

MSTS Tutorial: Korrektur KFZ Abstände V2.0

Autor: don.ultimax@gmx.at

Auf vielen Trasi-Strecken ist animierter Straßenverkehr zu sehen. Leider kommen diese Fahrzeuge vor Bahnübergängen oft ineinander steckend zu stehen, und fahren anschließend in diesem Zustand auch wieder weiter. Oft stecken die Fahrzeuge auch mit ihren Rädern tief in der Fahrbahn. Beides ist nicht wirklich schön anzusehen. Siehe untenstehende Grafik.



Die Ursache dieser Probleme basiert auf zwei Fehlern. Einerseits in einem falschen numerischen Zahlenwert im „CarSpawnerItem“ Eintrag des entsprechenden Fahrzeuges in der „Carspawndata“ Datei und / oder in Konstruktionsungenauigkeiten des entsprechenden Modells. Dies ist besonders häufig bei Modellen zu bemerken die mit dem TSM/TSD gebastelt wurden.

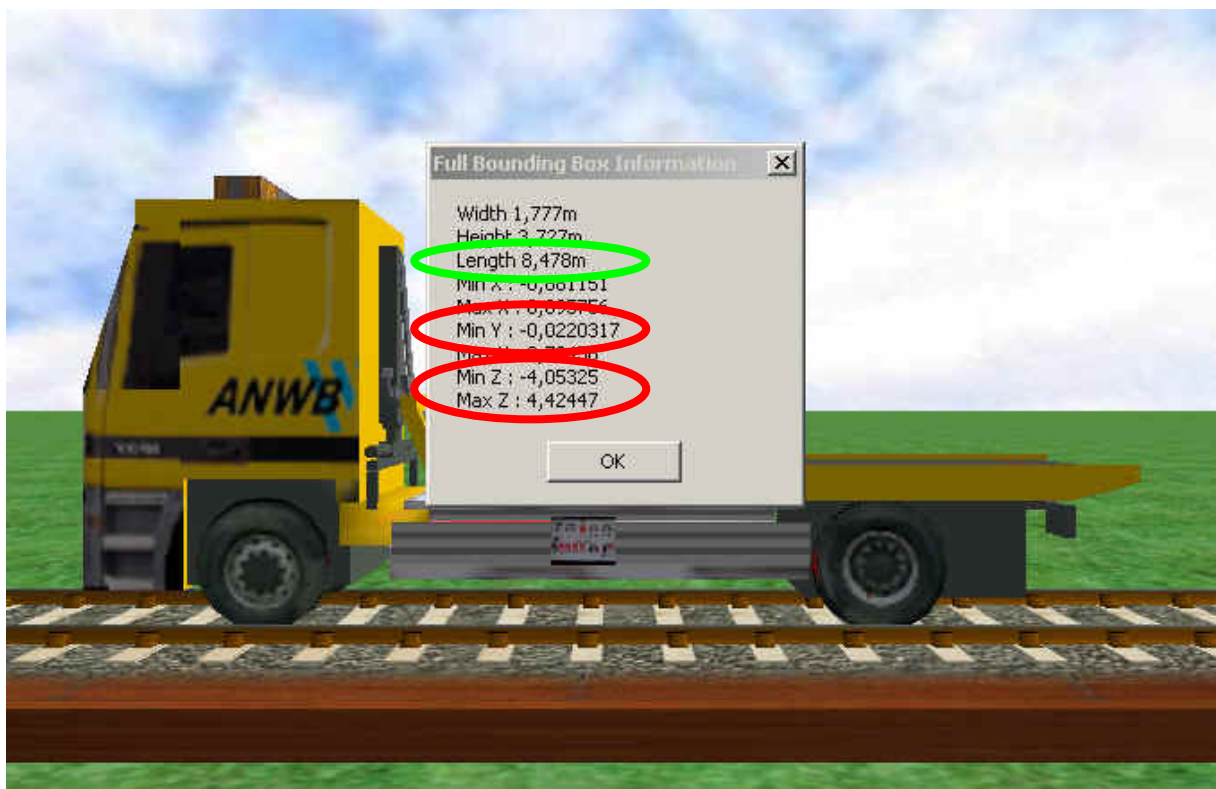
Bereits in meiner V1.0 dieses Tutorial habe ich ein damals doch eher aufwendiges Verfahren beschrieben um diese Missstände abstellen zu können. Inzwischen hat Paul Gausden sein geniales Tool „Shape Viewer“, auch unter „SView“ bekannt, zur raschen Betrachtung von Trasi Modellen veröffentlicht. Auf Grund meiner Anregung hat Paul in dieses Tool auch noch die Möglichkeit eingebaut, sich die für unsere Problematik relevanten Parameter anzeigen lassen zu können. Dafür meinen ganz besonderen Dank an Paul.

Womit es nun wesentlich einfacher möglich ist, die Ursachen dieses Problemkreises genau zu analysieren. Daher habe ich diese neue, nun auch illustrierte, Version meines Tutorial „Korrektur KFZ Abstände V2.0“ geschaffen.

Mit dem ebenfalls von Paul Gausden bereits vor längerem veröffentlichten fantastischen Tool „Shape File Manager“, auch unter „SFM“ bekannt, lassen sich die relevanten Parameter leicht und rasch anpassen. Der Ordnung halber weise ich an dieser Stelle jedoch darauf hin, dass leider viele der mit diesen Mängeln behafteten Fahrzeuge durch Copyrights derart geschützt sind, dass eine entsprechende Manipulation eigentlich nicht zulässig wäre.

Um ein perfektes Erscheinungsbild des Straßenverkehrs zu erzielen, müssen **ALLE** in der „Carspawn.dat“ Datei definierten Fahrzeug überprüft bzw. angepasst werden.

