

Trackdesign Tutorial

Version 1.2

Notwendige Programme:

Mindestvoraussetzungen:

T3ED - Version 0.07pre1.1 H-1 (für das Trackdesign)

NFS Wizard (zum extrahieren der Track - Grafiken)

Paintshop Pro ab Version 5 oder ein anderes vollwertiges Grafikprogramm

Zusätzliche Tools:

VIV Wizard (für das Zusammenfügen der BMP Grafiken: „Alphakanal- und Normalbild“ zu einer TGA Grafik)

EA Graphics Editor für das Exportieren von NFS6 Bilder aus den FSH Dateien

FSHTOOL Damit lassen sich auch qfs Dateien bearbeiten (siehe Anhang 3).

NFS String Editor (siehe hierzu Anhang 5) und CarCad (siehe hierzu Anhang 6)

Trackdateien:

Sky.qfs (Himmels -Texturen)

Tr.frd (Streckenverlaufdatei für den Tag)

Trn.frd (Streckenverlaufdatei für die Nacht - nicht immer enthalten)

Tr0.qfs (Texturen für den Tag)

Trn0.qfs (Texturen für die Nacht - nicht immer enthalten)

Wetterverhältnisse:

Tr.ini (Wetterdaten bei Tag und ohne Wetter)

Trw.ini (Wetterdaten bei Tag mit Wetter)

Trn.ini (Wetterdaten bei Nacht und ohne Wetter)

Trnw.ini (Wetterdaten bei Nacht mit Wetter)

Ladeschirme / Videowände:

Datei: **T(Nummer des Tracks)_00.qfs z.B. T13_00.qfs (für Rocky Pass)**

Der große Start - Ladebildschirm (Slide) befinden sich in Verzeichnis:

Data\Feart\Slide

Datei: **T(Nummer des Tracks).qfs z.B. T13.qfs (für Rocky Pass)**

Die kleine Videowand (VidWall) befindet sich im Verzeichnis:

Data\Feart\vidwall

Schritt 1 Öffnen des Tracks und Standardeinstellungen von T3ED

Hierzu wird im Programm T3ED mittels des Befehls OPEN (Menü File) die jeweilige **Tr.frd** Datei des zu ändernden Tracks aus dem Verzeichnis :

\Data\Tracks\Trackname geladen. Den Streckenverlauf des Tracks könnt Ihr nach dem Laden jetzt im Editor schon bewundern. Vor dem Bearbeiten sollten im Editor noch einige wichtigen und nützlichen Einstellungen vorgenommen werden.

1. Im Menü View:

- Toolbar und Status Bar
- Resolution auf FULL stellen
- Show Objects
- Show Lanes
- Show Track View

(für die 3D Ansicht des Tracks, beim ersten Aufruf kommt ein Dateiauswahlfenster. Hier einfach auf „Öffnen“ gehen - die richtige Texturendatei wird schon vorbelegt)





2. Im Menü Edit-Modes

- Extra Smoothing

Schritt 2 Einfache Trackbearbeitung

Um möglichst schnell und einfach den Streckenverlauf eines Tracks ändern zu können, gibt es im T3ED den sog. Block Modus.

Klickt hierzu bitte das Symbol Block Mode  in der Symbolleiste an.

Jetzt könnt Ihr ganz einfach irgendeinen Abschnitt des Tracks anklicken (wird rötlich hinterlegt) und mit der linken Maustaste sowie des Symbols  (seitlich) oder  (Höhe) verschieben. Die an diesen anhängenden Blöcke werden hierbei automatisch mit verschoben bzw. angepaßt. Mit den Laufleisten (unten und rechts) kann man die Ansicht des Tracks im Editor noch verändern um die Auswirkungen besser beobachten zu können.

Nützlich ist hierbei auch das Symbol . Mittels diesem wird die Ansicht immer auf das markierte Objekt zentriert.

Schritt 3 Objektbearbeitung

Objekte wie z. B. Bäume, Häuser, Wände usw. werden im Editor immer in grüner (können mittels Kollisionsabfrage in den Track eingebunden werden) oder in blauer



Farbe angezeigt und können mittels des Symbols  markiert, dann verschoben

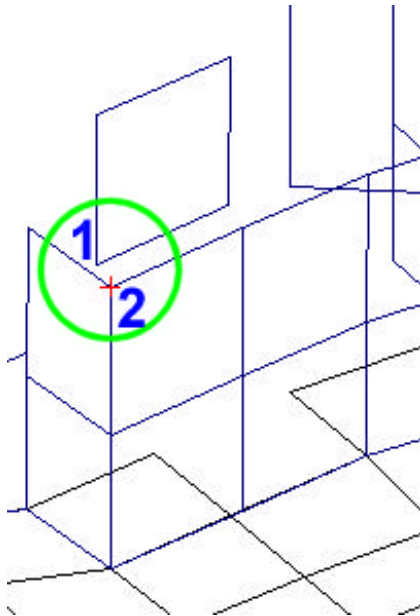


, kopiert  und ausgeschnitten  werden.




Natürlich lassen sich aus kopierten Objekten später auch neue andere Objekte wie z.B. größere Häuser herstellen. Hierzu muß das passende Objekt einfach kopiert



und an den neuen Platz verschoben  werden. Danach kann man mit dem Symbol  einzelne Rechtecke (Polygone) des Objektes kopieren, löschen und verschieben

(so wie bei Objekten). Hat man erst einmal ein einzelnes Rechteck des Objektes an die neue Position verschoben, kann man jetzt die Ecken mit dem Ursprungsobjekt verbinden. Hierzu schaltet man in den Point Modus  und aktiviert gleichzeitig noch die Merge Funktion . Als nächstes muß man nur die Ecke markieren an die später das neu erstellte Rechteck angefügt werden soll (nach dem Anklicken wird die jeweilige Ecke mit einem roten Kreuz markiert). Jetzt muß man nur noch mit gedrückter STRG Taste die anzuschließende Ecke des neuen Rechteckes anklicken und schon werden die beiden Ecken verbunden.

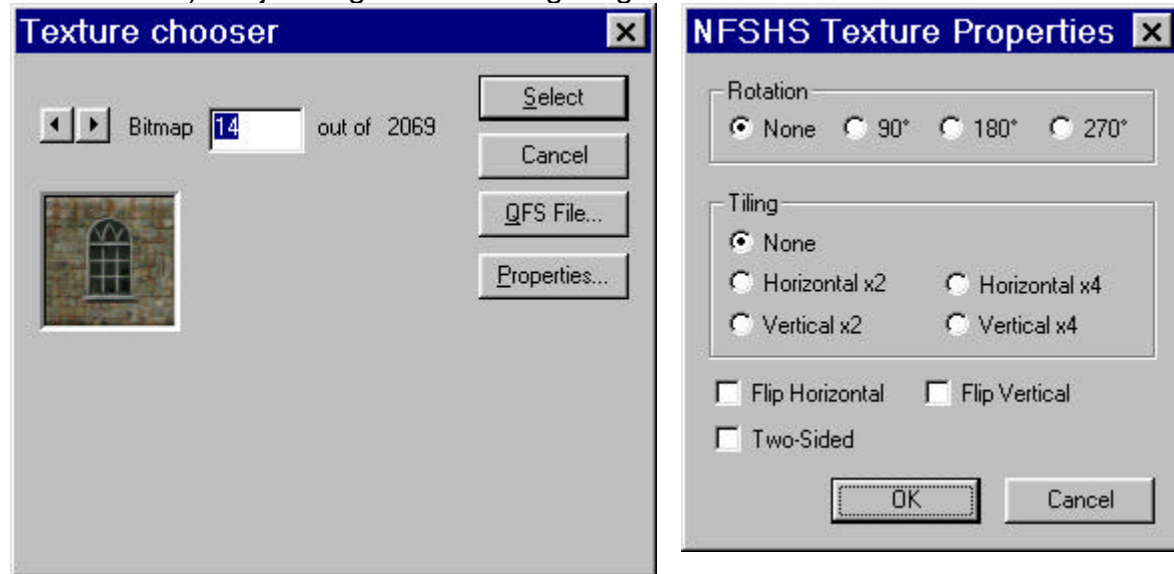


In der nebenstehenden Grafik könnt Ihr noch einmal das zuvor beschriebene Verfahren nachvollziehen. Punkt 2 ist der obere linke Punkt des Ursprungsobjektes (z.B. Hauswand). Punkt 1 ist der untere linke Punkt des zuvor kopierten

Rechteckes (wurde erst mittels  angeklickt und dann nach oben mit  verschoben). Jetzt werden die beiden Punkte 1 und 2 nur noch verbunden - mit  und gedrückter STRG Taste.

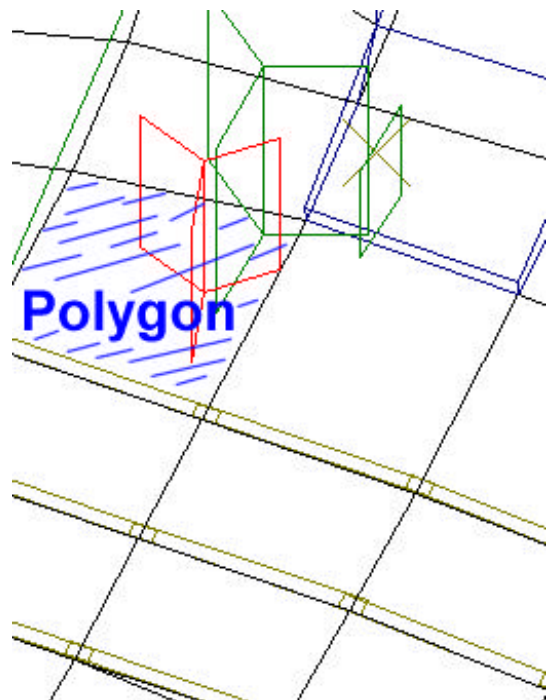
Ihr müßt jetzt natürlich nur noch mit den anderen Ecken genauso verfahren. Mittels dieser Vorgehensweise läßt sich ein Objekt vielseitig erweitern oder vergrößern. Natürlich ist dieses Verfahren auch später auf jedes Rechteck im Track anwendbar. Die einzelnen Rechtecke eines Objektes können auch mit passenden Texturen versehen werden. Hierzu muß das Rechteck markiert werden  (rot markiert) und dann der Befehl  aufgerufen werden.

