SFML Gui

0.1

Generated by Doxygen 1.5.6

Fri Jan 2 12:07:59 2009

Contents

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 2

Class Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 sfgui Namespace Reference

All graphics widgets.

Classes

- class Button

  A push button.

- struct Margin

  Struct which contains margin values.

- class Object

  A simple graphic item.

- class TextEdit

  A single line text entry.

Enumerations

- enum { Left, Right, Center }

5.1.1 Detailed Description

All graphics widgets.

This namespace contains all the graphics widgets

5.1.2 Enumeration Type Documentation

5.1.2.1 anonymous enum

This enumeration defines the position of an item (such as text). It is used by widget to define autoposition (like Button which has SetTextPosition...)

Enumerator:

Left Object is on the left.

Right Object is on the right.

Center Object is centered.

Chapter 6

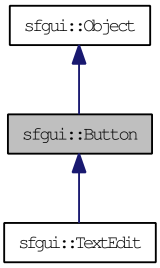
Class Documentation

6.1 sfgui::Button Class Reference

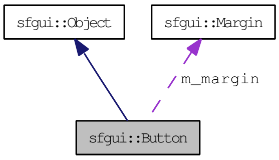
A push button.

```
#include <button.hpp>
```

Inheritance diagram for sfgui::Button:



Collaboration diagram for sfgui::Button:



Public Member Functions

- Button (sf::RenderWindow ∗parentWindow)
- ∼Button ()
- std::string getText ()
- void SetText (std::string text)
- std::string GetText ()
- void SetTextColor (sf::Color &)

- sf::Color GetTextColor ()
- void SetTextSize (float)
- float GetTextSize ()
- void SetTextFont (sf::Font &)
- sf::Font GetTextFont ()
- void SetTextAlignment (int)
- void SetTextMargin (float)
- void SetTextLeftMargin (float)
- void SetTextRightMargin (float)
- void SetTextTopMargin (float)
- void SetTextBottomMargin (float)
- void SetPosition (float x, float y)
- void Move (float x, float y)
- void Show ()

Protected Member Functions

- void updateTextPos ()

Protected Attributes

- sf::String m_text
- int m_textAlignment
- sfgui::Margin m_margin

6.1.1 Detailed Description

A push button.

This class represents a push button graphic item. It provides some signals (call some useful callbacks) like clicked, mouseOver...

6.1.2 Constructor & Destructor Documentation

6.1.2.1 sfgui::Button::Button (sf::RenderWindow ∗ *parentWindow*)

Create a Button on the parent window.

6.1.2.2 sfgui::Button::∼Button ()

6.1.3 Member Function Documentation

6.1.3.1 void sfgui::Button::updateTextPos () `[protected]`

6.1.3.2 std::string sfgui::Button::getText () `[inline]`

Get the button text

6.1.3.3 void sfgui::Button::SetText (std::string *text*)

Set the button text.

6.1.3.4 std::string sfgui::Button::GetText ()

6.1.3.5 void sfgui::Button::SetTextColor (sf::Color & *color*)

6.1.3.6 sf::Color sfgui::Button::GetTextColor ()

6.1.3.7 void sfgui::Button::SetTextSize (float *size*)

Set size of the text

6.1.3.8 float sfgui::Button::GetTextSize ()

6.1.3.9 void sfgui::Button::SetTextFont (sf::Font & *font*)

6.1.3.10 sf::Font sfgui::Button::GetTextFont ()

6.1.3.11 void sfgui::Button::SetTextAlignment (int *al*)

This set the position of the text on the button. You should use one of the following constant value sfgui::LEFT, sfgui::RIGHT, sfgui::CENTER

6.1.3.12 void sfgui::Button::SetTextMargin (float *margin*)

Set the global margin. The text will be spaced from the button each border by a number of pixels.

6.1.3.13 void sfgui::Button::SetTextLeftMargin (float *margin*)

Set the left margin. Text will be spaced from the left border of the button

6.1.3.14 void sfgui::Button::SetTextRightMargin (float *margin*)

Set the right margin. Text will be spaced form the right border of the button.

