TranslatorScript (1)

Inhaltsverzeichnis

- 1 Vorzüge & Kompatibilität
- 2 Vorbereitungen
  - 2.1 Einbinden von TranslatorScript
  - 2.2 Text richtig einbinden
  - 2.3 Codierung
- 3 Sprachdatei
  - 3.1 Speicherort
    - 3.1.1 Sprachdatei für gesamten Ordner
    - 3.1.2 Sprachdatei für einzelne *.mdl-Datei
  - 3.2 Syntax
  - 3.3 Codierung
- 4 Beispiel
- 5 Quellen & Links

[Blockierte Grafik: http://ftp.train-fever.net/flaggen/gb.png] Please find the english version here.

[Blockierte Grafik: http://ftp.train-fever.net/flaggen/de.png] Dieser kleine Script-Mod eröffnet die Möglichkeit, einen Mod nur durch hinzufügen einer Sprachdatei zu übersetzen.

[box]**Entwicklung wurde eingestellt!**
Durch die neuen Möglichkeiten des neuen Modding-Systems (strings.lua) wurde TranslatorScript überflüssig. Ich werde in den kommenden Tagen dieses Tutorial auf das neue System aktualisieren.[/box]

1 Vorzüge & Kompatibilität

Um ein Model zu übersetzen wird normalerweise für jede Sprache eine eigene *.mdl-Datei benötigt. So, du glaubst nun das wäre kein Problem? Dann stelle dir vor, du müsstest ein paar Zahlen in deinem Modell ändern. Du musst dies in jeder einzelnen Datei machen! Dies kann sehr komplex und zeitaufwendig werden.

Mit TranslatorScript muss nur eine einzige *.mdl-Datei für ein Modell bereitstellen. Die Sprachdaten sind in eine Sprachdatei ausgelagert. Der Benutzer installiert neben deinen Mod die Spache die er will. TranslatorScript übernimmt die gesamte Sprachintegration. Nur die Sprachdateien müssen von Hand gemacht werden.

Machst du dir sorgen, dass wenn TranslatorScript nicht bei deinen Benutzern installiert ist, sie deinen Mod nicht nutzen können? Weit gefehlt: Die Einbindung wird in diesem Fall einfach übersprungen. Dein Mod wird funktionieren.

[box]**Wenn der Nutzer TranslatorScript nicht installiert hat, wird dein Mod trotzdem funktionieren!**[/box]

2 Vorbereitungen

2.1 Einbinden von TranslatorScript

Füge den folgenden Code als erste Zeile in die *.mdl-Datei ein:
Code

```
pcall(require, 'BR146_translate_script')
```

Diese Zeile wird übersprungen, wenn TranslatorScript nicht installiert ist.

2.2 Text richtig einbinden

Jeder Text muss mit `_("DEIN TEXT")` umschlossen werden. Die wichtigen Zeichen sind der Unterstrich und die Klammern.
"DEIN TEXT" wird angezeigt, falls...

- keine Übersetzung installiert ist
- die Übersetzung für diesen Text fehlt
- TranslatorScript nicht installiert ist

2.3 Codierung

Die *.mdl-Datei muss als **UTF-8** codiert sein

3 Sprachdatei

3.1 Speicherort

[box]**Ein Nutzer kann pro Ordner bzw. *.mdl-Datei nur EINE Spachdatei installiert haben! Wenn keine Sprachdatei installiert ist, wird der Originaltext genutzt.**[/box]
Der Pfad zur Sprachdatei wird folgender Maßen gebildet:
Für ein Modell in `res\models\model\<Ordnername>`
lautet der Pfad
`res\strings\models\<Ordnername>`

3.1.1 Sprachdatei für gesamten Ordner

Für alle *.mdl-Dateien in einem gemeinsamen Ordner kann eine einzelne Sprachdatei erstellt werden. Der Name der Sprachdatei lautet dann `_TRANSLATION.lang`.

Beispiel:
Liegen die *.mdl-Dateien im Pfad `res\models\model\railroad\BR146`, so lautet der Pfad zur Sprachdatei:
`res\strings\models\railroad\BR146\_TRANSLATION.lang`.

3.1.2 Sprachdatei für einzelne *.mdl-Datei

Für einzelne *.mdl-Dateien - auch im selben Ordner - können einzelne, individuelle Sprachdateien erstellt werden. Der Name bildet sich dann aus dem Dateinamen der .mdl-Datei ohne Dateiendung (.mdl) und der Endung `.lang`.

Beispiel:
Lautet der Pfad zur *.mdl-Datei `res\models\model\railroad\yourModel.mdl`, so lautet der Pfad zur Sprachdatei:
`res\strings\models\railroad\yourModel.lang`.

3.2 Syntax

Die Sprachdatei besteht aus beliebig vielen Zeilen wie in diesem Beispiel:
Code

```
return                                                          {
["Orignaltext   mit   Leerzeichen"]   =   "the   translation   is   in   here",
["zweitens"]                  =                  "second                  line",
...
}
```

3.3 Codierung

Die Sprachdatei muss als **UTF-8** codiert sein

4 Beispiel

Diese Beispiel zeigt ein Grundgerüst einer *.mdl-Datei. Die Texte, die übersetzt werden sollen, lauten

"Signal" und "Signal von BR146". Wenn nur der Mod installiert ist, zeigt Train Fever diese beiden an. Wenn eine Sprachdatei (und natürlich TranslatorScript) installiert ist, wird "Segnale" statt "Signal" angezeigt (entsprechend beim zweiten Text). Sollte ein Text fehlen, dann wird der Originaltext vom Mod angezeigt.

*.mdl-Datei

Code

```
pcall(require,                                          'BR146_translate_script')


function                                                                    data()
return                                                                          {
[...]
metadata                                        =                               {



            },
[...]
}
end
```

Alles anzeigen

Sprachdatei

Code

```
return                                                                          {
                                                                ["Signal"]
                  ["Signal         von         BR146"]          =          "Segnale          (BR14
}
```

5 Quellen & Links

Thema im Forum
Download des Scripts