

---

# Symfonia – Cutscene

Das Nachschlagewerk zur Erstellung von Cutscenes

by totalwarANGEL

## Inhaltsverzeichnis

|         |   |    |
|---------|---|----|
| 1       | Vorwort.....                              | 3  |
| 2       | Das Internal-Menü.....                    | 3  |
| 2.1     | Wie bekomme ich den Cutscene Editor?..... | 3  |
| 2.2     | Was muss beachtet werden?.....            | 4  |
| 2.2.1   | Speichern und Laden der Map.....          | 4  |
| 2.3     | Alternativen zum Cutscene Editor.....     | 5  |
| 3       | Der Cutscene Editor.....                  | 5  |
| 3.1     | Die Oberfläche.....                       | 5  |
| 3.1.1   | Beschreibung.....                         | 5  |
| 3.2     | Events.....                               | 6  |
| 3.3     | Die Kamera.....                           | 7  |
| 3.3.1   | Der Richtungsvektor.....                  | 7  |
| 3.3.2   | Steuerung der freien Kamera.....          | 7  |
| 3.3.3   | Der Blick durch die Kamera.....           | 7  |
| 4       | Cuscenes einbinden.....                   | 8  |
| 4.1     | Cutscene konvertieren.....                | 8  |
| 4.1.1.a | Allgemein.....                            | 8  |
| 4.1.1.b | Die Konfigurationsdatei.....              | 9  |
| 4.1.1.c | Das erzeugte Skript.....                  | 10 |
| 4.1.2   | Cutscene importieren.....                 | 10 |
| 4.2     | Cutscene an Quest anbinden.....           | 10 |
| 5       | Nachwort.....                             | 11 |

## Abbildungsverzeichnis

|              |  |   |
|--------------|--|---|
| Abbildung 1: | LUA-Debugger einschalten.....            | 3 |
| Abbildung 2: | LUA-Debugger abschalten.....             | 3 |
| Abbildung 3: | Cutscene Editor starten.....             | 4 |
| Abbildung 4: | Ordner öffnen.....                       | 4 |
| Abbildung 5: | Ordner speichern.....                    | 4 |
| Abbildung 6: | Erklärung Fenster "Cutscene Editor"..... | 5 |
| Abbildung 7: | Blick durch ein Camera Event.....        | 8 |

## 1 Vorwort

Die Cutscenes sind eine ressourcensparende Methode Kameraflüge zu erstellen. Story Cutscenes sind ein gängiges Mittel in verschiedenen Arten von Spielen um die Handlung zu erzählen. Sie wurden in den Kampagnen von Siedler 5 verwendet, um die Geschehnisse besser zu präsentieren. Dieses Feature war auch für Siedler 6 geplant, hat es aber aus unbekannten Gründen nicht ins finale Spiel geschafft.

Aus diesem Grund wurde der Editor wohl auch niemals fertig entwickelt. Selbst mit freigeschalteten versteckten Bordmitteln ist es unmöglich Cutscenes sinnvoll zu nutzen. Aber mit ein paar zusätzlichen Werkzeugen und der QSB-S können diese Kameraflüge trotzdem verwendet werden. Zwar lädst Du Dir damit einiges an Mehraufwand auf, aber Du wirst mit wunderschönen und meist absolut ruckelfreien Kameraflügen belohnt!

Das folgende Dokument befasst sich mit Erstellung und Import von Cutscenes und mit der Verwendung in der Map. Es richtet sich eher an erfahrene Benutzer. Solltest Du also etwas nicht verstehen, stelle Fragen!

## 2 Das Internal-Menü

### 2.1 Wie bekomme ich den Cutscene Editor?

Der Cutscene Editor gehört zu den versteckten Tools im Internal-Menü. Das Menü ist für den normalen Nutzer nicht verfügbar. Wenn man allerdings den Dev-Machine-Key in der Registry setzt, wird das Menü nach Neustart des Editors angezeigt.

Ein bequemer Weg diesen Key zu setzen ist der LUA-Debugger.

