstellungen.

General Section:

- **Name:** Beinhaltet den Namen der Area. Zum Eingeben eines Namens, klicke auf die Namenseigenschaften auf der rechten Seite. Gebe „Cave“ ein:



Hier kannst Du auch eine Sprache für den Namen auswählen. Wähle eine Sprache aus der Liste:



Wähle „FinalEnglish_Short“

• **Comment:** Beinhaltet einen Bereich zum Schreiben eines Kommentars.

Das wird nicht vom Spiel benutzt, sondern es ist nur eine Information. Du kannst hier schreiben was auch immer Du willst, z.B. eine Info über den Bereich.

• **Tag:** Bestimmt die Punkte zum Beschreiben des ausgewählten Bereichs.

Es wird hauptsächlich mit Skripte benutzt, welche wir später im Handbuch beschreiben werden. Um den Text zu ändern, klicke in der rechts gegenüberliegenden Spalte von Tag und gebe „over_cave“ ein.

• **Last SET update by:** Beinhaltet Informationen über die Person die zuletzt die Area bearbeitet hat. Der Name wird direkt vom Windows Anmeldungskonto übernommen. Diese Eigenschaft kann nicht verändert werden. Diese Information aktualisiert sich automatisch.

• **Last SET update date:** Beinhaltet Zeit und Datum von der letzten Veränderung. Diese Eigenschaft kann nicht verändert werden. Diese Information aktualisiert sich automatisch.

• **Tileset:** Diese Eigenschaft beinhaltet den Namen des „meshes“ (3D Objekt) der Area. Du brauchst es nicht zu ändern, da es sich automatisch aktualisiert, wenn die Area erstellt wird. Wenn Du das Objekt ändern willst, z.B. Du willst eine andere Höhle als diese benutzen, klicke auf das schmale gelbe Icon auf der gegenüber liegenden Spalte:



Ein Windows Fenster zum Laden von Dateien öffnet sich. Gehe zu „Data\Meshes“. Das ist der Ordner der alle Areas, die vom TW Team erstellt wurden beinhaltet. Doppel-klicke auf den Ordner mit der Area den Du nutzen willst und Markieren die „.set“ Datei und klicken dann auf „öffnen“.

• **Persistent scene:** Wird nicht verwendet

Area Map Section:

• **Region bitmap override:** Beinhaltet ein Bild von der Area.

Mit dem D’jinni Editor und den Areas die vom TW-Team erstellt wurden, hast üblicherweise Bilder von den Areas. Die Bilder dienen als Karten die erscheinen wenn man im Spiel die Taste „m“ drückt. Um ein Bild zuweisen zu können, klicke auf das schmale gelbe Icon in der rechten Spalte:



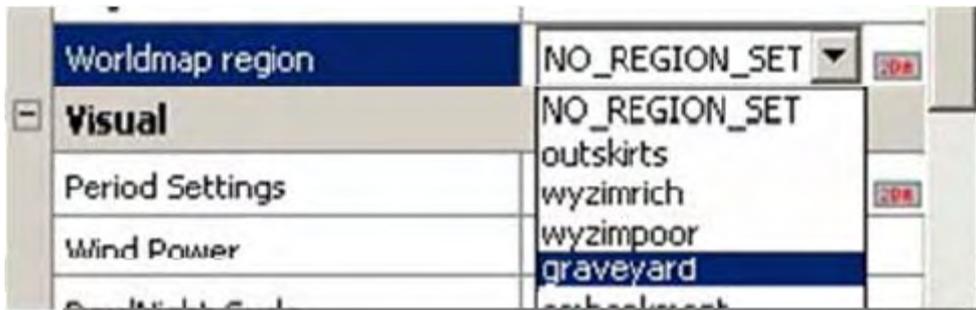
Es öffnet sich ein Windows Fenster zum laden von Dateien. Gehe zu „Data\Meshes\g31“ (das ist der Ordner der die Höhle beinhaltet die wir benutzen wollen), markiere die „regg31_m1.bmp“ und klicken dann auf „öffnen“.

- **Regions 2data override:** Beinhaltet die „.2da“ Dateien von unserer Area. „.2da“ Dateien sind nichts anderes als einfache Datenbanken. Mehr über diese Dateien wirst du in einem anderen Kapitel vom Handbuch lernen. Alle Areas vom TW-Team besitzen solche „.2da“ Dateien. Willst du so eine Datei eintragen, so klicke auf das schmale gelbe Icon in der rechten Spalte, gegenüber von „Regions 2data override“:



Es öffnet sich wieder ein Windows Fenster zum laden von Dateien. Gehe wieder in den Ordner „Data\Meshes\g31“ und wähle die „regg31_m1.2da“ Datei aus. Klicke auf „öffnen“.

- **Worldmap region:** Bestimmt die Position unserer Area in der gesamten Spielwelt. Eine Karte in TW besteht aus zwei Elementen: die Karte der Area in der sich Geralt (der Spieler) sich aktuell befindet; und die Karte der Region in der sich die momentane Area befindet. Mit dem benutzen der „Region bitmap override“ und „Region 2data override“ Eigenschaft haben wie die Karte von unserer Area (einer Höhle) definiert. Jetzt müssen wir festlegen wo sich die Höhle befindet. Wir tun dies in dem wir die „Worldmap region Attribute“ Eigenschaft verwenden. Wähle eine Region aus der Liste. Um die Einstellung zu ändern, klicke rechts neben die „Worldmap region“ Eigenschaft. Wähle, zum Beispiel die „graveyard“ region. Das bedeutet dass unsere Höhle auf dem Friedhof ist:



Wenn Du das schmale rote Icon der Auswahlliste anklickst, öffnet sich die „reg_worldmap.2da“ Datei. Dies ist die Datenbank die alle Regionen von TW beinhaltet:

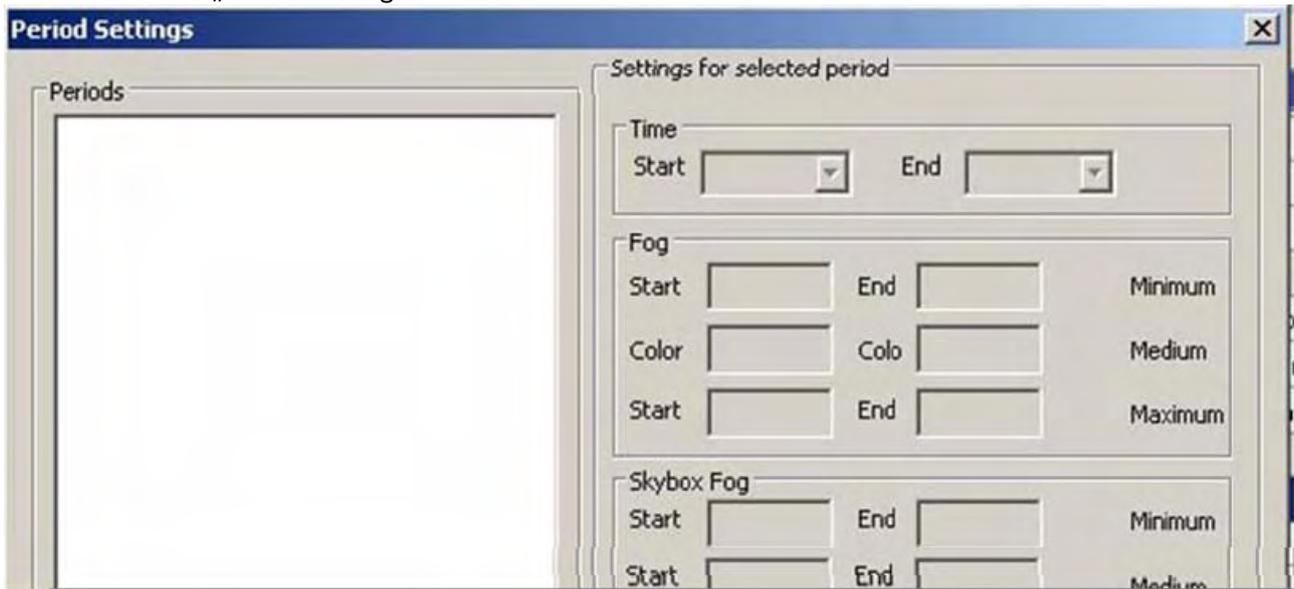


Visual Section:

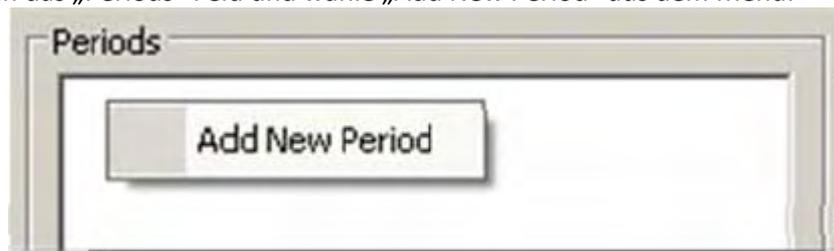
- **Period Settings:** Beinhaltet die Definitionen des Tag – Nacht Zyklus. Es ist einleuchtend das eine morgendliche Area anders aussieht als eine in der gerade Mittag, Abend oder Nacht ist. Das ist nicht weiter wichtig für uns, da unsere Area eine Höhle ist – es ist so zu sagen eine „Mono-Atmosphäre“. Ich werde Dir erzählen wie Du die Zyklen einstellst. Willst Du die Einstellungen ändern, klicke auf das schmale rote Icon in der rechten Spalte gegenüber von „Periods Settings“:



Es öffnet sich die „Period Settings“ Fenster:



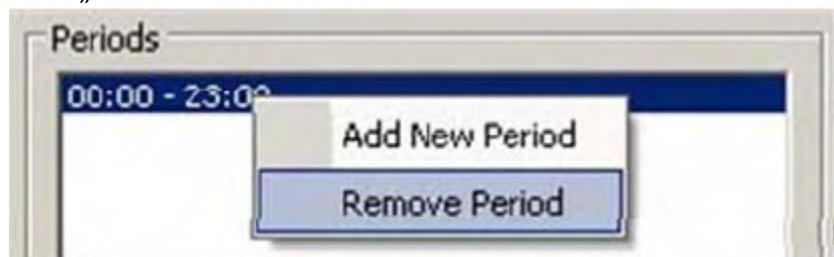
Das Fenster ist eingeteilt in zwei Bereiche: „Periods“, beinhaltet die definierten Zyklen; und „Settings for selected period“, beinhaltet die Einstellungen der gewählten Zyklen. Als erstes müssen wir einen Zyklus definieren. Klicke in das „Periods“ Feld und wähle „Add New Period“ aus dem Menü:



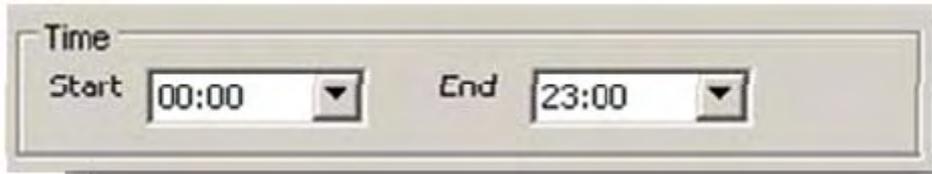
Ein neuer Zyklus erscheint im „Periods“ Feld:



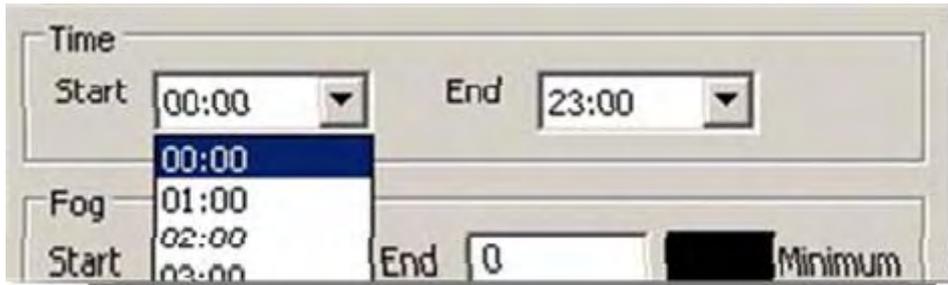
Die „Default“ Einstellung von den Zyklen ist von 00:00 bis 23:00, also 24 Stunden. Sicher, Du kannst das ändern und ich werde dir das in einer Minute erzählen. Wenn Du einen Zyklus löschen willst, rechtsklicke einfach drauf und wähle „Remove Period“ aus dem Menü:



Du kannst bis zu 24 Zyklen definieren, das währen jeweils einer pro Stunde. Es ist natürlich witzlos sowas zu tun. Es reicht vollkommen aus einen Zyklus für den Tag und einen für die Nacht zu nehmen. Im Falle unserer Höhle reicht einer. Gehe zum „Setting for selected period“ Feld. Als erstes setzen wir die Zeit für den Zyklus. Benutze dazu das „Time“ Feld:



Du trägst erst die Start Uhrzeit ein und dann die Endzeit. Wähle aus der Liste:



Für unsere Höhle brauchen wir nur einen Zyklus. Das nächste Feld ist „Fog“:



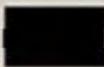
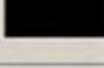
Dieses Feld ist für den Nebel in der Area zuständig. Was hat es mit dem Nebel auf sich? Erstens, es verändert die Szenarie – eine Area gepaart mit feinem Nebel schaut interessanter aus als ohne. Zweitens, es verstärkt die Wirkung unserer Area (höhle), weil weit entfernte Objekte, versteckt im Nebel, nicht sichtbar sind und nicht angezeigt werden. Eine großartige Möglichkeit für Außenbereiche.

Im „Fog“ Feld wähle die Entfernung von Geralt (dem Spieler) in der der Nebel beginnen soll. Setze auch die Farbe vom Nebel fest. Beachte auch das die Einstellungen in drei Teile aufgeteilt sind: Minimum, Medium und Maximum.

Zum verdeutlichen wie es funktioniert benutze ich ein anderes Beispiel.

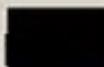
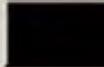
Lass uns annehmen wir haben einen Zyklus von 21:00 bis 07:00 Uhr definiert. Wäre nun keine Unterteilung von Minimum, Medium und Maximum, hätte der Nebel immer die gleiche Farbe. Es einfach nicht echt aussehen, besonders Anbetracht das es um 21:00 Uhr dunkel und um Mitternacht noch dunkler wird, während es gegen 05:00 Uhr wieder heller wird. Daher gibt es diese Einteilung in drei Kategorien. Um 21:00 Uhr soll der Nebel die Farbe in „Minimum“ um 01:00 Uhr die Farbe haben die in „Medium“ festgelegt ist und um 07:00 Uhr soll der Nebel die Farbe haben die unter „Maximum“ eingestellt ist. Die Farbe wechselt so sanft das es dem Spieler nicht auffallen wird. Alle drei Kategorien werden in unseren Zyklus den gleichen Wert haben. Gebe den Startwert 20 in allen drei Kategorien ein:

Fog

Start	20	End	0		Minimum
Color	20	Color	0		Medium
Start	20	End	0		Maximum

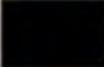
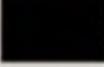
Als End wert geb 70 ein:

Fog

Start	20	End	70		Minimum
Color	20	Color	70		Medium
Start	20	End	70		Maximum

Das bedeutet dass der Nebel 20 Einheiten vor dem Spieler beginnt und 70 Einheiten nach dem Spieler endet. Lass uns nun die Farbe für den Nebel festlegen. Klicke auf das schwarze Feld neben „Minimum“:

Fog

Start	20	End	70		Minimum
Color	20	Color	70		Medium
Start	20	End	70		Maximum

Es öffnet sich ein Windows Fenster zum definieren von Farben:

Kolor [?] [X]

Kolory podstawowe:

Kolory niestandardowe:

Definiuj kolory niestandardowe >>

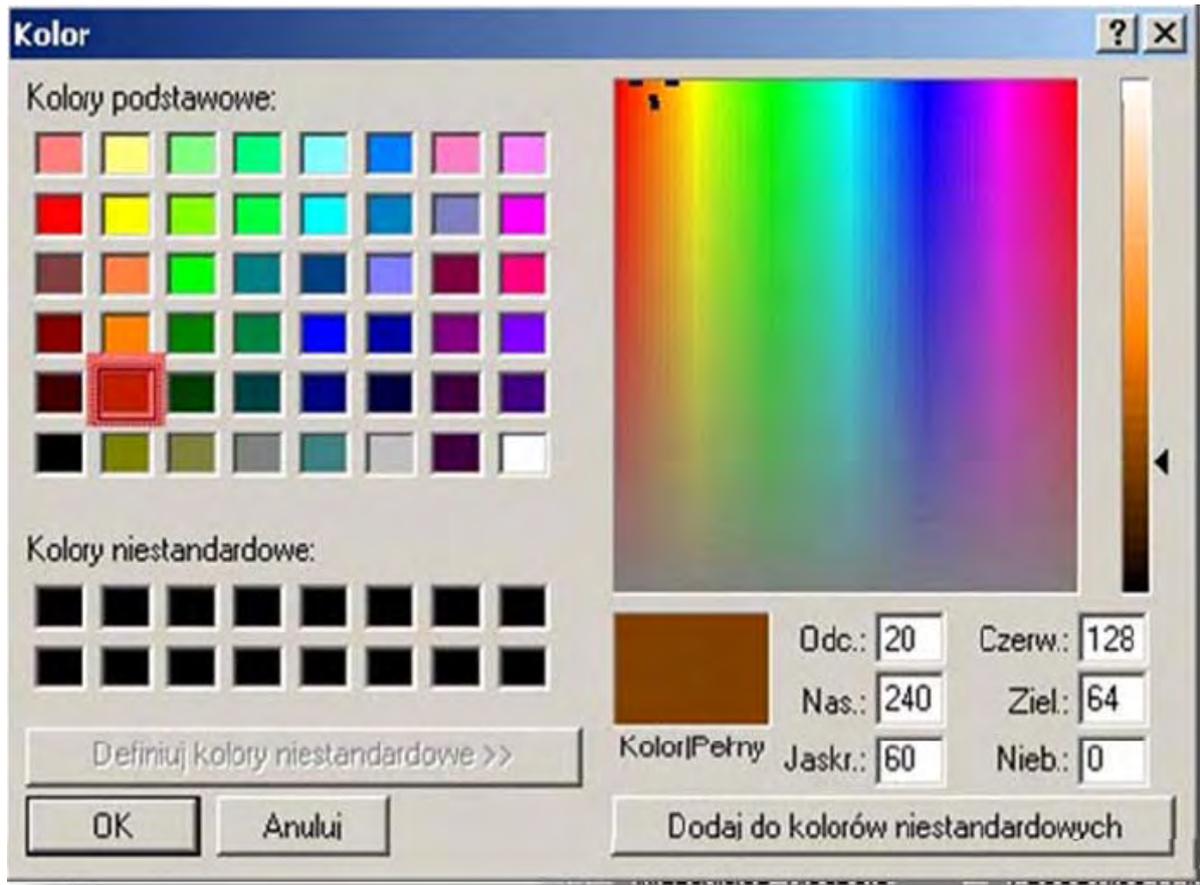
OK Anuluj

Dodaj do kolorów niestandardowych

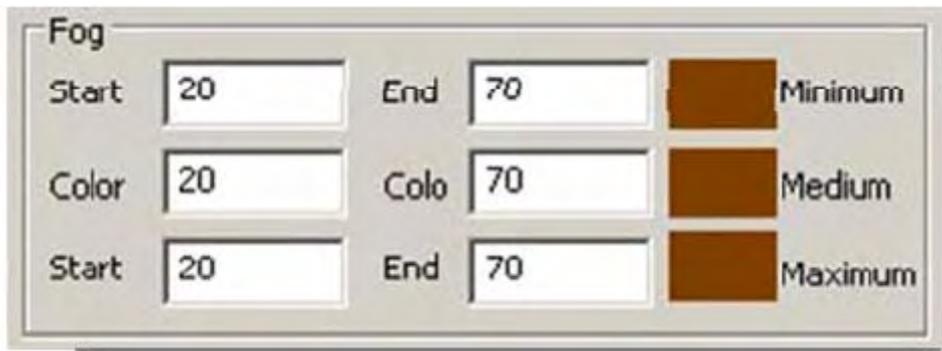
Kolor|Pełny

Odc.:	160	Czerw.:	0
Nas.:	0	Ziel.:	0
Jaskr.:	0	Nieb.:	0

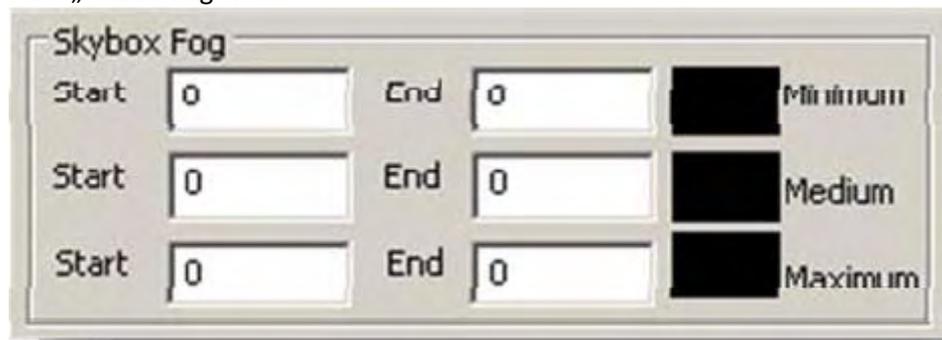
Wähle die braune Farbe:



Und klicken auf „Ok“. Wiederhole den Vorgang für die beiden anderen Kategorien „Medium“ Und „Maximum“:



Das nächste ist das „Skibob Fog“ Feld:



Dies ist für dieselben Sachen verantwortlich wie das „Fog“ Feld. Der einzige Unterschied ist das der Nebel für den Himmel gilt. Da aber unser Bereich eine Höhle ist, brauchen wir hier nichts einstellen. Falls Du einen offenen Bereich kreieren möchtest, kannst Du das Feld genauso benutzen wie das „Fog“ Feld. Das nächste Feld ist das „Clip plane“:



Dieses Feld ist hauptsächlich für das Ausblenden von Objekten zuständig. Entfernte Objekte sind nicht vom Spieler sichtbar und werden auch nicht angezeigt.

Um die optimale Distanz herauszufinden, musst Du damit eine wenig experimentieren. Starte die Module und schau nach wie es im Spiel aussieht und wirkt. Wenn du die Distanz zum Erscheinen zu nahe gemacht hast, ist das Ergebnis das die Objekte sofort dem Spieler angezeigt werden, ist die Distanz zu weit, wird das Modul sehr langsam da das Spiel auch weit entfernte Objekte darstellt.

Das „Clip plane“ Feld beinhaltet auch drei Kategorien: Minimum, Medium und Maximum. Damit kannst Du einen effektvollen Bereich erschaffen in dem die Distanzen um ein Objekt zu sehen näher ist bei Nacht. Wenn es also dunkel ist, sind entfernte Objekt für den Spieler nicht zu sehen.

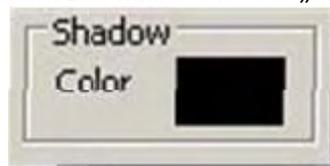
Ich empfehle die Startdistanz auf 0.1 zu setzen und die weiteste Distanz auf 200. Nehme diese Einstellung für alle drei Kategorien:



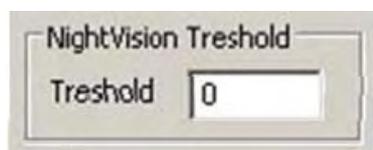
Das nächste Feld ist das „Light Mod.“:



Dieses Feld ist verantwortlich für die Lichteinstellung in der Area. Das Farbenfeld beinhaltet die Farbe wohingegen das „Faktor“ Feld den Multiplikator enthält. Die Lichteinstellung ist für außerhalb gedacht. In unserem Fall müssen wir hier nichts einstellen. Das nächste ist das „Shadow“ Feld:

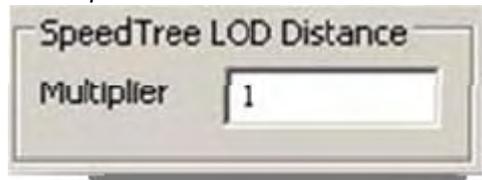


Dieses Feld bestimmt die Farbe der Schatten die von Objekten in der Area geworfen werden. Schwarz ist Standard, also können wir weitergehen. Alle Schatten, besonders in Höhlen, sind schwarz. Das nächste Feld ist das „Night Vision Threshold“ Feld:



Dieses Feld erlaubt Ihnen, den Schwellenabstand einzustellen, an dem der Spieler beginnt, um Gegenstände in der Dunkelheit zu erzeugen. Dies funktioniert, sofern der Spieler einen Nachtsicht Trank trinkt. Danach kommt das

SpeedTree LOD Abstandsfeld:



Dieses Feld wird ausschließlich für die Außenseite der Bäume verwendet. Hier kontrolliert der Multiplikator den Detaillierungsgrad von den Bäumen. Je Niedriger der Multiplikator, desto Realer sehen die Bäume aus, aber um so tiefer ist Leistungsfähigkeit des Gebiets. Ich empfehle Dir, dass du den Wert des Multiplikators steigerst, wenn Dein Bereich viele Bäume enthält, z.B in den Wäldern.

Danach kommt das *Gras*abstandsfeld:



Dieses Feld wird nur für die Gras bedeckten Außenseiten verwendet. Dieses Verwelkte Feld enthält die Entfernung vom Spieler, an dem das Gras beginnt zu verblassen. Wenn du das Feld zu hoch einstellen kann es die Leistungsfähigkeit des Bereiches verringern: wenn du es zu niedrig einstellst wird das Gras unter ihren Füßen Unrealistisch Das *Sichtbarkeitsfeld* enthält den Abstand, an dem der Spieler das Gras sehen kann Jetzt kommt das *Farbfilter*feld:



Hier setzt Du den Filter für die Farben. Das *Farbfeld* enthält eine Farbe, und der *Faktor* ist sein Multiplikator. , wenn es zu den Zykluseinrichten kommt. Klicke auf O.K. im Perioden Einstellung Fenster.

- **Wind Power** Bestimmt die Windgeschwindigkeiten in diesem Bereich. Wähle die Attribute von dieser Liste. Weil unser Bereich eine Höhle ist, setzen wir die Dinge und imitieren es in einem Wechsel. Um das zu tun, klicke auf die rechte Spalte gegenüber von den Wind Power Attributen und wähle Breeze von dieser Liste:



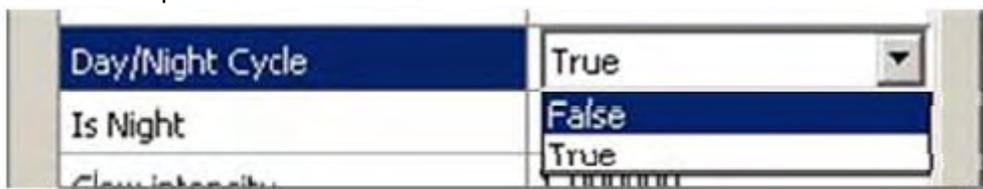
Die Werte, die Du diesen Attributen zuweisen kannst, sind:

- **None** - kein Wind
- **Breeze** – ein kleines bisschen Windiger
- **Light** – leichte Windgeschwindigkeiten
- **Medium** – mittlere Windgeschwindigkeiten
- **Heavy** – Starke Windgeschwindigkeiten
- **Very heavy** – wehende Windgeschwindigkeiten

Es ist wert das, zu erwähnen, dass Objekte in diesem Bereich mit einer zugeteilten Physik vom Wind weg-geblasen werden können

- **Day/Night Cycle.** Bestimmt, ob ein Bereich ein Tag-/Nachtzyklus hat. Wählen das Attribut von Dieser Liste. Weil unser Bereich eine Höhle ist, können wir den Zyklus ausschalten. Um das zu tun, klicken auf den Tag-/Nachtzyklus.

Zufinden in der rechten Spalte und wähle von dieser Liste False:



Die Werte, die du in diesem Attribut zuweisen kannst, sind:

- False – es gibt keinen Tag/Nacht Zyklus im diesen Bereich.
- True – es gibt einen Tag/Nacht Zyklus im diesem Bereich.

- **Is Night.** Stellt fest, ob dieser Bereich die Zeiträume der Nacht umfasst. Wähle dieses Attribut von der Liste. Weil es keinen Nachtbereich in diesem Bereich gibt, wähle false als Attribut von dieser Liste. Um das tun, klicke die rechte Spalte an und wähle False von dieser Liste:



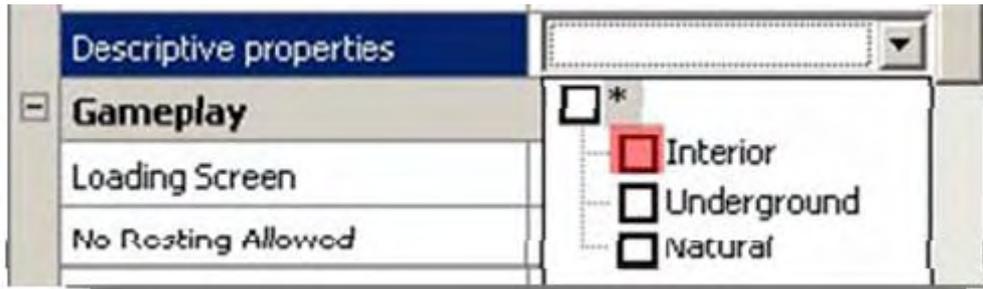
Die Werte, die Du in diesem Attribut zuweisen kannst, sind:

- False – Schließt nicht die Nacht in diese Periode mit ein
- True – schließt Periode der Nacht mit ein.

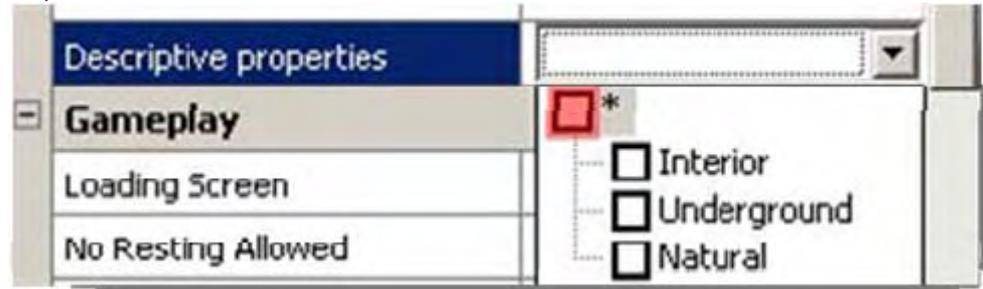
- **Glow intensity.** Bestimmt die Intensität der Glüh-Wirkung im diesen Gebiet. gebe den Wert 1 hier ein. Um dies zu tun, klicke auf die rechte Spalte gegenüber der Glüh Intensitätsattribute.

- **Glow threshold.** Stellt den Abstand vom Spieler fest, an dem der Glüheffekt beginnt in diesem Bereich. Gebe 0.2 ein. Um die Einstellung zu ändern, klicke auf die rechte Spalte gegenüber den Glühschwellenattributen.

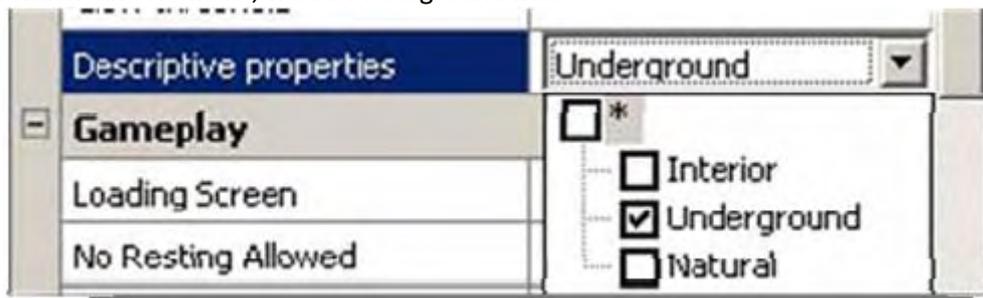
• **Descriptive properties.** Bestimmt die Beschreibung unserem Gebietes. Wähle und klicke von dieser Liste, die passende Auswahl an. Um das zu tun, klicke auf die rechte Spalte und mache einen Haken neben den kleinen Kasten neben der Auswahl Deiner Wahl:



Wenn Du alle Optionen wählen willst, klicke auf den Kasten neben dem Sternchen:



Weil unser Bereich eine Höhle ist, wähle Underground aus:

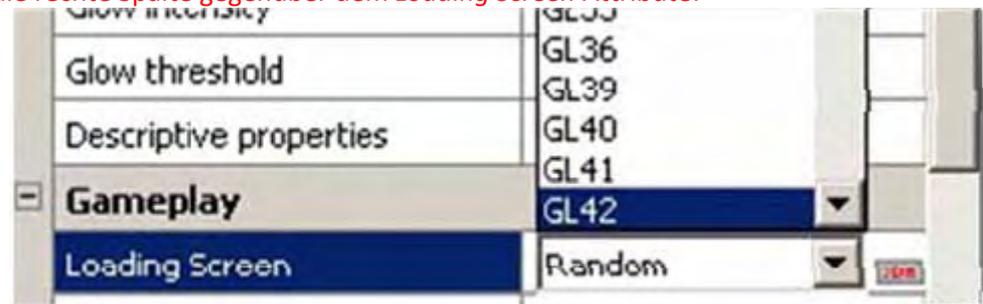


Die Werte, die Du in diesem Attribut zuweisen kannst, sind:

- *Interior* – dieser Bereich ist das Innere von einem Gebäude usw.
- *Underground* – dieser Bereich ist eine Höhle usw.
- *Natural* – dieser Bereich ist draußen, z.B. Wälder, Sumpf oder die Straßen von einer Stadt.

Gameplay Section:

• **Loading Screen.** Bestimmt, welches Bild erscheinen wird, wenn das Gebiet geladen wurde. Um das zu tun, klicke auf die rechte Spalte gegenüber dem *Loading Screen* Attribute:



Diese Liste enthält alle Bilder, die im Witcher Spiel vorhanden sind. Wähle *Choose Random* von dieser Liste. Das heißt, dass während des Ladevorgangs ein zufälliges Bild angezeigt wird. Klicke auf das kleine rote Icon nahe bei der Liste von den Auswahlmöglichkeiten:



Dieses öffnet die loadscreen.2da Datei, die Datenbank, welche verschiedene Definitionen aller Loading screens im Witcher enthält.

• **No Resting Allowed.** Bestimmt, ob der Spieler im Bereich ausruhen (oder meditieren) kann. Wähle das Attribut aus. Um die Einstellung zu ändern, klicke auf der rechten spalte gegenüber von No Resting Allowed Attribute. Der Spieler wird in der Lage sein sich auszuruhen wenn du False auswählst:



Die Werte, die du in diesem Attribut zuweisen kannst, sind:

- *False* – Player kann (meditieren) im diesem Bereich,
- *True* – Player kann nicht (meditieren) im diesen Bereich.

• **Safe during day.** Stellt fest, ob es im diesen Bereich während des Tages sicher ist. Wähle das Attribut aus. Um die Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom *Safe during day* Attribute. Weil unser Bereich eine Höhle ist, können wir annehmen, dass dort immer unsicher ist. So wähle false in der Auswahl:

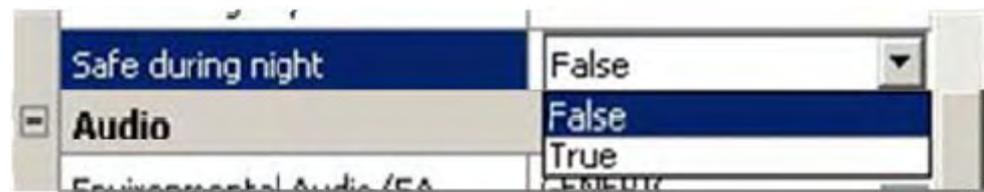


Die Werte, die Du in diesem Attribut zuweisen kannst, sind:

- False- es ist immer im diesen Bereich während des Tages unsicher
- True. Es ist immer im diesen Bereich während des Tages Sicher

Erinner dich bitte daran, dass, wenn es sicher ist, das der Spieler keine Waffen oder Zeichen benutzen kann.

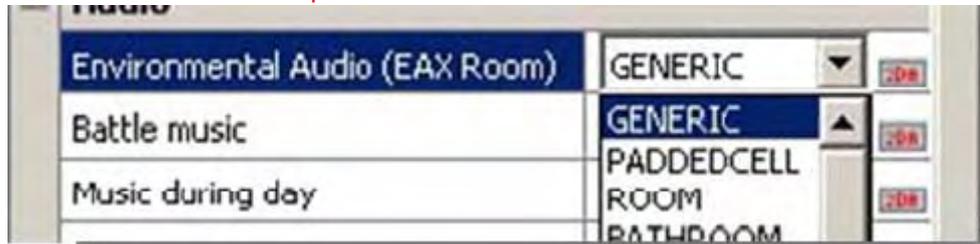
• **Safe during night.** Stellt fest, ob es im diesen Bereich während der Nacht sicher ist. Wähle das Attribut aus. Um die Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom *Safe during night* Attribute wähle false in der Auswahl:



- *False* – es ist immer im diesen Bereich während es Nacht ist unsicher
- *True* – es immer im diesen Bereich während es Nacht ist sicher

Audio Section:

• **Environmental Audio (EAX Room).** Bestimmt, welche Hintergrundtöne im Gebiet gehört werden können. So bestimmst du, wie sich der Ton in den verschiedenen Umgebungen ausbreitet. Derselbe Ton breitet sich verschieden in einem kleinen Zimmer aus wie in einem großen Saal oder einer Höhle. Wähle das Attribut aus. Um die Einstellung zu ändern, klicke auf die rechte Spalte gegenüber the *Environmental Audio (EAX Room)* Attribute wähle die *GENERIC* Option:



Diese Liste ist sehr lang und enthält eine Auswahl für praktisch jede Art von Umgebungen. Klicke auf das kleine rote Icon neben der Liste:



Dieses öffnet den `eax_rooms.2da` Ordner, die Datenbank, die alle Hintergrundtöne enthält, die im Witcher Spiel vorhanden sind. Wenn Sie nicht wissen, welcher Art sie vom Hintergrundton dem Bereich zuweisen sollen, wählen sie *GENERIC* Option.

• **Battle Music.** Enthält die Musik, die während eines Kampfes gespielt wird. Wähle das Attribut aus. Um die Einstellung zu ändern, klicke auf die rechte Spalte gegenüber von *Battle Music* Attribute. wähle Z.B. `mus_bat_boss1`:



Klicke auf das kleine rote Icon nahe der Liste von Choice:



Dieses öffnet den `ambientmusic.2da` Ordner, die Datenbanken, welche die ganze Musik enthält, die im Witcher Spiel vorhanden ist.

• **Music during day.** Enthält die Musik, die während des Tages gespielt wird. Wähle das Attribut aus. Die Liste enthält die gleichen Stücke von Musik wie *Battle Music* Attribute. Für unseren Bereich wählst Du `amb_catacombs` von der Liste. Wenn Du auf das kleine rote Icon nahe der Liste von Choices klickst, öffnet sich der `ambientmusic.2da` Ordner.

• **Music during night.** Enthält die Musik, die während der Nacht gespielt wird. Wähle das Attribut aus. Die Liste enthält die gleichen Stücke von der Musik wie *Battle Music* und *Music during day* attributes. Für unseren Bereich wähle `amb_dungeon` Piece von der Liste. Wenn Du auf das kleine rote Icon nahe der Liste von Choices klickst, öffnet sich der `ambientmusic.2da` Ordner

• **Music delay.** Stellt die Zeit zwischen dem Ende von einer Musik fest und wann es wieder beginnt. Die Zeit wird in den Sekunden gemessen. Klicke auf die rechte Spalte, auf das *Music delay* Attribute, tippe 2 ein und Drücke Enter. Dies heißt, dass die Musik anfängt, 2 Sekunden später zu spielen bevor sie anhält.

Jedes Icon stellt ein Werkzeug dar. Um ein Werkzeug auszuwählen, Linksklicke auf das jeweilige Icon. Wenn Du den Cursor über ein (Icon) schweben lässt und einen Moment wartest, wird ein kurzer tooltip unten gezeigt. Folgendes Werkzeug:



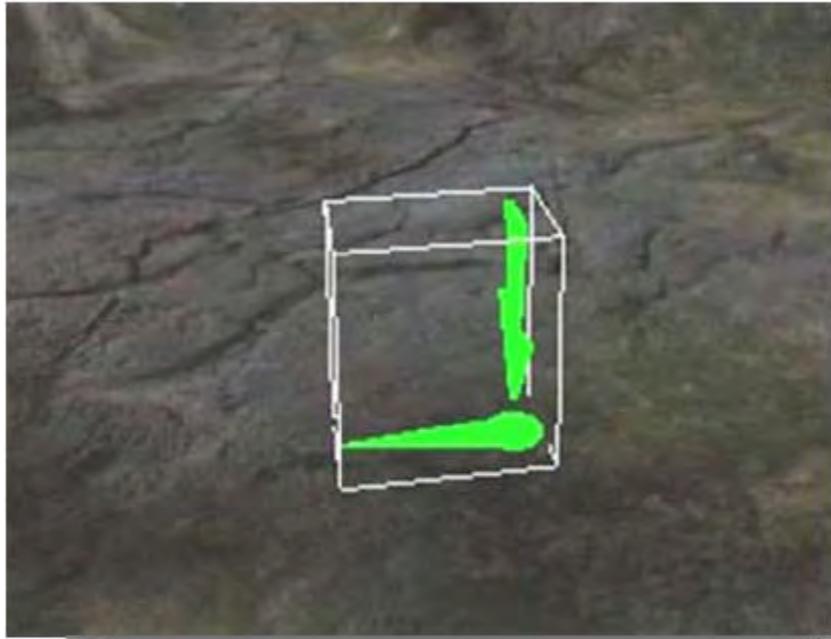
Dies ist das Werkzeug, um Gegenstände auszuwählen. Verwende es, um Objekte auszuwählen, die sich im diesen Bereich befinden. Um diese Handlung zuerst durchzuführen, musst du das Auswahl-Werkzeug auswählen, indem Du auf dieses Icon klickst, und dann auf diesen Gegenstand den du auswählen willst. Wenn Du auf den Gegenstand doppelklickst, wird ein Eigenschafts-Fenster, das die Attribute des Gegenstands enthält, auf der rechten Seite des Schirms erscheinen. Du kannst auch mehrere als ein Objekt auswählen, das sich im diesen Bereich befindet. Du kannst das auch auf mehrere Arten tun.

Zuallererst kannst du CTRL drücken und hältst es gedrückt, dann auf die Objekte klicken, dann auf das Objekt klicken das du auswählen willst. Wenn mehrere Objekte ausgewählt geworden sind, und du möchtest eins abwählen, drücke die Shift-Taste und klicke auf den Gegenstand. Du kannst auch einige Gegenstände auswählen, indem Du mit der Maus "zeichnest". Klicke und halte die linke Maustaste und verschiebe so die Maus, damit Du ein Viereck "zeichnest", das es die Gegenstände umgibt, die, Du auswählen möchtest.

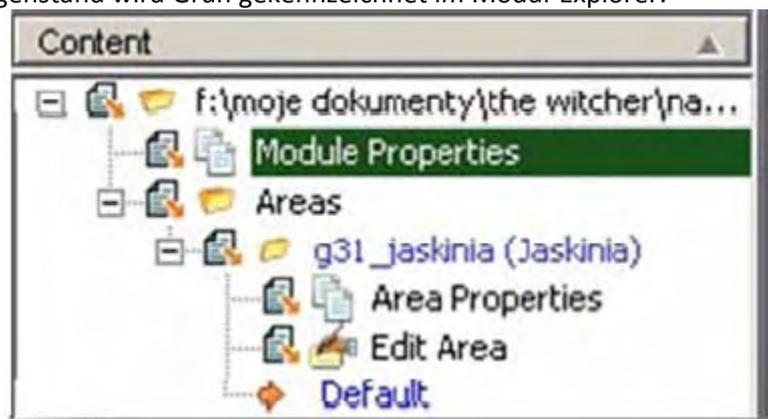
So sieht es jetzt aus:



In diesem Moment haben wir nur einen Gegenstand in diesem Gebiet, dem Start-Punkt, so kann nur dieser Gegenstand ausgewählt werden. Einmal ausgewählt kann der Gegenstand innerhalb eines weißen Rahmens gefunden werden:



Ein ausgewählter Gegenstand wird Grün gekennzeichnet im Modul-Explorer:



Wenn der Start-Punkt ausgewählt wurde, wird die Modul-Properties im Modul-Explorer Fenster auch ausgewählt. Das geschieht weil das Doppelklicken auf den Start-Point das Eigenschaftsfenster öffnet, welches die Modulattribute enthält. Diese Start-Points sind die einzigen Gegenstände die nicht entfernt werden können.

Bereich, nachdem es hinzugefügt worden ist. Du kannst es nur verschieben oder es in einen anderen Bereich legen. Gegenstand können im Modul-Explorer fenster auch ausgewählt werden.

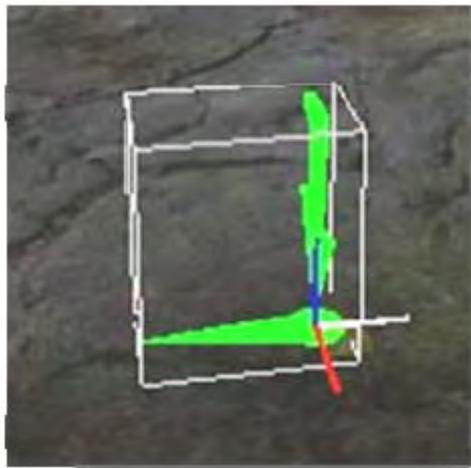


Dieses Werkzeug wird benutzt, um Gegenstände zu verschieben. Um das zu tun, wähle zuerst das Werkzeug aus indem Du auf das Icon und dann auf dem Gegenstand klickst, den Du verschieben möchtest. Der Gegenstand wird jetzt ausgewählt, und von einem weißen Rahmen umgeben die von 3 Achsen umgeben sind:



Das sind Achsen, entlang der Gegenstände die verschoben werden können. Du kannst die Gegenstände entlang der X Mittellinie (die rote Linie), der Y Mittellinie (die grüne Linie) und der Z Mittellinie (die blaue Linie) verschieben. Verschiebe den Cursor über der Mittellinie, entlang des Gegenstandes das du verschieben möchtest.

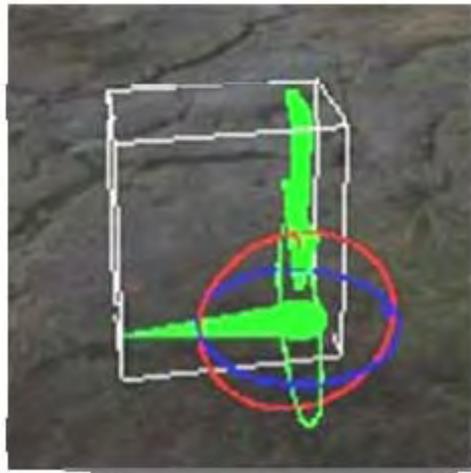
Die ausgewählte Mittellinie ist Weiß:



Das heißt, dass der Gegenstand entlang dieser Mittellinie verschoben wird. Klicke es an und halte die linke Maustaste gedrückt und verschiebe die Maus in der Richtung Deiner Wahl. Wenn der Gegenstand an seinen Bestimmungsort angelangt ist, lass die linke Maustaste los.



Dieses Werkzeug wird dazu benutzt, um Gegenstände rotieren zu lassen. Um das zu tun, wähle zuerst das Werkzeug aus indem Du auf das Icon und dann auf dem Gegenstand klickst, denn Du drehen möchtest. Das Objekt wird jetzt ausgewählt, ein weißer Rahmen wird erschaffen und drei Kreise erscheinen jetzt beim Objekt:



Diese Kreise stellen die Achsen dar, die Du benutzen möchtest, um die Gegenstände zu drehen. Du kannst entlang die X Mittellinie (der rote Kreis), die Y Mittellinie (der grüne Kreis) und die Z Mittellinie (der blaue Kreis) sich drehen. Verschiebe den Cursor über der Mittellinie, entlang über den Gegenstand den du drehen möchtest. Der ausgewählte Kreis färbt sich jetzt Gelb:



Das heißt, dass dieser Gegenstand entlang dieser Mittellinie gedreht werden kann. Klicke es an und halte die linke Maustaste gedrückt und verschiebe die Maus in eine Richtung Deiner Wahl. Wenn der Gegenstand gedreht worden ist, lass die linke Maustaste los.

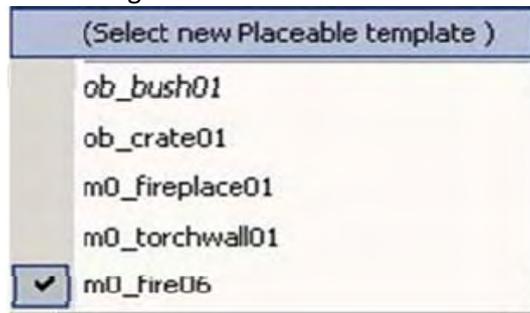
Das ist es, wenn es zu den Editierenden Werkzeugen kommt. Die anderen Werkzeuge werden dazu benutzt, um verschiedene Gegenstände in den Bereichen hinzuzufügen oder zu erstellen. Schaut aus wie folgt:



Dieses Werkzeug wird dazu benutzt, um Placeables (Stühle, Tabellen, Töpfe, Kästen, Feuer, usw.) hinzuzufügen. Indem Du dieses Werkzeug verwendest, kannst irgendein Detail in den verschiedenen Räumen in den Höhlen, Krypten hinzufügen und so weiter. Um einen Gegenstand Hinzuzufügen, klicke auf den abwärts Pfeil, der nahe bei diesem Werkzeug liegt:



Dieses öffnet ein Menü. Klicke auf die ausgewählte neue Placeable Schablone:



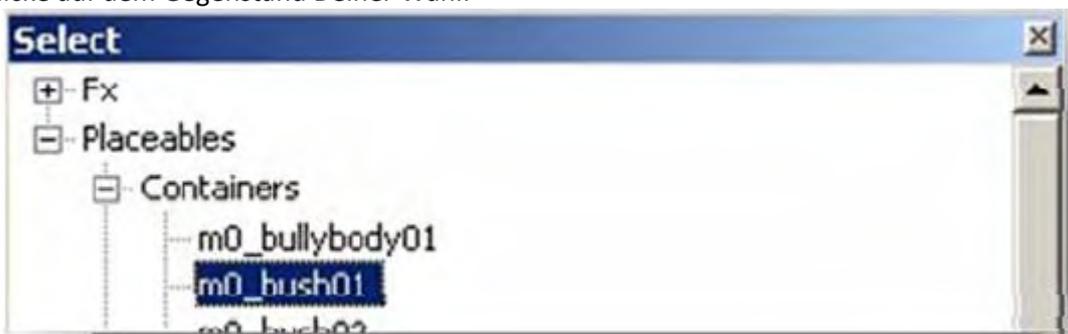
Das öffnet das folgende Fenster:



Dieses Fenster enthält alle vorhandenen Placeables, angezeigt als Äste und in Kategorien eingeteilt. Um die vollständige Kategorie zu sehen, klicke auf das kleine '+' Zeichen nahe des categoryname:



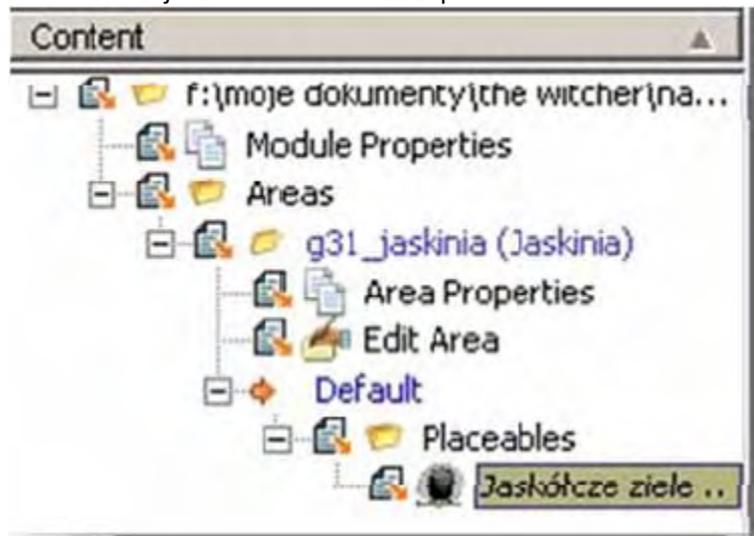
Doppelklicke auf dem Gegenstand Deiner Wahl:



Der Gegenstand wird am Cursor "gehaftet". Klicken jetzt auf den Punkt, wo du den Gegenstand setzen möchtest. Der Gegenstand wird jetzt in dem Bereich hinzugefügt:



Dieser neue Gegenstand erscheint jetzt auch im Modul-Explorer fenster:



Gegenstände im Modul-Explorer fenster werden in ihren jeweiligen Verästelungen hinzugefügt. In diesem Fall ist es die Placeables Verästelung, weil dieses die Art des Gegenstandes ist, den wir gerade unserem Bereich hinzugefügt haben. Dieses ist eine sehr nützliche Art des Anzeigens der Gegenstände, weil es die Dinge regelmäßig hält, während Du auf diesem Bereich arbeitest, besonders wenn es groß und von vielen Gegenständen voll ist. Da wir einen Gegenstand hinzugefügt haben, lass mich noch erklären, was wir alles mit ihm machen können. Rechts-Klicken auf diesem Bereich. Es macht nicht aus, ob Du dies im Bereich Editors ebene Fenster oder im Modul-Explorer tun.

Das öffnet ein Menü:



Nun werde ich die einzelnen Befehle beschreiben:

- **Open.** Öffnet das "Eigenschaften-Fenster" mit den Attributen des Objektes auf der rechten Seite des Bildschirms. Ich habe diese Attribute bereits in einem anderen Kapitel dieser Anleitung beschrieben. Um das Fenster zu öffnen kannst du auch doppelt auf den Namen des Objektes im "Objekt-Explorer" oder im "Area Editor" klicken.
- **Revert.** Wird nicht benutzt.
- **Focus on object.** Fokussiert die Sicht auf das angewählte Objekt. Diese Funktion ist sehr nützlich, wenn sich in einem Bereich sehr viele Objekte befinden und du ein Bestimmtes finden willst. Rechtsklicke auf das von dir gesuchte Objekt im „Objekt-Explorer“ und wähle den Menüpunkt „Focus on Object“. So kannst du das Objekt sehr schnell finden.
- **Export Placeables Position.** Exportiert die X, Y, Z Positionen eines Objekts in eine Textdatei. Wenn du diesen Befehl anwählst, öffnet sich der normale „Speichern unter...“ Dialog von Windows. Trage einen Namen für die Datei ein und klicke auf „Speichern“. Diese Datei kann später mit dem Windows Editor geöffnet werden.



Diese Datei enthält den Namen des Objekts und seine exakten X,Y,Z Koordinaten in diesem Gebiet (in eckigen Klammern).

- **Delete Instance:** Entfernt ein Objekt (oder genauer seine Kopie) aus dem Gebiet. Wenn du diesen Befehl auswählst, erscheint ein Fenster:



Klicke auf „Ja“ (Hier „Tak“) um das Objekt zu entfernen.

Denkt dran: Jedes verfügbare Objekt hat seine eigene Schablone. Das in den Bereich eingefügte Objekt ist nur eine Kopie auf Basis der Schablone. Du kannst ein Objekt auch durch das Klicken auf „Entfernen“ entfernen. Wähle das Objekt im „Area Editor“ aus und klicke auf „Delete“. Dadurch öffnet sich der „Removing objects“ Dialog:



Klicke auf „Yes“ um das Objekt zu entfernen.

- **Update from Template:** Aktualisiert die Attribute des Objekts anhand seiner Schablone. Wenn du die Attribute eines Objekts, das du eingefügt hast, änderst, gelten diese Änderungen nur für dieses Objekt; Wenn du die Schablone eines Objekts änderst, gelten diese Änderungen nicht für Objekte, die bereits platziert wurden. Möchtest du alle Objekte von einer veränderten Schablone aktualisieren, musst du den „Update from Template“ Befehl verwenden.
- **Edit Template:** Öffnet die Objekt Schablone auf der rechten Seite des Bildschirms. Dieses Fenster enthält die Attribute eines Objekts.

Denk dran: Jedes in den Bereich eingefügte Objekt ist nur eine Kopie auf Basis seiner Schablone.

- **Move to Layer:** Bewegt ein Objekt auf den ausgewählten Layer. Layer wurden in einem anderen Kapitel dieser Anleitung beschrieben.

Lass mich zurück zu dem Menü kommen, das erscheint, wenn du auf den Pfeil direkt neben dem Icon für das Tool zum Hinzufügen von Placeables.

Unter dem Befehl (Select new Placeable template) befindet sich eine Liste mit kürzlich benutzten Objekten. Jedes mal wenn du ein Objekt im Auswahl-Fenster auswählst und auf „Placeable template“ klickst, wird es zur Liste hinzugefügt. So musst du nicht immer den „Select new Placeable template“ Befehl verwenden. Beachte, dass ein Objekt bereits ausgewählt ist:



Dies ist sehr nützlich, wenn man mehrere Objekte der gleichen Art in den Bereich einfügen möchte. Anstatt das Objekt immer wieder über den "Select new Placeable template" Befehl auszuwählen, kannst du es aus der Liste der kürzlich verwendeten Objekte auswählen. Wenn du ein Objekt einmal ausgewählt hast, genügt es, auf den Icon zum hinzufügen von Placeables zu klicken:

Dieses Mal ist der Pfeil neben dem Icon unwichtig:



Nun klicke einfach auf den Punkt, wo du das Objekt einfügen möchtest.

Alles, was nun noch anzusprechen wäre, sind die Attribute von Placeables. Klicke doppelt auf das Objekt. Ein Fenster mit den Attributen des Placeables wird sich auf der echten Seite des Bildschirms öffnen:



Nun werde ich die einzelnen Attribute beschreiben:

General Section:

- **Name.** Beinhaltet den Namen des Objekts
- **Tag.** Beinhaltet den Tag zur Identifikation des Objekts. Tags werden vor allem bei Scripts verwendet.
- **Description.** Beinhaltet eine Beschreibung des Objekts.
- **Comment.** Beinhaltet Platz um einen Kommentar zu schreiben. Dies wird nicht im Spiel benutzt, es ist einfach ein Feld für Informationen. Du kannst alles reinschreiben, was du willst, zum Beispiel über Objekte.
- **TemplateResRef.** Beinhaltet die Schablone für ein Objekt. Wenn du zu einer anderen Schablone wechseln willst, klicke auf das kleine gelbe Icon rechts neben der Spalte „TemplateResRef“:



Dadurch öffnet sich das Standard Fenster für das Laden von Dateien. Gehe in das Verzeichnis Data\Templates\Placeables, wähle das Objekt, das du suchst und klicke auf „Open“.

Position Section:

- **Model Scale:** Legt die Größe eines bestimmten Placeables fest. Das Objekt wird gleichmäßig auf der X,Y,Z Achse vergrößert. Um die Größe eines Objektes zu ändern, klicke auf das Feld neben „Model Scale“ und trage den Wert ein, um den du das Objekt ändern möchtest.
- **XPosition:** Beinhaltet die Position des Placeables auf der X-Achse. Dies Attribut wird automatisch aktualisiert, wenn du das Objekt bewegst.
- **YPosition:** Beinhaltet die Position des Placeables auf der Y-Achse. Dies Attribut wird automatisch aktualisiert, wenn du das Objekt bewegst.
- **ZPosition:** Beinhaltet die Position des Placeables auf der Z-Achse. Dies Attribut wird automatisch aktualisiert, wenn du das Objekt bewegst.

Transition Section:

- **Waypoint Tag:** Enthält den Tag zur Identifizierung, der verwendet wird, wenn der Spieler zu einem neuen Gebiet teleportiert wird. Jedes Placeable kann als der Zielort der Teleportation dienen.

Denk dran, dass die Tags der Objekte des Spielers am Start- und Zielort der Teleportation die gleichen sein müssen. Mehr über die Bewegung zwischen Gebieten ist in einem anderen Kapitel dieses Handbuchs angesprochen worden.

Attributes Section:

- **Action:** Nicht benutzt
- **Model Name:** Wählt das 3D Modell dieses Placeables aus. Um ein anderes Modell auszuwählen, klicke auf die kleine gelbe Icon rechts neben „Model Name“:



Dies öffnet das Fenster um Dateien zu laden. Gehe in das Verzeichnis Data\Meshes, das alle Meshes enthält, wähle die Datei, die du suchst, aus und klicke auf „Load“.

- **Effect Name:** Wählt einen speziellen Effekt für das ausgewählte Placeable. Das könnte zum Beispiel eine brennende Fackel sein, die geschaffen wird, wenn Feuer auf einen Holzgegenstand angewandt wird. Um einen speziellen Effekt hinzuzufügen, klicke auf das kleine gelbe Icon rechts neben „Effect Name“:



Dies öffnet das Fenster zum Laden von Dateien. Gehe in das Verzeichnis Data\FX, das alle Effekte enthält, wähle die Datei, die du suchst, aus und klicke auf "Load".

- **Plot:** Bestimmt, ob das angewählte Placeable mit der Handlung der Geschichte verbunden ist. Wählen Sie das Attribut aus der Liste. Um die Einstellung zu ändern, klicke auf das Feld rechts neben „Plot“. Die Werte, die du diesem Attribut zuteilen kannst, sind:

- False nicht wichtig für den Plot

- True wichtig für den Plot

- **Durability:** Bestimmt die Lebensdauer des angewählten Placeables. Um die Lebensdauer zu setzen, klicken Sie auf das Feld rechts neben „Durability“ und gehen in einen Wert ein.

- **Usable:** Bestimmt, ob das angewählte Placeable vom Spieler verwendet werden kann. Wählen Sie das Attribut aus der Liste. Um die Einstellung zu ändern, klicke auf das Feld rechts neben „Usable“ und wähle die passende Auswahl aus. Die Werte, die Sie diesem Attribut zuteilen können, sind:

- False Objekt kann nicht benutzt werden

- True Objekt kann benutzt werden

- **Usable when empty:** Bestimmt, ob das angewählte Placeable einmal leer verwendet werden kann. Das bezieht sich auf Gegenstände wie Fässer oder Truhen. Abhängig von seinen Einstellungen wird es entweder möglich oder unmöglich sein, das Objekt wieder zu öffnen, nachdem Sie seinen Inhalt entfernt haben, Wähle das Attribut aus der Liste. Um die Einstellung zu ändern, klicke auf das Feld rechts neben „Usable when empty“ und wähle die passende Auswahl aus. Die Werte, die du diesem Attribut zuteilen kannst, sind:

- False Objekt kann nicht wieder verwendet werden

- True Objekt kann wieder verwendet werden.

- **Static Placeable:** Bestimmt, ob das angewählte Placeable statisch ist. Ein statischer Gegenstand kann nicht bewegt werden. Wähle das Attribut aus der Liste. Um die Einstellung zu ändern, klicke auf das Feld rechts neben „Static Placeable“ und wähle die passende Auswahl aus. Die Werte, die du diesem Attribut zuteilen kannst, sind:

False Objekt ist nicht statisch

True Objekt ist statisch

- **Hilight:** Wird nicht benutzt

• **No preload:** Bestimmt, ob das angewählte Placeable im Voraus, d. h. zusammen mit dem Modul selbst geladen werden soll, oder wenn es erforderlich im Spiel wird. Wenn ein Gegenstand zusammen mit dem Modul geladen wird, kann es die Modul-Leistung beschleunigen, aber kann das Laden des Moduls verlangsamen. Wähle das Attribut aus der Liste. Um die Einstellung zu ändern, klicke auf das Feld rechts neben „No preload“ und wähle die passende Auswahl aus. Die Werte, die du diesem Attribut zuteilen kannst, sind:

- False - Wird nicht vorher geladen

- True - Wird vorher geladen

• **Use Physics:** Bestimmt, ob das angewählte Placeable von der Physik beeinflusst wird. Wähle das Attribut aus der Liste. Um die Einstellung zu ändern, klicke auf das Feld rechts neben „Use Physics“ und wähle die passende Auswahl aus. Die Werte, die du diesem Attribut zuteilen kannst, sind:

- False - Nicht beeinflusst

- True - Beeinflusst

• **PWK collides with Physics:** Wird nicht benutzt

• **Improvised Weapon:** Wird nicht benutzt

• **Conversation:** Enthält eine Datei mit einer Konversation. Ein Konversations-Objekt“ wäre z.B. eine Statue, die eine Überschrift trägt. Ich habe bereits in einem anderen Kapitel über Konversationen geredet. Um eine Konversations-Datei hinzuzufügen, klicke auf das kleine gelbe Icon neben „Conversation“:



Darauf öffnet sich dann das Windows-Fenster zum Laden von Dateien. Gehe in den Ordner \Data\Dialogues\ (Der alle „Konversations-Dateien“ enthält), markiere die Datei, die du gesucht hast, und klicke auf „Open“.

• **Open State:** Bestimmt den anfänglichen Zustand des angewählten Placeable. Wähle das Attribut aus der Liste. Um die Einstellung zu ändern, klicke auf das Feld rechts neben „Open State“ und wähle die passende Option aus. Du kannst die folgenden Optionen wählen:

- Default, (Standard)

- Open, (Offen)

- Closed, (Geschlossen)

- Destroyed, (Zerstört)

- Activated, (Aktiv)

- Deactivated, (Deaktiviert)

- Dead. (Tot)

• **Sound Type:** Bestimmt den Ton, der gespielt wird, wenn der Spieler mit dem Placeable interagiert. Wähle das Attribut aus der Liste. Um die Einstellung zu ändern, klicke auf das Feld rechts neben „Sound Type“ und wähle die passende Option aus. Du kannst von den durch den folgenden Placeables erzeugten Tönen wählen:

- corpse,
- stone_plate,
- trap_spike,
- trap_knife,
- trap_blade,
- secr_stonewall,
- door_tomp,
- door_wooden,
- door_wooden2,
- door_fence,
- door_fence2,
- destroy_wall,
- chest_open,
- gate_wood.

Drücke auf das kleine rote Icon rechts neben „Sound Type“:



Dadurch öffnet sich die Datei placeableobjsnds.2da, die Datenbank mit den Definitionen aller „Sound Types“.

• **Usable in Combat:** Bestimmt, ob das ausgewählte Placeable im Kampf benutzt werden kann. Wähle das Attribut aus der Liste. Um die Einstellung zu ändern, klicke auf das Feld rechts neben „Usable in Combat“ und wähle die passende Auswahl aus. Die Werte, die du diesem Attribut zuteilen kannst, sind:

False kann nicht benutzt werden

True kann benutzt werden

• **Door Cursor:** Bestimmt, ob ein Tür-Cursor gezeigt werden soll, wenn du den Cursor über das ausgewählte Placeable führst. Wähle das Attribut aus der Liste. Um die Einstellung zu ändern, klicke auf das Feld rechts neben „Usable in Combat“ und wähle die passende Auswahl aus. Die Werte, die du diesem Attribut zuteilen kannst, sind:

False Wird nicht gezeigt

True Wird gezeigt

Inventory Section:

• **Has Inventory:** Bestimmt, ob das ausgewählte Placeable Ausrüstung enthält. Wähle das Attribut aus der Liste. Um die Einstellung zu ändern, klicke auf das Feld rechts neben „Has Inventory“ und wähle die passende Auswahl aus. Die Werte, die du diesem Attribut zuteilen kannst, sind:

False Keine Ausrüstung

True Ausrüstung

• **Items:** Bestimmt die Ausrüstung, die im ausgewählten Placeable gefunden werden kann. Um Ausrüstung hinzuzufügen, klicke auf das rote Icon rechts neben „Items“:



Mehr über die Erschaffung von Ausrüstung ist in einem anderen Kapitel dieses Handbuchs behandelt worden.

- **Store:** Enthält die „Storage Template“ Datei. Ein Lager kann eine Truhe sein, in der Geralt (der Spieler) Sachen unterbringen kann, die er im Moment nicht braucht. In Wahrheit werden Lager mittels des Storage Templates geschaffen, weswegen das Storage Template diesem Attribut zugeteilt wird (das Lager hat kein eigenes Template). Mehr über die Entwicklung eines Lagers ist in einem anderen Kapitel dieses Handbuchs beschrieben worden. Um ein „Store“ zuzuweisen, klicke auf das kleine, gelbe Icon rechts neben „Store“:



Dadurch öffnet sich das Windows-Fenster zum Laden von Dateien. Gehe in das Verzeichnis Data\Templates\Shops (das alle Shop-Templates enthält), markiere die gesuchte Datei und klicke auf „Open“.

- **Take only:** Bestimmt, ob es nur möglich ist, Sachen aus dem angegebenen Placeable zu nehmen oder man sowohl Dinge aus dem Placeable nehmen kann als auch Dinge hinzufügen kann. Ich bin sicher, dass dir im Verlauf des Spiels schon mehrere Truhen und andere Placeables begegnet sind. Wähle das Attribut aus der Liste. Um die Einstellung zu ändern, klicke auf das Feld rechts neben „Take only“ und wähle die passende Auswahl aus. Die Werte, die du diesem Attribut zuteilen kannst, sind:

- False - Es ist möglich, Dinge hinzuzufügen

- True - Es ist nicht möglich, Dinge hinzuzufügen

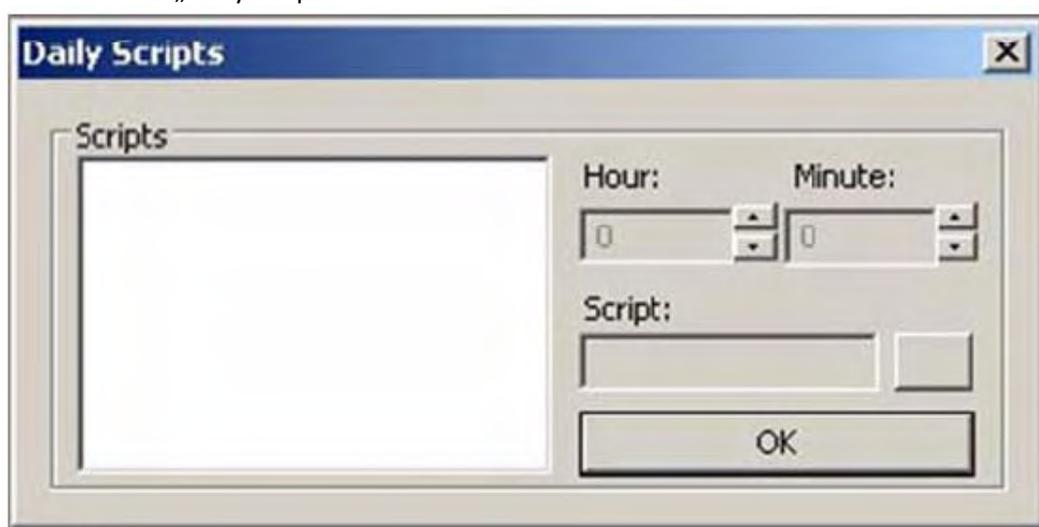
Scripts Section:

Die Attribute in dieser Sektion bestimmen die Scripts, die durch ein Ereignis ausgelöst werden, z.B wenn der Spieler einen vorhandenen Placeable verwendet. Mehr über Scripts steht in einem anderen Kapitel dieses Handbuchs.

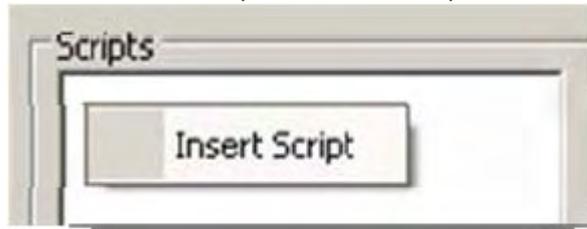
- **Daily Scripts:** Das Script wird jeden Tag zu einer bestimmten Zeit ausgeführt. Um das passende Script hinzuzufügen und um die Zeit festzulegen, an der es ausgelöst werden soll, musst du einfach auf das kleine, rote Icon rechts neben „Daily Script“ klicken:



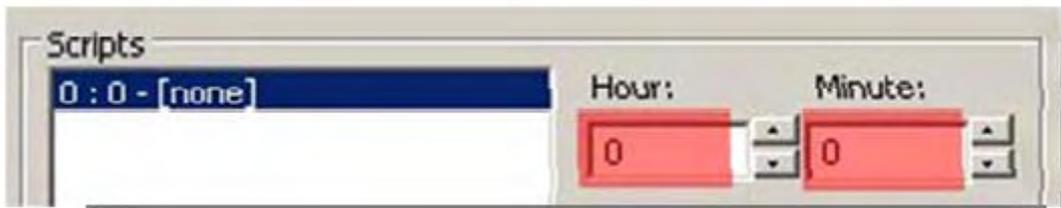
Dadurch öffnet sich das „Daily Script“ Fenster:



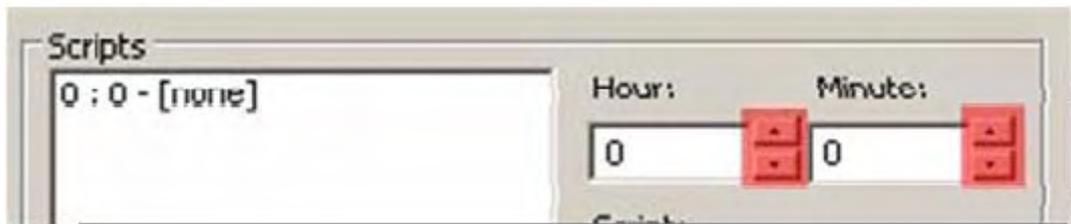
Um das passende Script hinzuzufügen, klicke mit der rechten Maustaste auf das „Script Panel“ und wähle den Menüpunkt „Insert Script“:



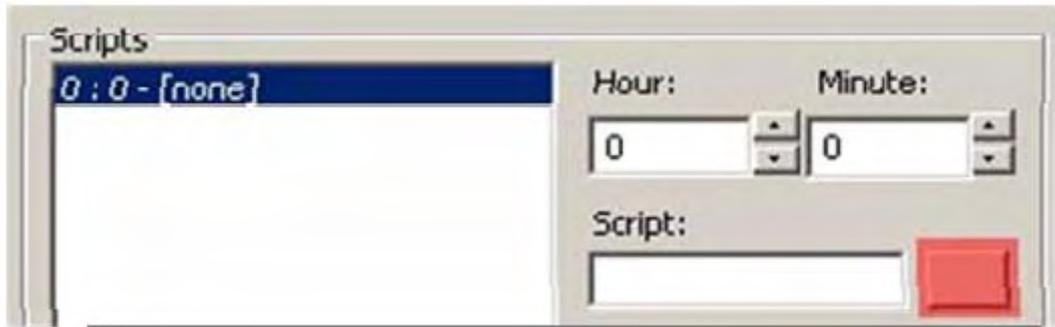
Klicke nun auf das kürzlich hinzugefügte Script und setze die Uhrzeit, zu der das Script ausgeführt werden soll. Trage die gewünschten Zeiten in die Fenster für Stunde (Hour) und Minute ein:



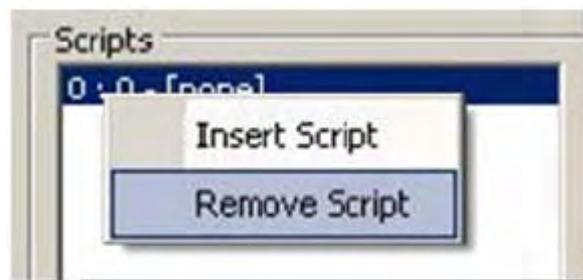
Du kannst die Zeit auch mithilfe der kleinen Pfeile rechts neben den Feldern einstellen:



Klicke als nächstes auf die kleine Box neben dem Feld „Script“:



Das öffnet das Standard-Fenster für das Laden der Dateien. Gehe in das Verzeichnis Data\Scripts (das alle Scripts des Spiels enthält), wähle das Script aus, das du laden möchtest, und klicke auf „Open“. Auf diese Weise wird das Script täglich in der voreingestellten Zeit ausgeführt. Klicke dann auf „OK“ im Script-Fenster. Natürlich kannst du mehrere Scripte hinzufügen. Wenn du ein Script entfernen willst, markiere das zu entfernende Script im „Script-Fenster“ und klicke darauf mit der rechten Maustaste; Wähle dann „Remove Script“ im Menü:



- **On closed:** Das Script wird ausgeführt, wenn der Placeable geschlossen wird.
- **On Damaged:** Das Script wird ausgeführt, wenn dem Placeable geschadet wird.
- **On Death:** Das Script wird ausgeführt, wenn das Placeable stirbt.

- **On Heartbeat:** Das Script wird alle 5 Sekunden ausgeführt.
- **On Inventory disturbed:** Das Script wird ausgeführt, wenn das Inventar geändert wird.
- **On locked:** Das Script wird ausgeführt, wenn das Placeable“ abgeschlossen wird.
- **On Melee Attacked:** Script wird ausgeführt, wenn das Placeable physisch angegriffen wird.
- **On Open:** Script wird ausgeführt, wenn das Placeable geöffnet wird
- **On Spell Cast At:** Script wird ausgeführt, wenn ein Zauber auf das Placeable gesprochen wird.
- **On unlocked:** Script wird ausgeführt, wenn das Placeable aufgeschlossen wird.
- **On Used:** Script wird ausgeführt, wenn das Placeable benutzt wird.
- **OnUserdefined:** Script wird vom Benutzer definiert.

Um ein Script auszuführen, musst du auf das kleine gelbe Icon neben dem Event, dem du ein Script zuweisen möchtest. Wenn du zum Beispiel dem Event „OnClosed“ ein Script zuweisen möchtest, klicke auf das gelbe Icon rechts neben „OnClosed“:



Physics Section:

Die Attribute in dieser Sektion sind für die Physik des Placeables zuständig.

• **Mass:** Bestimmt das Gewicht des angewählten Placeables. Wähle das Attribut aus der Liste. Um die Einstellung zu ändern, klicke auf das Feld rechts neben „Mass“ und wähle die passende Option aus. Du kannst aus den folgenden Optionen wählen:

- Feather,
- Candle,
- Sword,
- Plank,
- Axe,
- Heavy_Axe,
- Barrel,
- Stone,
- Great_Stone,
- Super_heavy,
- Hardcore_heave,
- BigBarrel,
- WaterBarrel,
- Key

Klicke auf das rote Icon rechts neben “Mass“:



Dadurch wird die Datei phy_ass.2da geöffnet, die Datenbank, Die die Definitionen der Gewichte enthält.

• **Material:** Bestimmt, woraus das angewählte Placeable besteht. Wähle das Attribut aus der Liste. Um die Einstellung zu ändern, klicke auf das Feld rechts neben „Material“ und wähle die passende Option aus. Du kannst aus den folgenden Optionen wählen:

- Wood,
- Stone,
- Gum,
- Metal

Klicke auf das rote Icon rechts neben Material:



Dadurch wird die Datei phy_material.2da, die Datenbank mit den Definitionen der einzelnen Materialien, geöffnet.

• **Primitive:** Bestimmt die geometrische Form, die den ausgewählten Placeable umgibt. Diese Form wird durch das Physik-System anerkannt. Wähle das Attribut aus der Liste. Um die Einstellung zu ändern, klicke auf das rote Icon rechts neben „Primitive“ und wähle die passende Option aus. Du kannst aus den folgenden Optionen wählen:

- Box,
- Cylinder,
- Sphere,
- Custom (of your choice)

• **Aard only:** Bestimmt, ob das angewählte Placeable nur auf das Aard-Zeichen reagiert. Wähle das Attribut aus der Liste. Um die Einstellung zu ändern, klicke auf das Icon rechts neben Aard und wähle die passende Option aus. Die Werte, die du diesem Attribut zuteilen kannst, sind:

- False - Reagiert auf alle Events und Einflüsse

- True - Reagiert nur auf Aard

• **Affected by wind:** Bestimmt, ob das angewählte Placeable auf Wind reagiert. Wähle das Attribut aus der Liste. Um die Einstellung zu ändern, klicke auf das Icon rechts neben „Affected by Wind“ und wähle die passende Option aus. Die Werte, die du diesem Attribut zuteilen kannst, sind:

- False - Reagiert nicht auf Wind

- True - Reagiert auf Wind

• **Pushable:** Bestimmt, ob das angewählte Placeable bewegt werden kann. Wähle das Attribut aus der Liste. Um die Einstellung zu ändern, klicke auf das Icon rechts neben „Pushable“ und wähle die passende Option aus. Die Werte, die du diesem Attribut zuteilen kannst, sind:

- False - Kann nicht bewegt werden

- True - kann bewegt werden

• **Sound enabled:** Bestimmt, ob ein Ton gespielt wird, wenn das ausgewählte Placeable von der Spiel-Physik beeinflusst wird. Wähle das Attribut aus der Liste. Um die Einstellung zu ändern, klicke auf das Icon rechts neben „Affected by Wind“ und wähle die passende Option aus. Die Werte, die du diesem Attribut zuteilen kannst, sind:

- False - Es wird kein Ton ausgegeben

- True - Es wird Ton ausgegeben

• **Sound type:** Bestimmt den Ton, der gespielt wird, wenn das ausgewählte Placeable von der Spiel-Physik beeinflusst wird. Wähle das Attribut aus der Liste. Um die Einstellung zu ändern, klicke auf das Icon rechts neben „Affected by Wind“ und wähle die passende Option aus. Du kannst folgende Optionen wählen:

- Basket,
- Candlestick,
- Ceramic_bottle,
- Pillow,
- Shield,
- Skull,
- Stone_block,
- Wood_barrel,
- Wood_box,
- Wood_chair,
- Body,
- Sword,
- Axe,
- Body_animal,
- HeavyGolem

Klicke auf das rote Icon rechts neben “Sound Type”:



Dadurch wird die Datei phy_sound.2da , die Datenbank, Die die Definitionen und den Umfang der Töne enthält.

• **Asset database.**

• **Asset name.**

• **Joint strength.** Bestimmt die Kraft, mit dem die Gegenstände miteinander agieren werden. Wähle dann dieses Attribut aus der Liste. Um dann die Einstellungen zu ändern, klicke dann auf die rechte Spalte gegenüberliegend vom Joint strength Attribute wähle dann die passende Auswahl dazu aus.

Du kannst auf folgendem wählen (Die Namen der Attribute sprechen für sich selbst):

- Unbreakable.
- Small.
- Medium.
- Big.

Klicke auf das rote Icon rechts neben „Joint Strenght“:



Das öffnet die Phy_joint.2da Datei, diese Datenbank enthält die Joint Strenght Definitionen.

• **Destructible.** Bestimmt ob das angegebene Platzier oder zerstörbar ist. Wähle dieses Attribut von der Liste aus. Chance zur Änderung, Klicke auf die rechte Spalte gegenüber von Destructible Attribute und wähle dann dazu die entsprechende Option, Diese wirst du dann diesen Attributen zuteilen kannst ist:

- False – Objekt kann nicht zerstört werden

- True –Objekt kann zerstört werden

• **Respawn.** Bestimmt, ob dieser angegebene hinzugefügte das zerstört wurde wieder erscheinen wird. (d.h das es auf demselben Platz wieder erscheinen kann). Wähle dann dieses Attribut aus der Liste. Um diese Einstellung zu ändern, Klicke auf die rechte Spalte gegenüber vom Respawn Attribute und wähle dann die passende Auswahl aus. Die Werte, die du dann in diesem Attribut zuteilen kannst, sind:

- False – Objekt kann nicht wiedererscheinen wenn es zerstört wurde.

- True – Objekt kann wieder erscheinen nach der Zerstörung.

• **Replace Subsection:** Diese Attribute in dieser Abteilung sind dafür verantwortlich, das ein altes 3D-Modell eines Platzier baren Gegenstandes wieder neu zu ersetzen, als zum Beispiel der Gegenstand zerstört worden ist. Das ist nützlich, weil das platzzierbare durch Wrackteile ersetzt werden sollte.

• **Asset database.**

• **Asset name.**

• **Joint strength.** Bestimmt die Kraft, wie stark die Gegenstände sich zusammenhalten

• **Model.** Bestimmt ob das 3D-Modell das alte ersetzen wird. Um ein Modell auszuwählen, klicke auf das kleine gelbe Icon das zu finden ist auf der rechten spalte gegenüber vom Modell-Attribute:



Das öffnet das Standardfenster von Windows, um dann die Dateien zu laden.

Gehe dann zum Data\Meshes ordner, wähle dann dein Modell deiner Wahl (mdl.file) und klicke auf öffnen.

• **Replace after time.** Bestimmt die Zeit wann dann diese 3D Modelle ausgetauscht werden. Um die zeit zu setzen, klicke auf die rechte Spalte gegenüber vom Replace after time Attribute und gebe einen wert ein.

• **Replace on collision.** Bestimmt ob ein Modeltausch nach einer Kollision zwischen den Gegenständen stattfinden wird. Wähle dann das Attribut aus der Liste. Um diese Einstellungen zu ändern, Klicke auf die rechte Spalte gegenüber vom the Replace on collision Attribute und wähle dann dazu die passende Auswahl aus. Die werte die du in diesen Attribut zuweisen kannst sind:

- *Off* – replacement after collision is disabled,

- *If_in_sim*,

- *Small* – after a small collision,

- *Medium* – after a medium collision,

- *Big* – after a big collision,

- *AARD* – only when the Aard sign has been used

Klicke auf das rote Icon neben „If_in_sim“:



Das wird die phy_replace_col 2da Datei öffnen, diese Datenbank enthält Definitionen der Kollisionskraft.

Destructible Options Subsection:

- **Destroy after time.** Bestimmt die Zeit, wann ein Gegenstand zerstört wird. um diesen Wert dann zu setzen, klicke auf die rechte Spalte gegenüber vom Destroy after time Attribute, und gebe dann einen wert ein.
- **Start destroy counter.** Bestimmt, wann der Countdown beginnen wird bei diesem angegebenen Gegenstand und es dann auch zu zerstören. Wähle dann dieses Attribut aus der Liste. Um diese Einstellungen zu ändern, klicke auf die rechte Spalte gegenüber vom Start destroy counter Attribute und wähle dann dazu die passende Auswahl aus. Diese Werte die du in diesen Attribut zuteilen kannst ,sind:

- *If_in_sim,*
- *Small_collision* – after a small collision,
- *Medium_collision* – after a medium collision,
- *Big_collision* – after a big collision.



Dieses Werkzeug wird dazu benutzt um Triggern(Ereigniss) in einem Gebiet hinzufügen. Was ist ein Trigger? Das ist eine abgetrennte Abteilung eines Gebietes das an eine Handlung angebunden wurde. Wenn Geralt(der Spieler) in der Abteilung sich bewegt wird etwas verursacht was den Triggern dazu beauftragt ein Ereignis auszulösen. Was genau passiert hängt vom Trigger selber ab.

Triggern werden genau wie die platzierbaren hinzugefügt. Klicke neben dem Trigger auf den Pfeil der nach unten zeigt. Wähle dazu die ausgesuchte Schablone in ihrem Menü aus und wähle dann den Trigger deiner Wahl im Fenster aus. Natürlich kannst du auch es von der Liste von den kürzlich verwendeten Triggern auswählen.

Das hinzufügen eines Triggers wird genauso benutzt wie im Falles der Platzier baren. Beachten dabei dass dieser Cursor sich in eine Rote Hand verändert hat. Klicke jetzt auf den Spot wo es zu finden sein sollte. Ein kleiner gelber Punkt wird jetzt erscheinen:



Beachte dabei bitte, dass ein Trigger ein unterschiedlicher Abschnitt eines Bereiches ist, er muss auch ziemlich groß sein. Theoretisch könnte er jede Form, bzw. eines Dreieckigen, Quadratischen annehmen. Wenn ein Punkt hinzugefügt wird, verschiebe dazu den Cursor und klicke noch einmal. Ein zweiter kleiner Punkt erscheint in diesem Bereich so wie auch eine weiße Linie die diese beiden punkten abschließt:



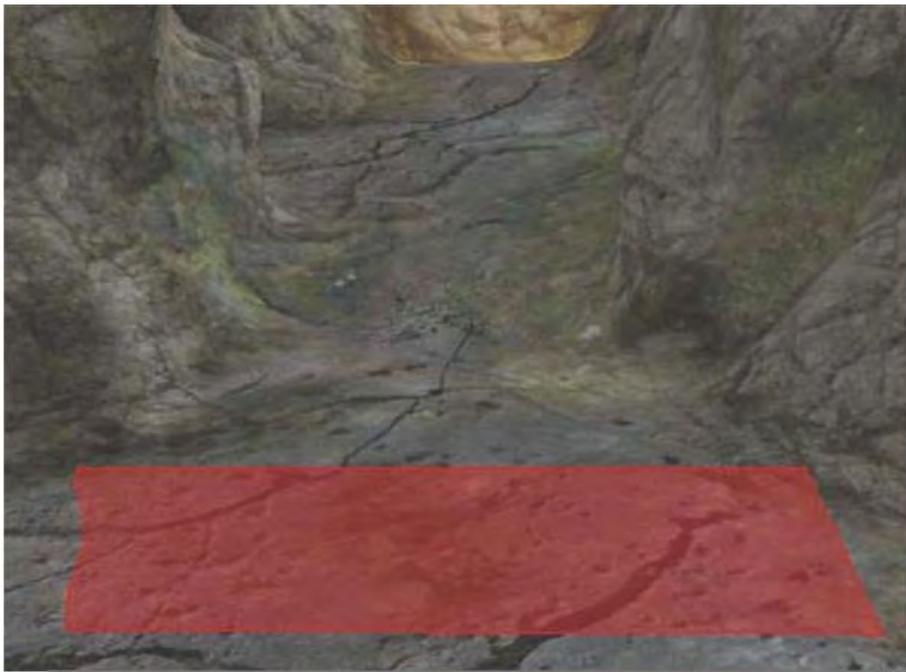
Lass uns einen rechteckigen Trigger bilden. Verschiebe dazu den Cursor auf einen anderen Punkt und klicke darauf. Jetzt haben wir drei Punkte und eine L-Förmige weiße Linie:



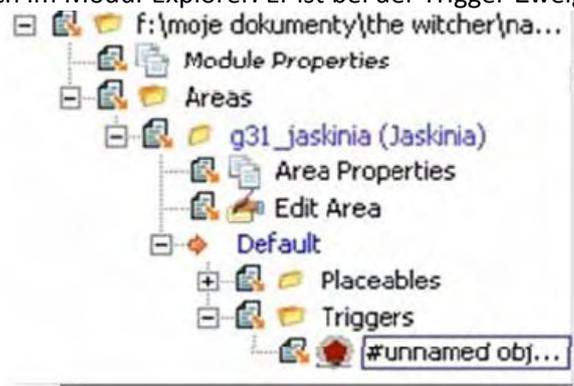
Bewege den Cursor wieder auf einen Punkt, der dann dieses Viereck abschließt klicke darauf. Jetzt haben wir vier gelbe Punkte und eine Ue- förmige weiße Linie:



Was du jetzt tun musst, ist sich das erste und den letzten Punkt anzuschließen. Um das zu tun, bringe deinen Cursor auf diesen ersten Punkt zurück und klicke ihn an. Diese Punkte werden jetzt verbunden und ein Trigger wird ausgelöst:



Der Trigger erscheint nun auch im Modul-Explorer. Er ist bei der Trigger-Zweigstelle zu finden:



Du kannst die gleichen Sachen mit den Triggern machen, das schon mit dem Platzier baren konntest, alles was wir tun müssen ist ein rechtsklick auf einen der angegebenen Trigger und dann dazu die passende Auswahl im Menü auswählen. Doppelklicke auf einen Trigger, um ein Fenster zu öffnen, das die Attribute enthält. Es wäre das Beste das im Modul-explorerfenster zu tun. Das Eigenschaften-Fenster, welches die Attribute des Triggers enthält öffnet sich jetzt auf der rechten Seite des Schirmes:



Hier betrachten wir die jeweiligen Attribute:

General Section:

- **Name.** Enthält die Namen des Triggers.
- **Tag.** Enthält die identification tags vom Trigger. Tags werden hauptsächlich mit Triggern verwendet.
- **Description.** Enthält die Beschreibung eines Triggers.
- **Comment.** Enthält einen Raum um Kommentare zu schreiben. Das wird nicht im Spiel selber verwendet, es verwendet nur ein Textfeld da kannst du schreiben was immer du möchtest. Z.B Informationen über das Modul.
- **TemplateResRef.** Enthält die Schablonen für den Trigger. Wenn du die Schablone unterschiedlich ändern möchtest, so klicke auf das kleine gelbe Icon, das auf der rechten Spalte gegenüber vom dem TemplateRes-Ref Attribut zu finden ist:



Das öffnet das Standard Windows-Fenster, das für Ladedateien benutzt wird. Gehe zum Daten- \ Schablonen- \ TriggerOrdner (das alle Triggerdaten enthält), wähle die Datei deiner Wahl aus und klicke auf öffnen.

Position Section:

- **XPosition.** Enthält die Positionen vom Trigger auf der X-Achse. Das wird automatisch aktualisiert jedes Mal wenn du die Die Position des Triggers im Bereich änderst, du kannst die Standard Einstellungen so lassen.
- **YPosition.** Enthält die Positionen vom Trigger auf der Y-Achse. Das wird automatisch aktualisiert jedes Mal wenn du Die Position des Triggers im Bereich änderst, du kannst die Standard Einstellungen so lassen.
- **ZPosition.** Enthält die Positionen vom Trigger auf der Z-Achse Das wird automatisch aktualisiert jedes Mal wenn du die Die Position des Triggers im Bereich änderst, du kannst die Standard Einstellungen lassen.

Attributes Section:

- **Type.** Stellt die Art des Triggers fest. Wähle dazu das Attribut von der Liste. Um diese Einstellung zu ändern klicke auf die rechte Spalte gegenüber von Type Attribut und wähle dazu die passende Auswahl aus. Die Werte die du diesem Attribut zuweisen kannst, sind:
 - *Generic* – General-Trigger.
 - *Area transition* – Der Trigger wird verwendet um zwischen den Bereichen sich zu bewegen.
- **Enabled.** Bestimmt ob der spezifizierte Trigger aktiviert wurde. Wähle dazu das Attribut von der Liste aus. Um diese Einstellung zu ändern. Klicke auf die rechte Spalte gegenüber vom Enabled Attribute und wähle dazu die passende Auswahl aus. Diese Werte die du dann zuteilen kannst, sind:
 - *False* – Trigger ist ausgeschaltet.
 - *True* – Trigger ist angeschaltet.
- **Highlight height.** nicht benutzbar

• **Cursor.** Stellt die Form eines Cursors ein der erscheinen wird, wenn du die Maus über den spezifizierten Trigger laufen lässt. Wähle dazu das Attribut aus der Liste. Um diese Einstellung zu ändern, klicken auf die rechte Spalte gegenüber vom Cursor Attribute und wähle dazu die passende Auswahl aus. Die Werte die du in diesen Attribut zuweisen kannst, sind:

- None – kein Cursor,
- Transition – ein Cursor der anzeigt das ein Übergang zwischen den Bereichen möglich ist,
- Use – ein Cursor, der anzeigt das ein verbrauchter Gegenstand möglich ist
- Examine – ein Cursor der anzeigt das eine Prüfung möglich ist
- Talk – ein Cursor der anzeigt das ein Gespräch möglich ist
- Walk – ein Cursor der anzeigt das Laufen möglich ist,
- XWalk,
- Attack – ein Cursor der anzeigt das ein Angriff möglich ist
- Magic – ein Cursor der anzeigt das Magie benutzen möglich ist,
- NoUse – ein Cursor der anzeigt das ein Objekt benutzen nicht möglich ist,
- Trap – ein Cursor das anzeigt das vorhanden sein einer Falle

Nun klicke auf das schmale rote Icon neben „Cursor“:



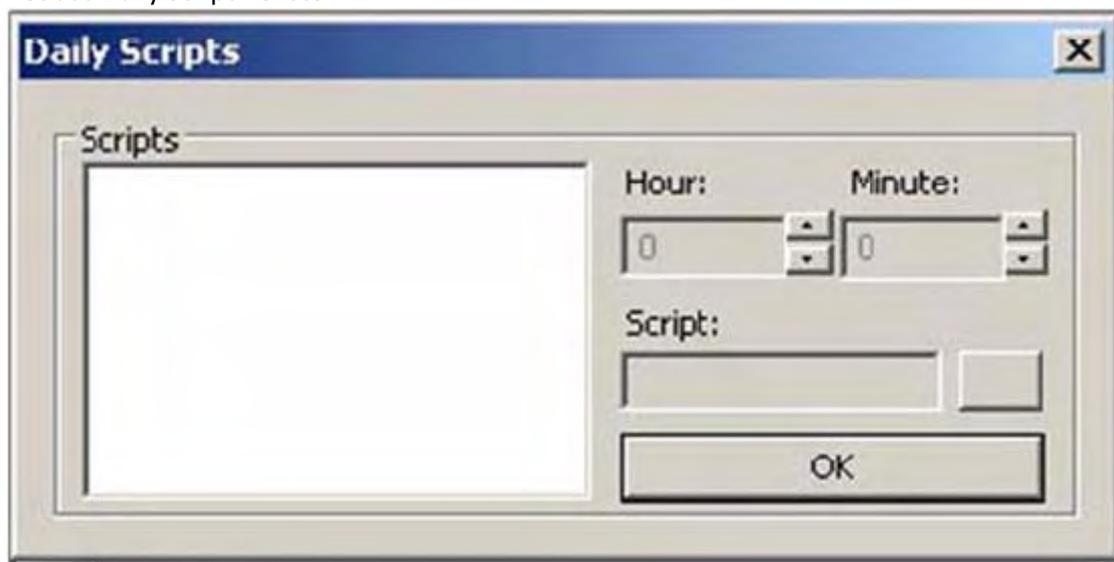
Das öffnet die cursors.2da Datei, diese Datenbank, welche die Definitionen aller Cursors enthält, die im Witcher Spiel vorhanden sind.

Scripts Section:

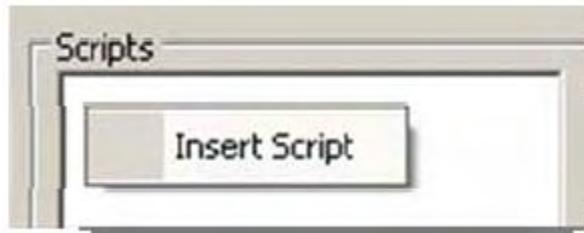
• **Daily Scripts.** Dieses Script wird täglich an einen spezifizierten Zeitpunkt laufen gelassen. Das bezieht sich auf die Spielzeit selbstverständlich. zum Hinzufügen eines passenden Scripts definiere auch die Zeit für die Ausführung, klicke dazu auf das kleine rote Icon das bei dem rechten fenster gegenüber vom Daily Scripts Attribute zu finden ist:



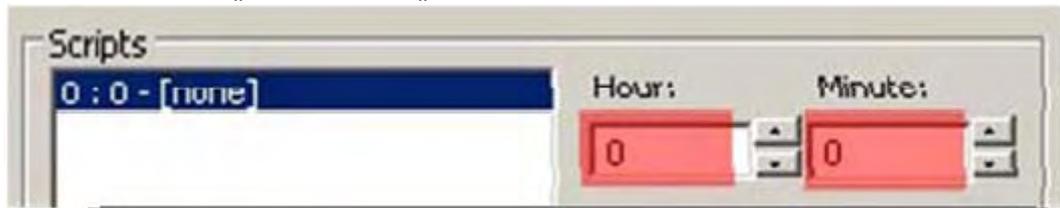
Das öffnet das Daily Script-Fenster:



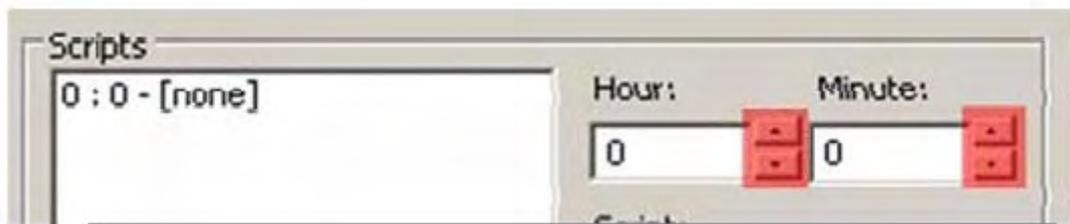
Rechtsklicke im Scripts Fenster und wähle „Insert Script“ im Menü:



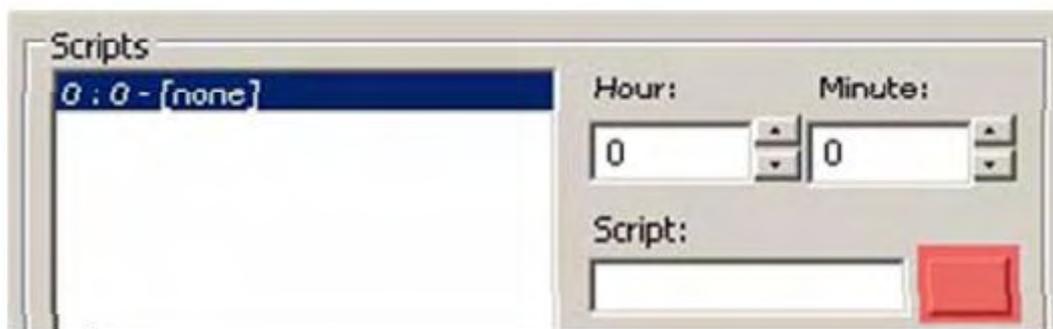
Klicke jetzt auf den vorher hinzugefügten Script und stelle die Zeit dazu ein, wann dieser Script laufen soll. Trage Dazu die Werte ein in „Stunden“ und „Minuten“ Feld:



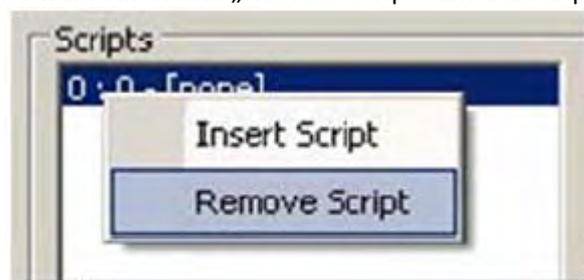
Du kannst auch die kleinen Pfeile bei den „Stunden“ und „Minuten“ Feld benutzen:



Klicke nun auf das kleine schmale Feld bei Scripts Diary:



Das öffnet das Standard Windows-Fenster, das für das laden von Ordner benutzt wird. Gehe dazu zum Data\Scripts Ordner(das alle Scripts enthält), wähle dann das Script was geladen werden soll und klicke dann auf öffnen. Auf diese Art wird das ausgesuchte Scripts täglich an der eingestellten Zeit gestartet. Doch zunächst klickst du auf O.k. Im Scriptfenster. Selbstverständlich kannst du mehr Scripte hinzufügen. Um einen Script zu entfernen, wähle dazu im Scriptsfenster das mit einem rechtsklick aus. Wähle dazu das Script aus, Und wähle im Menü „Remove Script“ um das Script zu löschen:



- **On Click.** Script wird gestartet wenn der Spieler auf den Trigger klickt.
- **On Enter.** Script wird gestartet wenn der Spieler oder ein Charakter den Trigger Bereich betreten.

- **On Exit.** Script wird gestartet wenn der Spieler oder ein Charakter den Trigger Bereich verlässt.
- **On Heartbeat.** Script wird gestartet innerhalb von 5 Sekunden.
- **On User Defined.** Script wird vom Benutzer definiert.

Um ein passendes Script zum Event zuzuweisen, klicke dazu auf das kleine gelbe Icon, das auf der rechten Spalte gegenüber vom Event zu finden ist. Z.B. Wenn du ein Script einem Event-Ereignis zuweisen möchtest, klicke dazu auf das gelbe Icon, das auf der rechten Spalte in der Nähe vom Attribut abgelegt ist:



Das öffnet das Standard Windows-Fenster, das für den Ladeordner benutzt wird. Gehe dazu zum Data\Scripts Folder (das alle Script-Dateien enthält,) wähle dann das Script deiner Wahl aus und klicke auf öffnen.

Advanced Section:

• **Auto remove key.** Bestimmt, ob ein Schlüssel automatisch vom Spieler entfernt wird bei einer Ausrüstung. Warte Welcher Schlüssel? Du siehst ein Trigger das einen Übergang in einen anderen Bereich benutzbar ist. Wenn Geralt (Der Spieler) den korrekten Schlüssel zu Verfügung hat. Wähle dazu das Attribut aus der Liste. Um diese Einstellung zu ändern klicke dazu auf die rechte Spalte gegenüber vom Auto remove key Attribute und wähle dazu die passende Auswahl dazu aus, Diese Werte die du diesem Attribut zuweisen kannst sind:

- False – Schlüssel wird nicht entfernt.

- True – Schlüssel wird entfernt.

• **Key name.** Enthält den Identifikations Tag des Schlüssels. Wenn ein Trigger, als eine Tür zwischen den Bereichen ist. Geralt (Der Spieler) muss den korrekten Schlüssel haben, um es zu Aktivieren. Dieses Attribut muss den gleichen Namen haben wie der Schlüssel der Ausrüstung von Geralt, den es wird dann nicht möglich sein einen Übergang in ein anderes Gebiet zu schaffen. Um in einen Tag zu gehen klicke dazu auf die rechte Spalte gegenüber vom Key name Attribute und gehe zum Text.

Area transition Section:

Diese Attribute von diesem Abschnitt trifft nur dann zu, wenn ein Trigger als eine Art Übergangspunkt zwischen den Bereichen dient.

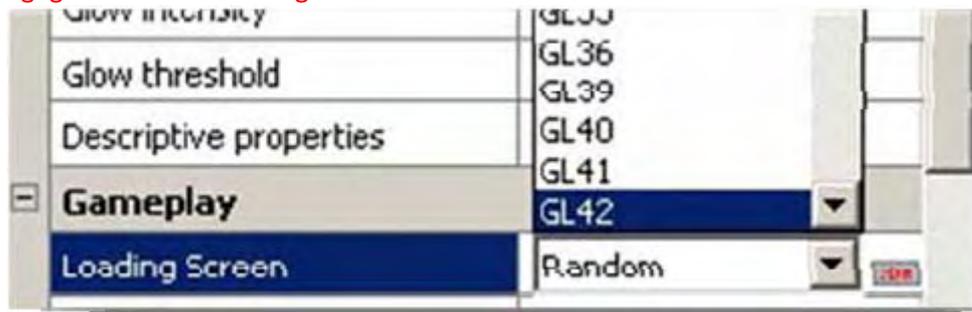
• **Target Module.** Enthält ein Modul, an dem der Spieler Teleportiert wird. Trifft aber nur dann zu wenn dieser Bereich zum Spieler Teleportiert werden soll, in einem anderen Modul. Wenn dieser Bereich das Gleiche Modul ist, müssen wir hier dann nichts ändern. Um dieses Modul zuzuweisen klicke auf das kleine gelbe Icon, das auf der rechten Spalte gegenüber vom Target Module Attribute zu finden ist:



Das öffnet das Standard Windows-Fenster, das für das Laden benutzt wird. Gehe dazu zu den Dokumenten in den Einstellungen \username\My documents\ im Witcher Ordner (Das alle Module enthält), wähle dazu die Datei deiner Wahl aus und klicke auf öffnen.

• **Destination tag.** Enthält Objekte mit den Identifikations Tag (Normalerweise ist es eine Zwischenstation) Der Geralt (der Spieler Teleportiert). Erinnerst du dich, dass ein Destinations tag Attribute das es den gleichen Tag wie das Objekt haben muss, an dem es Geralt teleportiert. Um diesen Tag zu betreten, klicke dazu auf die rechte Spalte gegenüber vom Destination tag Attribute und trage dazu den Text ein.

- **Loading Screen.** Bestimmt welches Ladebild erscheinen wird, während ein neuer Bereich geladen wird. Wähle dazu ein passendes Attribut von der Liste. Um diese Einstellung dann zu ändern, klicke dann auf die rechte Spalte gegenüber vom Loading Screen Attribute:



Diese Liste enthält alle Loadingscreens die im Witcher zu finden sind. Wähle dazu ein beliebiges(Random) aus der Liste aus. Das bedeutet dass nach einem Zufallsprinzip die Bilder angezeigt werden. Wenn es einen neuen Bereich laden tut.

Klicke dazu auf das kleine rote Icon bei „Loading Screen“:



Das öffnet die Loadscreen.2da Akte, diese Datenbank enthält die Definitionen aller Loadingscreens vom Witcher.

EAX Room Change Section:

Diese Attribute von diesem Abschnitt treffen nur dann zu, wenn ein Trigger den Hintergrund Sound verändert.

- **Set new EAX room.** Bestimmt, ob ein neuer Hintergrundsound eingestellt wurde. Wähle dazu dieses Attribut von der Liste. Um diese Einstellungen zu ändern, klicke dazu auf die rechte Spalte gegenüber vom new EAX room Attribute und wähle dazu die passende Auswahl aus. Die Werte die du in diesem Attribut zuweisen kannst, sind:

- *False* –Hintergrundsound wird sich nicht verändern
- *True* – Hintergrundsound wird sich ändern.

- **Effect volume override.**

- **New room.** Bestimmt welcher neue Sound typ in einem spezifizierten Bereich gehört wird. Wähle dazu dieses Attribut von der Liste. Um diese Einstellungen zu ändern, klicke dazu auf die rechte Spalte gegenüber von dem New room und wähle aus. Diese Liste ist sehr lang und enthält praktisch jede Definition des Bereichssounds.

Klicke dazu auf das kleine rote Icon neben „new room“:



Das öffnet die eax_rooms.2da Datei, Diese Datenbanken enthält alle Bereichssounds, Die im Witcher vorhanden sind, Wenn du nicht weisst was du in der Stichliste auswählen sollst für einen Bereich, wähle dann die GENERIC Option aus.

• **Only while on Trigger.** Bestimmt, ob ein Hintergrundsound sich verändert wenn Geralt(Der Spieler) oder ein anderer Charakter innerhalb eines Trigger-Bereiches bleibt. Wähle dazu ein Attribut von der Liste aus. Um diese Einstellung zu ändern, klicke dazu auf die rechte Spalte gegenüber vom Trigger Attribute und wähle dann die passende Auswahl aus. Diese Werte, die du in diesem Attribut zuweisen kannst sind:

- *False* –Hintergrundsound ändert sich permanent.

- *True* – Hintergrundsound soll sich nur ändern wenn Geralt oder rein Charakter innerhalb des Triggerbereiches bleibt.

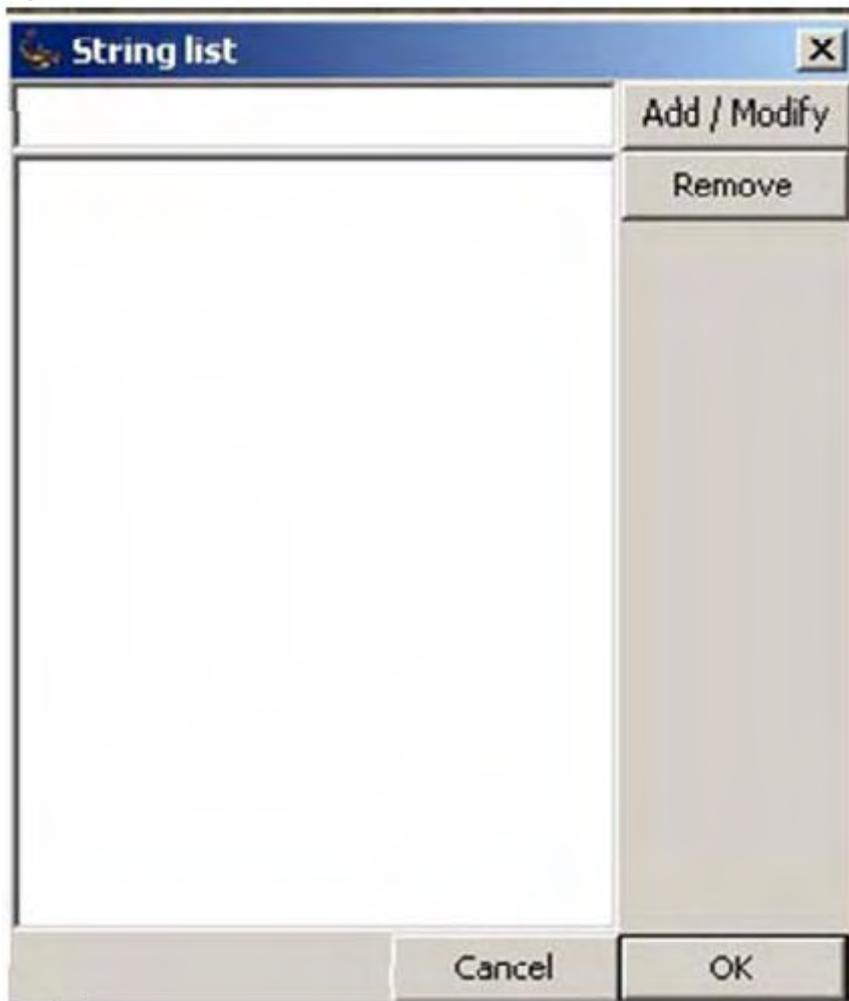
Sound groups enable / disable Section:

Diese Attribute von dieser Sectionsarbeit treffen nur dann zu, wenn ein Trigger verwendet wird der diese Sounds an und aus schaltet.

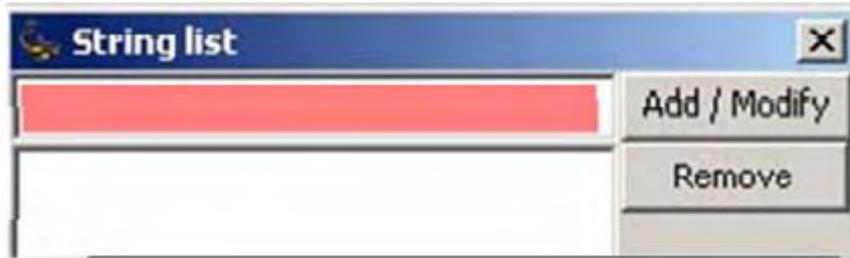
• **Enable sounds of tags.** Enthält eine liste von den Sounds Identifikations-Tag das angeschaltet werden kann. Um bei einen Auftrag im Tag Attribute etwas zuzuweisen, klicke dazu auf das kleine rote Icon auf der rechten spalte gegenüber vom dem Enable sounds of tag attribute zu finden ist:



Das öffnet das String-Listen Fenster:



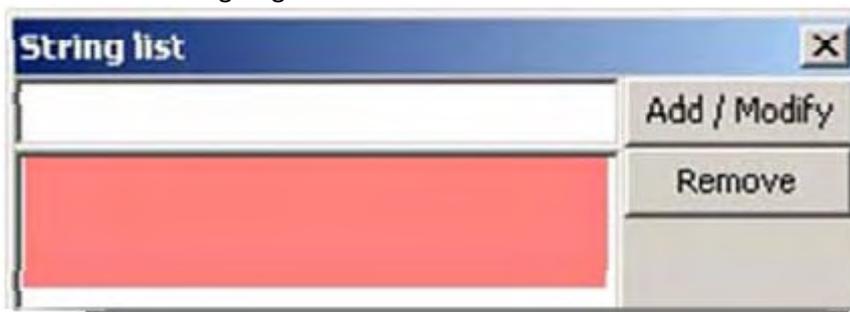
Um einen Tag zuzuweisen, klicke dazu auf das obere Feld des String listen-Fensters:



Betrete den Sound-Tag, den du anschalten möchtest, klicke dazu auf den „Add / Modify“ Button:



The entered tag wird zur Liste hinzugefügt:



Wenn Du einen vorhandenen Tag ändern möchtest, findest du bei der Liste im *String List* Fenster, klicke ihn an. Der Tag wird im oberen Teil des *String List* Fensters angezeigt. Jetzt kannst Du es ändern. Zunächst klicke, wieder auf den hinzufügen/ändern Button. Wenn Du einen vorhandenen Tag entfernen möchtest klicke ihn in der Liste im *String List* Fenster an. Klicke dann auf den „Remove“ Button um es zu entfernen:



Schließlich klicke auf den Ok Button, die im unteren Teil des String List Fensters abgelegt ist. Du kannst praktisch jede mögliche Zahl der Tags hinzufügen.

- **Disable Sounds of tag.** Enthält eine Liste von Sound identification tags, die unbrauchbar gemacht werden können. Um ein Tag dem Attribut zuzuteilen, klicke auf das kleine rote Icon abgelegt in der rechten spalte gegenüber den *Disable Sounds of tag* Attribute:



Dieses öffnet das *String List* Attribute. Der Prozess der Sperrung der Tags ist dasselbe wie in den *Enable Sounds of tag* Attribute

- **Fade Time.** Bestimmt die Zeitdauer bis ein Ton verklingen wird. Um diese Zeit zu setzen, klicke auf die rechte Spalte beim *Fade Time* Attribute und gebe einen Wert ein.

- **Reverse of Trigger.**

- **Only while on Trigger.** **Stellt fest, ob der Background Sound Disabled oder Enabled wird, nur wenn Geralt (der Spieler) oder andere Charaktere bleiben innerhalb des Triggerbereiches. Wähle das Attribut von der Liste. Um die Einstellung zu ändern, klicke auf die rechte Spalte gegenüber von dem einzigen Triggerattribut und wähle die dazu passende Auswahl. Diese Werte, die Du diesem Attribut zuweisen kannst, sind:**

- *False* – Hintergrundton soll Enabled(Aktiviert)/ Disabled(ausgeschaltet) werden

- *True* – Hintergrundton soll Enabled/Disabled werden, nur während Geralt oder ein anderer Charakter innerhalb des Trigger Bereichs bleibt



Dieses Werkzeug wird benutzt, um Waypoints einem Bereich hinzuzufügen. Trotz des Namens werden die Waypoints nicht im Allgemeinen für die Navigation verwendet. Im D'jinni Editor werden die Waypoints für zwei Zwecke verwendet: Darzustellen um die Computer gesteuerten Charaktere zu bewegen, wohin sie gehen sollen, wenn es anfängt zu regnen; und der Übergang zwischen den Bereichen. Wie man sowas erstellt wird in einem anderen Kapitel dieses Handbuchs beschrieben. Waypoints werden auf die gleiche Weise wie Placeables hinzugefügt. Klicke auf den abwärts Pfeil, der neben Werkzeug abgelegt ist *new Waypoint template* vom Menü. Zunächst wähle die Waypoints im auserwählten Fenster. Du kannst die Liste von deinen kürzlich verwendeten Waypoints auch benutzen. Klicke jetzt auf den Punkt, in dem Du die Waypoints setzen möchtest. Die Waypoints werden dem Bereich hinzugefügt:



Diese Waypoints erscheinen auch in den Waypoints in-unterzweigen des Modul-Explorer fenster:



Waypoints können auf die gleiche Weise wie Placeables geändert werden. Rechtsklicke auf den spezifizierten Waypoint und wähle die passende Auswahl vom Menü. Doppelklicke auf den Waypoint. Ein Eigenschaften Fenster, welches die Attribute der Waypoints enthält, erscheint auf der rechten Seite des Bildschirms:

#unnamed object [wp_]	
General	
Name	
Tag	wp_
Description	
Comment	
TemplateResRef	m0_wp_standard 
Position	
XPosition	1461.907349
YPosition	1505.327637
ZPosition	0.781392
Model Number	1
Attributes	
Type	Normal
Map marker	
Is map marker	False
Note text	
Visible as	NPC 
Marker enabled	False
Scripts	
Daily Scripts	0 Daily Scripts defined 

Jetzt betrachten wir die jeweiligen Attribute:

General Section:

- **Name.** Enthält die Namen der Wegpunkte
- **Tag.** Enthält die Waypoint identification tags. Tags werden hauptsächlich mit Skripten benutzt. Weil Waypoints normalerweise als Bestimmungsorte verwendet werden, zu denen Geralt sich teleportieren kann, erinnere dich daran, dass der gleiche Tag dem Gegenstand zugewiesen werden muss, von Geralts teleporter die, normalerweise die Türen Trigger sind.
- **Description.** enthält Beschreibungen der Wegpunkte
- **Comment.** Enthält einen Raum, in dem du eine Anmerkung schreiben kannst. Es wird nicht im Spiel verwendet, es ist nur dazu da um Informationen aufzufangen. Du kannst schreiben was du möchtest.
- **TemplateResRef.** Enthält eine Schablone für die Wegpunkte. Wenn Du zu einer anderen Schablone wechseln möchtest, klicke auf kleine gelbe Icon, die auf der rechten Spalte gegenüber von dem TemplateResRef Attribut Abgelegt wurden:



Dieses öffnet das Standard Windows Fenster, das für den Ladeordner benutzt wird. Gehe zu Data\Templates\Waypoints \ zum Ordner (das alle Wegpunkte enthält) wähle eine Akte Deiner Wahl aus und klicke auf öffnen.

Position Section:

- **XPosition.** Enthält die Position der Wegpunkte auf der X Mittellinie. Dieses Attribut wird automatisch aktualisiert, wann immer Du den Gegenstand verschiebst. Du kannst die Default Settings lassen.
- **YPosition.** Enthält die Position der Wegpunkte auf der Y Mittellinie. Dieses Attribut wird automatisch aktualisiert, wann immer du den Gegenstand verschiebst. du kannst die Default Setting lassen.
- **ZPosition.** Enthält die Position der Wegpunkte auf der Z Mittellinie. Dieses Attribut wird automatisch aktualisiert, wann immer Du den Gegenstand verschiebst. Du kannst die Default Setting lassen.
- **Model number.** Legt die Zahl des Modells in 3D fest, welches die Wegpunkte bildlich darstellt. Das wird als eine gelbe Markierungsfahne dargestellt. Um diese Zahl zu ändern, klicken auf die rechte Spalte gegenüber von dem *Model number* Attribut und trage einen Wert ein. Du kannst den Wert auch ändern, indem Du auf die kleinen Pfeile klickst, die auf der rechten Spalte abgelegt sind:



Attributes Section:

- **Type.** Stellt die Waypointstypen fest. Wähle dieses Attribut von der Liste. Um die Einstellung zu ändern, klicke auf die rechte Spalte gegenüber von der *Type* Attribute und wähle die passende Auswahl aus. Die Werte, die du diesem Attribut zuweisen kannst, sind:

- *Normal* – Typischer weg punkt,
- *Ingame Marker* – Wegpunkte werden markiert auf der Spiele Karte,
- *Editor Marker* – Wegpunkte werden markiert alleine für den D’jinni Editor

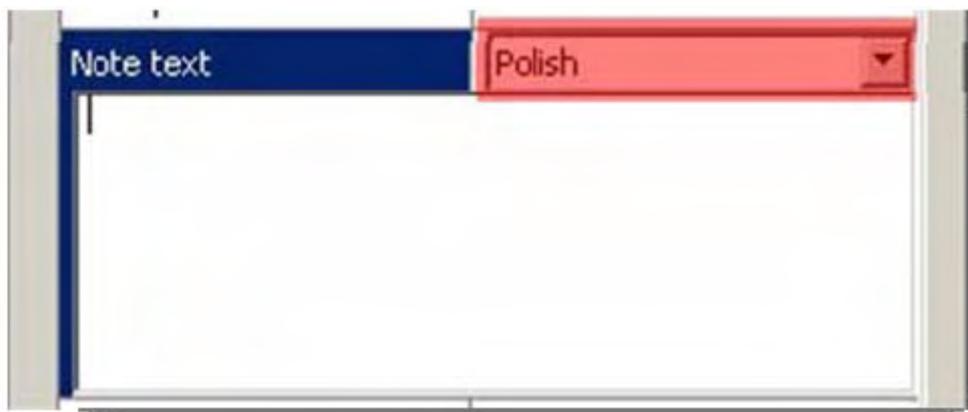
Map marker Section:

• **Is map marker.** Bestimmt, ob der angegebene Waypoint auf der Karte gekennzeichnet wird. Wird Waypoint verwendend, kannst du einen Platz kennzeichnen: Eine Quest muss ausgeführt werden; ein Höhlen Eingang wird lokalisiert; ein wichtiger Charakter steht dort usw. Wähle das Attribut aus der Liste. Um die Einstellung zu ändern, klicke auf der rechten spalte auf das Attribute appropriate Option. Und wähle die passende Auswahl aus. Diese Werte, die Du diesem Attribut zuteilen kannst, sind:

- *False* –Wegpunkte nicht markiert

- *True* – Wegpunkte wurden markiert

• **Note Text.** Enthält den Text, der erscheinen wird, wenn Du mit dem Cursor drüber fährst wird du eine Markierung auf der Landkarte sehen, Um einen Text einzugeben, klicke auf das Feld unter „Note Text Attribute“ und gebe einen Text ein der angezeigt werden soll. es ist auch möglich eine andere Sprache auszuwählen:



• **Visible as.** Bestimmt, wie der Markierung auf der Landkarte angezeigt wird. Markierung sind sichtbar und anders gefärbt je nach ihrem Zweck, ob das für Die Quest, Charaktere, usw. ist. Um auszuwählen wie die Markierung auf der Landkarte aussehen soll. Drücke auf das *Visible as* Attribute und select the appropriate Option auf der rechten spalte von Visible, und wähle die passende Auswahl aus. Du kannst aus dem folgenden wählen (Die Namen der Attribute sprechen für sich selbst):

- NPC,
- Quest,
- ImportantPlace,
- Shop,
- SwordUpgrade – the smith (hier kannst du dein Schwert upgraden),
- Campfire,
- Herb,
- PlaceOfPower,
- CircleOfElement,
- *User* – wird festgestellt durch den User

Klicke auf das kleine rote Icon, die nahe bei der Liste von Choice abgelegt sind:



Das öffnet die mapmarkers.2da Ordner, die Datenbank, die Markierungs Definitionen enthält.

• **Marker Enable.** Stellt fest, ob eine spezifizierte Markierung möglich ist. Wähle das Attribut von der Liste. Um die Einstellung zu ändern, klicke auf die rechte Spalte gegenüber *Marker Enable* Attribute und select the appropriate Option und wähle die passende Auswahl aus. Die Werte, die Du diesem Attribut zuweisen kannst, sind:

- *False* –Die Markierung ist abgeschaltet

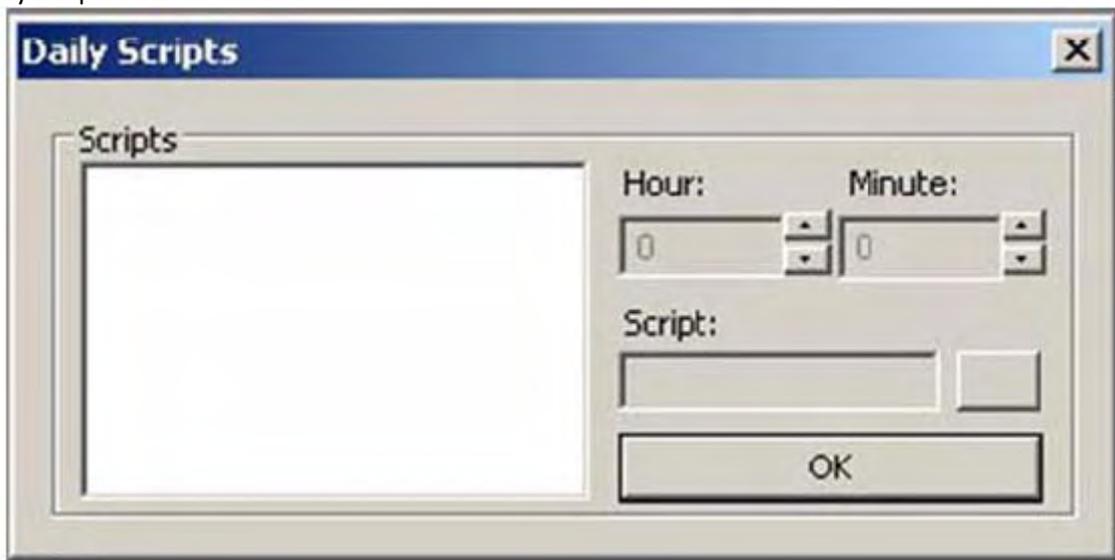
- *True* – Die Markierung ist angeschaltet.

Scripts Section:

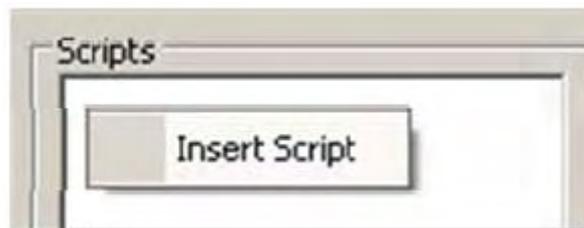
• **Daily Scripts.** Dieses Script wird täglich aus einer bestimmen Zeit laufen gelassen. Das bezieht sich natürlich auf die Spielzeit. Um den passende Script hinzuzufügen und es auch zu definieren, wenn es mal auftritt, klicke auf das kleine rote Icon das abgelegt ist in Windows opposite the *Daily Scripts* Attribute:



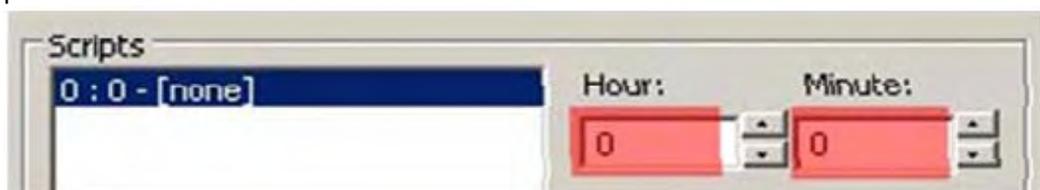
Das Daily Scripts Fenster öffnet sich:



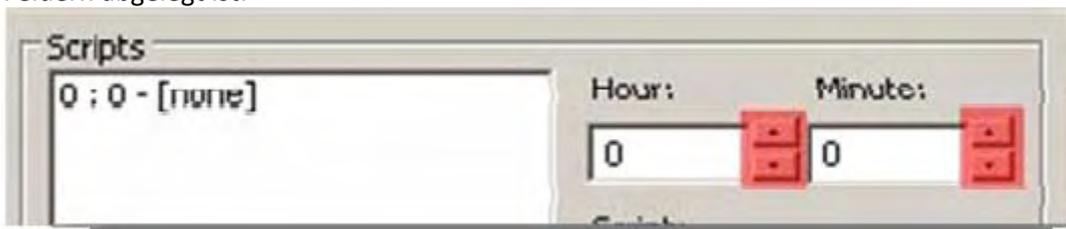
Um das passende Script hinzuzufügen klicke mit der rechten Maustaste auf die Scripte und wähle „Insert Script“ im Menü:



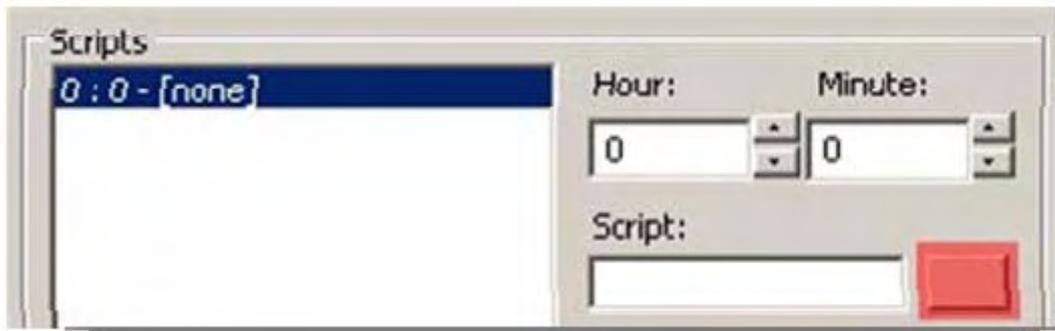
Jetzt klicke auf das vorher hinzugefügte script und stelle die Zeit ein, an dem der script laufen soll. Trage dann den passenden Wert der Stunde und Minute in die Felder ein:



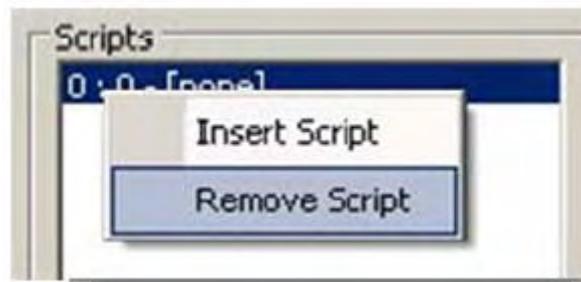
Du kannst die Zeit unter Verwendung der kleinen Pfeile auch einstellen, die nahe bei den Stunden- und Minuten Feldern abgelegt ist:



Zunächst klickst du auf den Kasten in der Nähe des Scriptes:



Das öffnet das Standard Windows Fenster, das für Ladeordner benutzt wird. Gehe zum Data\Scripts Ordner (dass alle Scripte enthält) und wähle das Script, der beim Klicken des Ordners geladen wird. Es wird geladen auf dieser Weise, dieser Script läuft täglich an dieser voreingestellten Zeit. Zunächst klicke auf O.K. im Script-fenster. Selbstverständlich kannst du mehr solcher Scripte Hinzufügen. Wenn Du ein script entfernen möchtest, wähle den Scripts Panel aus und wähle mit Rechtsklick „Remove Script“ vom Menü aus:

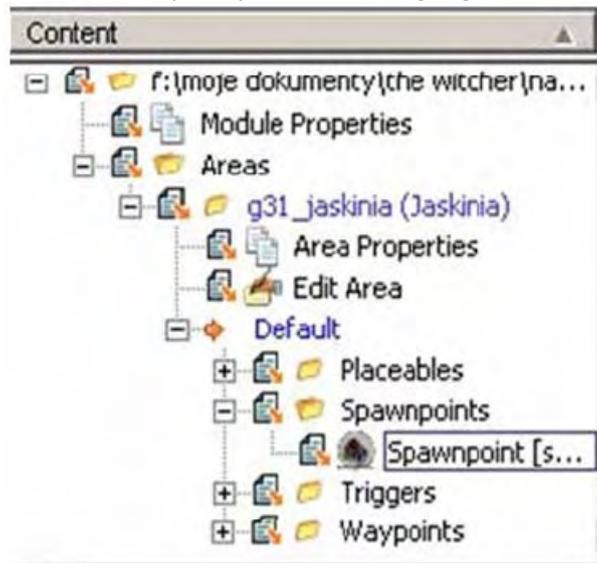


Dieses Werkzeug wird benutzt, um Spawn Punkte in einem Bereich hinzuzufügen. Ich habe bereits Spawn Punkte in einem anderen Kapitel dieses Handbuches umfasst. Jetzt werden wir besprechen, wie Spawn Punkte zu den respawn Charakteren benutzt werden. Es wird auf die gleiche Weise wie Placeables hinzugefügt. Klicke auf den abwärts Pfeil, der nahe bei dem Werkzeug abgelegt ist und wähle einen neuen Spawnpoint Schablone vom Menü aus. Zunächst wähle die Spawnpunkte im ausgewählten Fenster. Du kannst die Liste die vor kurzem benutzten Spawn Punkte auch benutzen. Klicke jetzt auf den Punkt, wo du den Spawnpoint setzen möchtest.

Es wird jetzt in dem Bereich hinzugefügt:



Diese Spawn Point erscheint auch in den Spawnpoints Verzweigungen des Modul-Explorers:



Spawnpunkte können auf die gleiche Weise wie Placeables geändert werden. rechtsklicke beim Spawnpoint und wähle die passende Auswahl im Menü. Doppelklicke bei den Spawnpoint. Ein Eigenschaften-Fenster, welches die Attribute der Spawnpunkte enthält, erscheint auf der rechten Seite des Bildschirms:

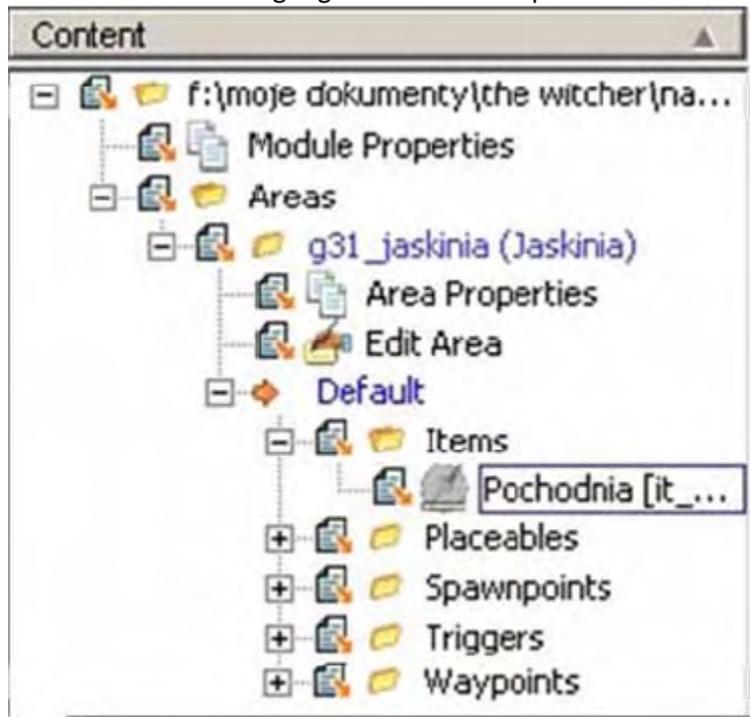
Spawnpoint [sp_standard]	
General	
Name	Spawnpoint
Tag	sp_standard
Comment	
Creature Uses Orientation	False
Spawning parameters	
Appear placeable template	
Spawn Categories	0 strings defined
Options	(Can appear here, Can dis.
TemplateResRef	m0_spawnpoint
Position	
X	1455.379761
Y	1506.546265
Z	0.269027

Das wurde bereits in einem anderen Kapitel dieses Handbuches beschrieben.



Dieses Werkzeug wird benutzt, um Items in diesem Bereich hinzuzufügen. Items bestehen hauptsächlich aus Waffen, Tränke, Rüstung, alchemical Bestandteile und so weiter. Es ist ratsam, Items in etwas Placeables, wie Kästen oder Fässer zu setzen. Sie sollen nicht irgendwo verstreut durcheinander Liegen. Items werden auf die gleiche Weise wie Placeables hinzugefügt. Klicke auf den abwärts Pfeil, der nahe bei dem Werkzeugleiste ist und wähle aus eine neue Item-Schablone vom Menü. Zunächst wähle ein Item aus im auserwählten Fenster. Du kannst die Liste die vor kurzem benutzten Items auch benutzen. Zunächst klicke auf den Punkt, wo du die Items setzen möchtest.

Es wird im Bereich und in den Items Verzweigungen des Modul-Explorer Fensters hinzugefügt:



Items können auf die gleiche Weise wie Placeables geändert werden. rechtsklicke auf einem Item und wähle die passende Auswahl vom Menü aus. doppelklicke auf dieses Item. Ein Eigenschaften-Fenster, welches die Attribute des Items enthält, erscheint auf der rechten Seite des Bildschirms:

Pochodnia [it_sclub_001; disarm_easy]	
General	
Name	Pochodnia
Tag	it_sclub_001; disarm_eas
Description	<cBOLD>Broń, źródło św
Comment	
TemplateResRef	m0_it_sclub10
MiniGameData	
Virtual Item	False
Readable Item	False
Name Color (hex)	
Desc Color (hex)	
Position	
Model Scale	1.000000
XPosition	1446.917480
YPosition	1509.807007
ZPosition	-1.547961
Attributes	
Base type name	short_sclub
Appearance	201
Model appearance	201
Plot	False
Quest Item	False

Lass uns die jeweiligen Attribute betrachten:

General Section:

- **Name.** Enthält die Namen der Items
- **Tag.** Enthält den item identification tags. Die Tags werden hauptsächlich mit den Scripten benutzt.
- **Description.** Enthält die Beschreibungen der Items, dass in der Ausrüstung des Spielers angezeigt wird.
- **Comment.** Enthält einen Raum, in dem du eine Anmerkung schreiben könntest. es wird nicht im Spiel verwendet, es ist nur dazu da um Informationen zu speichern. Du kannst schreiben was immer du möchtest.
- **TemplateResRef.** Enthält die Schablonen für die Items. Wenn Du zu einer anderen Schablone wechseln möchtest, klicke auf das kleine gelbe Icon, die auf der rechten Spalte gegenüber von dem TemplateResRef Attribut Abgelegt wurde:



Dieses öffnet das Standard Windows Fenster, das für Ladeordner benutzt wird. Gehen Sie zu Data\Templates\Items Ordner (das alle Items enthält), wählen Sie die den Ordner Ihrer Wahl aus und klicken Sie auf öffnen.

- **MiniGameData.**
- **Virtual Item.** Stellt fest, ob die Items virtuell sind. Die Virtuellen Items sind die mit den Fächern für die anderen Items. Wähle das Attribut von der Liste aus. Um die Einstellung zu ändern, klicke auf die rechte Spalte gegenüber von den virtuellen Item-Attributen und wähle das passende aus. Die Werte, die Du diesem Attribut zuweisen kannst, sind:
 - *False* – Die Items sind nicht virtuell
 - *True* – Die Items Sind Virtuell
- **Readable Item.** Lesbares Item. Stellt fest, ob die Items vom Spieler gelesen werden können. Das sind Items wie Bücher, Verträge oder Anmerkungen. Wähle das Attribut von der Liste aus. Um diese Einstellungen zu ändern, klicke auf die rechte Spalte gegenüber von dem lesbaren Item-Attribute und wähle die passende Auswahl aus. Die Werte, die Du diesem Attribut zuweisen kannst, sind:
 - *False* – Die Items können nicht gelesen wird
 - *True* – Die Items kann man Lesen
- **Name Color (hex).** Enthält eine Farbkennzeichnung, gespeichert im Hex-Format. Der Name dieses Items mit der Ausrüstung des Spielers wird in dieser Farbe angezeigt.
- **Desc Color (hex).** Enthält eine Farbkennzeichnung, gespeichert im Hex-Format. Der Name dieses Items mit der Ausrüstung des Spielers wird in dieser Farbe angezeigt.

Position Section:

- **Model Scale.** Stellt die Skala des Items ein. Dieses Item wird gleichmäßig auf die X, Y, Z Achsen eingestuft. Um die Skala der Items zu ändern, klicke auf die rechte Spalte gegenüber von dem *Model Scale* Attribut und trage den Wert ein wie du das Item ändern willst.

• **XPosition.** Enthält die Position der Items auf der X Mittellinie. Dieses Attribut wird automatisch aktualisiert, wann immer Du das Item verschieben musst. Du kannst die Default Setting so lassen.

• **YPosition.** Enthält die Position der Items auf der Y Mittellinie. Dieses Attribut wird automatisch aktualisiert, wann immer du das Item verschieben willst. Du kannst die Default Setting so lassen.

• **ZPosition.** Enthält die Position der Items auf der Z Mittellinie. Dieses Attribut wird automatisch aktualisiert, wann immer Du das Item verschieben willst. Du kannst die Default Setting so lassen.

Attributes Section:

• **Base type Name.** Bestimmt die Grundgruppe für das bestimmte Item. Zum Beispiel kann die Grundgruppe für eine Fackel eine Keule sein. Wähle das Attribut von der Liste aus. Um diese Einstellungen zu ändern, klicke auf die rechte Spalte gegenüber vom Typ Base und wähle die passende Auswahl aus. Diese Liste ist sehr lang. Du kannst aus dem folgenden auswählen (die Namen der Gruppen sprechen für sich selbst):

- *torch*, (Fackel)
- *steelsword*, (Stahlschwert)
- *silversword*, (Silberschwert)
- *long_axe*, (lange Axt)
- *long_club*, (Lange Keule)
- *Long flail*, (Lange flegel)
- *Crossbow*, (Armbrust)
- *bow*, (Bogen)
- *short_flail*, (kurzer flegel)
- *pole_halberd*, (Hellebarde)
- *potion_unfinished* – Ergebnis des erfolglosen Brauens von alchemistischen Zutaten,
- *potion_poison* – das Ergebnis des erfolglosen Brauens von alchemistischen Zutaten,
- *Short axe*, (kurze Axt)
- *Ammunition arrow*, (MunitionsPfeil)
- *Dagger*, (Dolch)
- *Short club*, (kurze Keule)
- *Neckle*, (kette)
- *Magical Ring*, (magischer Ring)

- *potion*, (Trank)
- *Ring*, (Ring)
- *shield*, (Schild)
- *key*, (schlüssel)
- *natural*, (Natur Talente)
- *witcher_armor*, (Witcher Rüstung)
- *scroll*, (rolle)
- *Gold*, (Gold)
- *gem.*, (Edelstein)
- *Ingredient*, (Inhaltstoffe)
- *Upgrade component*, (aufwerten der Komponente)
- *spell*, (Zauberspruch)
- *Fists*, (Fäuste)
- *magical_neckle*, (Magische Kette)
- *Medallion*, (Medallion)
- *Trophy*, (Trophäe)
- *Quest*, (quest)
- *mg_dice*, (Würfel)
- *natural_ranged* – (Waffen benutzt durch die Monster)
- *Food*, (essen)
- *other*, (sonstiges)
- *grease*, (Schmierfett)
- *bomb*, (Bombe)
- *drink*, (Trinks)
- *dice*, (Würfel)
- *spellranged* – (ordne den Zauberspruch)
- *sp_rngd_amm*, (?)

- *ammunition bolt* – (Munition für die Armbrust)
- *Thorn* –(geworfen von den echinops/Archespore)
- *echin_bow* – (ordnet die benutzte Waffe der echinops/Archespores)

Wenn du die Waffen herstellst, musst du die den Grund typ wählen, der deiner Kreation am ähnlichsten ist. Klicke auf das kleine rote Icon in der Nähe der Liste der Auswahl:



Dieses öffnet den basicitems.2da Ordner, die Datenbank, welche die Definitionen aller grundlegenden Items im Witcher enthält

- **Appearance.** Stellt das Aussehen eines angegebenen Items ein. Dieser numerische Wert in diesem Attribut verändert das Aussehen des Items. Es gibt zwei Möglichkeiten, diesen Wert zu ändern - trage den passenden Wert ein und drücke Enter oder benutze die kleinen Pfeile, die nahe bei dem Appearance Attribute in der rechten Spalte abgelegt sind:



- **Model appearance.** Stellt das Aussehen eines 3D Items ein. Dieser numerische Wert in diesem Attribut verändert das Aussehen des Items. Es gibt zwei Möglichkeiten, diesen Wert zu ändern - trage den passenden Wert ein und drücke Enter oder benutze die kleinen Pfeile, die nahe bei dem Model appearance Attribute das in der rechten Spalte abgelegt ist:



- **Plot.** Stellt fest, ob ein angegebenes Items für die storyline wichtig ist. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber von dem Plot Attribute und wähle die passende Auswahl aus. Die Werte, die Du bei diesem Attribut zuweisen kannst, sind:

- *False* – diese Item ist nicht für den Plot wichtig
- *True* Dieses Items ist für den Plot wichtig

- **Quest Item.** Stellt fest, ob ein das angegebenes Item für das Quest wichtig ist. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber von der appropriate Option und wähle die passende Auswahl aus. Die Werte, die Du bei diesem Attribut zuweisen kannst, sind:

- *False* Diese Item wird nicht für die Quest benutzt
- *True* – Dieses Item wird für die Quest benutzt

- **Quality.** Stellt Item-Qualität ein. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom Quality Attribute und wähle die passende Auswahl aus. Diese Werte, die Du bei diesem Attribut zuweisen kannst, sind:

- *Very Low* – sehr niedrige item Qualitäten,
- *Low* – Niedrige item Qualitäten,
- *Normal* – normale item Qualität,

- *High* – Hohe item Qualität,

- *Very High* – sehr hohe item Qualität.

• **Stack Size.** Bestimmt, wie viele Sachen desselben Typs in einem Warenbestand-Ablagefach (ein oben auf dem anderen) abgelegt werden soll. Das ist für Dinge wie Arzneitrank und alchemistische Zutaten besonders wichtig, von denen es mehrere der Warenbestände geben kann. Um die Menge zu ändern, klicke auf die rechte Spalte gegenüber dem Stack Size Attribute, gebe den passenden Wert ein und drücke ENTER.

• **Max Stack Size.** Stellt die maximale Menge der Items der gleichen Art fest, die in einem Warenbestand-Ablagefach gestapelt werden kann. Um diesen Wert zu ändern, klicke auf die rechte Spalte gegenüber vom Max Stack Size Attribute, gebe den passenden Wert dazu ein und drücke ENTER.

• **Use Physics.** Stellt fest ob ein Item z.B., das durch das Physiksystem beeinflusst wird durch dieses Spiel, ob es durch den Wind weggeblasen werden kann. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom Use Physics Attribute kannst du die passende Auswahl auswählen. Diese Werte, die Du diesem Attribut zuweisen kannst, sind:

- *False* –Dieses Items wird nicht beeinflusst durch das Physiksystem des Spiels

- *True* – Dieses Items wird beeinflusst durch das Physiksystem des Spiels.

• **Pickable in combat.** Stellt fest, ob ein Item während des Kampfes aufgenommen werden kann. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom Pickable Attribut und wähle die passende Auswahl aus. Die Werte, die du diesen Attributen zuweisen kannst, sind:

- *False* Dieses Item kann nicht während des Kampfes aufgenommen werden

- *True* Dieses Item kann während des Kampfes aufgenommen werden

• **Abilities.** Enthält die item Fähigkeiten Dieses Thema von den Fähigkeiten werden woanders im Handbuch beschrieben.

• **Persistent item.** Nicht benutzbar.

Item Description Section:

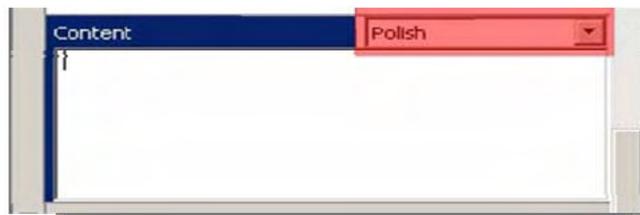
• **Picture.** Enthält ein Bild des Items. Diese Bilder werden im Warenbestand des Spielers angezeigt. Um ein Bild hinzuzufügen, klicke auf das kleine gelbe Icon.

Klicke auf das kleine gelbe Icon neben „Picture“:



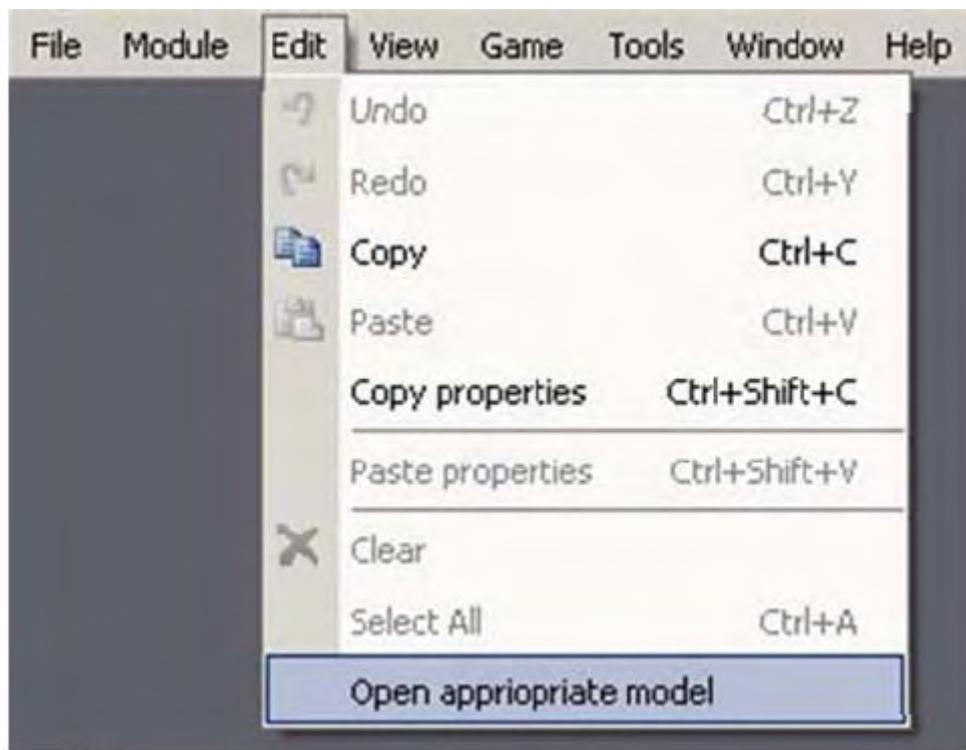
Das öffnet das Standard Windows Fenster, das für Ladeordner benutzt wurde. Wähle den passenden Ordner (einen tga Ordner) aus und klicke auf öffnen.

• **Content.** Enthält den Inhalt eines Buches, z.B. um einen Text einzutragen, klicke auf die rechte Spalte gegenüber vom Content Attribute, den passenden Text und drücke Enter. Es ist auch möglich, eine andere Sprache für eine Beschreibung auszuwählen:

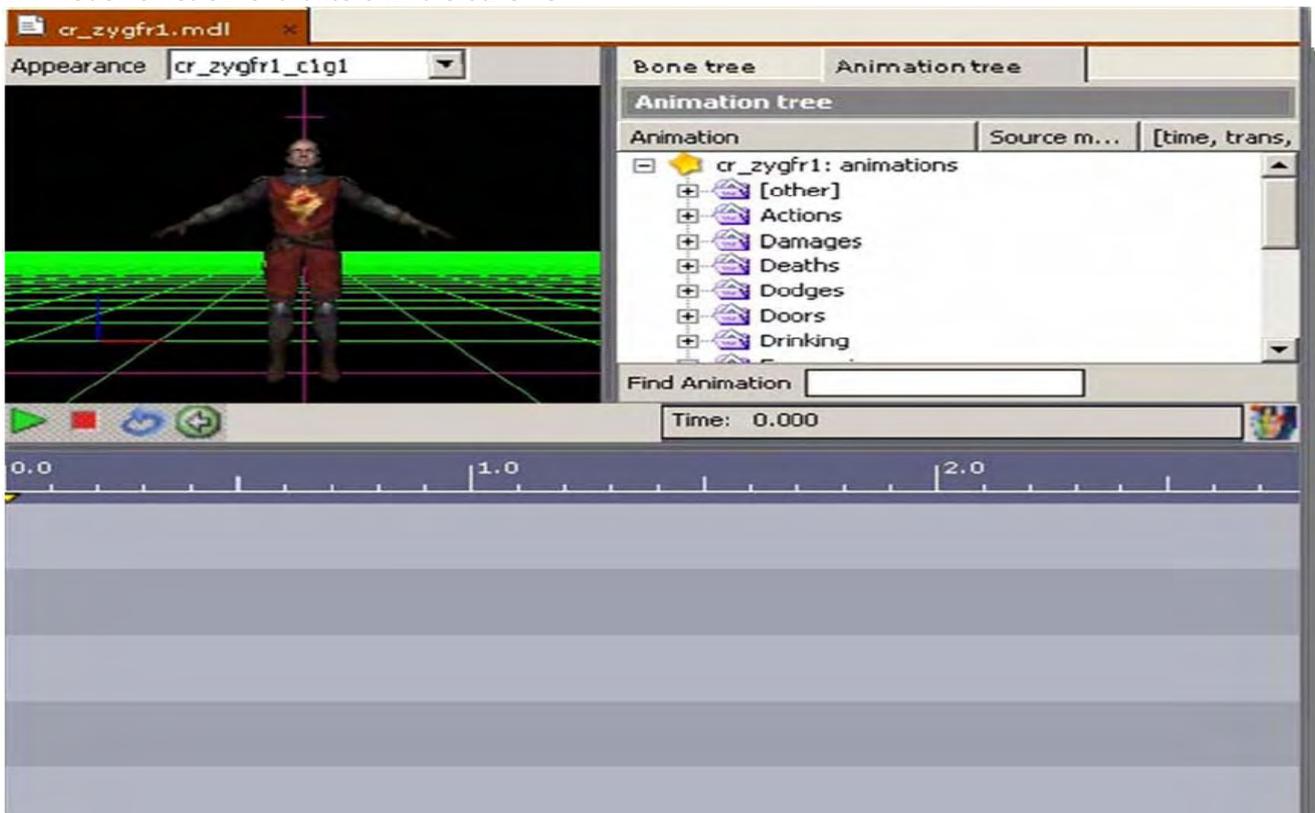


Armory Items Only Section:

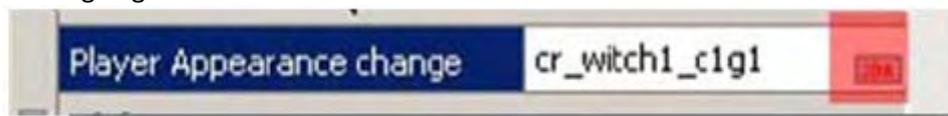
• **Player Appearance change.** Bestimmt das Äußere eines Charakters nach dem einstellen auf die spezifische Rüstung. Dieses Attribut ist nur wichtig, wenn das Item eine Rüstung ist und durch Geralt, d. h. den Spieler verwendet werden kann. Sonst ist das irrelevant. Um das passende Äußere auszuwählen, klicke auf die rechte Spalte gegenüber vom Player Appearance Change und wähle die passende Auswahl aus auf der Liste. Um eine dreidimensionale Ansicht vom Charakter zu sehen, wähle ein passendes Modell vom Editierten Menü:



Ein Modell eines 3D Charakters wird erscheinen:



Klicke auf das kleine rote Icon, die auf der rechten Spalte gegenüber vom Player Appearance Change Attribute abgelegt ist:



Das öffnet denn appearance.2da Ordner, die Datenbank, welche die Definitionen des Aussehens der bestimmten Charaktere im Witcher enthält

Alchemy Section:

Attribute in dieser Gruppe sind nur relevant, wenn ein bestimmtes Item eine alchemistische Zutat ist.

• **AlchemyIngredient.** Stellt die Art der alchemistischen Items fest. Dieses ist nur wichtig, wenn ein Item wirklich ein alchemistisches Item ist. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom Alchemy Ingredient Attribute und wähle die passende Auswahl aus. Klicke jetzt auf das kleine rote Icon, in der Nähe von der selection List abgelegt sind:



Dies öffnet die alchemy_ingre.2da Ordner, die Datenbank, welche die Definitionen aller alchemistischen Items enthält, die im Witcher vorhanden sind.

• **AlchemyBase.** Bestimmt den Grundursprung einer alchemistischen Zutat. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber dem AlchemyBase und wähle die passende Auswahl aus. Diese Werte, die Du bei diesem Attribut zuteilen kannst, sind:

- None, (nichts)

- Bottle Poor, (schwache Flasche)

- *Bottle Standard, (Normale Flasche)*
- *Bottle Good, (Gute Flasche)*
- *Bottle Superior, (außergewöhnliche Flasche)*
- *Grease Standard, (Normaler Schmierfett)*
- *Grease Good, (Guter Schmierfett)*
- *Powder Standard, (Normaler Schießpulver)*
- *Powder Good. (Guter Schießpulver)*

Klicke auf das kleine rote Icon neben „AlchemyBase“:



Das öffnet den alchemy_base.2da Ordner, die Datenbank, welche die Definitionen aller alchemistischen Items Inhaltstoffe enthält, die im Witcher vorhanden sind.

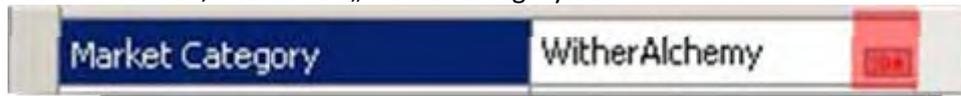
Economy Section:

• **Market Category.** Stellt die Kategorien eines Items fest, das im Geschäft vorhanden ist. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom Market Category Attribute und wähle die passende Auswahl aus. Diese Werte, Die du bei diesen Attributen zuweisen kannst, sind:

- *Alchemy, (alchemie)*
- *Alcohol, (Alkohol)*
- *Alcohol Strong, (Starker Alkohol)*
- *Armor, (Rüstung)*
- *Book, (Buch)*
- *Candy, (Süßigkeiten)*
- *Cloth, (Stoff)*
- *Dice, (Würfel)*
- *flower, (Blumen)*
- *Food, (essen)*
- *grease, (Schmierfett)*
- *herb, (Heilkraut)*
- *Illegal, (Verbotenes)*
- *Ingredient, (Inhaltstoffe)*

- *Monster-Ingredient, (Monster Inhaltstoffe)*
- *powder, (Pulver)*
- *SilverSword, (Silber-Schwert)*
- *torch, (Fackel)*
- *Trophy, (Trophäen)*
- *Value, (Kostbarkeit)*
- *Weapon, (Waffen)*
- *WeaponUp* – Waffen Verbesserungen,
- *WitcherAlchemy* – Witcher's' Alchemie Items.

Klicke auf das kleine rote Icon, was neben „Market Category“ ist:



Dieses öffnet die marketcat.2da Ordner, die Datenbank, die Item-Kategorien sowie Minimale und maximalen Geschäfts Preise enthält

- **Custom Price.** *Stellt den kundenspezifischen Preis eines Items ein. Preise werden in Orens angegeben (die Währung die verwendet wird im Witcher). Um einen Preis einzustellen, klicke auf der rechten Seite der Spalte auf Custom Price Attribute und drücke Enter.*

- **Restore Type.** *Stellt fest, welches Item in einem Geschäft wieder erscheinen wird, z.B. nachdem es von Geralt gekauft wurde. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber von dem Restore Type Attribute und wähle die passende Auswahl aus. Diese Werte, die Du in diesem Attribut zuweisen kannst, sind:*

- Next Day – Das item erscheint am nächsten Tag wieder

- Module Reload – nach dem das Modul neu geladen hat erscheint wieder das Item (d.h im selben Augenblick wenn Geralt zu einen anderem Spielabschnitt schreitet),

- Phase Change - das Item erscheint nur nach einer Änderung der Spielphase wieder. Die Spielphasen werden in einem anderen Kapitel des Handbuches beschrieben,

- Never – Item wird nicht wieder erscheinen

Weapon Section:

Attribute in dieser Gruppe sind nur relevant, wenn ein bestimmtes Item eine Waffe ist

- **Weapon Type.** *Stellt die Art der Waffe fest (wenn das Item eine Waffe ist). Um diese Einstellungen zu ändern, klicke auf die rechte Spalte gegenüber von Weapon Type Attribute und wähle die passende Auswahl aus. Diese Werte, die Du bei diesem Attribut zuweisen kannst, sind:*

- *WitcherSteelSword, (Witcher-Stahlschwert)*

- *WitcherSilverSword, (Witcher-Silberschwert)*
- *Fistfight, (Faustkampf)*
- *Monster – (Waffen die von Monstern benutzt würden wie z.B Krallen etc.,)*
- *Bow, (Bogen)*
- *Crossbow, (Armbrust)*
- *TwoHander –(Zwei Hände Waffen)*
- *ShortWeapon, (Kurze Waffe)*
- *Shield, (Schild)*
- *Pole, (Stab)*
- *Gabriel, (?)*
- *Sling, (schleuder)*
- *ThrownArms, (geworfene Waffen)*
- *Improvised, (Improvisiert)*
- *Dagger, (Messer) - PoleLight, (leichter-stab)*
- *SpellMele – (Magie im Nahkampf)*
- *SpellRanged – (lang reichende Magie)*
- *Not a weapon, (keine Waffe)*

• **Automatic Weapon Selection.** *Stellt fest, ob eine Waffe automatisch ausgewählt wurde. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom Automatic Weapon Selection Attribute und wähle die passende Auswahl aus. Diese Werte, die Du bei diesem Attribut zuweisen kannst, sind:*

- *False – Waffe wird nicht Automatisch ausgewählt,*
- *True – Waffe wird automatisch ausgewählt.*

• **Unarmed Combat Weapon.** *Stellt fest, ob eine Waffe während des unbewaffneten Kampfes benutzt wurde. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom Unarmed Combat Weapon Attribute und wähle die passende Auswahl aus. Diese Werte, die Du bei diesem Attribut zuweisen kannst, sind:*

- *False –Waffe wird nicht während des unbewaffneten Kampf benutzt ,*
- *True – Waffe wird benutzt im unbewaffneten Kampf,*

• **Weapon Ability Self.** *Stellt die Fähigkeiten einer Waffe ein. Um die Fähigkeiten einzustellen, klicke auf das kleine rote Icon, die auf der rechten Spalte gegenüber vom Weapon Ability Self Attribute abgelegt sind:*



Die Waffenfähigkeiten werden in einem anderen Kapitel dieses Handbuches beschrieben.

- **Weapon Ability Opponent.** Stellt die Fähigkeiten ein, die ein Gegner während des Kampfes hat, wenn er ein Item verwendet. Um diese Fähigkeiten einzustellen, klicke auf das kleine rote Icon, auf der rechten Spalte gegenüber vom opposite the Weapon Ability Opponent Attribute das dort abgelegt ist:



Die Waffenfähigkeiten der Gegner werden in einem anderen Kapitel dieses Handbuches beschrieben.

- **Weapon Slot.** Stellt ein, in welchem Slot diese Waffe gehalten wird. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom Weapon Slot Attribute und wähle die passende Auswahl aus. Das wird in einem anderen Kapitel dieses Handbuches beschrieben.
- **Override Attack Distance.** Stellt den Abstand ein, in dem ein Angriff, während ein Item verwendet wird, stattfinden kann. Um Den Abstand einzustellen, klicke das kleine rote Icon an, auf der rechten Spalte gegenüber vom, Override Attack Distance Attribute und gebe den passenden Wert ein und drücke Enter.

Inventory Section:

- **Has Inventory.** Stellt ein, ob ein Item einen Warenbestand hat. Um diese Einstellung zu ändern, klicken auf die rechte Spalte gegenüber vom Has Inventory Attribute und wähle die passende Auswahl aus. Diese Werte, die Du bei diesem Attribut zuweisen kannst, sind:

- *False* – Das Item hat keinen Warenbestand,

- *True* – Das Item hat einen Warenbestand.

- **Items.** Stellt ein, welche Items in einem Warenbestand gefunden werden können. Um Items Hinzuzufügen, klicke auf das kleine rote Icon, die auf der rechten Spalte gegenüber vom Items Attribute gefunden wird:



Die Warenbestandkreationen werden in einem anderen Teil dieses Handbuches beschrieben.

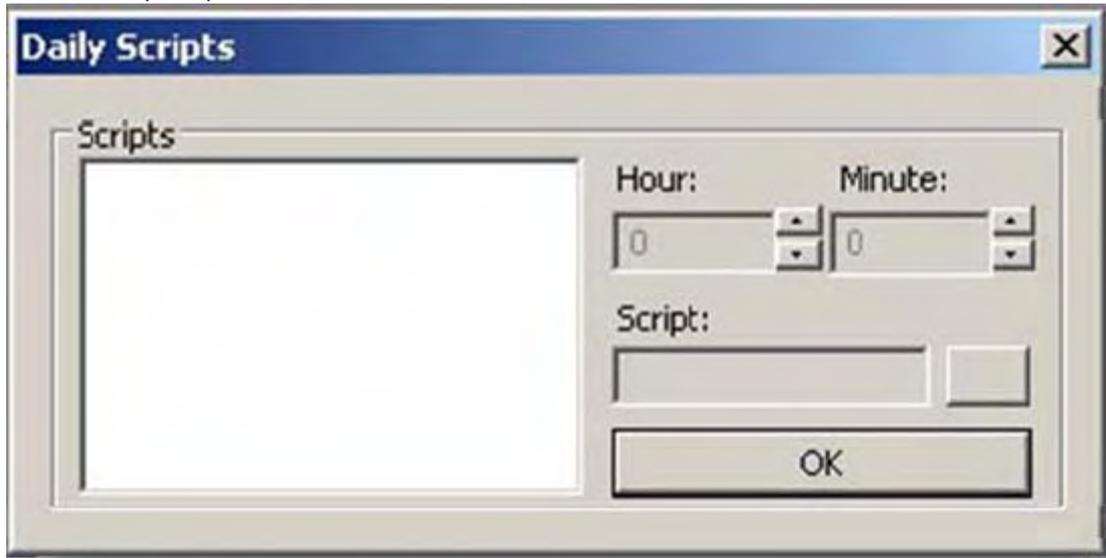
Scripts Section:

Diese Attribute in diesem Abschnitt stellen fest, welche Scripte aktiviert werden, wenn ein Ereignis auftritt, z.B. wenn Geralt ein Item benutzt

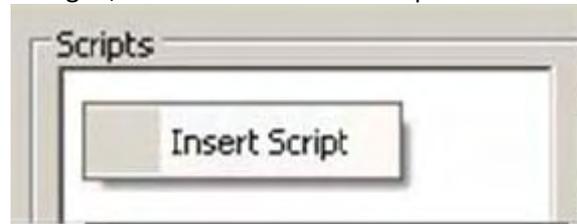
- **Daily Scripts.** Dieser Script wird täglich zu einer bestimmten Zeit gestartet. Das verweist selbstverständlich auf Die Spielzeit im Spiel. Um das passenden Script Hinzuzufügen und seine Aktivierung Zeit zu definieren, klicke auf das kleine rote Icon, die auf der rechten Spalte gegenüber vom Daily Scripts Attribute gefunden wird:



Das öffnet das Daily Scripts Fenster:



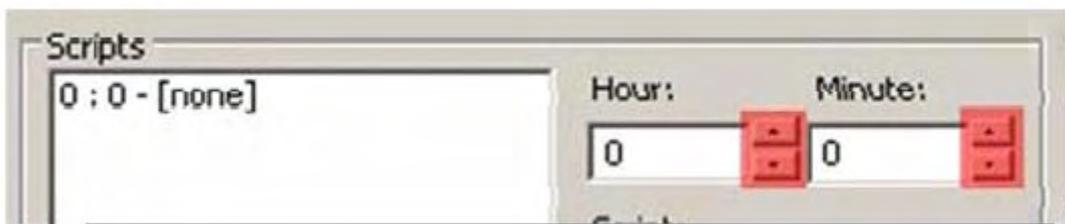
Um das passende Script hinzuzufügen, Rechtsklicke auf das Script und wähle „Insert Script“ vom Menü aus:



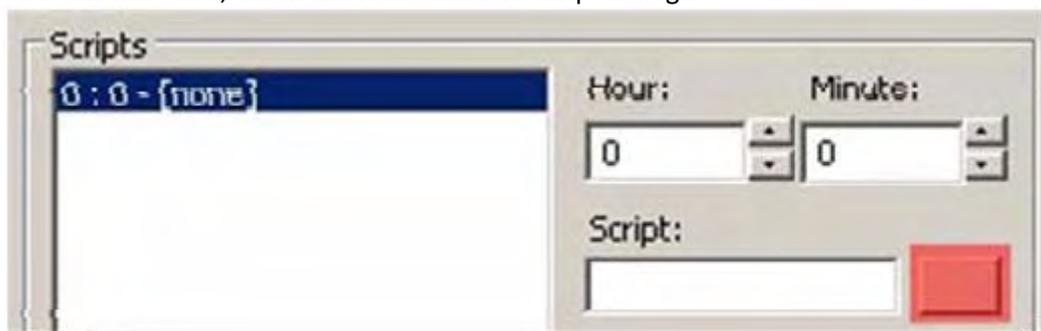
Klicke jetzt auf das vorher Hinzugefügte Script und stelle die Zeit ein, an dem der Script gestartet wird. Trage die gewünschten Werte der Stunden und Minuten ein:



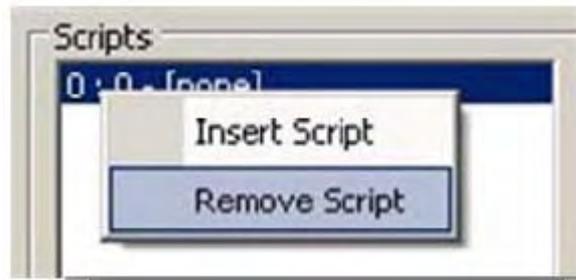
Die Stunden und Minuten können auch mit den kleinen Pfeilen eingestellt werden das in der Mitte der Stunde zu finden ist:



Klicke jetzt auf das Kästchen, das in der Nähe bei dem Scripten liegt:



Das öffnet das Standard Windows Fenster, das für Ladeordner benutzt wird. Gehe zum Data\Scripts Ordner (das alle Scripte im Spiel enthält), wähle den Script aus um geladen zu werden und klicke auf öffnen. Auf diese Weise wird der geladene Script täglich an der voreingestellten Zeit gestartet. Zunächst klicke auf O.K. im Script-fenster. Selbstverständlich kannst du mehr solcher Scripte Hinzufügen. Wenn Du ein Script entfernen möchtest, wähle sie im Scriptfenster aus mit einem Rechtsklick. Und wähle „Remove Scripts“ vom Menü:



- **OnIsUsable.** Die Scripte werden Aktiviert wenn ein Spezifiziertes Items benutzt wird.
- **OnJoinAttempt.** Dieses Script wird aktiviert, wenn Versucht wird, ein spezifiziertes Item zu kombinieren das erstellt worden ist.
- **OnUsed.** Dieses Script wird Aktiviert wenn ein Spezifiziertes Item verwendet wird
- **OnUserDefined.** Dieses Script wird definiert vom User.
- **OnPutIntoInventory.** Dieser Script wird aktiviert, wenn ein spezifiziertes Item im Warenbestand eingesetzt wird.
- **OnRemovedFromInventory.** Dieser Script wird aktiviert, wenn ein spezifiziertes Item im Warenbestand entfernt wird.

Um einen passendes Script zuzuweisen, klicke auf das kleine gelbe Icon, in der Spalte gegenüber vom Event. Um Z.B ein Script mit dem OnUsed Event zuzuweisen. Klicke auf das gelbe Icon gegenüber vom Attribute:



Das öffnet das Standard Windows Fenster, das für die Speicherung der Dateien benutzt wird. Gehe zum Data\Scripts Ordner (dass alle Scripte des Spiels enthält), wähle den passende Ordner Klicke Ihn an, und öffne es.

Physics Section:

Diese Attribute die in diesem Abschnitt zu finden sind. Sind für die Physik eines Items verantwortlich.

- **Mass.** Stellt Item Masse ein. Um die Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom Mass Attribute und wähle die passende Auswahl aus.
Diese Werte, die Du diesem Attribut zuweisen kannst, sind:

- Feather, (feder)
- Candle, (Kerze)
- Sword, (Schwert)
- Plank, (Steg)
- Axe, (Axt)

- *Heavy_Axe, (starke Axt)*
- *Barrel, (Fass)*
- *Stone, (stein)*
- *Great_Stone, (bedeutender Stein)*
- *Super_heavy, (Super Schwer)*
- *Hardcore_heave, (?)*
- *BigBarrel, (Großes Fass)*
- *WaterBarrel Key. (Wasserfass schlüssel)*

Klicke auf das kleine rote Icon, in der Nähe von der Selection Liste:



Das öffnet den phy_mass.2da Ordner, die Datenbank, die Definitionen der Masse enthält.

• **Material.** Bestimmt die materielle Zusammensetzung eines Items. Um dies Einstellung zu ändern, klicke auf die rechte Spalte das gegenüber liegt vom Material Attribute und wähle dann die passende Auswahl aus. Die Werte, die Du in diesem Attribut zuteilen kannst, sind:

- *Wood, (Holz)*
- *Stone, (stein)*
- *Gum, (Gummi)*
- *Metal, (Metall)*

Klicken Sie auf das kleine rote Icon, die in der Nähe von der Selection-Liste zu finden ist:



Das öffnet den phy_material.2da Ordner, eine Datenbank, die Definitionen der Materialien enthält

• **Primitive.** Bestimmt einen festen Rahmen, in dem ein Item für die Zwecke des Physik-Systems des Spiels beschrieben wird. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom Primitiven Attribute und wähle die passende Auswahl aus. Die Werte, die Du bei diesem Attribut zuteilen kannst, sind:

- *Box (cube), (Behälter Würfel)*
- *Cylinder, (Zylinder)*
- *Sphere, (Einflussbereich)*
- *Custom* – ein gewählter solider Rahmen

• **Aard only.** Stellt fest, ob ein Item durch das Aard Zeichen beeinflusst werden kann. Um die Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom Aard only Attribute und wähle die passende Auswahl aus. Die Werte, die Du bei diesem Attribut zuweisen kannst, sind:

- *False*, Items werden durch alle Events beeinflusst, die sich auf das Physiksystem des Spiels beziehen.

- *True* – Item wird durch das Aard Zeichen Beeinflusst

• **Affected by Wind.** Bestimmt, ob ein Item durch den Wind getroffen werden kann. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom Affected by wind Attribute und wähle die passenden Attribut-Option aus.

- *False*- Items werden nicht durch den Wind beeinflusst

- **True** – Item wird durch den Wind in mit einer Übereinstimmung der Physik beeinflusst, das für das Item eingestellt wurde.

• **Sound enabled.** Stellt ein, ob wir einen Ton hören werden, wenn ein Item durch das Physiksystem des Spiels beeinflusst wurde. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom Sound enabled Attribute und wähle die passende Wahl:

- *False* – Sound wird abgeschaltet

- *True* – Sound wird angeschaltet.

• **Sound type.** Stellt Die Art des Tones Fest die wir, hören werden, wenn ein Item durch das Physiksystem des Spiels beeinflusst wurde. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom Sound type Attribute und wähle die passende Wahl:

- *Basket, (Teller)*

- *Candlestick, (Kerzenhalter)*

- *Ceramic_bottle, (Keramik-Flasche)*

- *Pillow, (Kissen)*

- *Shield, (Schild)*

- *Skull, (Totenkopf)*

- *Stone_block, (Steinblock)*

- *Wood_barrel, (Holz fass)*

- *Wood_box, (Holzkiste)*

- *Wood_chair, (Holzstuhl)*

- *Body, (Körper)*

- *Sword, (Schwert)*

- *Axe, (Axt)*
- *Body_animal, (Körper der Tiere)*
- *HeavyGolem. (Schwerer Golem)*

Klicke auf das kleine rote Icon, das in der Selection-Liste zu finden ist:



Das öffnet den phy_sound.2da Ordner, die Datenbank, das die stichhaltigen Definitionen und die Sound-Skala enthält.

- **Asset database.**
- **Asset name.**
- **Joint strength.** *Stellt die Stärke ein, mit denen Die Items miteinander Agieren werden. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom Joint strength Attribut und wähle die passende Auswahl aus. Die Werte, die Du bei diesem Attribut zuweisen kannst, sind:*

- *Unbreakable (Unzerstörbar)*
- *Small (Gering)*
- *Medium (Medium)*
- *Big (dick)*

Klicke auf das kleine rote Icon, das in der Nähe von der Selection Liste zu finden ist:



Das öffnet den phy_joint.2da Ordner, die Datenbanken für, die gemeinsame Stärke Definitionen enthält.

- **Destructable.** *Stellt ein, ob ein Item zerstört werden kann. Um diese Einstellung zu ändern, klicke auf diese rechte Spalte gegenüber vom Destructible Attribut und wähle die passende Auswahl aus. Die Werte, die Du bei diesem Attribut zuweisen kannst, sind:*
- *False* – Item kann nicht zerstört werden,
- *True* – item kann zerstört werden,
- **Respawn.** *Stellt ein, ob ein Item respawnnt(Wiederscheinen) kann, nachdem es zerstört worden ist (erscheinen wird es immer an der derselben Stelle). Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom Respawn Attribut und wähle die passende Auswahl aus. Diese Werte, die Du bei diesem Attribut zuweisen kannst, sind:*
- *False* - Item wird nach Zerstörung nichtmehr wiederscheinen.
- *true* - Item wird nach Zerstörung wieder erscheinen.

Replace Subsection:

Die Attribute in diesem Abschnitt sind für das alte Modell 3D eines Items mit einem Item ersetzbar wenn z.B., das Item zerstört ist, Das ist sehr nützlich, weil ein Item durch einen wreckage(Bruch) ersetzt werden sollte, nachdem es zerstört wurde.

- **Asset database.**
- **Asset name.**
- **Joint strength.** Bestimmt die Kraft, wie Stark Items zusammen gebunden sind.
- **Model.** Stellt das neue Modell 3D des Items ein, das dieses alte ersetzen soll. Um ein Modell auszuwählen, klicke auf das kleine gelbe Icon, die auf der rechten Spalte gegenüber vom Model Attribute zu finden sind:



Das öffnet das Standard Windows Fenster, das für den Ladeordner benutzt wird. Gehe zum Data\Meshes Ordner, und wähle das Modell Deiner Wahl (eine mdl Ordner) aus und klicke auf öffnen.

