

Inhaltsverzeichnis

| | |
|--|----------|
| <u>Vorwort.....</u> | <u>2</u> |
| <u>Aufrufen einer Cutscene.....</u> | <u>2</u> |
| <u>Erstellen einer Cutscene.....</u> | <u>2</u> |
| <u>Cutscene Positionen festlegen.....</u> | <u>2</u> |
| <u>Steuerung des Ingame-Assistenten.....</u> | <u>3</u> |
| <u>Cutscene Bearbeiten und finalisieren.....</u> | <u>4</u> |
| <u>Funktionsbeschreibung.....</u> | <u>4</u> |
| <u>Die Buttons.....</u> | <u>5</u> |
| <u>Events.....</u> | <u>5</u> |

Vorwort

Dieses AddOn dient dazu eine einfache Schnittstelle zum abspielen von echten Cutscenes zur Verfügung zu stellen. Dabei handelt es sich um das Lost Feature¹ eines vom Spiel gerenderten Kamerafluges. Dieses Feature wurde niemals zu Ende entwickelt und stand deswegen bisher nicht zur Verfügung.

Eine Cutscene ist eine CS-Datei. Jede Datei, die genutzt werden soll, muss sich im Standardverzeichnis der Map befinden, also dort wo auch die QSB, oder die globalen und lokalen Skripte ihr Habitat haben. Gegenwärtig ist es nicht möglich eine Cutscene aus einem anderen Verzeichnis zu laden.

Aufrufen einer Cutscene

Der Aufruf einer Cutscene erfolgt folgendermaßen:

```
CS.StartCutscene(_cutscene)
```

Wo `_cutscene` der Name der CS-Datei als String ist. Der Datei-Suffix² wird aber nicht mit angegeben!

Diese Funktion gibt `true` zurück, falls die Cutscene gestartet wurde. Wenn gerade ein Briefing läuft, oder der Start der Cutscene aus einem anderen Grund fehlschlägt, wird `false` zurückgegeben. Läuft ein Briefing, muss auf das Ende gewartet werden, bevor die Cutscene gestartet werden kann.

Es können mehrere Cutscenes gleichzeitig „in Auftrag“ gegeben werden. In dem Fall wird die erste gestartet und die anderen werden in eine Warteschlange eingefügt, in der Reihenfolge in welcher sie im Skript aufgerufen werden.

Erstellen einer Cutscene

Zum Erstellen einer Cutscene geht man in zwei Schritten vor. Zuerst kann man die Positionen bestimmen, die von der Kamera abgeflogen werden sollen. Danach müssen die CS-Dateien generiert werden.

Cutscene Positionen festlegen

Es gibt zwei Möglichkeiten Positionen festzulegen, die die Kamera später entlangfliegen soll. Zum einen kann dazu der „Siedelwood Cutscene Assistant“ verwendet werden (dazu mehr im kommenden Abschnitt). Aber meistens wird es einfacher sein, die Positionen direkt innerhalb des Spiels festzulegen, da es dort einfacher ist das ganze zu visualisieren. Dazu muss folgende Funktion in der FMA geladen werden:

¹ Ein Lost Feature ist eine im Entwicklungsprozess einer Software untergegangenes Feature das gar nicht oder nur teilweise nutzbar ist.

² Ein Datei-Suffix ist der Teil des Dateinamen, der hinter dem letzten Punkt steht. Beispiel: mapscript.lua

```
CS.CreateCutscene_DEV_ONLY()
```

Wie der Name schon sagt, sollte diese nur während der Entwicklung genutzt werden und genau genommen, nur dann, wenn eine Cutscene innerhalb des Spiels bestimmt werden soll. Startet man nun die Map befindet man sich in einem angepassten Kameramodus, der es einem ermöglicht die Kamera nach belieben zu positionieren und bestimmte Kameraeinstellungen als Cutscene-Positionen zu speichern.