Das Programm kann auf [settlers.pro](http://settlers.pro) heruntergeladen werden.

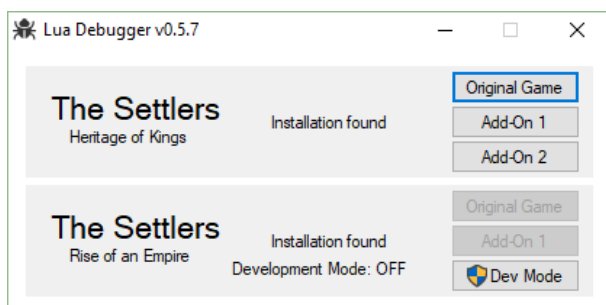


Abbildung 1: LUA-Debugger einschalten

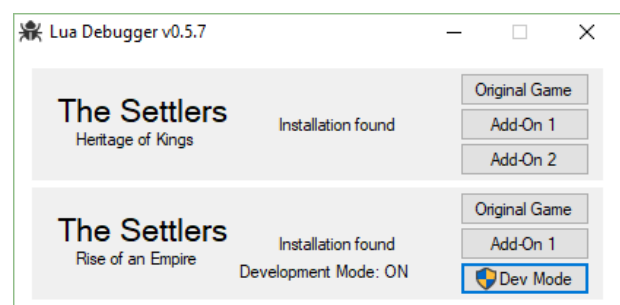


Abbildung 2: LUA-Debugger abschalten

Um den Dev-Machine-Key zu setzen, klickst Du auf die Schaltfläche „Dev Mode“. Du brauchst Administratorprivilegien um in die Registry zu schreiben. Wenn Du den normalen Zustand wiederherstellen willst, klicke einfach noch einmal auf „Dev Mode“. Wenn der Key gesetzt ist, ab sofort als Dev-Mode bezeichnet, wird Internal angezeigt.

Im Menü findest Du neben anderen hilfreichen Tools den Cutscene Editor.

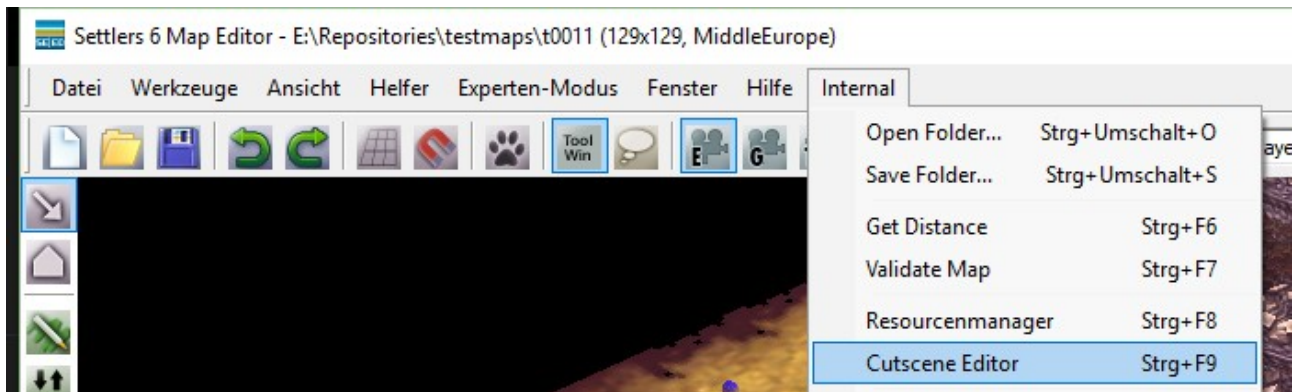


Abbildung 3: Cutscene Editor starten

Ein Klick auf „Cutscene Editor“ öffnet das Tool. Alternativ kannst Du auch die Hotkeys benutzen, welche im Menü angezeigt werden.

## 2.2 Was muss beachtet werden?

### 2.2.1 Speichern und Laden der Map

Wenn Du Cutscenes erstellst oder bearbeitest, kannst Du die Map nicht auf normalen Wege speichern, weil dann alle CS-Dateien verloren gehen. Du musst die Optionen „Save Folder“ und „Open Folder“ im Internal-Menü benutzen.