Im nun erscheinenden Fenster (Grafik links unten) können mittels der Pfeile andere Texturen zugewiesen werden. Außerdem kann über den Button **Properties** (Grafik rechts unten) die jeweilige Ausrichtung eingestellt werden.



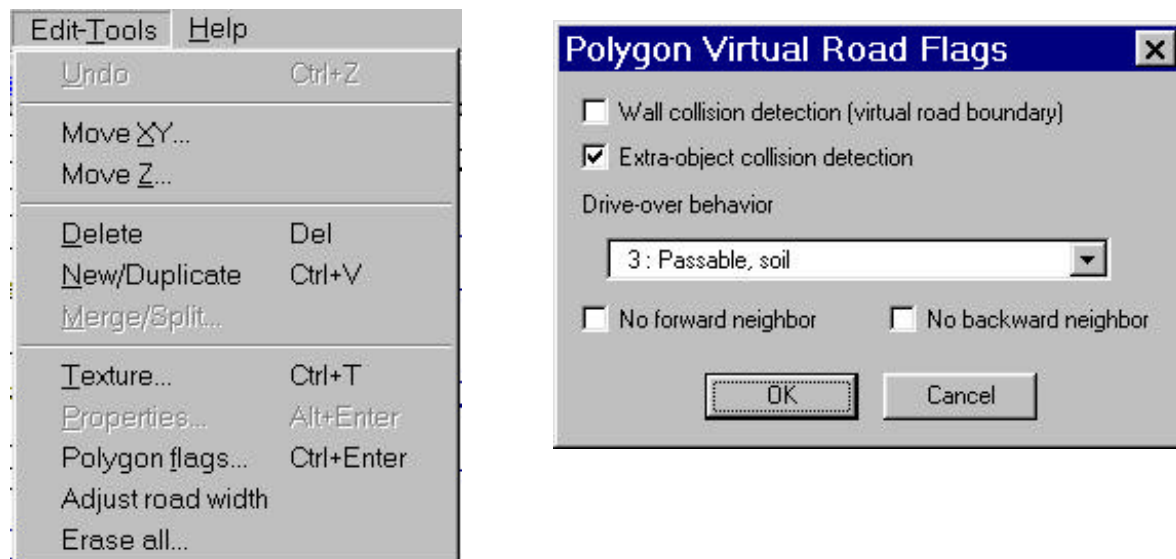
Schritt 4 Kollisionsabfragen von Objekten

Die zuvor neu eingefügten Objekte - insbesondere die grünen Bäume - können jetzt auch noch als Hindernisse in den Track eingefügt werden (siehe Grafik rechts). Nehmen wir einmal an Ihr wollt einen befahrbaren Seitenstreifen mit zusätzlichen Bäumen (z.B. beim Landstraße Track - die grünen Baumobjekte kopieren) versehen. Nach der Plazierung (siehe das rote Baumobjekt) muß dem unter dem Objekt liegenden Rechteck (Seitenstreifen) mitgeteilt werden, daß es nun beim Überfahren aufpassen soll ob das Auto gegen den Baum fährt. Das geht folgendermaßen : zuerst wird das Rechteck unter dem Objekt markiert (ist in der Grafik mit blauer Schraffierung hervorgehoben) und dann wird der Befehl



Edit Tools - Polygon Flag (siehe Grafik unten links) aufrufen.

Hier muß jetzt nur noch das Häkchen bei **Extra-object collision detection** (siehe Grafik unten rechts) gesetzt werden - Fertig!





Bei Schildern sieht die Sache ein wenig anders aus. Man kann zwar diese genauso wie z. B. Bäume als Objekte bearbeiten und auf der Fahrbahn plazieren, aber leider gibt es beim Einsatz eine kleine und entscheidende Einschränkung. Schilder lassen sich nur in unmittelbarer Nähe der Original - Fahrbahn mittels Kollisionsabfrage ansprechen. Dies bedeutet, daß eine Kollisionsabfrage auf Ersatzstrecken neben der Hauptstraße nicht möglich ist. Beim „Umfahren“ dieser Schilder auf einer selbst erstellten Nebenstraße wird das Schild nicht beschädigt bzw. es fliegt nicht in die Luft wie es z.B. bei den Hauptstraßen der Fall ist.

Allerdings lassen sich auch aus den Schildern noch andere sehr nützliche Objekte herstellen. Man kann z. B. aus einem Schild einen 3 D Stützbalken herstellen (siehe meinen Landstraße Track), der im Track auf der Hauptstrecke steht und dann beim Anprallen des Autos sich wie ein normales Schild verhält, in die Luft fliegt und dann wieder beschädigt auf der Fahrbahn liegen bleibt. Oder man baut eine Schranke, einen Zaun am Seitenstreifen die/den man umfahren kann. Der Phantasie sind hier keine Grenzen gesetzt - außer der zuvor beschriebenen Beschränkung der Kollisionsabfrage.

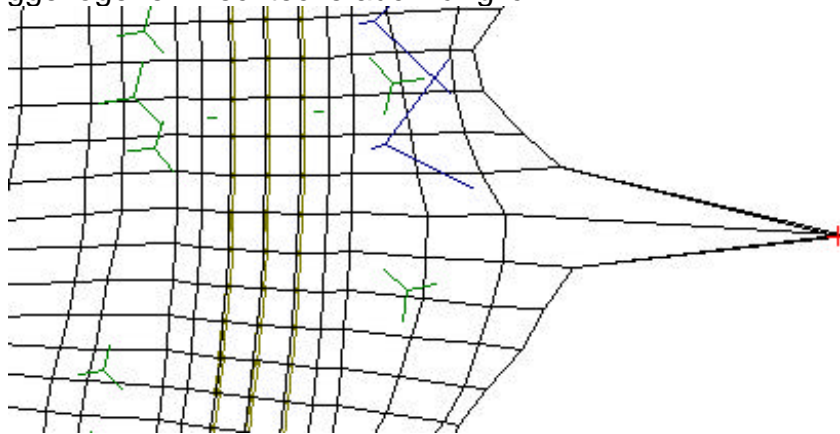
Schritt 5 Seitenstreifen / Zusatzwege

Mit dem zuvor beschriebenen Verfahren lassen sich auch Seitenstreifen erweitern oder sogar zusätzliche Ersatzstrecken einbauen.





Ihr müßt hierzu nur die jeweiligen Rechtecke markieren (im neuen T3ED kann man mit gedrückter STRG Taste mehrere Rechtecke gleichzeitig markieren) und dann wieder den Befehl **Edit Tools - Polygon Flag** aufrufen. Im nun erscheinenden Fenster kann man die Eigenschaften der neuen Fahrbahn mittels Auswahl des **Dropdowns** (siehe Grafik oben rechts) einrichten (z.B. snow = Schnee, dust = Sand usw.). Die jeweiligen neuen Abschnitte der Strecke sind jetzt befahrbar. Wem das

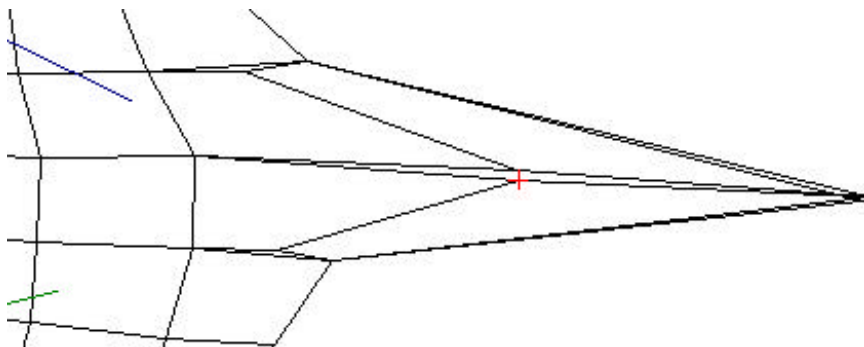
nicht reicht, der kann jetzt natürlich auch noch die sichtbaren Seitenstreifen vergrößern. Wie das geht? Zuerst müßt Ihr in den Point Modus  schalten. Danach kann man die seitlichen Begrenzungen der Strecke nach rechts oder links (je nach gewünschter Erweiterung) verschieben .

Markiert hierzu einfach einmal den äußersten Point eines Blockes z.B. rechts und zieht diesen dann nach rechts außen (siehe Grafik unten). Wie Ihr jetzt sehen könnt wird das jeweilige Rechteck verzerrt (d.h. die jeweilige Ecke wird nach außen verschoben). Jetzt könnt ihr auch die restlichen Ecken oder die benachbarten Points noch verschieben. Es wurde zwar die Strecke seitlich vergrößert aber dafür sind die jeweils langgezogenen Rechtecke auch länglich



