Mod erstellen: Leitfaden / Anleitung für Anfänger [TPF1]

Inhaltsverzeichnis

- 1 Struktur und Inhalt eines Mods
- 2 Das Modell
- 3 Die UV Map
- 4 Die Textur
- 5 Modell exportieren
- 6 Exportierte Dateien überprüfen
- 7 Materialien konfigurieren
- 8 Weitere Texturen erstellen
- 9 Texturen im DDS Format
- 10 Die ersten Tests
- 11 Multiple Units
- 12 Animationen
- 13 Model: Lods, Passagiere, Kamerapositionen, Wendezugfeature
- 14 Sound
- 15 Mod veröffentlichen

Hey,

Dies ist der Leitfaden für **Transport Fever 1**

Hier gibt es den Leitfaden für Transport Fever 2: Modding Guide: Anleitung für Anfänger zum Erstellen eines Mods

Sollte jemand den Artikel verwenden um das Modden zu lernen, bitte ich um Feedback. Auch sonst gerne weitere Vorschläge, Wünsche und Kritik. Vielen Dank!

bevor ich mit dem Modding angefangen habe, wünschte ich mir eigentlich nur meine Lieblingsfahrzeuge. Ich hatte keine Ahnung vom Modding, wie Mods funktionieren und wie man sie erstellt.

Ich möchte hier einen Überblick geben, um künftigen Moddern die Basics zu erklären und eine Vorstellung zu geben, was für einen Fahrzeugmod benötigt wird.
Hier geht es jetzt erstmal nur um Fahrzeuge - Bahnhöfe und platzierbare Assets unterscheiden sich jedoch nur in den letzten Schritten (Skript). Infos zum Aufbau der verschiedenen Mod Typen und deren Bestandteile gibt es auch im offiziellen Wiki

Hier ein sehr guter Leitfaden als Ergänzung von Seamon mit konkreteren Handlungsanweisungen: https://www.transportfever.net…sten-Mod-erstellen-2-pdf/
Ein etwas älterer Moddingleitfaden mit Videomaterial ist hier Moddingleitfaden
Hier gibt es ebenfalls Videos von Zargom: Moddingtutorials für TPF Vidcasts/Livestreams

Auf französisch (aber mit Bildern) gibt es hier eine Version: https://docs.google.com/docume…a5KtNCVKEkKVDm930e90/edit

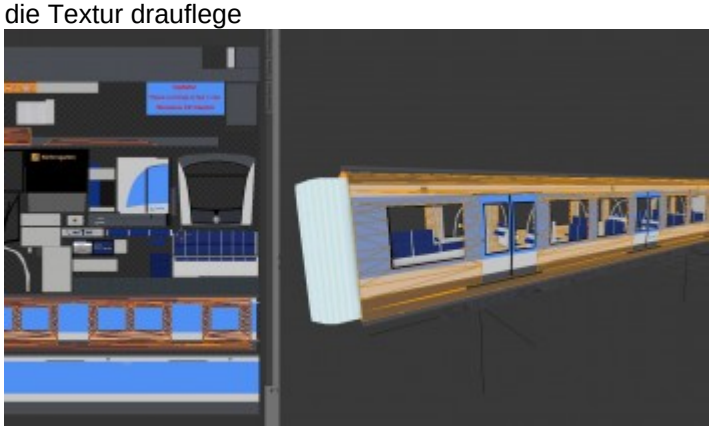
Vorab noch eine Information zum Zeitaufwand. Für eine Tram brauche ich einige Wochen, an meinem ersten Triebwagen habe ich über ein halbes Jahr gearbeitet.
Transport Fever Mods sind recht komplex und keine Sache von zwei Stunden. Es bedarf hier einiges an Motivation, Wille und Zielstrebigkeit.
Das eigene Modell dann (auch bei anderen) fahren zu sehen, ist jedoch eine tolle Bestätigung für die reingesteckte Arbeit.

1 Struktur und Inhalt eines Mods

Generell kann man sich viel von vorhandenen Mods abschauen. Die Struktur ist immer gleich.

Man hat ein 3D Modell, welches mithilfe von CAD-Programmen, wie zum Beispiel Blender, modelliert wird.
Dieses benötigt eine Textur, also Farben, Aufschriften etc. Bei der Textur handelt es sich um ein ganz normales Bild, welches man in Bildbearbeitungsprogrammen wie z.B. GIMP erstellt.
Damit das Spiel weiß, welche stelle vom Bild auf welche Stelle des Modells gehört, "mappt" man das Modell mithilfe einer UV Map. Mit dieser wird Modell und Textur vereint.

Das ganze sieht dann zum Beispiel so aus. Das orangene sind die Flächen meines Modells, welche ich auf die Textur drauflege



Nun legt man für einzelne Bereiche spezielle Materialien fest. Diese sagen Transport Fever, welche Eigenschaften die Oberfläche haben soll. Fenster sind transparent und benötigen ein anderes Material, als z.B. die Wände, oder die leuchtenden Scheinwerfer.
Mit Materialien stellt man neben Effekten wie Leuchtkraft (für Scheinwerfer) auch die korrekte Darstellung von Transparenz sicher.

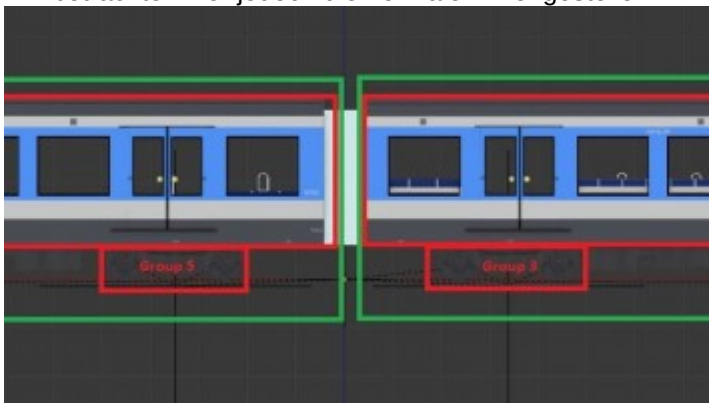
Dieses Modell wird jetzt mit einem speziellen Exporter in für Transport Fever lesbare Dateien konvertiert.
Das sind die .msh und die .msh.blob Dateien. Die .msh Dateien können mit einem einfachen Texteditor (z.B. Notepad++) verändert werden. Die Dateien basieren auf der Programmiersprache Lua und sind leicht verständlich und editierbar. Dort kann man beispielsweise das Material verändern oder Türanimationen einfügen.
In den msh.blob. Dateien ist unser Modell mit UV Map integriert, sie bestehen aus kryptischen Zeichenfolgen und können uns egal sein.
Jedes Objekt (z.B. Sitze, Außenhülle, Räder) wird einzeln exportiert, bekommt seine eigene .msh und .msh.blob Datei.

Diese Objektdateien werden nun weiter sortiert und im Skript zusammengebaut. Hier gibt es dann Unterschiede zwischen Tram, Triebwagen, oder Bahnhof. Wir schauen uns jetzt erstmal nur die normalen Wagons und Triebwagen an.

- Multiple Unit für Triebwagen(.lua)
--- Model (.mdl)
----- Group (.grp)
------- Mesh (.msh/.msh.blob)

Das Mesh/Objekt kennen wir schon.
Diese Meshs werden nun in Gruppen eingeteilt. Diese Gruppe gibt an, wo der Kontakt mit der Schiene im Spiel sein wird.
Bei einem Triebwagen nutzt man meist eine Group, welche die Meshs des Frontwagens beinhaltet, sowie eine für die Meshs des Mittelwagens.

Jede Gruppe z.B. "Wagon 1" besitzt dann noch zwei weitere Untergruppen "Drehgestell 1" und "Drehgestell 2", welche den eigentlichen Kontakt mit der Schiene festlegen, da wir unseren Wagon ja beweglich auf den Drehgestellen haben wollen.
Manchmal bieten sich als Alternativen, vor allem bei Trams und Bussen, Fake Bogies (fake Drehgestelle) an.
Wir betrachten hier jedoch die normalen Drehgestelle.



Meist erstellt man mehrere Exemplare der selben Group, welche verschieden detailreiche Meshs enthalten. Sobald man mit der Kamera weiter weg geht, wird die zweite Group (das zweite Lod = Level of detail)

eingeblendet, in der dann sehr detailreiche Meshs wie z.B. Haltestangen und Inneneinrichtung nicht mehr vorkommen. Dafür dürfen die einzelnen Objekte im CAD Programm nicht verschmolzen werden, sondern müssen nach dem Export als eigenes Mesh vorhanden sein (eigene .msh/.msh.blob Datei).

In der .mdl werden dann die verschiedenen Lods/Groups verlinkt und weitere Einstellungen wie Preis, Gewicht, Kapazität, oder Ladegeschwindigkeit getroffen.
Bei Triebwägen kann man dann in der MultipleUnit.lua mehrere .mdl aneinanderreihen, sodass man mit zwei Wagons (zwei .mdl) einen beliebig langen Zug bauen kann.

Des Weiteren gibt es Ordner, wo unsere Materialien liegen, in denen wir die Textur verlinken. Für die gibt es dann nochmal extra Ordner, genauso wie für z.B. Soundeffekte.
Neben unserer normalen Textur gibt es nämlich noch weitere, beispielsweise für Spiegelungen, Schmutz, oder Einfärbbarkeit.
Da man in Transport Fever möglichst ressourcensparend modelliert und nicht jede Niete einzeln modelliert, gibt es auch extra eine Textur (Normalmap), mit der der Zug noch etwas plastischer wirkt, ohne extra modellieren zu müssen. Dazu später mehr.

