Mod Portierung TpF -> TF

!WIP!

Dieser Lexikoneintrag befasst sich erstmal nur mit der Portierung von TpF Fahrzeugen nach TF.
Wenn jemand diesen vervollständigen möchte, darf dieser Eintrag gerne erweitert werden.

Allgemeines:
Die Struktur der meisten Dateien einer TpF basieren auf der bereits in TF verwendeten Struktur und wurden teilweise um weitere Attribute ergänzt. Diese zusätzlichen Attribute sind kein Problem für TF, da sie einfach ignoriert werden.
Problematisch wird es immer dann, wenn ein Modder auf TpF Dateien zugreift, die in TF noch nicht existierten, denn dann sind für eine Mod nicht alle Abhängigkeiten vorhanden und das Spiel wird dies einfach mit einer Fehlermeldung abstürzen.
Weiter ist es natürlich nicht möglich Kampagnen, Schiffe, Flugzeuge und Constructions funktionsfähig nach TF zu portieren. Eine Ausnahme bilden hier Assets, einfache Haltestellen und Depots. Diese werden allerdings in TpF in einer komplett neuen Struktur definiert.
Die Audio Dateien können grundsätzlich unverändert übernommen werden.
Bei den Texturen muss man Glück haben: verwendet die Mod tga-Texturen, ist alles gut. Werden hingegen dds-Texturen verwendet, müssen diese erst z.B. mit GIMP und dem dds Plugin in tga konvertiert werden.

Ich werde im Folgenden des öfteren beschreiben müssen welche Informationen an welche Stelle zu kopieren sind. Dazu verwende ich die für Lua übliche dot Notation. Das heißt, wenn ich z.B. schreibe "events.clacks.names muss nach clacks kopiert werden", heißt das, dass aus diesem Code

Code

```
events                                  =                                       {
  clacks                                  =                                       {
   names                                   =                                       {
    "vehicle/StadlerGiruno/clack.wav",
    "vehicle/StadlerGiruno/clack.wav"
   },
   refDist                                 =                                    15.0
  }
```

dieser hier wird

Code

```
clacks                                          =                                          {
 "vehicle/StadlerGiruno/clack.wav",
 "vehicle/StadlerGiruno/clack.wav"
}
```

Fahrzeugmods:
Ich kann natürlich nicht garantieren, dass das Vorgehen für alle TpF Fahrzeugmods funktioniert. Mit den wenigen getesteten Fahrzeugen hat es aber funktioniert.
Eine Fahrzeugmod besteht sowohl in TF als auch in TpF aus Texturen, audiodateien, configs um multiple units zusammenzubauen bzw. um zu beschreiben wie der Sound der Fahrzeuge verwendet wird und dem model an sich, welches wiederum aus aus der Form (meshes), Materialien (material), Gruppen von Formen (group) und dem Modell an sich (model), welches diese ganzen Sachen zu einem Fahrzeug zusammenklebt.

Nicht (direkt) kompatible Dateien:
In erster Linie sind alle Dateien die auf Dateien verweisen die in TF noch nicht existierten problematisch, denn diese führen dazu, dass das Spiel abstürzt.
Dazu gehören fast immer die Dateien in config/sound_set, denn diese benötigen das Script soundeffectsutil, welches in TF nicht existiert. Außerdem hat sich die Struktur der Datei stark verändert.

Lösungsvariante 1 "ganz einfach":
Man ersetzt den Inhalt dieser Datei durch die Minimal von TF benötigte Config:

Code

```
function                                                                    data()

return{
                                                                  type
 params                                          =                                          {
  tracks                                          =                                          {
  }
 }
}
end
```

Alles anzeigen

Schon bringt diese Datei das Spiel nicht mehr zum Absturz. Allerdings gibt man TF auch keinerlei Informationen über die Sounds des Fahrzeugs mit.

Lösungsvariante 2 "etwas besser":
Man verwendet wieder die Minimalconfig aus Variante 1 und ergänzt diese ein wenig:
Erstmal ergänzt man die Einträge refDist und refWeight und wählt jeweils "sinnvolle" Werte. Viele TF Mods setzen refDist=25 und refWeight=40
Dann ergänzt man noch clacks, horn openDoors, closeDoors, indem man events.clacks.names nach clacks, events.horn.names nach horn, events.openDoors.names nach openDoors und events.closeDoors .names nach closeDoors kopiert.
Die resultierende Datei sieht dann z.B. (ich verwende den Stadler Giruno von Seamon als Beispiel) so aus:

Code

```
function                                                                data()


return{
                                                                  type
 refDist                                      =                                  25.0,
 refWeight                                     =                                 40.0,
 horn                          =                    "vehicle/StadlerGiruno/horn.wav",
 openDoors                        =                 "vehicle/StadlerGiruno/open.wav",
 closeDoors                       =                "vehicle/StadlerGiruno/close.wav",
 clacks                                        =                                 {
  "vehicle/StadlerGiruno/clack.wav",
  "vehicle/StadlerGiruno/clack.wav"
 },
 params                                        =                                 {
  tracks                                        =                                 {
  }
 }
}
end
```

Alles anzeigen

Den tracks Block und die updateFn von TpF kann man nicht nach einem festen Schema nach TF rückportieren. Dazu mehr in Variante 3.

Lösungsvariante 3 "verstehen und selbst machen":
Hier findet man eien recht guten Wiki Eintrag zum Thema sound_set in TF.
Man kann sich an Variante 2 orientieren und mit den Informationen aus dem Wiki Eintrag noch den tracks/updateFn Teil so weit dies von TF unterstützt wird nach TF übersetzen. Das setzt aber vorraus, dass man versteht wie dieser Tracks Block und die updateFn in TpF funktionieren. Dazu gibt es noch keinen Wiki Eintrag und dies hier zu behandeln würde den Rahmen sprengen.
Dieser Wiki Eintrag beschreibt wie man diese sound_sets von TF nach TpF konvertiert und erklärt dabei zumindest ein wenig die Struktur dieser Dateien in TpF.

Wenn man eine dieser varianten durchgeführt hat, kann man meistens das Spiel mit der eben (teilweise) portierten Mod laden ohne, dass es abstürzt und wird das Fahrzeug auch im Depot sehen.
Es ist allerdings noch nicht garantiert, dass man das Fehrzeug kaufen kann ohne dass das Spiel abstürzt. Häufig verweisen Mods noch auf weitere Dateien die TF noch nicht hat.

Weitere Crash-Ursachen:
--fehlende Dateien, meistens materials

--Schönheitsfehler
Die mod.lua