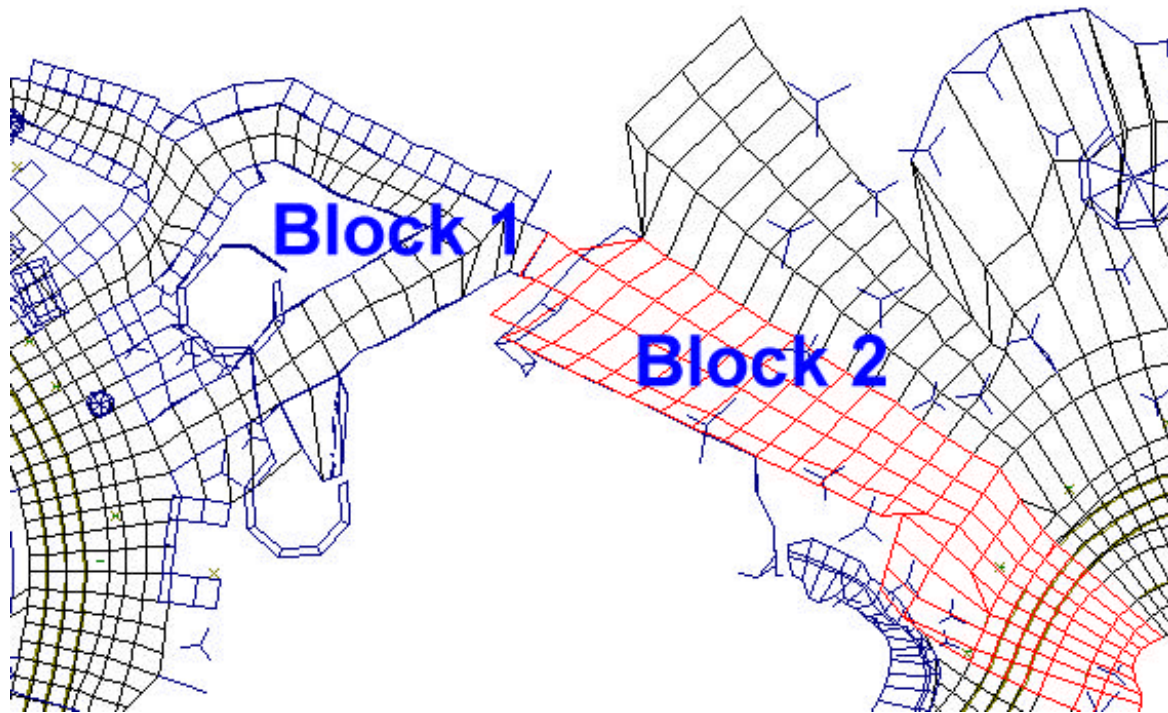
größer als die ursprünglichen. Außerdem werden bei diesem Verfahren auch die jeweiligen Texturen gestreckt. Um dieses optische Manko zu beheben gibt es im



Editor eine nützliche Zusatzfunktion. Mittels des Symbols  könnt Ihr die langgezogenen Rechtecke (vorher mit  markieren) halbieren. Die nun in der Mitte erscheinenden Ecken (siehe Grafik unten) lassen sich wie zuvor anhand der Objekte beschreiben, unter Zuhilfenahme von  (siehe Schritt 3) mit den benachbarten Eckpunkten verbinden. Außer dem zuvor beschriebenen Verfahren lassen sich auch die einzelnen Punkte durch das Symbol  verschieben. Hierbei werden auch die benachbarten Punkte mitverschoben. Hier gilt natürlich wie immer „Probieren geht vor Studieren“.



Schritt 6 Ersatzstrecken, Einsatz von Zäunen und Wänden

In Schritt 5 konntet Ihr den erweiterten Streckenbau lernen. Im nun beschriebenen Abschnitt geht es um die Verfeinerung des Erlernten. Wenn Ihr möglichst schnell ohne viel Zusatzarbeit eine neue Zusatzstrecke einbauen wollt, könnt Ihr auch nur zwei bis vier seitwärts angebrachte Rechtecke verlängern und an andere Blöcke des Track anfügen (siehe Bild unten).



Im obigen Bild seht Ihr eine solche einfache Zusatzstrecke. Beim Block 1 wurden einfach zwei (insgesamt sind es drei Zusatzstraßen) seitlich angebrachte Rechtecke nach außen verlängert und dann mittels der  Kopierfunktion in mehrere Rechtecke verkleinert (**nicht vergessen**: anschließend mit  die Eckpunkte verbinden). Am Schluß wurden die Endrechtecke mit denen aus Block 2 (rot markiert) verbunden. Leider gibt es jetzt beim Übergang von Block 1 auf Block 2 eine sog. unsichtbare Wand durch die man nicht hindurchfahren bzw. sehen kann. D.h. beim Befahren des ersten Streckenabschnitts von Block 1 bleibt das Auto plötzlich direkt beim Übergang auf Block 2 hängen und man kann den anschließenden Verlauf nicht sehen. Um dieses Problem zu beheben gibt es den Befehl **Edit-Tools - Adjust road width**. Leider funktioniert er nicht immer und die Wand verschwindet nicht. Man kann daher nur versuchen nicht zu weit entfernte Blöcke zu verbinden (NFS 4 zeigt während des Spiel ca. 3-4 Blöcke immer hintereinander an - d. h. beim Verbinden von Blöcken die max. 2 Blocks entfernt sind, dürfte es keine Sichtbeeinträchtigungen geben). Natürlich müssen auch die Eigenschaften (siehe Grafik rechts oben folgende Seite) der zu verbindenden Rechtecke beim Übergang geändert werden. Hierzu muß der Befehl **Edit Tools - Polygon Flag** aufgerufen und dann das Häkchen bei **Wall collision detection** entfernt werden.

Die jeweiligen Seitenstraßen können auch durch Wände oder Zäune begrenzt werden. Hierzu müßt Ihr nur schon vorhandene Zäune im jeweiligen Block kopieren und dann an die Ersatzstrecke anpassen. Nach dem Anpassen muß jetzt noch die Kollisionsabfrage integriert werden. D.h. den angrenzenden Rechtecken muß bekannt sein, daß dort jetzt ein Zaun oder Mauer ist. Ihr fragt euch sicher wie das geht? Naja eigentlich habe ich es zuvor schon anhand von Objekten beschrieben. Ihr müßt nur die nicht befahrbaren Rechtecke markieren und dann wieder den Befehl **Edit Tools - Polygon Flag** aufrufen. Im nun erscheinenden Fenster sollte für die **äußersten** Rechtecke folgendes eingestellt sein :



Für die **inneren** Rechtecke (also die zwischen der Hauptstraße und der Nebenstraße) sollte es folgendermaßen aussehen (siehe Bild links unten):




Damit es später beim „Anfahren“ der seitlichen Begrenzungen (Zäune, Wände, ...) keine Problem gibt muß man noch die Kollisionsabfrage für die betreffenden Polygone aktivieren (siehe Bild rechts unten - Option **Wall collision detection** - rotes Rechteck).

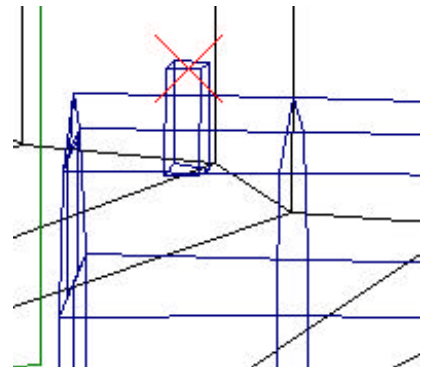


Anmerkungen: Mittels Zaun-Objekten (blau im Editor ersichtlich) lassen sich auch Mittelleitplanken für eine Autobahn erstellen. Leider hat das Einfügen dieser Hindernisse negativen Einfluß auf den Verkehr bzw. die Computergegner. Der Verkehr und die Computergegner bleiben zwar mittels einer Kollisionsabfrage an den Planken hängen und überqueren diese auch nicht, aber das ist auch schon alles. D.h. im NFS4 Trackformat ist anscheinend eine nachträgliche und künstliche Begrenzung der Orig.-Strecke nicht möglich. Dies bedeutet für den Verkehr und die Computergegner daß sie weiterhin verzweifelt versuchen werden das künstliche Hindernis (die Mittelleitplanke) zu überqueren bzw. die ganze **ursprüngliche** Fahrbahn zu benutzen. Eine Lösung dieses Problems ist mir momentan nicht bekannt. Wer hierzu mehr weis, sollte sich bei mir melden.

Schritt 7 Animationsobjekte






In der neuesten Version von T3ED kann man jetzt auch Lichtquellen und andere Ani-quellen (wie z.B. Rauch) als Objekte bearbeiten. Hierzu kann man mittels des

Objekt-Modi  das jeweilige Objekt (sieht wie ein grauer Stern aus und wird nach der Markierung rot angezeigt - siehe Grafik rechts) kopieren, verschieben oder ausschneiden. Sehr nützlich ist diese Zusatzfunktion beim Anbringen einer neuen Zusatzbeleuchtung auf Ersatzstrecken oder wenn man neue Häuser mit einem rauchenden Kamin versehen möchte.



Außer den zuvor beschriebenen Ani-Objekten gibt es noch sog. Animationsflächen wie z.B. Flüsse und Wasserfälle.

Zwar sind diese Objekte auf den ersten Blick nicht gerade als solche erkennbar, aber auch diese lassen sich ganz normal bearbeiten. Die Ani-Flächen sind meistens in blau dargestellt und bestehen aus einzelnen Rechtecken die in einer Reihe hintereinander zusammengefügt sind (Wasserläufe usw.). Mittels dieser Flächen kann man ganz schnell und gut aussehende Flußläufe in seinen Track einfügen.

Zuerst müßt Ihr einen künstlichen Fußlauf mittels Einsatz von  erschaffen (geht z.B. im  oder  Modus). Danach könnt Ihr eine kopierte Ani-Fläche an diese Untiefen anpassen (mittels  und ). Natürlich lassen sich diese Flächen auch auf der Hauptstrecke als Hindernis einbauen (z.B. bei Rallye - Strecken). Ich selber habe z. B. einen Wasserfall auf meiner Rocky Pass Ersatzstrecke (Name: Alpenpass) eingefügt, durch den man durchfahren kann.