Jedes Stückchen hat also seinen eigenen Ordner, wird über Dateien verlinkt und dann wie ein Puzzle zusammengebaut. Das erlaubt eine gute Anpassbarkeit immer die gleiche Struktur, egal ob Triebwagen, Lokomotive, oder Tram. Zwischen den Fahrzeugtypen gibt es dann nur noch Unterschiede in den Einstellungen oder dem Ordnernamen (tram statt train z.B.)

2 Das Modell

Für die Modellierung und die UV Map, sowie den Export verwende ich Blender, da es kostenlos und sehr umfangreich ist. Mit Version 2.8 wurde das Interface stark überarbeitet, wodurch es nun weniger Shortcutlastig und weniger verschachtelt ist.
Die meisten (Internet- und Video) Tutorials werden daher nicht ganz korrekt sein. Ich verwende hier ebenfalls das alte Blender 2.79.
Einige verwenden zur Modellierung auch Sketchup, da es übersichtlicher und einfacher zu bedienen ist. Bei Funktionen und vor allem Rundungen ist es jedoch begrenzt. Für UV Map und Export muss es dann auch nach Blender exportiert werden.
Ich persönlich empfehle Blender, da man hier alles beisammen hat.
Die Einstiegshürde ist hier zwar höher, jedoch macht sich das meiner Meinung nach bezahlt.

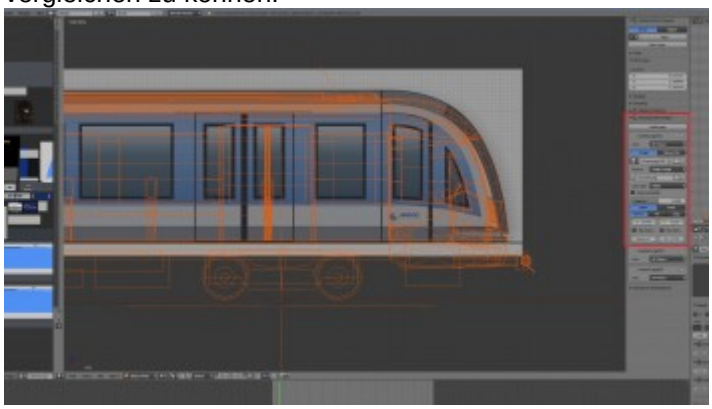
Bei der eigentlichen Modellierung schaut man sich am Besten die grundlegenden Funktionen in normalen Blender Tutorials an.
Es gilt learning by doing. Speziell die Tastenkombinationen, Bedienung und Menü sind nicht ganz einfach.
Jeder hat hier seine eigene Vorgehensweise, Funktionen und Strategie.
Das eigentliche Modellieren hier zu erklären würde den Umfang sprengen, im Folgenden erkläre ich jedoch kurz meine Vorgehensweise mit den wichtigsten Schritten.

https://www.transportfever.net/lexicon/entry/248-mod-erstellen-leitfaden-anleitung-f%C3%BCr-anf%C3%A4nger-tpf1/

Hier gibt es Infos zum Blendern selbst

Vor dem Modellieren ist jedoch die gute Ausgangslage das Wichtigste. Man benötigt genaue Maße, am besten mit technischen Zeichnungen und verzerrungsfreien Bildern. Technische Zeichnungen findet man häufig auf der Herstellerseite, in Büchern, bei anderen Moddern (https://docs.google.com/spread…un4ELYfkL_fGcBtL-a2c/edit hier oder einfach fragen), oder auch auf Anfrage bei den Herstellern. Youtube Videos helfen auch um Details zu erkennen.

Zeichnungen kann man in Blender als Hintergrund einfügen und so die Form vergleichen. Man sollte auf die Qualität der Zeichnung/Bilder achten und nicht blind nachbauen. Perspektive und Detailgrad haben einen großen Einfluss. Daher sollte man möglichst viel Material besitzen um aus verschiedenen Ansichten vergleichen zu können.



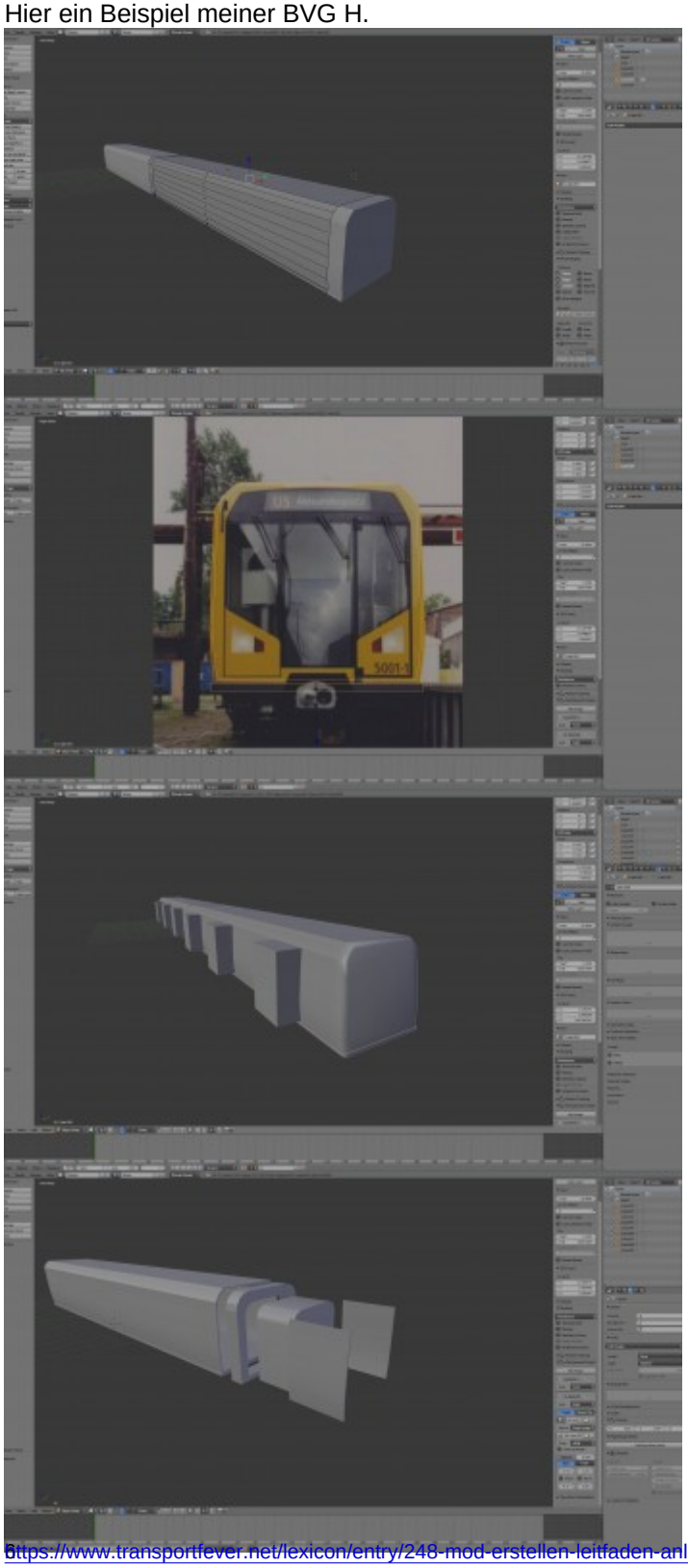
Mein Vorgehen:
Ich beginne meist mit einem Quader, den ich auf die richtigen Maße einstelle. Wichtig ist, dass das Koordinatensystem korrekt ausgerichtet ist. X zeigt nach rechts und Z nach oben. Der Zug fährt später von links nach rechts, der vordere Teil ist also rechts (Numpad 1 und evtl. 5 drücken)
Ich füge Schnittlinien mit Strg + R hinzu, schiebe diese etwas hin und her, runde die Kanten mit Strg + B ab, erweitere Flächen mit E und bastel so mein Modell aus verschiedenen Quadern, Zylindern und Flächen zusammen. Schneiden/Verbinden mit K, verbinden/füllen mit F.

