

# Widgets



## Inhaltsverzeichnis

Das Widet allgemein.....	2
Aufbau.....	2
Name.....	2
IsShown.....	2
ForceToHandleMouseEventsFlag.....	2
ForceToNeverBeFoundFlag.....	2
Group.....	2
Zpriority.....	2
Container Widget.....	2
Static Widget.....	2
Static Text Widget.....	3
Button Widget.....	3
Gfx Button.....	3
Text Button.....	3
Progress Bar Widget.....	4
Pure Tooltip Widget.....	4
Custom Widgets.....	5
Klassen.....	5
GGUI::CMiniMapCustomWidget.....	5
GGUI::CminiMapOverlayCustomWidget.....	5

# Das Widet allgemein

---

Ein Widet der S5 GUI hat immer folgende Eigenschaften.

## Aufbau

### *Name*

Der name des Widgets. Dieser muss einmalig im gesamten XML sein.

### *IsShown*

Eine einfache Flag die aussagt ob das Widget bei Spielbeginn sichtbar sein soll.

### *ForceToHandleMouseEventsFlag*

Noch nicht herausgefunden.

### *ForceToNeverBeFoundFlag*

Noch nicht herausgefunden.

### *Group*

Noch nicht herausgefunden.

### *Zpriority*

Die Zpriorität des Widgets. Das Widget mit der höchsten Zpriorität in einem Container wird als oberstes angezeigt.

# Container Widget

---

Das Container Widget ist das häufigst verwendete in der S5 GUI. Es dient allein zum zusammenfassen weiterer Widgets die sich in diesem Widget befinden. Alle Widgets die sich in ihm befinden haben ihre Position relativ zum Container Widget.

Das Widget kann verwendet werden um schnell viele Widgets an- und aus- zuschalten. Dazu werden alle Widgets im Container Widget angezeigt, das Container Widget selbst aber abgeschaltet. Durch das einblenden des Container Widgets werden nun alle Widgets die sich in ihm befinden angezeigt.

# Static Widget

---

Das Static Widget ist nichts anderes als ein Widget mit einem Hintergrund. Der Hintergrund kann aus einem Bild oder einer Farbe im ARGB Format bestehen.

Ein gutes Beispiel ist dieses Widget hier:



## Static Text Widget

Das Static Text Widget ist wie eine Erweiterung des Static Widgets und hat zusätzlich einen Schriftzug innerhalb. Zudem besitzt sie auch noch die Eigenschaft vom Spiel upgedated werden zu können.

Die Schrift hat ihre eigene Schriftart sowie eine Farbe.

## Button Widget

Das wohl wichtigste Widget im ganzen Spiel besitzt 5 States und zu jedem eine Textur. Die States sind:

- Normal
- Hover
- Pressed
- Disabled
- Highlighted



Da vom Spiel aus immer nur 4 Texturen eines Buttons existieren wird auch noch viel mit den RGBA Werten gespielt um weitere States zu erreichen.

Ein Button Widget hat auch immer die Möglichkeit einen Tooltip einblenden zu lassen wenn man die Maus darüber bewegt. Neben dieser Möglichkeit kann auch eine Lua Funktion ausgeführt werden wenn sich die Maus über dem Button bewegt.

Buttons gibt es in zwei Formen:

### *Gfx Button*

Der Gfx Button wird für rein grafische Buttons verwendet und hat eine zusätzliche Textur die über alle anderen Textur States gezeichnet wird.

### *Text Button*

Der Text Button hat wie der Name schon sagt einen Textkörper welche wiederum eine Schriftart besitzt.



## Progress Bar Widget

---

Dieses Widget wird verwendet um Fortschrittsbalken zu erstellen. Er besitzt einen Anfangs Wert der aber relativ sinnlos ist da dieses Widget sowieso per Skript kontrolliert werden muss und einen Maximalwert.

Ein allseits bekanntes Beispielwidget ist dieses:



## Pure Tooltip Widget

---

Das Pure Tooltip Widget ist ein unsichtbares Widget, welches nur den Zweck hat beim Eintritt der Maus einen Tooltip anzuzeigen. Wie der Button hat auch dieses Widget die Möglichkeit eine Lua Funktion ausführen zu lassen sobald sich die Maus über ihm befindet.

# Custom Widgets

---

Die Custom Widgets sind eine spezielle Form von Widgets welche auf Klassen beruhen die in C geschrieben wurden. Diese Klassen können nicht geändert oder selbst erstellt werden, jedoch aber wiederverwendet.

## Klassen

### *GGUI::CMiniMapCustomWidget*

Diese Klasse wird vom Spiel für Minimaps verwendet und stellt die Karte stark skaliert in einem kleinen Kreis dar.



### *GGUI::CminiMapOverlayCustomWidget*

Das klicken auf die Minimap und das folgende Springen der Kamera auf der echten Map erfolgt wiederum über ein weiteres Widget mit der MapOverlay Klasse

*Und noch viele mehr...*