Wir betrachten uns ein beliebiges Fahrzeug mit dem „Shape Viewer“ (z.B. V1.3.129) und lassen uns mittels „STRG“ + „C“ oder im Menü unter „Shape laden“ -> „Show Bounding Info“ die „Full Bounding Box Information“ anzeigen. Hierbei sind die Werte „Length“, „Min Y“, „Min Z“ und „Max Z“ für uns von Bedeutung. Wir notieren uns diese Werte unseres Probanden, wir werden sie später noch benötigen.



Vorab: Ist der entsprechende numerische Wert im „CarSpawnerItem“ um etwa 1,5 - 2,5m größer als der unter „Length“ im „Sview“ ausgewiesene Wert **UND** sind die beiden ausgewiesenen Werte für „Min Z“ und „Max Z“ auf etwa 0,2m identisch **UND** beträgt der Wert von „Min Y“ zwischen 0,10m und 0,12m, können wir dieses Fahrzeug belassen, wie es ist. Eine saubere Arbeit eines Entwicklers der wusste was er tat !

Stimmen jedoch ein oder mehrere dieser Parameter nicht mit den Anforderungen überein, so sollten wir handeln. Ein optisch ansprechend wirkender Straßenverkehr wird der Lohn unserer Bemühungen sein!

Wichtige Hinweise

Die Original-Straßenstücke des Trainsimulators liegen leider auf verschiedenen Niveaus. So liegen die „jp2rd...“ und die „us1rd...“ Straßenstücke 0,2m über Null, die „road...“ und „dirtroad...“ Straßenstücke jedoch 0,1m über Null. Je nach dem welche Original-Straßentypen in einer Strecke verbaut sind, kann der Straßenverkehr mehr oder weniger tief in der Fahrbahn stecken.

Das exzellente Freeware Straßen-Erweiterungspaket „New Roads“ beinhaltet nicht nur ein Vielzahl von neuen Straßenstücken, sondern auch die korrekte Korrektur der „jp2rd...“ und „us1rd...“ Straßenstücke auf das Standardniveau von 0,1m.

Die Anwendung des „Height Mis-Match Fix for Default Roads“ (rdhgfix.zip) kann zu Problemen führen, denn damit werden die „jp2rd...“ und „us1rd...“ Straßenstücke nicht wie angegeben auf das Standard-Niveau von 0,1m korrigiert, sondern auf das tatsächliche Niveau von 0,0m. Womit nicht nur wieder eine Diskrepanz zu den Standard „road...“ und „dirtroad...“ Straßenstücken entsteht, sondern auch zu all denen von „New Roads“.

1.) Korrektur der Fahrzeuglängssymmetrie

Dies geschieht durch Manipulation der Shape Datei, also der *.s Datei des Modells.

Dazu ist es nötig, falls erforderlich, diese zu dekomprimieren. Ob das Shape eines Modells komprimiert ist oder nicht, erkennt man, wenn man die entsprechende *.s Datei mit dem Editor öffnet. Ergibt dies „wirre“ Zeichen ist die Datei komprimiert. Zum dekomprimieren gibt es mehrere gute Tools. Ich empfehle ausdrücklich dafür **NICHT** die integrierte Funktionalität des doch schon älteren „ShapeFileManager“ („SFM“) zu verwenden. Geeignete Tools sind z.B. „Compress / Uncompress“ von Paul Gausden oder „Route_riter“ von Mike Simpson.

Betrachten wir nun noch einmal unseren Probanden, das Abschleppfahrzeug, so stellen wir folgende Werte fest:

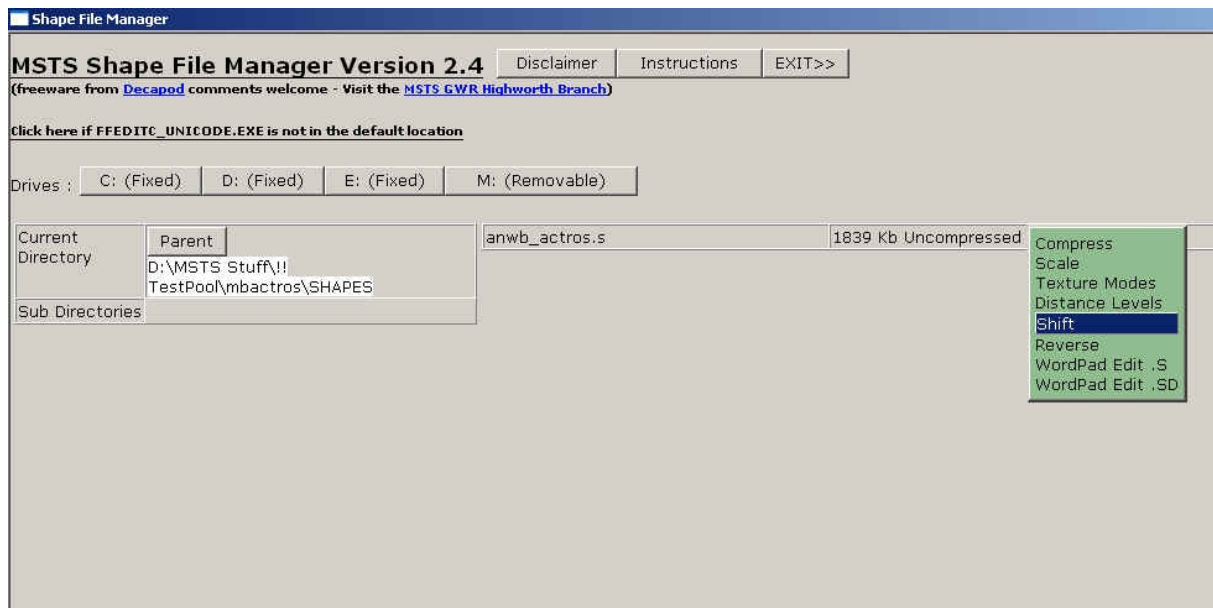
Min Z = -4,05325

Max Z = 4,42447

Daraus resultiert ein numerischer Unterschied von 0,37122 zwischen diesen beiden Werten. Ein **VIERTTEL** davon, also 0,092805, das ist die Größe die wir mit dem SFM (ShapeFileManager) korrigieren müssen.