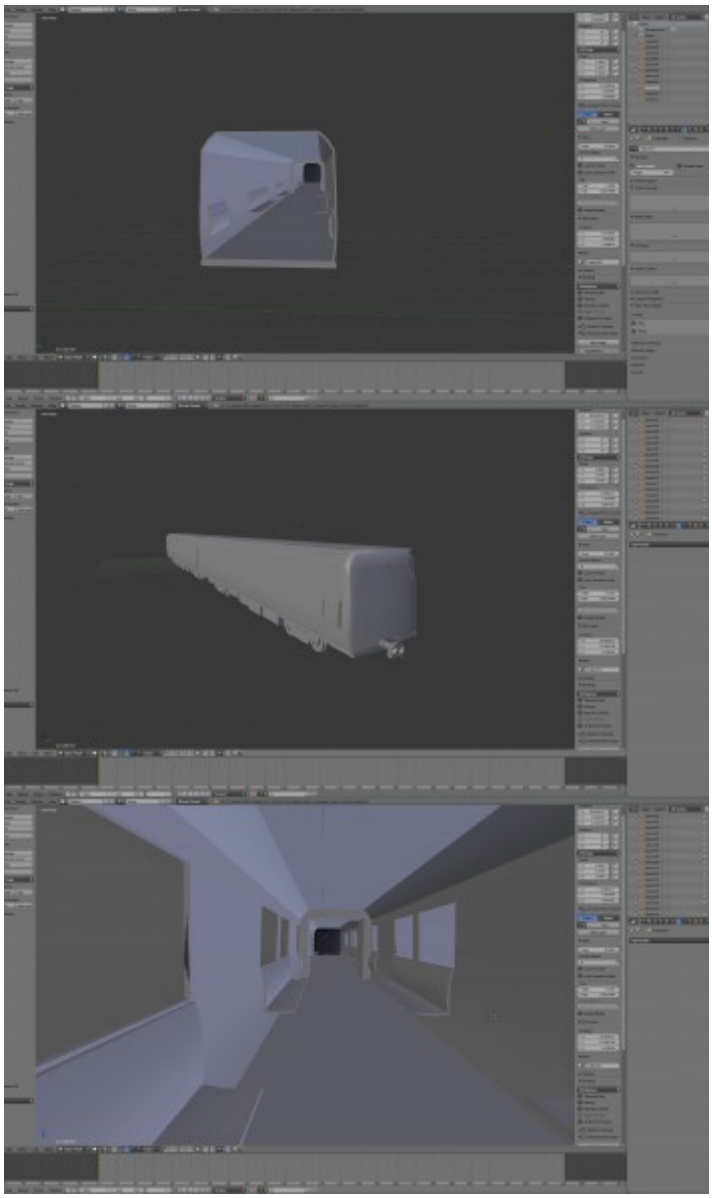
Die Wände sollte man doppelt machen, damit man die Innenverkleidung andersfarbig machen kann und nicht die Außentextur auch innen hat (z.B. blau). Dazu nehme ich die fertige Außenwand, kopiere sie (Shift + D) und skaliere sie etwas kleiner. Das ist nun meine Innenwand.
Man kann auch mit einer Fläche anfangen, diese mit E etwas dicker machen (ca. 0.02- 0.05 m) und dann abrunden. Dann spart man sich das Skalieren, weil die Innenwand bereits dabei ist, muss jedoch darauf achten, dass man außen und innen gleich bearbeitet.

Bei symmetrischen Teilen kann man auch nur eine Hälfte modellieren und dann mit dem Mirror Modifier spiegeln. Vor der UV Map und dem Texturieren sollte man den Modifier anwenden.
Türen muss man ausschneiden, damit sie später animiert werden können. Ich verwende hier den Modifier Boolean. Wenn man Außenschiebetüren hat, kann man die rechten Türen auf der rechten Seite (genauso linke Türen der rechten Seite, gleiches mit linker Seite und 2. Wagon) als ein Objekt zusammenfassen. Dann hat man zum Beispiel drei rechte Türen als ein Objekt, die man gemeinsam verschieben kann (später für die

Animation). Möchte man sie rotieren oder unabhängig voneinander animieren, müssen sie als einzelnes Objekt definiert werden.
Fenster kann man später aufmalen, müssen also nicht ausgeschnitten werden (auch wenn ich das hier gemacht habe). Wenn man eine Innenkamera verwenden möchte, sollte man die Fenster innen entfernen (nur einseitige Fenster)

Hier ein Beispiel meiner BVG H.





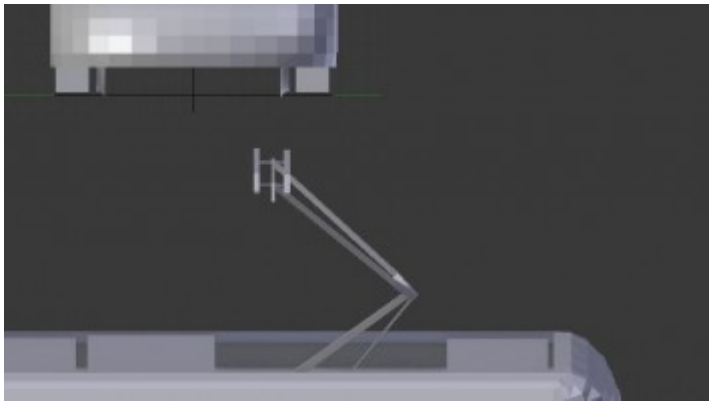
Zuerst habe ich wieder einen Quader nach Maßangaben und Zeichnung erstellt. Den habe ich nun abgerundet, in mehr Flächen aufgeteilt und dann nach Vorbild des Bild angepasst. In Bild 3 habe ich dann die Türen mit den Blöcken ausgeschnitten. Als nächstes der Faltenbalg und die ersten Sitze. Drehgestelle und Unterbau habe ich von meinem anderen Mod wiederverwendet, da die Modelle ähnlich sind. Außerdem habe ich in Bild 6 noch Details wie Scheibenwischer oder Einkerbungen an der Front vorgenommen. In Bild 7 nun die Fenster von innen ausgeschnitten (für Innenkamera).

Das Ziel ist möglichst wenige Dreiecke (Tris) zu verwenden, aber gerade so viele, dass das Objekt noch schön und charakteristisch aussieht. Je mehr Dreiecke man verwendet, desto mehr Ressourcen wird das Fahrzeug brauchen. Richtwerte für eine Tram oder einen Wagon eines Triebwagens sind 10 bis 20 Tausend Dreiecke. Wenn es das Doppelte ist, ist das auch kein Weltuntergang. Fünfach sollte man jedoch vermeiden.
Man kann später Details auch ausblenden, wenn man mit der Kamera weiter weg geht (Lods - Levels of detail), sodass die vielen Tris nicht ins Gewicht fallen.
Bei der Performance kann man auch darauf achten, wenige einzelne Objekte zu verwenden(nicht jeden Sitz

als ein Objekt), da sonst mehr Positionsberechnungen durchgeführt werden müssen.

Man sollte jedoch nicht zu viel zusammen zu fassen, damit man einzelne Objekte in verschiedenen Lods ausblenden kann. Für das Wendezugfeature ist es zudem wichtig, alle wendezugfähigen Objekte (Scheinwerfer rot, Scheinwerfer weiß, Zugzielanzeige Hauptbahnhof, Zugzielanzeige Ostbahnhof) als extra Objekt zu definieren.
Sie werden dann automatisch vom Spiel ein- oder ausgeblendet.
Mit Transport Fever 2 werden Blink- und Bremslichter wahrscheinlich nach dem gleichen Prinzip funktionieren.

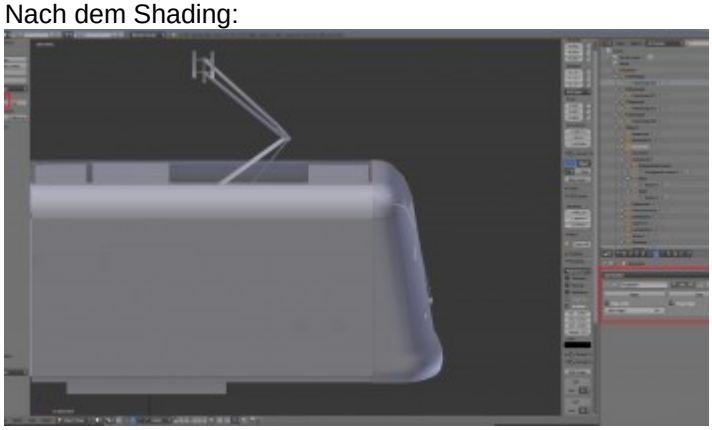
Hier die Variobahn mit Sitz. Man sieht die einzelnen Flächen sehr deutlich - genauer muss es auch garnicht sein.



Abrunden kann man auch ohne mehr Dreiecke zu verwenden, nämlich mit dem Shading (verwendet Schattierung). Damit dabei nicht jede Kante weggesmootht wird, stelle ich beim Edge Split Modifier z.B. 40 Grad ein. Diesen wende ich aber nicht an (sonst trennt er die Kanten, wodurch die Schattierung nicht mehr korrekt ist)!

Man sollte darauf achten, dass bereits in Blender die Schattierung keine komischen Flecken, Flackern, oder Kanten aufweist (die Flächen müssen verbunden sein). Ansonsten mal Strg + N (Richtung der Normals - Außen/Innen), Mesh - Vertices - Remove Doubles, oder Mesh - Clean up - Delete loose versuchen. Wenn nichts hilft manuell die getrennten Punkte auswählen und mit W - Merge zusammenfügen.

Nach dem Shading:



3 Die UV Map

Wenn unser Modell soweit fertig ist, benötigen wir die UV Map und unsere Textur.

Jedes Objekt bekommt mit seinen Flächen nun einen Platz auf der Textur zugewiesen.
In Blender öffnet man am Besten ein zweites Fenster. Auf einer Seite hat man das Objekt, auf der anderen Seite die UV Map (UV Image Editor).

Die Funktion zur Erstellung der UV Map heißt “Unwrap” (U im Edit Mode) und bietet verschiedene Arten, die Flächen aufzuteilen.
Hier muss man einfach ausprobieren und je nach Fall entscheiden, welche der Funktionen die beste Aufteilung bieten. Man kann nur bestimmte Flächen des Objekts auswählen und Fläche für Fläche unwrappen, oder das gesamte Objekt gleichzeitig.
Wichtig ist, dass die Fläche möglichst wenig verzerrt ist, da man sonst nicht sauber texturieren kann. Bei runden Objekten muss man meist in mehrere Teilflächen splitten.

Man sollte darauf achten, die Flächengruppen nicht bündig/zu nah aneinander zu legen, da es sonst später zu Pixelmatsch kommt und nicht mehr klar getrennt werden kann, welche Farbe zu welcher Fläche gehört. Gemeint sind nicht die einzelnen Dreiecke, sondern die unterschiedlichen Objekte, die nicht zusammen gehören.

