Parameter types in constructions

All parameter types have common properties:

Property	Value	Default	Description
key	`String`	mandatory	unique identifier for the parameter
name	`String`	""	Display name for the parameter menu
values	{`String`,`String`,...}	mandatory	Display values for the parameter menu, see below
defaultIndex	`Integer`	0	Pre-selected option on first call
uiType	`"BUTTON"` or `"SLIDER"` or `"COMBOBOX"` or `"ICON_BUTTON"` or `"CHECKBOX"`	"BUTTON"	Type of parameter, see below
yearFrom	`Integer`	0	Parameter is displayed from
yearTo	`Integer`	infinite	Parameter is displayed until
tooltip	`String`	""	Tooltip that is displayed above the parameter name when hovering

The characteristics of the different types are considered below:

BUTTON

Simple text buttons are already known from Transport Fever. They are lined up in the menu as a centered list. The labels of the buttons are passed as a string list in the `values` parameter, they can also be translated in strings.lua.

Lua

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

In the game the above code looks like this:

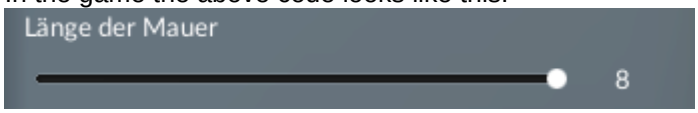
SLIDER

Especially for many linear values, such as numerical series or size increments, sliders can be used as easy-to-use controllers. The name of the currently selected position is displayed to the right of the slider. The display names of the buttons are passed as a string list in the `values` parameter, these can also be translated in the strings.lua.

Lua

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

In the game the above code looks like this:



COMBOBOX

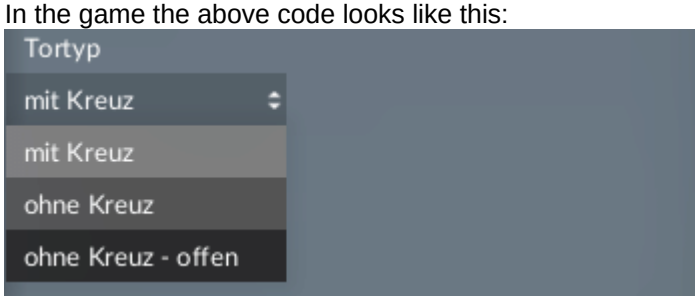
If a large number of text options should be available, a simple list of buttons is not sufficient. Then a combo box, also known as a drop-down menu, can be implemented to offer a compact number of options. The entries of the list are passed as a string list in the `values` parameter, they can also be translated in strings.lua.

Lua

```
{


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

In the game the above code looks like this:



https://www.transportfever.net/lexicon/entry/278-parameter-types-in-constructions/

ICON_BUTTON

Unlike the text buttons, the icon buttons display small images in TGA format. Unlike the text buttons, the list is currently left-aligned. If the list of buttons becomes wider than the parameter menu, it will wrap to another line. It is therefore advisable that the images for the buttons all have the same height. Also, the border of the buttons should be transparent so that you can see which option is currently selected. The file paths to the images are passed as a string list in the `values` parameter, the paths refer to the complete texture folder (see example)

Lua

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

In the game the above code looks like this:

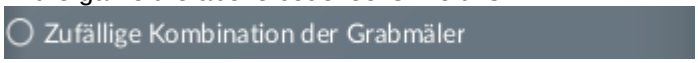
CHECKBOX

If only one value is assigned with yes or no, a checkbox is also suitable as parameter type. Only the name is displayed, but the `values` parameter must still be defined with two values, but it's values are never shown.

Lua

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

In the game the above code looks like this:



Return values of the parameters

In the `updateFn`-function of the Construction the selected options can be called up with `params.<key>`. Numbers starting from 0 are displayed. The first entry of a dropdown, for example, returns a 0. For a

checkbox, 0 corresponds to the non-activated state, 1 to the activated state.