Schritt 8 Texturen

Die Texturen kann man vor der Bearbeitung mit dem Programm **NFS Wizard** aus den jeweiligen *.qfs Dateien extrahieren. Danach können die TGA Dateien in jedem guten Grafikprogramm verändert werden (wichtig ist hierbei auf die max. Farbanzahl

von **256** zu achten - meistens). Anschließend können sie wiederum mit dem NFS Wizard in die QFS Datei (immer darauf achten, daß der Schreibschutz der Datei nicht aktiv ist) eingefügt werden. Wer einmal schon bei den Cars mit dem Alpha Kanal gearbeitet hat - wird diese Vorgehensweise natürlich auch bei den Trackgrafikdateien verwenden können. D.h. wenn Ihr z.B. Zäune oder Bäume verändern wollt, solltet ihr auch den Alpha-Kanal beachten. Ich verwende hierzu das Programm **VIV Wizard**. Hierzu wird eine **CAR VIV** Datei geladen und dann die Track TGA Datei einfach eingefügt. Nun kann man die Hauptgrafik und Alphakanalgrafik als BMP Datei abspeichern und nach der Bearbeitung wiederum einfügen. Die neue TGA Datei (die immer noch im VIV File enthalten ist) kann jetzt als ganz normale TGA Datei aus dem VIV File extrahiert und mittels NFS Wizard wiederum ins QFS File eingefügt werden. Dieses Verfahren soll nur für diejenigen als Orientierung dienen die kein teures Grafikprogramm haben, das die Bearbeitung des Alphakanals zuläßt.

Bei den Ladeschirmen kann man die TGA Dateien ohne Berücksichtigung von Beschränkungen der Farbanzahl oder des Alpha-Kanals bearbeiten.

Ladeschirme / Videowände Erklärungen:

Datei: T(Nummer des Tracks)_00.qfs

Der große Start - Ladebildschirm (Slide) befinden sich in Verzeichnis:

Data\Feat\Slide\

Datei: T(Nummer des Tracks).qfs

Die kleine Videowand (VidWall) befindet sich im Verzeichnis:

Data\Feat\vidwall\

Im Anhang 3 findet Ihr eine kleine Beschreibung über das Programm **FSHTool**. Mit diesem Programm kann man die QFS Files durch weitere Grafiken erweitern oder bestehende Grafiken ersetzen bzw. in der Größe (Auflösung z.B. 128*128) verändern.

Schritt 9 Wetterverhältnisse

Die jeweiligen Wetterverhältnisse lassen sich unter Zuhilfenahme einer sehr guten engl. Anleitung jeweils an die Trackstimmung (z. B. Wüstenlandschaft usw.) anpassen. Die Anleitung habe ich im Anhang 2 beigefügt. Wichtig ist hierbei auch die Bearbeitung der Hintergrundbilder (Datei **Sky.qfs** - Bearbeitung wie im Schritt 8 beschrieben). Wobei auch wiederum der Alpha-Kanal seine Bedeutung hat. Ich erspare mir wg. der guten engl. Anleitung eine weitere Erklärung der vers. Einstellungen der INI Dateien. Notfalls könnt Ihr auch die Original - Dateien belassen.

Schritt 10 Anmerkungen

Ich hoffe, daß mit dieser Anleitung eine kleiner aber nützlicher Schritt zum großen Thema „Trackbearbeitung“ getan wurde um Euch die Arbeit erleichtern zu können. Wie bei jedem so vielseitig angelegten Themengebiet, sollte man sich auch hier einige wichtige Grundkenntnisse aneignen (wie z.B. Grafikbearbeitung, Design mittels eines CAD Programms usw.) bevor man sich auf die Bearbeitung eines Tracks stürzt. Wer vorher schon einmal eigene Cars gebaut hat, wird es sich viel leichter tun als derjenige der bei NULL anfängt. Ein Tipp daher an jeden Neuling. Versucht erst einmal euch gute Strecken von anderen Designern zu besorgen und dann diese Beispiele zu studieren. Es ist manchmal echt verblüffend wenn man beim Betrachten dieser Tracks neue Ansätze und Lösungen findet an denen man vorher vielleicht schon seine Zähne ausgebissen hat. Ansonsten gibt es natürlich auch die Möglichkeit die jeweiligen Designer per Mail um Hilfe zu bitten oder in den zahlreichen Foren (ist meistens die bessere Alternative weil nicht jeder Designer die Zeit hat alle Fragen ausführlich zu beantworten ;-) entsprechende Hilferufe aufzugeben.

Ich wünsche daher Euch viel Spaß beim Umsetzen ☺

Gruß

Gert Steidle (alias FANGIO)

Anhang Inhalt:

1. NFS Seiten bei denen man die betr. Tools downloaden kann usw.
2. engl. Anleitung für die INI Dateien (Wetterverhältnisse)
3. Beschreibung des FSHTools
4. Point to Point Strecken - kurze Anleitung
5. NFS String Editor - Textdateien verändern
6. Vidwalls - Streckenführung anzeigen lassen

Anhang 1:

Nützliche deutsche NFS Seiten auf denen man die notwendigen Tools findet:

www.np4spd.de www.nfsfactory.de www.nfs2000.de www.nfsworld.de usw.

Dieses Tutorial wurde auch für die Aktion "Gastspiel" des wsm tuningvillage www.ctb.notrix.de geschrieben (vielen Dank für die Anregungen usw. ☺).

Natürlich könnt Ihr auch bei mir einmal wieder nachschauen ob es eine neue Version dieser Anleitung gibt ☺. **www.fangio.de**

Anhang 2: engl. Anleitung für die INI Dateien (Copyright by John Arnstrom, aka Zacabeb)

INI file Tutorial

1 The INI files

1.1 What's in the INI files?

The INI files contain weather and light specifications for the tracks in NFS:HS. They originate from the HRZ files found in NFS3, but with the enhancements made in NFS:HS they have been augmented with new features.

There are four INI files; one for each of the time/weather combinations available. The names of the files correspond to the conditions, as follows;

TR.INI	Daytime
TRW.INI	Daytime with weather turned on
TRN.INI	Nighttime
TRNW.INI	Nighttime with weather turned on

1.2 What do I need to edit the INI files?

All you need to edit the INI files is a text editor; by default Windows will use Notepad for editing INI files. If you have selected a different text editor, make sure you save the file in ASCII format or it will appear corrupted to the game and crash it.

1.3 What can I do by editing the INI files?

By editing the INI files, you can change the weather, light and sky color among other things. You can add thunder, rain or snow, change the position and size of the sun, alter the fog and edit the color and brightness of the light sources on the track. With careful editing you can change the atmosphere of a track only by changing values in the INI files - creating a misty summer morning or a chilling night scene. By editing the INI files you can add the final touch to the tracks you make - letting them appear just the way you want.

2 Variables in the INI file

2.1 Horizon Strip [strip]

The strip contains information on the horizon and its properties.

hrzShouldDrawBlack=b

This is a boolean which specifies whether the screen should be filled with black before each frame is drawn, in which case the value is **1**, if not the value is **0**. When set to 1, no pixmap or horizon is drawn. This is useful if the track does not contain any horizon or pixmap (see below).

This feature is not used on any of the original tracks in NFS:HS, the ones from NFS3 included. In NFS2 however, it was used on Monolithic Studios, and in NFS3 for the PlayStation it was used on the Easter Egg track Caverns. There is also a corresponding function in the .FRD files (where the track model is kept), which specifies the ShouldDrawBlack variable individually for each slice of the track. This is used in the Temple on the track Lost Canyons.

hrzMirror=b

This is a boolean and tells if the *pixmap* (see below) should be mirrored in the two halves of the horizon (**1**) or if it should be repeated with the same orientation in the two halves (**0**). For prominent horizons such as hills, it is often best to make a pixmap which can be repeated seamlessly in the two halves and set *hrzMirror* to 0, otherwise the mirroring of the two halves will cause a Rorschach effect at the seam. When you want there to be a scene only in the one end of the horizon, draw it in one end of your pixmap set *hrzMirror* to 1.

hrzHasPmx=b

This specifies whether the track has a *pixmap*, i.e. a background landscape. This is a boolean value set to **1** if a horizon pixmap is used, and a **0** if not. This can be used to leave out the pixmap if none is needed, such as on a track set on an island in the middle of the ocean.

hrzBottomYOff=y

This specifies the offset for the pixmap and clouds relative to the logical horizon (the actual middle of the 3D world itself). Normally this offset has a negative value, placing it below the logical horizon. In most cases, a value at around **-1000** to **-1500** is used.

hrzGouraudHeight=h

This is a single? precision floating point value indicating the height of the Gouraud shaded region of the sky and ground.

hrzGouraudMiddle=y

This is a single? precision floating point value indicating the position of the center of the Gouraud shaded region of the sky and ground.

hrzPmxTop=y

This value specifies how high in the 3D world the top of the pixmap is relative to the horizon offset, *hrzBottomYOff*. The pixmap will appear closer or further away depending on the radius of the horizon; *hrzRadius*, so you will probably want to adjust the latter before you adjust *hrzPmxTop*. Commonly the value is at around **1500.0-2000.0** for a horizon with a radius at around 2000.0 and a *hrzBottomYOff* at -1000.

hrzPmxBottom=y

This specifies the bottom of the pixmap in the same manner as *hrzPmxTop* specifies the top. Commonly the bottom of the pixmap is slightly above the physical horizon and lying at **500-1000**, placing the bottom of the pixmap near the logical horizon. When this is the case, the pixmap's bottom should be a bit below the horizon offset with a margin of about **50** units, to hide the seam where the sky and ground meet at the logical horizon.

hrzSunColor=[r,g,b]

This specifies the RGB color for the sky at the position of the horizon. The RGB values are expressed in normal 24-bit format and can each range from **0-255**, where **0** is dark and **255** is bright.

hrzOppositeSunColor=[r,g,b]

This specifies the RGB color of the sky at the point opposite of the sun.

hrzSkyTopColor=[r,g,b]

This specifies the RGB color at the top of the sky. This will gradually blend with the *hrzSunColor* and *hrzOppositeSunColor* values towards the horizon, above the top of the Gouraud shaded area the sky will be a solid color.

hrzEarthTopColor=[r,g,b]

This specifies the RGB color at the top of the ground visible outside the track area.

hrzEarthBotColor=[r,g,b]

This specifies the RGB color at the bottom of the ground visible outside the track area.

hrzfRadius=r

This specifies the radius of the horizon, and affects the appearance of the pixmap, clouds and shading. Common values range from **1500.0-2500.0**.

hrzAngle=?