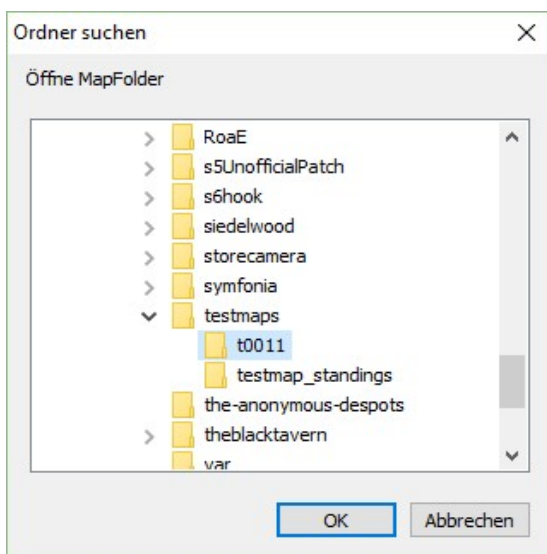


Abbildung 4: Ordner öffnen

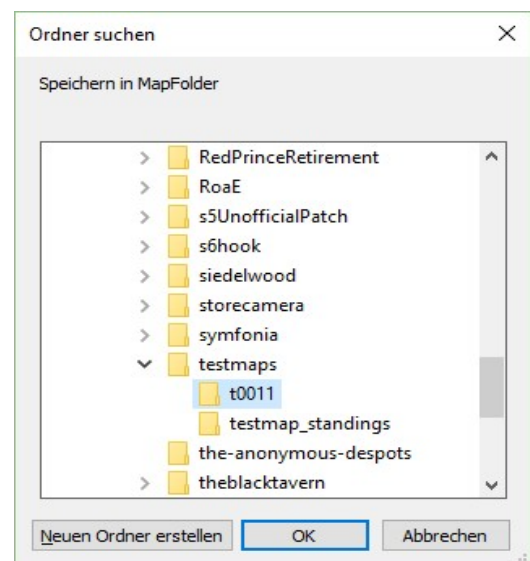


Abbildung 5: Ordner speichern

Damit Du Cutscenes erstellen und speichern kannst, muss zuerst die Map als Ordner exportiert werden. Dazu öffnest Du die Map wie gewohnt und speicherst sie dann über „Save Folder“. Navigiere zu einem Verzeichnis und erstelle einen neuen Ordner, vorzugsweise mit dem Namen der Map, damit Du ihn später leichter zuordnen kannst.

Alternativ kannst Du auch eine mit dem BBA-Tool entpackte Map Laden. Dazu musst Du den Ordner innerhalb von „externalmap“ auswählen! Das BBA-Tool ist ebenfalls auf [settlers.pro](http://settlers.pro) für Dich herunterladbar. Du wirst später sowieso das BBA-Tool benötigen, um die Cutscenes in eine normale Map zu packen!

Falls Du den Ordner über „Save Folder“ gespeichert hast, musst Du noch alle Dateinamen auf Kleinbuchstaben ändern, sonst kann dieser Ordner nicht zu einer Map konvertiert werden! Alternativ kannst Du später einfach die CS-Dateien herauskopieren.

## 2.3 Alternativen zum Cutscene Editor

Momentan ist der Cutscene Editor ALTERNATIVLOS.

Der alte Siedelwood Cutscene Assistent ist (momentan) nicht kompatibel!

Später wird es vielleicht ein Offline-Tool geben, welches das Erstellen von Cutscenes völlig losgelöst vom Cutscene Editor oder dem Spiel ermöglichen wird.

## 3 Der Cutscene Editor

### 3.1 Die Oberfläche

Der Cutscene Editor besteht aus 2 Dialogen. „Cutscene Event“ wird benutzt um die Eigenschaften eines Events einzustellen und die aktuellen Kamerakoordinaten zu übernehmen. „Cutscene Editor“ ist Dein Hauptwerkzeug bei der Erzeugung einer Cutscene.