Der Platz auf der Textur sollte zudem möglichst komplett ausgenutzt werden.
Je größer die UV Map (und je höher die Auflösung der Textur), desto detaillierter kann man später texturieren.
Flächen (Wagon, Türen, …) auf denen man später Beschriftungen, Piktogramme oder Details anbringen

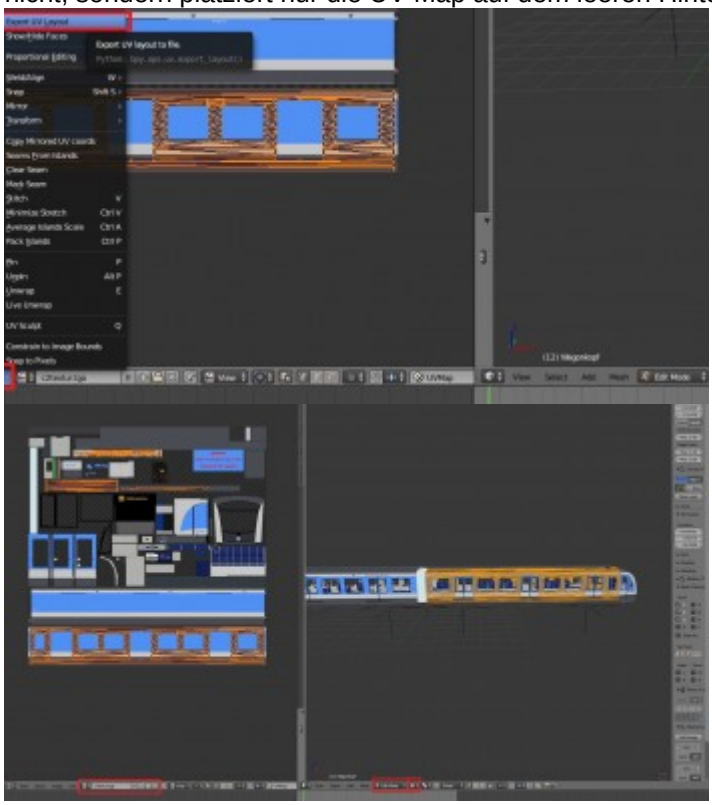
möchte, sollten daher größer gemappt werden, als zum Beispiel die graue Innenraumverkleidung.

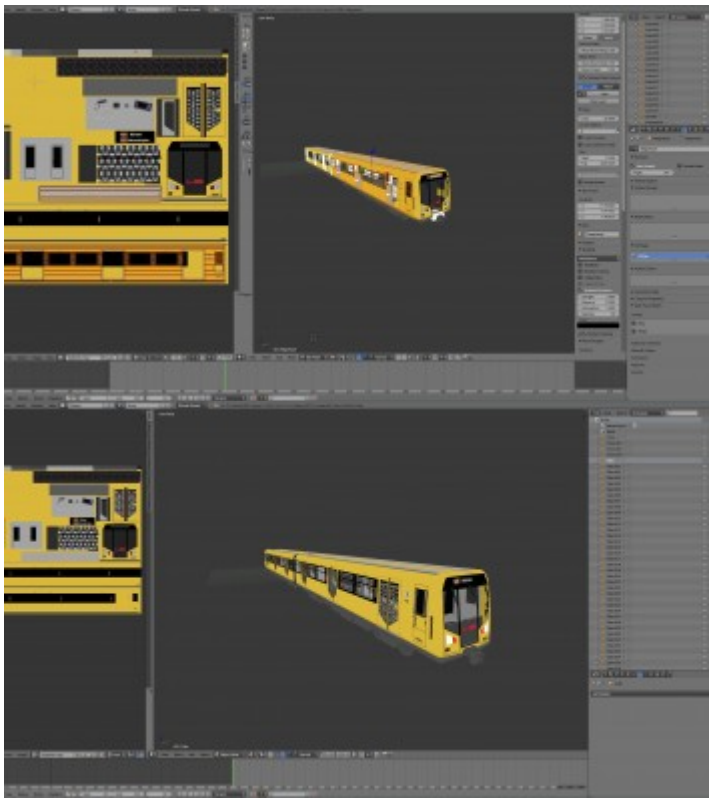
Beim Mappen kann man variieren von “jede graue Innenverkleidung einzeln mappen und texturieren” bis “alles Graue auf einen Fleck klatschen” ? Zitat Profi Modder “Noooooo” !D Image not found or type unknown
Je genauer man mappt, desto mehr kann man im Nachhinein bei Repaints verändern.
Wenn man alles übereinander auf eine einfarbige Fläche legt, kann man dort keine Spiegelungen, Einfärbungen, Schmutz, oder nachträgliche Veränderungen vornehmen, die über einen Farbwechsel hinausgehen.

Bei der Reihenfolge gibt es verschiedene Herangehensweisen.

- Man kann alle Objekte temporär zusammenfassen/verschmelzen (Strg + J), die UV Map für alles erstellen, exportieren und dann die gesamte Textur auf Basis der UV Map machen. Vorteil ist hier, dass man den Platz auf der Textur optimal ausnutzen kann. Am Ende muss man jedoch die Objekte wieder trennen (P im Edit Mode). Damit dies leichter geht, empfiehlt es sich, die Objekte in unterschiedlicher Höhe zu platzieren, sodass sie sich nicht überlappen.
- Man erstellt die UV Map für ein Objekt, texturiert nur dieses eine und wiederholt das für jedes Objekt. Ich arbeite nach diesem Verfahren, switche dabei häufig zwischen Textur und Blender, und kann dadurch leichter Korrekturen an Textur, sowie UV Map (Größe) vornehmen. Hier muss man aber bereits abschätzen, wie groß die einzelnen Objekte werden, damit am Ende alles draufpasst und keine Lücken bleiben. Für die äußeren Texturen verwende ich etwa die Hälfte bis zwei Drittel der Textur.

Exportieren kann man die UV Map im Edit Mode über Uvs - Export UV Layout (Bild 1). Ich habe die Textur direkt eingefügt (Bild 2), da ich gleichzeitig texturiere. Wenn man nach "1." arbeitet, hat man diese noch nicht, sondern platziert nur die UV Map auf dem leeren Hintergrund.





Bei der BVG H habe ich dann im ganz rechten Bild noch Zugzielanzeigen und Leuchten hinzugefügt. Türen sind dort auch eingeblendet.

4 Die Textur

Für die Textur taugt so gut wie jedes beliebige Zeichenprogramm, welches über Paint hinausgeht.
Ich verwende das kostenlose GIMP (2.8 - nicht das neue), viele kommen damit jedoch nicht zurecht. Einige benutzen auch PaintShop, Substance Painter, Photoshop, oder Paint.net.

Das verwendete Programm sollte verschiedene Ebenen unterstützen, mir reichen meistens 5 Stück. Andere texturieren mit 100 verschiedenen Ebenen.

Letzten Endes muss die Farbe drauf. Ebenen können helfen, das ganze variabel und nachträglich änderbar zu machen. Wenn man alles auf einer Ebene macht, kann man für andere Mods wenig wiederverwerten.

Eine wichtige Entscheidung ist die Auflösung der Textur.
Je höher die Auflösung, desto detaillierter kann man texturieren. Vor allem bei Rundungen begrenzen die Pixel die Qualität. Auch hier muss man einen Mittelweg zwischen ausreichender Qualität und verbrauchter Leistung finden. So klein wie möglich, so groß wie nötig.

1024x1024 oder 2048x2048 Pixel sind gängig. Ich verwende fast immer 2048x2048 und komme damit klar.

Die Textur speichert man bei GIMP als Gimp Datei, da dort alle Ebenen vorhanden bleiben. Hier kann man später leicht Änderungen vornehmen.
Für Transport Fever exportieren wir - dabei wird alles sichtbare in eine Ebene zusammengefasst. Als Format am Besten .tga verwenden (Den Haken bei RLE Kompression wegmachen!). Später werden wir dann auf DDS wechseln.

Wenn wir die Textur noch nicht in Blender eingefügt haben, tun wir das jetzt (siehe Bild oben).

5 Modell exportieren

Das Wichtigste steht nun.
Jetzt kommt der Export.

Für das Exportieren wird das Blender Addon von Merk benötigt. Es funktioniert noch nicht mit dem neuen Blender 2.8. Animationen lassen sich ebenfalls nicht exportieren, diese in Blender bereits zu erstellen ist unnötig. Wir erstellen sie später im Skript manuell.

Anleitung: Blender Import/Export Addon
Download: Blender Import/Export Addon (aktuell Version 0.5.3 - Alphatest)

Sobald es installiert ist, müssen wir noch einige Vorkehrungen treffen. Ansonsten bekommen wir Fehlermeldungen.