Steuerung des Ingame-Assistenten

Maus

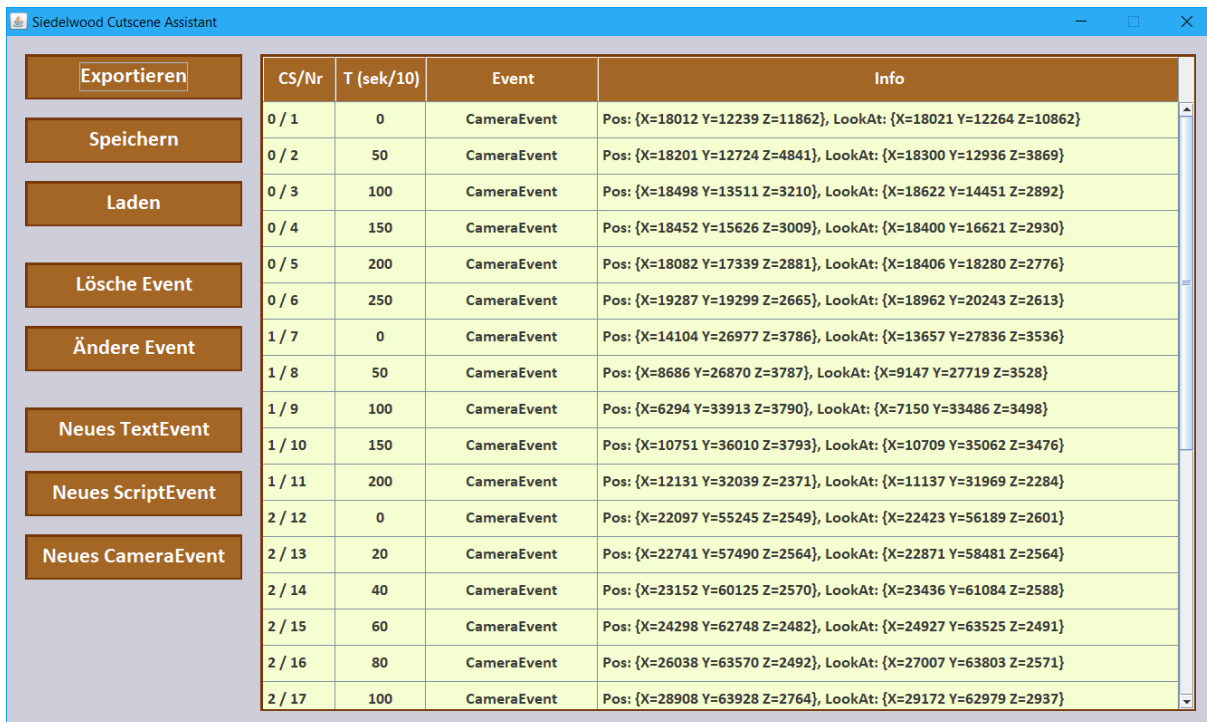
Die Blickrichtung der Kamera wird mit der Maus gesteuert. Bewege die Maus über den Bildschirm um die Kamera zu drehen.

Tastatur

| Taste | Beschreibung |
|-----------------------|---|
| W,A,S,D ← ↑ → ↓ | Bewegungen in der Kameraebene (Vor, Zurück, Rechts, Links). |
| Space | Bewegung nach oben |
| Backspace | Setzt die maximale Zeit zwischen zwei Punkten einer Cutscene auf den Standardwert zurück. Die maximale Zeit zwischen zwei Cutscene Positionen ist 99. Höhere Zahlen werden ignoriert. |
| Y | Bewegung nach unten |
| F | Speichert die Werte der aktuellen Position. Die Kamera wird zu dieser Position fliegen, wenn die Cutscene abgespielt wird. |
| J | Speichert die aktuelle Position als Sprungziel. Es wird nicht geflogen! |
| P | Erstellt eine Vorschau der Cutscene unter Nutzung des Briefing System. Achtung: Dies ist eine Simulation. Es handelt sich nicht um eine Cutsche und es kann vom endgültigen Ergebnis abweichen. |
| Nummernblock 0 - 9 | Jede Cutscene wird automatisch mit 5 Sekunden zwischen jeder Position versehen. Durch tippen auf dem Ziffernblock wird eine neue Zeit zur nächsten Position festgelegt. Diese kann wieder verändert werden. Zwischen jeder Änderung wird eine Notiz auf dem Bildschirm angezeigt, die die Zeit angibt, die zur nächsten Position genutzt werden wird. Diese Einstellung wird für alle folgenden Positionen beibehalten. |
| Num + | Erhöht die Bewegungsgeschwindigkeit. |
| Num - | Reduziert den Bewegungsgeschwindigkeit. |