6.1.3.15 void sfgui::Button::SetTextTopMargin (float *margin*)

Set the top margin. Text will be spaced from the top border of the button.

6.1.3.16 void sfgui::Button::SetTextBottomMargin (float *margin*)

set the bottom margin. Text will be spaced from the bottom border of the button

6.1.3.17 void sfgui::Button::SetPosition (float *x*, float *y*)

Set the button position. Adjust the text position to keep the text in the choosen position (center, left, right...)

6.1.3.18 void sfgui::Button::Move (float *x*, float *y*)

Move the button and adjust the text position

6.1.3.19 void sfgui::Button::Show ()

Display the button on the parent window

Reimplemented in sfgui::TextEdit.

6.1.4 Member Data Documentation

6.1.4.1 sf::String sfgui::Button::m_text `[protected]`

6.1.4.2 int sfgui::Button::m_textAlignment `[protected]`

6.1.4.3 sfgui::Margin sfgui::Button::m_margin `[protected]`

The documentation for this class was generated from the following files:

- button.hpp
- button.cpp

6.2 sfgui::Margin Struct Reference

Struct which contains margin values.

`#include <margin.hpp>`

Public Member Functions

- void SetMargin (float)

Public Attributes

- float Left

  Left margin value.

- float Right

  Right margin value.

- float Top

  Top margin value.

- float Bottom

  Bottom margin value.

6.2.1 Detailed Description

Struct which contains margin values.

This struc contains the pixel spacing from each border of an object. The function SetMargin is provided for convenience as it set all the margin with the same value.

6.2.2 Member Function Documentation

6.2.2.1 void sfgui::Margin::SetMargin (float *margin*)

6.2.3 Member Data Documentation

6.2.3.1 float sfgui::Margin::Left

Left margin value.

6.2.3.2 float sfgui::Margin::Right

Right margin value.

6.2.3.3 float sfgui::Margin::Top

Top margin value.

6.2.3.4 float sfgui::Margin::Bottom

Bottom margin value.

The documentation for this struct was generated from the following files:

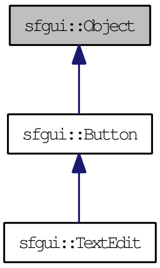
- margin.hpp
- margin.cpp

6.3 sfgui::Object Class Reference

A simple graphic item.

`#include <object.hpp>`

Inheritance diagram for sfgui::Object:



Public Member Functions

- Object (sf::RenderWindow ∗parentWindow)
- ∼Object ()
- void SetBackground (std::string)
- void SetBackground (sf::Image &)
- void SetBackground (sf::Color &)
- void SetBackground (sf::Color &, sf::Color &)
- void CheckEvent (sf::Event Event)
- void SetClickCallback (void(∗)())
- void SetMouseHoverCallback (void(∗)())
- void ManageMouse ()
- void Clicked ()
- void MouseHover ()

Protected Attributes

- sf::RenderWindow ∗ m_parentRenderWindow

  Pointer to the parent sf::RenderWindow.

- sf::Image ∗ m_BackgroundImg

  Curent background image.

- sf::Event m_Event

  Copy of the current sfml event.

Private Attributes

- void(∗ m_clickCallback)()

  Pointer to the click callback function.

- void(∗ m_mouseHoverCallback)()

6.3.1 Detailed Description

A simple graphic item.

This is a very simple object, which is rather useless. It's mostly present to be used as parent by other graphics items classes. This class create a sf::Sprite which represent a graphic item. There are some basic functions to set the general appearance like setBackground. This class can handle some common events which are possibly used by all widgets like clicks, mouse over...