- **Replace after time.** Stellt die Zeit ein, wann die 3D Modelle ausgetauscht werden sollen. Um eine Zeit einzustellen, klicke auf die rechte Spalte gegenüber vom Replace after time Attribute und trage einen Wert ein.
- **Replace on collision.** Bestimmt, ob ein Modelltausch nach einer Kollision(Zusammenstoß) zwischen den Items stattfinden wird. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom Replace on collision Attribute und wähle die passende Auswahl aus. Diese Werte, die Du bei diesem Attribut zuteilen kannst, sind:

- *Off* – der Ersatz nach der Kollision wird unbrauchbar gemacht,

- *If_in_sim*,

- *Small* –eine kleine Kollision,

- *Medium* –eine Mittlere Kollision,

- *Big* – eine sehr große Kollision,

- *AARD* – Nur nachdem dieses Aard Zeichen benutzt wurde.

Klicken Sie auf das kleine rote Icon, die in der Nähe von der Selection-Liste zu finden ist:



Das öffnet den phy_replace_col.2da Ordner, diese Datenbank, enthält die Definitionen der Zusammenstoß-stärke.

Destructable Options Subsection:

- **Destroy after time.** die Zeit nachdem einen Gegenstand zerstört wird. Um diesen Wert einzustellen, klicke auf die rechte Spalte gegenüber vom Destroy after time Attribute und trage den Wert ein.
- **Start destroy counter.** Stellt ein, wann der Countdown, des spezifizierten Items anfängt sich zu zerstören. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom Start destroy counter Attribute und wähle die passende Auswahl an:

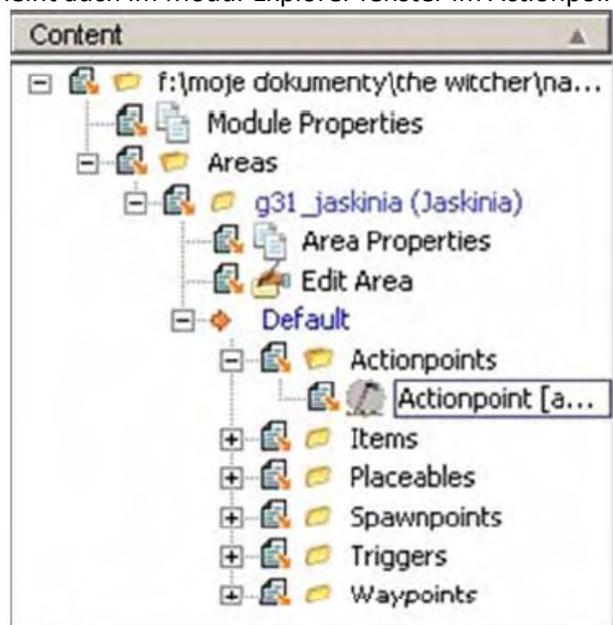
- *If_in_sim.*
- *Small_collision*, – eine kleine Kollision.
- *Medium_collision* – eine Mittlere Kollision.
- *Big_collision*- eine sehr große Kollision.



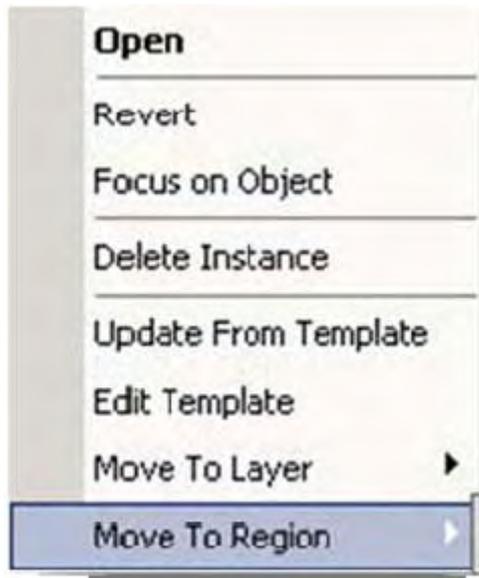
Dieses Werkzeug wird benutzt, um Aktions-Punkte in einem Bereich hinzuzufügen. Aktions-Punkte, an denen die Positionen der Charaktere in den verschiedenen Tätigkeiten agieren werden, wurde in einen anderen Kapitel des Handbuches beschrieben. Aktionspunkte werden genauso wie Placeables hinzugefügt. Klicke auf den abwärts Pfeil, der nahe bei dem Aktions-Punktwerkzeug zu finden ist. Im auserwählten Fenster wähle appropriate Action Point. Die Liste der neuen Aktions-Punkte kann auch benutzt werden. Klicke jetzt auf den Punkt, in dem der Aktionspunkt hinzugefügt werden soll:



Der neue Actionpoint erscheint auch im Modul-Explorer fenster im Actionpoints Unterbaum:



Dieselben Handlungen gelten als Handlungspunkte gemeint: Placeables. Klicke richtig auf die Handlungspunkte und suche das passende aus im Menü. Nein dieses Menü enthält keine Weiteren zusätzlichen befehle der Bewegungen zur Region:



Dieser Befehl verschiebt die Aktionspoints in einen anderen Bereich. Doppelklicke auf den Aktionspoint. Ein Eigenschaften-Fenster, welches die Attribute der Aktionspoints enthält, öffnet sich auf der rechten Seite des Schirmes:

Actionpoint [ap_standard]	
General	
Name	Actionpoint
Tag	ap_standard
Comment	
TemplateResRef	m0_ap_stand
Creature Uses Orientation	False
Action parameters	
Actions list	0 strings defined
Action placeable	
Region name	default
One Liner Dialog	
Dialog Category	0 strings defined
Waypoint Walk	
Waypoint Prefix	
Run	False
Position	
X	1456.046753
Y	1514.147339
Z	0.079862
Scripts	
On Used	

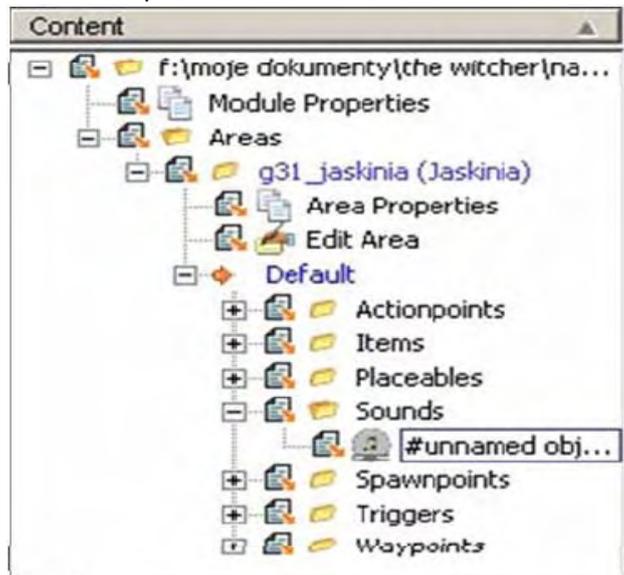
Diese Attribute haben wir bereits besprochen ich möchte sie hier nicht mehr wiederholen.



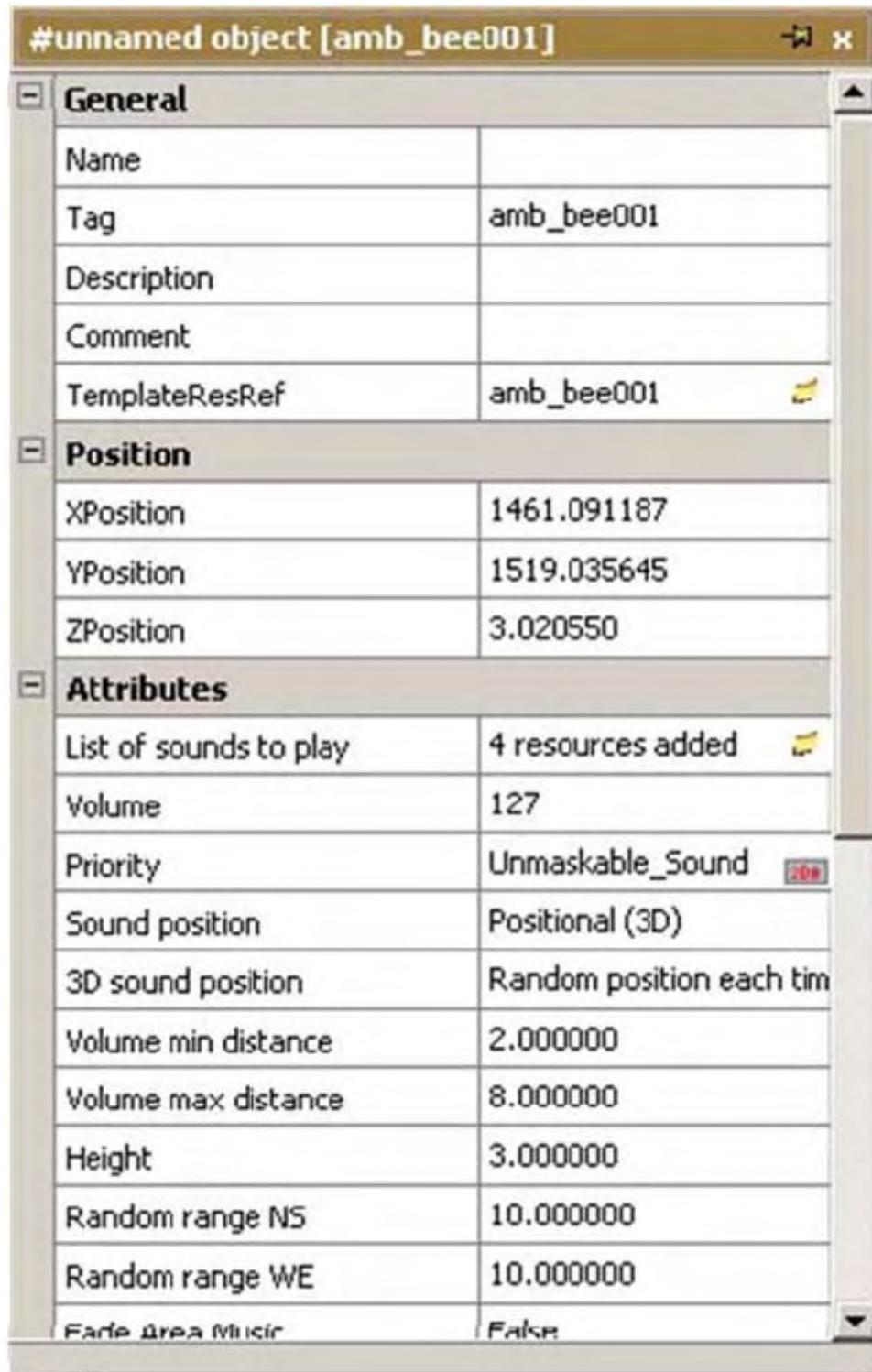
Dieses Werkzeug wird dazu benutzt, um Töne(Sounds) in den Bereichen hinzuzufügen. Der Ton bezieht sich auch auf die Musik im D'jinni Editor. Irgendein Ton, der kommt, kann hinzugefügt werden. Der Ton des Windes der einer Höhle, z.B.. Um ein Ton hinzuzufügen, klicke auf den abwärts Pfeil, der in der Nähe vom sound Tool zu finden ist. Vom Menü aus wähle New sound template command aus. im auserwählten Fenster wähle den passenden Ton aus. Diese Liste der vor kurzem benutzten Töne können auch benutzt werden. Klicke auf einen Bereich, in dem ein Ton hinzugefügt werden soll:



Der Sound erscheint auch im Modul-Explorer-fenster im Sound in der Baum-Verästelung:



Die gleichen Aktionen treffen auf den Sound hinsichtlich der Placeables zu. Rechtsklicke auf den Sound und wähle den passenden Befehl vom Menü aus. Dann Doppelklicke auf die Sounds Attributes. Ein Eigenschaften-Fenster, welches die Attribute der Sounds enthält, öffnet sich auf der rechten Seite des Schirmes:



Hier werden wir die jeweiligen Attribute betrachten:

General Section:

- **Name.** Enthält die Namen der Sounds. Dieser Name wird nicht im Spiel verwendet.
- **Tag.** Enthält die the identification tag of the sound. Tags werden hauptsächlich mit den Scripten benutzt. Mehr vom Scripten wird in einen anderen Kapitel beschrieben.
- **Description.** Enthält die Beschreibungen Der Sounds.

- **Comment.** Enthält einen Raum in, dem du eine Anmerkung schreiben kannst. Dies wird im Spiel nicht verwendet, das ist nur ein Informationsfeld. Du kannst schreiben was immer du möchtest.

- **TemplateResRef.** Enthält Schablonen für den Sound. Wenn Du zu einer anderen Schablone wechseln möchtest, klicke auf das kleine gelbe Icon, die auf der rechten Spalte gegenüber vom TemplateResRef Attribut zu finden sind:



Das öffnet das Standard Windows Fenster, das für den Ladeordner benutzt wird. Gehe zum Data\Templates\Sounds ordner (das alle stichhaltigen Ordnern enthält), wähle den passenden ordner aus und klicke es an, und klicke auf öffnen.

Position Section:

- **XPosition.** Enthält die Position des Sounds auf der X Mittellinie. Dieses wird automatisch jede Zeit aktualisiert, das die Position des Sounds im Bereich ändernd. Du kannst die Default Setting so lassen.

- **YPosition.** Enthält die Position des Sounds auf der y Mittellinie. Dieses wird automatisch jede Zeit aktualisiert, das die Position des Sounds im Bereich ändernd. Du kannst die Default Setting so lassen.

- **ZPosition.** Enthält die Position des Sounds auf der Z Mittellinie. Dieses wird automatisch jede Zeit aktualisiert, das die Position des Sounds im Bereich ändernd. Du kannst die Default setting so Lassen.

Attributes Section:

- **List of sounds to play.** Enthält die Liste der abgespielten Sounds. Um einen Sound zuzuweisen, klicke auf das kleine gelbe Icon, die gegenüber von der Liste der Sounds to Play Attribute liegt:



Das ressource list-Fenster öffnet sich. Dieses Fenster wurde vorher im Modul-attributs-abschnitt besprochen.

- **Volume.** Stellt Klangvolumen ein . um das Klangvolumen einzustellen, klicke auf die rechte Spalte gegenüber vom Volume Attribute an, um den passenden Wert einzugeben drücke danach Enter.

- **Priority.** Stellt die Priorität eines Sounds ein, der in den Relationen zu anderen Sounds abgespielt wird. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom Priority Attribute und wähle die passende Auswahl aus. Du kannst das kleine rote Icon auch anklicken, die in der Selection-Liste zu finden sind:



Das öffnet den prioritygroups.2da Ordner, diese Datenbanken enthalten, stichhaltige Einstellungen.

• **Sound Position.** Stellt die Position eines Sounds ein. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom Sound Position Attribute und wähle die passende Auswahl aus. Die Werte, die du bei diesem Attribut zuweisen kannst, sind:

- *Area – wide sound (2D),*

- *Positional (3D),*

- *Positional stereo (2D with position)*

• **3D sound position.** Stellt die Position eines 3D Sounds ein. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom 3D sound position Attribute und wähle die passende Auswahl aus. Die Werte, die Du bei diesem Attribut zuweisen kannst, sind:

- *Specific position – wird gesetzt vom Schöpfer des Moduls*

- *Random position each time it is the player– unterschiedliche Positionen des Sounds wenn es mal abgespielt wird*

• **Volume min distance.** Stellt das Klangvolumen auf minimale Distanz ein. um Das Volumen einzustellen, klicke auf die rechte Spalte gegenüber vom Volume min distance Attribute, gebe dazu den passenden wert ein und drücke Enter.

• **Volume max distance.** Stellt das Klangvolumen auf eine Maximale Distanz ein. um Das Volumen einzustellen, klicke auf die rechte Spalte gegenüber vom Volume max distance Attribute, gebe dazu den passenden wert ein und drücke Enter.

• **Height.** Bestimmt eine verlässliche höhe, um eine Höhe einzustellen klicke auf die rechte Spalte gegenüber vom Height Attribute, gebe den passenden Wert dazu ein und Drücke Enter.

• **Random range NS.** Enthält einen zufälligen verlässlichen Bereich in der Nord-Süd-Position um eine Position einzustellen. klicke auf die rechte Spalte gegenüber vom, *Random range NS* Attribute, und gebe dazu den passenden wert ein und drücke Enter.

• **Random range WE.** Enthält einen zufälligen verlässlichen Bereich in der Westen-Osten Position. um eine Position einzustellen. klicke auf die rechte Spalte gegenüber vom, *Random range WE* Attribute, und gebe dazu den passenden wert ein und drücke Enter.

• **Fade Area Music.** Stellt ein, ob eine Musik gespielt wird in einem Bereich, wenn, oder langsam zu verklingen. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom Fade Area Music Attribute und wähle dazu die passende Auswahl aus. Die Werte, die Du bei diesem Attribut zuweisen kannst, sind:

- *False – Die Musik soll nicht verklingen*

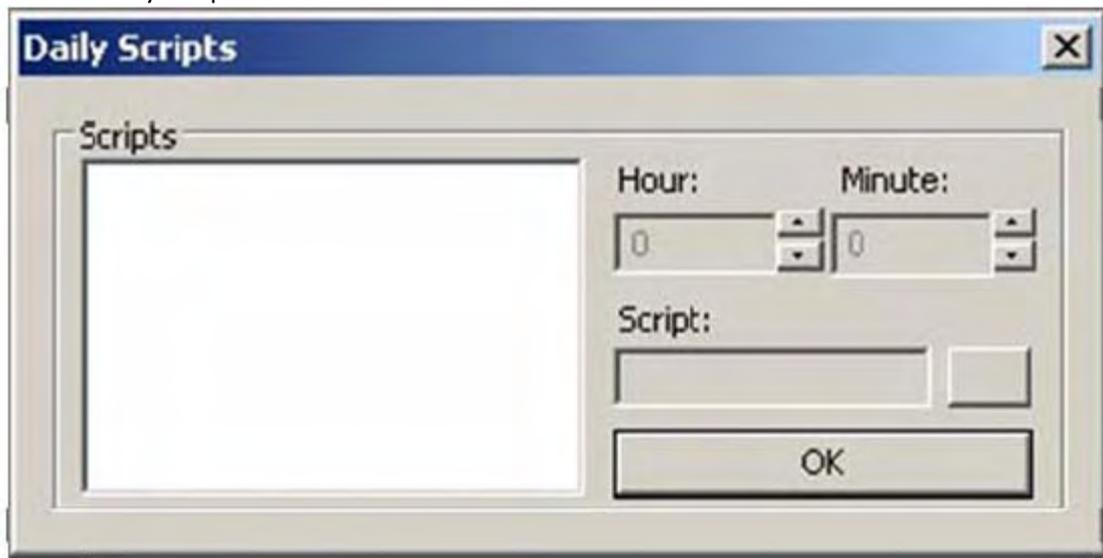
- *True – Die Musik soll verklingen(Verblassen)*

Scripts Section:

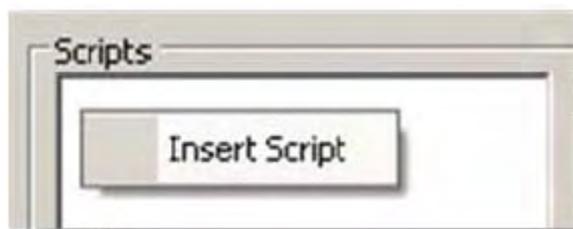
• **Daily Scripts.** Dieser Script wird täglich ab einer bestimmten Zeit gestartet. Um das passenden Script hinzuzufügen und die Aktivierungszeit zu definieren, klicke auf das kleine rote Icon, die in der rechten Spalte gegenüber vom Daily Scripts Attribute zu finden sind:



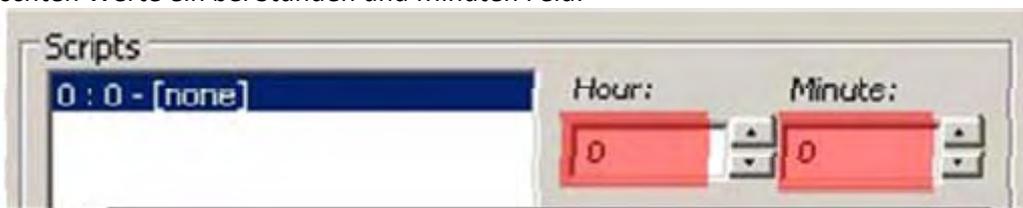
Dies öffnet das Daily-Script-Fenster:



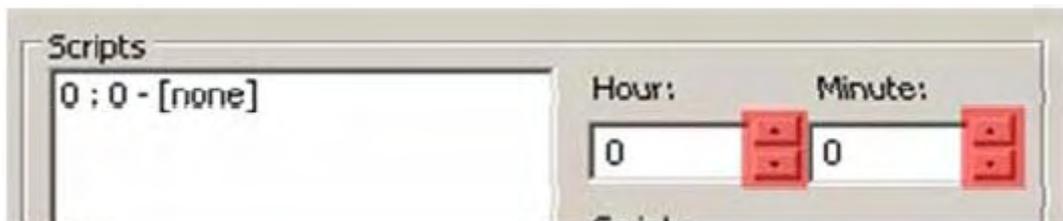
Um den passenden Script hinzuzufügen, Rechtsklicke auf den Script und wähle es vom Insert Script Menü aus:



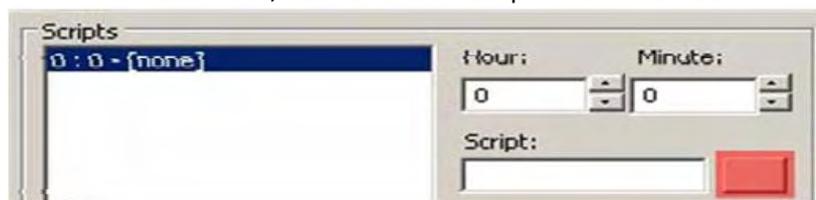
Klicke jetzt auf das vorher Hinzugefügte Script und stelle die Zeit ein, an dem der Script starten soll. Trage die gewünschten Werte ein bei Stunden und Minuten Feld:



Du kannst auch die Zeit mit den kleinen Pfeilen einstellen, die in der Nähe vom Stunden und Minutenfeld Zufinden sind:

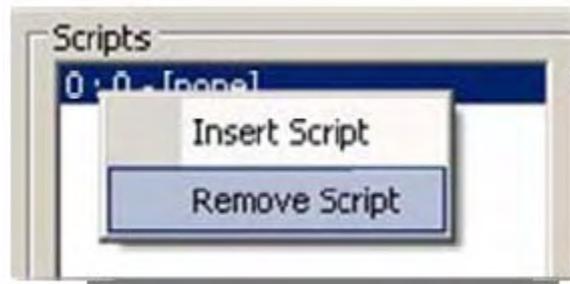


Klicke auf das kleine rote Kästchen an das, das neben den Scripten zu finden ist:



Das öffnet das Standard Windows Fenster, das für den Lade Ordner benutzt wird. Gehe zum Data\Scripts Ordner (das alle Scripte im Spiel enthält), wähle ein Script aus das geladen werden soll und klicke auf öffnen. Auf diese Weise wird das geladene Script täglich an der eingestellten Zeit gestartet. Zunächst klicke auf O.K. im Script-fenster. Selbstverständlich kannst Du mehr solcher Scripte Hinzufügen. Wenn Du ein Script entfernen möchtest, wähle es im Script-Panel aus und Drücke mit der rechten Maustaste. Und wähle im

Menü "Remove Script" aus:



Advanced Section:

• **Active.** Stellt ein, ob ein Sound aktiv ist, und sollte in Abfolge abgespielt werden. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom Active Attribute und wähle dazu die passende Auswahl aus. Diese Werte, die Du bei diesem Attribut zuweisen kannst, sind:

- *False* – Sound ist Inaktiv

- *True* – Sound ist Aktiv

• **When to Play.** Stellt ein, wann ein Sound abgespielt werden soll, Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom When to Play Attribute und wähle dazu die passende Auswahl aus. Die Werte, die Du bei diesem Attribut zuweisen kannst, sind:

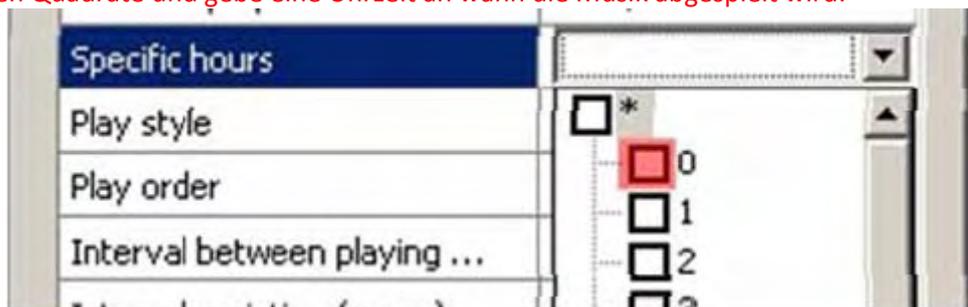
- *Specific hours, (bestimmte Uhrzeit)*

- *Day, (Tag)*

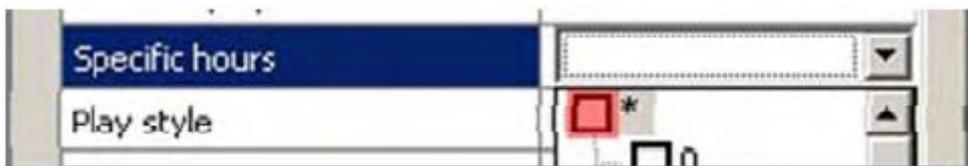
- *Night, (Nacht)*

- *Always. (Immer)*

• **Specific hours.** Stellt einen bestimmten Zeitpunkt ein, wann die Musik gespielt wird, wenn, man in den Attributen einstellt auf welche bestimmte Stunde Die Musik abgespielt wird. Um die Zeit einzustellen, klicke auf die kleinen Quadrate und gebe eine Uhrzeit an wann die Musik abgespielt wird:



Einige Stunden können auch ausgewählt werden. Um alle Stunden auszuwählen, klicke auf das kleine Quadrat nahe bei dem Sternchen:



Jedoch kannst du auch einfach einstellen, wann ein Attribut immer spielt, wenn Du einen Sound wünschst der ununterbrochen abgespielt wird oder stundelang abgespielt wird.

• **Play style.** Stellt den sound Play style ein. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom *Play style* Attribute und wähle dazu die passende Auswahl aus:

- *Play once* – Sound wird einmal abgespielt

- *Repeat* – Sound wiederholt sich

• **Play order.** Bestimmt ob ein Sound, in einer Ordnung abspielt. Das die Ressourcen aus der Liste von den Sounds verwenden, das dieses Attribut abspielen wird. Es können nämlich auch mehrere Sounds gleichzeitig abgespielt werden. Den Sounds können auch in einer besonderen Ordnung abgespielt werden. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom Play order Attribute und wähle dazu die passende Auswahl aus. Die Werte, die Du bei diesem Attribut zuteilen können, sind:

- *Sequential* – Sounds werden der Reihe nach abgespielt, den sie haben im Ressourcen-fenster ein Attribut gelegt das die Sounds so abspielen lässt.

- *Random* – Sounds werden nachdem Zufallsprinzip abgespielt

• **Interval between playing Sounds (msec.)**. Stellt die Distanz zwischen den Sound sein. Der Wert wird in Millisekunden angegeben. Um den Wert einzustellen, klicke auf die rechte Spalte gegenüber vom, Interval between playing Sounds (msec.) Attribute trage den passenden Wert ein und drücke Enter.

• **Interval variation (msec.)**. Stellt die Intervalle zwischen Tönen ein. Der Wert wird in Millisekunden angegeben. Um den Wert einzustellen, klicke auf die rechte Spalte gegenüber vom, Interval variation (msec.) attribute, trage den passenden Wert ein und drücke Enter.

• **Volume variation**. Stellt die Volumenveränderung ein. Um den Wert einzustellen, klicke auf die rechte Spalte gegenüber vom Volume variation attribute, Gebe den passenden Wert ein und drücke Enter.

• **Pitch variation**. stellt den Taktabstand ein. Um den Taktabstand einzustellen, klicke auf die rechte Spalte gegenüber vom opposite the Pitch variation Attribute, Gebe den passenden Wert ein und drücke Enter.

• **Seamlessly looping**. Stellt ein, ob der Sound – Looping nahtlos oder unterbrochen bei einem Event sein soll. Um diese Einstellung zu ändern, klicken auf die rechte Spalte gegenüber von dem Seamlessly looping Attribute und wähle die passende Auswahl aus. Die Werte, die Du bei diesem Attribut zuweisen kannst, sind:

- *False* – Sound liegt in einer Schleife,

- *True* – die Soundschleife kann unterbrochen werden



Dieses Werkzeug wird benutzt, um einen Start-Punkt hinzuzufügen. Da wir bereits einen Start-Punkt hinzugefügt haben, gehen wir auf weitere Details nicht ein.

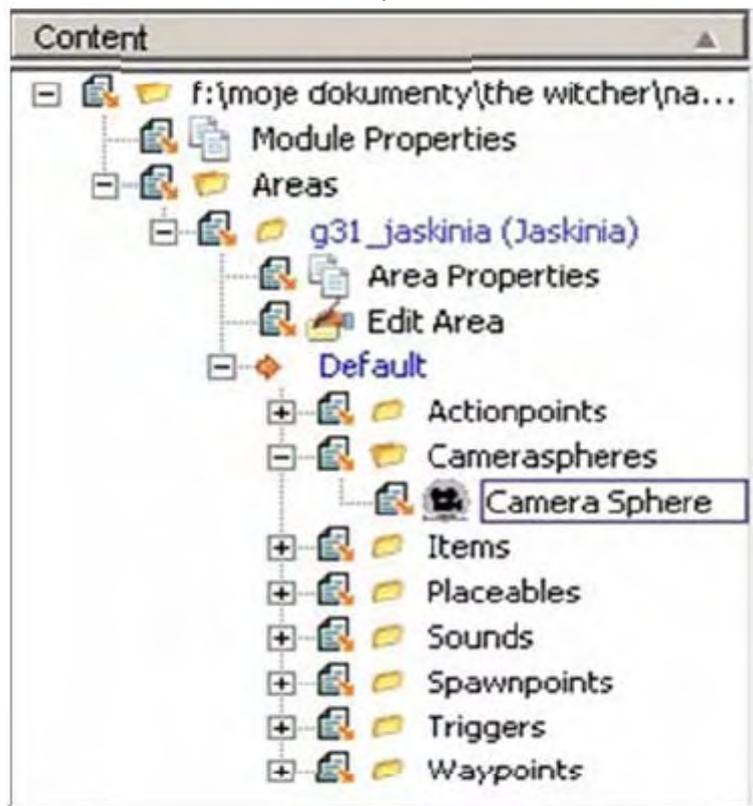
Als eine Art Erinnerung ist der Start-Punkt an dem Ort wo Geralt als Spieler erscheint im Modul Mit anderen Worten ist es das wo das Spiel beginnt, Nachdem man in einem Bereich einen Start-Punkt hinzufügt wird ist es nicht möglich seinen zu entfernen, Er kann nur an eine andere Stelle bewegt oder in einen anderen Bereich hinzugefügt werden.



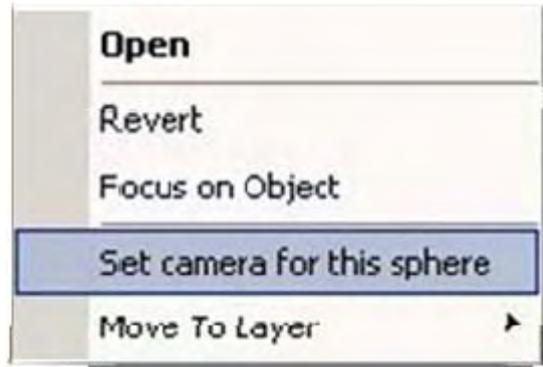
Dieses Werkzeug wird benutzt, um einen Kamerabereich hinzuzufügen es wird damit gezeigt wohin die Kamera "schaut". Um einen Bereich hinzuzufügen, klicke auf das Kamerabereichswerkzeug und klicke dann auf den Bereich in dem der Bereich hinzugefügt werden soll. Der Kamerabereich erscheint jetzt im Bereich:



Der neue Kamerabereich erscheint auch im Modul-Explorer im Kamerabereichsunterbaum:



Rechtsklicke auf die Camera Sphere im Modul-Explorer Fenster. Das öffnet ein Menü, das die setkamera für diese Sphere Befehlszeile enthält:



Wähle diesen Befehl aus, um die Kamera hinzuzufügen:



Die Kamera zeigt direkt auf diesen Bereich, den wir Hinzugefügt haben. Merke Dir, dass die Kamera in der Position erscheint, die im Editierten Bereich eingestellt worden ist. Wenn Du die Kamera in einer anderen Richtung zeigen möchtest, musst Du die Ansicht zuerst einstellen, dann, die auserwählte Setkamera für dieses Bereich. Doppelklicke auf den Kamerabereich. Ein Eigenschaften-Fenster, welches die Attribute der Camera Sphere enthält, öffnet sich auf der rechten Seite des Schirmes:

Camera Sphere	
General	
Comment	
Position	
PositionX	1452.486206
PositionY	1519.750244
PositionZ	2.477715
Camera	
PositionX	1458.804688
PositionY	1516.064087
PositionZ	3.953176
Attributes	
Radius	2.000000

Hier betrachten wir die jeweiligen Attribute:

General Section:

- **Comment.** Enthält einen Raum in dem sie eine Anmerkung schreiben können. Dies wird im Spiel nicht verwendet, das ist nur ein Informationsfeld. Du kannst schreiben was immer du möchtest.

Position Section:

- **XPosition.** Enthält die Position des Kamerabereichs auf der X Mittellinie. Dieses wird automatisch jede Zeit aktualisiert, Du kannst die Position des Bereichs im Bereich ändern. Du kannst die Default Setting so lassen.

- **YPosition.** Enthält die Position des Kamerabereichs auf der Y Mittellinie. Dieses wird automatisch jede Zeit aktualisiert, Du kannst die Position des Bereichs im Bereich ändern. Du kannst die Default Setting so lassen.

- **ZPosition.** Enthält die Position des Kamerabereichs auf der Z Mittellinie. Dieses wird automatisch jede Zeit aktualisiert, Du kannst die Position des Bereichs im Bereich ändern. Du kannst die Default Setting so lassen.

Camera Section:

- **XPosition.** Enthält die Position der Kamera auf der X Mittellinie. Dieses wird automatisch jede Zeit aktualisiert, da eine Kamera in einem Bereich zugewiesen wurde.

- **YPosition.** Enthält die Position der Kamera auf der y Mittellinie. Dieses wird automatisch jede Zeit aktualisiert, da eine Kamera in einem Bereich zugewiesen wurde.

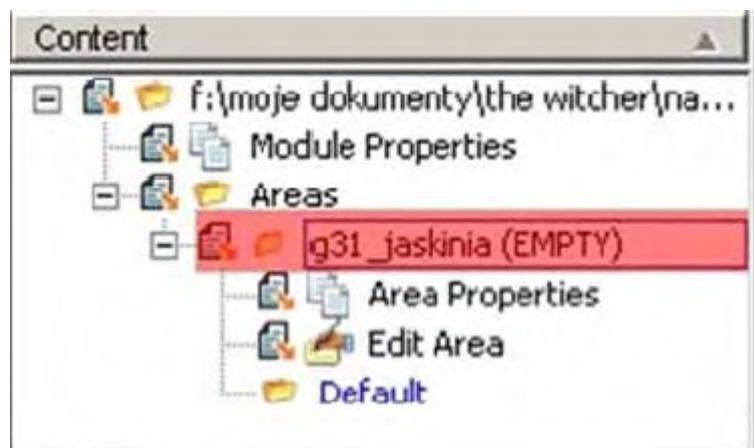
- **ZPosition.** Enthält die Position der Kamera auf der z Mittellinie. Dieses wird automatisch jede Zeit aktualisiert, da eine Kamera in einem Bereich zugewiesen wurde.

Attributes Section:

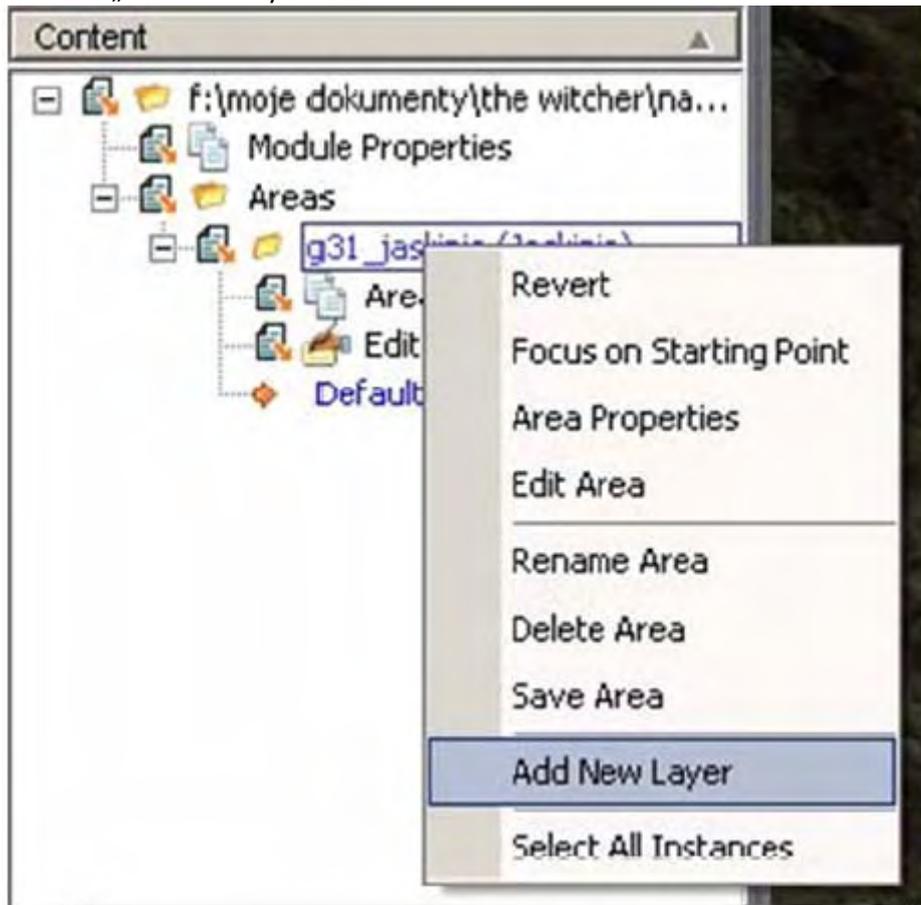
- **Radius.** Stellt den Radius des Bereichs ein. Um den Radius einzustellen, klicke auf die rechte Spalte gegenüber vom dem Radius-Attribut, und trage den passenden Wert dazu ein und Drücke Enter.

Das ist für das Camera-Sphere Attributes.

Lass uns jetzt etwas Beleuchtung hinzufügen mit einem neuen Layer des Bereichs. Um eine neuen Layer Hinzuzufügen, gehe zum Modul Explorer und Rechtsklicke auf dem Bereich in dem es hinzugefügt werden soll:



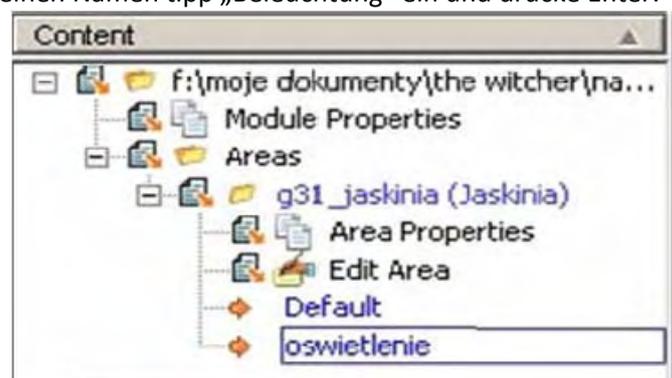
Dann wähle vom Menü „Add New Layer“ aus:



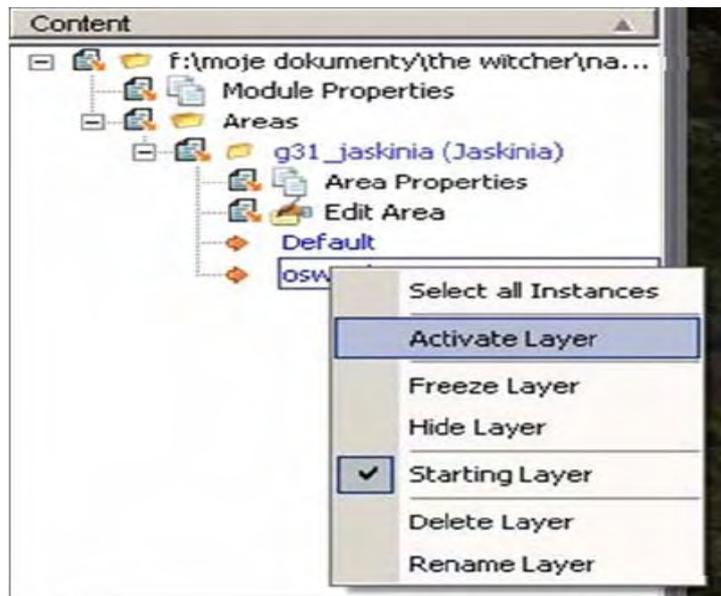
Dies fügt ein neuen Layer(Schicht) im diesen Bereich zu:



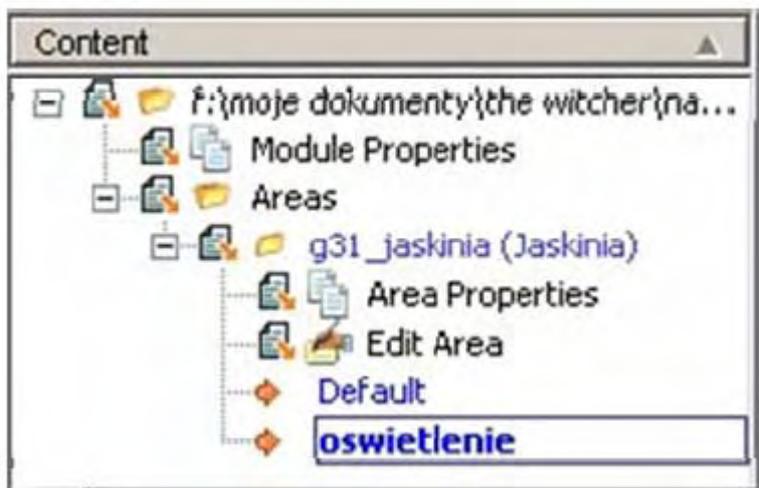
Der Neue Layer braucht einen Namen tipp „Beleuchtung“ ein und drücke Enter:



Jetzt muss der Layer aktiviert werden. Erinner dich daran, dass eine Schicht aktiv sein muss, wenn du die Items hinzugefügt hast. Andernfalls werden sich die Items oben in der Zeile beenden, egal welcher Layer dann wirklich aktiv war. Um ein Layer zu aktivieren, Rechtsklicke auf den Layer um es im Menü zu aktivieren:



Die Schicht wird aktiv und sichtbar:

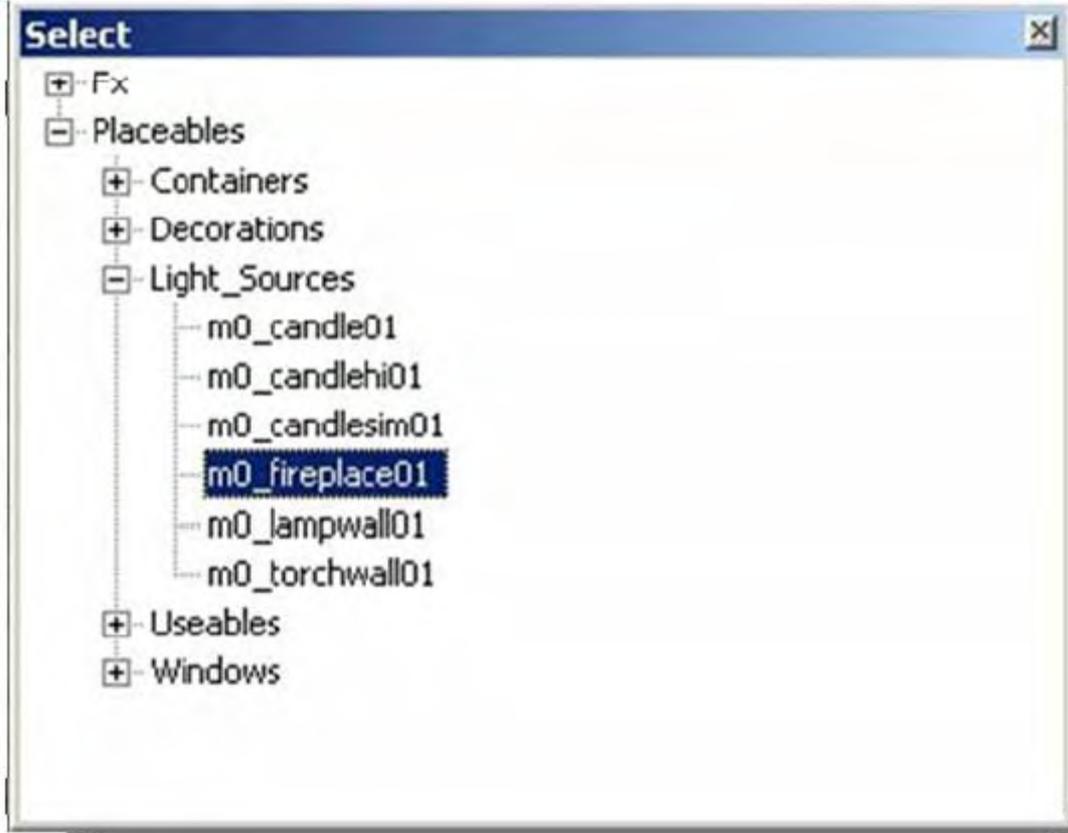


Jetzt können wir Items diesem Layer hinzufügen. Lass uns einige Feuerstellen beim Start-Punkt hinzufügen. Um eine Lichtquelle hinzuzufügen muss das Placeables Werkzeug benutzt werden.

Finde dieses Item m0_fireplace01 um Es zu finden klicken auf den Abwärts-Pfeil in der Nähe von den Placeable(platzierbar) Items:



Klicke auf die auserwählte neue Placeable Schablone. Öffne dann das Placeable - den Light-Sources Baum und Doppelklicke auf m0_fireplace01 im auserwählten Fenster:



Bemerke, dass diese fireplace object beim Cursor "haften" wird. Klicke jetzt irgendwo in der Nähe vom Anfang-Punkt. Die Feuerstelle wird mit einem Licht hinzugefügt:



Merke, dass diese Feuerstellen-Item noch am Cursor "gehaftet" wird und anzeigt, dass es wiederholt hinzugefügt werden kann. Um Gegenstände die hinzugefügt wurde zu stoppen betätige einfach die ESC-Taste.

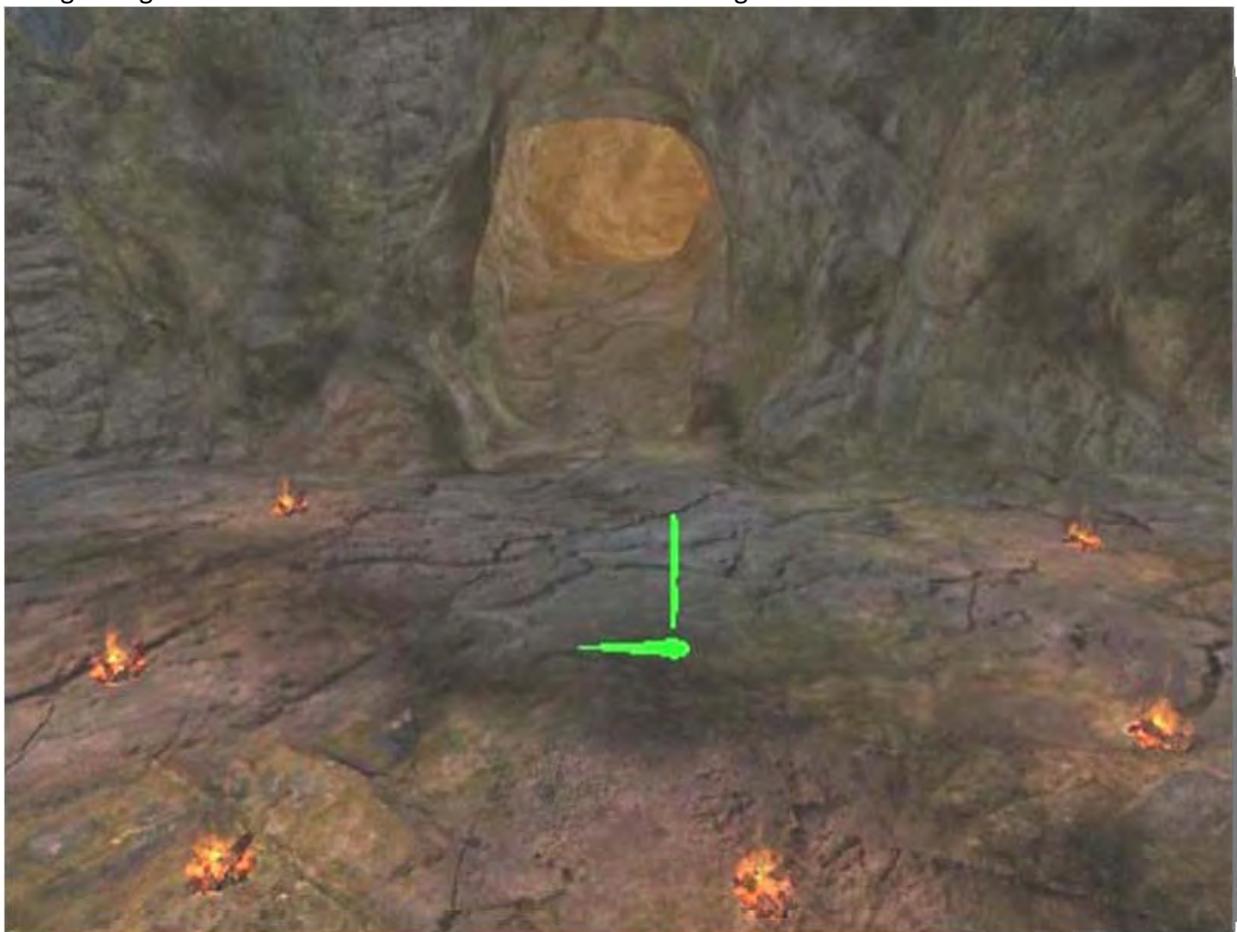
Jetzt lass uns noch einige mehr von diesen Feuerstellen hinzufügen. Dieses Mal kannst du das Placeables Werkzeugs-Icon an klicken:



Die Feuerstelle erscheint sofort auf dem Cursor. Das geschah, weil die Items der Liste der vor kurzem benutzten Placeables hinzugefügt wurde. Du kannst auch die andere vor kurzem benutzte Items sehen, indem Du auf dem abwärts Pfeil klicken, der nahe bei dem Placeables Werkzeug zu finden ist. Der Gegenstand m0_fireplace01 wird in der Liste markiert:



Jetzt lege einige Feuerstellen mehr beim Start-Punkt und betätige dann die ESC-Taste.

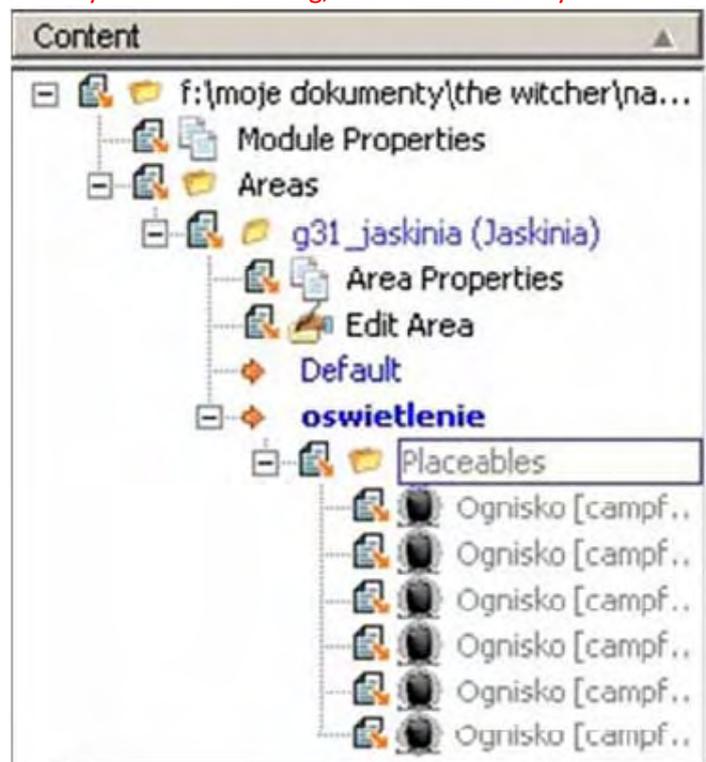


Alle neuen Feuerstellen-Items erscheinen im Modul-Explorer bei den Beleuchtungs Layern im Placeables Unterbaum:



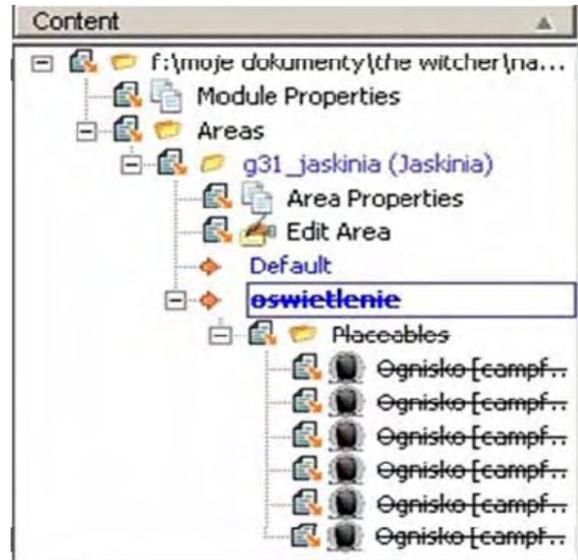
Da es bereits einige Items auf dem Layer gibt, lass uns einige der Befehle hier mal betrachten, die vorhanden sind. rechtsklicke auf den Beleuchtungslayer:

- **Select all Instances.** Alle Items von der Layer werden ausgewählt
- **Freeze Layer.** "friert" eine Layer ein. Anmerkung, alle Items eines Layers sind heller geworden:

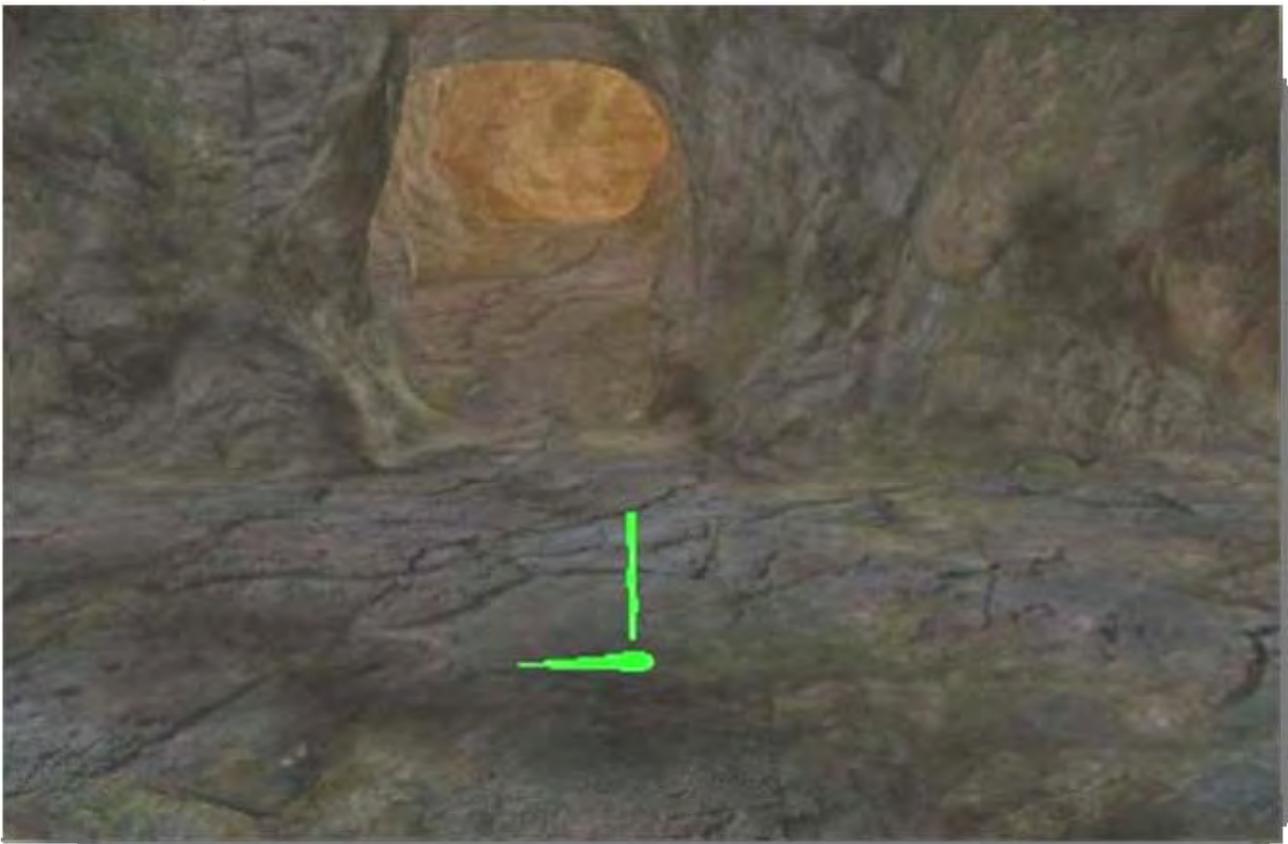


Das zeigt an dass alle Items auf einer Schicht nicht an wählbar sind, Sie können nicht ausgewählt werden, können nicht Rotiert werden, gedreht, entfernt werden, usw. Dieser Befehl ist sehr nützlich, besonders wenn es viele Items in einem Bereich gibt und Du nicht willst das Items auseinander sich miteinander Kreuzen. "unfreeze" Um eine Layer "aufzutauen" wähle den Befehl Freeze vom Layer-Menü.

- **Hide Layer.** versteckt alle Items auf einen bestimmten Layer. Merke Dir, dass alle Gegenstände auf einen Layer durchgestrichen sind:

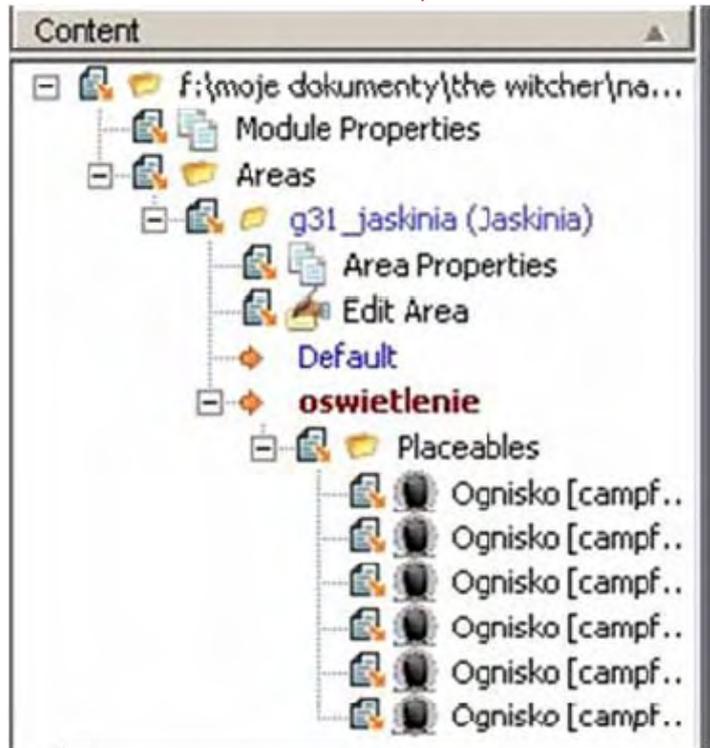


Diese Items sind jetzt auch vom Bereich verschwunden:



Sie werden noch dort versteckt. Dieser Befehl ist auch sehr nützlich. Da das Anzeigen vieler Gegenstände im Bereich kann nämlich den D'jinni Editor verlangsamen. Mit diesem Befehl können Items z.Z. Die nicht im Gebrauch sind versteckt werden und so die Arbeit des Editors beschleunigen. Um die Items wieder aufzu decken, wählen Sie den Hide Layer command aus dem Layer Menü aus.

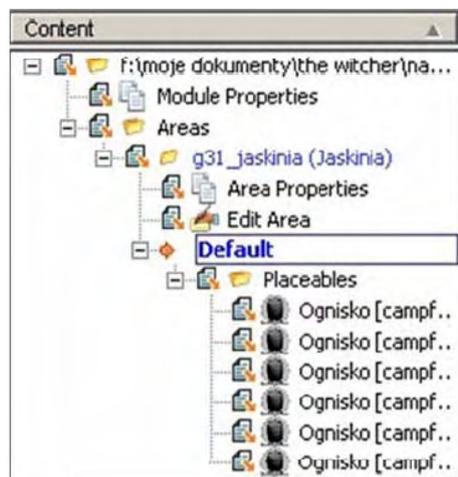
- **Starting Layer.** Eine Schicht die diesen Start-Punkt des Spielers darstellt.



- **Delete Layer.** Entfernt einen Layer. Nachdem man diesen Befehl ausgewählt hat erscheint das Layer Löschenfenster:



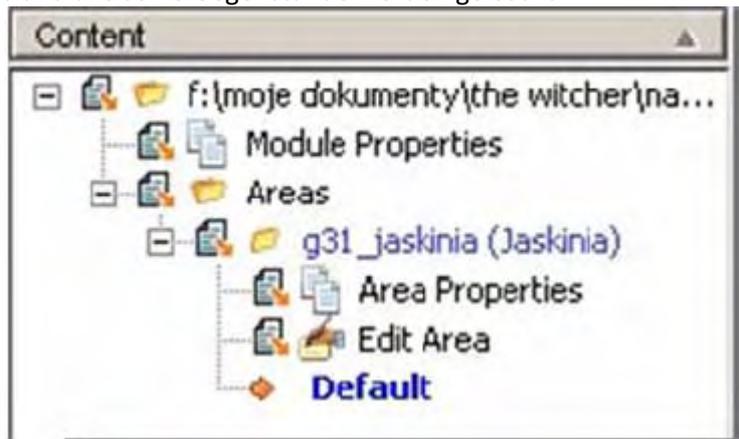
Wenn „Yes“ gewählt wird, wird der Layer gelöscht. Alle Gegenstände auf diesem Layer werden auf die Default Layer verschoben:



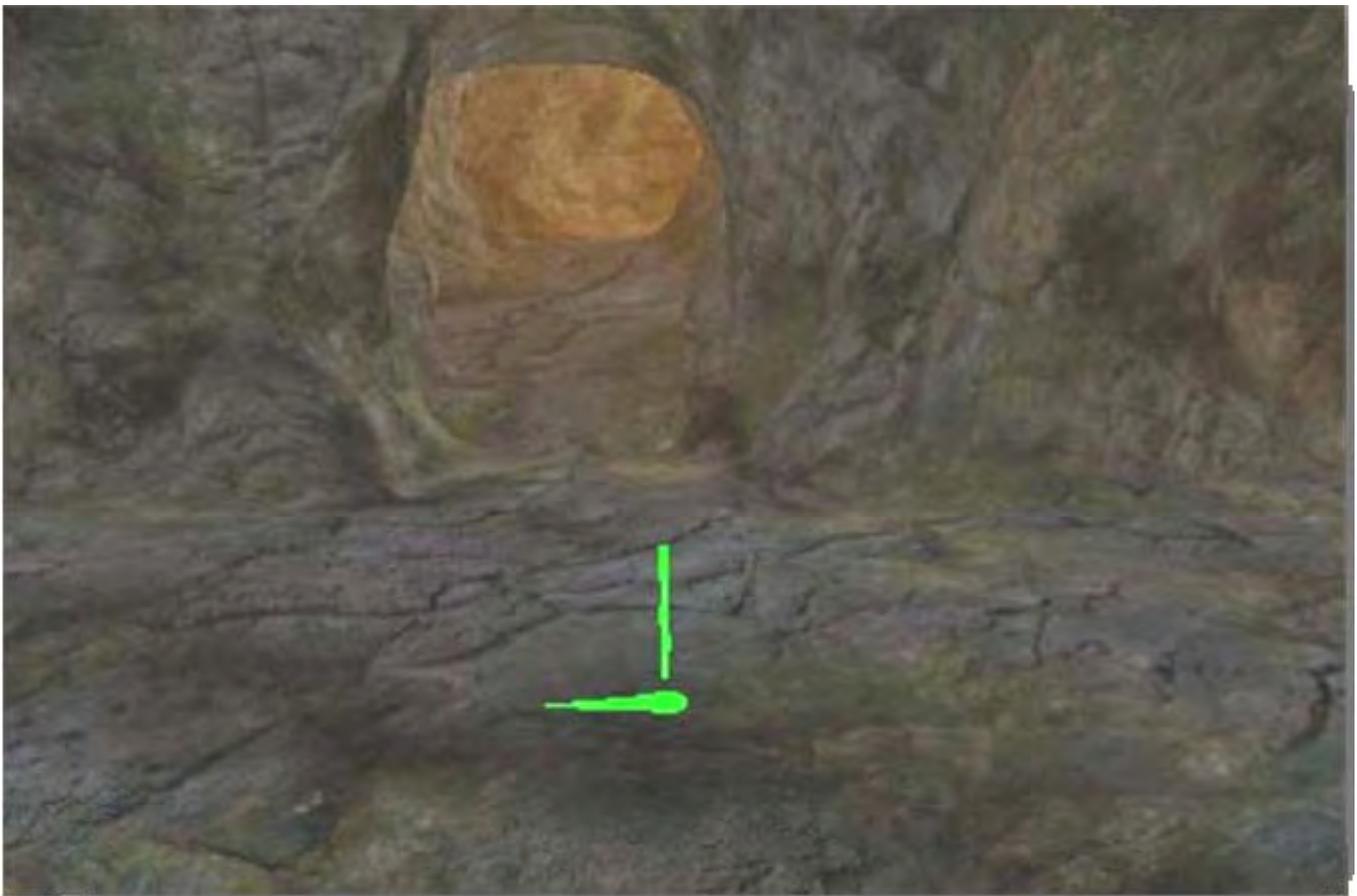
Wenn du einen Layer zusammen mit den Items löschen möchtest, so wähle das Deleting-Layer-Fenster:



Klicke „Yes“ .Die Layers und alle seine Gegenstände werden gelöscht:



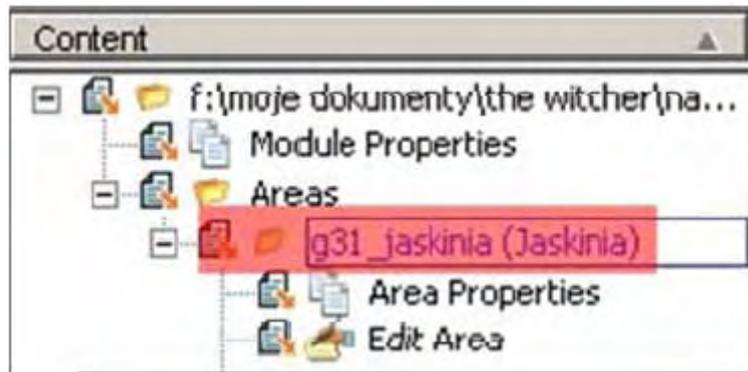
Die Items werden auch vom diesen Bereich entfernt:



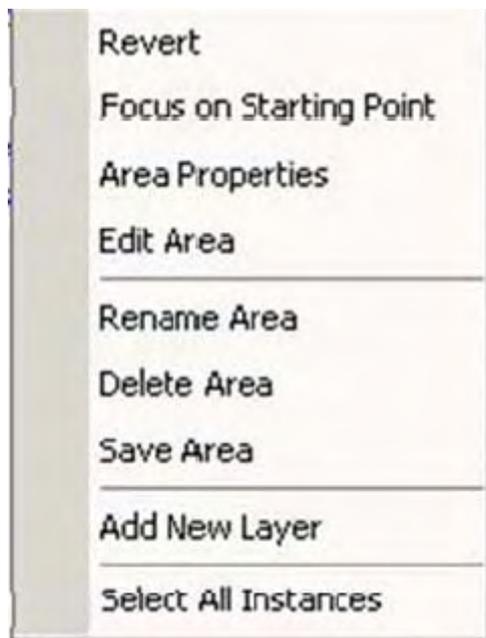
In diesem Fall sind die Gegenstände für immer weg. Es gibt keine Möglichkeit dazu, Die Gegenstände wieder herzustellen.

- **Rename Layer** – benennt eine Layer um. Nachdem Du diesen Befehl ausgewählt hast, tippe einen neuen Namen und drücke Enter.

Das ist für Die Layer. Lass uns einen schnellen Blick auf die Befehle im Bereichs Menü werfen. Du kannst das tun, wenn du mit der rechten Maustaste auf den Unterbau klickst:



Das folgende Menü öffnet sich:



- **Revert.** Nicht benutzbar.
- **Focus on Starting Point.** stellt das Area Editor Fenster auf den Start-Punkt im Mittelbereich ein.
- **Area Properties.** das Eigenschaften-Fenster, welches die Attribute der Bereiche enthält.
- **Edit Area.** Öffnet den Bereichs-Editor.
- **Rename Area.** Benennt einen Bereich um.
- **Delete Area.** löscht den Bereich.
- **Save Area.** speichert den Bereich.
- **Add New Layer.** hinzufügen eines neuen Layers.

- **Select All Instances.** wählt alle Items aus, die auf einer Layer liegen.

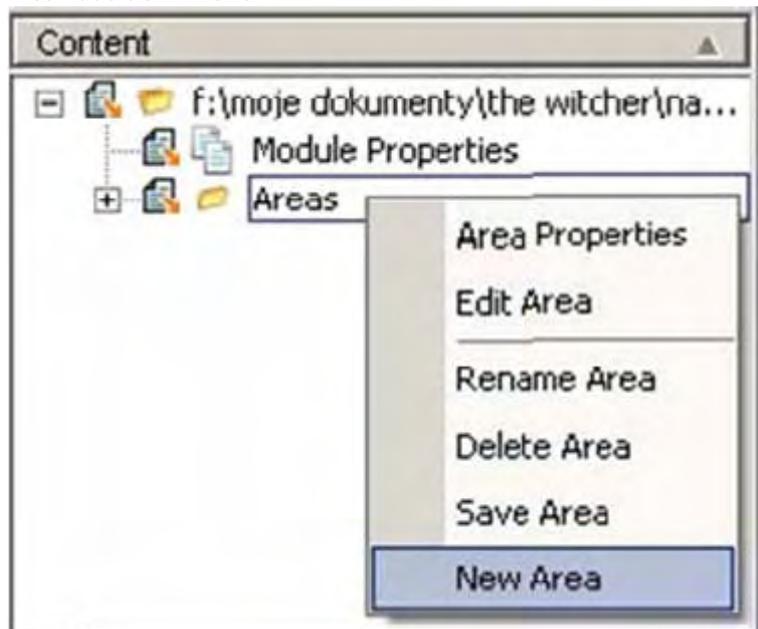
Bewegen zwischen Areas

Ein sehr wichtiges Element der Modulkreation ist die Bewegung zwischen den Bereichen. Einige komplette Bereiche sind zusammen mit dem D'jinni Editor von den Autoren des Witcher zur Verfügung gestellt worden. Dies schließt die Äußere und Innenbereiche mit ein. Mit unserem Modul wurde der erste Bereich (die Höhle) bereits Erstellt. Jedoch mangelt es hier an einem Eingang. Der Eingang zur Höhle sollte etwa vom Außengelände, z.B. ein Wald oder ein Sumpf erreichbar sein. In unserem Beispielm modul wird ein etwas anderer Weg genommen.

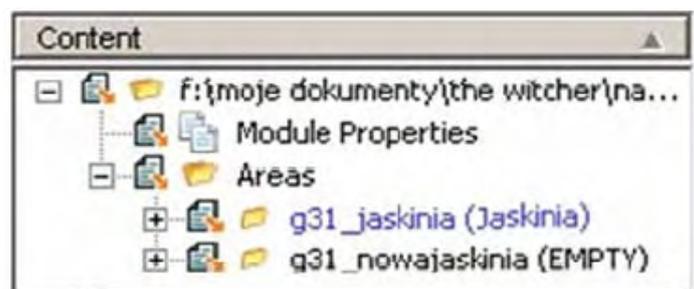
Der folgende Bereich, den wir Hinzufügen wollen, ist eine Höhle. Selbstverständlich gibt es einen Durchgang zwischen den zwei Bereichen. So kann Geralt (der Spieler) von einer Höhle zum anderen Höhle sich bewegen oder umgekehrt. Wirklich aus der Perspektive des Spielers, geschieht nicht viel.

Der Spieler denkt, dass er sich von einem Gebiet bis zu einen anderen bewegt. Den der Start-Punkt das dieses Modul erstellt für diese Person sind die Dinge ein bisschen verschieden, Der Spieler wird von einem Gebiet bis zu einem anderen teleportiert. Um von einem Bereich zum Nächsten zu gehen, sind zwei Elemente erforderlich: eine Tür oder ein Event von wo der Spieler Telepotiert wurde und vom Welchen Start-Punkt er kommt.

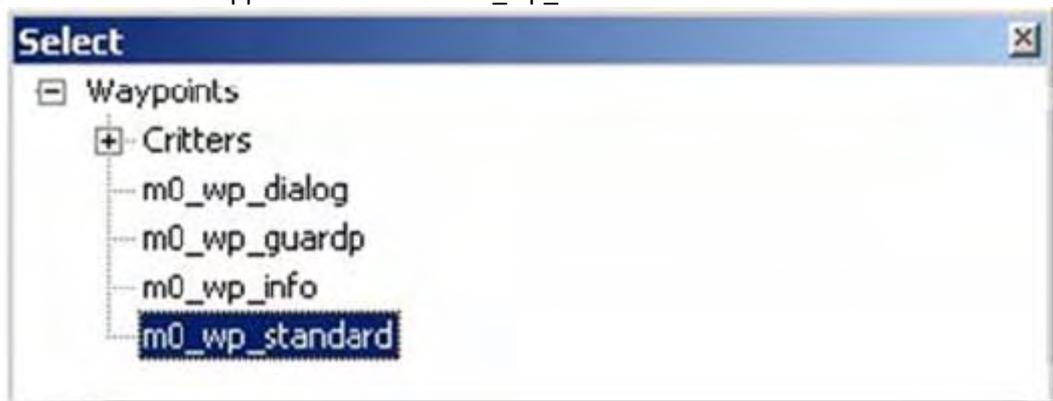
Lass uns zuerst den zweiten Bereich hinzufügen. Du weißt bereits wie man einen Layer hinzufügen kann, wir machen gerade auf die schnelle eine Überprüfung. Klicke rechts auf den Bereichsbaum im Modul Explorer, und wähle „New Area“ aus dem Menü:



Gehe zum Ordner Data\Meshes\G31, und wähle den g31.set Ordner und klicke an, und öffne es. Der gleiche Höhlenbereich, den wir für den ersten Bereich verwendeten, wird hier hinzugefügt. Im neuen Bereichsnamen Fenster trägst du g31_newcave ein und klicke auf O.K. Ein neuer Bereich erscheint im Modul-Explorerfenster:



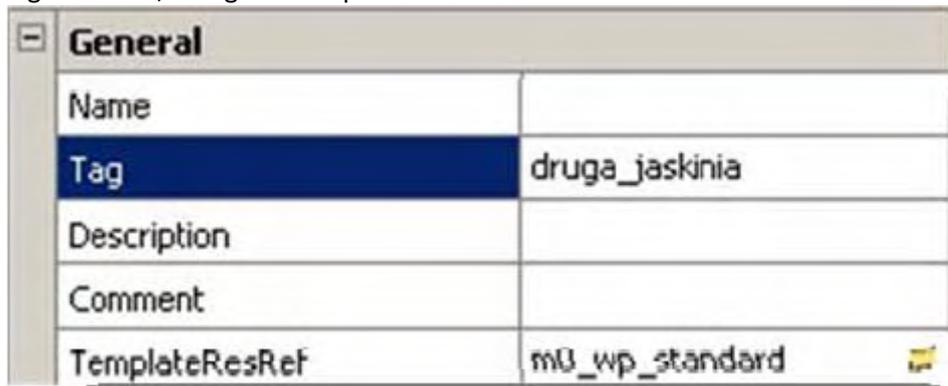
Öffne den g31_newcave (LEEREN) Baum und Doppelklicke auf den Editierten Bereich. Dann im Bereich Lademodusfenster, klicke auf Entire Geometry. Das öffnet das Bereichs Editor Fenster. Füge jetzt den Waypoint irgendwo in der Höhle ein. Um das zu tun, den abwärts Pfeil anklicken, der nahe vom Waypoints Tool ist, bearbeite es und wähle das neue Waypoint template vom Menü aus. Öffne dann den Waypoint-Baum und Doppelklicke auf den m0_wp_standard:



Klicke auf diese Umgebung, in der dieser Waypoints hinzugefügt werden soll. In diesem Beispiel fügen wir die Waypoints in dem Großen Raum in der Höhle ein:



Doppelklicke auf die neuen Waypoints. Ein Eigenschaften-Fenster, welches die Attribute der Waypoints enthält, öffnet sich jetzt auf der rechten Seite des Schirmes. Trage „second_cave“ in das Tag Attribute des allgemeinen Abschnittes ein. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber vom opposite the Tag Attribute, und gebe den passenden Text Drücke dann Enter:



Erinnern dich an diesen, Waypoint tag weil Du ihn später benötigst. Mit diesem Tag weiß das Spiel, wo der Spieler teleportiert werden soll. Wichtig: Damit die Teleportation, arbeitet muss der Ursprung und der bestimmungsort den gleichen Namen haben, Ordne einen Namen dem neuen Bereich zu.

Doppelklicke auf Area Properties im g31_newcave (LEEREN) Baum. Ein Bereich Eigenschaften Fenster wird erscheinen auf der rechten Seite des Schirmes. Tippe einen zweite Höhle auf der rechten Spalte gegenüber vom Namens Attribute im allgemeinen Abschnitt ein und drücke Enter. Das ist alles, was in diesem Bereich getan werden muss. Wollen wir die Änderungen des Moduls speichern. Speichern des Moduls im Modul-Menü:

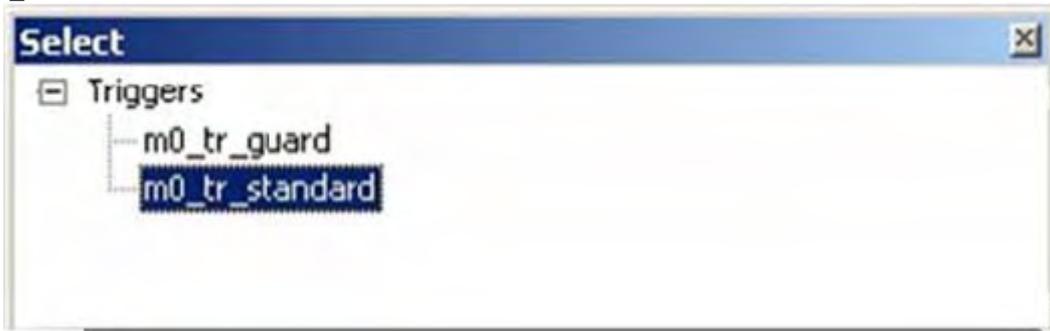


Wollen wir jetzt die alte Höhle editieren, Doppelklicke auf den Editierten Bereich bei g31_cave (Höhle). Du versuchst jetzt einen rechteckigen Trigger in der Nähe der Umgebung des Spielers. In diesem Beispiel wird der Trigger ein Durchgang zwischen den zwei Umgebungen.

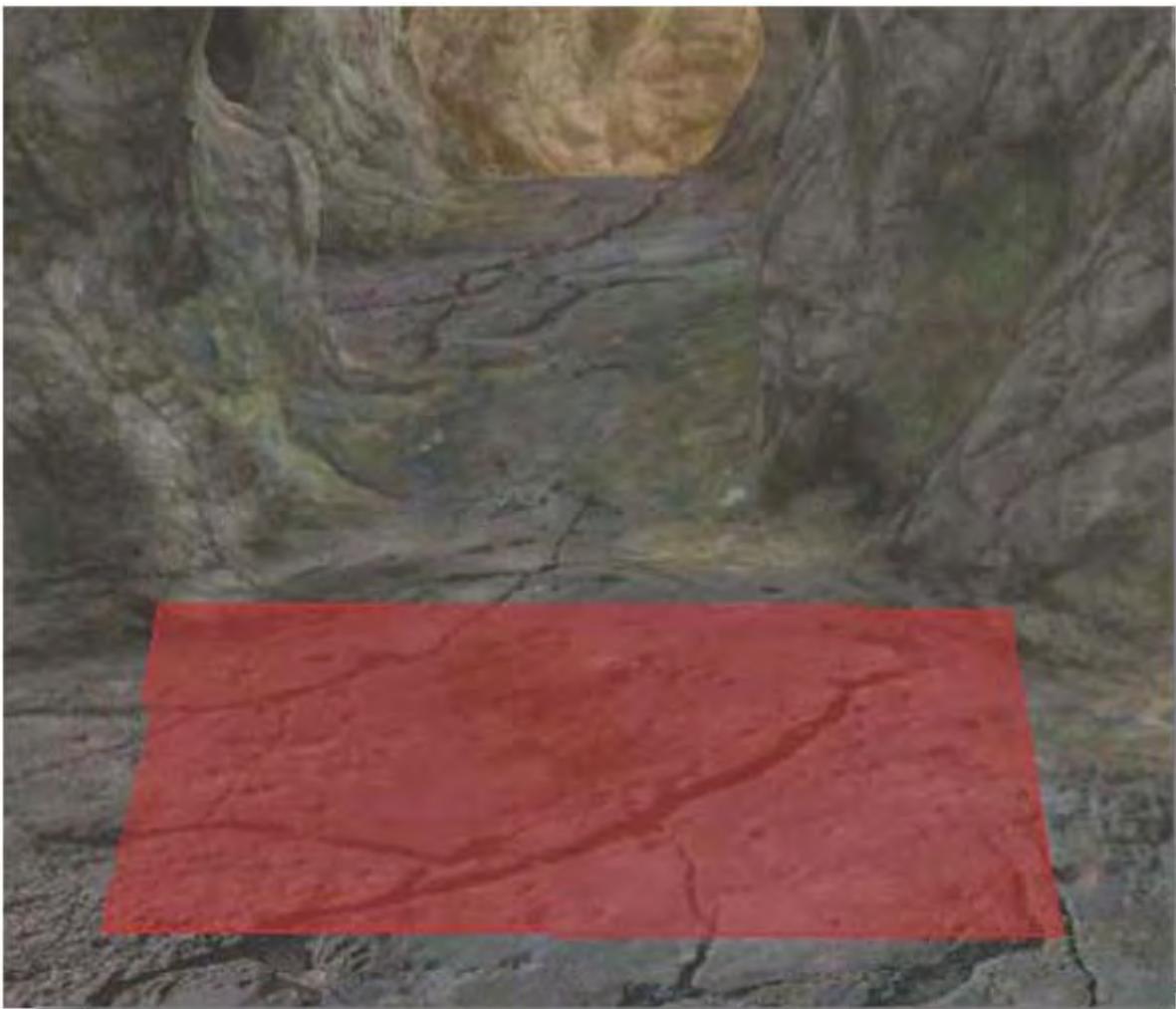
Türen können jetzt auch benutzt werden, um zwischen den Bereiche sich zu bewegen, aber die Türen können auch auf bestimmte Umgebungen gelegt werden. Wenn Du einen Durchgang bilden möchtest, der von einer Stadtstraße in ein Gebäude z.B. führt musst Du eine Türschablone benutzen. Jeder Gebäudewert schließt diese bestimmten Tür mit ein, eingesetzt durch die Schöpfer des Witcher ein Gebäude enthält mehrere Bereiche, zwischen denen der Spieler sich bewegen kann. Um eine Tür Hinzuzufügen, wähle eine neue -Schablone -Tür(utd) vom Ordner Menü aus.

Für unsere Höhle ist eine Tür nicht angebracht.

Stattdessen klicke auf den abwärts Pfeil, der nahe bei dem Triggerwerkzeug vom Bereich Editor Toolbar zu finden ist. Wähle eine neue Triggerschablone vom Menü aus, und öffne den Trigger baum und Doppelklicke auf m0_tr_standard:



Das Werkzeug verwenden, bestimmte Formen das zu einem Rechteck wurde wie früher im Handbuch beschrieben, klicken auf einen freien Platz , und Verbinde den letzten Punkt mit dem ersten, um einen neuen Trigger zu schaffen:



Doppelklicke auf dem neuen Trigger im Modul-Explorer .Jetzt müssen wir die Attribute des Triggers einstellen.

General Section:

- **Name.** Betrete den Durchgang zur folgenden Höhle. Diese Phrase(Ausdruck) erscheint während des Spiels der Cursor des Spielers über dem Trigger schwebt. Wenn eine Tür als ein Übergangs- Punkt benutzt worden ist, erscheint die Phrse auf der Tür.

Attributes Section:

• **Type.** Stell dieses Attribut auf Area transition ein Das Spiel weiß jetzt, dass dieser Trigger als ein Durchgang zu einem anderen Bereich dient.

Area Transition Section:

• **Destination tag.** Trage second_cave in dieses Attribut ein. Dieses Attribut stellt den Bestimmungsort fest, an dem der Spieler gebracht wird. Erinnern Dich daran, dass dieser Tag die Bestimmungs-ort-punkte zusammenbringen muss. In unserem Fall hat der zweite Punkt, den wir der neuen Höhle hinzufügen, genau diesen Namen.

So ist's recht! Lass uns das Modul und die Bereiche speichern wähle in der Anzeige Spiel speichern im Moduls-Menü aus.

Wenn das Modul jetzt gestartet wurde und Geralt (der Spieler) den Trigger einsetzte, würde er zur zweiten Höhle transportiert werden. Er würde durch die Finsternis eingewickelt werden, da wir noch keine Beleuchtung im zweiten Gebiet hinzufügen

Charaktere

Einleitung:

in diesem Kapitel werden wir auf die Entwicklung von Spielcharakteren schauen. Die Themen, die besprochen werden, schließen die Charakter-Schablonen und ihre Entwicklung da, Charakter-Spawn mit ein, das auch, den Spawn-Set-System verwendet, auch wie man einen Charakter leitet, mit einem Aktionspoints wird beschrieben.

-PCs (Nicht-Spielercharaktere), die mit einem Plot im Witcher verbunden würden. Das beinhaltet die Charaktere wie Triss und Jaskier

-Unabhängige Charaktere (Bürger mit anderen Worten bezeichnet), gehören die Charaktere zu einer oder eine bestimmte Gruppe Dies können Stadtbewohner, Wachen, Scoia'tael Mitglieder, Kaufleute oder Banditen sein.

-Alle Arten von Monstern

Templates:

Lass uns mit einer Charakter Schablone beginnen. Was ist eine Charakter Schablone? Es ist nur ein Set von Attributen welches die Beschreibung eines Charakters enthält. Eine Schablone enthält normalerweise eine definierte Charaktererscheinung, Charakternamen (dies betrifft hauptsächlich Handlungscharaktere, Fähigkeiten, Kampfstile usw.).

Für den Zweck dieses Handbuchs schaffen wir uns ein neues Abenteuer, wo Geralts Quest ist, Den Anführer der Banditen zu besiegen. Da Siegfried uns den Auftrag erteilt hat. Siegfried wird unser NPC sein. Jethro, der eifrige Würfelpokerspieler, ist ein freundlicher Charakter, es gibt aber mehrere Bösewichter. Fünf Charakterarten von den Charakteren werden wir jetzt erstellen.

- Siegfried – ist ein Ritter und unser Auftraggeber.

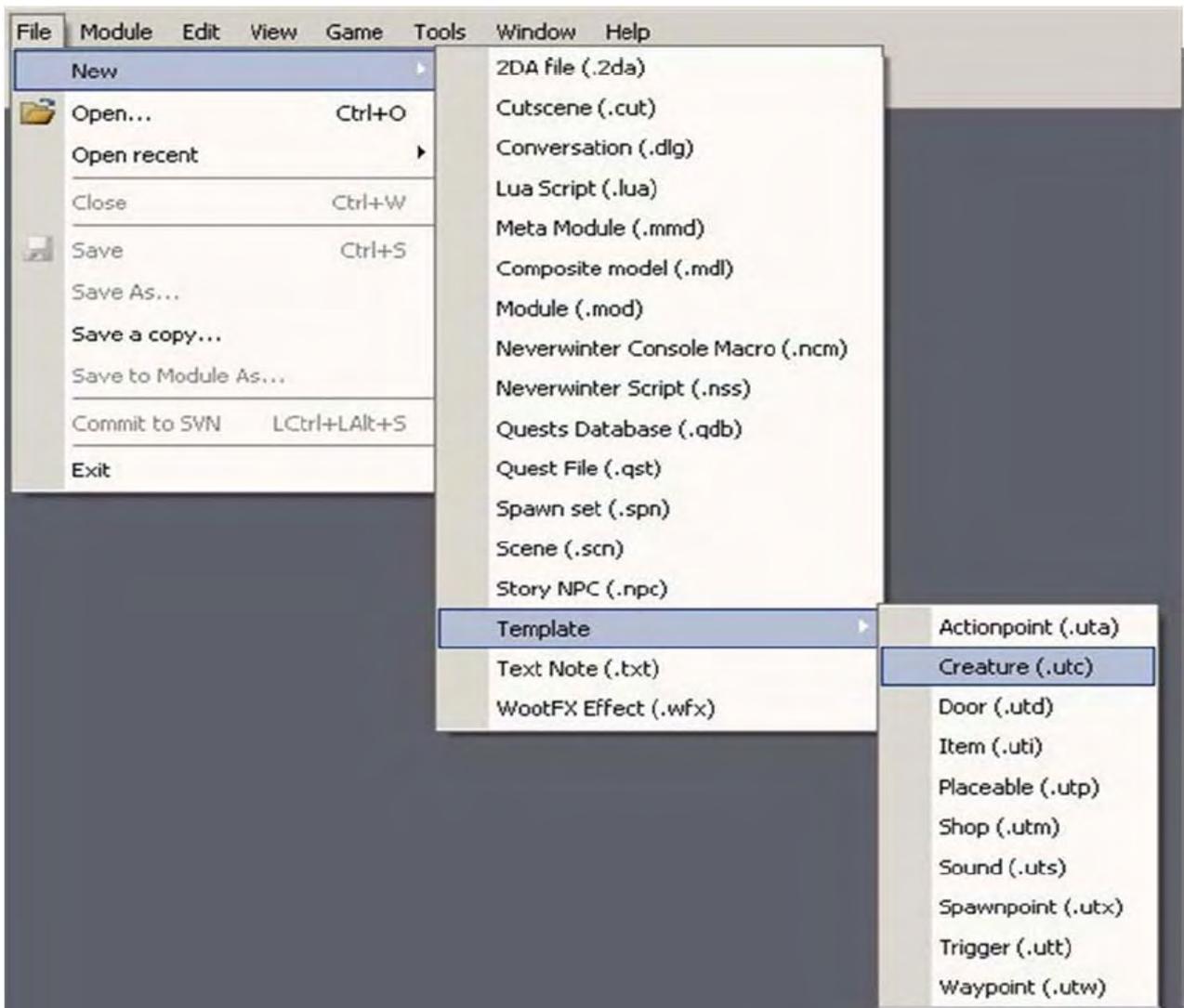
- Jethro – ist eine Wache und ein eifriger Würfelspieler.

- The Professor –Anführer der Banditen und ein Schlüsselcharakter bei unserer Quest.

- The Professors Suards – zwei seiner persönlichen Leibwächter (eine Schablone wird genug sein, um sie zu erstellen).

- The wolves – five wolves -(wie bei den Leibwächern genügt eine Schablone)

Die einzelnen Charaktere werden mit Hilfe von Schablonen geschaffen. beginne mit Siegfried. Zuerst starte D'jinni Editor und erstelle dann eine neue NPC Schablone, selecting New -> Template -> Creature (.utc) vom Menü:

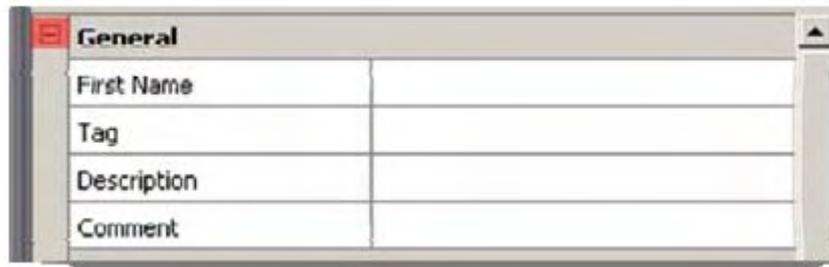


Das „New creature Template“ Fenster wird sich auf der rechten Seite des Bildschirms öffnen:

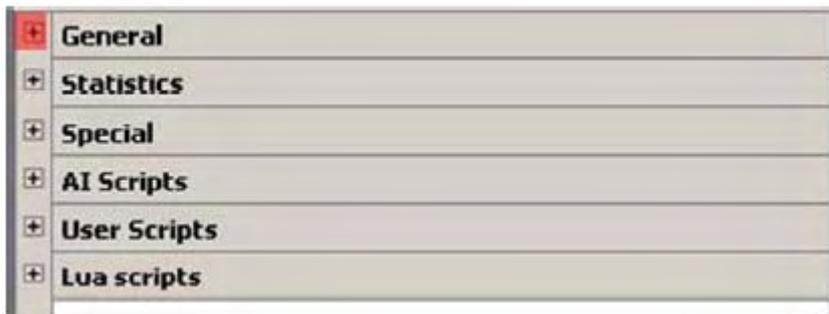
new creature template	
General	
First Name	
Tag	
Description	
Comment	
Statistics	
Appearance	cr_witch1_c1g1 
Level	0
Race	Nonhuman 
Gender	GENDER_MALE 
Experience Value	0
Movement Rate	PC_Movement 
Perception Range	Default 
Profiles	
Abilities	0 abilities defined 
Inventory Contents	
Combat Styles	0 styles defined 
Conversation	 
Voice Set	 
Runs To Despawn	False
Can be pushed	True
Special	
Player Character	False
Plot	False
Permanent Body	False
Don't Preload Model	False
Leaves blood stains	True
Immortal	False
Treasure Model	default 

Dieses Eigenschaften-Fenster hat dasselbe grundlegende Äußere wie das „Modul- und Bereichs- Eigenschaften Fenster, aber es enthält andere Attribute.

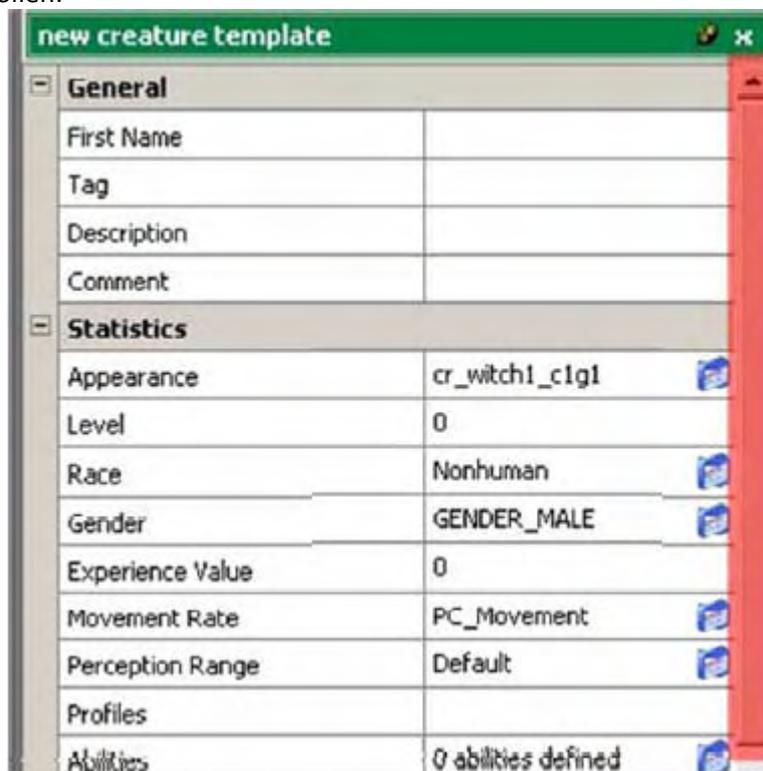
Als eine Gedächtnishilfe wird das Fenster in zwei Spalten geteilt, mit den Attributen auf der linken Seite. Das Fenster wird in die folgenden Abschnitte geteilt: General, Statistics, Special, AI Scripts, User Scripts und Lua Scripts. Um die Navigation zwischen den Abschnitten zu erleichtern, können sie minimiert werden, indem du auf den „-“, neben jedem Abschnittnamen klickst:



Um einen vorher minimierten Abschnitt wieder zu öffnen, musst du nur auf das kleine '+' Zeichen neben dessen Namen klicken:



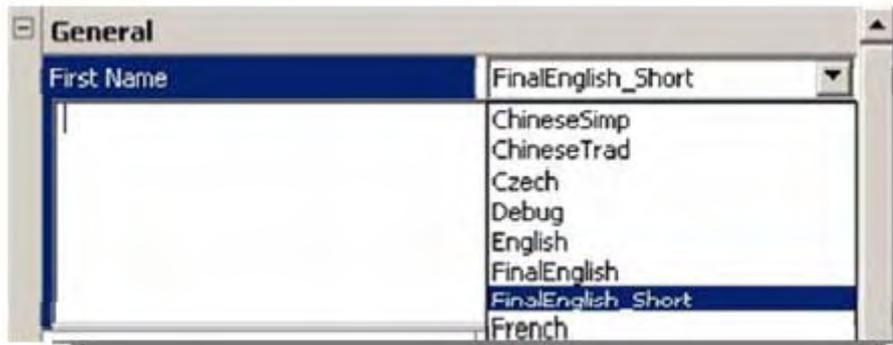
Den Wert jedes Attributes kannst du in der rechten Spalte finden. Jedes Attribut definiert einen bestimmten Teil eines Charakters. Die Liste von Attributen ist ziemlich lang, deshalb kannst du mithilfe der Scrollbar rechts auf und ab scrollen:



Lass uns nun die Attribute für Siegfried festlegen. Ich werde die Bedeutung der relevanten Attribute dabei näher erklären:

General Section:

- **First Name.** Enthält eine Beschreibung des Charakters. Genauer gesagt, ist das der Name, der über dem Charakter im Spiel sichtbar ist. Es ist möglich, eine andere Sprache für die Beschreibung auszuwählen. Klicke auf die rechte Spalte neben „First Name“ und wähle „FinalEnglish_Short“ aus der Liste aus, die erscheint:



Trage nun den Namen „Siegfried“ ein und drücke „Enter“:



- **Last Name.** Wird nicht benutzt

- **Tag.** Gewidmet den Fahnen, die einen besonderen Charakter beschreiben. Hauptsächlich verwendet mit Scripts, die später in diesem Handbuch besprochen werden. Nenne dies „our_siegfried“. Der Name Siegfried allein konnte technisch verwendet werden, aber ein Charakter mit diesem Tag ist bereits von den Schöpfern Des Hexers definiert worden. Um unnötige Probleme zu vermeiden geben wir diesem Charakter ein anderen Tag. Um den Text einzugeben, klicke auf die rechte Spalte neben „Tag“, trage "our_siegfried" ein und drücke auf „Enter“:

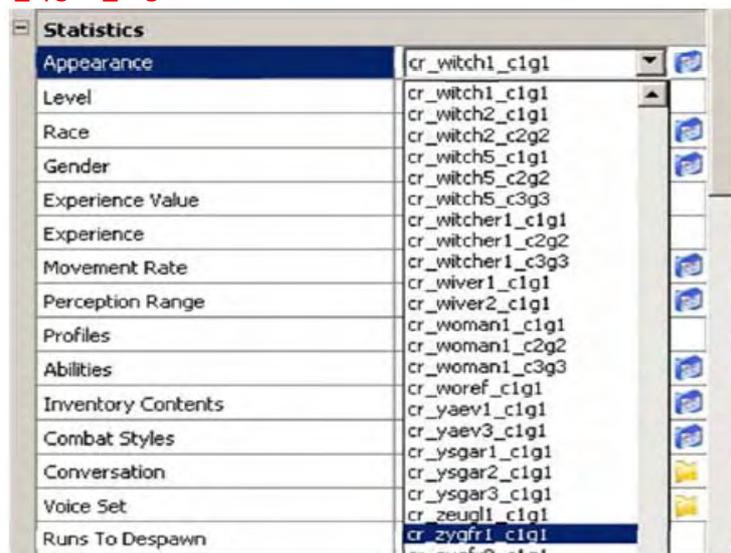


- **Description:** Wird nicht benutzt

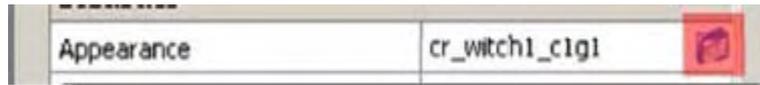
- **Comment:** Beinhaltet Platz um einen Kommentar zu schreiben. Dies wird nicht im Spiel benutzt, es ist einfach ein Feld für Informationen. Du kannst alles reinschreiben, was du willst, zum Beispiel über Objekte.

Statistics Section:

- **Appearance:** Dieses Attribut beschreibt das Aussehen eines Charakters. Klicke auf die Spalte neben „Appearance“ und wähle cr_zygfr1_c1g1 aus der Liste:



Die Liste ist umfangreich und du kannst mithilfe der „Scrollbar“ rechts navigieren. Klicke jetzt auf das kleine blaue Icon rechts neben der Liste:

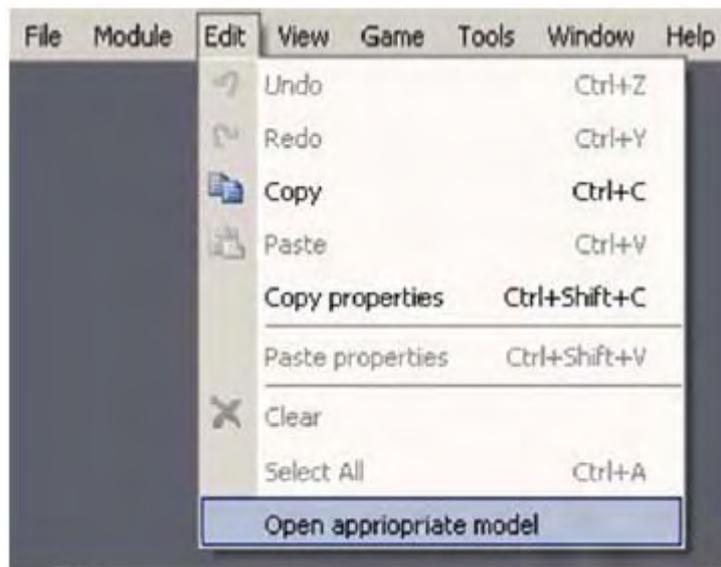


Dadurch öffnet sich die Datei *appearance.2da*:

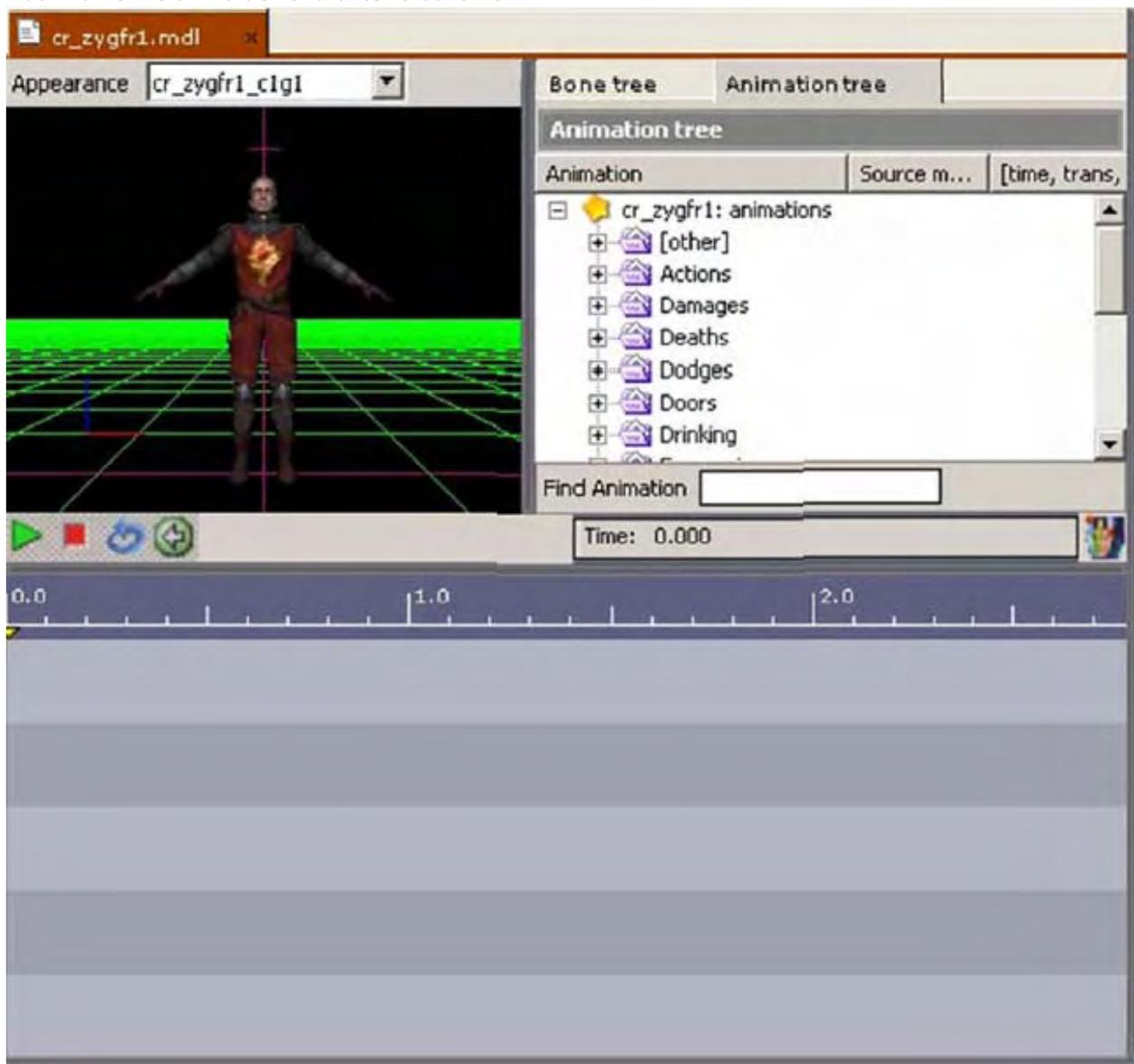
	LABEL	RACE	BODY_VERSION	HEAD_VERSION	BODY_SPECIAL_TEXTURES	HEAD_SPECI
0	cr_witch1_c1g1	cr_witch1	1	1	_n:_a	_f
1	cr_witch2_c1g1	cr_witch2	1	1	_n:_a	_f
2	cr_witch2_c2g2	chr_geralt	****	****	_n:_a	_f
3	Wiedzmin4	****	****	****	****	****
4	cr_witch5_c1g1	cr_witch4	1	1	_n:_a	_f
5	cr_witch5_c2g2	cr_witch6	1	1	_n:_a	_f
6	cr_witch5_c3g3	cr_witch5	1	1	_n:_a	_f
7	****	****	****	****	****	****
8	****	****	****	****	****	****
9	****	****	****	****	****	****

Diese Datenbank beschreibt das Aussehen der individuellen Charaktere in „The Witcher“. An dieser Stelle will ich nicht weiter auf „.2da Dateien“ eingehen. Die werden in einem anderen Kapitel besprochen.

Um dir ein dreidimensionales Modell des Charakters anzusehen, wähle „Open appropriate model“ aus dem „Edit“ Menü:

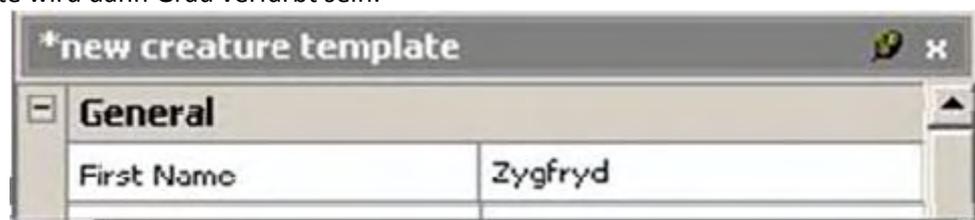


Nach kurzer Zeit wird der Charakter erscheinen:



Das Charakter-Modell ist in eins von D’jinni’s vielen Plugins geladen worden. In diesem Fall in den Model-Viewer, ein Plugin, das Modelle zeigt.

Wenn die *geöffnete entsprechende Model* Option nicht verfügbar oder zugänglich im *Editiermenü* ist, zeigt das auf eine *neue Kreatur*-Schablonen-fenster das wahrscheinlich inaktiv ist. Wenn das Fenster inaktiv ist, die Titel-leiste wird dann Grau verfärbt sein:



Um das Fenster zu aktivieren, klickst du irgendwo auf der Titelleiste, das für dich die beste Wahl ist. Wenn die Titel-Leiste anzeigt dass sie Grau ist, Bedeutet dass dieses fenster jetzt aktiv ist.

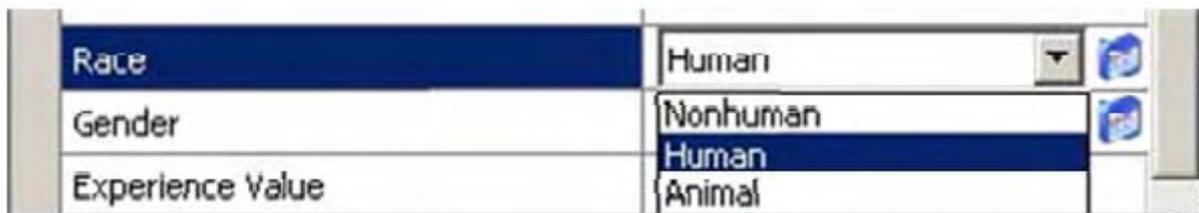
• **Level.** Bestimmt das Erfahrungslevel des Charakters. Diese Ebene wird verwendet, um die Anzahl der Erfahrungspunkte zu berechnen das Geralt verdient durch das töten eines Charakters. Wenn das Level des Charakters höher ist als Geralt seins, werden zusätzlich Erfahrungspunkte addiert, wenn aber das Level des Charakters kleiner ist als von Geralt, wird der Erfahrungswert um einen Punkt abgemildert, wie in einem minimalen Level. Da aber Siegfried ein Freund von Geralt ist kann er im Spiel nicht getötet werden. Den so ist der Wert auf 0 um das zu ändern klickst du in der rechten Spalte gegenüber vom the *Level* Attribute dann Drückst du Enter:



Der Wert kann auch mit Hilfe der kleinen Pfeile geändert werden, die sich gegenüber vom *Level* Attribute befinden:



• **Race.** Bestimmt die Rasse des Charakters. Um das zu ändern, klickst Du auf die rechte Spalte gegenüber vom *Race* Attribute. Für Siegfried wähle die Human Option:

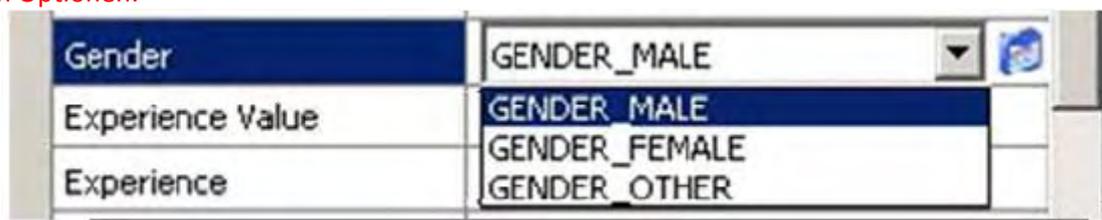


Die Werte, die du in diesem Attribut zuweisen kannst, sind:

- *Nonhuman* – Elf oder Zwerg, - *Mensch*,
- *Tier*,
- *Spectre* –Gespenst oder Geist, - *Vampire*,
- *Ornitoreptile* – kriechendes Monster (z.B. Wyvern, Basilisk, Cockatrice), - *gewöhnliche Monster*,
- *Insectoid*,
- *Aas*

Du kannst auch auf das blaue Icon klicken, die sich neben der Auswahlliste befindet. Das öffnet dann die *racialtypes.2da* Datei, die Datenbank von den Rassen, die im Witcher verfügbar sind.

• **Gender.** Bestimmt das Geschlecht des Charakters. Um das Setting zu ändern, klickst du auf die rechte Spalte gegen über vom *Gender* Attribute. Da Siegfried ein Mann ist, wählst du *GENDER_MALE* aus der Liste von den Optionen:



Die Werte, dass Du in diesem Attribut zuweisen kannst, sind:

- *GENDER_MALE* (Männlich)

- *GENDER_FEMALE* (Weiblich)
- *GENDER_OTHER* (Verschieden)

Du kannst auch auf das blaue Icon klicken, das sich neben der Auswahlliste befindet. Das öffnet dann die *gender.2da* Datei, die Datenbank von Geschlechtern, die im Witcher verfügbar sind

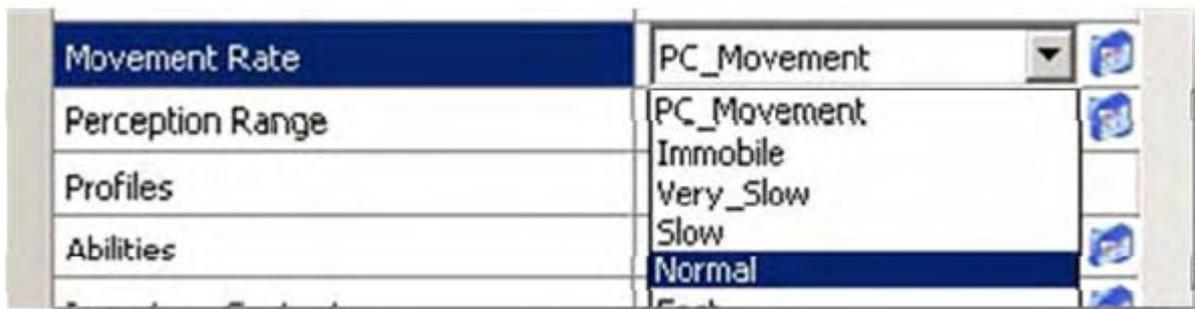
- **Experience Value.** Bestimmt das Grund Maß an einer zugeteilten Erfahrung, nachdem ein Charakter getötet worden ist von Geralt. Dieser Wert wird von einem entsprechenden Koeffizienten modifiziert, Das aus einem unterschiedlichen Level von Geralt und dem ermordeten Charakter zustande kommt. Im Fall von Siegfrieds trägst du einen Wert von 0 ein und Drückst Enter:



Der Wert kann auch mit den kleinen Pfeilen abgeändert werden zu finden gegenüber von Experience Value Attribute:



- **Experience.** Nicht benutzbar.
- **Movement Rate.** Bestimmt das Tempo, wie sich ein Charakter bewegt. Um das Setting zu ändern, klicke auf die rechte Spalte gegenüber vom. *Movement Rate* Attribute Für Siegfried wähle die normale Option von der Liste:



Du kannst auch auf das blaue Icon klicken, das sich neben der Auswahlliste befindet. Das öffnet die *creaturespeed.2da* Datei, die eine Datenbank, der Movement rate values (Laufen, Gehend usw.). Die Werte, die du diesen Attribut zuweisen kannst, sind:

- *PC_Movement*, (Spielergeschwindigkeit),
- *Immobile*, (unbeweglich)
- *Very_Slow*, (sehr langsam)
- *Slow*, (langsam)
- *Normal* – normale Geschwindigkeit,
- *Fast*, (schnell)
- *Very_Fast*, (sehr schnell)
- *Default* – Standardgeschwindigkeit,
- *DM_Fast*,

- *Witcher- Witcher* Geschwindigkeit, schnellstegeschwindigkeit im Spiel (gilt nur für den Witcher).

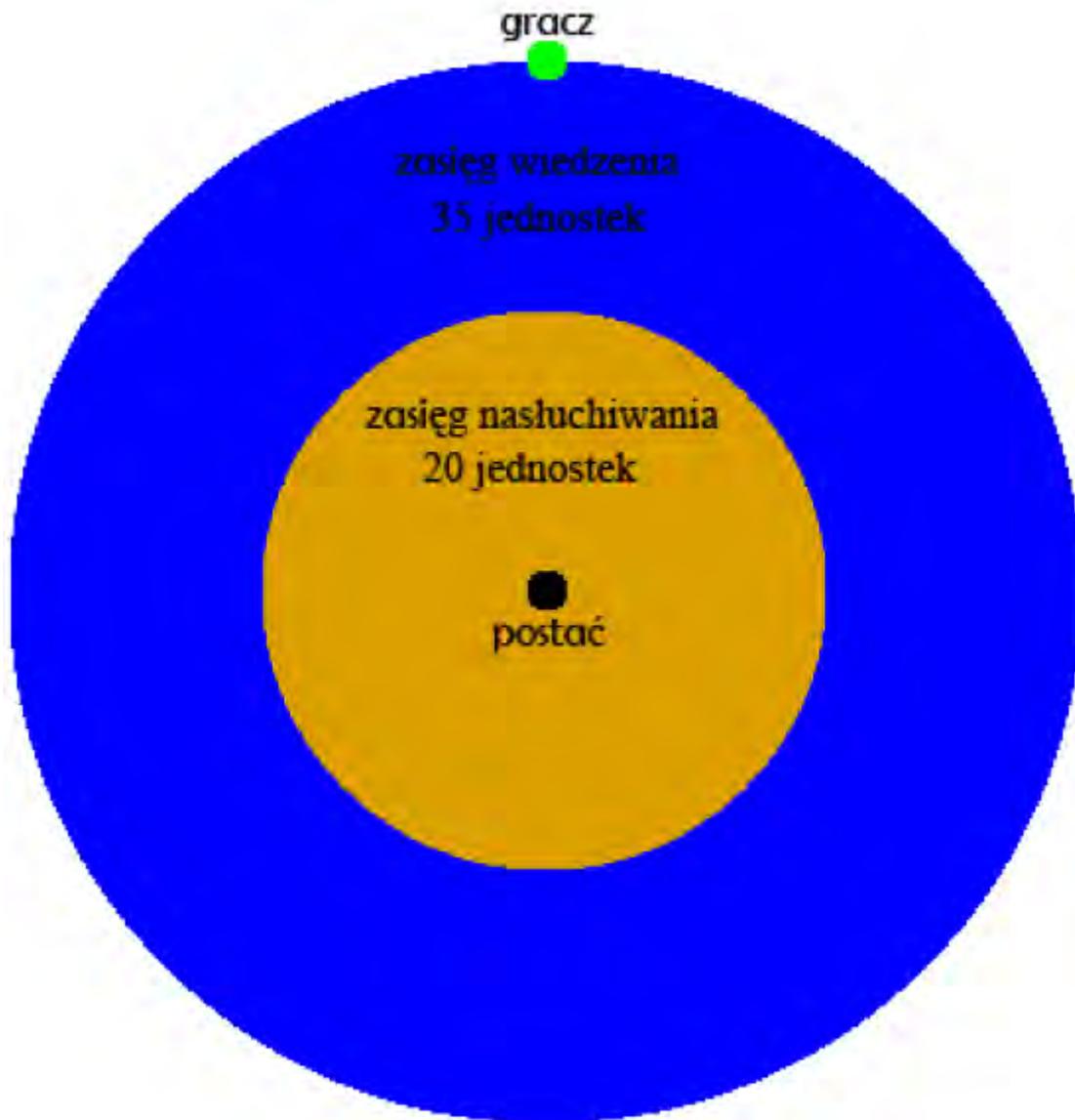
• **Perception Range.** Bestimmt den Wahrnehmungs-Bereich eines Charakters. Wenn ein Charakter im Bereich eines anderen Charakters erscheint, ergreift er die Maßnahme dass bei den Profil Attribute Einstellungen eingestellt wurde. Um diese Einstellung zu ändern, klickst Du auf die rechte Spalte gegenüber vom Perception Range Attribute. Für Siegfried wählst du die *VeryShort* bei der Auswahl aus:



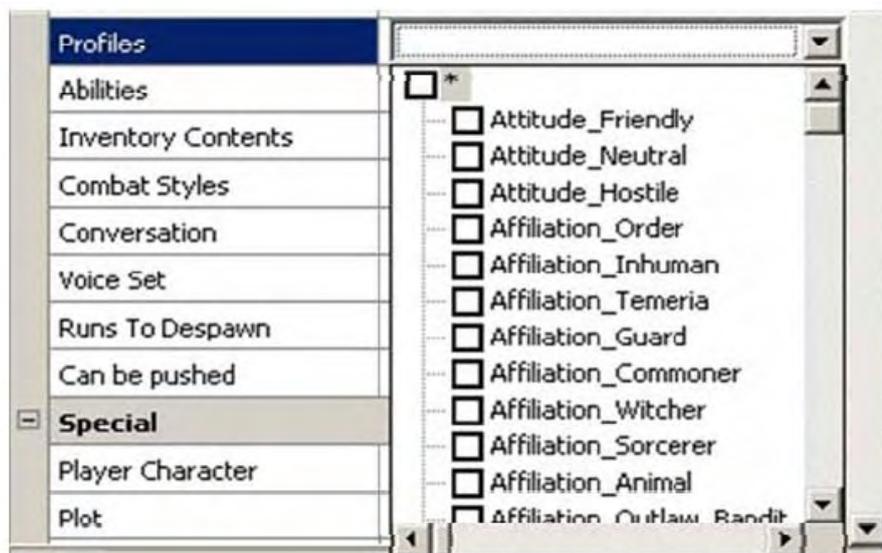
Du auch auf das blaue Icon klicken, das sich neben der Auswahlliste befindet. Das öffnest die *perceptionrng.2da* Datei, die eine Datenbank, mit Perception Ranges enthält. Du solltest Beachten das die Charaktere im Witcher zwei Beschaffenheit haben mit ihren Wahrnehmungsbereichen. Einmal zuhören und Entdecken. Die Werte, das du diesen Attribut zuweisen kannst, sind:

- *Default* – Default range für das hören und Entdecken,
- *Player* – zuhören range 20 Einheiten , entdecken range 35 Einheiten,
- *Short* – listening and spotting Ranges 15 Units,
- *Medium* – listening range 20 Units, spotting range 35 Units,
- *Long* – listening range 20 Units, spotting range 25 Units,
- *VeryShort* – listening range 5 Units, spotting range 7,
- *VeryLong* – listening range 38 Units, spotting range 45 Units.

Die Strecke im Witcher wird nicht in den Metereinheiten oder in irgendwelchen anderen vertrauten Meter einheiten der Masse gemessen. Die Strecke ist ein Abstand gerechterweise, das von den Schöpfern des Witcher definiert wurde. Z.B. zeigt die Einstellung des Perception Range Attribute Mittel an, dass der Charakter den Spieler beachten soll, wenn dich mit einer Strecke 35 Metern (oder kleiner) dich näherst, Egal ob der Charakter wird zum Spieler entsprechend ihres Profilattributeinstellungen reagieren, nur nachdem der Spieler sich mit einer Strecke von 20 Metern nähert, reagiert der Charakter dieses Beispiel wird oben durch das folgende Diagramm dargestellt:



- **Profiles.** Bestimmt das Verhalten eines Charakters je nach den auftretenden Umständen der Taten des Spiels oder des Spielers. Dieses Attribut enthält eine Serie von Optionen, die von der Liste gewählt und hervorgehoben werden:



Die Liste ist ziemlich lang und kann mit Hilfe der Leiste auf der rechten Seite der Liste navigiert werden. Die Optionen die verfügbar sind können in folgenden Kategorien eingeteilt werden:

- *Attitude* – definiert die Einstellung vom Charakter zum Spieler
- *Friendly* – Charakter hat eine freundliche Einstellung zum Spieler und greift ihn nicht an,
- *Neutral* – Charakter greift den Spieler nicht an, jedoch der Spieler kann den Charakter angreifen,
- *Hostile* – Charakter wird in Richtung des Spielers feindlich gesinnt sein und greift ihn an, wann immer es ihm möglich ist
- *Fistfighter* – Charakter nimmt an Faustkämpfen teil
- *Affiliation* – definiert die Gruppe, an dem ein Charakter angegliedert ist. Charaktere gehören zur gleichen Gruppe die Helfen in einem Kampf. Sie werden auch von Charakteren angegriffen, die sind aber in der Richtung dieser Gruppe feindlich:
- *Order*, (Kommando)
- *Inhuman* – Nicht-Menschen-Geschlechter (Kobold, Zwerg),
- *Temeria* Bewohner von Temerien,
- *Guard* – (für Duelle auch wichtig Charaktere mit diesem Stick helfen den Bürgern,)
- *Commoner*,
- *Witcher*, (Hexer)
- *Sorcerer*, (Zauberer)
- *Animal*, (Tier)
- *Outlaw_Bandit*, (Bandit)
- *Outlaw_City* – (geächteter Stadtbewohner,)
- *Outlaw_Salamander* – (geächtetes Mitglied der Salamandra,)
- *Outlaw_Scoiatael* – (geächtetes Mitglied der Scoia'tael Kommandos,)
- *Monster_Undead*, (Untote)
- *Monster_Magic* (magisches Monster,)
- *Monster_Experiment*, (experimentelles Monster,)
- *Monster_Animal* – (tierisches Monster,
- *Bandit_Neutral* – (neutraler Bandit,)
- *Lawful*, (Gesetz)
- *Vodyanoi* – (Charakter gehört zu den Vodyanoi,)

- *Kikimora_Pheromones* – (Character wird von einem Kikimoren Krieger markiert und wird angegriffen durch die Anderen Kikimores, die normalerweise neutral sind,)
- *Boss* –(eine spezielle Kategorie, um ein Boss im Spiel zu beschreiben)
- *Local1...Local* – (kann als Definition seiner eigenen Gruppe dienen,)
- *Monster_Ornitoreptile* (kriechendes Monster (Basilisk, Basilisk, Wyvern),
- *Monster_Insectoid* – (Insekten-mag Monster,)
- *Monster_Carrion* – (Ass-mag Monster,)
- *Monster_Spectre*, (Monster, Geister)
- *Zeugl.(?)*
- *Behavior* –definiert das Verhalten eines Charakters während Kampfs:
- *Aggressive* –Charakter versucht, an einen Feind schnell heranzugehen,
- *Defensive* – Charakter versucht, seine Position beizubehalten; wenn der Feind erfolglos, sich nähert
- *Coward* angegriffener Charakter flüchtet.
- *Tactics_DontSurround* – by Default, die Charaktere im Spiel, versuchen einen Rivalen in einer Weise zu umzingeln, damit einer mindestens von hinten angreift. Diese Einstellung der Wahl lässt die angreifenden Feinde das Ziel nach dem Zufall umzingeln.
- *Role* –definiert, welche Rolle ein Charakter spielt:
- *Victim* –Charakter wird Opfer
- *Leader* –Charakter wird ein führender Anführer sein,
- *Healer* –Charakter heilt die Wunden von anderen Charakteren,
- *Shooter* – Charakter greift bei einer Reichweite mit (Bogenschützen) an,
- *PC_Follower* – Character folgt dem Spieler,
- *DoNotHelpFriends* –Charakter hilft keinen Freunden,
- *IgnoreAll* –Charakter ignoriert jeden
- *IgnoreMe* –Charakter ignoriert Spieler,
- *PCAttacker*,
- *NPC_Follower* –Charakter folgt einem anderen Charakter,
- *DoNotHelpMe* –Charakter hilft keinem Spieler,
- *Alcohol* – definiert wie schnell ein Charakter betrunken wird:

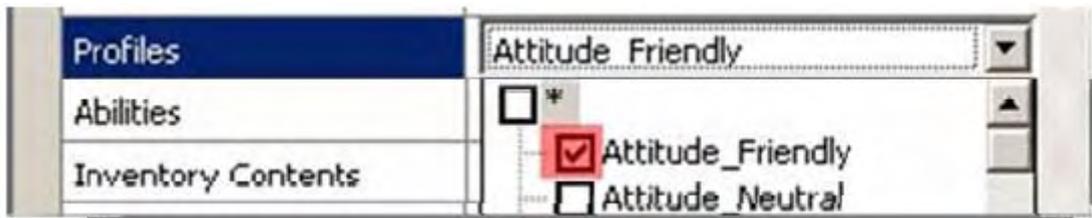
- *Mild_3* –schwacher Alkohol, Betrunken nach 3 Getränken
- *Mild_5* – schwacher Alkohol, Betrunken nach 5 Getränken,
- *Mild_7* –schwacher Alkohol, Betrunken nach 7 Getränken,
- *Medium_3* –mittlerer Alkohol, Betrunken nach 3 Getränken
- *Medium_5* –mittlerer Alkohol, Betrunken nach 5 Getränken
- *Medium_7*–mittlerer Alkohol, Betrunken nach 7 Getränken
- *Strong_3* – starker Alkohol, Betrunken nach 3 Getränken,
- *Strong_5* -starker Alkohol, Betrunken nach 5 Getränken,
- *Strong_7* – starker Alkohol, Betrunken nach 7 Getränken.
- *Enemy* – definiert, der Faction des Charakters in welcher Richtung sie feindlich sind. Sie Arbeiten mit einer Zugehörigkeit in den Affiliations Einstellungen.

Der Charakter nimmt jedermann in Angriff dessen Affiliations Liste mindestens eine Eintragung hat, die zu den Fraktionen der Charaktere unterschiedliche Verbindung hat. Die einzelnen Auswahlen sind dieselben wie in der Affiliation:

- *Species* –definiert die Art eines Charakters:
- *Sheep, Schaf*
- *Dog, Hund*
- *Echinops*.
- *Internal_Charmed* – Charakter ist unter einen Zauber
- *AfraidOf*–definiert, welche Gruppen einen Charakter fürchtet und zu flüchten versucht. Die Werte, die Du in diesem Attribut zuweisen kannst ,sind die gleichen wie in *den Affiliationen*
- *Ignore_Rain* –Charakter ignoriert Regen und versucht nicht, sich vor ihm zu verstecken,
- *Limited_Attackers_1*.
- *Limited_Attackers_2*.
- *Limited_Attackers_3*.
- *Immobile* –Charakter bewegt sich nicht,
- *Formation* – Formation, in der sich ein Charakter bewegt. Dies betrifft die Charaktere, gehören zu denselben Gruppen und zu den Akteuren wie im einen Teil einer Gruppe, z.B. Patrolieren:
- *Column*,
- *Circle*,

- Line.

Eine große Vielfalt von Profilkonfigurationen ist mit verfügbaren umfangreichen Optionen möglich. Wir sollten jetzt zu Siegfried zurück kommen und sein Profil einstellen. Du Wählst aus, indem du das kleine Kästchen nahe bei der Profilwahl anklickst:



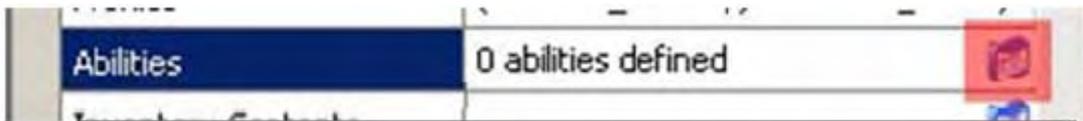
Du Klickst jetzt auf das vorher ausgewählte Profil un-check. Um alle Profil-Optionen einzustellen (nicht empfohlen), Du Klickst einfach auf das kleine Kästchen, das nah bei dem Sternchen zu finden ist:



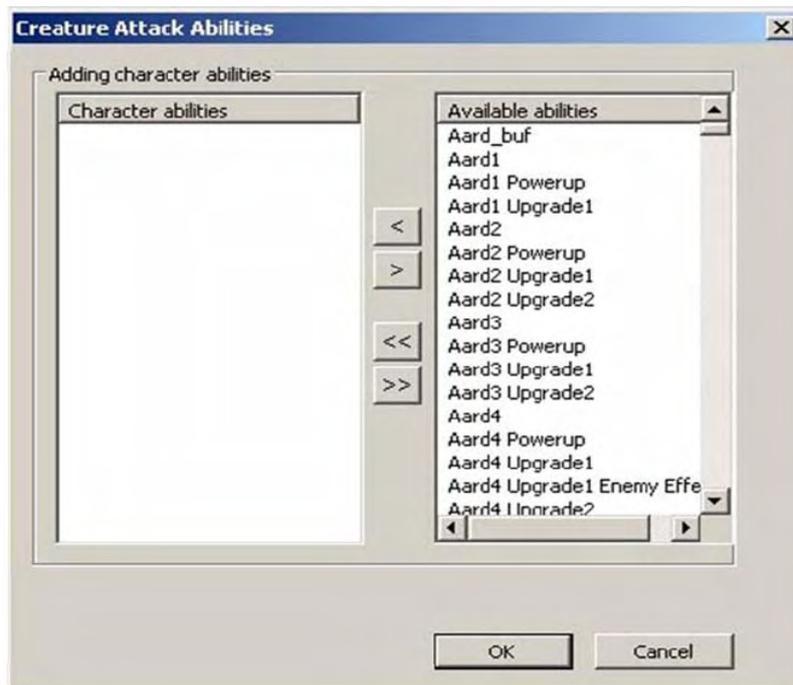
Du setzt folgende Profilooptionen für Siegfried:

Attitude_Friendly, Affiliation Order Affiliation_Witcher, Behavior Aggressive, Tactics_DontSurround, Enemy_Outlaw_Bandit, Enemy_Outlaw_Salamander, Enemy_Monster_Undead, Enemy_Monster_Magic, Enemy_Monster_Experiment, Enemy_Monster_Animal.

• **Abilities.** Bestimmt die Charakterstatistiken, d.h. Charakterfähigkeiten. Um die Fähigkeiten eines Charakters zu setzen, klickst Du auf das Blaue Icon das findest Du im rechten Spaltenteil bei *Abilities* Attribute. Das öffnet das *Creature Attack Abilities Fenster*:

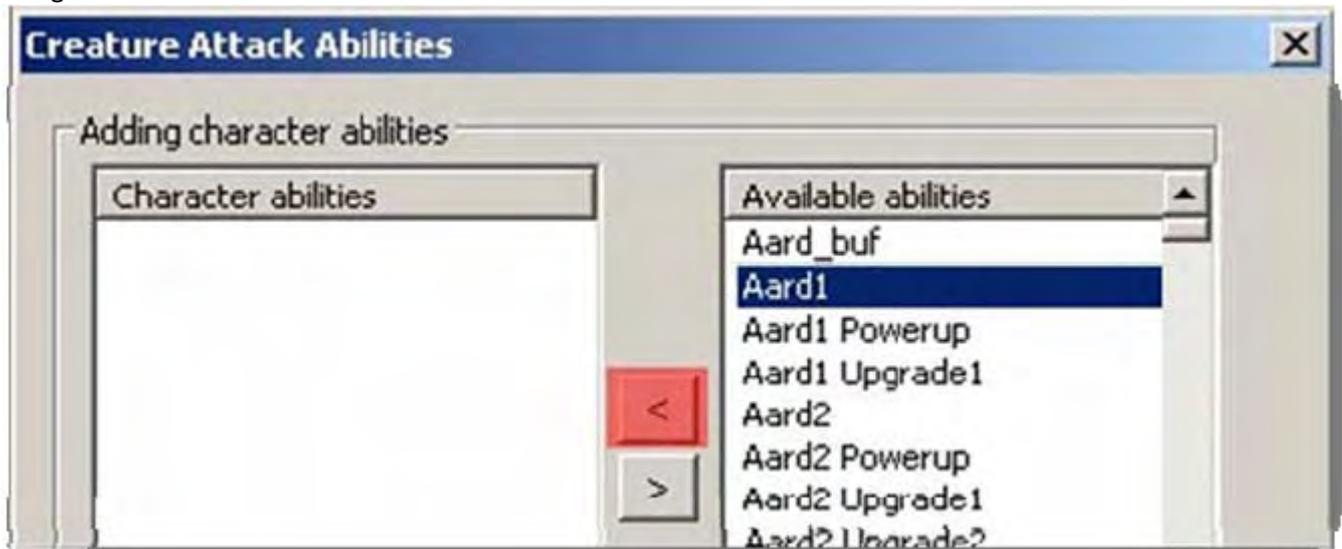


Das öffnet das *Creature Attack Abilities Fenster*:

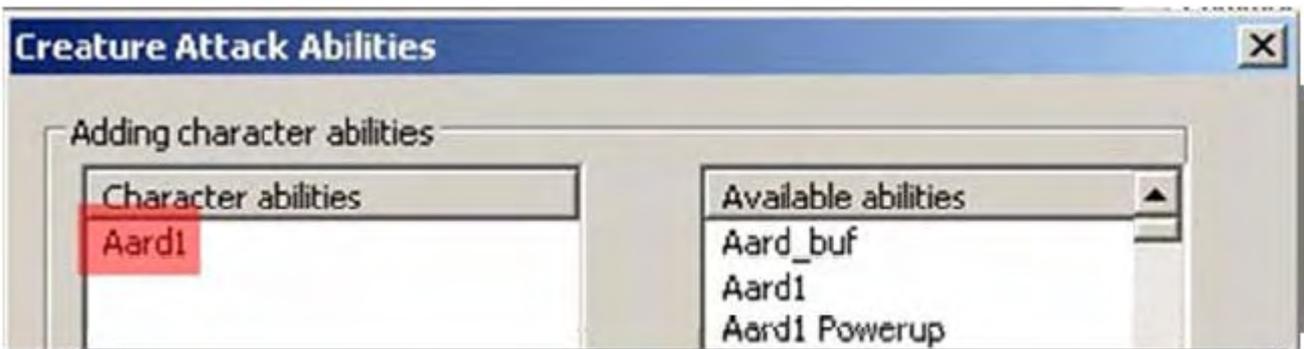


Alle Fähigkeiten, die z.Z. den Charakteren zugewiesen wurden, sind im Character Abilities Panel, während alle vorhandenen Fähigkeiten im vorhandenen Abilities Panel. Jedes der Fähigkeiten ist verbunden mit den unterschiedlichen Statistiken wie die Lebensanzahl oder die Energiepunkte, oder der Immunität/Widerstand es gibt verschiedene Zustände wie z.B Widerstand dagegen, betäubt zu werden oder in ein Zauber versetzt.

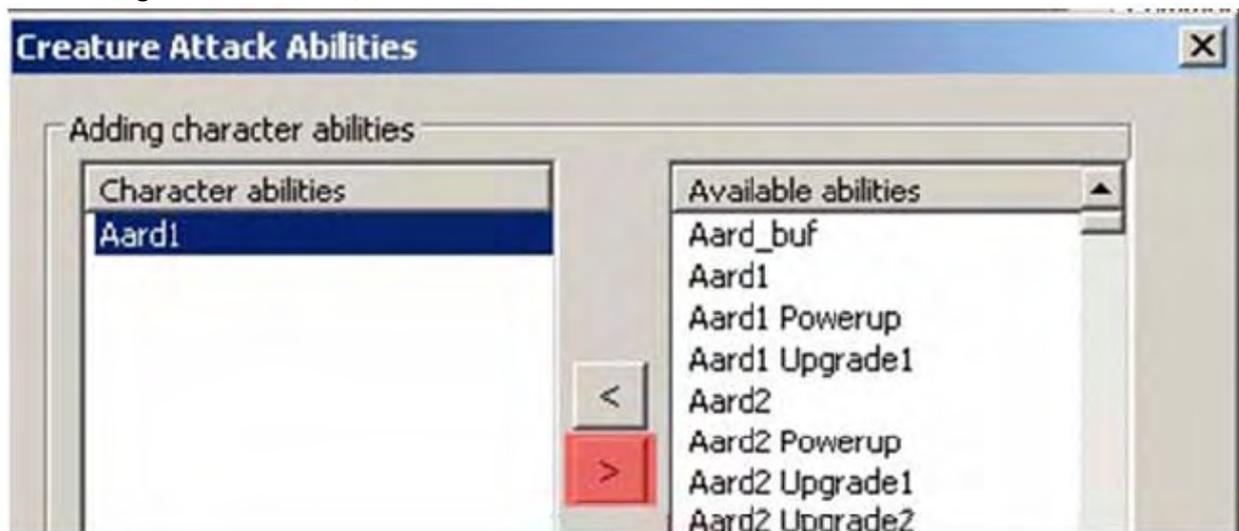
Um Fähigkeiten eines Charakter, hinzuzufügen wählst du eine passende Fähigkeit von den vorhandenen Fähigkeiten aus und klickst dann die Taste '<':



Die Fähigkeit erscheint automatisch im *Character Abilities* Panel, das anzeigt, dass du einen Charakter zuge- teilt hast:



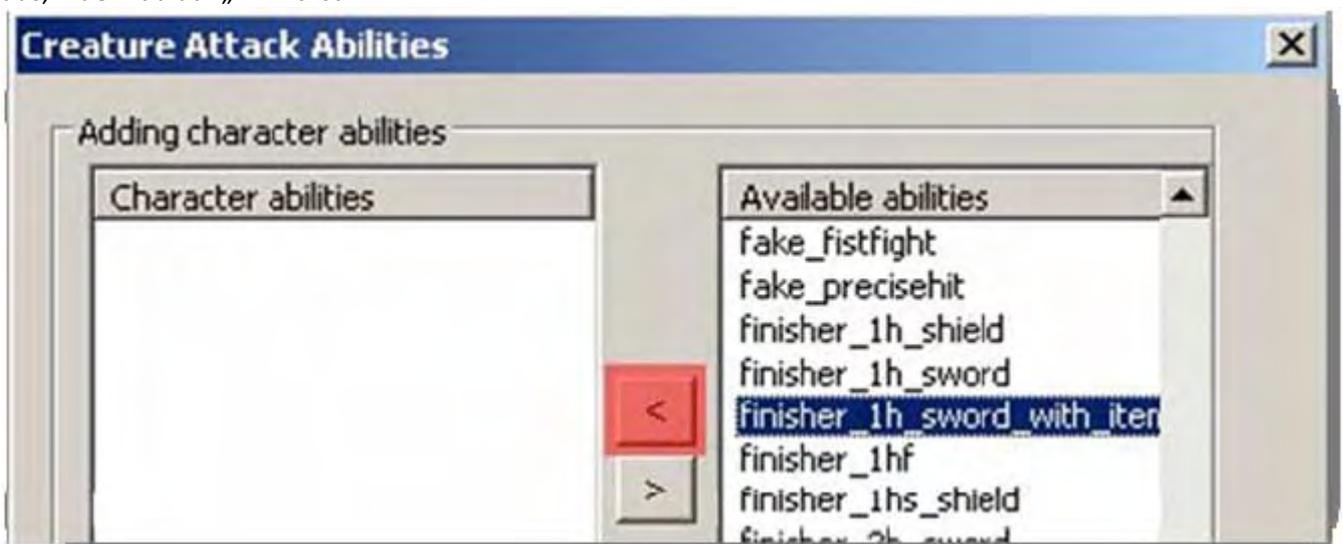
Um eine Fähigkeit zu entfernen, wähle es im *Character Abilities* Panel und drücke '>':



Mehr als eine Fähigkeit kann auch ausgewählt werden. Das kannst du so tun, wie in anderen Windows-Applikationen, durch das Klicken und Halten von CONTROL und SHIFT taste. Alle verfügbaren Aktionen sofort zuzuweisen durch das klicken von dieser Taste, '<<':



Das Klicken auf „>>“ entfernt alle Fähigkeiten vom Charakter. Die genaue Bedeutung aller Fähigkeiten wird nicht hier besprochen. Suche in der Liste nach „finisher_1h_sword_with_item“, und wähle es für Siegfried aus, indem du auf „<“ klickst:

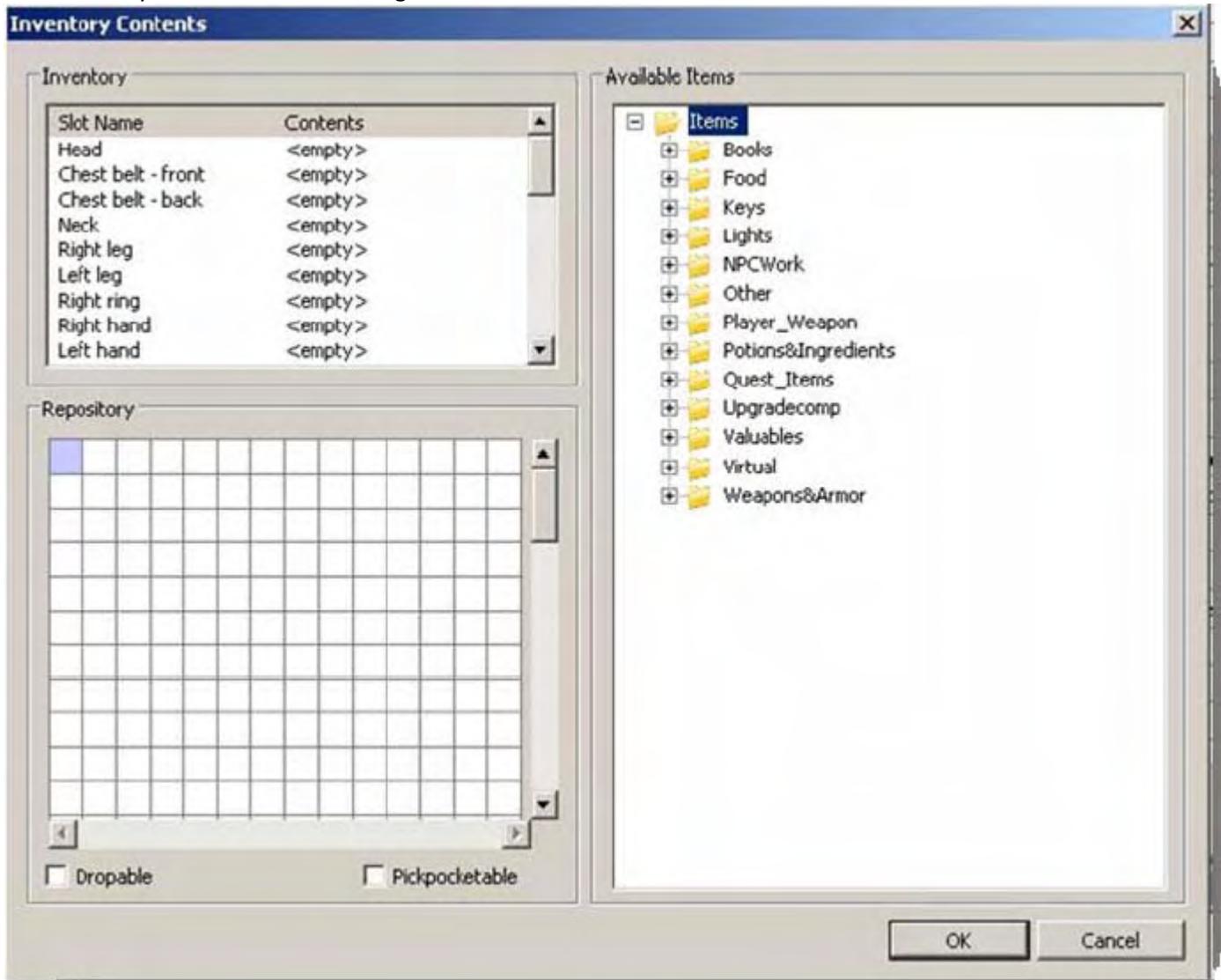


Du kannst auch auf den Ability Name Doppelklicken und Du bekommst dasselbe Ergebnis wie die Nutzung des '<' Knopfs. Weil diese Liste lang ist, kannst du die Suche etwas leichter machen, indem du „First Ability“ von dem Available Abilities Panel auswählst und dann 'F' tippst. Die Liste wird dann zu Abilities Ablage springen, die mit dem Kopf 'F' anfangen. Sobald du die Fähigkeit Siegfried zugeteilt hast, wähle „OK“ aus.

• **Inventory Contents.** Bestimmt das Inventar-item eines Charakters. Um ein Inventar für einen Charakter zu erstellen, klickst Du auf das kleine blaue Icon, die sich auf der rechten Spalte gegenüber vom *Inventory Contents* Attribute befindet:



Das Inventory Contents Fenster wird geöffnet



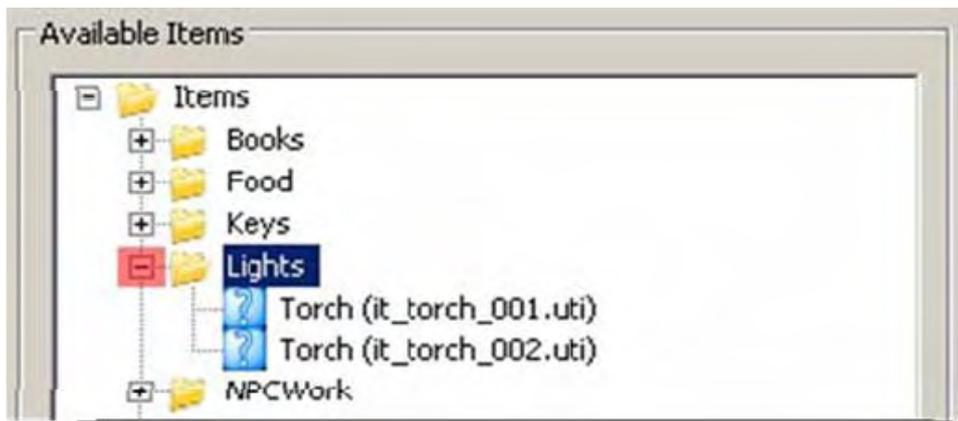
Drei Panels- können in diesem Fenster unterschieden werden:

- *Inventory* – Warenbestand eines Charakters aufgeteilt in Slots:
- *Head, Schädel*
- *Chest belt - front* – Riemen auf der Brust,
- *Chest belt - back* – Riemen auf der Brust,
- *Neck* – necklace, (Halskette)
- *Right leg,*
- *Left leg,*
- *Right Ring* – Ring in der recht Hand,

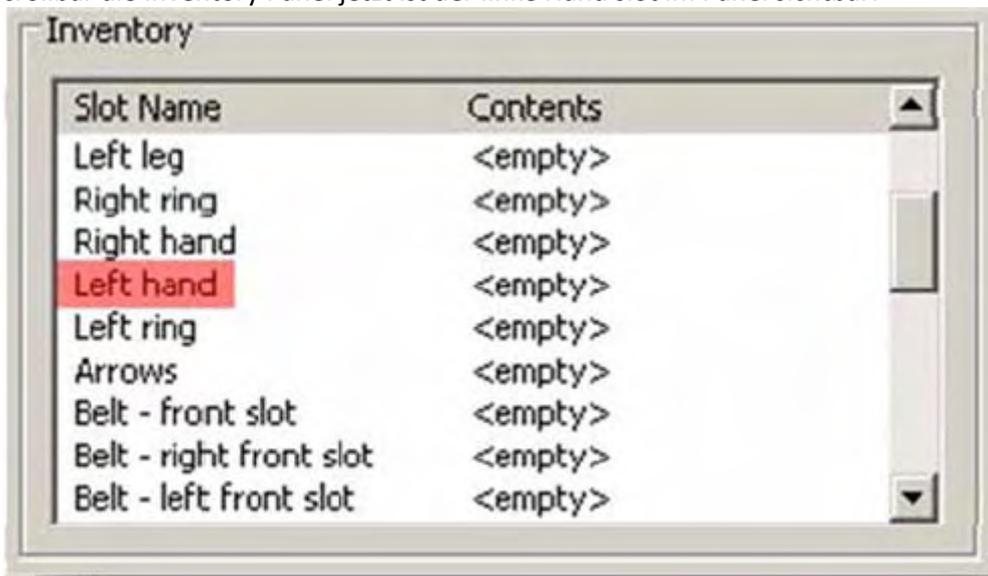
- *Right hand*,
- *Left hand*,
- *Left Ring* – Ring in der linken hand,
- *Arrows* – Raum für Pfeile
- *Belt - front slot* – front belt slot,
- *Belt - right front slot* – right front belt slot,
- *Belt - left front slot* – left front belt slot,
- *Belt - left slot* – left belt slot,
- *Belt - right slot* – right belt slot,
- *Silver sword* – Raum für das Silberschwert,
- *Steel sword* – Raum für das Stahlschwert,
- *Short weapon 1* – erster Raum für eine kurze Waffe,
- *Short weapon 2* –,der zweite Raum für eine kurze Waffe,
- *Big weapon* – der Raum für die große Waffe das sich auf der Rückseite befindet,
- *Trophy* - Raum für eine Trophäe (Haken) zeigt auf Geralt, - Medaillon,
- *Elixir 1* – erster Raum für Elixier,
- *Elixir 2* – zweiter Raum Für Elixier,
- *Elixir 3* – dritter Raum für Elixier ,
- *1st Creature Weapon*,
- *Creature Armor* – (betrifft nur Monster),
- Left arm,
- Right arm,
- *Repository* – Aufbewahrungsort eines Charakters (verwechsel das nicht mit dem Speichern eines Gasthaus-Inhabers). Dieses ist, wo die restlichen Items (Alchimiebestandteile, Bücher usw.) Die Charakter die etwas bei sich tragen, werden ihre Warenbestände lokalisieren.
- *Available Items* – die ganze Liste von den Items, die im Spiel verfügbar sind, befindet sich hier. Items werden in einer Baumstruktur in Kategorien eingeteilt und strukturiert und machen Die Orientierung und die Suche leichter, Items sind in folgenden Kategorien eingeteilt:
- *Books* – Bücher und alchemistische Rezepte

- *Foods*,
- *Keys*,
- *Lights* – verschiedene Lichtquellen, z.B. Taschenlampen
- *NPCWork* – spezielle Sachen, die von Charakteren verwendet werden, um Aktivitäten, z.B. Lauten auszuführen,
- *Other* – alle Restesachen, die nicht in die anderen Kategorien passen,
- *Player_Weapon* – die Waffen, des Spielers d.h. Stahl und silberne Schwerter
- *Potions&Ingredients* – Tränke und Zutaten; diese Kategorie ist in Unterkategorien eingeteilt:
 - *Base greases*,
 - *Bombs*,
 - *Drink* – Getränke einschließlich Alkohol,
 - *Greases*,
 - *Ingredients*,
 - *Potions*,
 - *Powders*,
 - *Uniques*,
- *Quest_Items* – Items, die mit Quest verbunden sind, (diese sind eingeteilt entsprechend die Taten im Spiel),
- *Upgradecomp* – Items , die verwendet werden, um Waffen zu verbessern, (Meteoriten und Runen),
- *Valuables* – Wertgegenstände (Orens, Ringe, usw.),
- *Virtual* – spezielle Typen von virtuellen Items. Jeder virtuelles Artikel hat sein eigenes Aufbewahrungsort, wo anderer echte Items als Beute (wie Bücher, Essen usw.) gefunden werden können. Wenn Geralt einen virtuellen Item findet, wird es vom Zufallsprinzip ausgewählt:
- *Weapons&Armor*.

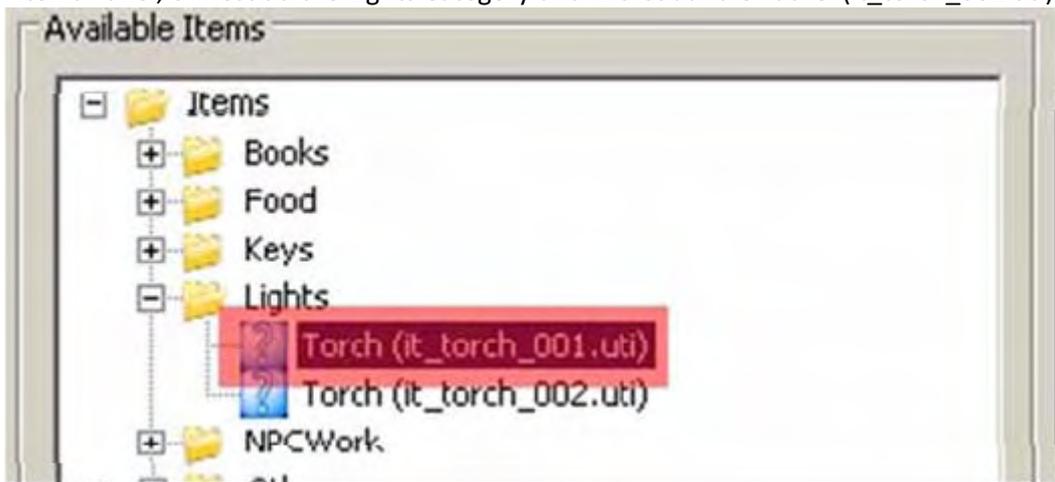
Um eine Kategorie zu öffnen und zu sehen welche Item es gibt, Doppelklickst du auf dem Namen oder auf das kleine ` + ' Zeichen, das nahe bei dem Namen dieser Kategorie zu finden ist:



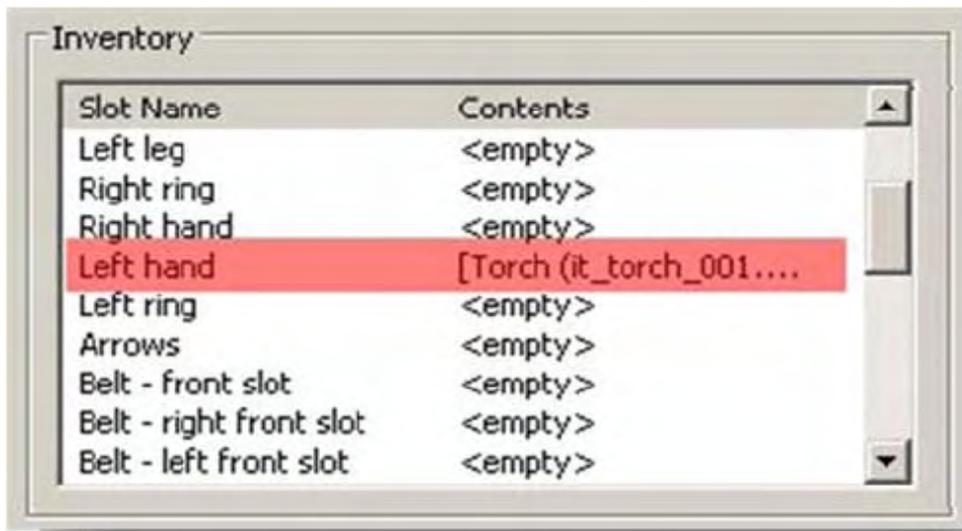
Warenbestand von Siegfried. Um mit ihm anzufangen, erhält er eine Fackel und eine Stahlklinge als Waffe. Beginnst du mit der Fackel und legst Sie sie in seine linke Hand. Um Items einfacher zu bilden, verschiebst du mit der Scrollbar die Inventory Panel jetzt ist der linke Hand slot im Panel sichtbar:



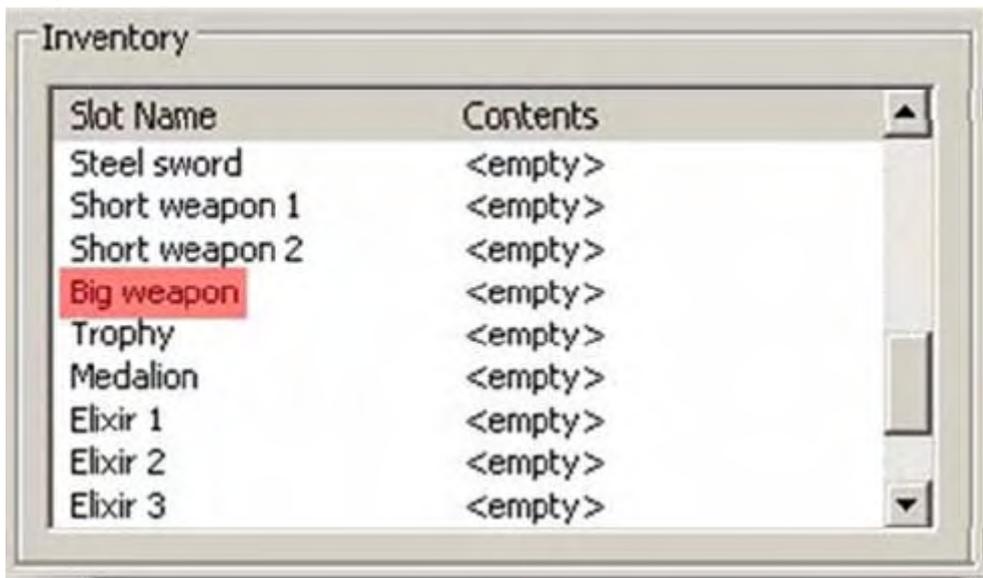
Gegenüber von der linken Hand ist ein Slot das ein Text, anzeigt, dass dieser Slot unbesetzt ist. Auf den vorhandenen Items Panel, öffnest du the Lights Category und klickst auf die Fackel (it_torch_001.uti):



Um eine Fackel in die Hand von Siegfried zu legen, klickst du nach links und fügst die Fackel (it_torch_001.uti) Item zum linken Handablagelot hinzu. Wenn sich der <leere> Ausdruck geändert hat, fügen wir die Fackel (it_torch_001.uti)> ein, und du drückst dafür die linke Maustaste. Siegfried hat jetzt eine Fackel erhalten, die er in seiner linken Hand tragen wird:



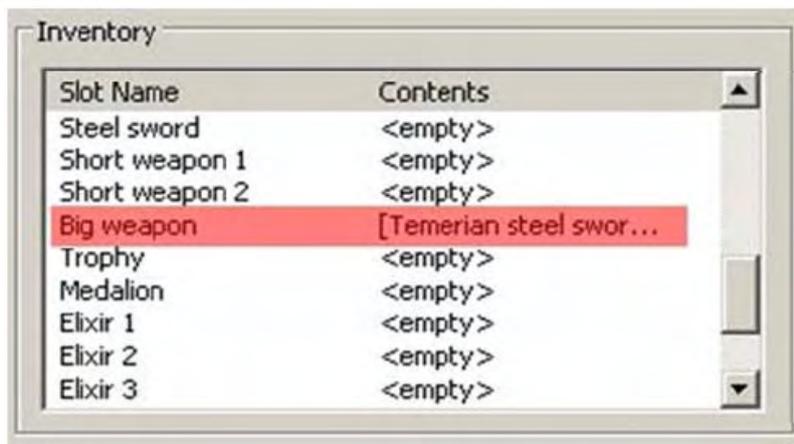
Wenn ein Item nicht in einen bestimmten Slot gelegt werden kann, verändert der Cursor sich zu einer Aktion verweigertes Zeichen. Wenn die Items passen in einen bestimmten Slot, verändert sich der Cursor zu einem Handzeichen. Du Wiederholst die gleichen Schritte also rüstest du Siegfried mit einer Klinge. Mit der Scrollbar des Inventory Panel, bewegst du es nach unten, damit der große Waffenslot, d.h. die große Waffe, in diesem Panel sichtbar wird:



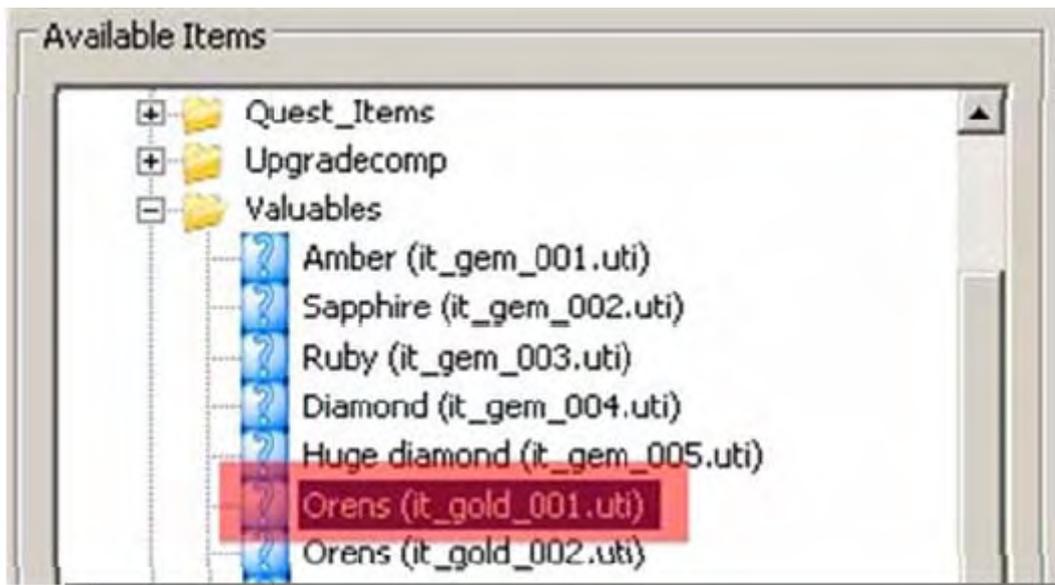
In dem verfügbaren Items-Panel öffnest du die Waffen & Panzer Kategorie und wählst dann das Temerian Stahlschwert (it_stlswd_005.uti) aus:



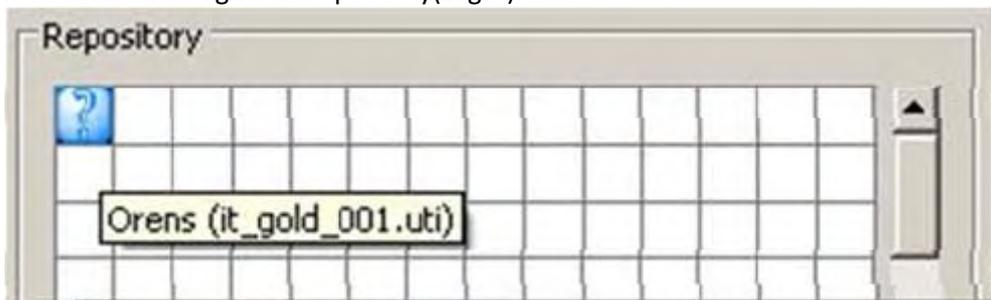
Wie mit einer Fackel, ziehst du das Temerian Stahl Item der Klinge (it_stlswd_005.uti) auf den großen Waffenslot und lässt dann die linke Maustaste los:



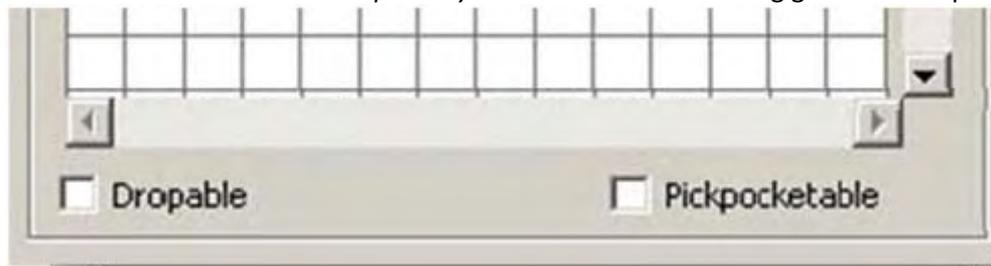
Siegfried ist jetzt mit einem Schwert bewaffnet. Es wäre nützlich, wenn er auch einige Orens hätte, (Währung des Witcher's). Um ihn mit irgendeinem Orens auszurüsten, öffnest du die nützliche Kategorie in Available Items Panel und wählst die Orens (it_gold_001.uti) aus:



Jetzt ziehst du die Orens (it_gold_001.uti) Item zum Repository Panel und dann lässt du linke Maustaste Los. Das Item erscheint im Siegfrieds Repository(Lager):



Es gibt noch zwei weitere Auswahlen im *Repository* Panel in den die ausfindig gemachten Optionen:



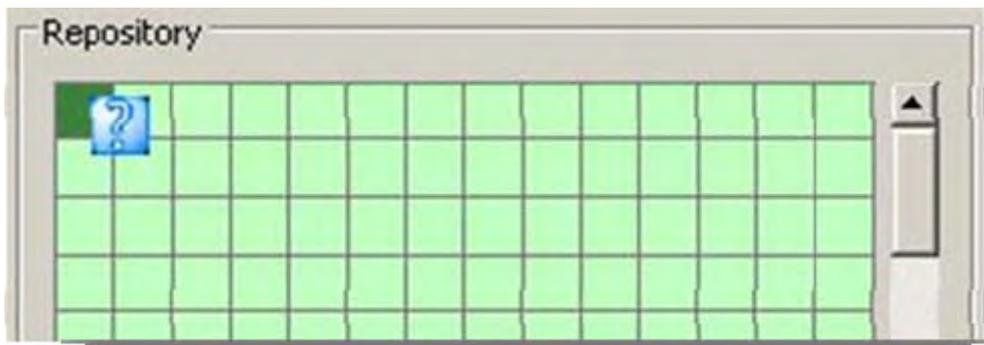
Zwei Optionen

- *Dropable* – ein Item vom Lager kann auch vom Charakter fallen gelassen werden.
- *Pickpocketable* – ein Item vom Lager kann auch in den Auswahlen Platziert werden.

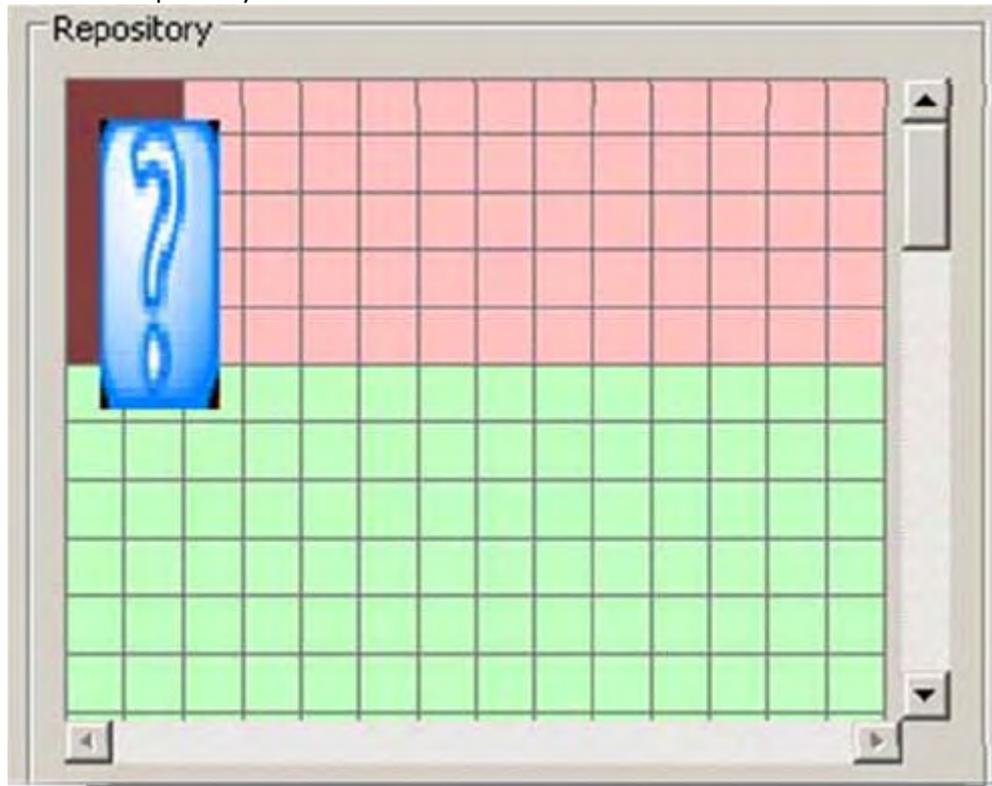
Um ein Item vom Lager, in einen *Repository* zuzuweisen, Klickst du auf das Item in dem *Repository* Panel und wählst dann die passende Auswahl in den Quadraten aus:



Wenn du die (it_gold_001.uti) Items zur Repository Panel ziehst ändert sich die Farbe des Hintergrundes merklich:



Das zeigt an ob, das Item im Lager des Charakters überall abgelegt werden kann. Versuche das Temerian Stahlitem der Klinge (it_stlswd_005.uti) auf das Repository Panel zu verschieben. Dieses Mal, hat sich die Hintergrundfarbe des Repository Panel Rot und Grün verfärbt:



Das zeigt an, dass dieses Item an einem spezifischen Platz im Lager des Charakters gelegt werden muss, in diesem Fall in die unterste Hälfte des Panels.

Selbstverständlich können Items vom Lager des Charakters auch entfernt werden. Das kannst du, indem du das Item auswählst, vom Inventory Panel oder du entfernst es vom Panel, in dem du es vom der Repository ziehst und dann die Taste loslässt.

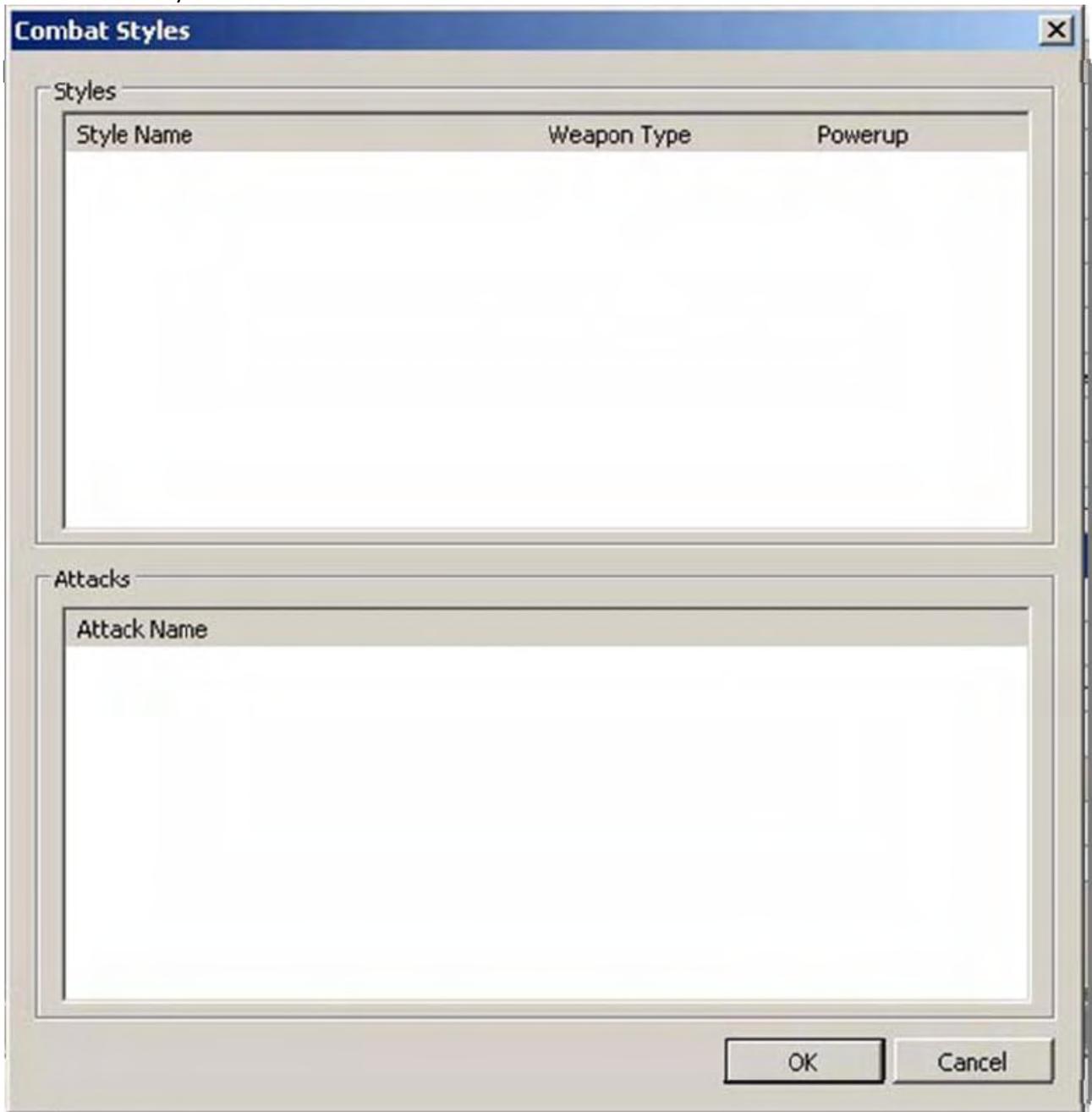
Nachdem du damit fertig bist, von Siegfrieds Lager Die Items hinzuzufügen, wählst du Ok unten im Inventar Inhalts fenster.

- **Combat Styles.** Definiert die Kampffarten, die durch einen Charakter verwendet wurden. Um die Kampffart zu definieren, Klickst du auf das kleine blaue Icon zu finden in der rechten spalte im Combat Styles Attribute:

Das öffnet das Combat Style Fenster:



Das öffnet das Style Fenster:



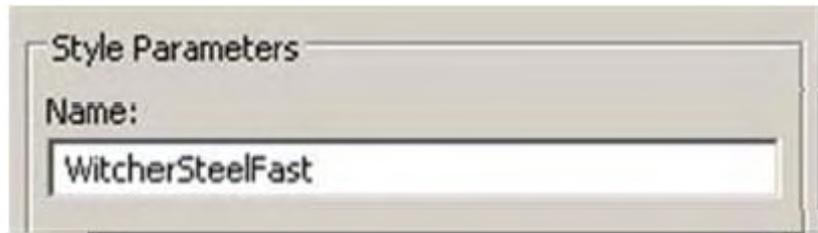
Das Fenster wird in zwei Panels geteilt: Die Styles, in denen die Kamparten und die Waffe, die einer bestimmten Art zugewiesen werden ,angezeigt werden und den Style, auf der Liste der Angriffs Sequenzen , die auf einen bestimmten Style zugewiesen wurden werden angezeigt. Ein Charakter kann einige Kamparten, die, abhängig von der Waffe, die definiert wurde ausgerüstet, und einige Angriffe haben, die für jede Kampart zugewiesen werden. Das betrifft nicht die Monster, die normalerweise eine Kampart und einen Angriff haben, die ihnen zugewiesen wurden. Um eine neue Kampart hinzuzufügen, Rechtsklicke im Style Panel, und wähle „Add Style“ aus dem Menü:



Hinzufügen eines Stylesnamens im Namensfeld .jedoch wird der Name nach Bedarf eingeteilt im Style listen-Fenster:

- *WitcherSteelFast* - ein Kampfstil verwendet einhändige Schwerter,
- *ShortWeaponFast* –ein Kampfstil, der Äxte und Knüppel verwendet
- *SwordWithShield* – ein Kampfstil, der eine Schild und eine einhändige Waffe benutzt,
- *TwoHanded* – ein Kampfstil, der eine zweihändige Waffe benutzt
- *PolearmHeavy* –ein Kampfstil, der ein polearm verwendet
- *TwoSwordFast* -ein Kampfstil, der zwei Waffen, hauptsächlich zwei Schwerter benutzt,
- *DaggerFast* –ein Kampfstil, der einen Dolch benutzt,
- *Bow* – ein Kampfstil, der einen Bogen verwendet,
- *Crossbow* – ein Kampfstil, der eine Armbrust verwendet,
- *Commoner* – ein von ungelerten Opponenten wie Banditen oder Miliz verwendeter Kampfstil,
- *Monster* –ein für Monster reservierter Kampfstil,
- *SpellRanged* – ein Kampfstil, der Zauberei verwendet,
- *FistfightCommoner* – eine Faust, die Kampfstil bekämpft.