This specifies the rotation around the horizontal axis, Theta or θ , of the horizon in a counter-clockwise direction and is expressed in radians. A radian is a fraction of pi, hence full circle or 360° roughly equals pi (3.14), and half circle or 180° half of pi (1.57). To convert degrees to radians, use the following formula:

$$r=0.00873*d$$

where d is the angle in degrees and r the result expressed in radians. Round the result off to 2 decimals. Note that you cannot use this formula directly in the INI file; you must calculate the result and insert it 'as is'.

2.2 Clouds [clouds]

cloudType=t

This is an integer and indicates the type of clouds to be applied. A value of **0** means no clouds will appear, a value of **1** means the clouds will be blended with the sky and finally a value of **2** means they will be blended as additive.

cloudBright=b

This is an integer and specifies the intensity, or brightness of the clouds. The value can range from **0** to **255**, where **0** is dark and **255** bright.

cloudVariance=v

This is an integer describing the variance of the clouds, i.e. the contrast between the bright and dark areas. Values may range from **0** to **255**, where **0** means no variance and **255** full variance.

cloudNimbus=n

This is a fixed point variable and tells if the clouds produce a Nimbus effect in thunderstorms. The Nimbus effect is simply the refraction of light within the clouds which make them light up with the flashes. Values may range from **0.00** to **1.00**, where **0.00** means there is no Nimbus effect at all and **1.00** there should be a strong Nimbus effect.

cloudFog=f

This is a boolean value which simply tells whether the fog on the track should affect the clouds, in which case it is set to **1**; if not it is set to **0**.

cloudVisualHeight=h

windSpeed=s

This is an integer specifying the windspeed. The windspeed affects the drifting of the clouds, as well as special effects such as rain and smoke. A value of **0** means the clouds and effects do not drift. For slow drift, use values around **20**, for faster drifts use values between **60** and **90**.

windDir=a

This is an integer describing the direction of the wind (in a counter-clockwise direction), again affecting both the clouds and special effects. Unlike the other angles in the INI file, this is expressed in degrees and can thus range from **0** to **359**.

cloudYOffset=200.0

cloudHeight=500.0

cloudUscale=1.0

2.3 Lightning [lightning]

These two values describe if lightning should occur and if so, how often. When *lightningChance* is set to a value above 0, there is *lightningChance* percent chance lightning will strike in *lightningOffTicks* time.

If you want frequent flashes, set the percentage of *lightningChance* reasonably high and *lightningOffTicks* low. If you want occasional flashes, set both *lightningChance* and *lightningOffTicks* low. Note that if you set the *lightningChance* too high, flashes will occur with a distinct recurrence which might appear unnatural.

lightningChance=c

This is an integer and tells, in percent, how great the chance is a flash will occur in the period specified in *lightningOffTicks*. Normally, keep this value below **80** for a naturally random recurrence and adjust the *lightningOffTicks* value to determine the periodicity of the flashes. If you don't want any lightning set this to **0**.

lightningOffTicks=o

This specifies the periodicity of the lightning in ticks.

2.4 Distance Fog [fog]

Fog comes in two forms; global and dynamic. The global fog affects the entire track in all areas where no dynamic fog zone has been defined. Dynamic fog zones let you add specific fog to a region of the track, and also fade smoothly between the different fogs.

fogColor=[r,g,b]

This specifies the color of the global fog in 24-bit RGB. Each value is an integer and may range from **0** to **255** where **0** is dark and **255** bright. The scenery is blended towards this color by the amount specified in *fogDensity*.

fogDensity=d

This specifies the density of the fog in percent, ranging from **0** for no fog at all up to **100** for completely opaque fog.

fogHorizon=h

This is a fixed point value and tells if the horizon should get affected by the fog, ranging from **0.00** for no effect up to **1.00** for full effect. Normally keep this at around **0.70-0.80**.

fogNumRegions=n

This value is an integer and specifies how many fog regions have been defined, thus it should equal the number of dynamic fog regions in your INI file.

[fog region n]

startSlice=ss

s_color=[sr,sg,sb]

s_density=sd

centerSlice=cs

c_color=[cr,cg,cb]

c_density=cd

endSlice=es

e_color=[er,eg,eb]

e_density=ed

These are dynamic fog regions which specify a fog unique to a certain part of the track. *fog region* specifies the number of the region in the list and should start with **0** for the first region, followed by **1,2,3...** etc for each subsequent fog region. *startSlice*, *centerSlice* and *endSlice* specify three positions along the track for the fog region. At each position the fog will have the properties specified in the **_color* and **_density* variables below the respective *Slice* entry. Between these positions, the game will gradually blend the nearest two values to fade between the different fogs.

The positions along the track specified in the *Slice* variables correspond to the rows of polygons making up the track. To determine the desired slice, you can open the track in T3ed and select Segment mode. Then move to the block you want and select it. In the lower right corner of the window you will see the number of the block. To determine the number of the first slice of that block, multiply the block number by **8**. This will give you a preliminary value to use which you can fine tune later.

Note that where two fog regions join, they should both have the same color and density at the joining ends. Otherwise the fog will instantly switch from the one value to the other as you drive inbetween the regions. Similarly, where either end of a fog region does not join another region, they should have the same values as the global fog specified in *fogColor* and *fogDensity* to avoid switching.

If you need to cover a large section of the track with an even fog area, use two fog regions where the values at the opposing ends are the same as their adjacent fogs, and the four values in between have the values you want for the large fog area. Then set the *centerSlice* of both fog regions **10-50** slices in from the *Slices* at the opposing ends, so to create a transition region at either end. Finally set the two adjacent *Slices* of the two fog regions to any value in between the two *centerSlices* to make them join and complete the fog area.

2.5 Weather [weather]

type=t

This specifies the type of weather to be used, where **0** means no weather, **1** means rain and **2** means snow.

startSlice=s

startSlice specifies an optional slice where the weather should start and can be used if you want to restrict the weather to a certain region of the track. Again, the *Slice* value refers to the slices of the track. Look above in the explanation of fog regions for information on how to determine the desired *Slice* value. If you want the weather to cover the entire track, set this variable and *endSlice* to **0**.

endSlice=e

This variable is connected to the *startSlice* and specifies the slice at which the weather should end.

fade=f

When you have restricted the weather to a certain part of the track using *startSlice* and *endSlice*, this specifies a number of slices at either ends of the weather region across which the weather should be faded, that is gradually become denser as you drive into it. Reasonable values range from **5** for a quick transition up to **50** for a smooth transition.

stayTimeOn=t

Together with the other *Time* variables, this lets you choose to have the weather come and go periodically. *stayTimeOn* specifies how long the weather should remain active before the break, and is counted in ticks.

fadeTimeOff=t

When you have chosen to have the weather come and go periodically in the *stayTimeOn* and *StayTimeOff* variables, this specifies a time during which the weather should fade out. Again, this is measured in ticks.

stayTimeOff=t

Like the *stayTimeOn* variable, this is used for making the weather break periodically and specifies how long the break should last before the weather comes back on. If you want the weather to remain constant, set this variable to **0**.

fadeTimeOn=t

While the *fadeTimeOff* variable specifies a time over which the weather should fade out, this specifies a time during which the weather should fade back in after the break. As is the case with the other *Time* values, this is measured in ticks.

density=100

This specifies the density of the weather in percent. 0 means there will be no downpour, and 100 that the downpour will be as dense as the game permits.

2.6 Ambient Light and Car Shadows [light]

AmbientRed=r

AmbientGreen=g

AmbientBlue=b

These specify the ambient light for the track and are measured in percent. At a value of **0**, the track will be completely dark, while at a value of **100** it will be as bright as has been specified in the internal shading values of the track file itself. If you want to make the track dark, choose values which lie at around **20** or more, anything less will make the track unplayable when the headlights have been busted. If you want the track to be fully lit, set the values to **100** or near.

lightTheta=?

lightRho=?

These two values are used to determine the source of the incoming light for the car shading. Both have two decimals and specify the angle around the horizontal axis (Theta) and vertical axis (Rho). Both are expressed in radians. *lightTheta* expresses the horizontal angle in a counter-clockwise direction, where **0.00** equals North, **0.79** equals West, **1.57** equals South and **2.36** equals East. *lightRho* expresses the vertical angle ranging from **0.00** for Zenith where the sun is placed at the top of the sky, and **0.39** where it lies at the horizon. Normally set *lightTheta* to the value opposite that of the *angleTheta* of the Sun (see below) and *lightRho* to the same value as the *angleRho* for the Sun.

carshadow=[i,r,g,b]

This specifies the intensity and color of the car shadow. *i* tells the intensity of the shadow and may range from **0** (none) to **255** (fully opaque). *r,g,b* specify the color in normal 24-bit RGB format, thus each value can range from **0** (dark) to **255** (bright). For most cases, set *i* to around **100**, and each of *r,g,b* to **255**.

2.7 Special Effects [special]

These variables specify certain special effects which can be used on the tracks.

meteor=b

This is a boolean and tells if there should be 'shooting stars' on the track, in which case it should be set to **1**, otherwise it should be set to **0**.

flare=[r,g,b]

When the track features a sun, this specifies the intensity of the sun and lens flares which appear when the sun is in view. Each of *r,g,b* can range from **0** to **255** and specify the intensity of the respective color, where **0** means no flare and **255** full intensity. If you don't want a flare to appear, for instance if you have a clouded sky with only a vague sun or a moon at night, set each of *r,g,b* to **0**. For an intense flare, set each of *r,g,b* to **255**.

fireworks=f

fireworksAngle=?

This lets you add fireworks to the track, but it is a restricted effect which depends on the date the game was installed. If the game was installed before the 4th of July, fireworks will only have appeared on the 4th of July, if it was installed later than the 4th of July fireworks will always appear when this option is set.

fireworks tells whether fireworks should appear. A value of **0** means no fireworks. *fireworksAngle* expresses the Rho or vertical angle of the fireworks and is expressed in radians, where **0.00** places them at the top of the sky and **0.39** at the horizon. However, the Theta or horizontal angle of the fireworks cannot be specified; they will always appear opposite the Sun.