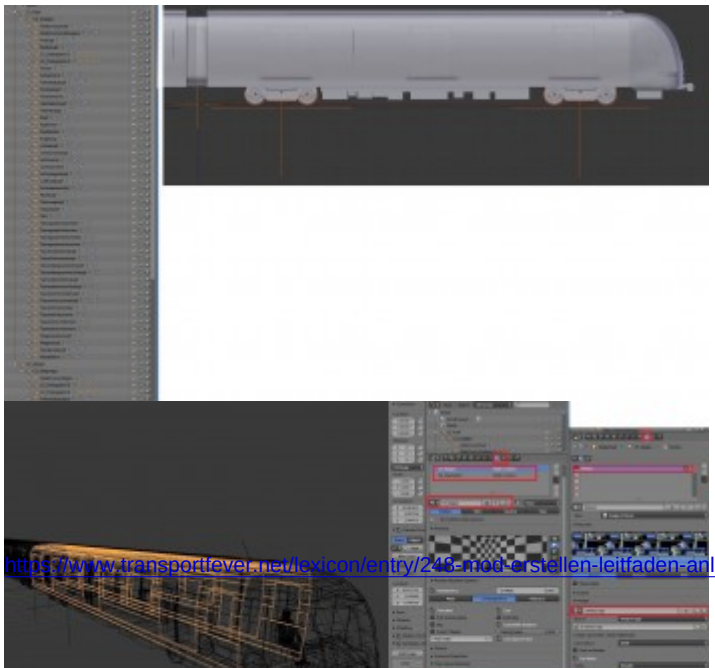
Man kann in Blender bereits alle Eigenschaften des Zuges einstellen, Texturen verlinken und Materialien konfigurieren. Da mir meistens viele Sachen noch fehlen und manche Funktionen nicht gehen (Animationsexport) und ich erstmal nur die Dateien haben möchte, exportiere ich erstmal nur Dummys und stelle den ganzen Rest später ein.
Nur die Materialanzahl, Benennung und Zuweisung sollte ich mir vorher überlegen, da das spätere Ändern umständlich ist (man kann auch später neu exportieren).

Ich verwende meist je ein Material für die äußere Wand, die äußere transparente Wand (Fenster), den Innenraum inkl. Innenwand (nichts transparent da Fenster innen entfernt/weggeschnitten) und für die

Scheinwerfer (leuchtend).

Zum Export:

- Das Modell darf nur Flächen mit Drei- oder Vierecke (nicht >4) enthalten. Dazu ein Objekt auswählen, exportieren, wenn die Fehlermeldung "contains ngons" angezeigt wird und man anschließend in den Edit Mode schaltet, sind bereits die ursächlichen Flächen markiert. Dann einfach Mesh - Faces - Triangulate faces wählen und sie werden korrekt in Dreiecke aufgetrennt. Mit dem Modifier Triangulate werden alle Flächen zerschnitten, auch wenn es erlaubte Vierecke waren, die erste Variante ist daher besser.
- Die meisten Modifier müssen vor dem Export angewendet werden. Der Edge Split Modifier sollte/darf nicht angewendet werden. Triangulate muss nicht angewendet, es kann jedoch helfen, wenn man durch den Modifier entstandene harte Kanten noch beheben möchte.
- Wir erstellen ein Model, eine Group für jeden Wagon und setzen sie in den Nullpunkt. Pro Drehgestell erstellen wir nun ebenfalls eine Group. Der Ursprung der Drehgestellgroup muss in der Mitte des Drehgestells (es rotiert um die Z-Achse) liegen. Dabei hilft Shift + S und “Set Origin”.
- Die Drehgestellgruppe wird als Parent der Drehgestellmeshs definiert (Meshs auswählen, dann Gruppe, dann Strg + P). Die Wagongruppe ist Parent der Wagonsmeshs sowie der Drehgestellgruppe. Das Model ist Parent von einer Wagongruppe. Für zwei Wagons haben wir zwei Model.
- Damit sich das Rad korrekt dreht, muss der Ursprung auf der Y-Achse (Rotationsachse) der Räder liegen. Für die Räder wird in den Einstellungen "Object Data" (links neben Material) die Box “Axle” angewählt (könnte man auch später hinzufügen). Für alle anderen Meshs (nicht Drehgestell und Räder) sollte der Ursprung auf dem Nullpunkt liegen. Dazu übernehme ich Position (evtl. Rotation und Skalierung) mit Strg + A.
- Jedes Mesh/Fläche braucht ein Material zugewiesen. Im Reiter Material erstellt man ein neues, benennt es, erstellt im Reiter daneben eine neue Textur (für jedes Material die gleiche Textur) und verlinkt als Image unsere erstellte Textur. Bei den Materialeinstellungen muss zudem für das Standardmaterial “Reflective” unsere Textur für map_color_reflect angegeben werden. Im Edit Mode kann man den einzelnen Flächen nun mit Assign unterschiedliche Materialien zuweisen.
- Wir wählen Wagonmodel1 aus, Wagongroup1, alles was dadrunter ist (Meshs, 2 Drehgestellgroups und dessen Meshs) und exportieren. Bei den Exporteinstellungen wählen wir als Typ train, man kann noch einen Unterordner angeben (z.B. den Namen des Zuges) und damit wir nicht das zweite Model gleichzeitig exportieren, wählen wir nur NGons und nicht Unselected.
- Später machen wir das mit dem zweiten Wagon und dem zweiten Model. Zur Vereinfachung benutzen wir erstmal nur eins.
- Fertig exportiert. Im Pfad befindet sich nun der Ordner res, in dem unsere Dateien liegen.



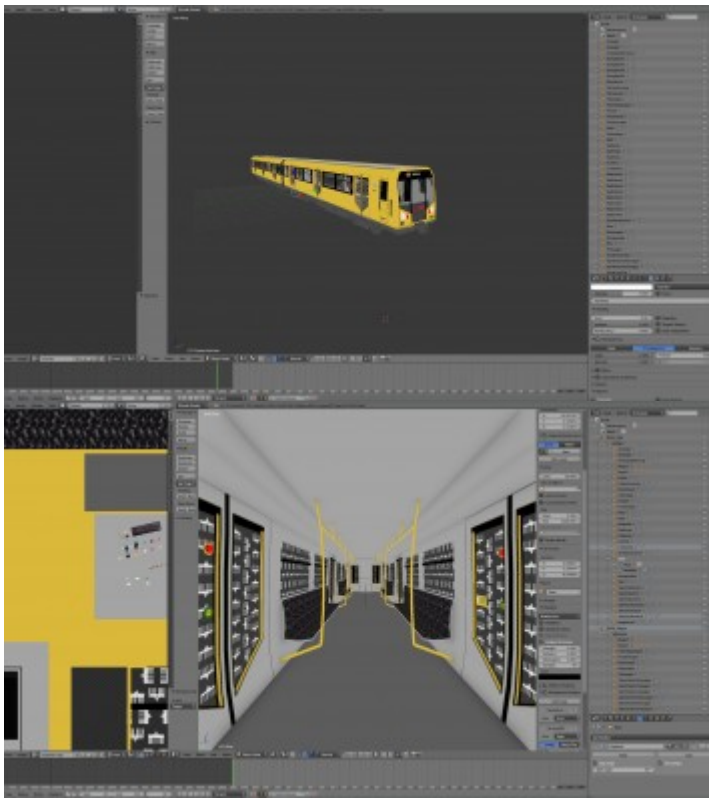


Bild 1: Ursprung für Rad und Drehgestell(gruppe) und korrekte Parentzuweisung, jedoch mit vielen Einzelmeshs (kann man zusammenfassen)
Bild 2: Material "Wagon" den äußeren Flächen und Material "Innenraum" den Inneren Flächen zugewiesen. Mit Assign übernommen. Zudem Texturen für Material eingestellt.
Bild 3: Bei der BVG H habe ich nun noch die Objekte richtig benannt (Mesh ist das wichtigste, zur Übersicht auch das Blenderobjekt)
Bild 4: Dort ist nun auch die korrekte Gruppierung rechts im Stammbaum, Haltestangen habe ich spontan auch noch hinzugefügt.

6 Exportierte Dateien überprüfen

Wir schauen uns jetzt mal die einzelnen Dateien an. Wir öffnen die .msh Dateien (models - mesh - vehicle - train - evtl. unterordner) mit einem Texteditor (z.B. Notepad++) und finden dort einen Verweis zu unserem Material. Hier ist es hilfreich, wenn man sie vorher in Blender schon richtig zugeordnet und benannt hat. Dann müssen wir nichts ändern. Später werden wir hier auch Türanimationen einfügen.

Als nächstes schauen wir uns die Groups an (models - group - vehicle - train - evtl. unterordner). Auch diese öffnen wir mit dem Editor. Dort vergewissern wir uns, dass die Pfade zu unseren Meshs korrekt sind. Hier sollte auch auf Groß- und Kleinschreibung geachtet werden. Bei Windows ist diese egal, bei Linux jedoch nicht, weshalb es dann bei manchen Usern zu Crashs kommen kann.
In den Wagongroups sind dann auch die Drehgestellgroups vorhanden und verlinkt. Die vielen Nullen und

Einsen bestimmen die Ausrichtung der Modelle. Man könnte sie hier mit der Transformationsmatrix https://www.transportfever.net…83-Transformationsmatrix/ noch ändern. Ich empfehle jedoch Änderungen direkt in Blender vorzunehmen und dann neu zu exportieren, sodass in Blender immer alles auf dem aktuellsten Stand bleibt.

Wenn das passt öffnen wir unser Model (models - model - vehicle - train). Dort prüfen wir, ob der Pfad zu unserer Wagongroup korrekt ist. Außerdem können wir hier bereits Werte in der metadata einfügen. Dazu gehören zum Beispiel Verfügbarkeit, angezeigter Name und Beschreibung, Kapazität, Geschwindigkeit, Lebensdauer, oder Gewicht. Am Besten kopiert man diesen Teil von einem anderen Mod und ersetzt die Werte. Hier Beispiel: Mdl_Beispiel_Basis.zip
-1 bedeutet automatische Berechnung und wird für Kaufpreis und Wartungskosten eingetragen.
Wenn wir in Blender die Räder noch nicht mit “Axle” markiert haben, müssen wir manuell den Pfad zu unseren Radmeshs hinzufügen.
Später fügen wir hier auch extras wie Passagiere, Wendezugfähigkeit, Kamerapositionen, oder Sounddatei ein.