Für Siegfried weist Du einen einhändigen Schwertkampfstil zu. Nach *WitcherSteelFast* im Namensfeld suchen und auswählen:

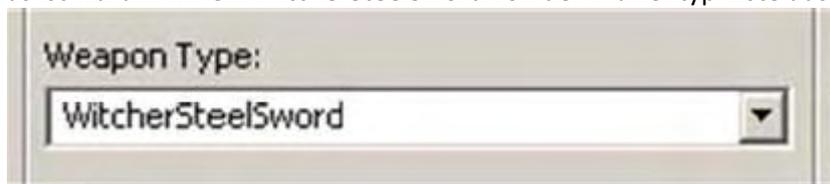


Style Parameters

Name:

WitcherSteelFast

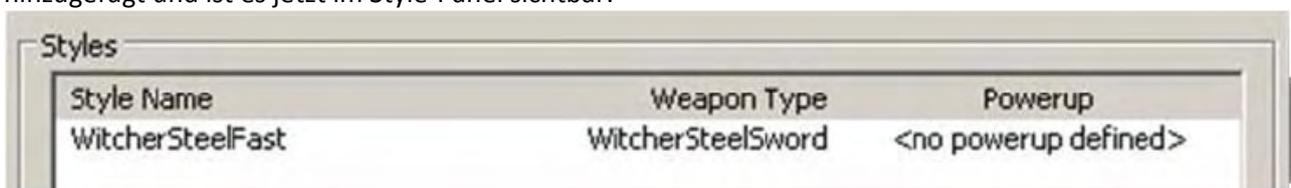
Von der Waffentyp-Auswahl-Liste, wählst du einen Waffentyp aus. Das muss dem im Namenfeld eines Types entsprechen, das ist warum wir ein *WitcherSteelSword* von der Waffentyp-Liste auswählen werden:



Weapon Type:

WitcherSteelSword

Die Powerup Liste werden nicht verwendet. Klickst du jetzt OK jetzt wurde eine Kampfarmt von Siegfried hinzugefügt und ist es jetzt im Style-Panel sichtbar:



Style Name	Weapon Type	Powerup
WitcherSteelFast	WitcherSteelSword	<no powerup defined>

Klickst du rechts auf den frisch hinzugefügten Kampfstil, um ein umfassendes Menü zu öffnen:

Die verfügbaren Menüauswahl Möglichkeiten:

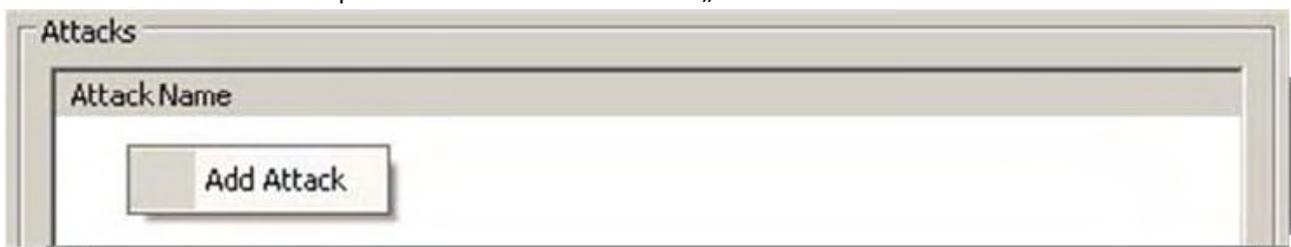
- *Add Style* – hinzufügen eines anderen Kampfstils,

- *Edit Style* - Du editierst den ausgewählten Kampfstil, und öffnest wieder das *Stilfenster*, so dass Korrekturen gemacht werden, (auf den Kampfstil zu doppelklicken, hat die selbe Wirkung).

- *Remove Style* – entfernt den gewählten Kampfstil (auf den Kampfstil und wenn du auf deiner Tastatur „Del“ drückst, hat das die selbe Wirkung).

Mehrere Kampfstile können für jeden Charakter geschaffen werden, je nach der Art der Waffe das dieser Charakter führt. Ein Stil für den Kampf für Siegfried genug.

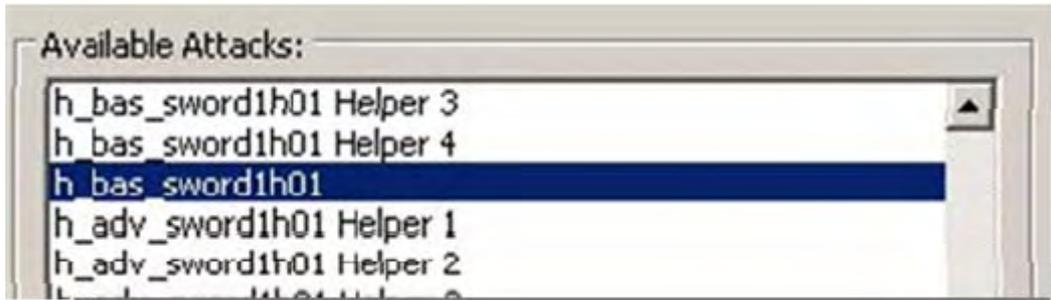
Jetzt fügst Du einige Angriffe zu diesem Kampfstil hinzu. Um das zu tun, wähle den Kampfstil aus und ein Rechtsklick auf dem *Attackspanel*. Wähle in diesem Menü „Add Attack“ aus dem Menü:



Das öffnet das Angriffs Fenster:



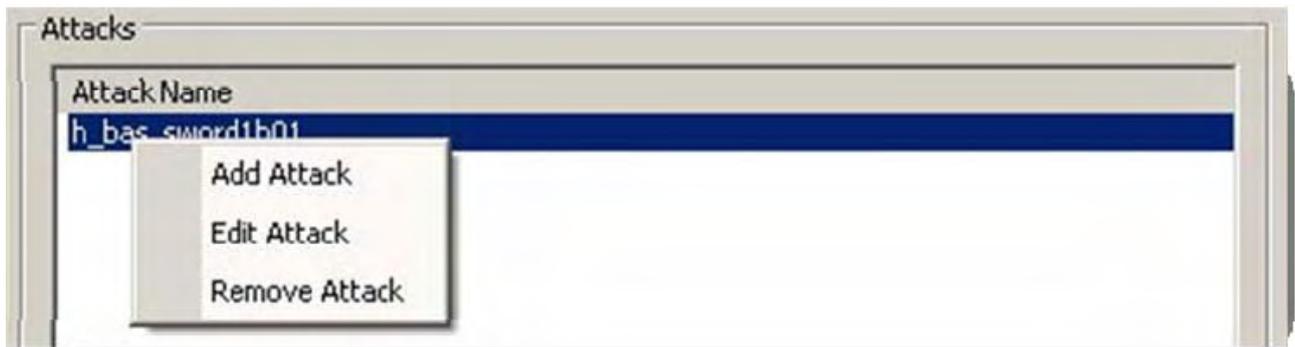
Die Details der einzelnen Attacken werden hier nicht näher beschrieben. Wähle h_bas_sword1h01 aus der Liste aus:



Dies ist die Standard-Attacke für ein Einhandschwert. Wenn du auf „OK“ klickst, erscheint die Attacke im „Attacks Panel“:



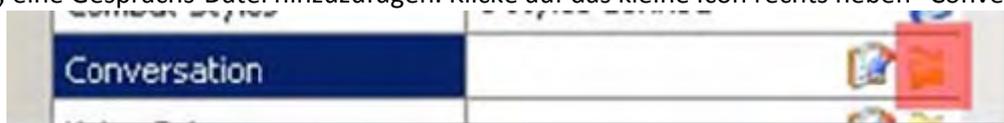
Klickst du nun, genau wie bei Kampfstilen, mit rechts auf neue Attacke, die wir erstellt haben, wird das gleiche Menü erscheinen:



Mehrere Attacken können zu einem Kampfstil hinzugefügt werden, praktisch eine Attacken-Sequenz. Siegfried wird in unserem Adventure nicht kämpfen, also wird eine Attacke genügen, um zu demonstrieren, wie man einen Kampfstil erstellt.

• **Conversation.** Verantwortlich für das „Conversation-File“ eines Charakters. Hier muss nicht eingefügt werden ,wenn der Spieler nicht mit anderen Charakteren redet. Wenn der Spieler auf irgendeine Art mit anderen Charakteren kommunizieren möchte, muss hier eine Datei eingefügt werden.

Da Siegfried ein Schlüssel NPC in unserem Abenteuer ist, wird der Spieler die Möglichkeit haben, mit ihm zu sprechen. Jedoch werden Gespräch-Dateien für die Schlüssel NPCs wo anders hinzugefügt. Gespräch-Dateien werden mehr unabhängigen Charakteren wie Stadteinwohner, Wächter usw. zugeteilt. Ein anderes Kapitel dieses Handbuches beschreibt, wie Gespräch-Dateien geschaffen werden. Alles, was wir noch tun müssen, ist, eine Gesprächs-Datei hinzuzufügen. Klicke auf das kleine Icon rechts neben "Conversation":



Dies öffnet das Fenster um Dateien zu laden. Gehe in das Verzeichnis Data\Dialogues, das alle Dialoge enthält, wähle die Datei, die du suchst, aus und klicke auf "Open".

- **Voice Sets.** Verantwortlich für das „Conversation-File“ eines Charakters. Der Unterschied hier ist, dass die Texte durch den Charakter in speziellen Spielsituationen gesprochen werden. Ein anderes Kapitel setzt sich mit diesem Thema auseinander. Um ein "Conversation-File" zuzuweisen, klicke auf das kleine gelbe Icon rechts neben "Voice Set":



- **Runs to despawn.** Legt fest, ob ein Charakter zu dem Punkt, an dem er aus dem Spiel verschwindet, rennt oder geht. Um die Einstellung zu ändern, klicke auf die Spalte rechts neben „Runs to despawn“:

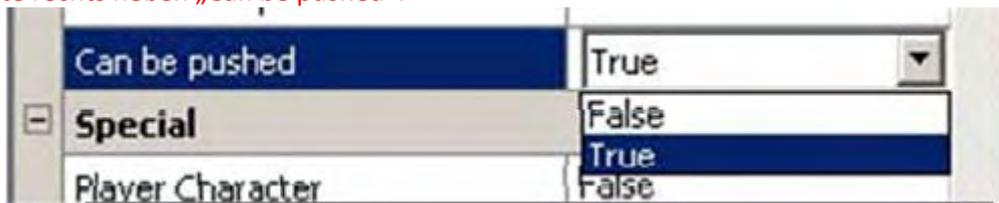


Die Werte, die du diesem Attribut zuteilen kannst, sind:

False Der Charakter geht zum Punkt.

True Der Charakter rennt zum Punkt.

- **Can be pushed.** Legt fest, ob ein Charakter geschubst werden kann. Um die Einstellung zu ändern, klicke auf die Spalte rechts neben „Can be pushed“:



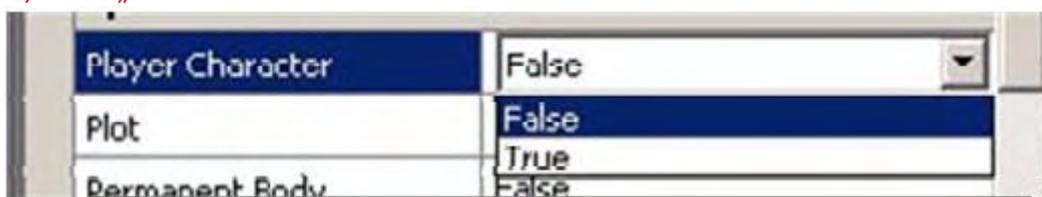
Die Werte, die du diesem Attribut zuteilen kannst, sind:

False Der Charakter kann nicht geschubst werden.

True Der Charakter kann geschubst werden.

Special Section:

- **Player Charakter:** Legt fest, ob ein Charakter als Spieler benutzt werden kann. Um die Einstellung zu ändern, klicke auf das kleine gelbe Icon rechts neben „Player Charakter“. Da Siegfried nicht als Spieler benutzt werden soll, wähle „False“:

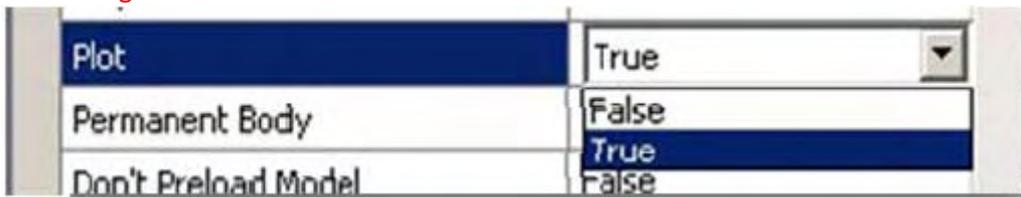


Die Werte, die du diesem Attribut zuteilen kannst, sind:

False Der Charakter kann nicht als Spieler benutzt werden.

True Der Charakter kann als Spieler benutzt werden.

- **Plot.** Legt fest, ob ein Charakter wichtig für den Plot ist. Um die Einstellung zu ändern, klicke auf die Spalte rechts neben „Plot“. Wenn dieses Attribut auf „True“ gesetzt wird, wird man den Charakter nicht töten können. Wenn die Lebensenergie dieses Charakters auf einen kritischen Wert sinkt, wird er auf den Boden fallen und von allen Gegnern ignoriert werden. Nach dem Kampf wird er wieder aufstehen, allerdings mit geringer Lebensenergie:

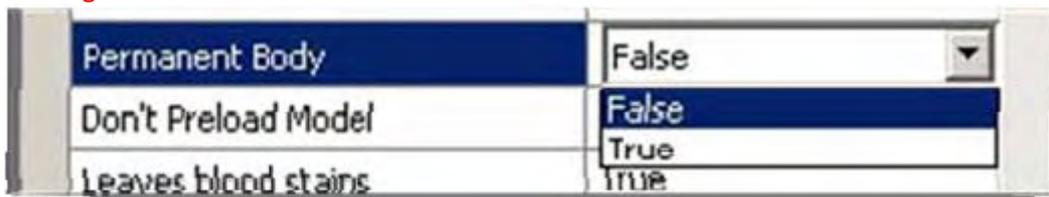


Die Werte, die du diesem Attribut zuteilen kannst, sind:

False Der Charakter kann getötet werden.

True Der Charakter kann nicht getötet werden.

- **Permanent Body.** Legt fest, ob der Körper eines Charakters nach dem Tod zu einem Skelett wird oder sein Aussehen behält. Um die Einstellung zu ändern, klicke auf die Spalte rechts neben „Permanent Body“. Wähle „False“ für Siegfried:



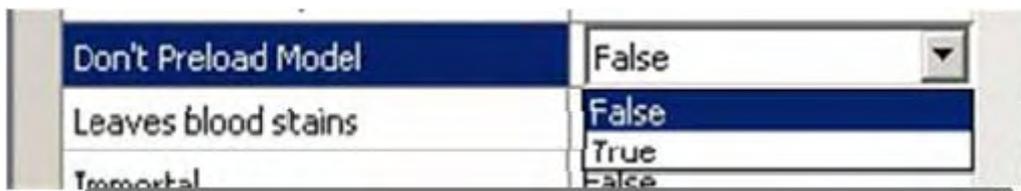
Die Werte, die du diesem Attribut zuteilen kannst, sind:

False Der Charakter wird nach seinem Tod zu einem Skelett

True Der Charakter behält sein Aussehen.

Es wird empfohlen, dieses Attribut auf „False“ zu setzen, um exzessive Ladezeiten zu vermeiden.

- **Don't Preload Model.** Legt fest, ob das Modell eines Charakters vorher geladen werden soll. Um die Einstellung zu ändern, klicke auf die Spalte rechts neben „Don't Preload Model“. Es wird empfohlen, „False“ zu wählen:



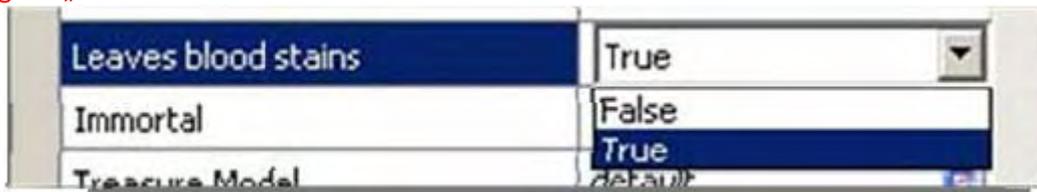
Die Werte, die du diesem Attribut zuteilen kannst, sind:

False Wird nicht vorher geladen.

True Wird vorher geladen.

Die Option „True“ wird manche Dinge beschleunigen, wird aber die Ladezeit in manchen Abschnitten verlängern.

- **Leaves blood stains.** Legt fest, ob Blut aus Wunden auch nach dem Kampf auf dem Charakter bleibt. Um die Einstellung zu ändern, klicke auf die Spalte rechts neben „Leaves Blood Stains“, Stelle bei Siegfried die Einstellung auf „True“:

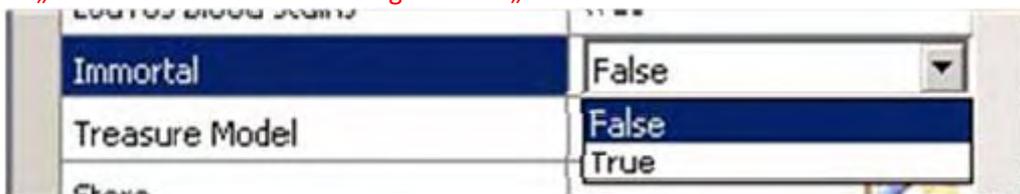


Die Werte, die du diesem Attribut zuteilen kannst, sind:

False Blut verschwindet.

True Blut bleibt.

- **Immortal:** Legt fest, ob ein Charakter unsterblich ist. Um die Einstellung zu ändern, klicke auf die Spalte rechts neben „Immortal“. Setze dies für Siegfried auf „False“:

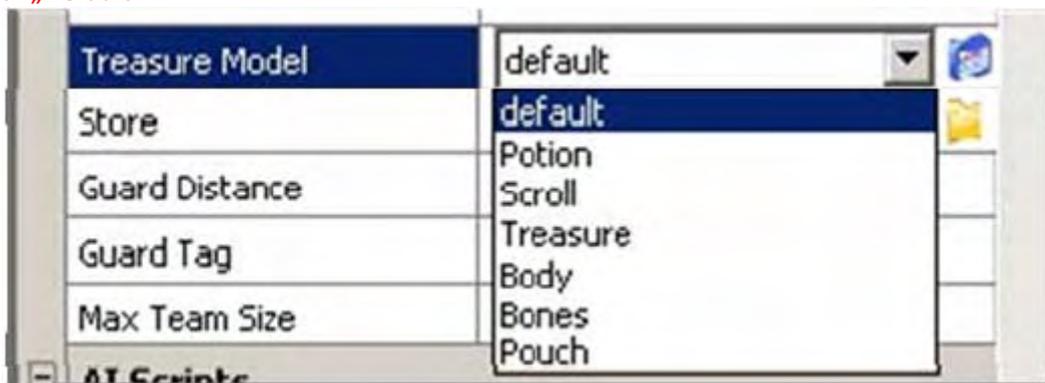


Die Werte, die du diesem Attribut zuteilen kannst, sind:

False Sterblich

True Unsterblich

- **Treasure Model.** Verantwortlich dafür, was erscheint, nachdem ein Charakter ermordet wurde. Um die Einstellung zu ändern, klicke auf die Spalte rechts neben „Treasure Model“. Da Siegfried nicht sterben soll, setze es auf „Default“:



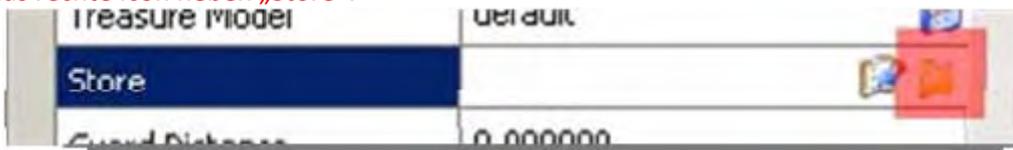
Die Werte, die du diesem Attribut zuteilen kannst, sind:

- Default – Die Standard-Einstellung
- Potion – ein Trank wird erscheinen
- Scroll – eine Schriftrolle wird erscheinen
- Treasure –ein Schatz wird erscheinen

- Body – ein Körper wird erscheinen
- Bones – Knochen werden erscheinen
- Pouch – ein Beutel wird erscheinen

Klicke auf das blaue Icon rechts neben „Treasure Model“, was die Datei bodybag.2da öffnet, die Datenbank mit allen Modellen für verstorbene Charaktere.

- **Store.** Verantwortlich für das „Store-Template“, auf das der Spieler zugreifen kann, um während eines Gesprächs mit dem Charakter die richtige Option zu wählen. Um ein neues „Store-Template“ zuzuweisen, klicke auf das rechte Icon neben „Store“:



Dadurch öffnet sich das Standard-Fenster zum Laden von Dateien. Wie man einen Store erstellt, wird in einem anderen Kapitel beschrieben.

- **Guard Distance.** Dieses Attribut bestimmt den Abstand, aus dem ein Wächter Geralt angreifen wird. Die folgenden zwei Attribute, Guard-Tag und max Team Size, werden in diesem vereinigt. Im Witcher können Gruppen von Charakteren geschaffen werden. Infolgedessen können Charaktere während eines Angriffs oder Verteidigung zusammenarbeiten. Wenn ein Charakter, der einer Gruppe angehört, von einem Gegner angegriffen wird, wird es möglich sein, Charaktere derselben Gruppe zur Hilfe zu rufen. Die Charaktere, die im „Guard-Distance“ Attribut gesetzt werden, werden rennend ankommen. Setze für Siegfried den Wert 0 und drücke ENTER:



- **Guard Tag.** Der Gruppen-Bezeichner. Alle Charaktere, die denselben Guard-Tag haben, gehören derselben Gruppe an und werden im Stande sein, während des Kampfs zusammenzuarbeiten. Für Siegfried müssen wir hier nichts eingeben. Um einen Tag einzugeben, klicke einfach auf die Spalte rechts neben Guard-Tag und gib die entsprechenden Werte ein.

- **Max Team Size.** Bestimmt, wie viele Charaktere einer Gruppe angehören und gerufen werden können, um einen anderen Charakter derselben Gruppe zu verteidigen. Im Siegfried's Fall ist dieses Attribut irrelevant, also trage den Wert 0 ein. Um die Einstellung zu ändern, klicke auf die Spalte rechts neben „Max Team Size“, trage den Wert ein und drücke dann ENTER:



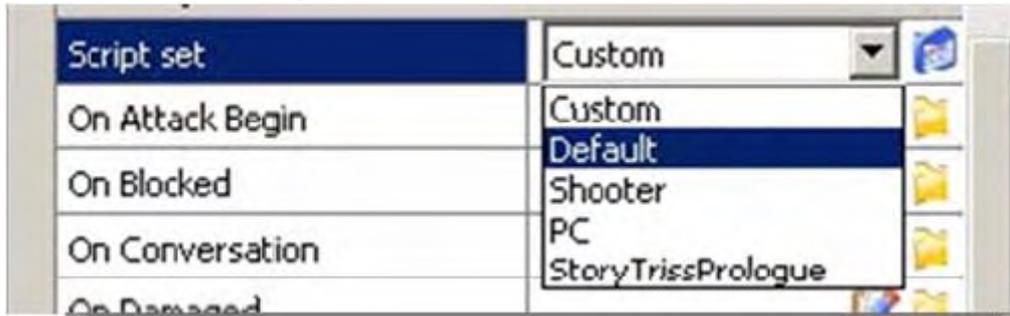
Den Wert kann man auch mithilfe der kleinen Pfeile, rechts neben der Spalte für die Werte, geändert werden:



AI Script Section:

Alle Attribute dieser Abteilung sind für die Scripts verantwortlich, die ausgeführt werden, wenn ein Charakter eine Handlung ausführt, einen Angriff beginnt oder mit einem anderen Charakter spricht. Mehr über Scripting ist in einem anderen Kapitel dieses Handbuchs besprochen worden. Hier werden nur Script-Optionen besprochen.

- **Script Set.** Bestimmt das Set mit vordefinierten Scripts für einen Charakter. Dieses Set enthält Standard-Scripts für bestimmte Ereignisse im Spiel. Um die Einstellung zu ändern, klicke auf die Spalte rechts neben „Script Set“. Nutze „Default“ für Siegfried:



Klicke auf das kleine blaue Icon rechts neben „Script Set“ um die Datei creaturescripts.2da, der Datenbank mit den Definitionen für die Standard-Scripts. Man kann folgende Werte zuweisen:

- Custom – der Charakter wird verschiedene Scripts verwenden.
- Default – der Charakter wird die Standard-Scripts verwenden.
- Shooter – der Charakter wird die Scripts für das Benutzen von Bögen verwenden.
- PC – der Charakter wird die Scripts von Geralt verwenden.
- StoryTrissPrologue – der Charakter wird die Scripts von Triss aus dem Prolog verwenden.

Die unten aufgeführten Ereignisse werden erstellte Scripts verwenden, wenn das Script-Set auf Custom gestellt ist:

- **On Attack Begin.** Script wird am Anfang eines Kampfes ausgeführt.
- **On Blocked.** Script wird ausgeführt, wenn sich der Charakter nicht mehr bewegen kann.
- **On Conversation.** Script wird ausgeführt, wenn ein Gespräch beginnt.
- **On Damaged.** Script wird ausgeführt, wenn der Charakter Schaden erleidet.
- **On Death.** Script wird ausgeführt, wenn der Charakter stirbt.
- **On Disturbed.** Script wird ausgeführt, wenn der Charakter bei einer Aktion unterbrochen wird.
- **On End Conversation.** Script wird ausgeführt, wenn der Charakter ein Gespräch beendet.
- **On End Combat Round.** Script wird ausgeführt, wenn der Charakter einen Angriff beendet.
- **On Heartbeat.** Script wird alle 5 Sekunden ausgeführt.
- **On Internal Event.** Script wird zusammen mit einem internen Event ausgeführt.
- **On Out Of Ammo.** Script wird ausgeführt, wenn der Charakter keine Munition mehr hat.

- **On Perception.** Script wird ausgeführt, wenn der Charakter einen anderen Charakter hört oder Sieht.
- **On Physical Attacked.** Script wird ausgeführt, wenn der Charakter physisch angegriffen wird.
- **On Rested.** Script wird ausgeführt, wenn der Charakter meditiert.
- **On Spawn It.** Script wird ausgeführt, wenn der Charakter an einem Spawn Punkt erscheint.
- **On Spell Cast At.** Script wird ausgeführt, wenn der Charakter von einem Zauber beeinflusst wird.
- **On Witness.** Script wird ausgeführt, wenn der Charakter von einem anderen Charakter entdeckt wird.

Um das ausgewählte Script zuzuweisen, klicke auf das kleine gelbe Icon rechts neben dem Event, dem du das Script zuweisen willst. Zum Beispiel, wenn du ein Script dem Event „On Attack Begin“ zuweisen willst, klicke hier:



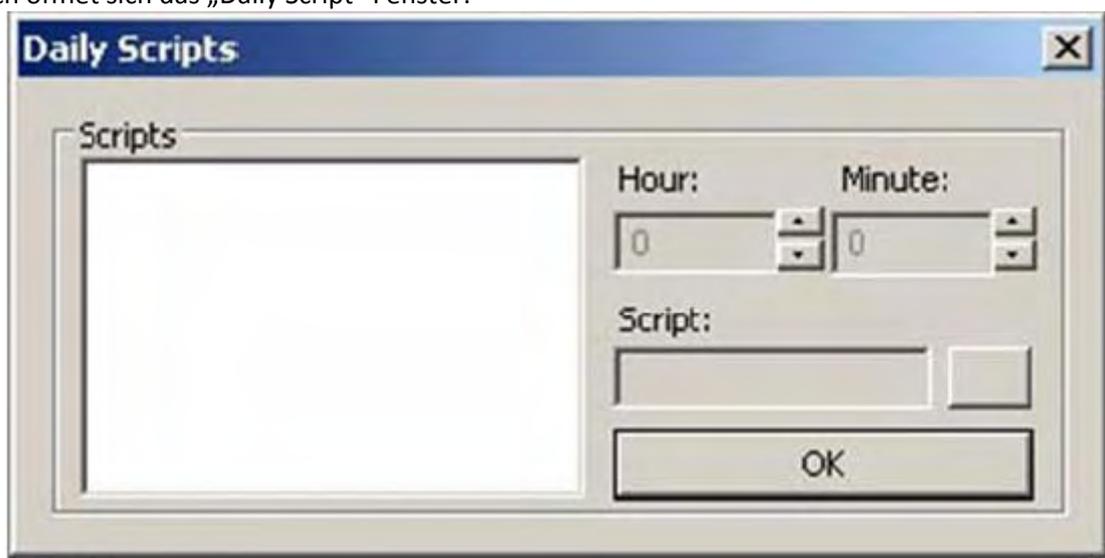
Dies öffnet das Fenster um Dateien zu laden. Gehe in das Verzeichnis Data\Scripts, das alle Scripts enthält, wähle die Datei, die du suchst, aus und klicke auf „Open“. Scripts werden in einem anderen Kapitel beschrieben.

User Scripts Section:

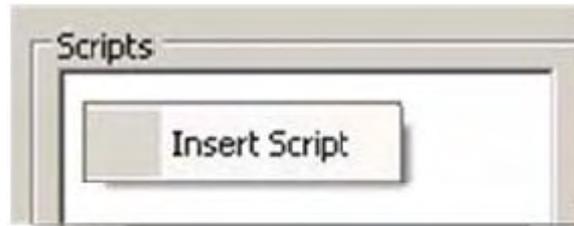
- **Daily Scripts.** Das Script wird jeden Tag zu einer bestimmten Zeit ausgeführt. Um das passende Script hinzuzufügen und um die Zeit, an der es ausgelöst werden soll, zu definieren, musst du einfach auf das kleine, rote Icon rechts neben „Daily Script“ klicken:



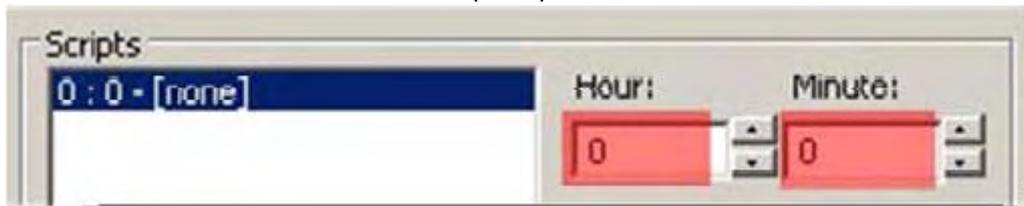
Dadurch öffnet sich das „Daily Script“ Fenster:



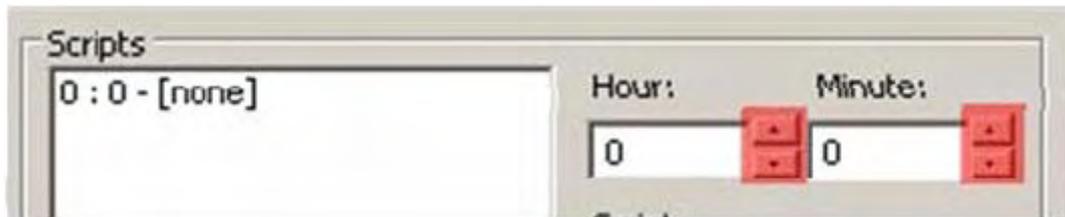
Um das passende Script hinzuzufügen, klicke mit der rechten Maustaste auf das „Script Panel“ und wähle den Menüpunkt „Insert Script“:



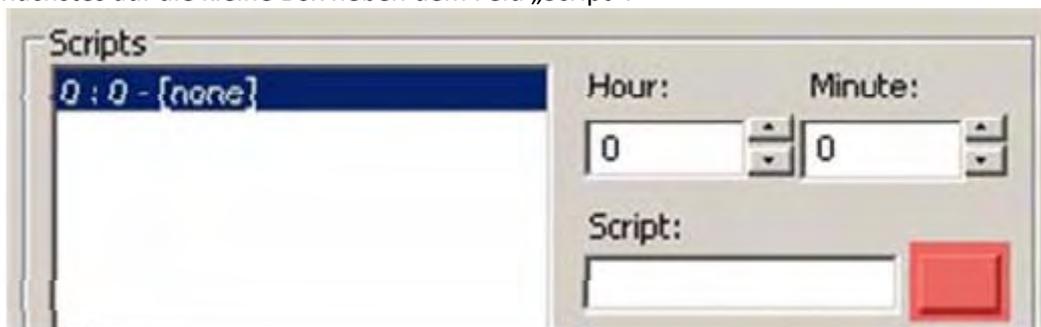
Klicke nun auf das hinzugefügte Script und setze die Uhrzeit, zu der das Script ausgeführt werden soll. Trage die gewünschten Zeiten in die Fenster für Stunde (Hour) und Minuten ein:



Du kannst die Zeit auch mithilfe der kleinen Pfeile rechts neben den Feldern einstellen:

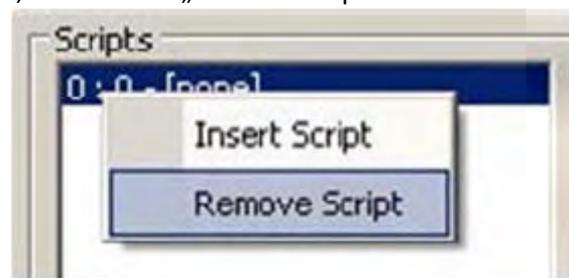


Klicke als nächstes auf die kleine Box neben dem Feld „Script“:



Das öffnet das Standard-Fenster für das Laden der Dateien. Wähle das Script aus, das du laden möchtest, und klicke auf „Open“. Auf diese Weise wird das Script täglich in der voreingestellten Zeit ausgeführt. Klicke dann auf „OK“ im Script-Fenster. Natürlich kannst du mehrere Scripte hinzufügen.

Wenn du ein Script entfernen willst, markiere das zu entfernende Script im „Script-Fenster“ und klicke darauf mit der rechten Maustaste, Wähle dann „Remove Script“ im Menü:



• **On User Defined.** Script wird vom User definiert. Um dies Art Script zu definieren, klicke auf das kleine gelbe Icon rechts neben „On User Defined“:



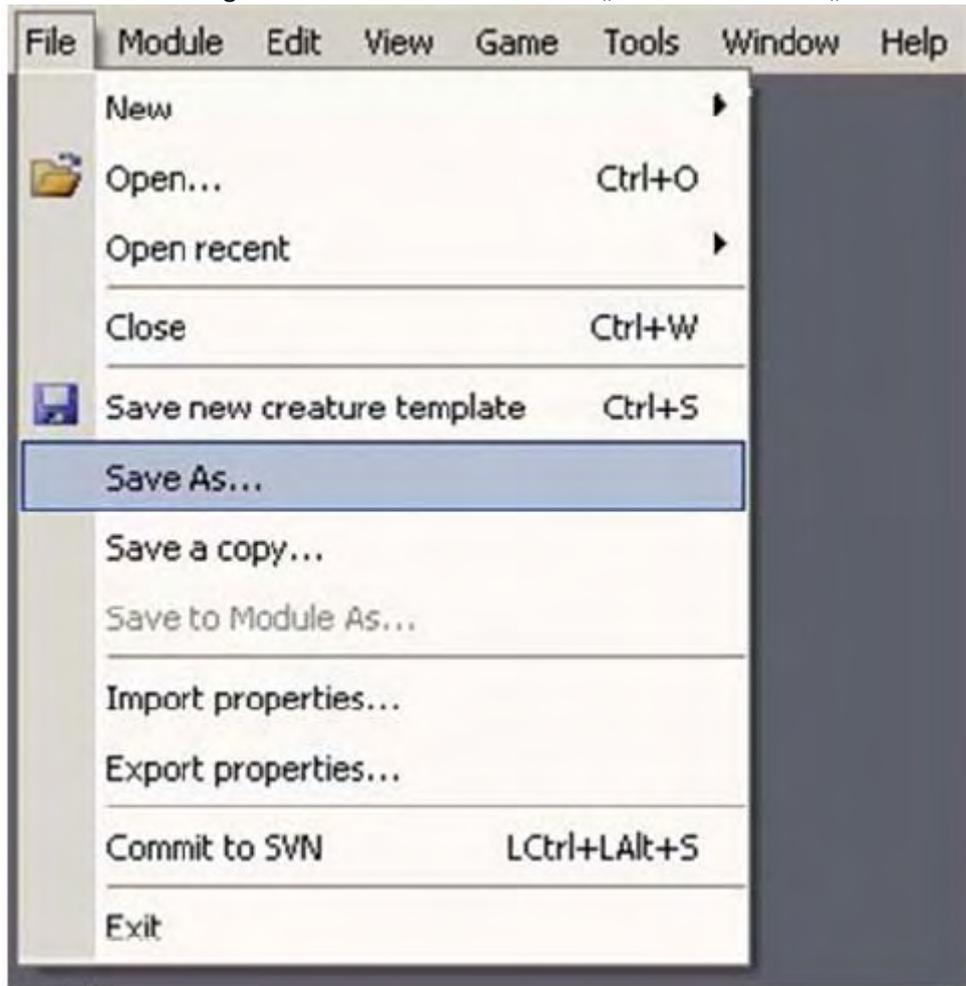
Das öffnet das Standard-Fenster für das Laden der Dateien. Wähle das Script aus, das du laden möchtest, und klicke auf „Open“.

LUA Scripts Section:

Diese Abteilung behandelt auch Scripts. Diese Scripts sind umfangreicher und in Lua geschrieben. Sie werden hier nicht behandelt.

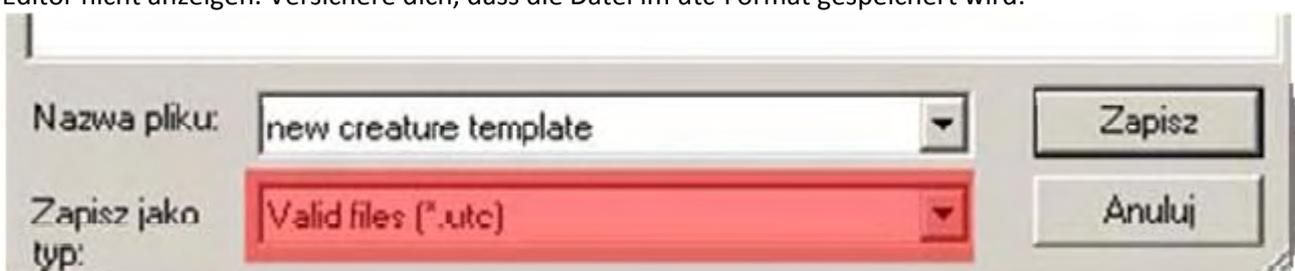
- **On Creation.** Script wird ausgeführt, wenn der Charakter erstellt wird.
- **On Model Loaded.** Script wird ausgeführt, wenn das Charakter-Modell geladen wird.

Das wars! Speichere nun die Siegfried's Schablone. Wähle dazu „Save as“ aus dem „File“ Menü:

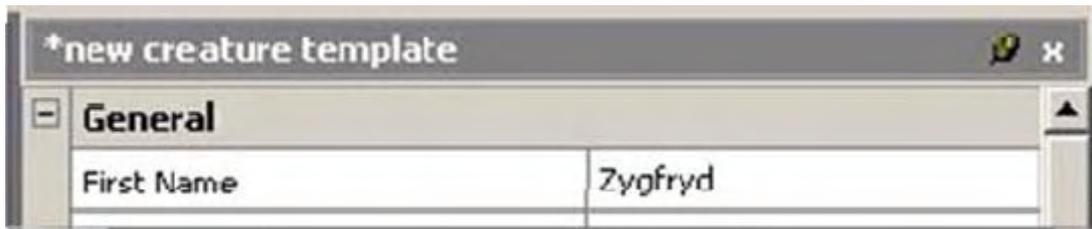


Dadurch wird das Standard-Fenster zum Speichern von Dateien geöffnet. Da Siegfried ein NPC ist, speichere im Verzeichnis Templates\Characters\NPC. Geh in das Verzeichnis, gib „our_siegfried“ ein und klicke auf „Save“.

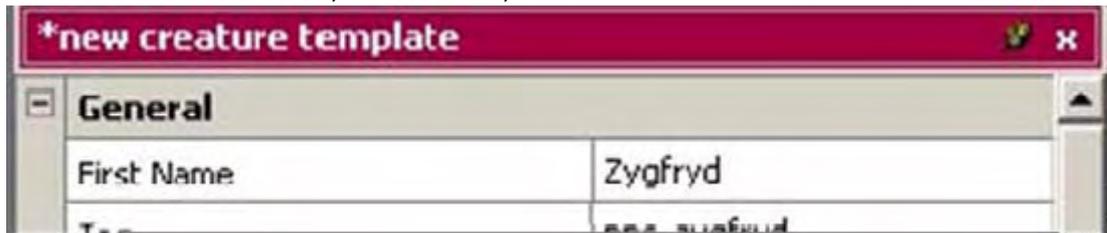
Denk dran, dass der Name der Datei nicht die Länge von 16 Zeichen überschreiten darf. Sonst wird ihn der Editor nicht anzeigen. Versichere dich, dass die Datei im utc-Format gespeichert wird:



Wenn stattdessen ein anderer Datei-Typ aufgelistet wird, bedeutet es wahrscheinlich, dass du versuchst, den Inhalt in einem anderen offenen Fenster zu speichern. Vor dem Speichern einer Charakter-Schablone, stelle sicher, dass das „New Creature Template“ Fenster aktiv ist. Wenn es inaktiv ist, wird die Titelbar grau sein:

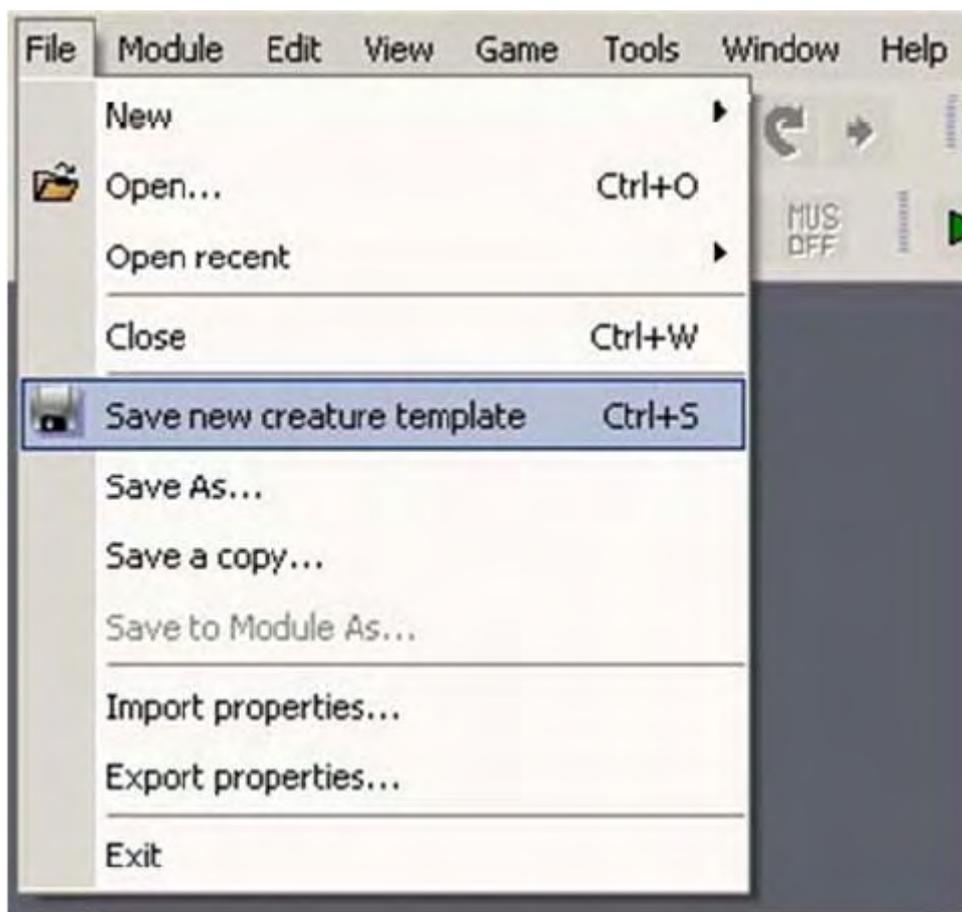


Hat die Titel bar eine andere Farbe, bedeutet das, dass das Fenster aktiv ist:



Ein Sternchen vor dem Namen in der Titel bar bedeutet, dass die Datei nicht mehr gespeichert wurde, nachdem du was verändert hast.

Gut gemacht. Die erste Kreaturen-Schablone wurde erschaffen. Bevor wir weitermachen, lass uns die Datei auf eine andere Art speichern. Wähle anstatt „Save as“ den Eintrag „Save new Creature Template“ im Menü „File“:

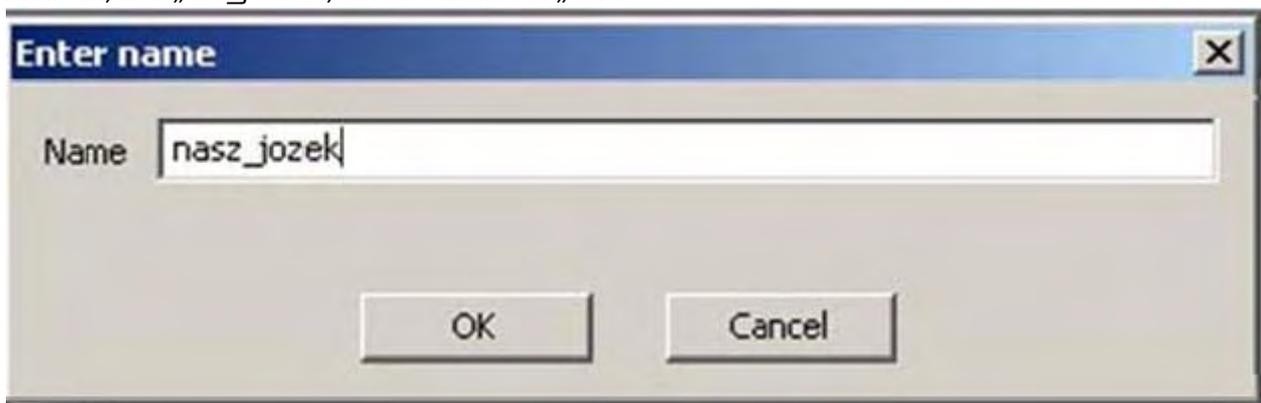


Ein Fenster mit dem Titel „Save File“ wird erscheinen:

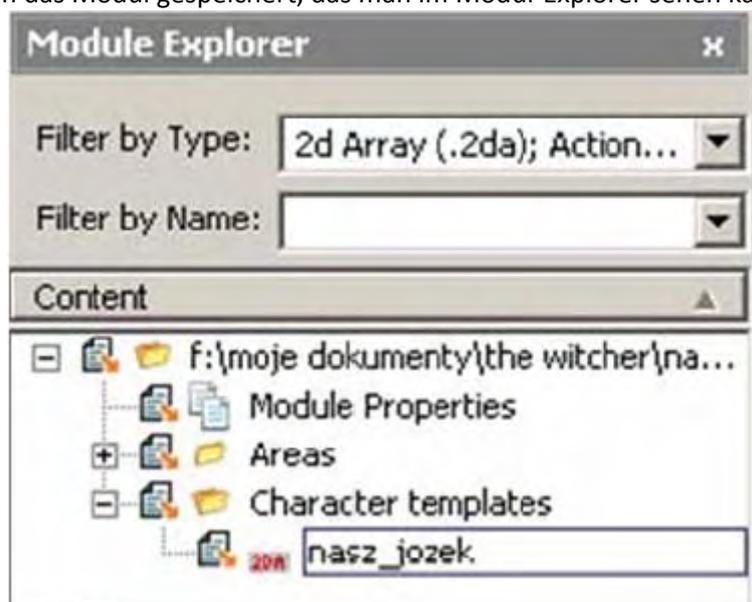


Global wird dasselbe Ergebnis wie das Auswählen des „Save as“ Befehls im File-Menü bringen. Alle Dateien auf diesem Weg gespeicherten Dateien sind globale Dateien, was bedeutet, dass sie in anderen Modulen verwendet werden können. Das Speichern auf diesem Weg hat seine Nachteile, denn die Dateien sind keinem bestimmten Modul zugewiesen. Um Module einfach zu benutzen und ins Internet hochzuladen, sollten alle Dateien mithilfe von „Module“ gespeichert werden.

Was passiert zum Beispiel, wenn 2 Personen den gleichen Namen verwenden? Das ist der Moment, wo uns das Speichern direkt in einem Modul die Rettung bringt. Um eine Datei so zu speichern, klicke auf „Module“ in dem „Save File“ Fenster. Darauf wird sich das Fenster mit dem Titel „Enter Name“ öffnen. Trage den Dateinamen, hier „our_jethro“, ein und klicke auf „OK“:



Die Datei wird direkt in das Modul gespeichert, das man im Modul-Explorer sehen kann:



Die in diesem Handbuch verwendete „Speichern“ Methode verwendet den „Save as“ Befehl vom File-Menü. Später in diesem Handbuch wird eine andere Methode, Dateien zum Modul hinzuzufügen, vorgestellt.

Vier zusätzliche Schablonen müssen geschaffen werden. Du bist bereits dazu fähig, diese Schablonen zu schaffen, aber um dir diese Aufgabe zu erleichtern, sind die Optionen, die verwendet wurden, um besondere Charakter-Attribute auszuwählen, unten aufgelistet. Wähle „New-> template-> Creature (.utc) im File-Menü aus. Das vertraute Fenster des „New Creature Template“ wird sich öffnen:

Jethro, der Poker-Spieler:

- **First name** – Jethro,
- **Tag** – our_jethro,
- **Appearance** – cr_guard1_c1g1,
- **Race** – Human,
- **Gender** – GENDER_MALE,
- **Movement Rate** – Normal,
- **Perception Range** – Default,
- **Profiles** – Attitude_Friendly, Behavior_Defensive,
- **Abilities** – cr_level10,
- **Inventory Contents** – Big weapon -> WitcherSteelSword -> Temerian steel sword (it_stlswd_005.uti),
- **Combat Styles** – WitcherSteelFast / WitcherSteelSword -> h_bas_sword1h01,
- **Script set** – Default

Lass die anderen Attribute in ihrer Standard-Einstellung. Dieses „Creature Template“ wird in das Verzeichnis „Templates\Characters\Commoners gespeichert unter dem Namen our_jethro. Versichere dich, dass die Datei im UTC-Format gespeichert wird.

The Professor, der Chef der Banditen:

- **First name** – The Professor,
- **Tag** – our_professor,
- **Appearance** – cr_prof2_c1g1,
- **Race** – Human,
- **Gender** – GENDER_MALE,
- **Experience Value** – 100,
- **Movement Rate** – Fast,
- **Perception Range** – Default,
- **Profiles** – Attitude_Hostile, Behavior_Aggressive, Affiliation_Boss, Role_DoNotHelpFriends, Role_IgnoreAll, Role_PCAttacker,
- **Abilities** – AutoJump, finisher_1h_sword, HeavyWounded, Wounded,
- **Inventory Contents** – Right hand -> Weapons & Armor -> Witcher's silver sword (it_svswd_004.uti),
- **Combat Styles** – WitcherSteelFast / WitcherSteelSword -> h_adv_sword1h01,
- **Script set** – Default,

Dieses „Creature Template“ wird in das Verzeichnis „Templates\Characters\NPC gespeichert unter dem Namen our_professor.

Bandit Leaders Guard:

- **First name** – Bandit,
- **Tag** – our_bandit,
- **Appearance** – cr_band2_c1g1,
- **Level** – 1,
- **Race** – Human,
- **Gender** – GENDER_MALE,
- **Experience Value** – 50,
- **Movement Rate** – Fast,
- **Perception Range** – Default,
- **Profiles** – Attitude_Hostile, Behavior_Aggressive, Enemy_Witcher, Role_PCAttacker,

- **Abilities** – AutoJump, finisher_1h_sword, HeavyWounded, Wounded,
- **Inventory Contents** – Right hand -> Weapons&Armor -> Witcher's silver sword (it_svswd_004.uti),
- **Combat Styles** – WitcherSteelFast / WitcherSteelSword -> h_bas_sword1h01,
- **Script set** – Default,

Speichere die Schablone im Verzeichnis Templates\Characters\Enemies unter dem Namen our_bandit.

Wolf:

- **First name** – Wolf,
- **Tag** – our_wolf,
- **Appearance** – cr_dog2_c1g1,
- **Level** – 1,
- **Race** – Animal,
- **Gender** – GENDER_MALE,
- **Experience Value** – 25,
- **Movement Rate** – Normal,
- **Perception Range** – Short,
- **Profiles** – Attitude_Hostile, Affiliation_Monster_Animal, Behavior_Aggressive, Tactics_DontSurround, Enemy_Witcher,
- **Abilities** – AutoJump, cr_m_f_dog01, HeavyWounded, LightyWounded, Wounded,
- **Inventory Contents** – Right hand -> Weapons&Armor -> Natural weapon (it_natural_001.uti),
- **Combat Styles** – Monster / Monster -> m_f_dog_att01,
- **Run To Despawn** – True,
- **Script set** – Default,

Speichere die Schablone im Verzeichnis Templates\Characters\Monsters unter dem Namen our_wolf.

Nun sind wir am Ende des Kapitels. Nach dem Lesen dieses Kapitels und Erstellen dieser fünf Schablonen solltest du ein allgemeines Gefühl für die Schablone-Entwicklung haben. Du wirst mehr lernen, wenn du mit ihnen auch experimentierst. Lass uns in der Zwischenzeit einen Blick darauf werfen, wie man das Verhalten von Charakteren definiert.

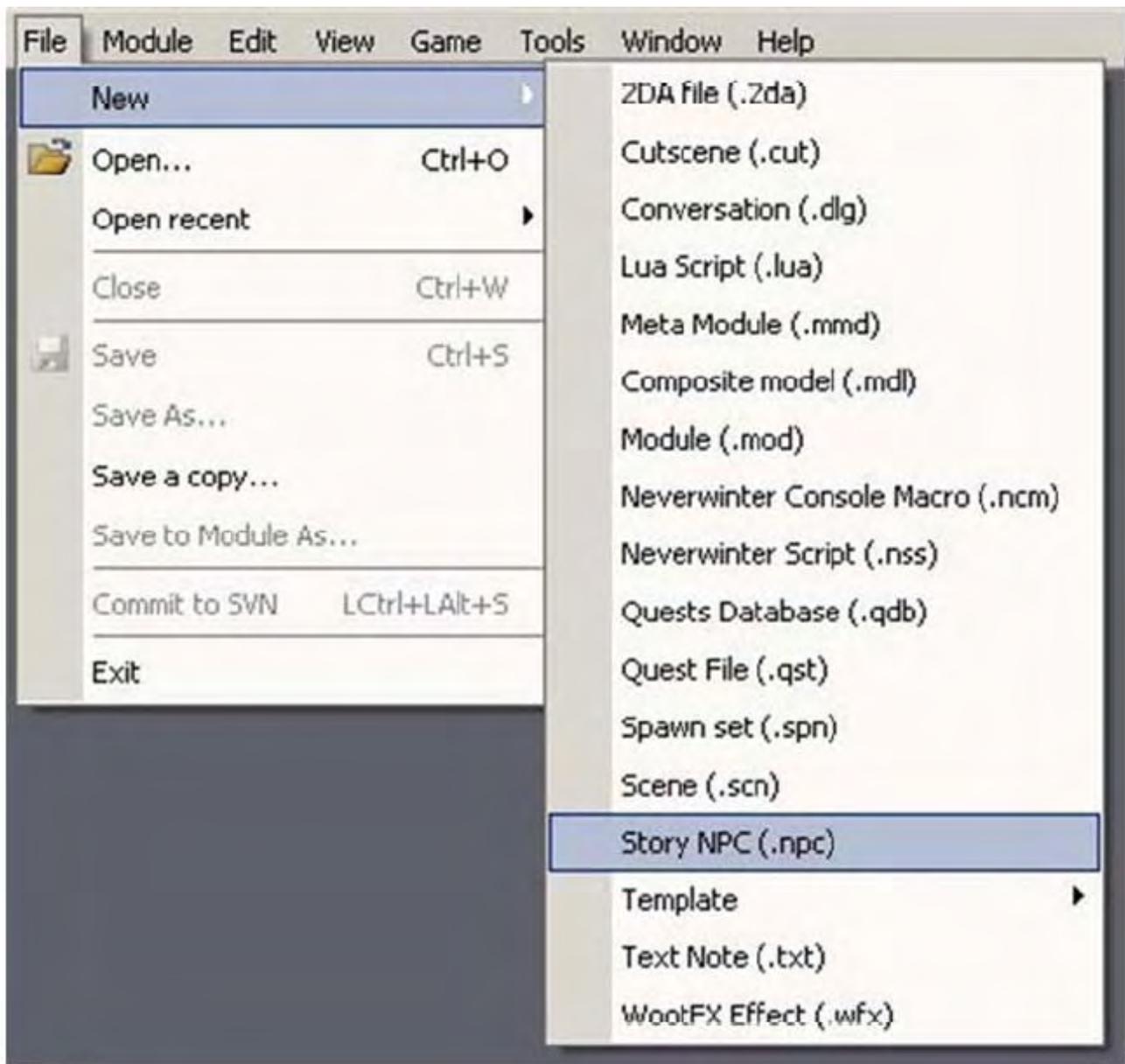
SPAWN SETS

Das Definieren des Charakters Verhaltens und das Einführen der Zeit und eines Platzes, in denen ein Charakter spawnen soll ist das, worüber wir jetzt reden werden. Das von den Schöpfern des Witcher's entwickelte Charakter-spawn-set ist wirklich ziemlich ordentlich und vor allem phantasievoll organisiert. Es ist möglich, zu definieren, wann und wo jeder Charakter spawnen soll im Spiel wie auch die Aktionen, Die die Charakter ausführen sollen.

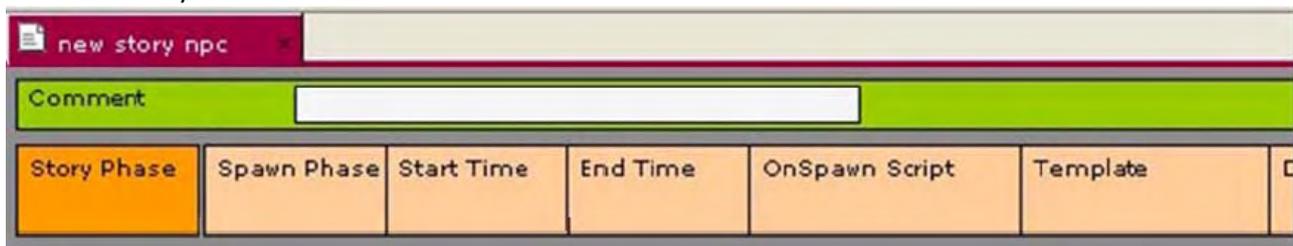
Mit dieser Lösung kann ein Charakter geschaffen werden, der zum Beispiel im Spiel um 7:00 Vormittags erscheinen wird beim Schmied und bleibt dort bis 3:00 Mittags und schmiedet Schwerter und Rüstungen. Der gleiche Charakter wird um 3:00 P.M. zur Taverne, um ein Getränk zu trinken, oder Poker zuzuspielen. Um 8:00 Uhr, wird der Charakter zu Bett gehen. Am nächsten Tag wird der Schmied um 7:00 wieder aufstehen. Die Möglichkeiten sind praktisch endlos.

Also am Anfang dieses Kapitels können Charaktere im Witcher in 3 Kategorien eingeteilt werden - NPC Charaktere, unabhängige Charaktere (z.B. Bürgerliche, wo auch die Stadtbewohner zählen können, Kaufleute, Banditen) und Monster. Lass uns die NPCs ansehen, und beginnen.

Ein NPC hat einen Namen, einzelne Eigenschaften, Plätze, wo er/die Zeit auf verschiedene weisen verbringt, im Spiel, Dialoge, die bei besonderen Situationen usw. ausgelöst werden. Mit anderen Worten ist da ein einzigartiger Charakter. Sobald er getötet wird, erscheint ein NPC nicht im Spiel. NPCs haben einen leicht anderen Spawn-Set als andere Charaktere (sowohl Bürgerliche als auch Monster). beginnen, wir und schauen, wie ein gesetzter NPC Charakter-spawn geschaffen ist. Unser NPC ist jetzt Siegfried und dieses Set wird jetzt für ihn geschaffen. Um einen Spawn-Set für ein NPC zu bauen, wähle im menü New -> Story NPC (.npc) aus:



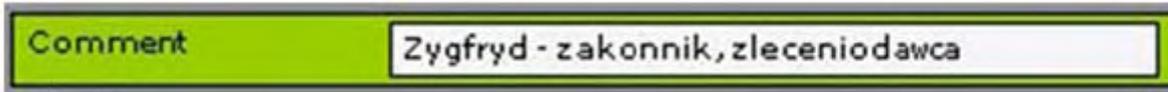
Ein neues Story NPC Fenster öffnet sich:



Dieses Fenster ist ziemlich lang und kann mit Hilfe der unten auf dem Bildschirm ausfindig gemachten Bildlaufleiste navigiert werden. Das Fenster ist in mehrere Spalten eingeteilt. Jetzt reden wir über die Bedeutung von jeder Spalte und wie man neue Einträge hinzufügen kann.

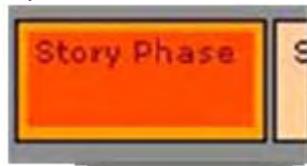
- **Comment.** Enthält ein Feld in, dem du eine Anmerkung schreiben kannst. Dies wird im Spiel nicht verwendet, das ist nur ein Informationsfeld. Du kannst schreiben was immer du möchtest. Um einen Eintrag hinzuzufügen, Doppelklicke auf das weiße Feld gegenüber vom Wort Kommentar:

Für Siegfried gibst du "Siegfried - Ritter, Arbeitgeber" ein und drückst Enter:



• **Story Phase.** Die Art, in der jedes NPC handelt, hängt von der Phase der Geschichte ab. Vom Script oder eines Quest ,es ist möglich, die Phase von der Geschichte des Spiels zu ändern. Dies wiederum verursacht eine Änderung im Verhalten des Charakters (solange die Charaktere sich in mehrere Phasen definieren lassen). In jeder Geschichte führst du schrittweise die Aktion ein, die Charakter ausführen soll, alles andere muss definiert werden. die Spawn-Points müssen aber gesetzt werden. Spawnpunkte und Aktionspunkte sind in einem anderen Kapitel dieses Handbuchs beschrieben.

Jeder NPC kann eine unbegrenzte Anzahl von Storylines haben. In jeder Phase, der Aktionen welches ein Charakter ausführen soll das von der Start-Linie die Spawnpunktkennzeichnungen setzen usw. Auf diese Weise kann ein NPC spezifisch für jeden Teil des Spiels definiert werden. Um eine neue Storyline hinzuzufügen, klickst du auf die rechte Seite des *Story Phase* Column:



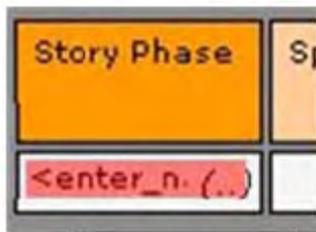
Klicke mit der rechten Maustaste auf Story Phase, in dem Menü was sich geöffnet hat, klickst du auf „Add Story Phase“:



Das fügt eine neue Story Phase hinzu:

Story Phase	Spawn Phase	Start Time	End Time	OnSpawn Script	Template	D
<enter_n. (...)		00:00:00	06:00:00			

Zuerst gibst du der neuen Story einen Namen. Der Name der Phase sollte einfach sein und einprägsam, es wird notwendig sein, wenn die Story für einen Charakter gewünscht wird:



Da Siegfried der Auftraggeber von Geralt sein wird, tippe in das Feld "employer" und drücke Enter:



• **Spawn Phase.** Die nächste Maßnahme wird sein, eine Spawnphase hinzuzufügen. Dieser Phasentyp ist im Allgemeinen für die verschiedenen Perioden am Tag, in dem der Charakter seine Aktionen ausführt, verantwortlich ist. Jede Story kann mehrere Spawnphasen haben. Um eine neue Spawnphase hinzuzufügen klicke auf das Feld wo „Spawn Phase“ steht:

Story Phase	Spawn Phase	St
zlecenie		0

Nun öffnet sich ein Menü, in diesem Menü wählst du „Add Spawn Phase“:

Spawn Phase	Start Time	En
		06

Add Spawn Phase

Delete Spawn Phase

Copy

Paste

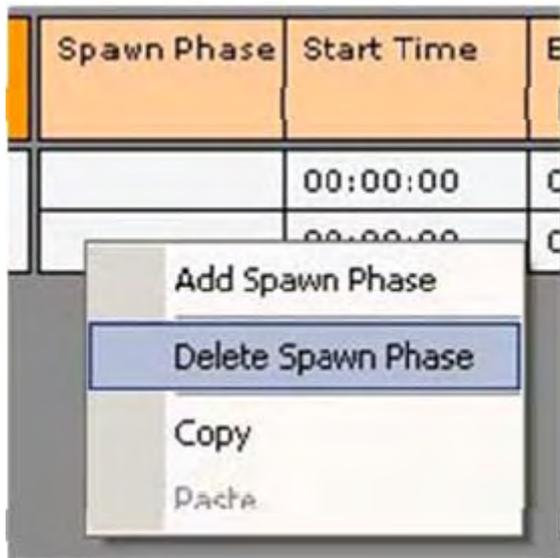
Das fügt eine neue Spawn-Phase hinzu, Achte darauf dass eine neue Zeile erschienen ist, die neue Zeile erscheint in der „Quest Story“ Linie:

Story Phase	Spawn Phase	Start Time	End Time	OnSpawn Script	Template	Di
zlecenie		00:00:00	06:00:00			
		00:00:00	06:00:00			

Die „Quest Story“ Phase hat z.Z. zwei Spawnphasen. Da Siegfried nur eine solche Phase in unserem Abenteuer haben wird, kann diese Sekunde entfernt werden. Um die Einstellung zu ändern, Rechtsklicke auf die Spawnphase um sie zu entfernen. Da wir gerade die zweite Spawnphase hinzugefügt haben, löschen wir dann diese:

Story Phase	Spawn Phase	St
zlecenie		00
		00

Wähle im Menü „Delete Spawn Phase“:



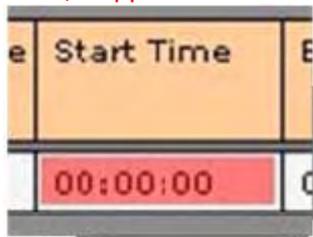
Die unnötige Spawnphase ist entfernt worden. Jede geschaffene Story enthält eine Spawnphase. Jetzt sollten wir es umbenennen, um das zu tun Doppelklicke auf das Feld unter der Spawnphase-Leiste:



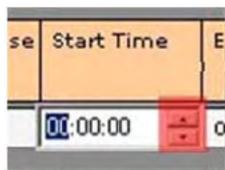
Gib „Waiting“ ein und drücke ENTER. Jetzt führt er eine Quest-Story Phase ein, indem es jetzt eine Warte-Spawn-Phase gibt die erstellt wurde:



• **Start time.** Die nächste Stufe wird sein, die Zeit zu setzen, an denen die NPCs Ihre Aktionen ausführen werden. Das wird getan durch das Eintragen der entsprechenden Zeit im „Start-Time“ Feld, welches die Zeit definiert eine Spawnphase beginnt, und die Endzeit welches die Zeit definiert, wenn eine angegebene Spawnphase endet. Im diesem Zeitrahmen führt der NPC Aktionen aus, die wir später definieren werden. Die Anfangszeit setzen wir in der Spawnphase, Doppelklicke auf das Feld unter der „Start Time“ Spalte:

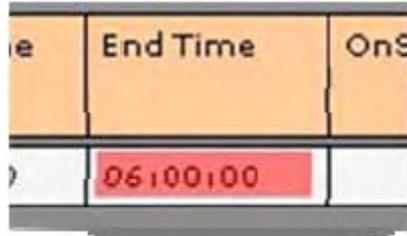


Die Zeit kann mit der Tastatur eingestellt werden oder, man klickt auf die kleinen Pfeile, die auf der rechten Seite der Spalte zu finden sind:



Lass die Zeit auf 00:00:00.

• **End Time.** Diese Spalte stellt die beendende Zeit für die Spawnphase ein. Die Default Zeit ist 06:00:00. Wenn das mal gelassen wurde, würde Siegfried bei 00:00:00 und bei 06:00:00 spawnen, aber das ist nicht, was wir hier wünschen. In diesem Abenteuer soll Siegfried im Spiel vorhanden sein die ganze Zeit, ob es Nacht oder Tag ist. Um das zu erzielen, müssen beide die Spawnphase Anfangs- und Endzeiten bis 00:00:00 eingestellt werden. Doppelklicke auf das Feld unter End Time:

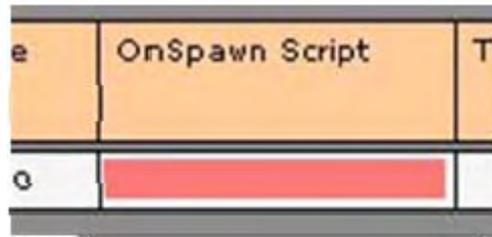


Tippe 00:00:00 ein und drücke Enter:

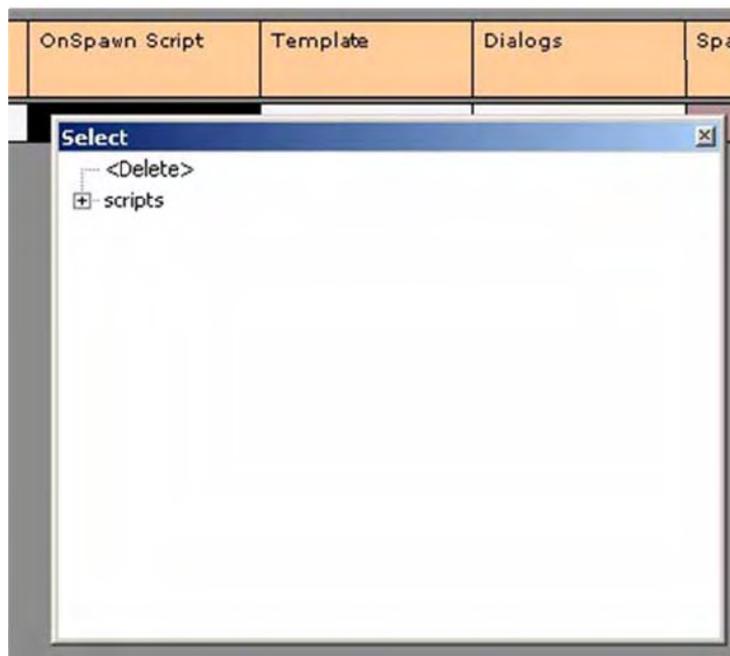


Erinnern dich daran, dass verschiedene Spawnphasen sich nicht überlappen können, weil kein Charakter in zwei unterschiedlichen Spawnphasen gleichzeitig sein können.

• **OnSpawn Script.** Diese Spalte stellt einen Script ein, der jede Zeit von einem NPC Spawn durchgeführt wird. Wir brauchen das nicht, für Siegfried zuweisen. Wenn du gerne einen Script einem NPC jedoch zuteilen möchtest, klickst du auf dem Feld unter der OnSpawn Script-spalte doppelt:



Ein Auswahl Fenster öffnet sich:



Das Fenster zeigt eine Liste aller im Spiel verfügbaren Scripte. Diese Liste wird in einer Baum Form gezeigt und reflektiert die Struktur der Daten\Scriptordners (in welchem alle Scripte aufbewahrt sind. Den Baum öffnest indem du auf das kleine '+' Zeichen klickst:



Um diesen Zweig zu schließen, klickst du auf die Kleine '-' Zeichen:



Nachdem das passende Script ausgewählt worden ist, Doppelklicke auf dieses Script und, wähle „2city“ für Siegfried aus:



Nachdem wir auf „2city“ doppelt geklickt haben, erscheint es im Feld unter der OnSpawn Scriptspalte:

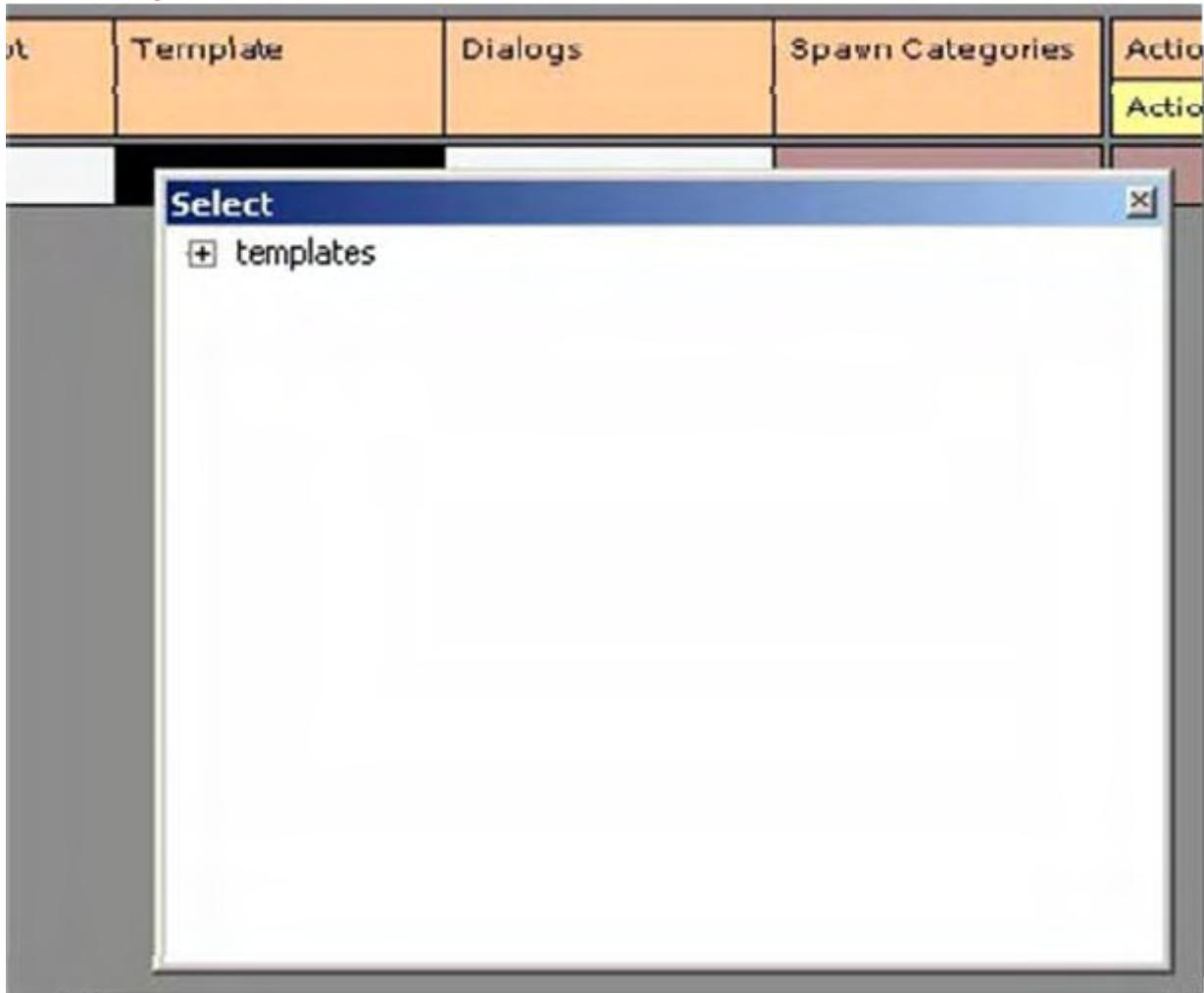
	OnSpawn Script	Te
0	2city	

Um das Script wieder zu entfernen, klickst du auf das Feld unter der OnSpawn Script-Spalte doppelt. Dieses Mal klickst Du jedoch auf die <Delete> Auswahl, zu finden auf der Oberseite der Liste im Ausgesuchten Fenster:

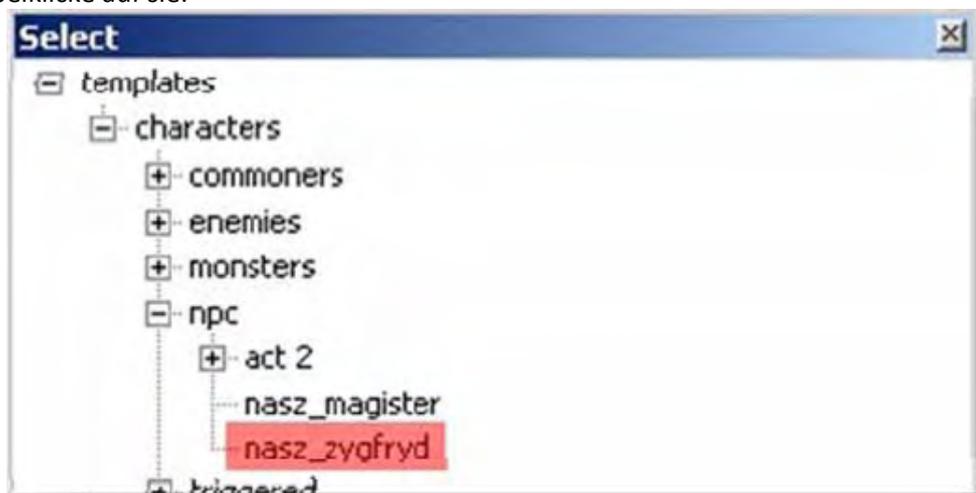


Das wird das „2city“ Script Entfernen. Du solltest beachten, dass ein verschiedenes Script für jede Spawn-Phase zugeteilt werden kann.

- **Template.** Ein anderer sehr wichtiger Schritt zum hinzufügen einer Charakter Schablone ist, Das die Charakter Schablone, die wir früher erstellt, und hinzugefügt hatten. Die Richtlinie für das Zuweisen einer Schablone ist dasselbe wie für das Zuweisen der Scripte. Doppelklicke auf Das Feld unter der Schablonen Spalte. Das ausgewählte Fenster öffnet sich:



Dieses Mal, erscheint eine Liste aller Charakter Schablonen, die im Spiel vorhanden sind. Die Schablonen werden in der Baum Form angezeigt, und die Struktur des Data\Templates Ordners reflektieren. Die Siegfried Schablone ist im Data\Templates\Characters\NPC Ordner gespeichert. Wähle „our_siegfried“ Schablone und Doppelklicke auf sie:



Das fügt „our_siegfried“ 's Charakter Schablone zur Quest Story Phase hinzu:

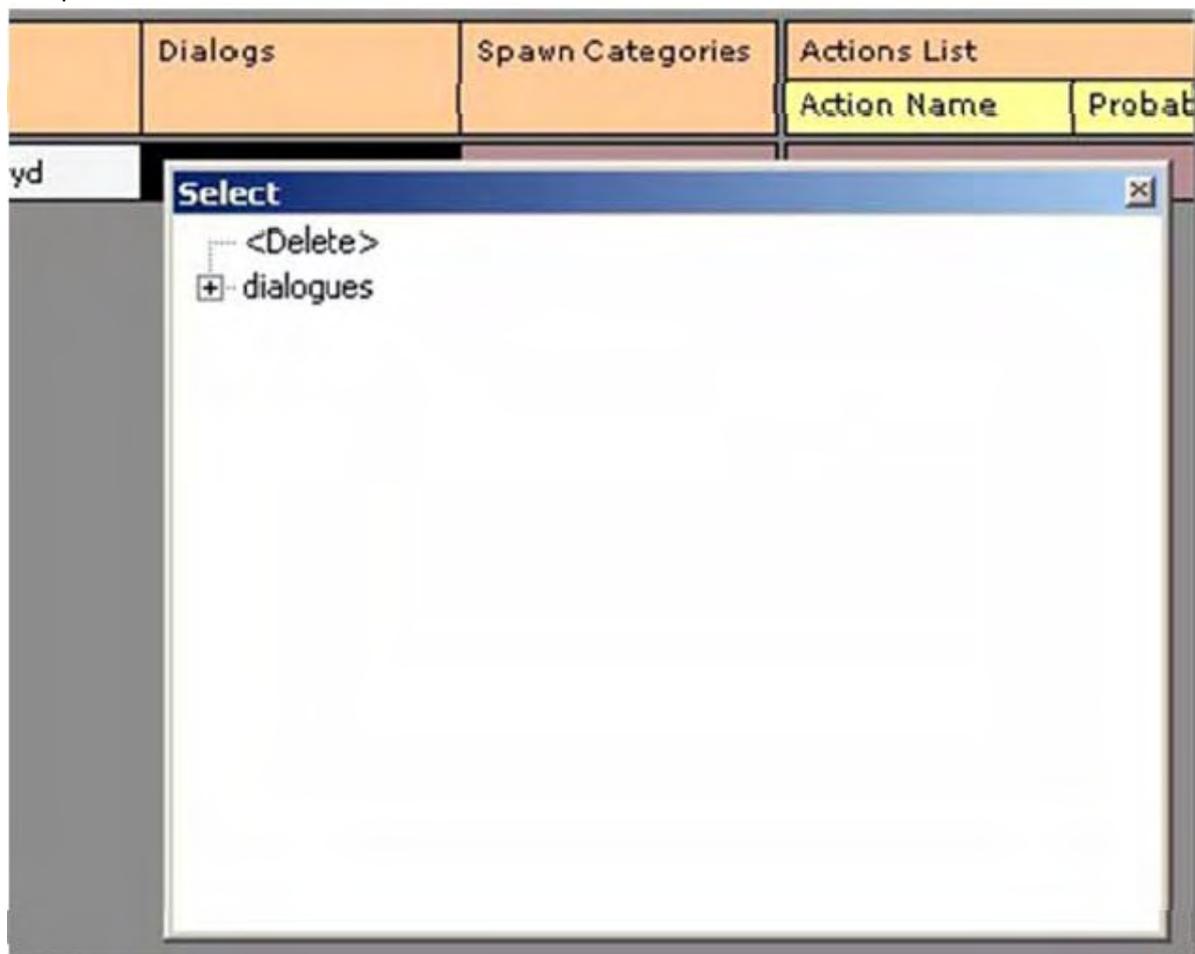
st	Template	Di
	nasz_zygfryd	

Sobald eine Charakter Schablone zugewiesen ist, kann sie vom Charakter Spawn-Set nicht mehr entfernt werden. Sie kann zu einer anderen Schablone nur hinzugefügt werden. Ein NPC kann eine andere Charakter Schablone in jeder Spawnphase haben, aber, damit die Änderung der Charaktere sichtbar ist, muss zuerst despawn und dann respawn sichtbar sein. Dies erfordert dem Spieler einen Wiedereintrag der Umgebung vorzunehmen.

• **Dialogs.** Gespräche verstellen, wird in einem anderen Kapitel dieses Handbuches beschrieben, das speziell zu diesem Thema eingegangen wird. Gespräche sind einfach Dialoge zwischen zwei oder mehrere Charaktere. Siegfried hat schon ein zugewiesenes Gespräch, aber dazu später mehr. schauen wir jetzt darauf, wie man eine Gesprächs-Datei in einem Spawn-set eines Charakters zuteilt. Das wird derselbe Weg wie das Zuweisen eines scripts oder der Charakter-Schablone gemacht. Klickst du auf das Feld unter der Dialogspalte doppelt:



Das entsprechende Fenster öffnet sich:



Die Fenster zeigen eine Liste von Dialogen an. Diese Liste wird in einer Baum Form angezeigt und die Struktur des Data\Dialogues Ordner reflektiert (wo alles Vorhandene im Spiel gespeichert wird Gespräche). Um einen Dialog zuzuweisen, lokalisiert du es auf der Liste und Doppelklickst. Z.B. auf den Dialog cn_boy01 und weist es Siegfried zu:



Der ausgewählte Dialog erscheint im Feld unter der Dialogspalte:

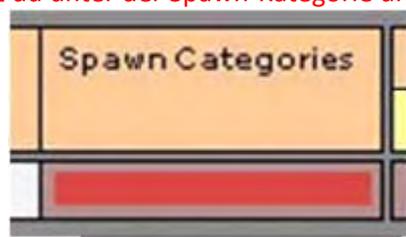


Da wir unseren eigenen Dialog später erstellen, Doppelklicke auf das Feld, daraufhin öffnet sich ein Fenster und dort wählst du „<Delete>“ aus:

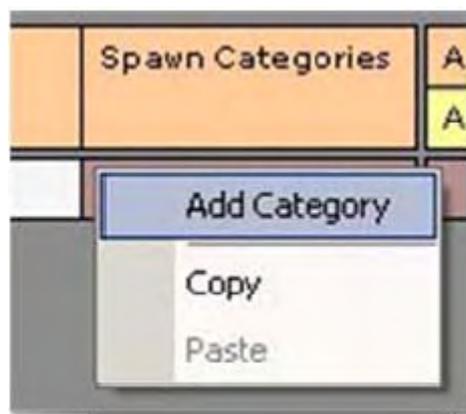


Der Dialog wird entfernt. Ein NPC kann unterschiedliche Gesprächs Ordner haben, die in jeder Spawnphase zugewiesen werden können. Das ist, warum diese Charakter nicht die Dialoge haben, die in der Charakter Schablone zugewiesen wurden.

• **Spawn Categories.** Die folgende Spalte sind Spawnkategorien., das ist ein Identifizierungszeichen das den NPC erklärt, an welchen Spawnpunkt er erscheinen oder verschwinden soll um eine Spawnkategorie, in einem Feld zuzuweisen Rechtsklickst du unter der Spawn-Kategorie und fügst es hinzu:



Wähle „Add Category“ im Menü:



Eine neue Spawnkategorie ist hinzugefügt worden(Name). Doppelklicke im Feld, das gerade unter der Spawn-Kategorien Spalte erschienen ist:



Für Siegfried lass uns die spawn Kategorie aufrufen. Tippe den Namen „spawn_siegfried“ ein und drücke Enter:



Damit Siegfried im Spiel zu einer neuen Zeit erscheint, muss ein Spawnpunkt mit demselben Tag „spawn_siegfried“, erstellt werden. Die Spawnpunkte werden in einem anderen Kapitel dieses Handbuches beschrieben

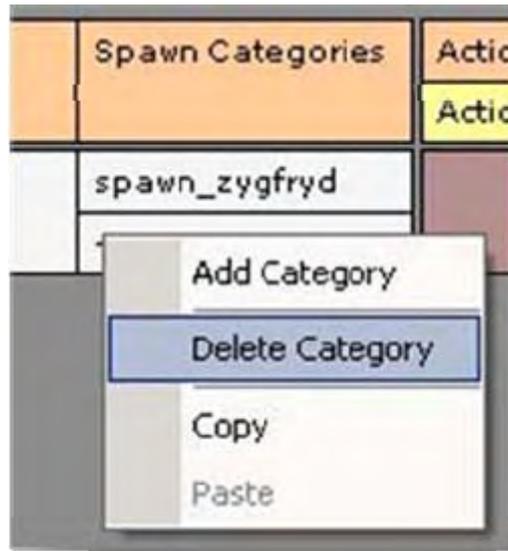
Jeder der Schlüssel NPC kann jede mögliche Zahl der Spawn Kategorie in einer Spawnphase haben. Wir sollten eine weitere Kategorie für Siegfried hinzufügen. Rechtsklicke unter der Spawn-Kategorie Spalte und die füge die ausgewählte Kategorie vom Menü hinzu. Das folgende Feld öffnet sich:



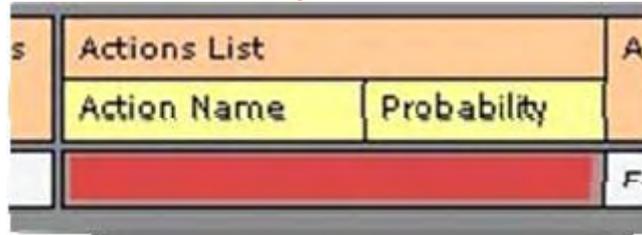
Diese zweite „Spawn Category“ brauchen wir nicht, Um Sie zu löschen, klicke mit der rechten Maustaste auf die 2 Spalte:



Wähle „Delete Category“ im Menü:



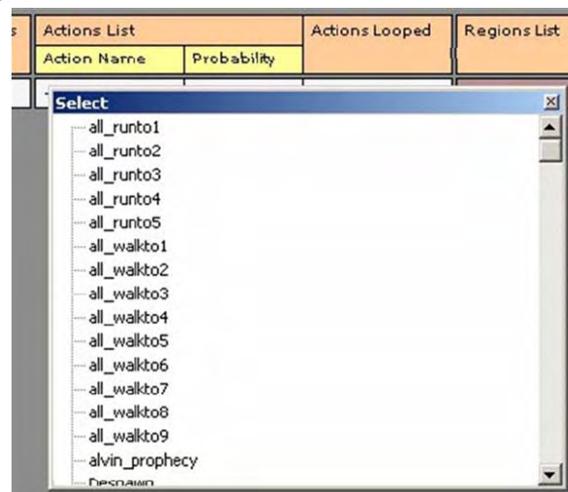
• **Action List.** Diese Spalte enthält die Aktionen, die Siegfried in einer gegebenen Spawnphase hinzugefügt wurde. Das sind set Animationen, die für einen bestimmten Charakter an einem bestimmten Moment abgespielt werden. Um eine Aktion im Feld hinzuzufügen mache einen Rechtsklick in der Actions Liste:



Wähle „Add Action“ Im Menü:



Das Auswahlfenster öffnet sich:



Dieses Fenster enthält eine umfangreiche Liste von Aktionen. Suchen die „w_h_pray“ Aktion auf der Liste (Siegfried wird dazu veranlasst zu beten) und mache einen Doppelklick:



Die Aktion wird jetzt in der Actions List erscheinen:

Actions List			A
Action Name	Probability		
w_h_pray	50%		F

Unter Probability(Chance) ist ein Wert von 50% erschienen. Das zeigt an, dass unser Charakter eine 50% Probability hat, und auch „w_h_pray“ Aktion anstatt eine andere Aktion zu einer bestimmten Zeit durchzuführen. Erinnern dich daran, nachdem du alle Aktionen, die ein Charakter in einer bestimmten Spawnphase durchführen soll, Eine Summe der Prozentsätze eingetragen werden muss, das es zusammen 100% ergeben. Wollen wir annehmen, dass Siegfried nur eine Chance von 5 % haben wird, die w_h_pray Aktion durchzuführen. Um das zu tun, klickst du auf das Feld unter der Probability doppelt:

Actions List			Ac
Action Name	Probability		
w_h_pray	50%		Fa

Tippe den Wert 5 ein und Drücke Enter. Es gibt keinen Grund, mit dem % Zeichen zu beginnen, es wurde automatisch hinzugefügt. Jetzt fügst du eine andere Aktion hinzu. Rechtsklicke auf die Aktion Story Phase Row. Dieses Mal, nachdem das auserwählte Fenster sich geöffnet hat, klickst du auf das kleine `x` um das Fenster zu schließen:



Das fügt eine Leere Action hinzu:

Actions List	
Action Name	Probability
w_h_pray	5%
<enter_name>	50%

Selbstverständlich kann es nicht leer gelassen werden. Öffnest du das auserwählte Fenster mit der Liste von Aktionen wieder, indem du auf das Empty Feld klickst, unter der Action Name List zu finden:

Actions List	
Action Name	Probability
w_h_pray	5%
<enter_name>	50%

Das *Ausgewählte* Fenster öffnet sich wieder. Dieses Mal machen wir die „w_h_alarmed“ Aktion ausfindig (wir verursachen das Siegfried sich während eines Alarms umsieht) dann doppelklickst du:



Diese Aktion erscheint in der Action Name List:

Actions List	
Action Name	Probability
w_h_pray	5%
w_h_alarmed	50%

Doppelklicke auf die Probability Spalte nahe bei der „w_h_alarmed“ Aktion ,und stelle die Probability auf 10% ein und drücke dann Enter:

Actions List	
Action Name	Probability
w_h_pray	5%
w_h_alarmed	50%

Wollen jetzt wir noch drei Aktionen für Siegfried hinzufügen: „all_walkto1“ (Wahrscheinlichkeit von 35 %), „all_walkto2“ (Wahrscheinlichkeit von 35 %) und „w_h_bored“ (Wahrscheinlichkeit von 15 %). Inzwischen ist es klar, wie man das macht. Die Aktion von Siegfried sollte wie folgt aussehen:

Actions List	
Action Name	Probability
w_h_pray	5%
w_h_alarmed	50%
all_walkto1	35%
all_walkto2	35%
w_h_bored1	15%

Erinnern dich daran, Der Gesamtwert der Probability sollte 100 % sein. Nach dem Spawnen wird Siegfried von Ort zu Ort gehen, knien oder von Zeit zu Zeit beten, ringsherum alles ansehen und sich manchmal vor Langeweile strecken. Wenn Du eine Handlung entfernen möchtest, Klicke mit der rechten Maustaste, und wähle „Delete Action“ Im Menü:

Actions List	
Action Name	Probability
w_h_pray	5%
w_h_alarmed	50%
all_walkto1	35%
all_walkto2	35%
v	%

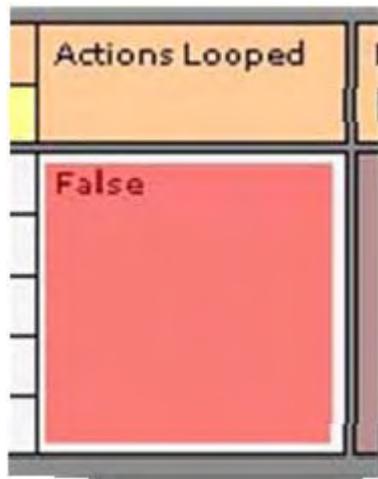
Add Action

Delete Action

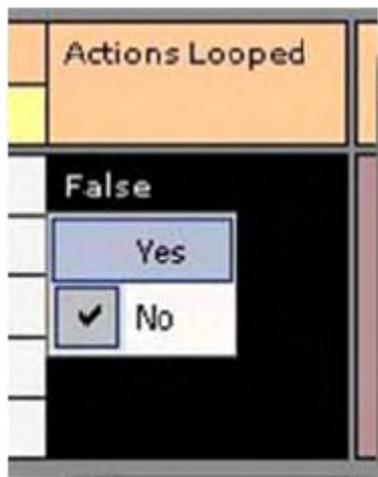
Copy

Paste

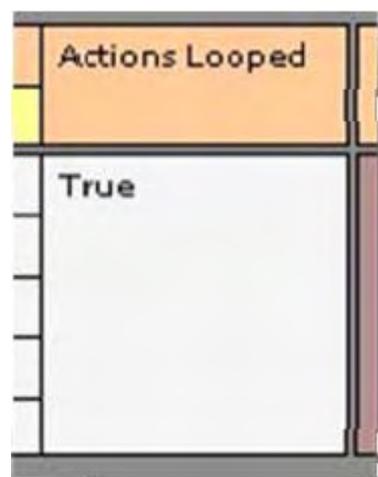
• **Actions Looped.** Das Setzen von "Actions Looped" auf True bildet aus den Aktionen in der Liste "Action Name" eine Schleife Dadurch wird Siegfried diese Aktivitäten kontinuierlich ausführen. Das Setzen von "Actions Looped" auf False sorgt dafür, dass Siegfried's Aktionen nur einmal durchgeführt werden und dann zum "Despawn Point" gehen. Um Aktionen (Actions) einer Schleife hinzuzufügen, Doppelklicke auf das Feld unter der "Actions Looped" Spalte:



Wähle „Yes“ aus im Menü:

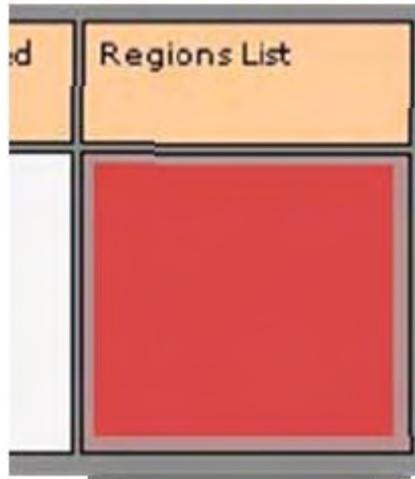


Das stellt den Loop der Aktion ein:

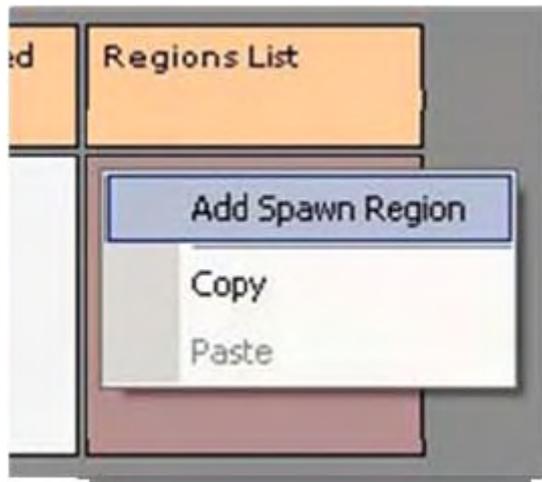


Um das zu ändern, ein Rechtsklick auf das Feld unter der Aktions Looped Spalte und im Menü „No“ auswählen.

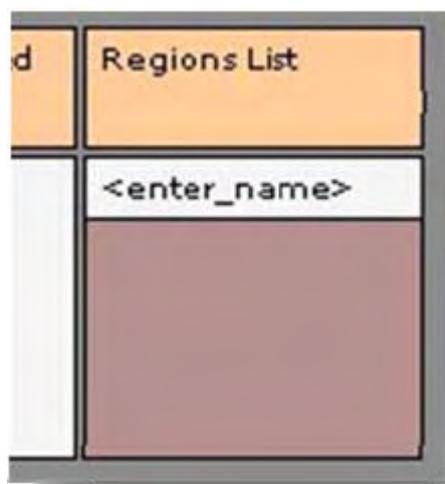
• **Regions List.** Die Spawn region ist ein Identifizierungszeichen für Aktionspunkte, die in einem anderen Kapitel dieses Handbuches beschrieben werden. Die Spawnregion erklärt dem Charakter, welche Aktion durchgeführt wurde, an den Aktionspunkten. Erinnerst Du Dich noch, ein Aktionspunkt ein Charakter eine Aktion durchführen zu können, muss eine identische Aktion auf der Liste haben. Um eine neue Spawnregion hinzuzufügen Rechtsklicke auf das Feld *Regions List*:



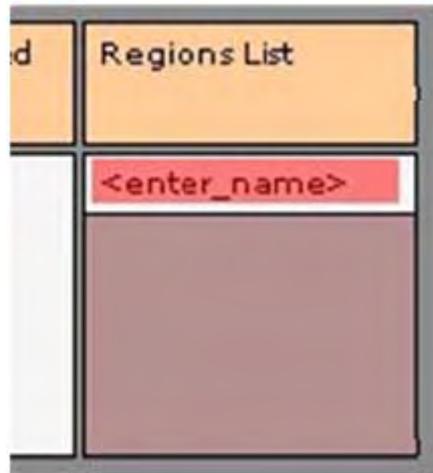
Wähle „Add Spawn Region“ Im Menü:



Dies fügt eine neue region hinzu:



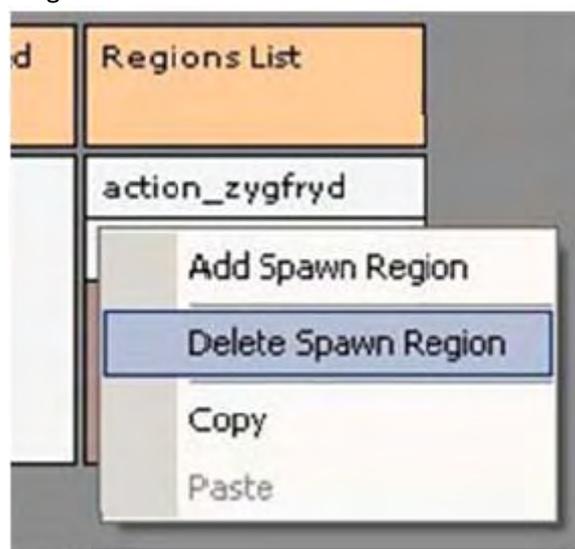
Nun sollten wir es umbenennen. Doppelklicke auf das Feld und füge unter der Regions-List den Namen „action_siegfried“ ein und Drücke Enter:



Erinnern dich daran, dass der Aktionspunkt, an dem Siegfried seine Aktionen durchführen soll, den gleichen Namen wie die Spawnregion haben muss. In jeder der Spawnphase kann ein Charakter jede mögliche Zahl der Spawnregion haben. Um eine andere Region hinzuzufügen, klicke mit der rechten Maustaste auf die Regions-List und wähle „Add Spawn Region“ vom Menü, ein neues Feld wird hinzugefügt:

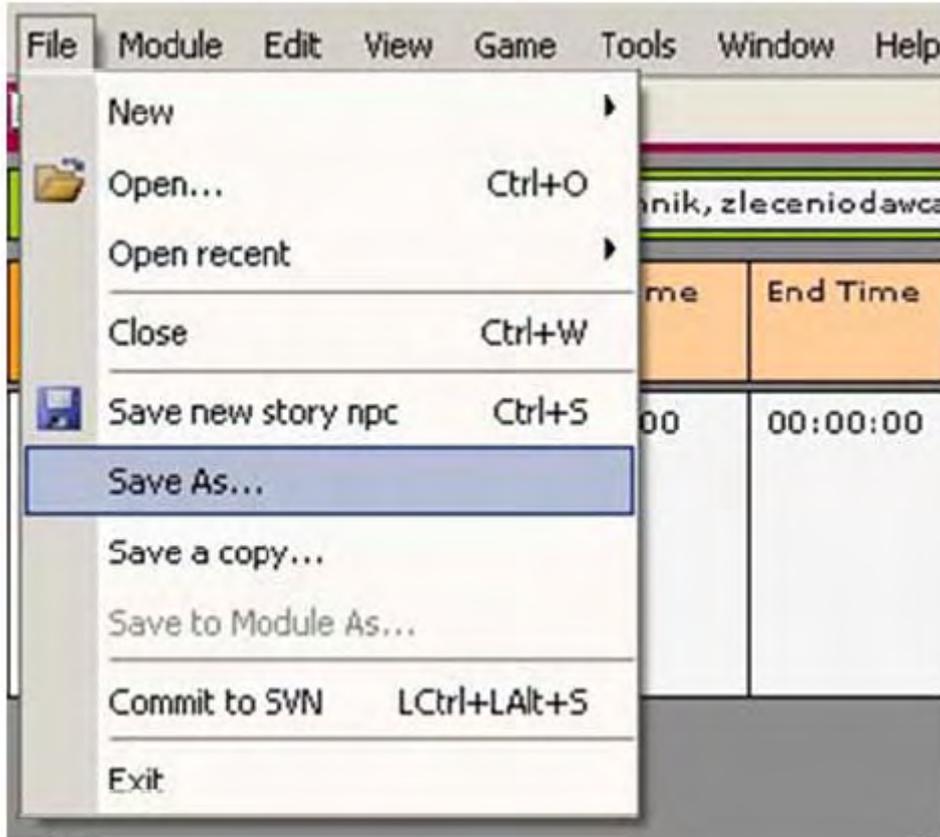


Diese zweite Spawn-Region ist für unsere Zwecke jedoch unnötig, um sie zu löschen, klicke mit der rechten Maustaste auf „Delete Spawn Region“ im Menü:



Das ist fast alles über das erstellen der Spawns, die für die Schlüssel-NPC Charaktere eingestellt werden. Es ist möglich, alle Haupt-NPC Charaktere zu einem Schlüsselcharakter im Spawn-Set zu speichern, aber das wird nicht empfohlen. Die beste Annäherung soll einem unterschiedlichen Set für jeden Charakter erstellen.

Jetzt sollten wir unser Set speichern. Um das zu tun, wählst du „Save As...“ im Menü:



Das öffnet das Standard Windows Fenster, das für die Speicherung der Daten benutzt wird. Außer der NPC Siegfried das stellen sie im Spawn Templates\NPCspawn Ordner ein. Gehe zu diesem Ordner, und trage den Namen „our_siegfried“ ein und klicke auf speichern. Erinnerung dich daran, dass der Name der Datei nicht über 16 Buchstaben enthalten darf. Andernfalls sieht der D'jinni Editor ihn nicht. Stelle sicher, dass die Datei im npc Format gespeichert wird:



Ist das Fenster grau untermalt, heißt das, dass das Fenster inaktiv ist:

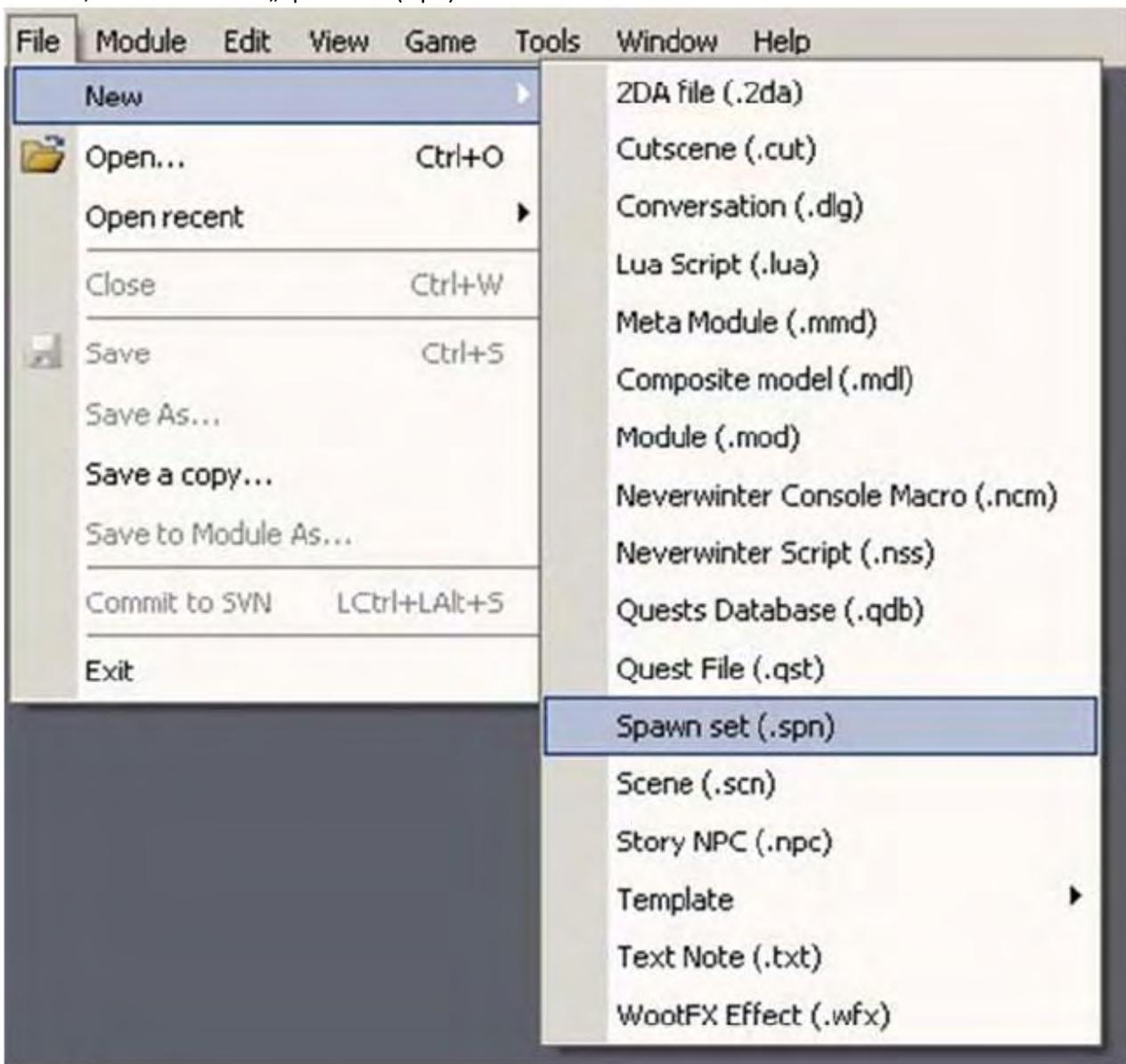


Eine andere Farbe bedeutet, dass dieses Fenster aktiv ist:



Ein Sternchen vor dem Namen in der Titelzeile bedeutet, dass die Datei seit der letzten Änderung nicht gespeichert wurde. Wir können auch statt „Save as“ aus dem Menü wählen, Man kann auch „Save new Story NPC“ auswählen. Das wird die Datei direkt im Modul speichern, diese Erläuterung wird im Handbuch nicht verwendet.

Jetzt werden wir einen anderen Charaktere gestalten d.h Bürger und Monster. Lass uns jetzt mit Jethro beginnen, ein Charakter, dessen Aufgabe sein wird Würfelpoker mit Geralt zu spielen. Um Jethro zu erschaffen, wähle im Menü „Spawn set (.spn)“:



Das „New spawn set“ Fenster öffnet sich:

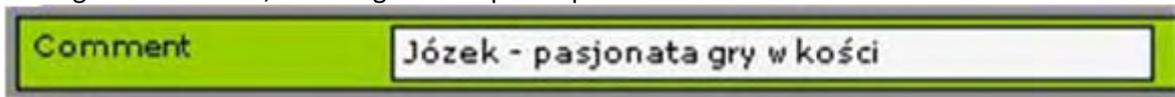


Wie mit dem „New Story npc“ Fenster ist das auch beträchtlich lang und kann mit der Scrollbar am unteren Bildschirmrand gesteuert werden. Das Fenster wird auch in einige Spalten aufgespaltet. Wenn auch einige spalten mit den neuen Spawns untergehen werden im Fenster, haben sie dieselbe Bedeutung wie in einer neuen „New Story“ fenster, wir besprechen nun wie man die Beschreibung von besonderen Sätzen hervor-gehoben werden.

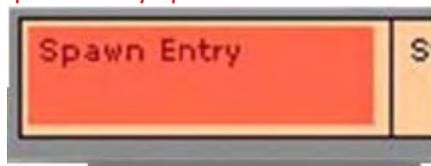
- **Comment.** Enthält einen Feld in den eine Anmerkung vermerkt werden kann. Dies wird im Spiel nicht verwendet, das ist nur ein Informationsfeld. du kannst schreiben was immer du möchtest. Um einen Eintrag hinzuzufügen, doppelklicke auf das weiße Feld gegenüber vom Wort *Kommentar*:



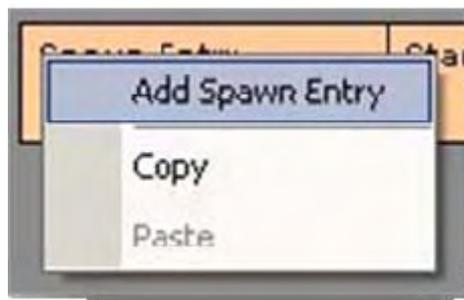
Für Jethro gibst Du "Jethro, der eifrige Würfelpokerspieler“ und drückst Enter:



- **Spawn Entry.** Spawn-Set für non-NPC Charaktere haben keine Spalte für Story-Phase und Spawn-Phasen. Das ist, weil Phasen nicht diese Arten der Charaktere betreffen. Spawneinträge kann mit Spawnphase verglichen werden, jedoch, Ein Charakter kann zu den unterschiedlichen Tageszeiten Spawn-Entry's, Aktionen zugewiesen werden, indem man unterschiedliche Spawneinträge Hinzufügt. Um eine neuen Spawneinträge Hinzuzufügen, Rechtsklicke auf die Spawn Entry Spalte:



Wähle „Add Spawn Entry“ im Menü:



Dies fügt eine neue Zeile hinzu:

Spawn Entry	Start Time	End Time	OnSpawn Script	Template	Spawn Cat
<enter_name>	00:00:00	06:00:00			

Zum umbenennen, klicke auf das Feld unter der Spawn Entry Spalte doppelt:

Spawn Entry	St
<enter_name>	00

Da Jethro gerne Würfelpoker spielt, tippt du „Poker“ ein und Drückst Enter:

Spawn Entry	St
poker	00

Um einen andere Spawneintrag hinzuzufügen, Rechtsklicke auf die Spawn-Entry Spalte und wähle im Menü „Add Spawn Entry“ um eine neue Spalte hinzuzufügen:

Spawn Entry	Start Time	End Time	OnSpawn Script	Template	Spawn C.
poker	00:00:00	06:00:00			
<enter_name>	00:00:00	06:00:00			

Wir brauchen den zweiten Eintrag nicht wirklich, selektiere beim zweiten Eintrag die „Start Time“ Spalte, und klicke mit der Rechten Maustaste:

Spawn Entry	Start Time	E
poker	00:00:00	0
<enter_name>	00:00:00	0

Daraufhin öffnet sich ein Menü in diesen wählst du „Delete Spawn Entry“:

Spawn Entry	Start Time	End Time	O
poker	00:00:00	06:00:00	
<enter_name>			

Add Spawn Entry

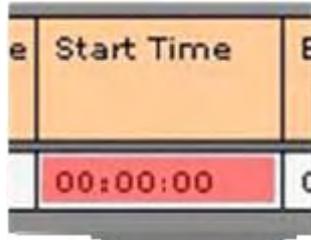
Delete Spawn Entry

Copy

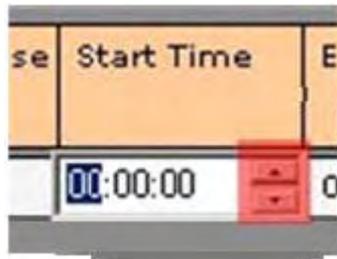
Paste

Klicke nicht mit der rechten Maustaste auf das Feld unter der „Spawn-Entry“ spalte da die Löschung des Spawns nicht dort zu finden ist.

• **Start time.** Unabhängige Charaktere, d.h. einfache Bürger, müssen auch Zeiten haben, während sie ihre eingestellten Aktionen durchführen. Die Start Zeit- und die Endzeit spalte haben die gleichen Funktionen wie mit den NPC Spawn-set. Eine Start Zeit für einen Spawneintragung zu definieren, Doppelklicke auf das Feld unter der Start Time spalte:

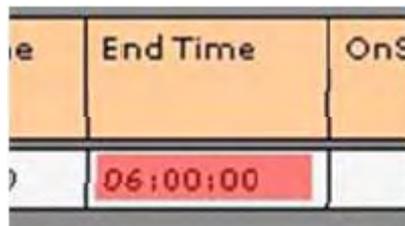


Die Zeit kann auch mit der Tastatur eingestellt werden oder, indem man die kleinen Pfeile verwendet, die auf der rechten Seite der Spalte zu finden sind:



Lass bei „Start Time“ 00:00:00 stehen.

• **End Time.** Das ist die Zeit die am Ende der Spalte ist, es ist jetzt 06:00:00. Wenn diese Zeit bliebe, würde Jethro um 00:00:00 erscheinen und um 06:00:00 verschwinden. In unserem Abenteuer, jedoch, müssen sowohl Siegfried als auch Jethro im Spiel die ganze Zeit, sowohl Tag als auch Nacht anwesend sein. Als mit NPC-Spawn-Set in der Größenordnung von unabhängigen Charakteren, gegenwärtig zu jeder Zeit beide bleiben, muss die Anfangs und Endzeit auf 00:00:00 gesetzt werden. Klicke auf dem Feld unter der End-Time spalte doppelt:

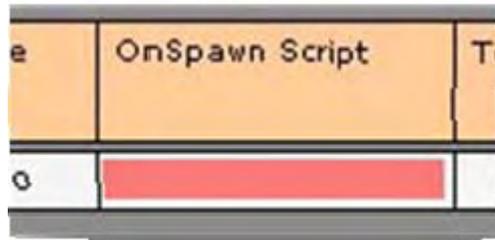


Tippe 00:00:00 ein und drücke Enter:

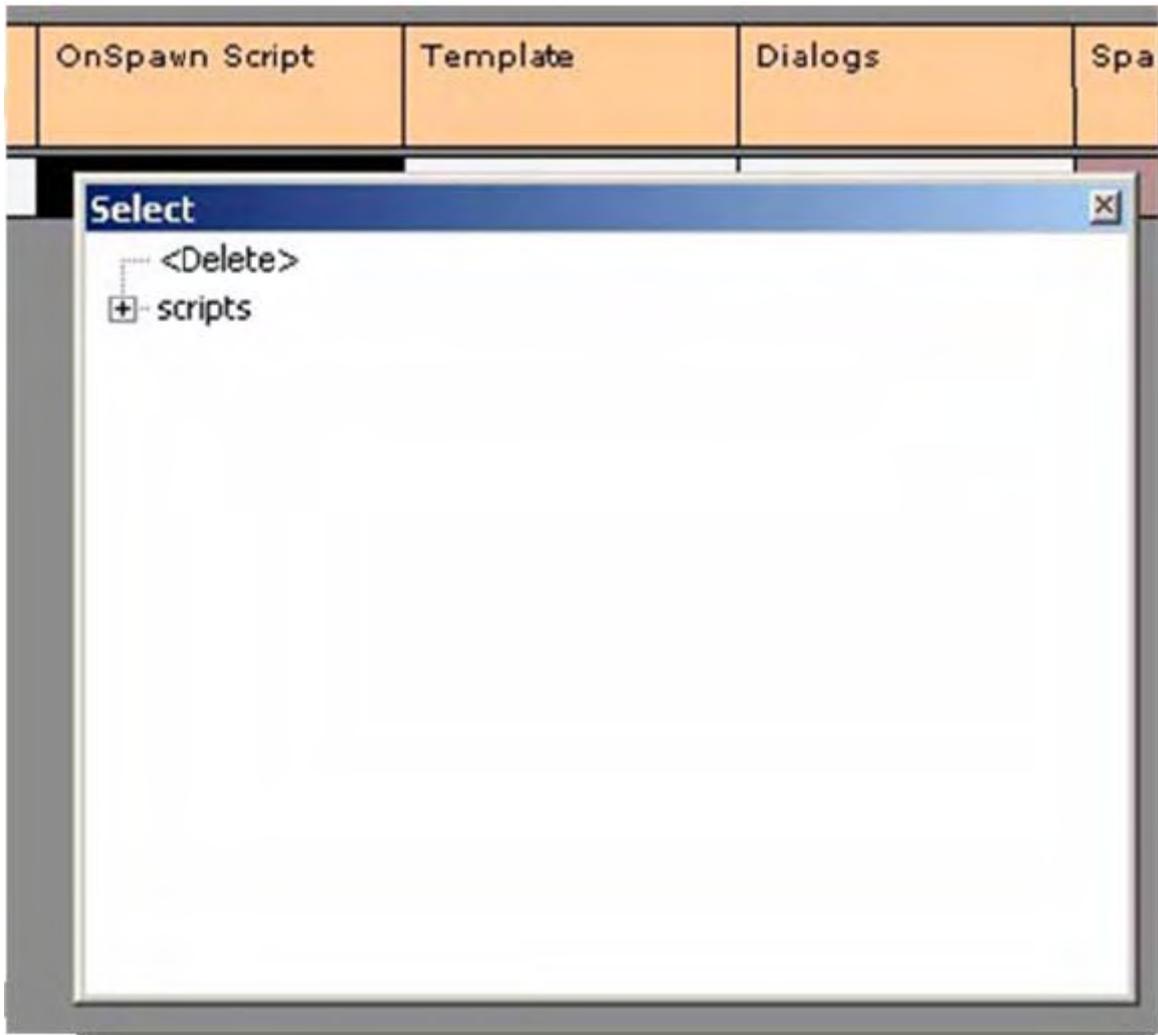


Erinnerst du Dich daran, das mit unterschiedliche Spawns, die Phasen sich nicht überschneiden dürfen, weil kein Charakter in zwei unterschiedlichen Zeiten an zwei Orten gleichzeitig sein kann.

• **OnSpawn Script.** In dieser Spalte wird das durchgeführte Script jede Zeit, die ein Unabhängigen Charakter im Spiel Spawnt, zugewiesen. Jethro erfordert nicht solch einen Script, aber wie man es einstellt wird beschrieben. Um Ein Script der Scriptspalte zuzuweisen. Doppelklickst du im Feld auf OnSpawn, um dann ein Script zuzuweisen:



Das folgende Fenster öffnet sich:



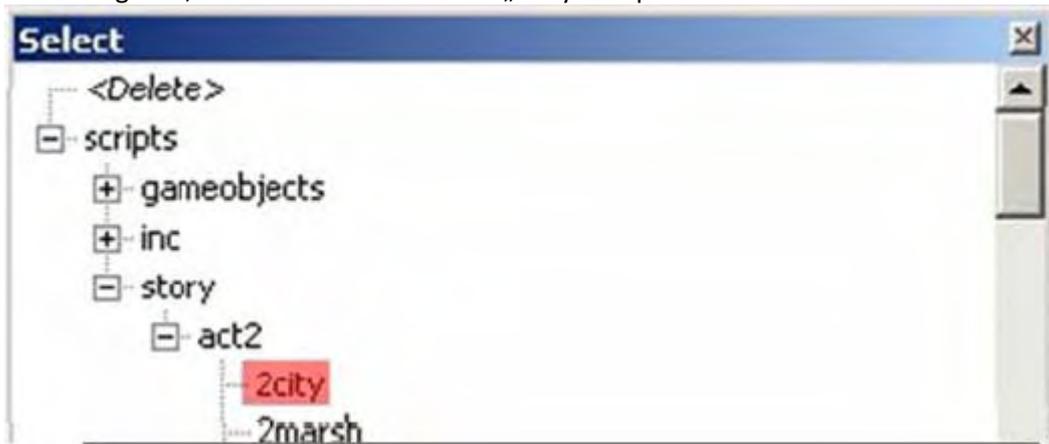
In diesem Fenster werden die ganzen Scripte angezeigt die es im Spiel gibt. Diese Liste wird in der Baumform angezeigt und die Struktur des Data\Scripts Ordner reflektiert (in, welchem alle Scripte gespeichert werden). Um einen Baum zu öffnen, klicken du auf das kleine '+' Zeichen:



Um den Baum zu schließen, klicke auf das kleine ` ` Zeichen:



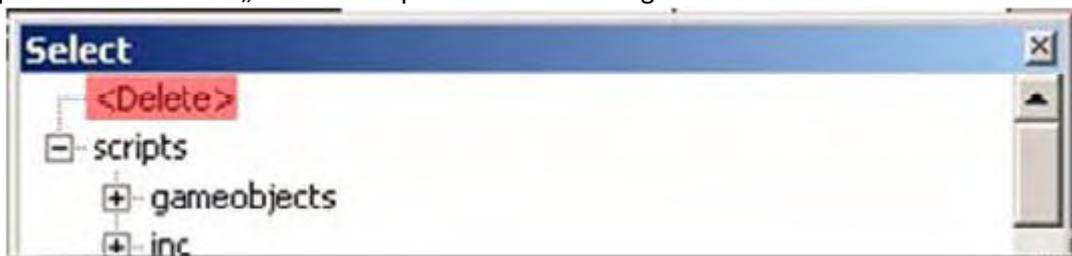
Nachdem du das richtige Script gefunden hast, klickst du doppelt drauf. Bei Jethro können wir das gleiche wählen, wie bei Siegfried, also wähle für Jethro das „2city“ Script:



Nachdem du auf das Script „2city“ doppelt geklickt hast, erscheint es im Feld unter der OnSpawn Scriptspalte:

	OnSpawn Script	Te
0	2city	

Um das Script wieder zu entfernen, doppelklicke auf das Feld unter der *OnSpawn script* Spalte doch dieses mal doppelklickst du auf die „<Delete>“ Option die findest du ganz oben im Fenster:

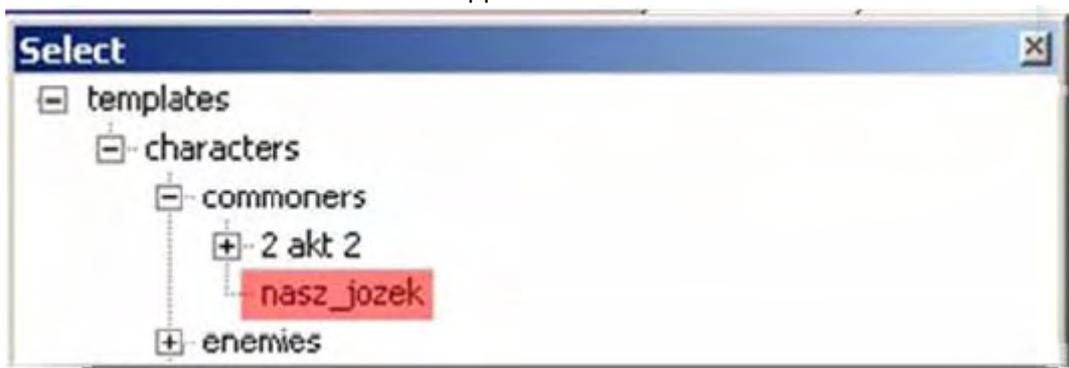


Merke Dir, dass ein anderes Script für jeden Spawneintrag zugewiesen werden kann.

- **Template.** Das ist, wo die Charakter Schablone, wo wir früher erstellten, Hinzugefügt wird. Doppelklicke im Feld unter der Template Spalte. Das auserwählte Fenster öffnet sich:



Das Fenster enthält eine Liste der Charakter Schablonen. Das in der Baum Form angezeigt und die Struktur des Data\Templates Ordners reflektiert. Dieses Mal, lassen wir Jethro's Schablone zuweisen. Erinnerst Du dich daran, dass wir es im Data\Templates\Characters\Commoners Ordner speicherten. Wähle die „our_jethro“ Schablone aus und mach einen Doppelklick drauf:



Das weist die our_jethro Charakter Schablone dem Spawneintrag hinzu:

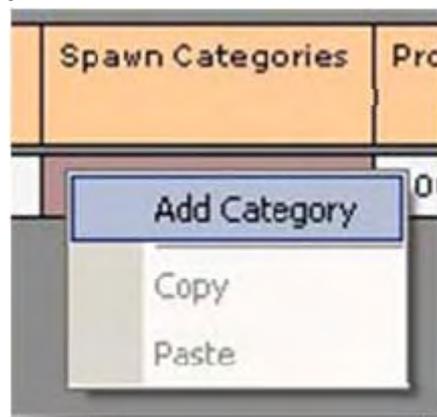


Wie mit einem NPCs, kann eine Charakter Schablone nicht entfernt werden. Sie kann zu einer anderen Schablone geändert werden. Ein unabhängiger Charakter kann eine andere Charakter Schablone in jedem Spawneintrag haben, aber damit die Änderung der Charaktere geht muss zuerst despawn und dann respawn sichtbar sein. Das erfordert wieder vom Spieler einige Eintragungen zu machen, für eine bestimmte Umgebung.

- **Spawn Categories.** Ein unabhängiger Charakter erfordert auch Spawnkategorien, die durch einen identifizierungszeichen, dem Charakter erklärt wird, wo es erscheint und/oder verschwindet soll. Um eine Spawnkategorie, im Feld der Spawn-Kategorie hinzuzufügen klicke mit einem Rechtsklick auf das Feld:



Wähle „Add Category“ im Menü aus:



Eine neue Spawnkategorie ist hinzugefügt worden. Um sie umzubenennen, Doppelklicke auf das Feld, das unter der Spawn-Kategorien Spalte zu finden ist:



Für Jethro, wollen wir die Spawn Categories „spawn_jethro“ nennen. Tippe den Namen ein und drücke Enter:



in der Größenordnung von Jethro, um im Spiel in einer späteren Zeit wiederzuerscheinen, muss ein spawn-Punkt mit demselben spawn_jethro tag geschaffen werden. Spawn-Punkte werden in einem anderen Kapitel dieses Handbuches besprochen.

Jeder unabhängige Charakter (d.h. einfacher Bürger oder Monster) kann jede mögliche Zahl der Spawnkategorien in jedem Spawneintrag haben. Lass uns eine weitere Kategorie von Jethro hinzufügen. Rechtsklicke unter der Spawn-Kategorien Spalte und füge die Category im Menü ein. das folgende Feld erscheint:

Spawn Categories	Pr
spawn_jozek	100%
<enter_name>	

Wir benötigen nicht noch eine zweite Spawnkategorie, also kann sie entfernt werden. Um eine Spawnkategorie zu entfernen, Rechtsklicke auf das passenden Feld unter der Spawn-Kategorie Spalte:

Spawn Categories	Pr
spawn_jozek	100%
<enter_name>	

Wähle „Delete Category“ vom Menü aus:

Spawn Categories	Prob
spawn_jozek	100%

Add Category

Delete Category

Copy

Paste

• **Probability.** Für Unabhängige Charaktere kann eine Wahrscheinlichkeit veranlasst werden um zu bestimmen, wie wahrscheinlich es sein wird, dass der Charakter im Spiel erscheinen wird. Das Setzen davon auf 100 % bedeutet, dass der Charakter immer im Spiel auftauchen wird. Ist der Wert weniger als 100 % bedeuten, dass der Charakter nicht erscheinen kann.

Zum Beispiel, ist eine Einstellung davon auf 10 %, heißt dass ein angegebener Charakter nur eine Chance von 10 % haben wird, im Spiel zu erscheinen. In einen bestimmten Raum, kann Geralt einen besonderen Charakter treffen. im dasselbe Zimmer bei einer verschiedenen Aktivität kann er, jedoch, nicht den Charakter treffen. Um die Probability eines Charakters zu setzen, der dann im Spiel erscheint, klicke auf das Feld unter dem Probability Feld:

Probability
100%

In unserem Abenteuer wollen wir, dass Jethro immer anwesend ist, seine Probability auf 100% setzen und dann Enter Drücken:

Probability	M
100%	1

Es gibt kein Bedürfnis auf dieses %-Zeichen einzugehen, da es automatisch hinzugefügt wird.

• **Min Creatures.** In dieser Spalte geben eine minimale Zahl von Charakteren dieses Types ein, der im Spiel gleichzeitig erscheinen wird . Um diese Zahl zu setzen, klicke auf das Feld unter der *Min Creatures* Spalte:

Min Creatures	M
1	4

Jethro ist einzigartig, also gebe eine 1 ein und drücke Enter:

Min Creatures	M
1	4

• **Max Creatures.** An dieser Stelle kannst Du die maximale Anzahl von Charakteren dieses Typs eingeben, die im Spiel gleichzeitig erscheinen können. Ändere den Wert durch einen Doppelklick in dieser Spalte:

Max Creatures	T
4	0

Gib wieder eine 1 ein. Und drücke Enter:

Max Creatures	T
1	0

• **Total Creatures.** Gib in dieser Spalte eine Zahl ein, Die die Gesamtzahl von unabhängigen Charakteren eines bestimmten Typs anzeigt, die während des Spiels erscheinen können. Ein Wert von 0 zeigt an, dass ein Charaktertyp unendlich oft erscheinen kann. Ein Wert, größer als 0, gibt einen fest definierten Wert an, wie oft dieser bestimmte Charaktertyp getötet werden kann, bis er nicht wieder erscheint.

Folgendes Beispiel macht die verschiedenen "Creature" Angaben deutlich:

Stell Dir ein Rudel von Wölfen vor. Der Wert in der „*Min Creature*“ Spalte ist 2, die „*Max Creature*“ Spalte hat einen Wert von 4 und „*Total Creature*“ einen Wert von 6. Zu einer bestimmten Zeit wird im Spiel ein Rudel von Wölfen erscheinen. Die Anzahl der Wölfe im Rudel liegt, Zufallsgeneriert, zwischen 2 und 4. Einmal können es 2 Wölfe sein, ein anderes Mal 3 oder 4. In unserem Beispiel werden 3 Wölfe erschienen.

Wenn Geralt (der Spieler) einen Wolf tötet, wird ein neuer an dessen Platz erscheinen. In diesen Fall ist ein Wolf tot und 3 lebendig. - eine Gesamtzahl von 4.
 Tötet Geralt einen weiteren Wolf, wird wieder ein neuer Wolf erscheinen, damit haben wir 2 tote und 3 lebendige Wölfe - eine Gesamtzahl von 5. Wenn die Gesamtzahl an Wölfen, ob tot oder lebendig 6 erreicht (Total Creatures), werden keine neuen Wölfe mehr erscheinen.

Um einen Wert für die Zahl der insgesamt möglichen Kreaturen einzugeben, doppelklicke in das Feld unter „Total Creatures“:



Gib für Jethro eine 0 ein, damit er immer wieder erscheinen kann, wenn er getötet wird (gleich aus welchem Grund). Drücke Enter:



• **Respawn Condition.** Diese Spalte definiert die Bedingung, nach der mit einem getöteten Charakter verfahren wird. Unter Auswahl folgender Bedingungen kannst Du Deinen Charakter wiederauferstehen lassen (oder auch nicht):

- After Time - Charakter wird nach einem in der Respawn Delay-Spalte definierten Zeitraum wieder erscheinen.
- Always - Charakter wird immer wieder neu erscheinen, solange alle Bedingungen (des Spawnsets oder der Quest) erfüllt sind.
- Never - Charakter bleibt tot.
- Next Entry - Charakter wird immer wieder erscheinen, wenn der Spieler eine entsprechende Region betritt.
- Next Day - Charakter wird erst am nächsten Tag wieder erscheinen.

Setze für Jethro die Bedingung auf Always. Doppelklicke in das Feld unter Respawn Conditions:



Wähle „Always“ aus dem Menü:



• **Respawn Delay.** Diese Spalte enthält die Zeit, nach der ein eigenständiger Charakter, ins Spiel zurückkehrt. Dieser Wert ist abhängig davon, ob in der Respawn Condition - Spalte After Time als Bedingung ausgewählt wurde. Wurde eine andere Respawn Condition gewählt, hat Respawn Delay keine Funktion. Um den Wert zu ändern, doppelklicke in das Feld unter Respawn Delay:



Weil wir die Respawn Condition Bedingung für Jethro auf „Always“ gesetzt haben, ist dieser Wert unwichtig. Wir lassen ihn auf 0:



• **Spawn Type.** In dieser Spalte legst du den Einstiegspunkt fest. Unter folgenden Bedingungen kannst du wählen:

On Entry - Charakter erscheint, wenn der Spieler die entsprechende Region betritt.

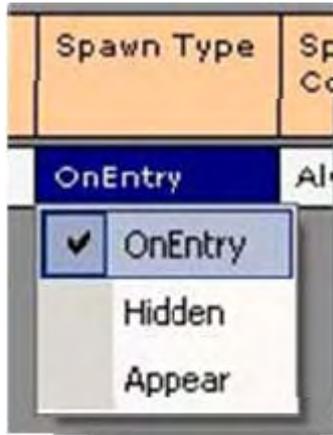
Hidden - Charakter versucht, in einem für den Spieler nicht sichtbaren Bereich zu erscheinen.

Appear - Charakter erscheinen, wenn der Spieler eine definierte Entfernung zum Einstiegspunkt (Spawnpoint) erreicht.

Wähle die Bedingung nach der Jethro erscheinen soll. Doppelklicke in das Feld unter der *Spawn Type* Spalte:



Wähle „OnEntry“ aus dem Menü:



Dies weist Jethro's den Spawn Type (Einstiegstyp) „OnEntry“ zu:



• **Spawn Condition.** Diese Spalte beschreibt die Bedingung unter dem ein eigenständiger Charakter im Spiel erscheinen wird. Unter folgenden Bedingungen kannst Du wählen:

- *Distance* - Der Charakter wird nur erscheinen, wenn der Spieler eine gewisse Entfernung vom Einstiegs-
punkt (Spawnpoint) hat. Die realen Entfernungswerte werden in den Spalten *Min Appear Distance*, *Appear*
Distance und *Disappear Distance* eingegeben.
- *Always* - Der Charakter wird ungeachtet bei der Spielerposition erscheinen.

Stell die Option Always für Jethro ein, Doppelklicke auf dem Feld unter der „Spawn Condition“:



Wähle „Always“ aus dem Menü:



Dies bestimmt die Bedingung unter der Jethro erscheint:



• **Min Appear Distance.** Wenn der Spieler näher am Charaktereinstiegspunkt ist, als der, in dieser Spalte definierte Wert, wird der Charakter nicht erscheinen. Wenn wir zum Beispiel hier einen Wert von 10 eintragen, der Spieler befindet sich aber nur 5 Einheiten vor dem Einstiegspunkt, wird der Charakter nicht erscheinen.

Erinnere Dich, dass für diese Einstellung *Distance* als *Spawn Condition* gesetzt sein muss. Wurde vorher *Always* ausgewählt, ist jeder Wert in der *Min Appear Distance* Spalte irrelevant. Nachdem wir *Always* unter *Spawn Condition* für Jethro gesetzt haben, besteht kein Bedarf mehr irgendetwas in der *Min Appear Distance* zu verändern. Wenn Du allerdings, etwas verändern möchtest, doppelklicke in das Feld unter die *Min Appear Distance* Spalte. Gib einen Wert ein und drücke Enter:



• **Appear Distance.** Bei der hier eingestellten Entfernung zwischen Spieler und Erscheinungspunkt (Spawnpoint) erscheint der Charakter im Spiel. Diese Spalte ist von der unter *Spawn Condition* eingestellten Bedingung abhängig.

Weil wir für Jethro, *Always*, als Einstiegsbedingung gesetzt haben, brauchen wir unter *Appear Distance* nichts einstellen. Wenn Du den Wert aber ändern willst, doppelklicke in das Feld unter *Appear Distance*:



Gib einen Wert ein und drücke auf Enter:



• **Disappear Distance.** In dieser Entfernung, zwischen Spieler Erscheinungspunkt (Spawnpoint) verschwindet der Charakter aus dem Spiel. Welchen Sinn hat diese Bedingung? Die Antwort ist relativ einfach. Es verringert den Rechenaufwand des PCs. Natürlich sollten Spawnpoints möglichst so platziert sein, dass Erscheinen und Verschwinden eines Charakters nicht im Blickfeld des Spielers stattfinden. (Die *Hidden* Option in der *Spawn Type* Liste, ist dafür auch hilfreich).

Damit der Wert dieser Spalte Auswirkungen auf das Spiel hat, muss *Distance* als Bedingung in der *Spawn Condition* Spalte gesetzt sein. Da Jethro *Always*, also immer erscheint, brauchen wir diesen Wert nicht ändern. Wenn Du allerdings den Wert verändern möchtest, doppelklicke in das Feld unter die *Distance* Spalte:



Setze einen Wert ein, und drück Enter:



• **SP Guard Distance.** Diese Spalte enthält die Entfernung, die sich ein Charakter maximal von seinem *Spawnpoint* entfernt. Wenn, zum Beispiel, Geralt zuerst auf einen Wolf zu läuft, dann aber flieht, wird der Wolf ihn nur, bis zu der hier eingestellten Entfernung, verfolgen. Danach kehrt der Wolf wieder zu seinem Erscheinungspunkt (Spawnpoint) zurück. Geralt und Jethro sind nicht verfeindet, darum hat diese Bedingung für uns keine Bedeutung. Solltest Du trotzdem den Wert ändern wollen, doppelklicke in das Feld unter die *SP Guard Distance* Spalte:

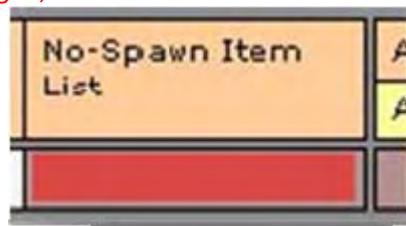


Gib den gewünschten Wert ein und drücke Enter.

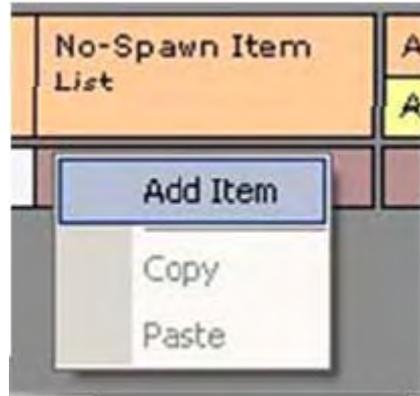
• **Spawn Interval.** Diese Spalte enthält die Zeitspanne, nach der ein Charakter (z.B. wenn gestorben) wieder im Spiel erscheint. Das ist nützlich, wenn mehrere Charaktere an einem Spawnpoint erscheinen sollen. Nachdem der erste Charakter erscheint, gibt die Spawn Interval Spalte vor, wie viel Zeit vergehen muss, bis der nächste Charakter am selben Spawnpoint erscheint. Der Wert in der Spawn Interval Spalte wird in Sekunden angegeben. Für Jethro tragen wir hier nichts ein, da er einmalig ist. Wenn Du möchtest kannst Du diesen Wert verändern. Doppelklicke in das Feld unter die Spawn Interval Spalte, lege einen Wert fest und drücke Enter:



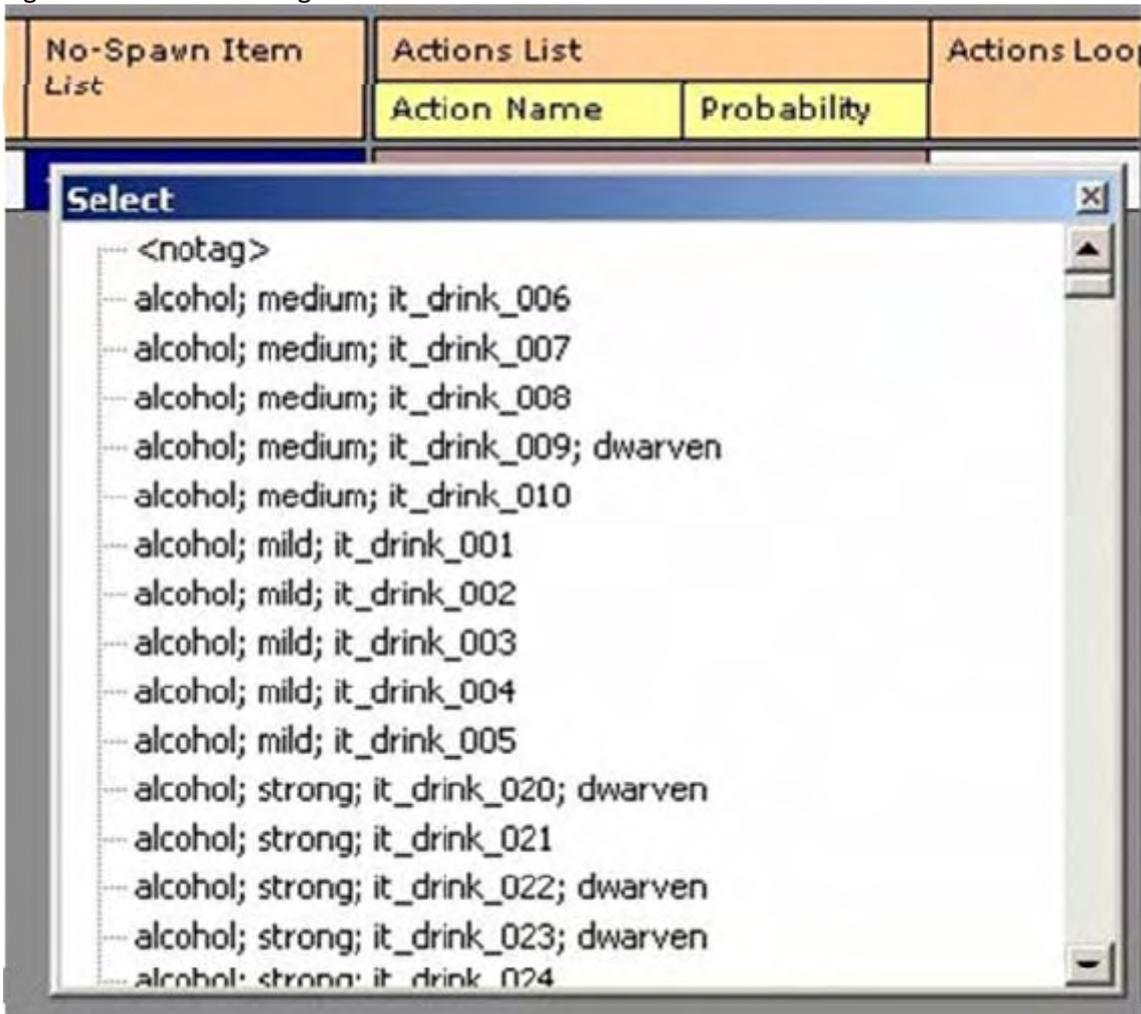
• **No-Spawn Item List.** In dieser Spalte kannst Du Gegenstände auswählen, die Geralt (der Spieler) vor dem Erscheinen von Charakteren schützen. Zum Beispiel könnte Geralt ein, ihn schützendes Amulett gegen Ertrunkene haben. Dieses Amulett gegen Ertrunkene (Charaktere) kann der *No-Spawn Item List* hinzugefügt werden. Es verhindert dann deren erscheinen, solange sich das Amulett in Geralts Besitz befindet. Um einen Gegenstand der Liste hinzuzufügen, rechtsklicke in das Feld unter der *No-Spawn Item List*:



Wähle "Add Item" aus den Menü:



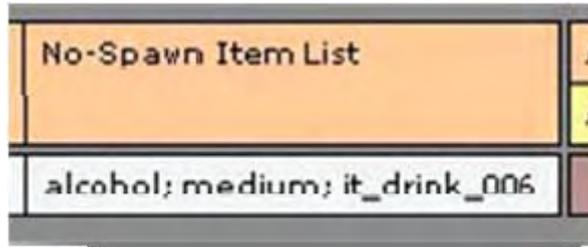
Das ausgewählte Fenster wird geöffnet:



Eine Liste der, im Spiel verfügbaren Gegenstände, wird angezeigt. Lass uns für (bzw. gegen) Jethro Alkohol auswählen. Doppelklicke dafür auf Alcohol; medium; it_drink_006 in der Liste:



Das fügt den Gegenstand der No-Spawn Item List hinzu:



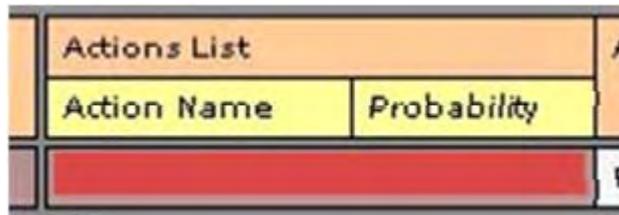
So lange Geralt nun den entsprechenden Alkohol, (Alcohol; medium; it_drink_006) in seinem Inventar hat, wird Jethro nicht im Spiel erscheinen. Für unser Spiel ist diese Bedingung ungünstig. Entferne deshalb den Gegenstand wieder aus der Liste. Mach es folgendermaßen. Rechtsklicke in das Feld unter die *No Spawn Item List*:



Wähle „Delete Item“ aus dem Menü:

Der *No-Spawn Items List* können natürlich auch mehrere Gegenstände hinzugefügt werden.

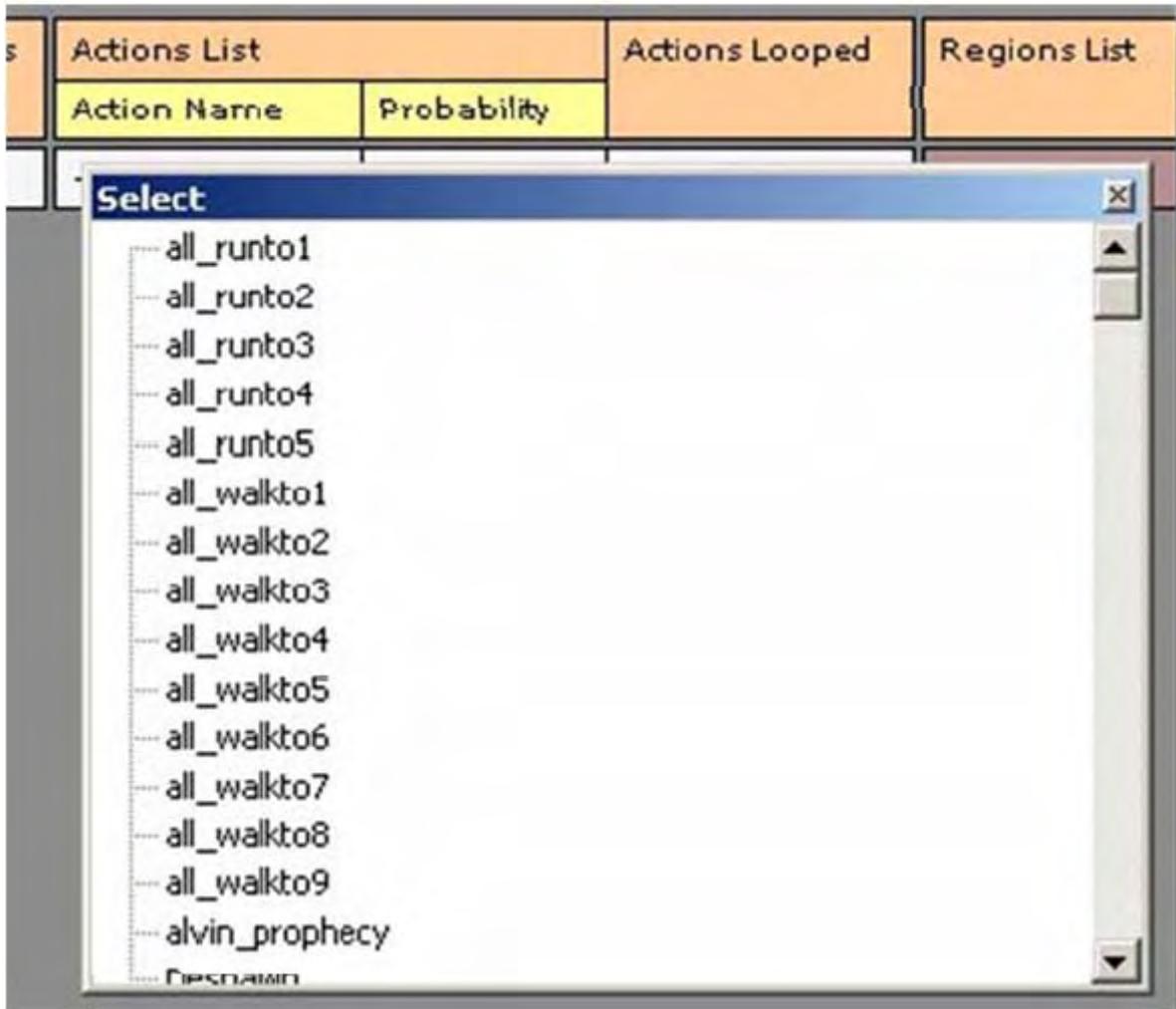
• **Action List.** Genau wie *NPCs* können auch *eigenständige Charaktere* Aktionen ausführen, sobald sie im Spiel erschienen sind. Dieses sind einfache Abläufe die der gewählte Charakter zu einer bestimmten Zeit ausführen soll. Um eine Aktion auszuwählen, rechtsklicke in das Feld unter die *Action List* Spalte:



Wähle „Add Action“ aus dem Menu:



Das gewählte Fenster wird geöffnet:



Lass uns jetzt Jethro einige Aktionen zuweisen. Such nach der w_h_tired1 Aktion aus der Liste und doppelklicke darauf:



Die Aktion wird in der Action Name Spalte erscheinen:

Actions List	
Action Name	Probability
w_h_tired1	50%

Ein Wert von 50% wird in der Probability Spalte (Wahrscheinlichkeit) angezeigt. Es besteht also die 50% Chance, dass unser Charakter die nebenstehende Aktion ausführt. Denk daran, wenn Du deinem Charakter mehrere Aktionen zuweist, dass die Summe der Werte unter Probability immer 100% ergeben muss. Setze für Jethro's Aktion w_h_tired1 einen Wert von 25% ein. Doppelklicke in das Feld unter die Probability Spalte:

Actions List	
Action Name	Probability
w_h_tired1	50%

Trage 25 ein und drücke Enter. Es ist nicht notwendig das % Zeichen einzugeben, da es automatisch hinzugefügt wird. Fügen wir Jethro noch eine weitere Aktion zu. Rechtsklicke wieder in das Feld unter die Action Name Spalte. Das gewählte Fenster wird geöffnet. Such diesmal nach der all_walkto1 Aktion (Jethro läuft zu einem bestimmten Punkt) und doppelklicke darauf:



Die Aktion wird in der Action Name Spalte erscheinen:

Actions List	
Action Name	Probability
w_h_tired1	25%
all_walkto1	50%

Setzt die Wahrscheinlichkeit, zu der Jethro diese Aktion ausführt auf 75%. Doppelklicke dafür in das zweite Feld unter die Probability Spalte:

Actions List	
Action Name	Probability
w_h_tired1	25%
all_walkto1	75%

Gib den Wert 75 ein und drücke Enter:

Actions List	
Action Name	Probability
w_h_tired1	25%
all_walkto1	75%

Diese zwei Aktionen sind für Jethro ausreichend. Erinnerung Dich: Die Werte unter der Probability Spalte müssen zusammen 100% ergeben. Wenn Du eine Aktion entfernen möchtest, markiere sie mit einem Rechtsklick, und wähle „Delete Action“ im Menü:

Actions List	
Action Name	Probability
w_h_tired1	25%
all_walkto1	75%

Add Action
Delete Action
Copy
Paste

• **Action Looped.** Wenn die Bedingung für die Action Looped Spalte auf True gesetzt wird, wiederholt Jethro die unter Action Name aufgeführten Aktionen immer wieder. Änderst Du die Bedingung auf False, wird Jethro die Aktion(en) nur einmal ausführen, und dann reglos am Erscheinungspunkt (Spawnpoint) stehen. Doppelklicke in das Feld unter die Action Looped Spalte:

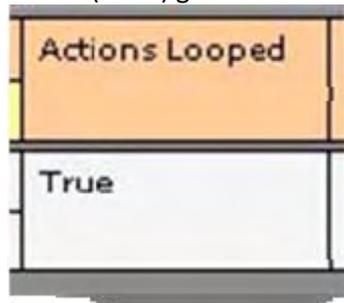
Actions Looped
False

Wähle Yes aus dem Menü:

Actions Looped
False

Yes
✓ No

Die Bedingung für die Schleife wurde auf *True* (wahr) geändert:



Willst Du sie erneut verändern, rechtsklicke in das Feld unter die *Action Looped* Spalte, und wähle *No*.

- **Always Start at Begin.** Beim setzen der *Always Start at Begin* Bedingung auf *True*, wird das *Spawnset* bei Spielbeginn aktiviert.
- **Regions List.** In dieser Liste werden Marker für *Actionpoints* eingetragen, bei denen die unter *Action Name* aufgeführten Aktionen stattfinden. *Actionpoints* werden in einem anderen Kapitel ausführlich beschrieben.
- **Selected actions.** Um unter *Region List* eine neue *spawn_region* (Marker) hinzuzufügen, rechtsklicke in das Feld unter die Spalte der *Regions List*:



Wähle „Add Spawn Region“ im Menü:



Eine neue Zeile, „<enter_name>“ ist erschienen:



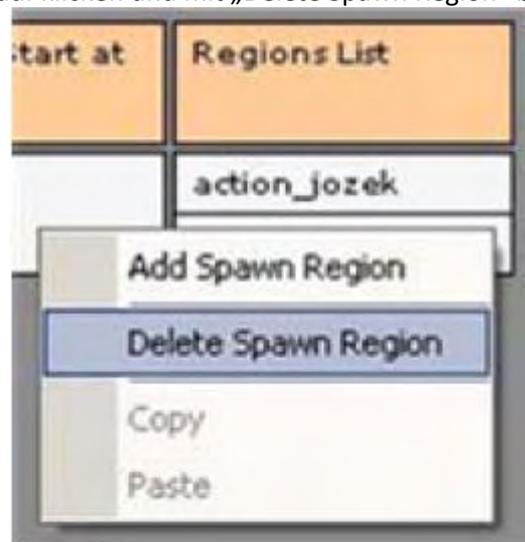
Jetzt müssen wir den Namen der spawn_region eintragen. Doppelklicke dafür auf „<enter_name>“:



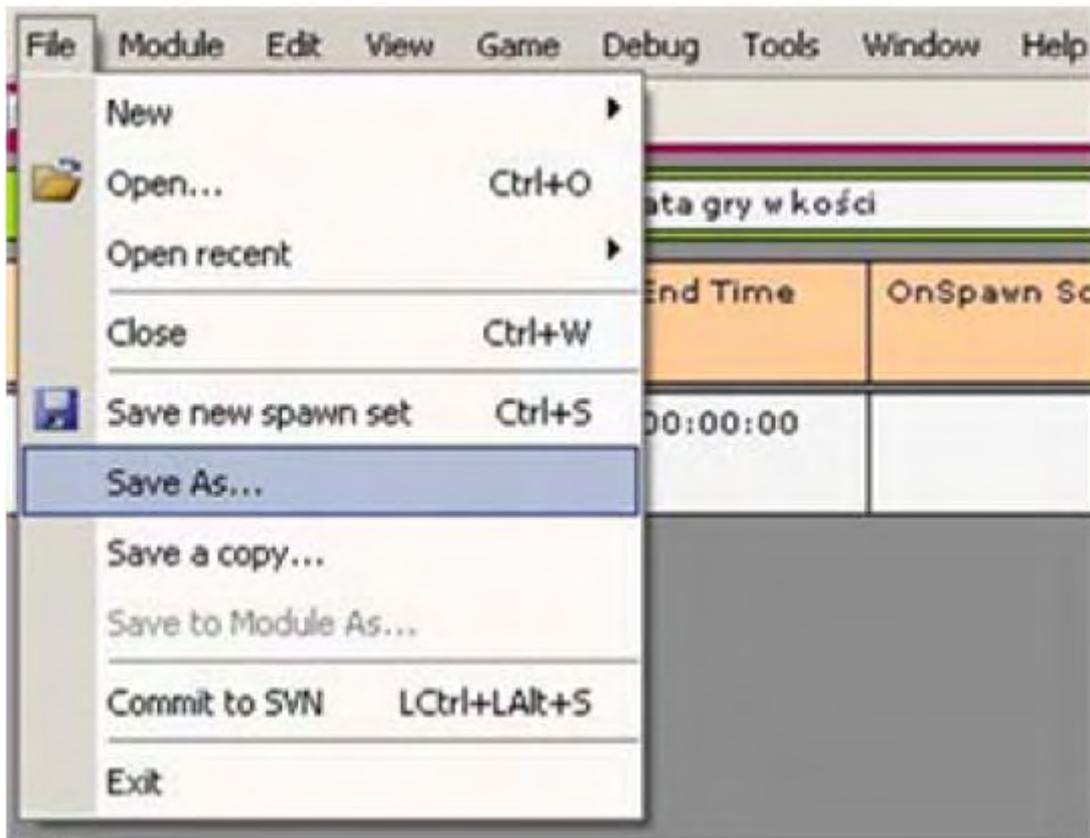
Für Jethro werden wir die Spawnregion „action_jethro“ nennen. Gib diesen Namen ein und drücke Enter. Dieser Marker muss später auch in den Einstellungen des zugehörigen Actionpoints unter Region_Name eingetragen werden. Mit jedem neuen Erscheinen im Spiel kann man einem Charakter auch eine neue Gruppe von Aktionen in einer neuen „spawn_region“ zuweisen. Um eine neue spawn_region in der Regions List anzulegen, rechtsklicke in das Feld in der Regions List und wähle „Add Spawn Region“ aus:



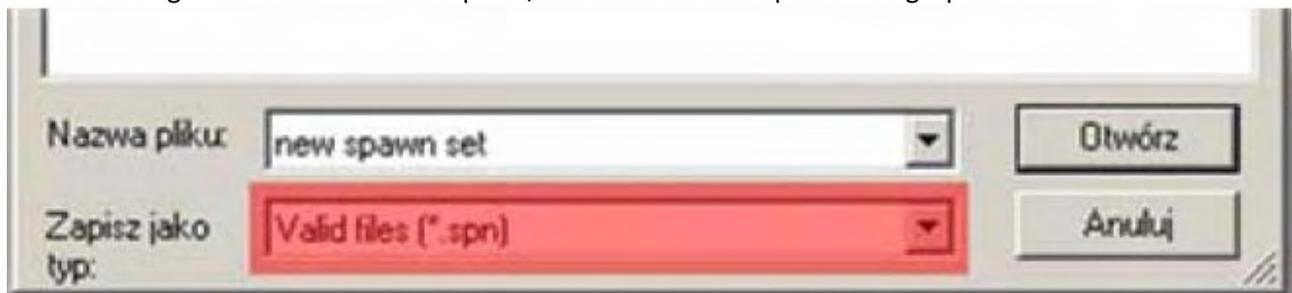
Jethro benötigt dieses Gebiet aber nicht. Um ein unwichtiges Spawn Gebiet löschen zu können, musst du mit der rechten Maustaste darauf klicken und mit „Delete Spawn Region“ bestätigen:



Nun ist das Spawn Gebiet gelöscht. Das Gleiche wird auch passieren, wenn normale Bürger gespawnt werden. Jetzt müssen wir speichern. Und um das zu tun, musst du auf „Save as“ klicken, welches im Menü „File“ zu finden ist:



Dann wird das Standard Windows Fenster automatisch geöffnet, welches für die Speicherungen der Daten zuständig ist. Das Spawn-Set des einfachen Bürgers muss im Ordner „Templates\Spawnsets“ gespeichert werden. Dort findest du auch den Ordner „Commoner“. Wir können Jethro dort abspeichern. Öffne den Ordner und gib dann „our_jethro“ ein, und klicke auf Speichern. Erinnerst du dich noch daran, dass der Name einer Datei nicht mehr als 16 Zeichen haben darf? Wenn er mehr hat, wird der D'jinni Editor diesen Namen nicht ganz lesen können. Überprüfe, ob der Ordner im .spn Format gespeichert wurde.



Wenn stattdessen eine andere Dateiart aufgelistet wird, bedeutet es wahrscheinlich, dass du versuchst, den Inhalt in einem anderem geöffneten Fenster zu speichern. Bevor du ein Spawn Set abspeicherst, solltest du dich vergewissern, ob das neue Spawn gesetzte Fenster aktiv ist. Wenn dies nicht der Fall ist, wird die Titelleiste grau:



Eine andere Farbe bedeutet, dass dieses Fenster aktiv ist:



Wenn ein Stern vor dem Namen in der Toolbar steht, bedeutet es, dass diese Datei nicht gespeichert wurden ist, seitdem die letzte Änderung durchgeführt wurde. Wähle „Schablonen“ aus zum Speichern eines neuen Spawn-Sets. Die Datei wird direkt im Modul gespeichert. Jetzt wollen wir 3 NPC Spawn-Sets erstellen. Einmal für den Banditen-Anführer, für den Professor und seinem Wächter sowie auch für den Wolf. Um dieses leichter zu machen, werden die Werte für jede Spalte eingegeben. Wähle New-> Spawn-Set (.spn) aus.

Der Professor, Der Banditenanführer:

- Comment – The Professor, the bandit leader,
- Spawn Entry – bandit,
- Start Time – 00:00:00,
- End Time – 00:00:00,
- Template – our_professor,
- Spawn Categories – spawn_professor,
- Probability – 100%,
- Min Creatures – 1,
- Max Creatures – 1,
- Total Creatures – 1,
- Respawn Condition – Always,
- Respawn Delay – 0.000,
- Spawn Type – OnEntry,
- Spawn Condition – Always,
- Action List – w_h_guard 100%,
- Action Looped – Yes (True),
- Region List – action_professor.

Lass die andere Werte unverändert, d.h. mit ihren Default-Werten. Außer bei den NPCs stellst du die Spawns in den Ordner Templates\Spawnset\Enemies unter dem Namen „our_professor“ ein. Vergiss nicht, dass die Datei im spn Format sein muss.

Für den Schutz des Banditenanführers müssen wir zwei Spawn-Einträge hinzufügen:

- Comment – The Professor’s guards,
- First spawn entry:
- Spawn Entry – guard1,
 - Start Time – 00:00:00,
 - End Time – 00:00:00,
 - Template – our_bandit,
 - Spawn Categories – spawn_bandit1,
 - Probability – 100%,
 - Min Creatures – 2,
 - Max Creatures – 2,

- Total Creatures – 2,
- Respawn Condition – Always,
- Respawn Delay – 0.000,
- Spawn Type – OnEntry,
- Spawn Condition – Always,
- Action List – w_h_guard2 100%,
- Action Looped – Yes (True),
- Region List – action_bandit1,

Second spawn entry:

- Spawn Entry – guard2,
- Start Time – 00:00:00,
- End Time – 00:00:00,
- Template – our_bandit,
- Spawn Categories – spawn_bandit2,
- Probability – 100%,
- Min Creatures – 2,
- Max Creatures – 2,
- Total Creatures – 2,
- Respawn Condition – Always,
- Respawn Delay – 0.000,

- **Spawn Type** – OnEntry,
- **Spawn Condition** – Always,
- **Action List** – w_h_guard2 100%,
- **Action Looped** – Yes (True),

Für die NPCs speicherst du im Spawns Ordner Templates\Spawnset\Enemies den Namen „our_bandit“.

Wolf:

- Comment – Wolves,
- Spawn Entry – hunt,
- Start Time – 00:00:00,
- End Time – 00:00:00,
- Template – our_bandit,
- Spawn Categories – spawn_wolf,
- Probability – 100%,
- Min Creatures – 5,
- Max Creatures – 5,
- Total Creatures – 5,
- Respawn Condition – Always,
- Respawn Delay – 0.000,
- Spawn Type – OnEntry,
- Spawn Condition – Always,

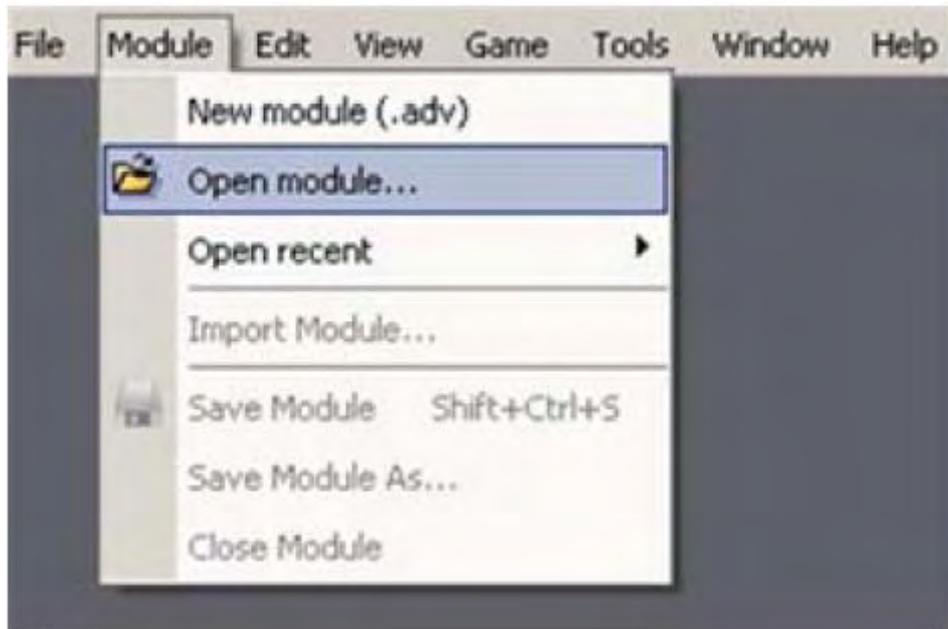
Weise keine Aktion oder Spawnregion den Wölfen zu. Speichere die Spawn Sets im Ordner Templates\Spawnset\Enemies unter dem Namen „our_wolf“.

Dir sollte bewusst sein, dass NPCs im Spiel vom allerersten Augenblick an sichtbar sein werden, aber andere Charaktere werden nicht sichtbar sein. Sie müssen erst erzeugt werden. Mehr darüber wurde in einem anderen Kapitel beschrieben. Kommen wir jetzt zu den NPCs und zu den Bürgern.

SPAWN POINTS

In diesem Kapitel gehen wir darauf ein, wie man Spawnpunkte erstellen kann. Was ist ein Spawnpunkt? Es ist ein Punkt im Spiel, wo ein angegebener Charakter gespawnt ist.

Die Bedingungen, unter denen ein Charakter erstellt wurde, hängen von den Einstellungen der ausgewählten NPCs oder/und den allgemeinen spawnenden Systemen ab. Wir brauchen 6 Spawnpunkte für unser Abenteuer. Wollen wir zuerst das Modul öffnen, in dem das Abenteuer erstellt wird. Wähle ein offenes Modul vom Modul-Menü aus:

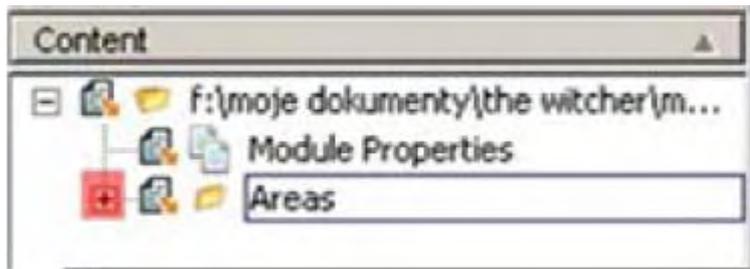


Das öffnet das Standard Windows Fenster, das für das Laden der Ordner benutzt wird. Suche nach dem Modul mit dem Zusatz unseres Abenteurers (Dateien werden in dem Ordner Settings\username\My documents\The Witcher gespeichert).

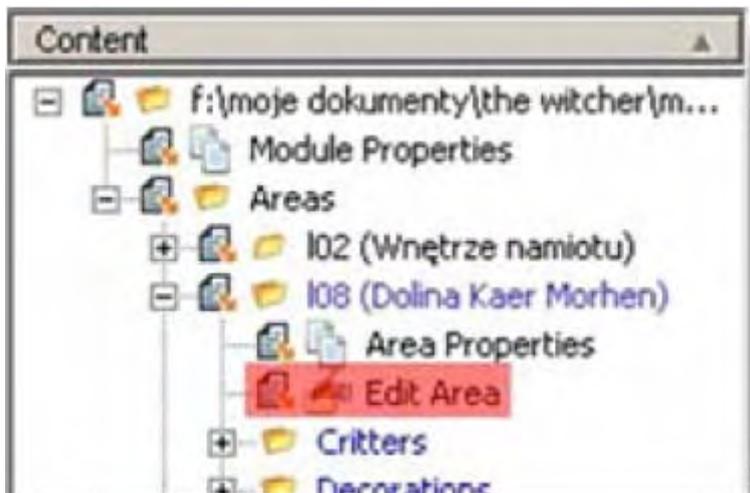
Hebe es hervor und klicke auf öffnen. Nachdem dieses Modul geöffnet ist, erscheint ein Modul-Explorer Fenster auf der linken Seite des Schirmes:



Mehr Informationen über das Interface des D'jinni's sind in einem anderen Kapitel dieses Handbuchs näher beschrieben. Um einen Spawnpunkt hinzuzufügen, müssen wir den Spielbereich editieren. Öffne den Baum, der diesen Bereich enthält, indem du auf das kleine '+' Zeichen klickst, welches nahe bei den Namen zu finden ist:



Nach dem der Bereich in diesen Baum erweitert wurde, machst du einen Doppelklick auf dem „Edit Area“ Button:



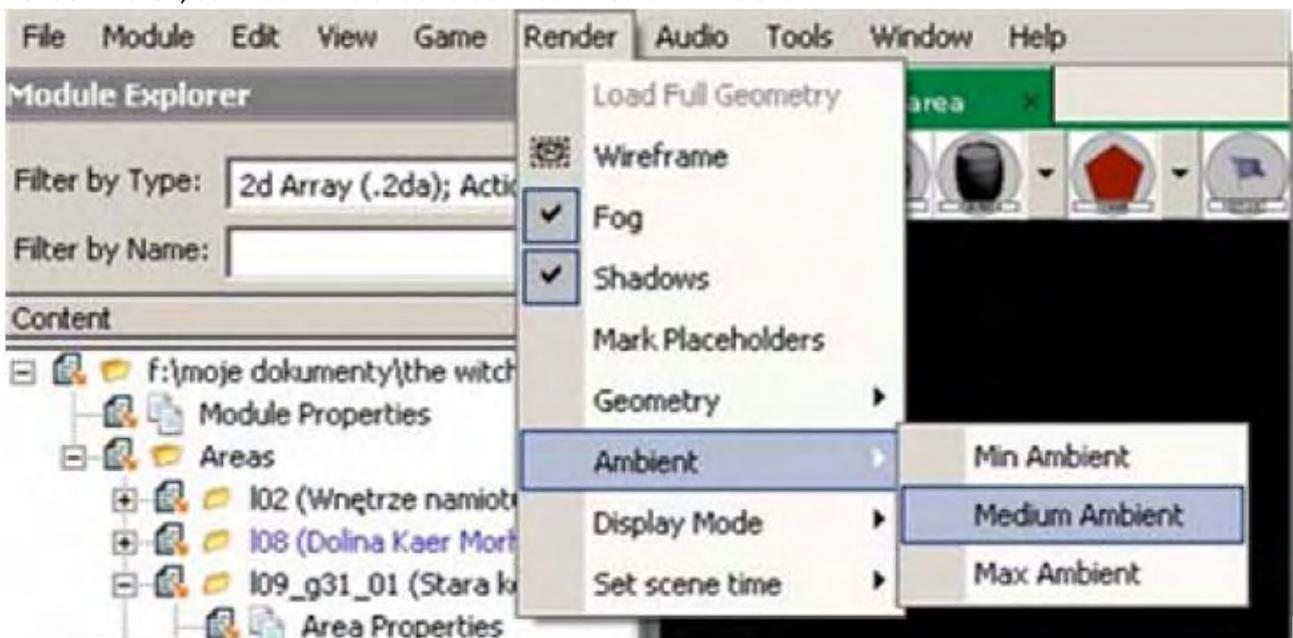
Jetzt wird ein Fenster erscheinen, wo du auf „Entire Geometry“ klickst. Damit wird das Gebiet geladen:



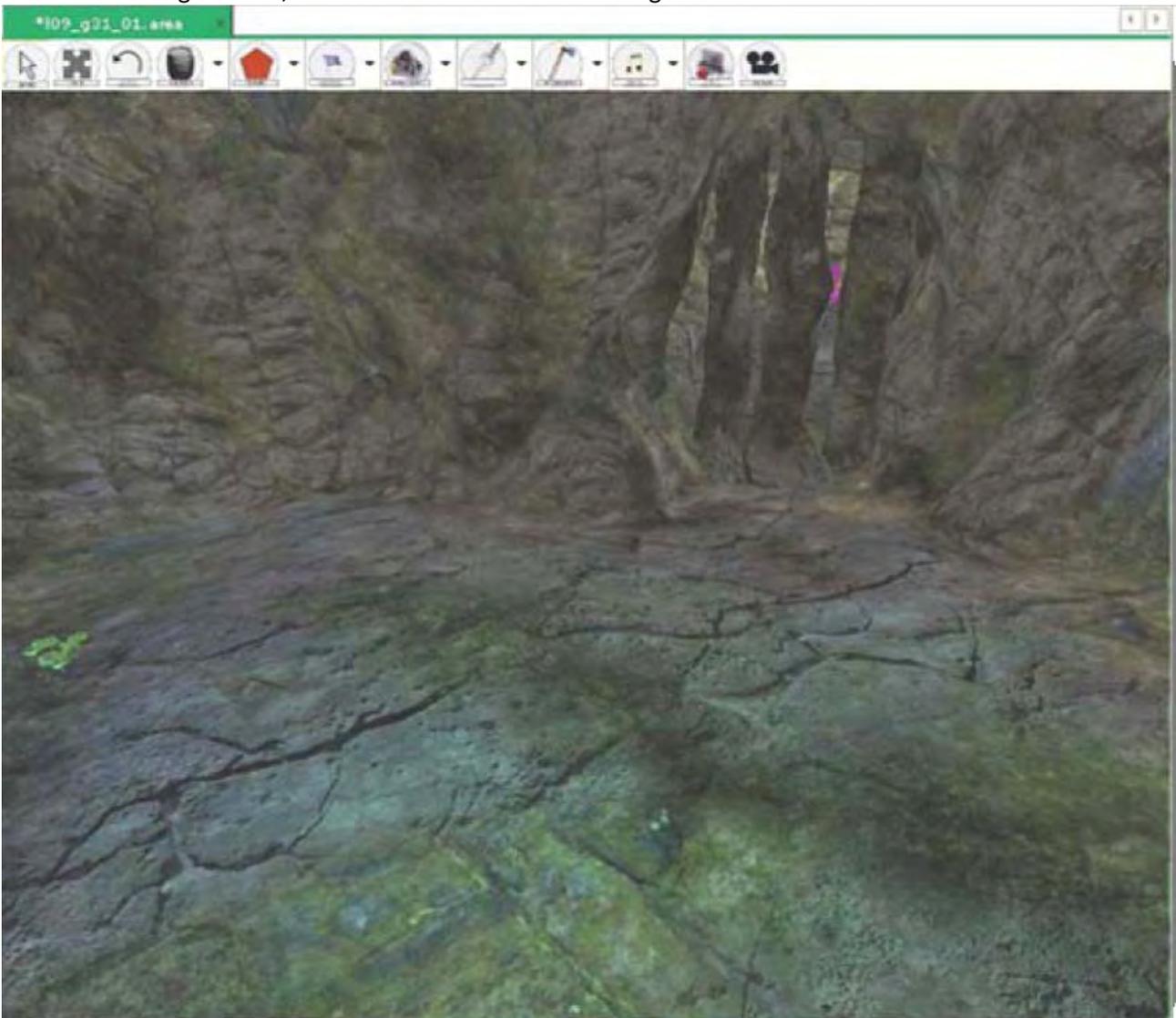
Nach einer Weile wird das Fenster im Editor für Gebiete geladen:



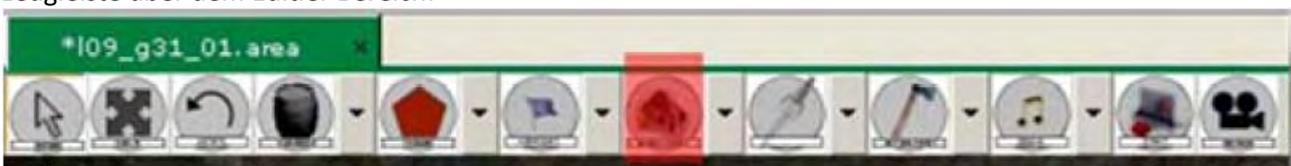
Falls das Gebiet zu dunkel ist, während es lädt, kannst du es erhellen. Dafür musst du oben in der Toolbar auf Render klicken, dann auf Ambient und dann auf Medium Ambient:



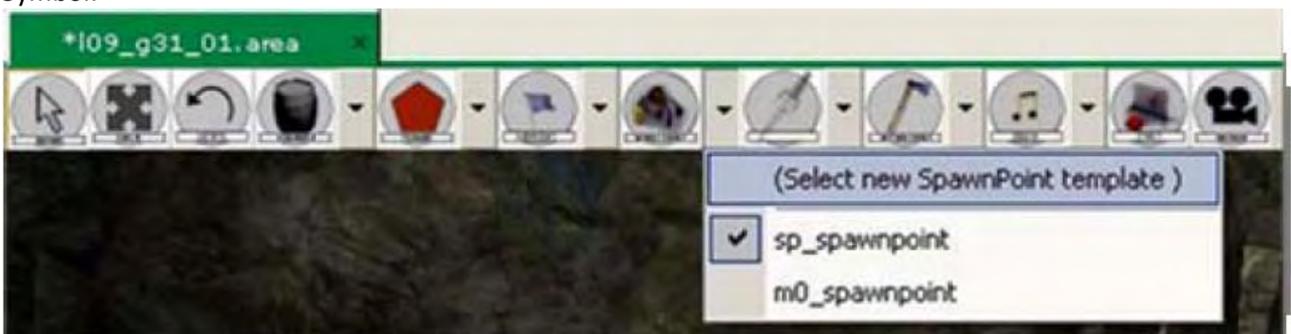
Wenn man dies getan hat, wird das Gebiet sofort heller dargestellt:



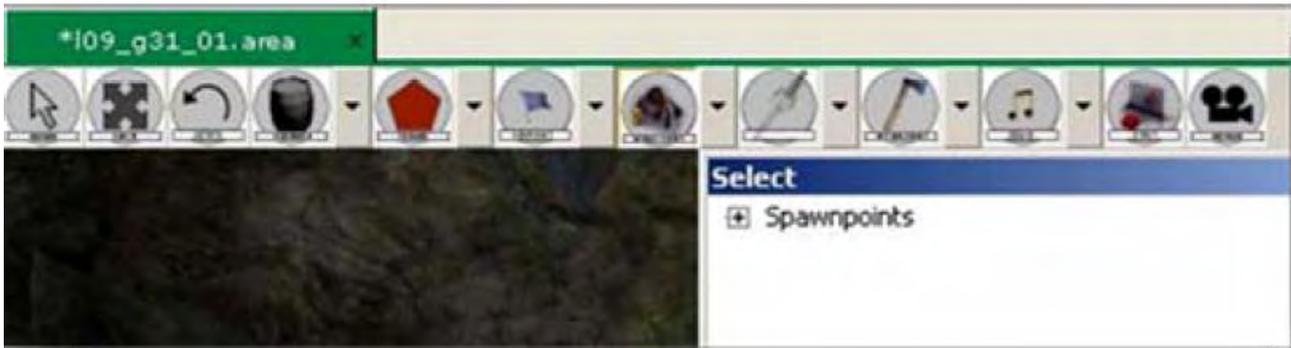
Zuerst erstellen wir einen Spawnpunkt für unseren NPC und benennen es Siegfried. Um ein neuen Spawnpunkt zu setzen, klickst du auf den siebten Icon von links gesehen. Dieser Icon befindet sich in der Werkzeugleiste über dem Editier Bereich:



Es ist möglich, einen Spawnpunkt mit einer Schablone zu erstellen. Also wähle eine aus der Liste aus. Klicke dann auf den Pfeil, der nach unten zeigt. Der Pfeil befindet sich in der Nähe von dem nächsten Spawn-Point Symbol:



Das „Select“ Fenster wird sich nun öffnen:



Öffnen den „Baum“ des Spawnpoints, indem du auf das kleine `+` Zeichen klickst und wähle dann die erforderliche Schablone von der Liste aus, indem du auf den Namen einen Doppelklick machst. Das Objekt wird sich mit dem Cursor verbinden. Jetzt Linksklicke auf den Punkt, wo du den Spawnpunkt setzen möchtest. Der Spawnpunkt wird jetzt im Gebiet hinzugefügt:



Jetzt sollten wir die Attribute für diesen Spawnpunkt einstellen. Um das zu tun, Doppelklick auf den neuen Spawnpunkt. Das Spawnpoint Fenster erscheint auf der rechten Seite des D'jinni Editors:

Spawnpoint [sp_standard]	
General	
Name	Spawnpoint
Tag	sp_standard
Comment	
Creature Uses Orientation	False
Spawning parameters	
Appear placeable template	
Spawn Categories	0 strings defined
Options	(Can appear here, Can di
TemplateResRef	sp_spawnpoint
Position	
X	1539.941040
Y	1494.355103
Z	7.500000

Die Attribute des neuen Spawnpoints sind nun in der linken Spalte zu finden. Diese sind in bestimmten Bereichen unterteilt. (General, Spawning Parameters und Position) Damit ist es nicht zu unübersichtlich wird, kannst du die Bereiche minimieren. Dafür musst du auf das kleine '-' Zeichen klicken:

General	
Name	Spawnpoint
Tag	sp_standard
Comment	
Creature Uses Orientation	False

Um Die zugeklappte Section wieder zu öffnen, Klicke auf das kleine '+' Zeichen neben dem Section-Name:

General	
Spawning parameters	
Position	

Die Werte der Attribute sind in der rechten Spalte zu finden. Die Attribute werden hier erklärt:

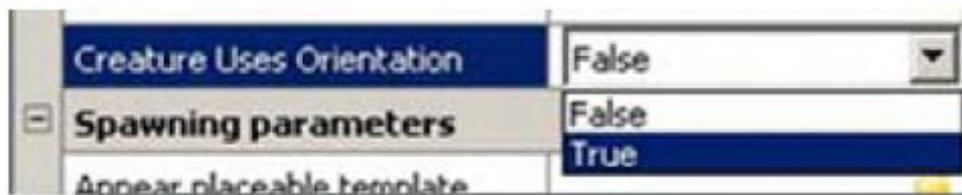
General Section:

- **Name.** Enthält den Namen des Spawnpunkts. Für gewöhnlich heißen alle anderen Spawnpoints in diesem Gebiet genau gleich.
- **Tag.** Enthält den identification tag des Spawnpoints. Alle anderen Spawnpoints haben den gleichen Tag.
- **Comment.** Hier kannst du eigene Kommentare hineinschreiben. Dies wird im Spiel nicht verwendet, es ist nur ein Informationsfeld. Du kannst schreiben was immer du möchtest. Klicke auf die rechte Spalte neben dem The Comment-Attribut. Dieses Mal wollen wir "Siegfried's spawn point" schreiben:



Drücke dann Enter um es zu bestätigen.

