Alle Ressourcen zur Auswahl in Konstruktionsparametern

Inhaltsverzeichnis

- 1 Parameter
- 2 Update-Funktion extern
- 3 Ressourcen ermitteln
- 4 Übergabe der Ressourcen
- 5 Verwendung in der Update-Funktion

Mich hat es schon immer gestört, dass die Auswahl von Straßentypen in Konstruktionen zu gering war oder Bahnhöfe nur die 2 Vanilla-Gleise angeboten haben. Schöner wäre es, wenn man automatisch die Auswahl von allen im Spiel vorhandenen Typen hätte, was die Modkompatibilitäten erhöhen würde.

In diesem Post habe ich gefragt, ob man dieses Problem nicht irgendwie lösen kann, worauf jemand von UG schrieb, dass das mit dem neuen dynamischen Ressourcensystem möglich sei, mit Verweis auf dieses Beispiel im Wiki. Umgesetzt habe ich es dann bei der Mod **Advanced Person Magnet**, welche hier auch als Beispiel dient. Dabei wird ein Parameter für alle **Straßentypen** implementiert, aber für Gleise, Modelle oder Gütertypen ist das Vorgehen ähnlich.

1 Parameter

Der Parameter, um den es geht, wird zunächst wie gewöhnlich in der Konstruktion definiert. Als Typ bietet sich **Combobox** an.

Code: res\construction\Advanced_Person_Magnet.con

```
...
params                                  =                                  {
    {


      values = {"dummy","standard/town_medium_new.lua"},  -- wird später ersetzt
    },
...
```

(In APM werden die Parameter eigentlich mithilfe von ParamBuilder definiert)

2 Update-Funktion extern

Die updateFn muss aus der Konstruktion genommen und in einer seperaten .script Datei definiert werden. In der .con Datei stehen dann nur noch die statischen Infos sowie die Parameter. Dazu muss man in der postRunFn die .script Datei benennen:

Code: mod.lua

```
postRunFn = function (settings, modparams)
                                                    local            APMcon              =
     APMcon.updateScript.fileName = "construction/Advanced_Person_Magnet.updateFn"
    ...
```

3 Ressourcen ermitteln

Außerdem muss man alle Typen der entsprechenden Ressource mithilfe der API ermitteln. Der Übersicht halber kann man das in einer eigenen Datei machen:

Code: res\scripts\APM_streetRep.lua

```
local                        s                        =                        {}

s.getStreetTypes       =       function()            --       stark       vereinfacht
                                              local           streetRep          =          api
                                                                                      local
                                                                                      local
foid,stypairs(streetRegionaFilterungnSichtbarkeit4,sowie Sortierungdergndstraßenachinten
                                                                                    local


    end

end

return s
```

Alles anzeigen

Es werden die Dateinamen (Typ), die später in der Update-Funktion gebraucht werden, und die Namen, zur Anzeige in der Liste, zurückgegeben. Dabei kann man die Typen eventuell filtern oder sortieren, um die Übersichtlichkeit im Konstruktions-UI zu bewahren. Ist das Spiel geladen, sind die Parameter fix und können nicht mehr geändert werden.

4 Übergabe der Ressourcen

In der postRunFn wird dann die obige Funktion aufgerufen und die Liste an die Script-Parameter (für die updateFn) übergeben sowie deren Bezeichnungen an den Konstruktions-Parameter:

Code: mod.lua

```
local           streetRep           =           require           "APM_streetRep"
...
postRunFn           =           function           (settings,           modparams)
    ...
                      local      streetTypes,      streetNames      =      streetRep.getStreet



    }

                                                        for            _,p            in            pai
 ifp.key=="APMstreetType"then--hierderselbekeywieinderParameterdefinition

        end
    end
```

Alles anzeigen

5 Verwendung in der Update-Funktion

In der externen updateFn kann dann mithilfe der scriptParams auf die Ressourcenliste zugegriffen werden. Der Index wird wie gewöhnlich anhand von constrParams ermittelt.

Code: res\construction\Advanced_Person_Magnet.script

```
function                                                                   data()
return                                                                          {
                                                                  updateFn                =




                  local  streetIndex   =  constrParams.APMstreetType     --  startet  mit   0
                                                local      streettype      =     scriptParam





            },

                                                                    {     {    0,    2,     0
                                                                    {     {    5,    2,     0
            },

        ...
```

Alles anzeigen

Ein anderes Beispiel ist **Vehicle Asset Builder** wo alle Fahrzeugmodelle ermittelt werden und zu Assets hinzugefügt werden. Dabei findet zusätzlich zur Filterung der Modelle eine Kategorisierung der Fahrzeugtypen statt.

Ich hoffe, einigen damit geholfen zu haben und würde mich freuen, wenn die ein oder andere zukünftige Mod so eine Auswahl anbietet.

Mein Script aus APM zur Ermittlung aller Straßentypen (inklusive Filter) sowie auch das Script von VAB stelle ich zur freien Verfügung.