Eigene Dateilisten und Zusatzmods für den Dummy Mod erstellen

Inhaltsverzeichnis

- 1 Dateilisten
- 2 Zusatzmods

1 Dateilisten

Wenn ihr eine Fehlermeldung bekommt, dass das Spiel beim Spielstand laden eine bestimmte Datei nicht finden kann und es für die Datei nicht schon einen Zusatzmod gibt, ist die einfachste Lösung eine eigene Dateiliste.
Diese Liste muss in einer Textdatei mit dem Namen `userList.lua` gespeichert werden, entweder direkt im Hauptverzeichnis des Mods (wo sich auch die mod.lua befindet) oder im Unterordner `res/scripts` (es ist auch möglich an beiden Orten unterscheidliche Dateien zu nutzen, wenn man unbedingt möchte). Der Inhalt der Datei sollte dann wie folgt aussehen:
Code: userList.lua

```
local modelList = {
 {
  fileName = "res/models/model/vehicle/train/br_120_0.mdl",
  dummyModel = {name = "trainEngine", length = 19.2, seats = 0}
 },
}
return modelList
```

Die Zeilen 1 und 6-7 solltet ihr nur ändern, wenn ihr wisst was sie bedeuten. Der Block aus den Zeilen 2-5 kann dagegen für beliebig viele Dateien wiederholt werden (wodurch sich natürlich die Zeilen 6 und 7 nach hinten verschieben).

Zeile 3 gibt den Dateinamen der fehlenden Datei an, ihr könnt ihn aus der Fehlermeldung ablesen, die das Spiel ausgibt:
[box]File: res/models/model/vehicle/train/br_120_0.mdl

cannot open res/models/model/vehicle/train/br_120_0.mdl: No such file or directory

This error is usually caused by modding. The Syntax of some game resources is not correct.[/box]
oder, nachdem das Spiel beendet wurde, am Ende der stdout.txt:
[box]`error in file res/models/model/vehicle/train/br_120_0.mdl: cannot open res/models/model/vehicle/train/br_120_0.mdl: No such file or directory`[/box]
Hier ist noch zu beachten, dass die Ordnerstruktur bis zur Datei im Mod vorhanden sein muss, da es nicht möglich ist Ordner über ein Lua-Script zu erstellen. In diesem Beispiel müsste im Ordner des Mods der

Unterordner `res` vorhanden sein, dort ein Unterordner `models` und so weiter bis zum Ordner `train` (diese speziellen Ordner sind im Mod schon enthalten, aber bei "exotischeren" Objekten kann es sein, dass manche Ordner fehlen).

Zeile 4 liefert Informationen über die Art der fehlenden Datei, diese sind leider nicht so einfach herauszufinden (da die Datei ja nicht mehr exisitert), aber leider auch nötig, damit die Objekte im Spiel richtig funktionieren.

`name` gibt an welches Objekt für diese Datei benutzt werden soll, folgende Werte sind möglich (alle anderen werden einfach ignoriert):

trainEngine	Loks
trainWagonPerson	Personenwagen
trainWagonCargo	Güterwagen
tram	Trams
tramCargo	Gütertrams
bus	Busse
truck	LKWs
ship	Passagierschiffe
shipCargo	Frachtschiffe
person	Personen
car	KI Autos
misc	Sonstige (Signale, Depots, Assets, etc.)
construction	Konstruktionen

`length` gibt die Länge des Modells an. Dieser Eintrag ist optional und nur bei Zügen relevant, dort können nämlich auch zu kurze oder zu lange Modelle zu einem Absturz führen. Es ist daher empfehlenswert hier möglichst genaue Werte einzutragen. Mir ist allerdings klar, dass es schwierig ist diese herauszufinden. Wenn man weiß um welches Fahrzeug es sich handelt, könnte man sich zum Beispiel am Vorbild orientieren (sofern eines existiert).

Mit `seats` kann man die Menge der Sitzplätz für Passagiere und Crew bestimmen. Auch dieser Eintrag ist optional und sollte nur genutzt werden, wenn man den genauen Wert kennt oder der Standardwert zu klein ist.

Werden sowohl `length` als auch `seats` nicht benötigt, kann die Zeile auch zu `dummyModel = "<name>"` (z.B. `dummyModel = "bus"` oder `dummyModel = "shipCargo"`) vereinfacht werden.

Sollte die Fehlermeldung immer noch erscheinen, nachdem ihr die Datei gespeichert und den Mod zu eurem Spielstand hinzugefügt habt, überprüft am besten nochmal ob sich die `userList.lua` im richtigen Ordner befindet und die Ordnerstruktur für die zu erstellenden Dateien vorhanden ist.

2 Zusatzmods

Ihr habt noch ein Backup eines Mods der gelöscht wurde und wollt anderen Leuten ohne Urheberrechtsverletzung die nötigen Dateien bereitstellen, damit sie ihre Spielstände wieterhin nutzen können? Dann ist ein eigener Zusatzmod eine Option.
Der Zusatzmod ist erstmal aufgebaut wie jeder andere Mod, man braucht also einen Ordner dessen Name folgendes Format hat: `<modname>_1` <modname> darf weitere Unterstriche enthalten, ihr könnt euch an den vorhandenen Zusatzmods orientieren, also `dummies_<Name des Originalsmods>_1` oder einen eigenen Namen verwenden.
In diesem Ordner muss eine mod.lua und die Ordnerstruktur zu allen Dateien, die erstellt werden sollen, vorhanden sein.

Der Inhalt der mod.lua sollte wie folgt aussehen:

Code: mod.lua

```
local modelList = {
 {
  fileName = "<dateiname>",
  possiblePaths = {
   "mods/<mod_ordner>/",
   "../../workshop/content/446800/<workshop_id>/"
  },
  dummyModel = {name = "misc", length = 0, seats = 0},
 },
}
local info = debug.getinfo(1,'S')
local modPath = string.sub(info.source, 2, string.len(info.source)-7)
function data()
 return {
  info = {},
  runFn = function (settings)
   if dummyModels then
    dummyModels.util.createModels(modPath, modelList)
   else
    print("Couldnotcreatedummymodels.Itseemsthemainmodismissingornotloadedatthispoint.")
   end
  end
 }
end
```

Alles anzeigen

Der Block aus Zeile 2-9 muss für jede Datei wiederholt werden, `fileName` und `dummyModel` haben die gleiche Funktion wie bei den eigenen Dateilisten, neu ist hier `possiblePaths`. Dort werden alle Verzeichnisse eingetragen, in denen die Datei vorhanden sein könnte, damit der Dummy Mod weiß wo er nach der Datei suchen muss, um dann zu enscheiden ob sie erstellt werden muss oder nicht. Zeile 5 zeigt ein Beispiel für einen Mod aus dem Mod-Ordner, Zeile 6 für einen Mod aus dem Workshop. Es müssen natürlich nicht immer diese zwei Einträge vorhanden sein, mehr oder weniger Einträge sind ebenfalls möglich.
Der Eintrag info in Zeile 15 entspricht größtenteils noch dem Inhalt der info.lua aus Train Fever, daher verweise ich dafür auf folgenen Artikel: info.lua

Eigene Zusatzmods können gerne veröffentlicht werden, wenn die folgenden Bedingungen eingehalten werden:

Spoiler anzeigen