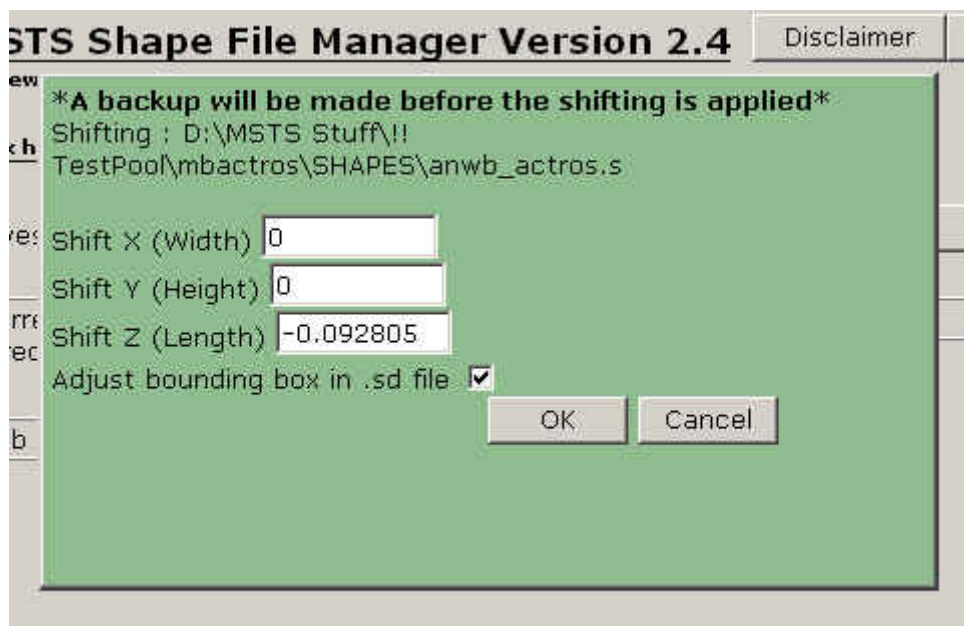
Wir starten nun den SFM und navigieren zu der entsprechenden Shape-Datei (*.s). Danach fahren wir mit der Maus über „Options“, nun öffnen sich meh-

rere Möglichkeiten. Wir fahren mit der Maus weiter auf „Shift“ und erhalten nun die Möglichkeit den konstruktiven Bezugspunkt des Fahrzeuges (Pivotpunkt) zu korrigieren.



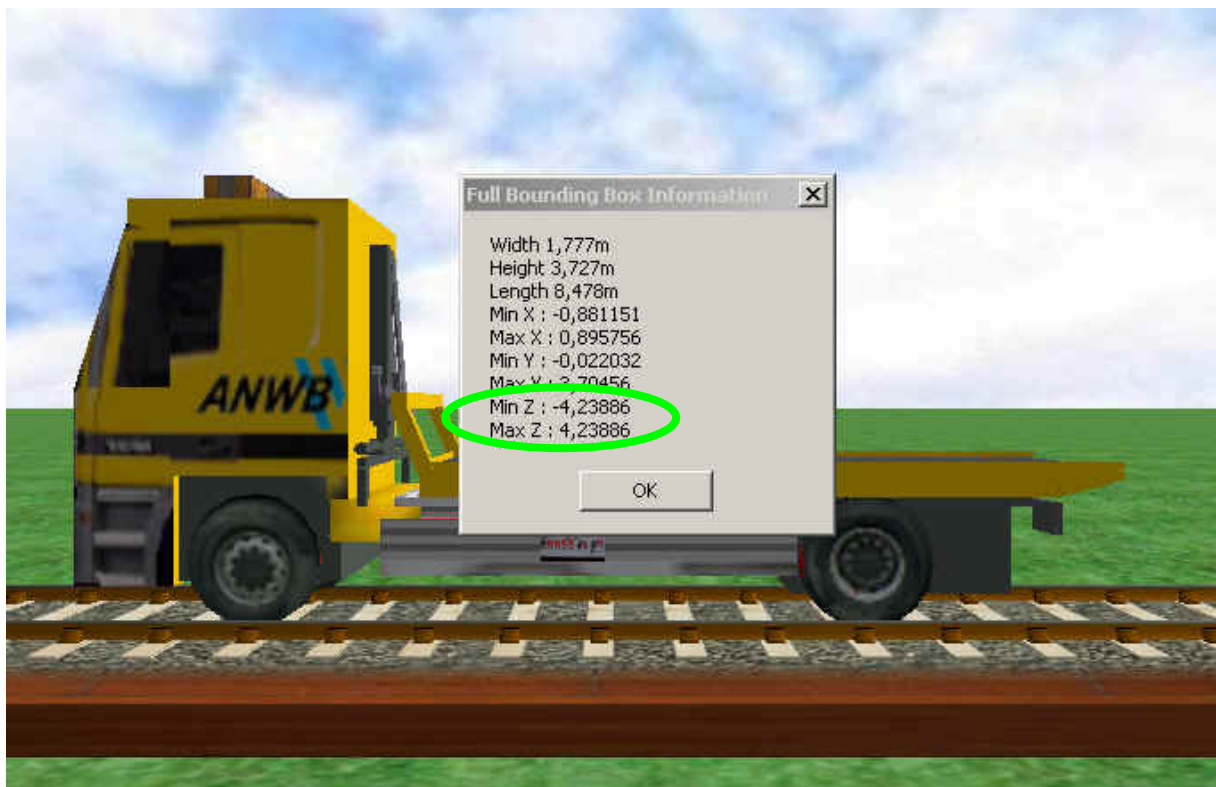
Nachdem „Min Z“ numerisch **KLEINER** als „Max Z“ ist, korrigieren wir unseren Pivotpunkt in die **MINUS** Richtung.