6.3.2 Constructor & Destructor Documentation

6.3.2.1 sfgui::Object::Object (sf::RenderWindow $*$ *parentWindow*)

Test de classe

Construct a simple graphic object

6.3.2.2 sfgui::Object::∼Object ()

6.3.3 Member Function Documentation

6.3.3.1 void sfgui::Object::SetBackground (std::string *path*)

Set a background from an image path

6.3.3.2 void sfgui::Object::SetBackground (sf::Image & *Img*)

Set a background from an existing sf::Image

6.3.3.3 void sfgui::Object::SetBackground (sf::Color & *color*)

Set a background color

6.3.3.4 void sfgui::Object::SetBackground (sf::Color &, sf::Color &)

6.3.3.5 void sfgui::Object::CheckEvent (sf::Event *Event*)

Callbacks

Call the callbacks functions if needed on the given Event.

Reimplemented in sfgui::TextEdit.

6.3.3.6 void sfgui::Object::SetClickCallback (void($*$)() *clickCallBack*)

Set a pointer to the callback function called when the user click on this object

6.3.3.7 void sfgui::Object::SetMouseHoverCallback (void($*$)() *mouseHoverCallback*)

Set a pointer to the callback function called when the mouse is hover the button.

6.3.3.8 void sfgui::Object::ManageMouse ()

Manage the mouse events.

6.3.3.9 void sfgui::Object::Clicked ()

Call the click callback function. It's automatically called by checkEvent, but you can also call it yourself if needed.

6.3.3.10 void sfgui::Object::MouseHover ()

6.3.4 Member Data Documentation

6.3.4.1 void(∗ sfgui::Object::m_clickCallback)() `[private]`

Pointer to the click callback function.

6.3.4.2 void(∗ sfgui::Object::m_mouseHoverCallback)() `[private]`

Pointer to the callback function called when mouse is hover the button

6.3.4.3 sf::RenderWindow∗ sfgui::Object::m_parentRenderWindow `[protected]`

Pointer to the parent sf::RenderWindow.

6.3.4.4 sf::Image∗ sfgui::Object::m_BackgroundImg `[protected]`

Curent background image.

6.3.4.5 sf::Event sfgui::Object::m_Event `[protected]`

Copy of the current sfml event.

The documentation for this class was generated from the following files:

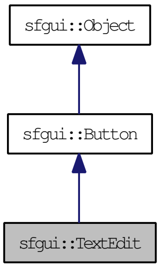
- object.hpp
- object.cpp

6.4 sfgui::TextEdit Class Reference

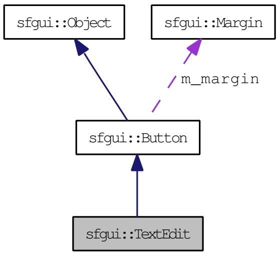
A single line text entry.

```
#include <textedit.hpp>
```

Inheritance diagram for sfgui::TextEdit:



Collaboration diagram for sfgui::TextEdit:



Public Member Functions

- TextEdit (sf::RenderWindow ∗parentWindow)
- void SetText (std::string &)
- void AddChar (char)
- void DelChar (int)
- void Activate ()
- void Deactivate ()
- void CheckEvent (sf::Event Event)
- void SetTextChangedCallback (void(∗)(std::string &))
- void SetReturnPressedCallback (void(∗)())
- void SetActivatedCallback (void(∗)())
- void SetDeactivatedCallback (void(∗)())
- void Show ()

Private Member Functions

- void textChanged ()
- void deactivated ()
- void activated ()

Private Attributes

- std::string m_stdText
- int m_nbCharToShow
- bool m_itemActive
- void(∗ m_textChangedCallback)(std::string &)
- void(∗ m_deactivatedCallback)()
- void(∗ m_activatedCallback)()
- void(∗ m_returnPressedCallback)()

6.4.1 Detailed Description

A single line text entry.

This class provides a single line text entry widget. Thanks to this class, the user can enter text directly in the application, and usefull callbacks are provided, like textChangedCallback, returnPressedCallback.

Small tutorial

If you want to enter text, you must have the item activated. You can activate it by calling the Activate() function (when user click on the TextEdit, this function is automatically called).

If you want to stop typing text in the widget, you must call the Deactivate() function.