- **Creature Uses Orientation.** Bestimmt, ob sich der Charakter so hinstellt, dass er zum Pfeil schaut. Merke dir, dass der Spawnpoint als Icon ein Pfeil hat. Setze diesem Attribut auf „True“ und die Kreatur wird am Spawnpoint erscheinen. Wir setzen nun dem Attribut von Siegfried auf „True“. Wenn du diese Einstellung ändern möchtest, musst du auf die rechte Spalte klicken. Gegenüberliegend vom „ Creature Uses Orientation attribute“:



Die Werte, die du diesen Attributen zuteilen kannst, sind:

- *False*- Der Charakter wird zufällig irgendwo hingestellt.
- *True* - Der Charakter stellt sich in die Richtung vom Pfeil hin.

Spawning Parameters Section:

- **Appear placeable template.** Verantwortlich für das, wie der Spawnpunkt im Spiel schaut. Ein Spawnpunkt hat normalerweise sein eigenes Item, das während des Spiels unsichtbar ist. Jedoch könnte ein Baum zum Beispiel auch als ein Spawn-Punkt dienen. Um den Item eines Spawn-Punktes zu ändern, musst du auf das kleine gelbe Icon klicken, welches du bei der rechten Spalte gegenüber vom Placeable template attribute finden kannst:

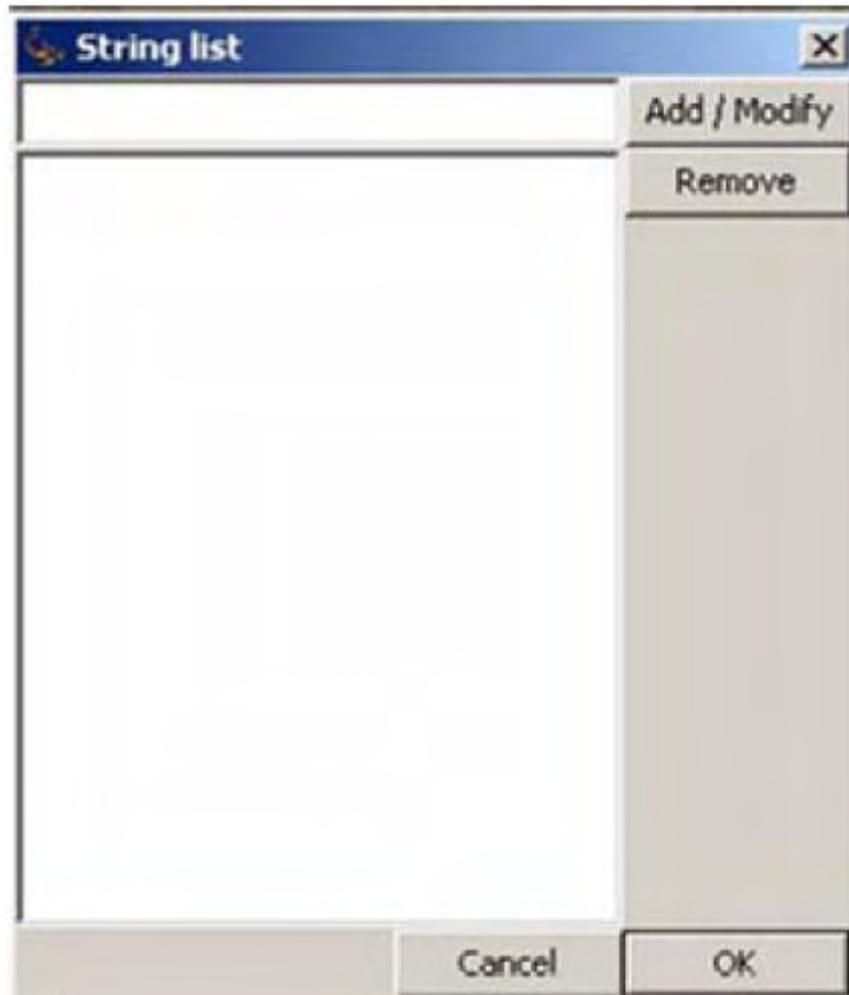


Das öffnet das Standardwindows-Fenster welches verwendet wird, um Dateien zu laden. Wähle das Item aus, welches du im (Dateien im .Utp-Format) laden kannst und öffne es mit einem Klick. Das Standard-spawn-Punkt-Item wird sich in das neue Item automatisch verändern.

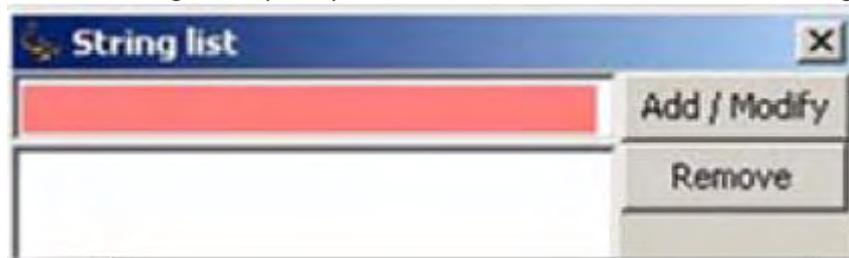
- **Spawn Categories.** Dies ist eins der wichtigsten Attribute für den Spawn-Punkt. Hier können wir einen Tag zuweisen, der dem NPC erklärt, wo es gespaunt werden muss. Um den Spawnpoint einen Tag zuweisen zu können, klickst du auf das kleine blaue Icon, welches man auf der rechten Spalte gegenüber vom Spawn Categories attribute findet:



Eine „String-List“ Fenster wird sich öffnen:



Erinnerst du dich an die Spalte mit dem Namen Spawn-Categories NPC/ Ein Bürger im Spawn-System? Wir fügten die Spawn-Kategorien zu dieser Spalte hinzu. Wenn wir einen Charakter haben wollen der im angegebenen Spawn-Punkt spawnen soll, spezifizieren wir die Punkte des Tag für beide. Die Spawn-Kategorie und der Spawn-Punkt sollten das gleiche sein. Wir teilen jetzt den Tag „spawn_siegfried“ für Siegfried dem NPC zu. Teile jetzt denselben Tag dem Spawnpunkt zu. Klick auf das obere Feld der String list Fenster:



Schreibe „spawn_siegfried“ und klicke auf Add/Modify:



Dieser Text wird im unteren Feld erscheinen:



Klicke auf die Unterseite im Stringlist-Fenster und drücke OK. Wenn du ein vorhandenes Tag modifizieren möchtest, klickst du auf das kleine blaue Icon zu finden auf der rechten spalte gegenüber vom Spawn Categories attribute. Wenn sich das Stringlist-Fenster öffnet, klicke dann auf das Tag, das du ändern möchtest:



Dieser Tag wird in einem kleinen Fenster mit einem Stringlist-Fenster geöffnet:

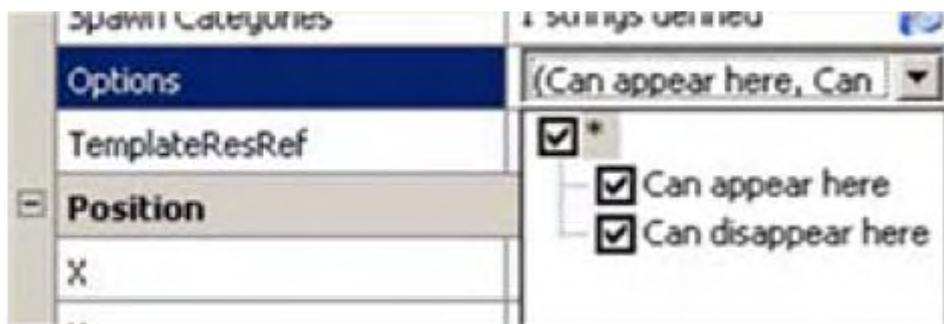


Jetzt kannst du den Tag modifizieren klicke dann auf Add /Modify möchtest du einen Tag entfernen ist der Spawn in der Stringlist zu finden und klicke dann auf „Remove“:

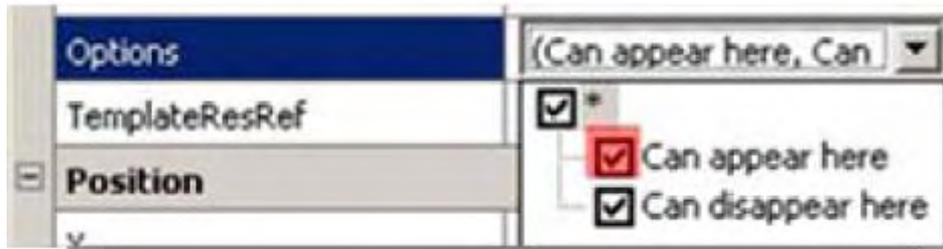


In unserem Fall jedoch, wollen wir nicht den Tag löschen. Wir wollen, dass Siegfried im Spawnpoint erzeugt wird. Erinnerst du dich, dass der Spawnpoint mehrere Tags haben kann, mehrere Charaktere erlauben, es dann an einem Spawnpoint zu spawnen.

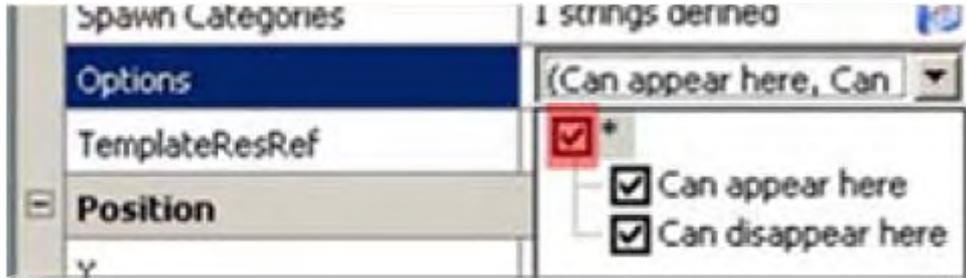
• **Options.** Bestimmt, ob eine Kreatur an einem angegebenen Spawnpoint erscheinen oder verschwinden soll. Die Default-Optionen reichen für unseren Siegfried aus. Wenn du die Einstellungen jedoch ändern willst, klickst du auf die rechte Spalte gegenüber vom dem Options Attribut. Diese folgende Liste wird jetzt erscheinen:



Ändere die Einstellungen, indem du auf die Kästchen klickst, die sich rechts neben den Optionen befinden:



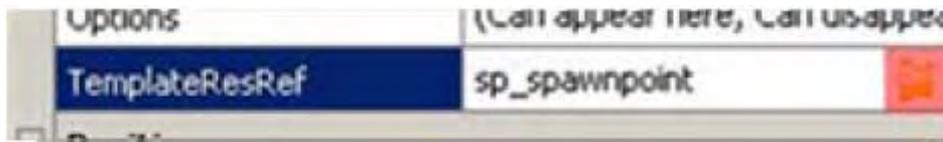
Wenn du auf eine nicht markierte Stelle klickst, wird diese dadurch markiert. Wenn du die Auswahl sofort ändern möchtest, klickst du auf das Kästchen nahe dem Sternchen:



Die Werte, die du in dieser Auswahl zuteilen kannst, sind:

- *Can appear here* -ein Charakter kann an diesem spawn-Punkt erscheinen,
- *Can disappear here* -Dein Charakter kann an diesen Spawn-Punkt verschwinden.

• **TemplateResRef.** Enthält die Spawn-Punkt-Schablone-Datei. Spawn-Punkte haben auch Schablonen. Diese Standardschablone für einen Spawn-Punkt ist eine *sp_spawnpoint.utx* Datei. Wenn du die Schablone wechseln willst, musst du auf dem gelben Icon klicken, der sich in der rechten Spalte gegenüberliegend vom *TemplateResRef* Attribut befindet:



Das öffnet das Standardwindows-Fenster welches dafür verwendet wird, um Dateien zu laden. Wähle die Schablone aus, die du willst, und klicke auf öffnen.

Position Section:

- **XPosition.** Enthält die Position der Spawnpunkte auf der X Mittellinie. Das wird automatisch jede Zeit aktualisiert, das sich Position der Spawnpunkte sich im Bereich ändernd. Du kannst die Default Settings so lassen.
- **YPosition.** Enthält die Position der Spawnpunkte auf der Y Mittellinie. Das wird automatisch jede Zeit aktualisiert, da sich die Position der Spawnpunkte sich im Bereich ändernd. Du kannst die Default Settings so lassen.
- **ZPosition.** Enthält die Position der Spawnpunkte auf der Z Mittellinie. Das wird automatisch jede Zeit aktualisiert, da sich die Position der Spawnpunkte sich im Bereich ändernd. Du kannst die Default Settings so lassen. Wir haben einen Spawnpoint für Siegfried hinzugefügt. Jetzt müssen wir noch fünf hinzufügen, für jedes der restlichen Charaktere in unserem Abenteuer. Füge den folgenden Spawnpoint nicht weit entfernt von Siegfried hinzu. Das wird jetzt der Spawnpoint für Jethro sein. Geben ihm diese folgenden Attribute:

- **Creature Uses Orientation** – True,
- **Spawn Categories** – spawn_jethro.

Füge den folgenden Spawnpunkt etwas weiter weg ein. Das ist ein Spawnpunkt für den Professor. Geb ihm folgende Attribute:

- **Creature Uses Orientation** – True,

- **Spawn Categories** – spawn_professor.

Füge in der Nähe vom Spawn Point vom Professor die nächsten 2 hinzu. Gib den folgenden Spawn Point folgende Attribute:

- **Creature Uses Orientation** – True,

- **Spawn Categories** – spawn_bandit1.

Für den zweiten

- **Creature Uses Orientation** – True,

- **Spawn Categories** – spawn_bandit2.

Es gibt einen weiteren Spawnpunkt für die Wölfe. Platziere ihn etwas weiter weg von den anderen. Gib ihm folgende Attribute:

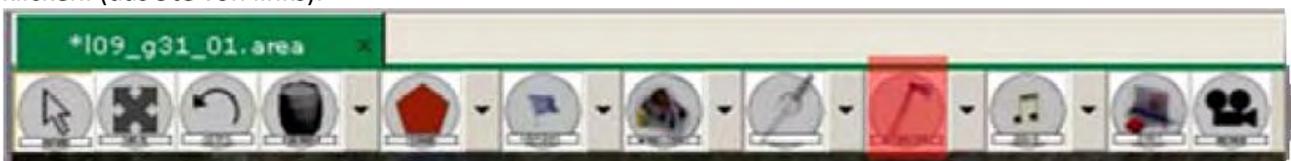
- **Creature Uses Orientation** – False,

- **Spawn Categories** – spawn_wolf.

Unser Gebiet hat nun genug Platz, damit 4 Charaktere erscheinen können.

ACTIONS POINTS

In diesem Kapitel geht es um Aktionspoints. Zuerst sollte ich jedoch erklären, was Aktionspoints sind. Charaktere bewegen sich zu diesen Punkten, nachdem sie gespawnt wurden. An diesem Aktionspoint tut er das, was du ihm zugeteilt hast. Aktionen beinhalten verschiedene Animationen, die ein Charakter ausführen wird an einem bestimmten Tag zu einer bestimmten Zeit. Lass uns Aktionspoints für Siegfried erstellen. Wie du dich sicherlich erinnern kannst, haben wir Siegfried 5 Aktionen zugeteilt, während wir ihn erstellt haben. Wie auch immer, wir müssen ihm nicht mehr 5 Aktionen zuweisen. Wir werden ihm nämlich nur 3 Aktionen geben. Um Aktionspoints zuweisen zu können, musst du auf das „axe“ (Axt) Icon in der Toolbar klicken. (das 9te von links):



Wie mit den Spawnpoints kannst du auch die Aktionspoint Schablonen nehmen. Klicke auf den nach unten zeigenden Pfeil, neben dem Aktionspoint Symbol:



Klicke in dem Menü auf „Select new Actionpoint template“:



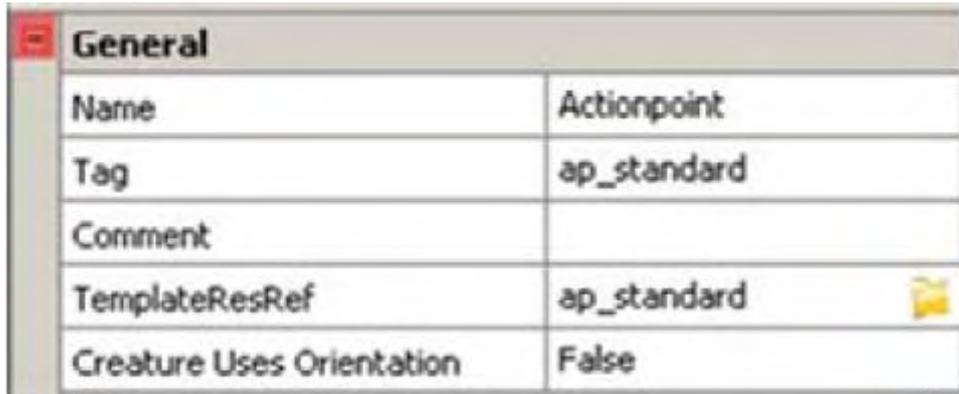
Öffnen wir den „Tree“ (Baum) für die Aktionspoint, indem auf du auf das „+“ Zeichen klickst. Klicke dann mit einem Doppelklick auf den benötigten Aktionspoint. Ein Objekt, welches den Aktionspoint repräsentiert, wird sich an den Cursor „Dranhängen“. Nun klicke mit einem Linksklicke dahin, wo du den Aktionspoint haben möchtest. Lass uns diesen neben den Spawnpoint von Siegfried setzen. Der Spawnpoint wird zum Gebiet hinzugefügt:



Jetzt müssen wir die Attribute für diesen Aktionspoint einstellen. Dafür musst du einen Doppelklick auf den Aktionspoint machen. Das Fenster vom Aktionspoint wird auf der rechten Seite des Editors erscheinen:

Actionpoint [ap_standard]	
General	
Name	Actionpoint
Tag	ap_standard
Comment	
TemplateResRef	ap_standard
Creature Uses Orientation	False
Action parameters	
Actions list	0 strings defined
Action placeable	
Region name	default
One Liner Dialog	
Dialog Category	0 strings defined
Waypoint Walk	
Waypoint Prefix	
Run	False
Position	
X	1539.981201
Y	1490.494141
Z	7.500000
Scripts	
On Used	

Dieses Fenster besteht aus 5 Sectionen, (General, Action Parameters, Waypoint Walk, Position und Scripts). Um nun die Sectionen zu schließen, klicke auf das kleine „-“ Zeichen:



General	
Name	Actionpoint
Tag	ap_standard
Comment	
TemplateResRef	ap_standard 
Creature Uses Orientation	False

Um ein zusammengeklapptes Fenster wieder ausklappen zu können, klicke auf das „+“ Zeichen:



General	
+ Action parameters	
+ Waypoint Walk	
+ Position	
+ Scripts	

Die Werte der Attribute findest du in der rechten Spalte. Einige von ihnen haben den gleichen „Task“, wie die Attribute für den Spawnpoint. Die Attribute werden hier beschrieben:

General Section:

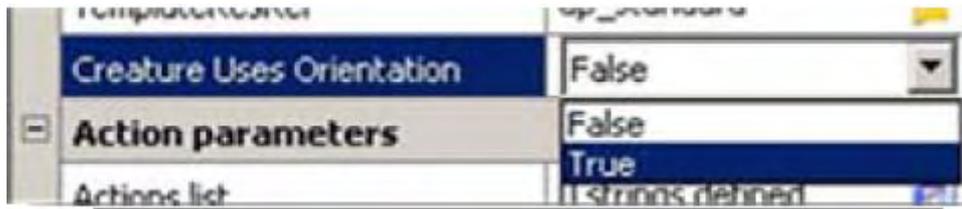
- **Name.** Beinhaltet den Namen von dem Aktionspunkt. So heißen auch alle anderen Aktionspoints in diesem Gebiet.
- **Tag.** Tag enthält den Identifikations Tag für den Aktionspunkt. Das ist normalerweise der gleiche Name wie für alle Aktionspoints in diesem Bereich. Tags sind für gewöhnlich gescrpted.
- **Comment.** Enthält einen Bereich, wo du Kommentare schreiben kannst. Wird nicht vom Spiel benutzt, dient dir nur als Information. Du kannst schreiben was immer du willst.
- **TemplateResRef.** Enthält die Schablonen-Datei für den Aktionspunkt. Aktionspunkte haben ihre eigenen Schablonen. Die Standard-Schablone ist eine ap_standard.uta Datei. Wenn du die Schablone wechseln willst, musst du auf das kleine gelbe Icon in der rechten Spalte klicken, welches sich gegenüberliegend dem TemplateResRef Attribut befindet:



TemplateResRef	ap_standard 
----------------	---

Dies öffnet das Windows-Ladefenster und dort kannst du dir eine Schablone aussuchen und anschließend klickst du auf Open.

- **Creature Uses Orientation.** Bestimmt, ob ein Charakter eine Aktion durchführt, nachdem er auf den Aktionspunkt schaut. Beachte, dass ein Aktionspunkt ein Pfeil als Icon hat, genauso wie ein Spawnpoint. Wenn du dieses Attribut auf True setzt, führt der Charakter die Aktion aus, nachdem er auf den Aktionspunkt schaut. Wir setzen dieses Attribut für Siegfried auf True. Wenn du die Einstellung ändern möchtest, klickst du in der rechten Spalte auf das Feld, gegenüberliegend vom Creature UsesOrientation Attribut:



Die Werte, die du den Attributen hinzufügen kannst, sind:

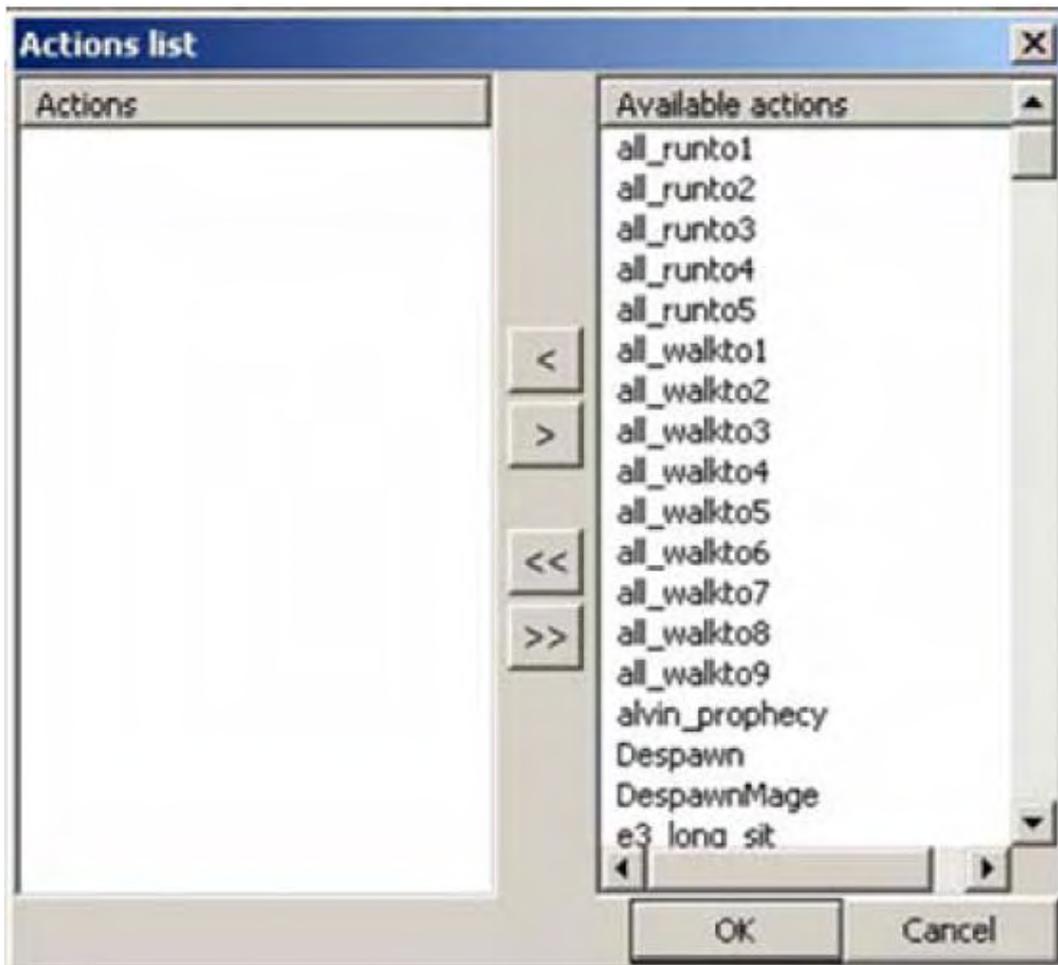
- *False* – Der Charakter wird eine Aktion durchführen, nachdem er auf eine zufällig gesetzte Anweisung trifft.
- *True* – Der Charakter wird eine Aktion durchführen, nachdem er auf die Anweisungen des Actionpunktes trifft.

Action Parameters Section:

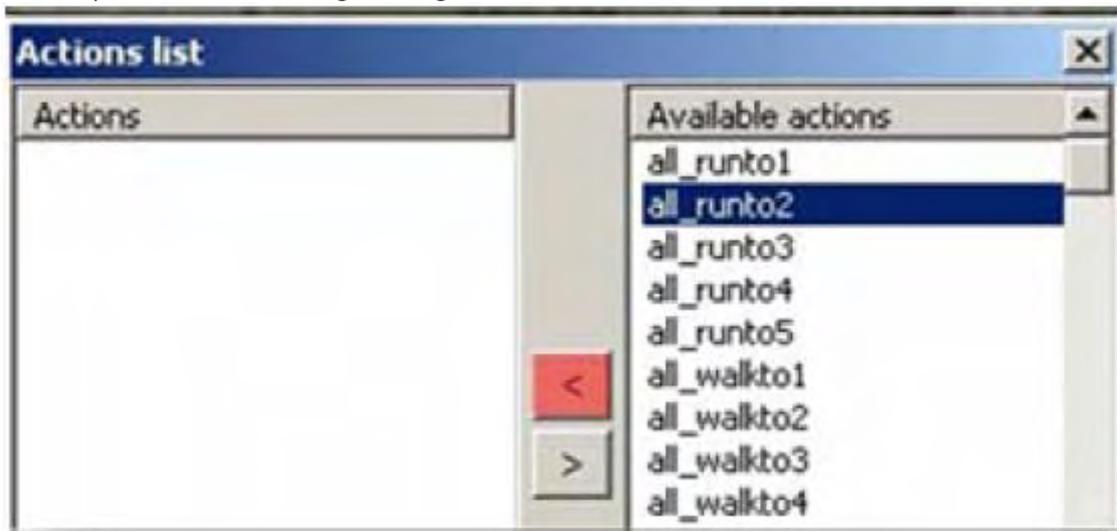
• **Action list.** Enthält eine Liste von Aktionen, die ein besonderer Charakter an einem besonderen Aktionspunkt durchführen wird. Berücksichtige, dass wenn du Einstellungen in dem Attribut vornehmen willst, dass der Charakter im Spawn Set identische, ihm aufgetragene Aufgaben haben muss. Zur Erinnerung: Wir gaben Siegfried 5 Aktionen: *w_h_pray*, *w_h_bored1*, *all_walkto1*, *all_walkto2* und *w_h_alarmed* zugewiesen. Lass uns ihm nun eine Aufgabe geben, die er am Aktionspunkt ausführen wird. Um dies zu tun, musst du auf das kleine blaue Icon klicken, welches sich in der rechten Spalte befindet. Gegenüberliegend dem Action list Attribute:



Die Actions Liste wird sich öffnen:



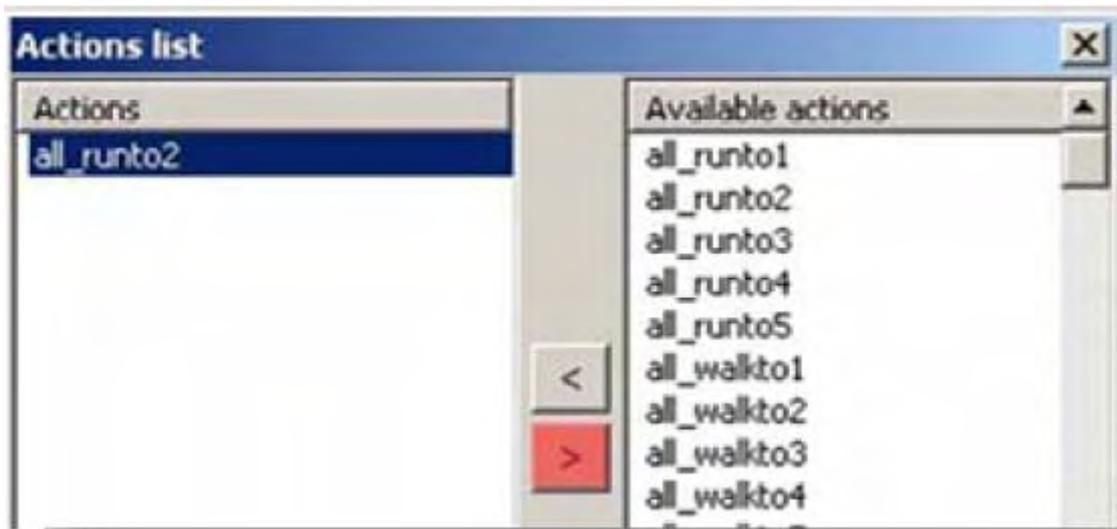
Die Aktionen, die z.B einem Aktionspunkt zugewiesen werden, sind im Actions Feld. Die Liste mit allen Aktionen findest du im Actions Panel. Die Liste von allen Aktionen findest du im Available Actions Feld. Um einem Aktionspunkt eine neue Aufgabe zu geben, musst im Available Actions Feld auf „<“ klicken:



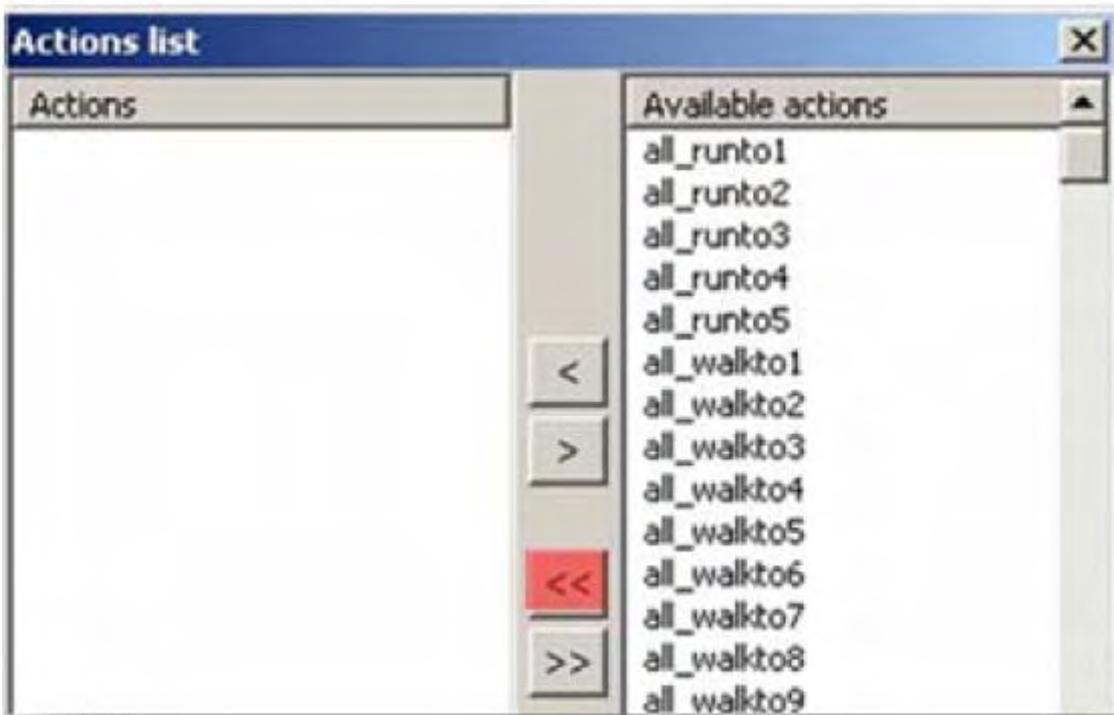
Die Aktion wird automatisch im Aktionen Feld erscheinen. Bedeutet also, dass diese Aufgabe dem Aktionspunkt zugewiesen wird:



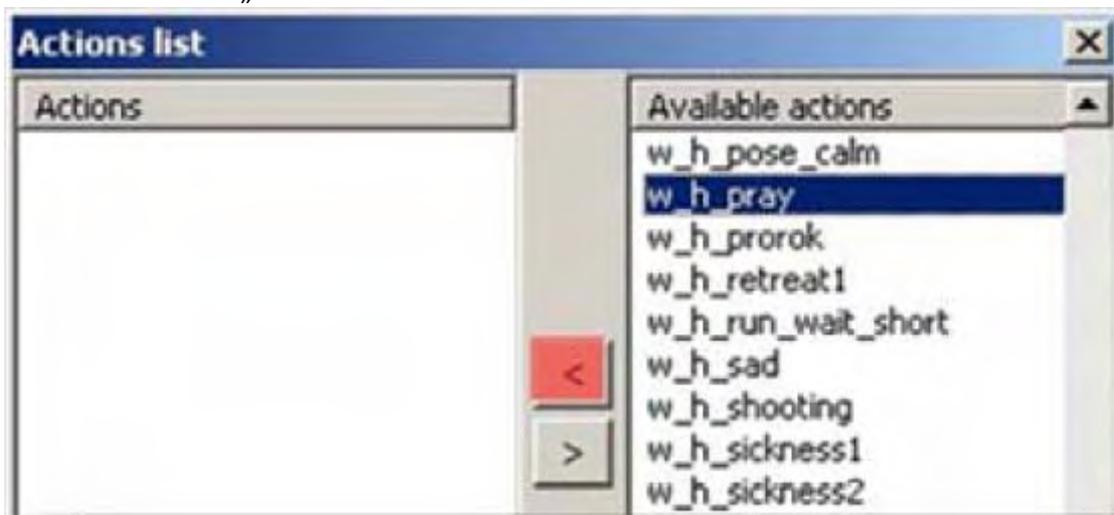
Wenn du eine Aktion entfernen willst, musst du diese markieren und auf „>“ klicken:



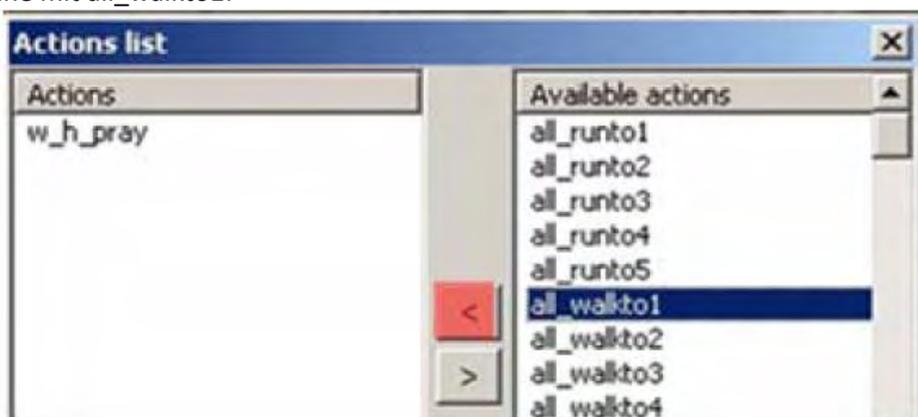
Du kannst auch mehrere Aktionen gleichzeitig auswählen. Dies machst du, indem du auf CONTROL (Zeiger nach unten) oder SHIFT mehrmals klickst und es gedrückt hältst. Um alle Aktionen gleichzeitig rüberschicken zu können, musst du auf „<<“ klicken:



Wenn du auf „>>“ klickst, werden alle Aktionen rückgängig gemacht. Lass uns 2 Aktionen für Siegfried dem Aktionspunkt geben. Wir geben ihm w_h_pray und all_walkto1. Suche w_h_pray im Available Actions Fach, markiere es und klicke auf „<“:



Mach das Gleiche mit all_walkto1:



Du kannst auch eine Aktion dem Punkt zuweisen, indem du auf ihren Namen doppelklickst. Da die Liste sehr lang ist, kannst du die Suche vereinfachen, indem du eine Aktion markierst und dann den Anfangsbuchstaben des Namens der Aktion drückst. Dadurch springt die Liste zu der Aktion, die mit dem entsprechenden Namen anfängt.

- **Action placeable.** Verantwortlich dafür, wie der "Actionpoint" im Spiel aussieht. Ein Actionpoint hat normalerweise ein eigenes Objekt, das während des Spiels unsichtbar ist. Aber andere Objekte, zum Beispiel ein Bett, kann auch als Actionpoint fungieren. Um das Objekt eines Actionpoints zu verändern, klicke auf das kleine gelbe Icon rechts neben dem „Action placeable“ Attribut:



Das öffnet den Standard-Dialog zum Laden von Dateien. Wähle das Objekt aus, das du suchst. Das Standard-Objekt wird dann mit dem, das du ausgewählt hast, ausgetauscht.

- **Region Name:** Eins der wichtigsten Attribute der Action Points. Hier weisen wir einen Tag zu, dass dem NPC mitteilt, welche Aktion er an diesem speziellen Action Point ausführen muss. Erinnerst du dich an den "Region List" Dialog, wo wir die Spawn Regions hinzugefügt haben? Wenn wir einem bestimmten Charakter eine Aktion an einem bestimmten Action Point zuweisen willst, müssen sowohl die Spawn Region als auch der Action Point den gleichen Namen haben. Wir haben Siegfried den Tag `action_siegfried` zugewiesen. Nun muss der gleiche Tag dem Action Point zugewiesen werden. Um das zu tun, klicke auf die Spalte rechts neben "Region Name" und tippe `action_siegfried` ein:

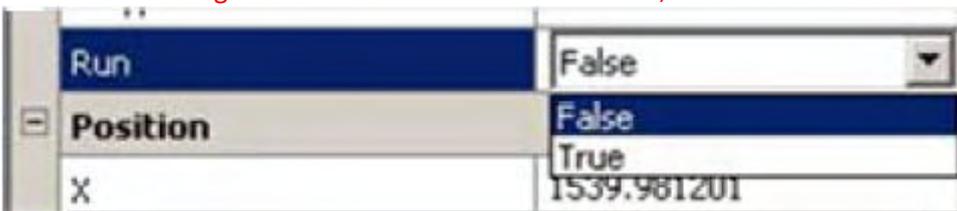


Drücke dann Enter.

- **One LinerDialog.**
- **Dialog Category.**

Waypoint Walk Section:

- **Waypoint Prefix.**
- **Run.** Bestimmt ob ein Charakter zu einem Action Point rennt oder geht. Wähle für Siegfried "False", da er von einem zum anderen Punkt gehen soll. Um das Attribut zu setzen, klicke auf das Feld neben "Run":



Wähle „False“ aus der Liste.

Die Werte die es annehmen kann, sind:

-False: Der Charakter geht.

-True: Der Charakter rennt.

Position Section:

- **XPosition:** Beinhaltet die Position des Action Points auf der X-Achse. Sie ändert sich automatisch, wenn du den Punkt verschiebst.
- **YPosition:** Beinhaltet die Position des Action Points auf der Y-Achse. Sie ändert sich automatisch, wenn du den Punkt verschiebst.
- **ZPosition:** Beinhaltet die Position des Action Points auf der Z-Achse. Sie ändert sich automatisch, wenn du den Punkt verschiebst.

Script Section:

- **On Used:** enthält das Script, dass der Charakter ausführt, wenn er mit dem Action Point interagiert. Um ein Script zuzuweisen, klicke auf das gelbe Icon rechts neben dem "On Used" Attribut:



Das öffnet das Fenster zum Laden von Dateien. Wähle das gesuchte Script aus und klicke auf Open.

So erstellst du Action Points. Wir müssen noch weitere Action Points für die anderen Charaktere erstellen und Siegfried braucht noch 2 in der Nähe seines ersten Action Points:



Setze die folgenden Attribute wie beim Ersten:

Creature Uses Orientation: True
Action List: all_walkto1, w_h_bored1
Region Name: action_siegfried

Mach dasselbe beim Zweiten:

Creature Uses Orientation: True
Action List: all_walkto2, w_h_alarmed
Region Name: action_siegfried

Nun kommen wir zu den Punkten von Jethro. Setze sie in kurzer Entfernung zu Jethro's Spawn Point.



Setze die folgenden Attribute für den Ersten:

Creature Uses Orientation: True

Action List: w_h_fired1

Region Name: action_jethro

Mach das selbe beim Zweiten:

Creature Uses Orientation: True

Action List: all_walkto1

Region Name: action_jethro

Nun ist es Zeit für den Professor. Füge einen Action Point in der Nähe des Spawn Points des Professors ein:



Gebe ihm die folgenden Attribute:

Creature Uses Orientation: True

Action List: w_h_guard

Region Name: action_professor

Zum Schluss die Wachen des Professors. Setze einen Action Point in die Nähe jedes Spawn Points.



Setze die folgenden Aktionen beim ersten Action Point:

Creature Uses Orientation: True

Action List: w_h_guard2

Region Name: action_bandit1

Mach das gleiche für den Zweiten:

Creature Uses Orientation: True

Action List: w_h_guard

Region Name: action_bandit2

Wir brauchen den Actionpoints der Wölfe keine Aktionen zuweisen.

CONVERSATION

Einleitung

In diesem Kapitel lernen wir, wie man Konversationen erstellt. Es müssen mindestens 2 Charaktere an einer Konversation teilnehmen. Man kann Konversationen in 3 Typen aufteilen:

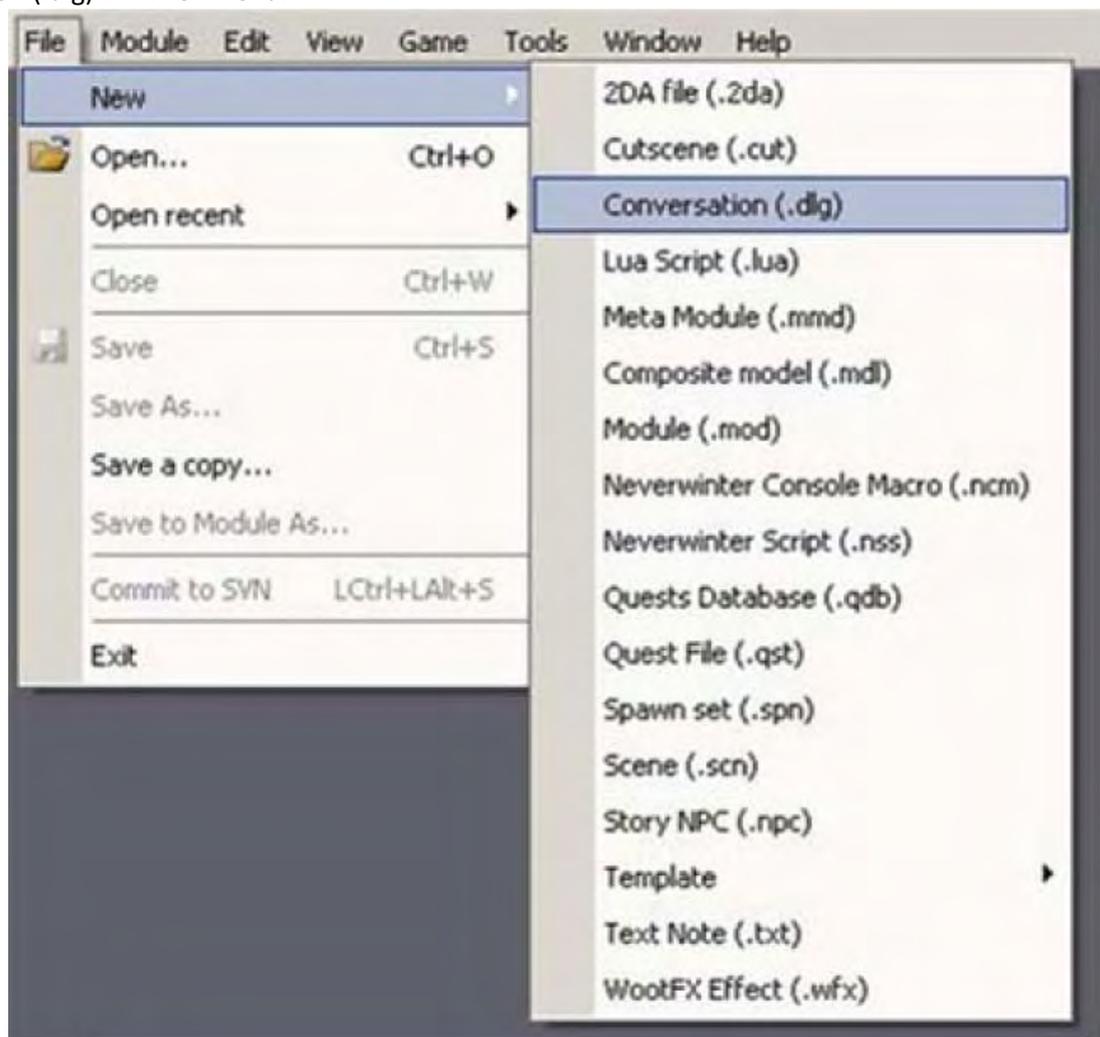
- Quest: Konversationen, die durch eine Quest mit der Handlung des Spiels verknüpft sind.
- Dialog: Konversation zwischen 2 oder mehr Kreaturen, die keinen Einfluss auf die Handlung hat.
- Cutscene: Konversationen, die während einer Cutscene stattfinden.

Cutscenes werden in einem anderen Kapitel erklärt.

Man kann zusätzlich noch ein sogenanntes "Gameplay" einer Konversation hinzufügen. "Gameplays" sind einfach die Möglichkeit Poker zu spielen, mit einem Charakter zu saufen oder mit ihm zu boxen.

Quest Conversation

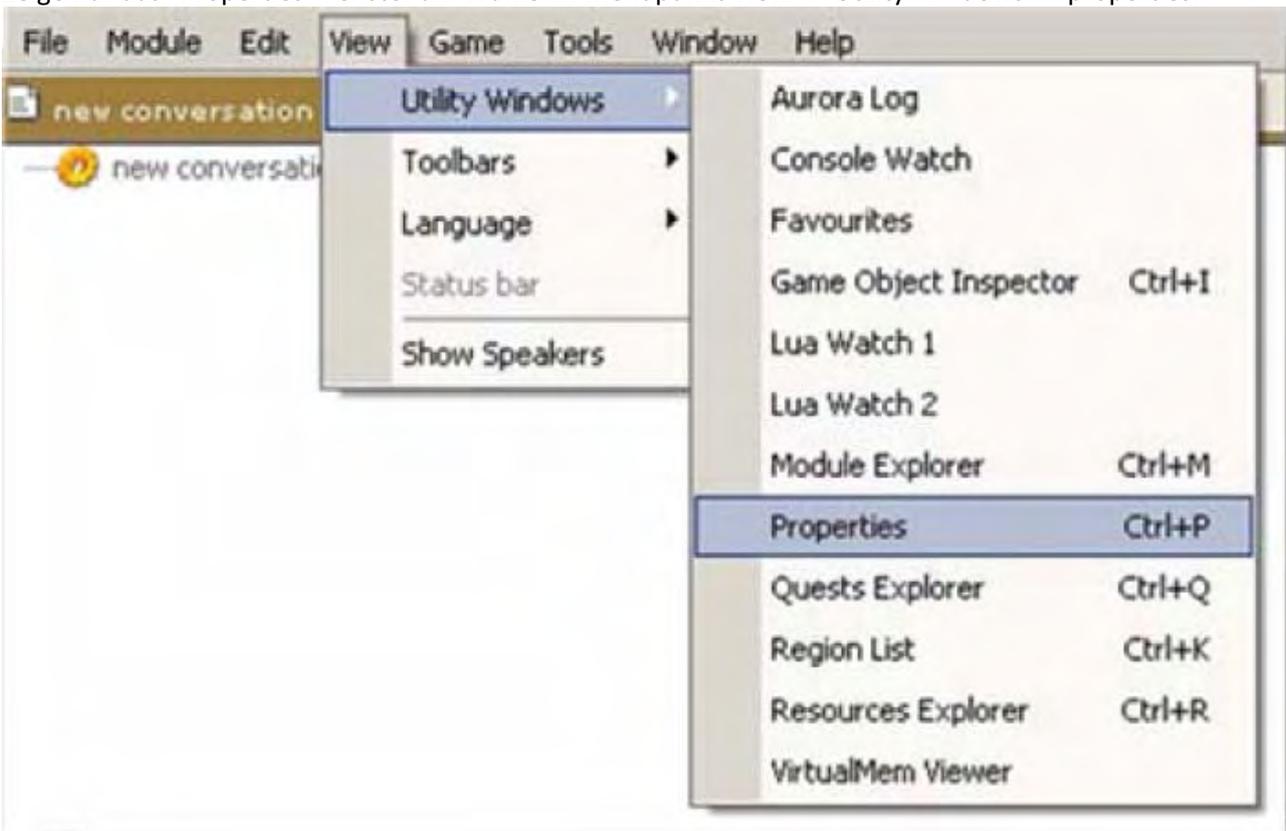
Um einen Einstieg zu finden, werden wir eine Conversation für Siegfried erstellen. Ehrlich gesagt sind es mehrere Conversation, die je nach Situation, angezeigt werden. Siegfried wird ein Quest für Geralt haben, in dem er den Anführer der Banditen töten soll. Um ein Conversation File zu erstellen, wähle *New => Conversation (.dlg)* im "File" Menü:



Das "New Conversation" Fenster wird sich öffnen:



Zeige nur das "Properties" Fenster an. Wähle im Menüpunkt View => Utility Windows => properties:



Das Fenster wird sich auf der rechten Seite des Bildschirms öffnen:

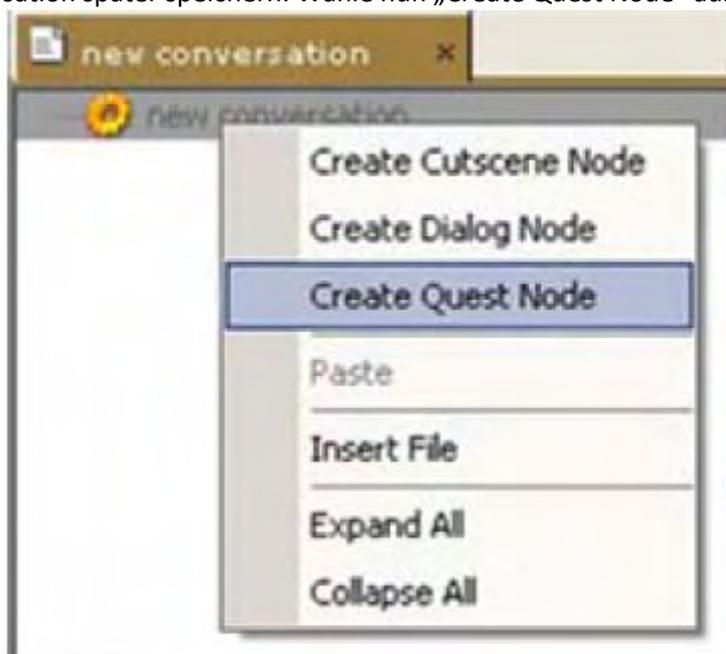


Füge nun den ersten Eintrag der Conversation hinzu. Um dies zu tun, wähle „new conversation“ und klicke mit der rechten Maustaste:

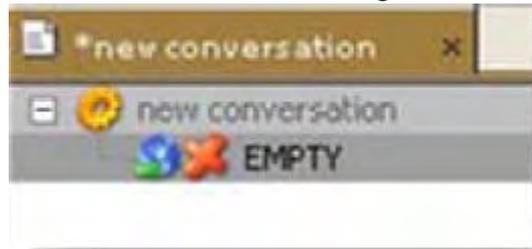


Wenn wir unsere Conversation speichern, wird der angezeigte Namen zum Beispiel „siegfried_conv“ sein, nicht mehr new conversation.

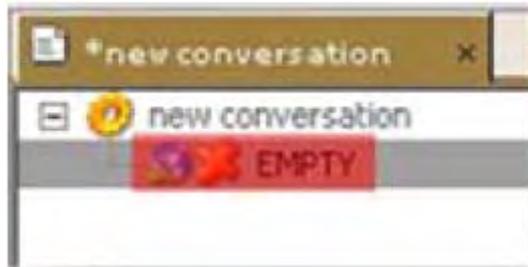
Wir werden die Conversation später speichern. Wähle nun „Create Quest Node“ aus dem Menü:



So erstellst du eine „Quest Conversation“. Der „EMPTY“ Eintrag erscheint:



Denk dran, dass jede Conversation, egal welcher Art (Quest, Dialog, oder Cutscene), von einem anderen Charakter begonnen wird. Sie wird niemals von Geralt begonnen, also dem Spieler. Darum ist der hinzugefügte Eintrag für Geralt. Um den Text hinzuzufügen, den Siegfried sagen wird, klicke doppelt auf EMPTY:



Dadurch öffnet sich ein Fenster:



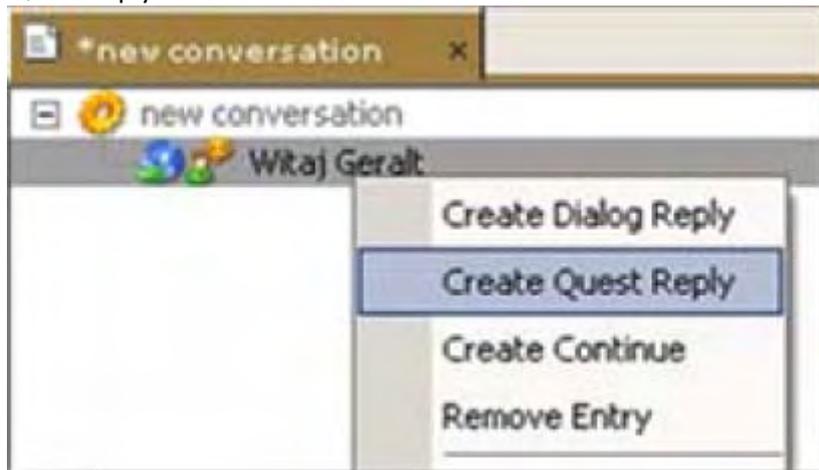
Schreibe "Hallo Geralt" und drücke ENTER. Der erste Eintrag ist fertig:



Nun muss der Spieler (Geralt) antworten. Um das zu tun, wähle den "Hallo Geralt" Eintrag aus und klicke mit der rechten Maustaste:



Wähle nun „Create Quest Reply“ aus dem Menü:



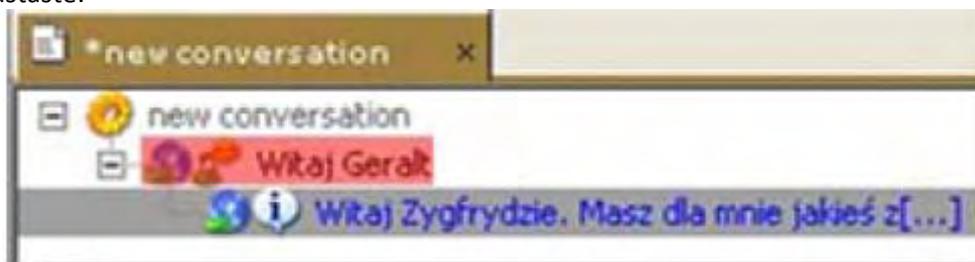
Dadurch wird ein weiterer „EMPTY“ Eintrag erstellt:



Du siehst, dass der neue Eintrag blau gekennzeichnet ist. Das heißt, dass ihn Geralt benutzen wird. Charaktere sprechen abwechselnd, zuerst spricht der eine, dann der andere Charakter. Um Geralts Antwort zu erstellen, klicke doppelt auf den neuen Eintrag, wodurch sich ein Fenster öffnet. Schreibe "Hallo Siegfried. Hast du neue Quests für mich?" hinein und drücke ENTER:



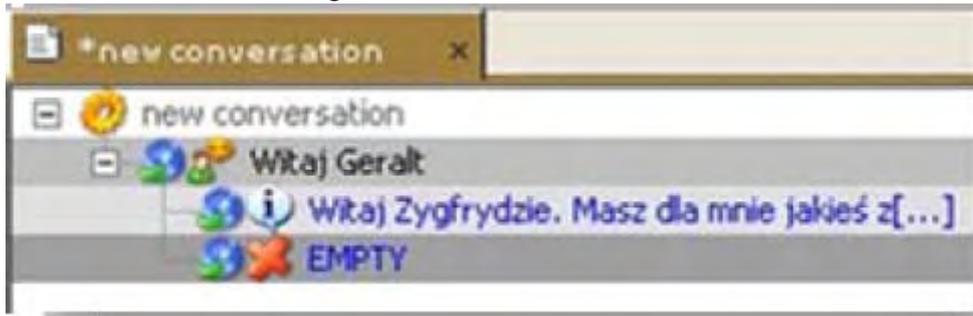
Lass uns nun einen weiteren Eintrag für Geralt hinzufügen. Wähle "Hallo Geralt" aus und klicke mit der rechten Maustaste:



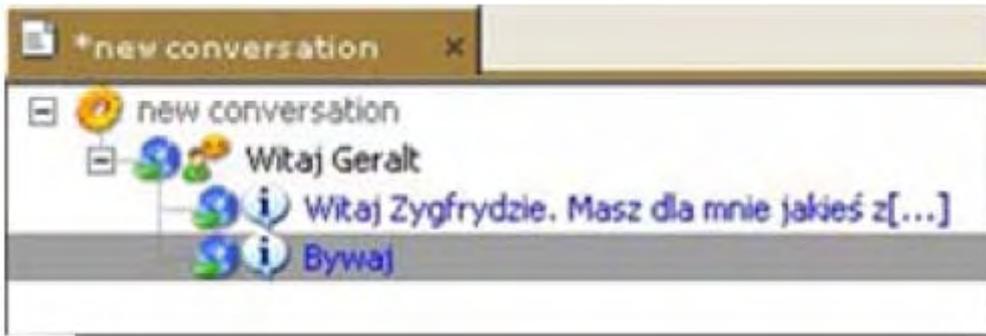
Wähle „Create Quest Reply“ aus dem Menü:



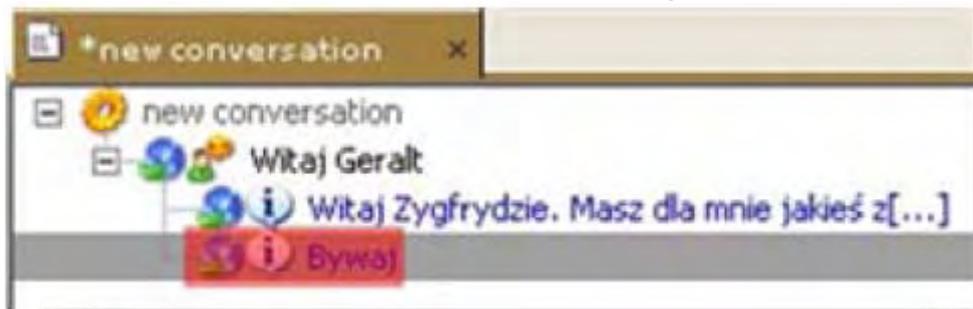
Das fügt einen weiteren „EMPTY“ Eintrag hinzu:



Klicke doppelt darauf und schreibe "Leb wohl". Drücke dann ENTER.



"Leb wohl" ist nun die zweite Möglichkeit, die der Spieler im Gespräch auswählen kann. Er kann entweder "Hallo Siegfried. Hast du neue Quests für mich?" oder "Leb wohl" sagen. Klicke nun rechts auf "Leb wohl":



Klicke nun im Eigenschaften-Fenster auf das Feld rechts neben "Conv Type". Wähle "Accompanying" aus der Liste:

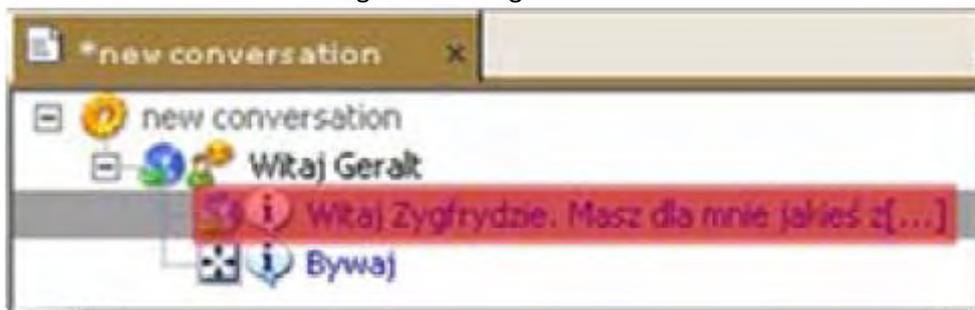
Node Text	
FinalEnglish_Short	Bywaj
Speaker	__player__
Interlocutor	
Voice File	
Voiceset tag	---None---
Comment	
Camera Shot	Auto
Action Script	
Quest Id	
Conv Type	Quest
JournalId	Dialog Quest
Node Emotions	
__player__	Accompanying Gameplay

Nun ist der "Leb wohl" Eintrag der zweite Punkt während der "Quest Conversation". Der geänderte Status ist erkennbar an dem neuen Symbol, das rechts neben dem Eintrag erscheint:

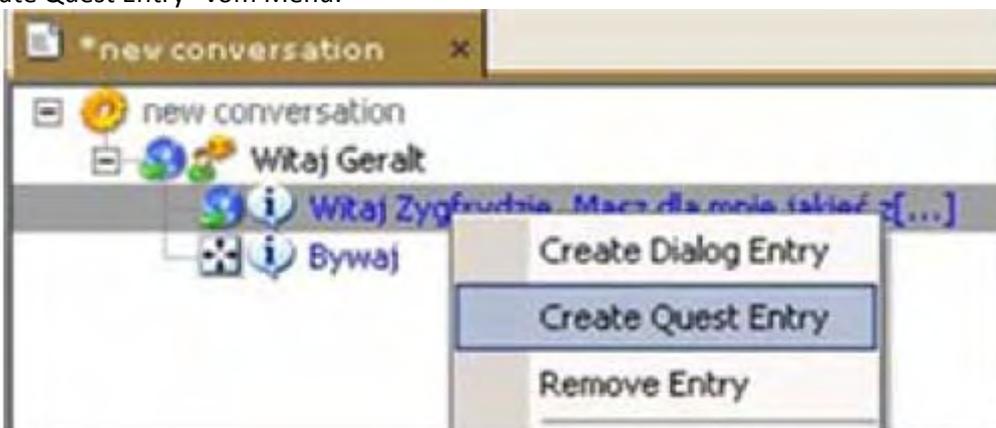


Was wir bisher getan haben, kann man folgendermaßen zusammenfassen:
Wenn die Conversation startet, sagt Siegfried "Hallo Geralt". Danach kann Geralt Siegfried nach neuen Quests fragen oder die Conversation beenden.

Nun werden wir einen neuen Eintrag hinzufügen, der es Siegfried erlaubt, die Frage nach Quests zu beantworten. Klicke dazu rechts auf denn Eintrag mit der Frage nach neuen Quests:



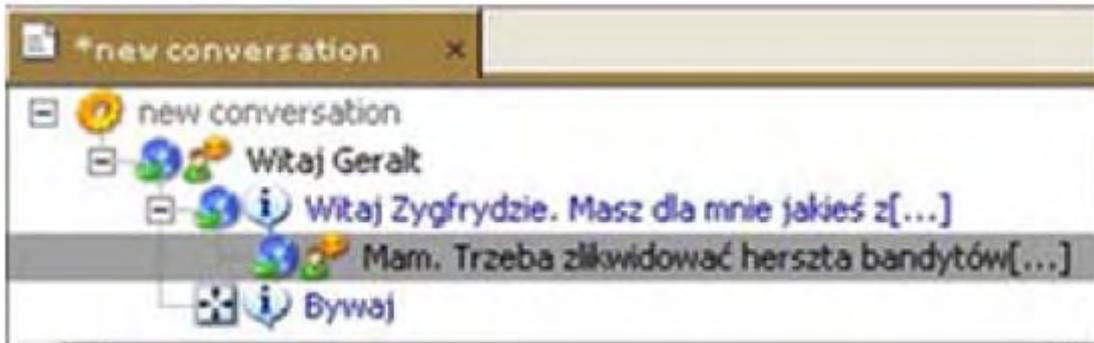
Wähle „Create Quest Entry“ vom Menü:



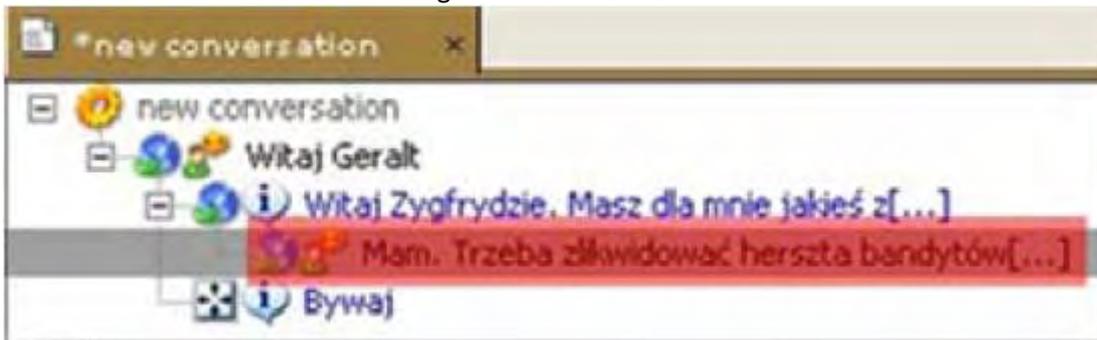
Das erzeugt einen leeren Eintrag für Siegfrieds Antwort hinzu:



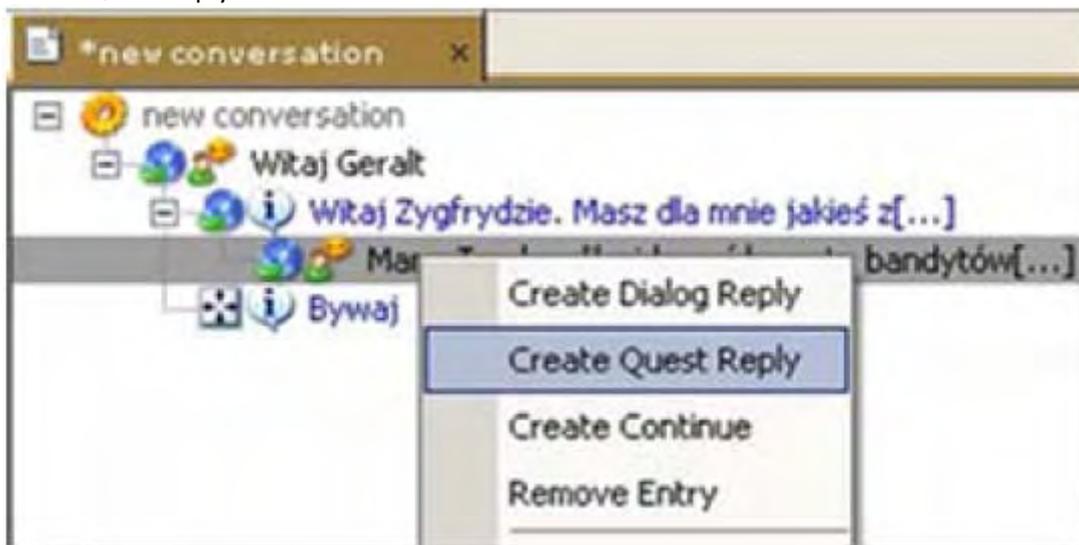
Klicke doppelt auf diesen Eintrag und füge den folgenden Text ein: "Hab ich. Du musst dich mit dem Banditenanführer befassen. Ich kann das nicht tun." Jetzt drücke Enter:



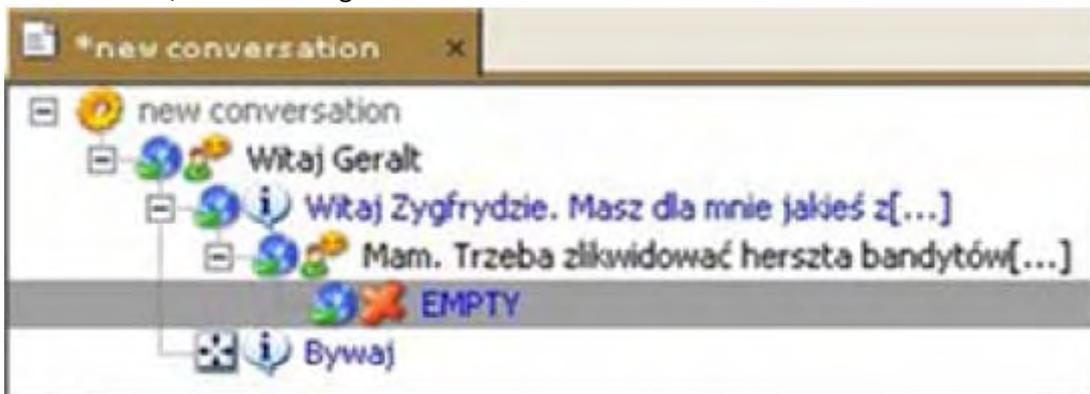
Jetzt müssen wir eine Antwort hinzufügen. Markiere den "Hab ich. Du musst dich mit dem Banditenanführer befassen. Ich kann das nicht tun." Eintrag mit einem Rechtsklick:



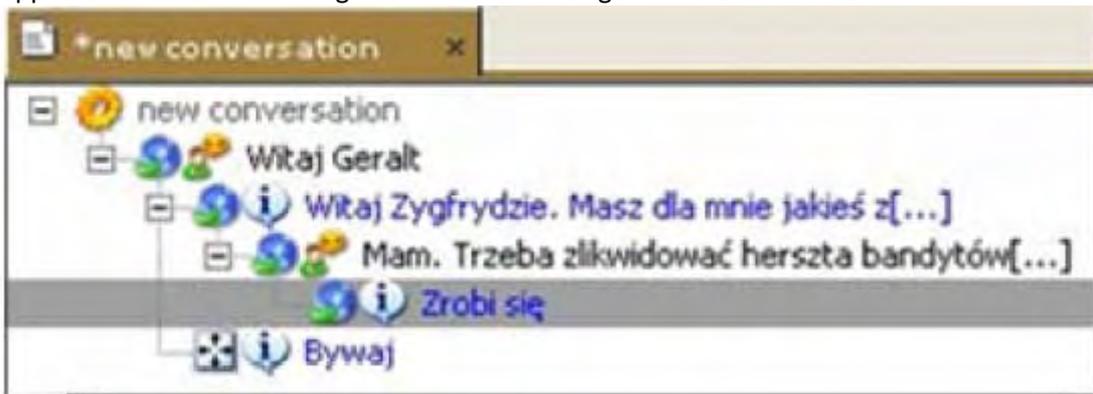
Wähle „Create Quest Reply“ aus dem Menü:



Das fügt einem neuen, leeren Eintrag hinzu:



Klicke doppelt auf den leeren Eintrag und schreibe "Erledigt". Drücke dann Enter:

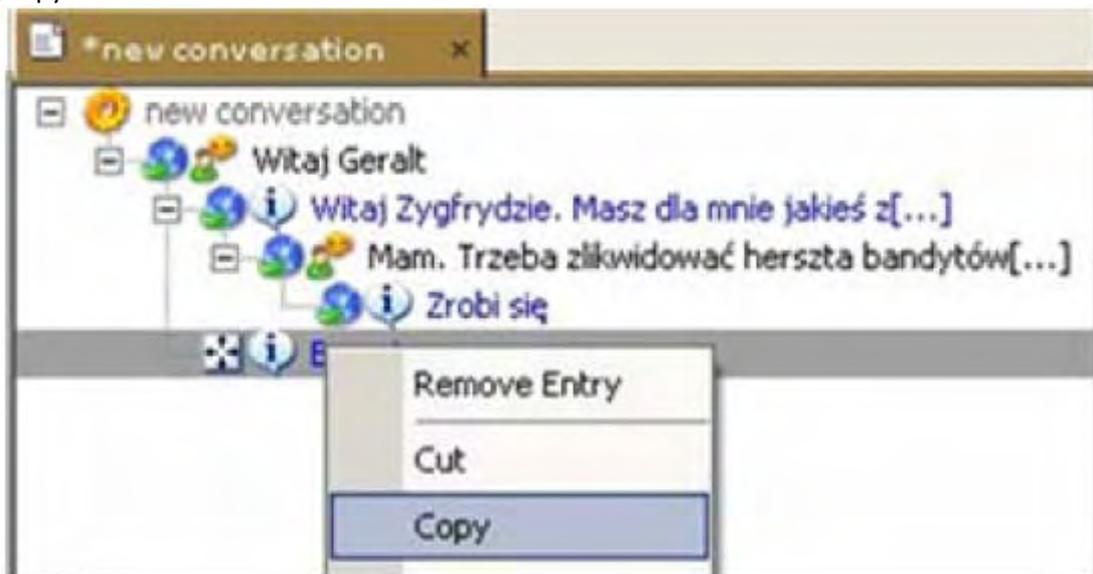


Da Geralt nicht verpflichtet ist, die Quest anzunehmen, müssen wir eine ablehnende Antwort erzeugen. Wenn Geralt, oder der Spieler, die von Siegfried angebotene Quest nicht annehmen will, kann er dann "Leb wohl" antworten.

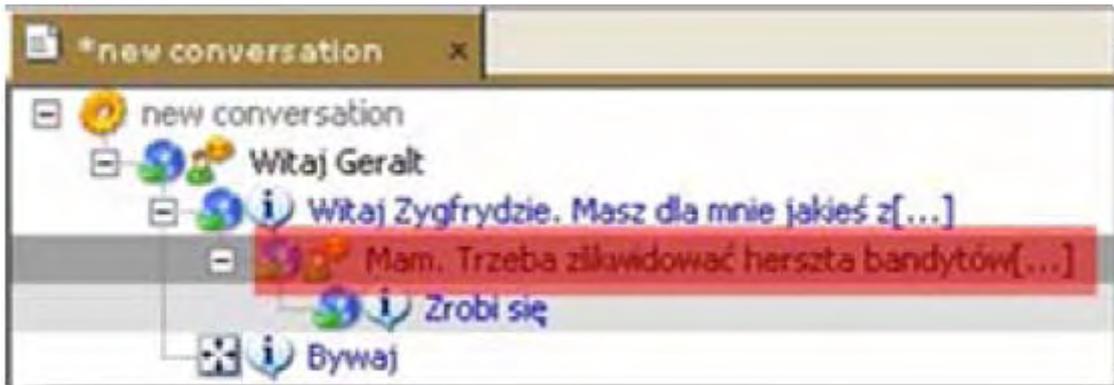
Da diese Antwort bereits besteht, müssen wir den gleichen Text nicht noch einmal schreiben. Stattdessen können wir einen Trick verwenden. Markiere den bestehenden "Leb wohl" Eintrag und drücke die rechte Maustaste:



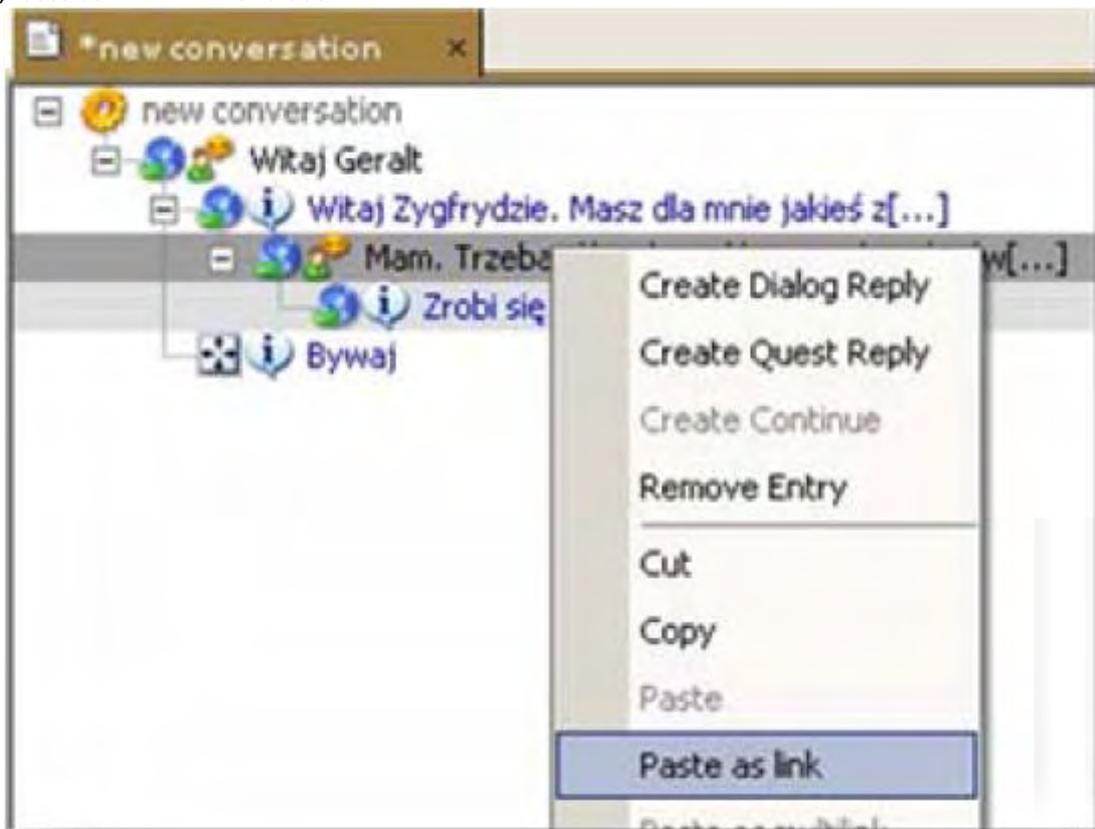
Wähle „Copy“ aus dem Menü:



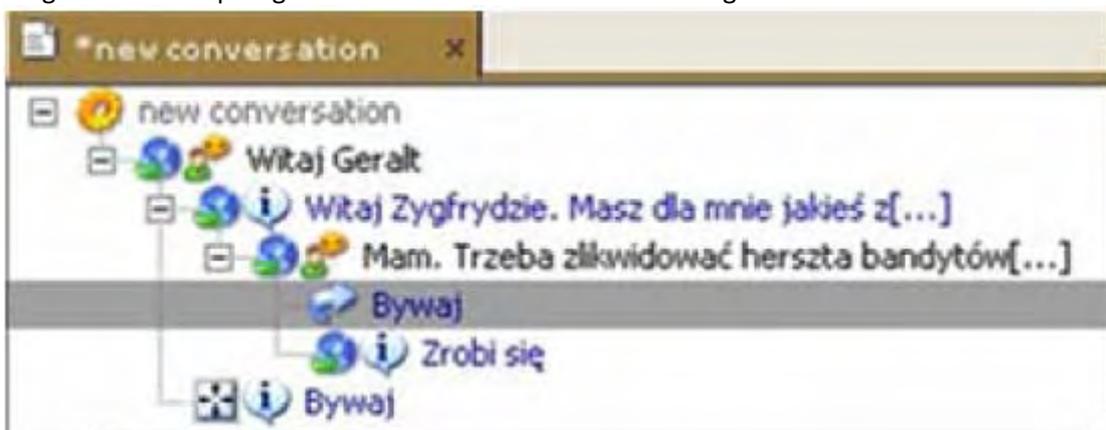
Markiere den "Hab ich. Du musst dich mit dem Banditenanführer befassen. Ich kann das nicht tun." Eintrag und drücke die rechte Maustaste:



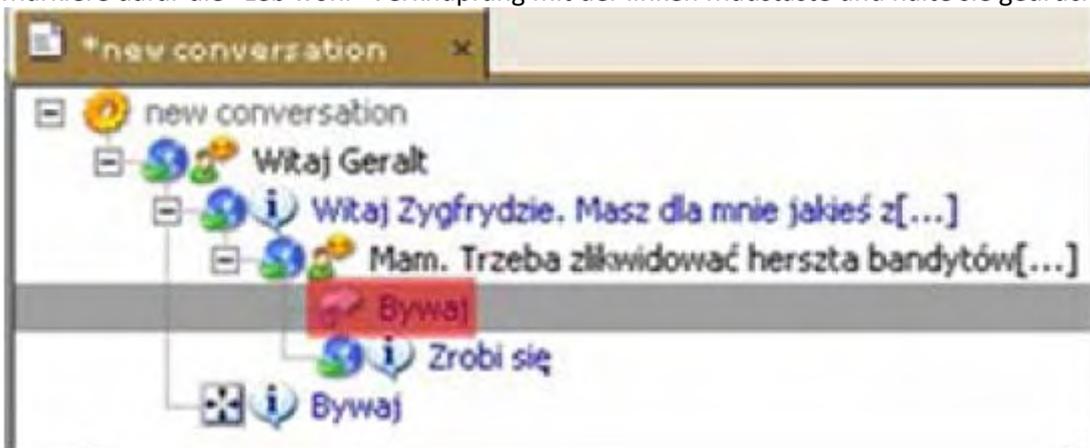
Wähle „Paste as link“ im Menü aus:



Dies erzeugt eine Verknüpfung zum bestehenden "Leb wohl" Eintrag:



Dank dieses Tricks können wir doppelte Einträge vermeiden, und der Spieler wird dann später im Spiel nur ein "Leb wohl" auf den Schirm sehen. Jetzt müssen wir die neue Verknüpfung unter den "Erledigt" Eintrag ziehen. Markiere dafür die "Leb wohl" Verknüpfung mit der linken Maustaste und halte sie gedrückt:



Ziehe es auf den "Erledigt" Eintrag und lass dann die linke Maustaste los:



Das ist eine einfache und nützliche Art die Einträge zu bewegen. Bisher haben wir folgendes getan: Am Anfang des Gespräches sagt Siegfried "Hallo Geralt". Als Entgegnung kann Geralt(der Spieler) fragen: "Hallo Siegfried, hast Du eine Quest für mich?" Oder das Gespräch einfach mit "Leb wohl" beenden.

Wenn jetzt der Spieler die Frage nach einer Quest stellt, dann antwortet Siegfried: "Hab ich. Du musst dich mit dem Banditenanführer befassen. Ich kann das nicht tun." Jetzt kann sich der Spieler entscheiden, ob er die Quest annimmt. In diesem Fall wählt er "Erledigt" oder mit "Leb wohl" ablehnt. Wurde die Quest angenommen, müssen wir ihr eine Quest Id zuordnen. Markiere dafür den "Erledigt" Eintrag:



Im Properties Fenster, klicke auf das Feld rechts neben *Quest Id* und schreibe *siegfried_task*:



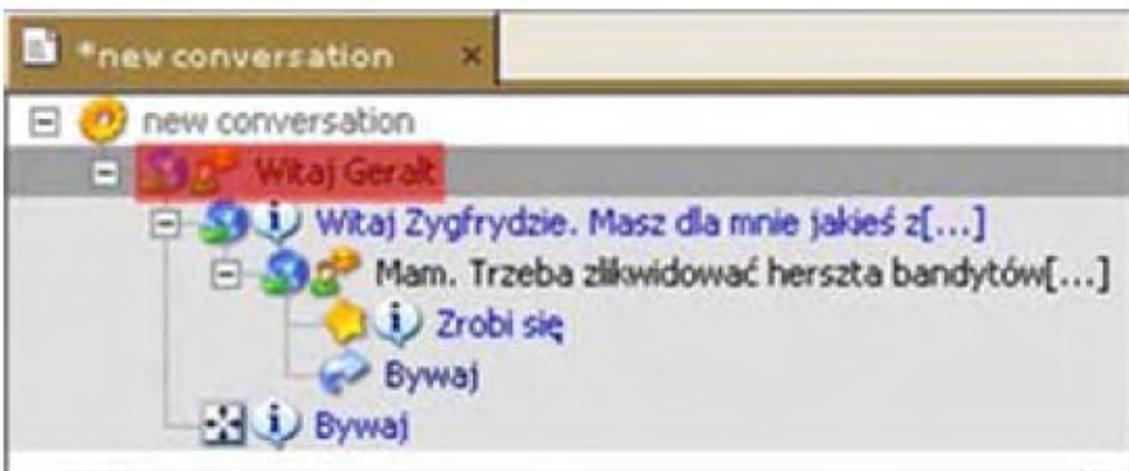
Beachte, dass sich das Symbol, vor dem "Erledigt" Eintrag verändert hat:



Wichtig: Nach einem Quest-Eintrag,(mit einer Quest id) kann der folgende Eintrag im selben Dialog nicht ebenfalls ein Quest-Eintrag sein. In unserem Fall ist das kein Problem, weil unsere Verknüpfung zum "Leb wohl" Eintrag führt. Der lediglich ein begleitender Eintrag ist.

Nachdem der Spieler Siegfrieds Quest angenommen hat, wird der vorherige Quest-Dialog nicht mehr angezeigt. Wir müssen einen neuen Dialog schaffen, in dem Siegfried Geralt fragt, ob er die Quest beendet hat. Lautet die Antwort ja, brauchen wir wieder einen Dialog, in dem Siegfried Geralt für die Erfüllung der Quest lobt.

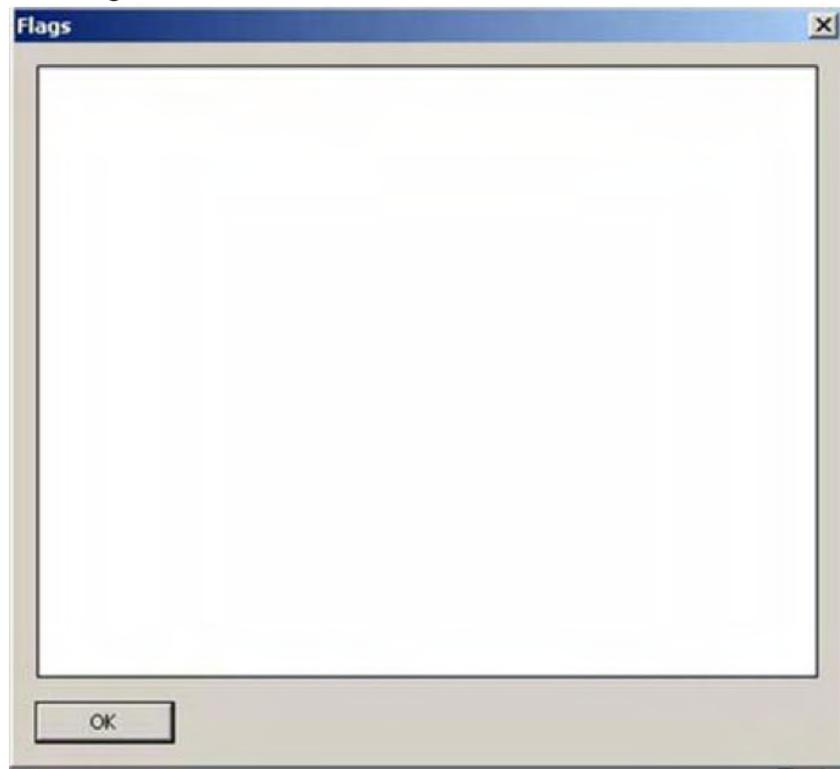
Wenn es aber drei Dialoge gibt, woher weiß dann das Spiel, welches angezeigt werden soll? Das erledigen wir mit Hilfe eines Flag Systems. Klicke auf „Hallo Geralt“:



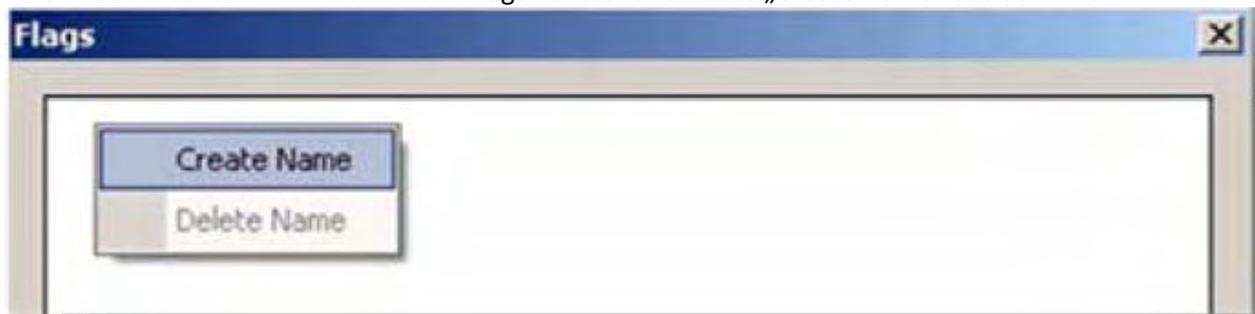
Klicke im Properties-Fenster auf das kleine gelbe Symbol im Feld rechts neben „Add Flag“:



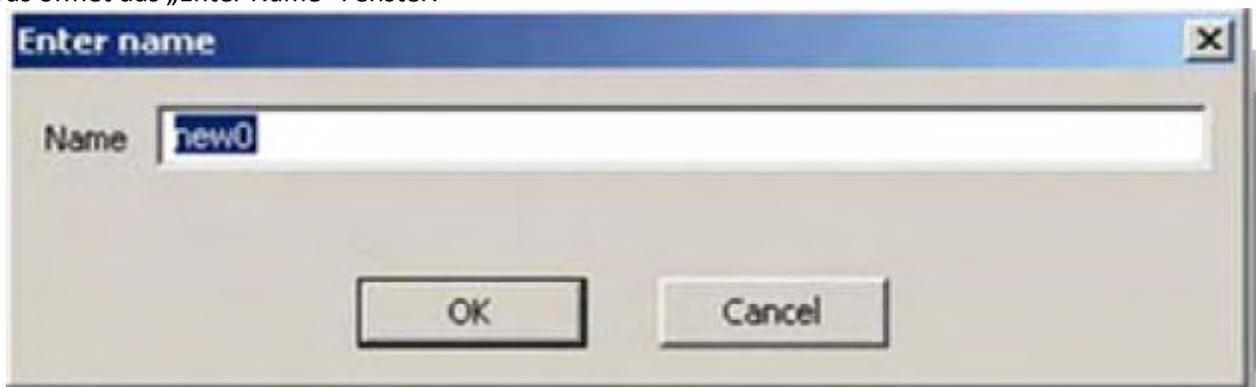
Dadurch öffnet sich das Flags Fenster:



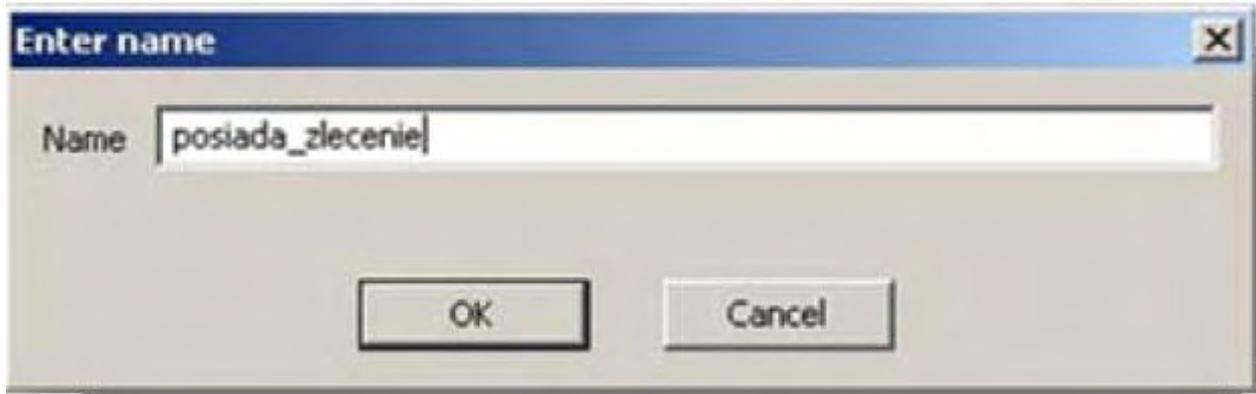
Klicke mit der rechten Maustaste in das Flags-Fenster und wähle „Create Name“ aus dem Menü:



Das öffnet das „Enter Name“ Fenster:



Schreibe „orders_received“ ins Fenster:



Jetzt klicke auf OK. Die Flag erscheint jetzt im Flag-Fenster:



Wenn du eine spezielle Flag löschen möchtest, klicke mit der rechten Maustaste darauf und wähle „Delete Name“ aus dem Menü aus:



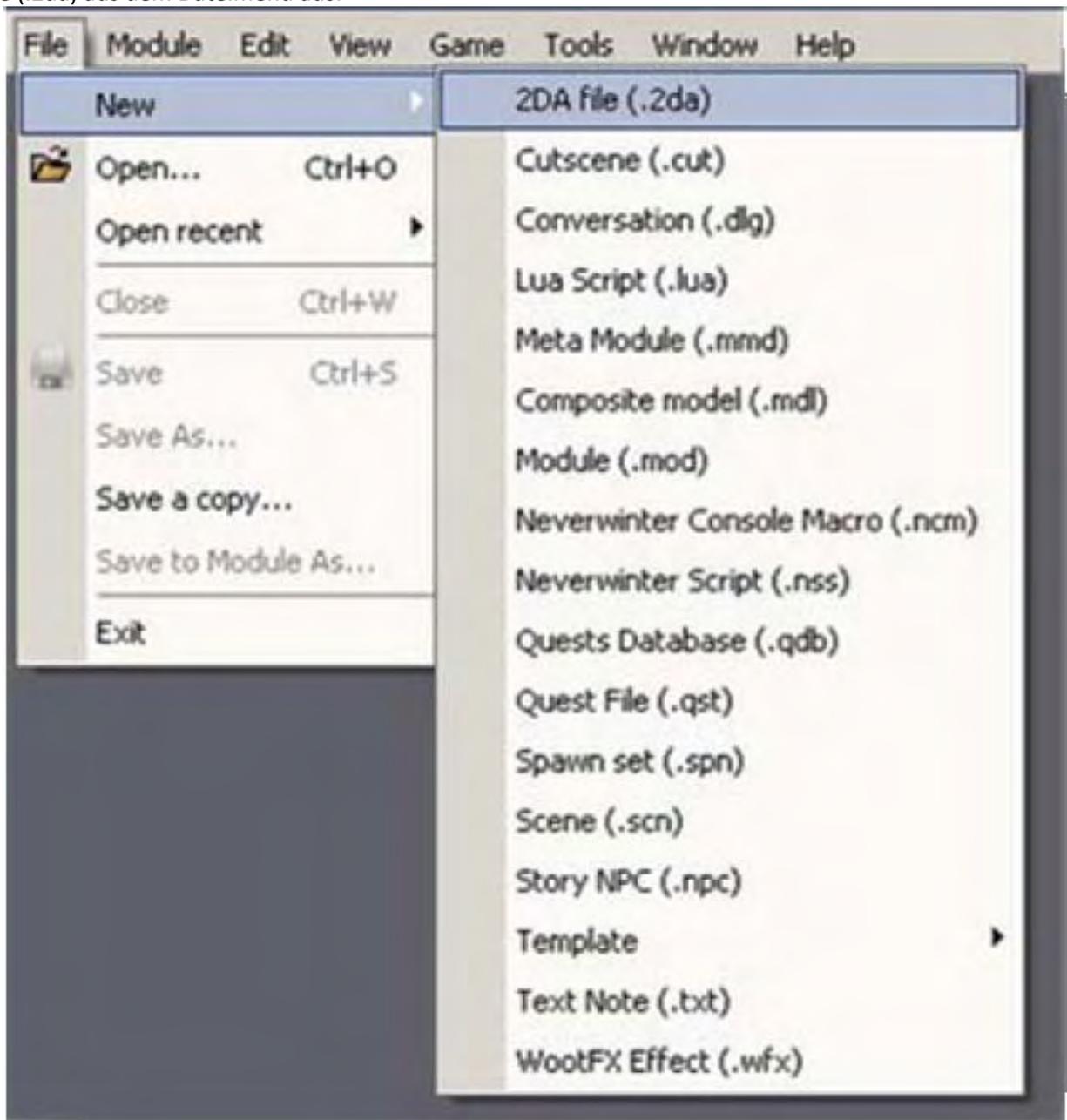
Wir wollen diese Flag aber nicht löschen, darum klicken wir nur auf Ok, im Flag-Fenster:



Wir fügen jetzt eine zweite Flag, *leader_killed* hinzu. Diese Flag wird den Wert 0(false) haben, wenn der Banditenanführer am Leben ist. Sobald Geralt(Der Spieler) den Banditenanführer tötet, ändert sich der Wert in der Flag auf 1(true).

Diese Flag muss eine "global flag" sein. Bisher haben wir "local flags" erstellt, die z.B. nur für Dialoge gelten. "global flags" sind im ganzen Spiel relevant, weshalb sie z. B. für Quests verwendet werden.

Darauf wird aber in einem späteren Kapitel eingegangen. "Global Flags" werden in der globaldlgflags.2da Datei gespeichert. Wir sollten jetzt so eine Datei erstellen. Um das zu bewerkstelligen wähle New -> 2DA File (.2da) aus dem Dateimenü aus:



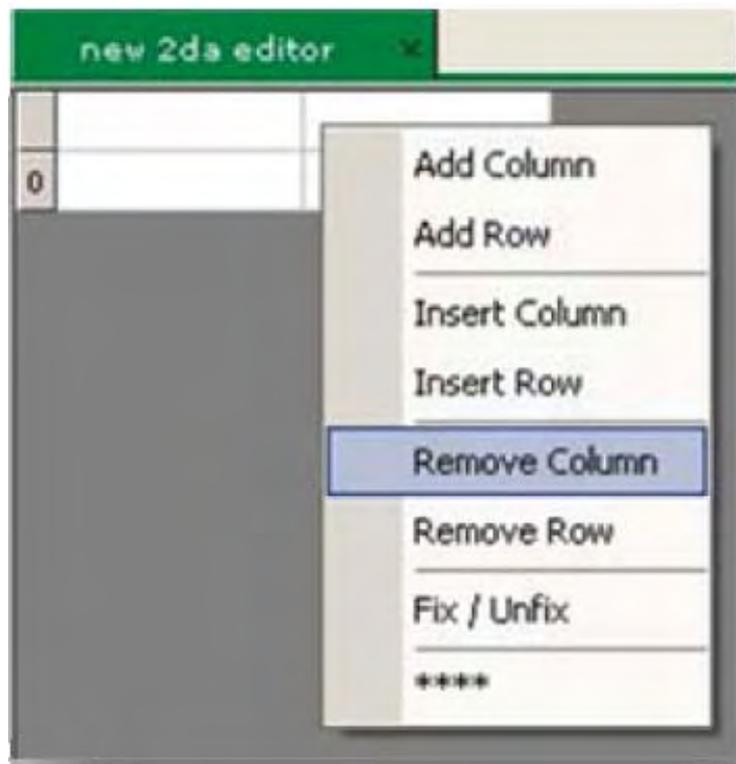
Das new 2da Editor Fenster öffnet sich:



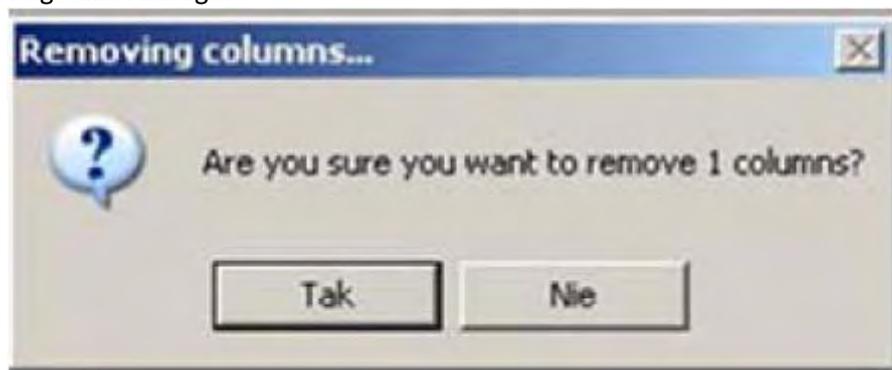
Die .2da Dateien aus The Witcher sind einfache Datenbanken, die Text in 2D Diagrammen enthalten. Eine .2da Datei kann aus einer unterschiedlichen Anzahl aus Zeilen und Spalten bestehen. Jede neue .2da Datei besteht aber zuerst nur aus 2 Zeilen und 2 Spalten. Die erste Zeile einer .2da Datei enthält einen Titel und die folgende Zeile ein Register. Die erste Spalte ist immer ein Register-Schlüssel durch den die Datenbank adressiert wird. In unserer globaldlgflags.2da Datei benötigen wir nur eine Spalte. Die andere entfernen wir mit einem Rechtsklick auf die rechte Spalte:



Wähle „Remove Column“ im Menü aus:



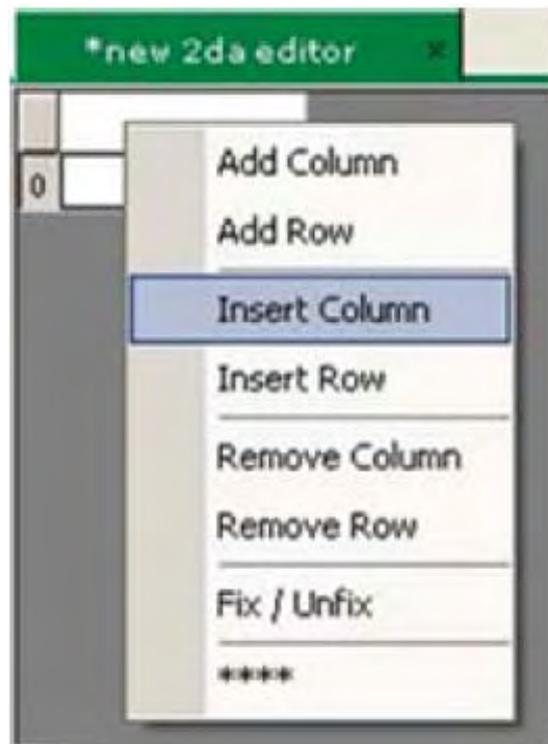
Zur Sicherheit erfolgt eine Abfrage:



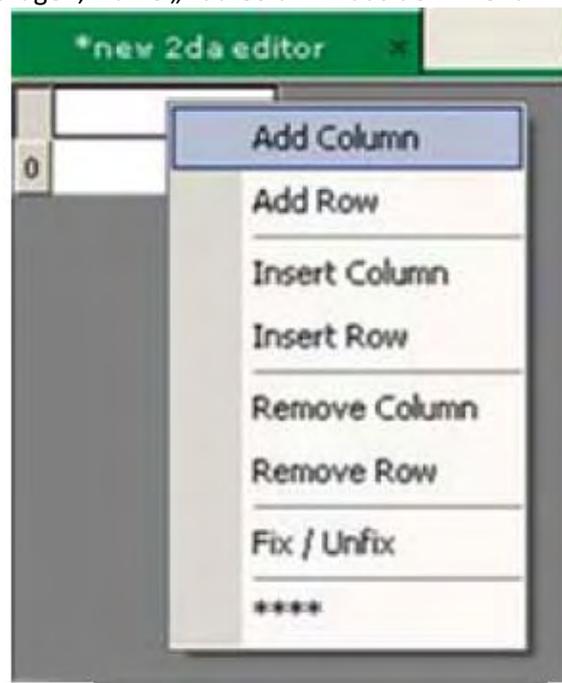
Das wird die Spalte löschen, klicke auf „Yes“:



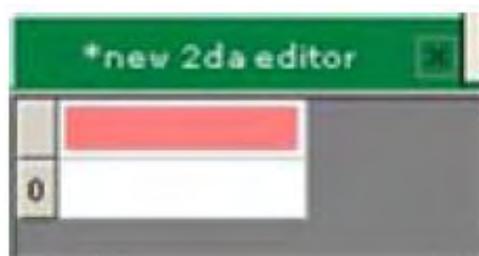
Jetzt haben wir eine Spalte entfernt. Wenn du eine neue Spalte hinzufügen möchtest, klicke mit einem Rechtsklick auf die vorhandene Spalte und wähle „Insert Column“ aus dem Menü:



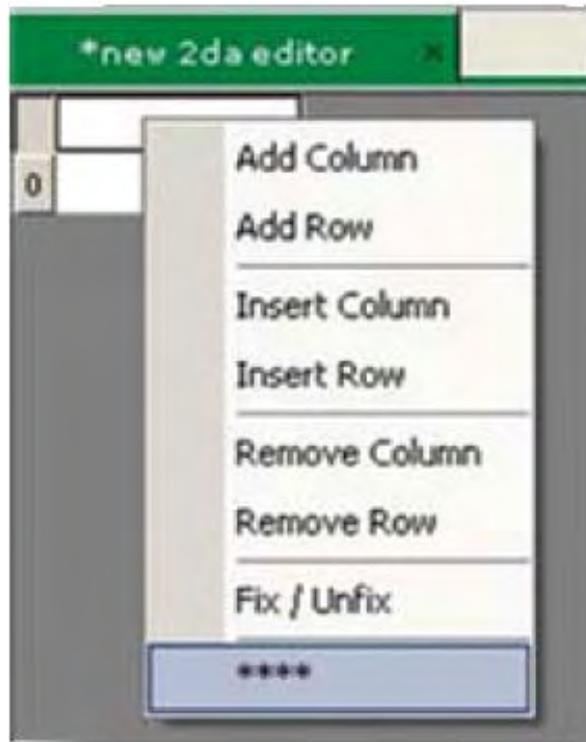
Um eine Spalte am Ende hinzufügen, wähle „Add Column“ aus dem Menü:



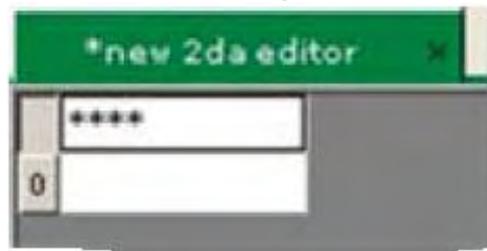
Unsere .2da Datei verfügt jetzt über eine Spalte. Im Moment brauchen wir nur eine Reihe. Allerdings lässt sich die zweite auch nicht entfernen. Beachte, dass die Reihen ab 0 gezählt werden, wodurch die eigentlichen Werte erst in der zweiten Spalte angefangen. Warum? Wie wir es vorher erwähnten kann die erste Spalte und die erste Reihe dazu dienen einen Titel zu erstellen. Das ist in unserem Fall nicht notwendig, deshalb klicke mit der rechten Maustaste auf die erste Zeile:



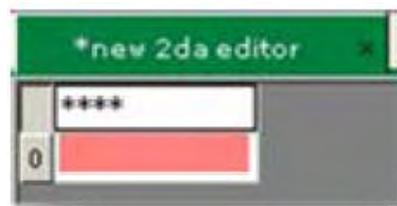
Wähle „****“ vom Menü:



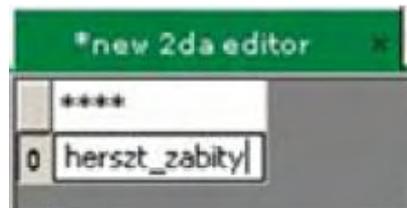
Dadurch bekommt unsere .2da Datei keinen Titel-Eintrag:



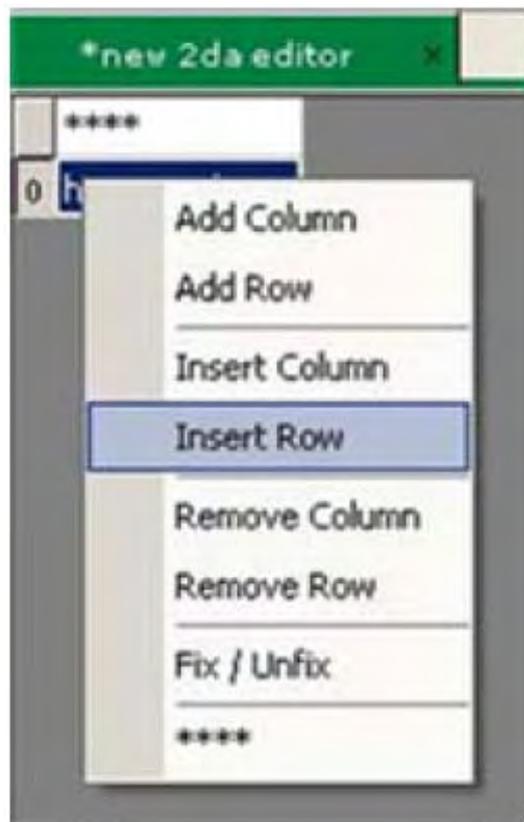
Klicke doppelt auf die zweite Zeile:



Schreibe leader_killed in das Feld:



Drücke Enter. Jetzt sind wir fertig. Wenn Du in Zukunft dieser Datei eine andere Zeile hinzufügen möchtest, klicke mit der rechten Maustaste an die Stelle, an der Du die neue Zeile einfügen möchtest und wähle „Insert Row“:



Um eine neue Zeile am Ende einzufügen, wähle „Add Row“ aus dem Menü:

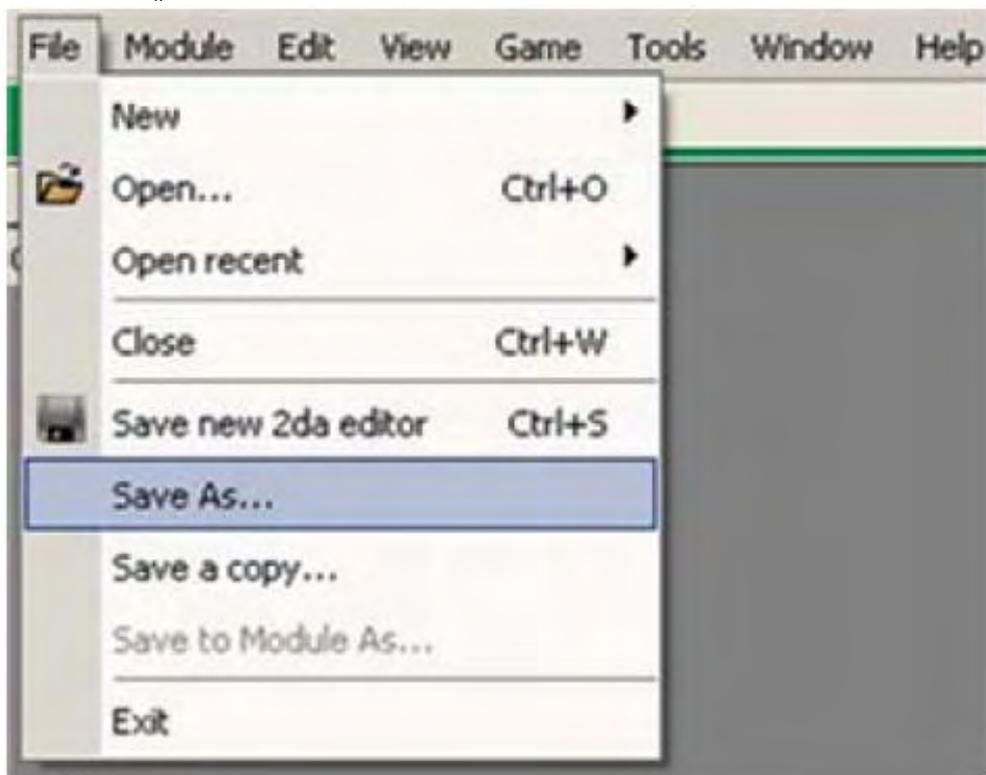


Wenn wir wollen, dass unsere 2da Datei Spalten- und die Zeilentitel enthält, müssen wir „Fix / Unfix“ aus dem Menü auswählen bevor wir die Datei speichern:

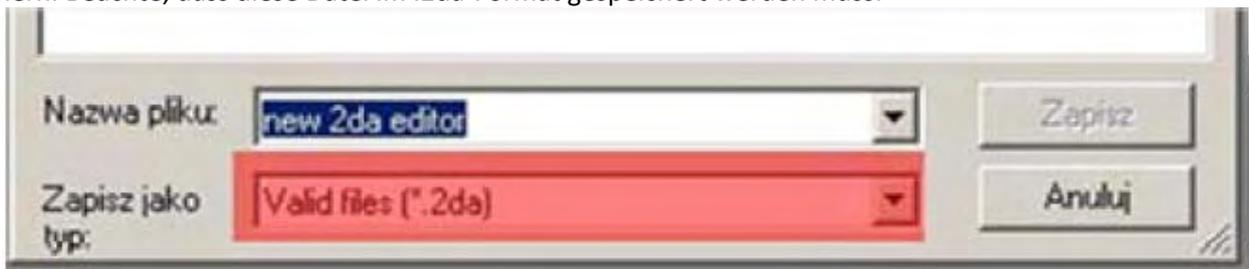


Dadurch würden wir erste Spalte und die erste Zeile in einen Titel ändern. Um die Datei später zu ändern müssten wir Fix / Unfix im Menü auswählen, sonst können neue Spalten und Reihen nicht hinzugefügt werden.

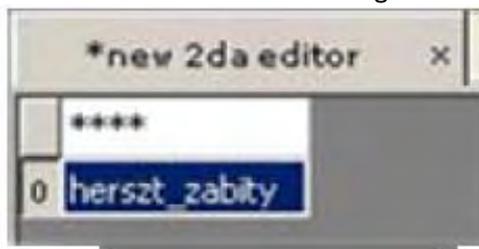
Unsere .2da Datei braucht keinen Titel, darum lassen wir ihn, wie es ist. Wir müssen nur die Datei speichern. Dafür wählen wir „Save As“ im Menü aus:



Das öffnet das Standard Windows Fenster zum Speichern der Datei. Die .2da Dateien vom Witcher findet man im Data\2DA Ordner. Öffne diesen Ordner, gib dann den Namen globaldflags ein und klicke auf speichern. Beachte, dass diese Datei im .2da-Format gespeichert werden muss:



Wenn ein andere Dateityp stattdessen angeboten wird, bedeutet das wahrscheinlich, dass ein anderes Windows Fenster als aktiv markiert ist. Stelle vor dem Speichern der Datei sicher, dass das neue 2da Editor Fenster aktiv ist. Wenn das Fenster inaktiv ist wird die Titelleiste grau sein:

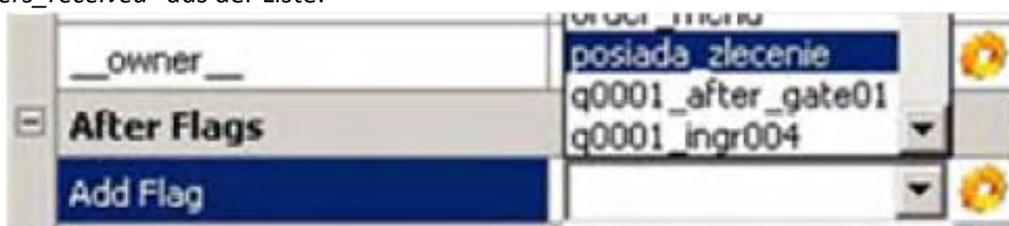


Eine andere Farbe bedeutet das dieses Fenster Aktiv ist:

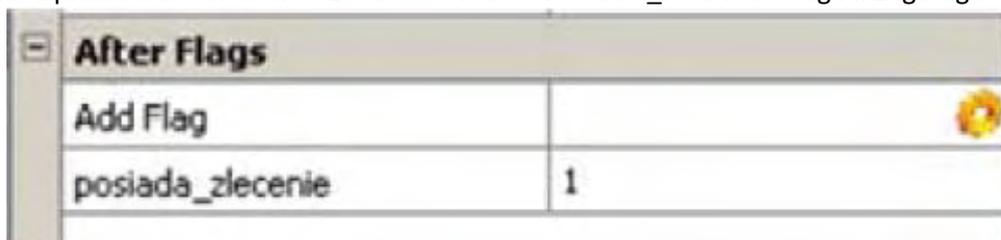


Das Sternchen vor dem Fensternamen zeigt an, dass die Datei nach der letzten Änderung noch nicht gespeichert wurde. Erst, wenn diese Dateivorlage(template) mit „Save as“ gespeichert wurde, können wir die new 2da Editor Datei speichern. Dadurch wird die 2da Datei direkt im Modul gespeichert. Mache das jetzt bitte **NICHT**.

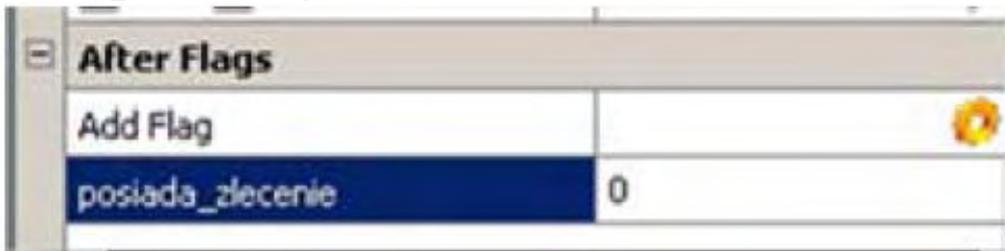
Zurück zu unserem Siegfried-Dialog. Klicke im Properties-Menü in das leere Feld rechts von „Add Flag“, und wähle „orders_received“ aus der Liste:



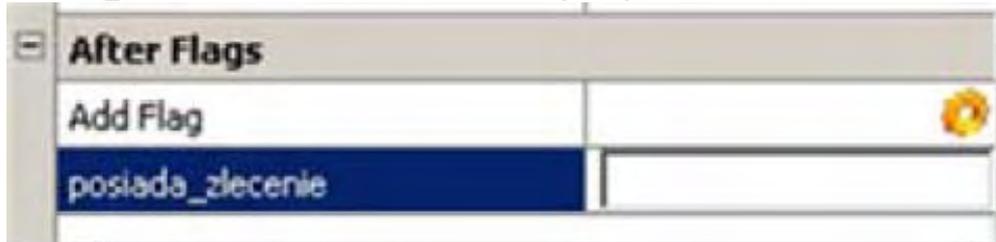
Diese Liste ist alphabetisch sortiert. Jetzt haben wir einen "orders_received" Flag hinzugefügt:



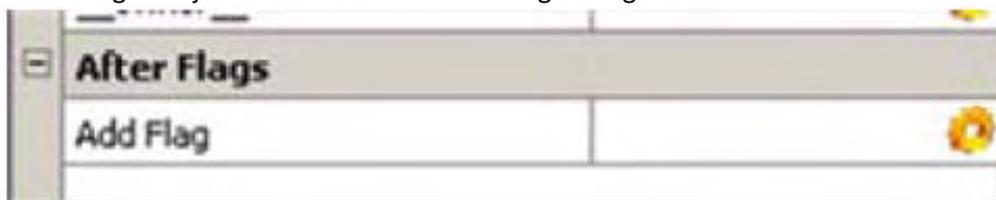
Flags werden in nur zwei Zuständen unterschieden. Entweder ist die Flag True, dann bekommt sie den Wert 1, oder False, dann ist ihr Wert 0. Wenn Geralt seinen Dialog mit Siegfried beginnt, hat er noch keine Quest von ihm, darum müssen wir den Wert der Flag auf 0 setzen. Um den Wert zu ändern, klicke in das Feld rechts neben "orders_received" und gib ihr den Wert 0. Drücke Enter:



Wenn du eine Flag im After Flags Fenster löschen willst, klicke in das Fenster rechts neben der zu löschenden Flag (z.B. orders_received) und lösche einfach den eingetragenen Wert (0 oder 1):



Drücke Enter. Die Flag wird jetzt automatisch aus der Flag-Liste gelöscht:

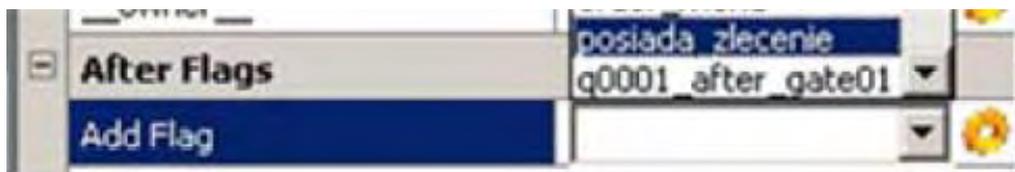


Wenn du eine Flag gelöscht hast, kannst Du sie, wie oben beschrieben, wieder hinzufügen, indem du sie aus der Liste neben Add Flag auswählst und ihr z. B. den Wert 0 zuweist.

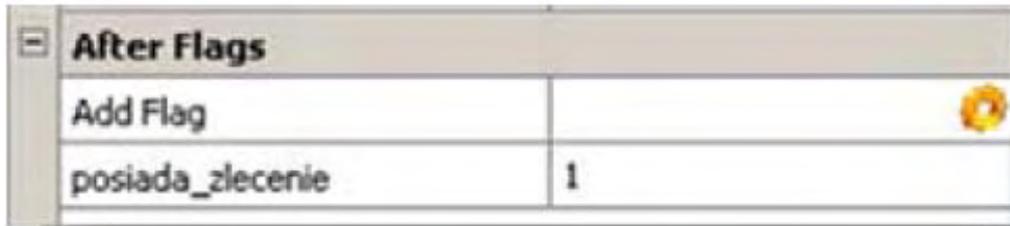
Klicke jetzt auf den "Erledigt" Eintrag:



Klicke im Properties Fenster, in das Feld rechts neben „Add Flag“ und wähle dann „orders_received“ aus der Liste aus:

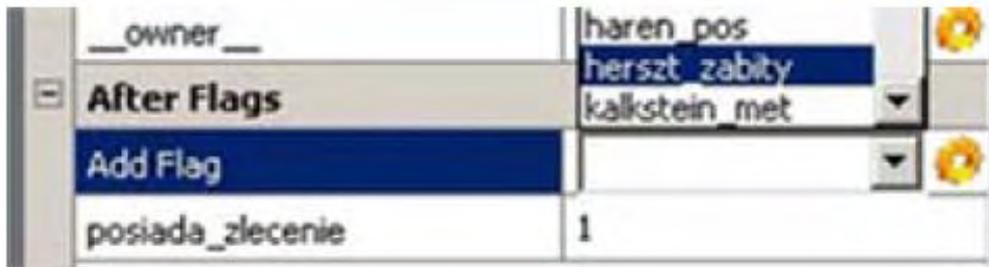


Weil die neue Flag automatisch den Wert 1(true) bekommt, brauchen wir hier nichts zu ändern:



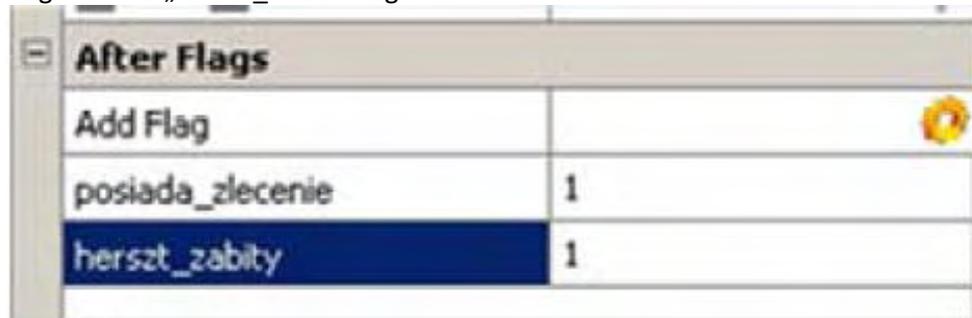
After Flags	
Add Flag	
posiada_zlecenie	1

Klicke auf das leere Feld rechts neben Add Flags und wähle dann „leader_killed“ von der Liste aus:



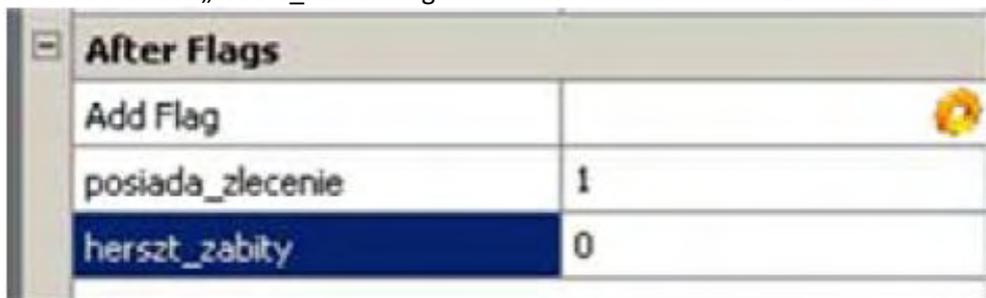
After Flags	
Add Flag	
posiada_zlecenie	1

Das fügt der Flag Liste die „leader_killed“ Flag hinzu:



After Flags	
Add Flag	
posiada_zlecenie	1
herszt_zabity	1

Hier müssen wir aber den Wert des Flags auf 0 setzen, weil der Banditenanführer noch nicht tot ist. Klicke in das Feld rechts neben der „leader_killed“ Flag und ändere den Wert auf 0. Drücke Enter:



After Flags	
Add Flag	
posiada_zlecenie	1
herszt_zabity	0

Diese Flags sagen dem Spiel wann welche Dialogteile angezeigt werden sollen. Der Dialog den wir in einer Minute hinzufügen werden, soll nur angezeigt werden, wenn der Wert der *orders_received* Flag 1(True) ist. Wir wissen bereits, dass diese Flag den Wert 1 erst bekommen wird, nachdem Geralt die Quest von Siegfried, mit dem *„Erledigt“* Eintrag angenommen hat.

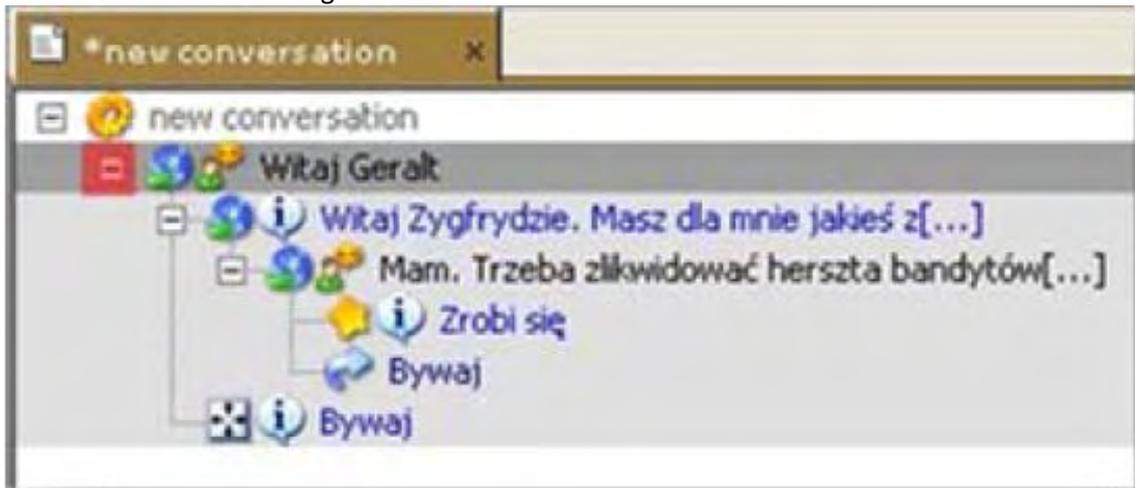
Quest Dialog Flags sind nur unter „Add Flags“ in den Properties änderbar. Ihre Werte werden nach dem Auswählen eines bestimmten Dialog-Eintrags gesetzt. Das bedeutet, wenn ein Spieler einen Quest-Eintrag mit einer Quest id annimmt, wird der vorherige Eintrag während des weiteren Gespräches nicht mehr angezeigt.

STANDARD UND CUTSCENE CONVERSATIONEN

Standard und Cutscene-Konversationen

Jetzt werden wir einen anderen Dialog für Siegfried erstellen. Dieser wird dann angezeigt, wenn Geralt (Der Spieler) die Quest von Siegfried annimmt, aber noch nicht beendet hat. Erst nach der Tötung des Banditenanführers, soll Geralt Siegfried von der Questerfüllung berichten können. Wir werden deshalb dasselbe Gespräch für diesen Zweck verwenden. Die in diesem Gespräch angezeigten Einträge werden von den Werten der Flags abhängen...

Zurück zu unserem Dialog. Wir müssen dafür keine neue Datei erstellen. Damit die vorhandenen Quest-Dialoge unserer Arbeit nicht stören, können wir die Verzweigung zusammenklappen, indem wir auf das "-" Zeichen vor "Hallo Geralt" Eintrag klicken:



Um die Verzweigung wieder zu öffnen, klicke auf das kleine "+" Zeichen vor dem "Hallo Geralt" Eintrag:



Um einen Standard-Dialog hinzuzufügen, wähle den „new conversation“ Eintrag mit einem Rechtsklick aus:



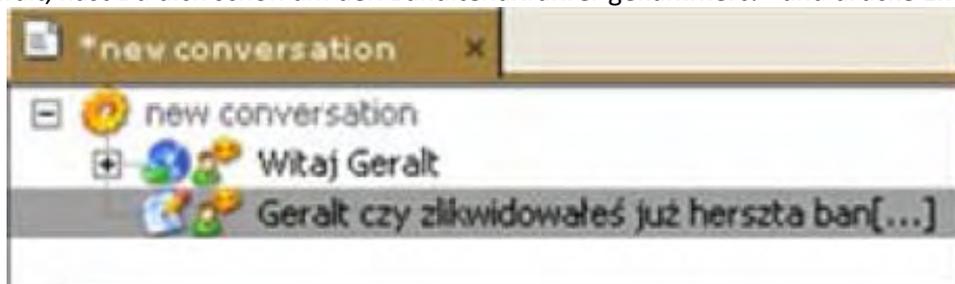
Wähle „Create Dialog Node“ aus dem Menü:



Das erstellt einen leeren Eintrag unter "Hallo Geralt." für einen neuen Standard Dialog:



Um einen Cutscene Dialog zu erstellen, der später in eine Witcher Cutscene (Filmsequenz) eingebaut werden kann, wähle „Create Cutscene Note“ aus dem Menü. Ein Cutscene-Dialog unterscheidet sich kaum von den bisher kennengelernten. Der einzige Unterschied liegt darin, dass Standard- und Questdialoge in derselben Datei gespeichert werden können. Bei einem Cutscene-Dialog geht das nicht. Er muss einzeln gespeichert werden. Es ist ebenfalls nicht möglich einen Standard- oder Quest-Dialog in einen Cutscene-Dialog einzubinden. Behalte das im Hinterkopf. Gehen wir jetzt zurück zu unserem Standard-Dialog. Bisher haben wir einen leeren Eintrag erstellt, der Siegfried zugeordnet ist. Klicke doppelt auf diesen leeren Eintrag und schreibe: "Geralt, hast Du dich schon um den Banditenanführer gekümmert?" und drücke Enter:

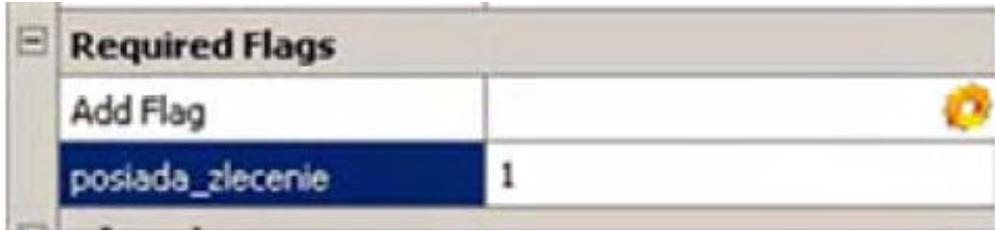


Jetzt müssen wir sicherstellen, dass dieser Dialog nur angezeigt wird, wenn die „orders_received“ Flag einen Wert 1 (True) hat. Weil Quest-Dialog-Eigenschaften nur in der „After Flag“ Liste modifizierbar sind, werden diese Werte erst gesetzt, nachdem eine Wahl durch Quest-Einträge getroffen wurde. Im Gegensatz dazu haben Standard-Dialoge sowohl Einträge in der „After Flag“ Liste, als auch in der „Required Flag“ Liste. In dieser anderen Liste werden die Flag Werte nur überprüft (aber nicht gesetzt). Entsprechend werden dann nur Dialog-Einträge mit dem passenden Wert angezeigt.

Jetzt wollen wir die „orders_received“ Flag zur „Required Flags“ Liste hinzufügen, damit das Spiel weiß, dass es diesen Wert abfragen soll. Klicke dafür auf das Feld rechts neben „Add Flag“ und wähle dann dann „orders_received“ aus der Menü Liste:

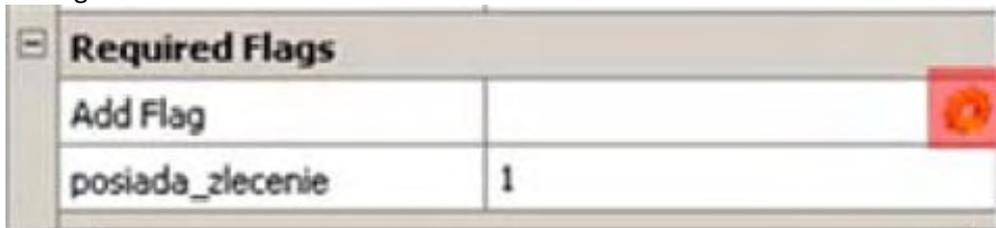


Dieses fügt die Flag der „Required Flags“ Liste zu:

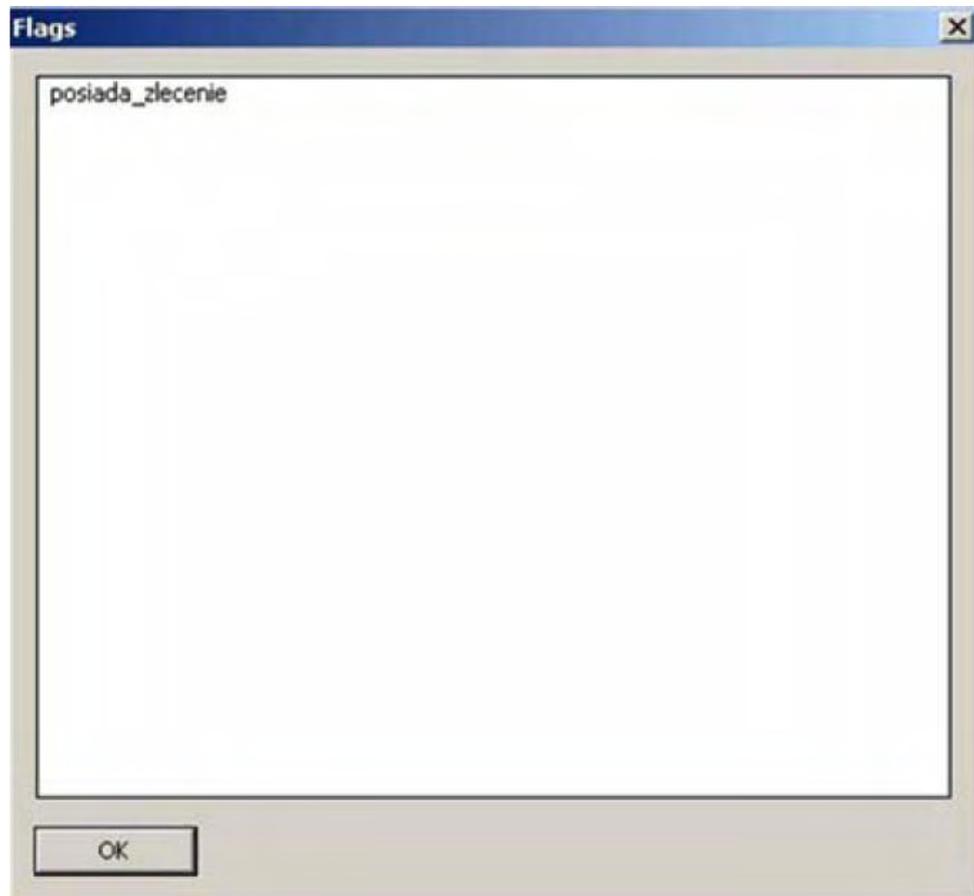


Um den Dialog mit folgendem Eintrag beginnen zu lassen, „Geralt, hast Du dich schon um den Banditenanführer gekümmert?“ muss der Wert der orders_received Flag 1 sein. Wir werden jetzt noch eine Flag hinzufügen, die den Wert 1 (True) haben soll, sobald Geralt(Der Spieler) den Banditenanführer getötet und Siegfried darüber informiert hat.

Um die Flag hinzuzufügen, klicke auf das kleine gelbe Symbol im Feld rechts neben „Add Flag“ im Bereich der „Required Flags“:



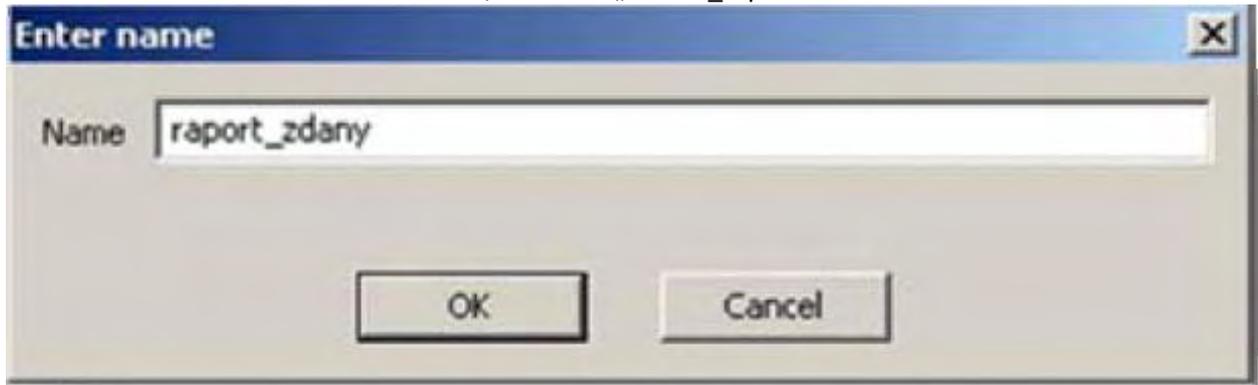
Das Flag Fenster öffnet sich:



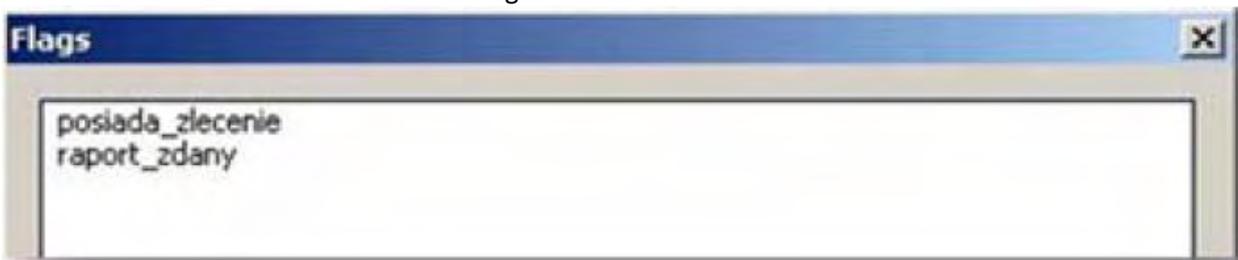
Rechts klicke auf das Feld und wähle „Create Name“ aus dem Menü:



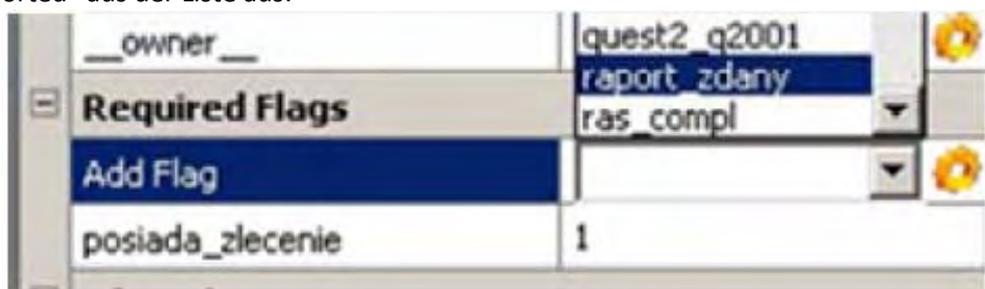
In das Enter Name Fenster das sich öffnet, schreibe: „results_reported“:



Klicke auf Ok. Jetzt haben wir bereits 3 Flags:



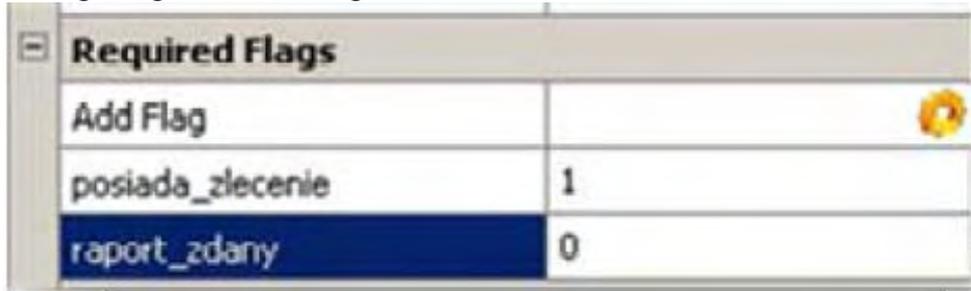
Klicke auf den kleinen Pfeil im Feld rechts neben „Add Flag“ im „Required Flags“ Bereich und wähle dann „results_reported“ aus der Liste aus:



Jetzt haben wir 2 Flags unter Required Flags:

Required Flags	
Add Flag	
posiada_zlecenie	1
raport_zdany	1

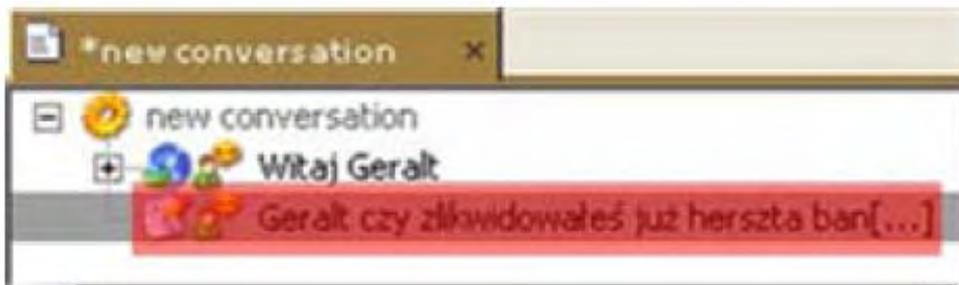
Setze jetzt den Wert der „results_reported“ Flag auf 0(False). Klicke dafür in das Feld rechts neben der „results_reported“ Flag und gib 0 ein. Bestätige mit Enter:



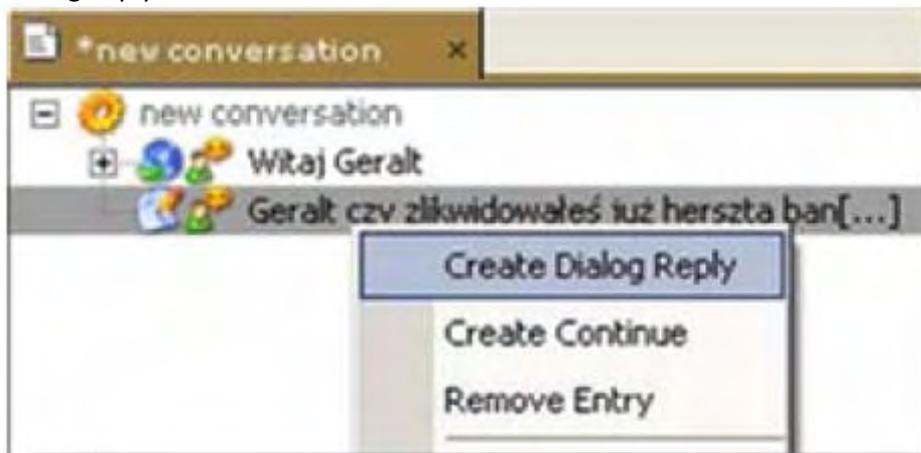
Required Flags	
Add Flag	
posiada_zlecenie	1
raport_zdany	0

Das Standardgespräch, das wir erstellen, wird nur angezeigt, wenn der Wert der „orders_received“ Flag 1 ist(d.h. Wenn der Spieler bereits die Quest von Siegfried angenommen hat) und der Wert der „results_reported“ Flag 0 ist(d.h. der Spieler hat Siegfried noch nicht vom Tod des Banditenanführers berichtet).

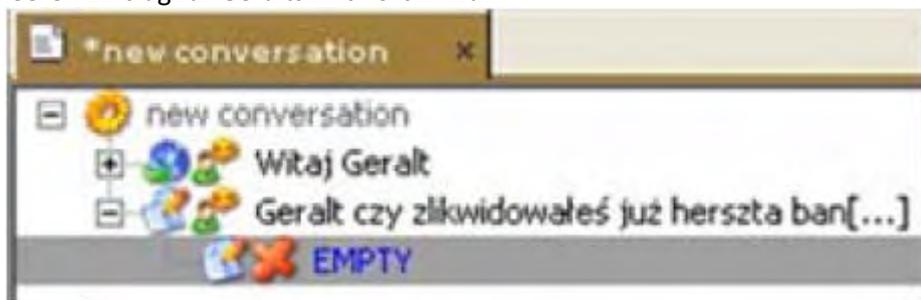
Lass uns weitere Dialog Einträge erstellen. Nun werden wir eine Antwort für Geralt hinzufügen. Wähle den "Geralt, hast du dich schon um den Banditenanführer gekümmert?" Eintrag mit einem Rechtsklick aus:



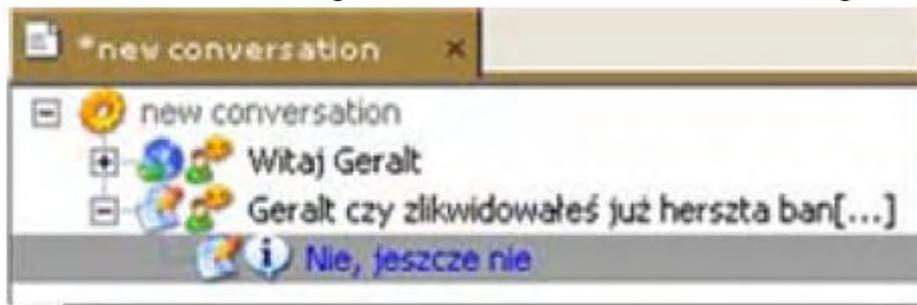
Wähle „Create Dialog Reply“ aus dem Menü aus:



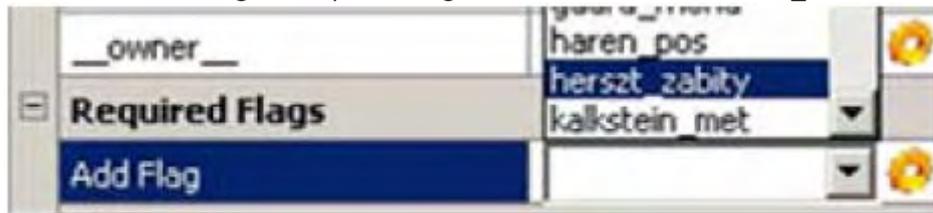
Das fügt einen leeren Eintrag für Geralts Antwort hinzu:



Klicke doppelt auf den neuen leeren Eintrag und schreibe: "Nein, noch nicht." Bestätige mit Enter:



Dieser Eintrag wird nur angezeigt, wenn Geralt den Banditenanführer noch nicht getötet hat. Klicke auf den kleinen Pfeil rechts neben Add Flag im Required Flags Bereich und wähle „leader_killed“ aus der Liste:



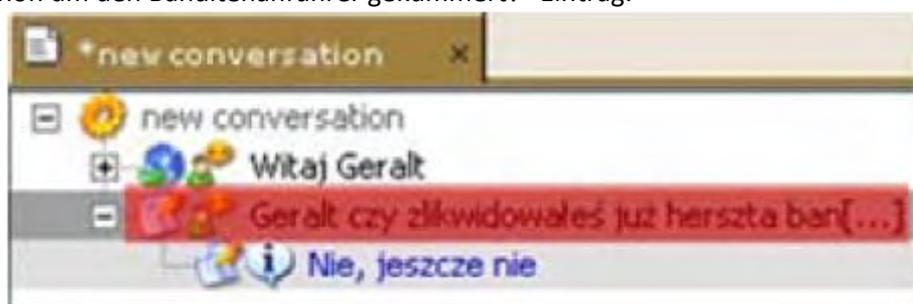
Das fügt die neue Flag dem Required Flags Bereich hinzu:



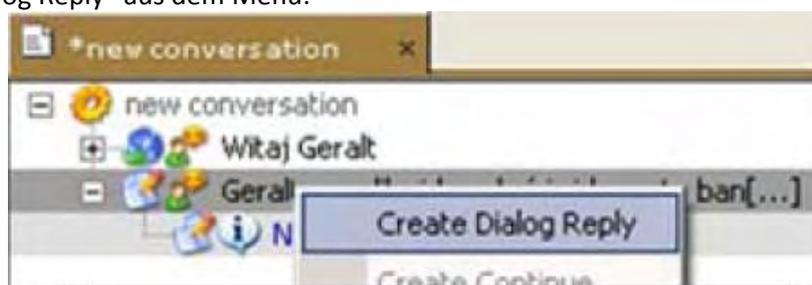
Ändere den Wert der Flag auf 0(False), so dass dieser Gesprächseintrag nur angezeigt wird, solange Geralt den Banditenanführer noch nicht getötet hat. Schreibe in das Feld rechts neben die leader_killed Flag eine 0 und drücke Enter:



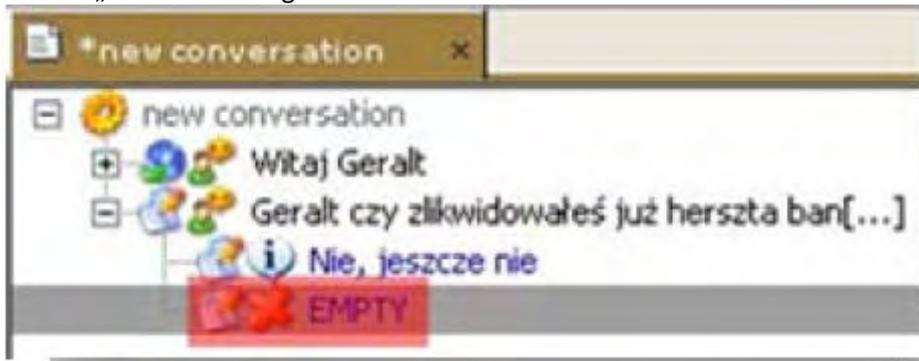
Jetzt fügen wir für Geralt eine alternative Antwort hinzu. Rechtsklick dafür noch einmal auf den "Geralt, hast du dich schon um den Banditenanführer gekümmert?" Eintrag:



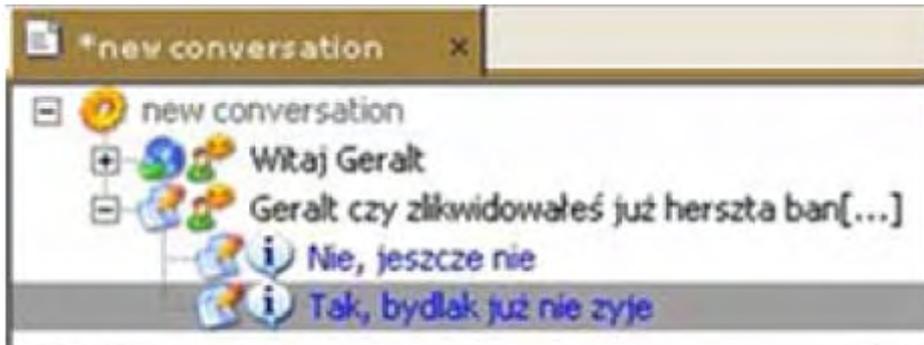
Wähle „Create Dialog Reply“ aus dem Menü:



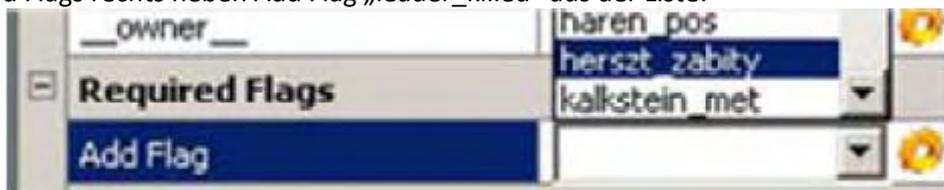
Klicke doppelt auf den „EMPTY“ Eintrag:



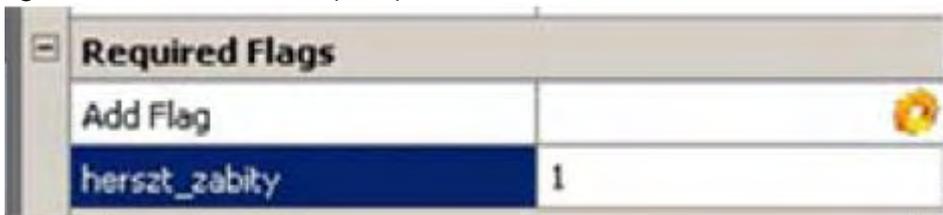
Schreibe: "Ja, der Bastard ist Tot". Und drücke Enter:



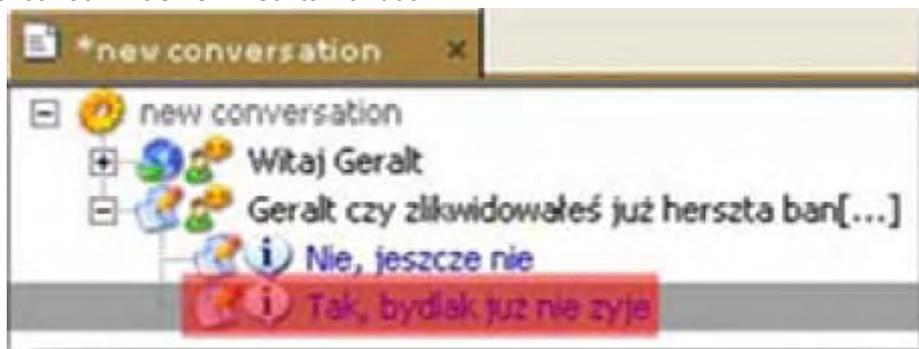
Dieser Eintrag soll erst angezeigt werden, nachdem Geralt den Banditenanführer getötet hat. Wähle dafür unter Required Flags rechts neben Add Flag „leader_killed“ aus der Liste:



Da die neue Flag automatisch den Wert 1(True) bekommen hat, müssen wir hier nichts verändern:



Als nächstes erstellen wir einen Dialogeintrag für Siegfried, als Antwort auf Geralts Erfolgsmeldung. Wähle: "Ja, der Bastard ist Tot" mit einem Rechtsklick aus:



Erstelle mit „Create Dialog Entry“ den „EMPTY“ Eintrag:



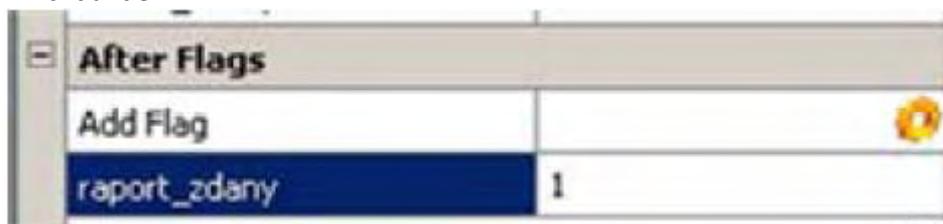
Klicke jetzt doppelt auf den leeren Eintrag und schreibe "Gut gemacht Geralt! Ich gebe dir 500 Orens Be-
lohnung". Dann drücke auf Enter:



Dieser Eintrag kann auch als Bericht für Siegfried dienen. Setze dafür den Wert der „results_reported“ Flag
auf 1 (True). Wähle dafür neben Add Flag „results_reported“ aus der Liste:



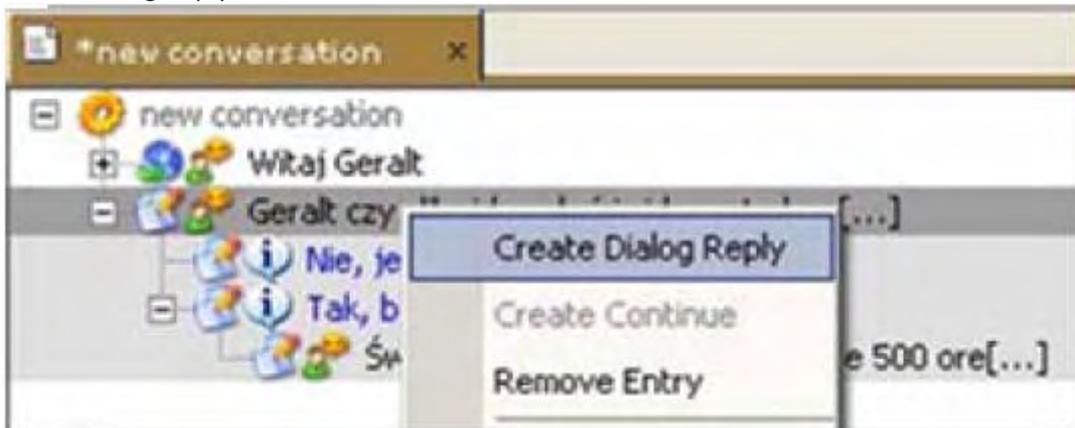
Das fügt die Flag der Flag Liste hinzu. Weil die Flag automatisch schon den Wert 1(True) bekommen hat,
müssen wir ihn nicht ändern:



Abschließend müssen wir eine Möglichkeit schaffen, das Gespräch mit Siegfried zu beenden. Wähle dafür
den "Geralt, hast du dich schon um den Banditenanführer gekümmert?" Eintrag mit einem Rechtsklick aus:



Wähle „Create Dialog Reply“ aus dem Menü:

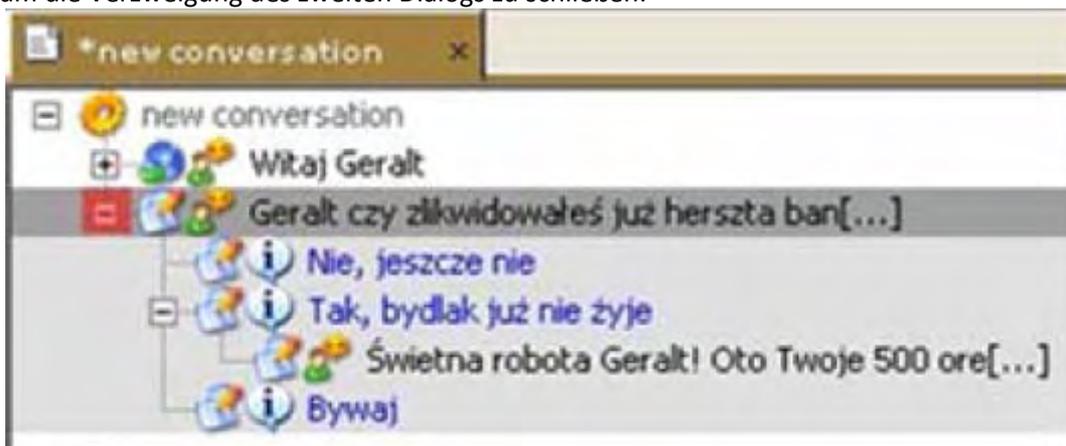


Klicke jetzt doppelt auf den leeren Eintrag und schreibe: "Leb wohl." Drücke Enter:

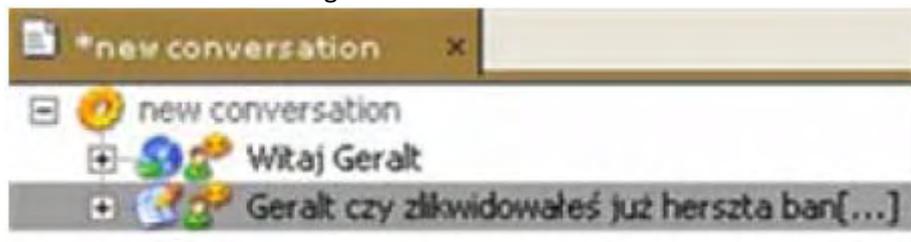


Damit ist Siegfrieds zweiter Dialog beendet. Wir werden noch einen dritten Dialog erstellen. Dieser einfache Dialog wird immer angezeigt, nachdem Geralt,(Der Spieler) Siegfrieds Quest beendet und die Belohnung erhalten hat.

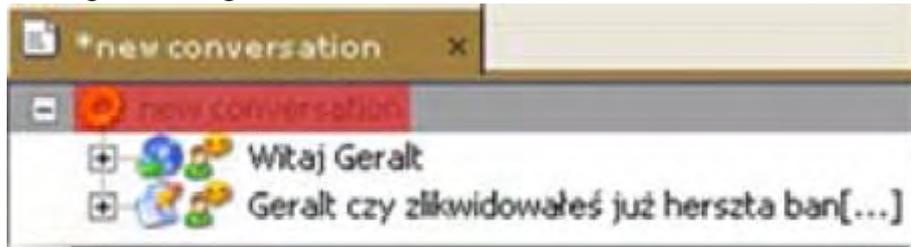
Klicke auf das Minus Zeichen vor dem: "Geralt, hast du dich schon um den Banditenanführer gekümmert." Eintrag, um die Verzweigung des zweiten Dialogs zu schließen:



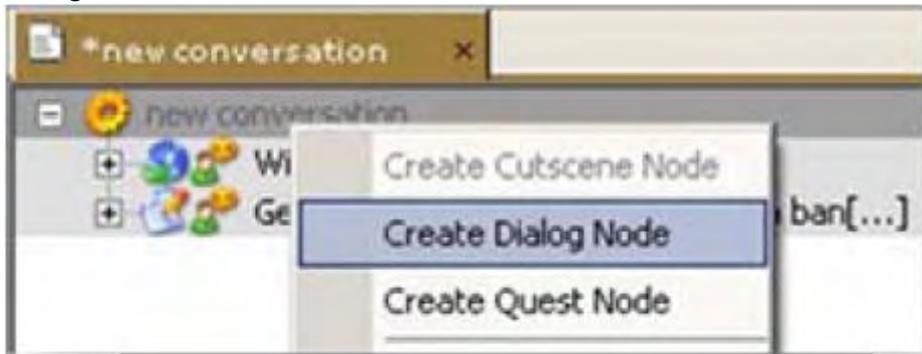
Dadurch wird das Weiterarbeiten im Dialogfenster übersichtlicher.



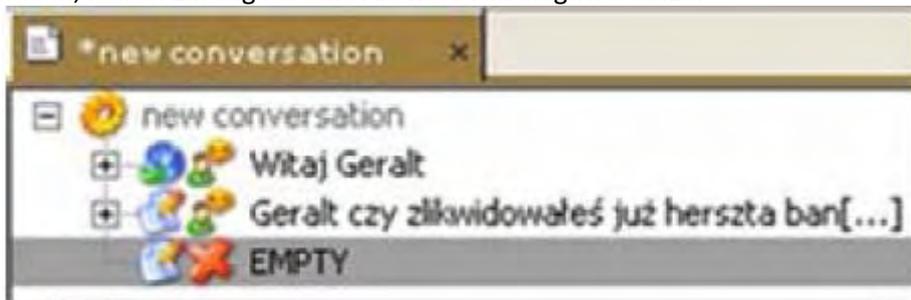
Um einen neuen Dialog hinzuzufügen, klicke mit der rechten Maustaste auf „new conversation“:



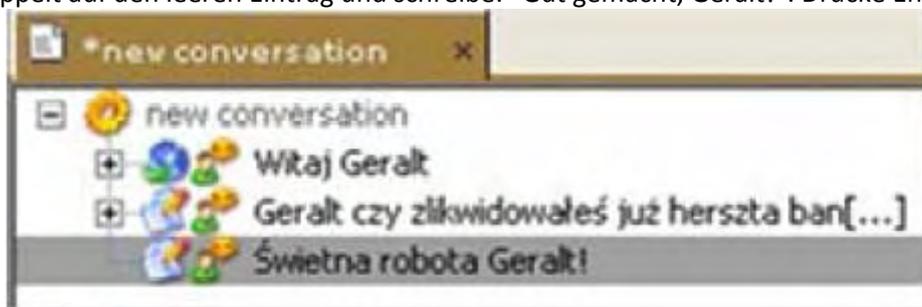
Wähle „Create Dialog Node“ aus dem Menü:



Das fügt einen neuen, leeren Eintrag für eine Antwort von Siegfried hinzu:



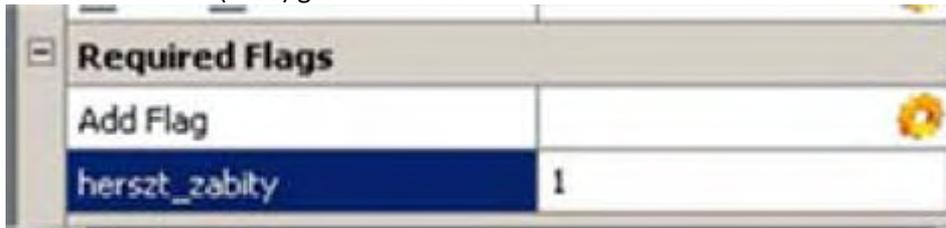
Klicke dann doppelt auf den leeren Eintrag und schreibe: „Gut gemacht, Geralt!“. Drücke Enter:



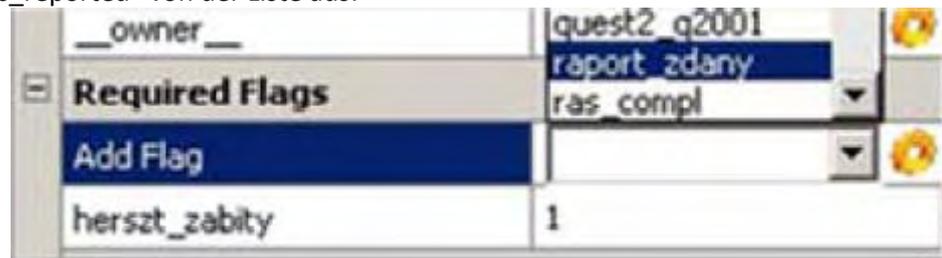
Wir sorgen als nächstes dafür, dass der "Gut gemacht, Geralt." Eintrag angezeigt wird, wenn die Flags „leader_killed“ und „results_reported“ den Wert 1 haben. Dafür wählen wir im Required Flags Bereich neben Add Flag zunächst die „leader_killed“ Flag aus:



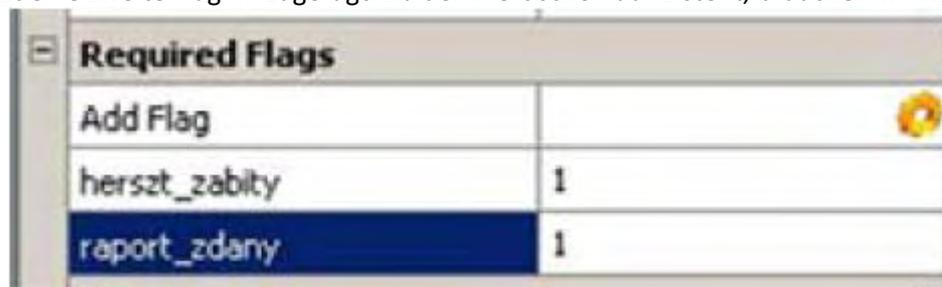
Dadurch erscheint die „Leader_killed“ Flag in den Required Flags. Wir brauchen ihren Wert nicht zu ändern, da er bereits automatisch auf 1 (True) gesetzt wurde:



Jetzt füge noch eine andere Flag hinzu. Klicke dafür wieder auf den kleinen Pfeil rechts neben Add Flag und wähle „results_reported“ von der Liste aus:

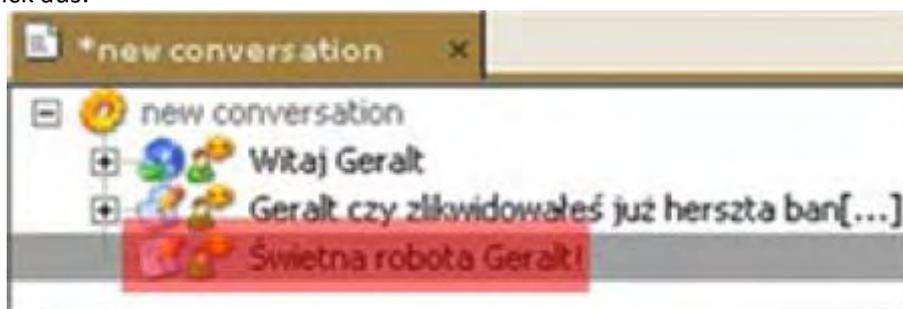


Wir haben jetzt eine zweite Flag hinzugefügt. Da der Wert schon auf 1 steht, brauchen wir nichts zu ändern:

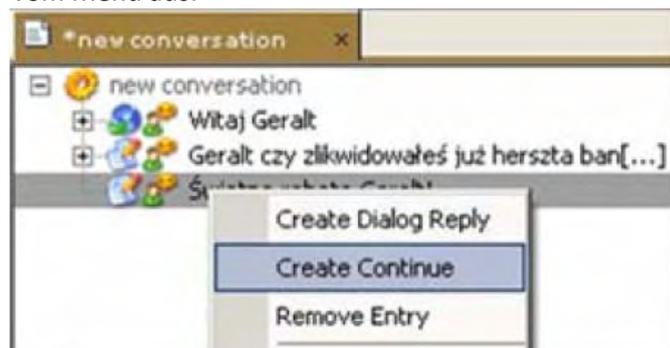


Dank dieser Flags wird Siegfrieds Antwort "Gut gemacht, Geralt!" angezeigt, wenn die Quest erfüllt ist und die Belohnung gezahlt wurde. Nun werden wir einen weiteren Gespräch Eintrag hinzufügen, der aber weiterhin von Siegfried gesprochen wird. Bisher haben in unseren Dialogen abwechselnd Siegfried und Geralt gesprochen.

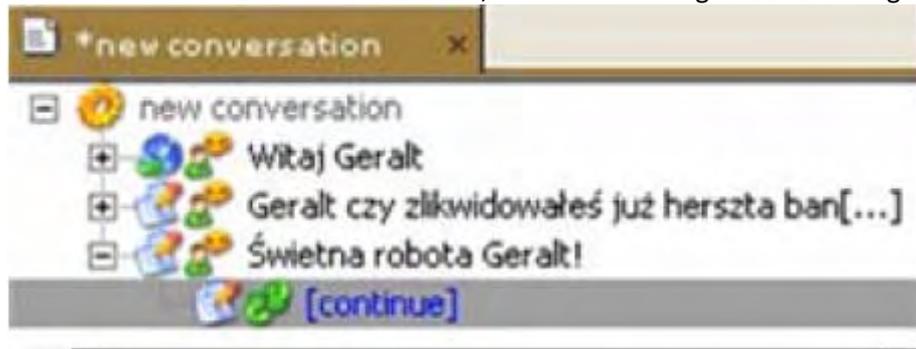
Um Siegfrieds Part um eine weitere Textzeile zu erweitern, wähle den "Gut gemacht, Geralt!" Eintrag mit einem Rechtsklick aus:



Wähle „Create Continue“ vom Menü aus:



Dadurch haben wir einen Eintrag erzeugt, der während eines Dialogs nicht angezeigt wird. Geralts Dialogtext wird hierbei einfach durch einen Mausklick ersetzt, auf den dann Siegfrieds Text fortgeführt wird:



Wähle den Neuen Eintrag (*Continue*) mit einem Rechtsklick aus, und dann „Create Dialog Entry“ im Menü:



Das fügt noch einen zweiten „EMPTY“ Eintrag hinzu:



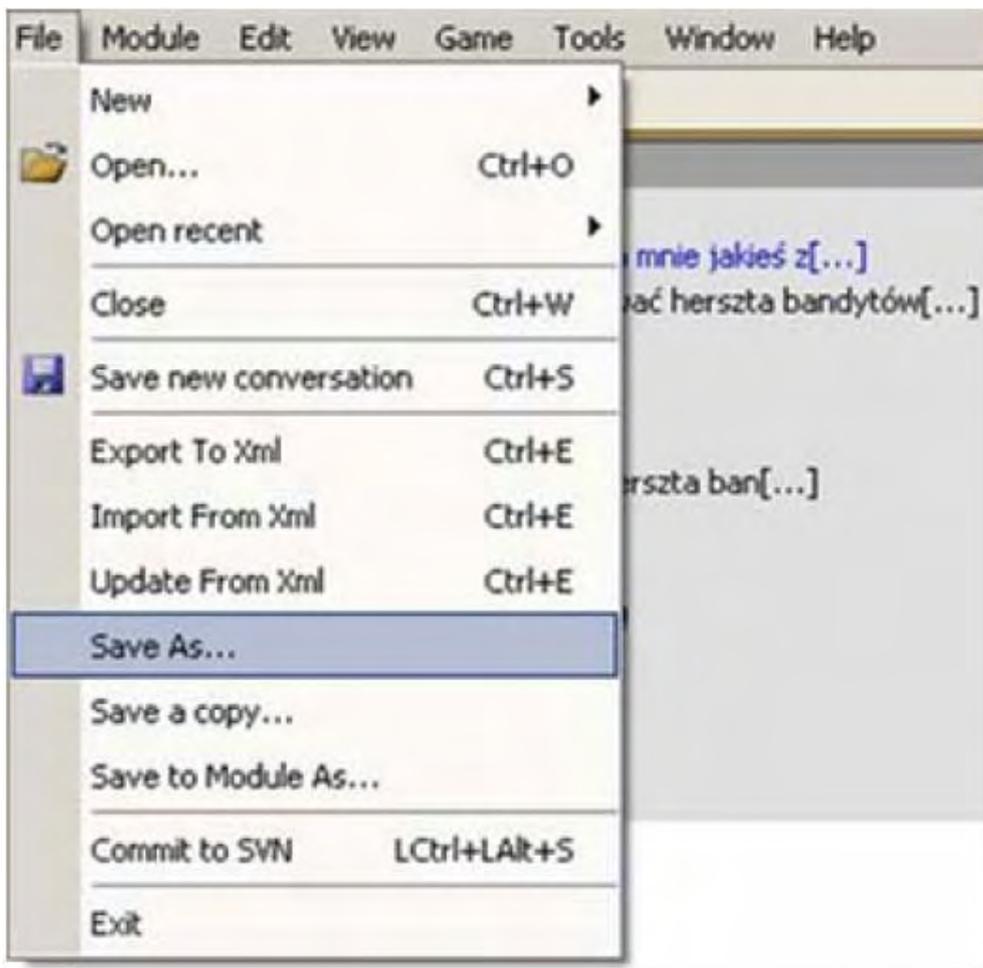
Dieser neue Eintrag gehört auch zu Siegfried. Klicke doppelt auf diesen Eintrag und schreibe dann "Weiter so!". Bestätige mit Enter:



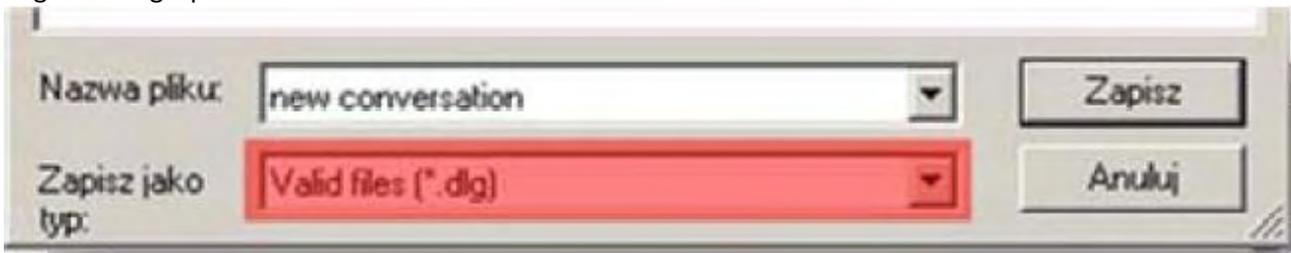
Jetzt ist unser Dialog mit Siegfried fertig und so sieht er aus:



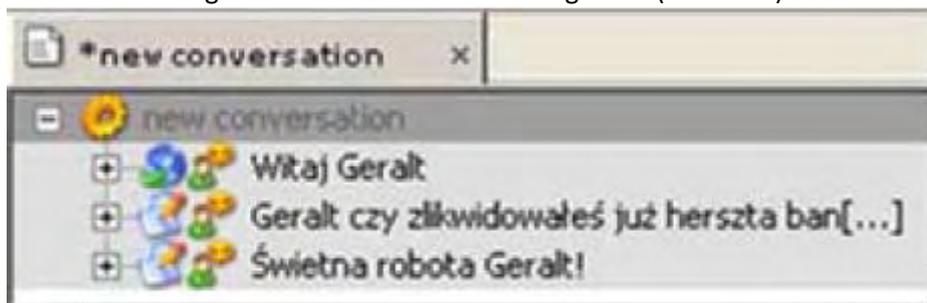
Als nächstes müssen wir unseren Dialog noch speichern. Klicke dafür auf „File“ und wähle „Save As...“ aus dem Menü:



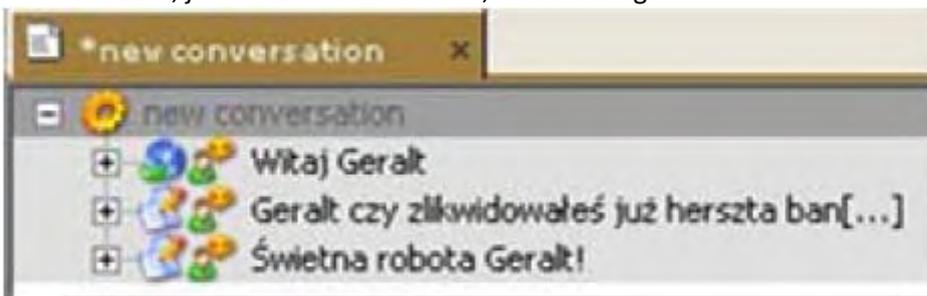
Das öffnet ein normales Windows Fenster für das Speichern von Dateien. Dialoge sollten in einem eigenen Dialog-Ordner gespeichert werden. In unserem Fall ist das der Data/Dialogues Ordner. Gehe zu diesem Ordner, und speichere die Datei als siegfried_conv. Denke daran, das der Name der Datei nicht länger als 16 Zeichen lang sein darf sonst kann der D'jinni Editor sie nicht verarbeiten. Achte darauf, dass die Datei im .dlg-Format gespeichert wird:



Wenn ein anderer Dateityp (statt .dlg) als Speichertyp angezeigt wird, war Dein letztes aktives Windows-fenster nicht das mit dem Dialog-Inhalt. Das ist auch an einer grauen (inaktiven) Titelleiste zu erkennen:



Ein aktives Fenster erscheint, je nach Windows-Theme, andersfarbig:

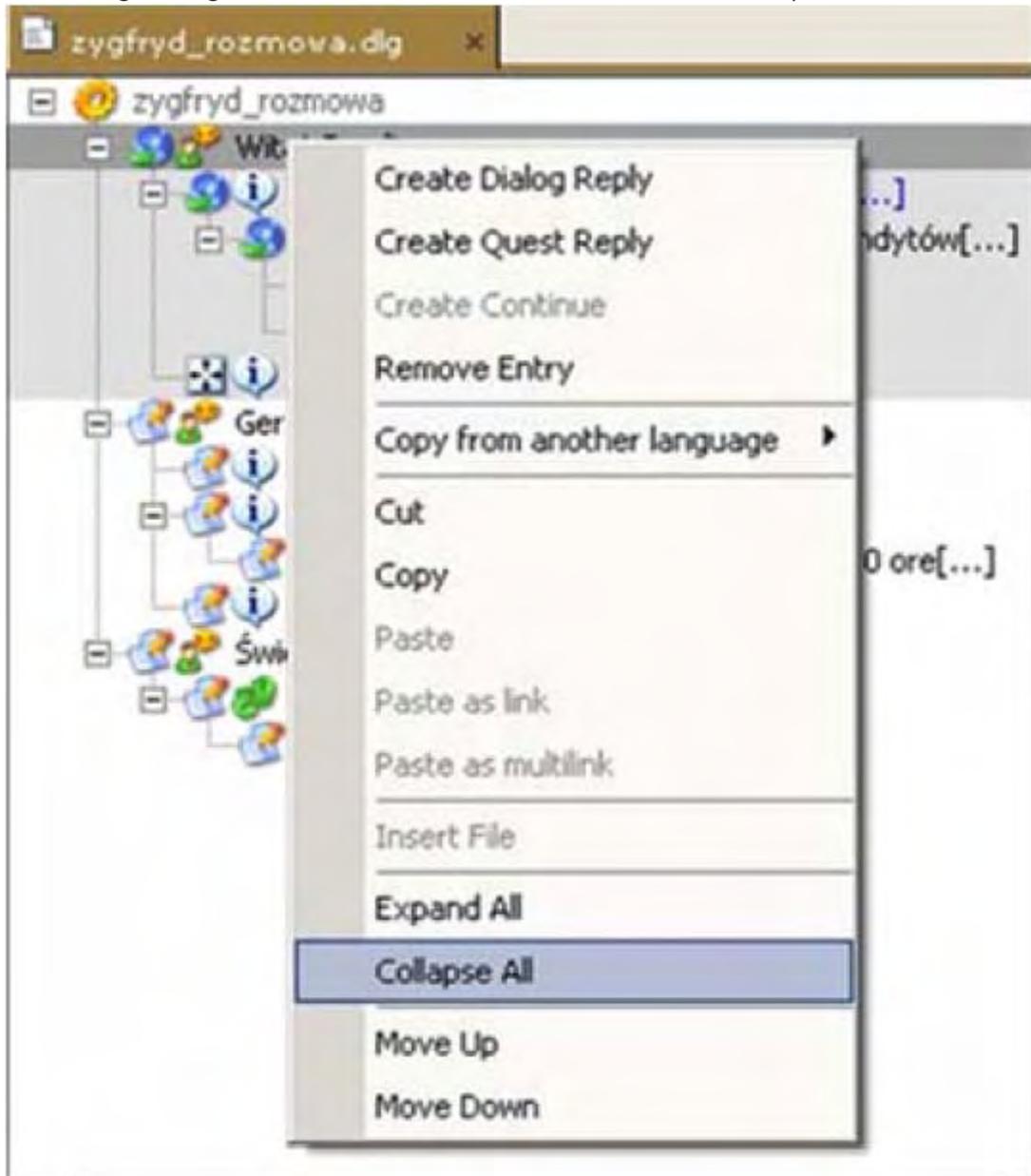


Ein Sternchen vor dem Namen in der Titelleiste bedeutet, dass diese Datei seit der letzten Änderung noch nicht gespeichert wurde. Um Vorlagen (Templates) zu speichern wähle anstelle von „Save As...“ besser „Save new conversation“ aus dem File Menü aus. Dadurch wird die Datei direkt im Modul gespeichert. Mache das aber jetzt noch nicht.

