mod.lua und strings.lua - Erklärung und Inhalt

Inhaltsverzeichnis

- 1 Offizielles Mod Format von Urban Games (mod.lua)
- 2 Zusätzliche Parameter für Mod Manager (TPFMM)
- 3 Zusätzliche Parameter für CommonAPI
- 4 Übersetzungen (strings.lua)
- 5 Beispiel Dateien

[Blockierte Grafik: https://ftp.train-fever.net/flaggen/gb.png] The English version can be found here: mod.lua and strings.lua - Explaining the content and requirements

1 Offizielles Mod Format von Urban Games (mod.lua)

Die Basis für die mod.lua, welche für jeden Mod benötigt wird und im Hauptverzeichnis des Mods liegen muss:
Code: mod.lua

```
function data()
return {
	info = {
		name = _("Sample mod"),
		description = _("modDesc"),
		minorVersion = 0,
		severityAdd = "WARNING",
		severityRemove = "NONE",
		visible = true,
		-- optional,
	},
	-- options
	-- runFn
	-- checkActiveFn
}
end
```

Alles anzeigen

2 Zusätzliche Parameter für Mod Manager (TPFMM)

Die folgenden Parameter *können* angegeben werden, um erweiterte Informationen zur Verfügung zu stellen, zB um eine Webseite oder den Ersteller der Mod zu verknüpfen.
Basisinformationen
Code

```
		tfnetId = 0,
		steamId = 0,
		url = "https://transportfever.net"  -- optional, URL zur Webseite des Mods
```

Autoren

Code

```
authors   =   {                                        --   optional,   Informationen   über   die
      {                                                                                --   Angaben   über
                                  name        =        "Author        1",
        role   =   "CREATOR",         --   CREATOR,   CO_CREATOR,   TESTER,   BASED_ON,   OTHER
       tfnetId   =   00000,          --   optional,   id   vom   Autor   auf   transportfever.net
                            steamId       =       0000,
    },                                          --   optional,   weiter   Autoren,   durch   Komma
},
```

Tags

Code

```
tags = { "tag1", "tag2", "tag3" },    --optional, Liste von Tags (strings)
```

Abhängigkeiten
Abhängigkeiten zu anderen Mods können angegeben werden. Das Spiel wertet diese Informationen nicht aus, aber Mod-Manager können damit sicher stellen, dass alle benötigten Mods installiert sind.
Um eine Abhängigkeit zu einem anderen Mod zu definieren, kann einfach der Ordnername benötigten Mods in die Liste aufgenommen werden:

Code

```
dependencies = { "other_mod_1", "another_mod_1", "something_else_1" },
```

3 Zusätzliche Parameter für CommonAPI

Es *können* erweiterte Informationen im info Bereich übergeben werden:

modid
Damit Mods auch aus einer Steam Quelle erkannt werden können , kann hier die originale modid übergeben werden. Bitte die Schreibweise beachten!

Code: mod.lua

```
modid = "beispiel_1"
```

requiredMods (ab Version 2019)
Eine Liste mit detaillierten Informationen zu Abhängigkeiten, im Gegensatz zu dependencies können hier zu jedem Mod optional modId, steamId (WorkshopID), tfnetId (Webdisk Eintrag), url angeben werden.
Es muss mindestens eine modId oder steamId angeben werden.
CommonAPI überprüft die Angabe und gibt eine Warnung in der stdout.txt aus, sollte die Mod nicht geladen sein. Darüber hinaus gibt CommonAPI automatisch Download URLs aus

Code: mod.lua

```
info                                        =                                        {
...
requiredMods    =    {    --    optional,    Informationen    über    geladene    Mods
       {                                                                                    --    Anga
                                            modId=          "example_1",
                  steamId     =     0000,                                 --     optional,     stea
      tfnetId  =  0000,          --  optional,  id  der  Webdisk  auf  transportfever.net
            url    =    "http://",                --    optional,    url    zum    Download    der
 minMinorVersion      =      0          --      optional,      minimale      minorVersion
    },                                                 --  optional,  weitere  Mods,  durch
},
...
```

Alles anzeigen

4 Übersetzungen (strings.lua)

Alle Strings, die der Übersetzungsfunktion _("") übergeben werden, werden mithilfe der strings.lua Datei übersetzt:
Code

```
function                                                                    data()

          ["my  mod"]  =  "Mein  Mod",    --    Alle  vorkommen  von  _("my  mod")  werden

                    -- falls die ausgewählte Sprache deutsch ist ("de")
                                                                                    ["the

   },
}
end
```

Alles anzeigen

[box]**Beachte:**
Wenn eine Variable an die _() Funktion übergeben wird, wird der *Inhalt* der Variable, nicht der Name der Variable selbst an die Funktion übergeben!
In Kombination mit globalen Variablennamen in LUA kann dies dazu führen, dass sich Variablen von unterschiedlichen Dateien oder sogar Mods gegenseitig überschreiben. In einem Beta-Patch von Transport

Fever konnte dieses Scenario beobachtet werden. Viele Mods nutzten die selbe Variable, um strings mit der `_( )` Funktion zu übersetzen - allerdings hat ein Mods bereits den Text in der Variable für alle anderen Mods überschrieben - so kann keine korrekte Übersetzung für den eigenen Mods gefunden werden.
Zwar wird dieses Verhalten in TPF mittleweile aktiv verhindert - allerdings rät Urban Games dazu, in Mods nur lokale Variablen (`local`) zu Benutzen.
Ich empfehle zusätzlich, immer mit Strings statt Variablen zu arbeiten.[/box]

Weitere Informationen zur strings.lua wurden hier von @Tom zusammen getragen.

5 Beispiel Dateien

Sehr einfaches Beispiel eines Mods mit einem Autor

Code: mod.lua

```
function data()
return {
    info = {

    },
    tags = {

    },
}
end
```

Alles anzeigen

Ein "Repaint"-Mod, welcher den Mod aus dem ersten Beispiel als Abhängigkeit benötigt

Code: mod.lua

```
function                                                                    data()
return                                                                          {
                                                                          info


      {




      },
      {


    },
  },
                                        dependencies            =           {
                                tags    =       {"repaint",         "train",
  },
}
end
```

Alles anzeigen

Code: strings.lua

```
function                                                                    data()


                                                                             ["name"]
                                                                          ["desc"]
   },

                                                                          ["name"]
                                                                    ["desc"]          =
   },
}
end
```

Alles anzeigen