Important : you must call the CheckEvent function in the event loop. If you don't, the widget will just be shown, but won't work

6.4.2 Constructor & Destructor Documentation

6.4.2.1 sfgui::TextEdit::TextEdit (sf::RenderWindow ∗ *parentWindow*)

6.4.3 Member Function Documentation

6.4.3.1 void sfgui::TextEdit::textChanged () `[private]`

If the text is modified, this function is called. It updates the sfString on the screen, and it call the textChanged callback (if exists)

6.4.3.2 void sfgui::TextEdit::deactivated () `[private]`

6.4.3.3 void sfgui::TextEdit::activated () `[private]`

6.4.3.4 void sfgui::TextEdit::SetText (std::string & *text*)

Set the text on the TextEdit

6.4.3.5 void sfgui::TextEdit::AddChar (char *ch*)

Add one char at the end of the current string

6.4.3.6 void sfgui::TextEdit::DelChar (int *pos*)

Delete one char from position pos

6.4.3.7 void sfgui::TextEdit::Activate ()

Activate the TextEdit. When the TextEdit is activated, all pressed keys are added to the TextEdit string

6.4.3.8 void sfgui::TextEdit::Deactivate ()

Deactivate the Textedit. Key you press will no longer be added to the TextEdit (until Activate() is called)

6.4.3.9 void sfgui::TextEdit::CheckEvent (sf::Event *Event*)

Callbacks

Call the callbacks functions if needed on the given Event.

Reimplemented from sfgui::Object.

6.4.3.10 void sfgui::TextEdit::SetTextChangedCallback (void(∗)(std::string &) *textChangedCallback*)

Set the callback called when the text is modified (add new char, delete one...)

6.4.3.11 void sfgui::TextEdit::SetReturnPressedCallback (void(∗)() *returnPressedCallback*)

Set the callback called when the retun key is pressed (it can be used to know when the user finished his text entry for instance

6.4.3.12 void sfgui::TextEdit::SetActivatedCallback (void(∗)() *activatedCallback*)

Set the callback called when the TextEdit is activated (ie has focus).

6.4.3.13 void sfgui::TextEdit::SetDeactivatedCallback (void(∗)() *deactivatedCallback*)

Set the callback called when the TextEdit is deactivated (ie hasn't focus).

6.4.3.14 void sfgui::TextEdit::Show ()

Show the TextEdit on the parent window

Reimplemented from sfgui::Button.

6.4.4 Member Data Documentation

6.4.4.1 std::string sfgui::TextEdit::m_stdText [private]

6.4.4.2 int sfgui::TextEdit::m_nbCharToShow [private]

6.4.4.3 bool sfgui::TextEdit::m_itemActive [private]

If true, user can enter text (textedit has focus), if false, it is disabled

6.4.4.4 void(∗ sfgui::TextEdit::m_textChangedCallback)(std::string &) [private]

6.4.4.5 void(∗ sfgui::TextEdit::m_deactivatedCallback)() [private]

6.4.4.6 void(∗ sfgui::TextEdit::m_activatedCallback)() [private]

6.4.4.7 void(∗ sfgui::TextEdit::m_returnPressedCallback)() [private]

The documentation for this class was generated from the following files:

- textedit.hpp
- textedit.cpp

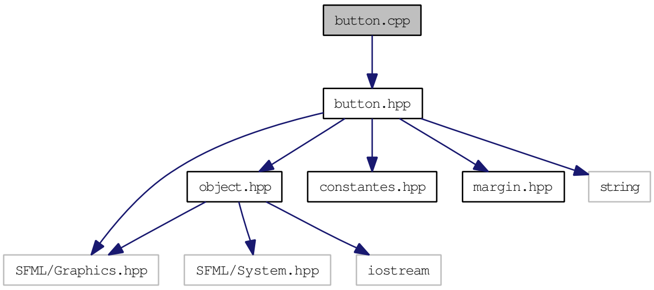
Chapter 7

File Documentation

7.1 button.cpp File Reference

```
#include "button.hpp"
```

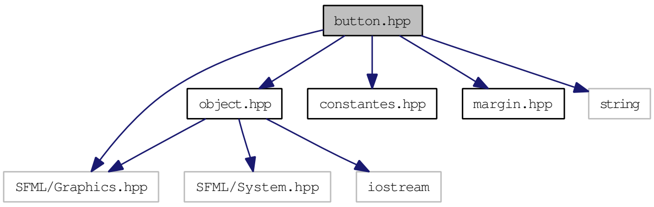
Include dependency graph for button.cpp:



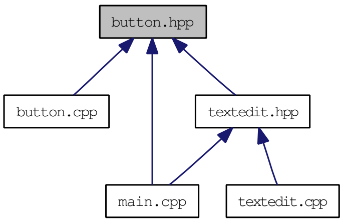
7.2 button.hpp File Reference

```
#include <SFML/Graphics.hpp>
#include "object.hpp"
#include "constantes.hpp"
#include "margin.hpp"
#include <string>
```

Include dependency graph for button.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace sfgui

Classes

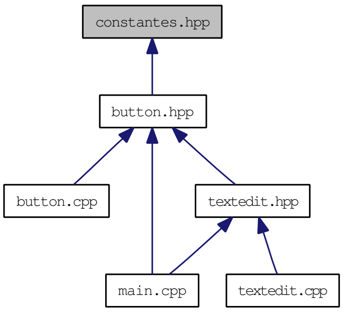
- class sfgui::Button

  A push button.

7.3 constantes.hpp File Reference

Description not set.

This graph shows which files directly or indirectly include this file:



Namespaces

- namespace sfgui

Enumerations

- enum { sfgui::Left, sfgui::Right, sfgui::Center }

7.3.1 Detailed Description

Description not set.

Author:

TANGUY Arnaud <arn.tanguy@gmail.com>

Date:

2009

Version:

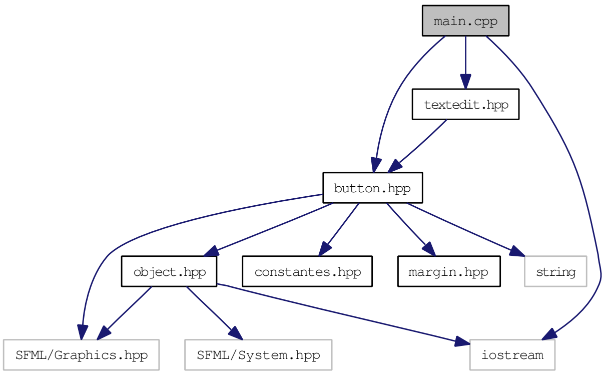
0.1

The big description for doxygen

7.4 main.cpp File Reference

```
#include "button.hpp"
#include "textedit.hpp"
#include <iostream>
```

Include dependency graph for main.cpp:



Functions

- sf::RenderWindow App (sf::VideoMode(800, 600, 32),"SFML GUI")
- void clickedCallBack ()
- void textChangeCallback (std::string &sr)
- int main ()

Variables

- sfgui::Button Sprite & App

7.4.1 Function Documentation

7.4.1.1 sf::RenderWindow App (sf:: *VideoMode*800, 600, 32, "SFML GUI")

7.4.1.2 void clickedCallBack ()

7.4.1.3 int main ()

7.4.1.4 void textChangeCallback (std::string & *sr*)

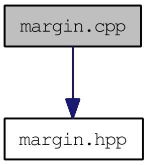
7.4.2 Variable Documentation

7.4.2.1 sfgui::Button Sprite& App

7.5 margin.cpp File Reference

```
#include "margin.hpp"
```

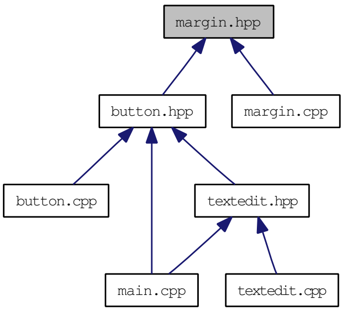
Include dependency graph for margin.cpp:



7.6 margin.hpp File Reference

A simple struct for margin.