7 Materialien konfigurieren

Als nächstes kümmern wir uns um die Materialien. Hier eine Liste und ihre Eigenschaften.

Die Materialdateien befinden sich bei material - vehicle - train. Möchte man sie in einen Unterordner tun, muss darauf geachtet werden, dass in jeder .msh Datei der richtige Pfad zum Material steht.

Für das Material Wagontransparent verwende ich PHYS_TRANSPARENT_NRML_MAP_CBLEND_DIRT , für Wagon und Innenraum (keine Transparenz) PHYSICAL_NRML_MAP_CBLEND_DIRT und für Emissive (leuchtend) EMISSIVE.

Zusatzinfo für EMISSIVE:
Es hat sich herausgestellt, dass für rote, leuchtende Scheinwerfer am Besten eine extra Textur verwendet wird, die nicht im .tga/DDS Format, sondern in .hdr (Photoshop) Format ist. Zudem funktioniert EMISSIVE hinter transparenten Flächen erstmal nicht.
Man kann mit einem nicht leuchtenden Typ und den Material-Koeffizienten eine leuchtende Zugzielanzeige oder rote Scheinwerfer erzeugen oder für einfarbige Leuchten (keine ZZA) diesen Trick verwenden Emissive hinter Glas

Je nach Materialtyp hat die Datei nun andere Inhalte. Am Besten kopiert man diese von anderen Mods und ersetzt nur den Pfad zur Textur. Hier sindBeispielmaterialien.zip
map_albedo (_opacity bei transparenz) ist unsere normale Textur, für die wir den Pfad angeben (befindet sich bei textures - models - vehicle - train - evtl. unterordner; der Pfad wird ohne textures angegeben).

Es gibt nun noch weitere benötigte Texturen. Das sind Glossmap/mga (metal gloss ao), Dirtmap/cbr (cblend dirt rust) und die Normalmap.
Die Glossmap gibt an, wie stark welche Stelle spiegeln soll (Metalleffekt, Reflektion und Ambient Occlusion = Schattierungen). Auf der Dirtmap legt man fest, an welchen Stellen wie viel Rost und Schmutz auftaucht und welche Stellen sich mit der Einfärbefunktion einfärben lassen.
Mit der Normalmap können wir Stellen plastisch wirken lassen, die eigentlich eben sind. Da wir ressourcensparend modelliert haben, kann man damit kleine Unebenheiten künstlich erzeugen.

Diese Texturen kann man entweder jetzt erstellen, oder einen Platzhalter verwenden. Dann hat man für einen ersten Test noch keine Spiegelungen oder Schmutz.
Die Platzhaltertexturen (Auflösung 2048x2048) einfach zu den anderen Texturen legen, den Pfad verlinken und die nächsten zwei Kapitel überspringen. platzhaltertexturen.zip

8 Weitere Texturen erstellen

Nun müssen wir die oben angesprochenen weiteren Texturen erstellen.
Hierfür gibt es einige nützliche Lexikoneinträge.

Texturen und die Kanäle
Glossmap und Texturen (ein zweiter Versuch)

Für jemanden, der keine Erfahrung in der Bildbearbeitung hat, sind diese jedoch teilweise zu kompliziert. Ich arbeite selbst etwas schlampig und nicht professionell, daher erkläre ich es hier auf meine sehr simple Weise.

Jede Farbe besteht aus den Bestandteilen Rot, Grün und Blau. Das sind die Kanäle. Jede Farbe bestimmt, wie stark eine Eigenschaft ausgeprägt ist.

Laut dem Artikel ist in der Glossmap (metal gloss ao) Rot für den schimmernden Metalleffekt zuständig. Ich öffne meine Farbeinstellung und stelle den Regler bei Rot zwischen 0 und 255 ein. Wie stark der Wert sein muss, muss man ausprobieren. Für moderne Hochglanzfahrzeuge braucht man meist garkein Metall (maximal Wert 30). Dafür benötigt man dort stärkere Spiegelungen/Umgebungsreflektionen. Dafür ist Grün zuständig. Ich möchte eine starke Spiegelung und verwende zum Beispiel 210. Blau gibt Ambient Occlusion an. Damit kann man Schattierungen auf der Textur erzeugen. Bei modernen Fahrzeugen ohne große Einkerbungen benötigt man dies meist nicht. Dort wählt man dann einfach 255 für die volle Farbkraft.
Bei älteren Fahrzeugen mit vielen Verwinkelungen kann die AO Map mehr Realismus und mehr Plastizität erreichen.
Wie man sie erstellt steht zum Beispiel in Seamons Leitfaden https://www.transportfever.net…sten-Mod-erstellen-2-pdf/

Je nach Bereich mische ich mir somit meine Farbe zusammen. Einige Farbcodes als Beispiel: Hochglanzwände/Fenster (man kann noch mehr grün geben) 15d7db, Innenraum/Sitze/nicht reflektierende Dinge (keine Reflektion, kein Metalleffekt) 0000ff, Mittelding für Fensterrahmen z.B. 0065e7

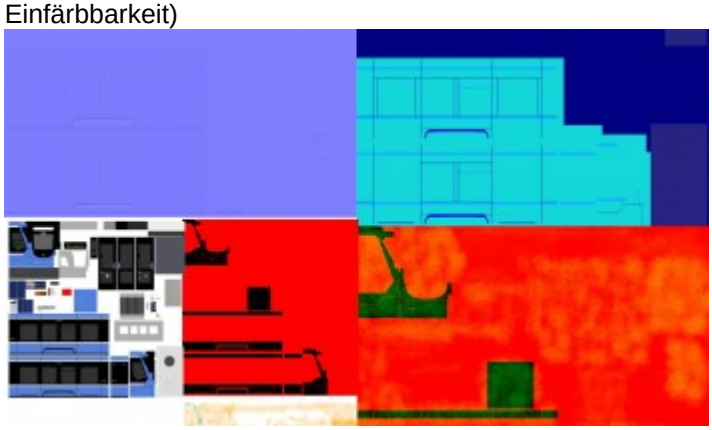
Bei der Dirtmap (cblend dirt rust) ist Rot für die Einfärbbarkeit zuständig. Ich nehme hier erstmal 255 (keine Einfärbung) und färbe die erste Ebene rot. Dann nehme ich die normale Textur in die zweite Ebene, wähle umfärbbare Bereiche aus und ändere sie in der Dirtmap (erste Ebene) von Rot 255 auf Rot 0 (schwarz). Diese Bereiche wird man dann einfärben können.
Nun kommt Schmutz (Grün) und Rost (Blau). Schmutz erstelle ich mit dem Sprühwerkzeug. Dafür habe ich mir eine Schmutzebene gemacht, die ich immer wieder verwenden kann.
Die Farbe ist dabei Rot 255 (erstmal nicht einfärbbar), Grün 162 (mittlerer Schmutz) und Blau 0 (Kein Rost) ? Farbcode ffa200. Blau wähle ich 0, weil ich bei modernen Fahrzeugen kaum Rost habe und dieser vernachlässigbar ist.
Ich persönlich brauche auch keinen Schmutz in der Nähe von Rädern, Auspuff oder an Kanten. Für mich ist es wichtig, dass das ganze Fahrzeug dunkel, dreckig und fleckig wird. Daher recht simpel eine Textur die ich immer wieder verwenden kann

. Ich fahre einmal horizontal und einmal vertikal in dirtlayeruniversal.zip

Diese Schmutzebene wird nun über die rot/schwarz Ebene gelegt. Jetzt wäre das Problem, dass die einfärbbaren Bereiche zwar einfärbbar wären, der Schmutz jedoch mit Rot 255 es nicht ist. Wir haben dadurch orange auf Schwarz und damit haben die Flächen, auf denen Schmutz liegt, bei Rot nicht mehr 0. Wir wählen also auch in der Schmutzebene die einfärbbaren Flächen aus und vertauschen die Farbe (orange ffa200) durch die Farbe ohne Rotton (Rot auf 0 drehen, dann haben wir 00a200).

Nun noch die Normalmap. Das ist die lila Map, die künstliche Erhebungen hinzufügt.
Auch da könnte man es professioneller machen mit Schwarz/Weiß Textur (? NormalMaps erstellen), ich mache es mir einfach, nehme meine normale Textur, verwende das GIMP Normalmap Plugin (im Internet runterladen) und wende es an. Ein paar Korrekturen nehme ich noch vor (Kontrast erhöhen/verringern für stärkeren/leichteren Effekt; X und Y vertauschen für Einkerbung statt Hervorhebung bzw. andersrum) und fertig.

Hier mal alle Texturen im Beispiel und die Dirtmap unterteilt in die zwei Ursprungslayer (Dreck und Einfärbbarkeit)



Bezüglich der Einfärbefunktion:
Häufig wirkt die Farbe ingame nicht so kräftig, wie im Auswahlmenü dargestellt wird. Dazu muss man die Helligkeit der einzufärbenden Farbe herausfinden.
Bei mir lassen sich blaue Flächen einfärben. Mit dem Farbpicker nehme ich den Farbwert und gebe ihn in diesen Converter ein um die Helligkeit zu bekommen. https://www.rapidtables.com/convert/color/rgb-to-hsl.html

Abhängig davon, ob wir den Wert als 0 - 1 oder 0 - 255 von unserem Grafikprogramm bekommen gibt es folgende einfache Formel:

0 - 1: `x = 0,5 / Helligkeit`

0 - 255: `x = 128 / Helligkeit`

Nun rechne ich 0.5/L Wert aus dem Converter also zum Beispiel 0.5/0.635 und bekomme den nötigen Faktor. Der wird nun im Material zum Beispiel vor dem polygon_offset{} eingefügt.
In dem Bereich kann man auch etwas rumspielen - evtl. findet man noch bessere Werte. Ich habe dann z.B. 0.81 genommen.

Erfahrungswert: Zähler (die 0.5) höher wählen, ca. 0.52-0.54

Code

```
color_blend                                    =                                    {

            },
```

9 Texturen im DDS Format

Das bisher verwendete .tga Format ist in Ordnung, für Transport Fever lesbar und funktioniert auch. Bei DDS hat man jedoch den Vorteil, dass die Textur nochmal komprimiert wird und dadurch Leistung/Speicher spart. Teilweise entsteht etwas Pixelmatsch, da hier mehrere Pixel zusammengefasst werden. Meistens merkt man dies jedoch nicht.

DDS-Texturen

Für Gimp gibt es ein DDS Plugin, welches man installiert. Beim Export sollte man hier alle Ebenen zusammenfügen oder unnötige löschen, da sonst das Plugin durcheinander kommt.
Wenn man wirklich nur noch eine (sichtbare) Ebene hat, muss man diese vertikal spiegeln (Bild - Transformation - Vertikal spieglen). Nun exportiert man sie mit der Dateiendung .dds und wählt je nach Textur folgende Einstellungen.

Für die normale Textur (besitzt Transparenz) DXT5 sowie “generate mipmaps”
Für die mga/Glossmap, sowie für cdr/Dirtmap (keine Transparenz) DXT1 sowie “generate mipmaps”
Für die Normalmap 3DC sowie “generate mipmaps”

Die Texturen ersetzen nun die alten tga Texturen bzw. die Platzhaltertexturen. In den Materialien muss nun der Pfad mit der korrekten Dateiendung .dds korrigiert werden.

Das Modell, UV Map, sowie Texturen sind jetzt fertig. Grundlegende Einstellungen haben wir im Skript bereits getroffen. Bevor wir zusätzliche Features wie Animationen, Lods, Wendezugfähigkeit, oder Passagiere hinzufügen, sollte man das bisherige testen.

10 Die ersten Tests

Beim Ändern von Dateien muss die Welt in Transport Fever immer neugeladen werden, um die Änderungen zu übernehmen. Das ganze geht auch etwas schneller über den Model Viewer.

https://www.transportfever.com…dding:developerinfo:tools

Der Model Viewer kommt in C:\Program Files (x86)\Steam\steamapps\common\Transport Fever

Damit der Viewer und Transport Fever unseren Mod erkennt, müssen wir noch einige Kleinigkeiten hinzufügen.
Der res Ordner kommt in einen Ordner, der nach folgendem Schema benannt wird: Name_Modname_1 . In meinem Fall (Steamname marc345) marc345_C2Inspiro_1.
Neben dem res Ordner, muss dort noch eine mod.lua Datei rein. Diese stellt allgemeine Informationen wie Mod Name, Mod Beschreibung, oder Autor bereit.
Dazu wieder eine bereits vorhandene kopieren und durch die eigenen Informationen ersetzen.mod lua Beispiel.zip
Unseren Mod platzieren wir jetzt in C:\Program Files (x86)\Steam\steamapps\common\Transport Fever\mods

In Transport Fever kann man den Mod nun ganz normal aktivieren und im Depot kaufen.
Im Model Viewer auf " …" klicken und dann ist er dort ebenfalls sichtbar. Man kann nun ein Model auswählen und anzeigen lassen. Dort findet man auch Einstellungen für Schmutz, Farbeinfärbung, oder Passagiere.

Man sieht nun die Farben, Spiegelungen und vorherigen Arbeitsschritte, wie sie im Spiel aussehen. Sollte es Unstimmigkeiten geben, kann man diese nun korrigieren.

Sollte das Spiel oder der Viewer abstürzen, sollte man sich die Fehlermeldung im Viewer (hat mehr Fehlermeldungen) oder in der stdout.txt (wenn in Transport Fever getestet) anschauen. Sollten sie unklar sein, am Besten nachfragen.

Ingame werden im Depot, Linienmanager etc. aktuell noch lila Platzhalter für unseren Zug verwendet. Im Model Viewer können wir über den Button Screenshots die Ui Bilder erstellen lassen (werden im Transport Fever Verzeichnis gespeichert).
Diese müssen jetzt nur noch im textures - ui Ordner eingefügt werden. Den genauen Pfad siehe andere Mods.

Wenn nun alles passt, können wir noch die restlichen Funktionen hinzufügen.

11 Multiple Units

Für multiple Units (Triebwägen) wird eine extra Datei erstellt (Textdatei mit Endung .lua Speichern), in der mehrere .mdls zu einem Triebwagen zusammengebaut werden.
Man kann dort auch Wagons mit forward = false spiegeln.
Sie befindet sich bei config - multiple unit und könnte so aussehen:
Code

```
function data()
return {

                                {    name    =    "vehicle/train/C2_Kopf.mdl",    forward    =
                               {    name    =    "vehicle/train/C2_Wagon.mdl",    forward    =    t
                               {    name    =    "vehicle/train/C2_Wagon.mdl",    forward    =    t
                              {    name    =    "vehicle/train/C2_Wagon.mdl",    forward    =    fal
                              {    name    =    "vehicle/train/C2_Wagon.mdl",    forward    =    fal
                        {   name   =   "vehicle/train/C2_Kopfhinten.mdl",   forward   =   false   }


    },
                                                                    name            =           _("Siemens
                                                                                                  fahrt.")
}
end
```

Alles anzeigen

Multiple Units können nur in Transport Fever im Depot gekauft werden - der Model Viewer unterstützt sie nicht.
Damit im Depot die einzelnen Wagons nicht mehr käuflich sind, muss in den .mdl Dateien MultipleUnitOnly = true gesetzt werden.

12 Animationen

Türen animieren

Animationen sind gefährlich. Wenn du bisher noch keinen Absturz hattest, dann wirst du spätestens jetzt einen bekommen. Sobald hier ein Element falsch ist, funktioniert es nicht mehr.
Animationen sind aber auch schön. Hier kann man richtig viel Zeit reinstecken, leuchtende LEDs hinzufügen und ein Türschließspektakel veranstalten.
Beim C2 Inspiro habe ich die Türleuchten alle (als extra Objekt damit unabhängig) animiert (pro Seite 6 Stück: rot Innen, grün innen, rot rechte Türe, rot linke Türe, grün rechte Türe, grün linke Türe)
Es geht aber auch einfach - damit fangen wir an.

In der Mdl wird bei den events automatisch eine Animation ausgelöst. Über die Angabe der Group wird diese Animation an das Mesh weitergegeben. Im Mesh ist dann die eigentliche Animation gespeichert.
Man kann Objekte rotieren und verschieben. Wenn das Schließen genau wie das Öffnen ablaufen soll (gleiche Dauer, gleicher Ablauf), dann muss man nur das Öffnen erstellen und spielt dann das Schließen einfach rückwärts ab.

Das versuchen wir jetzt. Wir beginnen in der Mesh Datei einer Türe. Hat man bei Außenschiebetüren zum Beispiel alle rechten Türen der rechten Seite als ein Objekt definiert, weil man sie gleich animieren möchte, sparen wir Arbeit.

Wir öffnen also nun das Mesh der rechten Türen der rechten Seite.

Code

```
animations = {

{

},
{

},
{

},
{

},
},

},

},
},
```

Alles anzeigen

Die Animation wird direkt nach return vor den MatConfigs eingesetzt.

Wir erstellen eine neue Animation namens opendoor. Jeder Block ist ein Keyframe. Dort gibt man an, zu welchem Zeitpunkt sich das Objekt wo befindet. Ausgangslage 0.
Nach einigen Zwischenschritten ist die Türe nach 2300 Millisekunden um 0.765 Meter in X verschoben und um 6 Zentimeter in Y verschoben. Die Werte findet man am Besten in Blender heraus.
Zum Test reicht auch erstmal nur die Start und die Endposition. Zwischenschritte für eine rundere Öffnung kann man immernoch hinzufügen.

Dieses Mesh ist in unserer Group des ersten Wagons enthalten. Dort müssen wir jetzt die Animation hinzufügen.
Hier fügen wir zwischen children und Mat Configs folgendes ein:

Code

```
events = {
        },
        },
        [21] = { forward = true, name =
        },
        [21] = { forward = false, name =
        },
    },
```

Alles anzeigen

Wir haben erstmal nur ein Objekt der rechten Seite. Wir zählen (bei den children), welche Position dieses Objekt (unser Türmesh) in der Group hat. Das erste Objekt hat die Id 1, unser Objekt ist zum Beispiel das 21. in der Liste.

Wir öffnen die Türe vorwärts forward=true mit der Animation namens "opendoor" und schließen genau rückwärts mit forward = false die gleiche Animation "opendoor".

Nun gehen wir in die Mdl und fügen dort folgendes zwischen children und MatConfigs ein:

Code

```
events = {
        },
        },
        [1] = { forward = true, name = "open_doors_ri
        },
        [1] = { forward = true, name = "close_doors_rig
        },
    },
```

Alles anzeigen

Nummer der Group zählen = 1 (wir haben nur eine Wagongroup verlinkt). In der Group hieß die Animation "open_doors_right", diese spielen wir vorwärts ab. Genauso die Schließanimation "close_doors_right", welche wir ebenfalls vorwärts abspielen.
In der Group selbst wird dann ja gesagt, dass wir die selbe Animation rückwärts abspielen.
Alternative: Man könnte auch in der Group nur open_doors_right hinzufügen und dann im Model sagen, man möchte für close_doors_right einfach "open_doors_right" in der Group rückwärts abgespielt haben.