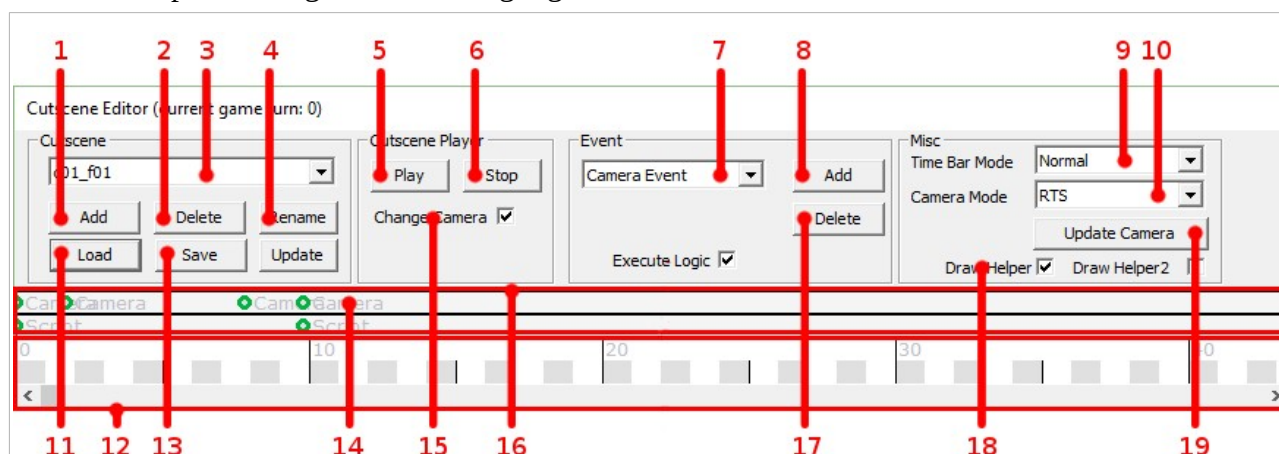


Abbildung 6: Erklärung Fenster "Cutscene Editor"

#### 3.1.1 Beschreibung

Folgende Tabelle erklärt die einzelnen Elemente der Oberfläche:

|   |  |
|---|--|
| 1 | Einen neuen Flight erstellen. Du musst einen Namen eingeben. Achte auf eindeutige Namen, die sich gut zuordnen lassen!   |
| 2 | Löscht den aktuellen Flight. Die Datei wird erst nach erneutem Export über „Save Folder“ tatsächlich gelöscht.   |
| 3 | Der aktuell ausgewählte Flight. Mit diesem Dropdown-Menü kannst Du zwischen den einzelnen Flights wechseln.  |
| 4 | Den aktuellen Flight unter neuen Namen kopieren.   |
| 5 | Startet die Preview des aktuellen Flight im Cutscene Mode. Hier kannst Du die Kamerabewegung, so wie sie im Spiel sein wird, vorher anschauen. Bitte vorher speichern! |

|    |   |
|----|---|
| 6  | Die Preview wird angehalten.  |
| 7  | Hier wird ausgewählt, welches Event erzeugt werden soll. Du brauchst allerdings nur Camera Events. Den Rest kannst Du getrost ignorieren.                                 |
| 8  | Fügt ein neues Event hinzu.   |
| 9  | Ändert den Modus des Zeitstrahl.<br>Additiv: Ein Event und alle Events danach werden gleichzeitig verschoben<br>Normal: Nur das ausgewählte Event wird verschoben         |
| 10 | Kameramodus einstellen.<br>Cutscene: Cutscene Modus zur Vorschau eines Flights<br>Free View: Freie Kamera zum Einstellen der Kameraposition<br>RTS: Normale Editor-Kamera |
| 11 | Alle CS-Dateien neu Laden. Nützlich, wenn Du per Hand andere CS-Dateien in das Verzeichnis kopiert hast.  |
| 12 | Der Zeitstrahl. Nutze ihn um Events zu platzieren.  |
| 13 | Die Änderungen im aktuellen Flight in die CS-Datei speichern.   |
| 14 | Ein Event in der Event-Leiste.  |
| 15 | Kameramodus automatisch verändern. Das sollte immer aktiv sein, sonst siehst Du keine Kameraanimationen.  |
| 16 | Die Event-Leiste. Hier werden alle Events auf ihrer Position im Zeitstrahl angezeigt. Du kannst ein Event auswählen und verschieben. Ausgewählte Events sind rot.         |
| 17 | Löscht das in der Event-Leiste ausgewählte Event  |
| 18 | Kamera Entity sichtbar/unsichtbar schalten. Das Kamera Entity zeigt den Ausschnitt an, den der Spieler in der Kameraeinstellung sehen wird.                               |
| 19 | Setzt die Kamerakoordinaten aus dem Event und lässt sie die Kamera anzeigen.  |