2.8 Sun [sun]

This contains information on the sun of the track, however the sun may also be a moon (!).

hasSun=b

This is a boolean and indicates whether the track should have a sun (**1**) or not (**0**). If this variable is left out on tracks made from the NFS3 tracks, the sun will appear ???where???

angleTheta=?

angleRho=?

These specify the position of the sun on the horizon in radians. *angleTheta* expresses the horizontal angle and may range from **0.00** to **3.14**. *angleRho* expresses the vertical angle and may range from **0.00** to **0.39**. To make the sun appear within the view, normally set the *angleRho* between **0.10** and **0.30**, however it might be obscured by the *pixmap* if the latter is high, and the *inFront* variable (see below) has been set to 0.

radius=r

This is a decimal value expressing the radius of the sun. For a normal sun, set this to around **200.0**, for a big sun set it to around **300.0**, and for an oversize novelty sun set it to **500.0** - or a bit above :).

rotates=b

This boolean tells if the sun should rotate around its own axis as you turn, in which case it should be set to **1**. If your sun is a moon (!), for obvious reasons you might not want to use this, so then set it to **0**.

additive=b

This boolean tells if the sun should be pasted on top of the sky (**0**) or blended with it (**1**).

inFront=b

A boolean which specifies whether the sun should appear in front of the *pixmap* (**1**) or behind it (**0**).

2.9 Track Glows [track glows]

The track glows contain the specifications of the track's glows, which are the light sources seen on lamp posts and similar. The values specified in the track glows were previously common for all the tracks, but since the inclusion of specific variables in the INI files their appearance can be edited. Normally you will find most of the glows to have the same values on all the original tracks. Not all of the glows are used, on many tracks only two or three are.

glow*=[i,r,g,b], f, t, ?, d

These arrays specify the color and behaviour of the glows. A total of **32**, named *glow0* through to *glow31* appear in the original files, however not all of them are used on every track.

[i,r,g,b] specify the intensity and RGB color of the light, and can all range from **0** to **255**. An intensity of **0** means the glow will not appear, and can thus be used to put out light sources which have become misplaced when editing the track layout.

f is a boolean and tells if the glow is solid (**0**) or flashes (**1**), in which case ***t*** tells how long the glow should wait in between changing its state. At a low ***t*** value, such as **2** or **3**, the glow will flash very quickly. I would not recommend using values lower than **5** as the resulting strobe effect can appear disturbing. At a slightly higher value, such as **6** or **7**, the flashing is slow enough to be suitable for objects such as flashing stoplights or similar.

d is a fixed point value with two decimals, specifying the diameter of the glow itself.

There are also three unknown variables which only appear in some of the INI files, and always with the same values (**0,0**), **T0**. Experimenting with these values does not seem to have any noticeable effect other than in some cases eliminating the respective glow. Since these variables are not present in many of the INI files and when they appear always have the same consistent values, it can be assumed they do not have any proper use in the game. These variables might be part of features which were left out or never put in. They are however redundant and can be left out.

Anhang 3:

Bearbeitung des QFS - Files mit dem FSHTool

Mit dem FSHTool ist es möglich die Bildformate der Grafiken in QFS Files zu verändern und weitere Grafiken einzufügen.

Anhand dem „NFS Redrock Ridge Track“ möchte ich diese sehr nützlichen Funktionen einmal erklären.

Zuerst müsst Ihr das QFS - File mit dem FSHTool verknüpfen (siehe auch die dem Tool beigefügte Readme Datei - unter Zuhilfenahme der Funktion „**Öffnen mit**“ im MS-Explorer usw.)

Erst danach lässt sich das QFS - File durch einen Doppelklick in ein Unterverzeichnis (z. B. tr0) entpacken. Danach befinden sich in diesem Verzeichnis alle im QFS - File enthaltenen Grafiken (als BMP Datei).

Diese Grafikdateien lassen sich nun mit jedem Standardgrafikprogramm bearbeiten (bitte hierbei auf die jeweilige Farbanzahl - meisten 256 - achten) und in einer anderen Auflösung (z.B. 64*64 wird durch eine bessere Grafik im 128*128 Format ersetzt) abspeichern.

Leider kann man den Alpha - Kanal in den betr. BMP Dateien nicht mitbearbeiten. Deshalb ist nach dem Zusammenfügen eine Nachbearbeitung mit z.B. Paintshop Pro (ab Version 5) notwendig (TGA Datei mittels NFSWizard aus der QFS Datei exportieren, Alpha-Kanal einfügen und nachher importieren usw.). Dazu aber nachher mehr.

Vorerst wollen Wir uns mit der Erweiterung oder Veränderung des QFS Files beschäftigen.

In dem gleichen Verzeichnis, in dem die Grafiken hinterlegt sind, gibt es noch die Datei „index.fsh“. Einen kleinen Abschnitt dieser Datei (hier als Beispiel die von meinem Deadly Outback 2 in 1 Track) habe ich im folgenden eingefügt. Anhand meiner kleinen Beschreibung könnt Ihr die Bearbeitung dieser Datei studieren.

Beispieldatei mit Beschreibungen:

FSHTool generated file -- be very careful when editing

C:\Programme\Electronic Arts\Need for Speed\Data\Tracks\Redrock\tr0.qfs

QFS

SHPI 249 objects, tag GIMX

BUFSZ 1032632

NOGLOBPAL

0000 0000.BMP

BMP 7B +1 1 1 {0 0 0 0}

!PAD 7 0000000000000000

PAL 2D 256 1 {256 0 0 0}

Hier steht die genaue Grafikanzahl im QFS File.

Wenn Ihr am Ende der Datei weitere Grafiken einfügt müßt Ihr hier (rosa markiert) entsprechend die Anzahl erhöhen. Sonst bekommt Ihr eine Fehlermeldung.

0205 0216.BMP

BMP 7D +0 128 128 {0 0 0 0}

alpha 0216-a.BMP

Natürlich lassen sich auch Alpha Kanäle (grün) der Grafiken einbinden (siehe Beispiel rechts). Hier wurde eine Grafik im 32 BIT Format (rosa) verwendet. Alpha-Kanal Bilder werden immer im Format **alpha lfd. Nr.-a.BMP** eingefügt (geht aber nicht bei 8 BIT Bildern, ist nur ab 16 BIT möglich - bitte beachten)

0206 0217.BMP

BMP 7B +1 32 32 {0 0 0 -1}

PAL 2D 256 1 {256 0 0 0}

0207 0218.BMP

BMP 7B +1 32 32 {0 0 0 0}

PAL 2D 256 1 {256 0 0 0}

0208 0219.BMP

BMP 7B +1 32 32 {0 0 0 0}

PAL 2D 256 1 {256 0 0 0}

0209 0220.BMP

BMP 7B +1 32 32 {0 0 0 0}

PAL 2D 256 1 {256 0 0 0}

Rechts seht Ihr ein kleines Beispiel wie man animierte Texturen in seinen Track, sofern dieser animierte Texturen wie z.B. ein Fluss, eine Flagge unterstützt, ins QFS File einfügen kann. Als Beispiel habe ich eine Flagge (siehe unten), bestehend aus vier Einzelgrafiken (laufende Nummern ist gelb und Grafikname ist in grün markiert) im 32*32 Format (blau), zusätzlich zu den schon bestehen Ani-Texturen in Redrock Ridge (Fluss, Wasserfall) gewählt.



0217 0218 0219 0220

0237 0248.BMP

BMP 7B +1 32 32 {0 0 0 0}

PAL 2D 256 1 {256 0 0 0}

#END

Neue Grafiken könnt Ihr einfügen, wenn Ihr einen kleinen Abschnitt, wie hier umrahmt, am Ende der Datei einfügt (vor dem WORT #END - rosa markiert). Wichtig ist hierbei, dass Ihr auch die laufenden Zahlen der Bilder - gelb ist die laufende Zahl im QFS File und grün ist die laufende Zahl der BMP Datei im jeweiligen Verzeichnis -richtig einträgt. Die Auflösung der jeweiligen Grafik lässt sich an der blau markierten Position verändern (z.B. das bisherige 32*32 wird durch ein 128*128 Format ersetzt).

Beschreibungen der einzelnen Textbefehle in der index.fsh Datei:

Beispiel 1

```
0011 0009.BMP
BMP 7B +1 64 64 {0 0 0 0}
PAL 2D 256 1 {256 0 0 0}
```

Kurzbeschreibung (Beispiel 1):

blau (0011 1. Zeile links) = laufende Nummer im QFS File
 rot (0009.BMP 1. Zeile rechts) = laufender Bildname im (Entpack-) Verzeichnis
 dunkelgrün (2. Zeile) = BMP (Bildformat) 7B (BIT Format siehe folgende Aufstellung)
 +1 (Anzahl der Erweiterungen Palette/Text/Data - hier ist es nur eine Erweiterung und zwar PAL - deshalb +1), 64 64 (Höhe und Breite der Grafik)
 rosa (3. Zeile) Format, 2D (Bit Format hier sind es 16 BIT), 256 (Farbanzahl)
 (die jeweils in Klammern beigefügten Zahlen sind für NFS 4 unbedeutend - sie sollten aber dennoch wie in den Beispielen mit eingefügt werden - die genaue Funktion/Auswirkung ist mir auch noch nicht ganz klar :-)

Hier ein kleines Beispiel für Schilder die auch bei spiegelverkehrter Strecke funktionieren sollen.

Hierzu müssen zwei Versionen der Grafik (des Schildes - Beispielgrafik siehe unten) im QFS File hinterlegt werden.

```
0102 0099.BMP
BMP 7B +2 64 64 {0 0 0 0}
PAL 2D 256 1 {256 0 0 0}
TXT 6F 16 0
<nonmirrored>%00%00%00
```

Die erste Version besteht aus der normalen Grafik (siehe rechts) Der Unterschied zu den bisherigen Einträge besteht aus der unteren Zeile des Abschnittes. Option <nonmirrod>.... bedeutet, dass die Grafik ohne einer Spiegelung angezeigt wird (also bei normaler Streckenfahrt).

```
0101 0100.BMP
BMP 7B +2 64 64 {64 0 0 0}
PAL 2D 256 1 {256 0 0 0}
TXT 6F 16 0
<mirrored>%00%00%00%00%00%00
```

Im zweiten Abschnitt handelt es sich um die gespiegelte Version der Grafik. D.h. die Grafik aus dem ersten Abschnitt muss hier in einer gespiegelten Version eingefügt werden (siehe auch den Hinweis <mirrored>... in der unteren Zeile). Diese Grafik wird dann in einem „gespiegelten“ Track (sofern dies im Game eingestellt wurde) angezeigt.



