

Dateien in Notepad++ und LUA-Formatierung

Wer zum Öffnen der Spieldateien noch den Windows Editor benutzt, sollte sich spätestens jetzt überlegen, einen geeigneten Texteditor zu benutzen. Vorteile wie bessere Suche, Dateienmanagement und Code-Formatierung liegen klar auf der Hand. Ich kann jedem [Notepad++](#) empfehlen, da es sehr viele Funktionen bietet und dazu noch kostenlos ist.

Doch auch wer Notepad++ schon kennt, für den gibt es sicher noch ein paar Tricks zu entdecken.

Da es sich bei den (Text-)dateien von TF_n ja fast immer um LUA-Code handelt, ist es natürlich vorteilhaft, eine entsprechende Formatierung zu haben, die Syntax farblich darstellt und Kommentare, Klammern und Einrückungen deutlich angezeigt. Dadurch passieren auch weniger Fehler.

Zwar erkennt Notepad++ Dateien schon an der Endung, allerdings gibt es in TF_n ja viele unterschiedliche Dateiendungen, die erstmal nicht erkannt werden. Das kann man aber folgendermaßen einstellen:

Einstellungen > Stile > Sprache > Lua

Dann trägt man bei *Benutzer-Erw* ein: `con mdl module script mtl msh`

Außerdem noch ein paar Tipps:

- Kommentare gehen einfach mit STRG+Q (auch über mehrere Zeilen)
- Einrücken über mehrere Zeilen mit Tab (zurück mit Shift+Tab)
- Automatischer Zeilenumbruch würde ich einschalten, damit man nicht mehr horizontal scrollen muss
- In der Leiste findet man weitere nützliche Tools wie eine Miniaturansicht
- Wenn man viele Dateien von verschiedenen Projekten offen hat, ist die Sitzung-Funktion (bei Datei>) sehr praktisch
- Ich würde mir die stdout immer nach ganz links legen (wird diese geändert kommt eine Meldung)

```

34     debugPrint(loadedState)
35 end
36
37 local function guiInit() --- called once at startup
38     --print("-----GUI INIT-----")
39 end
40
41 local function guiUpdate() --- called every GUI frame
42     --print("guiUpdate", getRealTimeStr())
43 end
44
45 local function guiHandleEvent(id, name, param, ...)
46     --if name=="visibilityChange" then return end
47     --if name=="destroy" then return end
48     --if name=="idAdded" then return end
49     --print("guiHandleEvent:", id, name, param, ...)
50     --debugPrint(param)
51
52     if id=="trackBuilder" or id=="streetBuilder" or id=="streetTrackModifier" then
53         if name=="builder.proposalCreate" then
54             if param.data.errorState.critical or #param.proposal.proposal.addedSegments==0 then
55                 toolTipCont.destroy()
56                 return
57             end
58             --local status, ret = pcall(GeomInfo.calc, param)
59             local status, err = pcall(function()
60                 local sttime = os.clock()
61                 local ginf = GeomInfo.calc(param)

```