## 3.2 Events

Über den Button „Add“ in der Event-Gruppe kannst Du ein neues Event erstellen. Es wird ein Event des ausgewählten Typ erstellt. Für unsere Zwecke ist nur das Camera Event von Bedeutung. Alle anderen kannst Du ignorieren. Die Koordinaten der Kamera können entweder übernommen oder per Hand eingegeben werden. Je nach dem, wie viel von der Map innerhalb des Bildausschnittes liegt, muss ggf. der Wert bei „Far Clip Plane“ reduziert werden.

Ein Flight benötigt immer mindestens 4 Camera Events um zu funktionieren!

Wenn der Flight schon einmal für die QSB-S angepasst wurde, wirst Du zwei Steuerevents finden. Mit ihnen teilt der Flight dem Skript mit, dass er gerade begonnen hat oder, dass er gerade beendet ist. Solltest Du die Dauer des Flights verändern, empfiehlt es sich diese Events zu löschen!

Wenn Du Events veränderst, solltest Du jedes Mal speichern. Vor allem auf schwachen Systemen läuft der Cutscene Editor sehr instabil. Es kommt dann oft zu Abstürzen. Die Ursache ist unbekannt, es hängt aber wohl damit zusammen, dass dieses Feature niemals zu Ende entwickelt wurde.

## 3.3 Die Kamera

### 3.3.1 Der Richtungsvektor

Die Position und Blickrichtung der Kamera wird durch zwei Punkte gesteuert. Diese Punkte spannen einen Richtungsvektor auf. Über die einzelnen Punkte eines Flights wird dann die Bewegung als Kurve berechnet. Diese Kurve wird von der Kamera abgefahren. Das dadurch resultierende Ergebnis ist wesentlich flüssiger als die LERP<sup>1</sup> der Briefings. Es sollte nicht zu Ruckelern kommen.

### 3.3.2 Steuerung der freien Kamera

Die Koordinaten für Blickrichtung und Position der Kamera lassen sich bequem aus der freien Kamera übernehmen. Dazu muss man jedoch die freie Kamera richtig bedienen. Die Kamera wird über die Pfeiltasten und die mittlere Maustaste gesteuert.

#### Mittlere Maustaste / Mausrad

Durch gedrückt halten der mittleren Maustaste und bewegen der Maus wird die Blickrichtung der Kamera verändert. Das allein bewegt die Kamera jedoch kein Stück!

#### Pfeiltasten

Die Pfeiltasten bewegen die Kamera vor, zurück und seitlich. Und das relativ zur Blickrichtung!

- Nach Oben (↑) bewegt die Kamera vorwärts hin zum Blickpunkt.
- Nach Unten (↓) bewegt die Kamera weg vom Blickpunkt.
- Nach Links (←) bewegt die Kamera nach links.
- Nach Rechts (→) bewegt die Kamera nach rechts.

Pfeiltasten und Mausrad lassen sich auch gleichzeitig benutzen.

### 3.3.3 Der Blick durch die Kamera

Die nützlichsten Features des Cutszene Editors sind die Übernahme der aktuellen Kameraeigenschaften und die Vorschau auf den resultierenden Flight.