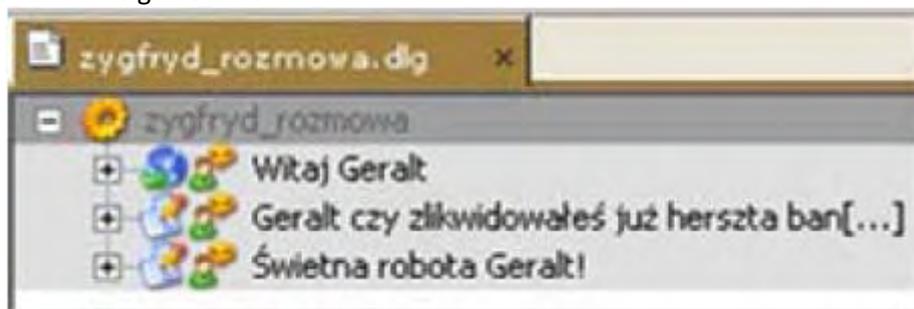
Wir sollten auch die Möglichkeiten probieren, Dialog-Einträge zu löschen, oder Dialog-Verzweigungen aufzufächern bzw. einzuklappen. Wie du sehen kannst wird unser Dialog in der Form eines Baums angezeigt. Es gibt drei Hauptzweige in diesem Baum:



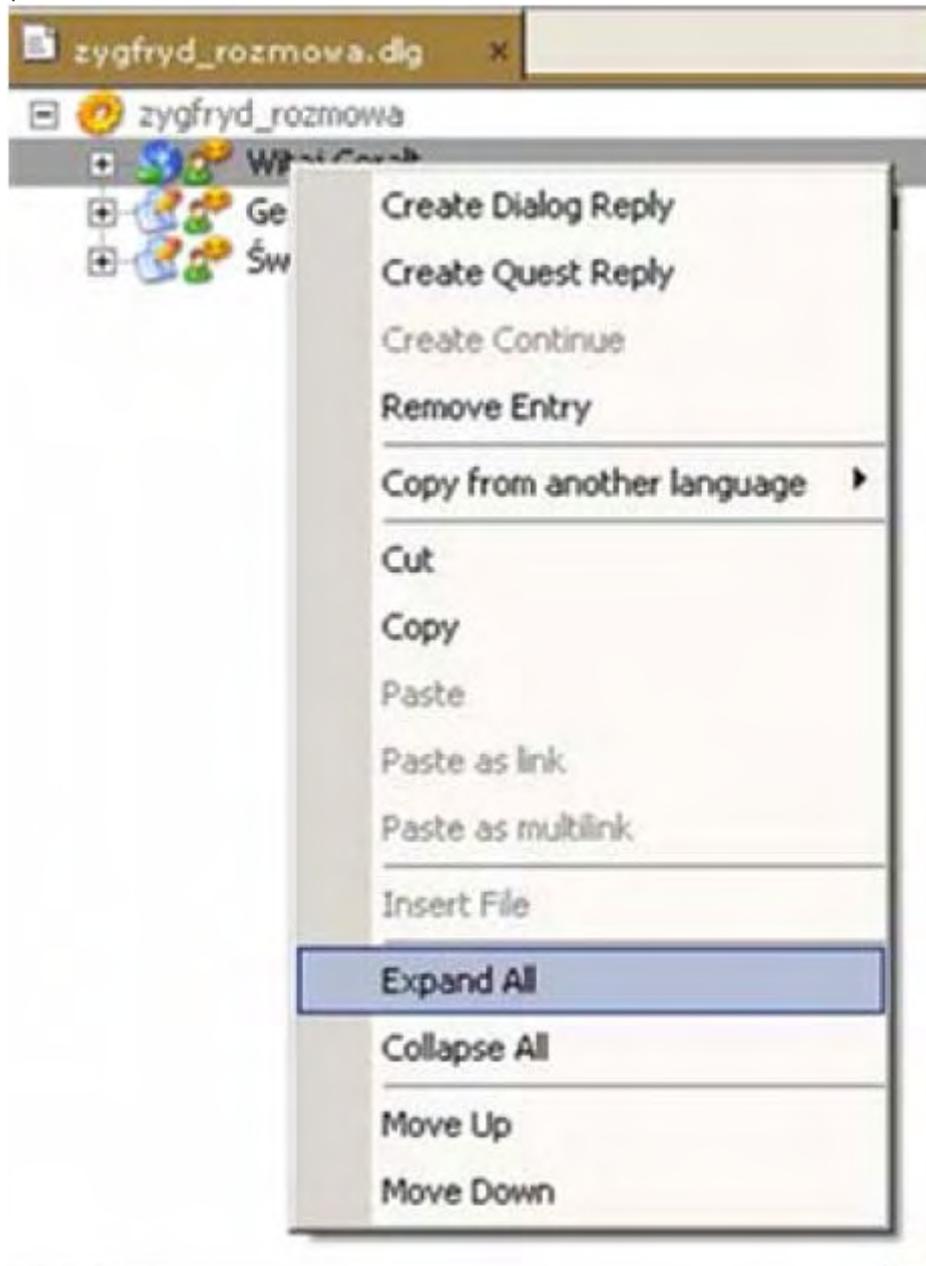
Wähle einen Dialog-Eintrag mit einem Rechtsklick aus, und wähle dann „Collapse All“ aus dem Menü:



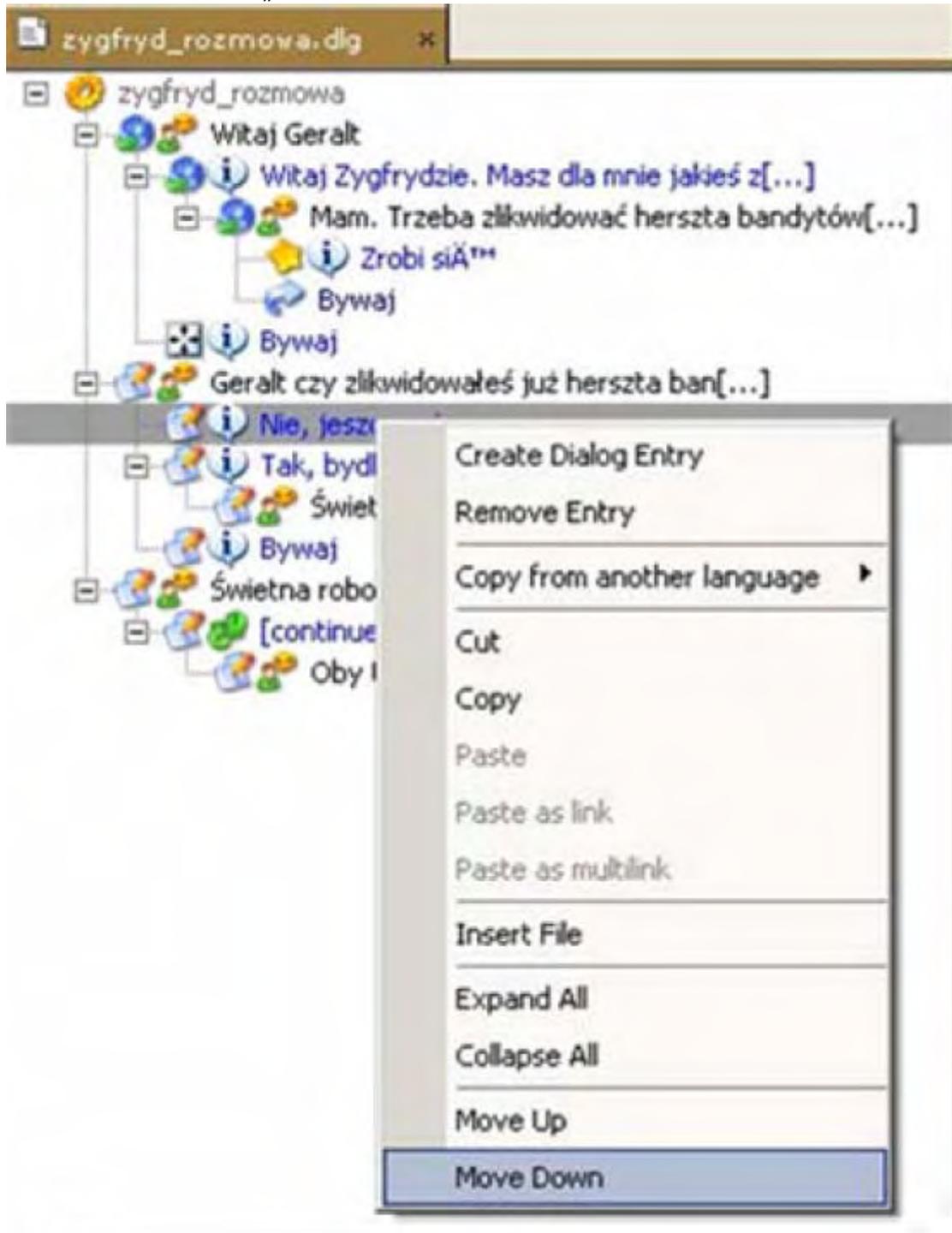
Dadurch klappen alle Zweige des Baums zusammen:



Wähle jetzt „Expand All“ aus dem Menü:



Das fächert alle Zweige des Baums wieder auf. Wähle jetzt den "Nein, noch nicht!" Eintrag mit einem Rechtsklick aus und wähle dann „Move Down“ aus dem Menü:



Das verschiebt den "Nein, noch nicht!" Eintrag unter den "Ja, der Bastard ist tot" Eintrag:



Um den Eintrag jetzt wieder an seine ursprüngliche Position zu bewegen, wähle "Nein, noch nicht" mit einem Rechtsklick aus, und wähle dann Move Up aus dem Menü.

Wähle jetzt einen beliebigen Eintrag mit einem Rechtsklick aus und wähle dann Insert File aus dem Menü:



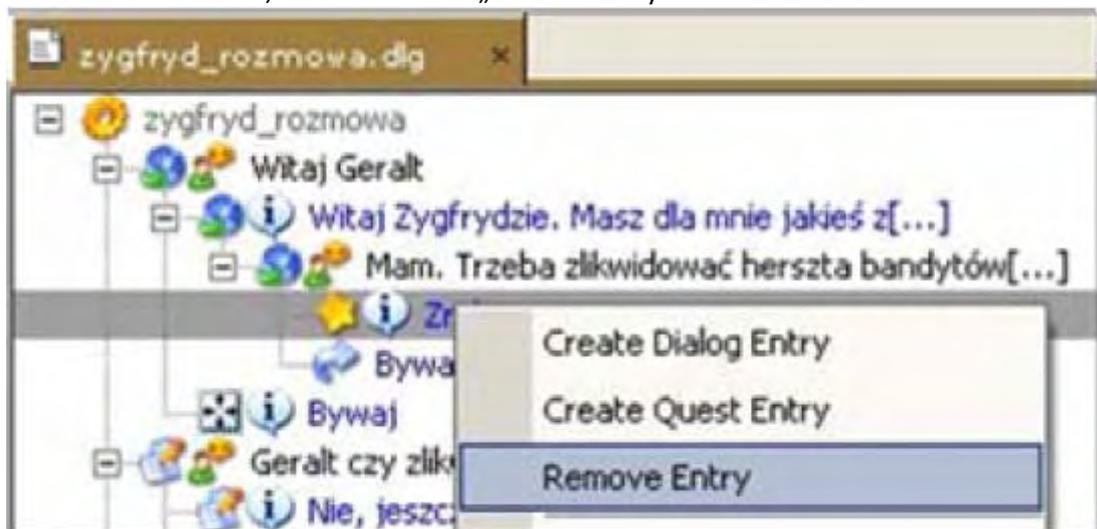
Das öffnet zuerst einmal das normale Windows Fenster zum laden von Dateien. „Insert File“ kann dazu genutzt werden um einen bereits existierenden Dialog mit einem ausgewählten Eintrag zu verknüpfen(verlinken). Erwinnere dich, dass alle Gespräche im Data\Dialogues Ordner gespeichert werden.

Die weiteren gebräuchlichen Befehle des Menüs sind: Cut, Copy und Paste:



Diese werden für das ausschneiden, kopieren und einfügen von Dialog Einträgen verwendet, der Paste as Link Befehl wird verwendet um einen Eintrag mit einem anderen Eintrag im gleichen Dialog zu verknüpfen.

Jetzt noch ein paar Worte über das löschen von Einträgen. Um einen einzelnen Eintrag zu löschen, wähle ihn mit einem Rechtsklick aus, und wähle dann „Remove Entry“ aus dem Menü:



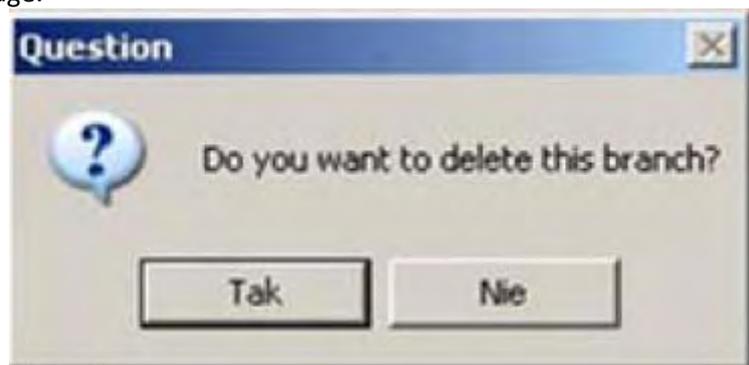
Der ausgewählte Dialog Eintrag wird jetzt gelöscht. Sei also sehr vorsichtig, wenn du einen Eintrag von einem Hauptzweig des Dialogs wie z.B. "Hallo Geralt" entfernst:



Als Resultat wird dann der komplette Zweig gelöscht:



Um aber zu verhindern, dass der komplette Zweig eines Dialogs irrtümlich entfernt wird, erscheint sicherheitshalber eine Abfrage:



Klickt man hier auf „Yes“, wird der ganze Zweig gelöscht. So weit sind die Menüpunkte erläutert. Als nächstes müssen wir unseren neuen Dialog Siegfried zuordnen.

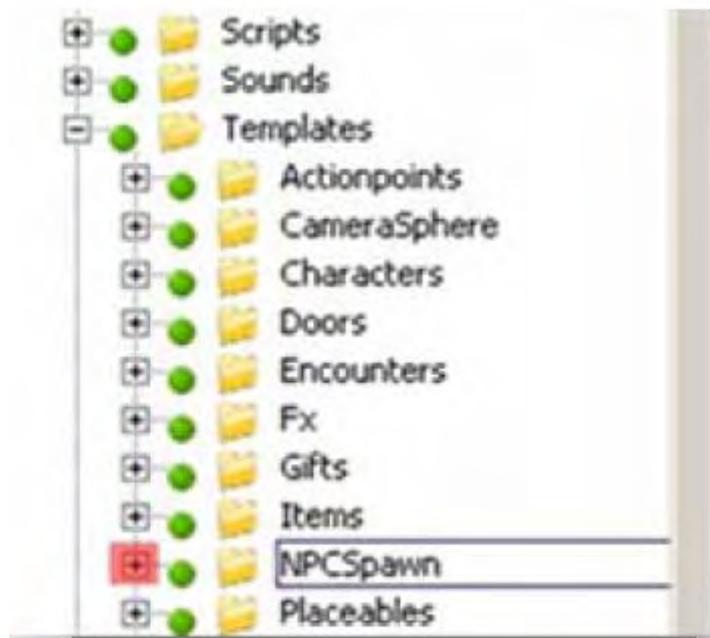
Das Gespräch kann Siegfrieds Profil (template) zugeordnet werden, obwohl das bei einem NPC nicht besonders ratsam ist. Da Siegfried im Laufe des Spiels mehrere Dialoge haben kann, ordnen wir ihn stattdessen einem von Siegfrieds Spawn-Sets zu.

Zuerst müssen wir die Datei öffnen, die Siegfrieds Spawn-Set enthält. Wir werden dafür den Resources Explorer verwenden. Wähle dafür unter Utility Windows -> Resources Explorer aus dem Menü. Der Resources Explorer sollte sich jetzt auf der linken Seite des Bildschirms öffnen. Der Resources Explorer wird ausführlicher in einem anderen Kapitel des Handbuches beschrieben.

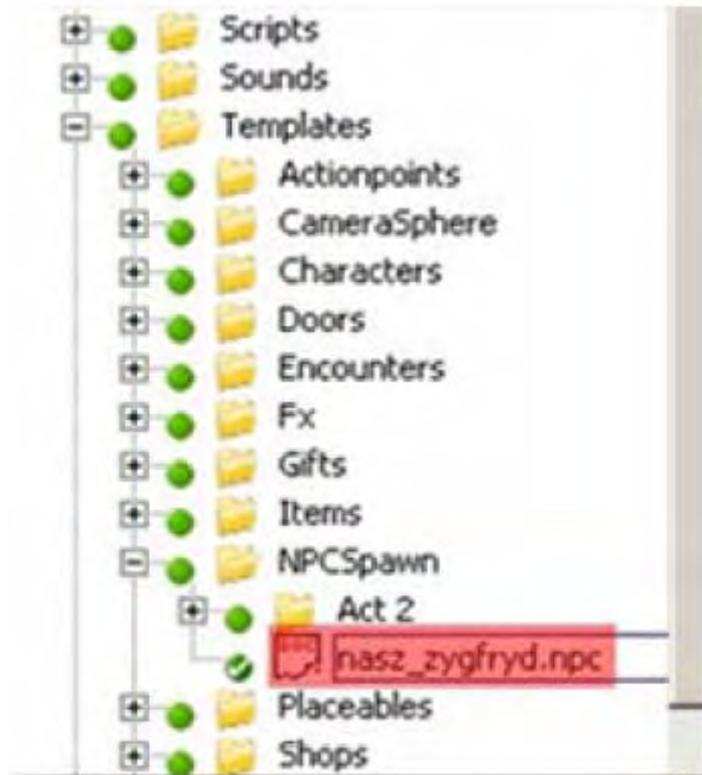
Wir haben Siegfrieds Spawn-Set im Data\Templates\NPCSpawn Ordner unter dem Namen „our_siegfried.npc“ gespeichert. Um dieses Datei zu öffnen, klicke im o.a. Verzeichnis auf das kleine "+" Zeichen vor dem Ordner Templates:



Klicke als nächstes auf das kleine "+" Zeichen vor dem Ordner „NPCSpawn“:



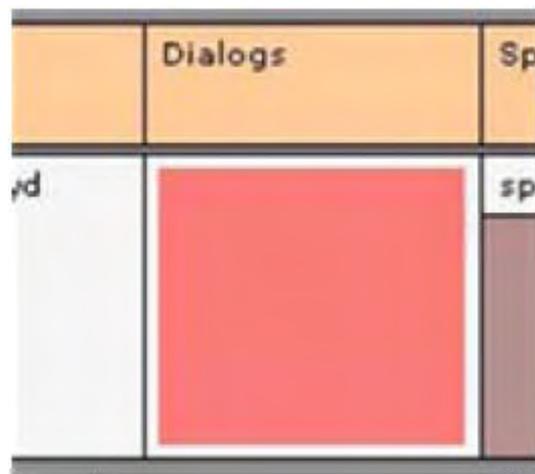
Klicke doppelt auf die „our_siegfried.npc“ Datei:



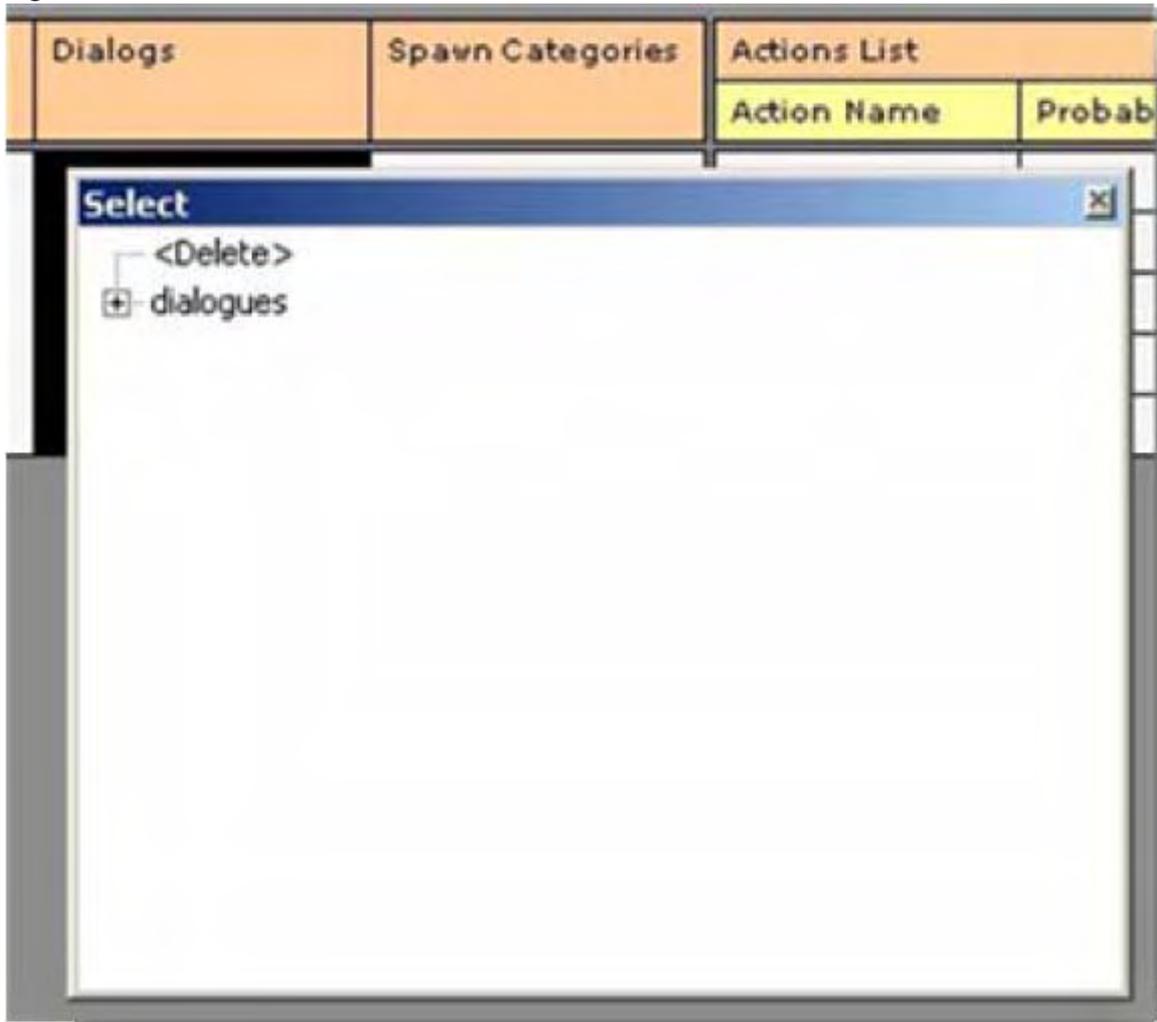
Das „our_siegfried“ Fenster öffnet sich, Dies Beinhaltet Siegfried’s Spawnset:

nasz_zygfryd.npc					
Comment					
Zygfyrd - zakonnik, zleceniodawca					
Story Phase	Spawn Phase	Start Time	End Time	OnSpawn Script	Template
zlecenie	oczekiwanie	00:00:00	00:00:00		nasz_zygfryd

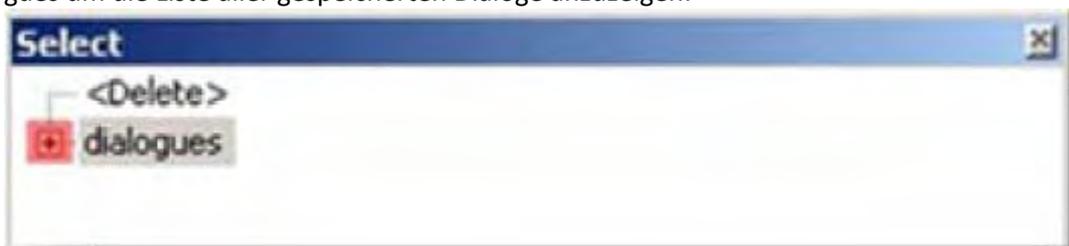
Ich möchte Dir zeigen, wie man einen Dialog einem Spawn Set zuteilt. Klicke doppelt auf das Feld unter der „Dialogs“ Spalte:



Das ausgewählte Fenster öffnet sich:



Der Ordner „Dialogues“ wird angezeigt. Um eine Dialog-Datei zuzuordnen klicke zuerst auf das kleine "+" vor Dialogues um die Liste aller gespeicherten Dialoge anzuzeigen:



Klicke doppelt auf siegfried_conv:



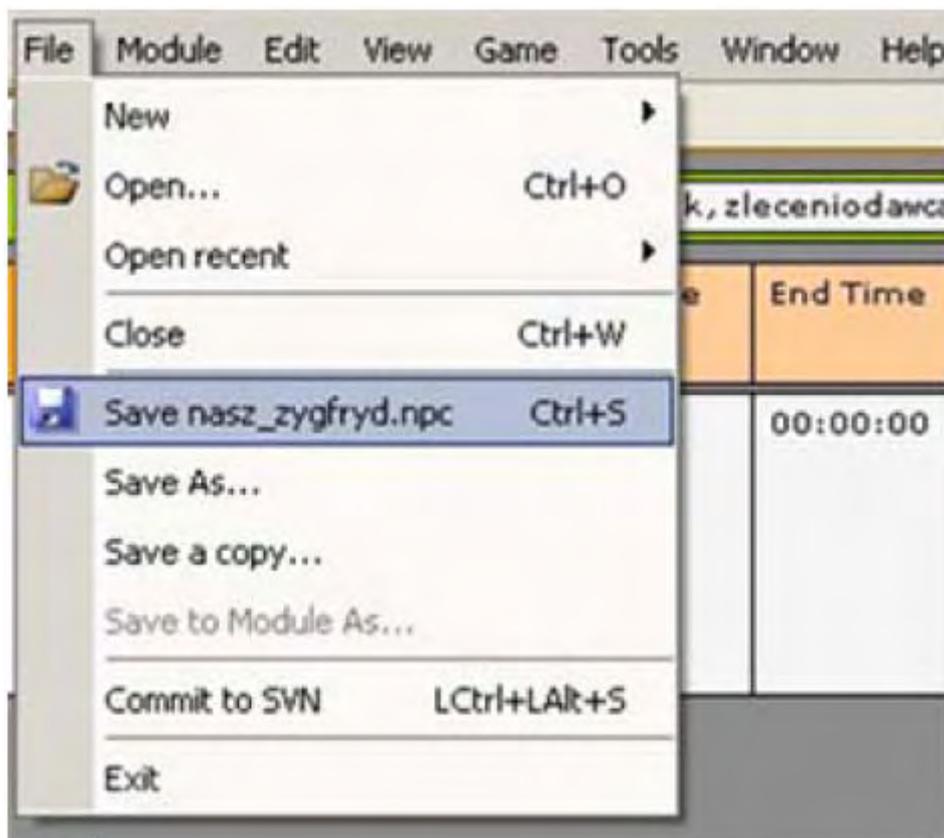
Das weist die neue Unterhaltung Siegfried's Spawn-Set zu:

Dialogs	
id	zygfryd_rozmowa

Nun sollten wir die neue „our_siegfried.npc“ speichern. Wenn eine Datei verändert und noch nicht gespeichert wurde, wird die Datei mit einem * vor dem Namen im Titel markiert:

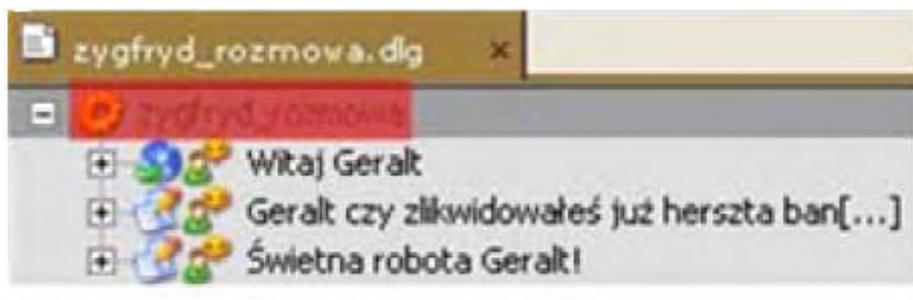
*nasz_zygfryd.npc		
Comment		Zygfryd -
Story Phase	Spawn Phase	St
zlecenie	oczekiwanie	00

Um eine Datei unter demselben Namen zu speichern, zum Beispiel „our_siegfried.npc“, kannst du einfach "Save our_siegfried.npc" aus dem File-Menü wählen:



Wenn du den "Save our_siegfried.npc" Befehl nicht finden kannst, oder ein anderer Name sichtbar ist, kann das bedeuten, dass das "our_siegfried.npc" Fenster nicht aktiv ist. Du kannst es durch klicken auf den Namen des "our_siegfried" Fensters aktivieren. Dann kannst du die Datei our_siegfried.npc speichern.

Um diese Untersektion zu vervollständigen, werde ich noch ein paar weitere Attribute beschreiben. Klicke dazu auf "siegfried_conv" (Wenn du die Konversation nicht gespeichert hast, wird dort "new conversation" stehen):



Die folgenden Attribute werden im Eigenschaften-Fenster auf der rechten Seite erscheinen:

Properties	
Conversation Properties	
Enable cine view on start	True
Disable cine view on end	True
Force Interactive Mode	False
Stats for Lang [FinalEnglish_Short]	
Words	51
Chars	320
Translated	13/13

Conversation Properties Section:

• **Enable cine view on start.** Verantwortlich für die Panorama-Sicht und Text-Box, die vor der Konversation erscheinen. Die Werte, die das Attribut annehmen kann, sind:

False: Keine Panorama-Sicht

True: Panorama-Sicht

• **Disable cine view on end.** Verantwortlich dafür, dass nach der Konversation die Panorama-Sicht und die Text-Box nach der Konversation verschwinden. Das Attribut kann folgende Werte annehmen:

False: Panorama-Sicht verschwindet nicht

True: Panorama-Sicht verschwindet

• **Force Interactive mode**

Stats for Lang Sektion:

Hier sind keine Attribute, aber Statistiken zu finden, Die die Konversation betreffen:

- Words: Die Anzahl der Wörter der Konversation

- Chars: Die Anzahl der Charaktere der Konversation

- Translated: Die Anzahl Nodes, die in einer anderen Sprache geschrieben wurden.

Klicke nun auf einen Eintrag der Conversation, der nicht vom Spieler gesprochen wird. bei uns ist dies Siegfried. Klicke auf "Hallo Geralt":



Es werden andere Attribute im Eigenschaften-Fenster erscheinen:



Node Text	
FinalEnglish_Short	Witaj Geralt
Speaker	__owner__
Interlocutor	
Voice File	
Voiceset tag	---None---
Comment	
Camera Shot	Auto
Response	Default
Action Script	
Conv Type	Quest
JournalId	Q1012_For_sale
Node Emotions	
__player__	
__owner__	
Rotate Towards	
__player__	
__owner__	
After Flags	
Add Flag	
posiada_zlecenie	0

Node Text Sektion:

• **FinalEnglish_Short.** Beinhaltet den Text, den du in dem Eintrag finden kannst. Wir haben den direkt mit einem Doppelklick in dem Eintrag gespeichert, aber hier kannst du das auch tun. Schreibe den Text in das Feld rechts neben "FinalEnglish_Short" und drücke Enter. Der Name dieses Attributs kann sich je nach Version verändern, zum Beispiel wird je nach der Einstellung des Editors dort nicht English sondern German stehen! Mehr dazu steht in einem anderen Kapitel.

• **Speaker.** Legt fest, wem der Eintrag zugewiesen wird. Du kannst folgende Werte zuweisen:

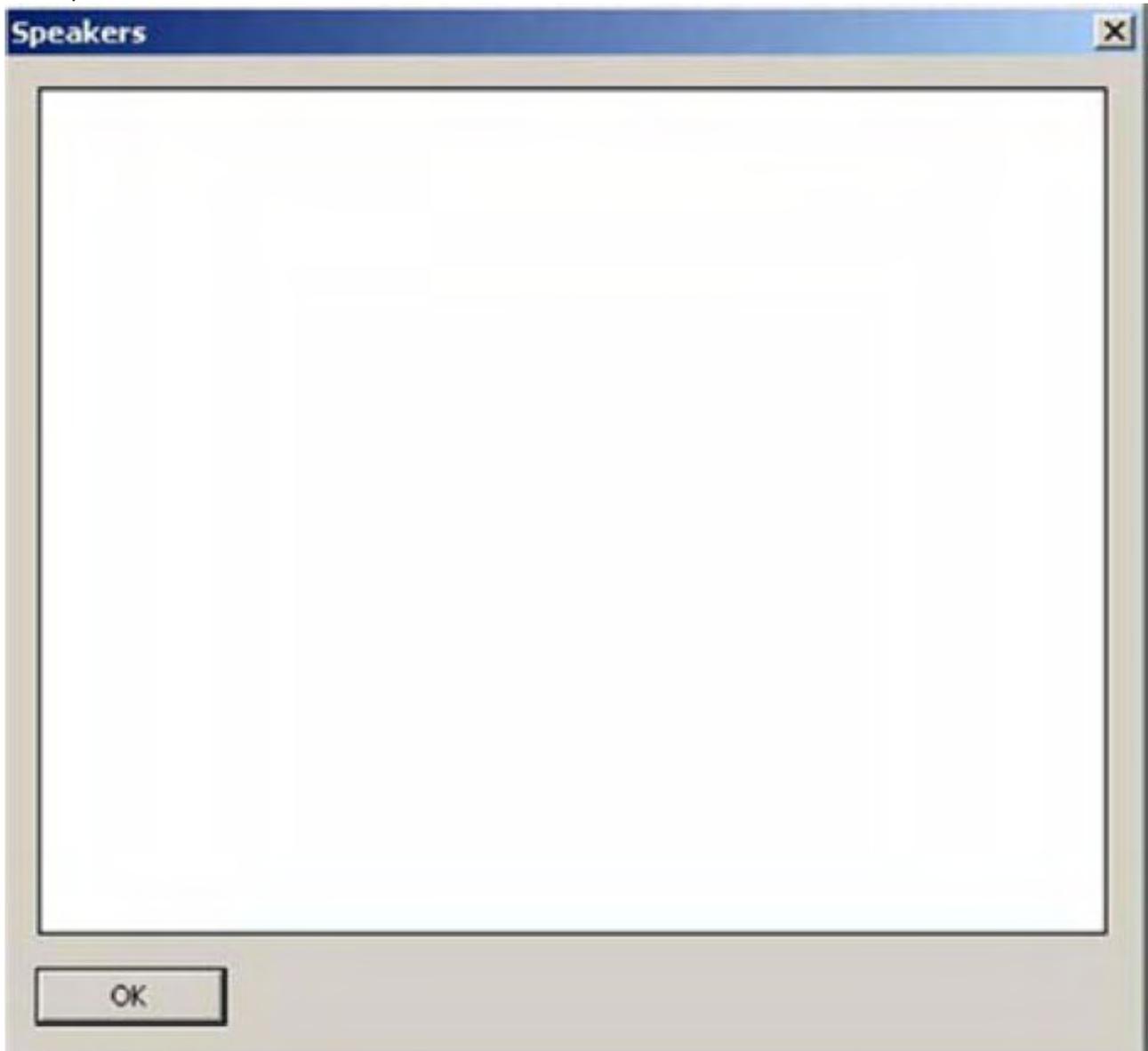
__owner__: Wird von einem anderen Charakter gesprochen.

__player__: Wird vom Spieler, hier Geralt, gesprochen.

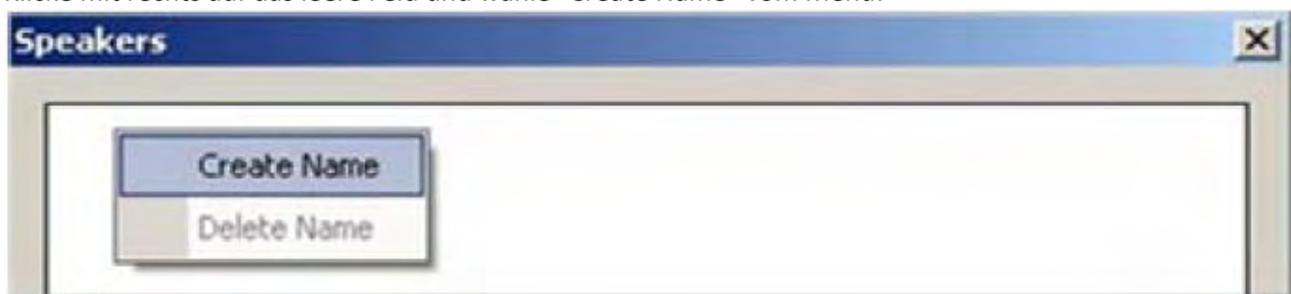
Natürlich können mehr als 2 Charaktere an einer Konversation teilnehmen. Zum Beispiel könnte Jethro zu der Konversation hinzustoßen. Da wir die Konversation nur für Geralt und Siegfried ausgelegt haben, müssen wir Jethro als einen weiteren Charakter hinzufügen. Klicke auf das kleine gelbe Icon rechts neben dem Attribut „Speaker“:



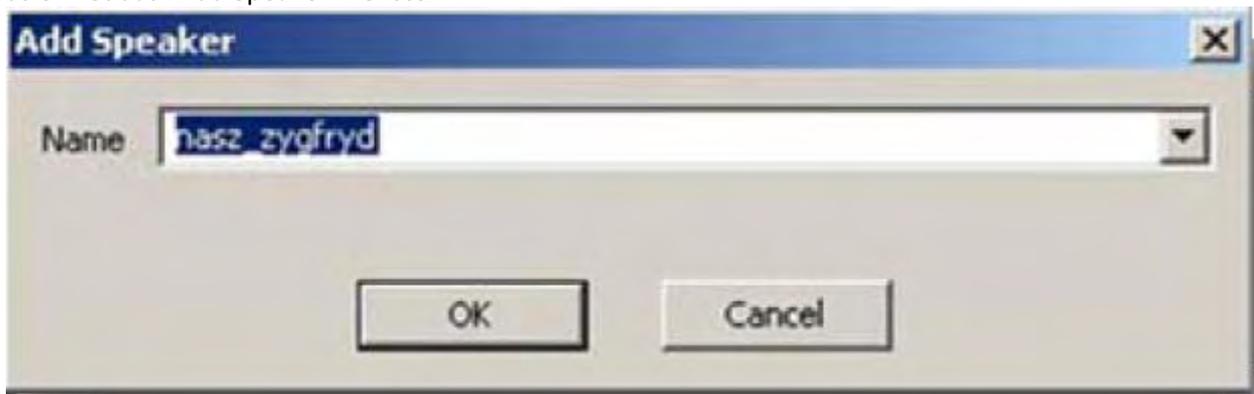
Das "Speaker's" Fenster wird sich öffnen:



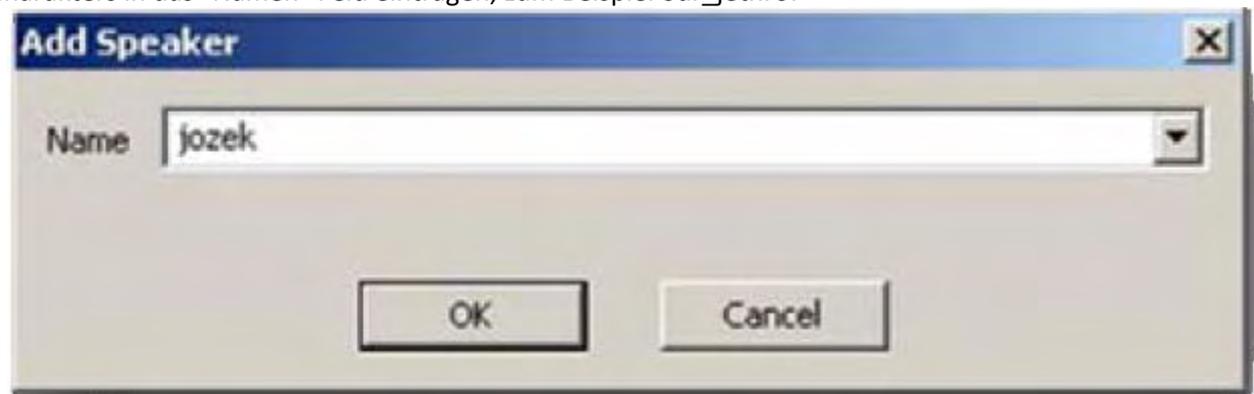
Klicke mit rechts auf das leere Feld und wähle "Create Name" vom Menü:



Das öffnet das "Add Speaker" Fenster:



Du kannst einen weiteren Sprecher auf 2 Arten zu der Unterhaltung hinzufügen. Du kannst den Tag des Charakters in das "Namen" Feld eintragen, zum Beispiel our_jethro:



Oder du wählst ihn aus der Liste aus:



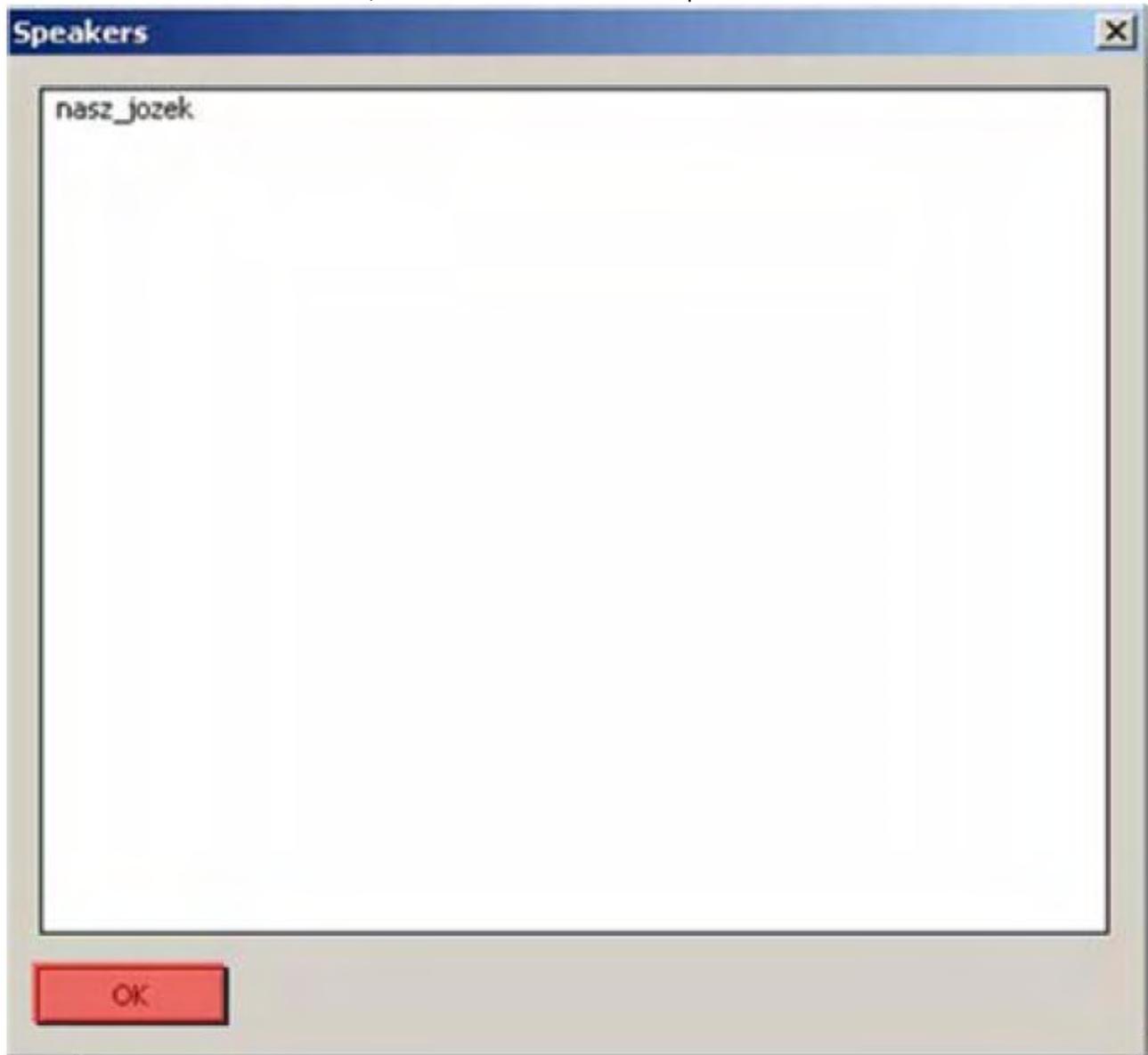
Die Liste enthält nur NPCs mit Tags. Nachdem du den Charakter hinzugefügt hast, klicke auf Ok im "Add Speaker" Fenster. **Jethro** erscheint jetzt in der Liste:



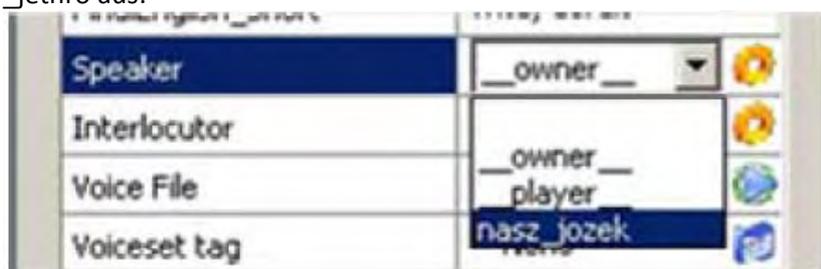
Um einen Charakter zu löschen, klicke mit rechts auf den Namen und wähle "Delete Name":



Da wir Jethro nicht löschen wollen, klicke einfach auf "OK" im "Speakers" Fenster:



Um einen Eintrag einem neuen Charakter zuzuweisen, klicke auf das Feld rechts neben dem Attribut "Speaker" und wähle our_jethro aus:



Dadurch wird eine Konversation mit mehr als 2 Charakteren erzeugt. Wenn nur 2 Charaktere an dieser Konversation teilnehmen sollen, müssen wir an diesem Attribut nichts ändern. Die Nodes werden automatisch in der Reihenfolge, in der Sie erstellt wurden, "_owner_" oder "_player_" zugewiesen.

• **Interlocutor.** Legt fest, an wen der Charakter seinen Text richtet. Du kannst hier die folgenden Attribute zuweisen:

owner: Der Charakter spricht sich selbst an.

owner: Der Charakter spricht den Spieler an.

Bei Konversationen mit 2 Charakteren müssen wir hier nichts einstellen. Sind mehr als 2 Personen beteiligt, kann man hier den Empfänger einstellen.

Sollten die Charaktere einen anderen Charakter ansprechen muss der wie beim "Speaker" Attribut noch hinzugefügt werden. Dies funktioniert auf die gleiche Weise wie beim "Speaker" Attribut.

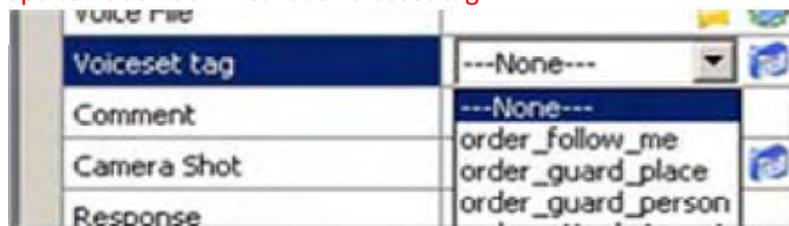
• **Voice File.** Hier könnt ihr eine Sound-Datei einem Eintrag zuweisen. Um dies zu tun, klicke auf das gelbe Icon neben "Voice File":



Dies öffnet das Fenster zum Laden von Dateien. Suche die Datei mit der passenden Stimme und klicke auf OK. Um dir die Stimme anzuhören, klicke auf das blaue Icon neben "Voice File":



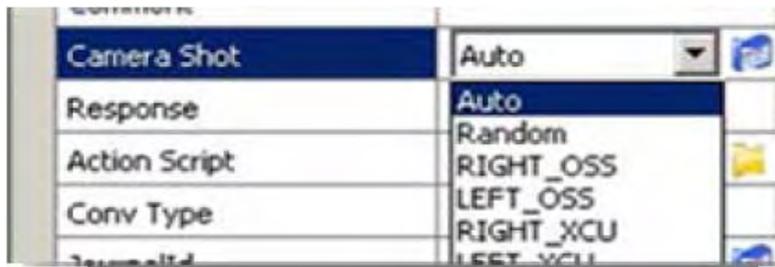
• **Voiceset tag.** Zuständig für das Voiceset, das während eines Dialogs im Hintergrund abgespielt wird. Das kann z. B. das Geräusch von Regen sein, das während eines Gesprächs im Hintergrund zu hören ist. Du kannst aus einer Liste vorgegebener Geräusche etwas Passendes auswählen. Um die Einstellung zu ändern, klicke auf die rechte Spalte neben dem Attribut Voiceset tag:



Ein Klick auf das kleine blaue Symbol neben der Auswahlliste öffnet die Datei voiceset_tags.2da. Dies ist die Datenbank, die die Informationen über vordefinierte Geräusche enthält.

• **Comment.** Ein Feld, in das Du einen Kommentar schreiben kannst. Dieser Text wird vom Spiel nicht verwendet, er dient nur zur Information. Du kannst hier etwas Beliebiges eintragen.

• **Camera Shot.** Zuständig für den Kamerawinkel während des Dialogs. Die Kamera kann für jeden Knoten des Dialogs unterschiedlich positioniert werden. Du kannst die Einstellung aus einer Liste auswählen. Um die Einstellung zu ändern, klicke auf die rechte Spalte neben dem Attribut Camera Shot:



Die Werte, die dieses Attribut annehmen kann, sind:

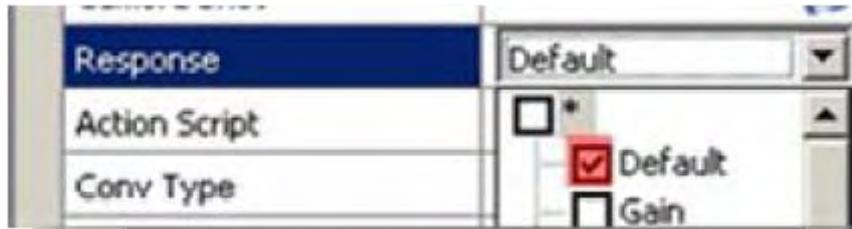
- Auto - Standardeinstellung. Dies ist die empfohlene Einstellung.
- Random - zufällige Einstellung.
- NoChange - Die Kamera wird nicht positioniert.

Die Optionen des Attributs können verwendet werden, um die Kamera an verschiedenen Positionen zu platzieren. Die empfohlene Einstellung ist Auto, außer, die Kamera ist während des Gesprächs ungünstig positioniert. Ein Klick auf das kleine blaue Icon neben der Auswahlliste öffnet die Datei dlcamshots.2da. Dies ist die Datenbank, die die Einstellungen für die jeweiligen Kamerawinkel enthält.

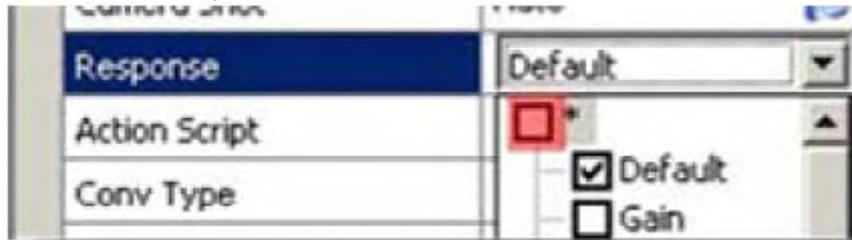
• **Response.** Zuständig für die Reaktion des Charakters, mit dem Geralt sich gerade unterhält. Zum Beispiel können wir festlegen, dass Geralt nur dann eine Quest von Siegfried bekommt, nachdem er Siegfried einen Ring gegeben hat. Oder wir können bestimmen, dass Siegfried nicht mit Geralt reden will, weil er gerade beschäftigt ist:



Setze dieses Attribut, auf diesem kleinen Kasten neben dem Namen der „desired reaction“:



Das klicken auf eine der vorher markierten Reaktionen wird gekennzeichnet um alle Reaktionen sofort setzen zu können (Wird aber nicht empfohlen), klicke dazu auf dieses kleine Kästchen neben dem Sternchen:



Die Werte, die du zur dieser Auswahl zuteilen kannst, sind:

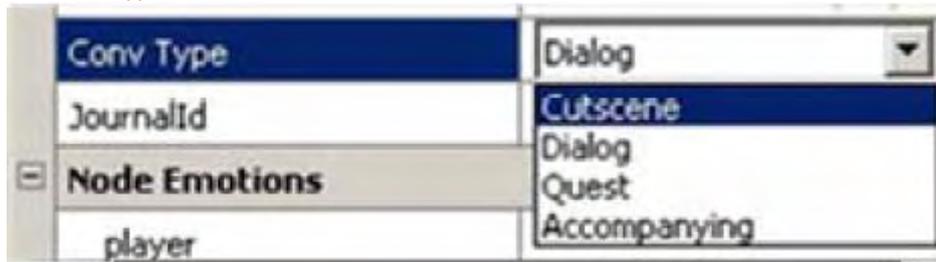
- *Default* – Standardname, ein Charakter verlangt nichts
- *Gain* – der Charakter will etwas gewinnen,
- *Gain / Signet* – der Charakter will einen Siegelring,
- *Gain / Drink* – der Charakter will etwas Alkohol,,
- *Gain / Bribe* – der Charakter will Bestechungsgeld,
- *Gain / Gift* – the Charakter will ein Geschenk,
- *Gain / Fistfight* – der Charakter will einen Faustkampf haben,
- *Reject* – der Charakter will nicht sprechen,
- *Reject / Busy* – der Charakter will nicht sprechen, weil er beschäftigt ist,
- *Reject / Drink* – der Charakter will nicht sprechen, weil er betrunken ist,
- *Reject / Offend* – der Charakter will nicht sprechen, weil er verletzt wurde
- *Reject / Sleep* – der Charakter will nicht sprechen, weil er schläft,
- *Reject / Fistfight* – der Charakter will nicht sprechen, weil er kämpft,
- *Reject / Gift* – der Charakter will nicht sprechen, weil er ein Geschenk nicht erhalten hat.

• **Action Script.** Verantwortlich für ein Script, die nach einer besonderen Conversation aktiviert wird. Um einen Script zuzuteilen, klicke auf das kleine gelbe Icon in der rechten Spalte gegenüber vom „Action Script“:



Das öffnet das normale Windows-Fenster, das verwendet wird, um Dateien zu laden. Gehe dazu zum Data\Scripts Ordner (der alle Script-Dateien enthält). Suche dann nach der erforderlichen Datei und klicke auf öffnen. Mehr über Scripte wird in einem anderen Kapitel dieses Handbuchs beschrieben.

• **Conv Type.** Verantwortlich dafür, den Typ dieses Gespräches zu setzen. Das wird allgemein automatisch je nachdem gesetzt, ob es eine Quest oder Standardgespräch ist, jedoch kann es notwendig sein, dieses Attribut zu ändern, zum Beispiel um eine Quest-Conversation zu kennzeichnen, dass ein besonderer Conversations Eintrag Für Geralts Auswahl ist. Um die Einstellung zu ändern, klicken dazu auf die rechte Spalte gegenüber vom „Conv Typ“:

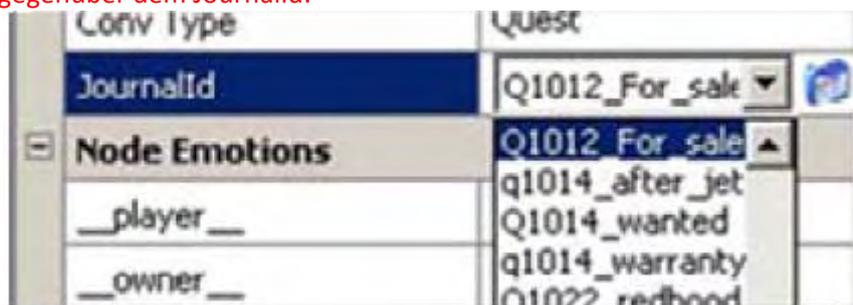


Die Werte, die Du bei dieser Auswahl zuteilen kannst, sind:

- *Cutscene* –eine Cutscene Gesprächs Eintrag,
- *Dialog* – ein Standard Eintrag,
- *Quest* – ein quest Eintrag,
- *Accompanying* – Der Quest Eintrag ist eine Option für Geralt, Dem Spieler.

Es gibt mehr Optionen auszuwählen, nämlich das Gameplay. Das ist nur verfügbar, wenn ein Conversations Eintrag durch Geralt gesprochen wurde . Mehr darüber wird im folgenden Unterkapiteln beschrieben.

• **JournalId.** Dieses Attribut enthält Informationen, die im Journal von Geralt erscheinen werden. Das könnte zum Beispiel ein Eintrag von Siegfried oder Jethro sein. Um diese Einstellung zu ändern, klicke dazu auf die rechte Spalte gegenüber dem JournalId:



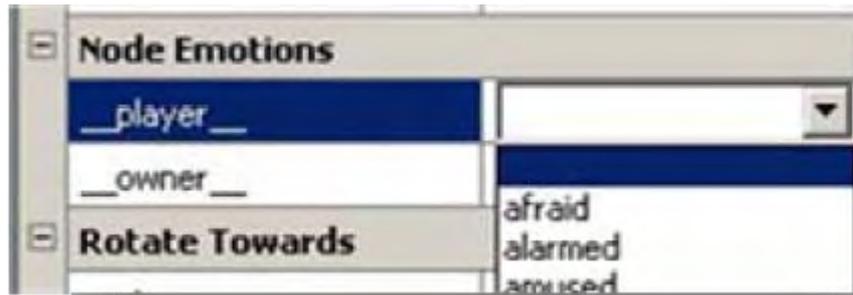
Das Klicken auf eines der blauen Icons gegenüber der choice list wird die journal.2da Datei eine Datenbank öffnen, die Information über Einträge im Journal des Spielers enthält.

Node Emotions Section:

Die Attribute dieser Abteilung sind für die Gefühle verantwortlich, die durch die Charaktere während eines besonderen Conversations Eintrag ausgedrückt sind. Gefühle sind einfach eine Reihe von Animationen, die während eines Conversations Eintrag angezeigt werden.

- **__player__.** Verantwortlich für Geralt, d. h. die Gefühle des Spielers.
- **__owner__.** Verantwortlich für Gefühle des Charakters, wem das Gespräch (in unserem Fall, Siegfried) zugeteilt wurde.

Um die Einstellung zu ändern, klicke dazu auf die rechte Spalte gegenüber vom `__player__` oder `__owner__` Attribute:



Die Werte, die Du bei dieser Auswahl zuteilen kannst, sind:

- *empty* – keine Gefühle,
- *afraid* – ängstlich,
- *alarmed* – alarmiert,
- *amused* – amüsiert,
- *amused_kid* – witzbold,
- *Angry* – böse,
- *armed* – bewaffnet,
- *bored* – gelangweilt,
- *default* – default,
- *despaired* – verzweifelt,
- *drunk* – Trinker,
- *excited* – gereizt
- *fistfight* –Faustkämpfer,
- *flirting_female* – Sexy Frau,
- *friendly* – Freund,
- *hostile* – feindlich,
- *hostile_female* – feindliche Frau,
- *interested* – interessiert,
- *Lofty* – arrogant,
- *Pain* – leidend,
- *Sad* – traurig,

- *Shy* – scheu,
- *Terrified* – entsetzt,
- *Timid* – zaghaft,
- *Tired* – ermüdet,
- *Witcher* – Witcher (Hexer).

Rotate Towards Section:

Attribute in dieser Abteilung sind für die Positionierung von Charakteren verantwortlich, die an einem Gespräch teilnehmen, d. h. wer auf wen während des Gespräches schauen soll. In einem Gespräch zwischen gerade zwei Charakteren müssen wir nichts ändern. Aber wenn mehr als zwei Charaktere teilnehmen, können wir diese Attribute verwenden, um die Richtung zu setzen, in der sie schauen sollen.

- **__player__**. Enthält den Charakter, auf wem der Spieler schaut, indem er ein besonderen Conversations Eintrag spricht,
- **__owner__**. Enthält den Charakter, dem ein besonderes Gespräch (in unserem Fall, Siegfried) gerichtet wurde.

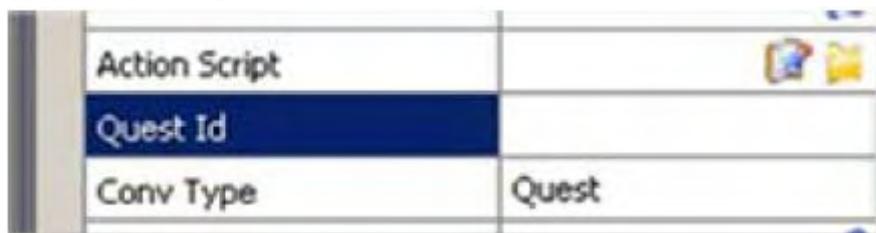
Wenn mehr als zwei Menschen an einem Gespräch teilnehmen, muss der Rest hinzugefügt werden. Das wird zusammen mit dem Autor-Attribut beschrieben.

After Flags Section:

Die Details Nach der Flag Section sind vorher beschrieben worden. Unter Verwendung dieser Gruppe können wir Markierungsfahnen mit den gesetzten Werten einem bestimmten Gesprächseintragungspunkt hinzufügen Die Attribute der Quest-Conversation-Nodes von Geralt unterscheiden sich von anderen Charakteren auf nur zwei verschiedene Weisen. Klicken dazu auf " Hallo Siegfried. Hast du neue Quests für mich?" Eintrag:

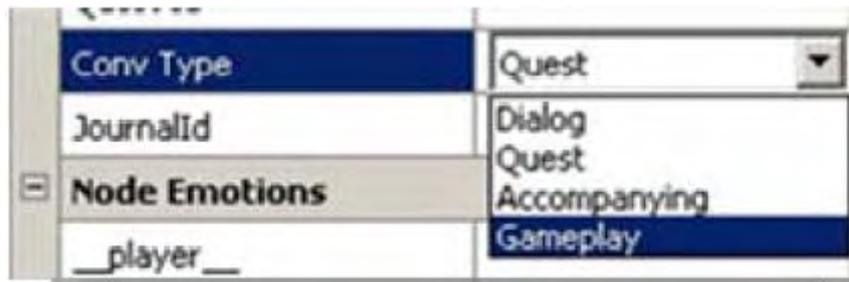


Wenn wir einen näheren Blick auf dieses Attribute werfen, werden wir sehen, dass dieses Ansprechattribut durch die „Quest Id“ Attribut ersetzt worden ist:

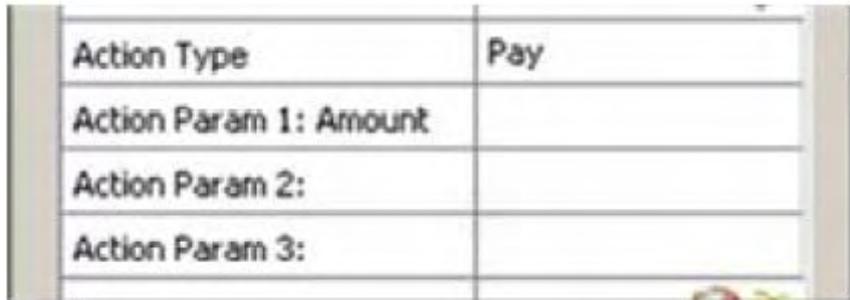


Wie früher erklärt ist dieses Attribut für die Quest Id verantwortlich.

Der andere Unterschied zwischen den Quest-Gesprächen von Geralt und anderer Charaktere ist, dass eine zusätzliche Auswahl unter dem „Conv Typ“ verfügbar ist:



Nach dem Setzen des Gameplay Auswahl im Conv Typ-Attribut werden mehrere andere Attribute erscheinen:

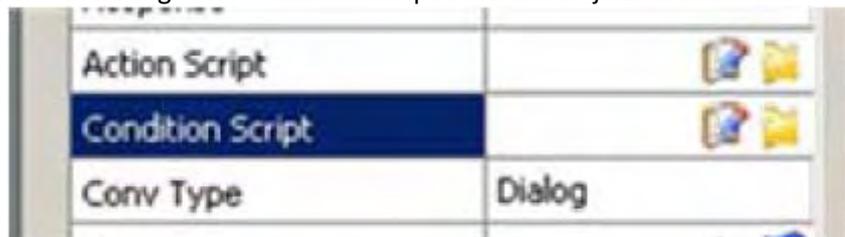


Diese werden im Detail im folgenden Unterkapiteln beschrieben, wenn wir besprechen, wie man einen Gameplay erstellt.

Klicken jetzt auf ein Conversations Eintrag, der durch einen Charakter ausgesprochen ist, außer von Geralt, aber dieses Mal auf einem Standardgespräch.



Hier gibt es auch eine Änderung – der Condition Script attribute ist jetzt erschienen:

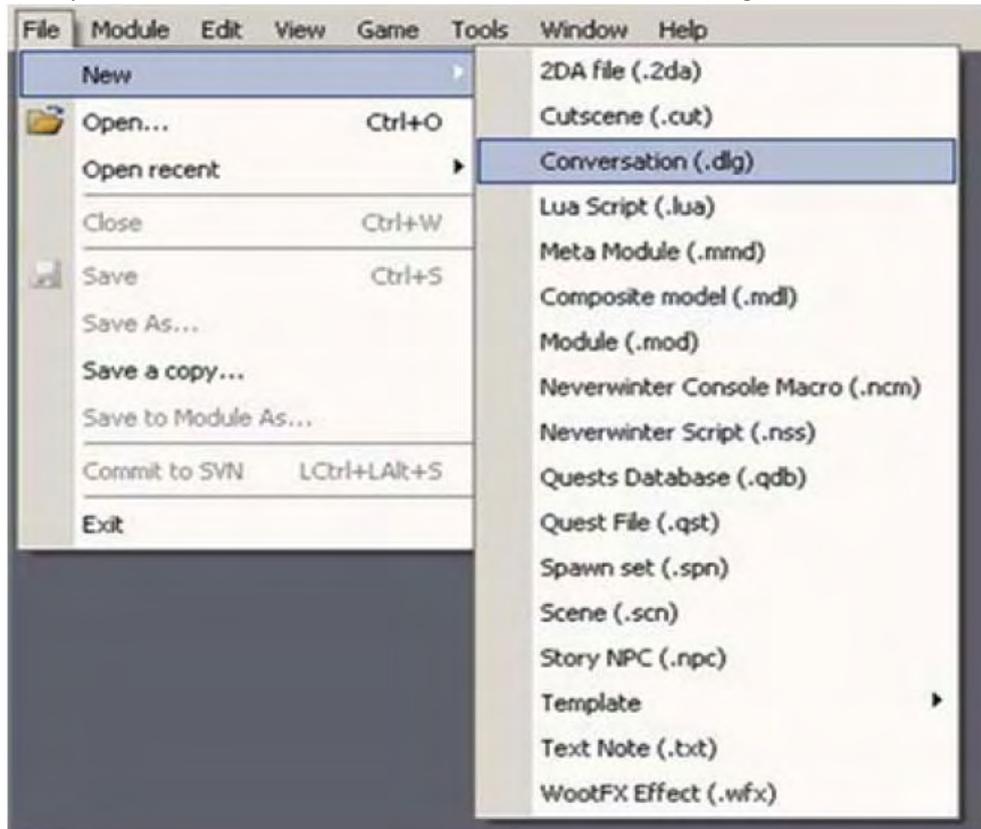


Eine Actionscript zugeteilter Script wird aktiviert, kurz nachdem ein Conversations Eintrag beendet wurde wird dann ein Conditions Script aktiviert bevor ein besonderer Conversations Eintrag stattfinden wird. Das kann dann verwendet um etwa zu überprüfen, ob irgendwelche zusätzlichen Bedingungen erfüllt wurden.

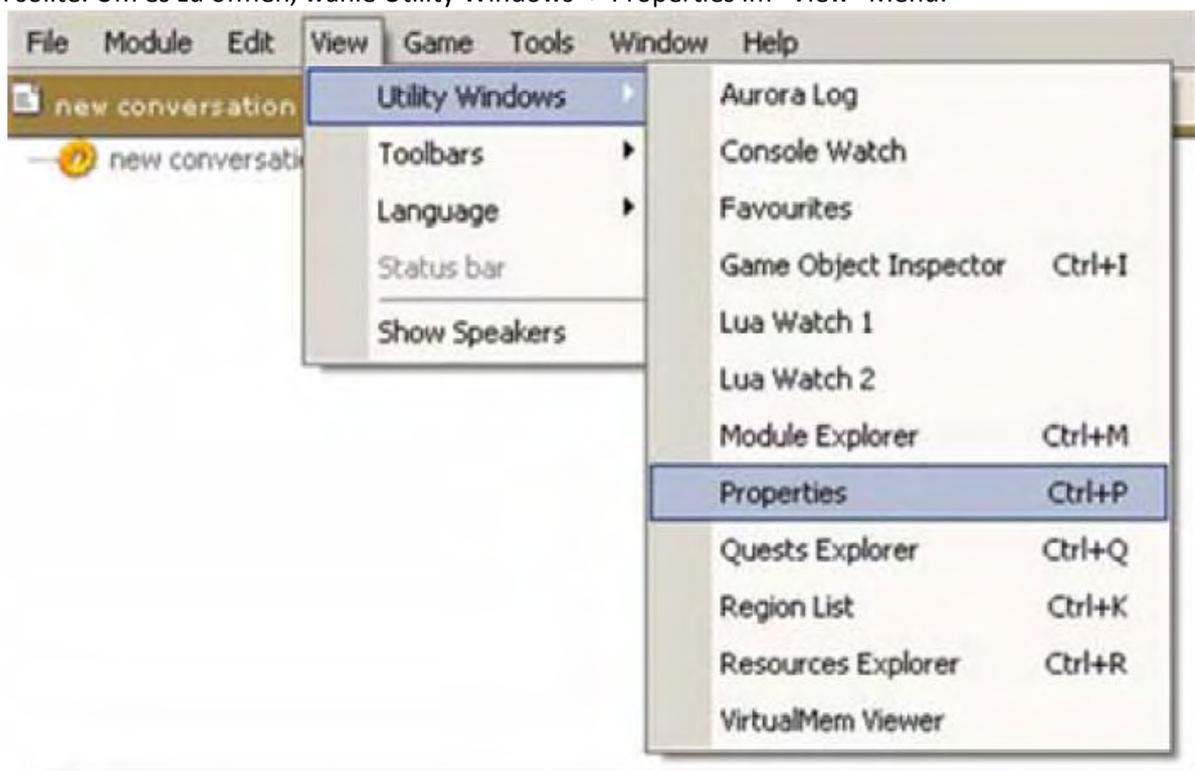
So ist's recht!. Wenn es zum Conversations-Attribut kommt.

ENTERTAINMENT IN CONVERSATIONS

Jetzt werden wir uns damit befassen, wie man Gespräche für Jethro erstellt. Als eine Gedächtnishilfe ist Jethro nicht ein NPC. Er spielt gerne würfeln, so in seinen Gesprächen werden wir die Möglichkeit der Spielwürfe hinzufügen. Deshalb wird Geralt (der Spieler) im Stande sein, Spielwürfel mit Jethro zu spielen. Um ein neues Gespräch zu schaffen, wählen dazu New-> Conversation (.dlg) vom Menü File aus:



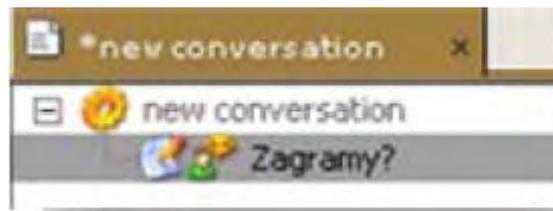
Es wird sich ein neues "Conversation"- Fenster öffnen. Denk dran, dass das Eigenschaften-Fenster offen sein sollte. Um es zu öffnen, wähle Utility Windows -> Properties im "View" Menü:



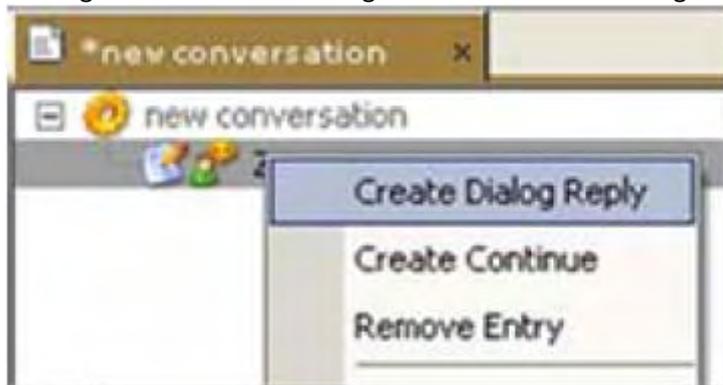
Nun werden wir uns eine Standard-Unterhaltung für Jethro erstellen. Wähle "New Conversation" und klicke mit rechts. Wähle dann "Create Dialog Node":



Klicke doppelt auf den leeren Eintrag, den wir grade hinzugefügt haben, schreibe "Werden wir spielen?" und drücke dann auf ENTER:



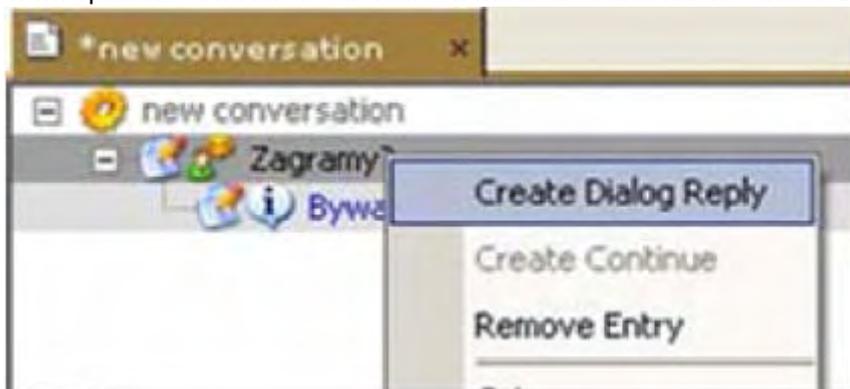
Klicke nun mit rechts auf den gerade erstellten Eintrag und wähle "Create Dialog Reply" aus:



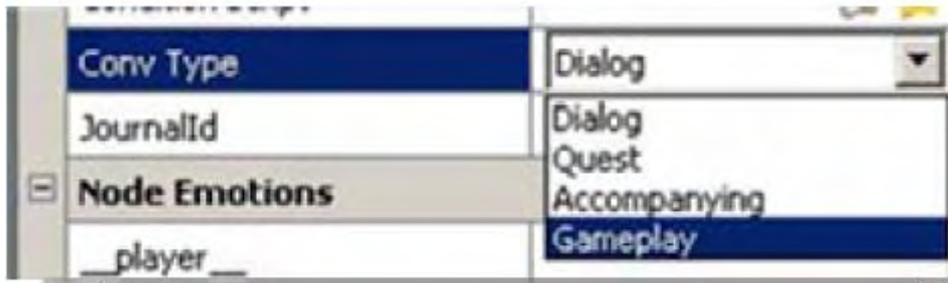
Klicke nun auf den neuen Eintrag und schreibe "Leb wohl". Und Drücke dann auf ENTER:



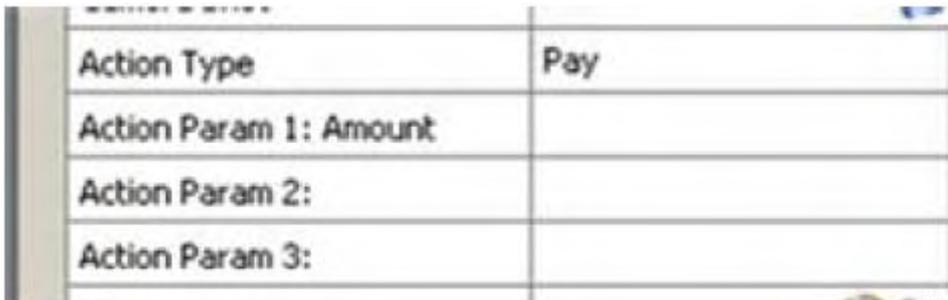
Selektiere „Werden wir spielen?“ Und klicke mit rechter Maustaste und wähle "Create Dialog Reply" aus:



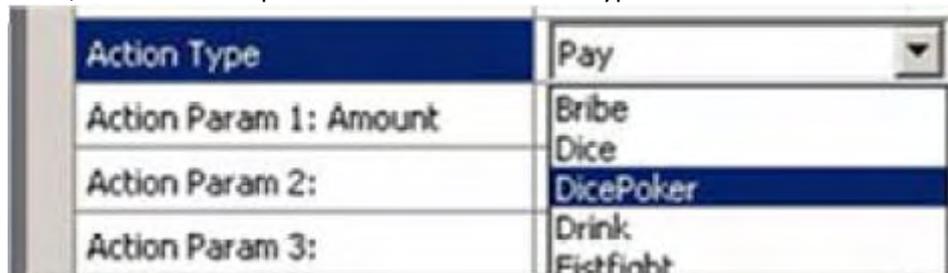
Klicke nun rechts auf den leeren Eintrag und gehe in das Eigenschaften-Fenster. Setze das "Conv Type" Attribut in der "Node Text" Sektion auf *Gameplay*.



Es sind ein paar neue Attribute erschienen:



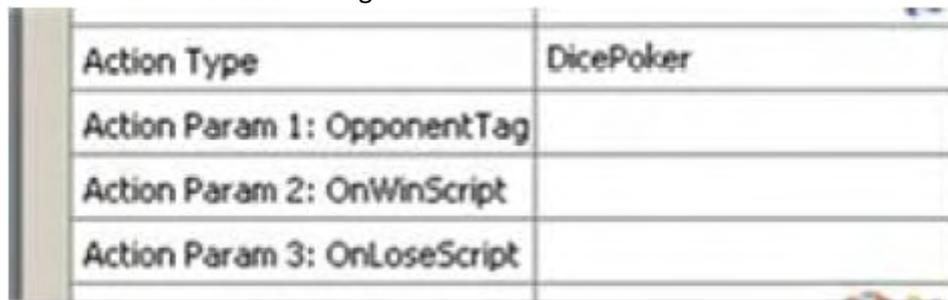
Jehro spielt gerne Poker, also setzen wir das "Action Type" Attribut auf "DicePoker". Um den Wert des Attributs zu ändern, klicke auf die Spalte rechts neben "Action Type" und wähle "DicePoker":



Der leere Eintrag in unserer Conversation hat automatisch seinen Namen in "Let's play Poker!" geändert:



Nun wird während der Unterhaltung mit Jehro die Option erscheinen, Poker zu spielen. Achtung! Die Werte der „Action Param“ Attribute haben sich geändert:



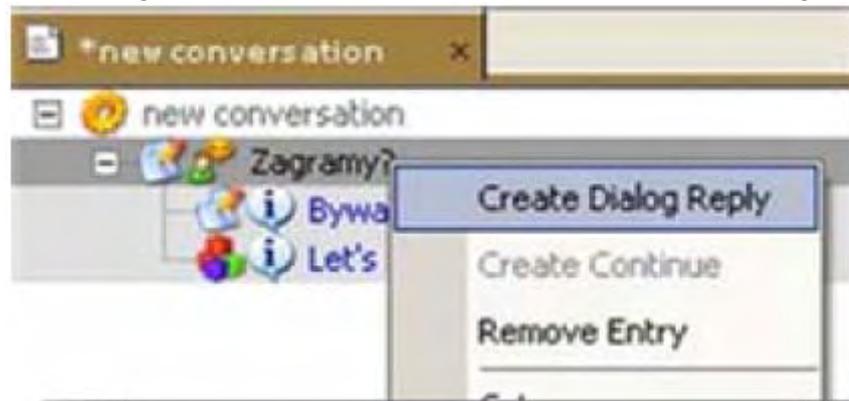
Hier müssen wir nichts mehr ändern. Der Sinn der Attribute ist Folgender:

Action Param 1: OpponentTag. Beinhaltet den Tag des Charakters, mit dem wir Poker spielen.

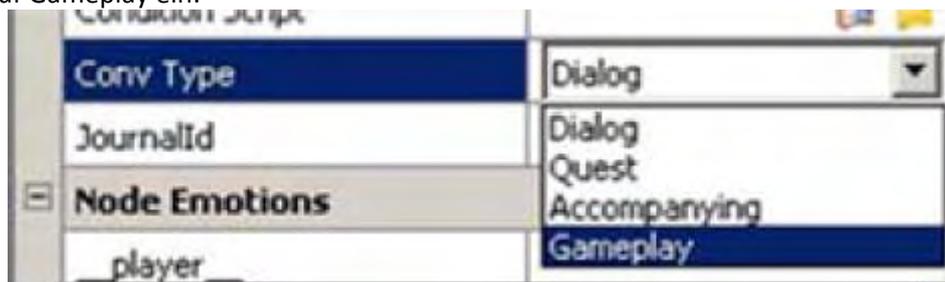
Action Param 2: OnWinScript. Beinhaltet das Script, das aktiviert wird, wenn Geralt gewinnt.

Action Param 3: OnLoseScript. Beinhaltet das Script, das aktiviert wird, wenn Geralt verliert.

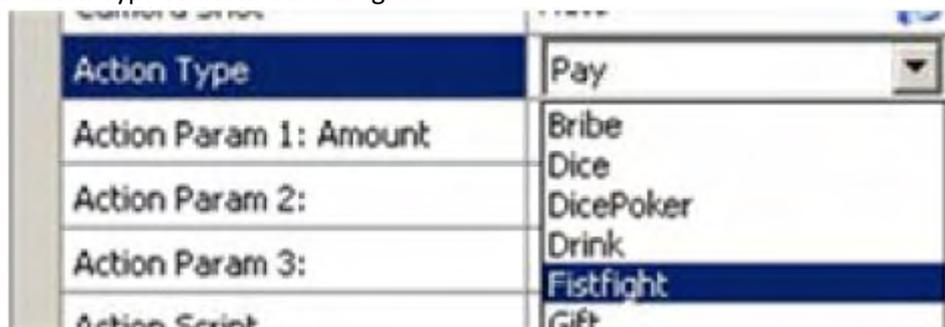
Das Auswählen der Gameplay Auswahl unter dem Conv Typ-Attribut entscheidet unter verschiedenen Unterhaltungen, für die sich ein besonderer Charakter interessiert. Im Fall von Jethro ist das dass Würfeln. Wollen wir das in die Auswahl hinzufügen, um einen Faustkampf mit Jethro zu haben. Wähle dazu denn "Werden wir spielen?" Eintrag und rechtsklicke. Dann wählen dazu die „Create Dialog Reply“ vom Menü:



Wähle aus und rechtsklicke auf den leeren Eintrag, den wir gerade hinzugefügt hatten. Stelle jetzt die Conv Art Attribut auf Gameplay ein:



Setze jetzt das Action Type-Attribut auf Fistfight:



Im Falle von Fistfight der Handlungen sind Param Attribute verschieden:

- *Action Param 1: Min_Gold* – setzt den minimalen Anteil
- *Action Param 2: Max_Gold* – setzt den maximalen Anteil
- *Action Param 3: Waypoint_Prefix* – setzt den Wegpunktenprefix das notwendig ist den Weg einen Charakters zu erstellen.

Unser Gespräch änderte sich jetzt auch. Der leere Conversations Eintrag hat sich geändert in: "Show me what you've got!":



Geralt wird im Stande sein, Würfeln während seines Gespräches mit Jethro zu wählen. Er wird auch im Stande sein, in einen Faustkampf mit ihm zu kommen oder einfach abzureisen. Bevor wir aber fortsetzen, werden die anderen Optionen des Action Type Attribute zusammen mit der verschiedenen Handlung Param Attribute beschrieben:

- **Bribe.** Geralt kann einen Charakter bestechen, um etwas Information zum Beispiel zu bekommen: - Handlung Param 1: Betrag - setzt die minimale Summe von Orens (die im Witcher verwendete Währung), dass Geralt in der Größenordnung vom Bestechungsgeld geben muss, um zu arbeiten.

- **Dice.** Nicht benutzbar.

- **DicePoker.** Das Würfel-Spiel ist oben beschrieben worden.

- **Drink.** Geralt kann Alkohol trinken:

- Action Param 1: *Fail_Script* – enthält den Script, das aktiviert wurde, wenn Geralt leichtsinnig und schwummerig in Folge des zu viel Trinkens wird.

- **Fistfight.** Das Faust-Kämpfen ist oben beschrieben worden.

- **Gift.** Geralt kann einem anderen Charakter ein Geschenk geben:

- Action Param 1: *Gift_Tag* – enthält das Anhängsel des Geschenks, das der Charakter von Geralt erhalten will,

- Action Param 2: *Bad_Gift_Tag* – enthält das Anhängsel eines Geschenks, das der Charakter ablehnen wird,

- Action Param 3: *Unknown_Gift_Script* – enthält den Script, die aktiviert wird, wenn Geralt versucht, dem Charakter ein unbekanntes Geschenk zu geben,

- **Pay.** Geralt kann einen Charakter für etwas Information bezahlen oder ihm einfach ein Bestechungsgeld geben: - Handlung Param 1: Betrag - setzt die minimale Summe von Orens (die im Witcher verwendete Währung), dass Geralt geben muss.

- **Signet.** Geralt kann einem Charakter einen Siegelring geben:

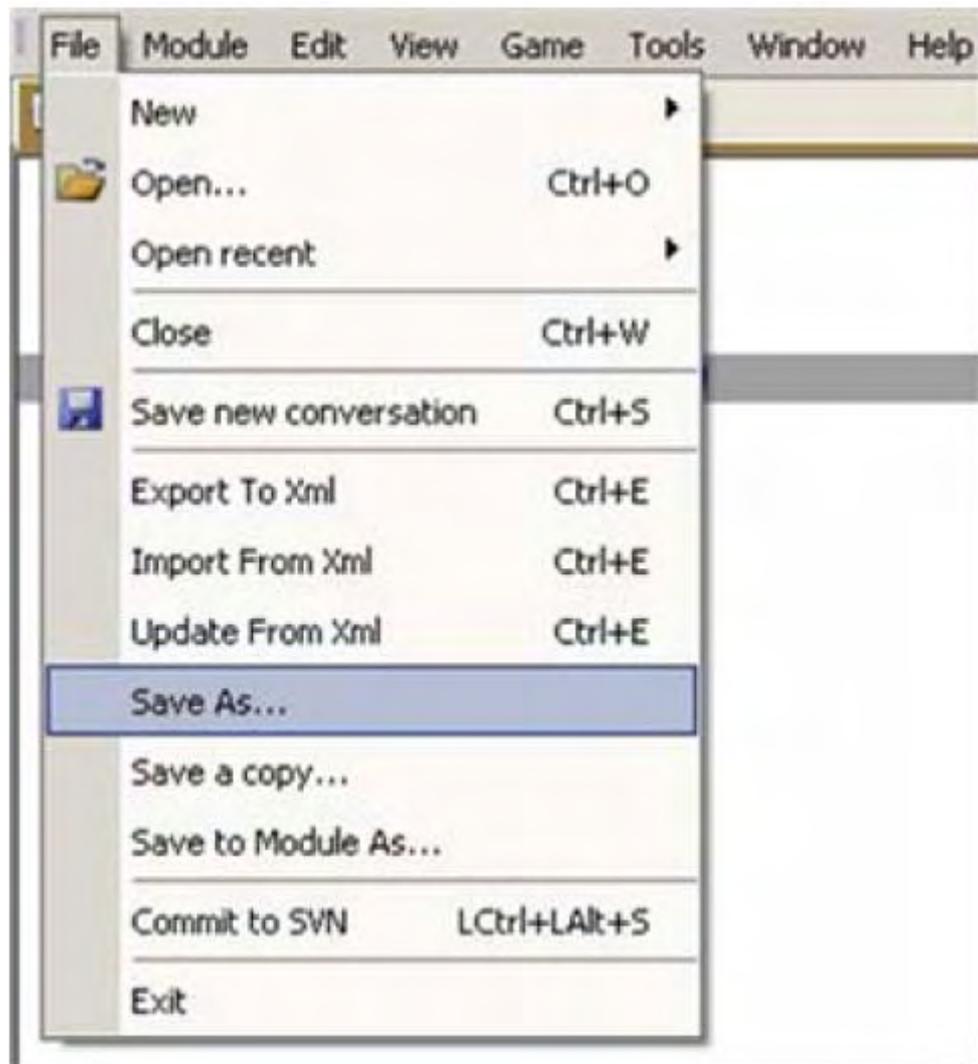
- Action Param 1: *Signet_Tag* – enthält das Tag des Siegelrings, den Geralt geben muss,

- Action Param 2: *Unknown_Signet_Script* – enthält den Script, die aktiviert wird, wenn Geralt versucht, dem Charakter ein unbekanntes Siegel zu geben,

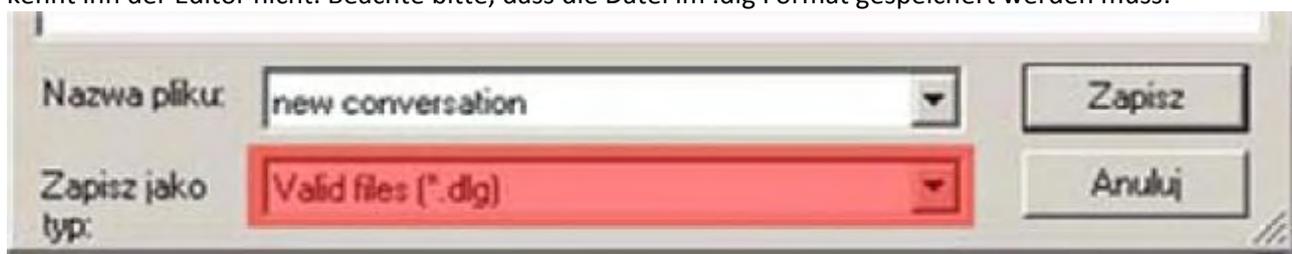
- **Sleep.** Geralt kann einen Charakter bezahlen, so dass er etwas Schlaf bekommen kann: - Handlung Param 1: Preis - enthält den Preis, den Geralt zum Schlafen bezahlen muss.

- **Store.** Geralt (der Spieler) Lager. Das ist, wo er unnötige Dinge versorgen kann. Mehr darüber ist in einem anderen Kapitel beschrieben worden.
- **SwordUpgrade.** Der Charakter hat eine Schmiede. Mehr darüber ist in einem anderen Kapitel besprochen worden.
- **Trade.** Der Charakter hat ein Geschäft. Mehr darüber ist in einem anderen Kapitel beschrieben worden.

Gehen wir zurück zu Jethro und seinem Gespräch und speichern wir es jetzt. wähle „Save as...“ vom Menü „File“:



Dadurch öffnet sich das Standard-Fenster zum Speichern von Dateien. Konversationen müssen im "Dialogues Ordner" gespeichert werden. Begib dich in den Ordner, gebe den Namen jethro_conv" ein und klicke auf "Save". Denk dran, dass der Name der Datei nicht mehr als 16 Zeichen lang sein darf, denn sonst erkennt ihn der Editor nicht. Beachte bitte, dass die Datei im .dlg Format gespeichert werden muss:



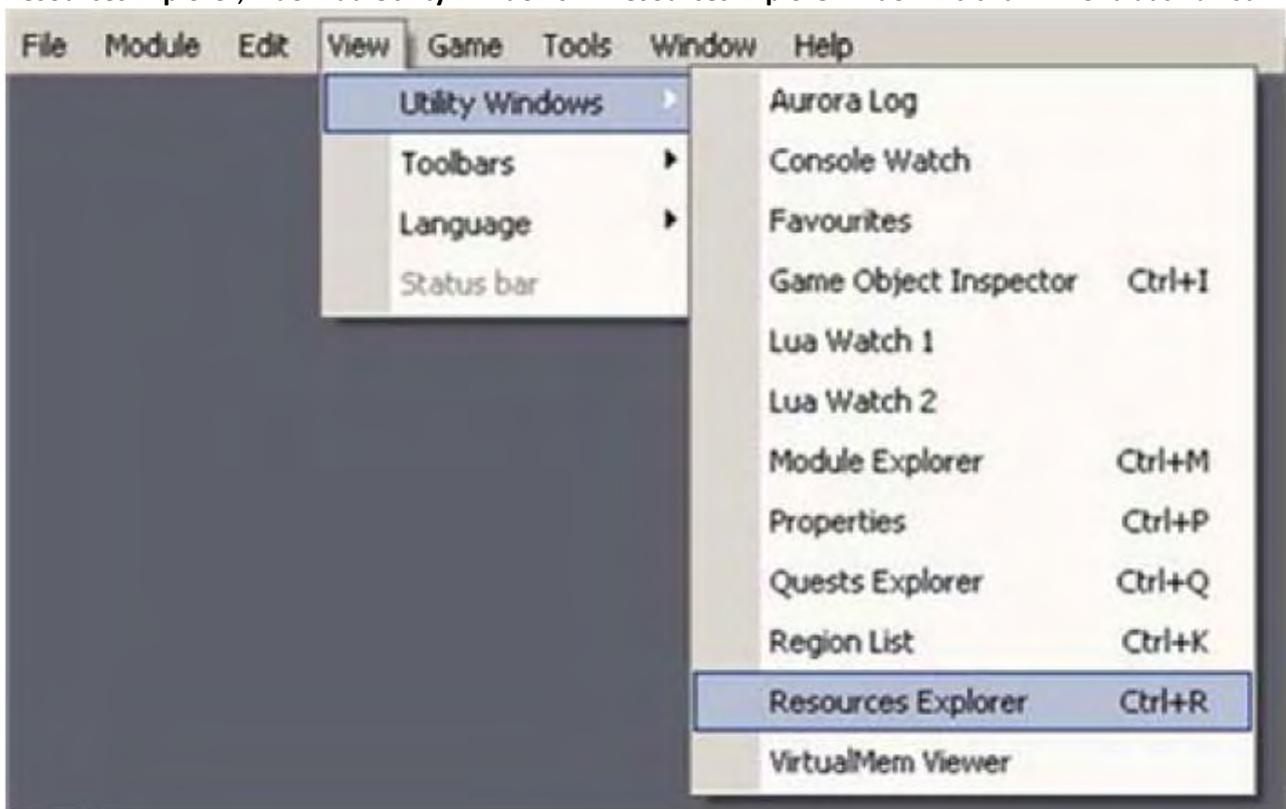
Wenn dort ein anderes Format angezeigt wird, bist du dabei, den Inhalt aus einem anderen aktiven Fenster zu speichern. Sei dir also sicher, dass "New Conversation" Fenster aktiv hast, bevor du speicherst. Ob das Fenster aktiv ist, erkennst du an der Titel bar. Ist sie grau, ist das Fenster inaktiv:



Hat es eine andere Farbe, ist es aktiv:



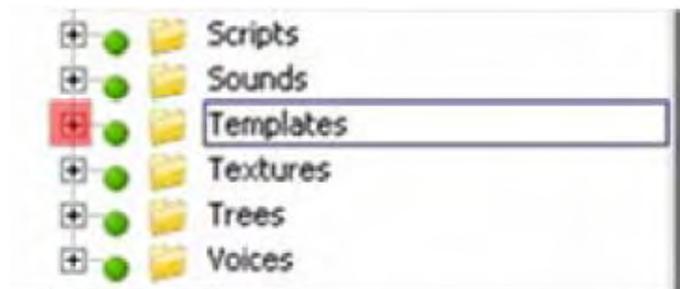
Ein Sternchen vor dem Namen in der Titelseite bedeutet, dass die Datei, seit der letzten Änderung, noch nicht gespeichert wurde. Wir müssen nun das Gespräch noch Jethro zuteilen. Weil er ein Commoner und kein NPC ist, werden wir jetzt dieses Gespräch der Vorlage (template) von Jethro zuteilen. Öffne dafür den **Resources Explorer**, indem du **Utility Windows** -> **Resources Explorer** in der Ansicht im Menü auswählst:



Das **Ressourcen Explorer** Fenster wird sich auf der linken Seite öffnen:



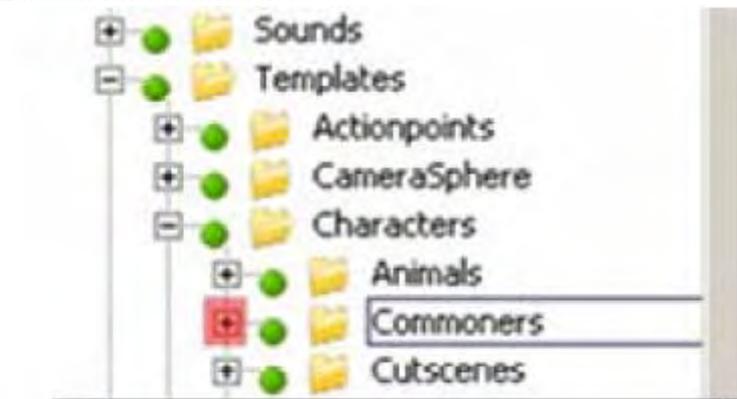
Wir speicherten die Vorlage von Jethro im *Data\Templates\Charakters\Commoners* Ordner unter dem Namen *our_jethro.utc*. Öffne die Verzweigung im Verzeichnisbaum, indem du auf das kleine "+" Zeichen neben dem Namen des Ordners klickst:



Klicke jetzt auf das kleine "+" Zeichen vor *Characters* um in das Unterverzeichnis zu gelangen:



Jetzt noch auf das "+" Zeichen vor *Commoners*:



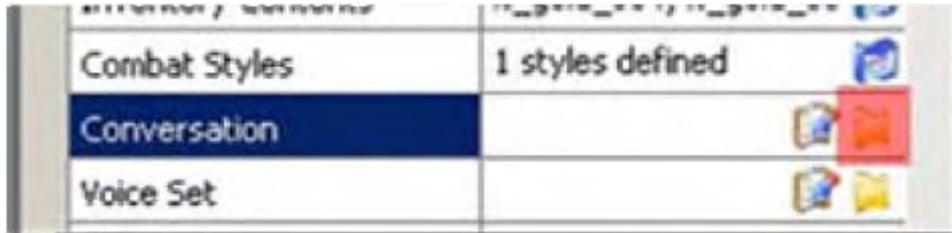
Doppelklicke auf die *our_jethro.utc* Datei:



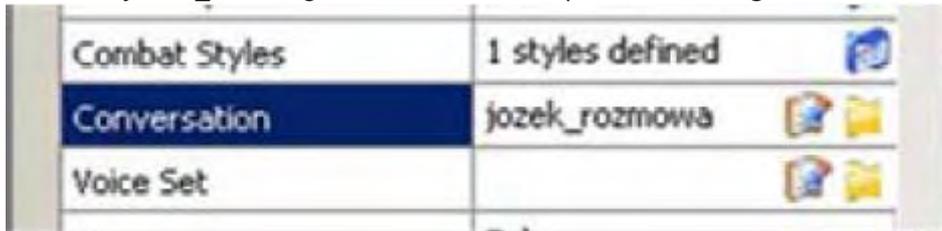
Das Einstellungsfenster der *our_jethro* Datei öffnet sich auf der rechten Seite des Schirms:



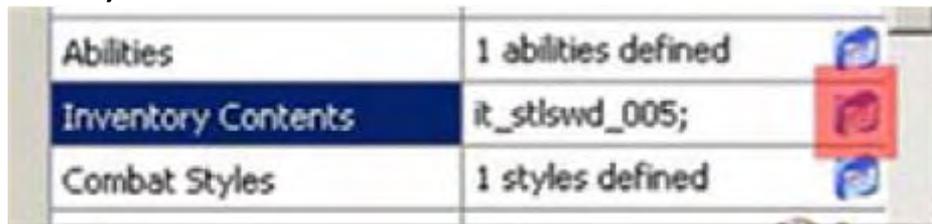
Klicke jetzt auf das kleine gelbe Symbol in der Zeile rechts neben **Conversation** unter der Überschrift **Statistics**:



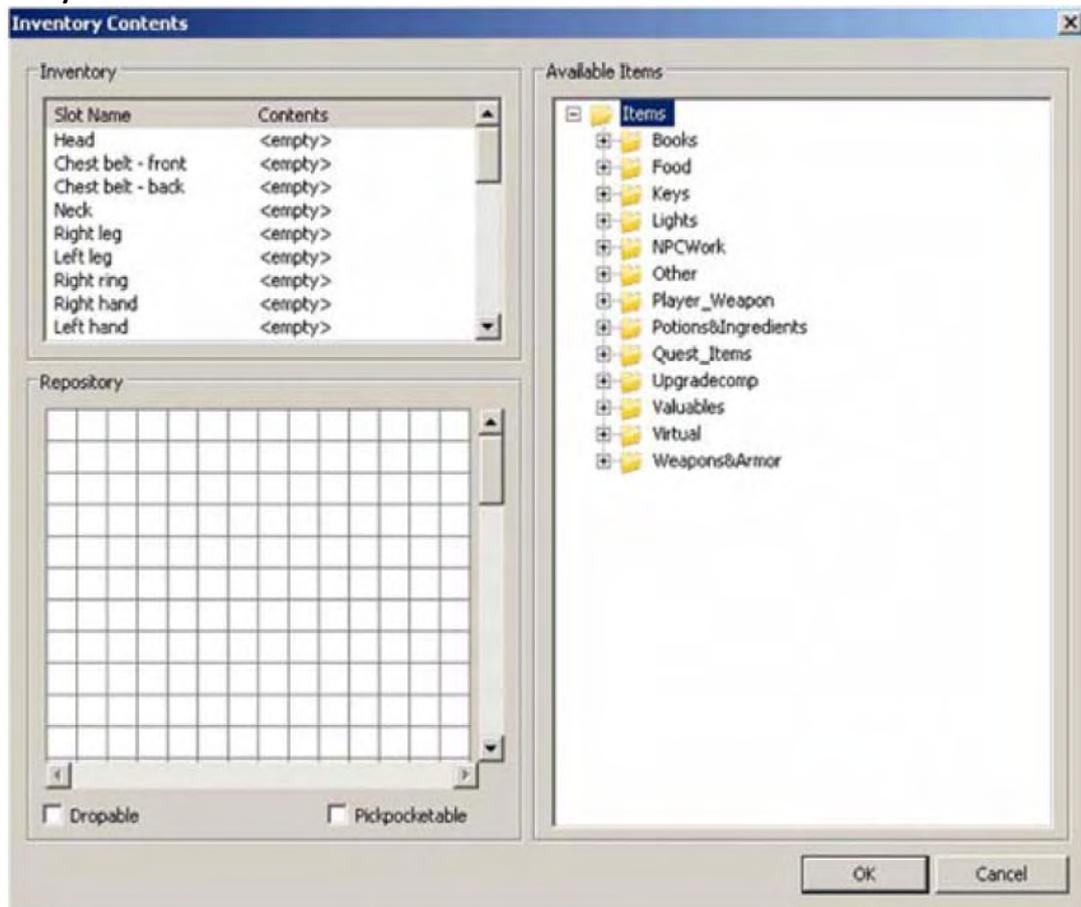
Das öffnet das Standard Windows Fenster für das Speichern von Dateien. Gehe zum *Data\Dialogues* Ordner und speichere die Datei *jethro_conv.dlg*. Dieses teilt das Gespräch der Vorlage von Jethro zu:



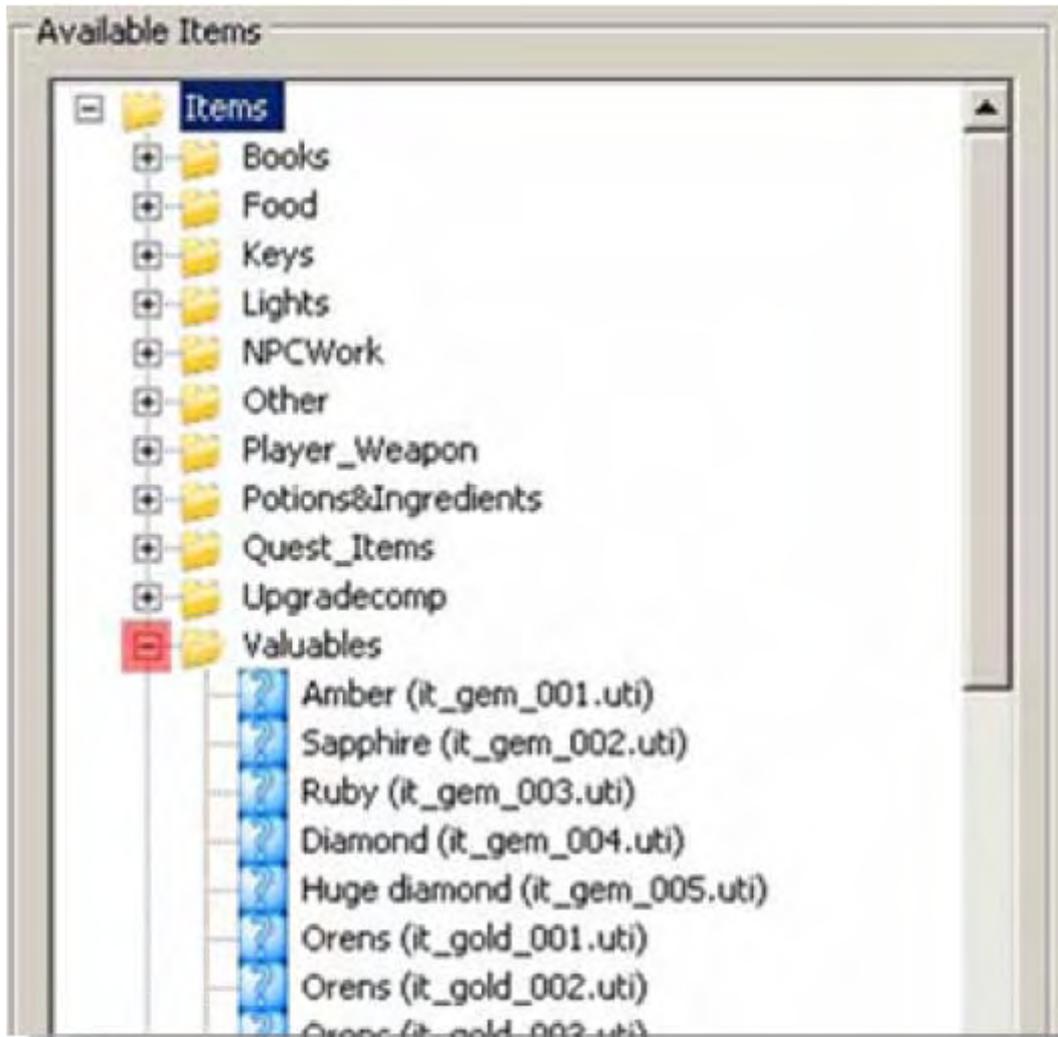
Jetzt müssen wir Jethro Orens geben (die in The Witcher verwendete Währung), sonst wird er sich weigern mit uns Poker zu spielen. Um Jethro's Besitz anzuzeigen, klickst du auf das kleine blaue Symbol in der Zeile rechts neben **Inventory Contents**:



Das **Inventory Contents** Fenster öffnet sich:



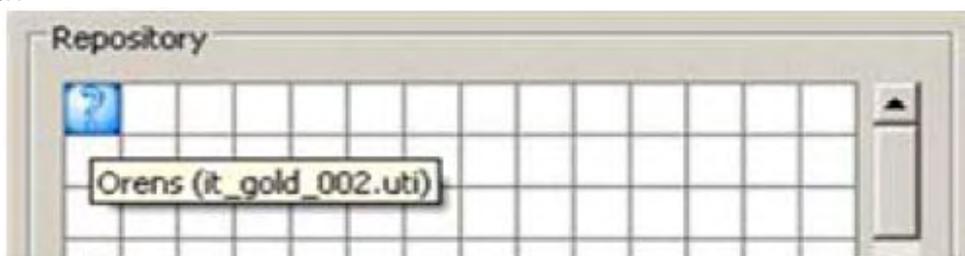
Klicke unter **Available Items** auf das kleine "+" Zeichen vor **Valuables** um eine Liste von Wertgegenständen anzuzeigen:



Wähle Orens(it_gold_002.uti) aus:



Klicke die Datei mit der linken Maustaste an, halte sie gedrückt und ziehe sie in ein beliebiges Feld im **Repository** Fenster:



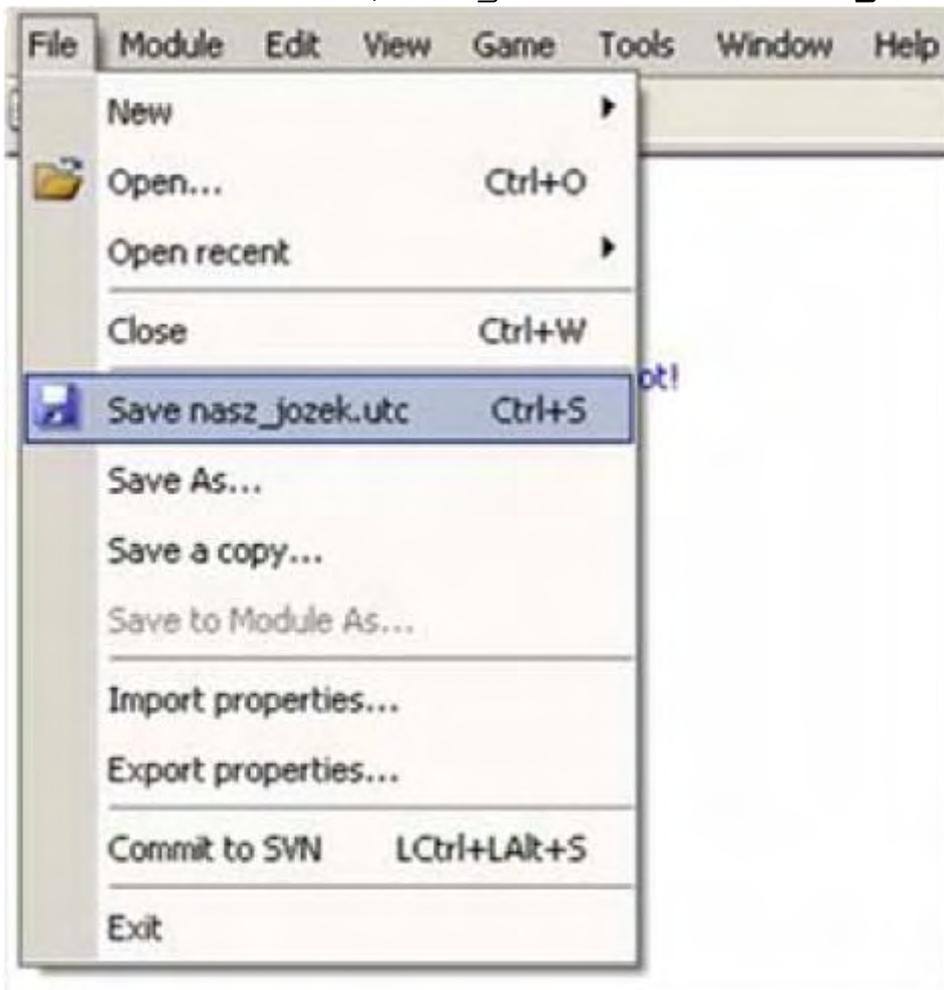
Lass die linke Maustaste los. Du kannst das mehrfach wiederholen, um Jethro mehr Orens zu geben:



Klicke abschließend auf **Ok** im **Inventory Content** Fenster. Jetzt müssen wir die aktualisierte Version der *our_jethro.utc* Datei speichern. Ein Sternchen vor dem Namen *our_jethro.utc* bedeutet, dass die Datei nicht gespeichert wurde, seitdem die letzte Änderung eingeführt wurde:



Speichere die Datei unter demselben Namen, d.h. *our_jethro.utc* und wähle **Save our_jethro.utc** im Menü:



Wenn **Save our_jethro.utc** nicht als Option angeboten wird, bedeutet es, dass dieses *our_jethro.utc* Fenster nicht aktiv ist. Um es zu aktivieren, klicke auf den Namen in der Titelleiste, dann wähle **save our_jethro.utc** vom Dateimenü aus. Fertig. Du hast Jethro jetzt seiner Dialog Datei zugewiesen.

QUESTS

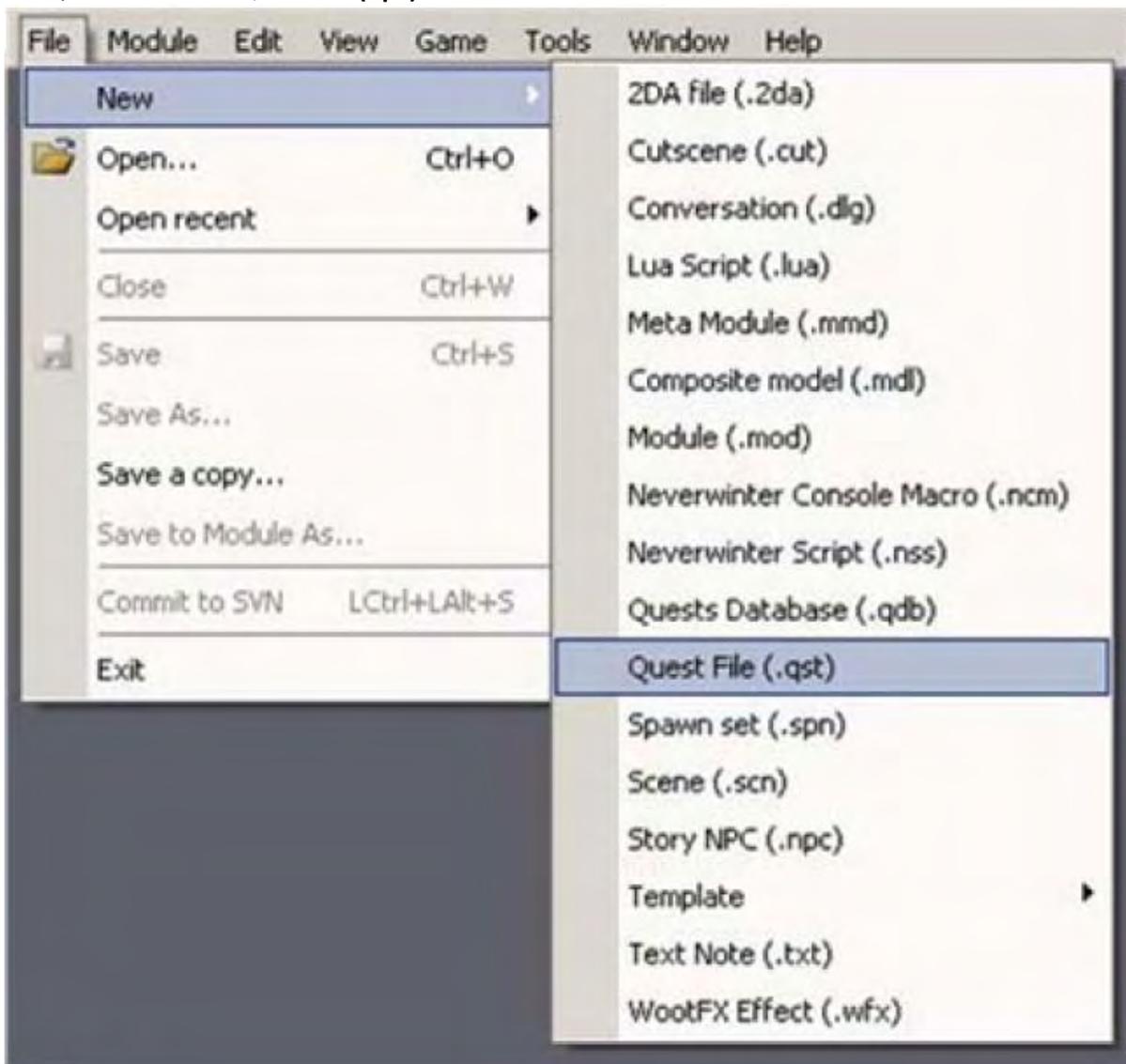
Einleitung:

In diesem Kapitel werden wir uns mit der Quest-Erstellung befassen. Handelt sich dabei hauptsächlich um Quests, die der Spieler von anderen Charakteren im Spiel bekommt. Wie du schon beim Spielen von The Witcher gemerkt hast, können Quests ganz unterschiedlich sein. So zum Beispiel, dass Töten eines Ungeheuers, dass abliefern eines Gegenstands, oder einen anderen Charakter zu eskortieren. Deine Phantasie entscheidet. Quests werden in Primary und Secondary unterschieden. Primary Quests sind für die Haupthandlung des Spieles wichtig, Secondary sind lediglich Nebenquests.

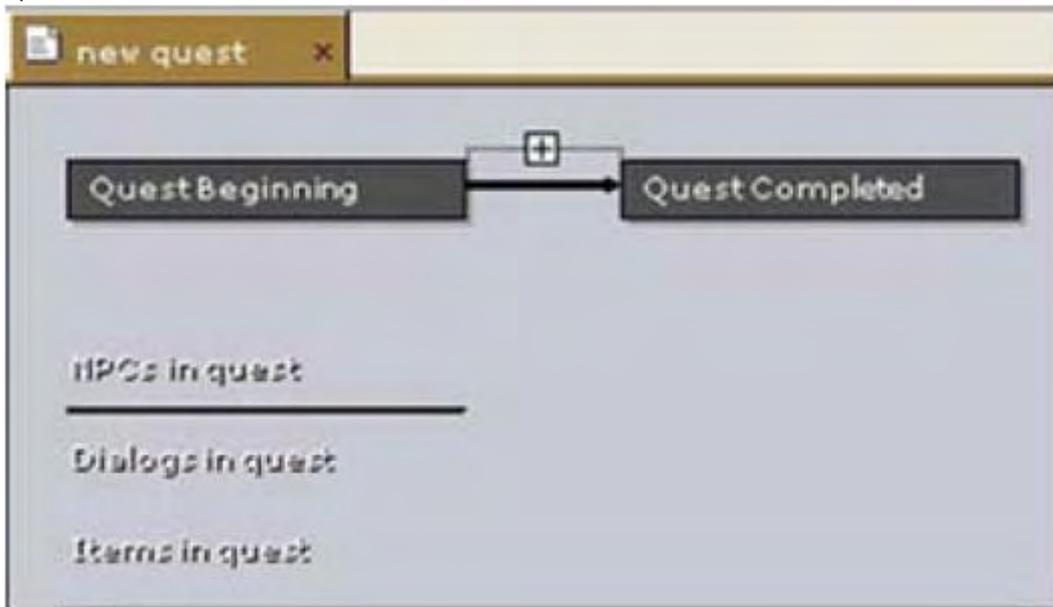
Jede Quest im Witcher wird in Phasen eingeteilt. Die Anzahl der Phasen einer Quest mag unterschiedlich sein. Allerdings hat jede Quest mindestens zwei Phasen. Einen Anfang, *QuestBeginning*, und ein Ende, *QuestCompleted*. Quests können sehr einfach sein, gerade Mal 2 Phasen umfassen, sie können aber auch komplex sein und viele Phasen, und Subquests haben. Quest die man erstellt, können zu einer Quest-Datenbank hinzugefügt werden, dadurch wird das Arbeiten mit ihnen leichter.

Quests:

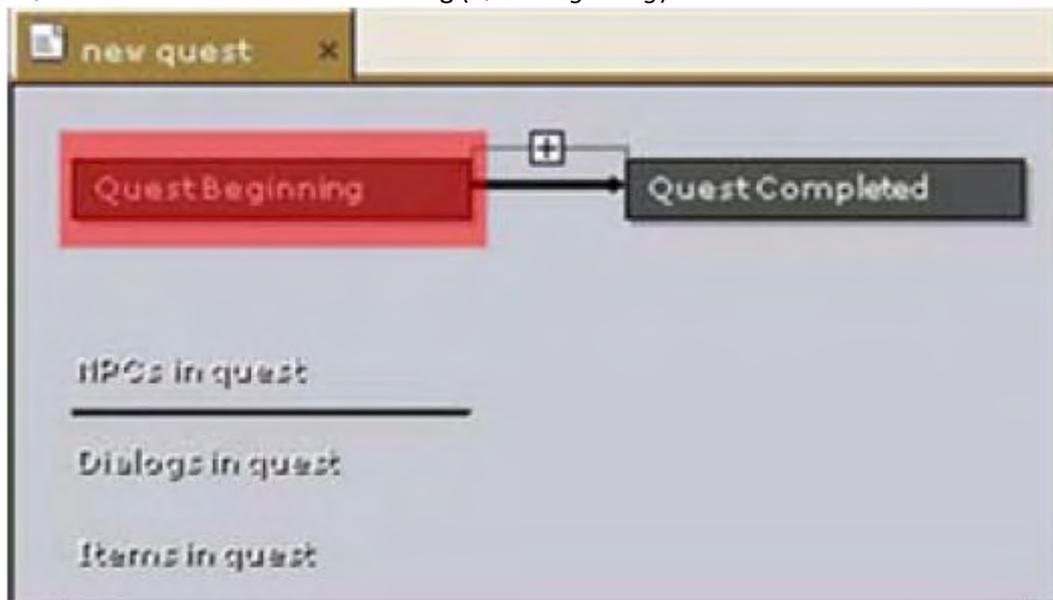
Die Quest in unserem Abenteuer wird ziemlich einfach sein. Es geht darin um die Tötung eines Banditenanführers. Sie besteht aus drei Phasen: Einen Questanfang (*QuestBeginning*) eine ExekutiveQuest (in dem Geralt den Banditenanführer töten muss) und ein Questende (*QuestCompleted*). Um eine neue Quest zu erstellen, wähle **New** -> **Quest File (.qst)** vom Menü aus:



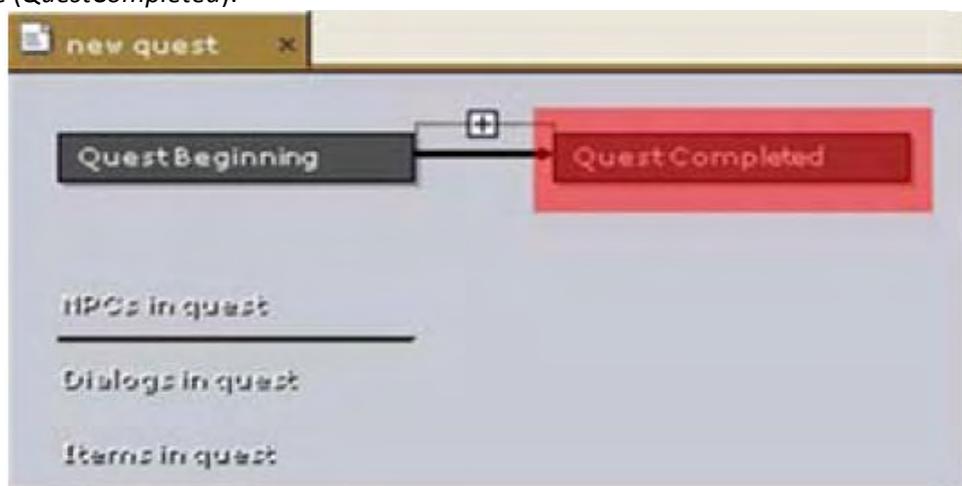
Das „new quest“ Fenster öffnet sich:



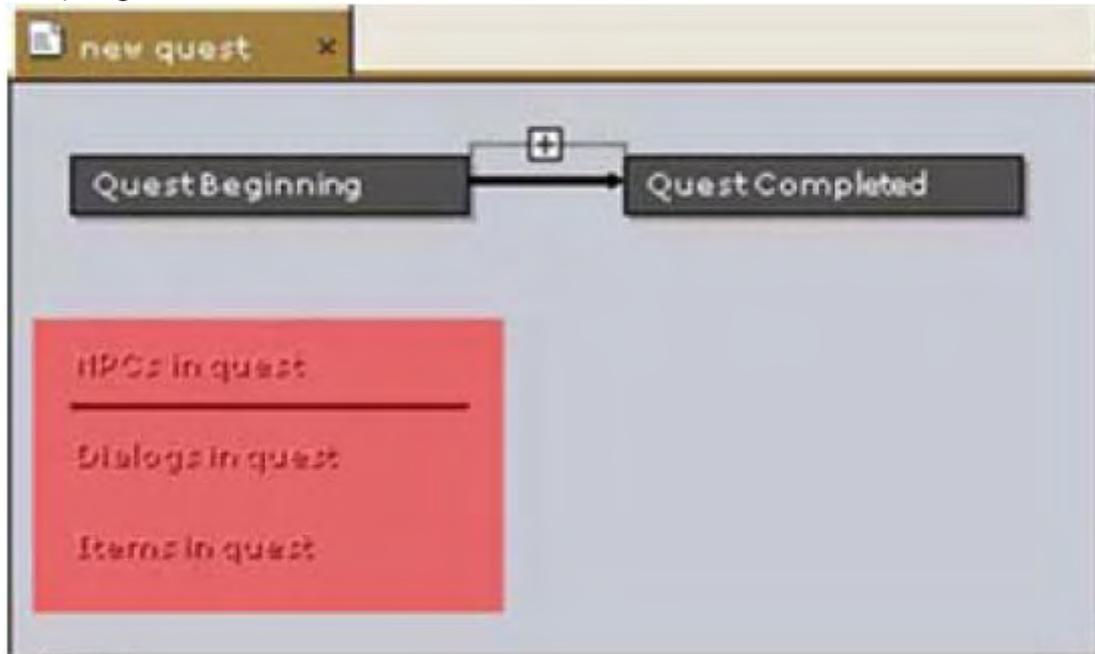
Jede neue Quest hat zwei Phasen – den Anfang (*QuestBeginning*):



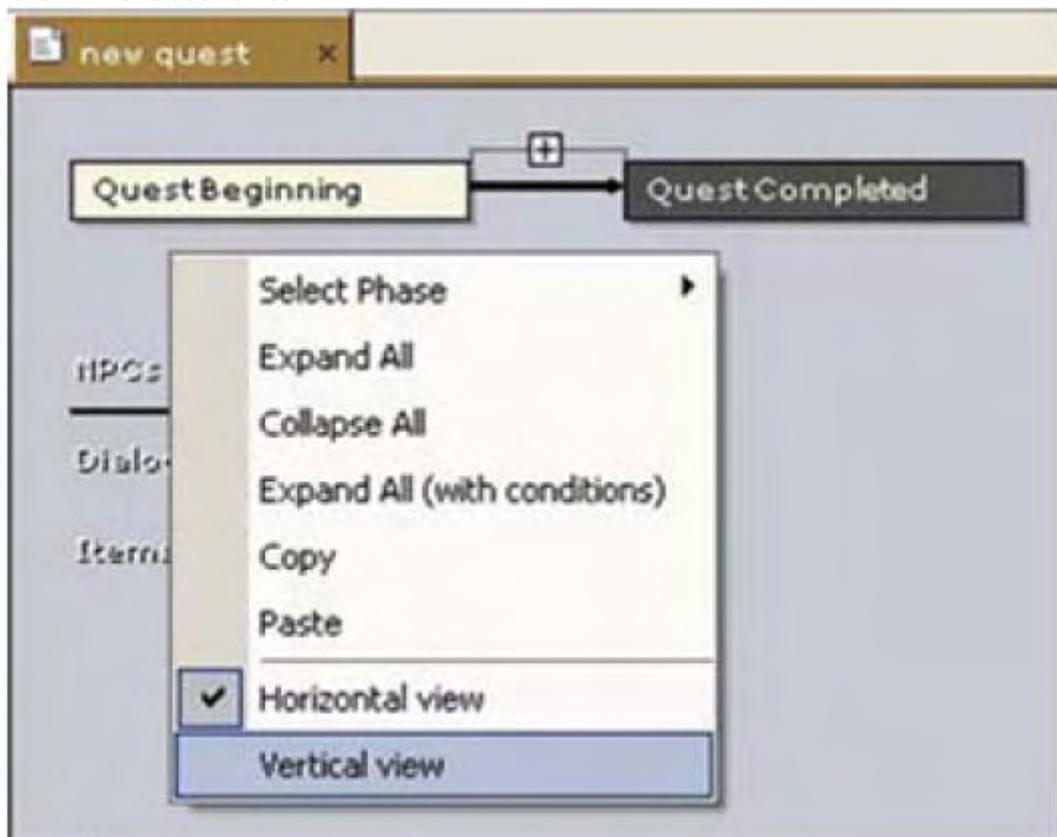
Und ein Ende (*QuestCompleted*):



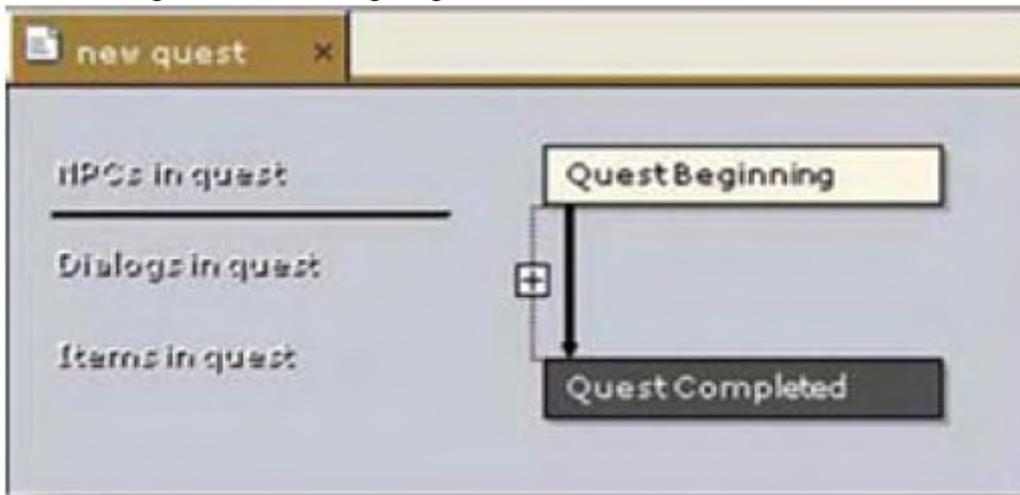
Zusätzlich werden Informationen über Quest relevante Charaktere (NPCs), Dialoge (Dialogue), und Gegenstände (Items) aufgelistet:



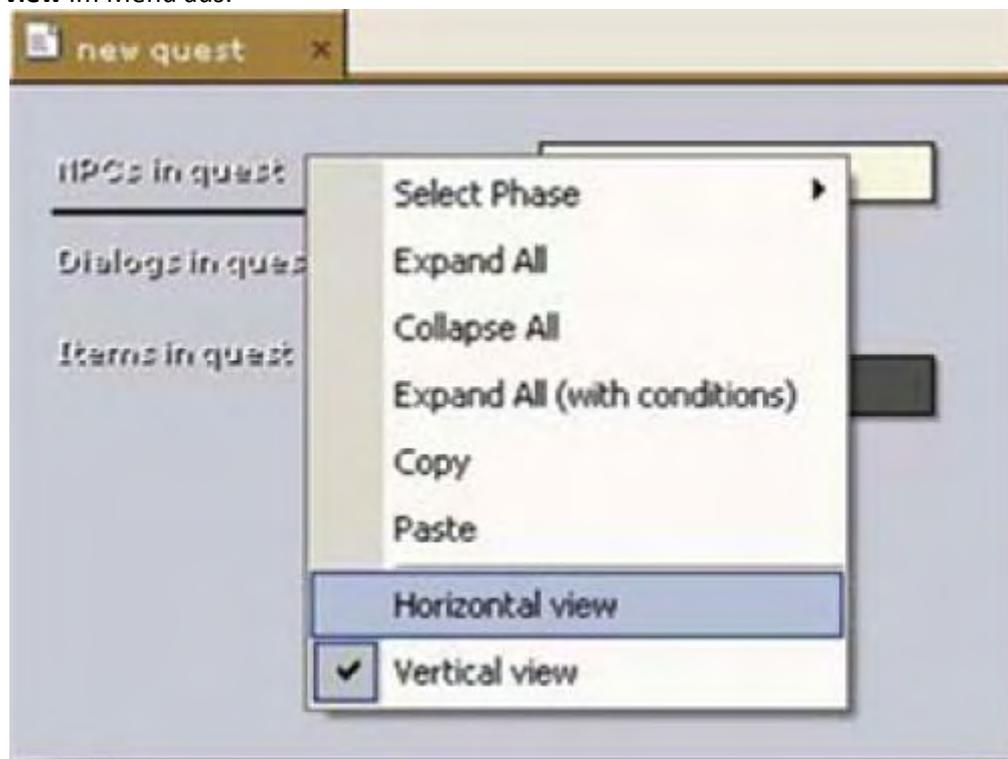
Die Quest wird in Form eines horizontales Diagrammes angezeigt. Wenn du willst, kannst du es auch als vertikales Diagramm anzeigen lassen, indem du mit einem Rechtsklick auf das **new quest** Fenster eine vertikale Ansicht im Menü auswählst:



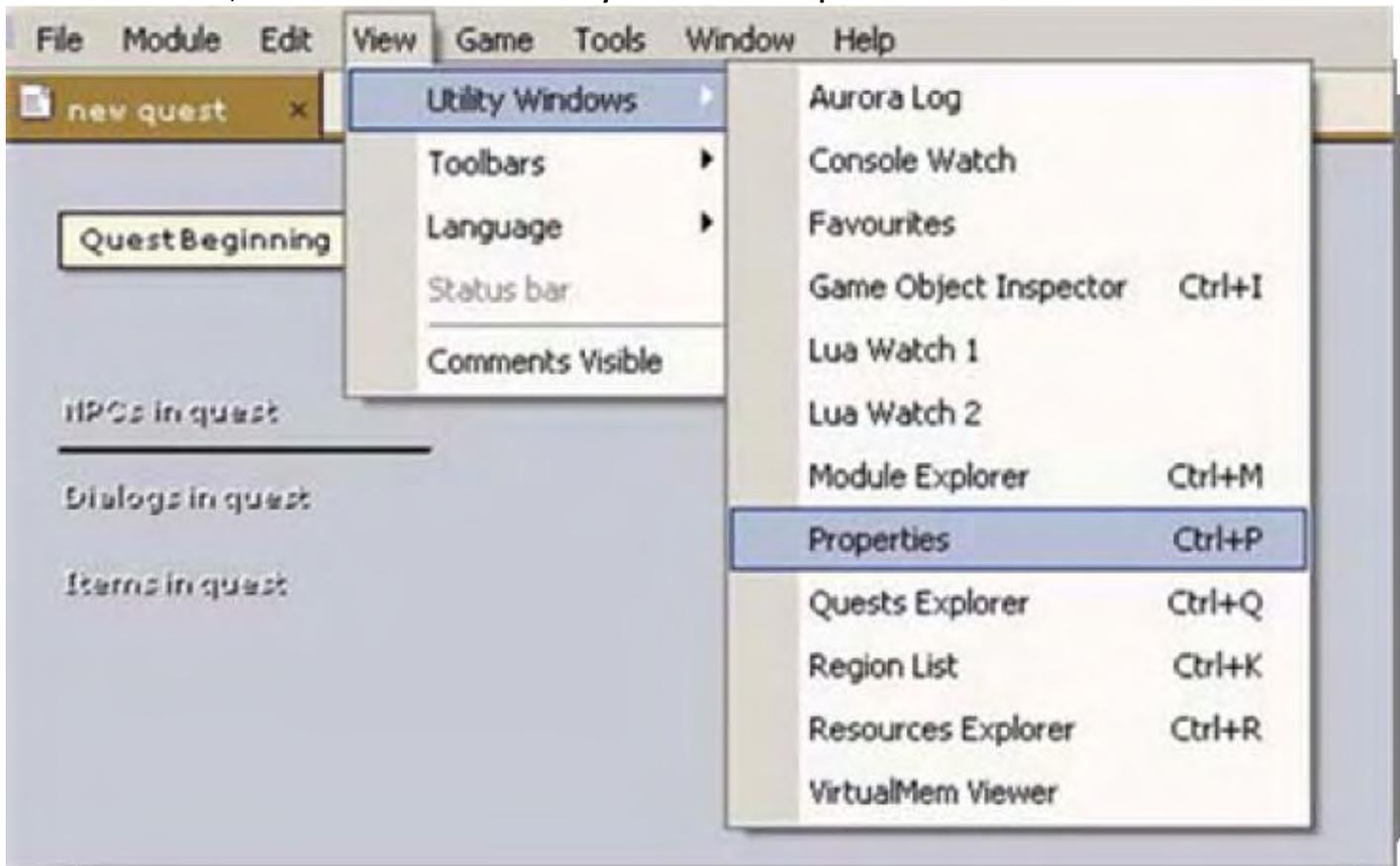
Jetzt wird das Questdiagramm vertikal angezeigt:



Um zur horizontalen Ansicht zurückzukehren, rechtsklicke wieder auf das **new quest** Fenster und wähle **horizontal view** im Menü aus:



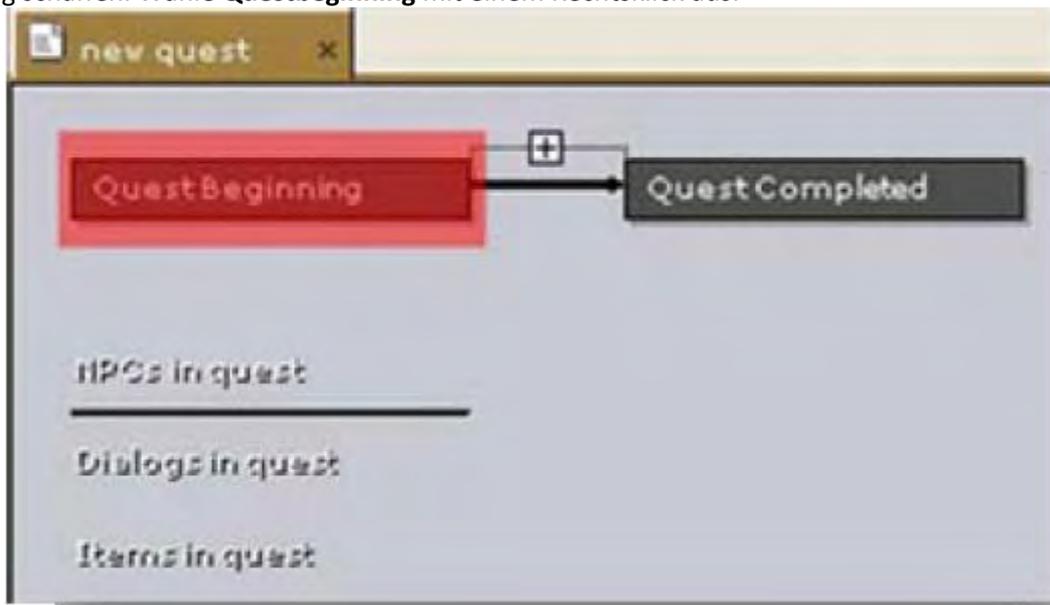
Welche Ansicht du bevorzugst, bleibt dir überlassen. Um jetzt mit der Quest-Erstellung weiter zu machen, brauchen wir das Eigenschafts-Fenster (Properties) unserer Quest. Sollte es nicht schon auf der rechten Seite zu sehen sein, wähle im Menü **View => Utility Windows => Properties** aus:



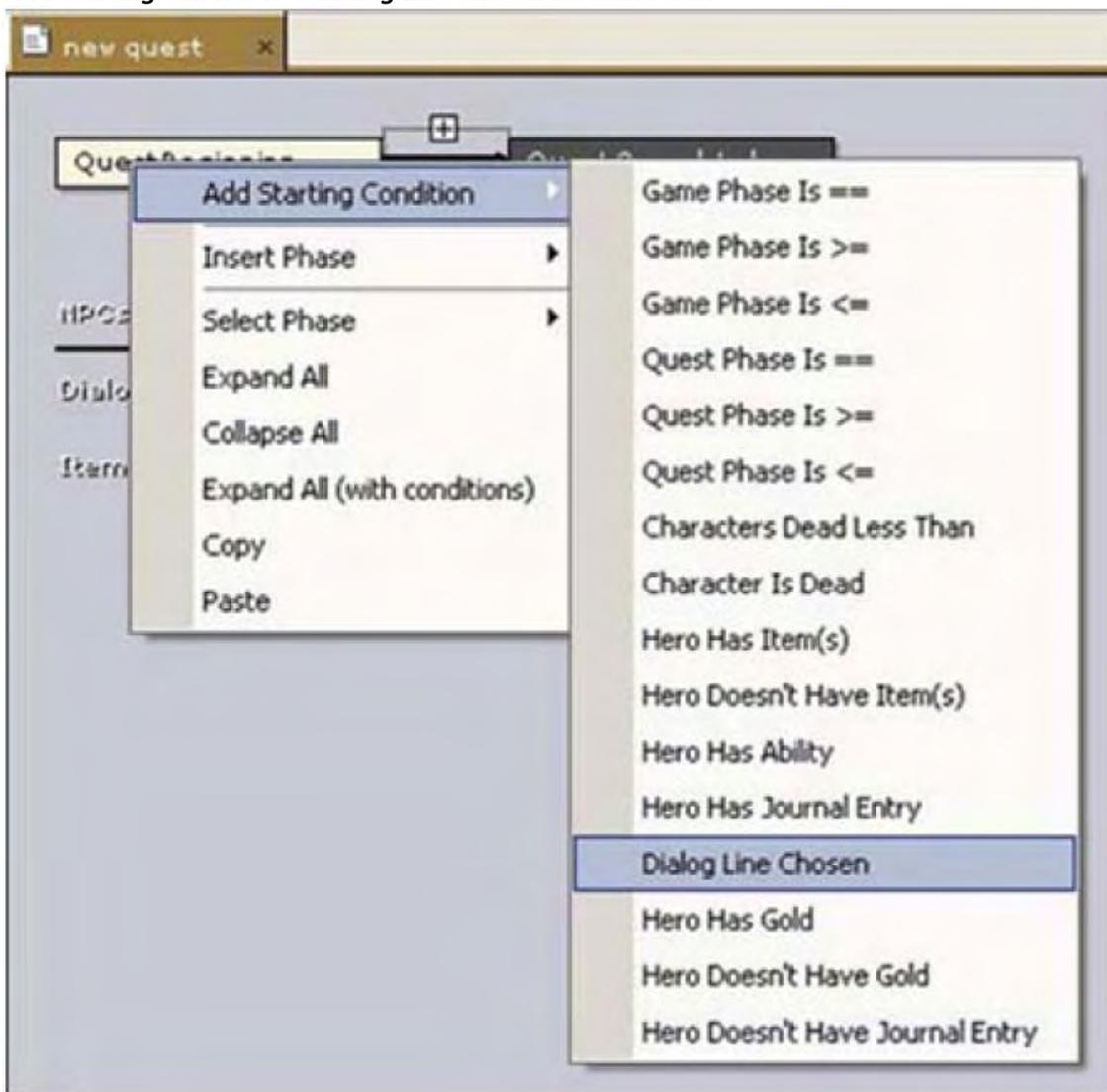
Ein leeres **Properties** Fenster öffnet sich auf der rechten Seite des Schirmes:



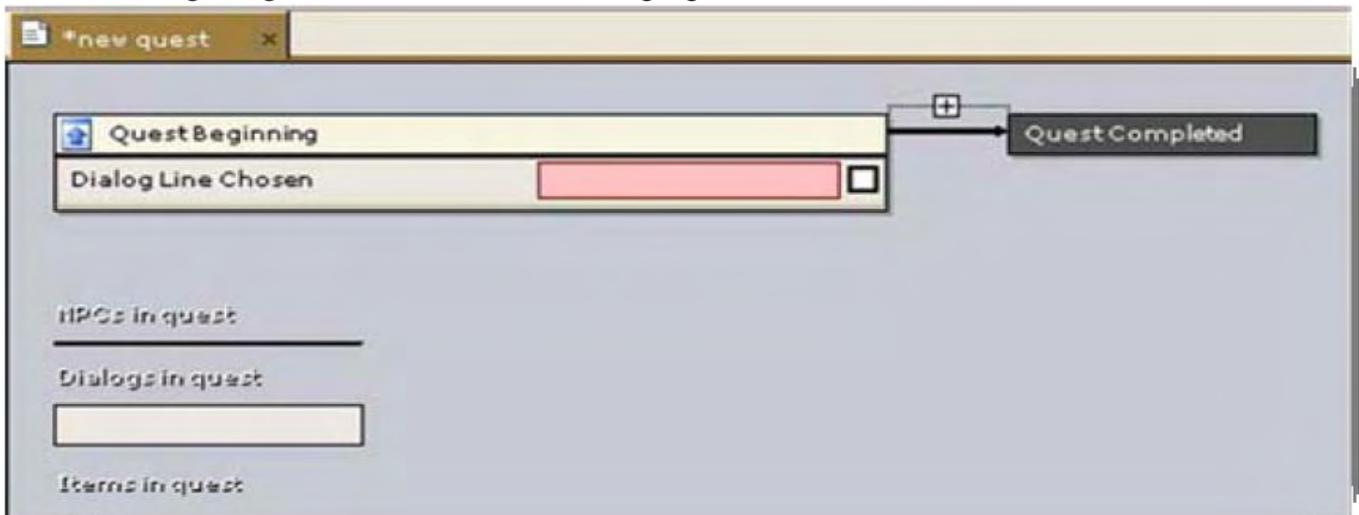
Geralt wird jetzt eine Quest erhalten wenn er den "Erledigt" Eintrag während seine Gespräches mit Siegfried auswählt. Für den Anfang der Quest, müssen wir diesen Gesprächs-Eintrag auswählen, und eine Startbedingung schaffen. Wähle **Questbeginning** mit einem Rechtsklick aus:



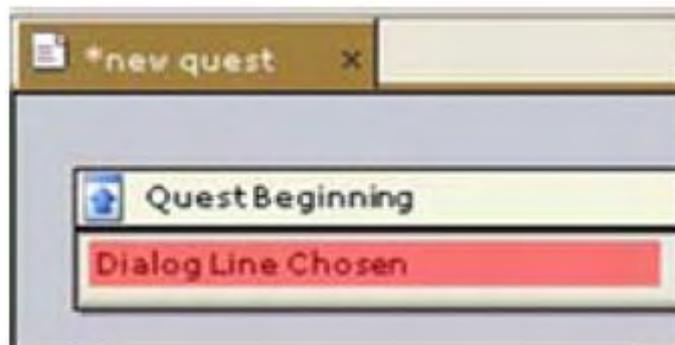
Wähle **Add Starting Condition -> Dialog Line Chosen** im Menü aus:



Unter **Quest Beginning** erscheint die neue Startbedingung:



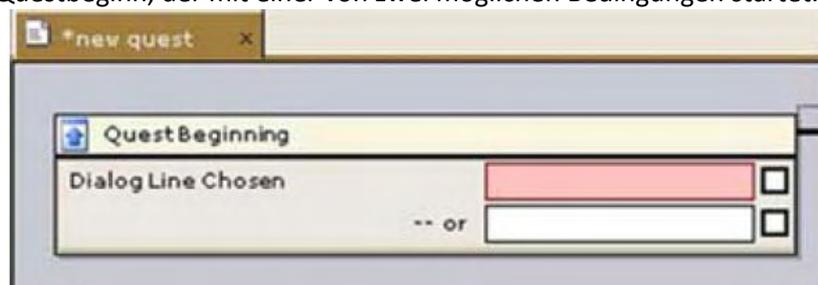
Wenn es in dem Gespräch mehrere Nodes geben soll, mit denen die Quest beginnen kann, dann kannst Du eine weitere Bedingung hinzufügen. Dazu markierst Du eine Bedingung unter **Quest Beginning** und öffnest mit einem Rechtsklick ein weiteres Menü:



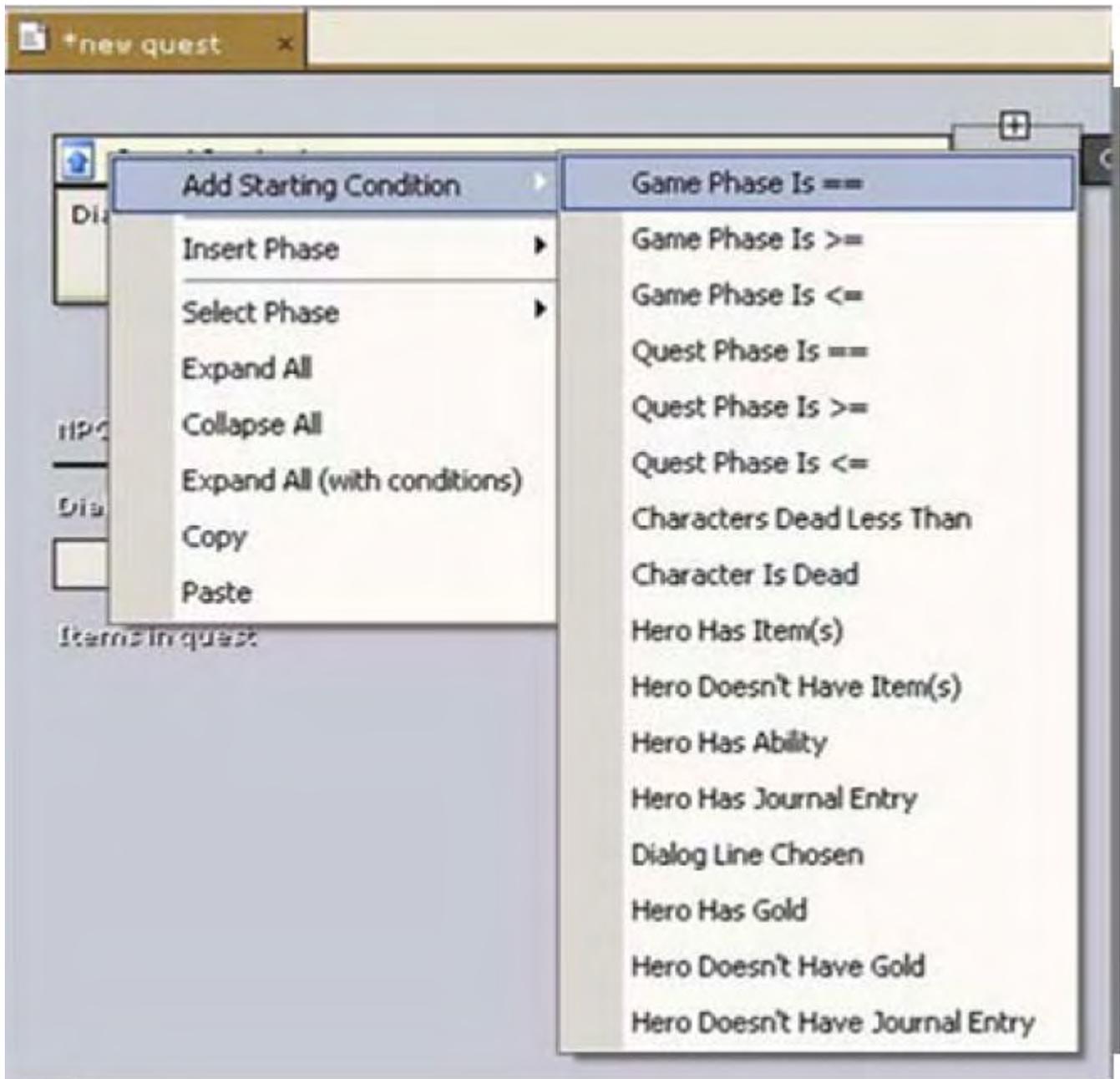
Wähle **Add 'or' Condition** im Menü aus:



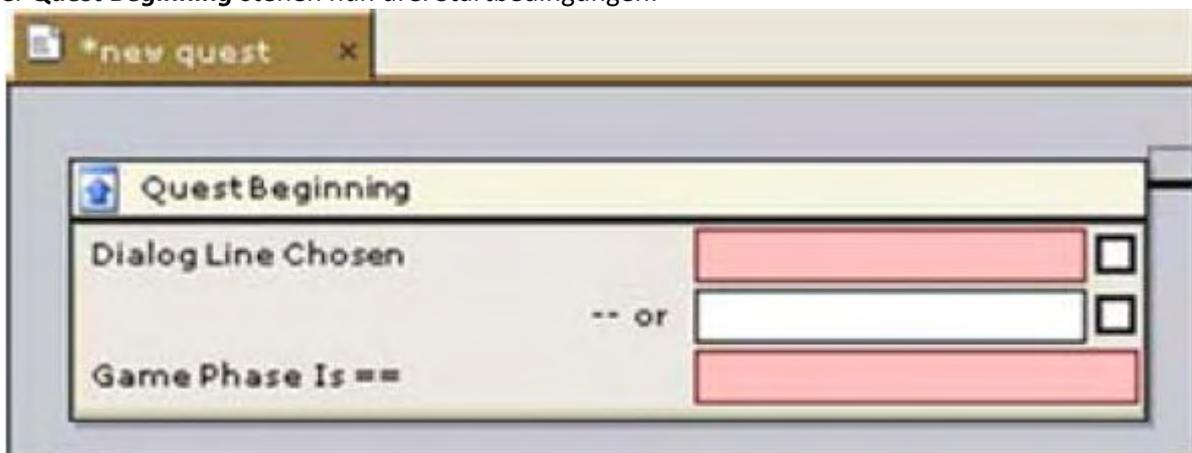
Das erzeugt einen Questbeginn, der mit einer von zwei möglichen Bedingungen startet.



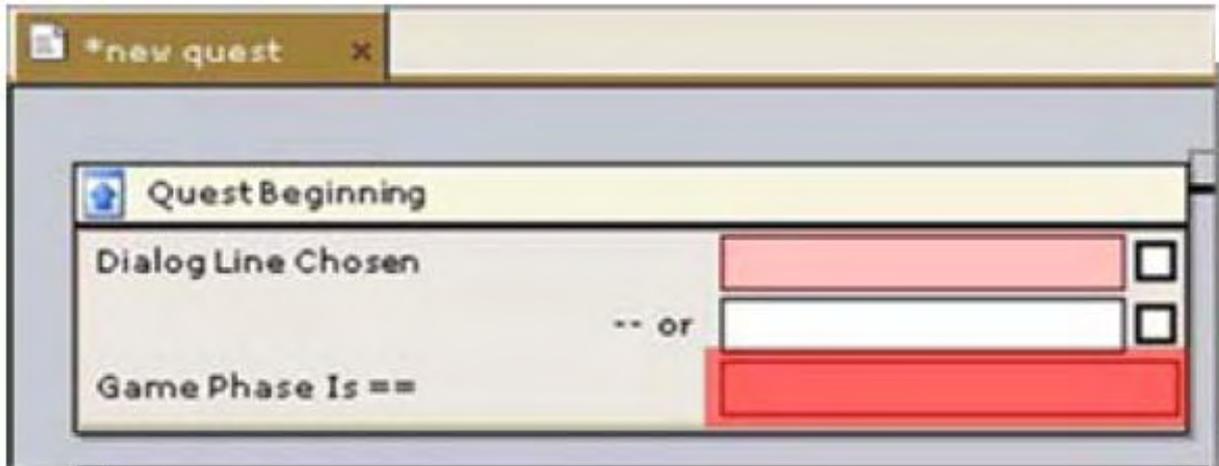
Außer Dialogoptionen kannst Du auch andere Startbedingungen für die Quest auswählen. Markiere **Quest Beginning** und öffne mit einem Rechtsklick das Menü. Wähle dann **Add Starting Condition -> Game Phase Is ==** vom Menü:



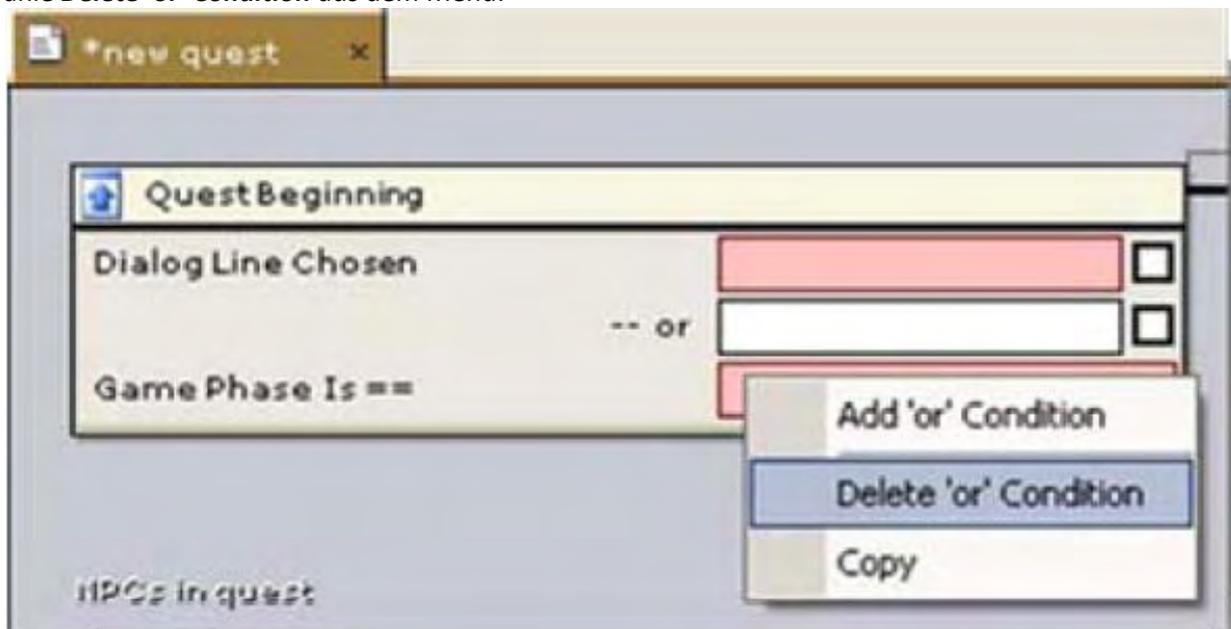
Unter **Quest Beginning** stehen nun drei Startbedingungen:



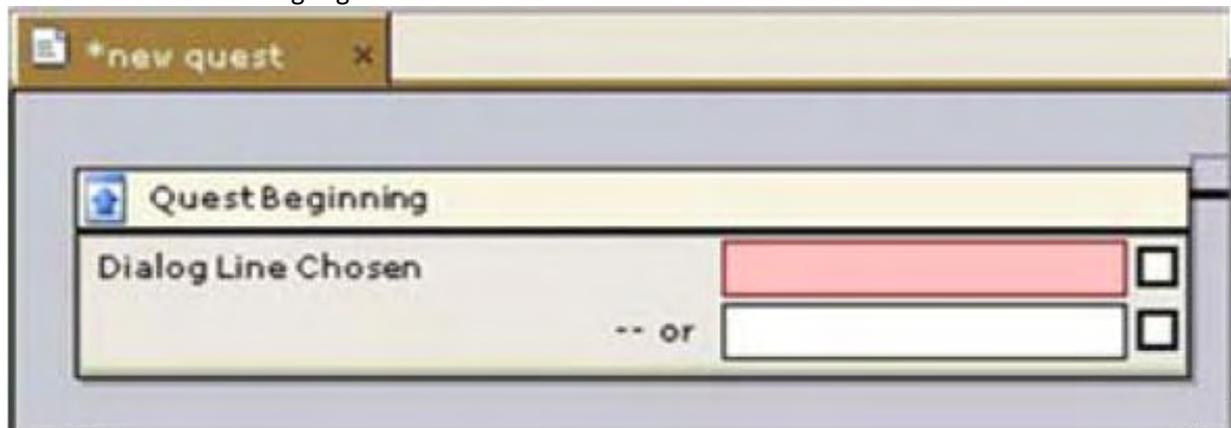
Es müssen jedoch nur zwei der Bedingungen erfüllt sein, um die Quest zu starten, weil die zweite **Dialog Line Chosen**-Bedingung eine 'oder'-Bedingung ist. Das heißt, es genügt, wenn eine der beiden Bedingungen erfüllt ist. Für unsere Quest soll es aber reichen, dass eine Bedingung erfüllt ist, also brauchen wir die beiden anderen nicht. Mit einem Rechtsklick auf das kleine Feld neben **Game Phase Is ==** öffnest Du ein Menü:



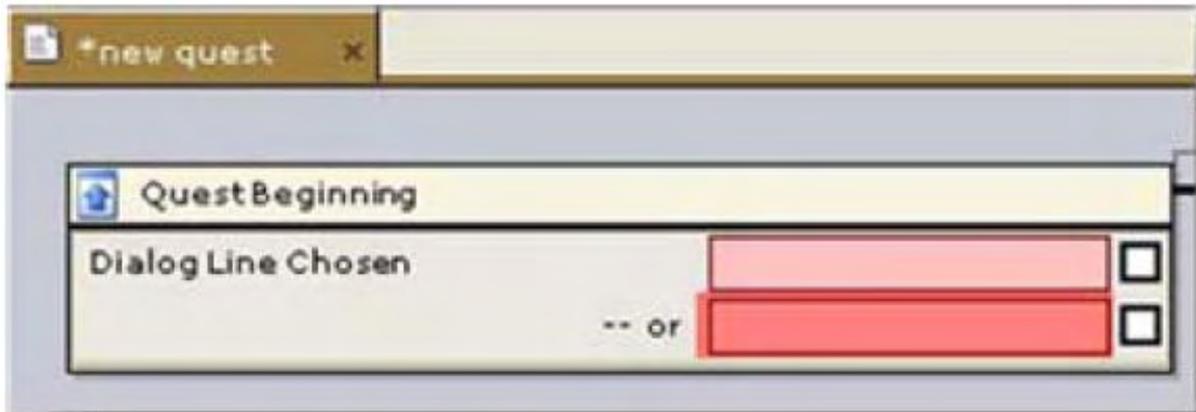
Wähle **Delete 'or' Condition** aus dem Menü:



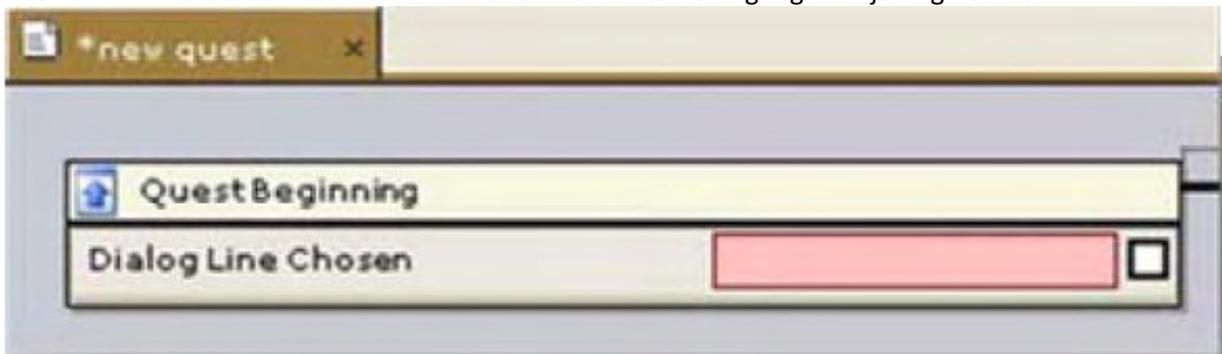
Das löscht die dritte Bedingung:



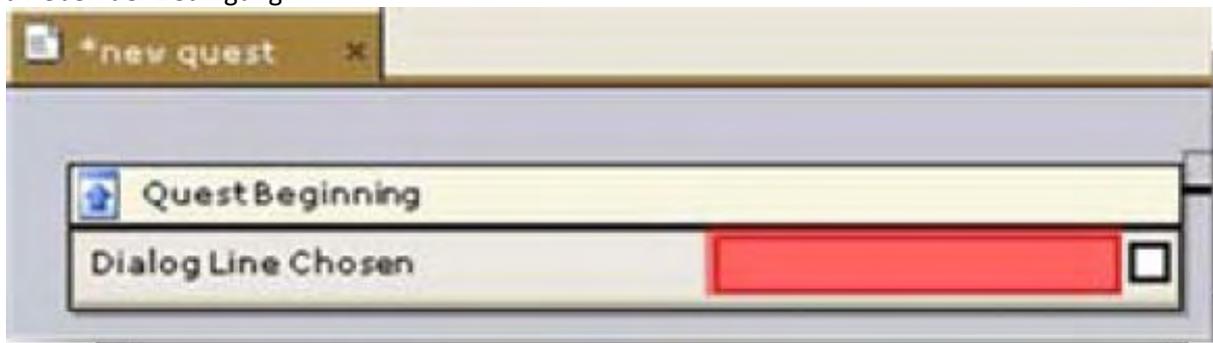
Für die zweite Bedingung wiederholst Du das Ganze nochmal. Rechtsklick auf das Feld neben **Dialog Line Chosen** öffnet wieder das Menü:



Wähle nochmal **Delete 'or' Condition** im Menü. Die zweite Bedingung wird jetzt gelöscht:



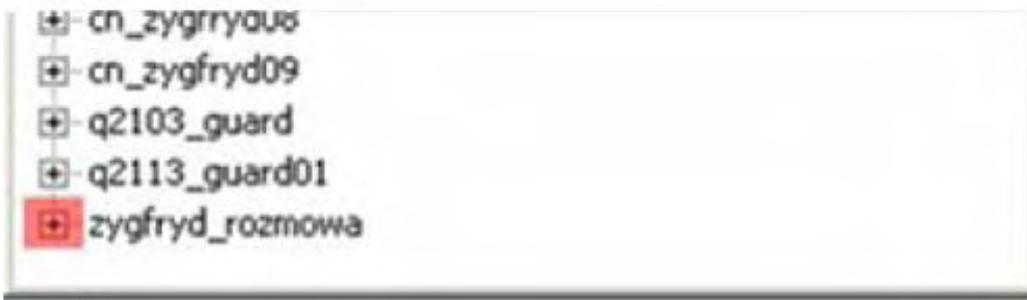
Jetzt müssen wir der Bedingung **Dialog Line Chosen** den passenden Dialog zuordnen. Klicke dazu auf das Feld neben der Bedingung:



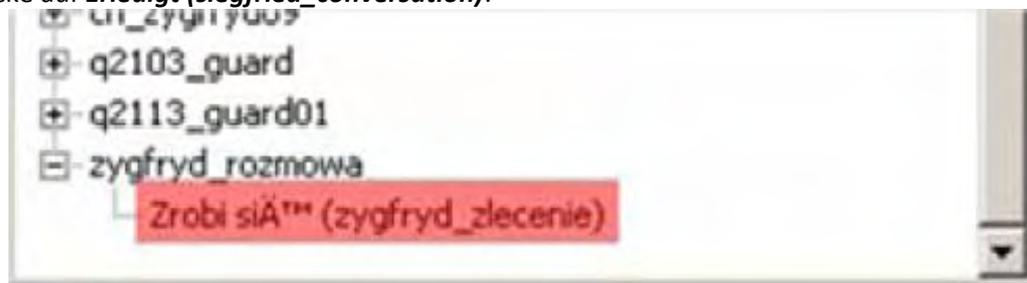
Das folgende Fenster öffnet sich:



Das Fenster enthält alle Quest-Dialoge im Ordner Data\Dialogues, die einen Quest-Node mit einer QuestID enthalten. Die Dialoge werden in der Form eines Baums gezeigt. Unser "siegfried_conv" Dialog wird in der Liste abgelegt. Um den passenden Dialog-Node auszuwählen, klappt Du den "siegfried_conv" Zweig aus, indem Du auf das kleine '+' Zeichen neben dem Namen des Dialogs klickst:



Doppelklicke auf **Erledigt (siegfried_conversation)**:



Der **Dialog Line Chosen** Bedingung wird der Dialog-Node "Erledigt" zugeordnet:



Dialog Line Chosen-Bedingungen können progressiv sein (d. h. eine bestimmte Questphase abschließen) oder nicht progressiv (d. h. sie beenden die Questphase nicht). Damit die Bedingung nicht progressiv ist, wählst Du das Kästchen neben dem zugeordneten Dialog-Node:



Wenn der Dialog-Node nicht progressiv ist, dann bleibt die Quest unverändert und der betreffende Node steht beim nächsten Gespräch mit dem Charakter wieder zur Verfügung. Das ist sinnvoll, wenn der Spieler von dem Charakter eine Quest annehmen soll, aber noch die Möglichkeit haben soll, vor ihrem Abschluss nach Einzelheiten zu fragen.

Wichtig: Wenn ein Quest-Dialog erstellt wird, aber der zugehörige Node keiner Quest zugeordnet wird, dann wird der Dialog nie angezeigt. Als nächstes müssen wir die Attribute von **Quest Beginning** setzen. Klicke auf **Quest Beginning**:



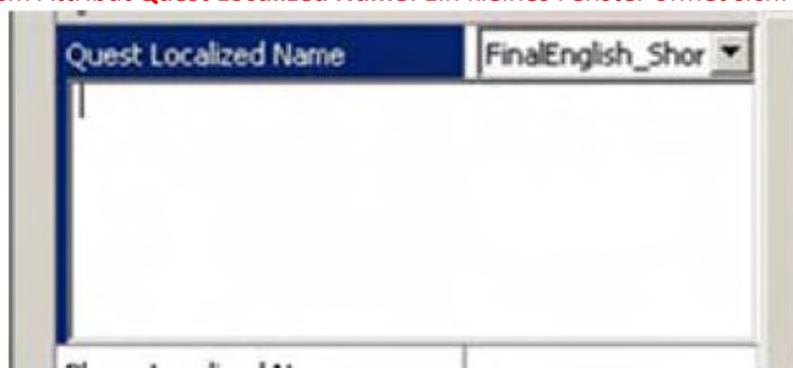
Das Fenster **Properties** enthält eine Liste von Attributen:



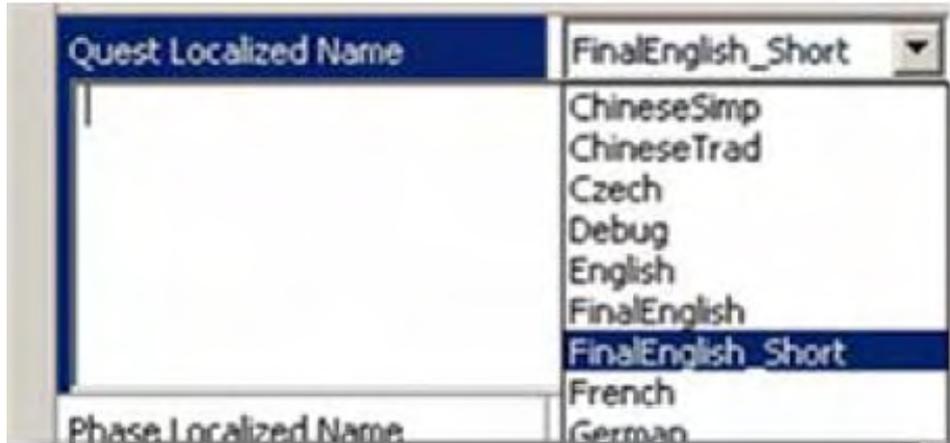
Quest	
Quest Localized Name	
Phase Localized Name	
Phase Localized Description	
Phase Localized Short Des...	
Phase Failed Localized De...	
Phase Failed Localized Sh...	
Journal Entry Id	Q1012_For_sale 
Waypoint assigned (tag)	
Worldmap region	NO_REGION_SET 
OnPhaseCompleted Action	
OnPhaseFailed Action	
Comment	
QuestFile	new quest
Is Primary	False
OnPhaseFinished Actions	
Available Actions	Add XP to player 
OnQuestFinished Actions	
Available Actions	Add XP to player 

Quest Section:

- **Quest Localized Name.** Enthält den Namen der Quest. Um einen Text einzufügen, klicke auf die rechte Spalte gegenüber dem Attribut **Quest Localized Name**. Ein kleines Fenster öffnet sich:



Gib "Befehle" ein und drücke Enter. Außerdem kannst Du in der Liste eine Sprache auswählen:



Der Name "Befehle" wird jetzt in der Quest-Liste im Journal von Geralt erscheinen:



• **Phase Localized Name.** Enthält den Namen der Quest-Phase. Um einen Text einzufügen, klicke auf die rechte Spalte gegenüber dem Attribut **Phase Localized Name**, gib "Jagd" ein und drücke Enter. Der Name "Jagd" wird jetzt im Journal von Geralt in der ersten Phase der Quest "Befehle" erscheinen:



- **Phase Localized Description.** Dieses Attribut enthält die Beschreibung der Questphase.

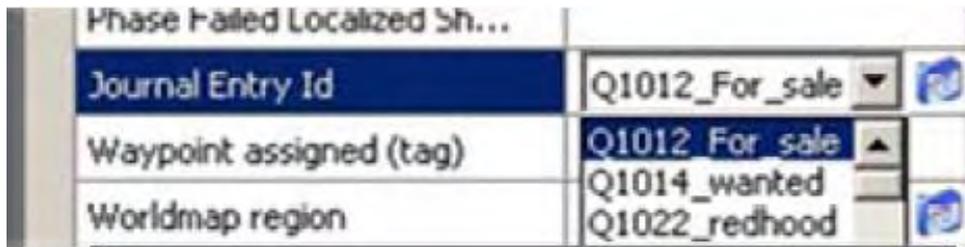
Klicke auf die rechte Spalte gegenüber dem Attribut **Phase Localized Description** und schreibe: *"Siegfried hat das Urteil über den Anführer der Banditen gesprochen. Er muss sterben. Vielleicht springen dabei ein paar Orens heraus."* Dieser Text wird in der Beschreibung der Quest "Befehle" unter der Phase "Jagd" erscheinen:



- **Phase Localized Short Description.** Enthält eine kurze Beschreibung der Quest-Phase. Das ist der beste Platz für Schlüsselinformationen wie z. B. das Ziel der Questphase. Klicke auf die rechte Spalte gegenüber dem Attribut **Phase Localized Short Description** und schreibe *"Ich muss den Anführer der Banditen töten."*. Dieser Text wird jetzt gelb unter der Questbeschreibung erscheinen:



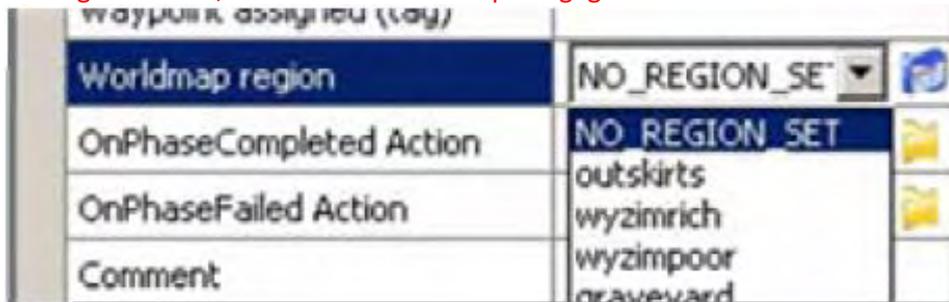
- **Phase Failed Localized Description.** Enthält eine ausführlichere Beschreibung, warum eine Quest-Phase fehlschlug. Wir brauchen das für unsere Quest nicht wirklich, hier musst du nichts schreiben.
- **Phase Failed Localized Short Description.** Enthält eine kurze Beschreibung, warum eine Quest-Phase fehlschlug. Hier musst Du auch nichts schreiben.
- **Journal Entry Id.** Enthält Informationen, die in Geralts Journal erscheinen können. In diesem Fall könnte es z. B. ein Eintrag über Siegfried sein, der etwas mit der Quest zu tun hat. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber dem Attribut **Journal Entry Id**:



Ein Klick auf das blaue Icon neben der Auswahlliste öffnet die Datei *journal.2da*, eine Datenbank, die Informationen über Journaleinträge enthält.

- **Waypoint assigned (tag).** Enthält eine Markierung für einen Wegpunkt, die auf die Stelle auf der Karte zeigt, an der die Quest abgeschlossen wird. Für den Moment tragen wir hier nichts ein.

- **Worldmap region.** Enthält die Karte des Gebiets, wo eine bestimmte Quest abgeschlossen werden soll. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber dem Attribut **Worldmap region**:



Diese Liste enthält alle im Witcher verfügbaren Gebiete. Wähle NO_REGION_SET aus der Liste aus.