This graph shows which files directly or indirectly include this file:



Namespaces

- namespace sfgui

Classes

- struct sfgui::Margin

  Struct which contains margin values.

7.6.1 Detailed Description

A simple struct for margin.

Author:

TANGUY Arnaud <`arn.tanguy@gmail.com`>

Date:

2009

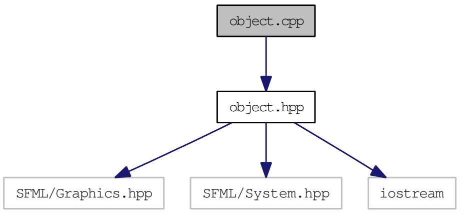
Version:

0.1

7.7 object.cpp File Reference

```
#include "object.hpp"
```

Include dependency graph for object.cpp:



Functions

- void SetBackground (sf::Color &, sf::Color &)

7.7.1 Function Documentation

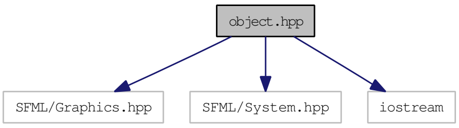
7.7.1.1 void SetBackground (sf::Color &, sf::Color &)

Shaded of between two colors. Not implemented yet.

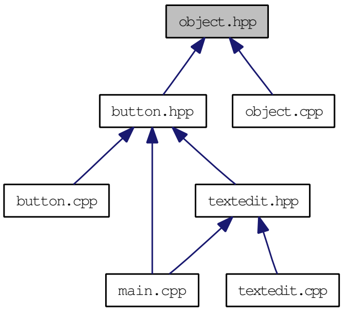
7.8 object.hpp File Reference

```
#include <SFML/Graphics.hpp>
#include <SFML/System.hpp>
#include <iostream>
```

Include dependency graph for object.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace sfgui

Classes

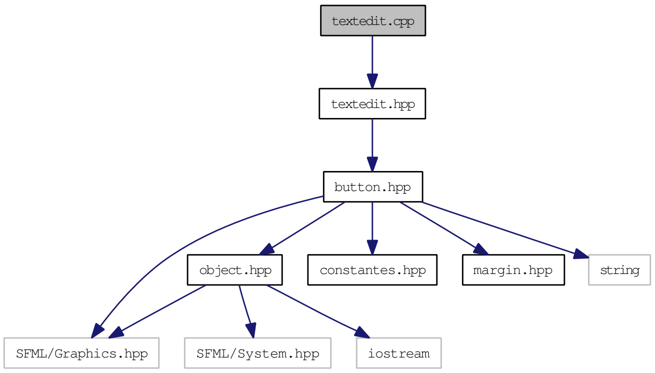
- class sfgui::Object

  A simple graphic item.

7.9 textedit.cpp File Reference

```
#include "textedit.hpp"
```

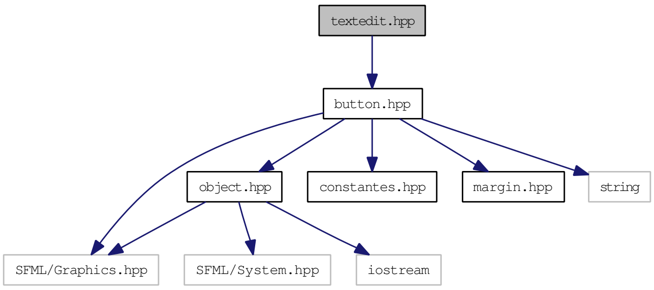
Include dependency graph for textedit.cpp:



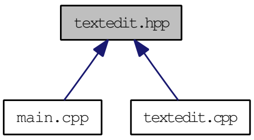
7.10 textedit.hpp File Reference

```
#include "button.hpp"
```

Include dependency graph for textedit.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace sfgui

Classes

- class sfgui::TextEdit

    A single line text entry.

Index