Wir geben im SFM in der Zeile „Shift Z (Length)“ den ermittelten Wert von 0,092805 als MINUS Wert ein. **Wobei das Komma als Punkt anzuführen ist.** Also als -0.092805. Sonst werden die Nachkommastellen abgeschnitten!



Ist bei einem beliebigen anderen Probanden „Min Z“ numerisch **GROESSER** als „Max Z“ korrigieren wir im SFM natürlich in **PLUS** Richtung!

Betrachten wir nach dieser Manipulation unseren Probanden zur Kontrolle wieder im „SView“, so sollten „Min Z“ und „Max Z“ exakt übereinstimmen.



Bingo ! Erste wichtige Hürde erfolgreich bewältigt.

2.) Korrektur der Lage des Fahrzeuges zur Fahrbahnhöhe

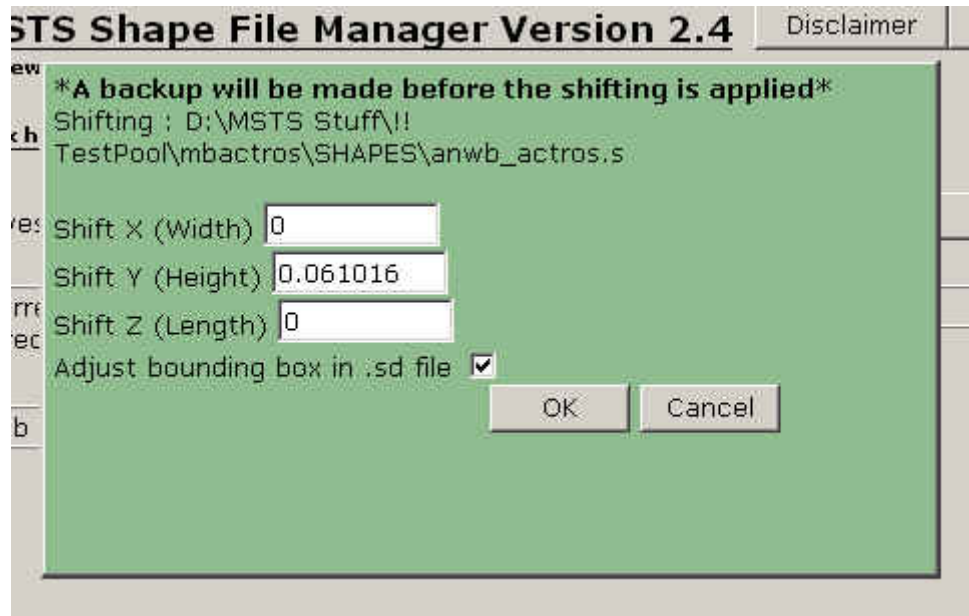
Daß die Straßenfahrzeuge des Trasis ihre Fahrbahnen in Streifen schneiden ist nicht primär die Schuld der Fahrzeugbauer. Vielmehr resultiert dieser Effekt daher, dass die Straßen einen gewissen Abstand zum darunter liegenden Gelände haben und die Autos dem Gelände folgen. Was man jedoch als Modellbauer wissen und bei seiner Arbeit auch entsprechend berücksichtigen sollte. Diesen Fehler gilt es nun im nächsten Schritt zu korrigieren.

Indem wir den Pivotpunkt der Fahrzeuge entsprechend nach unten, oder anders betrachtet, das Fahrzeug entsprechend nach oben verschieben.