Um die Kamera einzustellen, kannst du entweder die aktuellen Einstellungen der freien Kamera übernehmen oder in ein Camera Event hineinspringen und die freie Kamera die Einstellungen aus dem Event verwenden lassen.

Wenn Du dich in einem Camera Event befindest, kannst Du die Draw Helper nutzen, um den Bildausschnitt anzuzeigen, der im Spiel zusehen sein wird. Der äußere Rahmen ist der Bildausschnitt. Sinn des inneren Rahmens ist es, die Kamera besser justieren zu können. Auch außerhalb des Events sind die Helper nützlich, da sie die Lage der Kamera darstellen.

---

<sup>1</sup> LERP (lineare Interpolation): Es wird mittels Multiplikation eines Vektors ein Punkt auf einer Geraden zwischen zwei Punkten bestimmt.



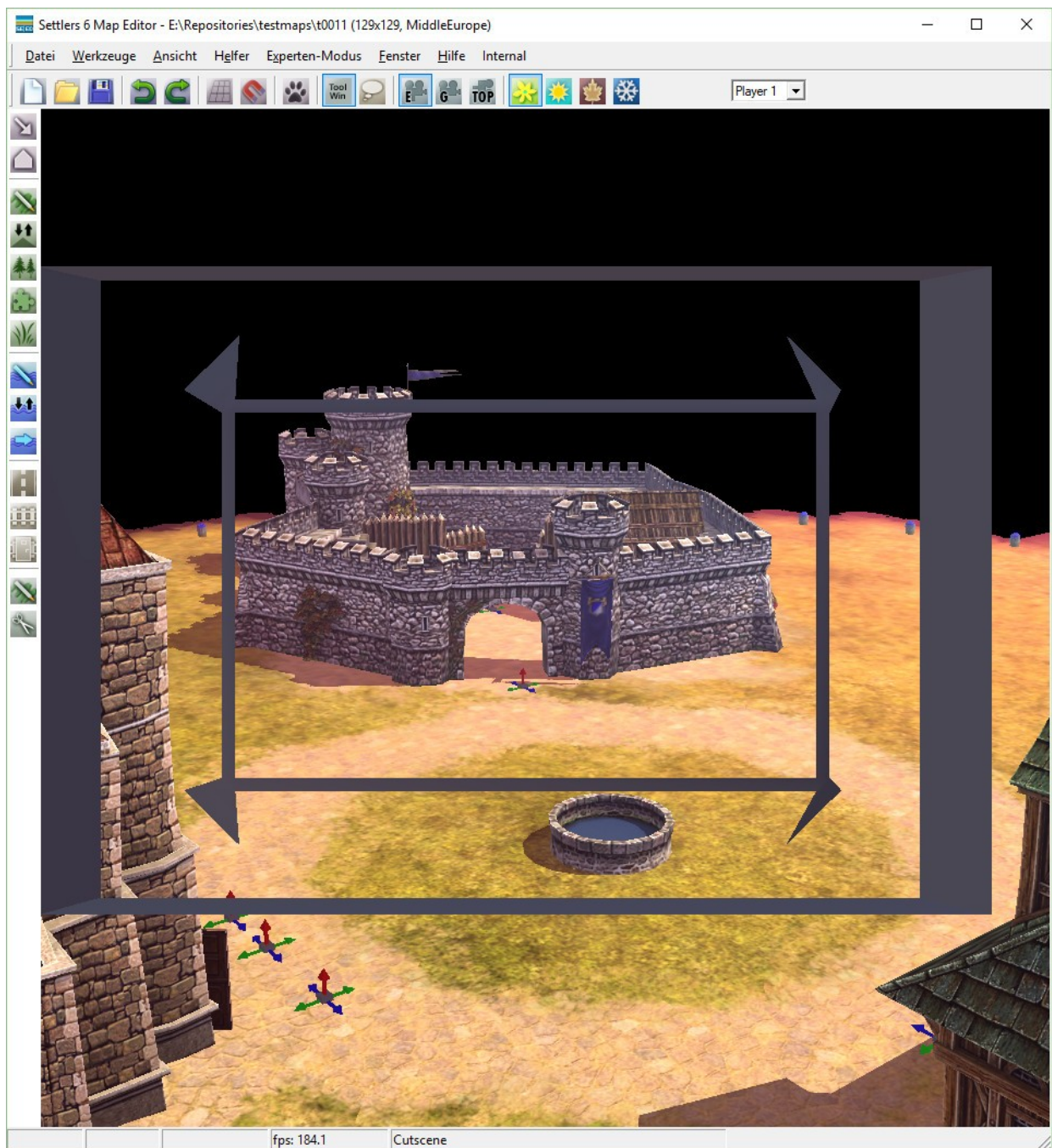


Abbildung 7: Blick durch ein Camera Event

## 4 Cuscenos einbinden

### 4.1 Cutscene konvertieren

#### 4.1.1.a Allgemein

Um die Flights im Spiel nutzen zu können, müssen diese angepasst werden und es muss ein entsprechendes Steuerskript geschrieben werden. All dies kann man auch per Hand erledigen. Aber wieso, wenn man es nicht muss? Diese Aufgabe übernimmt der Cutscene Maker!



Bei dem Programm handelt es sich um eine Konsolenanwendung. Das heißt, es wird über die Eingabeaufforderung, Power Shell oder Bash gestartet. Wenn Du bei der Installation die PATH Variable gesetzt hast, sollte dies nach einem Neustart funktionieren.

Folgender Befehl bearbeitet alle CS-Dateien in einem Ordner:

```
CutsceneMaker <Name> <Directory>
```

