Repaints

Manches Mal fehlt einem genau ein Anstrich eines Modells im Spiel zum Glück. Man kann sich nun vertrauensvoll an einen Modder wenden und darauf hoffen und warten, daß der einem diesen Anstrich liefert.

Oder man macht es einfach selbst. Das ist in TPF 2 überraschend leicht und problemlos möglich.

Und so wird´s gemacht:

1. Etikette und guter Stil

Beides ist elementar und wird natürlich auch bei Repaints gewahrt. Wenn man ein Modell, das man nicht selbst gebaut hat, nur für sich selbst neu anstreichen möchte, ist das sicher kein Problem. Aber sobald man das Modell mit dem neuen Anstrich irgendwo hochladen möchte, egal, ob auf Transportfever.net, Steam, oder wo auch immer, ist das ausdrückliche Einverständnis des Erstellers des Ursprungsmodells vorher einzuholen. Das ist nicht nur eine Frage des Stils, sondern auch eine des Copyrights: das Modell gehört immer noch dem, der es erstellt hat; das Copyright des Repainters beschränkt sich auf die von ihm neu erstellten Dateien.

2. Ein paar Grundlagen zum Anfang.

Ein Modell besteht aus einer Anzahl von verschiedenen Einzelteilen, den Meshes, die man an der Dateierweiterung ".msh" erkennt. Ich werde sie in diesem Text daher als .msh bezeichnen. Die optischen Eigenschaften eines .msh werden ihm in einem Material zugewiesen, das die Erweiterung ".mtl" trägt und daher hier auch so heißen soll. Eine Modelldatei mit der Erweiterung ".mdl" führt die Dateien alle zusammen und bewirkt, daß sie zu einem Modell werden. In einer .mdl wird also aus .msh und .mtl ein Modell !) So weit, so einfach.

Image not found or type unknow

Um ein Modell also für einen Neuanstrich zu duplizieren, müssen wir:

- eine neue .mdl erschaffen,

- in dieser neue .mtl anfordern,

- und zum Schluß neue Texturdateien auf dieses Duplikat legen.

Das ist ohne große Probleme möglich, da es nur minimale Änderungen am Originalmodell erfordert. Der Zeitaufwand für die Vorbereitung ist unter 5 Minuten; derjenige für den Anstrich selbst allerdings nicht zu definieren.

Und wie sieht das jetzt in den Dateien aus?

Wenn wir einmal eine .mdl öffnen, so finden wir darin ganz zu Anfang zahlreiche Blöcke, die wie dieser hier aussehen:

Code

```
                    {
                                                                materials   =   {   "vehicl


                transf = { 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, },
                  },
```

In diesen Blöcken können noch einige andere Dinge mit drinstehen, die sind aber für einen Repaint nicht von Bedeutung. Wichtig ist hier, daß dem .msh "vehicle/plane/Tu154B/Tu154B_Rumpf.msh" das .mtl "vehicle/plane/Tu-154B CU.mtl" zugewiesen wird. Möglich ist, daß einem .msh auch mehrere .mtls zugeteilt werden; in dem Fall ist das .msh in einzelne Segmente unterteilt wie zum Beispiel eine Tür, die ein durchsichtiges Fenster enthält. Die anderen Teile einer .mdl sind für einen Repaint ebenfalls nicht wichtig.

Schauen wir uns jetzt einmal eine .mtl an. Die oben angeforderte Tu-154B CU.mtl sieht so aus:

Code

```
function data()
return {

        },

            },

            },

        },

        },

                                                                f

        },

                                                fileName    =    "models/vehicle/plane

        },

        },

        },

                                                                fileName    =    "models/vehicl
```

Alles anzeigen

Auch hier sind wieder viele Blöcke enthalten, die jeder eine einzelne Texturdatei anfordern, die das Aussehen des Materials alle zusammen definieren. Nicht jeder Block ist in jeder .mtl enthalten; in der für Repainter relevanten .mtl ist jedoch immer ein Block "map_albedo" und "map_metal_gloss_ao", meist auch ein Block "map_cblend" (evtl mit _dirt_rust) enthalten. Für eine wunderschöne Erklärung der verschiedenen Texturarten empfehle ich SkyJoes hervorragende Lexikonartikel - das brauche ich hier nicht nochmal zu schreiben.

Interessant sind wie gesagt je nach Modell und Anstrichwunsch auf jeden Fall die "map_albedo", die den farbigen Anstrich definiert, wahrscheinlich auch die "map_metal_gloss_ao" (Metalloberfläche, Glanz und Schattenwurf in finsteren Winkeln des Modells) und eventuell die "map_cblend", die immer die wählbaren Anstrichbereiche und meist auch die Dreck- und Rostebenen beinhaltet. Für genaue Erklärungen dieser Dateien darf ich wieder an Skyjoe verweisen.