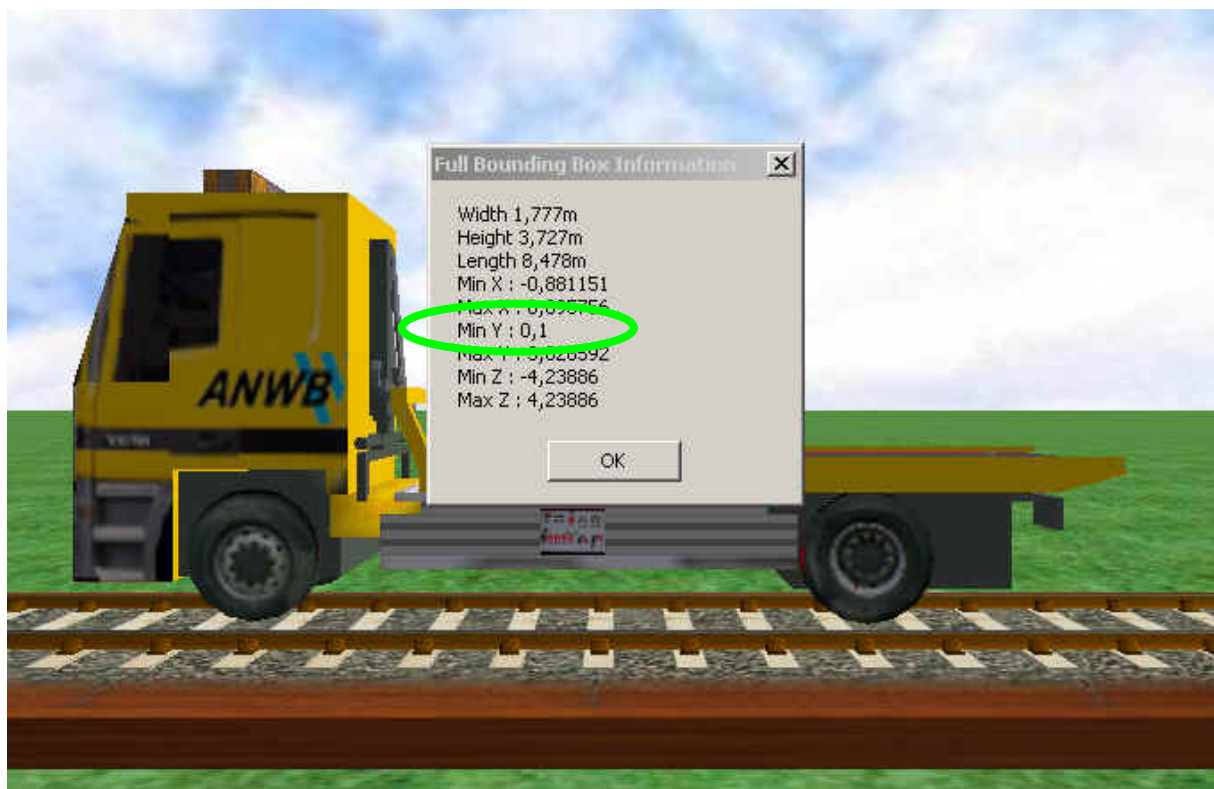
Dazu kontrollieren wir, wieder im „SView“, den Parameter „Min Y“. Beträgt dieser Wert +0,10m stimmt das wahrscheinlich exakt. „Wahrscheinlich“ nur deshalb, weil in diesem Wert auch wieder Konstruktionsungenauigkeiten des Fahrzeuges verankert liegen können, die unter Umständen in einem weiteren Schritt korrigiert werden müssten. Dies ist jedoch nur äußerst selten nötig.

In unserem konkreten Beispiel beträgt der aktuelle Parameter für „Min Y“ noch -0,022032. Das Fahrzeug steckt sogar im Gelände. Siehe ersten Screenshot dieses Tutorial.

Um nun den Abstand zum darunter liegenden Gelände auf +0,10m zu korrigieren, verschieben wir im „SFM“ den Pivotpunkt um den **HALBEN** Unterschied der beiden Werte nach unten, oder wieder anders betrachtet, eben das Fahrzeug nach oben. Also im SFM um **PLUS** 0,061016 (der Hälfte von 0,122032), dazu ist bei „Shift“ der Wert 0.061016 in der Zeile „Shift Y (Height)“ einzutragen.



Wir werden nun mit dem angestrebten Abstand von 0,1m des Fahrzeuges zum Untergrund belohnt!



Nochmals Bingo ! Zweite wichtige Hürde erfolgreich bewältigt.

Bei Ausgangswerten des Parameters von „Min Y“, die sich im positiven Zahlenbereich befinden, gestaltet sich die ganze Rechnerei etwas einfacher.

Liegt der Wert den uns „SView“ liefert z.B. bei +0,03, und ist damit **KLEINER** als unser angestrebter Wert von +0,10, korrigieren wir einfach um den **HALBEN** Unterschied in **PLUS** Richtung. Also um 0.035, in der Zeile „Shift Y (Height)“

Liegt der Wert den uns „SView“ liefert z.B. bei +0,12, und ist damit **GROESSER** als unser angestrebter Wert von +0,10, korrigieren wir einfach um den **HALBEN** Unterschied in **MINUS** Richtung. Also um -0.01, in der Zeile „Shift Y (Height)“

Das alles liest sich komplizierter als es in Wirklichkeit ist. Spätestens nach dem dritten Fahrzeug hat man den Dreh heraus und die Anpassung eines Fahrzeuges ist in wenigen Minuten erledigt !!!

3.) Korrektur des Parameters im „CarSpawnerItem“

Der Wert des Parameters für das entsprechende Fahrzeug im „CarSpawnerItem“ der „Carspawn.dat“ Datei bezieht sich auf die jeweilige Fahrzeuglänge unter Berücksichtigung der nötigen Sicherheitsabstände zu anderen Fahrzeugen. Also wie weit sich unser Fahrzeug dem zuvor fahrenden nähert, bzw. wie weit sich das nachfolgende Fahrzeug unserem nähern darf.

Für unser konkretes Beispiel empfiehlt der Entwickler den Eintrag:

```
CarSpawnerItem( „anwb_actros.s“ 8 )
```

Betrachten wir im „SView“ den Wert von „Length“, sehen wir dass unser Fahrzeug 8,478m lang ist. Das bedeutet, kommen zwei dieser Fahrzeuge vor einem Bahnübergang zu stehen, werden diese 0,478m ineinander dringen. Siehe erster Screenshot dieses Tutorial. Leider fahren dann, so zum Stehen gekommenen Fahrzeuge, so ineinander gedrungen auch weiter entlang des gesamten Straßenverlaufes.

Um nun zu einem brauchbaren Ergebnis zu gelangen vergrößern wir einfach den Wert im „CarSpawnerItem“ um etwa 1,5 bis 2,5m. Je nach dem wie groß unser Fahrzeug ist. Kleinere Fahrzeuge (PKW's) um etwa 1,5m, größere Fahrzeuge (Laster mit Anhänger) um etwa 2,5m. In unserem Beispiel sind 1,5m ein guter Wert. Also:

```
CarSpawnerItem( „anwb_actros.s“ 10 )
```

(Übrigens, der Wert im „CarSpawnerItem“ muss immer ganzzahlig sein)

Für unser Beispiel heißt das also, dass unser Fahrzeug mit einer Länge von etwa 8,5m jeweils nach vorne und nach hinten einen Abstand von 0,75m erzwingt.