- **OnPhaseCompleted Action.** Enthält ein Script, das aktiviert wird, wenn eine bestimmte Questphase erfolgreich beendet wurde. Um ein Script zuzuordnen, klicke auf das kleine gelbe Icon in der rechten Spalte gegenüber dem Attribut **OnPhaseCompleted Action**:



Das öffnet das übliche Windows-Fenster zum Laden von Dateien. Öffne den Ordner Data\Scripts, markiere die passende Datei und klicke auf *Öffnen*:

- **OnPhaseFailed Action.** Enthält ein Script, das aktiviert wird, wenn eine bestimmte Questphase fehlgeschlagen ist. Dieses Script wird genauso zugeordnet wie unter **OnPhaseCompleted Action** beschrieben.

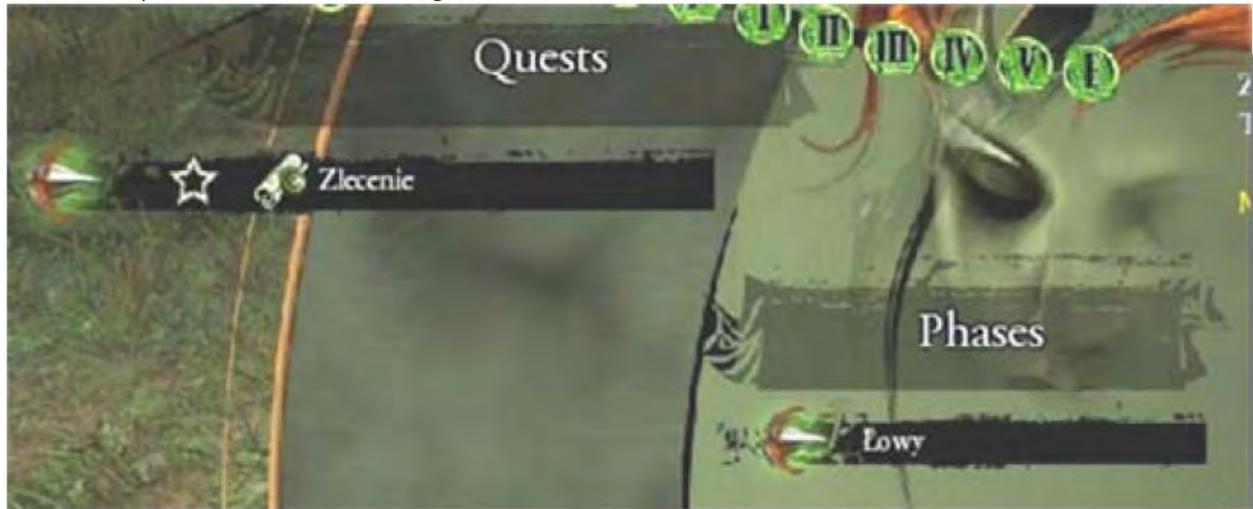
- **Comment.** Enthält Platz für einen Kommentar. Dieses Feld wird im Spiel nicht verwendet, es ist nur ein Informationsfeld. Hier kannst Du eintragen, was Du möchtest. Um diese Einstellung zu ändern, klicke auf die rechte Spalte gegenüber dem Attribut **Comment**.

- **QuestFile.** Enthält den Namen der Questdatei. Nach dem Speichern unserer Quest wird hier der Name der Questdatei erscheinen.

- **IsPrimary.** Bestimmt, ob es sich um eine primäre Quest (Hauptquest) handelt, die für den Fortschritt der Handlung notwendig ist, oder um eine sekundäre Quest (Nebenquest). Unsere Quest wird eine Hauptquest sein. Klicke auf die rechte Spalte gegenüber dem Attribut **IsPrimary** und wähle *True* von der Liste aus:



False bedeutet, dass die Quest nicht primär ist (also keine Hauptquest). Eine primäre Quest wird auch im Journal des Spielers in der Questliste gekennzeichnet:



OnPhaseFinishedActions Section:

In diesem Abschnitt wird definiert, welche Aktionen beim Abschluss einer bestimmten Questphase ausgeführt werden. Um zu sehen, wie das funktioniert, geben wir Geralt 100 Orens (die Währung, die im Witcher verwendet wird). Mit diesem Geld kann er dann mit Jethro pokern. Klicke auf die rechte Spalte gegenüber dem Attribut **Available Actions** und wähle **Transfer Gold to Player** von der Liste aus:



Die anderen Werte, die Du dieser Option zuweisen kannst, sind :

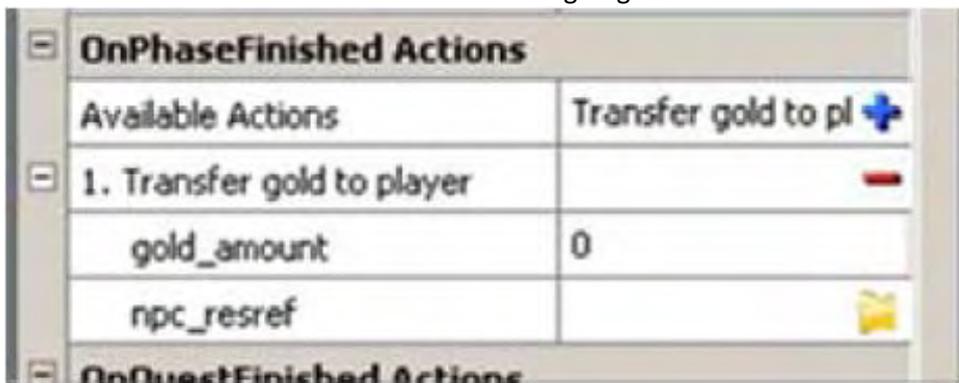
- *Add XP to player* - gibt dem Spielercharakter eine bestimmte Anzahl von Erfahrungspunkten
- *Transfer item to player* - gibt dem Spielercharakter einen Gegenstand
- *Transfer item to NPC* - gibt einem NPC (nicht dem Spielercharakter) einen Gegenstand
- *Change game phase* - wechselt die Spielphase
- *Enable spawnset* - aktiviert ein Spawnset
- *Disable spawnset* - deaktiviert ein Spawnset
- *Change NPC Story phase* - Änderung einer NPC Story phase

- *Transfer gold to NPC* - gibt einem NPC (nicht dem Spielercharakter) einen Geldbetrag
- *Transfer gold to player* - gibt dem Spielercharakter einen Geldbetrag
- *Set Dialog flag* - setzt ein Flag für einen Dialog
- *Enable trigger* - aktiviert einen Trigger

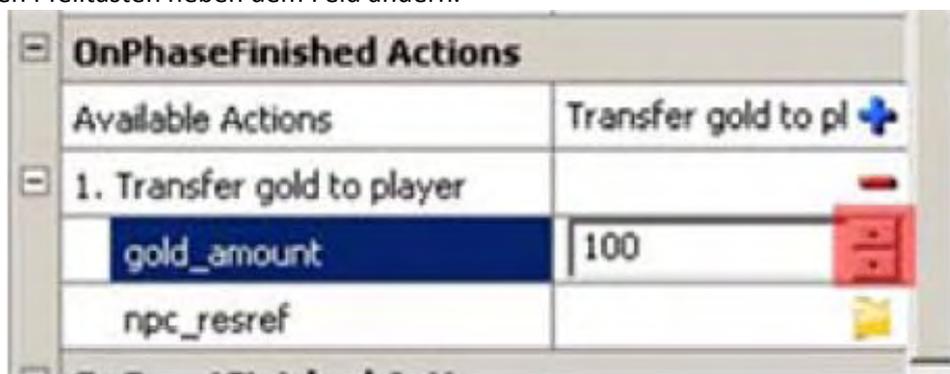
Nachdem Du eine Aktion gewählt hast, klicke auf das blaue '+' neben der zugehörigen Aktion:



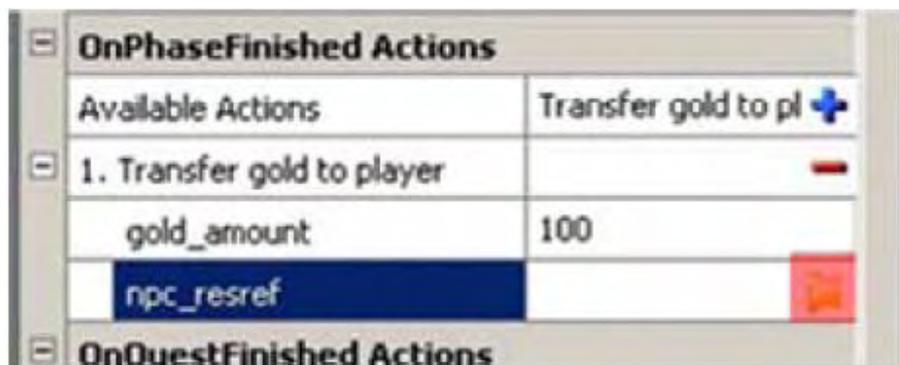
Die Aktion wird im Abschnitt **OnPhaseFinishedActions** hinzugefügt:



Jetzt geben wir Geralt 100 Orens. Dafür gibt es das Attribut *gold_amount*. Um die Einstellung zu ändern, klicke in die Spalte rechts neben dem Attribut, schreibe *100* und drücke Enter. Du kannst den Betrag auch mit den kleinen Pfeiltasten neben dem Feld ändern.



Dem Attribut *npc_resref* wird jetzt ein NPC Spawnset zugewiesen. Von diesem NPC soll Geralt das Gold bekommen. Denke daran, dass der NPC genug Gold in seinem Inventar haben muss, um den Betrag abgeben zu können. Um den NPC zuzuweisen, klicke auf das kleine gelbe Icon in der Spalte rechts neben *npc_resref*:



Das übliche Windows-Fenster zum Laden von Dateien wird geöffnet. Öffne den Ordner Data\Templates\NPCSpawn, wähle die Datei *our_Siegfried.npc* und klicke auf *Öffnen*. Der NPC wird zugewiesen. Es muss kein Charakter zugewiesen werden - es genügt, wenn das Attribut *gold_amount* gesetzt ist. Wenn Du eine Aktion löschen möchtest, klicke auf das rote '-' neben der Aktion:

OnPhaseFinished Actions	
Available Actions	Transfer gold to pl +
1. Transfer gold to player	-
gold_amount	100
npc_resref	nasz_zygfryd

OnQuestFinished Actions Section:

In diesem Abschnitt wird definiert, welche Aktionen beim Abschluss einer bestimmten Quest ausgeführt werden. Das funktioniert genauso wie unter *OnPhaseFinished Actions* beschrieben. Auch hier fügen wir *Transfer gold to player* ein, allerdings soll Geralt diesmal 500 Orens bekommen:

OnQuestFinished Actions	
Available Actions	Transfer gold to pl +
1. Transfer gold to player	-
gold_amount	500
npc_resref	nasz_zygfryd

Wir müssen eine weitere Aktion hinzufügen - *Add XP to player*. Damit bekommt der Spielercharakter Erfahrungspunkte für die abgeschlossene Quest. Nachdem Du die Aktion *Add XP to player* eingefügt hast, gib neben dem Attribut *xp 500* ein, damit der Spielercharakter 500 Erfahrungspunkte bekommt. Der Abschnitt *OnQuestFinished Actions* sollte jetzt so aussehen:

OnQuestFinished Actions	
Available Actions	Transfer gold to pl +
1. Transfer gold to player	-
gold_amount	500
npc_resref	nasz_zygfryd

Das war alles zu den Attributen von *Quest Beginning*. Bevor wir weitermachen, schauen wir uns die Bedingungen genauer an. Bedingungen sind sowohl während der Start- als auch der Endphase einer Quest relevant:

OnQuestFinished Actions	
Available Actions	Add XP to player 
1. Transfer gold to player 	
gold_amount	500
npc_resref	nasz_zygfryd 
2. Add XP to player 	
xp	500

- Game Phase Is ==
- Game Phase Is >=
- Game Phase Is <=
- Quest Phase Is ==
- Quest Phase Is >=
- Quest Phase Is <=
- Characters Dead Less Than
- Character Is Dead
- Hero Has Item(s)
- Hero Doesn't Have Item(s)
- Hero Has Ability
- Hero Has Journal Entry
- Dialog Line Chosen
- Hero Has Gold
- Hero Doesn't Have Gold
- Hero Doesn't Have Journal Entry

Game Phase Is == - die Bedingung ist erfüllt, wenn die Spielphase den angegebenen Wert hat

Game Phase Is >= - die Bedingung ist erfüllt, wenn die Spielphase den angegebenen oder einen größeren Wert hat

Game Phase Is <= - die Bedingung ist erfüllt, wenn die Spielphase den angegebenen oder einen kleineren Wert hat

Quest Phase Is == - die Bedingung ist erfüllt, wenn die Questphase den angegebenen Wert hat

Quest Phase Is >= - die Bedingung ist erfüllt, wenn die Questphase den angegebenen oder einen größeren Wert hat

Quest Phase Is <= - die Bedingung ist erfüllt, wenn die Questphase den angegebenen oder einen kleineren Wert hat

Characters Dead Less Than - die Bedingung ist erfüllt, wenn mehr als die angegebene Anzahl von Charakteren noch am Leben ist

Character is Dead - die Bedingung ist erfüllt, wenn der betreffende Charakter tot ist.

Hero has Item(s) - die Bedingung ist erfüllt, wenn Geralt den angegebenen Gegenstand im Inventar hat.

Hero doesn't have item(s) - die Bedingung ist erfüllt, wenn Geralt den angegebenen Gegenstand **nicht** im Inventar hat.

Hero Has Ability - die Bedingung ist erfüllt, wenn Geralt die angegebene Fähigkeit hat

Hero Has Journal Entry - die Bedingung ist erfüllt, wenn Geralt den angegebenen Tagebucheintrag hat

Dialog Line Chosen - die Bedingung ist erfüllt, wenn Geralt die angegebene Dialogoption wählt

Hero Has Gold - die Bedingung ist erfüllt, wenn Geralt den angegebenen Betrag bei sich hat

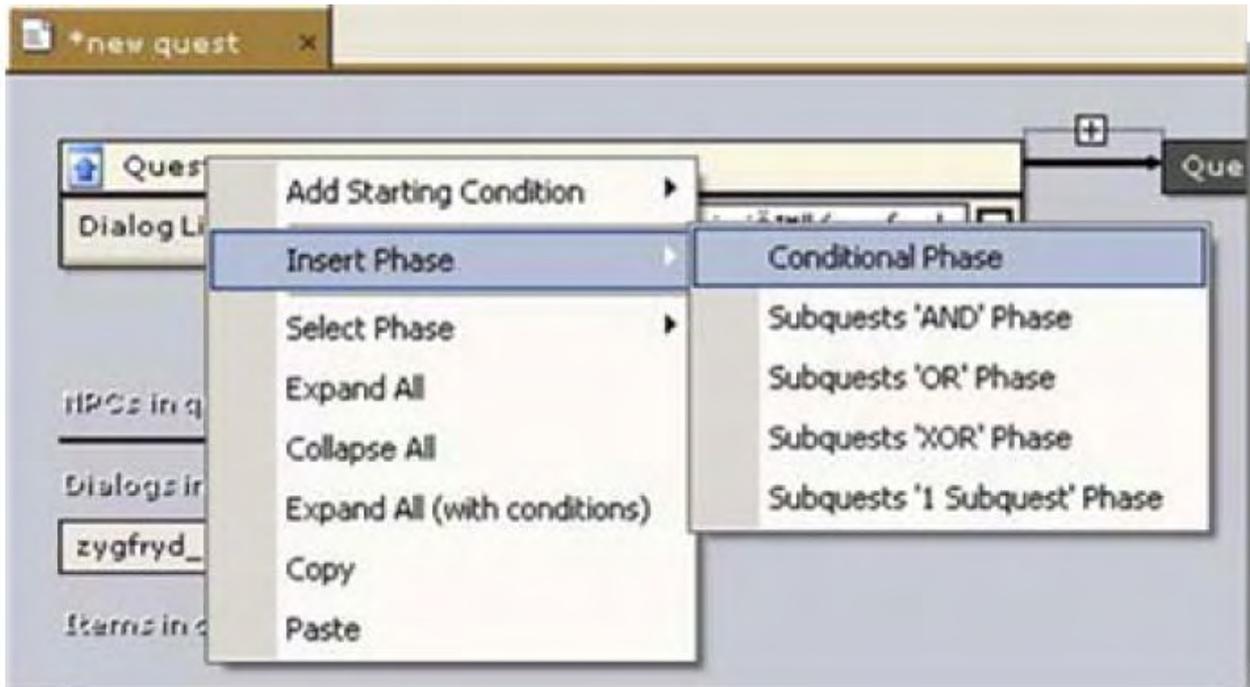
Hero Doesn't Have Gold - die Bedingung ist erfüllt, wenn Geralt den angegebenen Betrag **nicht** bei sich hat

Hero Doesn't Have Journal Entry - die Bedingung ist erfüllt, wenn Geralt den angegebenen Tagebucheintrag **nicht** hat.

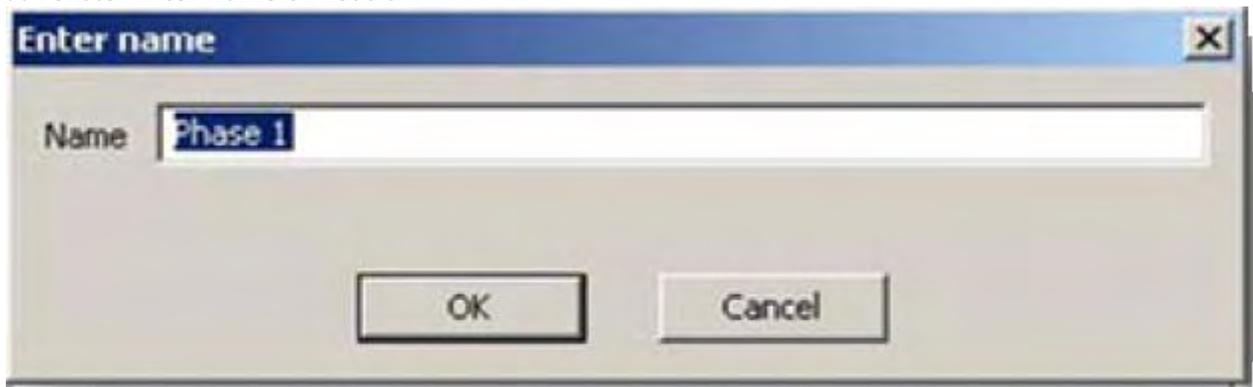
Zurück zu unserer Quest. Als nächstes wollen wir eine neue Phase erstellen. Klicke dazu mit der rechten Maustaste auf *Quest Beginning*.



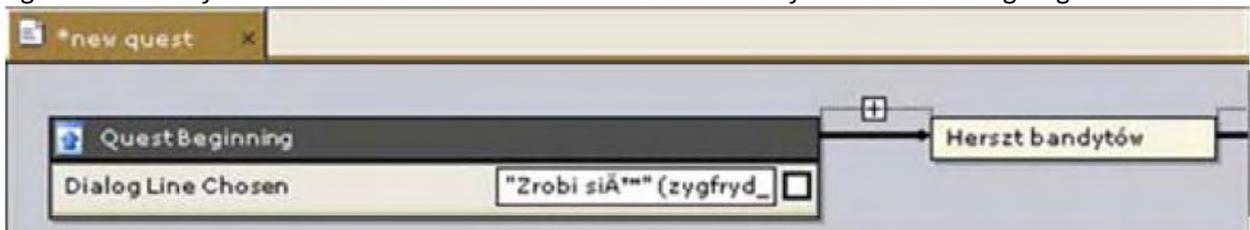
Wähle **Insert Phase -> Conditional Phase** aus dem Menü:



Das Fenster **Enter Name** öffnet sich:



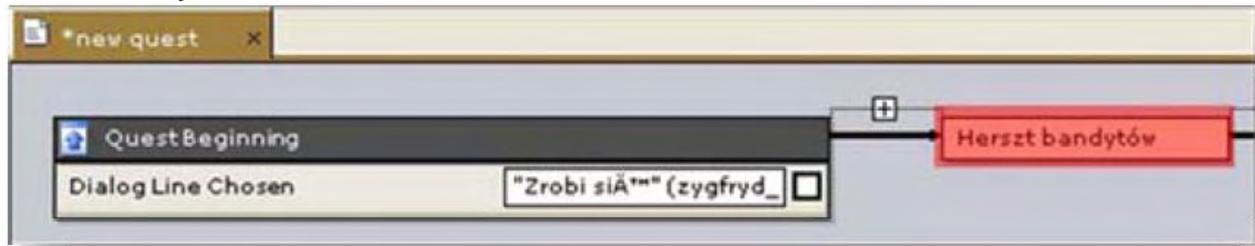
Trage einen Namen für die Phase ein. Denk daran, dass jede Phase einen anderen Namen haben muss. Trage **"Banditenanführer"** ein. Klicke auf OK. Die Phase **"Banditenanführer"** wird hinzugefügt:



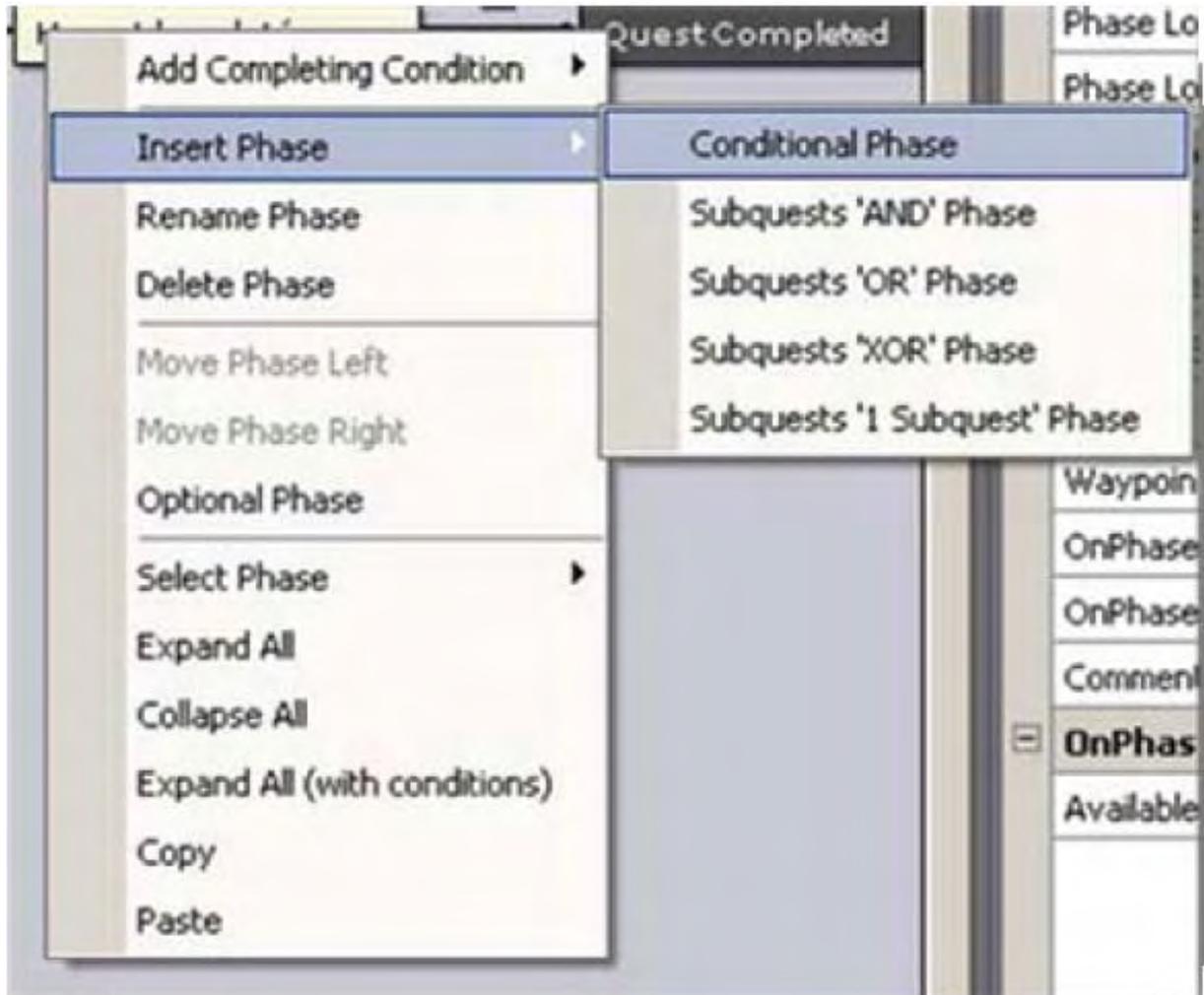
Natürlich können mehrere Phasen hinzugefügt werden. Wenn Du vor der Phase **"Banditenanführer"** eine weitere Phase hinzufügen möchtest, klicke mit der rechten Maustaste auf **Quest Beginning**:



Wenn Du der neuen Phase nach "Banditenanführer" einfügen möchtest, klicke mit der rechten Maustaste auf "Banditenanführer".



Wähle **Conditional Phase** aus dem Menü:



Das Fenster **Enter Name** öffnet sich wieder. Gib den Namen der Phase ein und klicke auf OK. Dies fügt die nächste Phase hinzu:



In beiden Fällen haben wir eine bedingte Phase hinzugefügt. Dies heißt, dass die Phase nur abgeschlossen wird, wenn die Bedingung erfüllt wird. Wir wählen diese Option, weil wir davon ausgehen, dass der Banditenanführer in der Phase getötet wird. Anstelle einer bedingten Phase könnten wir andere Phasen hinzufügen, je nachdem, was für unsere Quest sinnvoll ist. Die folgenden Phasen sind verfügbar:

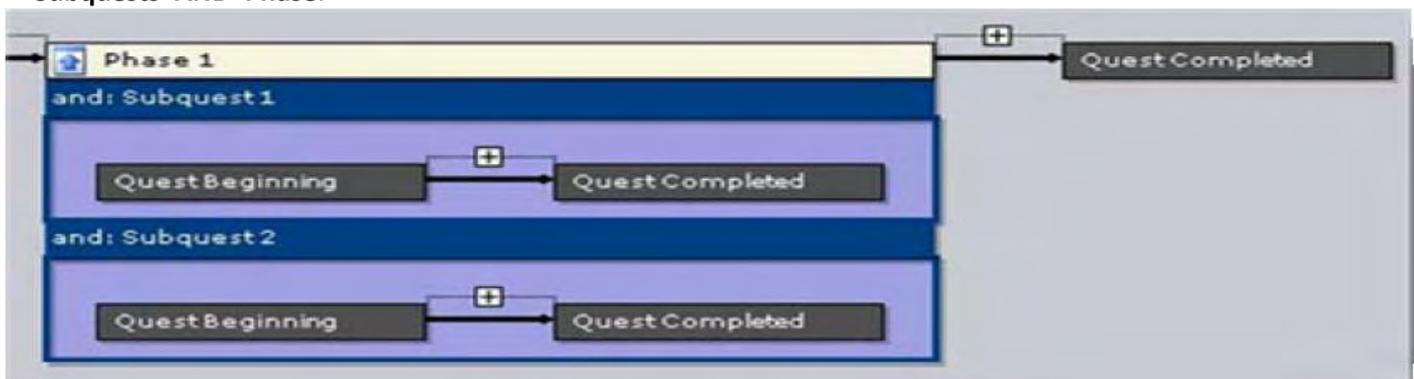
- Conditional Phase
- Subquests 'AND' Phase
- Subquests 'OR' Phase
- Subquests 'XOR' Phase
- Subquests '1 Subquest' Phase

-> Conditional Phase:



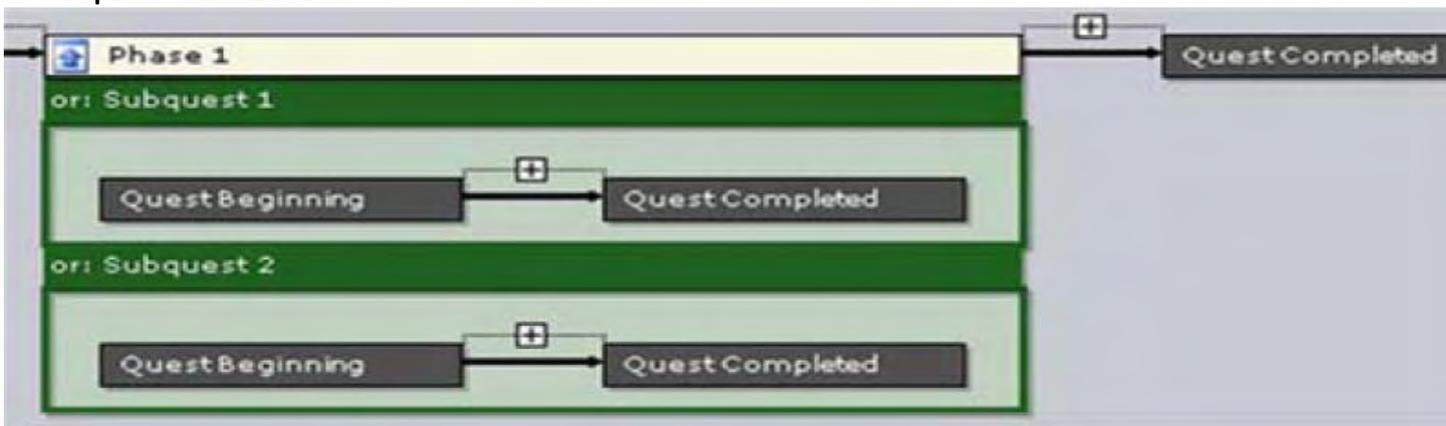
Diese Phase kann nur abgeschlossen werden, wenn alle Bedingungen erfüllt sind,

-> Subquests 'AND' Phase:



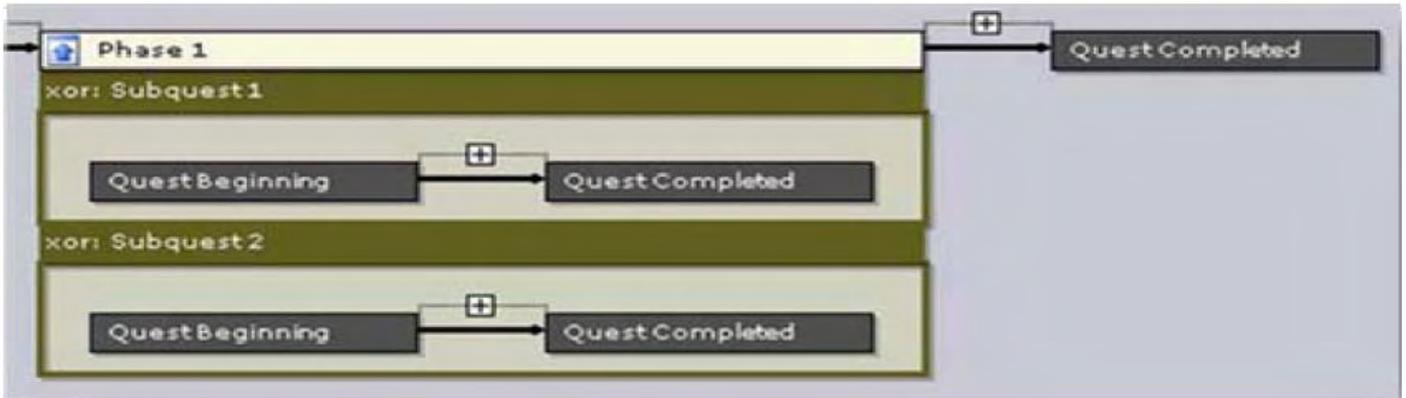
Diese Phase kann nur abgeschlossen werden, wenn alle Subquests abgeschlossen sind. Du kannst hier mehrere Subquests angeben und sie nach Bedarf hinzufügen oder entfernen. Jede Subquest kann wiederum mehrere Phasen haben.

-> Subquests 'OR' Phase:



Diese Phase hat Ähnlichkeit mit **Subquests 'AND' Phase**. Der einzige Unterschied ist, dass es genügt, eine der Subquests abzuschließen, damit die Phase abgeschlossen wird. Die übrigen Subquests können abgeschlossen werden, müssen es aber nicht. Auch hier kann es mehrere Subquests geben, und jede kann aus mehreren Phasen bestehen.

-> **Subquests 'XOR' Phase:**



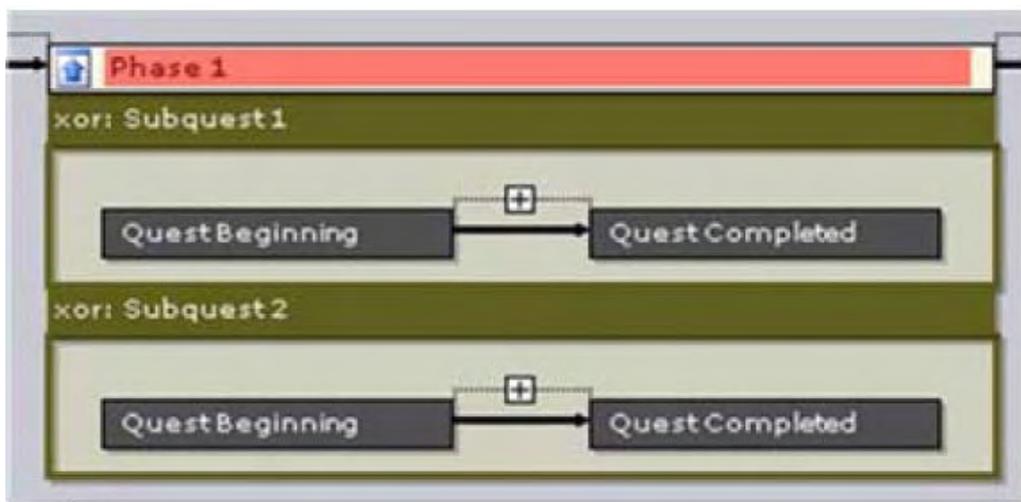
Diese Phase ähnelt den beiden vorherigen, allerdings muss hier genau eine Subquest abgeschlossen sein, damit die Phase abgeschlossen ist. Die übrigen Subquests können nicht abgeschlossen werden.

-> **Subquests '1 Subquest' Phase:**



Diese Phase wird abgeschlossen, wenn eine Subquest abgeschlossen ist. Die Phase kann aus mehreren anderen Phasen bestehen.

In jedem Fall müssen wir zuerst "Phase 1" in das *Enter Name* Fenster schreiben. Jetzt schauen wir uns die Menüs *Subquest 'AND' Phase*, *Subquest 'OR' Phase* und *Subquest 'XOR' Phase* genauer an. Klicke mit der rechten Maustaste auf den Namen der Phase:



Das folgende Menü öffnet sich:



Add Subquest - fügt eine weitere Subquest hinzu

Change Subquest Color - ändert die Farbe der Subquest

Convert to 'AND' Phase - macht aus der Phase eine Phase vom Typ *Subquest 'AND' Phase* (wenn sie diesen Typ nicht bereits hat)

Convert to 'OR' Phase - macht aus der Phase eine Phase vom Typ *Subquest 'OR' Phase* (wenn sie diesen Typ nicht bereits hat)

Convert to 'XOR' Phase - macht aus der Phase eine Phase vom Typ *Subquest 'XOR' Phase* (wenn sie diesen Typ nicht bereits hat)

Insert Phase - fügt eine weitere Phase ein

Rename Phase - benennt die Phase um

Delete Phase - löscht eine Phase

Move Phase Left - bewegt eine Phase nach links (wenn möglich)

Move Phase Right - bewegt eine Phase nach rechts (wenn möglich)

Optional Phase - markiert eine Phase als optional. Das bedeutet, die Phase kann abgeschlossen werden, muss es aber nicht, um den Plot voranzubringen.

Select Phase - wähle eine Phase aus

Expand All - expandiert eine Phase

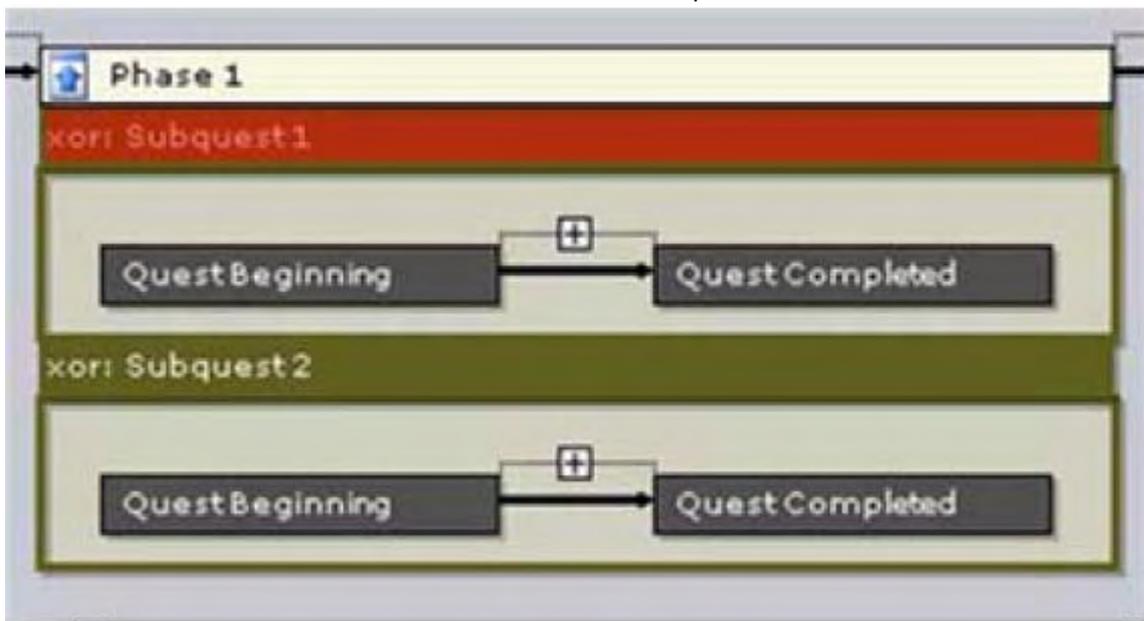
Collapse All - minimiert eine Phase

Expand All (with conditions) - expandiert eine Phase zusammen mit allen Bedingungen

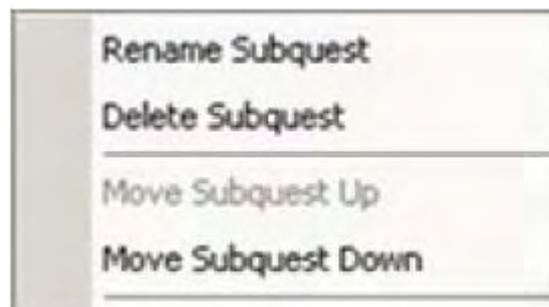
Copy - kopiert eine Phase

Paste - fügt eine Phase ein

Wenn Du mit der rechten Maustaste auf den Namen einer Subquest klickst:



Das folgende Menü öffnet sich:



- *Rename Subquest* - benennt eine Subquest um,

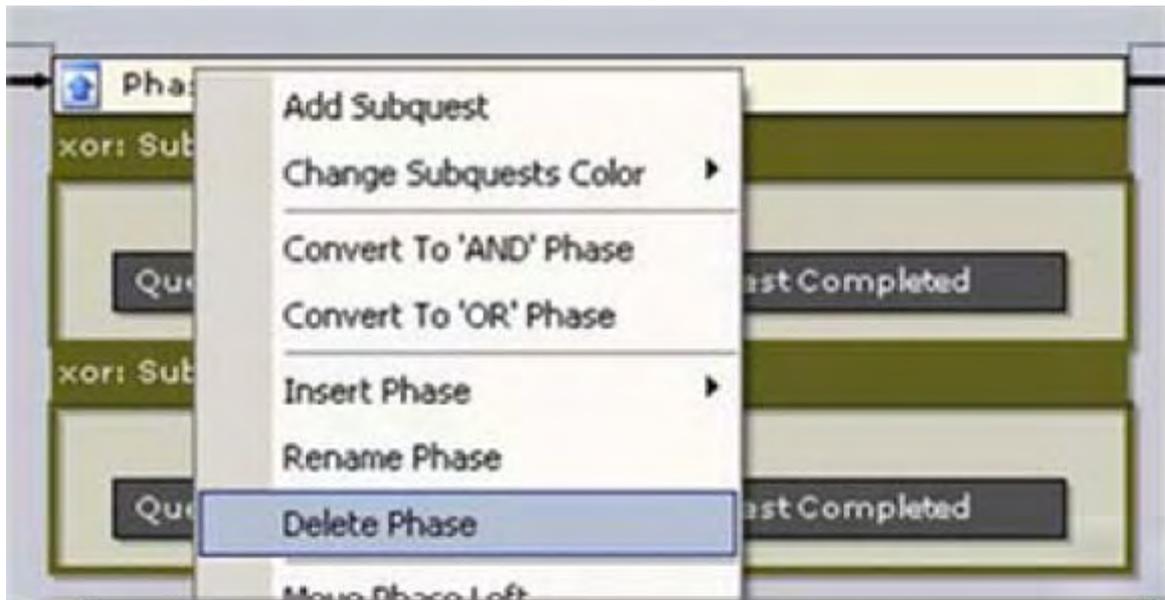
- *Delete Subquest* - lösche eine Subquest,

- *Move Subquest up* - verschiebe die Subquest nach oben (wenn möglich),

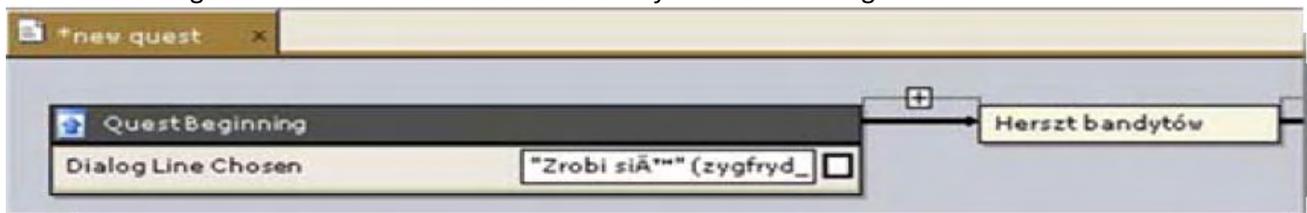
- *Move Subquest Down* - verschiebe die Subquest nach unten (wenn möglich).

Wir brauchen diese Phase jedoch nicht, also können wir sie löschen. Klicke mit der rechten Maustaste auf

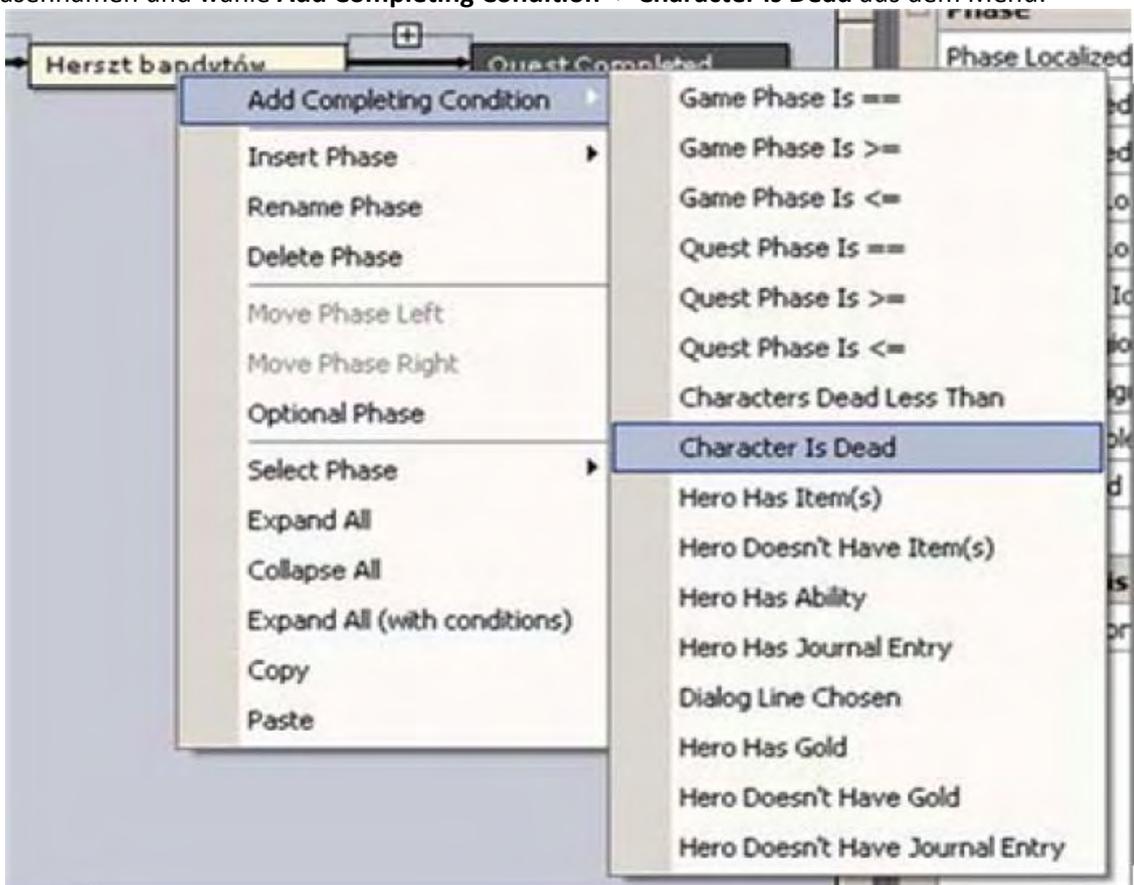
den Namen der Phase und wähle *Delete Phase* aus dem Menü:



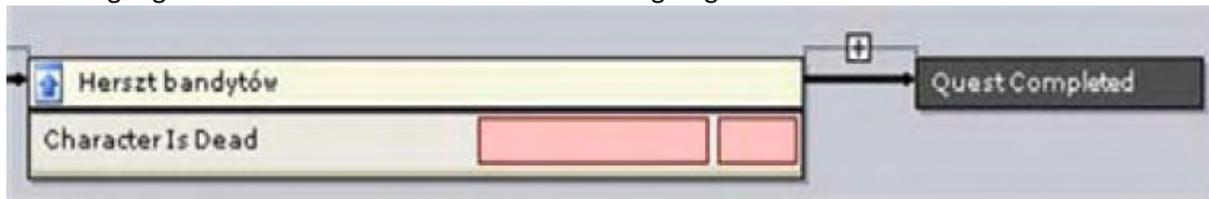
Die Phase wird gelöscht und nur die Phase *Banditenanführer* bleibt übrig:



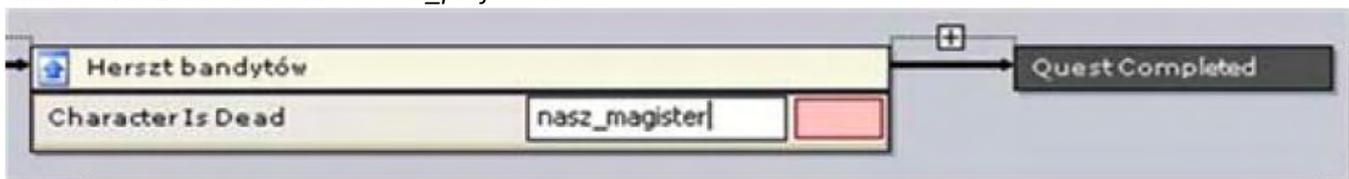
Weil die Banditenanführer Phase eine bedingungsabhängige Phase ist, müssen wir jetzt eine abschließende Bedingung hinzufügen. Diese Bedingung spezifiziert was notwendig ist, um die Phase abzuschließen. In unseren Fall muss der Banditenanführer von Geralt erschlagen werden. Rechtsklick auf den Banditenanführer Phasennamen und wähle **Add Completing Condition** -> **Character is Dead** aus dem Menü:



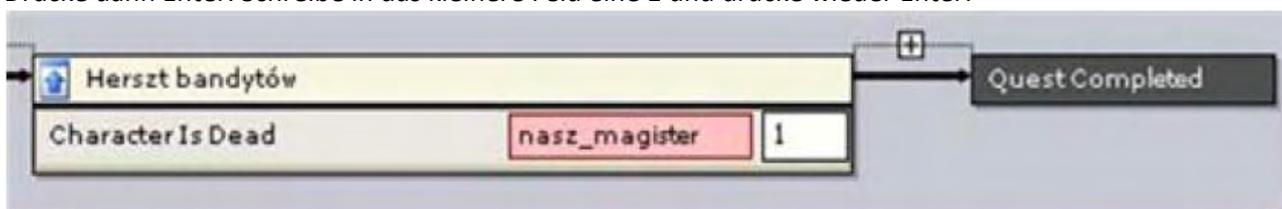
Diese Bedingung wird der Banditenanführer- Phase hinzugefügt:



Jetzt müssen wir den Charaktertyp und die zu tötende Anzahl angeben, die nötig ist, um die Bedingung zu erfüllen. In unserem Fall muss Geralt den Banditenanführer töten. Die notwendige Anzahl lautet 1. Zur Erinnerung: Wir nannten den Banditenanführer *our_professor*. Klicke auf das größere Feld rechts neben **Character Is Dead** und schreibe *our_professor*:



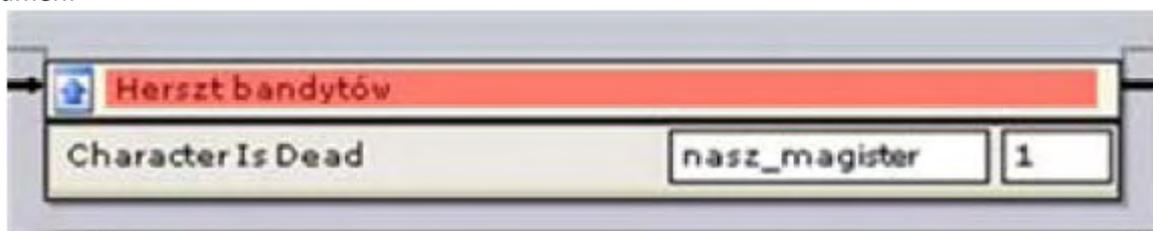
Drücke dann Enter. Schreibe in das kleinere Feld eine 1 und drücke wieder Enter:



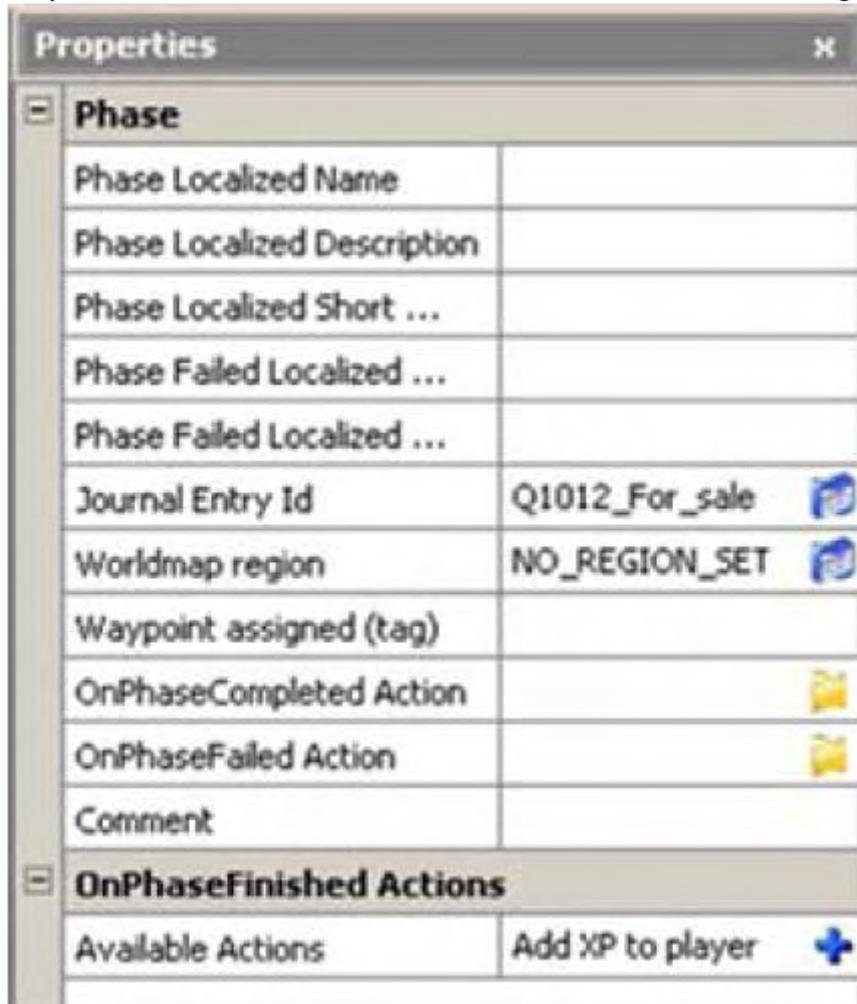
Die Banditenanführer- Phase wird abgeschlossen, wenn **ein** Charakter mit der Bezeichnung *our_professor* getötet wurde.

Wichtig: Die Anzahl an Charakteren, die getötet werden müssen, um diese Bedingung zu erfüllen, bezieht sich nicht nur auf einen einzelnen Spielabschnitt, sondern global auf das gesamte Spiel. Wenn Geralt z. B. in einer Quest fünf Wölfe töten muss, und in der nächsten sieben, dann ist die globale Bedingung erst erfüllt, wenn in Summe 12 (nicht sieben) Wölfe getötet wurden. Es ist geschickter für jede eigenständige Quest einen neuen Charaktertyp zu wählen.

Jetzt müssen wir die Attribute der Banditenanführer- Phase setzen. Klicke auf den Banditenanführer- Phasennamen:



Gehe dann in das **Properties**-Fenster, in dem die Banditenanführer- Phasenattribute angezeigt werden:



Da die Attribute praktisch gleich denen, unter **Quest Beginning** sind, werden sie hier nicht noch einmal beschrieben. Gib folgendes ein:

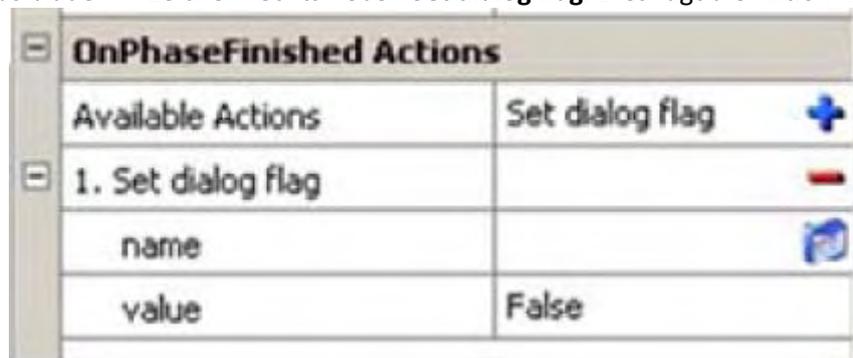
- **Phase Localized Name** – *Banditenanführer*
- **Phase Localized Description** – “ *Der Banditenanführer ist tot. Jetzt sollte es hier friedlich sein.*“
- **Phase Localized Short Description** - “ *Ich habe den Irren getötet. Nun sollte ich zu Siegfried gehen.*“

Lass die übrigen Attribute unverändert. Wir müssen lediglich Aktionen im **On Phase Finished Actions** Bereich definieren. Die Aktionen werden ausgeführt, wenn die Banditenanführer Phase abgeschlossen ist, in unseren Fall, nachdem der Banditenanführer getötet wurde. Hier ist unsere einzige Aktion, eine **Dialog flag** (Marker) zu setzen. Wie Du dich erinnerst, haben wir die **global flag leader_killed** auf 1 (true) gesetzt. Wegen dieser global flag, können wir, nachdem der Banditenanführer getötet wurde, mit Siegfried sprechen und von unserem Erfolg berichten.

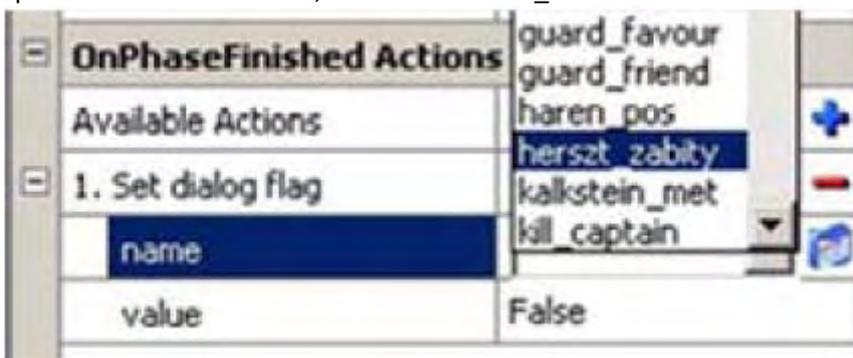
Klicke in die Spalte rechts neben **Available Actions** und wähle **Set dialog flag** aus:



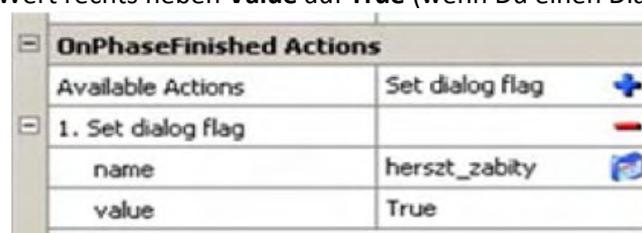
Klicke dann auf das blaue "+" Zeichen rechts neben **Set dialog flag**. Dies fügt die Aktion hinzu.



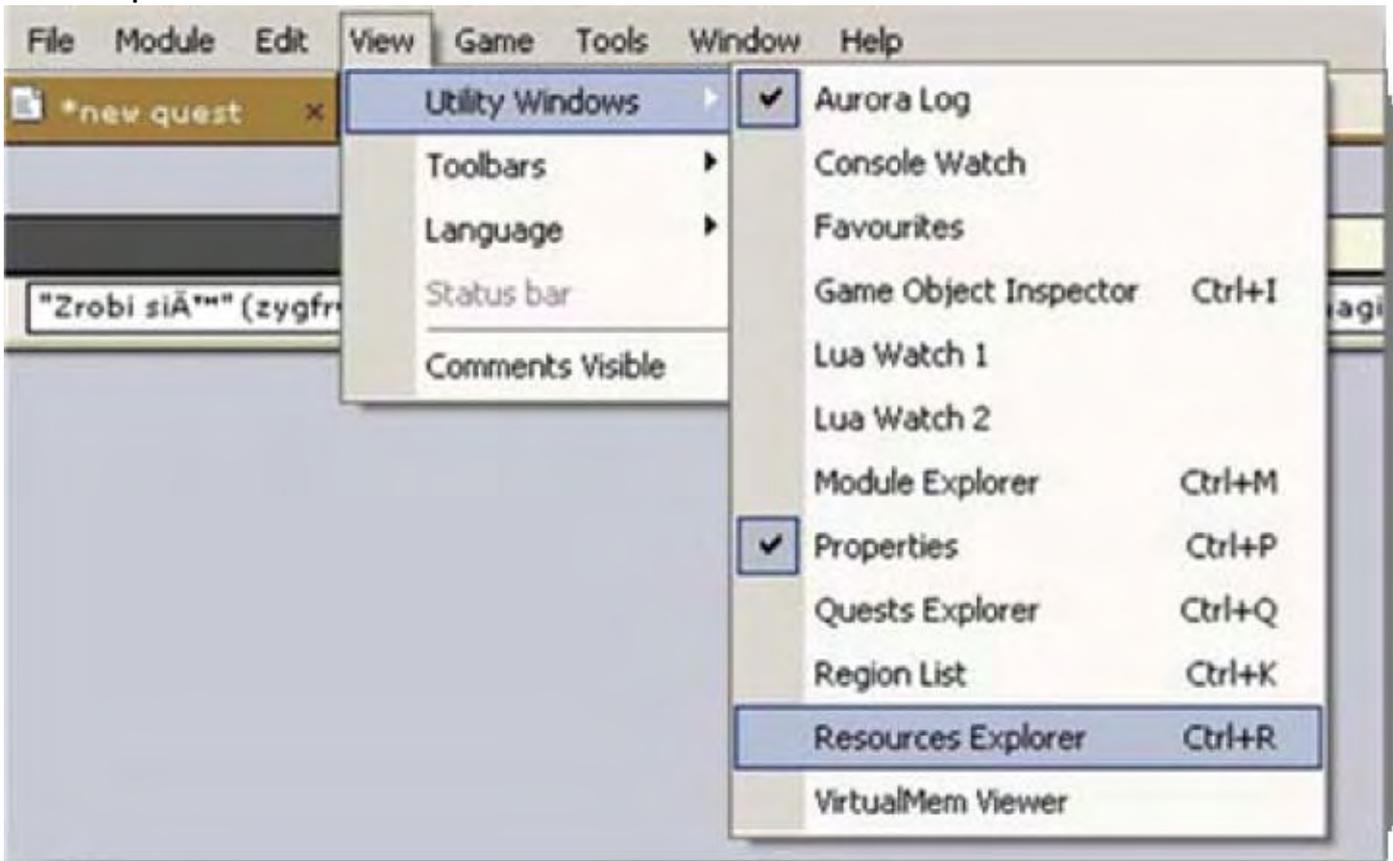
Klicke jetzt in die Spalte rechts neben **name**, und wähle *leader_killed* aus der Liste aus:



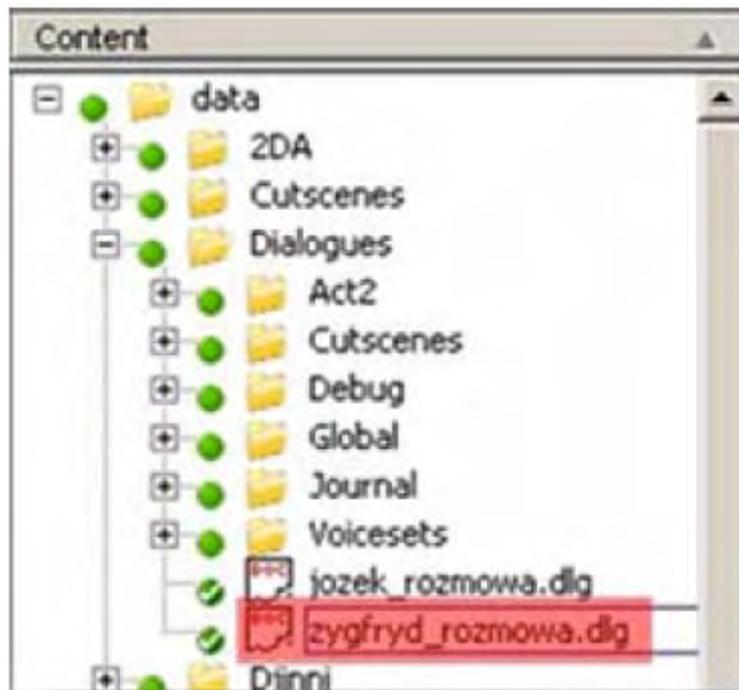
Nimm zur Kenntnis, dass **flags**, wie *orders_received* und *results_reported*, nicht in der Liste erscheinen. Diese **flags** sind **lokale (local) flags** die nur bei der Erstellung der zugehörigen Dialogdateien verfügbar sind. Wir definierten allerdings die *leader_killed* **flag** in der *globaldialogflags.2da* Datei. Somit ist sie eine **global flag** die auch außerhalb unseres Gespräches verfügbar ist. Sie kann sowohl zur Dialog-, als auch zur Questerstellung genutzt werden. Klicke auf das kleine blaue Symbol neben der Liste und öffne die *globaldialog.2da* Datei. Setze den Wert rechts neben **Value** auf **True** (wenn Du einen Dialog erstellst, wäre es die 1):



Die Banditenanführer Phase ist fertig. Wir müssen jetzt die letzte Phase erstellen, mit der wir die Quest beenden wollen. Bevor wir das machen, lass uns den Dialog mit Siegfried ein bisschen modifizieren. Wenn der **Resources Explorer** geschlossen ist, öffne ihn aus dem Menü. Wähle **View -> Utility Windows -> Resources Explorer**:



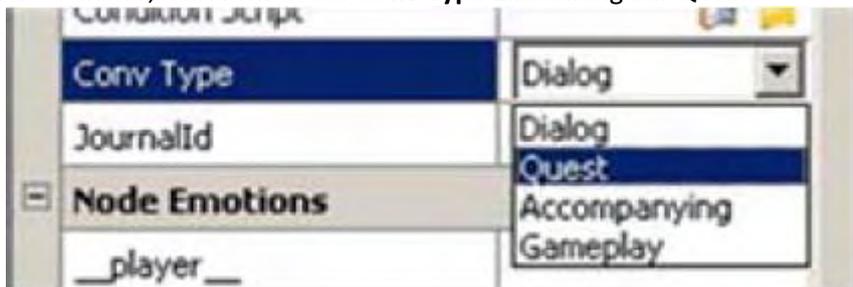
Öffne den **Dialogue** Zweig im **Resources Explorer**. Wähle dann die *siegfried_conv.dlg* Datei mit einem Doppelklick aus:



Ein Fenster, das unser Gespräch mit Siegfried enthält, wird geöffnet. Markiere jetzt den "Ja, der Bastard ist tot" Eintrag:



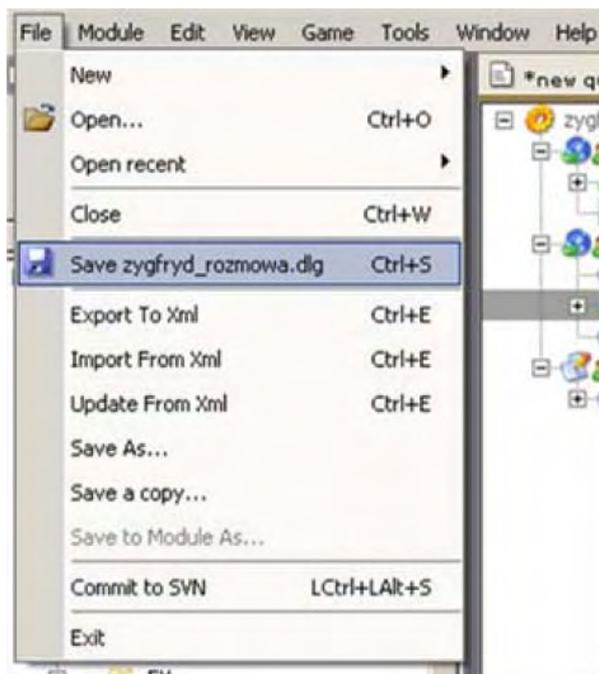
Ändere im **Properties** Fenster, das Attribut der **Conv Type** Einstellung auf **Quest**:



Schreibe in das Fenster rechts neben **Quest Id** *Leader Dead*:

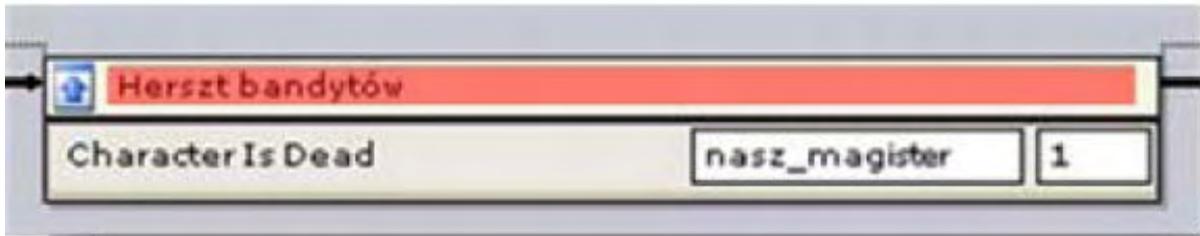


Abschließend müssen wir die Änderung noch speichern. Wähle dafür im Menü, **File -> Save siegfried_conv.dlg** aus:

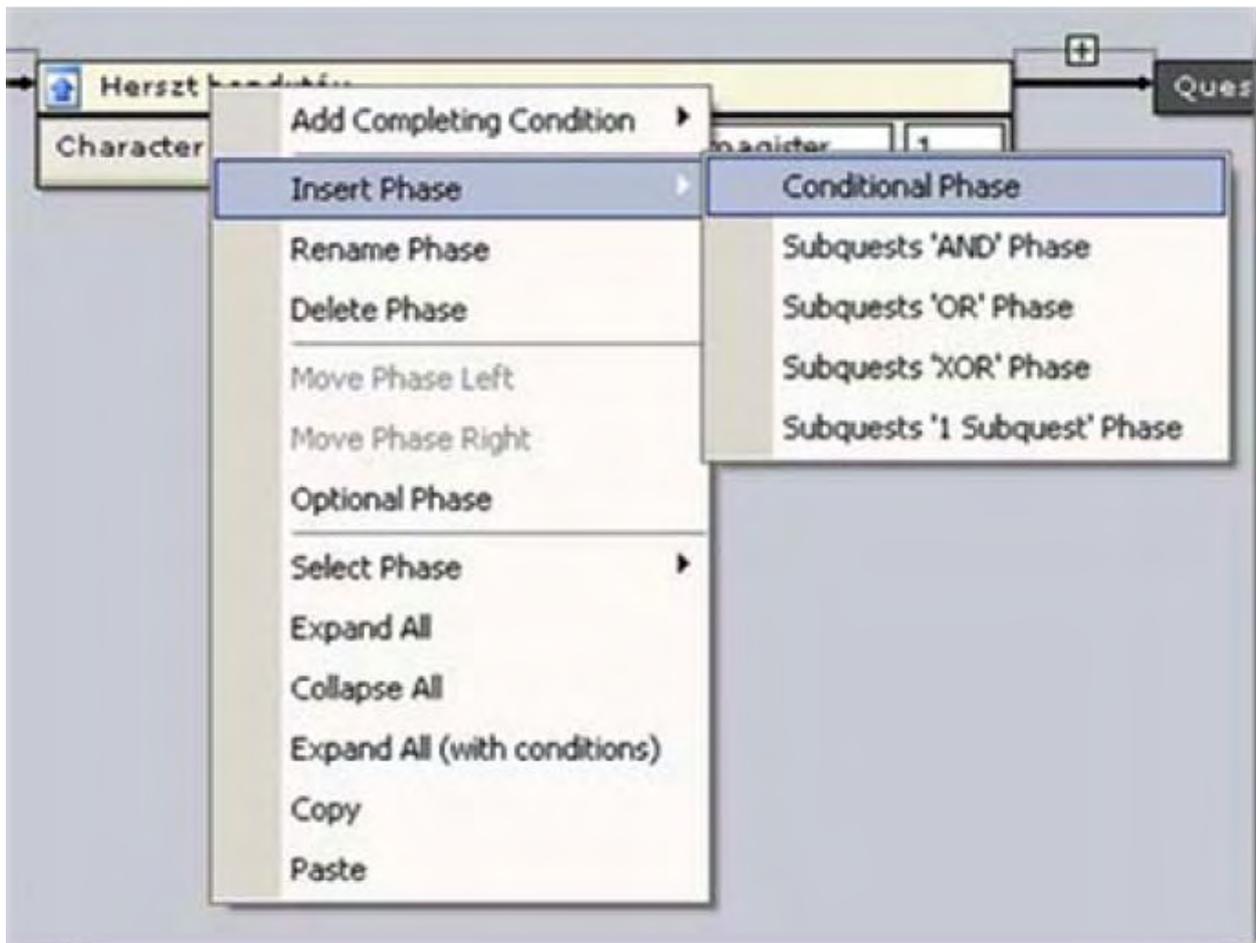


Wenn unter **File** nicht **Save siegfried_conv.dlg** wählbar ist, oder ein andere Dateiname angezeigt wird, ist das **siegfried_conv.dlg** Fenster im Moment nicht als aktives Fenster ausgewählt. Klicke in diesem Fall auf die Titelzeile vom **siegfried_conv.dlg** Fenster und wähle dann **Save siegfried_conv.dlg** im **File** Menü aus.

Wir haben diese Änderung am Dialog vorgenommen, damit wir unsere Quest mit der Dialogzeile *"Ja, der Bastard ist tot"* beenden können. Lass uns jetzt die letzte Phase unserer Quest hinzufügen. Klicke dafür mit einem Rechtsklick auf den Banditenanführer- Phasenamen:

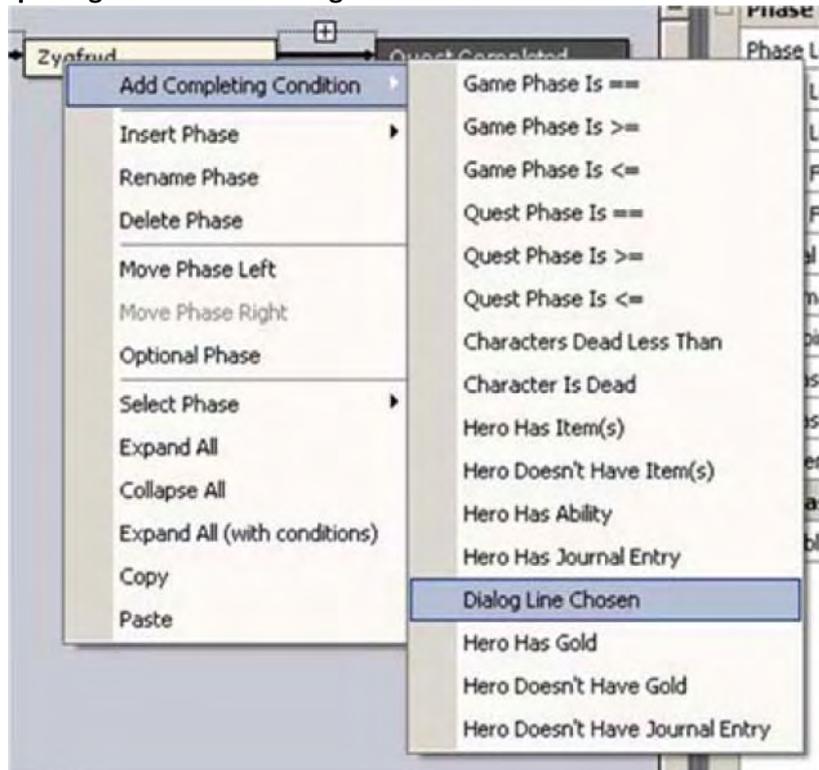


Markiere **Insert-Phase- > Conditional Phase** aus dem Menü:



Schreibe, Siegfried, in das **Enter Name** Fenster und klicke auf OK. Die Siegfried- Phase ist jetzt hinzugefügt.

Jetzt sollten wir eine Abschlussbedingung hinzufügen. Rechtsklicke dafür auf den *Siegfried* Phasenamen und wähle **Add Completing Condition -> Dialog Line Chosen** aus dem Menü:

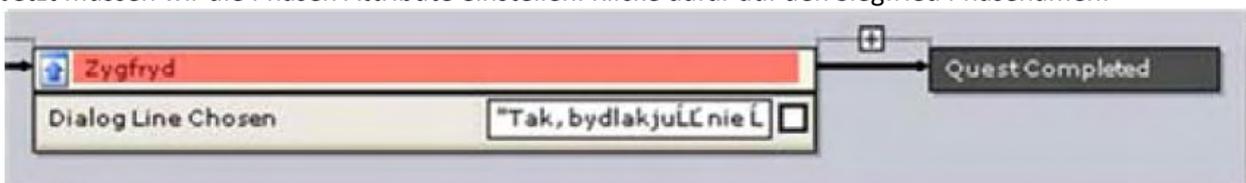


Auf diese Weise wird die Bedingung zugefügt. Doppelklicke in das Feld neben die **Dialog Line Chosen** Bedingung. Das **Select** Fenster öffnet sich. Klicke auf das "+" Zeichen vor **siegfried-conv** um diesen Zweig zu öffnen. Doppelklicke auf den "Ja, der Bastard ist tot" Eintrag:



Das ist der Dialog Eintrag das wir gerade geändert haben. Die Siegfried-Phase wird beendet, wenn vom Spieler der "ja, der Bastard ist tot" Eintrag im Dialog mit Siegfried ausgewählt wurde.

Jetzt müssen wir die Phasen Attribute einstellen. Klicke dafür auf den Siegfried Phasenamen:



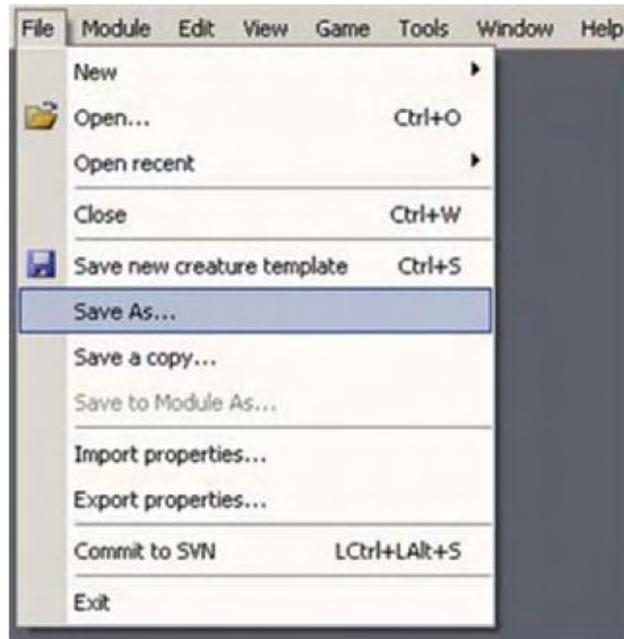
Gehe jetzt zum **Properties** Fenster und stelle folgende Attribute ein:

- **Phase Localized Name** – *Siegfried*,

- **Phase Localized Description** – *Siegfried war mit dem Ergebnis zufrieden.*

- **Phase Localized Short Description** – *Ich habe 500 Orens von Siegfried erhalten. Es hat sich gelohnt.*

Lass die restlichen Attribute unverändert. Unsere Quest ist fertig und wir können jetzt speichern. Wähle dafür **Save As...** aus dem **File** Menü:



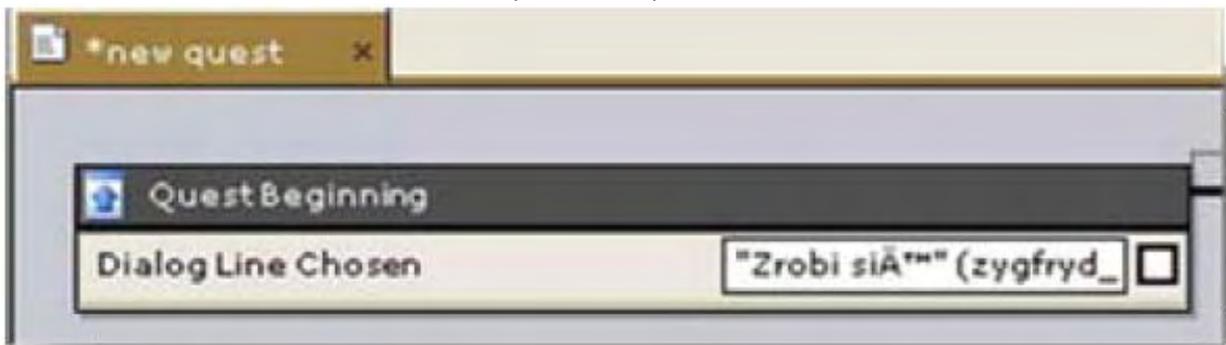
Es öffnet sich das Standard Windows-Fenster, zum speichern von Dateien. Gehen zum `\Data\Quests` Ordner und gib `our_task` ein. Klicke dann auf Speichern. Erinnerung: Der Dateiname darf nicht mehr als 16 Zeichen haben. Achte darauf, dass die Datei im `.qst` Format gespeichert wird:



Wenn für die Datei ein anderer Speichertyp, als `.dlg`, angeboten wird, war das `new quest` Fenster nicht das aktiv markierte Fenster. Achte vor der Speicherung darauf, dass das `new quest` Fenster aktiv ist. Inaktive Fenster haben eine graue Titelleiste:



Bei einer anderen Farbe ist das Fenster aktiv (hier: braun):



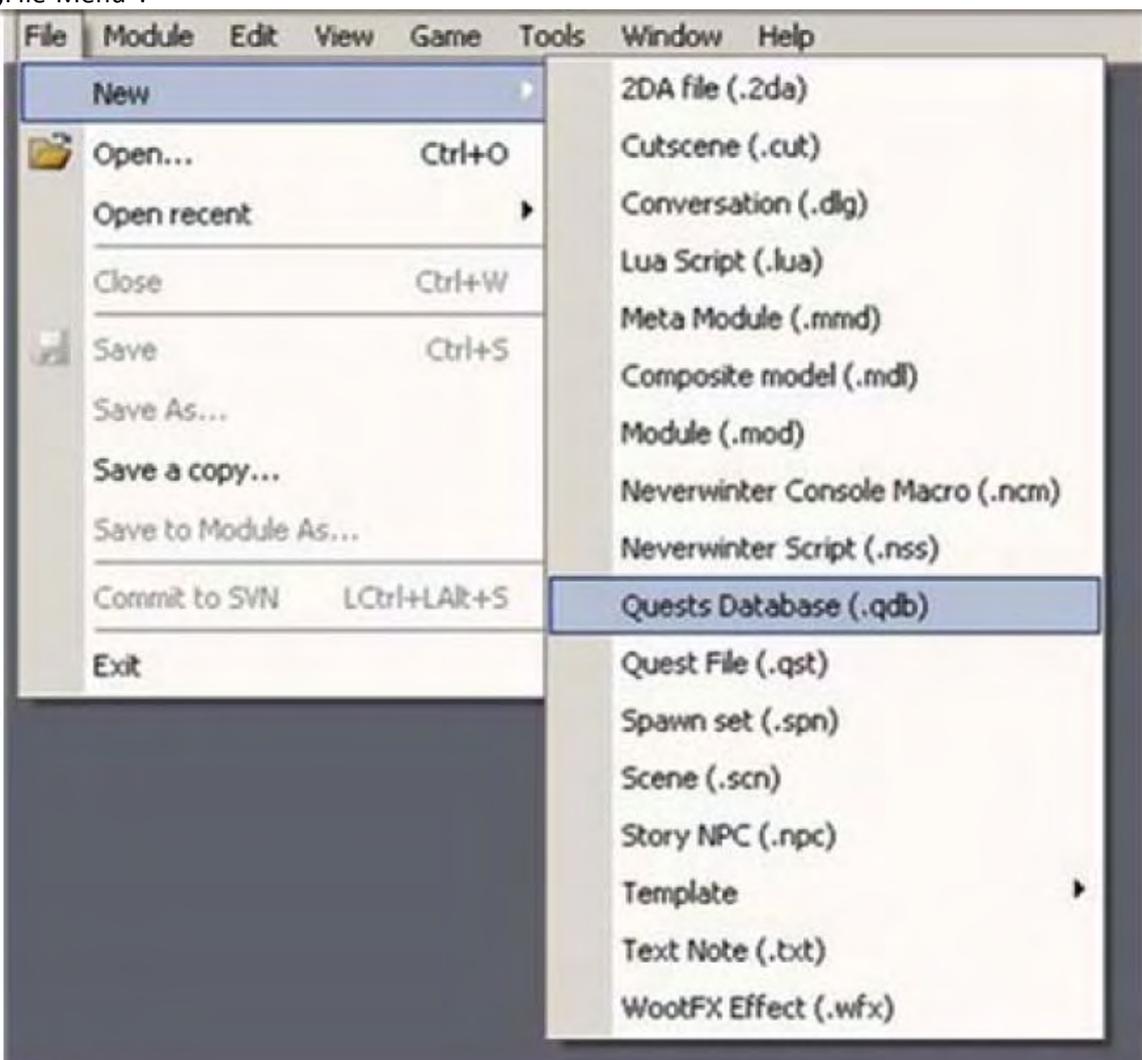
Ein Sternchen vor dem Namen in der Titelleiste bedeutet, dass die Datei seit der letzten Änderung noch nicht gespeichert worden ist.

So ist es gut. Die Quest ist jetzt bereit für das Abenteuer das wir erstellt.

QUEST DATABASES

Jetzt werden wir uns mit der Entwicklung von Quest-Datenbanken befassen.. Wofür brauchen wir die? Die Antwort ist ziemlich einfach: Um Quests besser und einfacher zu verwalten.

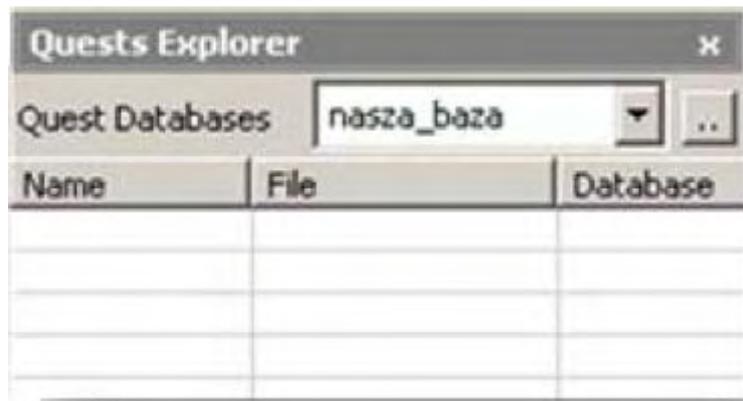
In unserem Fall ist eine Datenbank nicht wirklich notwendig, weil wir nur eine Quest haben. Aber stell dir vor, dass wir 50 Quests hätten und jede einzelne öffnen wollten. Im Normalfall würde das einige Zeit in Anspruch nehmen. Um eine neue Datenbank zu erstellen wählst Du: New -> Quests Database (.qdb) aus dem „File-Menü“:



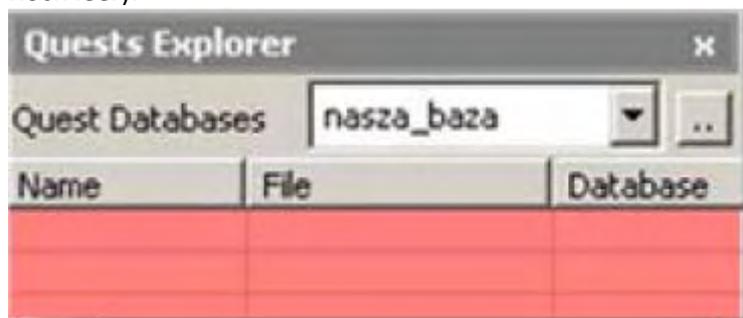
Es öffnet sich das Windows Fenster zum Speichern von Dateien. Wechsele zum Data\Quests Ordner (Hier findest du sowohl Quests als auch Quest-Datenbanken), Gib „our_base“ ein und wähle save aus. Denk daran, dass die Namen der Dateien die 16 Zeichen nicht überschreiten dürfen, Achte darauf, dass die Datei im Qdb-Format gespeichert wird:



Der Quest-Explorer öffnet sich auf der linken Seite des Bildschirms:



Jetzt wollen wir unsere Quest zur Datenbank hinzufügen. Rechtsklicke auf das Feld mit den Datenbank-Inhalten(zur Zeit ist es noch leer):



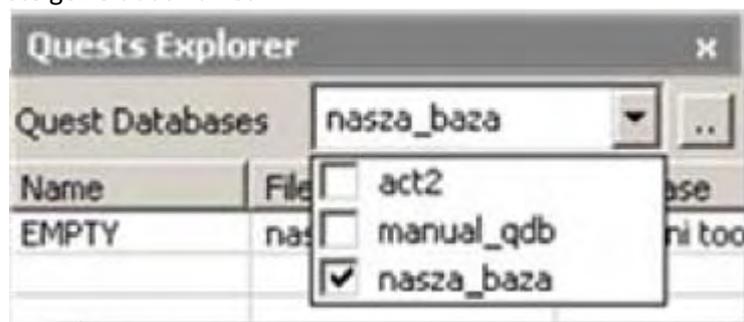
Wähle **Add Existing Quest To** -> **our_base** aus dem Menü:



Das Windows Fenster zum laden von Dateien öffnet sich. Wechsele in den Data\Quests Ordner, markiere our_task.qst und wähle **Open**, Die Quest ist jetzt unserer Datenbank hinzugefügt:

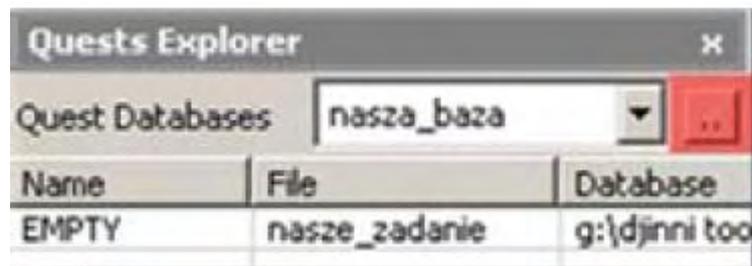


Die Datenbank wird automatisch aktualisiert, Diese Änderungen brauchen wir (im Gegensatz zur anderen Dateien) nicht speichern. Die Datenbank kann jetzt in unserem Modul verwendet werden. Wenn mehr als eine Datenbank existiert, kannst du ihre Inhalte durchsuchen, indem Du die entsprechende Datenbank in der Quest-Database Liste gezielt auswählst:



Wähle jetzt die Datenbank aus, die du durchsuchen möchtest. Klicke in die kleinen Kästchen vor den jeweiligen Datenbanken die Du durchsuchen möchtest.

Du kannst auf diese Weise mehrere gleichzeitig durchsuchen. Du kannst auch andere Operationen auf einer bereits geöffneten Datenbank ausführen. Klicke dafür auf das Kästchen mit den beiden Punkten rechts neben der Datenbankliste:



Ein Menü öffnet sich:



-> **Load Database** – Lädt eine Datenbank, Die sich noch nicht im Quest Explorer Fenster befindet.

-> **Unload Database** – Löscht eine Datenbank aus dem Quest Explorer Fenster, dieses löscht aber nicht die Datenbank an sich.

-> **Create New Database** – Erstellt eine neue Quest Datenbank.

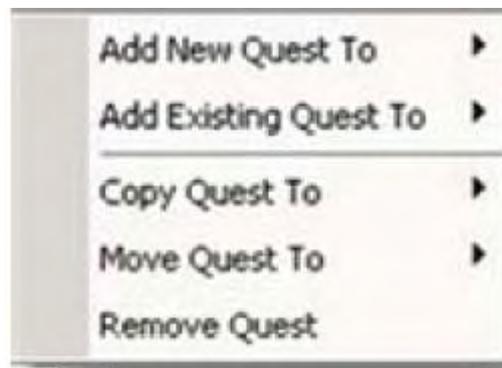
-> **Delete Database** – Löscht eine Quest Datenbank. Für diese Option ist die Bestätigung einer Sicherheitsabfrage erforderlich:



Wenn du auf Yes klickst, wird die Datenbank gelöscht, Du kannst auch Änderungen in Verbindung mit einzelnen Quests durchführen, Rechtsklicke auf Quest:



Ein großes Menü öffnet sich:



-> **Add New Quest To** – Fügt eine neue Quest zur Datenbank hinzu. Der Vorgang gleicht dem bekannten: -> New -> Quest File (.qst) aus dem File Menü. Der Unterschied besteht darin, dass hier Name und Zielordner angegeben werden müssen.

-> **Add Existing Quest To** – Fügt eine vorhandene (bereits erstellte) Quest der Datenbank hinzu.

-> **Copy Quest To** – Kopiert eine Quest zur einer anderen Datenbank.

-> **Move Quest To** –bewegt eine Quest in eine andere Datenbank.

-> **Remove Quest** – entfernt eine bestimmte Quest aus der Datenbank (aber es löscht nicht die Quest Datei an sich.)

Damit ist das Kapitel Quest Datenbanken abgeschlossen und wir können zum nächsten übergehen.

Running and testing the game

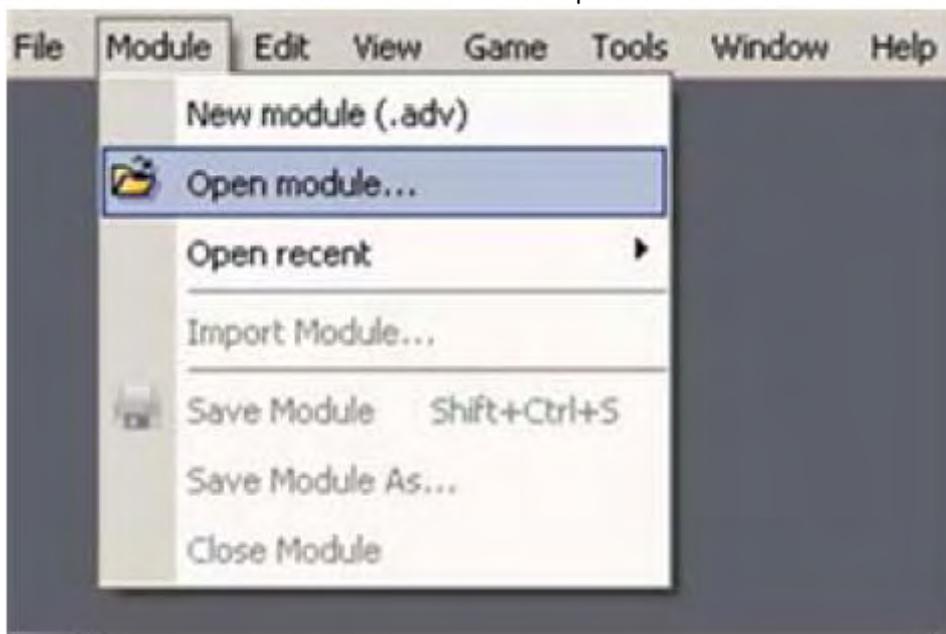
Einleitung:

Es ist jetzt Zeit für den nächste Schritt bei der Erstellung eines Abenteuers. Wir haben Charaktere, ein Spawn System, Gespräche und eine einfache Quest. Jetzt müssen wir alles zusammenfügen und einen Testlauf starten.

ASSIGNING AND ADDING FILES TO A MODULE

Wir müssen jetzt alle Komponenten, die wir erstellt haben. Den Attributen unseres Moduls hinzufügen. Wenn wir das nicht tun. Wird das Modul nicht korrekt laufen.

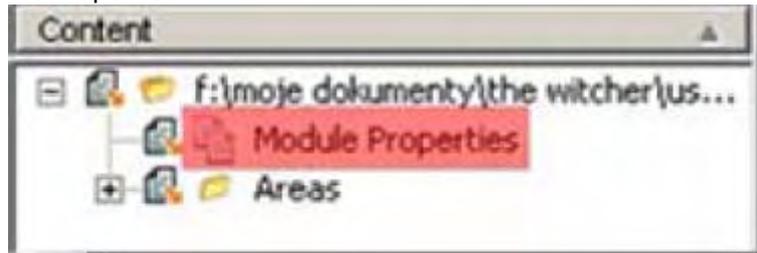
Außerdem erleichtert diese Zuweisung die Veröffentlichung unseres Abenteuers über das Internet. Dazu später mehr. Zuerst müssen wir unser Modul öffnen. wähle Open module... aus dem Modul Menü:



Das öffnet sich das Windows Fenster, zum laden von Dateien. Suche Dein Modul im Dokumente und Einstellungen\username\Eigene Dateien\The Witcher Ordner, Markiere es und klicke auf öffnen. Das Modul Explorer Fenster öffnet sich:



Doppelklicke auf Module Properties:



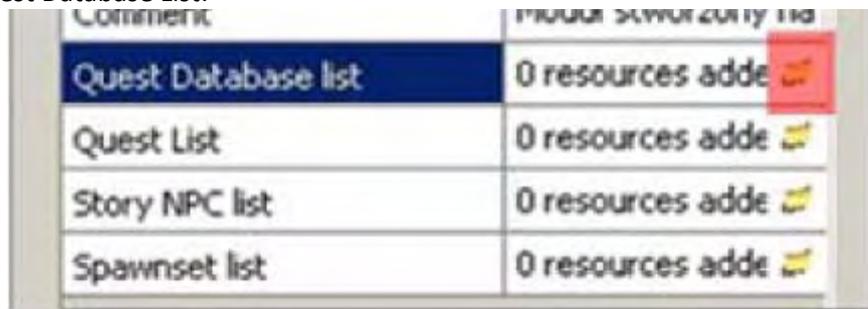
Das Modul Properties Fenster öffnet sich auf der rechten Seite des Bildschirms:



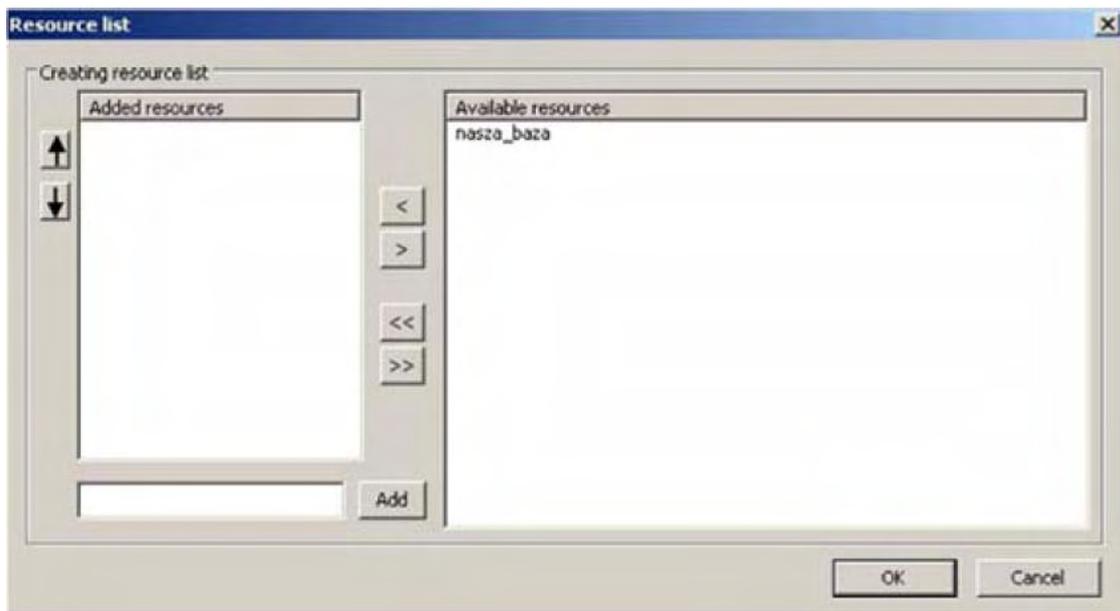
Module Einträge werden detailliert in einem anderen Kapitel dieses Handbuchs beschrieben. In diesem Augenblick interessieren wir uns für 4 Einträge aus der General-Section:



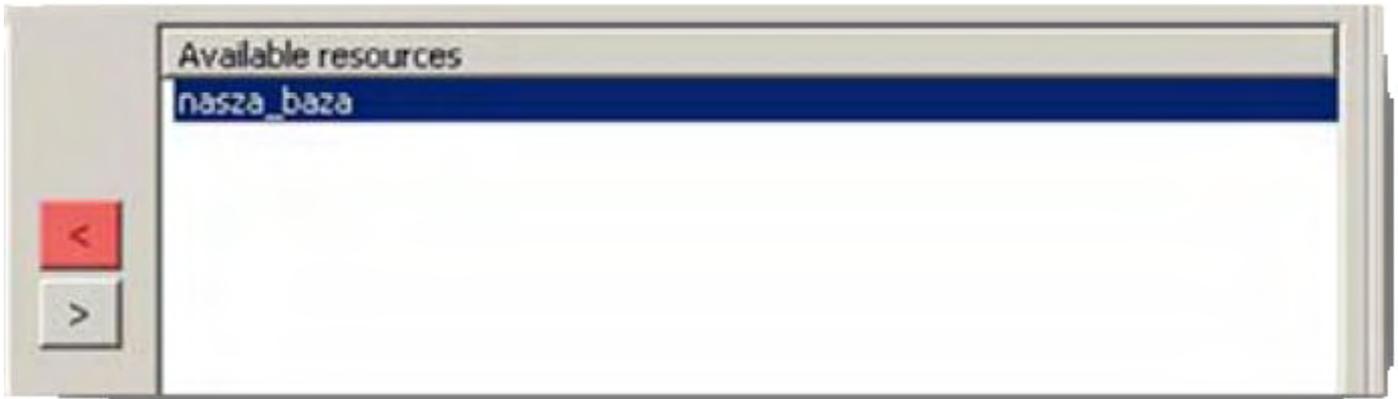
Der Eintrag der Quest Database list lässt uns Questdatenbanken dem Modul zuweisen, Unter Quest List werden einzelne Quests dem Modul zugewiesen. Die Story NPC List verwaltet die NPC Spawnsets (z.B. Siegfried's Spawnset) und die Spawnset List die aller übrigen Kreaturen (z.B. Jethro's Spawnset und das des Banditenanführers). Lass uns zuerst die Quest Datenbank zuweisen. Klicke dafür auf das gelbe Symbol rechts neben Quest Database List:



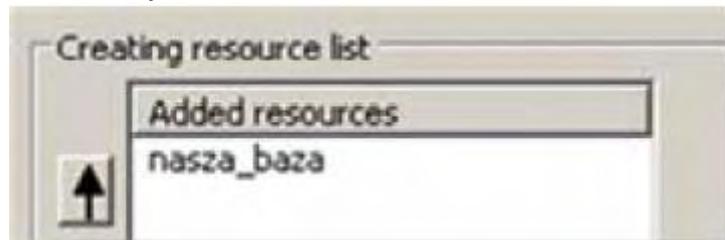
Das Ressource list Fenster öffnet sich:



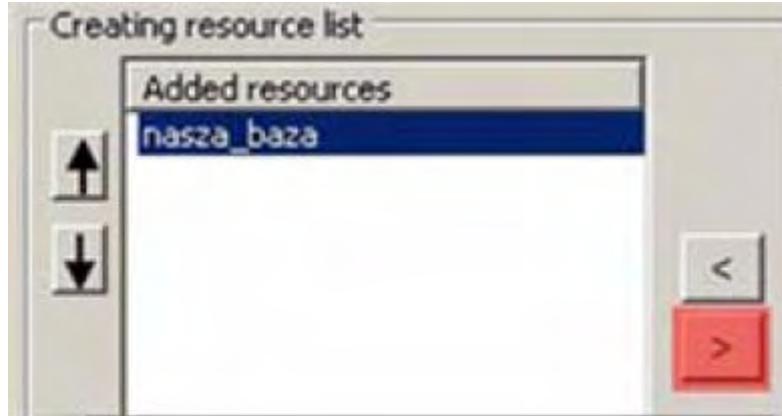
Wir sehen unsere Datenbank im Available resources Fenster. Um sie unserem Modul zuzuweisen, klicke zuerst auf den Datenbank Namen und dann auf "<":



Die our_base Datenbank erscheint jetzt im Added resources Fenster:



Das bedeutet, dass die Datenbank unserem Modul zugeteilt ist. Du kannst auch eine Datenbank hinzufügen, indem du auf einen Doppelklick auf den Datenbank Namen im Available Resources Fenster machst. Die Wirkung ist dieselbe – die Datenbank wird im Added Resources Fenster erscheinen. Wenn du diese Datenbank aus der Liste der Added Resources Fenster entfernen möchtest, Doppelklicke auf den Datenbank Namen oder auf „>“:



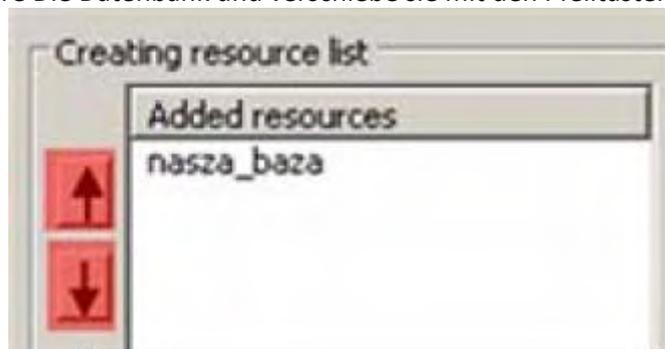
Dadurch wird die Datenbank aus dem Modul entfernen. Wenn du alle vorhandenen Quest-Datenbanken dem Modul zuteilen möchtest(im Falle, das es mehrere gibt), klicke auf '<<':



Wenn du gerne alle Datenbanken vom Modul entfernen möchtest, klicke auf '>>':



Wenn aus irgendeinem Grund, gerne die Reihenfolge der Datenbanken im Added Resources Fenster ändern möchtest. markiere Die Datenbank und verschiebe sie mit den Pfeiltasten:



Da wir jetzt fertig sind, klicke auf Ok in der Resources List. Wirf jetzt einen Blick auf den Eintrag hinter Quest Database List:

Quest Database list	1 resources adde 🟡
Quest List	0 resources adde 🟡
Story NPC list	0 resources adde 🟡
Spawnset list	0 resources adde 🟡

Die Liste zeigt jetzt eine zugefügte Ressource an.

Betrachten wir jetzt den Eintrag für Quest List Ressourcen. An dieser Stelle können Quests dem Modul zugefügt werden. In unserem Fall befindet sich unsere Quest aber schon in der Quest Datenbank. Wir brauchen hier also nichts zugefügen.

Jetzt werden wir Siegfried's Spawnset zugefügen. Klicke dafür auf das gelbe Symbol, rechts neben dem Story NPC list Eintrag:

Quest Database list	1 resources adde 🟡
Quest List	0 resources adde 🟡
Story NPC list	0 resources adde 🟡
Spawnset list	0 resources adde 🟡

Das Ressource List Fenster öffnet sich. Dieses Mal zeigt das Available resources Fenster die NPC-Spawnsets und keine Quest Datenbanken:



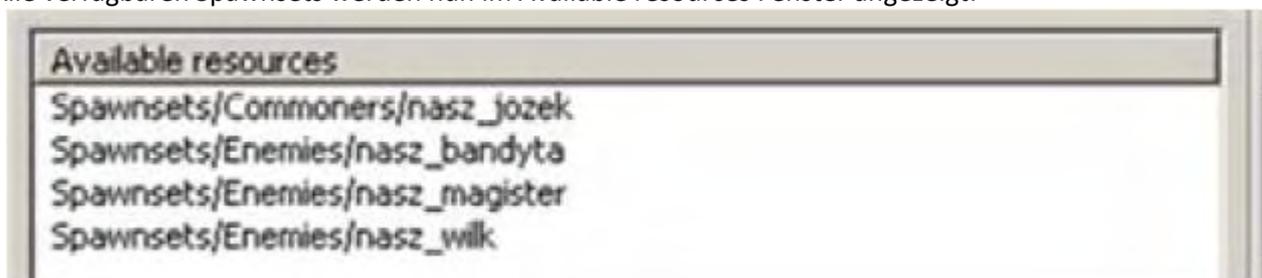
Klicke auf das NPC-Spawn/our_siegfried in Available resources Fenster und klicke dann auf "<" Wähle jetzt Ok, im Ressource List Fenster. Rechts neben dem Story NPC list Eintrag wird jetzt angezeigt, dass eine Ressource zugefügt wurde:

Quest Database list	1 resources added
Quest List	0 resources added
Story NPC list	1 resources added
Spawnset list	0 resources added

Jetzt müssen wir noch die Spawnsets der restlichen Kreaturen hinzufügen. Klicke dazu wieder auf das kleine gelbe Symbol in der Spalte rechts neben Spawnset List:

Quest Database list	1 resources added
Quest List	0 resources added
Story NPC list	1 resources added
Spawnset list	0 resources added

Alle verfügbaren Spawnsets werden nun im Available resources Fenster angezeigt:



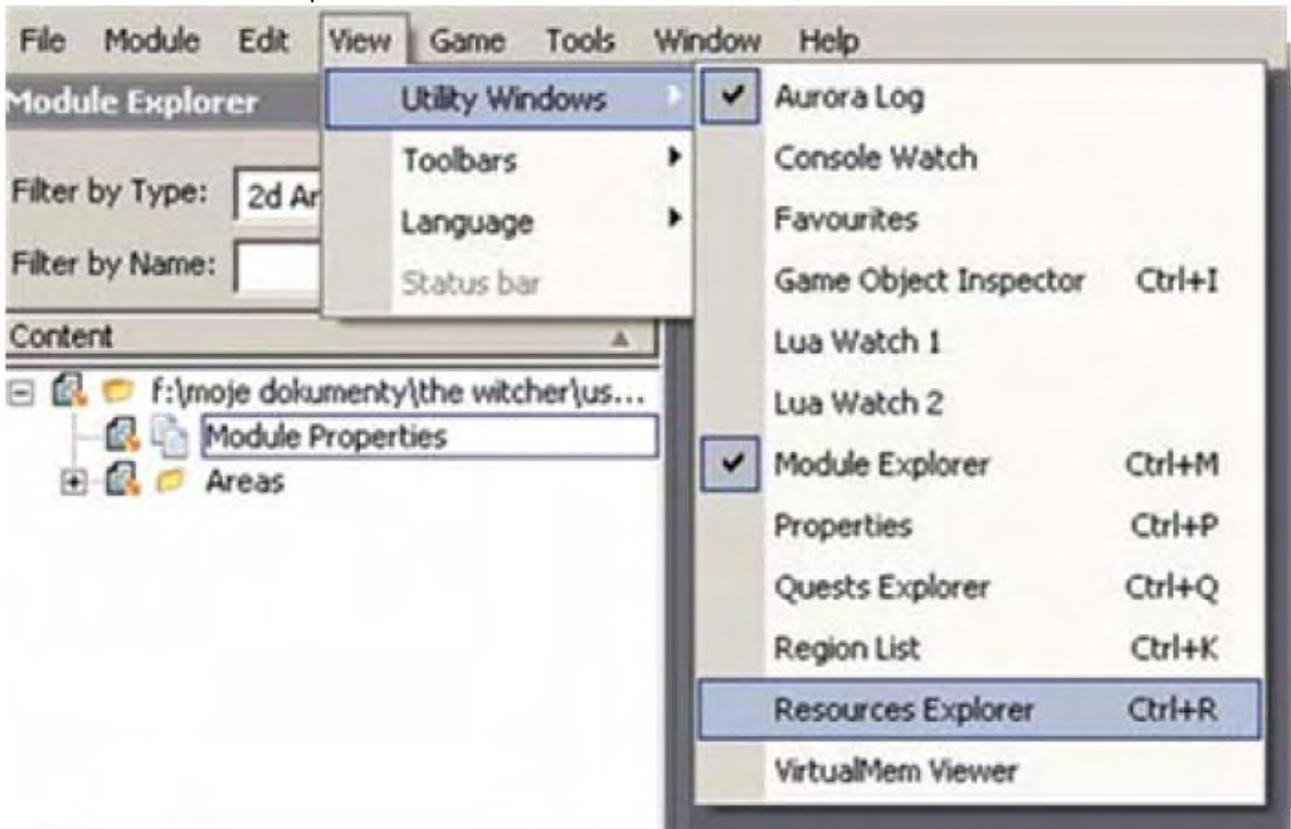
Du kannst den '<<' Knopf verwenden, wenn Du alle Spawnsets aus dem Available resources Fenster deinem Modul hinzufügen willst. Wenn du aber nur einzelne Spawnsets aus der Liste auswählen willst, markiere sie und klicke dann auf "<". Wenn Du eine große Anzahl Spawnsets markieren willst, kannst Du das mit SHIFT und CONTROL tun, genau wie Du es von anderen Windows-Applikationen gewohnt bist. Nach dem hinzufügen des Spawnsets, klicke auf OK im Resources List Fenster. Rechts neben Spawnset list, wird angezeigt dass wir 4 Ressourcen zugefügt haben:

Quest Database list	1 resources added
Quest List	0 resources added
Story NPC list	1 resources added
Spawnset list	4 resources added

Kommen wir jetzt zu dem Punkt, an dem wir dem Modul Dateien zuweisen.

Die Einführung erklärte, dass du Dateien zum Modul hinzufügen solltest. Warum? Weil sie, wenn du diese Dateien einem Modul zufügst, in eine einzige große .adv Datei zusammengefasst werden, die Dein komplettes Abenteuer enthält. Auf diese Weise musst du Dich nicht darüber sorgen, ob die Dateien in den richtigen Ordnern liegen, Wenn Du Dein Abenteuer an einen Freund senden willst, sende Ihm einfach die adv. Datei. Sie enthält alle relevanten Dateien

Um Dateien zum Modul hinzufügen, brauchst du den Resources Explorer. Um Ihn zu öffnen, wähle Utility Windows -> Resources Explorer vom View Menü aus:



Ein **Resources Explorer** Fenster wird auf der linken Bildschirmseite erscheinen:



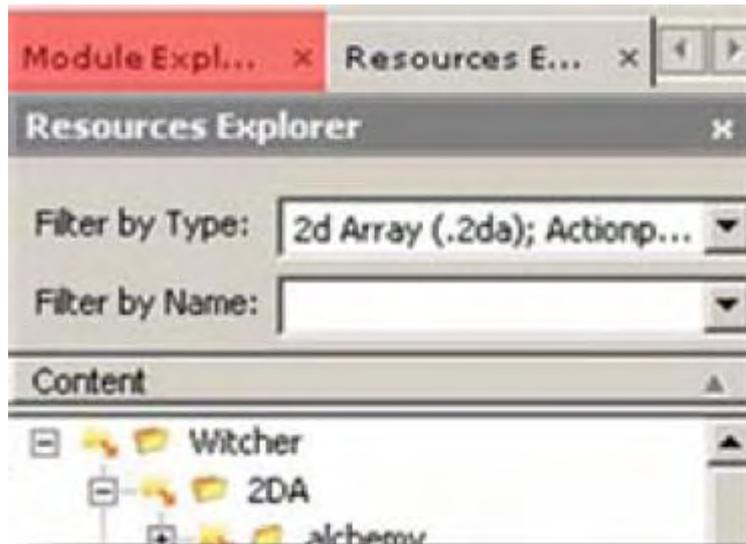
Dies ist eine Liste von Dateien und Verzeichnissen, die unserem Modul hinzugefügt werden müssen:

- \2DA\globaldlgflags.2da,
- \Dialogues\jethro_conversation.dlg,
- \Dialogues\siegfried_conversation.dlg,
- \Quests\our_task.qst,
- \Templates\Characters\Commoners\our_jethro.utc,
- \Templates\Characters\Enemies\our_bandit.utc,
- \Templates\Characters\Monsters\our_wolf.utc,
- \Templates\Characters\NPC\our_professor.utc,
- \Templates\Characters\NPC\our_siegfried.utc,
- \Templates\NPCSpawns\our_siegfried.npc,
- \Templates\Spawnsets\Commoners\our_jethro.spn,
- \Templates\Spawnsets\Enemies\our_bandit.spn,
- \Templates\Spawnsets\Enemies\our_professor.spn,
- \Templates\Spawnsets\Enemies\our_wolf.spn.

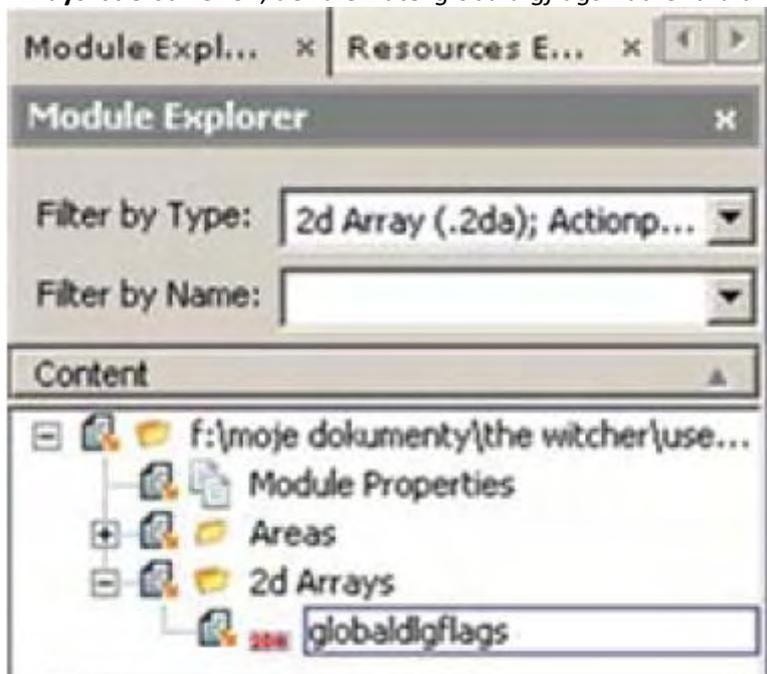
Alle diese Dateien, haben wir bisher erstellt. Um eine Datei einem Modul hinzuzufügen, öffne die Verzweigung des Ordners, der die Datei enthält. Dann Rechtsklick auf die Datei und wähle **Add to Module** aus den Menu:



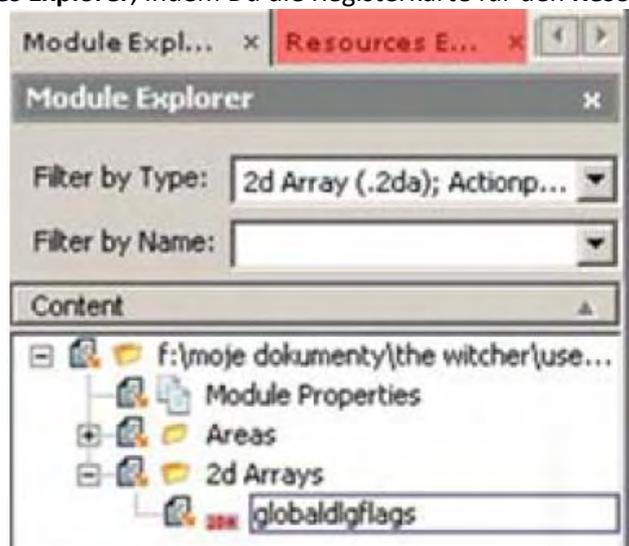
Wechsele jetzt zum **Module Explorer** Fenster, indem du auf Die Registerkarte für den **Module Explorer** klickst:



Ein neuer Ordner, **2d Arrays** ist erschienen, der die Datei *globaldigflags.2da* enthält:



Gehe wieder zum **Resources Explorer**, indem Du die Registerkarte für den **Resources Explorer** wählst:

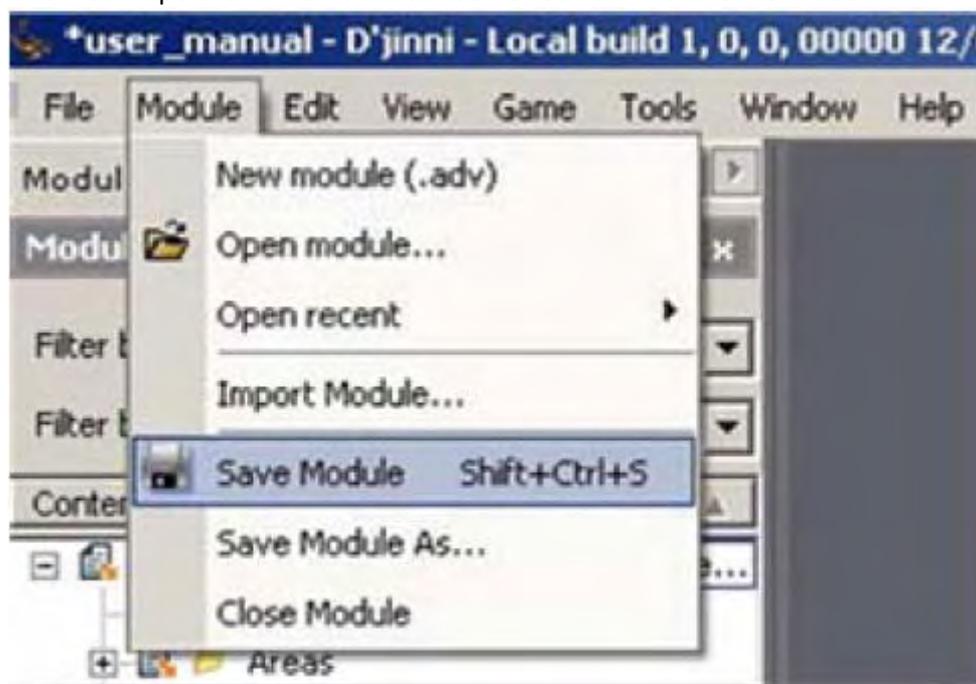


Füge jetzt die restlichen Dateien auf die gleiche Weise dem Modul hinzu. Nachdem man alle Dateien dem **Module Explorer** hinzugefügt hat, sollte die folgende Dateistruktur erscheinen:



Wenn Du jetzt Änderungen an einer Datei vornehmen möchtest, die Du bereits dem Modul zugefügt hast, musst Du dafür den **Module Explorer** benutzen. Dies ist sehr wichtig. Wenn Du versuchst eine Datei im **Resources Explorer** zu modifizieren, wird die Änderung nicht im Modul vorhanden sein. Behalte das im Hinterkopf.

Jetzt lass uns das Modul speichern. Wähle **Save Module** aus dem **Module** Menü aus.



Fertig. Wir haben jetzt alle selbst erstellten Dateien dem Modul zugeordnet. Jetzt, ist es Zeit für die nächste Stufe...

STARTEN EINES MODULES

Es ist jetzt an der Zeit, das Modul zu starten, um zu sehen, wie es funktioniert. Eine der besten Funktionen vom D'jinni Editor ist die Fähigkeit, das Spiel innerhalb des Editors zu starten. Ich muss Dir nicht sagen, wie viel Zeit Du dadurch sparst, besonders wenn Du das Spiel oft starten musst. Außerdem enthält der D'jinni Editor viele hilfreiche Tools, die nützlich für das Testen des Spiels sind. Du wirst schon sehen.

Lass uns das Modul starten. Wähle dafür **Play** aus den **Game** Menü:



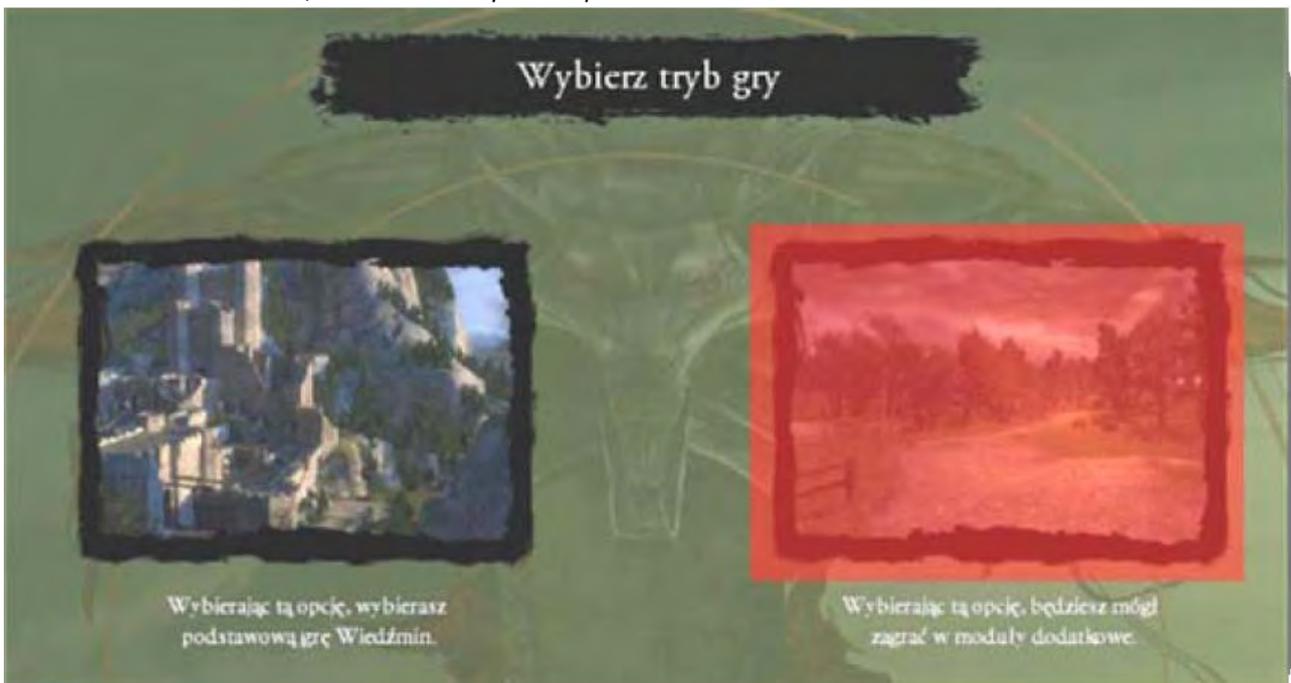
Du kannst auch F5 drücken. Wenn Du das Modul lieber im Fullscreen Modus, als im Editor Fenster starten willst, wähle **Fullscreen** im **Game** Menü. Mache das aber, bevor Du das Spiel startest:



Allerdings ist es nicht erforderlich, das Spiel im Fullscreen Modus zu starten. Du kannst auch zwischen Editor Fenster und Fullscreen umschalten, indem Du ALT+ENTER drückst, nachdem das Spiel bereits gestartet ist. Wenn das Modul startet, wird der normale Titelschirm von The Witcher erscheinen:



Um das Modul zu starten, wähle *Neues Spiel* -> *Spielmodus wählen*:



Klicke auf den Namen des Moduls („Unser neues Abenteuer“) aus der Liste verfügbarer Module. Klick dann auf Abenteuer laden:



Wähle jetzt den Schwierigkeitsgrad aus:

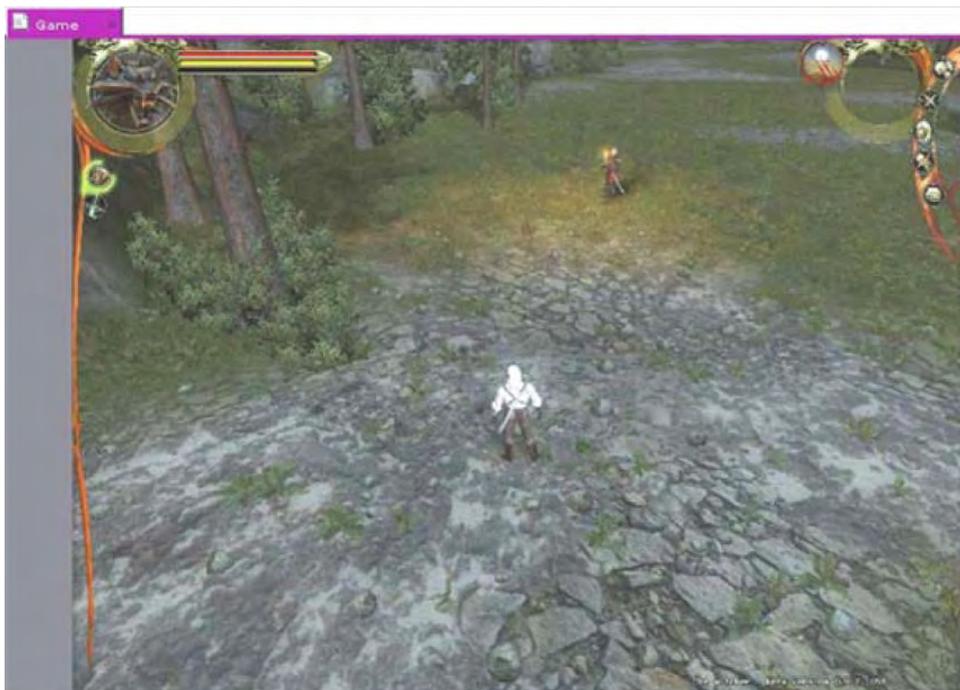


Wähle nun die Steuerungsmethode aus. Wenn man aus dem Editor heraus arbeitet, ist die Maus kontrollierte Steuerungsmethode am besten geeignet:



Das erlaubt Dir, aus dem Editor Menu ebenso gut zu agieren, wie aus dem Spiel selbst, was sehr hilfreich ist. Während des Spiels kannst du zwischen den verschiedenen Steuerungsmethoden durch die Benutzung der F1, F2 und F3 Tasten wählen.

Jetzt widmen wir uns dem neuen Modul und... Geralt natürlich:



Was übrig bleibt, ist, unser Modul zu testen.

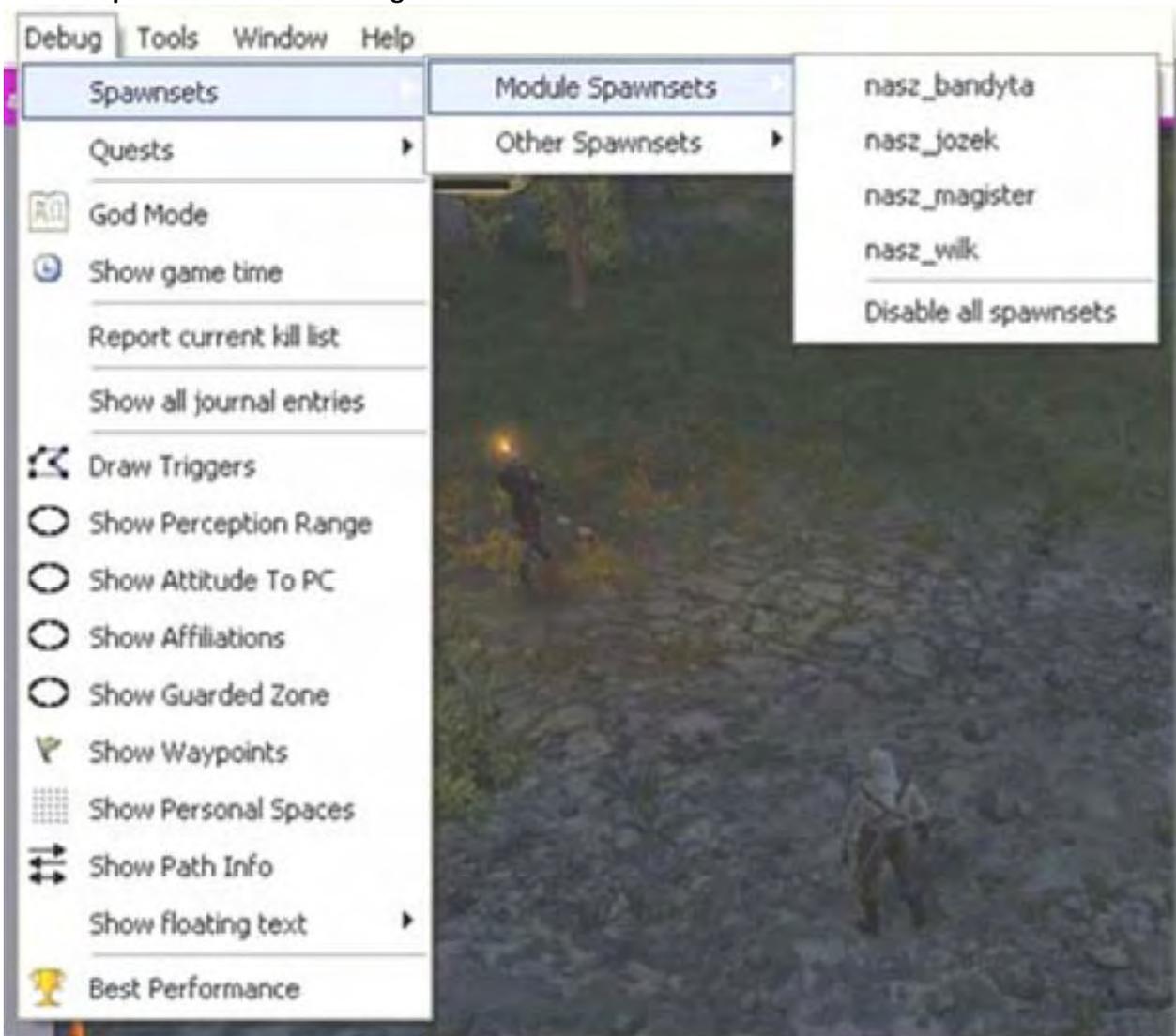
TESTEN EINES MODULES

Wir haben unserem Modul Dateien hinzugefügt und zugewiesen. Jetzt müssen wir es testen. Warum wir es testen sollten? Zu aller erst um zu sehen ob es richtig funktioniert. Die Hauptaufgabe ist es, Fehler zu finden, die in unserem Abenteuer auftreten könnten.

Zunächst schlage ich vor, dass du Geralt ein wenig im Modul bewegst. Laufe zu der Stelle an der sich Jethro, der Banditenanführer und die Wölfe befinden. Aber was ist das? Wo sind denn alle? Es gibt nur Jethro? Warum? Haben wir was falsch gemacht?

Die Antwort ist Nein! Alles ist gut. Spawnsets sind im Normalfall immer deaktiviert. Das trifft aber nicht auf NPC Spawnsets zu. Darum ist Siegfried sichtbar, nachdem das Modul gestartet ist. Unsere nächste Aufgabe wird es sein, dass richtige Spawnset zu aktivieren, so dass ein Spieler, der unser Modul startet, alles wie geplant sehen wird. Du kannst Spawnsets auf 2 Wege aktivieren und deaktivieren. Entweder, durch die Benutzung von Scripts (nicht im Handbuch beschrieben), oder durch eine spezielle Quest, die das Set aktiviert.

Davon später mehr. Zum Testen der Spawnsets im Editor kann man diese noch auf eine andere Art aktivieren. Öffne das **Debug** Menü, das erschienen ist, nachdem du das Modul gestartet hast. Wähle **Spawnsets** -> **Module Spawnsets** aus dem **Debug** Menü:



Das „Module Spawnsets“-Menü beinhaltet alle Spawn-Sets, die wir unserem Modul zugewiesen haben. Vergiss nicht, dass es ein großer Unterschied ist, ob eine Datei einem Modul zugewiesen wird, oder ob sie dem Modul hinzugefügt wird. Starten wir zuerst Jethro's Spawn-Set. Um dies zu machen, musst du „our_jethro“ aus dem „Debug -> Spawnsets -> Module Spawnsets“-Menü wählen. Jethro sollte jetzt an seinem Spawnpoint erscheinen:



Schau noch einmal ins „Debug -> Spawnsets -> Module Spawnsets“-Menü:



Wie du siehst, erscheint nun ein Häkchen neben „our_jethro“. Das bedeutet, dass dieses Spawn-Set aktiviert wurde. Wenn du „our_jethro“ aus diesem Menü erneut anklickst, wird Jethro aus dem Modul verschwinden, genauso wie das Häkchen im Menü. Alle Kommandos im „Debug“-Menü sind so genannte „Toggles“ (Schalter). Das bedeutet, sie können an- und ausgeschaltet werden. Logischerweise kannst du alle Kommandos aus dem „Debug“-Menü aktivieren.

Wenn du „Disable all spawnsets“ auswählst:



Dann werden alle kürzlich aktivierten Spawn-Sets wieder deaktiviert.

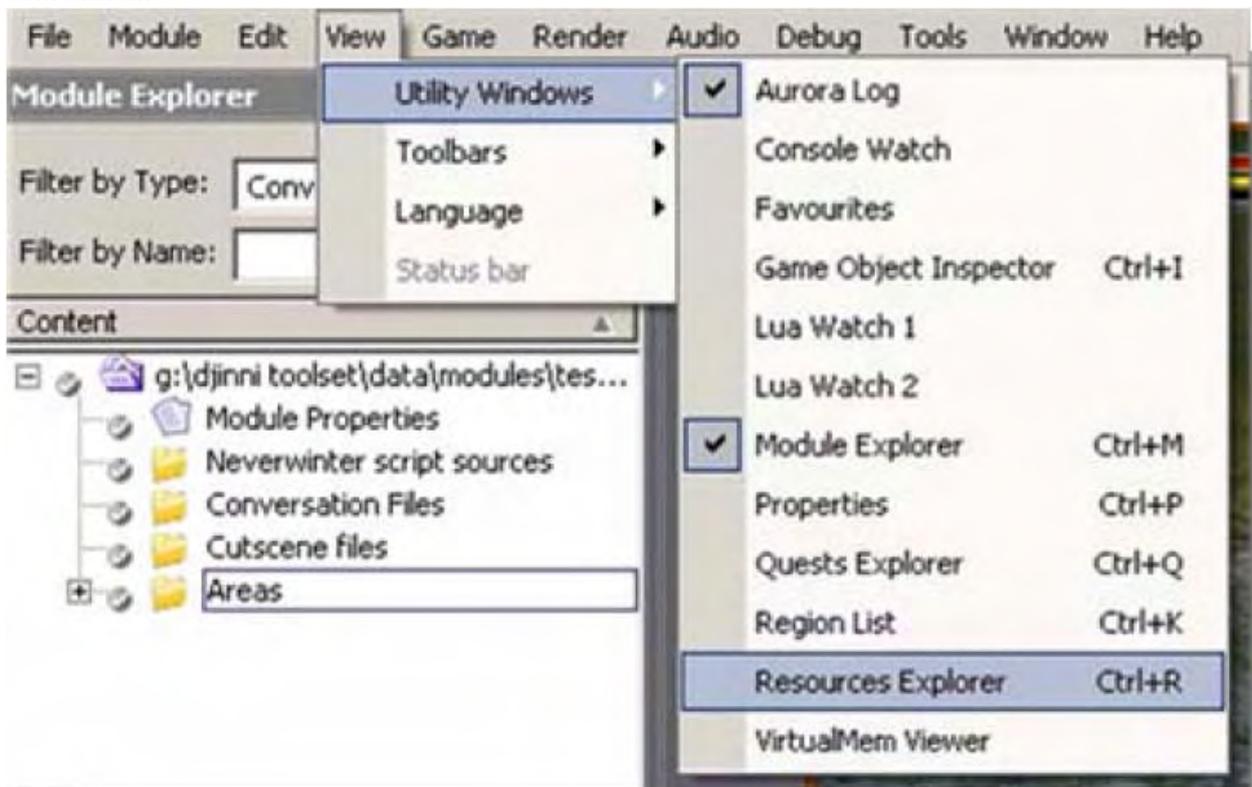
Ok, aktivieren wir nun alle Spawn-Sets, indem wir sie unter „Debug -> Spawnsets -> Module Spawnsets“ nacheinander anklicken:



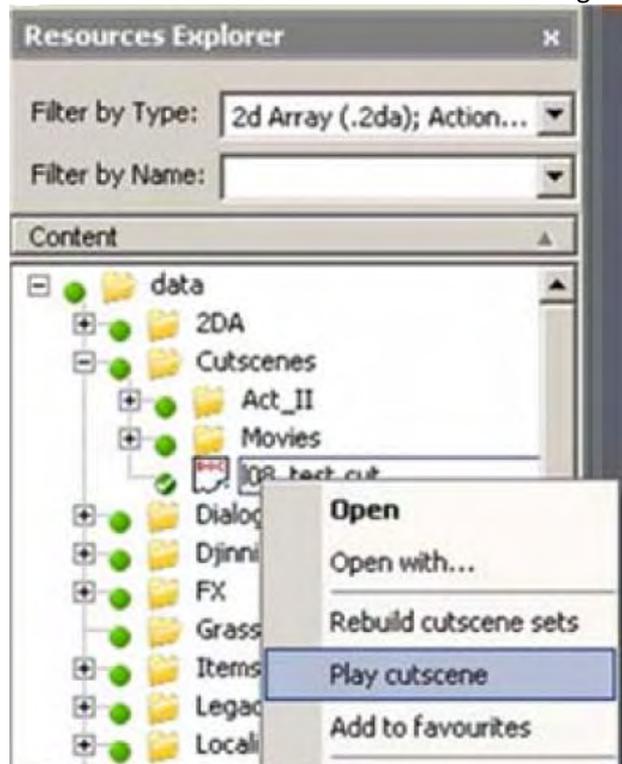
Jetzt sollten alle unsere Charaktere im aktivierten Modul erscheinen. Da du jetzt Spawn-Sets über Quests (per „Enable Spawnset“ bzw. „Disable Spawnset“ aus der „OnPhaseFinished Actions“ und „OnQuestFinished Actions“ Sektion) aktivieren und deaktivieren kannst, ist es an der Zeit unsere Quest zu verbessern.

Und wie? Der Banditenanführer und dessen Wächter werden im Modul nur erscheinen, wenn Geralt (der Spieler) von Siegfried die Quest zum Töten des Banditenanführers erhält. Wieso? Dies hilft uns Komplikationen mit der Quest zu vermeiden – was würde zum Beispiel passieren, wenn Geralt zuerst den Banditenanführer tötet und anschließend erst mit Siegfried redet? Wir müssten einen anderen Conversations-Punkt erstellen, um dies zu berücksichtigen. Allerdings werden wir solche Probleme vermeiden, da wir bereits früher die Quest verändert und ihr eine zweite Stufe hinzugefügt haben.

Wir wissen jetzt, wie wir Spawn-Sets aktivieren und deaktivieren während wir das Modul testen. Wir können noch viele andere Dinge testen wenn das Modul läuft. Weitere Kommandos aus dem „Debug“-Menü werden in Kürze erklärt. Zuerst schauen wir uns jedoch das Ressourcen-Explorer Fenster an. Dieses Fenster hat einige interessante Befehle, die wir während des Testdurchlaufs ausführen können. Sollte das Ressourcen-Fenster deaktiviert sein, kannst du es über „Utility Windows -> Resources Explorer“ aus dem „View“-Menü öffnen:

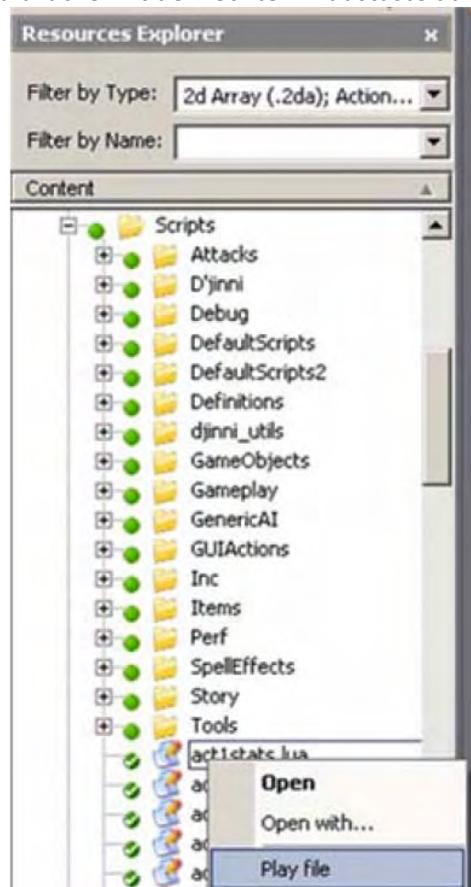


Der Ressourcen-Explorer wird auf der linken Seite des Bildschirms erscheinen. Klappe „Cutscenes“ aus, indem du auf das kleine + Zeichen daneben klickst. Rechte Maustaste auf irgendeine „.cut“-Datei:

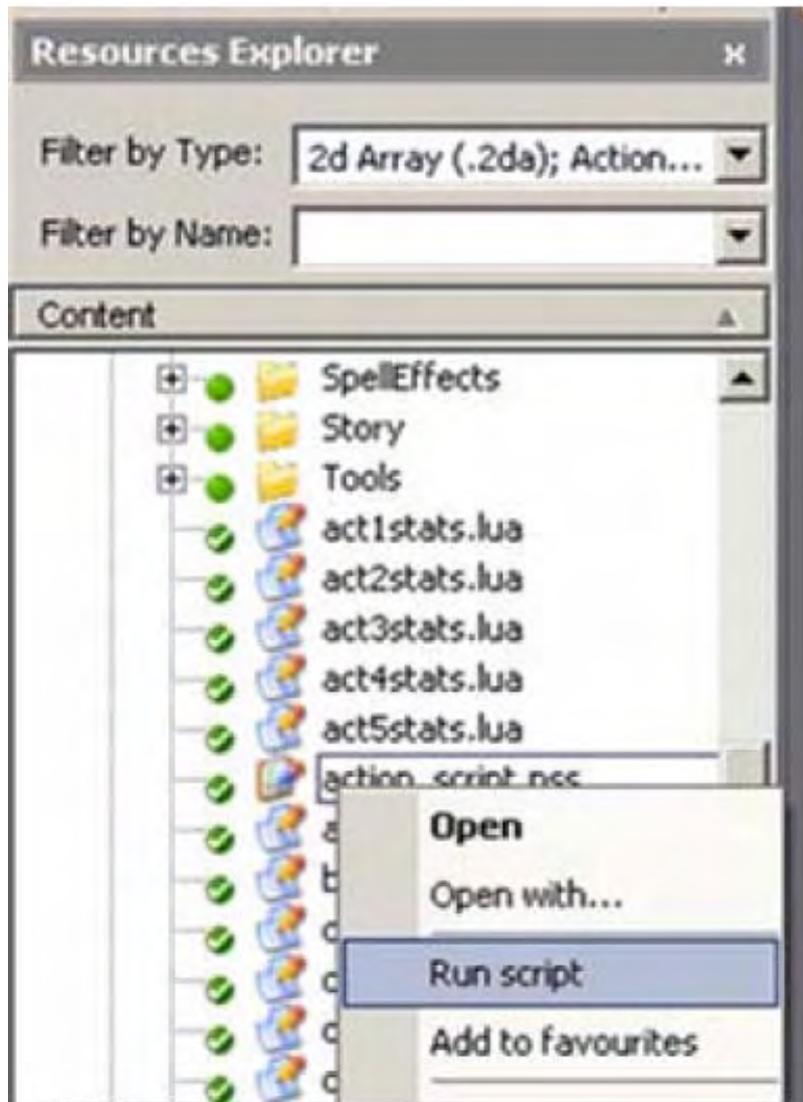


Ein neuer „Play Cutscene“ Befehl ist im Untermenü verfügbar. Dieser Befehl ist nur verfügbar, wenn das Modul läuft und es dir erlaubt einen Film zu öffnen. Das Erstellen von solchen Cutscenes wird in einem anderen Kapitel dieses Handbuchs beschrieben.

Klappe den „Script“-Baum aus und drücke mit der rechten Maustaste auf irgendeine „.lua“-Datei:

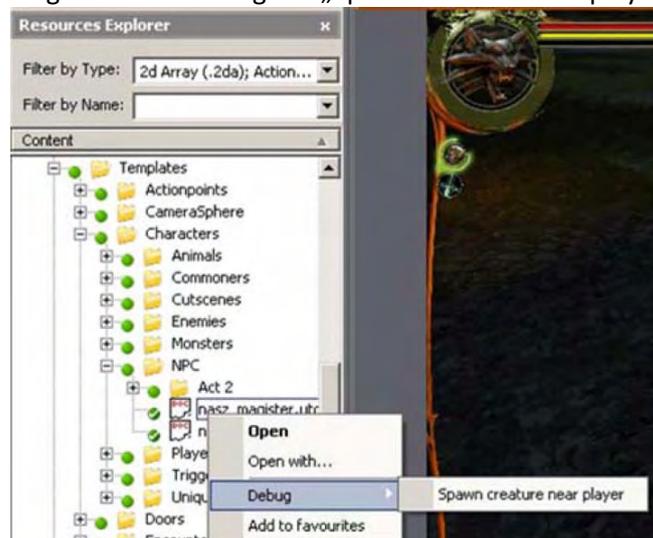


Auch hier ist ein neuer Befehl verfügbar. Es ist „Play file“, welches die Scripte aktiviert, die in Lua geschrieben wurden. Wenn du auf eine „.nss“-Datei rechtsklickst wirst du auch dort einen neuen Befehl vorfinden:



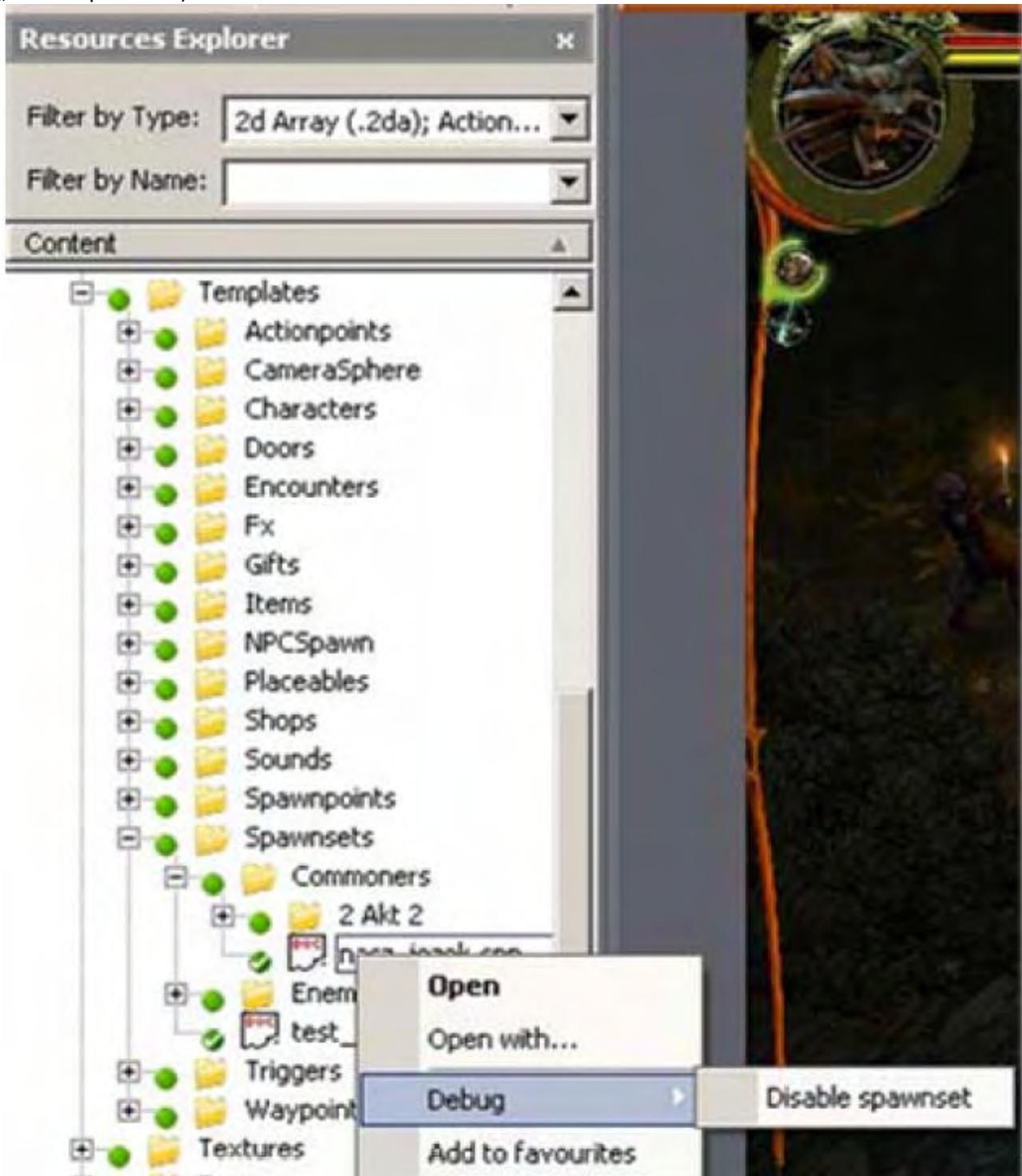
„Run Script“ erlaubt es dir Scripte auszuführen, die in „Neverwinter Nights Script“ geschrieben wurden. Auch das Skripten ist in einem anderen Kapitel dieses Handbuchs näher beschrieben.

Klappe nun den „Templates -> Characters“-Baum auf, und wähle z.B. „NPC“. Klicke nun mit rechts auf eine „.utc“-Datei. Ein neuer „Debug“-Befehl ist verfügbar: „Spawn creature near player“:



Dieser Befehl lässt einen bestimmten Charakter direkt neben Geralt erscheinen (nochmal zu Erinnerung: „.utc“-Dateien sind Charakter-Vorlagen“). Ein sehr nützliches Kommando.

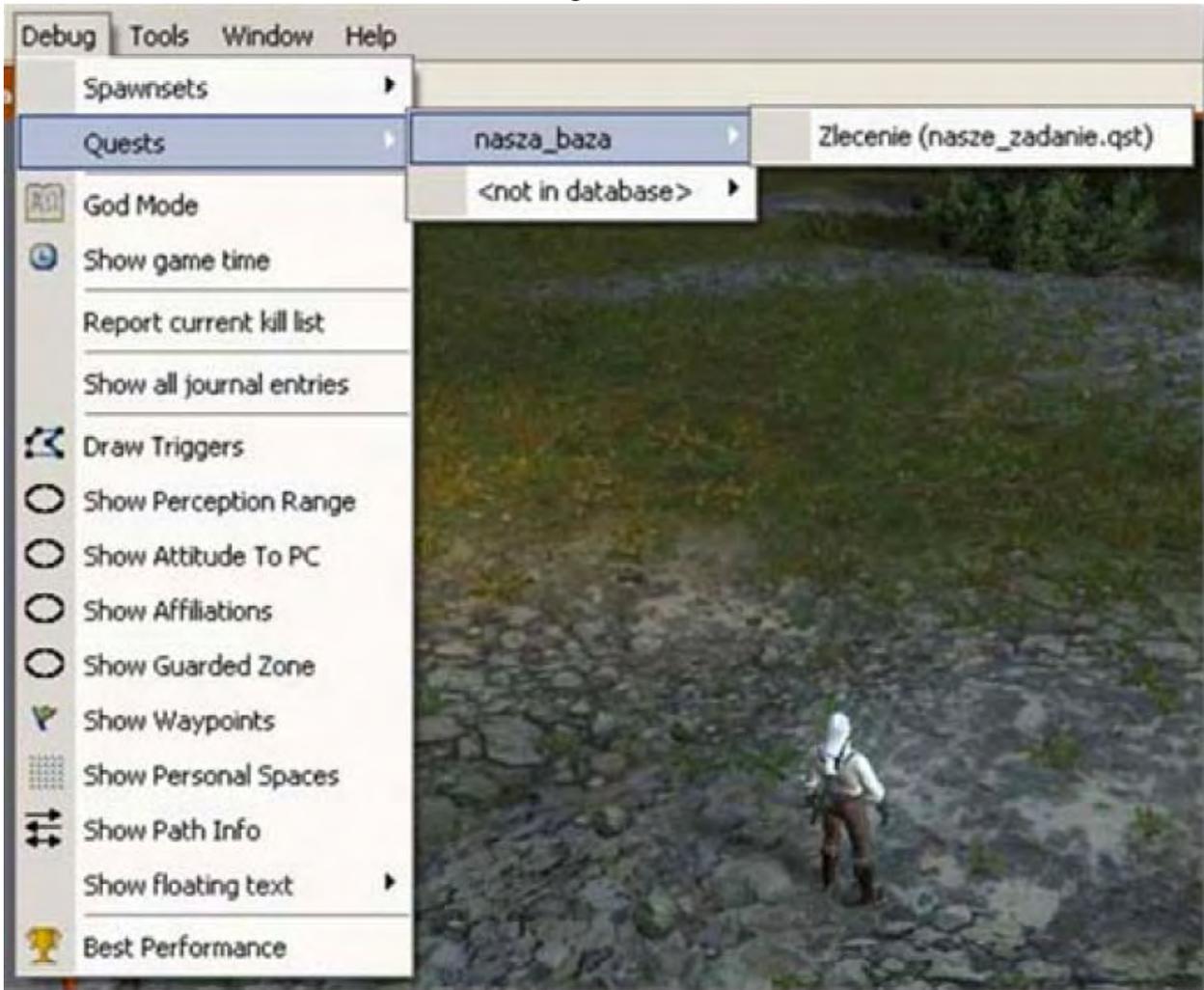
Klappen wir nun den „Templates -> Spawnsets -> Commoners“-Baum aus und klicken auf eine „.spn“-Datei mit der rechten Maustaste. Hier gibt es ebenfalls einen neuen Befehl: „Debug -> Disable Spawnset“ (bzw. „Enable Spawnset“):



Dieser Befehl hat die gleiche Funktion, wie der aus dem „Debug“-Menü. Er aktiviert bzw. deaktiviert Spawnsets. Vergiss nicht, „.spn“-Dateien sind die mit den Spawn-Sets.

Das ist alles über den Ressourcen-Explorer. Behalte immer vor Augen, dass diese Befehle nur verfügbar sind, wenn das Modul gerade läuft.

Kommen wir zurück zum Erklären anderer „Debug“-Menüs. Wähle „Quests“ aus diesem Menü:



Hier wirst du unsere „our_base“ Datenbank sehen, zusammen mit der „Task (our_task.qst)“ Aufgabe. Mit Hilfe des „Quests“-Befehls können wir unsere Quests im „Debug“-Menü testen. Wähle jetzt „Task (our_task.qst)“. Ein neues Fenster mit dem Titel „Running quest: EMPTY[Read Only]“ wird sich öffnen:



Dieses Fenster enthält die Quest, die wir früher erstellt haben. Diese Quest ist im Testdurchlauf schreibgeschützt (bestätigt wird dies durch die „[Read Only]“-Information, die im Namen des Fensters angezeigt wird). Das bedeutet, dass wir sie nicht verändern können.

Merke dir, dass der Quest-Beginn einen gelben Rahmen hat. Das heißt, die Quest wird in der Anfangsphase getestet und ist noch nicht wirklich aktiv.

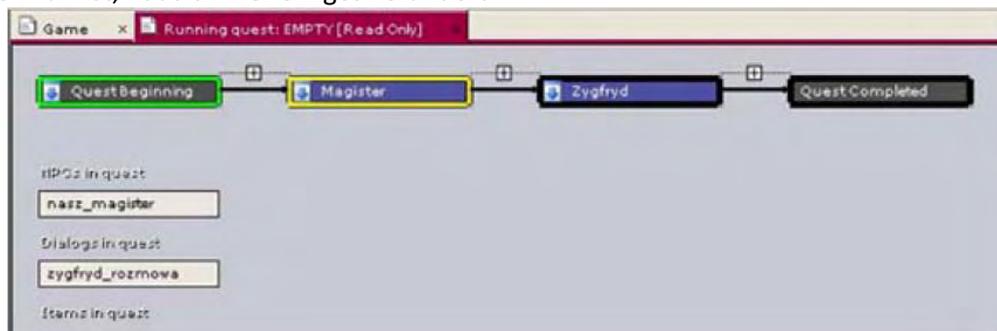
Kehren wir zum Modul zurück indem wir auf den „Game“-Reiter im oberen Bereiches des Fensters klicken:



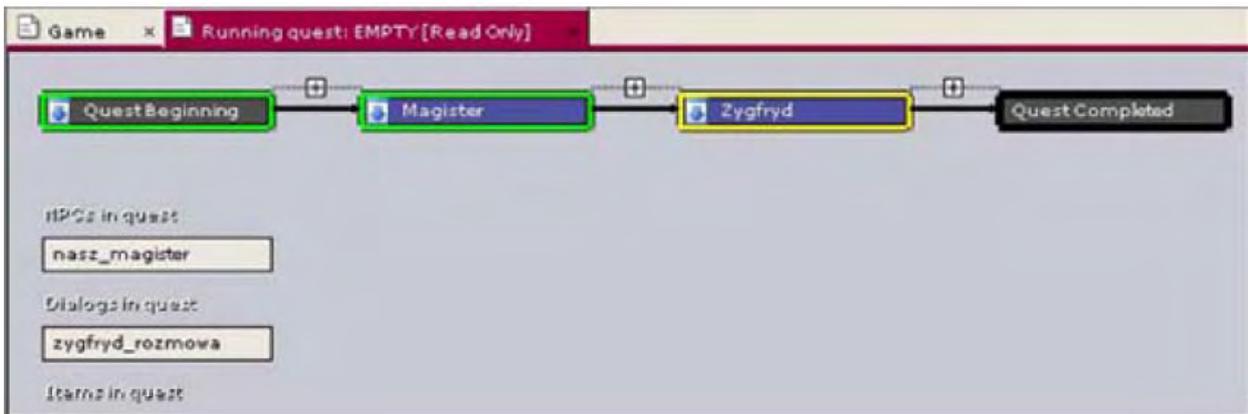
Suche jetzt Siegfried und rede mit ihm. Nimm die Quest von ihm an, indem du den entsprechenden Gesprächs-Punkt auswählst. Nachdem du Siegfrieds Quest angenommen hast, kehrst du zum Quest-Fenster zurück, indem du auf den Reiter mit dem Titel „Running Quest: EMPTY[Read Only]“ klickst:



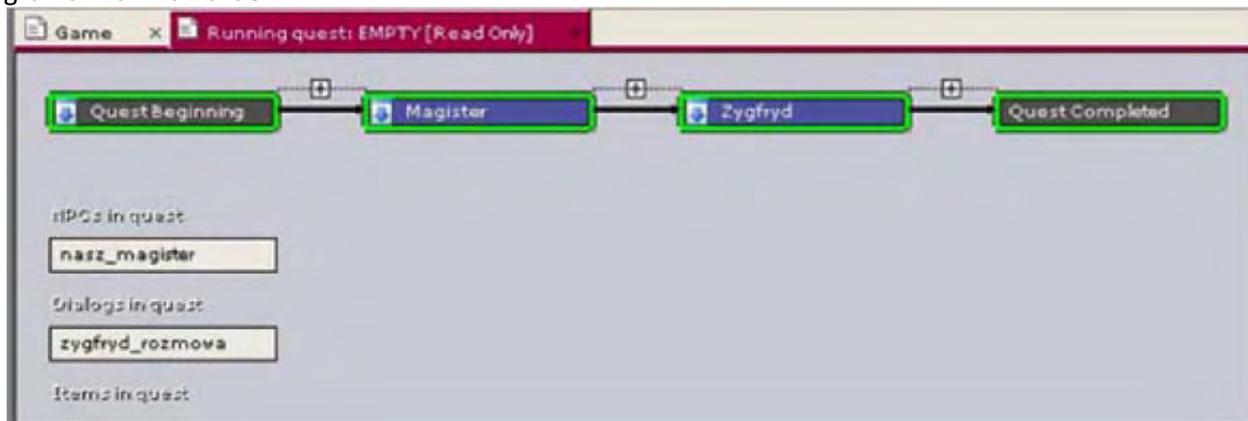
Wie du sehen kannst, hat sich hier einiges verändert:



Der Quest-Beginn ist jetzt grün umrahmt. Das bedeutet, dass diese Quest-Phase abgeschlossen wurde. Die „Professor“-Phase ist nun gelb umrahmt, was wiederum bedeutet, dass dieser Questabschnitt gerade getestet wird. Wenn du zum Modul zurückgehst, den Banditenanführer tötest und danach wieder zum Quest-Fenster wechselst wirst du sehen, dass die „Professor“-Phase abgeschlossen wurde (grüner Rahmen) und der Test die „Siegfried“-Phase erreicht hat (gelber Rahmen):



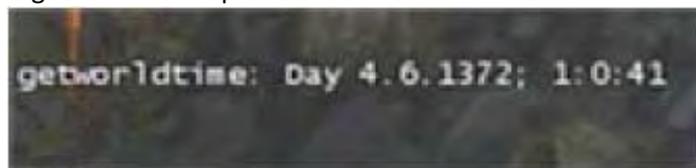
Wenn du mit Siegfried redest und ihm sagst, dass du den Banditenanführer bereits getötet hast wirst du im Quest-Fenster sehen, dass die Quest abgeschlossen wurde und die „Quest Completed“-Phase ebenfalls grün umrahmt wurde.



Das zeigt, dass sowohl die Konversation, als auch die Quest, ohne Probleme funktionieren. Später erfahren wir noch ein paar kleine Veränderungen im Zusammenhang mit dem Erscheinen des Banditenanführers. Die restlichen Befehle aus dem „Debug“-Menü sind:

-> **God Mode** -> Geralt wird unsterblich. Dies ist sehr hilfreich, um Kämpfe zu testen. Sollte Geralt nämlich sterben, muss der Test wiederholt werden.

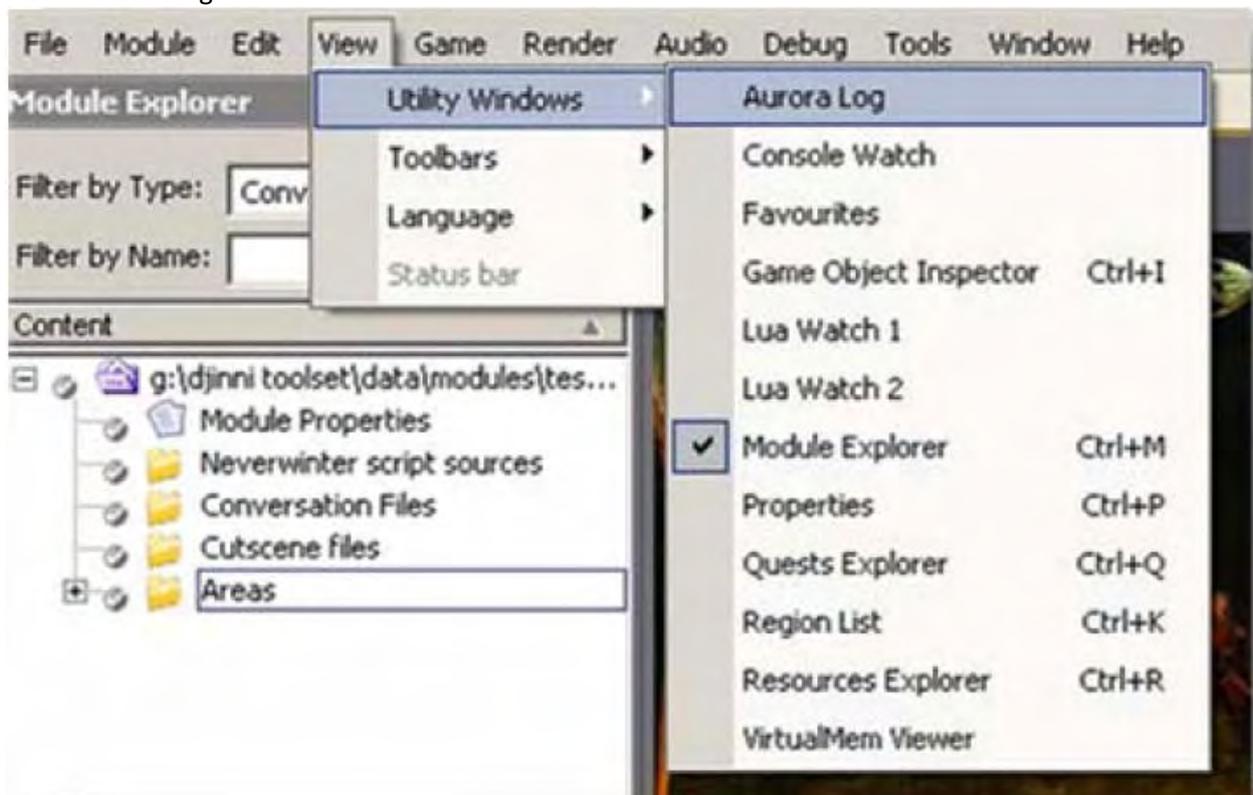
-> **Show Game Time** -> Zeigt die aktuelle Spielzeit in der oberen linken Ecke an:



-> **Report current kill list** -> Zeigt im unteren Bereich des Bildschirms eine Liste mit momentan toten Charakteren aus dem „Aurora“-Log an:



Der Charaktername sowie die Anzahl von Charakteren mit diesem Namen, welche getötet wurden, werden angezeigt. Wenn das „Aurora log“-Fenster nicht angezeigt wird, klicke im „View“-Menü auf „Utility Windows -> Aurora log“:



-> **Show all journal entries**. Zeigt alle Journaleinträge im „Aurora log“-Fenster an:



-> **Draw Triggers**. Zeigt die Trigger (Auslöser) im aktuellen Module an.

-> **Show Perception Range.** Zeigt die Sichtweite jedes erschienenen Charakters an:



Mehr über die Reichweiten eines Charakters wurden bereits in einem früheren Kapitel erzählt.

-> **Show Attitude to PC.** Zeigt die Gesinnung eines NPC gegenüber Geralt an. Es wird mit Hilfe eines farbigen Kreises um den Charakter angezeigt:



Grüne Kreise bedeuten, dass der Charakter Geralt gegenüber freundlich eingestellt ist, rot bedeutet eine feindliche Gesinnung, blau signalisiert Gleichgültigkeit.

-> **Show Affiliations.** Zeigt die Zugehörigkeit eines Charakters an. Diese wird mit einem Kreis über dem Kopf dargestellt:



Charaktere, die zu einer Fraktion gehören, haben die gleichen Farbkreise über ihren Köpfen.

-> **Show Guarded Zone.** Zeigt die Zonen an, die von einem bestimmten Charakter bewacht werden. In unserem Abenteuer spielt niemand diese Rolle.

-> **Show Waypoint.** Zeigt alle Wegpunkte an. Auch Wegpunkte haben wir in unserem Modul nicht.

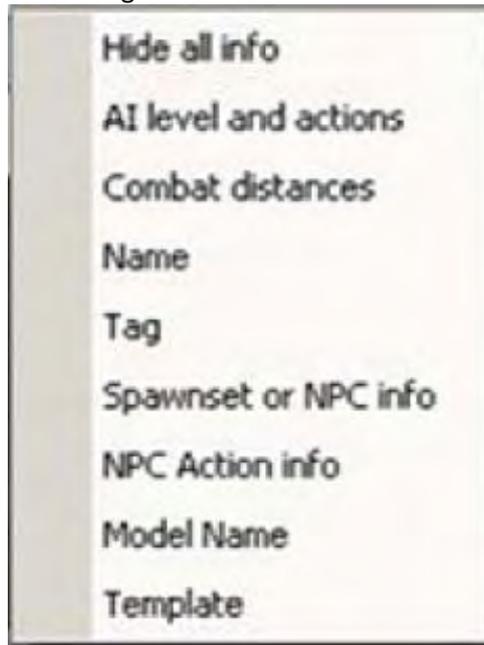
-> **Show Personal Spaces.** Zeigt den Platz an, den ein bestimmter Charakter im Module einnimmt. Dieser wird mit einem pinkfarbenen Kreis unter dem Charakter angezeigt:



-> **Show Path Info.** Zeigt den Weg zu einem bestimmten Actionspoint an, zu welchem der Charakter sich bewegt:



-> **Show floating text.** Zeigt Testinformationen über den Charakter. Der Text wird über dem Kopf des Charakters erscheinen. Wähle die Art der Information, welche anschließend angezeigt werden soll, aus dem Menü, das nach drücken von „Show floating text“ erscheint:



Diese Befehle sind wie Schalter. Du kannst jeden davon aktivieren bzw. deaktivieren, oder auch alle gleichzeitig anzeigen.

-> **Hide all info.** Versteckt alle Informationen.

-> **AI level and actions.** Zeigt Informationen zum Level des Charakters, sowie dessen Aktionen:



→ **Combat distances.** Zeigt die Entfernung von Charakteren während des Kampfes:



→ **Name.** Zeigt den Namen des Charakters an:



→ **Tag.** Zeigt die Bezeichnung eines Charakters an:



- **Spawnset or NPC Info.** Zeigt den Name des Spawnsets sowie die aktuellen Spiel-Phase eines Charakters an:



- **NPC Action Info.** Zeigt die momentan aktive Aktion an, die der Charakter ausführt:



→ **Model name.** Zeigt den Namen des Modells vom Charakter an:



→ **Template.** Zeigt die Vorlage des Charakters an:

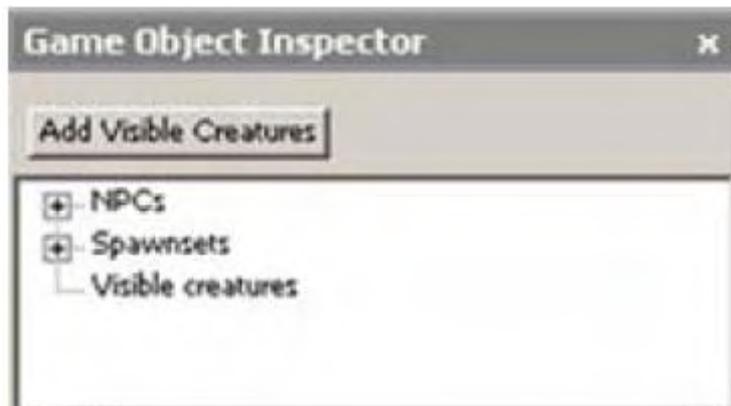


Der letzte Befehl im „Debug“-Menü ist „Best Performance“. Wenn dieser ausgewählt wurde, werden Grafikeffekte im Module auf das notwendige Minimum reduziert um die beste Effizienz während des Testens zu erreichen. Dies kann sehr nützlich sein.

Das war alles über das „Debug“-Menü. Schauen wir uns nun das letzte Werkzeug zum Testen von Modulen an, nämlich den „Game Object Inspector“. Um das entsprechende Fenster anzuzeigen wählst du vom „View“-Menü den Punkt „Utility Windows -> Game Object Inspector“ aus:



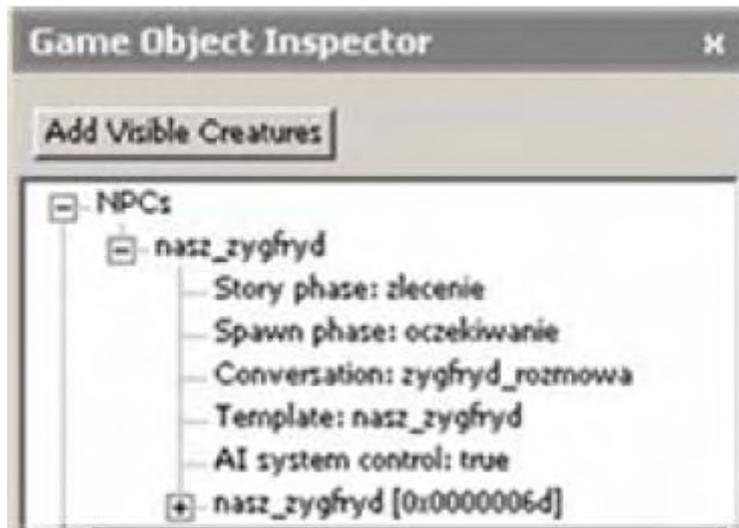
Das „Game Object Inspector“-Fenster wird auf der linken Seite des Bildschirms erscheinen:



Das Fenster zeigt alle notwendigen Informationen über das, was im laufenden Modul gerade passiert. Die Informationen werden in Form eines Baumes dargestellt. Klappe die „NPC“ und „Spawnsets“-Bäume aus, indem du auf das kleine „+“ daneben klickst:



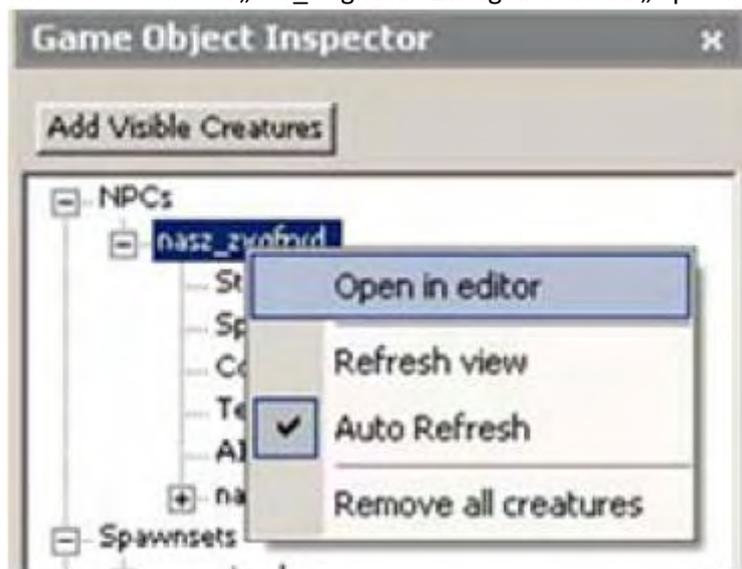
Der „NPC“-Zweig zeigt uns Informationen über alle NPCs des Moduls an. Bei uns gibt es nur einen NPC, nämlich Siegfried. Wir haben sein Spawn-Set unter „our_siegfried.npc“ gespeichert. Klappe seinen Zweig jetzt noch weiter aus:



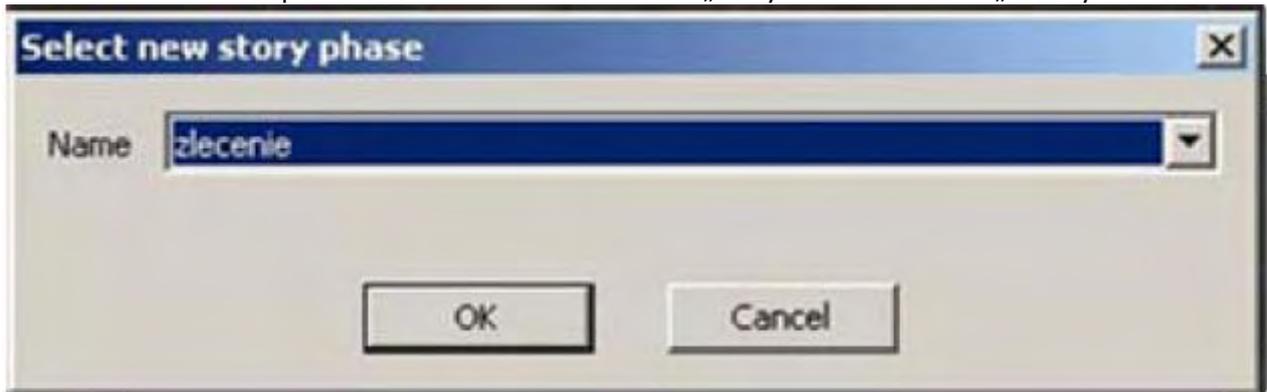
Eine Auswahl von Informationen über Siegfried werde nun angezeigt:

- „Story Phase“ -> Spiel-Phase, in der sich der NPC befindet
- „Spawn Phase“ -> Die aktuelle Spawn-Phase des NPCs
- „Conversation“ -> Die Unterhaltung, die dem NPC momentan zugewiesen ist
- „Template“ -> Name der Vorlage für den Charakter
- „AI System Control“ -> Ob ein bestimmter Charakter vom Computer (true) oder vom Spieler (false) gesteuert wird.

Klicke mit der rechten Maustaste auf den „our_siegfried“-Zweig und wähle „Open in editor“:

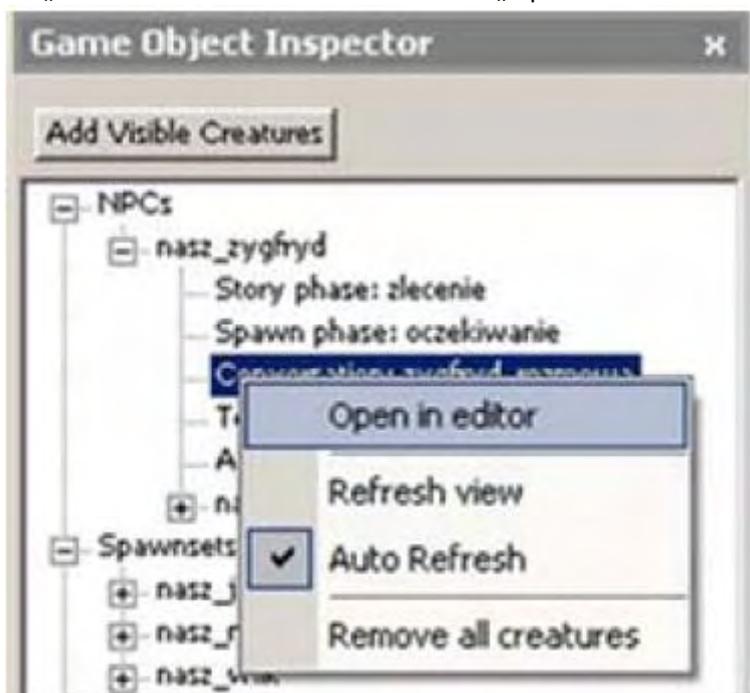


Dies öffnet Siegfrieds Spawn-Set. Du kannst auch auf andere Zweige rechtsklicken um weitere Befehle abzurufen. Klicke zum Beispiel mit der rechten Maustaste auf „Story Phase“ und dann „Modify value“:



Nun öffnet sich ein neues Fenster mit dem Titel „Select new story phase“, welches benutzt werden kann, um die Spiel-Phase des Charakters zu ändern. In unserem Fall ist dies nicht möglich, da Siegfried nur eine Spiel-Phase besitzt.

