ModUtil: Ein Repaint mit der Repaint Funktion erstellen

Inhaltsverzeichnis

- 1 Einleitung
- 2 mod.lua
- 3 .mdl

1 Einleitung

Die Repaint Funktionen des ModUtil Scripts haben sowohl das Ziel, Repaints zu vereinfachen, als auch Dateien und damit auch Speicherplatz zu sparen (auf der Festplatte und im Arbeits-/Grafikspeicher). Man braucht daher nur noch folgende Dateien:

- das ModUtil Script in "res/scripts"
- die mod.lua
- die geänderten Materials mit den dazugehörigen Texturen (brauchen immer noch einen eigenen Namen, damit sie nicht die original Texturen überschreiben)
- .mdls für die Repaint Objekte, bei Fahrzeugen inklusive der UI-Icons

Allerdings müssen Repaints dann etwas anders erstellt werden als bei anderen Objekten. Dieses Tutorial soll die nötigen Änderungen vorstellen.
[info=info]Wenn beim original Mod nichts gegenteiliges steht, sind Repaints mit dem ModUtil Script immer erlaubt, solange folgende Bedingungen beachtet werden:

- Es werden keine .msh und .grp Dateien mitgeliefert (ist dann auch gar nicht mehr nötig)
- Im Download wird auf das original Objekt verlinkt und der Autor erwähnt.

[/info]

2 mod.lua

Hier muss zunächst das ModUtil Script eingebunden werden (sofern es nicht wegen der Einstellungen schon eingebunden ist):

```
local modUtil = require "merk_modutil_1"
```

Ansonsten muss in der runFn nur noch eine Funktion aufgerufen werden, die dem Script mitteilt, welche Materials des Repaints welchen Materials des original Objekts entsprechen:
Code

```
modUtil.vehicles.addRepaint("mod_id",                "repaint_id",                {
 {"vehicle/waggon/4yg.mtl",                           "vehicle/waggon/4yg_64.mtl"},
 {"vehicle/waggon/4yg_2.mtl",                       "vehicle/waggon/4yg_64_2.mtl"},
 {"vehicle/waggon/4yg_transparent.mtl",  "vehicle/waggon/4yg_64_transparent.mtl"},
})
```

"mod_id" ist die ID der original Mod (steht in der entsprechenden mod.lua) und "repaint_id" die ID für das Repaint (die Kombination mod_id + repaint_id sollte eindeutig sein). Danach folgt noch eine Liste mit Materials, der erste Eintrag in jeder Zeile ist das Material des original Objekts, der zweite das entsprechende

Material für euer Repaint.
In Beispiel oben bedeutet das, dass jedes Mesh mit dem Material "vehicle/waggon/4yg.mtl" auch noch das Material "vehicle/waggon/4yg_64.mtl" zugewiesen bekommt, jedes Mesh mit "vehicle/waggon/4yg_2.mtl" auch "vehicle/waggon/4yg_64_2.mtl", und so weiter.

Eine vollständige mod.lua sieht dann z.B. so aus (wie der "info" Teil aussehen muss, könnt ihr hier nachlesen: mod.lua und strings.lua - Erklärung und Inhalt): Image not found or type unknown

Code: mod.lua

```
local modUtil = require "merk_modutil_1"

-- optional: Einstellungen
local settings_def = {


    },
}
-- Einstellungen Ende

function data()
 return {
  info = {
   -- ...
  },
  settings = settings_def, -- nur wenn Einstellungen genutzt werden
  runFn = function(settings)
   modUtil.vehicles.addRepaint("mod_id", "repaint_id", {
    {"vehicle/waggon/4yg.mtl", "vehicle/waggon/4yg_64.mtl"},
    {"vehicle/waggon/4yg_2.mtl", "vehicle/waggon/4yg_64_2.mtl"},
    {"vehicle/waggon/4yg_transparent.mtl", "vehicle/waggon/4yg_64_transparent.mtl"},
   })
   modUtil.userSettings.create("merk_4yg_64", settings_def) -- nur wenn Einstellungen genutzt werden
  end
 }
end
```

Alles anzeigen

3 .mdl

Es ist sinnvoll die .mdl des original Objektes als Ausgangspunkt zu nutzen, diese einfach in den eigenen Mod Ordner kopieren und umbenennen. Danach den "metadata" Teil anpassen (insbesondere"name" und "description") und für ein einfaches Repaint auch nur diesen.
Die Datei sollte dann immer noch folgende Struktur haben:
Code

```
function data()
return {...}
end
```

die noch wie folgt geändert werden muss:

Code

```
local                       result                       =                       {...}


if          merk_vehicleUtil          and          merk_vehicleUtil[1]          then
 merk_vehicleUtil[1].getRepaint("mod_id",           "repaint_id",           result)
end


function                                                                    data()
 return                                                                     result
end
```

Alles anzeigen

Der Teil `{...}` ändert sich dabei nicht und in Zeile 4 müssen "mod_id" und "repaint_id" den gleichen Wert haben wie in der mod.lua.
Alle anderen Dateien bleiben so, wie sie auch bei anderen Repaints sind.