0099

0100

BIT Farbtiefe des Bildes:**Beispiel:**

0011 0009.BMP

BMP 7B +1 64 64 {0 0 0 0}

PAL 2D 256 1 {256 0 0 0}

Das BMP BIT Format in grüner Farbe markiert (Zeile 2):**6D** 16-bit 4:4:4:4 (mit Alpha Kanal)**78** 16-bit 0:5:6:5**7B** 8-bit (mit Palette)**7D** 32-bit 8:8:8:8 (mit Alpha Kanal)**7E** 16-bit 1:5:5:5 (mit Alpha Kanal)**7F** 24-bit 0:8:8:8**Das PAL BIT Format in gelber Farbe markiert (Zeile 3 links):****22** 24-bit DOS**24** 24-bit**29** 16-bit NFS5**2A** 32-bit**2D** 16-bit**Farbanzahl in der Palette - rosa markiert (Zeile 3 rechts):**

im obigen Beispiel wären es 256 Farben

*Demnach besteht das obige Beispiel aus einer 8-BIT (256 Farben aus einer 16 BIT Palette) Grafik mit einer Auflösung von 64*64.*

Anhand dem schon vorher aufgeführten Beispiel der gespiegelten Bilder möchte ich noch einmal genauer auf die möglichen BMP Anhänge eingehen.

Hier noch einmal das betr. Beispiel:

0102 0099.BMP

BMP 7B +2 64 64 {0 0 0 0}

PAL 2D 256 1 {256 0 0 0}

TXT 6F 16 0

<nonmirrored>%00%00%00

0101 0100.BMP

BMP 7B +2 64 64 {64 0 0 0}

PAL 2D 256 1 {256 0 0 0}

TXT 6F 16 0

<mirrored>%00%00%00%00%00%00

In der Zeile 2 (grün markiert) könnt Ihr die Zahl + 2 sehen. Dies bedeutet, dass nach der BMP Beschreibung noch zwei weitere Zusätze erscheinen (Zeile 3 und 4/5).

Zeile 3 also die Option mit der Bezeichnung PAL usw. habe ich schon zuvor beschrieben. In der Zeile 4 kommt jetzt aber noch eine weitere wichtige Option hinzu. Mit der Option TXT usw. könnt Ihr noch bestimmte Eigenschaften der Grafik zuordnen z.B. hier <nonmirrored> = ungespiegelt oder <mirrored> gespiegelt (näheres zu dieser Funktion habe ich schon zuvor beschrieben).

Aufbau der Option TXT:

TXT 6F <Anzahl der folgenden Beschreibungen> 0

Danach folgt in der nächsten Zeile die Beschreibung.

z.B.

<mirrored>%00%00%00%00%00%00

d.h. diese Grafik wird nur bei gespiegelter Streckeneinstellung angezeigt.

Kurze Zusammenfassung des Erlernten:**Bisheriges Bild/Format verändern:**

- QFS File mit FSHTool entpacken.
- die betreffende Grafik verändern bzw. vergrößern z.B. aus einer 64*64 großen Grafik wird eine im 128*128 Format.
- die Datei **index.fsh** entsprechend ergänzen bzw. anpassen.
- Doppelklick auf **index.fsh** (das neue QFS File wird automatisch erstellt).
- Mit z.B. NFSWizard die veränderten Grafiken aus dem QFS-File exportieren (als TGA Datei) und den Alpha-Kanal in z.B. Paintshop Pro (ab Version 5) bearbeiten und dann wieder importieren. Abspeichern - FERTIG!

Neue zusätzlichen Grafiken einfügen:

- QFS File mit FSHTool entpacken.
- die betreffenden zusätzliche Grafiken in das Verzeichnis (in dem die zuvor entpackten Grafiken enthalten sind) kopieren (bitte die genaue laufende Namensbezeichnung beachten z.B. 0249.BMP, 0250.BMP usw.).
- die Datei **index.fsh** entsprechend ergänzen bzw. anpassen.
- Doppelklick auf **index.fsh** (das neue QFS File wird automatisch erstellt).
- Mit z.B. NFSWizard die veränderten oder neu eingefügten Grafiken aus dem QFS-File exportieren (als TGA Datei) und den Alpha-Kanal in z.B. Paintshop Pro (ab Version 5) bearbeiten und dann wieder importieren. Abspeichern - FERTIG!

Hoffentlich sind hiermit einige Unklarheiten im Zusammenhang mit dem FSH Tool aus der Welt geräumt und Ich hoffe, daß Ihr jetzt auch das QFS File entsprechend eurer Wünsche erweitern oder verändern könnt.

Eine ausführlichere Anleitung, dafür aber leider in englischer Sprache, ist in der ZIP Datei des FSHTools enthalten (wer mehr wissen möchte).

Anhang 4:**kurze Beschreibung von Beni C. (Email: Beni_C@web.de) zum Thema Punkt-zu-Punkt Strecken für NFS4.**

Anders als die Strecken von NFS1 oder NFS5 sind die Strecken von NFS4 alle Rundstrecken, d.h. man kommt wieder dorthin zurück wo man gestartet ist und fährt mehrere Runden. Die Punkt-zu-Punkt (Point-to-Point) Strecken hingegen haben einen Start und einen Endpunkt und man fährt die Strecke nur einmal.

Ich hab mir vor einiger Zeit schon Gedanken dazu gemacht, aber erst kürzlich hab ich es wirklich probiert.

Hier die Beschreibung:

- Eine Strecke ist unterteilt in Blocks. Jeder Block besteht der Länge nach aus 8 Polygonen. Die Landstrasse hat z.B. 135 Blocks. Die untenstehenden Zahlenkolonnen, z.B. (0,1,2,3,4...99) bezeichnen die Nummer des jeweiligen Blocks (steht in der unteren Leiste des Trackeditors) und deren Anordnung.

- Als Beispiel nehmen wir hier eine Strecke mit 100 Blocks (von 0 bis 99). Wir fahren ein 2 Runden Rennen und vorwärts.

- NFS sieht ein Runde als beendet an, wenn man die 100 Blocks in der richtigen Reihenfolge abgefahren hat (0-99) und wieder auf den 0. Block kommt. Wenn man das dann ein 2. Mal macht, sieht NFS das Rennen als beendet an. Das ist der normale Ablauf.

- Man muß also NFS vorspielen das man zweimal die 100 Blocks abfährt, ohne das man tatsächlich die 2 Runden gefahren ist.

- Hierzu muß man am Ende der Strecke (nachdem die 1. 100 Blocks abgefahren wurden) jeweils noch mal eine Reihe (keinen ganzen Block) von Polygonen des (0,1,2,3,4,5,6...99). Blocks einfügen. NFS "denkt" dann, das die 2 Runden beendet wären und das Rennen ist beendet.

- Um das zu vereinfachen habe ich 2 wichtige Dinge herausgefunden:

- NFS sieht eine Runde schon nach ca. 3/4 der Strecke als beendet an, die 2. 100 Polygone können danach schon starten. D.h. man kann die Strecke auch (0,1,2,3,4...75) bauen und dann die Zweite Staffel Blocks anfangen.

- Man muß nicht aus jedem Block eine Reihe Polygone einfügen (0,1,2,3,4...99), es reicht jeder 10. oder sogar jeder 15. Block (0,10,20,30...99).

- Wichtig ist jedoch das man nach dem 99. Block noch mal eine Reihe des 0. Blocks einfügt, das ist dann das Ziel.

*- Die Strecke sieht von den Blöcken her also wie folgt aus:
0,1,2,3,4,5...99,0,10,20,30...99,0.*

- Wenn man die Strecke so bauen will, daß man auf ihr auch 8 Runden Rennen fahren kann, muß man nach den ersten 100 original Streckenblocks noch 7 mal die Polygone aus den Blocks einfügen (wie oben, nur statt 1 mal, 7 mal).

Probleme:

- Beim fahren auf den neuen Polygonen wird sowohl das Auto wie auch die Strecke unsichtbar.

- Die Strecken sind wie oben beschrieben nicht rückwärts befahrbar.

- Bei mehreren Mitfahrern kommt es zu Problemen, da der Startpunkt mancher Spieler mehrere Blocks vor der Ziellinie liegt, so das diese dann nicht in der Lage wären am Rennen teilzunehmen, weil sie sofort in die 2. Staffel Blocks käme. Deshalb ist es anzuraten die Strecke nach dem Muster (0,1,2,3,4...75,0,10,20,30..99,0) usw. bastelt.

- Man sollte die Strecke (0,1,2,3,4...75,0,10,20,30..99,0) nach dem 75. Block vom 76. Block trennen, da sonst, wenn man nach Überqueren der Blocks wieder auf den 76. trifft das Rennen eventuell nicht als beendet angesehen wird.

Da ich leider nicht allzu viel Zeit hatte es zu testen kann ich euch nur sagen das es funktioniert, die Feinabstimmung und die Fehlervermeidung müßt ihr allerdings selbst herausfinden. Ich hoffe das Prinzip ist einigermaßenklar geworden, es ist zwar nicht so einfach, aber leichter als es sich anhört. Falls ihr Fragen habt, könnt ihr mir gerne mailen, falls ihr etwas rausgefunden habt, teilt mir das doch bitte auch mit!

Viel Spaß

Beni C.

Anmerkung von Fangio:

Ich hatte leider auch noch keine Möglichkeit bzw. Zeit die Beschreibung von Beni C. in die Praxis umzusetzen.