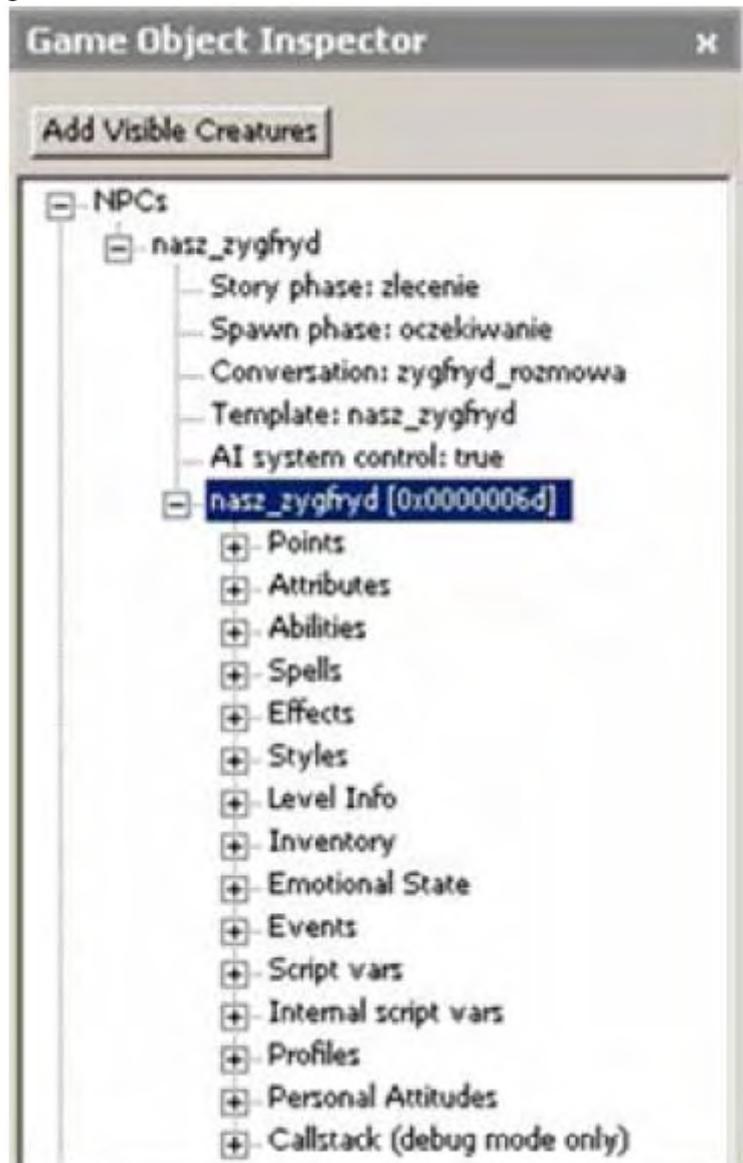
Klicke nun mit rechts auf „Conversation“ und anschließend auf „Open in editor“:



Dies öffnet eine Conversations-Datei.

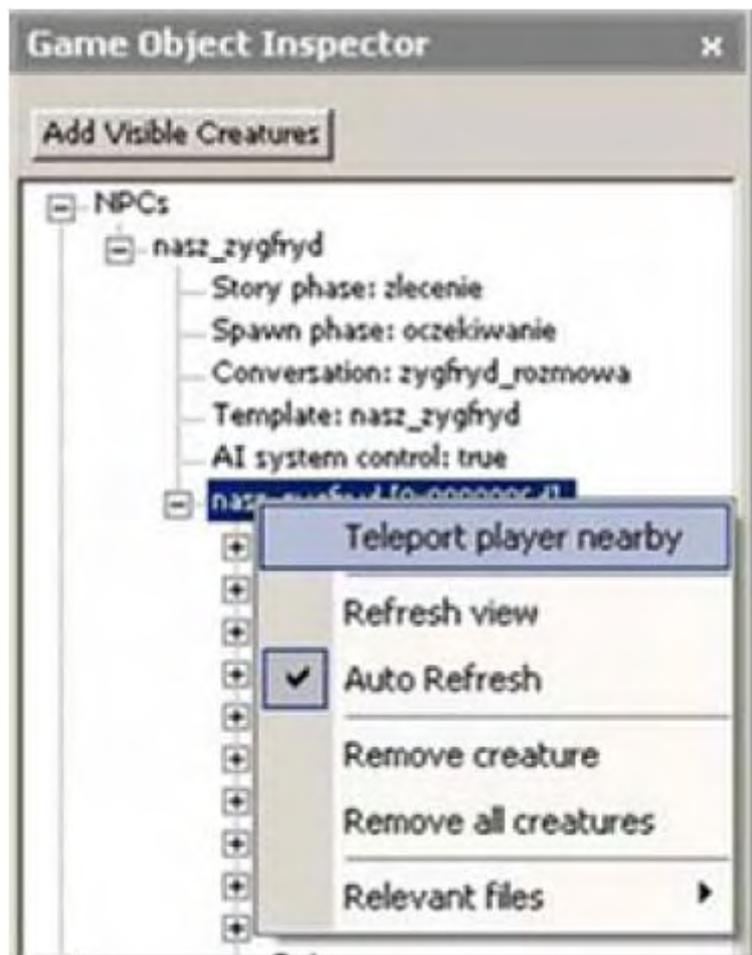
Der Befehl „Refresh View“ aktualisiert die Anzeige im „Game Object Inspector“-Fenster. „Remove all creatures“ entfernt alle Charaktere aus dem „Game Object Inspector“-Fenster, was bedeutet, dass diese nicht weiter untersucht werden.

Klappen wir nun den nächsten Teil unseres „our_siegfried“-Zweiges aus. Es werden nun Informationen über den Charakter angezeigt:

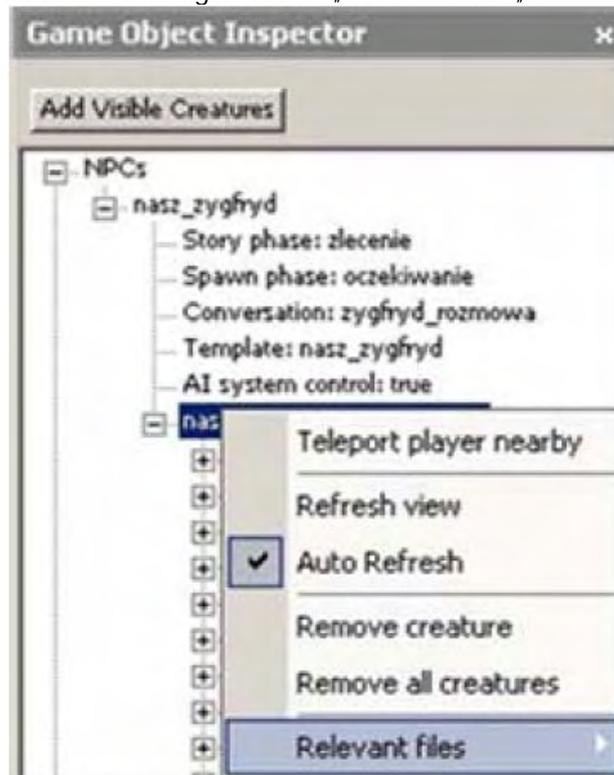


Viele Zweige werden nun sichtbar. Sie beinhalten Informationen über nahezu jedes Thema, von Leben und Energiepunkten bis zu gelernten Fertigkeiten und Profilen. Um Details zu erhalten erweitere einfach den entsprechenden Zweig.

Ein Rechtsklick auf einen Unter-Zweig ermöglicht weitere Befehle wie „Modify value“ (Wert verändern), usw. Klicke nun mir rechts auf den Zweig, der die Siegfried-Bezeichnung hat und wähle „Teleport player near“ aus:



Damit bewegst du Geralt in die Nähe von Siegfried. Dies ist sehr nützlich, da du dir damit Zeit sparen kannst, die du sonst zum aufsuchen benötigst. Wähle „Relevant files „vom Menü:



So werden alle Dateien angezeigt, die mit diesem Charakter verbunden sind:

```
Story: nasz_zygfryd.npc
Template: nasz_zygfryd.utc
Conversation: zygfryd_rozmowa.dlg
Script ScriptHeartbeat: def_onheartbeat.nss
Script ScriptOnNotice: def_onperception.nss
Script ScriptSpellAt: def_onspellcast.nss
Script ScriptAttacked: def_onattacked.nss
Script ScriptDamaged: def_ondamaged.nss
Script ScriptDisturbed: def_ondisturbed.nss
Script ScriptEndRound: def_onecr.nss
Script ScriptDialogue: def_convers.nss
Script ScriptSpawn: def_onspawnin.nss
Script ScriptRested: def_onrested.nss
Script ScriptDeath: def_ondeath.nss
Script ScriptOnBlocked: def_onblocked.nss
Script ScriptAtkBegin: def_onatkbegin.nss
Script ScriptOnWitness: def_onwitness.nss
Script ScriptEndConvers: def_endconv.nss
Script ScriptInternal: def_oninternal.nss
```

Jede Datei davon kann mit einem Klick darauf geöffnet werden.

Gehen wir zurück zum Spawnset-Zweig. Erweitere den „our_jethro“-Zweig (welcher den Dateinamen von Jethro's Spawn-Set anzeigt):

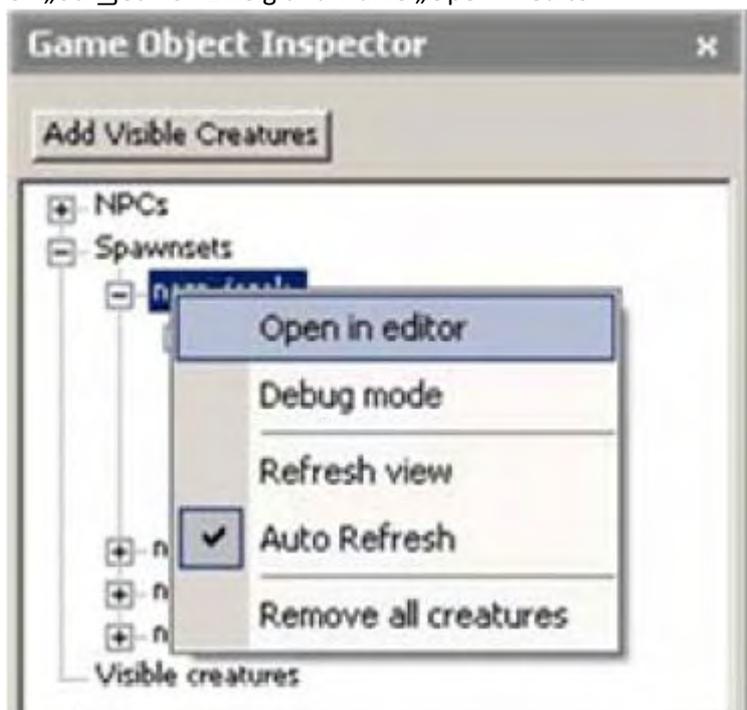


Hier gibt es einen weiteren Zweig, der sich „poker“ nennt. Zur Erinnerung: „poker“ ist der Name eines Spawn-Eintrages in Jethro’s Spawn-Set. Erweitere den „poker“-Zweig. Er enthält weitere Informationen über Jethro:



- Slots -> Anzahl der Charaktere die gespawnt werden
- Spawned -> Anzahl der bereits gespawnten Charaktere
- Time to next spawn -> Zeit, bis der nächste gespawnt wird
- Killed -> Anzahl der Charaktere, die getötet wurden

Rechtsklicke nun auf den „our_jethro“-Zweig und wähle „Open in editor“:

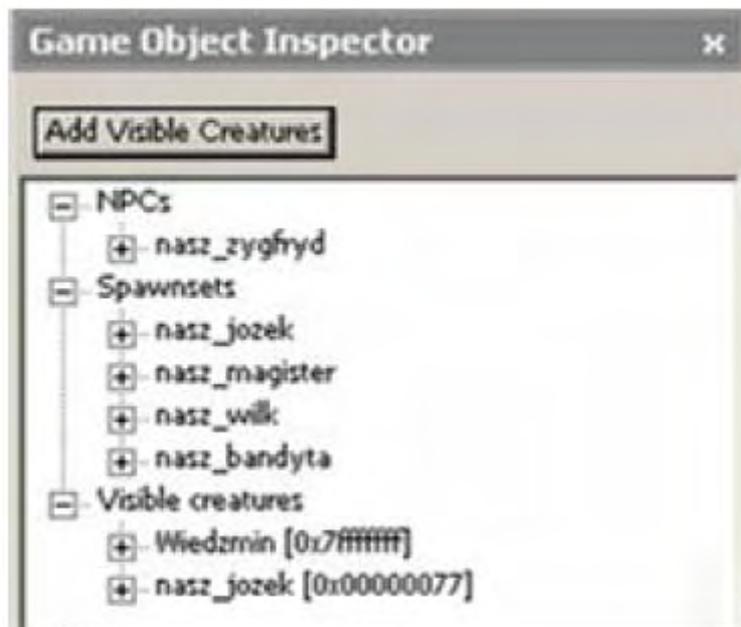


Damit öffnest du Jethro’s Spawn-Set. Ein Rechtsklick auf den „poker“-Zweig und die Auswahl „Open in editor“ lässt Jethro’s Vorlage öffnen.

Platziere Geralt nun so, dass er einen Charakter sieht (aber keinen NPC). Dies könnte Jethro sein. Klicke anschließend auf „Add Visible Creatures“ im oberen Teil des „Game Object Inspector“-Fensters:



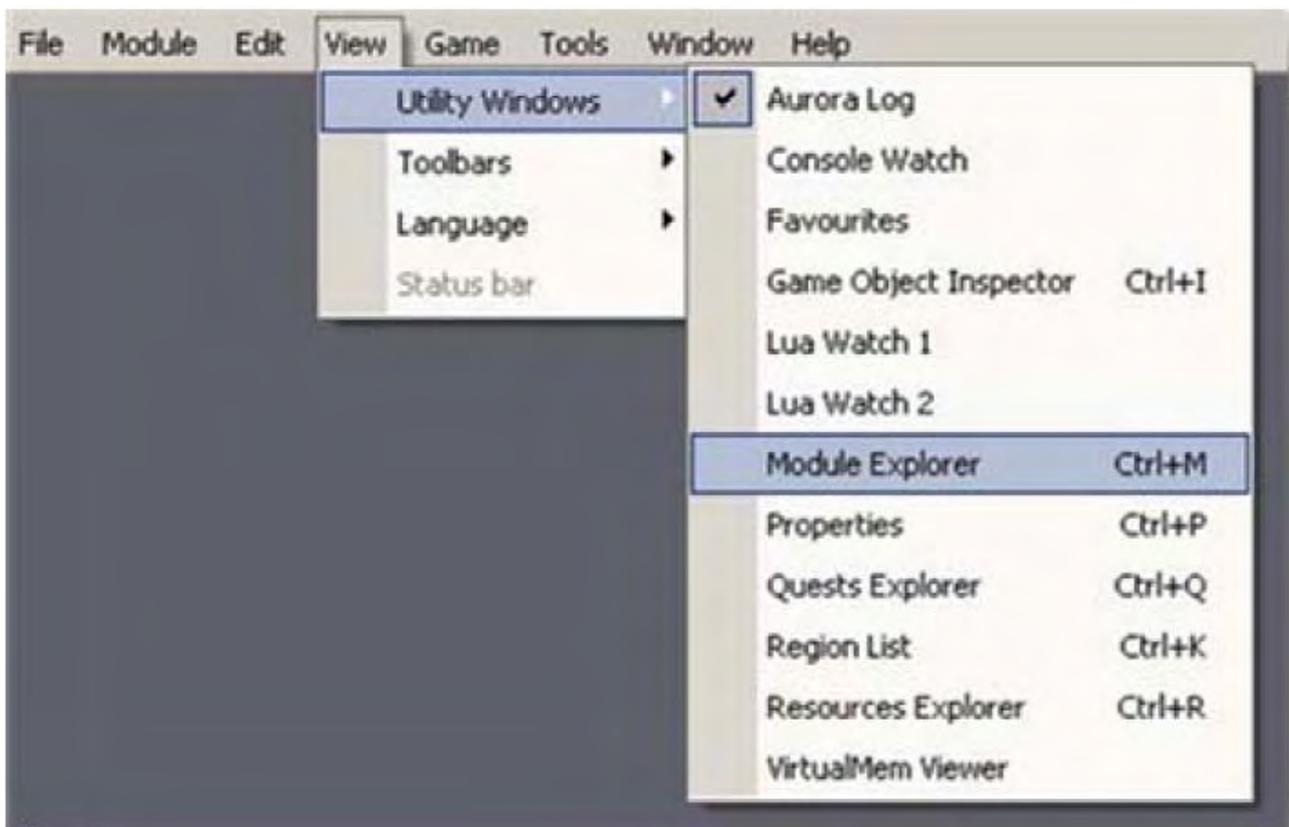
Dieser Befehl fügt neue Unterzweige im Bereich von „Visible creatures“ zu, welche die Charaktere auflistet, welche momentan auf dem Bildschirm zu sehen sind:



Damit bekommst du Zugriff auf dieselben Daten wie im Falle von NPCs.

Das ist nahezu Alles, was du zum Testen eines Moduls wissen musst. Trotzdem gibt es noch ein paar kleine Tricks. Wenn du mit der Maus über einem Charakter bist und die „Y“-Taste drückst, wird dieser getötet. „SHIFT + Y“ zusammen tötet alle in der des Mausursors befindlichen Charaktere. Mit der „+“-Taste (Zahlenfeld) kannst du Geralt an die Stelle des Mausursors teleportieren.

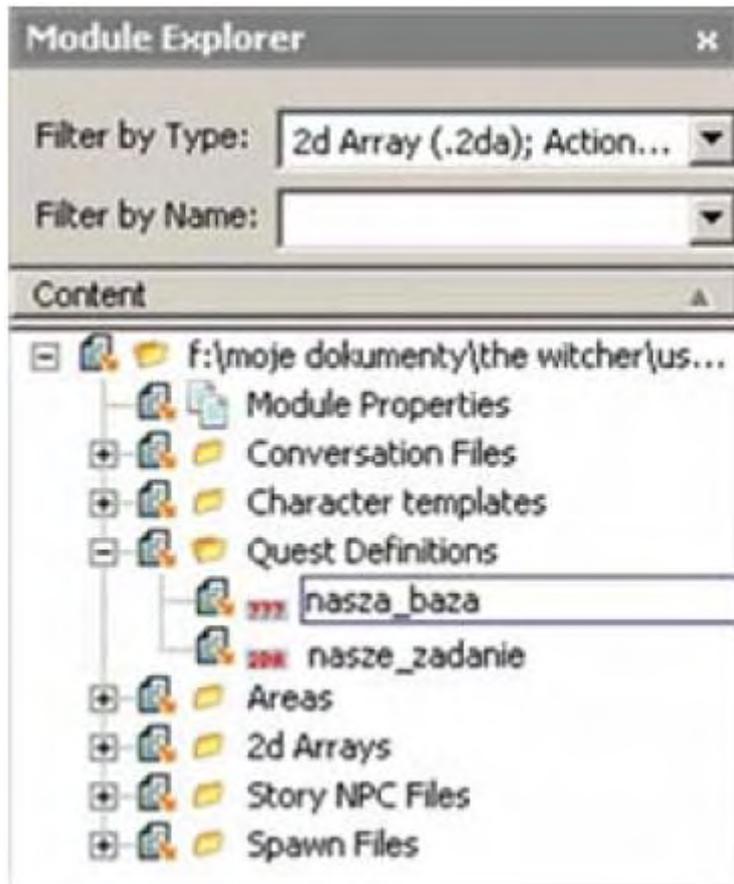
Jetzt werden wir noch die bereits angesprochenen Änderungen durchführen. Zuerst werden wir unsere Quest verändern, sodass der Banditenanführer und dessen Wachen erst erscheinen, wenn Geralt die Quest von Siegfried bekommen hat. Wie bereits erklärt, müssen wir nach dem hinzufügen der Datei zum Modul alle Änderungen im „Module Explorer“ einbringen. Gehe zum „Module Explorer“-Fenster. Wenn es geschlossen ist, kannst du es über das „View“-Menü mit „Utility Windows -> Module Explorer“ wieder aufrufen:



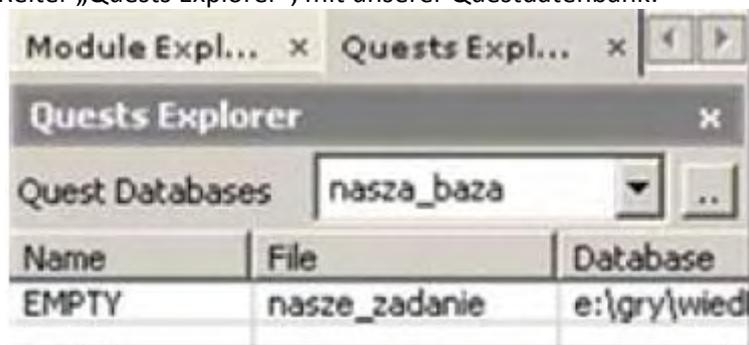
Der „Module Explorer“ wird auf der linken Seite des Bildschirms geöffnet:



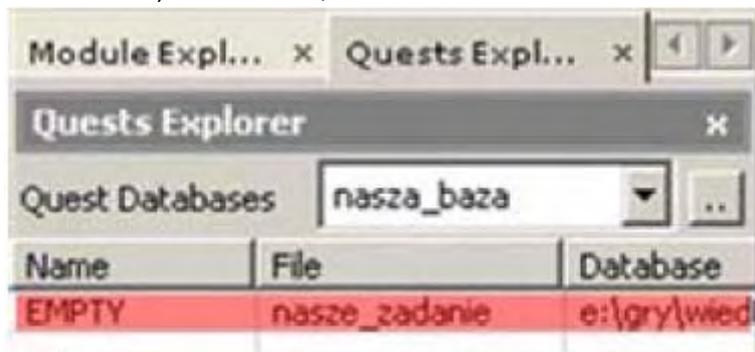
Erweitere den „Quest Definitions“-Zweig, indem du auf das „+“ daneben klickst. Doppelklicke danach auf „our_base“:



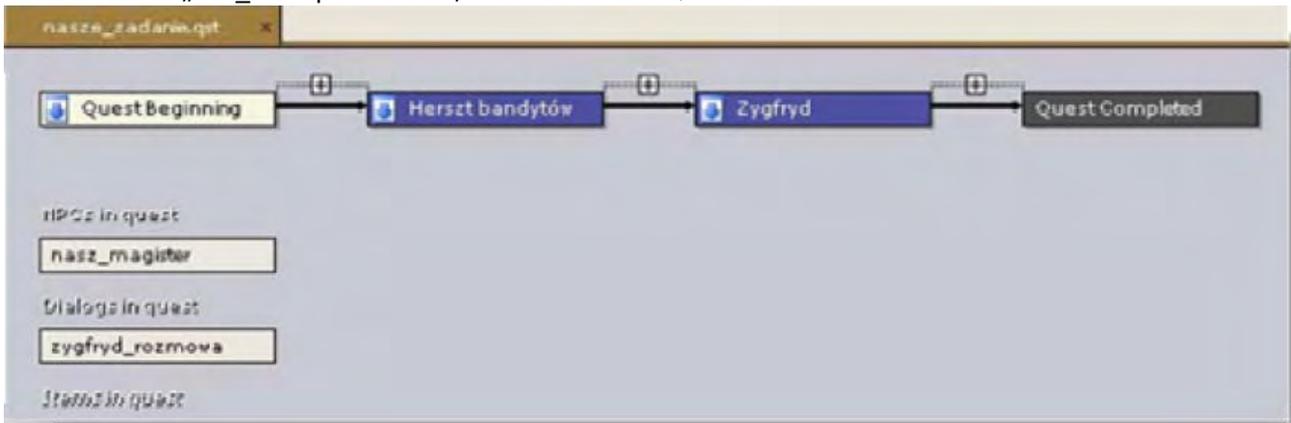
Damit öffnet sich der Reiter „Quests Explorer“, mit unserer Questdatenbank:



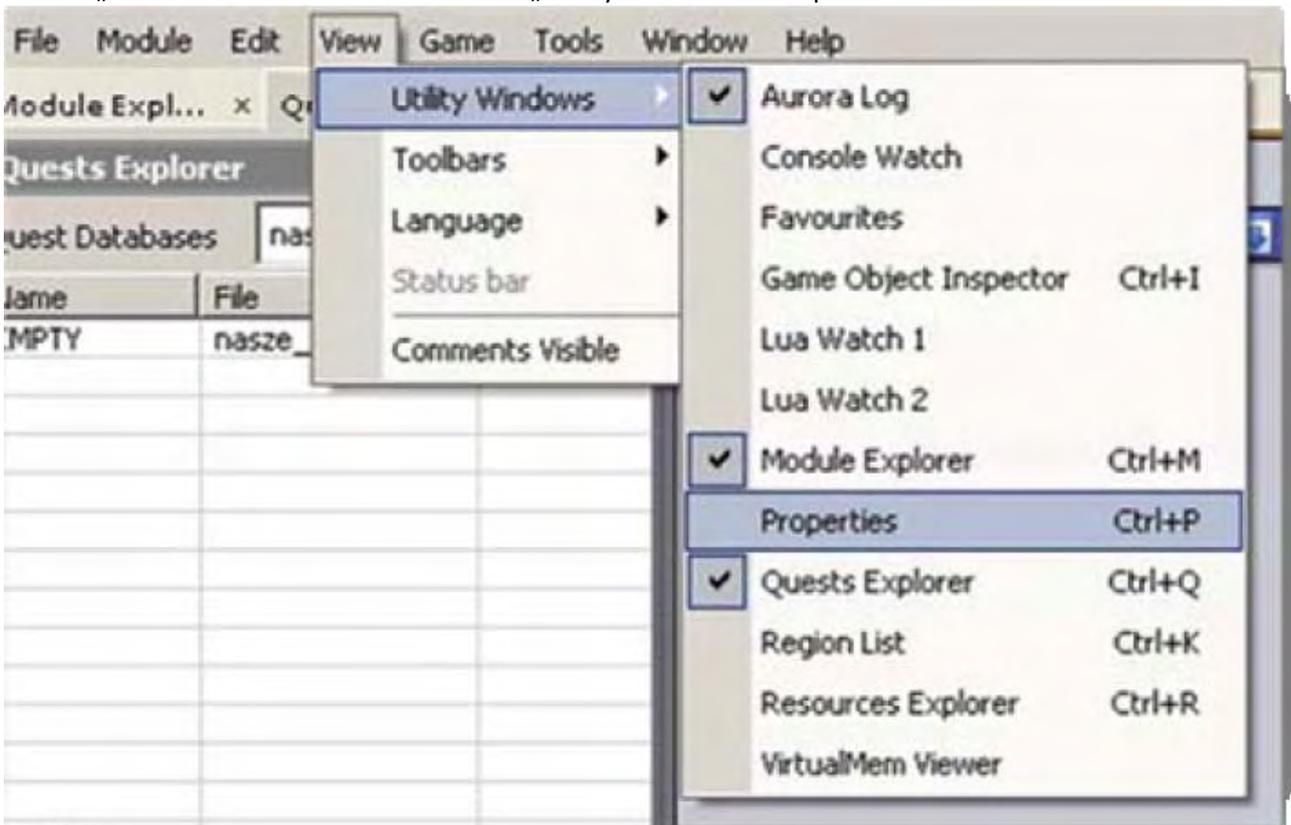
Klicke danach doppelt auf die Zeile, die unsere Quest beinhaltet:



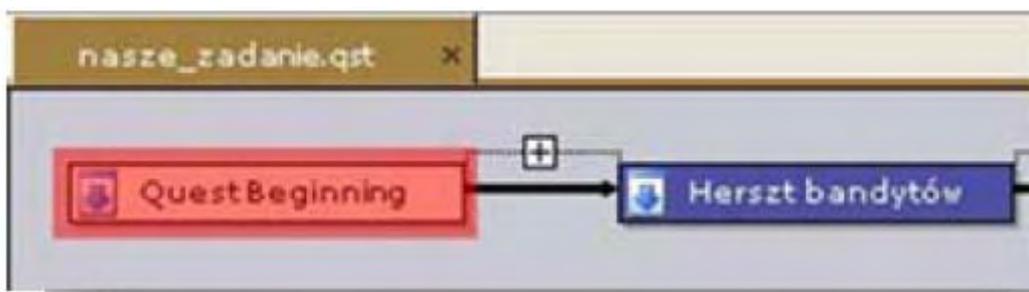
Das öffnet das „our_task.qst“-Fenster, welches unsere Quest enthält:



Jetzt müssen wir unsere bereits erwähnten Änderungen einarbeiten. Um das zu machen, klicke zunächst auf das „View“-Menü und anschließend auf „Utility Windows -> Properties“:



Das „Properties“-Fenster wird sich auf der rechten Seite des Bildschirms öffnen. Klicke nun auf „Quest Beginning“:



Gehe dann ins „Properties“-Fenster zurück. Wird werden die Änderungen in der Sektion „OnPhaseFinished Actions“ einarbeiten:

OnPhaseFinished Actions	
Available Actions	Transfer gold to 
1. Transfer gold to player	
gold_amount	100
npc_resref	nasz_zygfryd 

Wieso dort? Weil alle Aktionen aus dem „OnPhaseFinished Actions“ Bereich nach der Beendigung der jeweiligen Questphase auftreten. In diesem Falle ist „Quest Beginning“ für alles verantwortlich, was passiert, nachdem Geralt die Quest von Siegfried erhalten hat. Klicke auf die rechte Spalte, gegenüber des „Available Actions“-Attributes, und wähle „Enable spawnset“ aus der Liste:

OnPhaseFinished Actions	
Available Actions	Transfer go  
1. Transfer gold to player	
gold_amount	
npc_resref	
OnPhaseFinished Action	<ul style="list-style-type: none"> Add XP to playe  Transfer item to  Transfer item to  Change game p  Enable spawnse  Disabic spawnse 

Danach klickst du auf das blaue „+“ Symbol:

OnPhaseFinished Actions	
Available Actions	Enable spawnset 
1. Transfer gold to player	
gold_amount	100
npc_resref	nasz_zygfryd 

Damit wird die „Enable spawnset“-Aktion im Bereich „OnPhaseFinished Actions“ hinzugefügt:

OnPhaseFinished Actions	
Available Actions	Enable spawnset 
1. Transfer gold to player	
gold_amount	100
npc_resref	nasz_zygfryd 
2. Enable spawnset	
spn_resref	

Anschließend klickst du auf das kleine gelbe Symbol in der rechten Spalte, gegenüber vom „spn_resref“-Attribut:

OnPhaseFinished Actions	
Available Actions	Enable spawnset 
1. Transfer gold to player	
gold_amount	100
npc_resref	nasz_zygfryd 
2. Enable spawnset	
spn_resref	

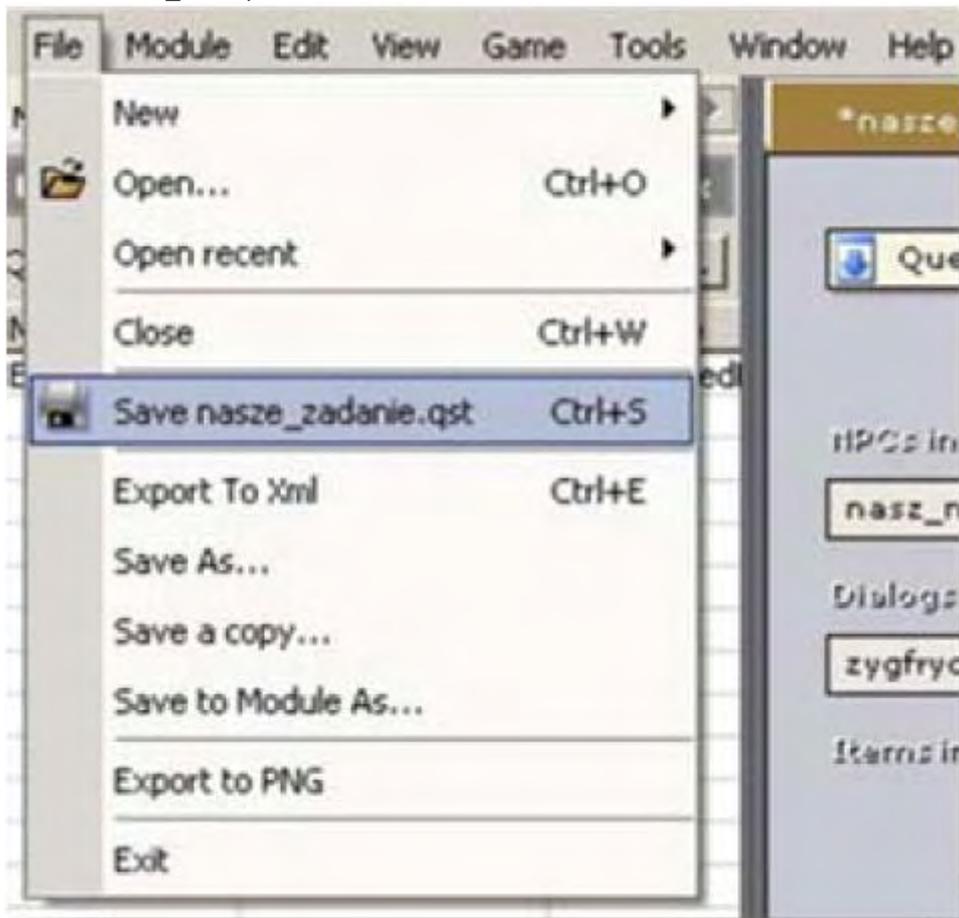
Damit öffnet sich das Standard-Fenster von Windows, zum Laden von Dateien. Lade das Spawn-Set des Professors, dem Anführer der Banditen. Gehe zum „Data\Templates\Spawnsets\Enemies“-Ordner, in welchem das Spawn-Set gespeichert ist. Wähle die Datei mit dem Namen „our_professor.spn“ aus und drücke auf „Öffnen“. Das „our_professor“ Spawn-Set wird gegenüber vom „spn_resref“-Attribut erscheinen:

OnPhaseFinished Actions	
Available Actions	Enable spawnset 
1. Transfer gold to player	
gold_amount	100
npc_resref	nasz_zygfryd 
2. Enable spawnset	
spn_resref	nasz_magister 

Nun müssen wir das Gleiche für die Wächter des Professors tun. Für jeden muss eine „Enable Spawnset“-Aktion im „OnPhaseFinished Actions“-Bereich hinzugefügt werden. Klicke auf das kleine gelbe Symbol gegenüber von „spn_resref“ und ordne die Datei „our_bandit.spn“ zu:

OnPhaseFinished Actions	
Available Actions	Enable spawnset 
1. Transfer gold to player	
gold_amount	100
npc_resref	nasz_zygfryd 
2. Enable spawnset	
spn_resref	nasz_magister 
3. Enable spawnset	
spn_resref	nasz_bandyta 

Das war's! Die Änderungen wurden vorgenommen und müssen nur noch gespeichert werden. Wir müssen die aktualisierte Aktion und das Modul speichern. Fangen wir mit der Quest an, wähle dazu aus dem „File“-Menü den Punkt „Save our_task.qst“:



Sollte „Save our_task.qst“ nicht in dem Menü erscheinen, oder ein anderer Name angezeigt werden, dann bedeutet dies, dass das „our_task.qst“-Fenster momentan nicht aktiv ist. Um es wieder zu aktivieren, brauchst du nur einmal das entsprechende Fenster anklicken und dann mit „Save our_task.qst“ speichern. Danach speicherst du das Modul. Hierzu musst du im „Module“-Menü den Befehl „Save Module“ auswählen:

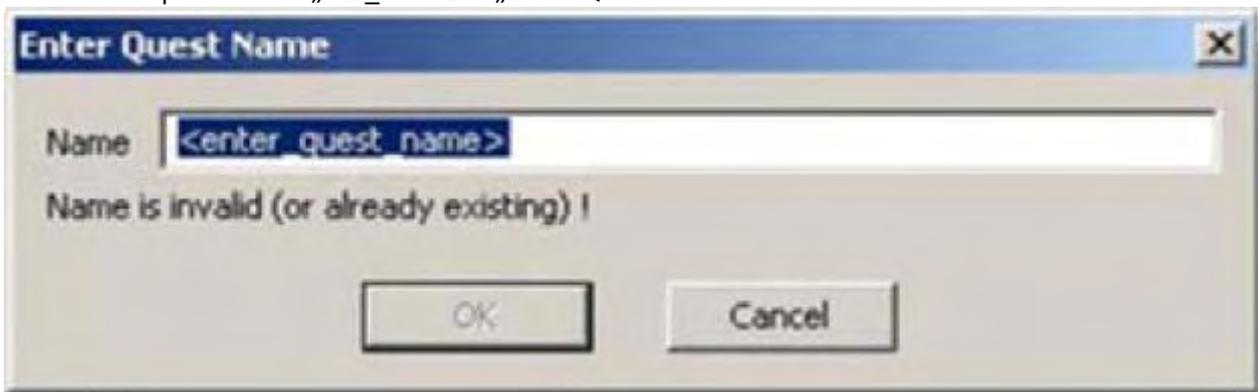


Wir müssen trotzdem noch eine zweite Quest erstellen. Diese wird als „einleitende Quest“ bezeichnet, welche automatisch mit unserem Adventure aktiviert wird, sobald das Spiel startet. Normalerweise wird der Spieler von dieser Quest nichts bemerken – sie schaltet nur die notwendigen Charakter Spawn-Sets ein.

Nun werden wir uns damit auseinandersetzen, wie eine Quest automatisch zur Quest-Datenbank hinzugefügt wird. Um das zu erreichen gehe zu erst ins „Quest Explorer“-Fenster. Klicke mit der rechten Maustaste auf irgendeine Zeile und wähle „Add New Quest To -> our_base“ aus dem Menü:



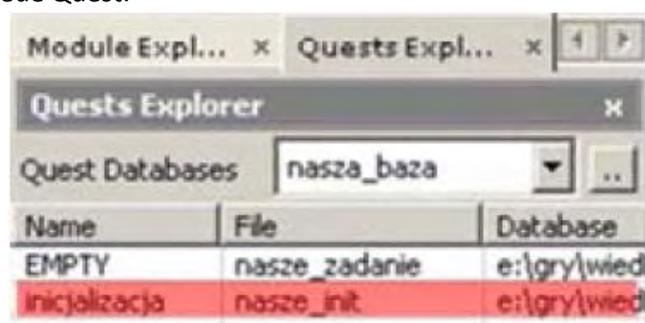
Das öffnet das Standard-Fenster von Windows zum Speichern von Dateien. Gehe in den „Data\Quests“-Ordner und speichere als „our_init“. Das „Enter Quest Name“-Fenster erscheint darauf hin:



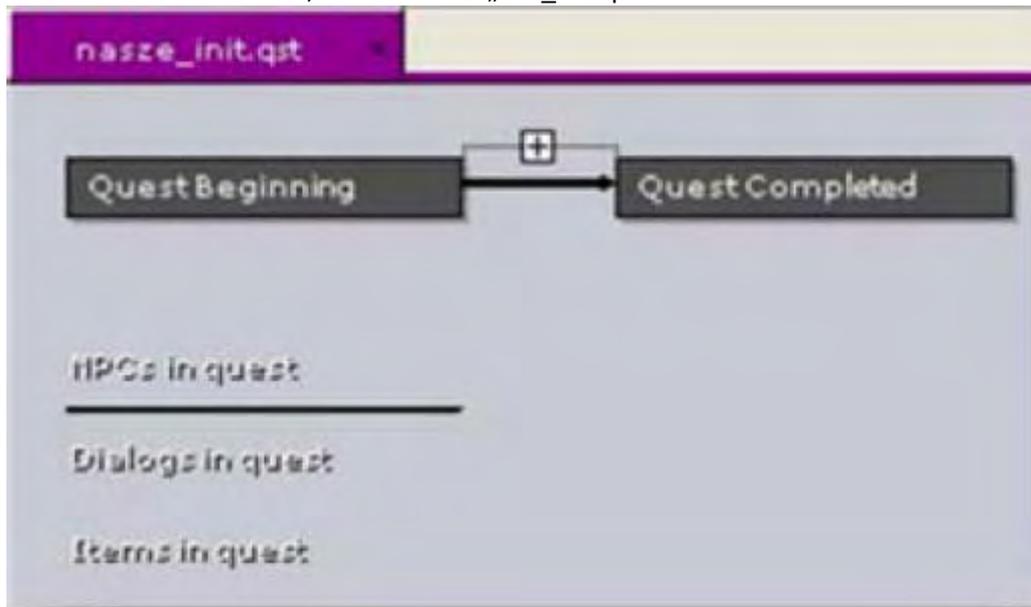
Gib „initialization“ ein und klicke auf „OK“. Die „initialization“-Quest wird nun automatisch in unsere Datenbank eingefügt.



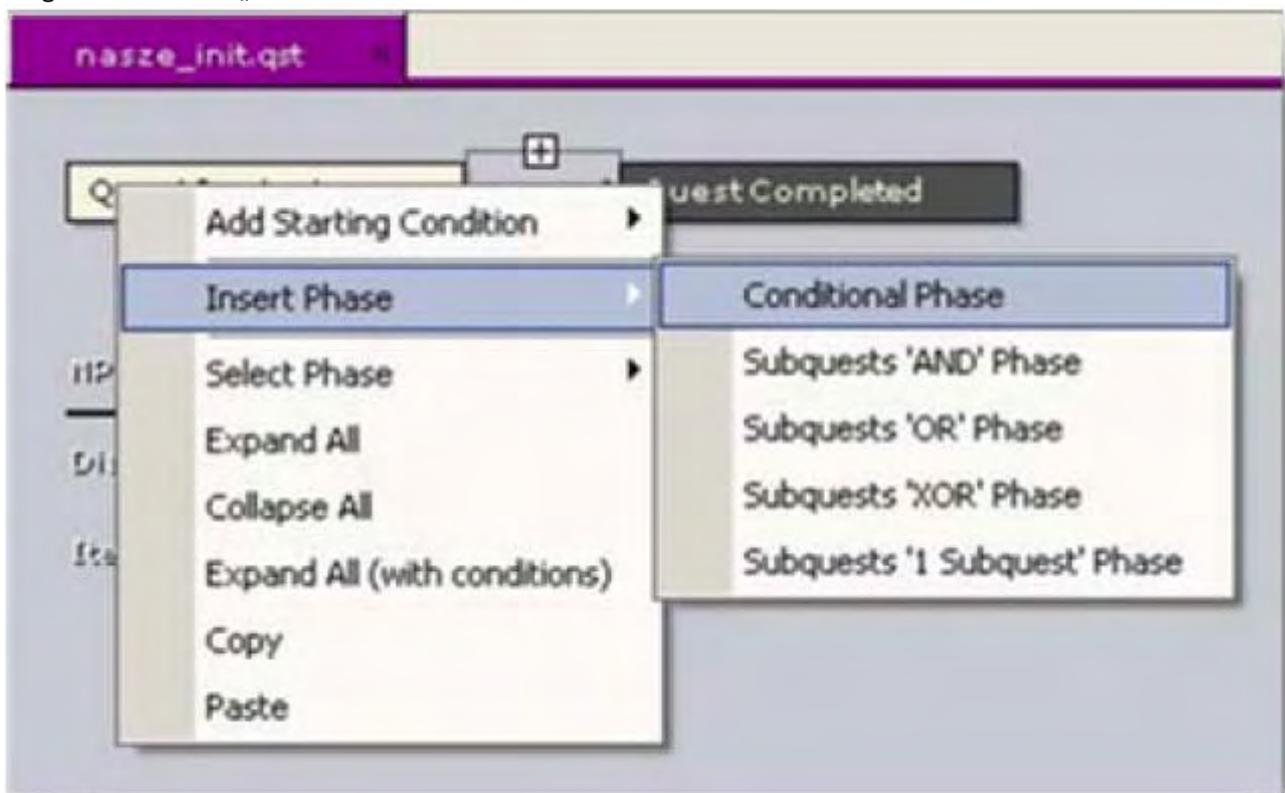
Doppelklicke auf unsere neue Quest:



Damit öffnet sich ein neues Fenster, mit dem Titel „our_init.qst“:

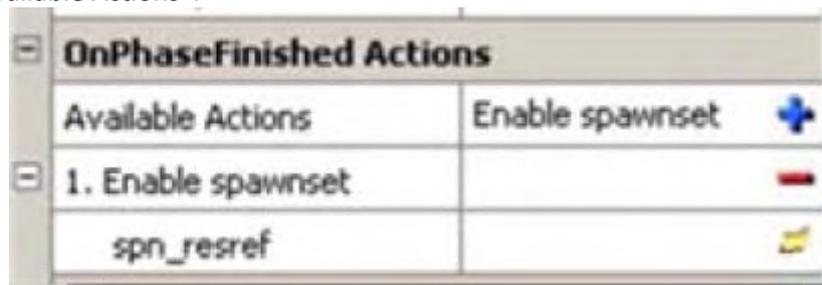


Da die Quest nur benötigt wird um unser Abenteuer zu starten brauchen wir keine Quest-Phasen, Namen, usw. einzugeben. Die Quest benötigt lediglich eine Bedingungs-Phase. Klicke mit rechts auf „Quest Beginning“ und dann auf „Insert Phase -> Conditional Phase“:

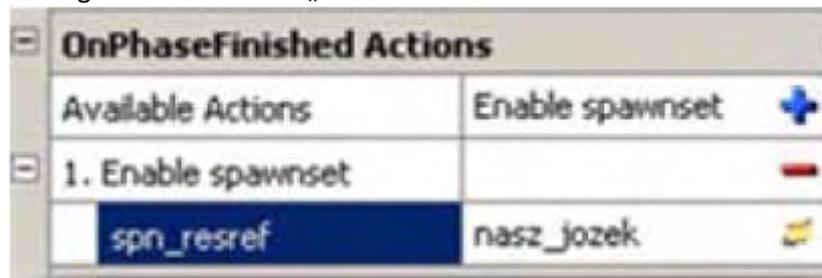


Ein neues „Enter name“-Fenster erscheint. Lasse den vorgegebenen Name auf „Phase 1“ und klicke einfach auf „OK“. Wechsle nun ins „Properties“-Fenster und öffne die „OnPhaseFinished Actions“-Sektion. Da „Quest-Beginning“ keine Aktionen zugeordnet wurden wird diese Quest automatisch aktiv, wenn das Modul geladen wird. Damit werden auch automatisch alle Aktionen der „OnPhaseFinished Actions“-Gruppe zu beim Laden des Moduls ausgeführt.

Diese Quest wird nur benutzt, um die Spawn-Sets von Jethro und den Wölfen zu aktivieren. Dafür gehst du zur „Enable spawnset“-Aktion im „OnPhaseFinished Actions“-Bereich. Klicke dazu auf das blaue „+“ Zeichen gegenüber von „Available Actions“:

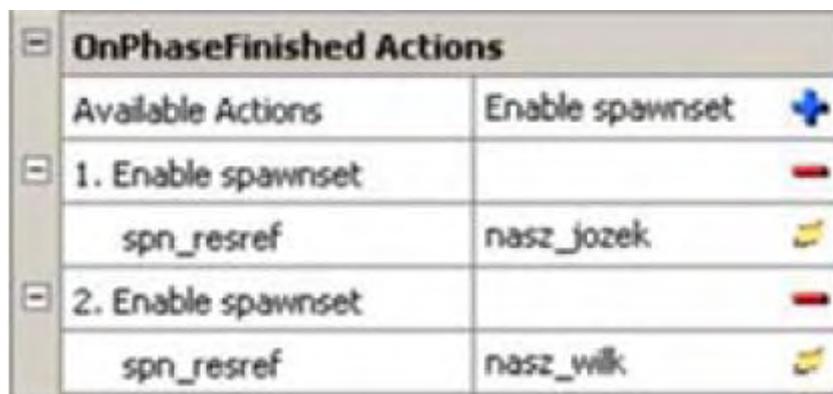


Anschließend klickst du wieder auf das gelbe Symbol gegenüber von „spn_resref“. Wähle im folgenden Auswahl-Bildschirm den Ordner „Data\Templates\Spawnsets\Commoners“ und dort die Datei „our_jethro.spn“. Bestätige mit dem Button „Öffnen“:

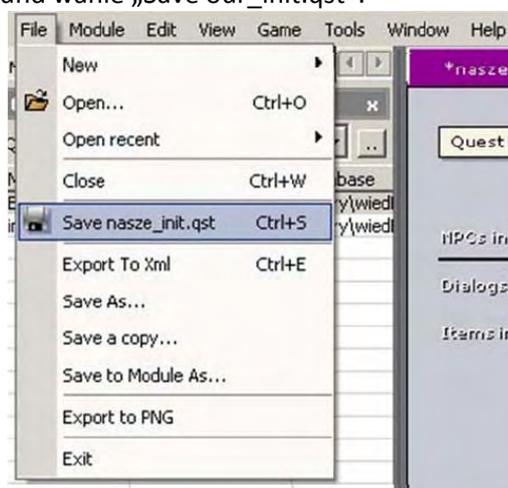


Die Datei „our_jethro.spn“ wurde dem Attribut „spn_resref“ zugewiesen.

Jetzt müssen wir nur noch eine weitere „Enable spawnset“-Aktion für die Wölfe hinzufügen. Weise die „our_wolf.spn“-Datei, welche im „Data\Templates\Spawnsets\Enemies“-Ordner liegt, dem „spn_resref“-Attribut zu:



Das war's für diese Aufgabe. Vergiss nicht, dass unsere Änderungen natürlich noch gespeichert werden müssen. Öffne das „File“-Menü und wähle „Save our_init.qst“:



Sollte „Save our_init.qst“ nicht vorhanden sein, heißt das, dass du zuerst das „our_init.qst“-Fenster anklicken musst, sodass dieses aktiv ist. Dazu reicht ein einfacher Klick auf das entsprechende Fenster. Anschließend solltest du auch „Save our_init.qst“ aus dem „File“-Menü auswählen können.

Damit beenden wir eine bestimmte Phase in der Erstellung unseres Abenteuers. Alles sollte nun so eingerichtet sein, damit unser Abenteuer funktioniert. Wenn du dieses Handbuch von Anfang an bis hier her aufmerksam gelesen hast, solltest du nun in der Lage sein, eigene Abenteuer zu kreieren. Lass dich nicht entmutigen, wenn deine ersten Versuche nur einfach und schlicht sind – mit der Zeit werden sich dein Können und dein Einfallsreichtum unweigerlich verbessern und du wirst viel Erfahrung im Umgang mit D’Jinni sammeln.

Im nächsten Kapitel werden wir uns mit den etwas komplizierteren Dingen des Editors beschäftigen. Das könnte unter anderem die Erstellung von Zwischensequenzen und Spezialeffekten sein, wie auch das Schreiben von eigenen Scripten.

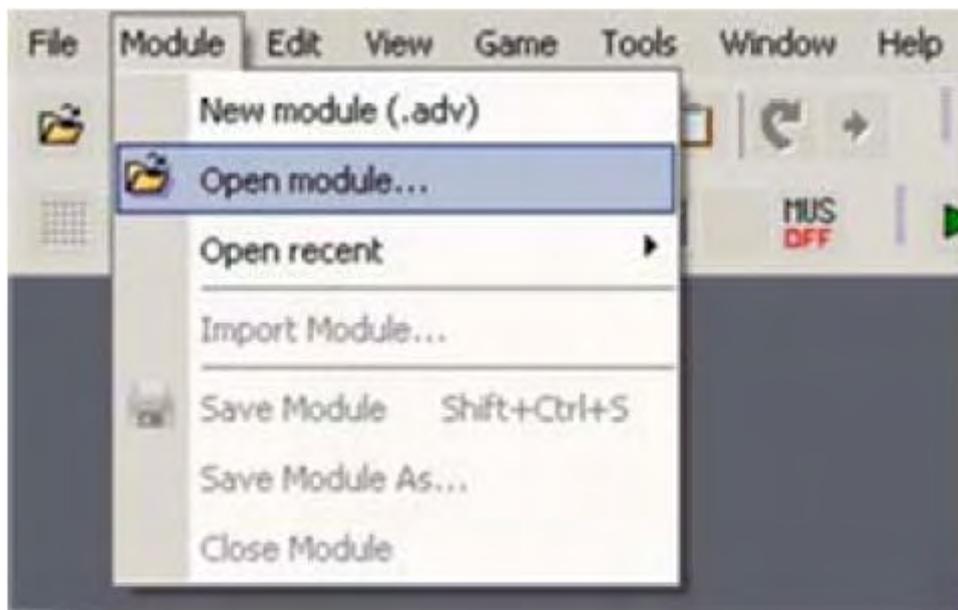
Cutscenes / Zwischensequenzen

Einleitung:

In diesem Kapitel werden wir uns mit der Erstellung einer Zwischensequenz auf Basis der Witcher-Engine beschäftigen. Während des Spielens hast du sicher schon einige dieser Zwischensequenzen gesehen, die vom Entwicklerteam erstellt wurden. Unsere Cutscene wird natürlich nicht so komplex sein.

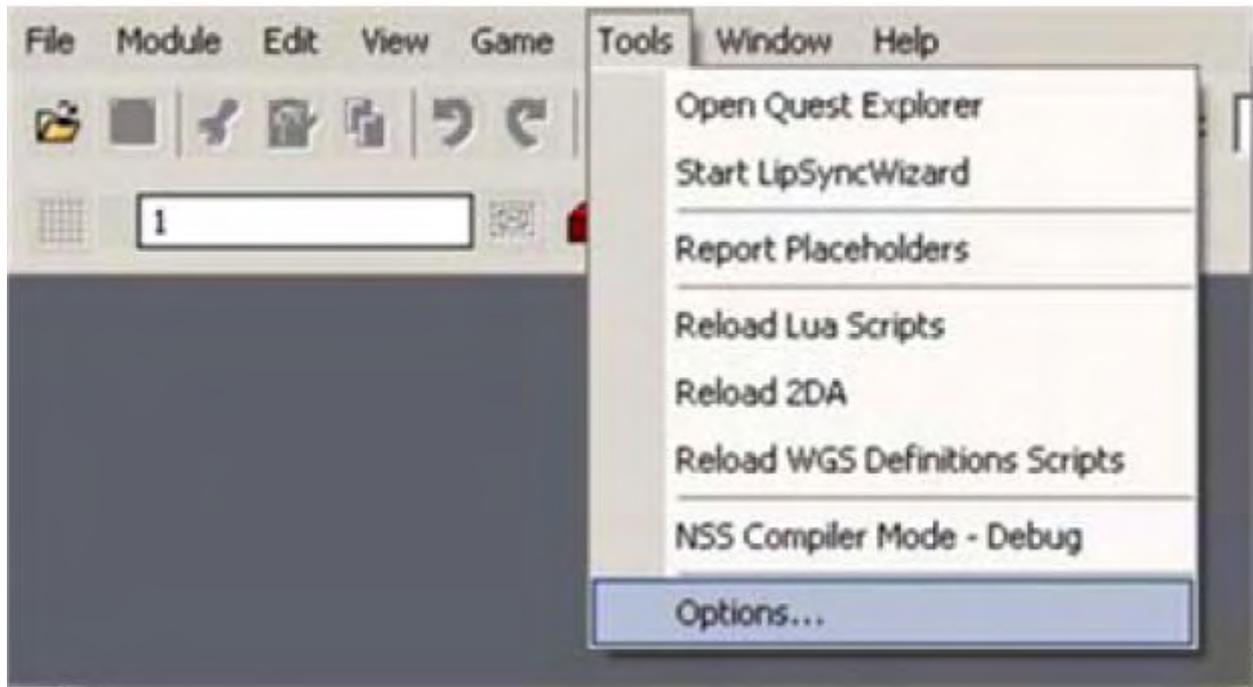
Wir werden eine relativ einfache Zwischensequenz erstellen, bei der die Kamera durch eine vorgegebene Landschaft fliegt.

Um dies zu machen, muss auf jeden Fall zuerst ein Modul geladen werden. Dies ist wichtig, denn du kannst keine Cutscene erstellen ohne bereits vorher ein Modul vorbereitet zu haben, welches mindestens eine Area enthalten muss. Wenn noch kein Modul geladen ist, klicke auf „Open Modul“ aus dem „Module“-Menü:

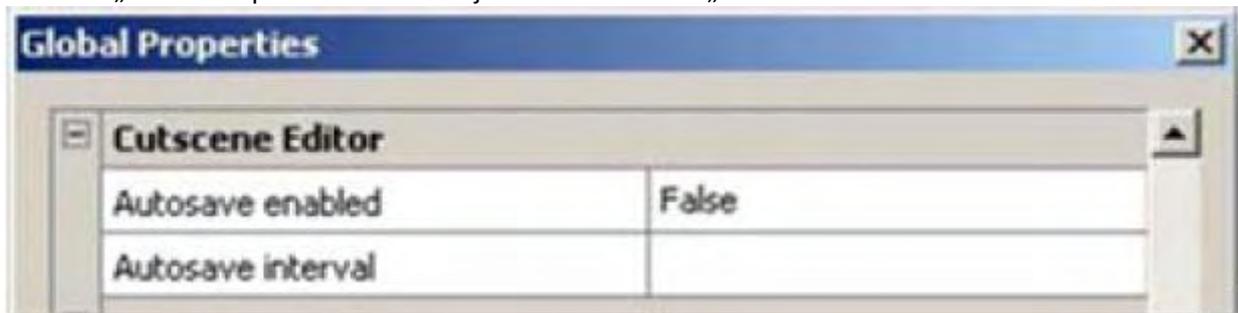


Nun öffnet sich das Standard Fenster von Windows zum Dateien auswählen. Navigiere zu „Dokumente und Einstellungen \ dein Benutzername \ Eigene Dateien \ The Witcher“ (hier sind alle Module gespeichert), wähle „our_adventure.adv“ aus und klicke auf „Öffnen“. Das Modul wird nun geladen.

Der Cutscene-Editor braucht jetzt möglicherweise ein wenig Zeit und Leistung. Denke daran, deinen Fortschritt ständig zu speichern, denn leider ist der Editor nicht so stabil, wie wir es gerne hätten. Du kannst auch die Autosave-Funktion einschalten, dann speichert das Programm dein Projekt gelegentlich automatisch (hängt von deiner Einstellung ab). Um diese Funktion einzuschalten, gehe auf das „Tools“-Menü und wähle „Options“:



Im Fenster „Global Properties“ suchst du jetzt den Abschnitt „Cutscene Editor“:



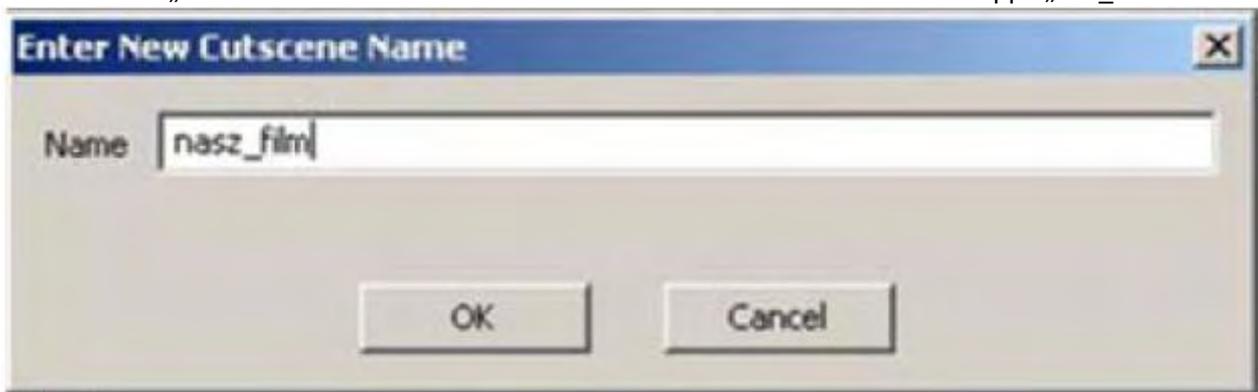
Um das automatische Speichern zu aktivieren, musst du bei „Autosave enabled“ den Wert auf „True“ setzen. Darunter kannst du bei „Autosave interval“ bestimmen, in welchen Abständen automatisch gespeichert werden soll. Klicke auf OK wenn du fertig bist, um diese Einstellungen zu speichern.

Erstellen einer Cutscene

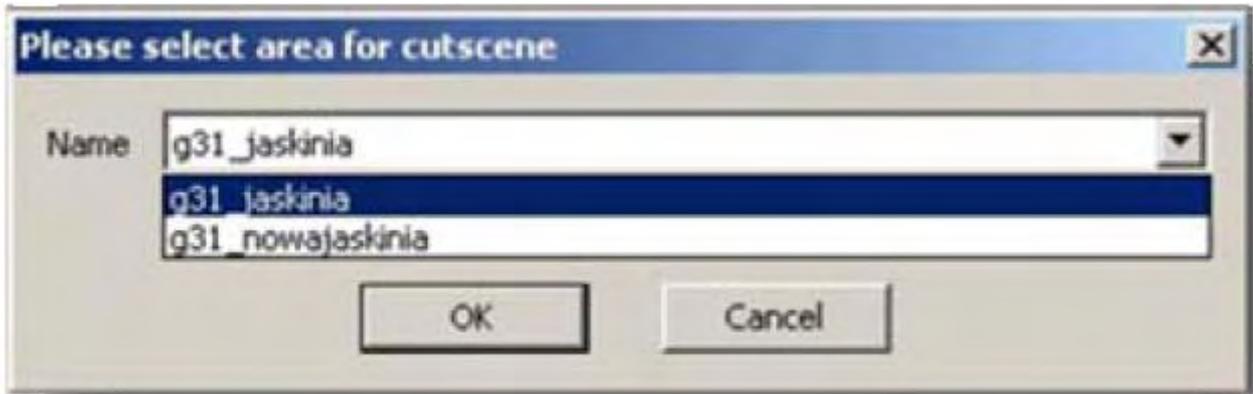
Um eine Zwischensequenz zu erstellen wähle zuerst „New -> Cutscene (.cut)“ aus dem „File-Menü“:



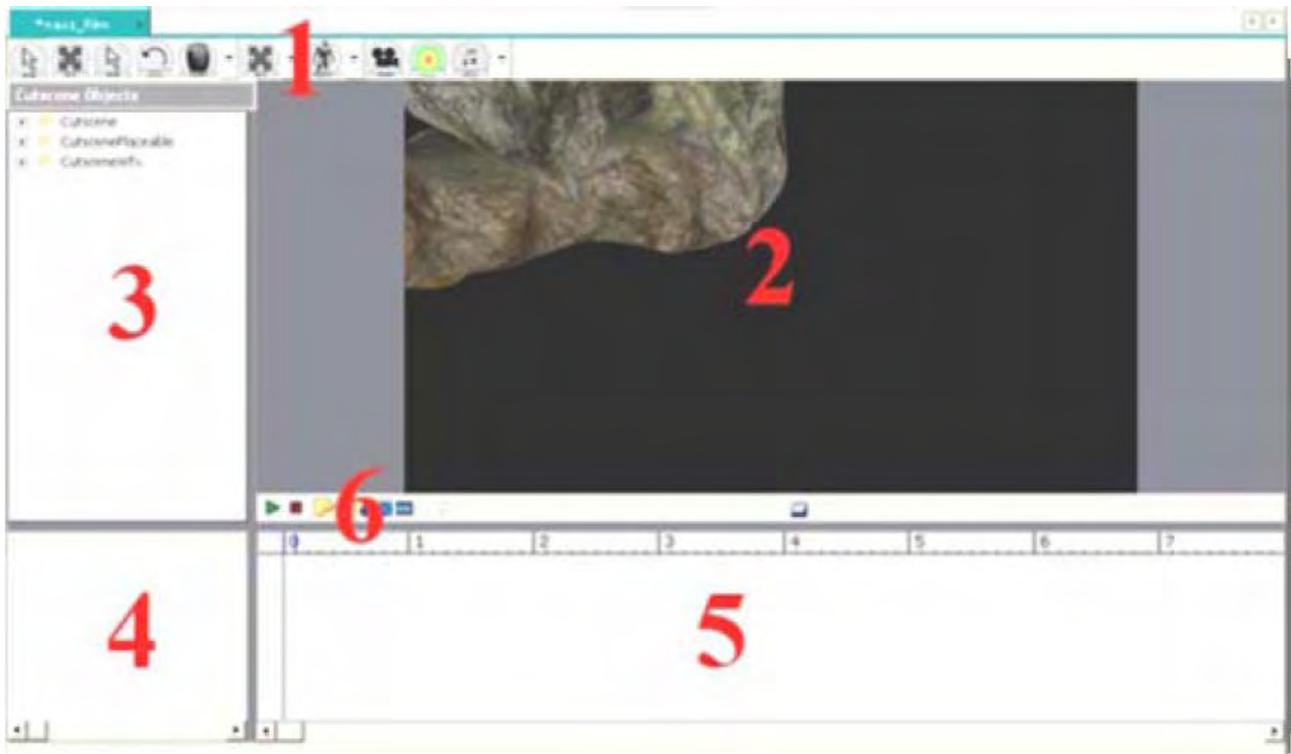
Gib im Fenster „Enter New Cutscene Name“ einen Namen für deine Cutscene ein. Tippe „our_film“ ein:



Bestätige mit OK. Weil unser Modul zwei Areale enthält erscheint ein neues Fenster mit dem Titel „Please select area for cutscene“. Das Fenster enthält eine Liste mit allen verfügbaren Arealen. Wähle „g31_cave“ aus:



Das war die erste Area, die wir erstellt haben. Klicke auf „OK“ und das Film-Editor Fenster öffnet sich. Bevor wir weitermachen schauen wir uns die Oberfläche des Editors an:



1. Die Werkzeugleiste des Editors. Sie ist sehr ähnlich mit der aus dem Area-Editor.
2. Die Arbeitsfläche, in der die aktuelle Cutscene erstellt wird. Sie funktioniert auf dieselbe Weise wie der Area-Editor. Der einzige Unterschied besteht darin, wie Objekte dargestellt werden. Zum Beispiel beinhaltet diese Arbeitsfläche keine Objekte wie Start-, Spawn- und Actionpunkte, etc.
3. Die Liste mit Objekten, die in dieser Cutscene verwendet wurden.
4. Die Liste mit Objekt-Pfaden.
5. Cutscene-Symboleiste (wird zum Abspielen oder anhalten der Sequenz benutzt).

Wie du sehen kannst, ist unser Cutszene-Editor in 4 Fenster aufgeteilt, bei dem das größte Fenster unsere Area enthält. Im Wesentlichen ist es das gleiche wie der Area-Editor, nur ohne ein paar bestimmte Werkzeuge (zum hinzufügen von Start-, Spawn- und Actionpunkten, etc.), dafür aber auch ein paar neue (zum Beispiel zum hinzufügen von Charakteren). Zusätzlich werden Objekte wie Spawn- und Actionpunkte nicht angezeigt. Die Cutszene-Werkzeugleiste ist im Grunde die gleiche wie beim Area-Editor, deswegen gibt es nachfolgend nur eine kurze Information für die Wichtigsten. Eine detaillierte Beschreibung zur Benutzung dieser Werkzeuge kann im Modul-Erstellungs-Abschnitt dieses Handbuchs gefunden werden.



Dieses Werkzeug wird benutzt, um Objekte auszuwählen, die sich in der Area befinden.



Dieses wird zum verschieben von Objekten benutzt. Um dies zu bewerkstelligen, musst du zuerst das Icon dieses Werkzeugs anklicken und anschließend das Objekt, welches du bewegen möchtest.



Dieses Werkzeug wird benutzt, um Objekte mit allen ihren zugehörigen Keyframes zu verschieben. Keyframes sind Wegweiser in einem Sequenz-Pfad, welche eine bestimmte Aktion beginnen, z.B. das Drehen der Kamera. Mehr über Keyframes erfährst du später in diesem Kapitel.



Mit diesem Werkzeug kannst du Objekte drehen, indem du zuerst dieses Icon und dann das zu drehende Objekt anklickst.



Hiermit kannst du Gegenstände, wie z.B. Stühle, Tische, Kisten, Kamine,... hinzufügen.



Dieses Werkzeug fügt Spezialeffekte ein. Spezialeffekte sind z.B. verschiedene Flammen, Zauberspruch-Effekte, usw.



Hiermit fügt man Charaktere ein. Um einen bestimmten Charakter in einer Cutscene zu benutzen, müssen wir ihn zuerst hinzufügen. Wenn wir zum Beispiel „Siegfried“ oder „Jethro“ in unserer Sequenz haben möchten, fügen wir ihn erst mit diesem Tool hinzu. Entweder muss eine Vorlage für diesen Charakter erstellt werden oder eine bereits existierende benutzt werden. Die Grundregeln die damit zusammenhängen, sind die gleichen wie beim hinzufügen von Gegenständen – drücke auf den Pfeil direkt neben dem Werkzeug und wähle „Select new Cutscene Character template“ aus. Anschließend die passende Datei im „Auswählen“-Fenster selektieren und danach im Area-Fenster auf den Punkt klicken, wo der Charakter eingefügt werden soll.



Dient zum hinzufügen von Kameras. Die Cutscene wird durch ihren Linsen dargestellt.



Hiermit fügt man Objekte ein, wie z.B. „look at“ („schaue auf“). Das sind unterstützende oder zusätzliche Objekte.

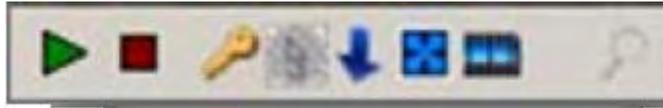


Dieses Werkzeug fügt Sounds in die Area ein.

Nun haben wir die Werkzeuge besprochen, nun wird es Zeit eine Cutscene zu erstellen. Unsere Sequenz wird relativ leicht zu erstellen und ziemlich kurz sein. Grundsätzlich ist es eine Kamera, die sich quer durch eine Höhle bewegt. Um dies zu verwirklichen werden wir zwei Kameras benutzen. Die erste Kamera wird einem linearen Pfad folgen, während die zweite sich dreht und rotiert. Zuerst müssen wir einen sinnvollen Startpunkt finden. Dies wird genauso wie im Area-Editor gemacht: Zum Beispiel werden die Pfeiltasten \leftarrow \uparrow \downarrow \rightarrow oder WASD benutzt um sich zu bewegen. Eine Rotation wird bewerkstelligt, indem man die rechte Maustaste drückt und hält. Da die erste Kamera einem linearen Pfad folgt wird sie in dem Korridor platziert, welcher zu einer Kammer mit dem Startpunkt führt:



Bevor wir die Kamera einfügen schauen wir uns die Cutszene-Symbolleiste mal an:



1. Abspielen.
2. Stop.
3. Automatisches hinzufügen von Keyframes aktivieren/deaktivieren.
4. Wird nicht benutzt.
5. Wird nicht benutzt.
6. In 16:9 Formate wechseln (dieses Format wird bei Cutszenes verwendet).
7. Schaltet den oberen und unteren Rand der Anzeige an/aus.
8. Wird nicht benutzt.
9. Zusätzliche Objekte in der Area anzeigen/verstecken.

Vergewissere dich, dass das automatische hinzufügen von Keyframes aktiviert ist, also das der entsprechende Button gedrückt und gelb hervorgehoben ist:

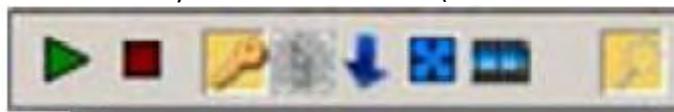


Klicke jetzt auf das „Kamera hinzufügen“-Werkzeug:



Ein Kamera-Icon wird jetzt auf dem Mauscursor erscheinen. Klicke auf den Punkt in der Area, an dem die Kamera platziert werden soll. Mach dir keine Gedanken über deren exakte Position, denn die Kamera hat interessante Funktionen, die wir in Kürze kennenlernen.

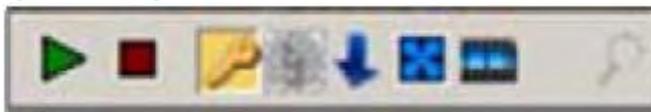
Platziere die Kamera irgendwo am Boden. Wie du vielleicht bemerkt hast, wird die Kamera nicht in der Area angezeigt. Um die Kamera sichtbar zu machen musst du den Button für „Zusätzliche Objekte in der Area anzeigen/verstecken“ in der Cutszene-Symbolleiste anklicken (das ist der Button mit der Lupe):



Wenn dieser Button gelb hervorgehoben ist, sollten die Kamera und andere zusätzliche Objekte in der Area sichtbar sein:



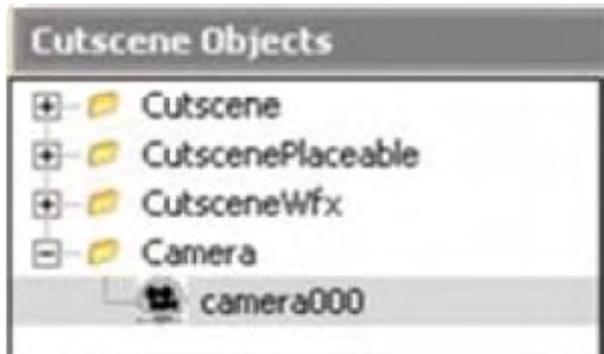
Vorerst solltest du die zusätzlichen Objekte wieder verstecken, also klicke erneut auf den Button mit der Lupe, sodass er nicht mehr gelb hervorgehoben ist.



Die Kamera wurde außerdem im „Cutszene Objects“-Feld hinzugefügt. Ein neuer Baum mit dem Namen „Camera“ ist erschienen. Mit einem Klick darauf klappt sich dieser aus und das Objekt „camera000“ wird angezeigt:



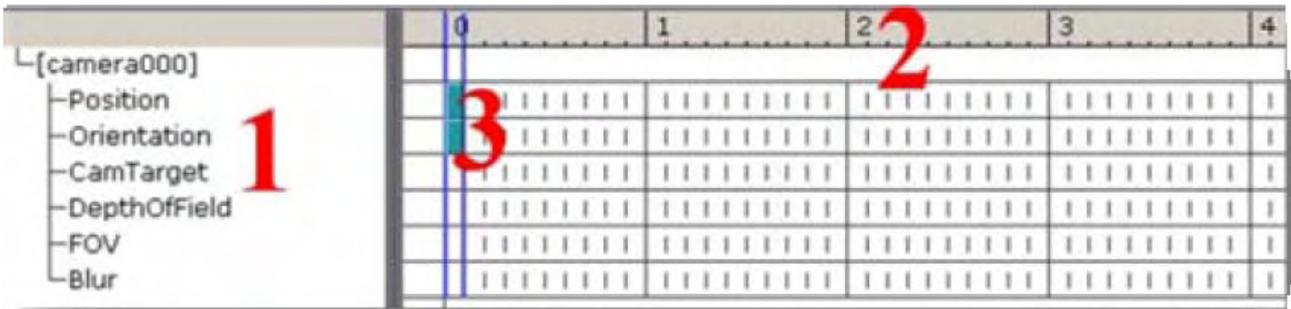
Dies ist die Kamera, welche wir eben hinzugefügt haben. Wähle im „Cutszene Object“-Feld mit einem Klick aus, sodass sie hervorgehoben wird:



Damit wird die Kamera auch im Area-Editor hervorgehoben:



Jedes ausgewählte Objekt wird als weiße Kontur eines Würfels erscheinen. Wenn die Kamera ausgewählt wurde schauen wir einmal auf den unteren Teil des Cutszene-Editors:



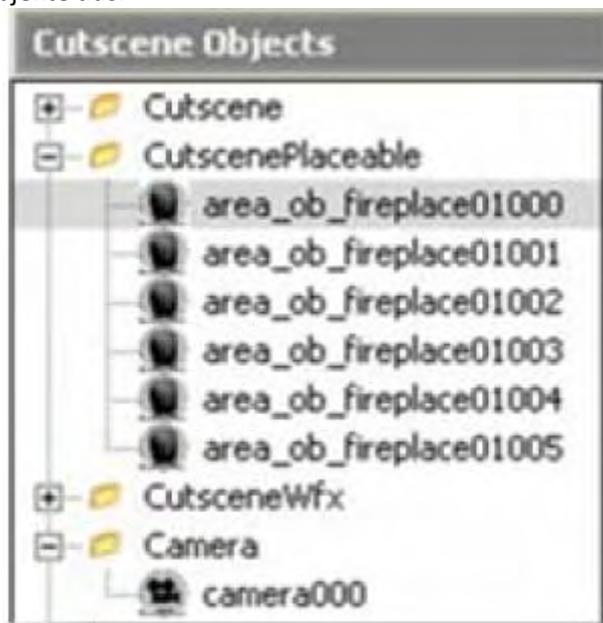
1. Liste mit Objekt-Pfaden
2. Zeitleiste
3. Keyframes

Jedes Objekt einer Sequenz hat eine bestimmte Nummer von Pfaden. Die Art der Pfade, die mit einem Objekt in Verbindung stehen, hängt von dessen Wesen ab. Die Kamera weist sechs solcher Pfade auf:

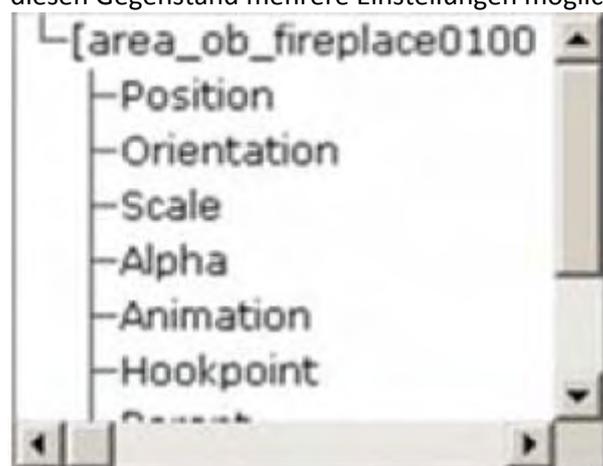


- [camera000] -> Name des Objekts; besitzt keinen eigenen Pfad
- Position -> Position der Kamera
- Orientation -> Orientierung der Kamera
- CamTarget -> Ziel, auf welches die Kamera gerichtet ist
- DepthOfField -> Tiefe der Kameraansicht
- FOV -> Blickwinkel der Kamera
- Blur -> Unschärfe der Kamera

Klicke nun im „Cutszene Objects“-Fenster auf den „Cutszene Placeable“-Baum um ihn auszuklappen. Wähle nun eins der verfügbaren Objekte aus:

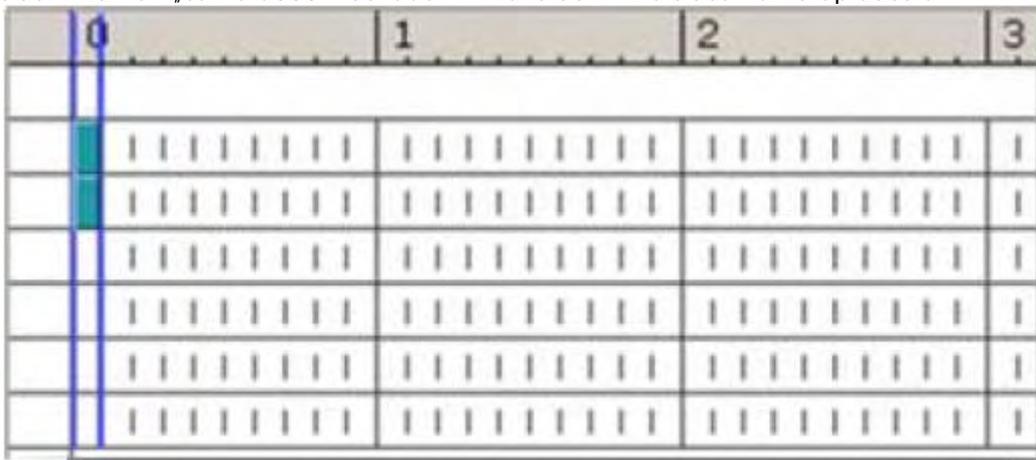


Du wirst bemerken, dass für diesen Gegenstand mehrere Einstellungen möglich sind:

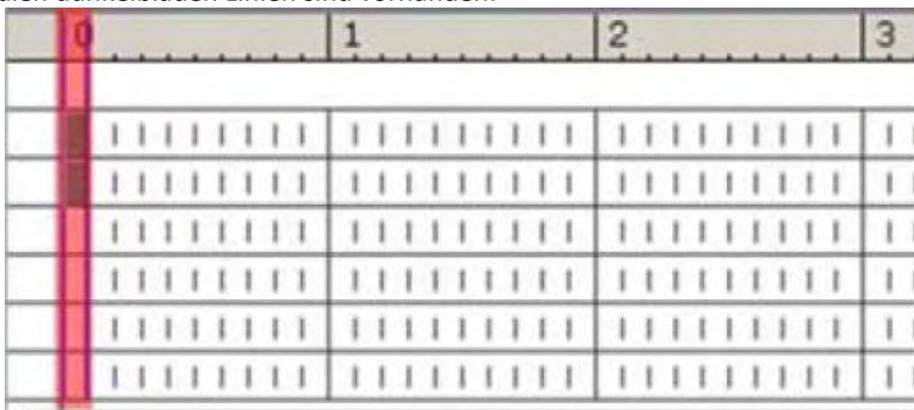


- [area_ob_fireplace0100] -> Der Name des Objekts; besitzt keinen eigenen Pfad
- Position -> Position des Objekts
- Orientation -> Ausrichtung des Objekts
- Scale -> Skalierung des Objekts (Größenverhältnis)
- Alpha -> Durchsichtigkeit des Objekts
- Animation -> Animation des Objekts
- Hookpoint -> Punkt, von welchem das Objekt abhängig ist
- Parent -> Elternteil des Objekts (wenn das Objekt mit einer anderen Hierarchie verbunden ist)
- LocalPosition -> Lokale Position des Objekts
- Visible -> Sichtbarkeit des Objekts

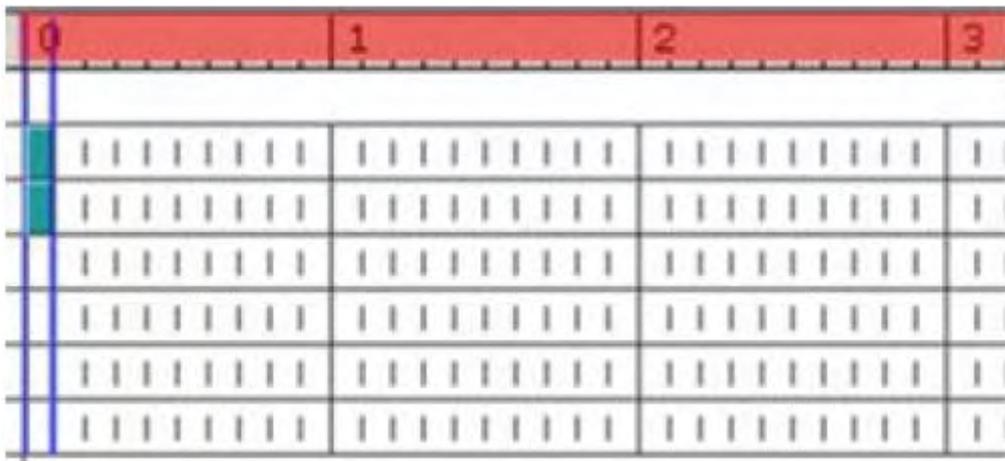
Jeder dieser Pfade kann zu gegebener Zeit verändert werden, um bestimmte Effekte zu erzielen. Um dies zu demonstrieren erhöhen wir die Skalierung des Objekts. Dafür muss der „Scale“-Pfad bearbeitet werden, z.B. soll das Objekt nach zwei Sekunden doppelt so groß sein. Zunächst kommen wir aber zurück zu unserer Kamera. Klicke im „Cutscene Objects“-Fenster auf das Objekt mit dem Namen „camera000“. Schauen wir uns den Inhalt des Kamerapfades an:



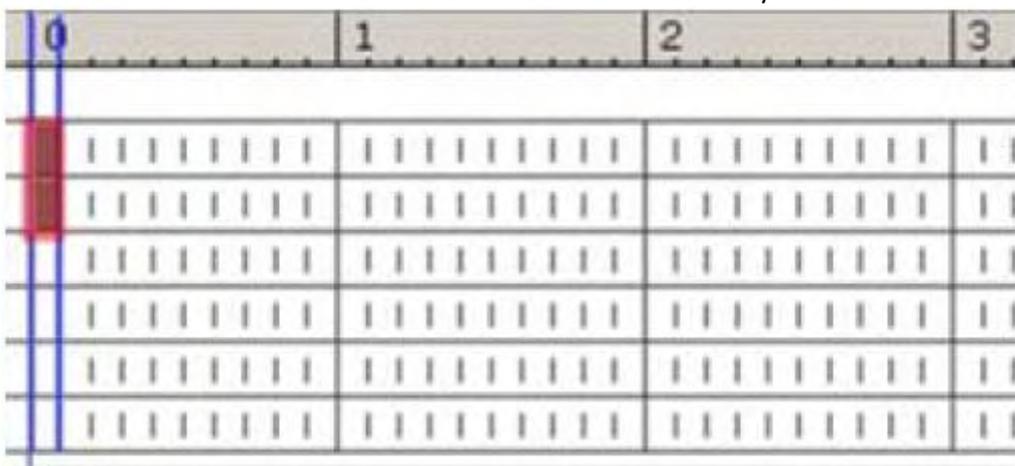
Die zwei vertikalen dunkelblauen Linien sind vorhanden:



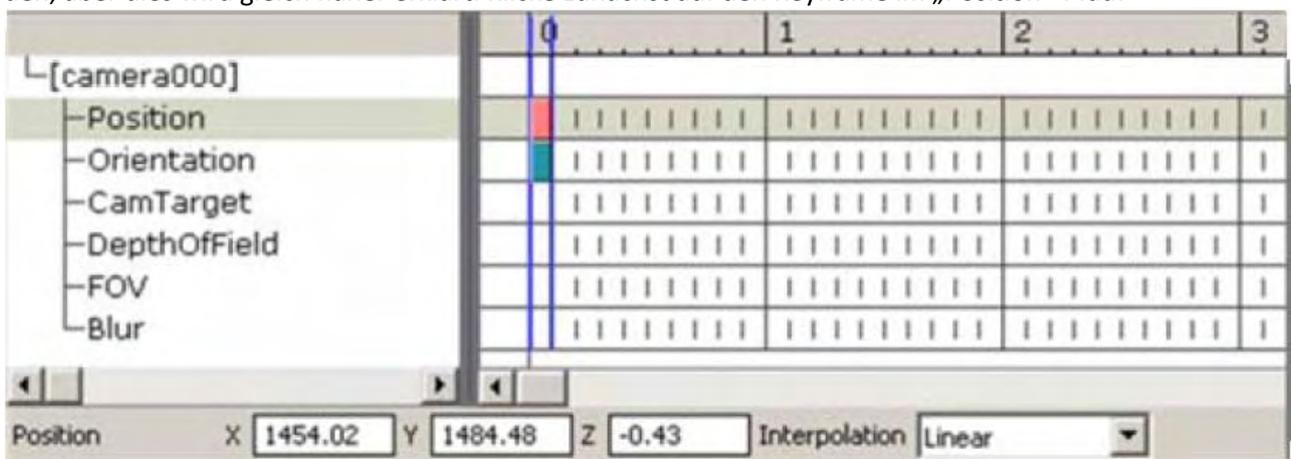
Dies sind Zeitmarkierungen, welche uns die aktuelle Position anzeigen. Die Zeitmarkierungen sind an der Position „0“, was bedeutet, dass die Cutscene gerade erst beginnt. Es gibt auch eine Zeitskala im oberen Teil des Fensters:



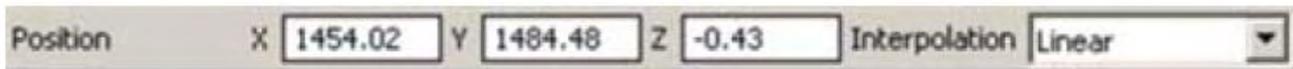
Die Zahlen 0, 1, 2, 3, ... repräsentieren die entsprechenden Sekunden der Sequenz. Jede der 10 vertikalen Kerben einer Sekunde zeigt 0.1 einer Sekunde an. Um die Zeitmarkierung zu bewegen musst du einfach irgendwo auf die Zeitskala klicken. Die kleinen hellblauen Vierecke sind Keyframes:



Da wir die Einstellung „automatisches Hinzufügen von Keyframes“ aktiviert haben, wurden diese Keyframes für uns bereits hinzugefügt. Beachte, dass die Keyframes, in den Pfaden für Position und Orientierung hinzugefügt worden. Normalerweise können Keyframes von Hand hinzugefügt, verschoben oder gelöscht werden, aber dies wird gleich näher erklärt. Klicke zunächst auf den Keyframe im „Position“-Pfad:



Der ausgewählte Keyframe wird rot hervorgehoben. Zusätzlich wurde der zugehörige Pfad in grau hervorgehoben. Beachte außerdem, dass zusätzliche Felder, die mit diesem Pfad in Verbindung stehen am unteren Fensterrand erscheinen:

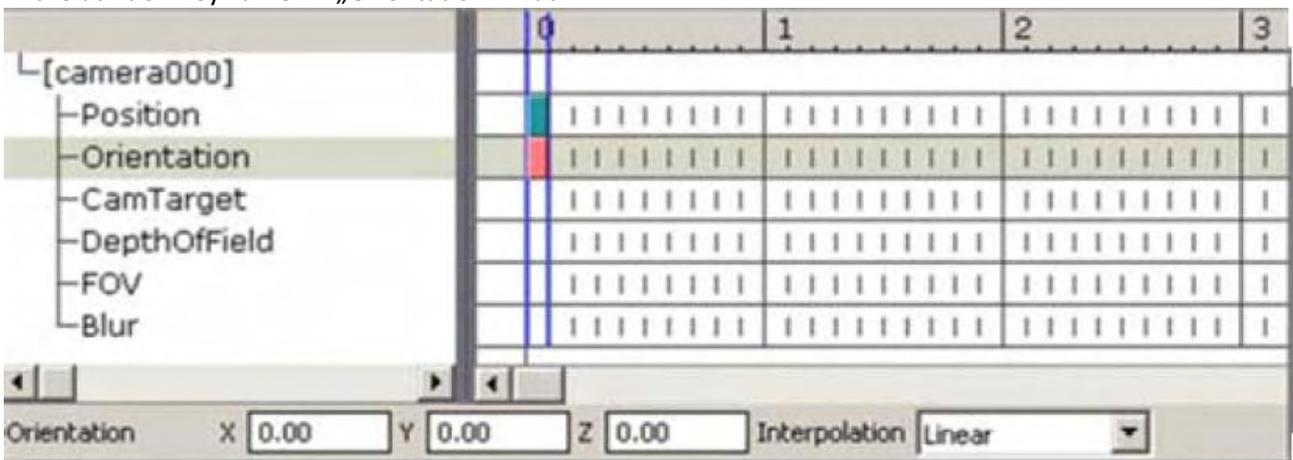


Dies sind die X, Y und Z Positionen von „camera000“. Diese Werte können manuell verändert werden. Dazu musst du einfach in das entsprechende Feld klicken, den gewünschten Wert eingeben und mit der „Enter“-Taste bestätigen.

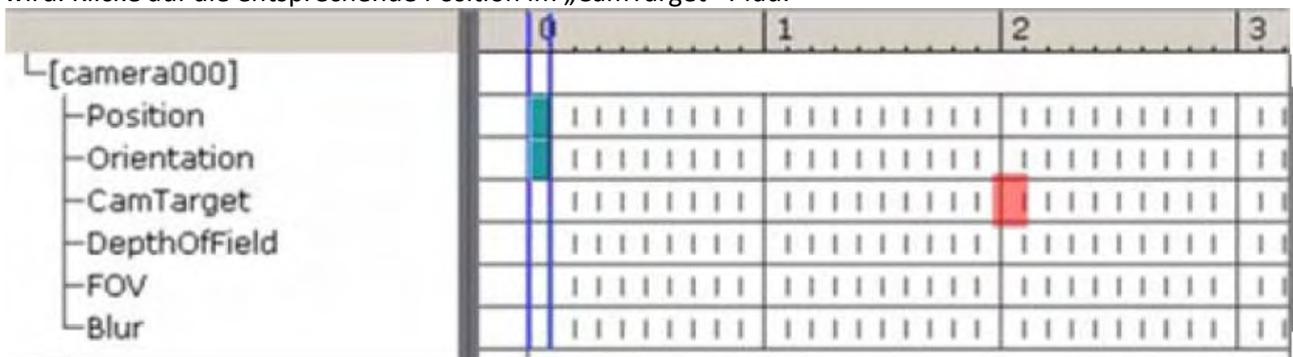
Das Feld mit dem Titel „Interpolation“ ist sehr wichtig. Es bestimmt die Methode, mit der ein Keyframe in den nächsten übergeht. Folgende Werte können hier eingestellt werden:

- „Stepped“ -> Der Durchlauf von einem Keyframe zum nächsten erscheint in Schritten
- „Linear“ -> Der Durchlauf erscheint in einer linearen Art
- „Smooth“ -> Der Durchlauf erscheint sanft.

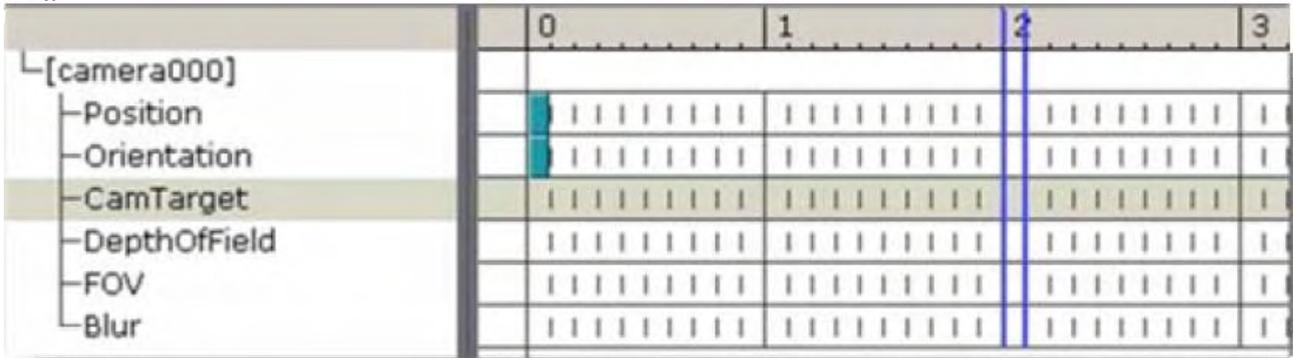
Klicke auf den Keyframe im „Orientation“-Pfad:



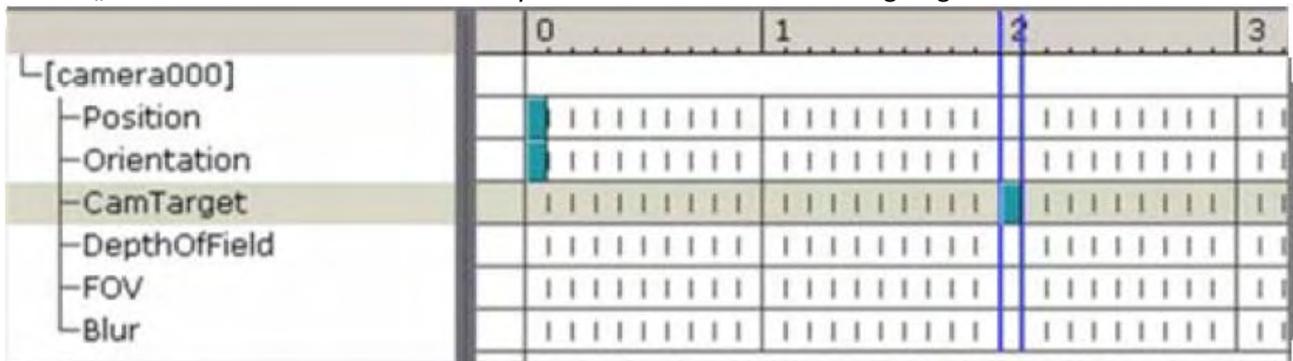
Dies zeigt Einstellungen an, die in Verbindung mit der Ausrichtung des Objekts stehen. Warum nicht von Hand einen Keyframe einfügen? Um das zu demonstrieren wollen wir bei Sekunde 2 einen Keyframe im „CamTarget“-Pfad anlegen. Zur Erinnerung: „CamTarget“ legt den Punkt fest, auf den die Kamera schauen wird. Klicke auf die entsprechende Position im „CamTarget“-Pfad:



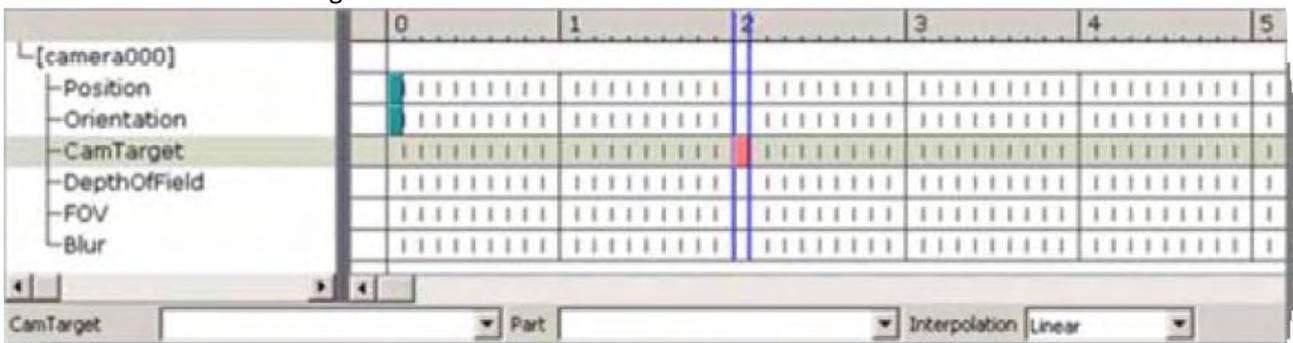
Die Zeitmarkierung wechselt zu der ausgewählten Position und der „CamTarget“-Pfad wird in grau hervorgehoben:



Drücke „EINFG“ auf deiner Tastatur. Ein Keyframe wird in diesem Feld eingefügt.



Natürlich müssen die Einstellungen des Keyframes noch eingestellt werden. Klicke auf den neu erstellten Keyframe. Er wird in rot hervorgehoben, was bedeutet, dass er aktiv ist und die Einstellungen in den Feldern unter diesem Fenster geändert werden können:



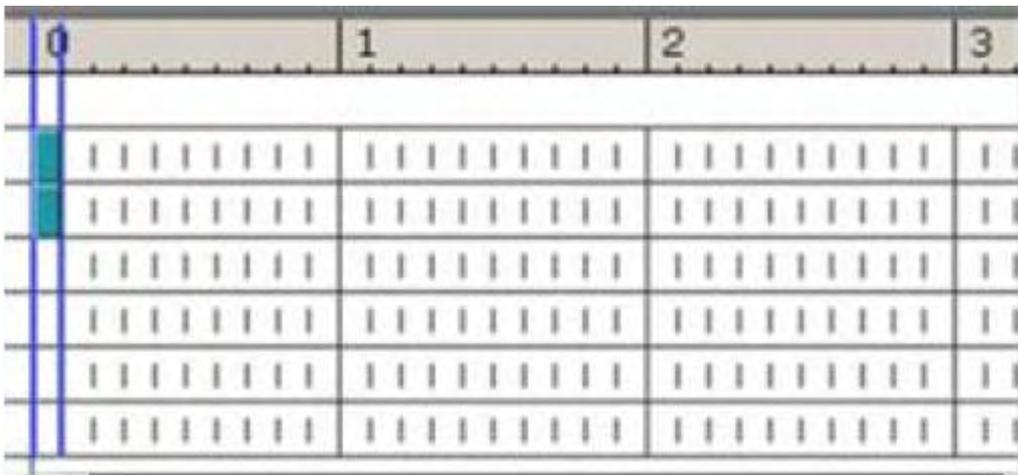
Die „CamTarget“-Liste enthält alle Objekte, die in dieser Cutscene-Area vorhanden sind. Die „Part“-Liste beinhaltet alle individuellen Teile eines bestimmten Objekts, welches in der „CamTarget“-Liste ausgewählt wurde (sofern es welche hat).

Keyframes können entlang ihres jeweiligen Pfades verschoben werden. Um dies zu machen, wähle einen Keyframe aus. Er wird wieder rot hervorgehoben, was anzeigt, dass er aktiv ist. Nun mit der linken Maustaste auf den Keyframe klicken und halten, dann nach links oder rechts verschieben. Wie du siehst, folgt der Keyframe den Bewegungen der Maus. Wenn der Keyframe sich an der gewünschten Stelle befindet, musst du lediglich die linke Maustaste wieder loslassen. Vergiss nicht, dass Keyframes nicht zwischen verschiedenen Pfaden hin und her verschoben werden können.

Bis jetzt wissen wir, wie man Keyframes hinzufügt und bewegt, jetzt lernen wir noch, wie man welche löscht. Um einen Keyframe zu löschen, muss er zuerst ausgewählt werden, damit er aktiv wird (rot hervorgehoben). Klicke auf unseren gerade erst erstellen Keyframe und drücke anschließend die „ENTF“-Taste auf deiner Tastatur. Der Keyframe wurde somit gelöscht. Vorsicht beim Löschen von Keyframes – es gibt keine erneute Bestätigung. Sobald du „ENTF“ drückst, wird der Keyframe ohne Rückfragen gelöscht. Jetzt schauen wir uns einmal die Kamerapositionierung näher an. Momentan befindet sich die Kamera auf dem Boden:



Allerdings wollen wir, dass sie an der Decke hängt. Das kann mit Hilfe von einer der Werkzeuge gemacht werden, aber es gibt auch eine einfachere Methode. Zuerst vergewissere dich, dass die Zeitmarkierung an der Position 0 ist.



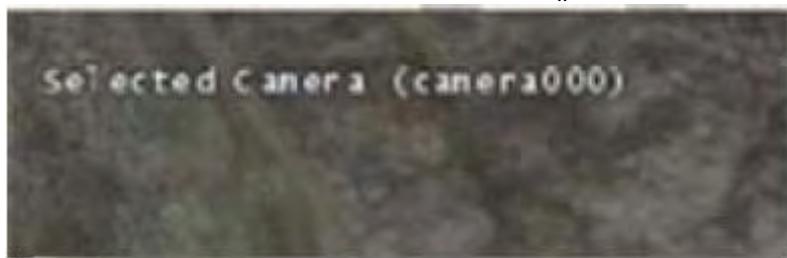
Selbstverständlich muss „camera000“ im „Cutszene Objects“-Fenster ausgewählt werden. Anschließend klickst du mit der rechten Maustaste auf „camera000“ und wählst „Set active camera“ aus dem Menü:



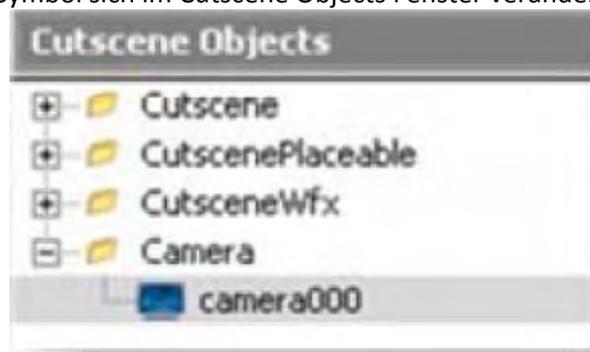
Was ist passiert? Schauen wir mal auf die Area:



Das ist, was die Kamera sieht, im diesen Augenblick bedeutet dass, wenn wir anfangen sie, zu verschieben, bewegt sich die Kamera selbst aktiv mit. Das ist ein sehr nützlicher Aspekt der Kamera, Das bedeutet, dass wir seine Position genau festlegen können und wohin sie zeigen soll. Betrachten wir die obere linke Ecke des Bereichs. Eine Informations Linie sollte über Der Kamera erscheinen die gewählt worden ist. Den Namen der Kamera sieht man in den '()' Klammern:



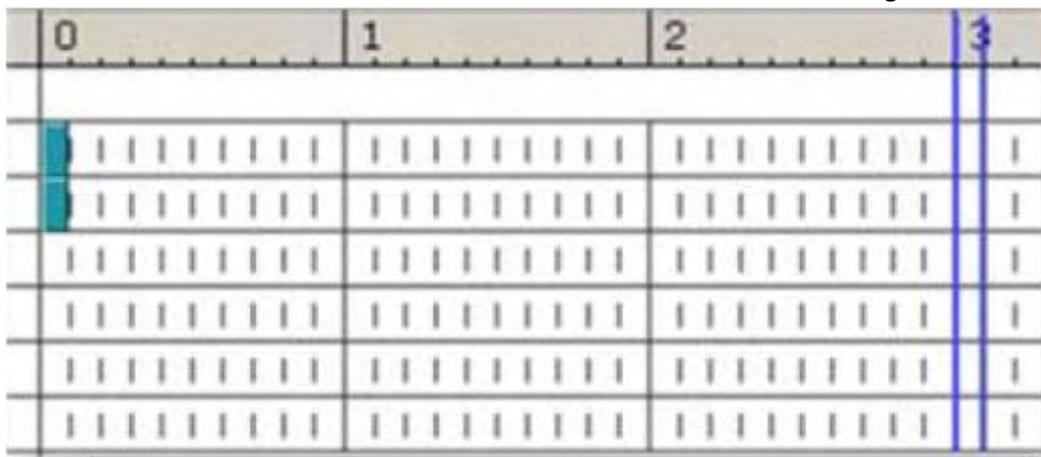
Merke dir das dass Kamera Symbol sich im Cutscene Objects Fenster verändert hat:



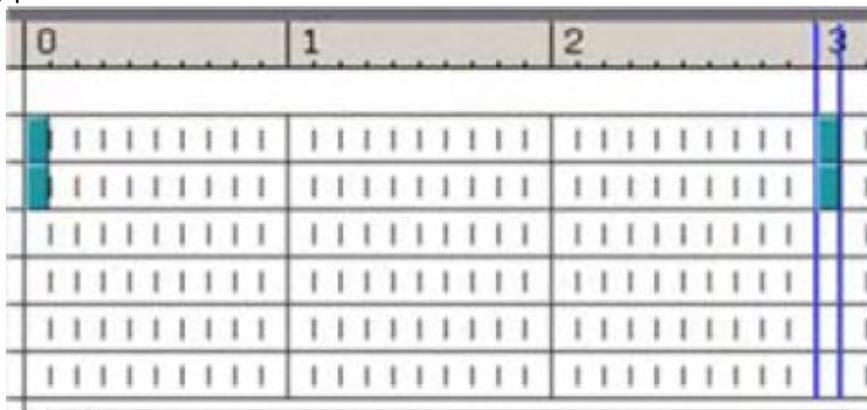
Das neue Symbol bedeutet, dass die ausgewählte Kamera jetzt aktiv ist und im Bereich des Editors verschoben werden kann. Wie verschieben wir die Kamera eigentlich? Auf Die gleiche Weise, wie wir den Bereich verschieben, d.h. verwende die ← ↑ ↓ → und die W A S D Tasten. Klicke und Halte die Tasten um die Kamera zu Rotieren, verwende jetzt die WASD Tasten um die Kamera bis zur Höhlendecke zu bewegen:



Da wir jetzt die Kamera veranlassen, sich entlang eines geradlinigen Pfades 3 Sekunden lang zu bewegen. Stelle wir im Set the time Marker 3 Sekunden ein um es dann in der timescale anzeigen zu lassen:



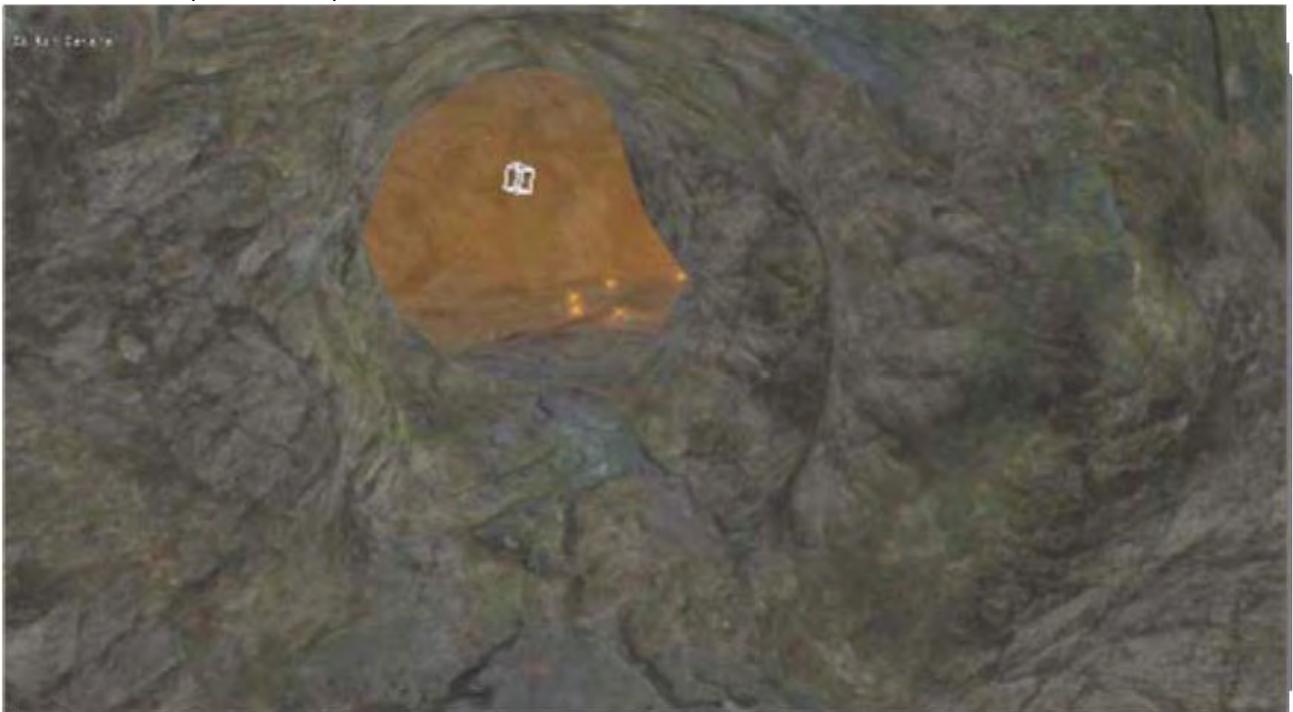
Mit der ↑ Taste, bewegst du die aktive Kamera vorwärts. Beachte, dass was wir für die Bewegungen der Kamera wählen, erscheint ein Keys in der 3. Sekunde der cutscene sowohl auch in den Positions- als auch in den Orientierungspfaden:



Das ist, was passieren soll. Es kommt vor, weil der automatische Key der Funktion hinzugefügt wurde, und damit wird dann ermöglicht. Wenn es unbrauchbar wäre, würden diese Keys nicht automatisch hinzugefügt werden, und uns veranlassen, es manuell hinzuzufügen. Klicke und halte die Tasten, um die Kamera rotieren zu lassen, damit diese Punkte nach unten und nach oben zeigen:



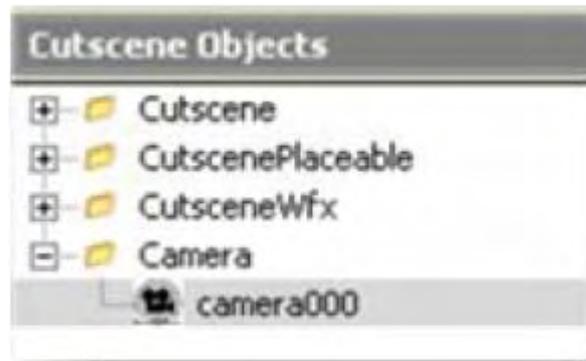
Das ist für die Kamera, wir können sie jetzt deaktivieren. Um das zu tun, Drücke auf den 3 Knopf über der Zehntastatur (NUMPAD 3):



Die Phase der Editor Kamera erscheint jetzt in der oberen linken Ecke:



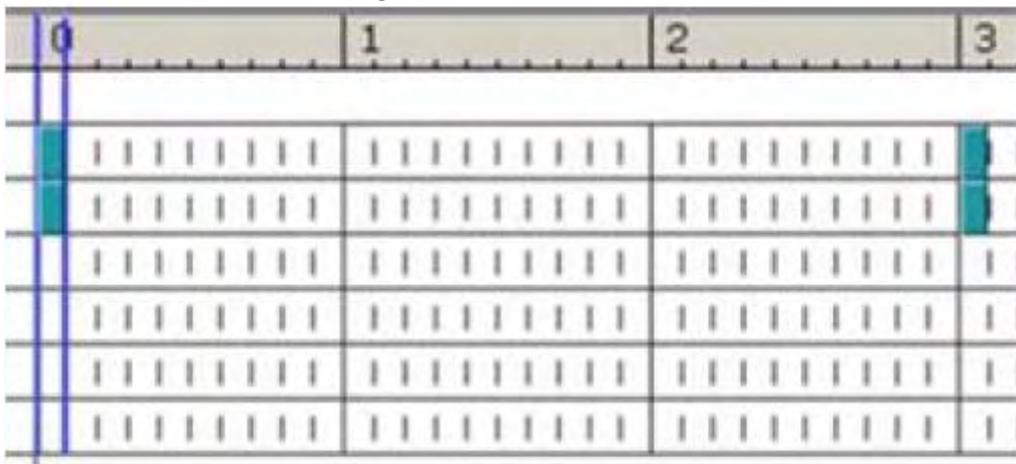
Das heißt, dass die Kamera jetzt zur Standardeditor Kamera gewechselt ist, dass für das Bewegen der Bereiche benutzt wird. Beachte außerdem, dass das Symbol im Cutszene Objekt Fenster sich verändert hat, Und das sich auch das Symbol im Fenster zur Standardkamera verändert hat, Das bedeutet, dass die Kamera nicht mehr aktiv ist:



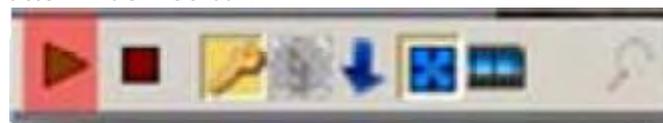
Klicke auf den STOP Button in der Cutszene Leiste:



Das bringt den Timemarker auf den Anfang der Cutszene zurück:



Klicke nun auf den PLAY Button in der Toolbar:



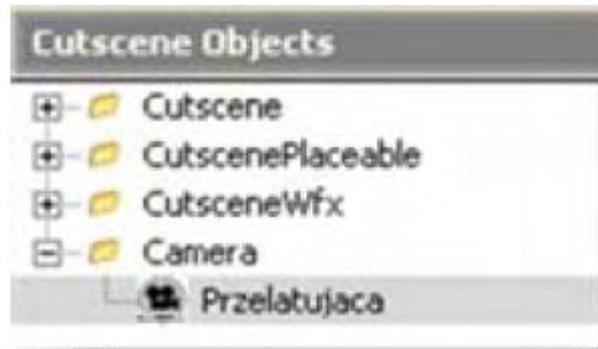
Du solltest jetzt sehen, wie sich die Kamera bewegt, (vergewissere Dich, dass du Die Kamera im Cutszene Objektfenster ausgewählt haben.) Beachte, wie sich der Timemarker auf den Kamerapfad bewegt. Diese Wiederholung kannst du beenden indem Du auf den STOP Button drückst.

Wichtig: Sobald der cutscene Reply Stoppt, wird die camera000 automatisch aktiviert. Das ist jedoch nicht unerwünscht, so ist es am besten, irgendwo auf ein Gebiet zu klicken und dann mit dem NUMPAD 3 die Cutscene zu stoppen. Auf diese Weise verhinderst Du irgendwelche unerwünschten Folgen, denn die camera000, Deaktiviert sich und Die Kamera wird wieder ausgeschaltet. Beachte, dass, wenn Du fortsetzen möchtest, muss man eine Anpassung an der Kamera machen, und es wieder im Cutscene Objects auswählen, da die Kamera nach einem Klick irgendwo auf deinem Gebiet nicht mehr ausgewählt werden kann.

Im Wesentlichen haben wir bereits eine Grundlage für einen Film hergestellt. Das ist noch nicht das Ende den wir müssen immer noch zwei rotierende Kameras hinzufügen. Bevor wir das aber tun, müssen wir die camera000 mit einem richtigen Namen speichern. Rechtsklicke auf die Camera000 im Cutscenen Fenster, Rechtsklicke und wähle „Rename“ aus dem folgenden Menü:



Schreibe „Moving toward“ und drücke Enter. Dieser Name der Kamera wurde jetzt geändert:



Vor dem Hinzufügen der zweiten Kamera, wollen wir uns ein wenig weg von der ersten Kamera bewegen, so dass sich diese zwei nicht überlappen. An dem Screenshots unter dir kann dieser Punkt angewählt werden, wie dargestellt im Handbuch genauso auch das erste Bewegen an der Position von der 3. Sekunde in der Cutscene zu sehen ist:



Klicke jetzt auf das Werkzeug, um eine zusätzliche Kamera hinzuzufügen:



Klicke jetzt auf eine Position im Gebiet, das im Grunde die beste Wahl ist. Eine zweite Kamera wird jetzt zum Gebiet hinzugefügt. Klicke darauf im Cutscene Objects Fenster. Natürlich ist die Kamera standartmäßig Camera000 genannt worden:



Die neue Kamera wird jetzt in diesem Gebiet besonders hervorgehoben:



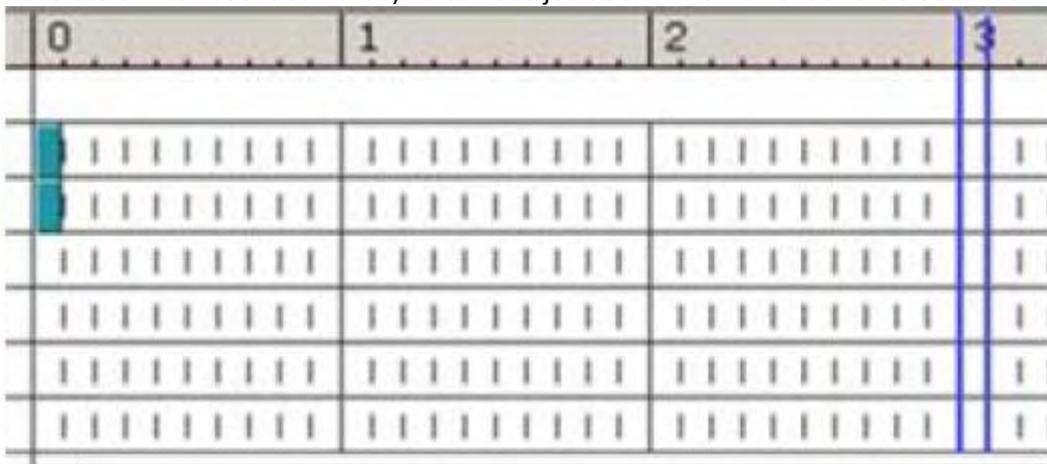
Wollen wir den Namen dieser Kamera sofort ändern. Rechtsklicke auf camera000 im Cutscenen Fenster und wähle „Rename“ im Menü:



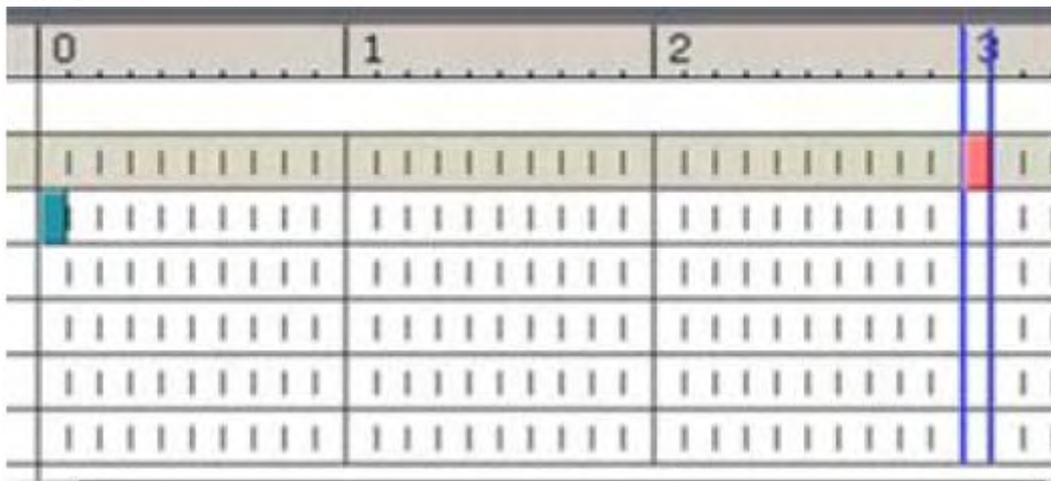
Schreibe Orbiting und Drücke Enter .Der Name der Kamera wird jetzt geändert:



Die Orbiting(Umkreisend) Kamera sollte seinen Durchlauf des Bereichs ab der 3 Sekunde beginnen(d.h, nachdem der erste umlauf beendet wurde) verschiebe jetzt den Timemarker auf die 3 Sekunde:



Beachte dass die ersten schlüssel auf den Positions- Und Orentierungspfaden auf der Sekunde 0 eingestellt ist. Wir müssen das Korrigieren und lassen jetzt die zwei Schlüssel zu der 3 Sekunde gleiten. Klicke und halte den ersten Schlüssel auf dem Positionspfad fest(er wird jetzt rot angezeigt) da wir ihn ausgewählt haben jetzt gehen wir zur 3 Position:



Du kannst die schlüsseln auch ohne eine Unterbrechung bewegen. Um das zu tun, Klicke auf den Schlüssel. Den du bewegen möchtest und halte CTRL.

Benutze die W A S D Tasten, und bewege die Kamera unter die Decke. Rechtsklicke und halte die Taste, Um die Kamera rotieren zu lassen, damit es Punkte unten aufweist:



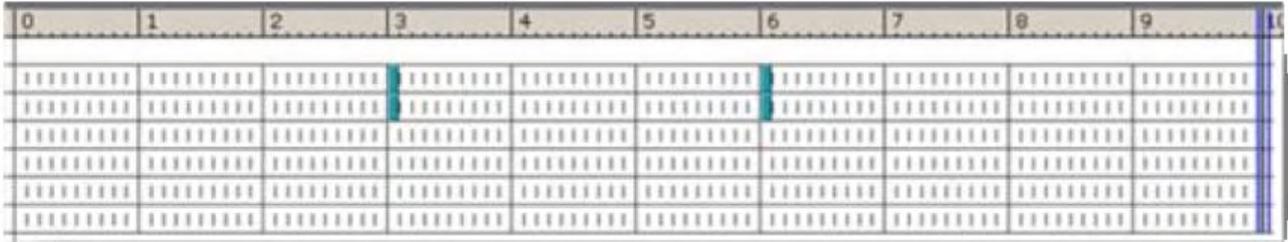
Bewege den Timemarker auf die 6 Sekunde (d.h eine halbe Bewegung der Orbiting Kamera):

0	1	2	3	4	5	6

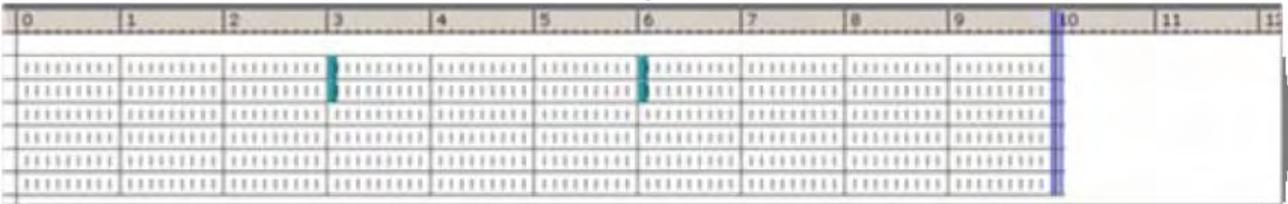
Benutze die ← ↑ ↓ → Tasten, und halte die rechte Maustaste gedrückt, das lässt die Kamera um ungefähr 90 ° rotieren:



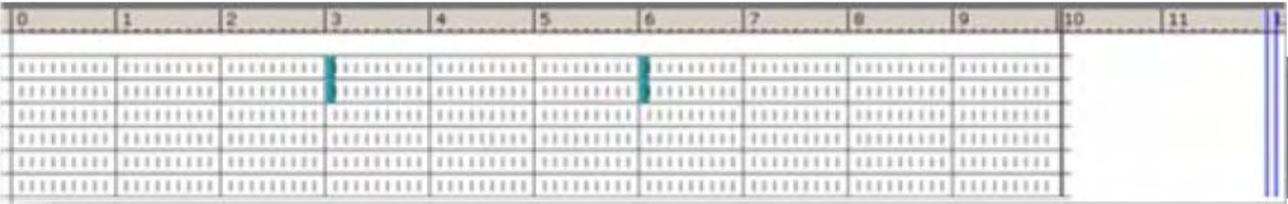
Bewege den Time-Marker auf die 10 Sekunde:



Mit dem Mausrad kannst du Heranzoomen oder auf die Zeitskala schauen, so dass du eine größere Ansicht hast(wenn du gezoomt hast). Natürlich muss das Fenster mit einer Zeitskala Aktiv sein, um auch im Stande zu sein das zu tun. Um das Zeitskala-Fenster zu Aktivieren, Klicke auf den time marker. Beachte, das diese Zeitskala nur zum 10 vom zweiten geht:

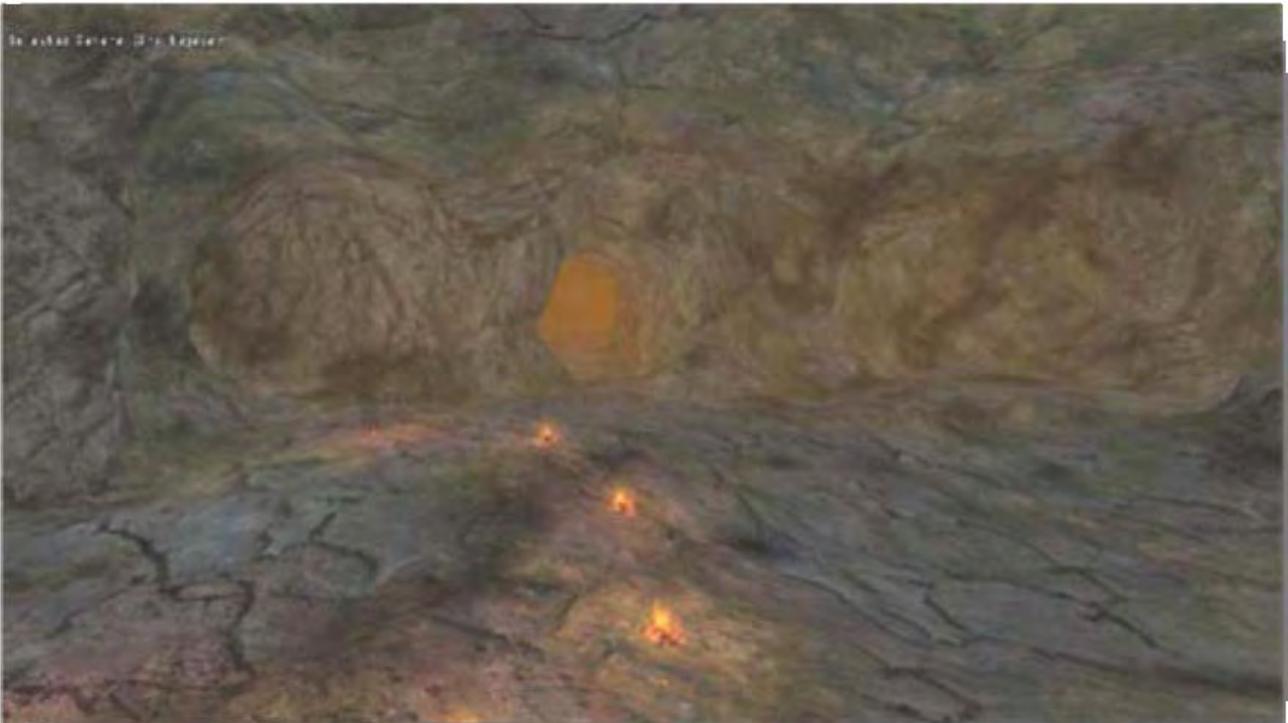


Klicke auf irgendeinen Pfad, z. B. *Position*, und drücke den END-Knopf. Die Zeitskala wird jetzt auf 12 Sekunden erweitert:

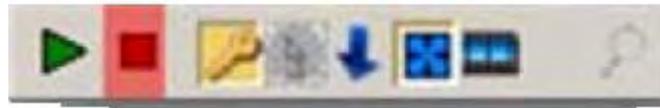


Wir brauchen aber nur 10 Sekunden. Bewege den Zeitmarker zur Sekunde 10 und drücke wieder auf END.

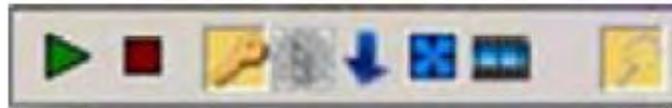
Zurück zu unserer Kamera. Rotiere im Area Editor die Kamera um 90° nach links, indem Du die Pfeiltasten (← ↑ ↓ →) verwendest und dabei die rechte Maustaste gedrückt hältst:



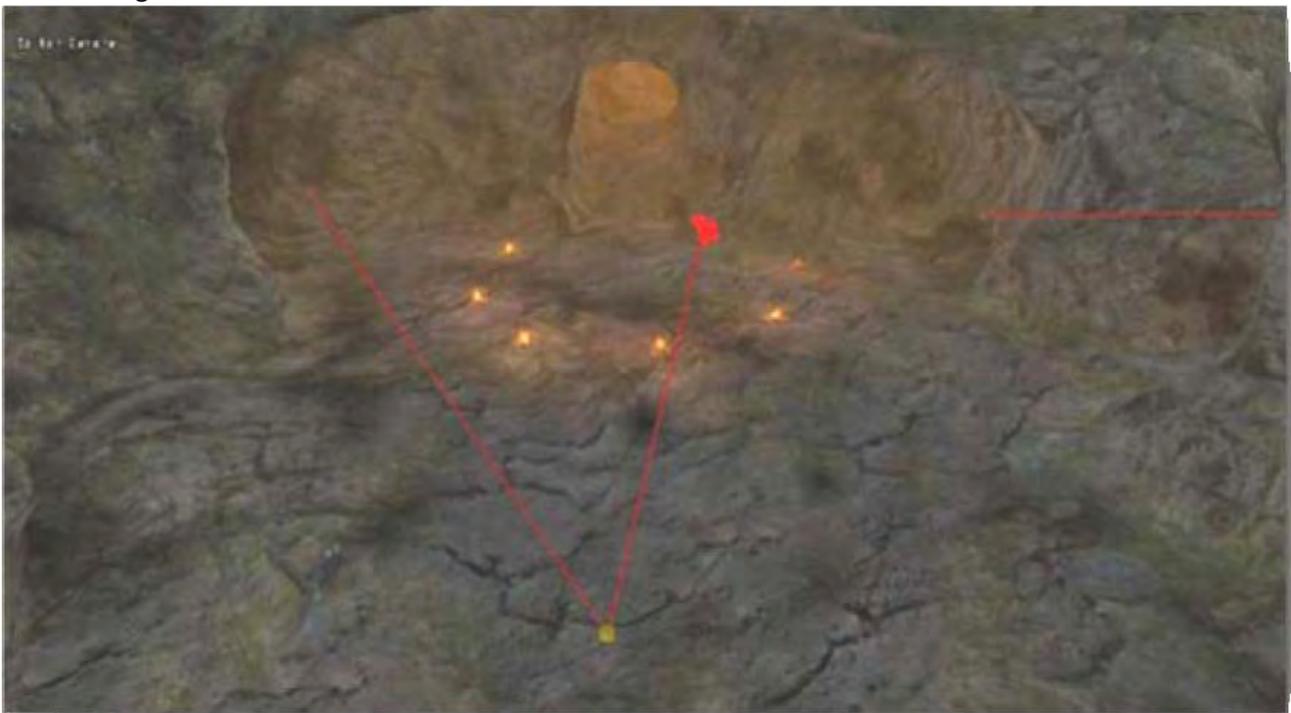
Die Kamera wird jetzt innerhalb von 7 Sekunden eine Drehung um 180° ausführen. Jetzt können wir die Kamera deaktivieren. Drücke dazu die 3 auf dem Zehnerblock der Tastatur (NUMPAD 3). Bewege den Zeitmarker auf 0, indem Du auf den STOP-Button in der Cutscene-Werkzeugleiste klickst:



Lass Dir jetzt die Hilfsobjekte anzeigen, indem Du auf das Vergrößerungsglas in der Cutscene-Werkzeugleiste klickst:

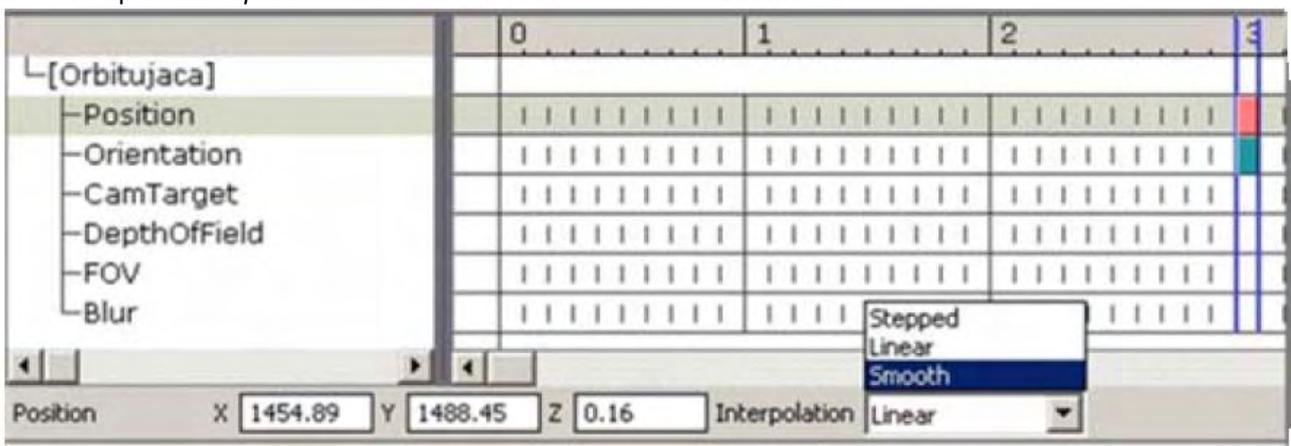


Beide Kameras werden jetzt angezeigt. Die Bewegungen der Kameras werden als Linien dargestellt, die Frames als gelbe Punkte:

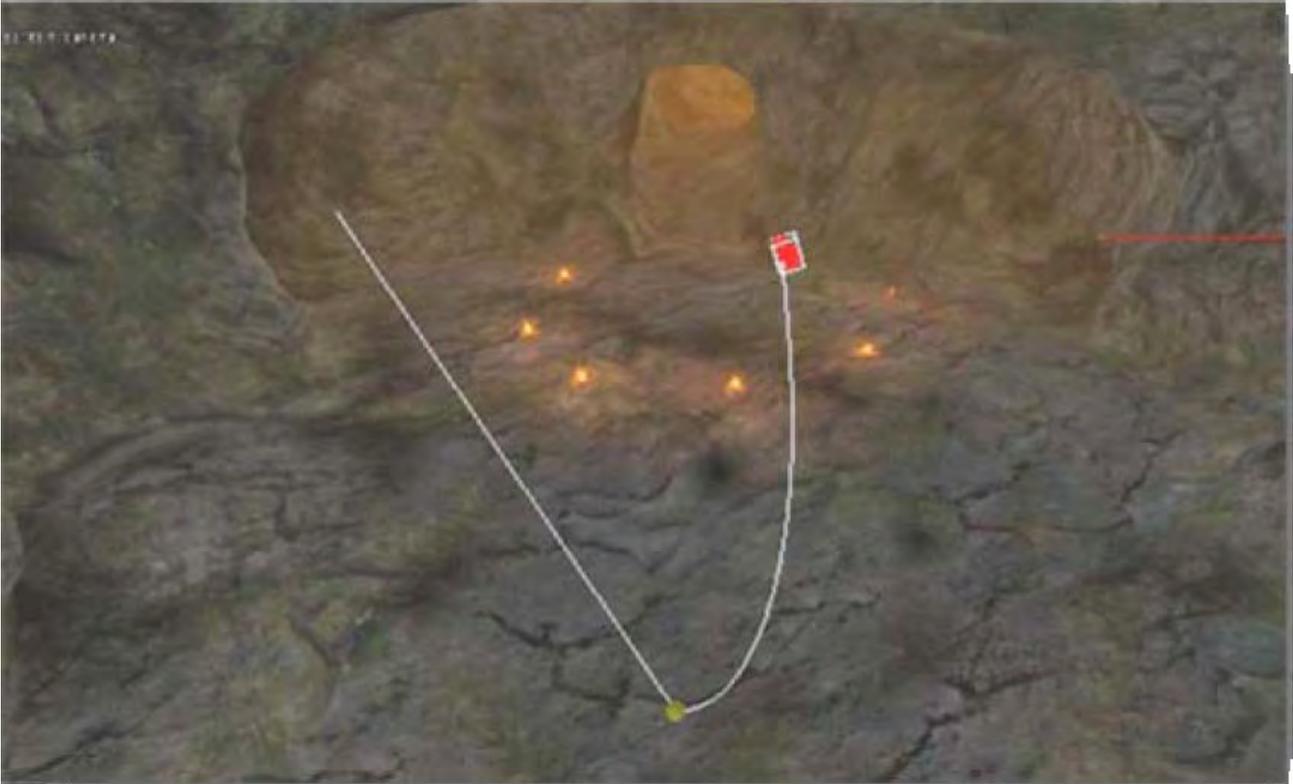


Wir sehen, dass zwar die erste Kamera richtig funktioniert, die Orbitalkamera aber nicht. Sie sollte sich entlang einer Kurve um 180° bewegen, was offenbar nicht der Fall ist. Das bringen wir jetzt in Ordnung.

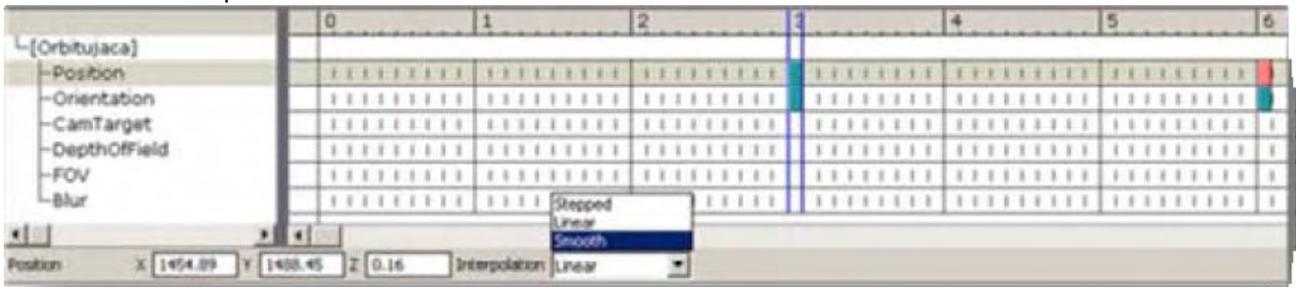
Ändere zuerst die Interpolation der Kamerabewegung. Klicke im Fenster *Cutscene Objects* auf die Orbitalkamera und wähle im Pfad *Position* den ersten Key. Der Key wird nun rot dargestellt, da er aktiv ist. Ändere nun die Option *Interpolation* am unteren Rand des Fensters zu *smooth*:



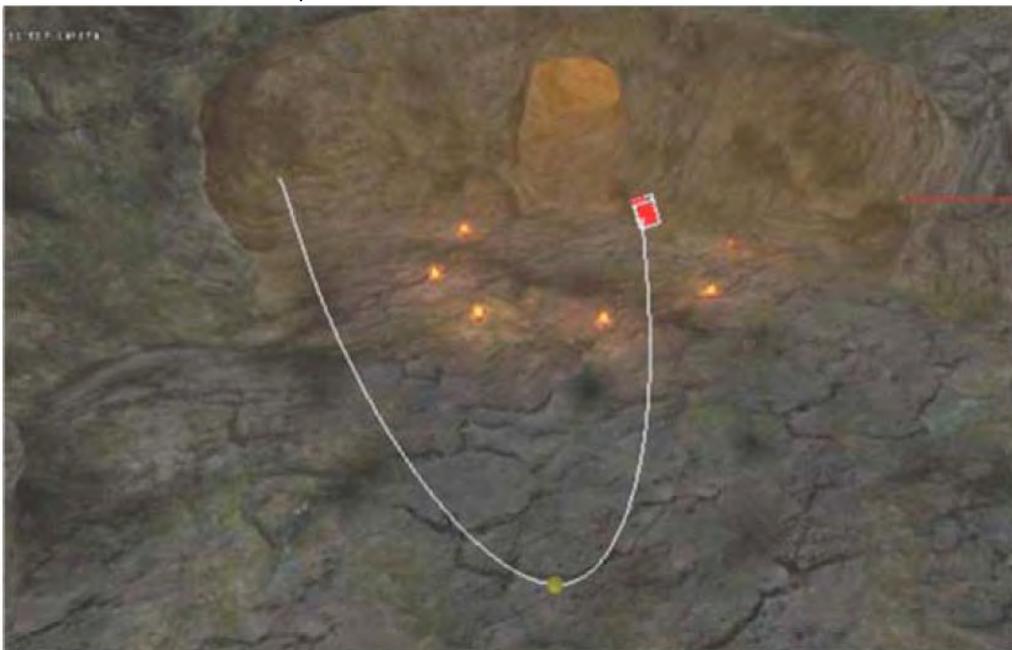
Schau Dir jetzt die Bewegung der Kamera an. Sie beschreibt nun eine Kurve:



Genau das wollten wir, also wählst Du nun den zweiten Key auf dem Pfad *Position* (Sekunde 6) und änderst auch hier die Interpolation zu *smooth*:



Wenn Du Dir jetzt noch einmal die Kamerabewegung anschaust, ist sie zwar immer noch keine perfekte Kurve, aber immerhin besser als das, was wir vorher hatten:



Klicke jetzt im Area Editor auf den gelben Punkt, der vom Anfang und vom Ende des Bogens am weitesten entfernt ist. Der Punkt ist nun selektiert, was durch einen weißen Rahmen angezeigt wird:



Damit ist der zweite Key auf dem Pfad *Position* der Orbitalkamera selektiert. Wähle nun das Objekt-Bewegungswerkzeug (das zweite Symbol von links) in der Cutscene Editor-Werkzeuggestreife aus:



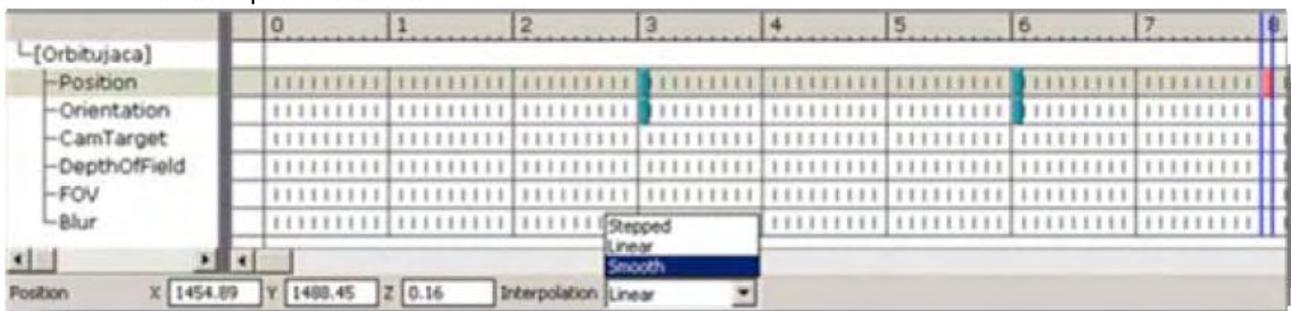
Grundsätzlich funktioniert das Bewegen von Objekten mit diesem Werkzeug genauso wie im normalen Area Editor. Aus dem ausgewählten Objekt ragen nun drei verschiedenfarbige Linien heraus. Mit diesen Linien kannst Du das Objekt entlang der X-Achse (rote Linie), Y-Achse (grüne Linie) oder Z-Achse (blaue Linie) bewegen:



Achte darauf, dass der Pfad *Position* selektiert ist, da wir hier den neuen Key einfügen werden. Du siehst, dass sich die Kamera auf die entsprechende Position bewegt hat:



Drücke den INSERT-Knopf. Auf dem Pfad *Position* wird ein neuer Key hinzugefügt. Selektiere den neuen Key und ändere die Interpolation zu *smooth*:



Klicke auf dem STOP-Knopf in der Cutscene-Werkzeuggestreife:



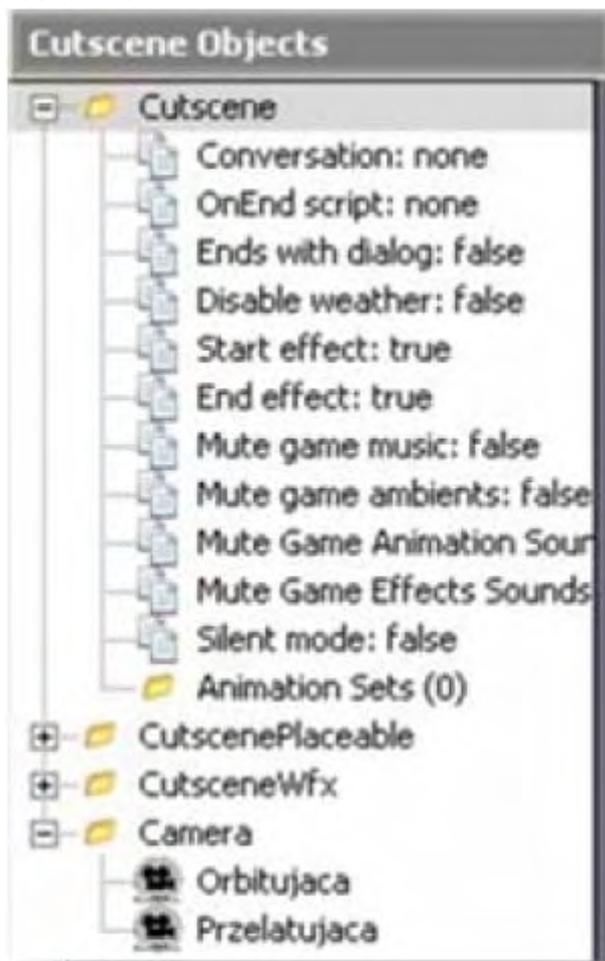
Der Zeitmarker wird auf 0 gesetzt, also auf den Anfang der Szene. Wähle im Area Editor den neu eingefügten gelben Punkt aus:



Bewege den Punkt mit dem Objekt-Bewegungswerkzeug ein wenig nach links, so dass die rote Linie eine schöne 180°-Kurve ergibt:



Diese Änderungen genügen fürs Erste, aber wir müssen die Bewegungen der beiden Kameras noch festlegen. Was wir zum Schluss haben wollen, ist Folgendes: zum Anfang der Szene soll die Ansicht der geraden Kamera folgen, und dann in die Orbitalperspektive der zweiten Kamera übergehen. Um den Hauptpfad für die Szene zu bearbeiten, klicke auf dem Baum *Cutscene* im Fenster *Cutscene Objects* und klappe ihn auf. Jetzt sind die Hauptoptionen der Szene sichtbar:



- **Conversation.** Enthält den Namen der Dialogdatei, die während der Szene abgespielt wird.
- **OnEnd script.** Enthält den Namen des Skripts, das am Ende der Szene ausgeführt wird.
- **Ends with dialog.** Diese Option bestimmt, ob am Ende der Szene ein Dialog abgespielt wird. Dieses Attribut kann folgende Werte annehmen:
 - *False* – Am Ende der Szene wird kein Dialog abgespielt.
 - *True* – Am Ende der Szene wird ein Dialog abgespielt.
- **Disable weather.** Legt fest, ob während der Szene Wettereffekte (z. B. Sturm, Regen, usw.) abgeschaltet werden. Dieses Attribut kann folgende Werte annehmen:
 - *False* – keine Wettereffekte
 - *True* – Wettereffekte bleiben eingeschaltet
- **Start effect.** Bestimmt, ob während der Szene das Seitenverhältnis des Bildschirms auf 16:9 geändert wird. Dieses Attribut kann folgende Werte annehmen:
 - *False* – Das Seitenverhältnis wird auf 16:9 geändert.
 - *True* – Das Seitenverhältnis bleibt unverändert.

• **End effect.** Bestimmt, ob am Ende der Szene das Seitenverhältnis des Bildschirms wieder zurückgesetzt wird. Dieses Attribut kann folgende Werte annehmen:

- *False* – Das Seitenverhältnis bleibt unverändert.
- *True* – Das Seitenverhältnis wird zurückgesetzt.

• **Mute game music.** Bestimmt, ob die Hintergrundmusik während der Szene ausgeblendet wird. Dieses Attribut kann folgende Werte annehmen:

- *False* – Die Musik wird nicht ausgeblendet.
- *True* – Die Musik wird ausgeblendet.

• **Mute game ambients.** Bestimmt, ob die Hintergrundgeräusche während der Szene ausgeblendet werden. Dieses Attribut kann folgende Werte annehmen:

- *False* – Die Hintergrundgeräusche werden nicht ausgeblendet.
- *True* – Die Hintergrundgeräusche werden ausgeblendet.

• **Mute Game Animation Sounds.** Bestimmt, ob die Geräusche bei Animationen während der Szene ausgeblendet werden. Dieses Attribut kann folgende Werte annehmen:

- *False* – Die Geräusche werden nicht ausgeblendet.
- *True* – Die Geräusche werden ausgeblendet.

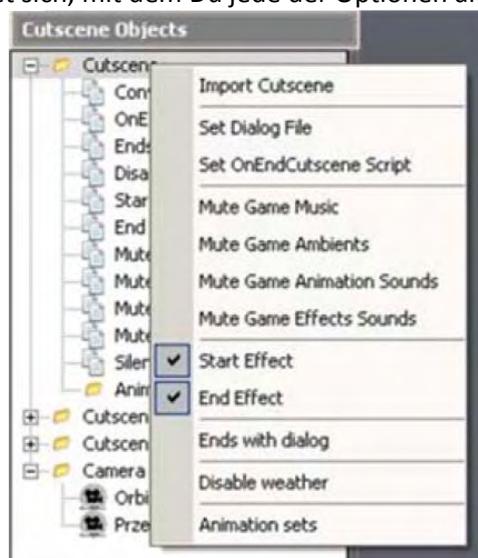
• **Mute Game Effects Sounds.** Bestimmt, ob Geräuscheffekte während der Szene ausgeblendet werden. Dieses Attribut kann folgende Werte annehmen:

- *False* – Die Geräuscheffekte werden nicht ausgeblendet.
- *True* – Die Geräuscheffekte werden ausgeblendet.

• **Silent mode.** Legt fest, ob während der Szene vollständige Stille herrschen soll. Dieses Attribut kann folgende Werte annehmen:

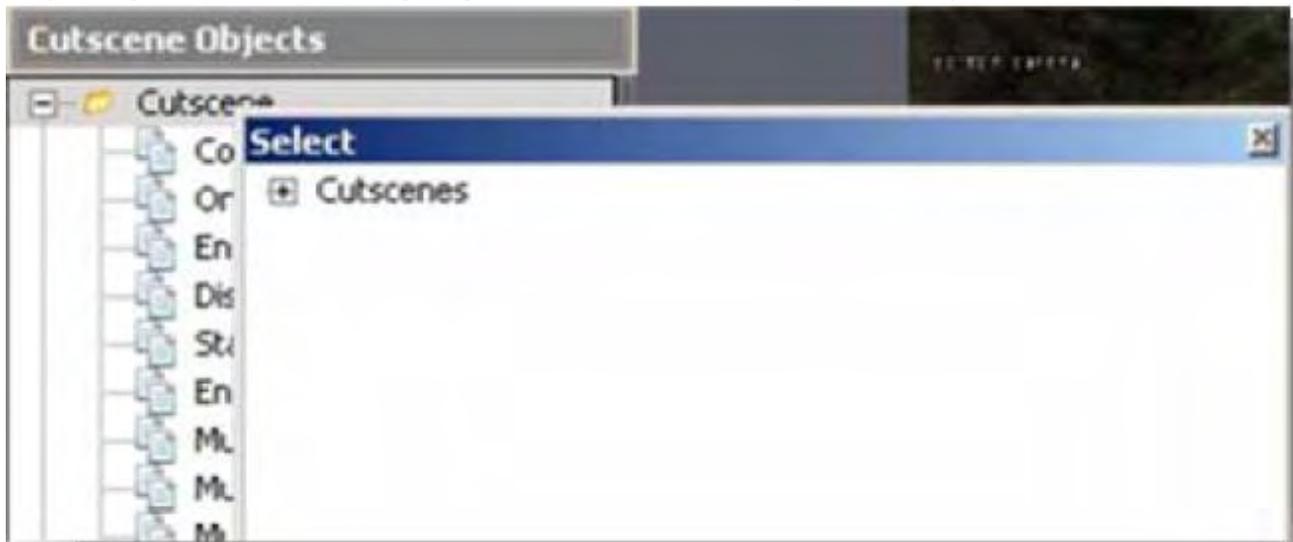
- *False* – Musik und Geräusche werden nicht ausgeblendet.
- *True* – Jegliche Musik und Geräusche werden ausgeblendet.

Um eine der Optionen zu ändern, klicke mit der rechten Maustaste auf den Baum *Cutscene* im Fenster *Cutscene Objects*. Ein Menü öffnet sich, mit dem Du jede der Optionen ändern kannst:



Der Befehl *Import Cutscene* ist selbsterklärend - damit importierst Du eine Szene.

Die Befehle *Import Cutscene*, *Set Dialog File* und *Set OnEndCutscene Script* öffnen Auswahlfenster, in denen die jeweiligen Baumansichten angezeigt werden (*Cutscenes*, *Dialogues* und *Scripts*).



Die übrigen Menübefehle funktionieren als Schalter mit den Einstellungen *true* und *false*.
Jetzt schauen wir uns die verschiedenen Typen von Pfaden an. Es gibt insgesamt 14:



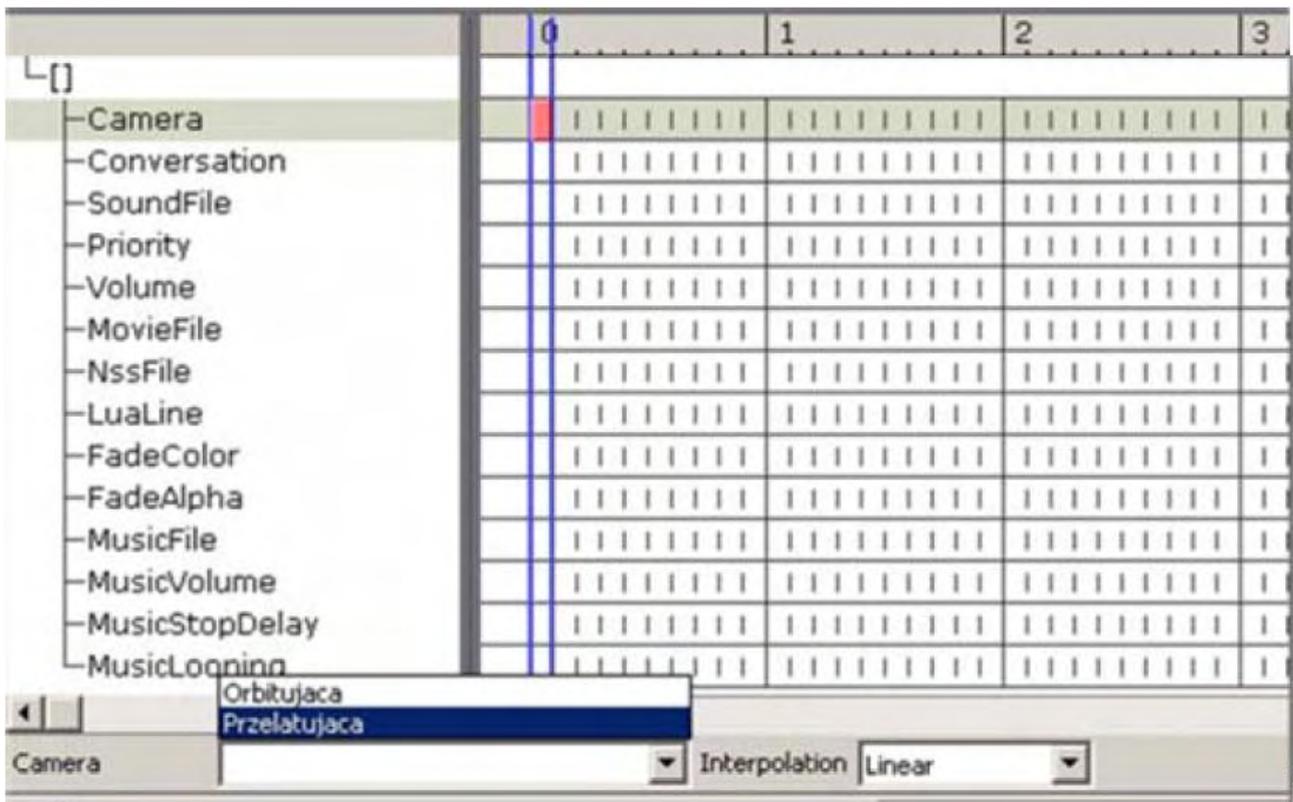
- *Camera* – Dieser Pfad enthält die Kameras. Hier werden wir unsere beiden Kameras einfügen.
- *Conversation* – Dieser Pfad enthält Dialoge.
- *SoundFile* – Dieser Pfad enthält Geräusche.
- *Priority* – Dieser Pfad enthält priority sounds.

- *Volume* – Dieser Pfad enthält die Geräuschlautstärke.
- *MovieFile* – Dieser Pfad enthält Szenen im Format .bik
- *NssFile* – Dieser Pfad enthält Neverwinter Nights Sprachskripts.
- *LuaLine* – Dieser Pfad enthält LUA Skripts,
- *FadeColor* – Dieser Pfad enthält color fading,
- *MusicFile* – Dieser Pfad enthält die Musik.
- *MusicVolume* - Dieser Pfad enthält die Musiklautstärke.
- *MusicStopDelay* - Dieser Pfad enthält die Verzögerung, mit der die Musik gestoppt wird.
- *MusicLooping* Dieser Pfad enthält Informationen, die benötigt werden, um Musik in einer Endlosschleife abzuspielen.

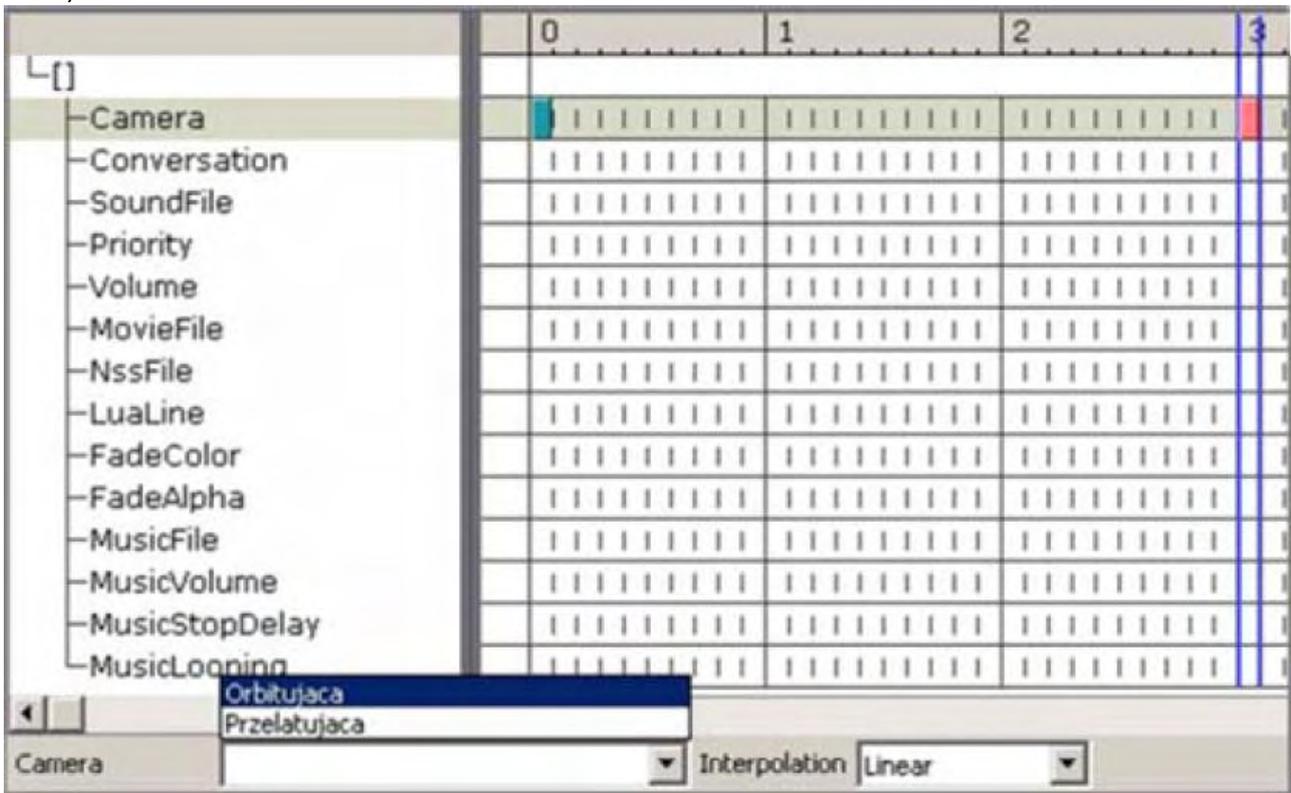
Grundsätzlich funktionieren die Szenenpfade genau wie die Kamerapfade. Setze den Zeitmarker auf 0, am Besten, indem Du auf den Knopf STOP in der Cutscene-Werkzeugleiste klickst:



Klicke jetzt im Pfad *Camera* auf 0 Sekunden und drücke den Knopf INSERT. Ein Key wird hinzugefügt. Klicke darauf. Der Key ist jetzt rot, da er aktiv ist. Wähle unten auf dem Bildschirm unsere erste Kamera aus der Liste aus:



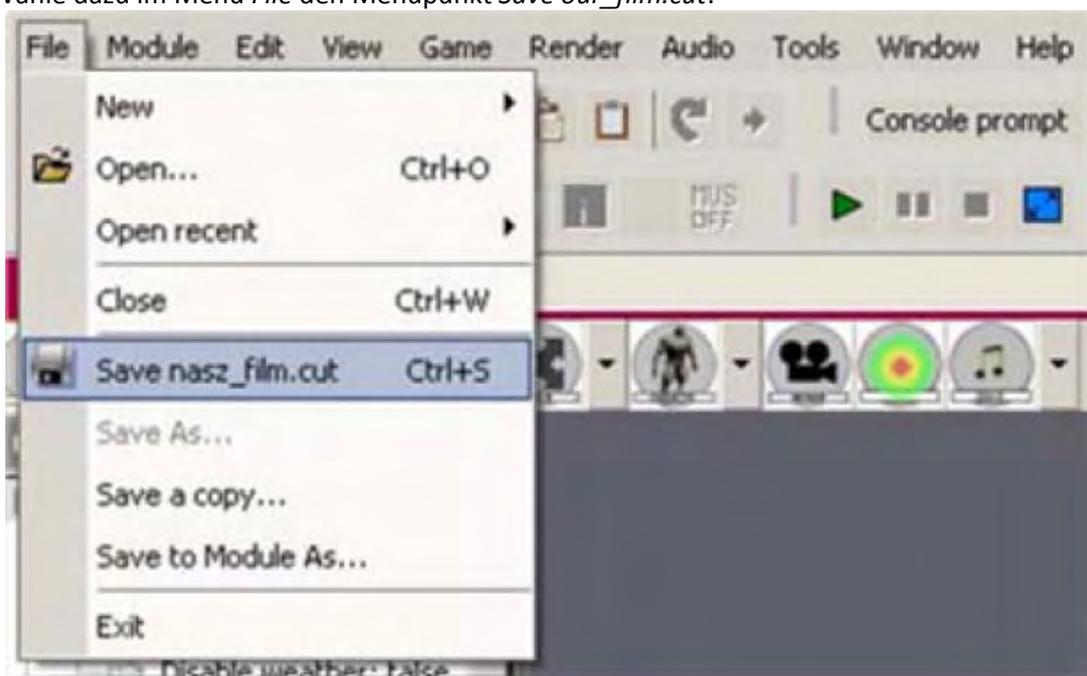
Da die Bewegung dieser Kamera 3 Sekunden dauern soll, verschiebe den Zeitmarker auf die Sekunde 3. Klicke auf dem Pfad *Camera* auf die Sekunde 3 und drücke den INSERT-Knopf. Ein zusätzlicher Key wird hinzugefügt. Klicke auf den neuen Key und wähle dann aus der Liste die zweite Kamera (d. h. die Orbitalkamera) aus:



Das wars - wir haben gerade unsere erste Cutscene erstellt. Es ist eine sehr einfache Szene mit nur zwei Kamerabewegungen, aber sie genügt, um die Grundlagen der Erstellung von Cutscenes mit D'jinni zu erläutern. Jetzt kannst Du die ganze Szene abspielen, indem Du in der Cutscene-Werkzeugleiste auf PLAY klickst:



Das Ergebnis unserer Arbeit kannst Du im Area Editor betrachten. Dazu musst Du die Szene nur noch speichern. Wähle dazu im Menü *File* den Menüpunkt *Save our_film.cut*:



Wenn dieser Menüpunkt nicht im Menü erscheint, oder wenn statt *our_film.cut* ein anderer Dateiname erscheint, dann ist das Fenster *our_film.cut* gerade nicht aktiv. Klicke auf die Titelleiste des Fensters, um es zu aktivieren, und wähle dann den Menüpunkt *Save our_film.cut* aus. Das Fenster *Save File* öffnet sich:



Beim Speichern der Datei kannst Du wählen, ob Du die Datei global speichern willst, damit sie auch in anderen Modulen zur Verfügung steht, oder ob Du sie nur im aktuellen Modul speichern willst. Weil wir uns bisher für die erste Möglichkeit entschieden haben, klicke auf *Global*.

Das Windows-Standardfenster zum Speichern von Dateien öffnet sich. Gehe zum Ordner *Data\Cutscenes*, in dem alle Cutscenes gespeichert werden, und gib *our_film* ein, falls der Name nicht automatisch erschienen ist. Klicke auf *Save*. Du kannst die Szene später manuell zu einem Modul hinzufügen. Wie das funktioniert, ist in den früheren Kapiteln erklärt. Diese Möglichkeit ist sehr praktisch, wenn Du Deine Module z. B. ins Internet hochladen willst, um sie Freunden oder der Community zugänglich zu machen.

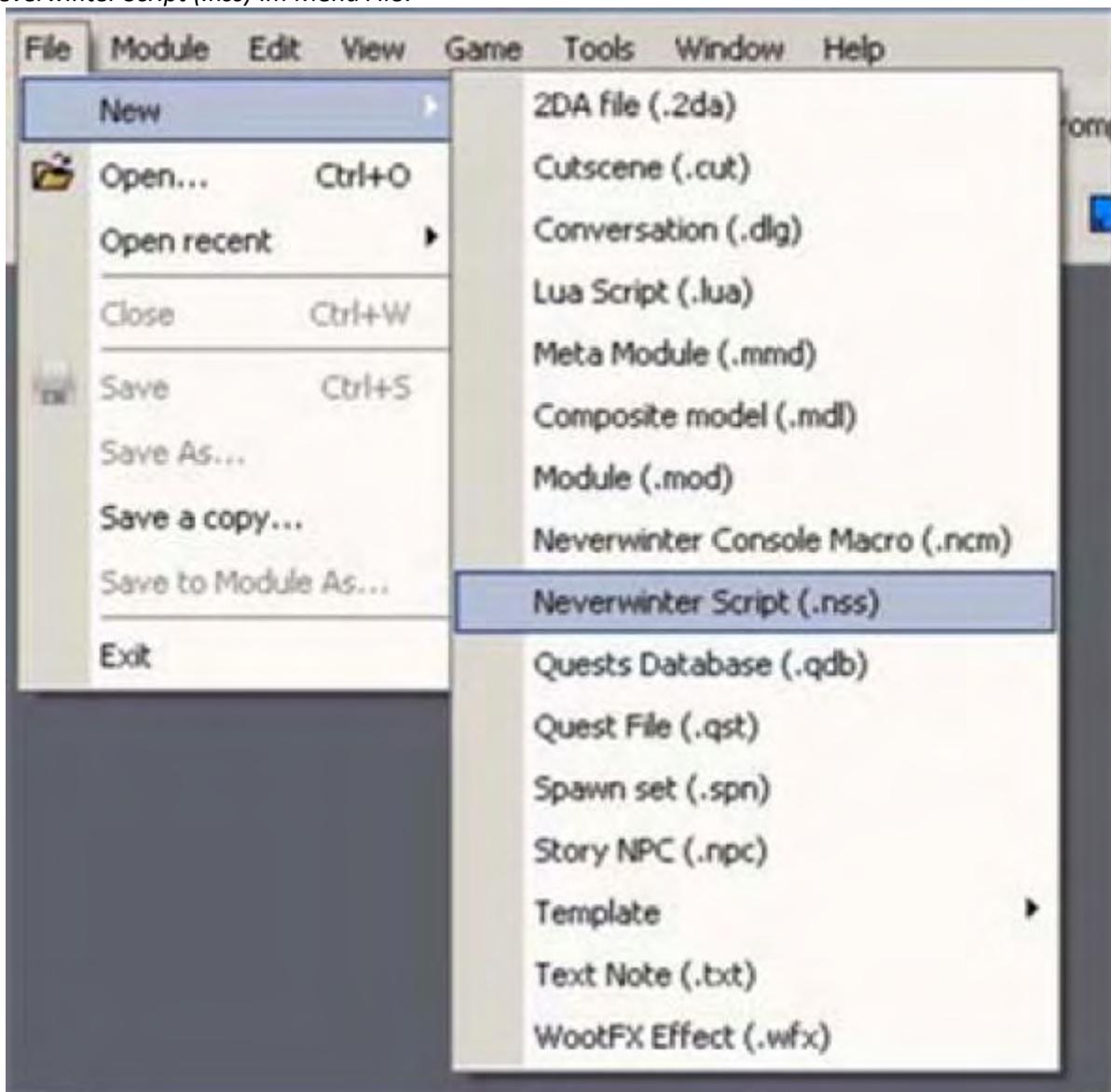
Der Film ist jetzt fertig, aber wir müssen ihn noch in unserem Modul laden. Dafür brauchen wir Trigger und ein Script, und darum geht es im nächsten Kapitel.

Einführung ins Scripting

Einleitung:

Dieses Kapitel befasst sich mit Scripts. Ideal ist dieses Kapitel für diejenigen, die bereits Skriptsprachen kennen (Java, C ++, C #), oder die schon einmal mit Neverwinter Nights Scripts zu tun hatten. Es handelt sich hier nur um eine Einführung in das Scripting, weil eine ausführlichere Beschäftigung mit diesem umfangreichen Thema dieses Handbuch leicht um 200 Seiten umfangreicher machen könnte. Ich werde hier nur einige Beispielskripts behandeln.

"The Witcher" verwendet zwei Skriptsprachen: LUA und Neverwinter Nights Scripts. Wir werden uns in diesem Kapitel mit letzterem beschäftigen. Diese Skriptsprache wird auch in Neverwinter Nights verwendet. Die Scripts können natürlich in D'Jinni geschrieben werden. D'Jinni verfügt über einen eingebauten Scripteditor, der drei Hauptfunktionen mitbringt: Syntaxhervorhebung, Zeilennummern und Hilfen zur Scriptsyntax. Es gibt zwei Wege, den Scripteditor zu starten. Die erste Möglichkeit ist der Menüpunkt *New -> Neverwinter Script (.nss)* im Menü *File*:

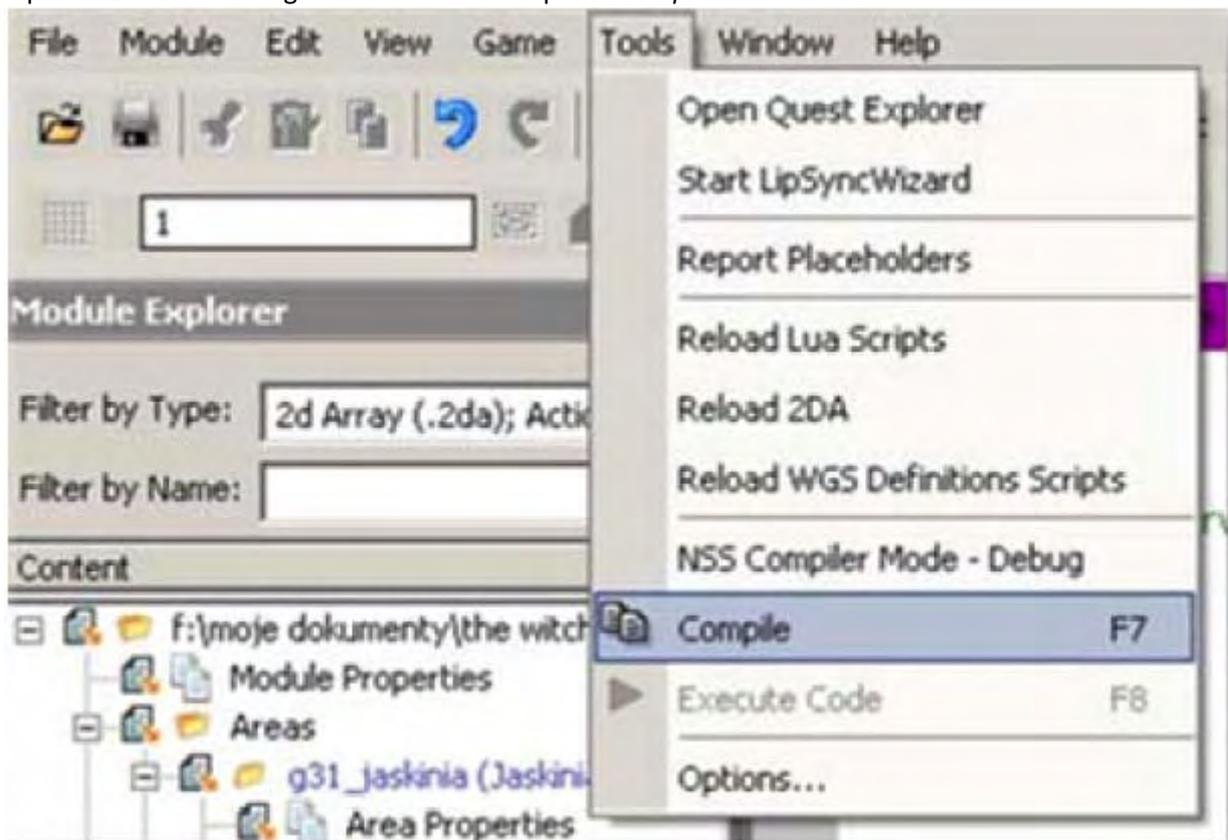


Das Fenster *new neverwinternights script* öffnet sich:



```
1 void main()  
2 (  
3  
4 )
```

Dies ist der Scripteditor. Die zweite Möglichkeit, ihn zu starten, wird später in diesem Kapitel beschrieben. Jedes Script muss kompiliert oder in einen Code "übersetzt" werden, den der Computer versteht. Zum Kompilieren gibt es zwei Möglichkeiten: die erste ist das Speichern der Datei. Dabei wird sie automatisch kompiliert. Die zweite Möglichkeit ist der Menüpunkt *Compile* im Menü *Tools*:

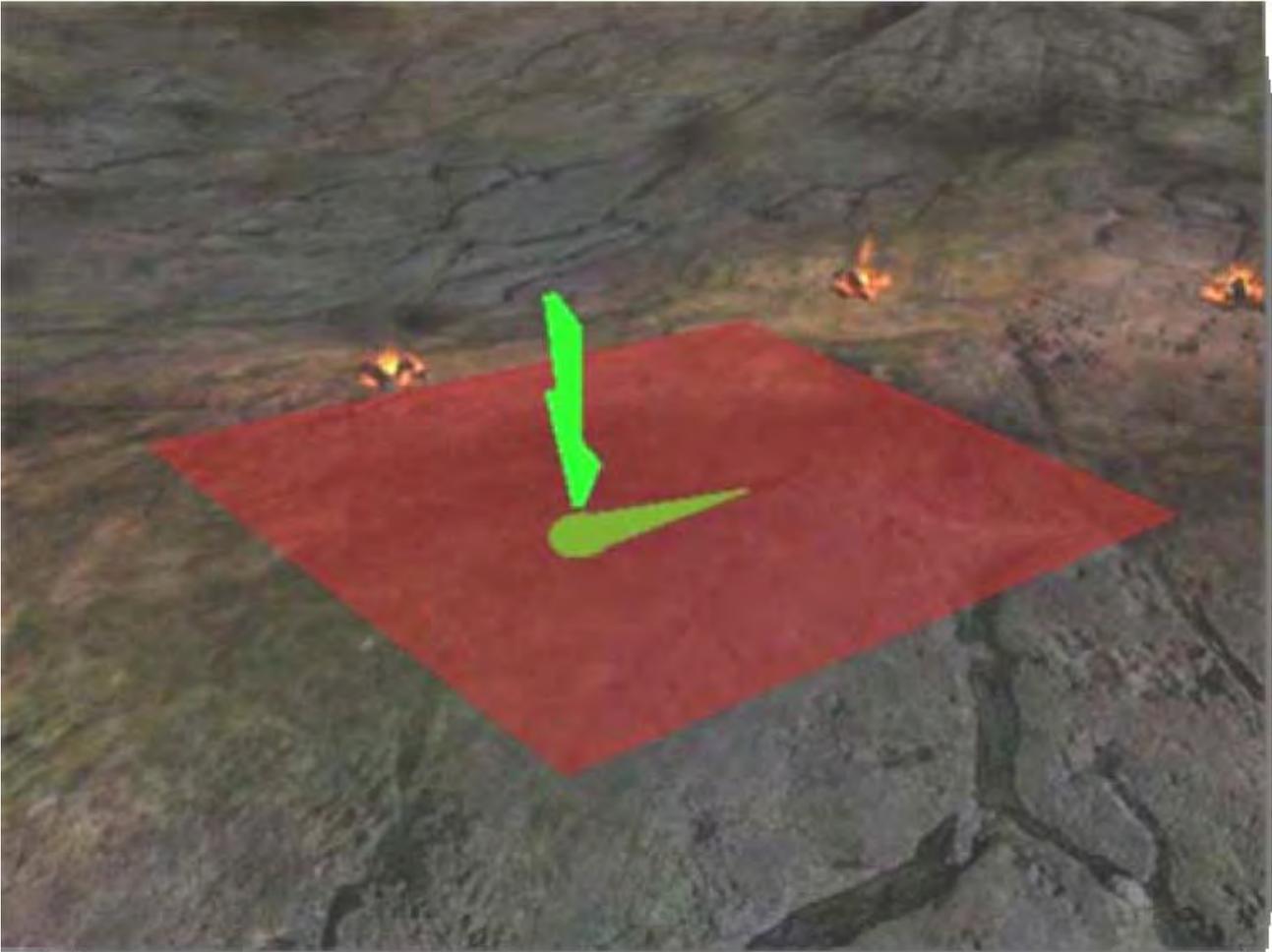


Der Befehl *Compile* steht nur zur Verfügung, wenn der Scripteditor das aktive Fenster ist. Der Befehl bewirkt außerdem, dass das Script vor dem Kompilieren gespeichert wird.

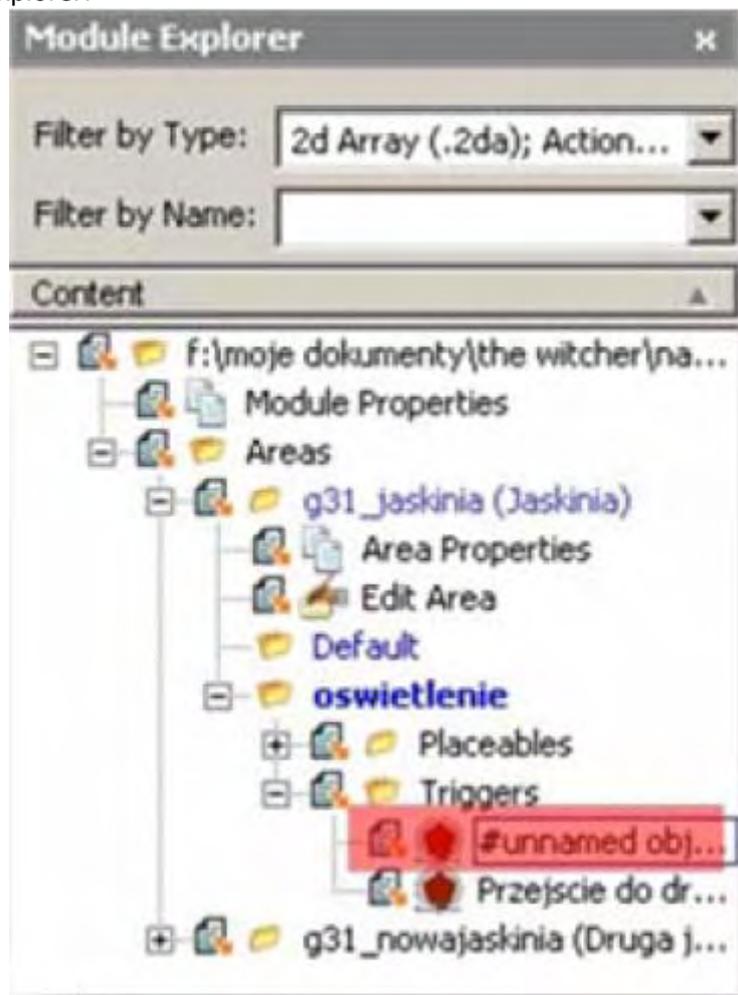
Starten einer Cutscene

Im vorigen Kapitel haben wir gesehen, wie man eine einfache Cutscene erstellt. Jetzt müssen wir noch einige Optionen hinzufügen, die dafür sorgen, dass die Szene, die wir erstellt haben, auch abgespielt wird. Dazu erstellen wir einen Trigger und ein Script, die dafür sorgen, dass die Szene abgespielt wird, sobald Geralt (der Spieler) eine bestimmte Stelle betritt. Da wir außerdem wollen, dass die Szene nur einmal abgespielt wird, muss das Script den Trigger anschließend abschalten.