Du musst <Name> mit dem Funktionsnamen des Steuerskriptes ersetzen. Den Namen kannst du frei wählen. Er darf aber keine Umlaute oder Sonderzeichen (außer `_`) enthalten. Dieser Name wird gleichzeitig als Dateiname der erzeugten LUA-Datei verwendet.

<Directory> muss mit dem Pfad des Ordners ersetzt werden, indem die CS-Dateien liegen. Der Pfad kann relativ<sup>2</sup> oder absolut angegeben<sup>3</sup> werden.

#### 4.1.1.b Die Konfigurationsdatei

Alle CS-Dateien im Verzeichnis werden zu einer Cutscene zusammengesetzt. Als Standard gibt es weder Texte noch werden die CS-Dateien angepasst. Die Eigenschaften der Cutscene und der einzelnen Flights können jedoch in einer INI-Datei mit beliebigen Namen angegeben werden.

```
[Cutscene]
HideBorderPins=true
RestoreGameSpeed=false
BigBars=false
BarOpacity=1.0
FastForward=true
StartingFunction=SomeFunctionAtStart
FinishedFunction=SomeFunctionAtEnd
```

In der ersten Sektion werden die Eigenschaften der Cutscene selbst definiert. Diese Sektion muss immer `Cutscene` heißen! Die Optionen sollten Dir bereits von Briefings bekannt sein. Sie funktionieren hier analog. Die Option `FastForward` steuert, ob die Cutscene beschleunigt vorgespielt werden darf, oder nicht.

```
[c01_f01]
Title=Flight 1
Text=Das ist ein Test!
FadeIn=3.0
FadeOut=nil
Action=nil
```

Danach folgen beliebig viele Sektionen mit Angaben zu einzelnen Flights. Eine Sektion heißt immer genauso, wie ihre zugehörige CS-Datei! Die Optionen eines Flights sind an die Optionen einer Briefing-Seite angelehnt. Auch hier kannst du Text, Titel, Fader und ein Callback setzen. Wenn ein Flight fehlt, werden Defaults verwendet.