3. Vorbereitung eines Modells zum Neuanstrich

Um also ein vorhandenes Modell neu anzustreichen, benötigt man eine neue .mdl, in der die neu anzustreichenden Teile eine neue .mtl zugeteilt bekommen. Das ist mit wenigen Handgriffen zu erledigen.

a. Vorbereitung der .mdl

Öffnet eine .mdl des gewünschten Modells. Zuerst muß man herausfinden, welches Material auf den neu anzustreichenden Teilen liegt. Das ergibt sich meistens logisch aus den Namen der .msh - wenn eine .msh "Rumpf", "Kessel" oder ähnliche Worte im Namen trägt, ist die Wahrscheinlichkeit hoch, daß das der korrekte Kandidat ist.

Schaut euch an, wie die .mtl dazu heißt. Nehmen wir mal an, sie heiße "Altes Material.mtl". Nun kann man einfach mit "Suchen und Ersetzen" alle vorkommenden "Altes Material.mtl" durch eine noch zu erstellende neue Materialdatei ersetzen lassen. Für diese Datei sollte man sich einen sinnvollen neuen Namen überlegen, der im Spiel einmalig ist: wenn das Spiel gleich benannte Dateien findet, dann führt das zu nichts Gutem. Ganz schlecht wäre der Name "Repaint.mtl", gut wäre ein Name wie (Repainter)_(Modell)_(Anstrich).mtl - in unserem Fall, wenn ich die Tu-154 neu anstreichen wollte, wäre es also etwa "DH106_Tu154_Lufthansa.mtl". Danach wird die .mdl **unter einem neuen Namen!** im selben Verzeichnis gespeichert. Solange die neue .mtl nicht existiert, führt das Öffnen der neuen .mdl zum CTD - also machen wir gleich weiter.

b. Vorbereitung der .mtl

Öffnet die noch vorhandene Datei "Altes Material.mtl". Hier sucht ihr euch die in "map_albedo" und bei Bedarf auch in anderen Blöcken verlangten Bilddateien (.tga oder .dds) heraus und ersetzt deren Namen durch neue, frei gewählte, die ihr für euren Repaint verwenden werdet. Speichert sie dann unter dem Namen, den ihr oben in der .mdl eingetragen habt, also in diesem Beispiel "Repaintmaterial.mtl". **Wichtig** ist, daß ihr

euch hier einen Namen aussucht, der im Spiel einmalig ist. Sinnvoll ist zum Beispiel ein Name wie "(Ersteller)_(Modell)_(Version).mtl". Für die hier zitierte Tu-154 würde also zum Beispiel passen "DH106_Tu154_Lufthansa.mtl". Wenn man einen Namen wählt, den es im Spiel schon gibt, dann wird das Spiel je nach Ladereihenfolge das schon vorhandene Material entweder durch eures ersetzen oder anstelle eures neuen das schon vorhandene nehmen. Beides sieht bescheiden aus und ist leicht vermieden, wenn man diese einfache Regel einhält.

Es ist hier nicht nötig, daß die neuen Bilddateien schon vorhanden sind. Wenn der Modelviewer oder das Spiel eine Texturdatei vergeblich sucht, dann wird sie halt nicht verwendet; typisch ist, daß das Modell dann violett ist, extrem schnell altert oder sonstwie eigenartig aussieht. Aber einen CTD gibt es dadurch nicht.

Jetzt ist ein neues Modell als leere Leinwand zum Anstrich bereit. Schwingt den Pinsel, erstellt die in der .mtl verlangten Texturdateien.

4. Integration des Modells im Spiel

Sobald eine .mdl und die in ihr verlangten .mtls fertig sind, kann man das Modell im Spiel schon benutzen. Aber im Kaufmenü gibt es noch kein Vorschaubild, sondern die violetten Quadrate. Um diese Bilder zu bekommen, bietet sich der Model Editor an, den man hier herunterladen kann (nur für GOG-Nutzer, bei Steam wird er mitgeliefert). Mit einem Klick auf "Screenshot" werden automatisch die richtigen Bilder am richtigen Ort gespeichert.

Wenn man auch die Beschreibungszeile im Menü angepaßt haben möchte, dann geht das über die .mdl. Darin steht beinahe ganz am Ende ein Block:

Code

```
description                                    =                                    {
                                                                description     =     _("Dreistra
                                                                                                   n
        },
```

..., der selbsterklärend sein sollte und leicht korrigiert ist.

Und danach sollte es funktionieren - viel Spaß mit dem neuen Repaint ;)

Image not found or type unknown