Cutscene Bearbeiten und finalisieren

Hierzu wird der „Siedelwood Cutscene Assitant“ verwendet.



| CS/Nr | T (sek/10) | Event | Info |
|--------|------------|-------------|---|
| 0 / 1 | 0 | CameraEvent | Pos: {X=18012 Y=12239 Z=11862}, LookAt: {X=18021 Y=12264 Z=10862} |
| 0 / 2 | 50 | CameraEvent | Pos: {X=18201 Y=12724 Z=4841}, LookAt: {X=18300 Y=12936 Z=3869} |
| 0 / 3 | 100 | CameraEvent | Pos: {X=18498 Y=13511 Z=3210}, LookAt: {X=18622 Y=14451 Z=2892} |
| 0 / 4 | 150 | CameraEvent | Pos: {X=18452 Y=15626 Z=3009}, LookAt: {X=18400 Y=16621 Z=2930} |
| 0 / 5 | 200 | CameraEvent | Pos: {X=18082 Y=17339 Z=2881}, LookAt: {X=18406 Y=18280 Z=2776} |
| 0 / 6 | 250 | CameraEvent | Pos: {X=19287 Y=19299 Z=2665}, LookAt: {X=18962 Y=20243 Z=2613} |
| 1 / 7 | 0 | CameraEvent | Pos: {X=14104 Y=26977 Z=3786}, LookAt: {X=13657 Y=27836 Z=3536} |
| 1 / 8 | 50 | CameraEvent | Pos: {X=8686 Y=26870 Z=3787}, LookAt: {X=9147 Y=27719 Z=3528} |
| 1 / 9 | 100 | CameraEvent | Pos: {X=6294 Y=33913 Z=3790}, LookAt: {X=7150 Y=33486 Z=3498} |
| 1 / 10 | 150 | CameraEvent | Pos: {X=10751 Y=36010 Z=3793}, LookAt: {X=10709 Y=35062 Z=3476} |
| 1 / 11 | 200 | CameraEvent | Pos: {X=12131 Y=32039 Z=2371}, LookAt: {X=11137 Y=31969 Z=2284} |
| 2 / 12 | 0 | CameraEvent | Pos: {X=22097 Y=55245 Z=2549}, LookAt: {X=22423 Y=56189 Z=2601} |
| 2 / 13 | 20 | CameraEvent | Pos: {X=22741 Y=57490 Z=2564}, LookAt: {X=22871 Y=58481 Z=2564} |
| 2 / 14 | 40 | CameraEvent | Pos: {X=23152 Y=60125 Z=2570}, LookAt: {X=23436 Y=61084 Z=2588} |
| 2 / 15 | 60 | CameraEvent | Pos: {X=24298 Y=62748 Z=2482}, LookAt: {X=24927 Y=63525 Z=2491} |
| 2 / 16 | 80 | CameraEvent | Pos: {X=26038 Y=63570 Z=2492}, LookAt: {X=27007 Y=63803 Z=2571} |
| 2 / 17 | 100 | CameraEvent | Pos: {X=28908 Y=63928 Z=2764}, LookAt: {X=29172 Y=62979 Z=2937} |

Funktionsbeschreibung

Beim Start des Tools wird automatisch die letzte im Spiel erzeugte Cutscene geladen. Auf dem Bild sieht man wie das aussehen könnte. Hier wurden größtenteils Flüge verwendet, aber auch zwei Sprünge. In der ersten Spalte sind zwei Informationen enthalten. Die Nummer der Position in der gesamten Cutscene ist an zweiter Stelle. Die erste Stelle zeigt an zu welcher Teil-Cutscene die Position gehört. Teil-Cutscenes sind notwendig, da es nicht möglich ist Sprünge in CS-Dateien zu machen. Dies wird umgangen indem einfach mehrere CS-Dateien aneinandergehängt werden, die Teil-Cutscenes. Damit ist auch jeder Sprung die erste Position einer neuen TeilCutscene, somit sind diese leicht zu erkennen. Außerdem beginnt die Zeitrechnung mit jeder Teil-Cutscene neu.

Die Zeit befindet sich in der zweiten Spalte. Da die CS-Datei eine Genauigkeit im Bereich der Zehntelsekunde erlaubt werden hier Zehntelsekunden für die Zeit verwendet. Daran sollte man sich erinnern, wenn man sich wundert warum die Cutscene viel zu schnell vorbei ist. Die Zeit ist innerhalb der Tabelle anpassbar.

Die Spalten Drei und Vier enthalten mehr Informationen dazu was an der Stelle passiert. Hier wird angegeben, welche Art von Event zu dem Zeitpunkt aufgerufen wird, sowie etwas Details geliefert.

Die Buttons

| Button | Beschreibung |
|-------------------|---|
| Exportieren | Erzeugt die CS-Dateien. Alle diese Dateien müssen in die Map gepackt werden. Der Name darf keine Großbuchstaben enthalten. Der Name wird später für den Aufruf verwendet. Die anderen erstellten Cutszenes sind in der Form Name..“appendix“..Nummer benannt und <i>dürfen nicht umbenannt werden</i> , da sonst der automatische Aufruf nicht funktioniert! Nur die erste CS-Datei darf umbenannt werden, allerdings muss sie dann auch mit dem neuen Namen aufgerufen werden. |
| Speichern | Da mit jedem Start immer nur die letzte bearbeitete Cutscene aus dem Spiel geladen wird soll es möglich sein auch eine bereits angepasste Cutscene weiterzubearbeiten. Daher kann man den derzeitigen Stand hiermit als TXT-Datei abspeichern. |
| Laden | Lädt den derzeitigen Stand der als TXT-Datei gespeichert wurde. |
| Lösche Event | Löscht ein Event/Zeile. |
| Ändere Event | Ändert ein Event/Zeile. |
| Neues TextEvent | Erstellt ein Event, das einen Text anzeigt. |
| Neues ScriptEvent | Erstellt ein Event, dass einen Skriptbefehl ausführt. |
| Neues CameraEvent | Erstellt ein Kamera-Event. |

Events

Events werden genutzt um die Cutscene zu steuern.

Das TextEvent

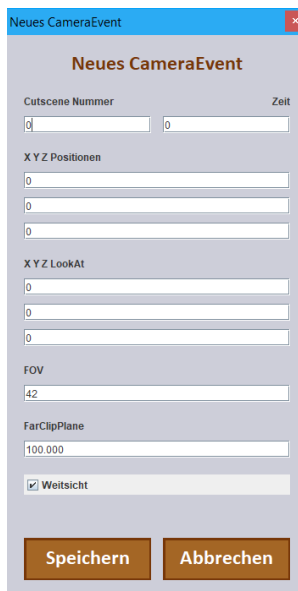
Ein TextEvent bestimmt über den anzuzeigenden Text und Titel, ob dieser Zentriert sein soll und welche Art von Balken angezeigt werden soll. Jedes neue TextEvent überschreibt die bisherigen Daten völlig. Ein TextEvent kann zu einem völlig beliebigen Zeitpunkt aufgerufen werden und hängt nicht davon ab wann eine Kameraposition erreicht ist (kann aber auch gleichzeitig sein). Es ist nur nicht möglich zwei TextEvents zur gleichen Zeit aufzurufen.

Das ScriptEvent



Ein ScriptEvent bestimmt Code, der in einem der beiden Skripte ausgeführt werden soll. Die Defaultfunktion hat keine Auswirkung, da innerhalb von Cutscenes Notizen nicht angezeigt werden.

Das CameraEvent



CameraEvents sind einfacher zu handhaben, wenn sie innerhalb des Spiels definiert worden, sind, da die Zahlen nicht sehr repräsentativ sind. Es ist aber trotzdem möglich diese über das Tool anzulegen und die Positionen selbst zu bestimmen.