Wenn alle anderen Fahrzeuge in unserer „Carspawn.dat“ Datei mit einem ebensolchen entsprechenden Übermaß im jeweiligen „CarSpawnerItem“ definiert sind, können sich die Fahrzeuge nie mehr als auf etwa 1,5m bis 2,5m einander nähern.

Womit im Wesentlichen eigentlich unsere Fahrzeugabstimmung beendet ist.

4.) Testen

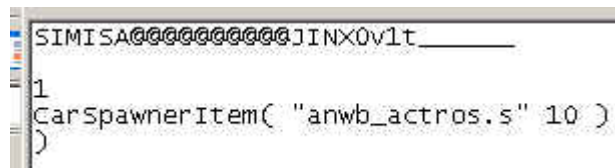
Zu guter letzt sollten wir aber auf jeden Fall noch unser Fahrzeug in der Simulation betrachten. Es könnten sich Fehler eingeschlichen haben.

Eine deutliche Erleichterung zum Testen stellen Tools dar, mit denen man einen erweiterten Beobachtungsspielraum hat. Z.B. CamHackII.

Zur Begutachtung unserer Probanden erstellen wir am besten einmalig im Aufgabeneditor eine einfache Activity, oder auch nur einen „Path“, bei dessen Start eine beliebige Lok genau vor einem Bahnübergang zu stehen kommt, um damit den anrollenden Straßenverkehr zu blockieren.

Danach sichern wir unsere vollständige „Carspawn.dat“ Datei weg und erstellen anstatt dieser daraus eine neue Minimalversion. In diese neue „Carspawn.dat“ Datei nehmen wir nur unseren Probanden auf.

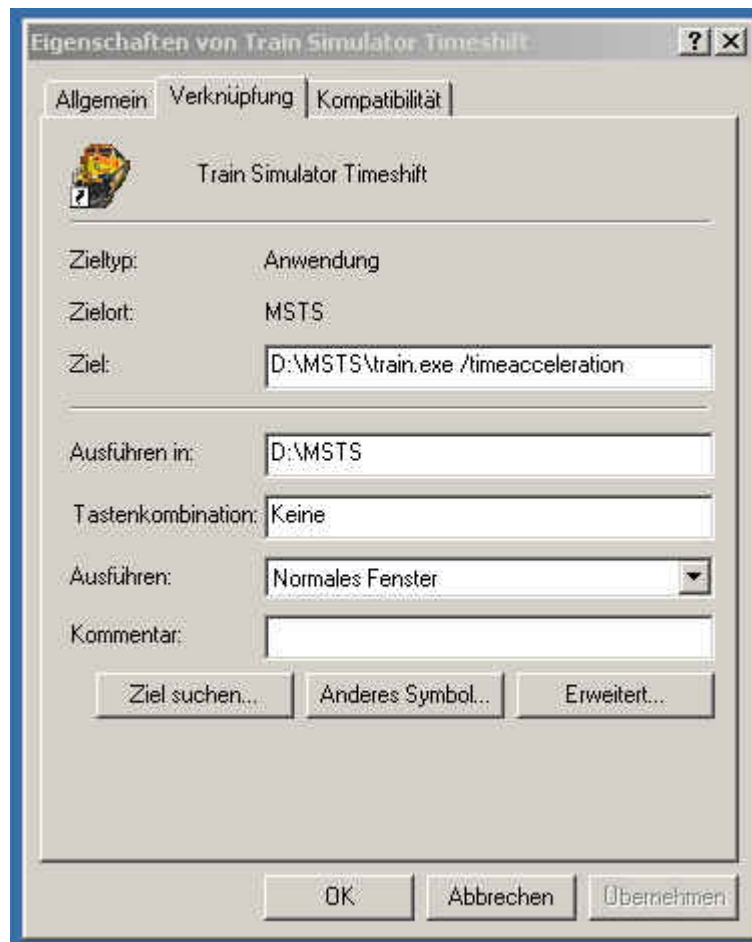
Das könnte in etwa so aussehen:



```
SIMISA#####JINXOVlt_____  
1  
CarSpawnerItem( "anwb_actros.s" 10 )  
)
```

Nun können wir die erstellte Activity starten und warten bis sich die Fahrzeuge dem Bahnübergang nähern. Da dies ganz schön lange dauern kann bis endlich Fahrzeuge vor dem Bahnübergang zum Stehen kommen, empfehle ich zu diesem Zweck den Start des Trainsimulators mit der Option „timeacceleration“.

Dazu gilt es eine neue Verknüpfung zum Trasi, mittels kopieren und einfügen, am Desktop zu erstellen und dann im Kontextmenü unter „Eigenschaften“ als neues Ziel z.B. D:\MSTS\train.exe /timeacceleration anzugeben, sofern D:\MSTS\ der tatsächliche Ort der Trasi-Installation ist.



Danach kann man, wenn man den Trainsimulator durch Doppelklick auf diese Verknüpfung startet, mit den Tasten STRG + T den Ablauf der Simulation entsprechend mehrfach beschleunigen und mit den Taste STRG + SHIFT + T den Ablauf auch wieder entsprechend verzögern. Und siehe da, plötzlich rauschen die Autos nur so daher.

Mit der Taste „4“ wechseln wir in die Außenansicht und können mittels schwenken und zoomen sehr gut bewerten, ob das Ergebnis unserer Abstimmungen auch entsprechend ausgefallen ist.