Jetzt kann man im Model Viewer die Animation testen. Wenn er abstürzt (animations end Fehler), ist meist die Id falsch.
Wenn alles funktioniert, kann man die restlichen Türobjekte programmieren. In den Meshs ändert sich bei der Animation meist nur das Vorzeichen.

13 Model: Lods, Passagiere, Kamerapositionen, Wendezugfeature

Zum Aufbau der Mdl findet man hier weitere Infos https://www.transportfever.com…ding:developerinfo:models

Nun zu den **Lods**. Wenn wir dies nicht bereits getan haben, erstellen wir nun für jede Group ein zweites Exemplar.
In Lod 2/dem zweiten Exemplar werden detailreiche Meshs gelöscht, die auf weitere Entferung eh nicht mehr gesehen werden, z.B. Haltestangen, Innenanzeigen, Fahrertisch, oder Sitze.

In der .mdl wird dann der Block bei lods für jedes Lod kopiert und der Name der ursprünglichen Group (Lod1) durch die Neue (2.Lod) ersetzt.
Ich mache meistens nur zwei Lods, da ich nicht viele Details habe, die man ausblenden muss/kann. Man kann auch 3 oder mehr machen.

Bei visibleFrom/To wird eingegeben, bei welcher Entfernung welches Lod sichtbar ist.
Ich verwende meist 0-250 und 250-2500.

Bei den axles muss man ebenfalls ein zweites Lod hinzufügen, sonst fährt der Zug nicht gescheid um die Kurve, wenn man von weiterer Entfernung schaut. (Beispieldatei am Ende des Kapitels)

Wichtig: Wenn man in der .mdl im zweiten Lod ebenfalls die Animationen verlinkt hat, müssen diese auch in den zweiten Groups eingetragen sind. Hier ändern sich die Ids, weil man weniger Objekte hat. Diese müssen neu gezählt werden!

Passagiere werden so angegeben:

Code

```
seats = {
oRad(000,0,0,15),false,
        0(000,0,0,15),false,
        0(000,0,0,15),false,
        },
```

Hier kann man beliebig viele Einträge hinzufügen. Die Gruppe ist eine MeshId. Wenn der Passagier im ersten Wagon sitzt, sollte man hier ein Mesh angeben, welches sich in der Gruppe des ersten Wagons

befindet.
Die MeshId bekommt man mit dem TF ID Counter
Einfach damit die .mdl Datei öffnen und er zeigt alle Ids der Groups/Meshs an.
mit transf.scale... kann man die Passagiere skalieren, mit degtoRad rotieren, mit vec3.new platzieren (Positionen).
Standing = true, ob sie stehen sollen (sonst sitzen sie)
crew = true, ob es der Fahrer ist (sonst normaler Passagier

Wir müssen zudem das hier (Zeile 1 und 4 sind bereits vorhanden) am Anfang der Mdl angeben:

Code

```
function                                                                    data()
local              vec3              =              require              "vec3"
local             transf             =             require             "transf"
return {
```

Hier ein extra Lexikoneintrag: https://www.transportfever.net…mitglieder-in-Fahrzeugen/

Kamerapositionen lassen sich hiermit setzen

Code

```
cameraConfig                                =                                {

                {

oRad(0.0,0,0),vec3.new(16,2,3)),

                        },
                {

ec3.new(18,5,4)),

                        },
                      },
                },
```

Alles anzeigen

Group gibt die MeshId/GroupId an (Id Counter).
fov ist das field of view - Je größer, desto größer der Bildausschnitt bis hin zum Fischauge.
Rotation und Position der Kamera wie bisher in den hinteren Klammern.

Nun das **Wendezugfeature**.
Hier gibt es einen Wikipediaeintrag https://www.transportfever.com…models:reversiblevehicles

Normalerweise werden Züge am Kopfbahnhof umgedreht und wechseln nicht einfach die Fahrtrichtung.
Mit dem Wendezugfeature lässt sich das ändern.

Dazu wird in jeder .mdl die einen Eintrag bei engine hat reversible = true, gesetzt. Am Besten bei allen reinschreiben.

Außerdem kann man wechselnde Zugzielanzeigen, oder sogar verschiedene Pantographen einstellen.
Statt einen vorderen Wagon mit weißen Leuchten und einen hinteren Wagon mit roten Leuchten zu konfigurieren, verwenden wir nun einen universal einsetzbaren Wagon.
Hierbei weiße und rote Leuchten, genauso wie Zugzielanzeigen mit allen Zielen eingefügt. Ohne zusätzliche Einstellungen wäre alles gleichzeitig sichtbar.
Mit ihnen kann man bestimmen, welche Objekte ausgeblendet und zum richtigen Zeitpunkt wieder eingeblendet werden. Eine Erklärung befindet sich im verlinkten Wikipediaeintrag.
Als Id verwendet man wieder die Ids aus dem bereits verlinkten Id Counter/Viewer.

Hier die **fertige Mdl als Beispiel**. Mdl_Beispiel_Fertig.zip
Das Soundset ist hier bereits eingetragen und wird im Folgenden erklärt.

14 Sound

https://www.transportfever.net/lexicon/index.php?entry/86-sound-set/

Der Sound bringt nochmal den letzten Schliff und verleiht dem Fahrzeug einen eigenen Charakter.
Dazu benötigt man Audiodateien und eine Sound Konfigurationsdatei.

Die Konfigurationsdatei (.lua Datei) kommt in config - sound_set.
Den Namen verlinken wir dann in unserer .mdl Datei (im Beispiel bereits geschehen).

Bevor wir die Sounddatei konfigurieren, benötigen wir die Audiodateien.
Im Idealfall nimmt man diese selbst auf.
Auf Youtube finden sich ebenfalls Videos, bei denen man Soundaufnahmen mit Erlaubnis des Autors verwenden kann.
Wenn alles nichts hilft, kann man auch einfach die Standarddateien aus dem Spiel benutzen.

Man kann folgende Sounddateien einbinden:

Einmalige Sounds:
1. Türe auf
2. Türe zu
3. Horn

Wiederholbare Sounds. Dabei wird ein kurzer Abschnitt in Endlosschleife wiederholt.
Die Schwierigkeit ist es, einen Bereich zu finden, der beim Übergang von Ende zum Anfang gut klingt und keine Ruckler oder Störgeräusche besitzt. Das bedeutet, dass man keine Beschleunigungen in einem Sound haben kann.
Wir müssen unsere Beschleunigung daher in mehrere konstante Tonhöhen splitten und diese dann möglichst schön kombinieren, sodass ein gleichmäßiger Beschleunigungssound entsteht. Hier ein Tutorial wie man in Audacity gute Schleifenübergänge macht.
Diese Dateien werden verwendet für:

1. Standgeräusche
2. Fahrgeräusche
3. Beschleunigungsgeräusche

Für die Erstellung solcher Audiodateien verwende ich Audacity - ebenfalls kostenlos.
Alle Dateien dürfen nur eine Monospur besitzen. Dazu kann man in Audacity Spuren - Stereospur in Mono umwandeln verwenden.
Bei Transport - Schleifenwiedergabe kann man die wiederholbaren Sounds probehören.

In der Sounddatei werden nun die Pfade für die Audiodateien angegeben. Events wie Türe auf, Zu und Horn können einfach mit Lautstärke und hörbare Distanz angegeben werden.
Für die wiederholbaren Sounds wird je ein Track erstellt. Dort gibt man die Lautstärke sowie den Pitch abhängig von Geschwindigkeit oder Leistung an.

Hinweise:
Man kann das Horn auch weglassen. Dann nervt es nicht.
Damit das Türpiepen überall gleichmäßig zu hören ist, könnte man es in jedem Wagon abspielen. Dann überlagern sich jedoch die Töne. Ich habe deswegen eine DummyMdl genau in der Mitte des Triebwagens platziert, welche aus einem transparenten Rechteck besteht und nur Sound abspielt.
Die standardmäßigen Clacks und Bremsenquietschen kann man ebenfalls hinzufügen.
Wenn man in Abhängigkeit der Leistung konfiguriert: Leistung ist hier nur positive Leistung, also Beschleunigung. Bremsen ist 0 und führt daher zu keinem einstellbaren Sound.

Hier ein kommentiertes Beispiel der Avenio, hier ist alles beisammen: Soundconfig.zip

15 Mod veröffentlichen

https://www.transportfever.net…ion-ver%C3%B6ffentlichen/
Für den Steam Upload ist dies eine gute Anleitung.

Hier im Forum benötigt man lediglich die gepackte Version des Mods hochzuladen.
Nach der ganzen Arbeit wäre es nun verschenkte Zeit, wenn wir an der Beschreibung und den Bildern sparen.

Mit dem Model Viewer einige Screenshots aufnehmen und im Eintrag einfügen. Ingame ist mit dem Kameratool auch einiges möglich.
Manche Youtuber machen auch häufiger Modvorstellungen.

Ich hoffe diese Anleitung ist hilfreich und konnte zum Verständnis von Transport Fever Mods beitragen und eine Anleitung für neue Modder bieten.
Über Feedback, Verbesserungsvorschläge, Anmerkungen zum Verständnis etc. würde ich mich sehr freuen, damit dieser Leitfaden stetig verbessert, sowie aktuell gehalten werden kann und die viele Arbeit nicht umsonst war.

Vielen Dank fürs Lesen.
Grüße Marc