- 
- 2 Relative Pfadangabe: Der Pfad eines Ordners wird relativ zum Arbeitsverzeichnis angegeben. In diesem Fall ist das Arbeitsverzeichnis das, in dem CutsceneMaker ausgeführt wird.  
Beispiel: `foo/bar/muh`
  - 3 Relative Pfadangabe: Der Pfad eines Ordners wird komplett von Beginn der Partition bis zum Ziel angegeben.  
Beispiel: `C:/Dateien/Siedler/foo/bar/muh`

### 4.1.1.c Das erzeugte Skript

Wenn alles richtig angegeben wurde, erscheint eine LUA-Datei im Arbeitsverzeichnis. Diese Datei beinhaltet Dein Steuerskript für die Cutscene. Eine solche Datei könnte wie folgt aussehen:

```
function Cutscene01()
    local Cutscene = {
        RestoreGameSpeed = false,
        HideBorderPins    = true,
        BigBars           = false,
        BarOpacity         = 1.0,
        FastForward        = true,
    };
    local AF = API.AddFlights(Cutscene);

    AF {
        Flight  = "c01_f01",
        Title   = "Flight 1",
        Text    = "Das ist ein Test!",
        Action  = nil,
        FadeIn  = 3.0,
        FadeOut = nil,
    }

    ...

    Cutscene.Starting = SomeFunctionAtBegin;
    Cutscene.Finished = SomeFunctionAtEnd;
    return API.CutsceneStart(Cutscene);
end
```

An der Stelle von ... können beliebig viele weitere Flights stehen, je nach dem wie es von Dir in der INI-Datei konfiguriert wurde.

### 4.1.2 Cutscene importieren

Eine Map kann nicht normal gespeichert werden, wenn sie CS-Dateien enthält. Genau wie alle anderen Zusatzinhalte, gehen auch die CS-Dateien verloren. Deshalb wird das BBA-Tool benötigt, um die Map zusammen mit den CS-Dateien zu packen.

CS-Dateien müssen immer im Hauptverzeichnis der Map liegen. Das Hauptverzeichnis ist nach dem Entpacken in der Regel folgendes Verzeichnis:

```
.../<DATEINAME>.<DATEIENDUNG>.unpacked/maps/externalmap/<MAPNAME>
```

Beispiel:

```
C:/MapArchiv/testmap01.s6xmap.unpacked/maps/externalmap/testmap01
```

Nach dem Kopieren der CS-Dateien in das Verzeichnis die Map wie gewohnt zurückpacken.

## 4.2 Cutscene an Quest anbinden

Eine Cutscene kann durch Aufruf der Funktion, welche sie definiert, gestartet werden.

```
Cutscene01()
```

Auf diese Weise wird die Cutscene umgehend ausgeführt, wenn keine andere Cutscene und kein anderes Briefing aktiv ist. Allerdings ist die Cutscene so nicht an einen Quest gebunden und kann nicht mit `Trigger_Briefing` überwacht werden!

Cutscenes werden von der QSB-S wie ein Briefing behandelt. Das bedeutet, dass eine Cutscene nur dann starten kann, wenn kein anderes Briefing und keine andere Cutscene gerade laufen. Ferner können Cutscenes wie Briefings mit den entsprechenden Behavior gestartet und überwacht werden.

```
Reprisal_Briefing("Cutscene01")
```

Startet die Cutscene, wenn der Quest fehlschlägt.

```
Reward_Briefing("Cutscene01")
```

Startet die Cutscene, wenn der Quest erfolgreich ist.

```
Trigger_Briefing("QuestCutscene01", "All")
```

Löst einen Quest aus, wenn die Cutscene eines anderen Quests beendet ist.

## 5 Nachwort

Jetzt weißt Du, wie du Cutscenes erstellen und für das Spiel nutzen kannst. Cutscenes geben Dir die Möglichkeit Deine Map optisch aufzupeppen. Wenn Du sie benutzt, um die Geschichte zu erzählen, achte darauf, nicht zu viel Text anzugeben. Die Kameraanimationen sollen genossen werden. Zu viel Text, der gelesen werden muss, stört das Erlebnis des Spielers. Mehr als zwei Zeilen sollten es also nicht werden.

Falls zum Handling von Cutscenes noch Fragen offen sind, oder falls etwas nicht hinreichend beschrieben wurde, kannst Du dich jeder Zeit an totalwarANGEL wenden.