Parametertypen in Konstruktionen

Inhaltsverzeichnis

- 1 Eigenschaften aller Parametertypen
- 2 Parametertypen: uiType
  - 2.1 BUTTON
  - 2.2 SLIDER
  - 2.3 COMBOBOX
  - 2.4 ICON_BUTTON
  - 2.5 CHECKBOX
- 3 Rückgabewerte der Parameter

1 Eigenschaften aller Parametertypen

Alle Parametertypen haben gemeinsame Eigenschaften:

Eigenschaft	Inhalt	Default	Beschreibung
key	`String`	-	eindeutiger Identifier für den Parameter
name	`String`	""	Anzeigename für das Parametermenü
values	{`String`,`String`,...}	?	Anzeigewerte für das Parametermenü, siehe unten
defaultIndex	`Integer`	0	Vorgewählte Option beim ersten Aufruf
uiType	`"BUTTON"` oder `"SLIDER"` oder `"COMBOBOX"` oder `"ICON_BUTTON"` oder `"CHECKBOX"`	"BUTTON"	Art des Parameters, siehe unten
yearFrom	`Integer`	0	Parameter wird angezeigt ab
yearTo	`Integer`	unendlich	Parameter wird angezeigt bis
tooltip	`String`	""	Tooltip der beim Hovern über dem Namen des Parameters angezeigt wird

2 Parametertypen: uiType

Die Eigenheiten der verschiedenen Arten wird im Folgenden betrachtet:

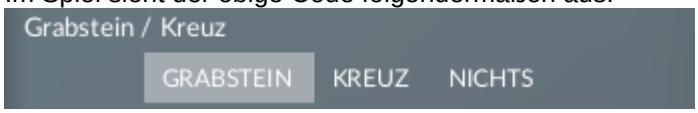
2.1 BUTTON

Einfache Text-Buttons sind bereits aus Transport Fever bekannt. Sie werden im Menü als zentrierte Liste aneinander gereiht. Die Beschriftung der Buttons wird im `values`-Parameter als Stringliste übergeben, diese können auch in der strings.lua übersetzt werden.

Lua

```
{
                                                name              =              _("Grabste
                  values     =     {     _("Grabstein"),     _("Kreuz"),     _("nichts"),
                   tooltip     =     _("Wähle     den     Aufbau     für     das     Grabr
},
```

Im Spiel sieht der obige Code folgendermaßen aus:



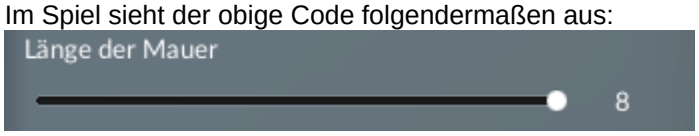
2.2 SLIDER

Gerade für viele lineare Werte, wie z.b. Zahlenreihen oder Größenabstufungen lassen sich Slider als einfach zu bedienende Regler einsetzen. Die Bezeichnung der aktuell ausgewählten Position wird rechts neben dem Slider angezeigt. Die Anzeigenamen der Buttons wird im `values`-Parameter als Stringliste übergeben, diese können auch in der strings.lua übersetzt werden.

Lua

```
{
                                                  name              =              _("Länge
    values = { _("1"), _("2"), _("3"), _("4"), _("5"), _("6"), _("7"), _("8"), },
                     tooltip     =     _("Wähle     die     Anzahl     der     Mauerelem
},
```

Im Spiel sieht der obige Code folgendermaßen aus:



2.3 COMBOBOX

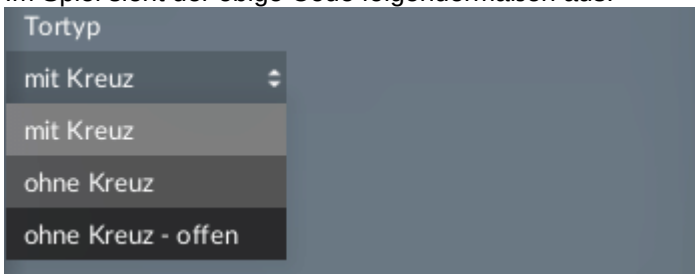
Wenn eine große Menge an Textoptionen zur Auswahl stehen soll, ist eine einfache Liste aus Buttons nicht ausreichend. Dann lässt sich eine Combobox, auch als Dropdown-Menü bekannt, realisieren, um kompakt viele Optionen anzubieten. Die Einträge der Liste werden im `values`-Parameter als Stringliste übergeben, diese können auch in der strings.lua übersetzt werden.

Lua

```
{


      values  =  {  _("mit  Kreuz"),  _("ohne  Kreuz"),  _("ohne  Kreuz  -  offen"),  },
                        tooltip      =      _("Wähle      die      Art      des      Friedl
},
```

Im Spiel sieht der obige Code folgendermaßen aus:



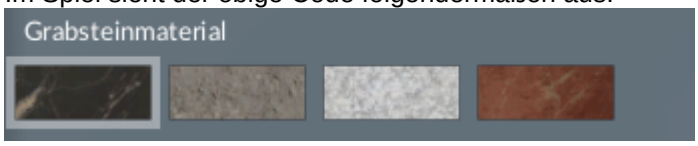
2.4 ICON_BUTTON

Im Gegensatz zu den Text-Buttons zeigen die Icon-Buttons kleine Bilder im TGA-Format an. Die Liste ist im Gegensatz zu den Text-Buttons aktuell linksbündig angeordnet. Wenn die Liste der Buttons breiter als das Parametermenü wird, bricht sie auf eine weitere Zeile um. Es ist deshalb ratsam, dass die Bilder für die Buttons alle die gleiche Höhe haben. Außerdem sollte der Rand der Buttons transparent sein, damit man erkennt, welche Option aktuell ausgewählt wurde. Die Dateipfade zu den Bildern werden im `values`-Parameter als Stringliste übergeben, die Pfade beziehen sich auf den kompletten Texturenordner (siehe Beispiel).

Lua

```
{
    "textures/grabstein_weiss.tga",    "textures/grabstein_rot.tga",
    tooltip = _("Wähle das Material für den Grabstein
},
```

Im Spiel sieht der obige Code folgendermaßen aus:



2.5 CHECKBOX

Wenn lediglich ein Wert mit ja oder nein belegt ist, eignet sich auch eine Checkbox als Parametertyp. Dabei wird lediglich der Name angezeigt, der `values`-Parameter muss aber dennoch definiert sein.

Lua

```
{
    name = _("Zufällige Kombination der
    values
    tooltip = _("Lege fest, ob die Grabmäler zufällig zusammengestellt werden"),
},
```

Im Spiel sieht der obige Code folgendermaßen aus:

3 Rückgabewerte der Parameter

In der `updateFn`-Funktion der Construction können die gewählten Optionen mit `params.<key>` abgerufen werden. Dabei werden Zahlen beginnend ab 0 ausgegeben. Der erste Eintrag eines Dropdowns liefert beispielsweise entsprechend eine 0. Bei der Checkbox entspricht 0 dem nicht aktivierten Zustand, 1 dem aktivierten Zustand.