5.) Aufräumen und weitere Optimierungen

Nachdem wir unsere *.s Datei noch dekomprimiert haben, sollten wir die Gelegenheit nutzen und auch noch ein paar andere Optimierungen vornehmen.

a.) Als erstes suchen wir mit einem Editor in unserer *.s Datei nach dem Text „named_filter_mode“. Dieser Eintrag befindet sich ziemlich im oberen Bereich der Datei. Im Regelfall werden wir dort den Text „named_filter_mode (**MipLinear**)“ vorfinden.

```

        named_shader ( TexDiff )
    }
    texture_filter_names ( 1
        named_filter_mode ( MipLinear )
    )
    points ( 1586

```

Diesen ändern wir auf „named_filter_mode (**Linear**)“

```

        named_shader ( TexDiff )
    }
    texture_filter_names ( 1
        named_filter_mode ( Linear )
    )
    points ( 1586

```

Damit werden lästige Moire-Effekte von zu detaillierten Texturen unterbunden. Das kostet zwar ein wenig an Rechnerressourcen, aber das machen wir bereits durch den nächsten Optimierungsschritt mehr als wett!

b.) Nun suchen wir in der *.s Datei mit dem Editor nach dem Text „**distance_levels (** „. Dieser befindet sich etwa im oberen Drittel der Datei. Finden wir in dieser Zeile den Parameterwert „1“ bedeutet dies, dass das Fahrzeug nur einen LOD (Level of Distance) besitzt. Ist dies der Fall, sollten wir auf jeden Fall diese Sichtbarkeitsgrenze anpassen. Dies geschieht durch Manipulation des Eintrages des Wertes in „dlevel_selection“ ein paar Zeilen darunter.

Finden wir:

dlevel_selection (**2000**)

```

distance_levels_header ( 0 )
distance_levels ( 1
    distance_level (
        distance_level_header (
            dlevel_selection ( 2000 )
            hierarchy ( 1 -1 )

```

so ändern wir z.B. auf:

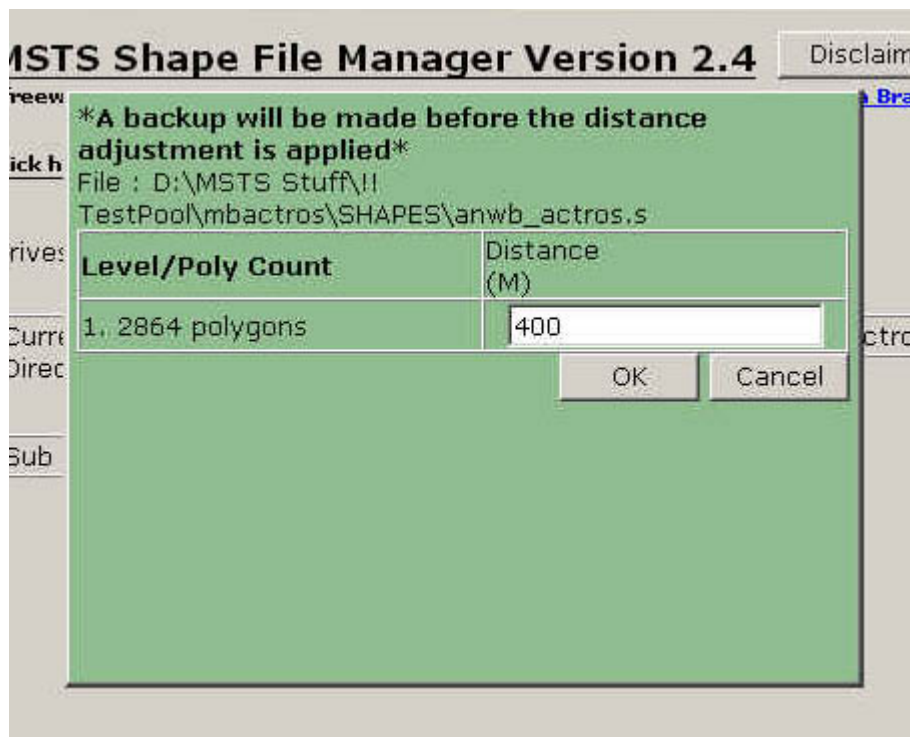
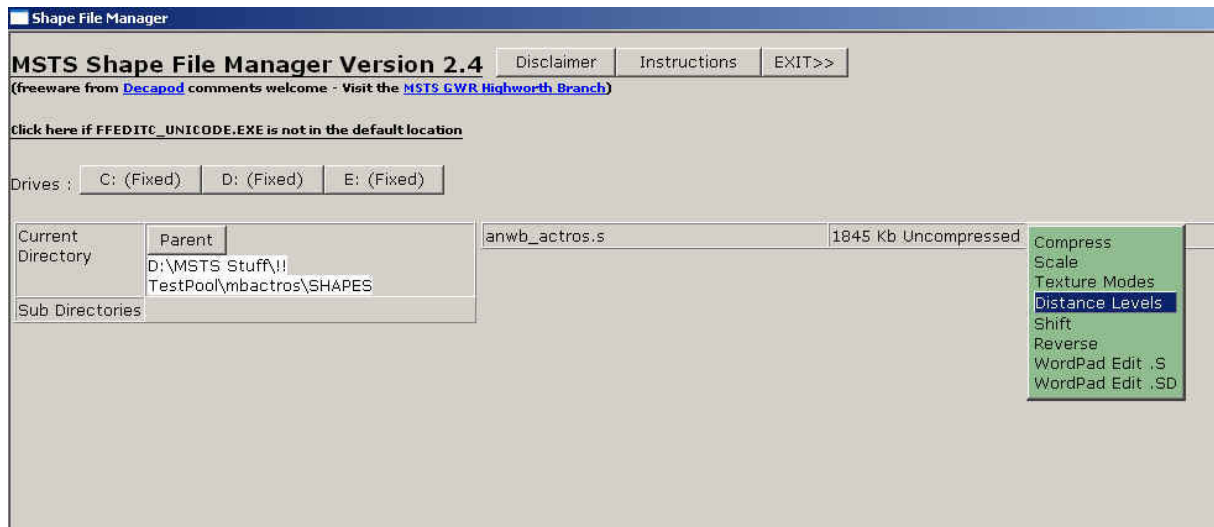
dlevel_selection (**400**)

```

distance_levels_header ( 0 )
distance_levels ( 1
    distance_level (
        distance_level_header (
            dlevel_selection ( 400 )
            hierarchy ( 1 -1 )

```

Diese Anpassung der Sichtbarkeitsgrenzen lässt sich auch mit dem SFM unter „**Distance Levels**“ leicht durchführen.