Demnach kann Ich euch nur sagen „Probieren geht vor Studieren“. Also probiert es einfach einmal aus und stellt dann den ersten Point to Point Track für NFS 4 online (bisher, zumindest zum Zeitpunkt des Erstellungsdatums der Anleitung, habe ich noch keine P. to P. Strecke gesehen).

Anhang 5:

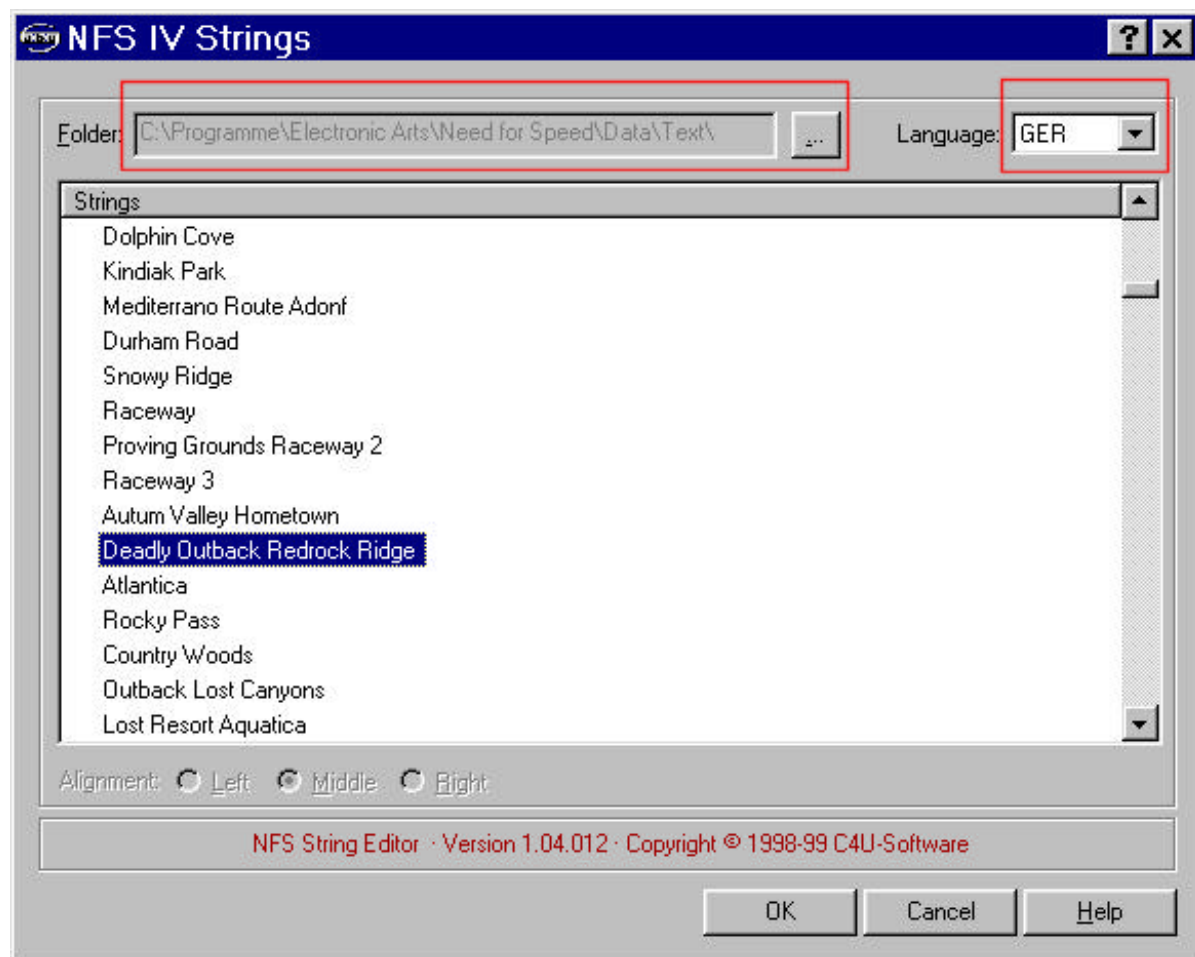
NFS String Editor

Mit diesem Programm kann man alle Texte in NFS 4 editieren - also auch bei Bedarf die angezeigten Tracknamen ☺.

D. h. Ihr könnt als kleine Zugabe zu euerem Track auch eine berichtigte Textdatei beifügen in der euerer Trackname schon eingefügt wurde.

Die Textdateien von NFS 4 sind im Verzeichnis **C:\Programme\Electronic Arts\Need for Speed\Data\Text** hinterlegt. Dort gibt es für jede Sprachversion eine eigene Datei (z.B. für Deutschland ist es die Datei **text.GER**).

Im NFS String Editor müßt Ihr nur noch nach dem Programmstart den richtigen Pfad (links oben rot markiert) und die jeweilige Sprachversion (rechts oben rot markiert) einstellen (siehe Grafik unten).



Editieren kann man die jeweiligen Texte im mittleren Fenster (siehe blaue Markierung) indem man den jeweiligen Text markiert und dann über das Kontextmenü (rechte Maustaste drücken) den Befehl **EDIT** aufruft. Die betr. Zeile kann man jetzt mit einem eigenen Text versehen (als Beispiel habe ich meinen Track Deadly Outback genommen). Nach der Eingabe müßt Ihr nur noch einen anderen Texteintrag anklicken (danach erscheint an der zuvor veränderten Stelle

ein Ausrufezeichen) und das Programm beenden bzw. auf OK drücken. Die jetzt veränderte Text Datei könnt Ihr eurem Track (bzw. ZIP File) beifügen so dass jetzt jeder User in seinem Spiel den Tracknamen im Auswahlmenü sehen kann (siehe Grafik unten).



Anhang 6:

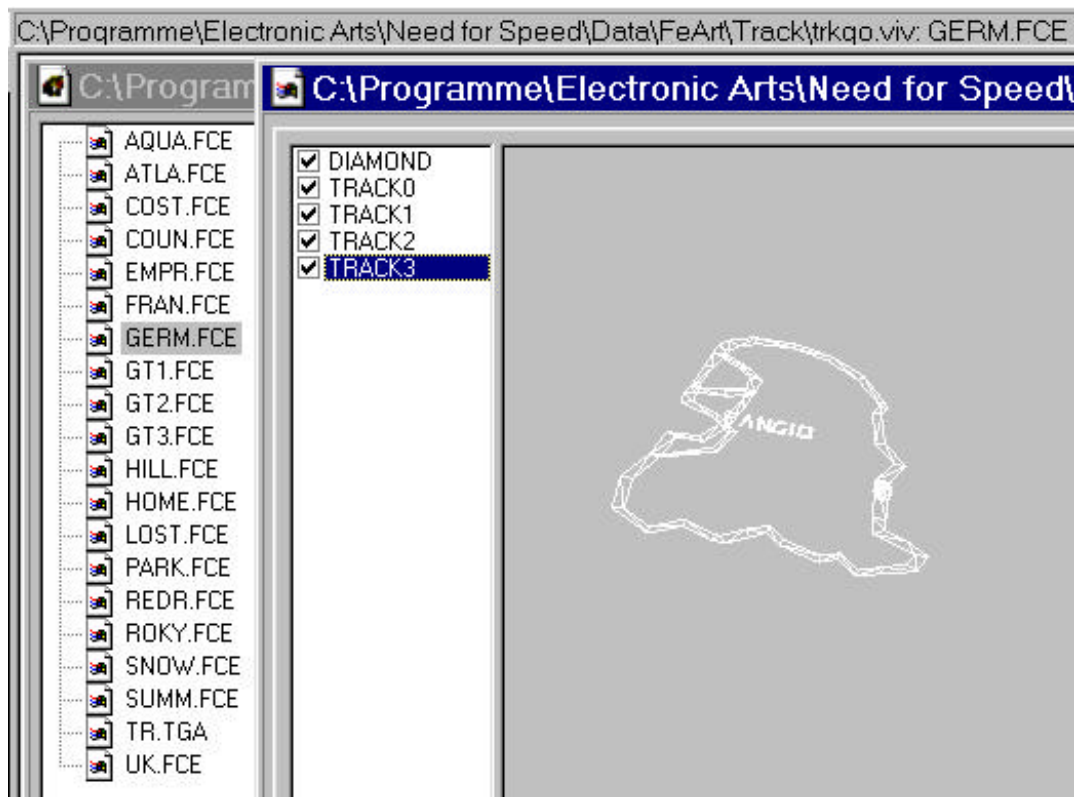
Habt Ihr einmal überlegt wie schön es wäre, wenn man die Streckenführung eueres Tracks schon vor der Fahrt anschauen könnte ?

Tja kein Problem - das geht ☺.

Im Verzeichnis **C:\Programme\Electronic Arts\Need for Speed\Data\FeArt\Track** gib es die Datei **trkgo.viv** in der mittels einer FCE Datei die Streckenführung in der **Vidwall Ansicht** (also dort wo man die Wetterverhältnisse, die Fahrtrichtung usw. einstellt) angezeigt wird. Zum Verändern müßt Ihr nur diese Datei mit dem NFSWizard öffnen, die jeweilige FCE Datei exportieren und dann mit CarCad bearbeiten.

Nach dem Bearbeiten mittels CarCad (geht genauso wie bei den Cars nur dass in diesem Fall die Strecke aus fünf Teilbereichen besteht **siehe Grafik unten - Ansicht im NFSWizard**) könnt Ihr die FCE Datei wieder mit dem NFSWizard in die Datei **trkgo.viv** einfügen und ebenfalls eurem Track beifügen. Wer sich dieses Verfahren einmal genauer anschauen möchte der kann sich auf meiner Seite www.fangio.de/tracks.htm ein kleines Beispiel für meine Tracks **Kindiak Track special** und **Landstrasse Track special** downloaden (siehe meine Bemerkungen oben auf der jeweiligen Internetseite).

Hier könnt Ihr die Streckenführung für meinen Landstrasse Track (im NFSWizard) sehen. Links stehen die jeweiligen Kurzbezeichnungen für die Tracks und recht sieht man die fünf Teilbereiche der FCE Datei.



So sieht es dann im Spiel aus ☺:

