

Strecken in Assetto Corsa konvertieren

Benötigte Programme:

- 3DSimED3
- KS-Editor SDK
- Grafikprogramm zum Erstellen und bearbeiten von .png /.dds texturen (z.B. GIMP)

Projektstart:


1. Erstelle die richtige Ordnerstruktur im Track-Ordner:

\STEAM\SteamApps\common\assetto corsa\content\tracks\

Die normale Struktur sieht folgendermaßen aus (in „“ der Name des Ordners):

- „trackname“ Der Name der Strecke. ACHTUNG der Name muss identisch mit dem Namen der.kn5 Datei sein!
Übergeordneter Ordner in dem folgende Ordner und Dateien stehen:
- „ai“ dieser Ordner enthält Informationen, welche von der AI benötigt werden, um auf der Strecke zu fahren. Wenn keine Informationen für die AI vorhanden sind, kann der Ordner auch einfach leer bleiben.
- „data“ hier kann man, wenn man möchte, spezifische Oberflächen für die Strecke, Lichteffekte, Audio Quellen definieren, Kameras editieren und mehr. Am einfachsten man kopiert den Ordner aus einer anderen Strecke und ändert was man möchte.
- „ui“ Hier werden die Basisdaten, welche im Spiele-Auswahlmenü angezeigt werden, definiert:
preview.png - das Bild für die Streckenauswahl
outline.png - das Streckenlayout das über dem Vorschau-Bild zu sehen ist
ui_track.json – Textinformationen zur Strecke
- trackname.kn5 Das ist die eigentliche Streckendatei, welche alle Daten wie Geometrie, Texturen und Materialien enthält. (diese Datei zu erstellen ist das Ziel)
- trackname.fbx ist die von 3DSimED3 exportierte Datei, welche im KS-Editor geöffnet und für AC weiterbearbeitet wird.
Wenn am Ende alles fertig ist kann diese Datei gelöscht werden, sie ist für das Spiel AC nicht notwendig !
- „texture“ Dies ist ein wichtiger Ordner, in dem alle Texturen liegen, welche der KS-Editor benötigt.
Wenn am Ende alles fertig ist kann dieser Ordner gelöscht werden, er ist für das Spiel AC nicht notwendig !
- „gmt“ In diesen Ordner exportiere ich die .gmt Objekte aus 3DSimED
Wenn am Ende alles fertig ist kann dieser Ordner gelöscht werden, er ist für das Spiel AC nicht notwendig !
- trackname.fbx.ini Diese Datei wird vom KS-Editor automatisch angelegt, wenn man die Änderungen „save persistence“ abspeichert.
Wenn am Ende alles fertig ist kann diese Datei gelöscht werden, sie ist für das Spiel AC nicht notwendig !

2. Vorbereiten der Strecke für den .fbx Export:

- öffne die Strecke in 3DSimED
- alles zentrieren „center model“
- lösche die skyboxi
- lösche alle timegates, wie xsector, xfinish, xpitout usw.
- passe die Texturen an: da in AC nur einseitige Texturen verwendet werden, muss man diese, wenn nötig (z.B. Bäume, Schilder usw.) im Objekt doppelseitig machen. Am besten sieht man das, wenn man unter **Display, Back Face** anklickt, dann sieht man nur die Seite, die man im Spiel auch sieht. Ist es nun erforderlich, dass man beide Seiten sieht, wählt man das Objekt aus öffnet unter dem  Icon das Menü und wählt: „Make Double Sided“
- beachte die Benennung der Objekte. Punkt 2.1
- setze die AC_timing Objekte, auch hier ist auf die Benennung zu achten s. Punkt 2.2
- exportiere die Strecke als .fbx in den vorher angelegten Streckenordner.
(Einstellungen FBX 6.0 binary)

2.1 Benennungen:

In 3DSimED müssen alle Objekte, welche physikalische Eigenschaften besitzen, nach einem bestimmten Schema benannt werden: <ID><Name><optionale Zusatzbezeichnung>

<ID> ist eine Zahl größer 0, wenn physikalische Eigenschaften wirken sollen

<Name> Name des Objektes: ROAD, GRASS, GRAVEL, KERB, SAND, WALL

Erst wenn man also eine Zahl größer 0 vor den Namen setzt, erhält man physikalische Eigenschaften. Bei ROAD z.B. kann man auf der Oberfläche fahren, die WALL erhält dadurch Kollision usw. Man kann also jedem Objekt welches eine Kollision verursachen soll den Namen WALL geben, einfacher ist es aber ein Objekt so zu benennen wie es normal heißt und dann eine einfache, nicht sichtbare Mauer als WALL davor zu setzen.

Der Name bei den Oberflächen auf denen man fährt ist sehr wichtig und beeinflusst das Rollverhalten. Wie sich welche Oberfläche verhält ist in der „surfaces.ini“ definiert.

ACHTUNG: wichtig ist auch, dass in 3DSimED, sowohl als „Object-name“ als auch als „Instance“ derselbe Name steht, als „Instance“ muss der Name in Großbuchstaben stehen.

Namensbeispiele: 1WALL_mauer, 1ROAD_02, 1GRASS_dunkel usw.

Wichtig ist es auch, alle 3D Vegetation als KSTREE zu benennen.



2.2 AC_TIMING

Ich beschreibe hier nur die für AB-Strecken notwendigen Timing-Objekte.

- AC_PIT_0
Beschreibung: Platz in der Box, bzw. Vorstart bei Bergrennen; einfach durchnummeriert so viele Plätze es geben soll, beginnend bei 0
- AC_START_0
Beschreibung: Platz in der das Fahrzeug beim Start steht, bei Bergrennen gleich wie PIT ; einfach durchnummeriert so viele Plätze es geben soll, beginnend bei 0
- AC_HOTLAP_START_0
Beschreibung: Dieses Objekt steht direkt am Start und wird für den Hotlapping-Modus benötigt.
- AC_AB_START_L bzw. AC_AB_START_R
Beschreibung: Diese Objekte stehen links „L“ und rechts „R“ der Startlinie und lösen die Zeitmessung aus.
- AC_AB_FINISH_L bzw. AC_AB_FINISH_R
Beschreibung: Diese Objekte stehen links „L“ und rechts „R“ der Ziellinie und beenden die Zeitmessung.
- AC_TIME_0_L bzw. AC_TIME_0_R
Beschreibung: Diese Objekte stehen links „L“ und rechts „R“ der Zwischenzeitmessung.

3. Arbeiten und Einstellungen mit SDK (KS-Editor)

Hierzu werde ich nicht viel schreiben, da gibt es ein sehr gutes Video-Tutorial von Kunos, das das wichtigste beschreibt (https://www.youtube.com/watch?v=qj3z_yzdwbs).

Ich mache hier nur ein paar für mich wichtige Anmerkungen und Erklärungen und beschreibe Dinge, welche in diesem Video noch nicht erklärt werden :

3.1-Shader-Einstellungen

Diese Einstellungen findet man im Reiter **Material**

- Straße: ksPerPixelNM (NM = normal map = bump-map)
- Objekte mit Transparenz: ksPerPixelAT (AT = Alpha Test)
- 3D Vegetation: ksTree (entfernt die Schatten an den Bäumen und Sträuchern selbst)

3.2 AI-Path

Der AI-Path ist wichtig für die Track Kameras, welche sich daran orientieren.

- gehe zu \STEAM\SteamApps\common\assetocorsa\system\cfg und editiere die Datei: assetto_corsa.ini
ändere den Wert des folgenden Eintrags von 0 auf 1
[AC_APPS]
ENABLE_DEV_APPS=1 (Damit werden verschiedene Apps im Spiel aktiviert, u.a. eine APP Namens AI)
- Starte AC und wähle deine Strecke, stelle das Auto auf die Startposition und öffne die App: AI
Drücke **Start Recording** und fahre die Strecke ab (dabei ist es nicht wichtig schnell zu sein, sondern eine möglichst schöne und runde Linie zu fahren). Nach dem Ziel beende die Aufzeichnung und drücke **SAVE ai**.
- Beende das Spiel, im Ordner „ai“ gibt es jetzt eine Datei namens: fast_lane.ai.candidate
- benenne diese um in: fast_lane.ai
Fertig
- In SDK lädt man diese Datei über: AI Spline/Load

3.3 Track-Cameras Einstellungen

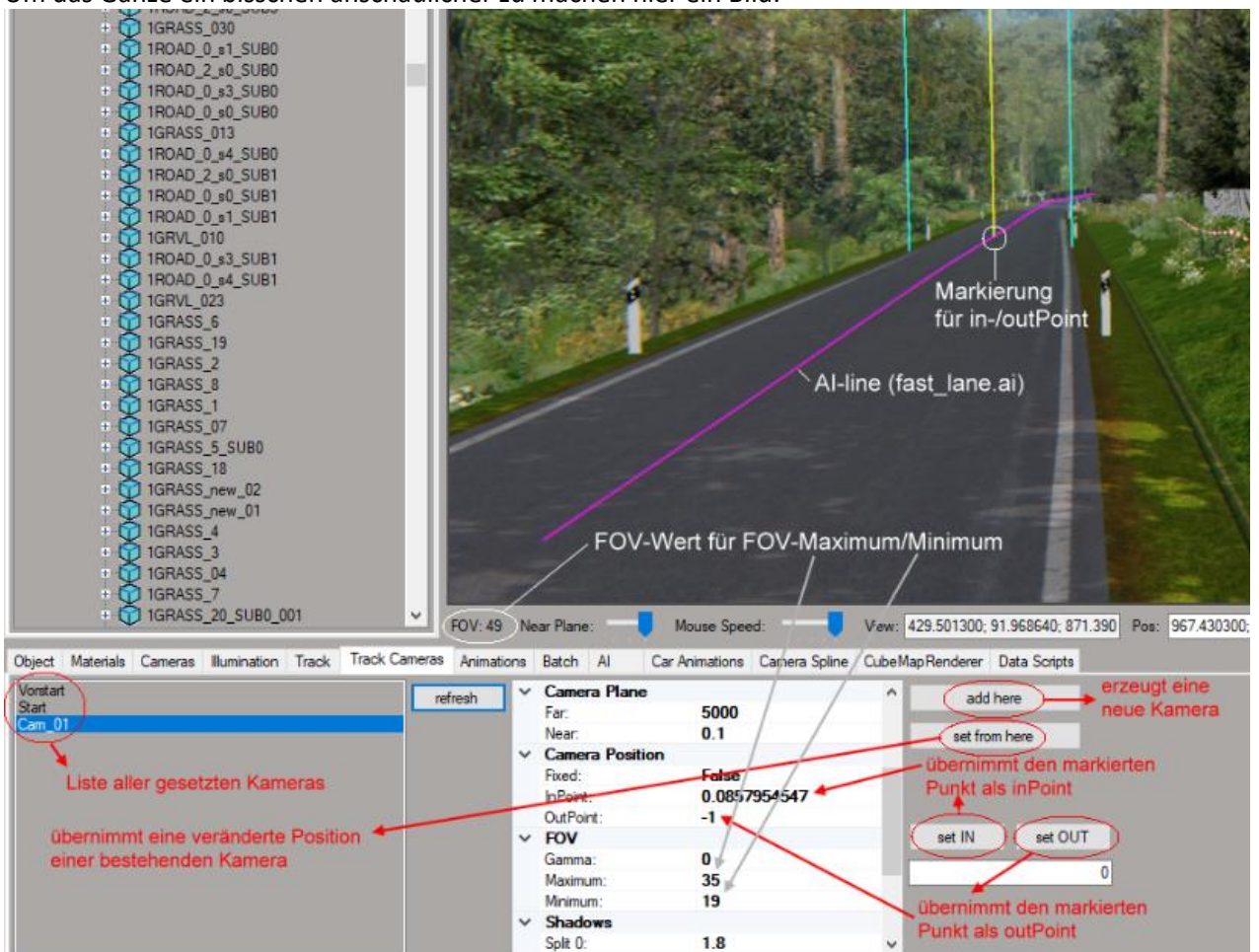
Hier zuerst mal eine Beschreibung der Einstellungsmöglichkeiten (zumindest die, die ich bereits herausgefunden habe)

SDK Menü	Beschreibung
Camera DOF Factor: Focus: Manual: Range:	Depth Of Field = Tiefenschärfe ? ? ? ?
Camera Exposure Maximum: Minimum:	Camera Belichtung ? ?
Camera Plane Far: Near:	Kameraumgebung: wieviel die Kamera von der Umgebung zeigt (in Meter). z.B. liegt in der Entfernung von 500 m ein Objekt, das man nicht sehen soll, so ist <i>Far</i> auf 499 einzustellen, oder steht direkt vor der Kamera ein Baum (0,2m), so ist <i>Near</i> auf 0,3m einzustellen.
Camera Position Fixed: InPoint: OutPoint:	Kameraposition False = Kamera schwenkt mit True = Kamera ist fest ab wann die Kamera aktiv wird, -1 steht für unendlich der Punkt an dem an die andere Kamera übergeben wird
FOV Gamma: Maximum: Minimum:	Field of view = Sichtfeld ? Zoom (mit Mausrad zoomen und den FOV-Wert ablesen) Zoom (mit Mausrad zoomen und den FOV-Wert ablesen)
Misc Name SplineAnimationLength SplineFileName SplineRotation	Verschiedenes Name der Kamera ? ? ?
Shadows Split0: Split1: Split2:	Schatten ? ? ?

3.4 Track-Cameras erstellen/platzieren

- zuerst den AI-Path der Strecke laden (AI Spline/Load/fast_lane.ai)
- jetzt das Track Camera Menü in der unteren Menüleiste aufrufen
- fahre an die Position, an der die Kamera platziert werden soll und drücke **add here** (Kamera wird gesetzt)
- gib der Kamera einen Namen
- InPoint (der Punkt ab dem die Kamera aktiviert wird):
 - bei A-B Strecken ist der 1. In-Point -1 für unendlich !
 - bei Rundstrecken ist der 1. In-Point 0 = Start-Ziellinie
- OutPoint (der Punkt bis zu dem die Kamera aktiviert bleibt)
 - Der letzte OutPoint muss immer 1 sein !
- WICHTIG: der OutPoint der vorherigen Kamera ist zugleich auch der InPoint der folgenden Kamera.
- Alle weiteren Punkte werden über die „fast_lane.ai“ festgelegt, in dem man mit der rechten-Maustaste eine Stelle auf der Linie auswählt (gelbe Markierung). Ist ein Punkt gewählt wird dieser mit den Feldern:
 - set IN (für den InPoint)
 - set OUT (für den OutPoint)
 übernommen.
- nun alle anderen gewünschten Einstellungen vornehmen
- Kameras speichern: Track **Cameras/Save** im Ordner .../data/cameras.ini
- Man kann jederzeit die Kameras verändern, in dem man die cameras.ini erneut lädt.
 - Mit Doppelklick auf die gewünschte Kamera (Name) gelangt man direkt in die Ansicht der Kamera, hier kann man nun im Menü die Einstellungen der Kamera ändern, oder man kann auch die Kamera selbst verschieben und dann mit dem Button **set from here** die neue Position der Kamera festlegen. Nach den Änderungen die cameras.ini erneut speichern.

Um das Ganze ein bisschen anschaulicher zu machen hier ein Bild:



3.5 Animierte Flaggen

- Zuerst bei den Materialeigenschaften ksPerPixelAT wählen und danach ksFlags
- Einstellungen bei Flaggen

SDK Menü	Beschreibung
frequency	wie schnell sich die Fahne bewegt
distortion	Wieviel sich die Fahne bewegt (Ausschlag)
boh	?
ksAmbient	helligkeit
ksDiffuse	?
ksSpecular	Wieviel Licht das Material spiegelt
ksSpecularEXP	Exponent der spiegelung
ksEmissive	?
ksAlphaRef	Intensität der Schatten