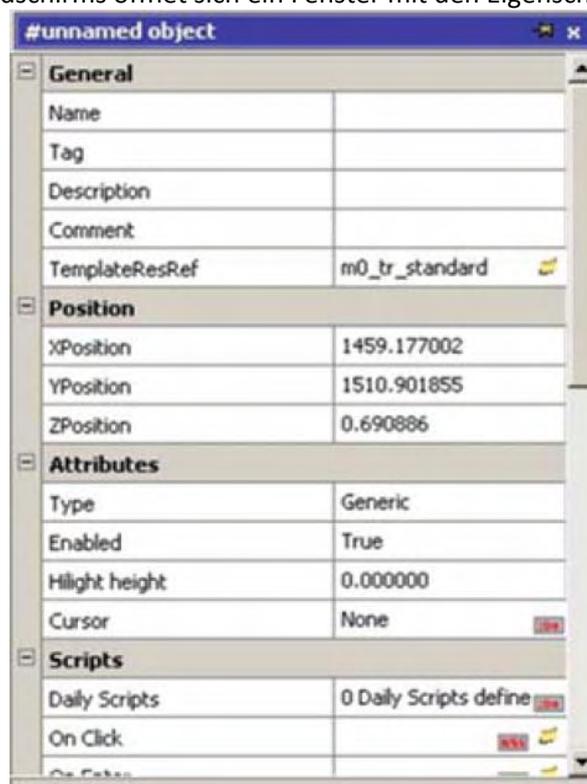
Als erstes erstellen wir einen Trigger. Wie das funktioniert, ist in einem früheren Abschnitt des Handbuchs beschrieben. Platziere den Trigger so, dass der Startpunkt im Triggerbereich liegt:



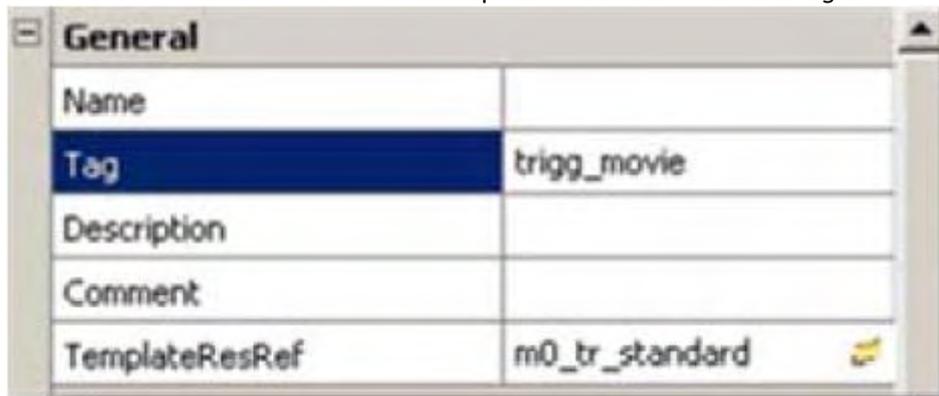
Wie du Dich vielleicht erinnerst, liegt der Startpunkt, den wir vorher erstellt haben, im ersten Gebiet, das wir erstellt haben, also g31_cave. Platziere den Trigger an dieser Stelle und doppelklicke darauf. Am besten geht das im Module Explorer:



Auf der rechten Seite des Bildschirms öffnet sich ein Fenster mit den Eigenschaften des Triggers:

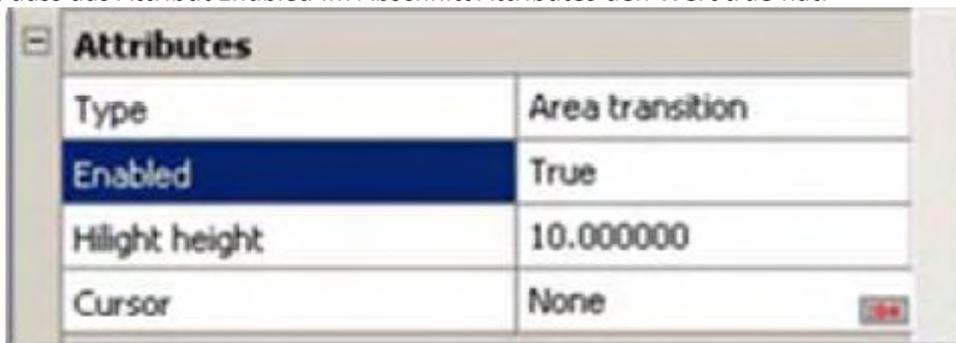


Gib *trigg_movie* im Abschnitt *General* in der rechten Spalte neben dem Attribut *Tag* ein:



General	
Name	
Tag	trigg_movie
Description	
Comment	
TemplateResRef	m0_tr_standard

Achte darauf, dass das Attribut *Enabled* im Abschnitt *Attributes* den Wert *true* hat:

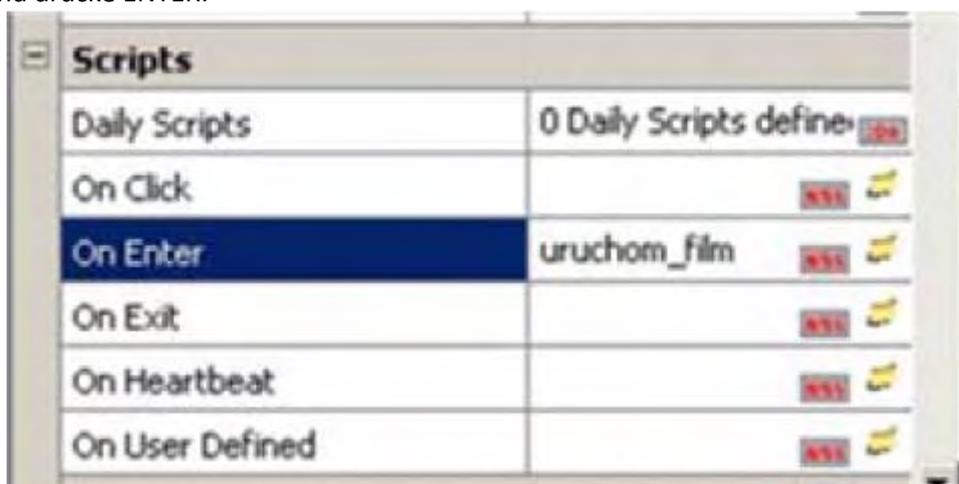


Attributes	
Type	Area transition
Enabled	True
Highlight height	10.000000
Cursor	None

Das bedeutet, dass der Trigger eingeschaltet ist. Schau Dir jetzt den Abschnitt *Scripts* an. Dort sollte es ein Attribut namens *On Enter* geben. Das Script, das diesem Attribut zugeordnet ist, wird aufgerufen, sobald ein Charakter (Geralt oder ein NPC) den Bereich betritt. Diesem Attribut weisen wir jetzt das Script zu, das die Cutscene auslöst.

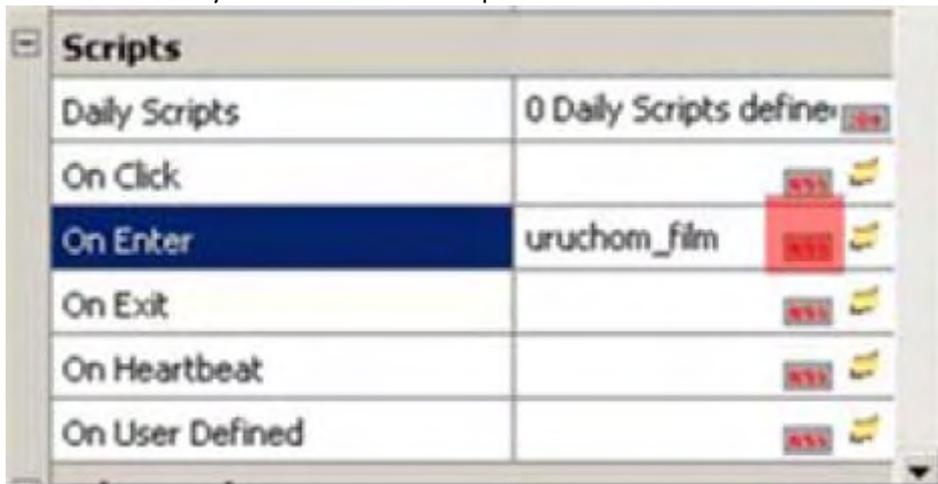
Der Startpunkt liegt im Triggerbereich, also wird die Szene direkt nach dem Start des Moduls abgespielt, weil sich Geralt in den Triggerbereich bewegt.

Jetzt schauen wir uns an, wie man den Scripteditor startet, ohne dabei das Menü *File* zu verwenden. Füge den Dateinamen *launch_film* hinzu. Klicke dazu auf die rechte Spalte neben dem Attribut *On Enter*, gib den Namen ein und drücke ENTER:

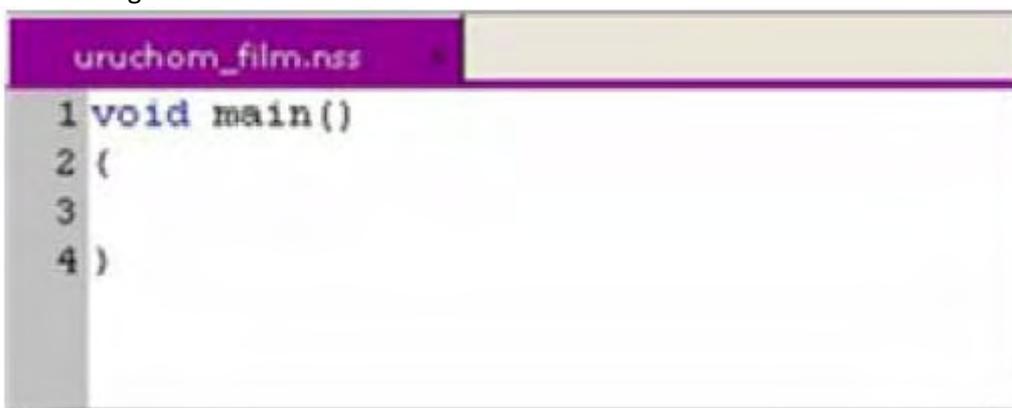


Scripts	
Daily Scripts	0 Daily Scripts define 
On Click	 
On Enter	uruchom_film  
On Exit	 
On Heartbeat	 
On User Defined	 

Klicke jetzt auf das kleine rote Symbol in der rechten Spalte:



Der Scripteditor wird geöffnet:

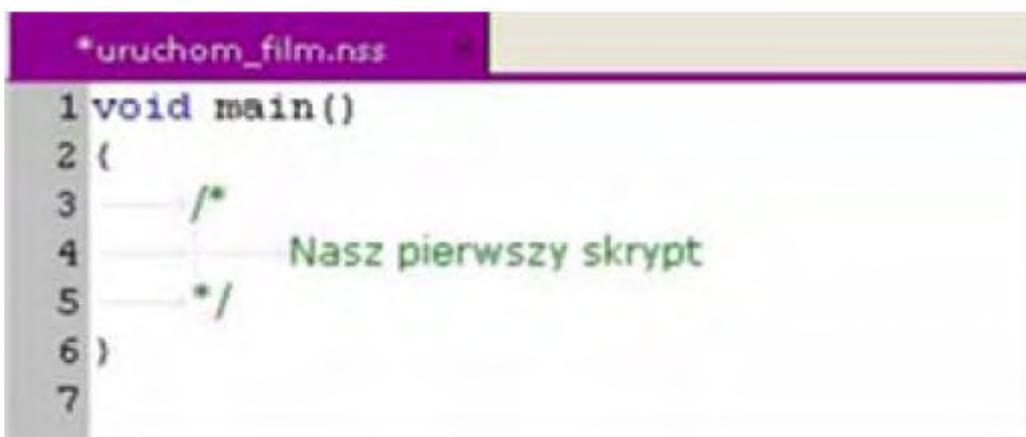


Das Script bekommt automatisch den Namen, den Du vorher eingegeben hast. Tatsächlich ist es besser, den Scripteditor auf diese Art zu öffnen, als über das Menü *File*, weil das Script so automatisch einen Namen hat und einem Attribut zugeordnet wird.

Jetzt müssen wir unser erstes Script schreiben. Wie Du siehst, beginnt jedes Script mit der Funktion *void main()*. Der Inhalt des Scripts steht zwischen den geschweiften Klammern "{}".

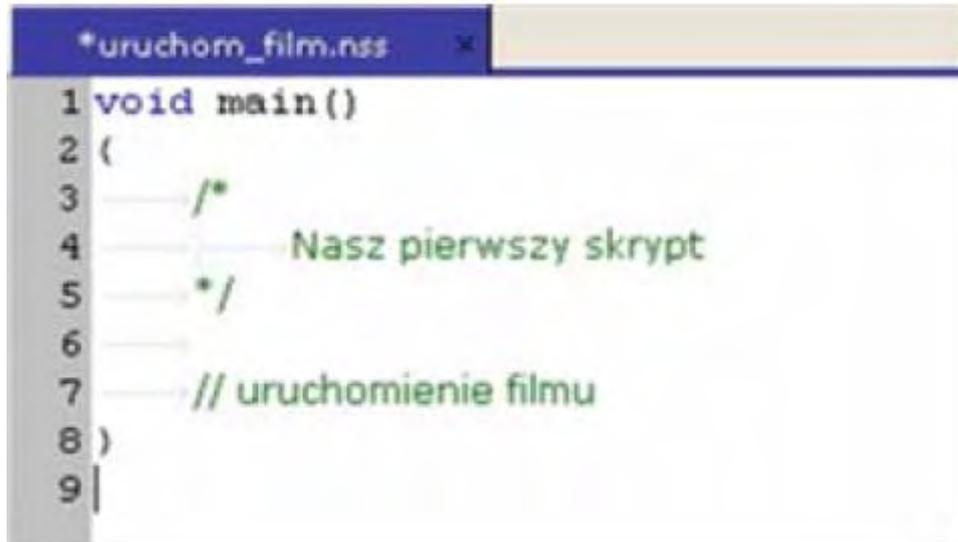
Zuerst fügen wir einen Kommentar ein, damit jemand, der das Script liest, auch weiß, worum es geht. Gib folgendes ein:

```
/*
Unser erstes Script
*/
```



Alles was zwischen `/*` und `*/` steht, ist ein Kommentar und wird vom Compiler nicht übersetzt. Der Kommentar ist eine zusätzliche Information für den Leser, damit er sieht, wofür das Script gut ist. Diese Art von Kommentar kann über mehrere Zeilen geschrieben werden. Eine andere Art von Kommentar verwendet die Zeichen `//`. Schreibe folgendes unter den ersten Kommentar:

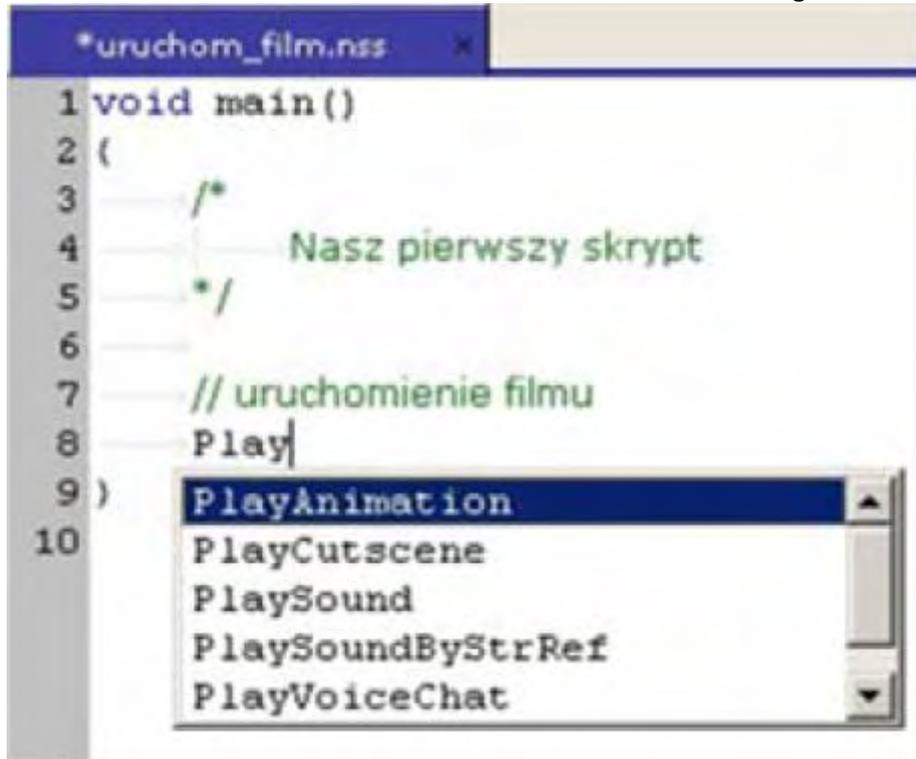
`// Film starten:`



```
1 void main()
2 (
3  /*
4  Nasz pierwszy skrypt
5  */
6
7  // uruchomienie filmu
8 )
9 |
```

Dieser Kommentar ist ein Einzeiler. Wie der Name schon sagt, ist er nur eine Zeile lang. Wenn die nächste Zeile ebenfalls ein Kommentar sein soll, muss sie auch wieder mit `//` beginnen.

Jetzt schreiben wir unsere erste Funktion. Gib unter dem zweiten Kommentar folgendes ein:

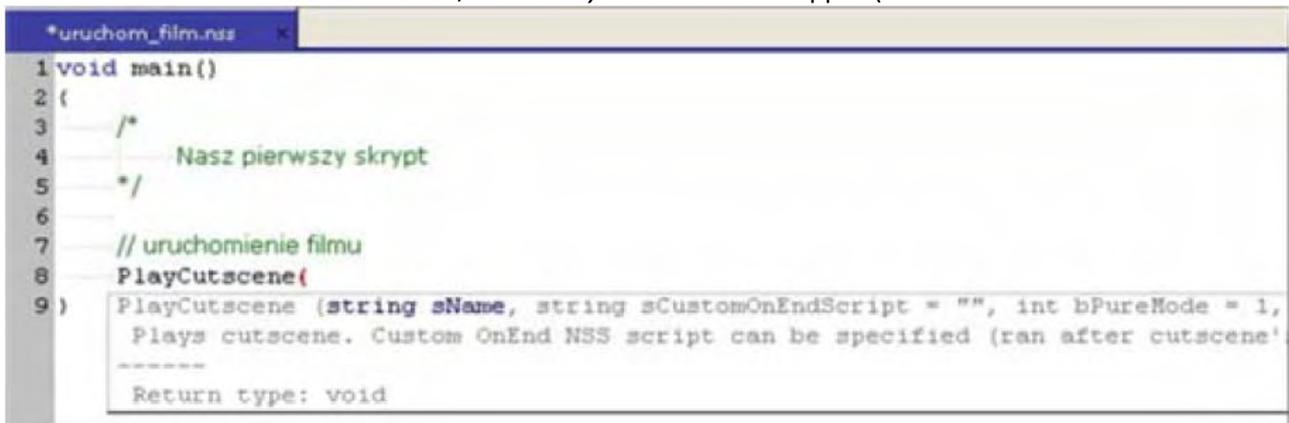


```
1 void main()
2 (
3  /*
4  Nasz pierwszy skrypt
5  */
6
7  // uruchomienie filmu
8  Play
9 )
10 |
```

- PlayAnimation
- PlayCutscene
- PlaySound
- PlaySoundByStringRef
- PlayVoiceChat

Du siehst, dass sich ein Fenster öffnet, sobald Du `Play` eingegeben hast. Das Fenster enthält eine Liste von Funktionen, die in diesem Kontext zulässig sind. Es gibt 6 Funktionen, die das Schlüsselwort `Play` enthalten - verwende die Tasten `↓` und `↑`, um Dich in der Liste zu bewegen. Wenn Du die passende Funktion ausgewählt hast, drücke entweder `TAB` oder `ENTER`. Natürlich kannst Du die Funktion auch von Hand eingeben.

Weil wir eine Cutscene starten wollen, wähle *PlayCutscene*. Dann tippe "(".

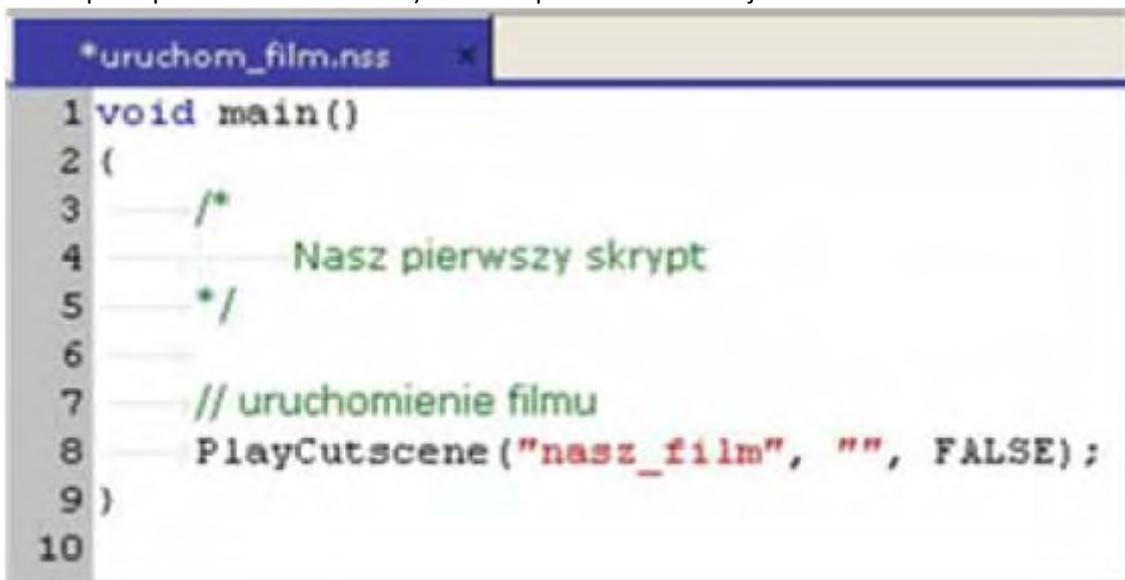


```
1 void main()
2 (
3     /*
4     Nasz pierwszy skrypt
5     */
6
7     // uruchomienie filmu
8     PlayCutscene(
9     PlayCutscene (string sName, string sCustomOnEndScript = "", int bPureMode = 1,
        Plays cutscene. Custom OnEnd NSS script can be specified (ran after cutscene'
        -----
        Return type: void
```

Wie Du siehst, öffnet sich ein neues Fenster. Diesmal enthält es Hinweise zur Syntax der Funktion. Wir folgen dem Vorschlag und geben als nächstes den Namen unserer Szene ein. Das ist natürlich ein Dateiname, aber ohne die Erweiterung *.cut*.

Das nächste Argument der Funktion ist der Name des Scripts, das am Ende der Szene gestartet werden soll. Das brauchen wir nicht, also lassen wir die Anführungszeichen "" einfach leer.

Das dritte Argument ist für uns wichtiger. Es beschreibt den Abspielmodus für unseren Film. Wenn dieses Argument auf *true* (1) steht, dann werden die Charaktere nicht angezeigt. Steht es auf *false* (0), dann werden die Charaktere angezeigt, und das ist genau das, was wir wollen. Während sich die beiden Kameras bewegen, können wir Jethro, Siegfried und Geralt sehen (vorausgesetzt, das Sichtfeld der Kameras befindet sich bei den Spawnpoints der Charaktere). Die komplette Zeile sollte jetzt so aussehen:



```
1 void main()
2 (
3     /*
4     Nasz pierwszy skrypt
5     */
6
7     // uruchomienie filmu
8     PlayCutscene("nasz_film", "", FALSE);
9 )
10
```

Denke daran, ans Ende jeder Zeile des Scripts ein Semikolon ";" zu setzen. Die Funktion *PlayCutscene* startet nun die angegebene Szene. Jetzt müssen wir nur noch zum Ende der Szene den Trigger abschalten.

Bevor wir das jedoch tun, fügen wir einen weiteren Kommentar ein:

```
*uruchom_film.nss
1 void main()
2 {
3     /*
4     /*      Nasz pierwszy skrypt
5     */
6
7     // uruchomienie filmu
8     PlayCutscene("nasz_film", "", FALSE);
9
10    // wyłączenie wyzwalacza
11 }
12
```

Jetzt fügen wir die nächste Funktion ein. Gib folgendes ein:

```
*uruchom_film.nss
1 void main()
2 {
3     /*
4     /*      Nasz pierwszy skrypt
5     */
6
7     // uruchomienie filmu
8     PlayCutscene("nasz_film", "", FALSE);
9
10    // wyłączenie wyzwalacza
11    Enable|
12 }
13
```

- EnableSequenceJumps
- EnableSpawnPhase
- EnableStoryNPCActions
- EnableTrigger
- EndFistfightKnockOutEffect

Wie Du siehst, gibt es eine ganze Reihe von Funktionen, die mit *Enable* beginnen. Aber wieso verwenden wir überhaupt eine *Enable*-Funktion, wenn wir versuchen, den Trigger auszuschalten? Dazu kommen wir gleich.

Die Funktion, die wir suchen, heißt *EnableTrigger*. Gib das ein und öffne eine Klammer (:

```
*uruchom_film.nss x
1 void main()
2 (
3   /*
4   |   Nasz pierwszy skrypt
5   */
6
7   // uruchomienie filmu
8   PlayCutsScene("nasz_film", "", FALSE);
9
10  // wyłączenie wyzwalacza
11  EnableTrigger(
12 )
13
```

```
EnableTrigger (object oTrigger, int bValue)
  Enables or disables trigger. Inactive trigger doesn't react on events.
-----
Return type: void
```

Wie Du im Hinweis sehen kannst, ist das erste Argument dieser Funktion das Triggerobjekt. Zunächst müssen wir das Triggerobjekt referenzieren. Dafür gibt es zwei Möglichkeiten. Die erste besteht darin, folgende Zeile in das Script einzufügen:

```
object oTrigger = GetObjectByTag("trigg_movie");
```

Das Schlüsselwort *object* zeigt an, dass wir mit einem Objekt arbeiten. *oTrigger* ist ein willkürlich gewählter Name, dem das Objekt zugewiesen wird. Die Funktion *GetObjectByTag* greift auf das Objekt mit dem angegebenen Tag zu. In unserem Fall wollen wir auf das Triggerobjekt zugreifen, dem wir den Tag *trigg_movie* zugewiesen haben. Dazu müssen wir nur noch folgendes eingeben:

```
EnableTrigger(oTrigger)
```

Es gibt noch eine einfachere Möglichkeit: das Schlüsselwort *OBJECT_SELF*. Schlüsselwörter werden im Scripteditor in blau dargestellt. *OBJECT_SELF* bezieht sich auf das Objekt, dem das Script gehört. Da das Script dem Trigger *trigg_movie* zugeordnet ist, wird *OBJECT_SELF* automatisch auf den Trigger *trigg_movie* verweisen. Das bedeutet, dass wir die Zeile *object oTrigger* nicht brauchen.

Das Script sollte jetzt wie folgt aussehen:

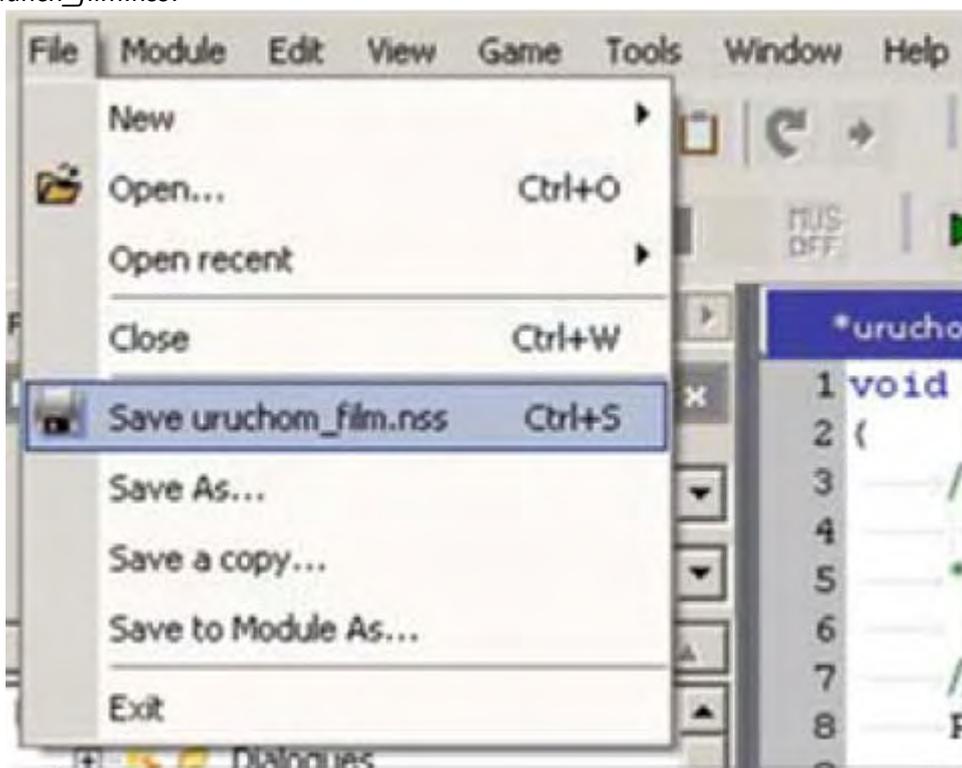
```
*uruchom_film.nss x
1 void main()
2 (
3   /*
4   |   Nasz pierwszy skrypt
5   */
6
7   // uruchomienie filmu
8   PlayCutsScene("nasz_film", "", FALSE);
9
10  // wyłączenie wyzwalacza
11  EnableTrigger(OBJECT_SELF
12 )
13
```

Die Funktion *EnableTrigger* kann den Trigger ein- oder ausschalten. Das zweite Argument der Funktion kann die Werte *true* oder *false* annehmen. Gib nach dem Komma *false* ein und schließe die Klammer):

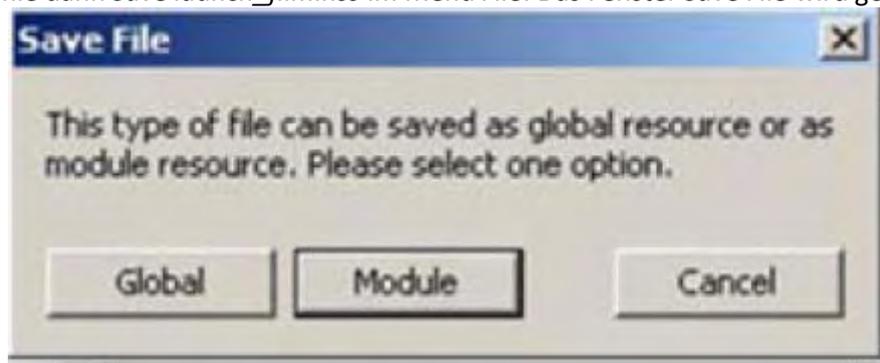
```
*uruchom_film.nss
1 void main()
2 (
3     /*
4     Nasz pierwszy skrypt
5     */
6
7     // uruchomienie filmu
8     PlayCutscene("nasz_film", "", FALSE);
9
10    // wyłączenie wyzwalacza
11    EnableTrigger(OBJECT_SELF, FALSE);
12 )
13
```

Denke daran, die Zeile mit einem Semikolon ";" abzuschließen.

Das wars - unser Script ist fertig. Jetzt müssen wir es nur noch speichern. Wähle im Menü *File* den Menüpunkt *Save launch_film.nss*:

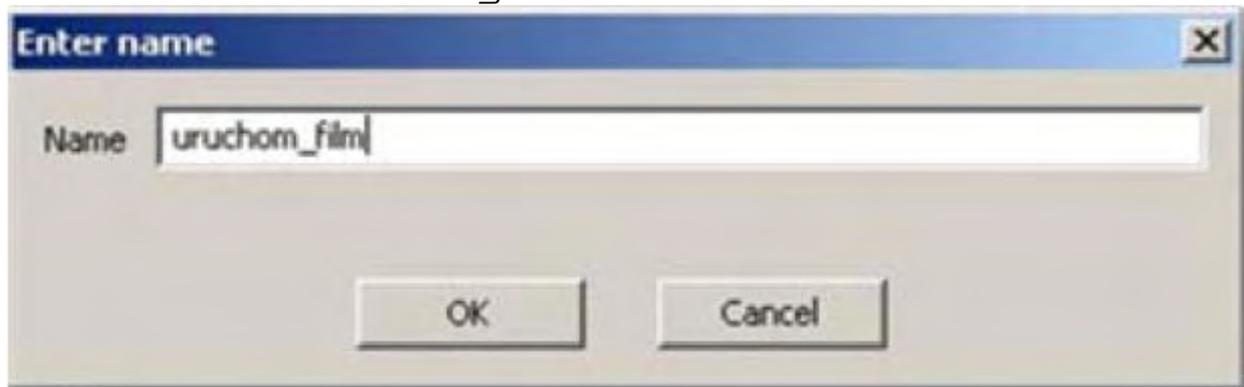


Wenn der Menüpunkt *Save launch_film.nss* nicht im Menü erscheint, oder wenn stattdessen ein anderer Dateiname angezeigt wird, dann ist das Fenster *launch_film.nss* nicht aktiv. Klicke auf die Titelleiste des Fensters und wähle dann *Save launch_film.nss* im Menü *File*. Das Fenster *Save File* wird geöffnet:



Beim Speichern der Datei kannst Du wählen, ob die Datei global gespeichert wird oder nur im aktuellen Modul. Global bedeutet, dass das Script auch in anderen Modulen zur Verfügung steht. Klicke für dieses Mal auf *Global*, aber behalte im Hinterkopf, dass Du ein Script immer auch modulgebunden speichern kannst. Das ist die bessere Wahl, wenn Du Dein Modul ins Internet hochladen und anderen zugänglich machen willst. Das komplette Modul wird dann in einer *.adv* Datei gespeichert, so wie wir es bereits an früherer Stelle in diesem Handbuch besprochen haben.

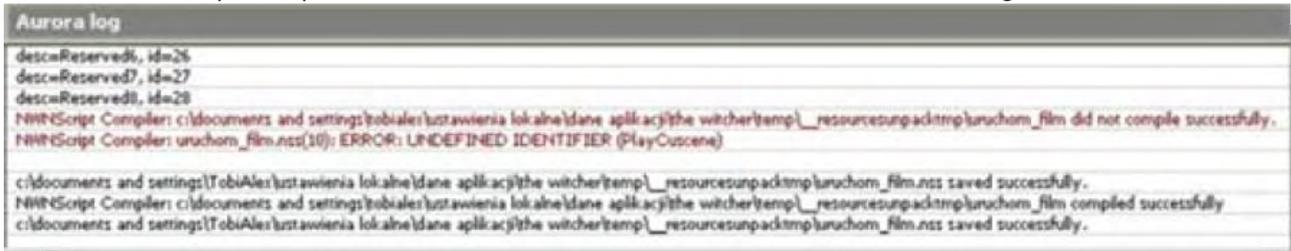
Ein Klick auf die Schaltfläche *Global* öffnet das übliche Windows-Fenster zum Speichern von Dateien. Gehe zum Verzeichnis *Data\Scripts*, in dem alle Scripts gespeichert sind, gib den Dateinamen ein und klicke auf *Speichern*. Eine Datei, die auf diesem Weg gespeichert wurde, kannst Du natürlich später zu einem Modul hinzufügen. Das machen wir jetzt: klicke auf die Schaltfläche *Module*. Das Fenster *Enter Name* wird geöffnet. Gib im Feld *Name* den Namen *launch_film* ein:



Klicke auf *OK*. Das Script wird jetzt kompiliert und erscheint dann im Baum *Neverwinter scripts sources* im Module Explorer:



Nachdem das Script kompiliert ist, werfen wir einen Blick auf das Fenster *Aurora-Log*.



```
Aurora log
desc=Reserved6, id=26
desc=Reserved7, id=27
desc=Reserved8, id=28
NWNScript Compiler: c:\documents and settings\tobialer\zstawienia lokalne\dane aplikacji\the witcher\temp\_resourcesunpacked\uruchom_film did not compile successfully.
NWNScript Compiler: uruchom_film.nss(10): ERROR: UNDEFINED IDENTIFIER (PlayCuscene)

c:\documents and settings\tobialer\zstawienia lokalne\dane aplikacji\the witcher\temp\_resourcesunpacked\uruchom_film.nss saved successfully.
NWNScript Compiler: c:\documents and settings\tobialer\zstawienia lokalne\dane aplikacji\the witcher\temp\_resourcesunpacked\uruchom_film compiled successfully
c:\documents and settings\tobialer\zstawienia lokalne\dane aplikacji\the witcher\temp\_resourcesunpacked\uruchom_film.nss saved successfully.
```

Wenn während des Kompilierens Fehler auftreten, werden sie im Log violett dargestellt. Wie Du siehst, endete das Kompilieren hier mit solch einem Fehler:

NWNScript Compiler: \.\ launch_film.nss did not Compile successfully

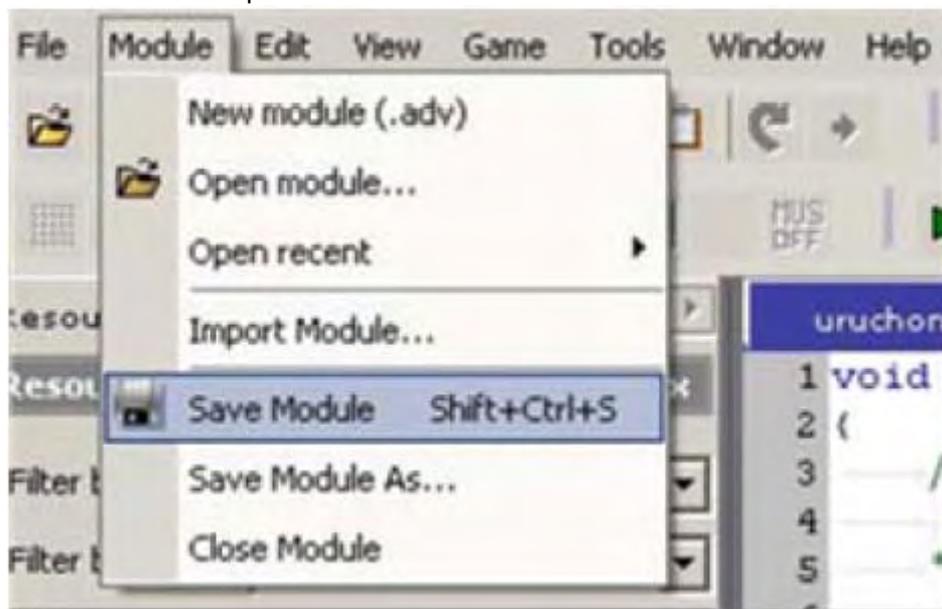
Das bedeutet, dass das Script nicht erfolgreich kompiliert werden konnte. In der nächsten Zeile wird der Fehler beschrieben:

NWNScript Compiler: launch_film.nss(10): ERROR: UNDEFINED IDENTIFER (PlayCuscene)

In den Klammern "(" hinter dem Namen des Scripts steht die Nummer der Zeile, in der der Fehler aufgetreten ist. Dann wird der Fehlertyp angegeben - UNDEFINED IDENTIFIER. In den Klammern dahinter steht die Bezeichnung, auf die sich die Meldung bezieht. Und natürlich stimmt die Meldung - eine Funktion namens *PlayCuscene* gibt es nicht, es muss *PlayCutscene* heißen. Wenn der Tippfehler korrigiert ist, muss das Script erneut kompiliert werden:

NWNScript Compiler: \.\ launch_film compiled successfully

Das bedeutet, dass das Script jetzt kompiliert wurde, aber wir müssen noch das Modul speichern. Wähle dazu im Menü *Modules* den Menüpunkt *Save Module*:

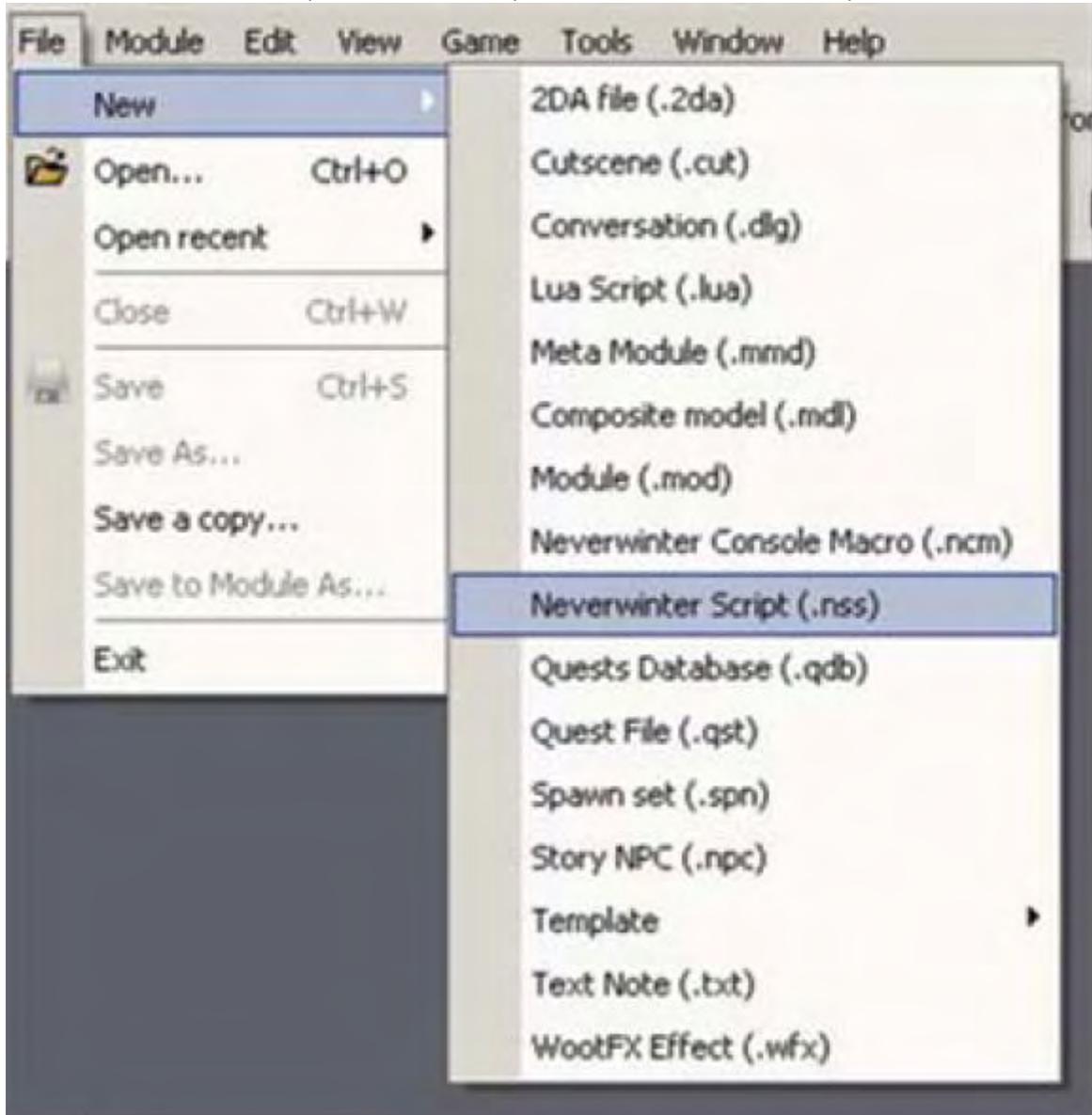


Das wars, wir haben unser erstes Script erstellt. Es ist zwar nicht furchtbar kompliziert, aber für den Moment wird es genügen.

Der Hexer und seine Ausrüstung

Ein sehr wichtiger Schritt beim Erstellen eines Moduls besteht darin, festzulegen, welche Ausrüstung Geralt haben soll. Das ist leider nicht ganz einfach. Damit es funktioniert, müssen wir ein Script schreiben, das Geralts Ausrüstung definiert.

Dieses Mal erstellen wir das Script über den Menüpunkt *New-> Neverwinter Script (.nss)* im Menü *File*:



Das Fenster *new neverwinter nights script* öffnet sich:

```
new neverwinternights script
1 void main()
2 {
3
4 }
```

Gib zuerst einen Kommentar ein, damit man sieht, wofür das Script gut ist:

```
*new neverwinterights script
1 void main()
2 (
3   /*
4   |----- Ekwipunek Geralta
5   */
6
7 )
```

Die Funktion, mit der wir Geralts Ausrüstung erstellen werden, heißt *CreateItemOnObject*. Eines der Argumente, die wir definieren müssen, ist das Objekt, das dem Gegenstand zugeordnet wird. In unserem Fall ist das Geralt. Dazu gibt es die Funktion *GetFirstPC*. Zunächst aber fügen wir einen weiteren Kommentar ein, damit unser Script verständlicher wird. Dann gib als nächste Zeile die Funktion *GetFirstPC* ein:

```
*new neverwinterights script
1 void main()
2 (
3   /*
4   |----- Ekwipunek Geralta
5   */
6
7   // pobranie obiektu gracza
8   |
9 )
```

Die Zeile kann wie folgt gelesen werden: Lies das Objekt des ersten Spielers aus und weise es dem Objekt oPC zu.

```
*new neverwinterights script
1 void main()
2 (
3   /*
4   |----- Ekwipunek Geralta
5   */
6
7   // pobranie obiektu gracza
8   object oPC = GetFirstPC();
9   |
10 )
```

Jetzt kommt die Funktion zum Erstellen der Ausrüstung ins Spiel. In der Funktion *CreateOnObjekt* müssen wir den Namen der Objektvorlage definieren. Alle Objektvorlagen sind im Ordner "Data\Templates\Items" gespeichert. Als Beispiel werden wir zu Geralts Ausrüstung ein paar Orens (die Währung, die in The Witcher verwendet wird) und ein Silberschwert hinzufügen - ein Stahlschwert hat Geralt standardmäßig dabei, darum müssen wir uns also nicht kümmern.

Die Vorlagen für Orens haben Dateinamen von "it_gold_001.uti" bis "it_gold_014.uti". Sie sind unter "Data\Templates\Items\Valuables" gespeichert. Die Vorlage für das Silberschwert heißt "it_svswd_004.uti" und ist unter "Data\Templates\Items\Weapon&Armor" gespeichert. Denke daran, dass die Funktion *CreateItemOnObject* die Dateinamen der Vorlagen ohne die Erweiterung *.uti* benötigt.

Nun zurück zu unserem Skript. Als erstes schreiben wir die Funktion *CreateOnObjekt*, beginnend mit einer Klammer "(":

```
*new neverwinter nights script
1 void main()
2 {
3     /*
4     Ekwipunek Geralta
5     */
6
7     // pobranie obiektu gracza
8     object oPC = GetFirstPC();
9
10    CreateItemOnObject(
11    CreateItemOnObject (string sItemTemplate, object oTarget=OBJECT_SELF, int nStackSize=1)
12)    Create an item with the template sItemTemplate in oTarget's inventory.
    - nStackSize: This is the stack size of the item to be created
    * Return value: The object that has been created. On error, this returns
    OBJECT_INVALID.
    -----
    Return type: object
```

Der Hinweis zu den Argumenten der Funktion öffnet sich. Nach der Objektvorlage und dem Zielobjekt muss als letztes Argument der Funktion die Menge übergeben werden, also die Anzahl der Objekte, die aus der selben Vorlage erstellt werden. Gib folgendes ein:

```
*new neverwinter nights script
1 void main()
2 {
3     /*
4     Ekwipunek Geralta
5     */
6
7     // pobranie obiektu gracza
8     object oPC = GetFirstPC();
9
10    CreateItemOnObject ("it_gold_005", oPC, 50);
11    |
12 )
```

Denke daran, die Zeile mit einem ";" abzuschließen. Jetzt fügen wir noch einen Kommentar ein, allerdings an einer anderen Stelle als gewöhnlich:

```
*new nevervinternights script
1 void main()
2 {
3     /*
4     Ekwipunek Geralta
5     */
6
7     // pobranie obiektu gracza
8     object oPC = GetFirstPC();
9
10    CreateItemOnObject("it_gold_005", oPC, 50); // dodanie orenow
11
12 }
```

Wie Du siehst, können wir Kommentare am Ende einer Zeile hinzufügen. Man kann keinen Kommentar am Anfang der Zeile hinzufügen, jedenfalls nicht mit den "/*" Zeichen. Wenn Du wirklich einen Kommentar am Anfang der Zeile schreiben willst (nicht zu empfehlen, da es das Script unleserlich macht), benutze die "/*" "*/" Zeichen.

```
*new nevervinternights script
1 void main()
2 {
3     /*
4     Ekwipunek Geralta
5     */
6
7     // pobranie obiektu gracza
8     object oPC = GetFirstPC();
9
10    /* dodanie orenow */ CreateItemOnObject("it_gold_005", oPC, 50);
11
12 }
```

Die Zeile kann wie folgt gelesen werden: Füge 50 Objekte, basierend auf der Vorlage "it_gold_005", zu dem Objekt oPC hinzu.

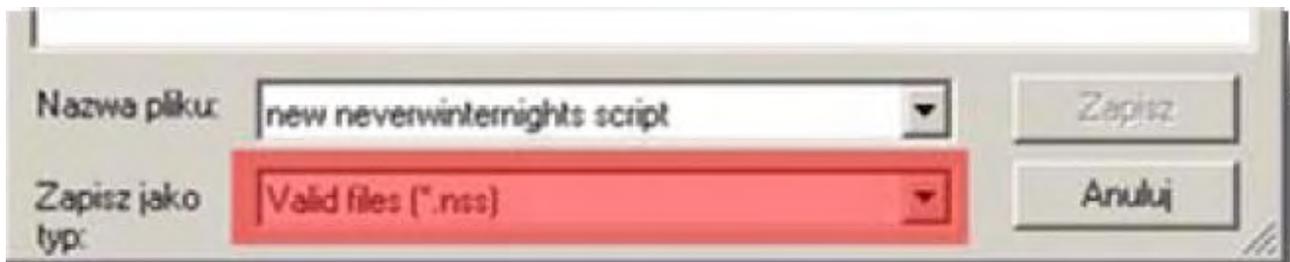
Nun fügen wir das Silberschwert hinzu:

```
*new nevervinternights script
1 void main()
2 {
3     /*
4     Ekwipunek Geralta
5     */
6
7     // pobranie obiektu gracza
8     object oPC = GetFirstPC();
9
10    CreateItemOnObject("it_gold_005", oPC, 50); // dodanie orenow
11    CreateItemOnObject("it_svswd_004", oPC, 1); // dodanie srebrnego miecza
12 }
```

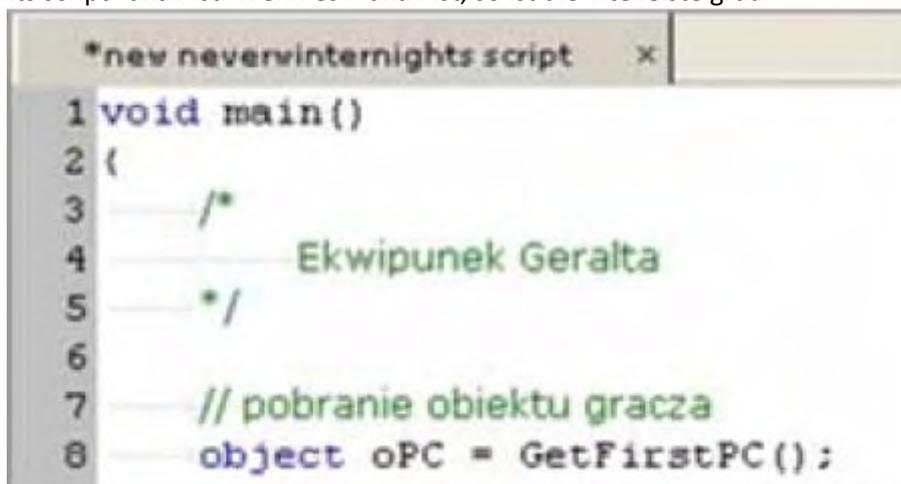
Das wars. Unser Skript ist fertig. Nun speichere es mit "Save as..." im Menü "File":



Es öffnet sich das übliche Windows-Fenster zum Speichern von Dateien. Gehe zum Ordner "Data\Scripts" (wo alle Scripts gespeichert werden), gib einen Namen ein und klicke auf "Save". Denke daran, dass der Name höchstens 16 Zeichen lang sein darf. Vergewissere Dich, dass die Datei im Format .nss gespeichert wird:



Wenn stattdessen andere Typen von Dateien aufgelistet werden, versuchst Du wahrscheinlich, eine Datei aus einem falschen Fenster zu speichern. Vergewissere Dich vor dem Speichern, dass das Fenster "new neverwinternights script" aktiv ist. Wenn es inaktiv ist, so ist die Titelleiste grau:



Eine andere Farbe zeigt an, dass das Fenster aktiv ist:

A screenshot of a script editor window. The title bar shows '*new neverwinternights script'. The script content is as follows:

```
1 void main()
2 {
3     /*
4     Ekwipunek Geralta
5     */
6
7     // pobranie obiektu gracza
8     object oPC = GetFirstPC();
```

Ein Sternchen vor dem Namen in der Titelleiste zeigt an, dass das Script seit der letzten Änderung noch nicht gespeichert wurde. Das Script kann direkt im Modul gespeichert werden. Wähle dazu "Save new neverwinternights scripts" im Menü "File" und wähle "Modul" im Fenster "Save File".

In diesem Handbuch haben wir diese Methode nicht verwendet. Stattdessen haben wir alle Dateien global gespeichert und sie dann manuell über den Resources Explorer zu den Modulen hinzugefügt. In einem der früheren Kapitel wird darauf genauer eingegangen. Schau jetzt im *Aurora log*, ob das Kompilieren und Speichern des Scripts erfolgreich war:

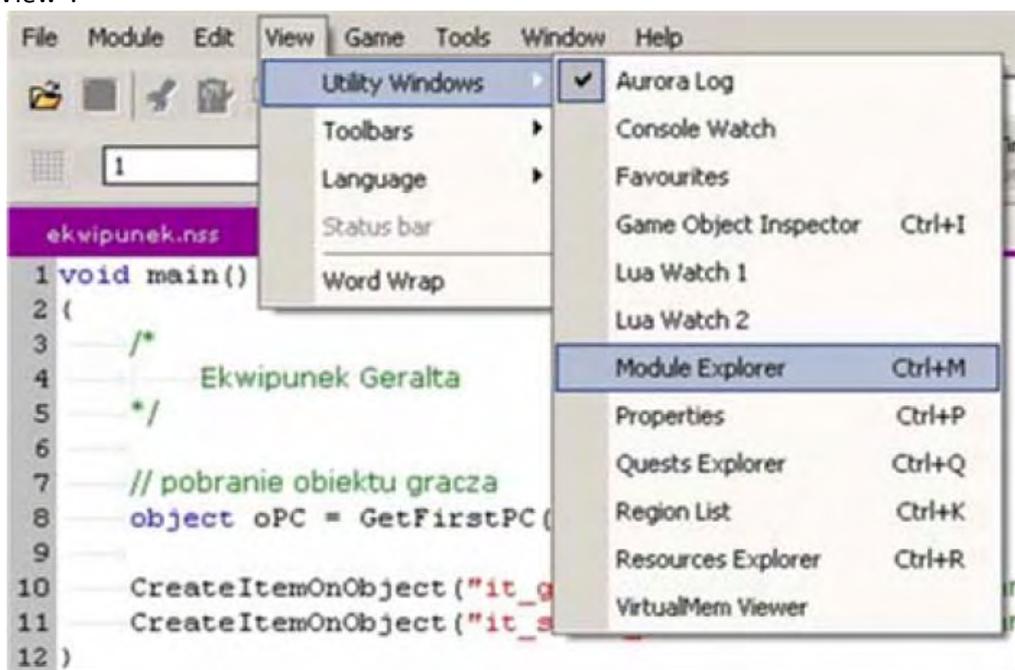
A screenshot of the Aurora Log window showing three messages:

```
e:\Gry\Wiedlgmin\Data\Scripts\ekwipunek.nss saved successfully.
NWNScript Compiler: e:\gry\wiedlgmin\data\scripts\ekwipunek compiled successfully
e:\Gry\Wiedlgmin\Data\Scripts\ekwipunek.nss saved successfully.
```

Der erste Eintrag besagt, dass der Speicherprozess erfolgreich war, während der zweite anzeigt, dass das Kompilieren ohne Fehler verlief. Der dritte Eintrag ist noch einmal ein Speichern.

Nun fügen wir unser Script dem Modul hinzu. Am besten geht das, indem man im Module Explorer das Script dem Ereignis "OnClientEnter" zuweist.

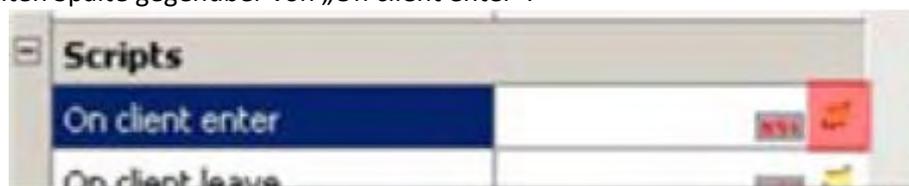
Das bedeutet, dass das Script gestartet wird, wenn Geralt im Modul erscheint. Lade das Modul, wenn es noch nicht geladen ist, und öffne den Module Explorer. Wähle dazu "Utility Windows => Module Explorer" im Menü "View":



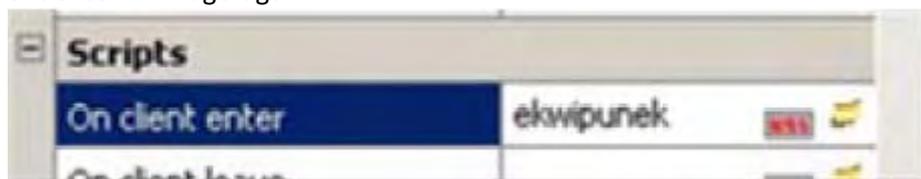
Doppelklicke im Module Explorer auf „Module Properties“:



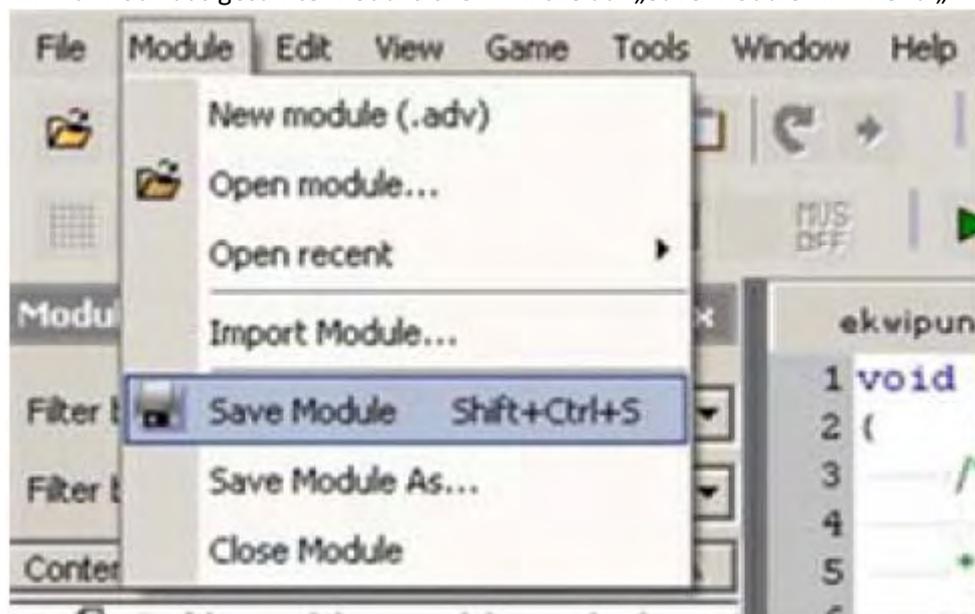
Das Fenster "Properties", das die Attribute des Moduls enthält, wird geöffnet. Klicke auf das kleine gelbe Icon in der rechten Spalte gegenüber von „On client enter“:



Es öffnet sich das übliche Windows-Fenster zum Laden von Dateien. Gehe zu „Data\Scripts (wo das Skript gespeichert ist), wähle „equipment.nss“ und klicke auf „Open“. Das Skript „equipment.nss“ wird dem Ereignis „On client enter“ hinzugefügt:



Jetzt müssen wir nur noch das gesamte Modul sichern. Klicke auf „Save Module“ im Menü „Module“:



Wird das Modul gestartet, hat Geralt nun 50 Orens und ein Silberschwert. Die hier beschriebene Methode, um einen Charakter mit Ausrüstung zu versehen, kann auch für andere Charaktere außer Geralt verwendet werden.

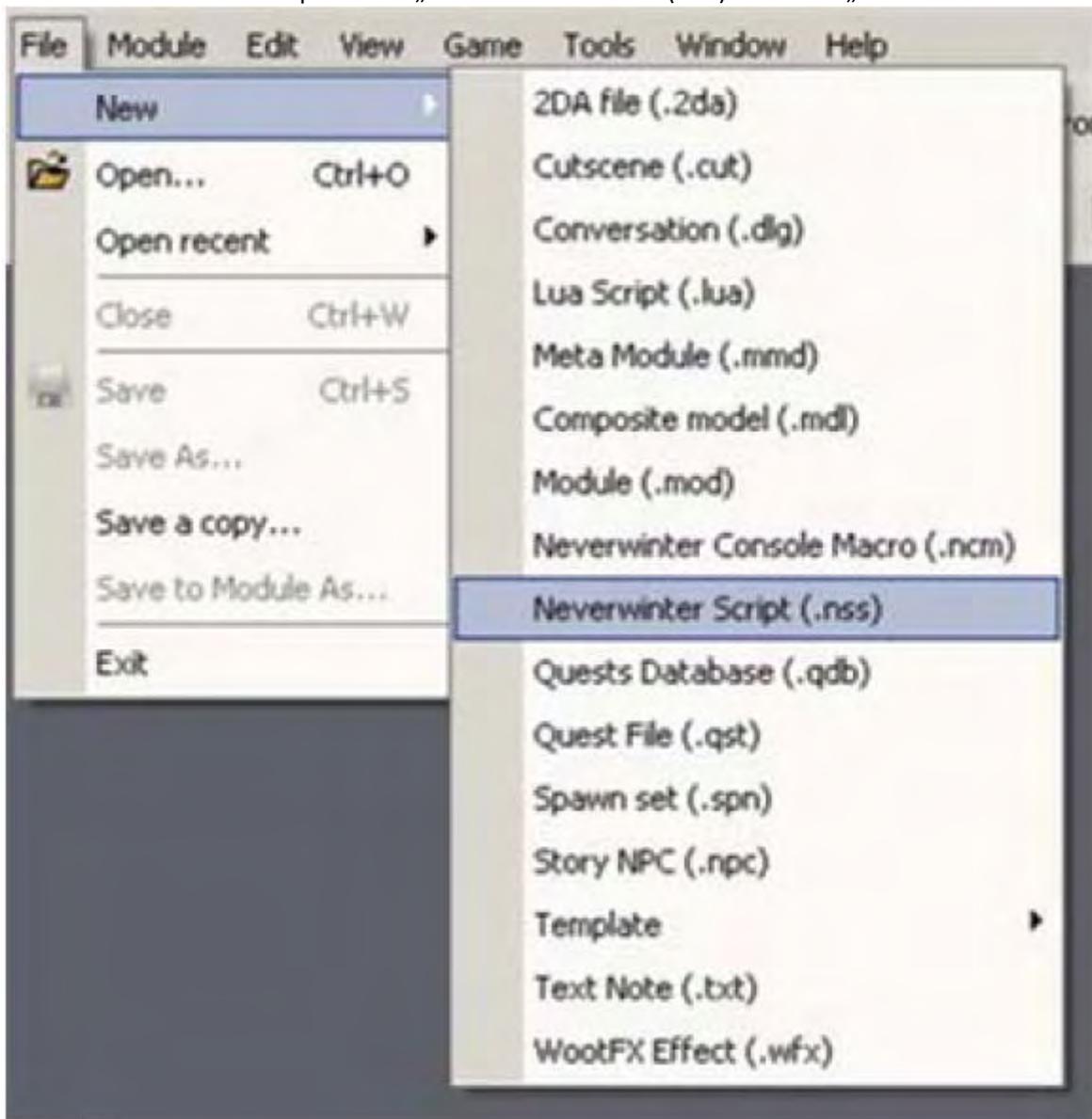
TESTEN DER FLAGS

Beim Erstellen von Dialogen und Quests im vorigen Kapitel haben wir Flags verwendet, die abhängig von der Situation den Wert 0 (false) oder 1 (true) annehmen konnten. Das größte Problem, das ich beim Erstellen der Quest für dieses Handbuch hatte, bestand darin, zu überprüfen, ob ein bestimmtes Flag tatsächlich den gewünschten Wert hatte.

Leider gibt es im D'Jinni keine einfache Möglichkeit, den Status von Flags zu prüfen. Man kann sich jedoch mit einem Script behelfen. Zur Erinnerung hier noch einmal die Flags, die wir verwendet haben:

-orders_received,
-leader_killed,
-results_reported;

Nun erstellen wir ein neues Script. Wähle „New => Neverwinter (.nss)“ aus dem „File“ Menü:



Das Fenster „new neverwinternights script“ öffnet sich:

```
new neverwinternights script x
1 void main()
2 (
3
4 )
```

Wie gewöhnlich beginnen wir mit einem Kommentar:

```
*new neverwinternights script x
1 void main()
2 (
3  /*
4  |----- Testowanie flag
5  */
6 )
```

Gib nun folgende Zeile ein:

```
*new neverwinternights script x
1 void main()
2 (
3  /*
4  |----- Testowanie flag
5  */
6
7  if(GetDialogFlagValue("posiada_zlecenie") == 0)
8 )
```

Achtung: Wenn eine Zeile mit dem Schlüsselwort *if* beginnt, dann darf am Ende der Zeile ausnahmsweise kein Semikolon stehen!. Das ist sehr wichtig. Wird eine solche Zeile mit einem Semikolon abgeschlossen, dann funktioniert das Script nicht richtig.

Das Schlüsselwort „if“ kennzeichnet eine Bedingung. Damit wird gesteuert, welche Teile eines Scripts unter welchen Umständen ausgeführt werden sollen. Die "if"-Anweisung liefert 0 (false) zurück, wenn die angegebene Voraussetzung nicht erfüllt ist. Sie liefert 1 (wahr) zurück, wenn die Voraussetzung erfüllt ist. Die Funktion *GetDialogFlagValue* liest den Wert des Flags aus, dessen Name als Argument übergeben wird.

Zusammengefasst bedeutet die obige Zeile also folgendes: "Wenn das Dialog-Flag mit dem Namen "orders_received" den Wert 0 hat, dann..." Natürlich müssen wir das Script noch fertigstellen und dabei festlegen, was genau passieren soll, wenn das Flag den Wert 0 hat. Schreibe den nächsten Teil der if-Anweisung wie folgt:

```
*new neverninternights script x
1 void main()
2 (
3     /*
4     Testowanie flag
5     */
6
7     if(GetDialogFlagValue("posiada_zlecenie") == 0)
8     (
9         PrintString("Flaga posiada_zlecenie = 0");
10    )
11    |
12 )
```

Die Funktion „PrintString“ schreibt den Text, der als Argument übergeben wird, in das Aurora-Log. Die ganze Zeile wie folgt gelesen werden: wenn das Flag mit dem Namen "orders_received" den Wert 0 hat, dann schreibe den Text "Flag orders_received = 0" in das „Event registry“ Fenster. Leicht, oder? Nun schreiben wir noch eine „if“ Anweisung für den Fall, dass das Flag „orders_received“ gleich 1 ist:

```
*new neverninternights script x
1 void main()
2 (
3     /*
4     Testowanie flag
5     */
6
7     if(GetDialogFlagValue("posiada_zlecenie") == 0)
8     (
9         PrintString("Flaga posiada_zlecenie = 0");
10    )
11    if(GetDialogFlagValue("posiada_zlecenie") == 1)
12    (
13        PrintString("Flaga posiada_zlecenie = 1");
14    )
15
16 )
```

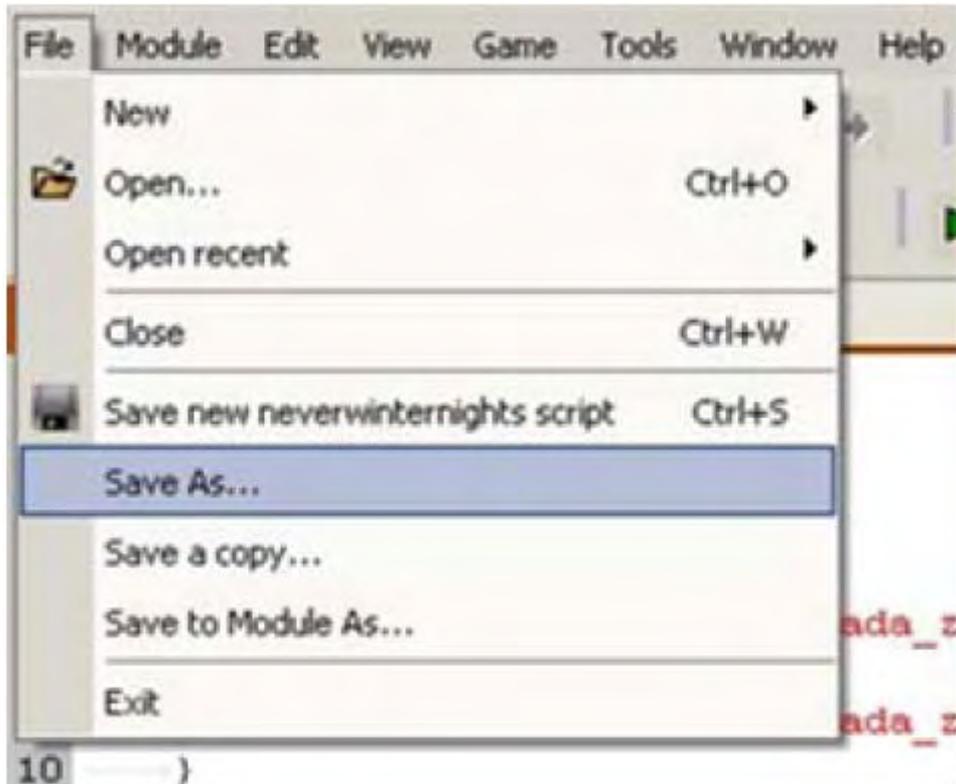
Dasselbe Script kann auch die anderen beiden Flags überprüfen. Kopiere den ganzen Block Code und ersetze „orders_received“ mit den anderen Namen:

```
*new neverwinternights script
1 void main()
2 (
3   /*
4   Testowanie flag
5   */
6
7   if(GetDialogFlagValue("posiada_zlecenie") == 0)
8   (
9     PrintString("Flaga posiada_zlecenie = 0");
10  )
11  if(GetDialogFlagValue("posiada_zlecenie") == 1)
12  (
13    PrintString("Flaga posiada_zlecenie = 1");
14  )
15
16  if(GetDialogFlagValue("herszt_zabity") == 0)
17  (
18    PrintString("Flaga herszt_zabity = 0");
19  )
20  if(GetDialogFlagValue("herszt_zabity") == 1)
21  (
22    PrintString("Flaga herszt_zabity = 1");
23  )
24
25  if(GetDialogFlagValue("raport_zdany") == 0)
26  (
27    PrintString("Flaga raport_zdany = 0");
28  )
29  if(GetDialogFlagValue("raport_zdany") == 1)
30  (
31    PrintString("Flaga raport_zdany = 1");
32  )
33  |
34 )
```

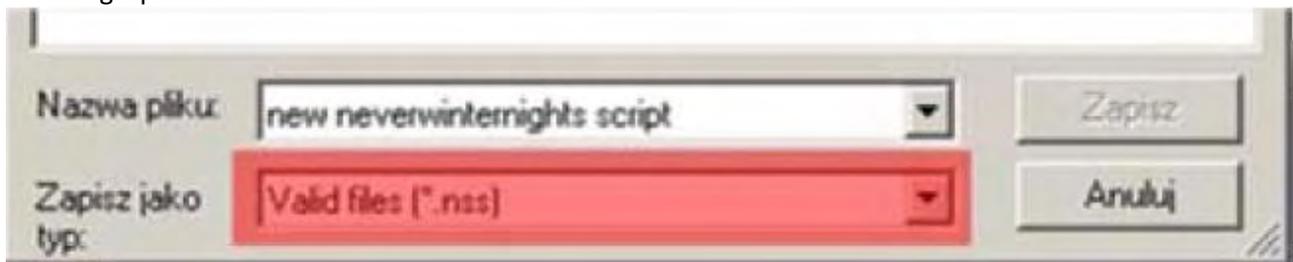
Nun noch zu etwas anderem. Jeder der alt genug ist, um Programmieren aus Büchern zu lernen, wird sich vielleicht erinnern, dass frühere Computerprogramme nur Text ausgaben, und der Text „Hello World“ war eine allgemeine Einführung in das Programmieren. Wir wollen also den alten Sitten treu bleiben und fügen am Ende des Scripts noch folgendes hinzu:

```
33
34 PrintString("Hello World! Hello Witcher! Hello D'jinni!");
35
36 )
```

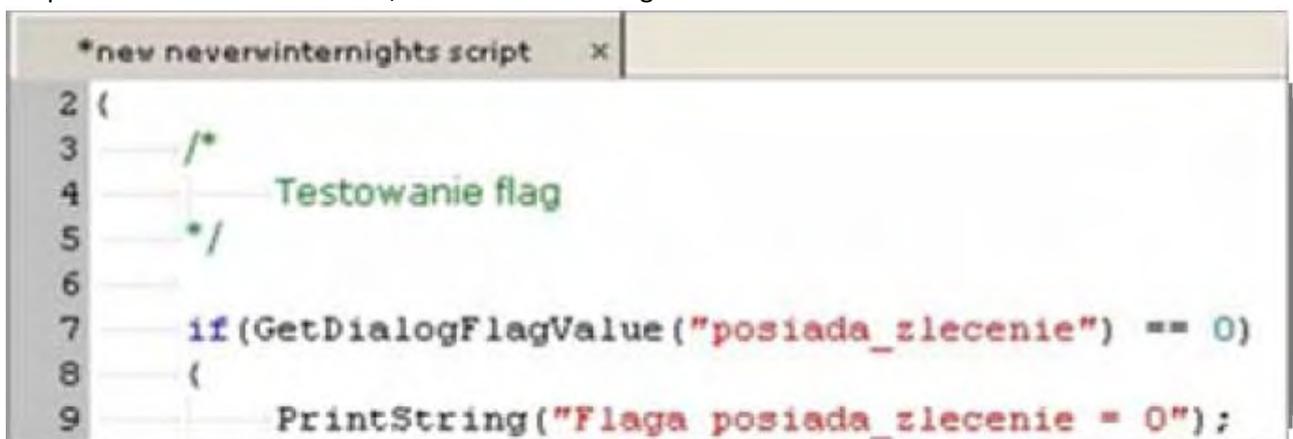
Das ist vielleicht ein wenig länger, aber es hat auch mehr Stil. Wähle nun „Save As...“ aus dem Menü „File“:



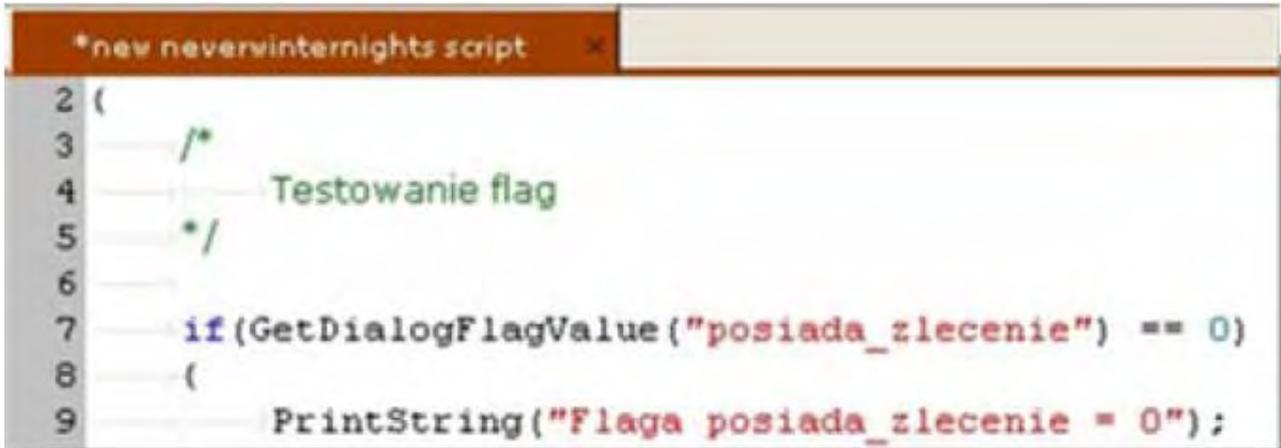
Es öffnet sich das übliche Windows-Fenster zum Speichern von Dateien. Gehe zu dem Ordner „Data\Scripts“ (wo alle Scripts gespeichert werden), gib den Namen „test_flag“ ein und klicke auf „Save“. Denk daran, dass der Dateiname höchstens 16 Zeichen lang sein darf. Vergewissere Dich, dass die Datei im „.nss“ Format gespeichert wird:



Wird ein anderer Dateityp aufgelistet, versuchst Du wahrscheinlich versehentlich, den Inhalt eines anderen geöffneten Fensters zu speichern. Schau vor dem Speichern nach, ob das Fenster „new neverwinternights script“ aktiv ist. Ist es nicht aktiv, so ist die Titelleiste grau:



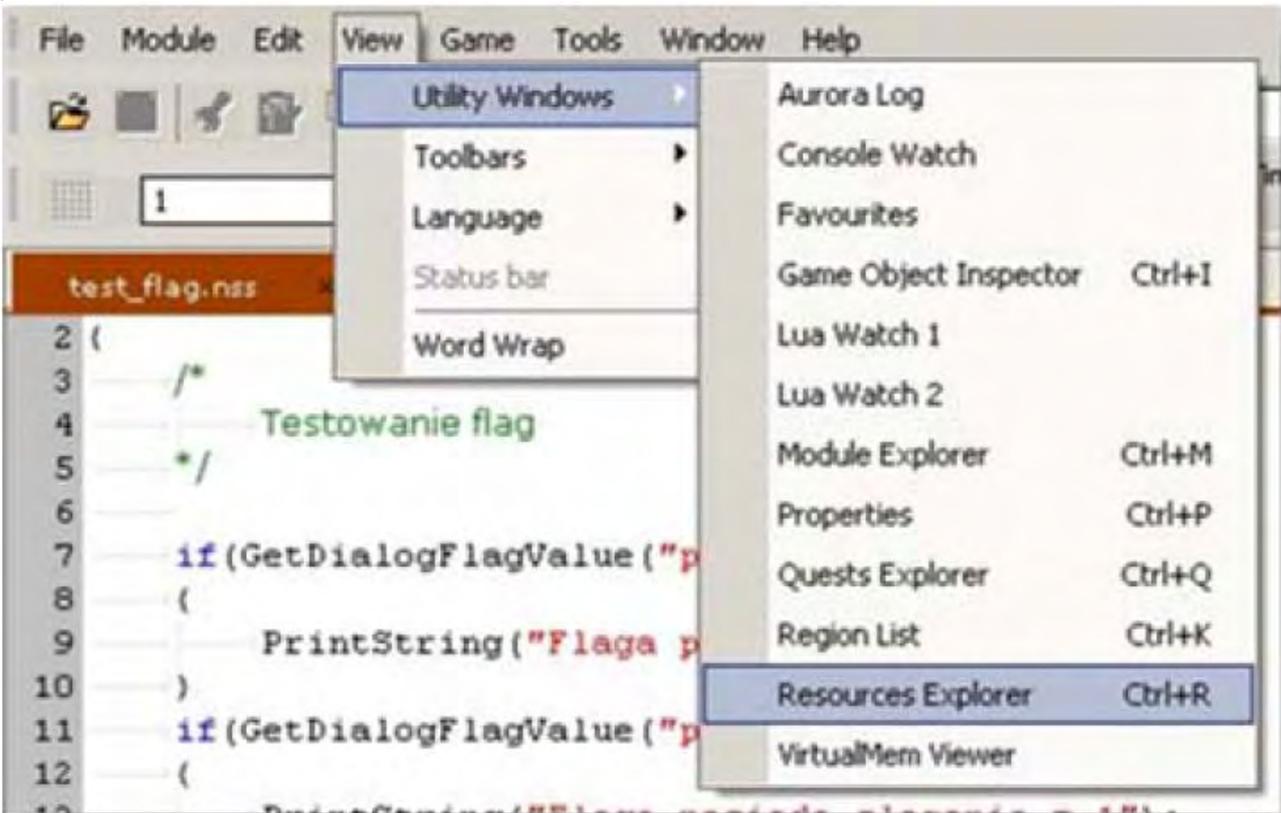
Eine andere Farbe bedeutet, dass das Fenster aktiv ist:



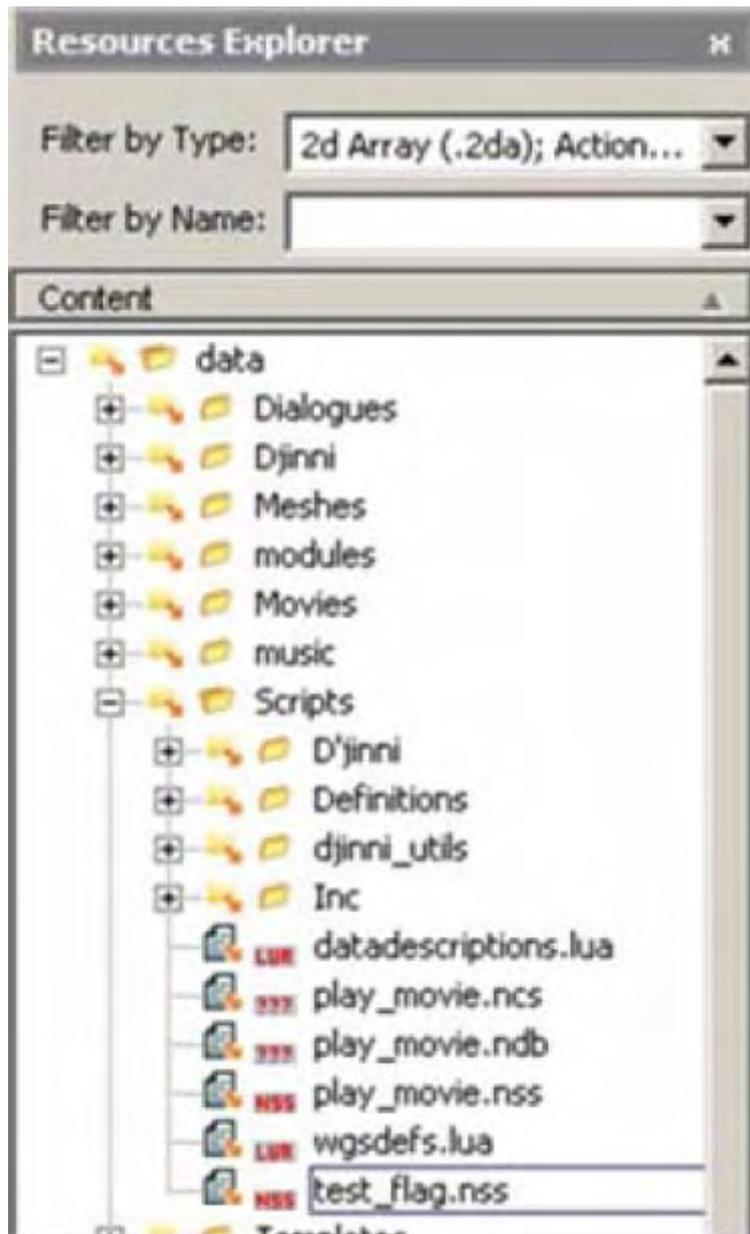
```
2 (
3  /*
4  Testowanie flag
5  */
6
7  if(GetDialogFlagValue("posiada_zlecenie") == 0)
8  (
9  PrintString("Flaga posiada_zlecenie = 0");
```

Ein Sternchen vor dem Namen in der Titelleiste zeigt an, dass das Script seit der letzten Änderung noch nicht gespeichert wurde. Wir erinnern uns, dass man Scripts immer in einzelnen Modulen speichern kann. Um dies zu tun, wähle „Save new neverwinternights scripts“ im Menü „File“, dann wähle "Modul" im Fenster „Save File“.

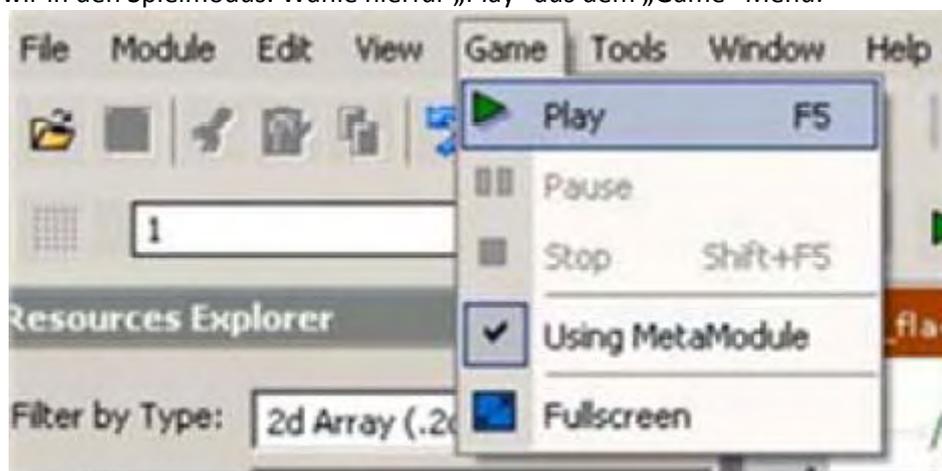
Nach dem Speichern der Datei sollten wir uns noch vergewissern, dass beim Kompilieren des Scripts keine Fehler aufgetreten sind. Öffne den „Resources Explorer“. Wähle dazu „Utility Windows => Resources Explorer“ aus dem Menü „View“:



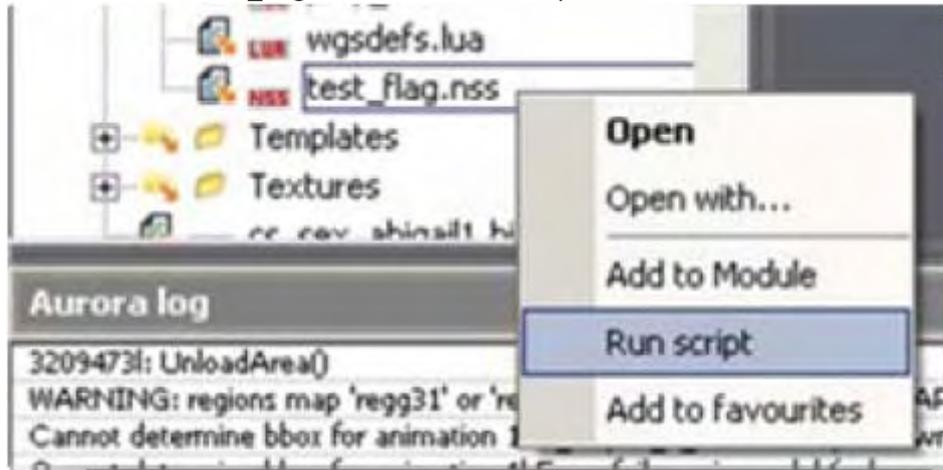
Klappe im Fenster "Resources" die Bäume "Data" und "Scripts" aus. Das Script "test_flag.nss" sollte jetzt zu sehen sein:



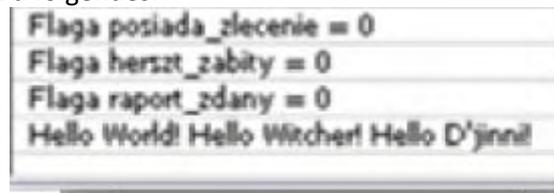
Nun schalten wir in den Spielmodus. Wähle hierfür „Play“ aus dem „Game“ Menü:



Nach dem schalten in den Spielmodus, wird das Modul geladen. Ist es geladen, gehe in den „Resources Explorer“ und rechtsklicke auf test_flag.nss. Wähle „Run script“ aus der Auswahlliste:



Im Aurora Log Fenster siehst Du folgendes:



Wie Du sehen kannst, haben alle Flags den Wert 0. Nun nehme die Quest von Jethro und lade das Skript erneut. Töte den Banditenanführer und starte das Skript erneut. Du siehst die Änderungen. So, wie Du sehen kannst, machen Skripte das Leben leichter.

[HILFE BEIM SKRIPTEN](#)

Die Skriptsprache in The Witcher ist exakt die gleiche wie in „Neverwinter Nights“. Dennoch sind die zwei Sprachen wenn es um Funktionen, oder den Weg um Funktionen aufzurufen geht unterschiedlich. Wenn Du wirklich ernsthaft mit D`jinni Skripten willst, solltest Du ein wenig im „Neverwinter Nights“ Editor üben. Die meisten Funktionen sind in der Praxis gleich.

Auch solltest Du einen Blick auf die Guides und Anleitungen über den NWN Editor werfen die Online zu finden sind. Das NWN Lexikon ist zu empfehlen und Du findest es unter <http://www.nwnlexicon.com/>. Alle Funktionen die in NWN benutzt werden sind hier zu finden.

Anhang

Einleitung:

Dieses Kapitel handelt um eine anderes mögliches Element in TW, welches in den vorherigen Kapiteln noch nicht drangenommen wurde. Wir werden hier zeigen wie man einen Lagerraum (storage area), einen Laden (wo man tolle Sachen kaufen kann) und einen Rastplatz (oder einen Meditationsplatz) erstellt.

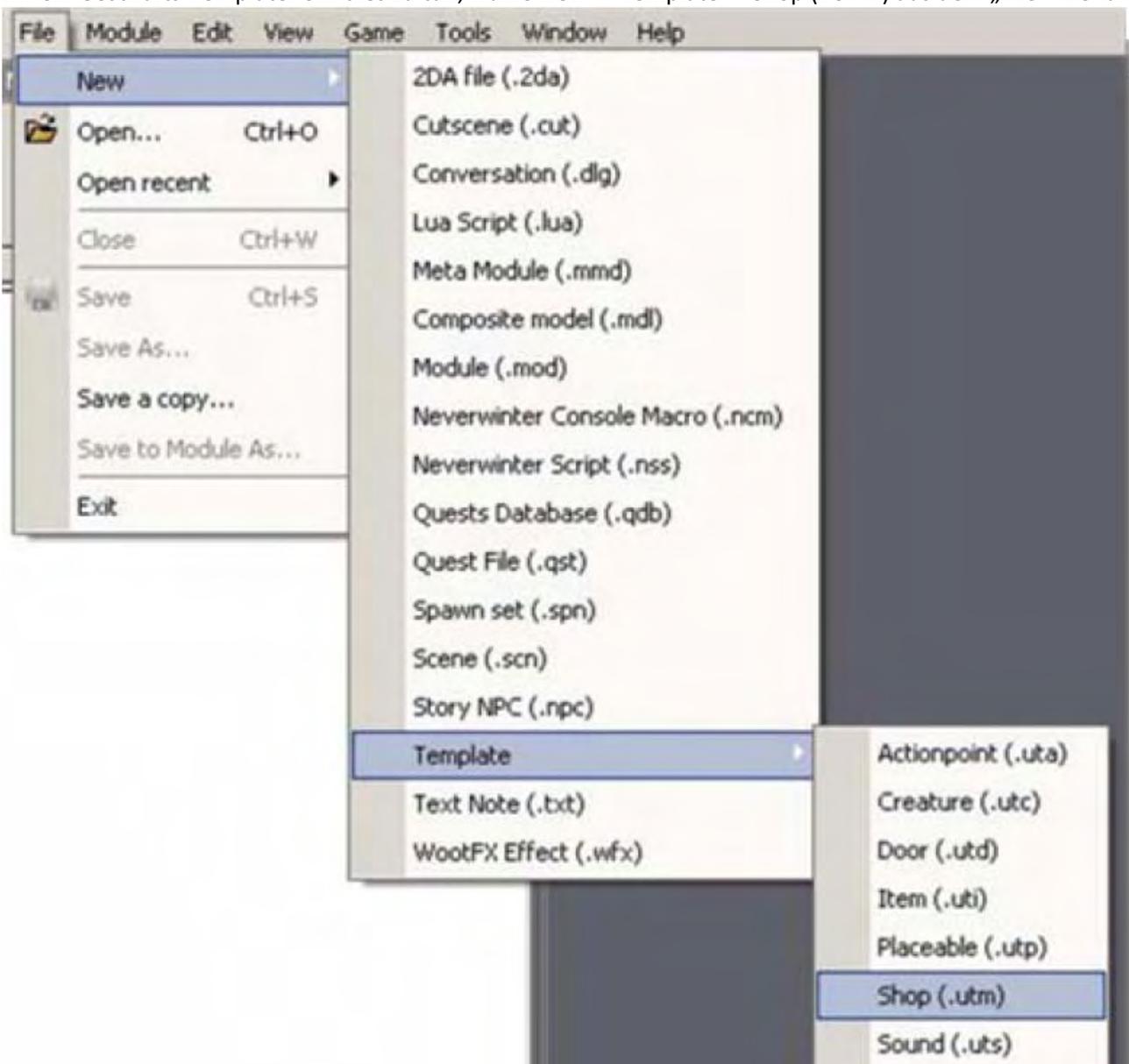
Da alle drei Elemente in Gesprächen mit anderen Charakteren möglich sind, werden wir unser zuvor erstellte Gespräch mit Jethro benutzen. Wenn Du irgendwelche Probleme in diesem Kapitel hast, kannst Du immer zurück zum vorherigen Abschnitt und dort nach Lösungen suchen.

Lager:

Ein Lager ist, wo Geralt seine nicht benötigten Sachen lagern kann, die er nicht im Inventar braucht. Es ist im Grunde eine Art Schrank.

Es gibt zwei Arten von Lagern - eins in Form einer Truhe, die andere in Form eines NPC mit Lagerungs Fähigkeiten, z.B. ein Gastwirt.

Hier werden wir einen Blick auf die letztere Möglichkeit werfen. Wir wollen, dass Geralt Sachen bei Jethro hinterlegen kann. Die Schaffung von einem Lagerplatz basiert auf der Grundlage eines Geschäfts. Erstellen wir ein Geschäfts-Template. Um dies zu tun, wähle New -> Template -> Shop (. UTM) aus dem „File“-Menü:



Das „new shop template“ Fenster erscheint auf der rechten Seite vom Bildschirms:

The screenshot shows a window titled "new shop template" with a green header. It contains two main sections: "General" and "Profiles".

General	
Name	
Tag	
Description	
Comment	
Is Storage	False
Profiles	
Profile - Weapon	Default
Profile - Armor	Default
Profile - Food	Default
Profile - Alcohol	Default
Profile - WitherAlchemy	Default
Profile - Ingredient	Default
Profile - Value	Default
Profile - Book	Default
Profile - Illegal	Default
Profile - Trophy	Default
Profile - MonsterIngredient	Default
Profile - SilverSword	Default
Profile - Dice	Default
Profile - WeaponUp	Default
Profile - Flower	Default

Diese Fenster beinhaltet die Attribute von dem "shop" Template. In der „General Section“ setze das „Is Storage“ Attribut auf „True“. Klicke einfach in der rechten Spalte gegenüber von „Is Storage“:

This close-up shows the "Is Storage" attribute in the "General" section. The value is currently "False" and is being changed to "True".

General	
Name	
Tag	
Description	
Comment	
Is Storage	False
Profiles	False
Profile - Weapon	True

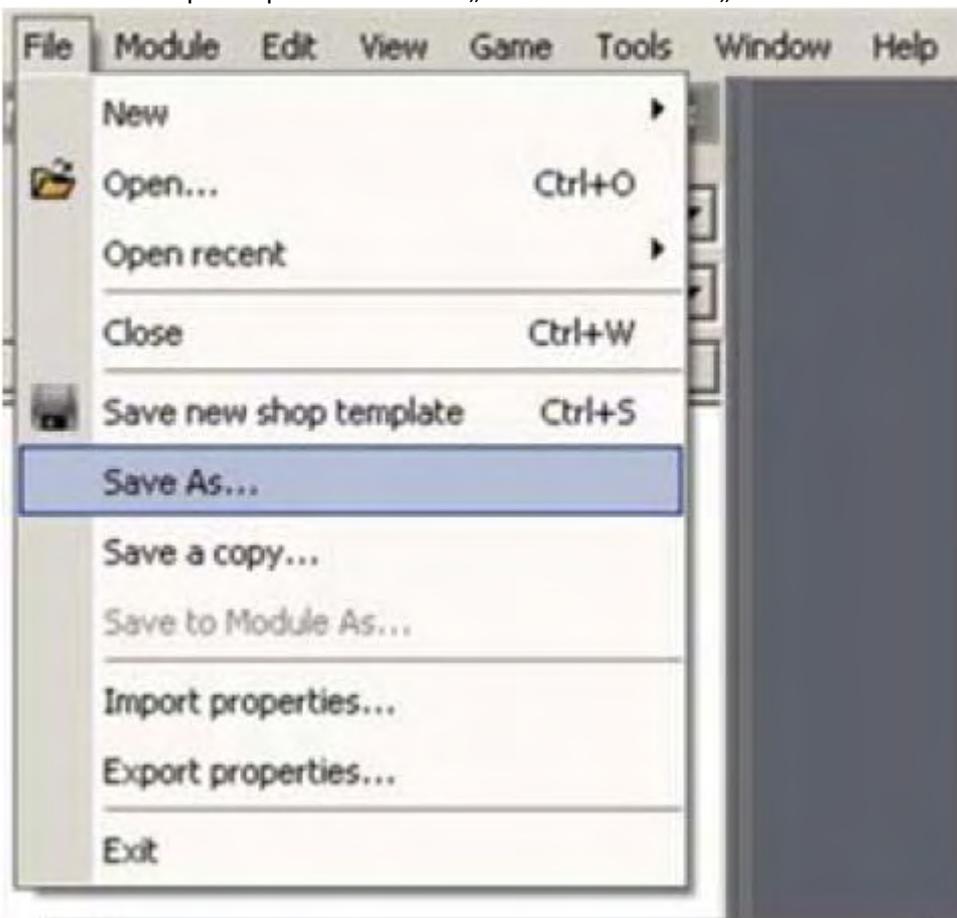
Diese Einstellung zeigt uns an, was wir erstellen. Erstellen wir einen Lagerplatz (Is Storage = True) oder ein Geschäft (Is Storage = false). Wir sollten auch dem Lagerplatz einen Namen geben. Klicke einfach auf der rechten Spalte gegenüber von Name und tippe „Warehouse“ ein und drücke ENTER:



Nun füge einige Items hinzu. Klicke auf das schmale rote Symbol auf der rechten Seite gegenüber von „Inventory contents“ in der „Attributes section“:

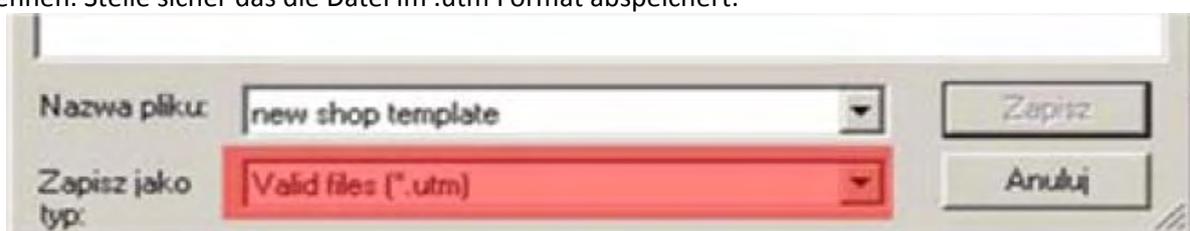


Die Grundlagen zum hinzufügen von Items zum Charakter Zubehör wurde im vorherigen Kapitel besprochen. Lasse uns nun das Template speichern. Wähle „Save As...“ aus dem „File“ Menü:

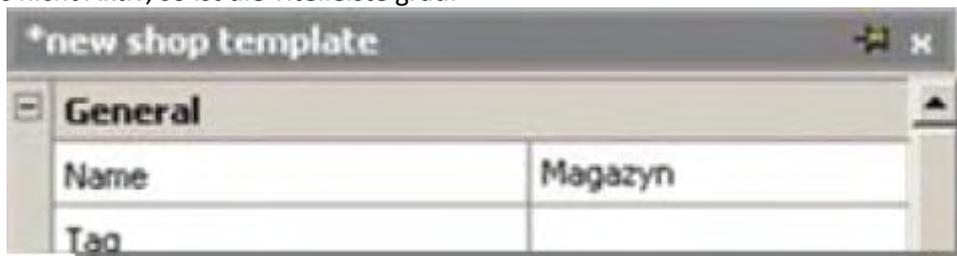


Es öffnet sich ein Windows Fenster zum Speichern von Dateien. Gehe zum „Data\Templates\Shops“ Ordner (dort sind alle Laden und Lagerplatz Templates gespeichert), vergebe „our_warehouse“ als Namen und Klicke auf „Save“.

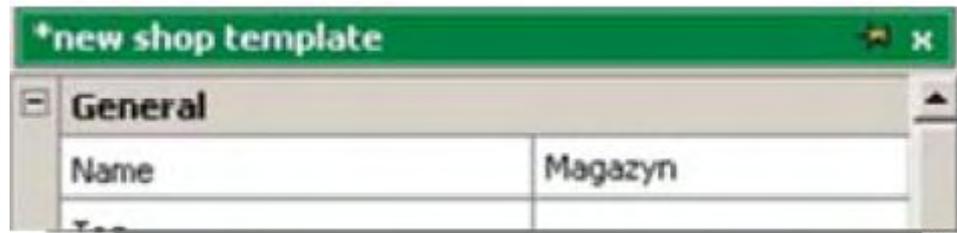
Vergess bitte nicht das nicht mehr als 16 Zeichen erlaubt sind. Andernfalls wird der D’jinni Editor sie nicht erkennen. Stelle sicher das die Datei im .utm Format abspeichert:



Wird ein anderer Dateityp aufgelistet, versuchst Du wahrscheinlich die Datei über ein anderes geöffnetes Fenster zu speichern. Vor dem Speichern solltest Du sicher gehen das das „new shop template“ Fenster aktiv ist. Ist es nicht Aktiv, so ist die Titelleiste grau:

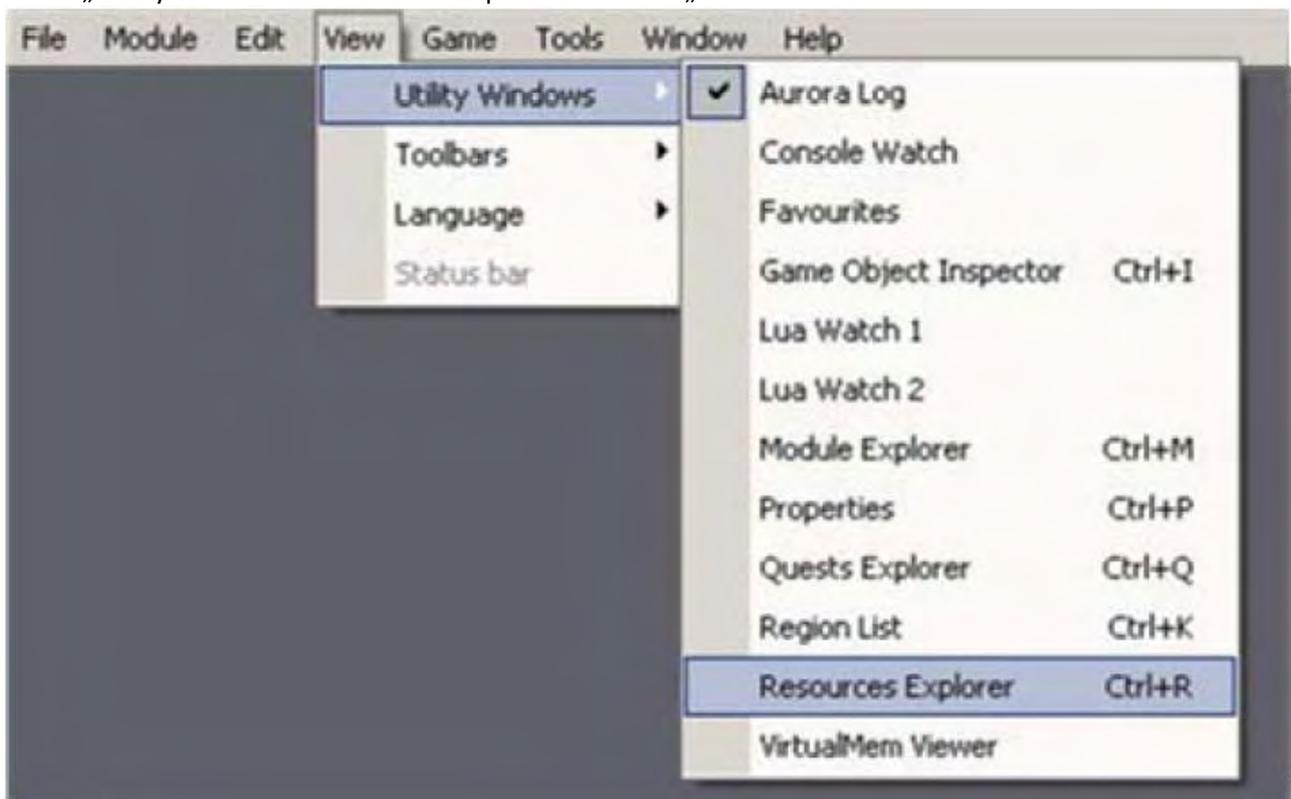


Eine andere Farbe deutet an das das Fenster aktiv ist:

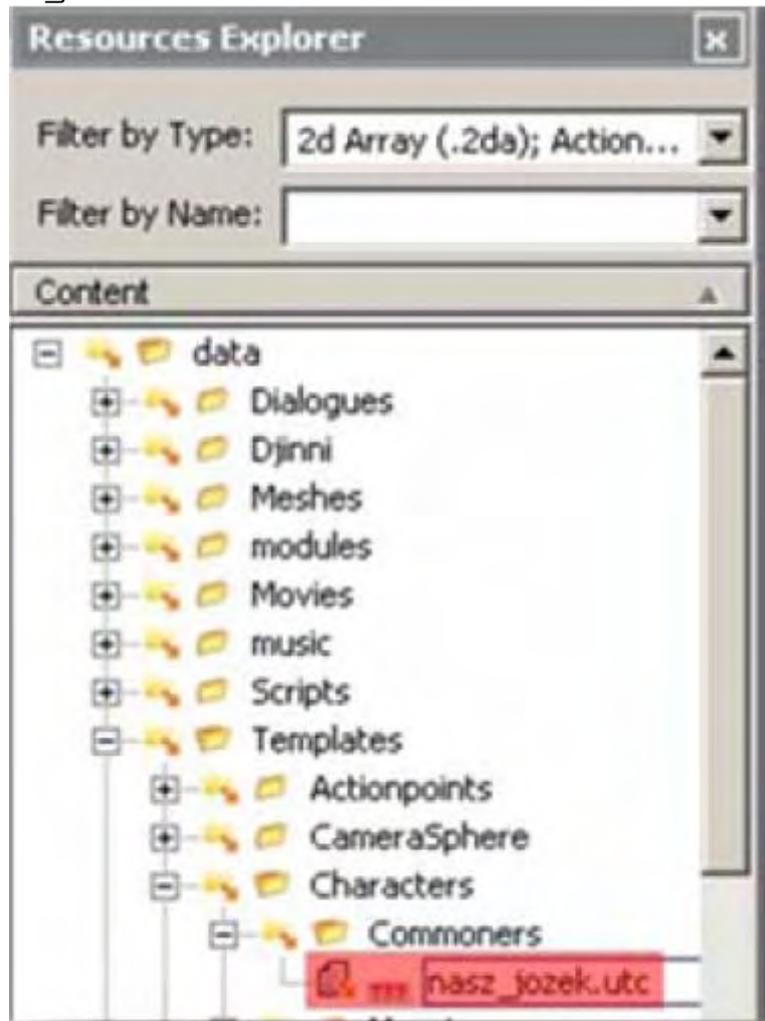


Ein Sternchen vor dem Namen in der Titelleiste zeigt an das das Skript seit der letzten Änderung noch nicht gespeichert wurde.

Jetzt müssen wir noch das Template für den Laden (eher den Lagerplatz) einem Charakter Template hinzufügen. Wie vorher erwähnt, benutzen wir den Jethro Charakter dafür. Öffne den „Resources Explorer“ und wähle „Utility Windows => Resources Explorer“ aus dem „View“ Menü:



Im „Resources Explorer“ Fenster wählst Du „Data => Templates => Character => Commoners“ und doppelklickst auf die Datei „our_jethro.utc“:



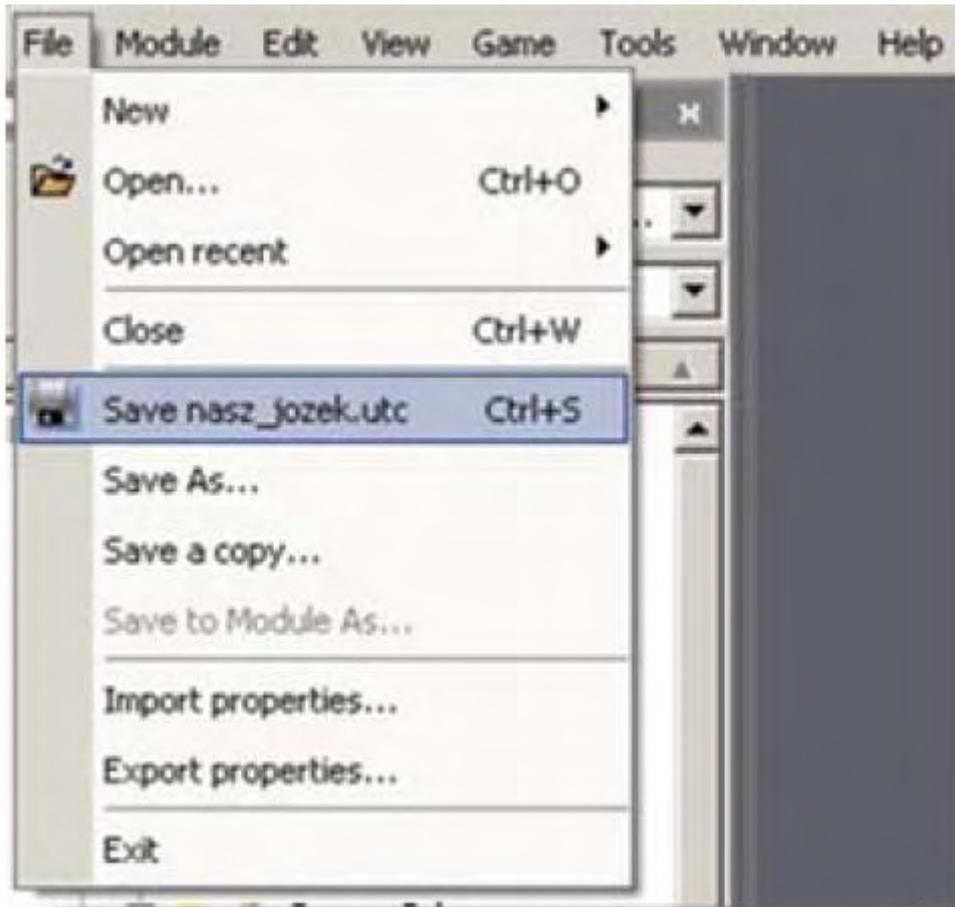
Ein Fenster das die Attribute vom Charakter Template beinhaltet öffnet sich auf der rechten Seite vom Bildschirm. Klicke auf das schmale gelbe Symbol in der rechten Spalte gegenüber vom „Store“ Attribut in der „Special Section“:



Es öffnet sich ein Windows Fenster zum laden von Dateien. Gehen Sie zum „Data\Templates\Shops“ Ordner und wählen Sie die „our_warehouse.utm“ Datei aus. Klicken Sie auf „Open“. Das Laden Template wird nun dem Charakter Template hinzu gefügt:



Wir müssen nun noch die Änderungen am Jethro template speichern. Klicken Sie auf „Save our_jethro.utc“ im „File“ Menü:

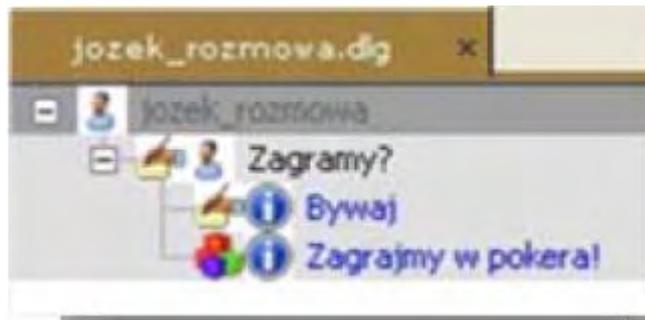


Wenn „Save our_jethro.utc“ nicht angezeigt wird, oder ein anderer Name dort steht, ist das „our_jethro.utc“ Fenster nicht aktiv. Zum aktivieren klicke auf die Titelleiste vom „Our_jethro.utc“ Fenster und wähle dann „Save our_jethro.utc“ aus dem „File“ Menü.

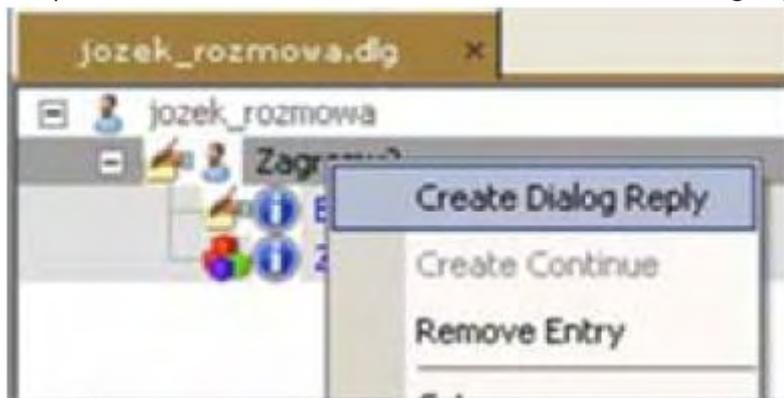
So, nun müssen wir noch, um die Option zu wählen, um den Laden während Geralt's Gespräch mit Jethro zu benutzen. Wir verwenden die Ressourcen Explorer erneut. Wähle Data -> Dialoge und doppelklicke dann auf die Datei jethro_conv.dlg:



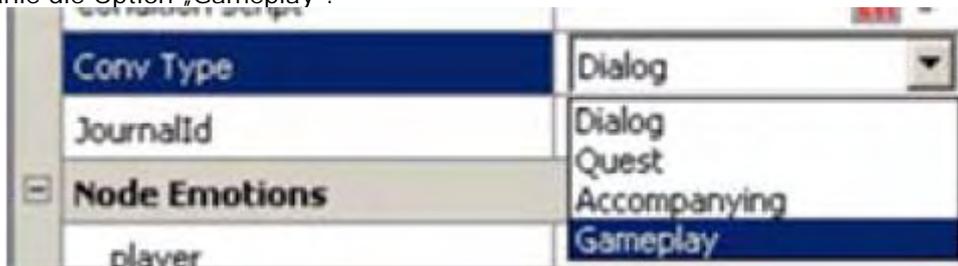
Das „jethro_conv.dlg“ Fenster erscheint:



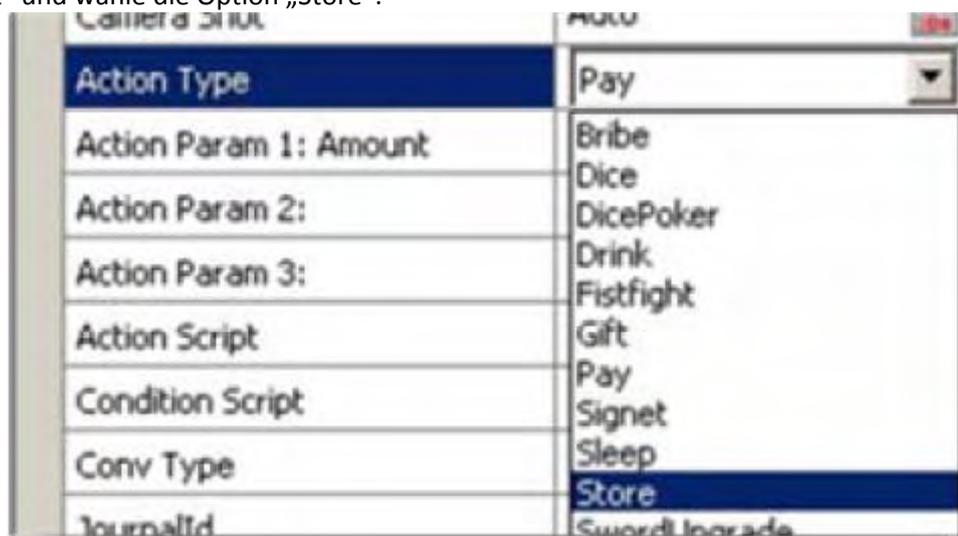
Wähle nun "Wollen wir spielen?" aus und rechtsklicke drauf. Wähle „Create Dialog Reply“:



Dies erzeugt einen leeren Eintrag. Linksklicke auf ihn und gehe zum Fenster "Eigenschaften" zu Attribute und Konversation. Klicke in der rechten Spalte gegenüber dem „Conv Type“ Attribut im Abschnitt „Node Text“ und wähle die Option „Gameplay“:



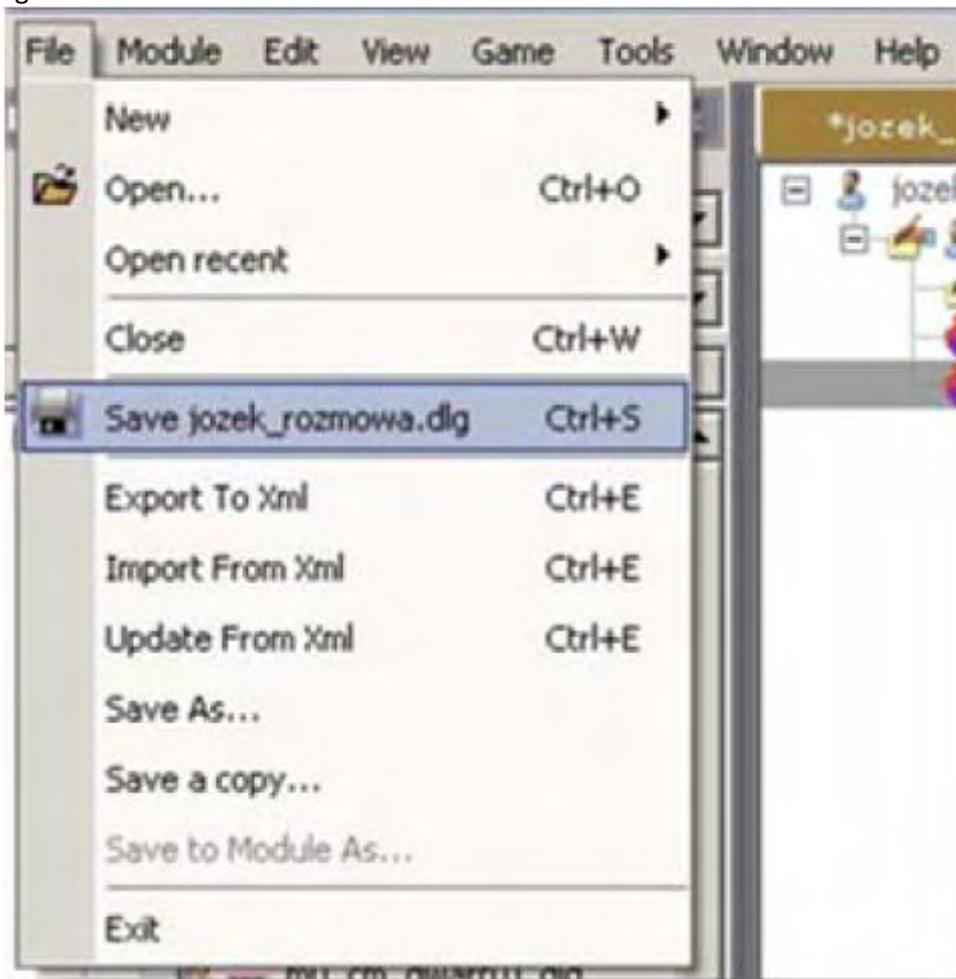
Ein neues Attribut erscheint im Abschnitt „Node Text“. Klicke auf der rechten Spalte gegenüber der Aktion „Typ-Attribut“ und wähle die Option „Store“:



Der leere Eintrag wird automatisch zu "Ich will etwas hinterlegen":



Alles, was jetzt noch fehlt ist die Änderungen in Jethro-Gespräch zu speichern. Um dies zu tun, wähle „Save jethro_conv.dlg“ aus dem File-Menü:



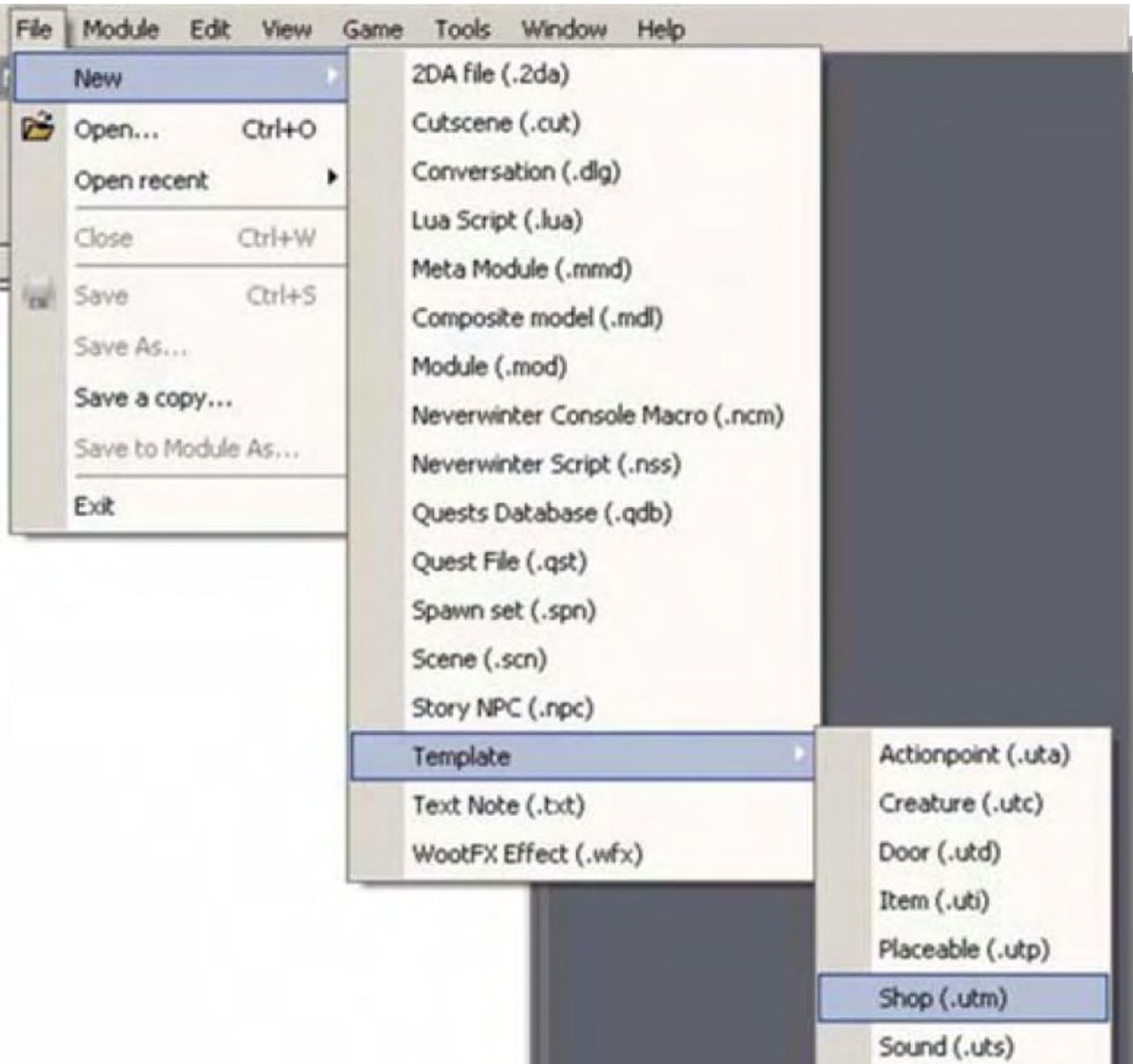
Wenn jethro_conversation.dlg nicht angezeigt wird, oder ein anderer Dateiname angezeigt wird, bedeutet, dass das „jethro_conv.dlg“ - Fenster nicht aktiv ist in diesem Moment. Um es zu aktivieren, klicke einfach auf die Titelleiste des „jethro_conv.dlg“ Fenster und wähle dann „Save our_jethro.dlg“ aus dem Menü "Datei".

Das war's schon. Nun, wenn wir wollen, können wir mit Jethro sprechen. Wir werden nicht nur in der Lage sein mit ihm Würfel zu spielen, sondern auch können auch bestimmte Sachen bei ihm hinterlegen.

Geschäft:

Dieses Unterkapitel befasst sich mit der Schaffung eines Shops. Im Wesentlichen ist die Erstellung eines Geschäfts nicht viel anders als die Erstellung eines Lagers.

Lasse uns ein neues Geschäfts-Template erzeugen. Um dies zu tun, wähle New -> Template -> Shop (. UTM) aus dem File-Menü:



Auf der rechten Seite des Bildschirms öffnet sich das Fenster *new shop template*:

The screenshot shows a window titled "new shop template" with two main sections: "General" and "Profiles".

General	
Name	
Tag	
Description	
Comment	
Is Storage	False

Profiles	
Profile - Weapon	Default
Profile - Armor	Default
Profile - Food	Default
Profile - Alcohol	Default
Profile - WitherAlchemy	Default
Profile - Ingredient	Default
Profile - Value	Default
Profile - Book	Default
Profile - Illegal	Default
Profile - Trophy	Default
Profile - MonsterIngredient	Default
Profile - SilverSword	Default
Profile - Dice	Default
Profile - WeaponUp	Default
Profile - Flower	Default

Das Fenster enthält die Attribute des Ladens. Setze im Abschnitt *General* das Attribut *Is Storage* auf *false*. Klicke dazu auf die rechte Spalte neben dem Attribut *Is Storage*:

This close-up shows the "General" section with the "Is Storage" attribute highlighted. The value "False" is selected in a dropdown menu. Below it, the "Profiles" section is partially visible, showing the "Profile - Weapon" attribute with a dropdown menu containing "False", "True", and "Default".

General	
Name	
Tag	
Description	
Comment	
Is Storage	False

Profiles	
Profile - Weapon	False
	True
	Default

Nachdem Du die gewünschten Gegenstände hinzugefügt hast, klicke im Fenster *Inventory Contents* auf OK. Schau Dir jetzt die Attribute im Abschnitt *Profiles* an:



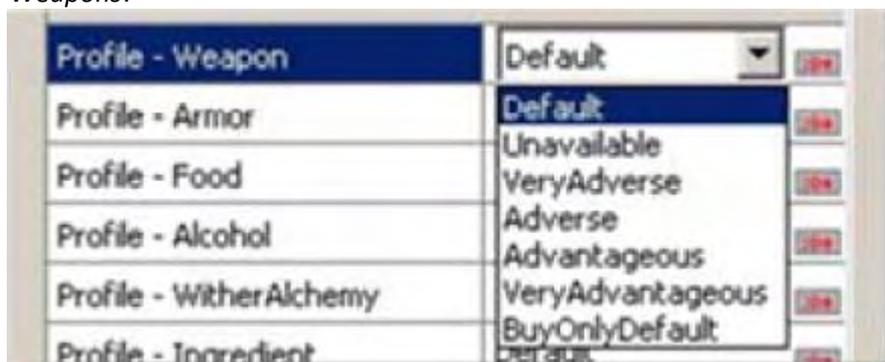
Profiles	
Profile - Weapon	Default 
Profile - Armor	Default 
Profile - Food	Default 
Profile - Alcohol	Default 
Profile - WitherAlchemy	Default 
Profile - Ingredient	Default 
Profile - Value	Default 
Profile - Book	Default 
Profile - Illegal	Default 
Profile - Trophy	Default 
Profile - MonsterIngredient	Default 
Profile - SilverSword	Default 
Profile - Dice	Default 
Profile - WeaponUp	Default 
Profile - flower	Default 
Profile - cloth	Default 
Profile - torch	Default 
Profile - powder	Default 
Profile - grease	Default 
Profile - herb	Default 
Profile - AlcoholStrong	Default 
Profile - Alchemy	Default 
Profile - Candy	Default 

Diese Attribute werden dazu verwendet, um Preisoptionen für die verschiedenen Gegenstände zu setzen. Gegenstände werden in The Witcher in 23 Profile eingeteilt, von denen jedes zu den Preisoptionen eines bestimmten Gegenstands gehört. Folgende Profile gibt es:

- *Profile – Weapon* - die Preisoptionen für Waffen,
- *Profile – Armor* - die Preisoptionen für Rüstungen,
- *Profile – Food* – die Preisoption für Nahrung,
- *Profile – Alcohol* die Preisoption für Alkohol,

- *Profile – WitcherAlchemy* - die Preisoptionen für Alchemieobjekte von Hexern
- *Profile – Ingredient* – die Preisoptionen für Zutaten,
- *Profile – Value* – die Preisoptionen für wertvolle Gegenstände,
- *Profile – Book* – die Preisoptionen für Bücher,
- *Profile – Illegal* – die Preisoptionen für illegale Gegenstände,
- *Profile – Trophy* – die Preisoptionen für Trophäen,
- *Profile – MonsterIngredients* – die Preisoptionen für Zutaten, die von besiegten Monstern gewonnen wurden,
- *Profile – SilverSword* – Preisoptionen für Silberschwerter,
- *Profile – Dice* – die Preisoptionen für Würfel,
- *Profile – WeaponUp* – die Preisoptionen für Waffenerweiterungen,
- *Profile – flower* – die Preisoptionen für Blumen,
- *Profile – cloth* – die Preisoptionen für Kleidung,
- *Profile – torch* – die Preisoptionen für Fackeln,
- *Profile – powder*-die Preisoptionen für Pulver, also Grundmaterial für Bomben,
- *Profile – grease* – die Preisoptionen für Fett, also Grundmaterial für Öle,
- *Profile – herb* – die Preisoptionen für Kräuter,
- *Profile – AlcoholStrong* – die Preisoptionen für starken Alkohol,
- *Profile – Alchemy* die Preisoptionen für Alchemieobjekte,
- *Profile – Candy* – die Preisoptionen für Süßigkeiten.

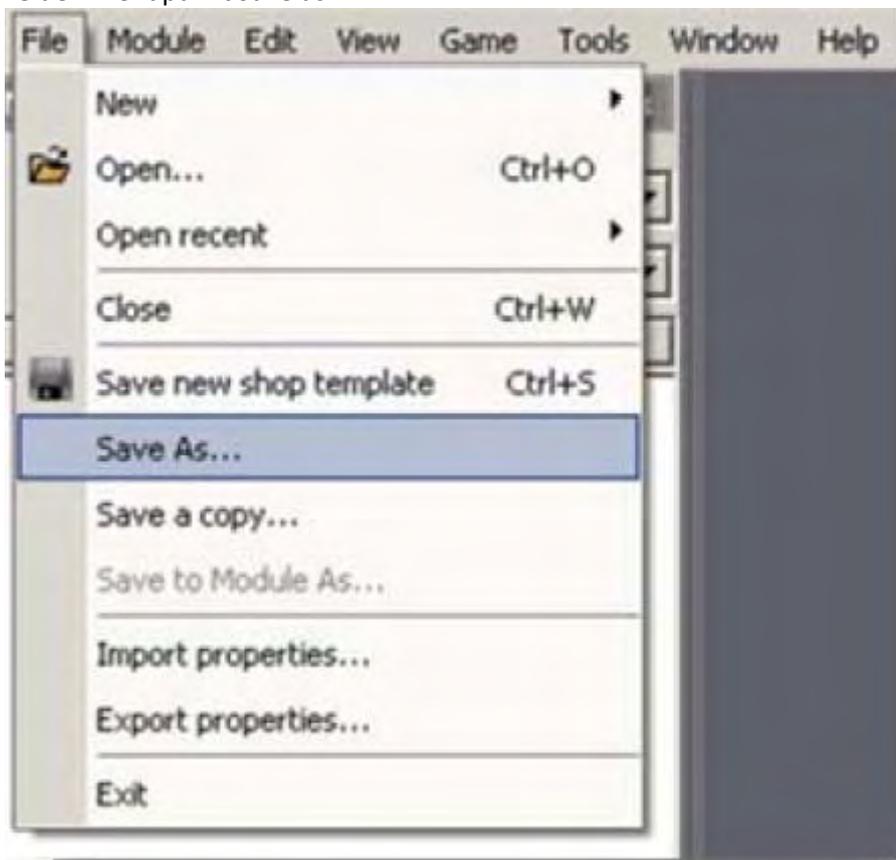
Um das Preisprofil eines Gegenstands zu setzen, klicke auf die rechte Spalte neben dem zugehörigen Profil. Wenn Du also z. B. die Preisoptionen für eine Waffe setzen willst, klicke auf die rechte Spalte neben dem Attribut *Profile - Weapons*:



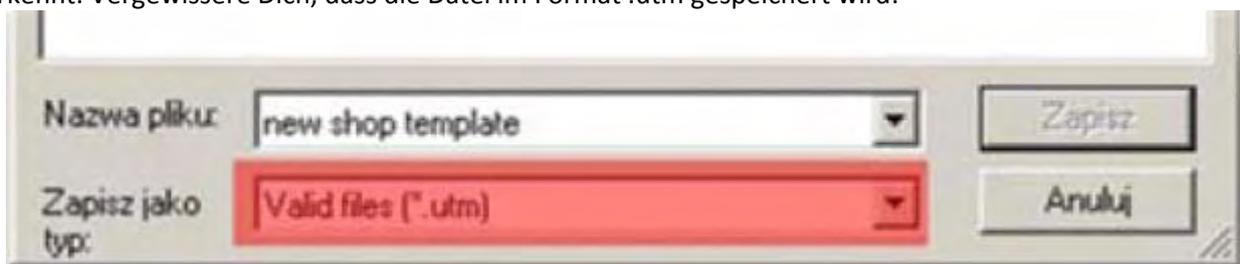
Wähle den passenden Wert aus. Folgende Werte stehen zur Verfügung:

- *Default* – Gegenstände werden zum vollen Preis verkauft und für 1/5 des Wertes gekauft,
- *Unavailable* – Gegenstände werden nicht gekauft,
- *VeryAdverse* – Gegenstände werden zum vollen Preis verkauft und für 1/100 des Werts gekauft,
- *Adverse* – Gegenstände werden zum vollen Preis verkauft und für 1/10 des Werts gekauft,
- *Advantageous* – Gegenstände werden zum vollen Preis verkauft und für die Hälfte des Werts gekauft,
- *VeryAdvantageous* – Gegenstände werden zum vollen Preis verkauft und für den vollen Wert gekauft,
- *BuyOnlyDefault* – Gegenstände werden zum vollen Preis verkauft, aber nicht zurückgekauft.

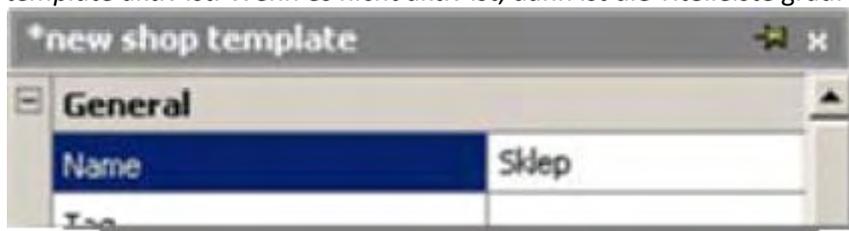
Nachdem wir die Preisoptionen gesetzt haben, müssen wir die Vorlage unseres Ladens speichern. Wähle dazu im Menü *File* den Menüpunkt *Save as...*:



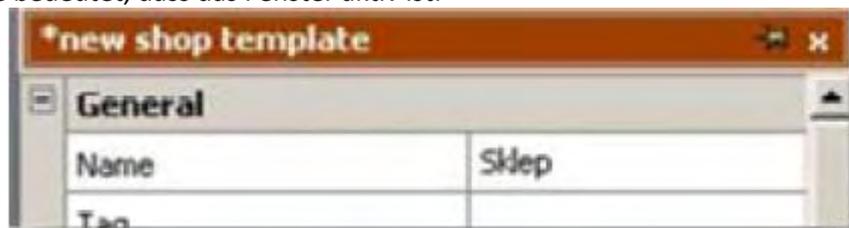
Das übliche Windows-Fenster zum Speichern von Dateien öffnet sich. Gehe zum Ordner *Data\Templates\Shops* (in dem alle Vorlagen für Läden gespeichert sind), gib "our_shop" ein und klicke auf *Speichern*. Denke daran, dass der Dateiname höchstens 16 Zeichen lang sein darf, da D'Jinni ihn sonst nicht erkennt. Vergewissere Dich, dass die Datei im Format *.utm* gespeichert wird:



Wenn stattdessen ein anderer Dateityp angezeigt wird, versuchst Du wahrscheinlich versehentlich, den Inhalt eines anderen geöffneten Fensters zu speichern. Vergewissere Dich vor dem Speichern, dass das Fenster *new shop template* aktiv ist. Wenn es nicht aktiv ist, dann ist die Titelleiste grau:

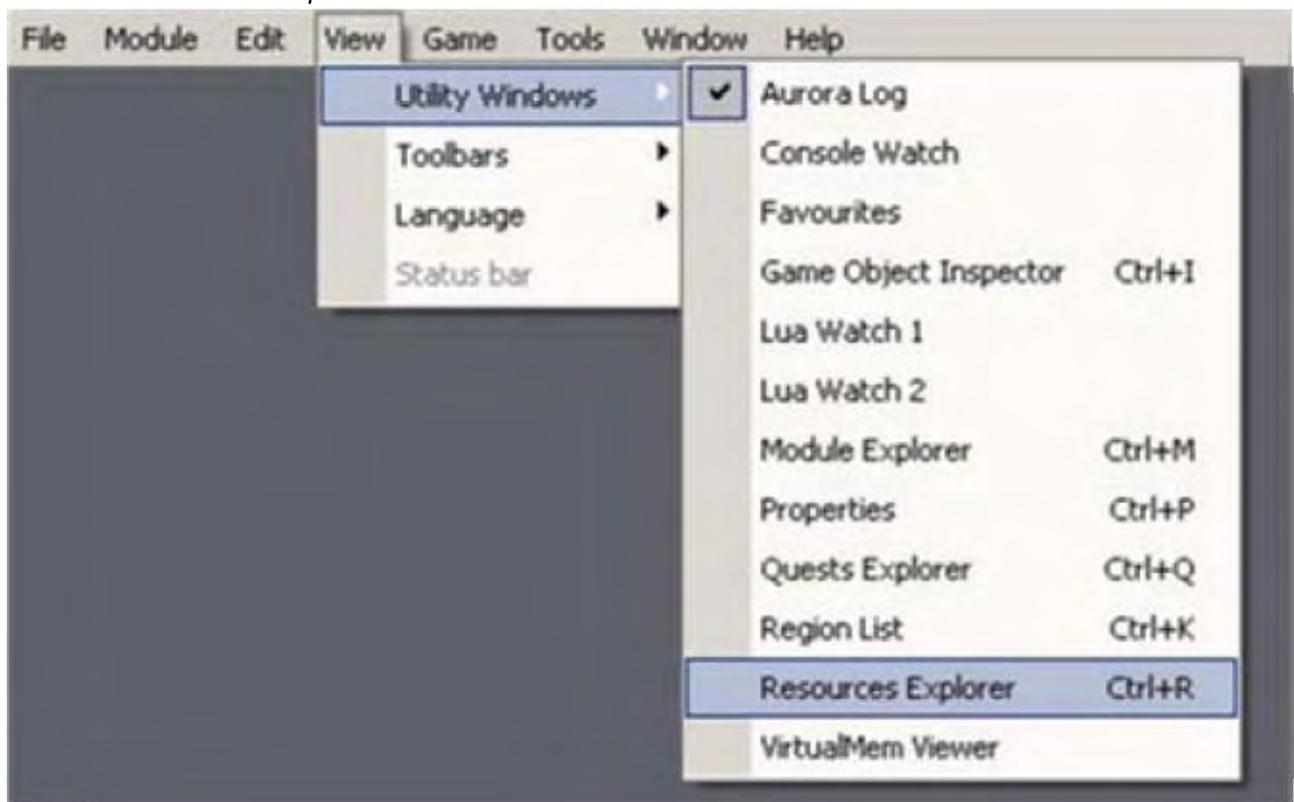


Eine andere Farbe bedeutet, dass das Fenster aktiv ist:

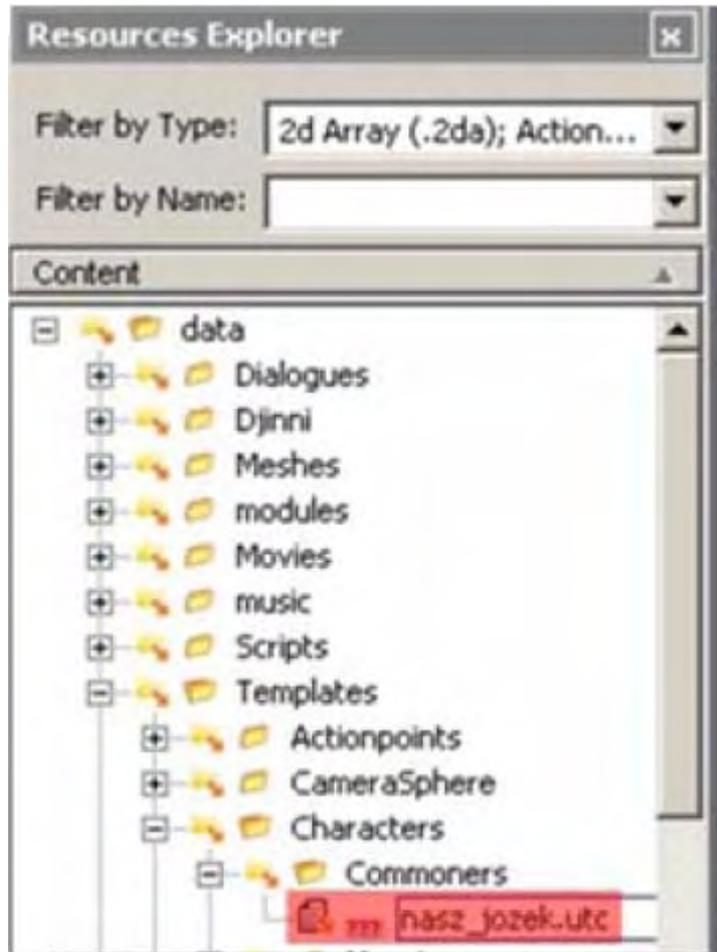


Ein Stern vor dem Namen in der Titelleiste bedeutet, dass die Datei seit der letzten Änderung nicht gespeichert wurde.

Jetzt müssen wir die Vorlage für das Lager noch der Vorlage des Charakters zuweisen. Auch dafür benutzen wir den Charakter Jethro. Öffne den Resources Explorer, indem Du im Menü *View* den Menüpunkt *Utility Windows -> Resources Explorer* wählst:



Wähle im Resources Explorer *Data -> Templates -> Character -> Commoners* und doppelklicke auf die Datei *our_jethro.utc*:



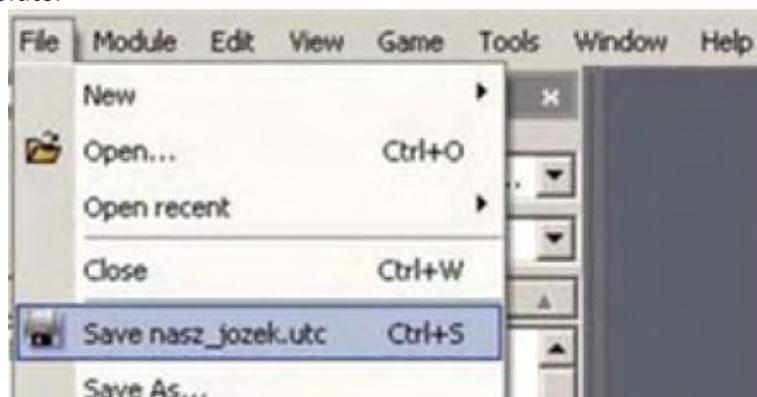
Auf der rechten Seite des Bildschirms öffnet sich ein Fenster mit dem Attributen der Charakter-Vorlage. Klicke auf das keine gelbe Icon in der rechten Spalte neben dem Attribut *Store* im Abschnitt *Special*:



Das übliche Windows-Fenster zum Laden von Dateien wird geöffnet. Gehe zum Ordner *Data\Templates\Shops* (wo wir die Vorlage für das Lager gespeichert haben), wähle die Datei *"our_warehouse.utm"* und klicke auf *Öffnen*. Die Vorlage für das Lager wird jetzt der Charaktervorlage zugewiesen:

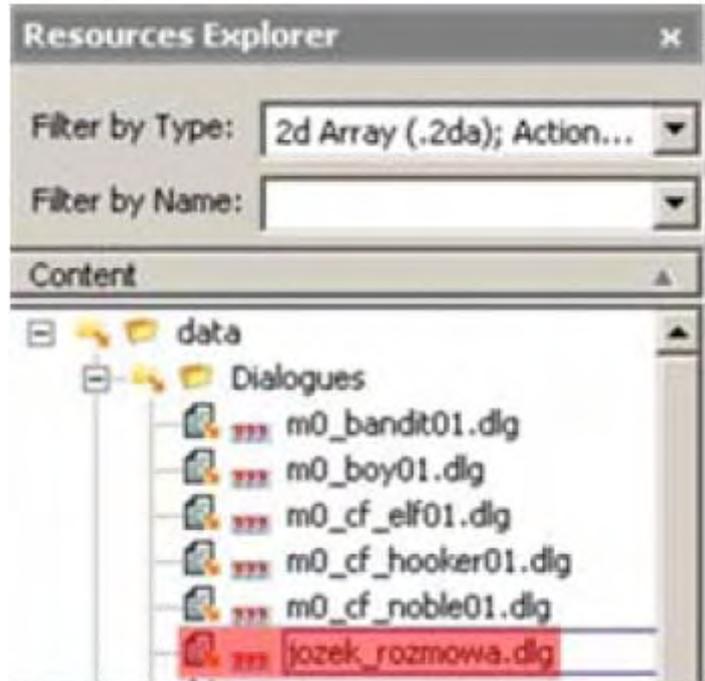


Jetzt müssen wir noch die Änderung in Jethro's Vorlage speichern. Wähle dazu im Menü *File* den Menüpunkt *Save our_jethro.utc*:



Wenn dieser Menüpunkt nicht im Menü *File* erscheint oder stattdessen ein anderer Dateiname angezeigt wird, dann ist das Fenster *our_jethro.utc* gerade nicht aktiv. Um es zu aktivieren, klicke auf die Titelleiste des Fensters, und wähle dann im Menü *File* den Menüpunkt *Save our_jethro.utc*.

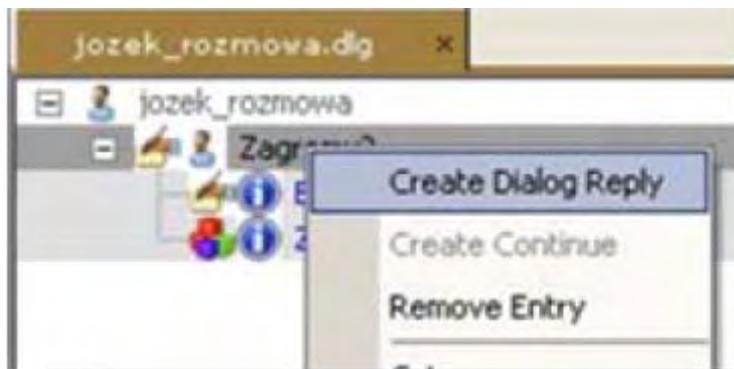
Jetzt müssen wir noch eine Option hinzufügen, die es Geralt erlaubt, während des Gesprächs mit Jethro auf den Laden zuzugreifen. Dazu verwenden wir wieder den Resources Explorer. Wähle *Data* -> *Dialogues* und doppelklicke auf die Datei *jethro_conv.dlg*:



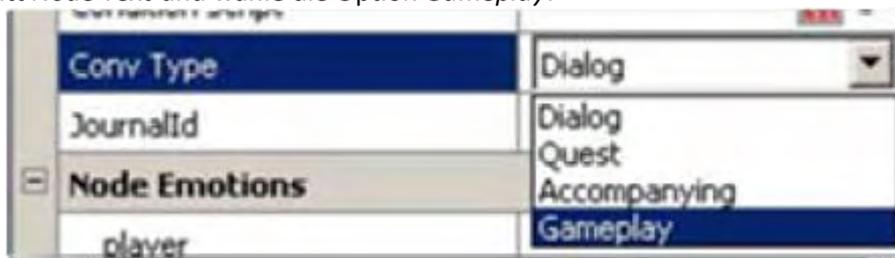
Das Fenster *jethro_conv.dlg* wird geöffnet:



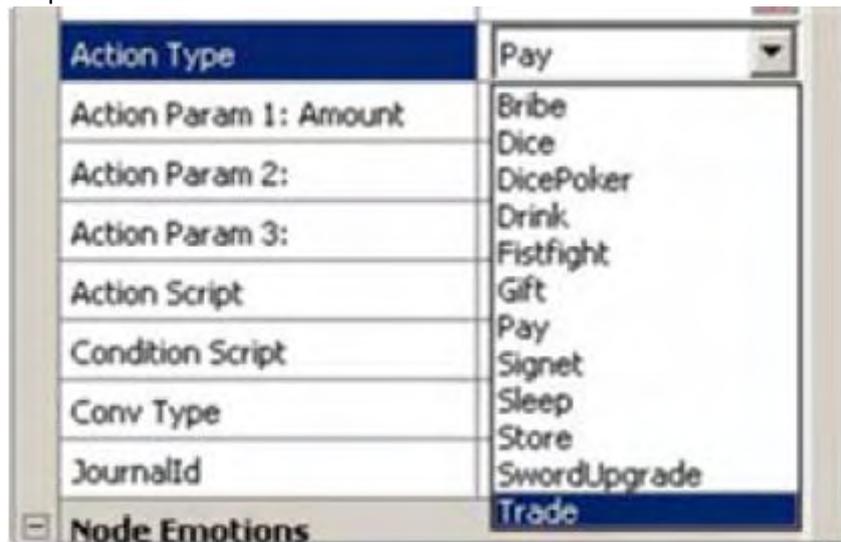
Wähle "Wollen wir spielen?", klicke mit der rechten Maustaste darauf und wähle *Create Dialog Reply* aus dem Menü:



Damit wird ein leerer Knoten erzeugt. Klicke darauf und wechsele zum Fenster *Properties*, in dem die Attribute und Dialoge für den Knoten angezeigt werden. Klicke auf die rechte Spalte neben dem Attribut *Conv Type* im Abschnitt *Node Text* und wähle die Option *Gameplay*:



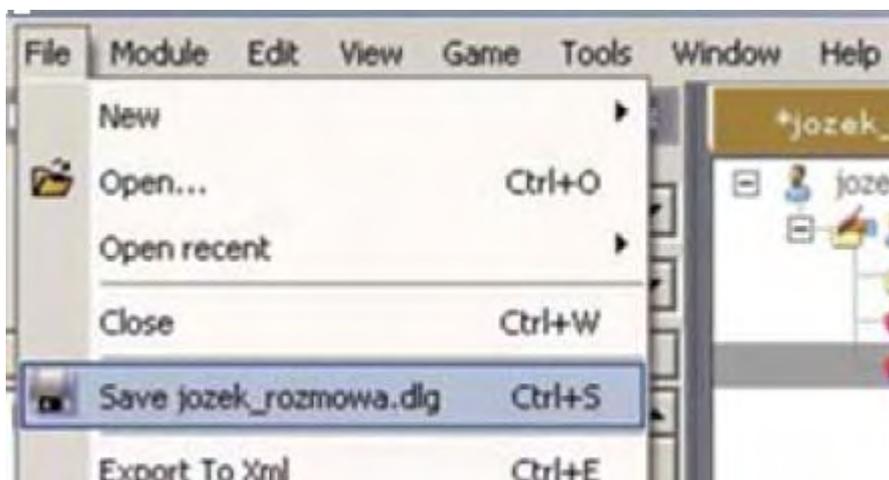
Im Abschnitt *Node Text* erscheint ein neues Attribut. Klicke auf die rechte Spalte neben dem Attribut *Action Type* und wähle die Option *Trade*:



Der leere Eintrag wird automatisch zu "Zeig mir Deine Ware.":



Wir müssen noch die Änderungen an Jethro's Dialog speichern. Wähle dazu *Save jethro_conv.dlg* aus dem Menü *File*:



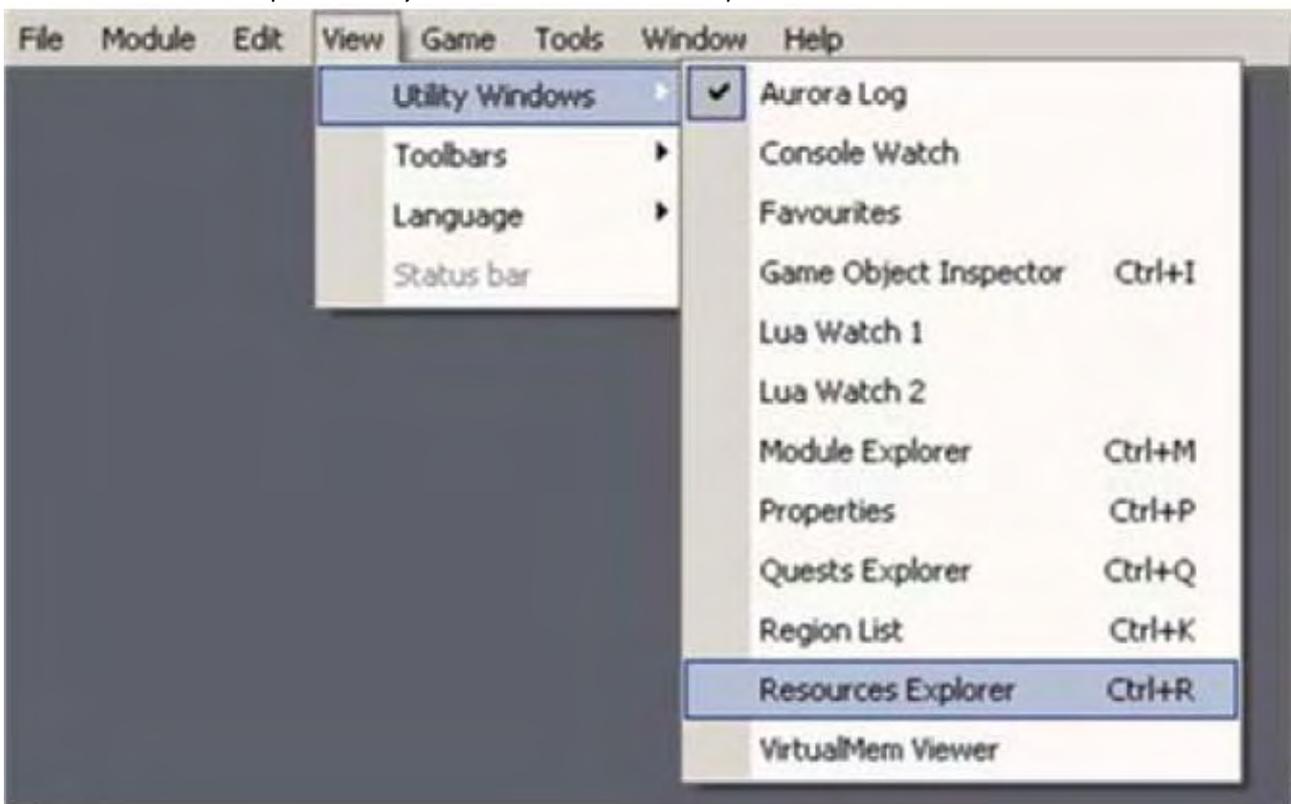
Wenn der Menüpunkt nicht sichtbar ist oder stattdessen ein anderer Dateiname angezeigt wird, dann ist das Fenster *jethro_conversation.dlg* gerade nicht aktiv. Um es zu aktivieren, klicke auf die Titelleiste des Fensters, dann wähle *Save jethro_conversation.dlg* aus dem Menü.

Das war alles. Wenn wir jetzt unser Modul laden und mit Jethro sprechen, dann können wir auch mit ihm handeln.

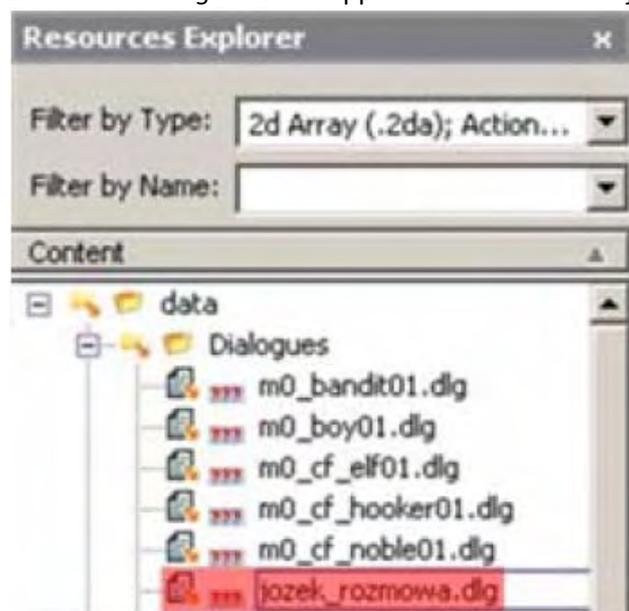
RASTPLÄTZE:

Zum Schluss wollen wir für Geralt noch einen Ort schaffen, an dem er meditieren kann. Der Rastplatz soll verfügbar sein, wenn er Jethro dafür bezahlt.

Auch dafür verwenden wir wieder unser Gespräch mit Jethro. Öffne den Resources Explorer, indem Du im Menü *View* den Menüpunkt *Utility Windows -> Resources Explorer*:



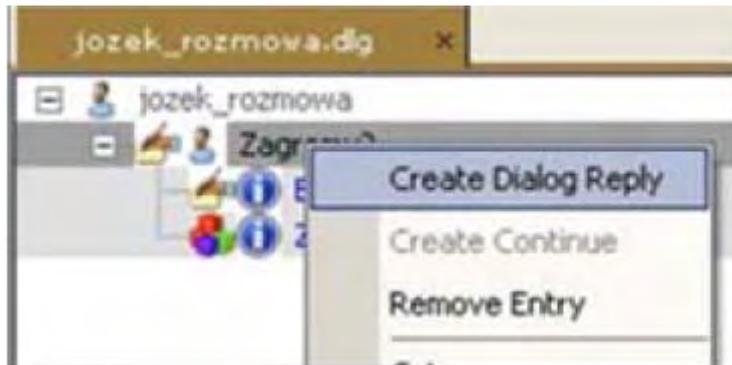
Wähle im Resources Explorer *Data -> Dialogues* und doppelklicke auf die Datei *jethro_conv.dlg*:



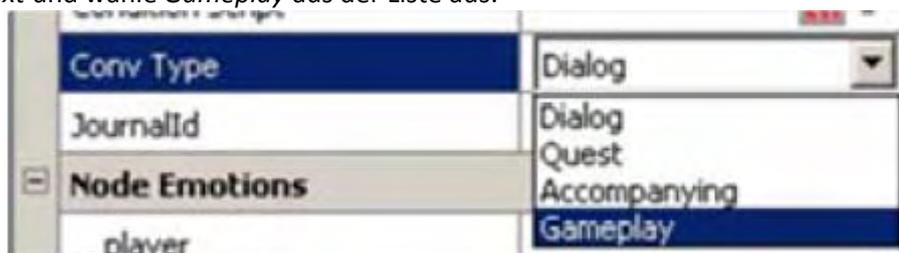
Das Fenster *jethro_conv.dlg* öffnet sich:



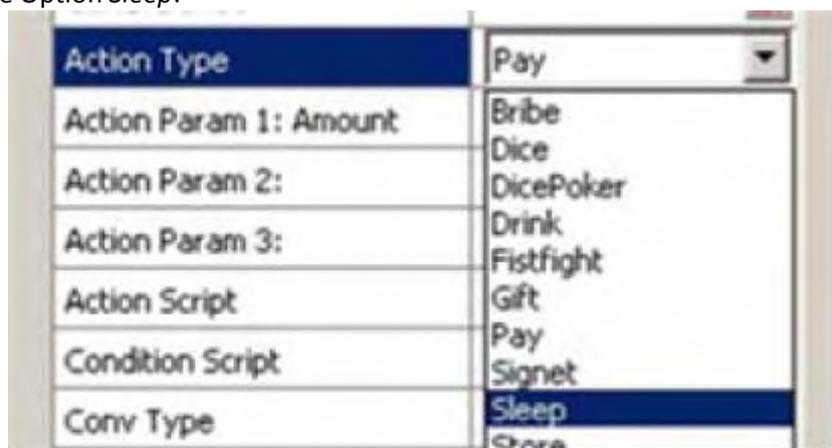
Wähle den Eintrag "Wollen wir spielen?" aus und klicke mit der rechten Maustaste darauf. Wähle *Create Dialog Reply* aus dem Menü:



Ein leerer Eintrag wird erstellt. Klicke darauf und wechsele zum Fenster *Properties*, das alle Eigenschaften und den Dialog für den Eintrag enthält. Klicke auf die rechte Spalte neben dem Attribut *ConvType* im Abschnitt *Node Text* und wähle *Gameplay* aus der Liste aus:



Im Abschnitt *Node Text* erscheint ein neues Attribut. Klicke auf die rechte Spalte neben dem Attribut *Action Type* und wähle die Option *Sleep*:



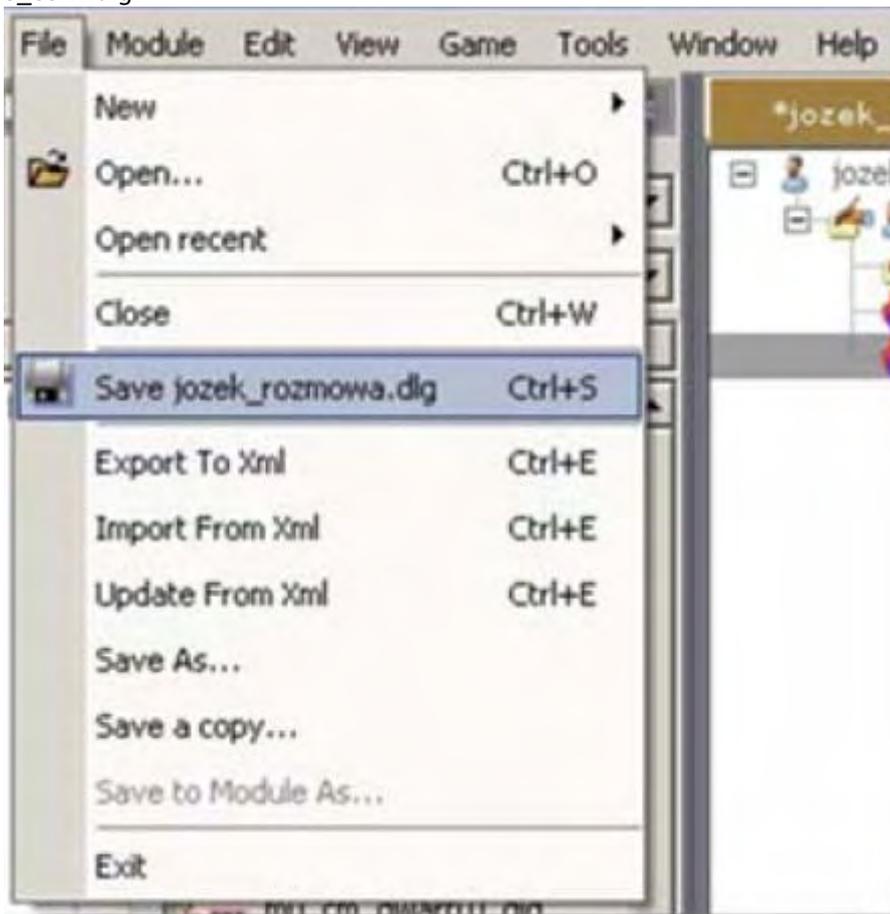
Der leere Eintrag ändert sich automatisch zu "Ich werde mich ein wenig ausruhen."



Wir können auch den Preis festlegen, den Geralt für das Ausruhen bezahlen muss. Dazu setzen wir das Attribut *Action Param 1: Price*. Wir können den Wert z. B. auf 5 setzen. Klicke dazu auf die rechte Spalte neben dem Attribut, gib den gewünschten Wert ein und drücke Enter:

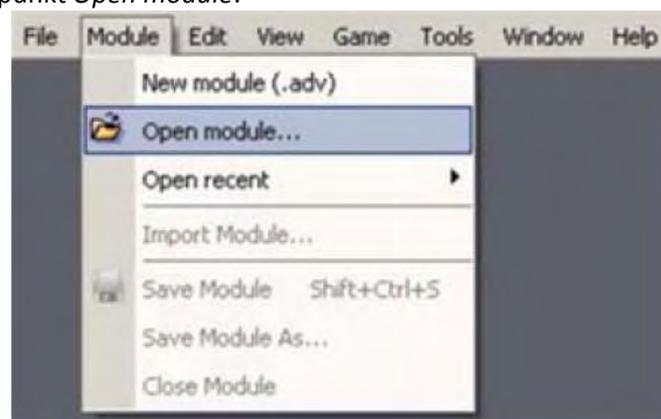
Action Type	Sleep
Action Param 1: Price	5

Jetzt muss Geralt 5 Orens an Jethro zahlen, damit er sich ausruhen kann. Die Änderungen, die wir am Dialog vorgenommen haben, müssen natürlich noch gespeichert werden. Wähle dazu im Menü *File* den Menüpunkt *Save jethro_conv.dlg*:



Wenn der Menüpunkt nicht angezeigt wird oder stattdessen ein anderer Dateiname angezeigt wird, dann ist das Fenster *jethro_conversation.dlg* gerade nicht aktiv. Klicke auf die Titelleiste des Fensters, um es zu aktivieren, und wähle dann *Save jethro_conversation.dlg* aus dem Menü *File*.

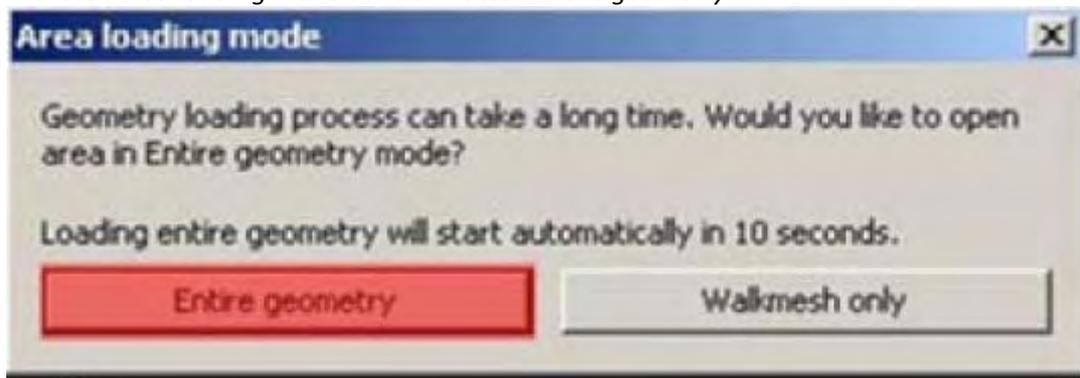
Als nächstes müssen wir das Modul öffnen und den Ort festlegen, an dem Geralt rasten soll. Wähle im Menü *Module* den Menüpunkt *Open module*:



Das übliche Windows-Fenster zum Laden von Dateien öffnet sich. Gehe zum Ordner Eigene Dateien\The Witcher (dort sind alle Module gespeichert), wähle die Datei our_adventure.adv und klicke auf *Öffnen*. Auf der linken Seite des Bildschirms öffnet sich der Module Explorer. Wähle *Areas* -> *g31_cave* und doppelklicke auf *Edit Area*:



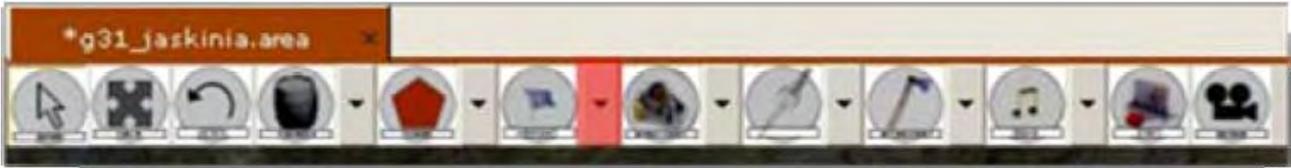
Klicke im Fenster *Area loading mode* auf den Button *Entire geometry*:



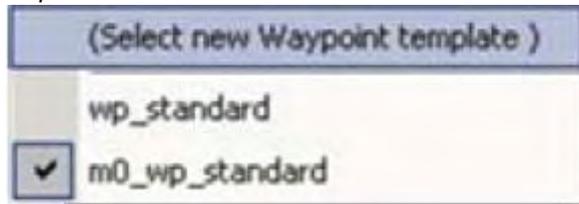
Jetzt wird der Bereich geladen, in den wir den Rastplatz legen wollen. Damit er leichter zu bearbeiten ist, sorgen wir zunächst für bessere Beleuchtung. Wähle im Menü *Render* den Menüpunkt *Ambient* -> *Medium Ambient*:



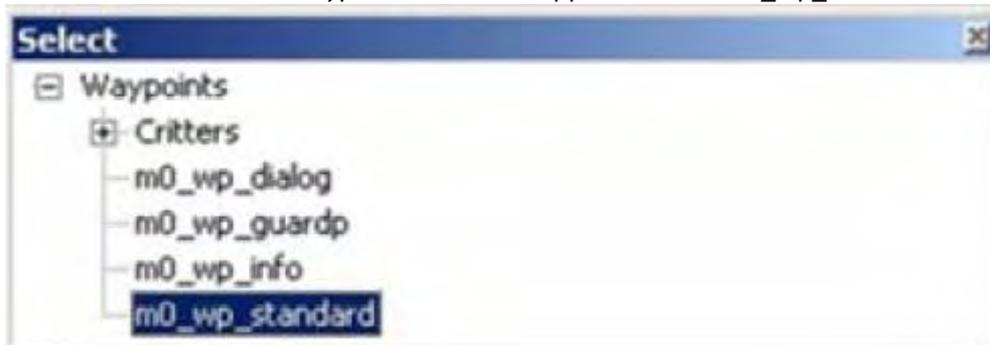
Um den Meditationspunkt zu erstellen, müssen wir einen neuen Wegpunkt setzen. Klicke auf den kleinen Pfeil neben dem Wegpunkt-Symbol:



Wähle *Select new waypoint template* aus dem Menü:



Klappe im Fenster *Select* den Baum *Waypoints* aus und doppelklicke auf *m0_wp_standard*:

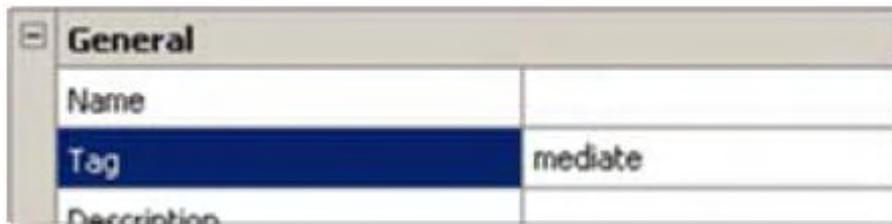


In der Nähe des Mauszeigers erscheint eine kleine violette Fahne. Das ist der neue Wegpunkt. Klicke jetzt in die Nähe von Jethro's Spawnpunkt, um den Wegpunkt zum Gebiet hinzuzufügen:



Dort kann Geralt meditieren und sich ausruhen. Der Pfeil zeigt dabei an, in welche Richtung Geralt beim Meditieren schaut. Wenn Geralt in eine andere Richtung schauen soll, drehe den Wegpunkt mit dem passenden Werkzeug. Wie man die Werkzeuge verwendet, ist in einem der vorherigen Kapitel erklärt.

Doppelklicke jetzt auf den Wegpunkt. Auf der rechten Seite des Bildschirms öffnet sich das Fenster *Properties*, in dem die Attribute des Wegpunkts aufgelistet sind. Gib beim Attribut *Tag* den Text "meditate" ein. Klicke dazu auf die rechte Spalte neben dem Attribut im Abschnitt *General*, gib den Text ein und drücke Enter:



Was passiert, wenn in einem Gebiet mehrere Wegpunkte das Tag "meditate" haben? Nichts weiter, außer dass Geralt zum nächstgelegenen Rastplatz geht.

Zum Schluss müssen wir das Modul noch speichern. Wähle im Menü *Module* den Menüpunkt *Save Module*:



Das wars. Jetzt kann Geralt, wenn er mit Jethro spricht, für 5 Orens eine Rast einlegen.

Hier endet das Handbuch. Es folgt nun das Schlusswort.

Schlusswort

Master of the Darkness:

So nun ist es endlich geschafft nach Monate langer Arbeit erblickte die Übersetzung das Licht Der Welt. Am Anfang war das nur ein Wunschgedanke von mir, ich überlegte die ganze Zeit ob es nicht sinnvoller wär eine Deutsche Übersetzung zu machen für die nicht englisch mächtigen—Schließlich entfasste ich den Entschluss etwas der Community zu schenken und so stand ich Nachts auf und begann mit den Seiten, am Anfang war es eine Seite und je mehr mich der eifer packte kam immer mehr dazu... so der 1. Schritt war also getan ein beginn für ein großartiges Projekt das viel Schweiß und Zeit kostete aber letztendlich sich auszahlte...nachdem mein Projekt anklang bekam gesellten sich weitere Übersetzer hinzu und eine Gemeinschaft entstand und wir arbeiteten hand in hand zusammen man muss bedenken das alles wurde in unserer Freizeit gemacht langsam wuchsen wir immer weiter und weiter und je mehr wir voranschritten umso mehr hatten wir den Ehrgeiz das Projekt nicht zum stoppen zu bringen es gab auch holpersteine und manch einer musste einspringen wenn einer nicht konnte.. es verlief alles Harmonisch untereinander es war total schön in der Gruppe zu sein aber jedes Projekt hat mal ein Ende und mit einer leichten träne verabschiede ich mich für die tolle Zeit mit euch! Keiner von uns hat die Übersetzung für sich gemacht sondern jeder wusste das es nur für die großartige Witcher Community ist, ich habe dieses Handbuch für die Community insbesondere aber Für Meditate als ein Dankeschön von mir übersetzt für mich war sie in dieser Zeit eine wichtige Person! Ein ganz wichtiger Mensch ist für mich Trainzfreak geworden er hat für euch das PDF erstellt und jeden Tag uns mit neuen Versionen versorgt das PDF wurde so schnell fertig weil er täglich daran gearbeitet hat eine ganz wichtige Person im Team! Ich hoffe dass sich unser eifer sich ausgezahlt hat und die Community sich an unser Handbuch erfreuen wird wir wünschen euch alle viel Spaß beim Modden, ich möchte aber an dieser Stelle einige Personen Danken!

Trainzfreak:

Ich wollte Master of the Darkness danken das er mich aufgenommen hat als ich mich gemeldet habe, und möchte allen Mitgliedern der WoP Community danken, die so ein großes Project verwirklicht haben, Hoffe das nach dem Project trotzdem noch in Kontakt bleiben, mehr fällt mir nicht ein, außer einen GROSSEN DANK an die Leute des WoP's Forums.

Zizous:

An 1. Stelle möchte ich meinen 2 wichtigsten Menschen in meinen Leben bedanken für mich sind sie etwas besonderes als ich am Boden war habt ihr mich aufgebaut und mir Kraft gegeben das Handbuch zum Abschluss zu bringen ich bedanke mich das ich euch kennenlernen durfte Danke an meine Partnerin Nadiana und Zizou so etwas wie euch gibt es kein 2. Mal! Danke geht auch an mein Team ☺ jeder einzelne von euch ist eine sehr wichtige Person ihr könnt alle stolz auf euch sein.. wenn ihr es nicht seit ich bin es aufjedenfall ihr habt Tag und Nacht daran gesessen und mich nicht im Stich gelassen alles wurde perfekt von euch strukturiert und geplant ich kann es immer wieder betonen aber ohne euch würde das Projekt scheitern...Dank geht an mein Staff im Team ich kann euch nur danken, danke das ihr mich aufgenommen habt und gezeigt habt wie man etwas für die Community erschaffen tut ☺ Danke an Meditate und Petra durch euch wurde ich ermuntert im Team zu arbeiten Petra weil sie Die News über uns verbreitet hat danke an euch beiden ☺ Danke an Trainzfreak ein guter Freund der mit mir zusammen gearbeitet hat um das PDF zu erstellen und mich immer bei not gefragt durch dein Druck haste es geschafft das es Fertig ist. Und jetzt stelle ich das Team vor:

Mitglieder der Übersetzungen:

- 1. Master Of Darkness alias Dustin (Teamleiter Handbuch-Übersetzer)**
Trainzfreak alias Thomas. (Ein wichtiger Mann wenn es um Die PDF Erstellung geht)
- 2. Meditate hatte immer ein Auge darauf eine sehr wichtige Person**
- 3. Leonard (Ansprechpartner und Moderator)**

4. .pl (Handbuch-Übersetzer)
5. The Projekt alias? (Handbuch-Übersetzer)
6. Tulzar (Handbuch-Übersetzer)
7. Zizou (ehem. Vanyar) 🌸😊(Handbuch-Übersetzerin)
8. LeeWicKeD (Handbuch-Übersetzer)

Ganz wichtige Leute die Lektoren:

1. **Canis Lupus (1. Korrektor)**
(wichtigster Mann: Rechtschreib- und Grammatik- Bereiniger)
2. **Leviticus (2. Korrektor)**
(genauso wichtig wie der 1. Korrektor)
3. **kingralf2005 (3. Korrektor)**
(genauso wichtig wie der 1. Korrektor)
4. **Vinyamar (4. Korrektor)**
(genauso wichtig wie der 1. Korrektor)
5. **Nimue (5. Korrektorin)**
(Genauso wichtig wie der 1. Korrektor)

Die fleißigen Leute die ihr Oberhalb seht haben dieses Projekt erst verwirklicht, und tatkräftig übersetzt und Korrigiert, besonders muss ich Master of the Darkness nochmal Danken!

Nun Folgen die Namen die genannt werden wollten:

1. Master of the Darkness - Dustin
2. Tulzar - Bernard
3. Trainzfreak - Thomas
4. Canis Lupus - ????????
5. Leewicked – Name unbekannt
6. Vinyamar
7. Nimue
8. pl
9. Zizou
10. Kingralf 2005
11. Leviticus
12. The Projekt
13. Leonard

So das war es nun von allen die genannt werden wollten, falls jemand fehlt meldet sich einfach im Forum.