Mit dieser Manipulation werden unsere Rechnerressourcen immens geschont, denn das Fahrzeug braucht nun nicht mehr über die volle Reichweite von 2000m laufend berechnet werden, sondern nur mehr über 400m. Also eine Ressourceneinsparung von etwa 80 % !!!!

Günstige Werte für diesen Parameter wären in etwa:

200 für Fahrräder, Mopeds, Motorräder,

300 für PKW's

400 für LKW's

500 für LKW's mit Anhänger, oder Sattelschlepper.

Größere Sichtbarkeitsgrenzen würden, außer unnötiger Ressourcenbelastung, nichts bringen, denn man kann die Fahrzeuge in der Simulation ab einer gewissen Entfernung sowieso nicht mehr, oder nur mehr schemenhaft, wahrnehmen.

c.) Nun sehen wir die *.sd Datei unseres Fahrzeuges an. Diese sieht bei unserem konkreten Beispiel folgendermaßen aus:

```
SIMISA#####JINX0t1t_____  
Shape ( anwb_actros.s  
    ESD_Detail_Level ( 0 )  
    ESD_Alternative_Texture ( 0 )  
    ESD_Bounding_Box ( -0.881151 0.038984 -4.146051 0.895756 3.765576 4.331665 )  
)
```





Wir entfernen nun die Zeile mit „ESD_Bounding_Box“ komplett. Danach sieht die Datei folgendermaßen aus:

```
SIMISA#####JINX0t1t_____  
Shape ( anwb_actros.s  
    ESD_Detail_Level ( 0 )  
    ESD_Alternative_Texture ( 0 )  
)
```



Damit sparen wir wieder einiges an Rechnerressourcen, denn nun braucht die Kollisionsabfrage des Fahrzeuges nicht mehr mitberechnet werden. Dies ist auch nicht nötig, denn selbst bei einer definierten Zusammenstoßwahrscheinlichkeit von 100% konnte ich bisher keine Kollision von Straßenfahrzeugen mit Zügen beobachten.

d.) Nun entfernen wir noch die bei der Manipulation der Pivotpunkte automatisch vom „SFM“ erstellten Backupdateien. Diese sind leicht an der Endung „PreShift“ zu erkennen.

Vorher:



 anwb_actros.s	1.846 KB	MSTS Shape
 anwb_actros.s.PreShift	1.843 KB	PRESHIFT-Datei
 anwb_actros.sd	1 KB	SD-Datei
 anwb_actros.sd.PreShift	1 KB	PRESHIFT-Datei

Nachher:



 anwb_actros.s	1.846 KB	MSTS Shape
 anwb_actros.sd	1 KB	SD-Datei

e.) Als letzten Schritt sollten wir unsere *.s Datei wieder komprimieren. Damit beschleunigen wir das Nachladen der Dateien und sparen auch Platz auf der Festplatte. Mehr nicht.

Vorher:

 anwb_actros.s	1.840 KB	MST5 Shape
 anwb_actros.sd	1 KB	SD-Datei

Nachher:

 anwb_actros.s	72 KB	MST5 Shape
 anwb_actros.sd	1 KB	SD-Datei

Womit wir die **Abstimmung** des Fahrzeuges endgültig **abgeschlossen** haben. Der Lohn unserer Arbeit ist ein optisch ansprechendes Bild des gesamten Straßenverkehrs in der Simulation. Vergleiche den nächsten Screenshot mit dem ersten Screenshot dieses Tutorial.



Das oben beschriebene Verfahren konnte ich in den allermeisten Fällen erfolgreich anwenden. Es gab lediglich bei ganz wenigen Fahrzeugen geringe Probleme. Diese wirkten sich jedoch auch nur insofern aus, dass sich nicht das komplette Fahrzeug gegenüber seinem Pivotpunkt verschieben ließ, sondern nur Teile davon.

Dieser Effekt trat jedoch überhaupt nur bei Fahrzeugen mit Anhängern auf. In so einem Falle greift man einfach wieder auf die ursprüngliche Datei oder das Backup des „SFM“ zurück und setzt einen entsprechend großen Wert im „CarSpawnerItem“.

6.) Abschließende Tipps und Hinweise

Meine persönlichen Erfahrungen haben gezeigt, dass der Einsatz von vielen Straßenfahrzeugen mit vielen Polygonen (so ab etwa 1500 - 2000 Polygonen, ohne LOD's) gewisse Risiken in sich birgt. So kann es passieren, nachdem sich solche Fahrzeuge vor Bahnübergängen stauten, dass danach die Simulation abstürzt, da Formen benachbarter Tiles nicht mehr nachgeladen werden können.

Um dies zu vermeiden, sollte man generell Ressourcen fressende Straßenfahrzeuge ausmustern, den Zeitabstand mit dem der Carspawner Fahrzeuge ausspuckt auf Maximum stellen, und auch die Länge der befahrenen Straßenstücke möglichst minimieren.

Viel Freude mit dem Ergebnis !

Liebe Grüße aus Kärnten
(c) by Don Ultimax 2004
Mail: don.ultimax@gmx.at



Mein Spendenprojekt „Rosental“: <http://msts.rosental.shipstut.com>
<http://msts.ultimax.shipstut.com>