

OpenGui

Generated by Doxygen 1.8.2

Thu Nov 1 2012 01:55:26



# Contents

<b>1</b>	<b>barbs-bandits</b>	<b>1</b>
<b>2</b>	<b>Hierarchical Index</b>	<b>3</b>
2.1	Class Hierarchy . . . . .	3
<b>3</b>	<b>Class Index</b>	<b>5</b>
3.1	Class List . . . . .	5
<b>4</b>	<b>File Index</b>	<b>7</b>
4.1	File List . . . . .	7
<b>5</b>	<b>Class Documentation</b>	<b>9</b>
5.1	Button Class Reference . . . . .	9
5.1.1	Detailed Description . . . . .	9
5.1.2	Constructor & Destructor Documentation . . . . .	10
5.1.2.1	Button . . . . .	10
5.1.2.2	Button . . . . .	10
5.1.2.3	Button . . . . .	10
5.1.2.4	Button . . . . .	10
5.1.2.5	Button . . . . .	10
5.1.3	Member Function Documentation . . . . .	10
5.1.3.1	setBgImg . . . . .	10
5.1.3.2	setText . . . . .	10
5.1.4	Member Data Documentation . . . . .	10
5.1.4.1	_imageE . . . . .	10
5.1.4.2	_textE . . . . .	10
5.2	CheckBox Class Reference . . . . .	11
5.2.1	Detailed Description . . . . .	11
5.2.2	Constructor & Destructor Documentation . . . . .	11
5.2.2.1	CheckBox . . . . .	11

5.2.2.2	CheckBox	11
5.3	Element Class Reference	11
5.3.1	Detailed Description	13
5.3.2	Constructor & Destructor Documentation	13
5.3.2.1	Element	13
5.3.2.2	Element	13
5.3.2.3	Element	13
5.3.2.4	~Element	13
5.3.3	Member Function Documentation	13
5.3.3.1	addChild	14
5.3.3.2	clearResult	14
5.3.3.3	getId	14
5.3.3.4	mouseInput	14
5.3.3.5	operator<	14
5.3.3.6	registerCallback	14
5.3.3.7	render	14
5.3.3.8	setDirty	15
5.3.3.9	setHeight	15
5.3.3.10	setWidth	15
5.3.3.11	setX	15
5.3.3.12	setY	15
5.3.3.13	setZ	15
5.3.4	Member Data Documentation	15
5.3.4.1	_height	15
5.3.4.2	_result	16
5.3.4.3	_width	16
5.3.4.4	_xCoord	16
5.3.4.5	_yCoord	16
5.4	Image Class Reference	16
5.4.1	Detailed Description	17
5.4.2	Constructor & Destructor Documentation	17
5.4.2.1	Image	17
5.4.2.2	Image	17
5.4.2.3	Image	17
5.4.2.4	Image	17
5.4.2.5	Image	17
5.4.2.6	~Image	17

5.4.3	Member Function Documentation	17
5.4.3.1	blit	17
5.4.3.2	get	17
5.4.3.3	getPixels	18
5.4.3.4	height	18
5.4.3.5	set	18
5.4.3.6	set	18
5.4.3.7	width	18
5.4.4	Friends And Related Function Documentation	18
5.4.4.1	operator<<	18
5.5	ImageElement Class Reference	18
5.5.1	Detailed Description	19
5.5.2	Constructor & Destructor Documentation	19
5.5.2.1	ImageElement	19
5.5.2.2	ImageElement	19
5.5.2.3	ImageElement	19
5.5.2.4	ImageElement	19
5.5.2.5	ImageElement	19
5.5.3	Member Function Documentation	19
5.5.3.1	clearResult	19
5.6	Pixel Class Reference	20
5.6.1	Detailed Description	20
5.6.2	Constructor & Destructor Documentation	20
5.6.2.1	Pixel	20
5.6.2.2	Pixel	20
5.6.2.3	Pixel	20
5.6.2.4	Pixel	20
5.6.3	Member Function Documentation	21
5.6.3.1	getA	21
5.6.3.2	getB	21
5.6.3.3	getG	21
5.6.3.4	getR	21
5.6.3.5	setRGB	21
5.6.3.6	setRGBA	21
5.6.4	Friends And Related Function Documentation	21
5.6.4.1	operator<<	21
5.7	Text Class Reference	21

5.7.1	Detailed Description	22
5.7.2	Constructor & Destructor Documentation	22
5.7.2.1	Text	22
5.7.2.2	Text	22
5.7.2.3	Text	22
5.7.2.4	Text	22
5.7.2.5	~Text	22
5.7.3	Member Function Documentation	22
5.7.3.1	getImage	22
5.7.3.2	getText	22
5.7.3.3	setText	22
5.8	TextElement Class Reference	23
5.8.1	Detailed Description	23
5.8.2	Constructor & Destructor Documentation	23
5.8.2.1	TextElement	23
5.8.2.2	TextElement	23
5.8.2.3	TextElement	23
5.8.2.4	~TextElement	24
5.8.3	Member Function Documentation	24
5.8.3.1	clearResult	24
5.8.3.2	setText	24
5.9	ToggleButton Class Reference	24
5.9.1	Detailed Description	25
5.9.2	Constructor & Destructor Documentation	25
5.9.2.1	ToggleButton	25
5.9.2.2	ToggleButton	25
5.9.2.3	ToggleButton	25
5.9.2.4	ToggleButton	25
5.9.2.5	ToggleButton	25
<b>6</b>	<b>File Documentation</b>	<b>27</b>
6.1	button/Button.h File Reference	27
6.2	button/main.cpp File Reference	27
6.2.1	Function Documentation	27
6.2.1.1	main	27
6.3	main.cpp File Reference	28
6.3.1	Function Documentation	28

6.3.1.1	<a href="#">buttonClicked</a>	28
6.3.1.2	<a href="#">closeWindow</a>	28
6.3.1.3	<a href="#">draw</a>	28
6.3.1.4	<a href="#">init</a>	29
6.3.1.5	<a href="#">loadGuiTexture</a>	29
6.3.1.6	<a href="#">main</a>	29
6.3.1.7	<a href="#">mainLoop</a>	29
6.3.1.8	<a href="#">mouseClicked</a>	29
6.3.1.9	<a href="#">shutDown</a>	29
6.3.1.10	<a href="#">windowResize</a>	29
6.3.2	<a href="#">Variable Documentation</a>	29
6.3.2.1	<a href="#">texture</a>	29
6.4	<a href="#">text/main.cpp File Reference</a>	29
6.4.1	<a href="#">Function Documentation</a>	30
6.4.1.1	<a href="#">main</a>	30
6.5	<a href="#">checkboxbox/CheckBox.h File Reference</a>	30
6.6	<a href="#">element/Element.cpp File Reference</a>	30
6.7	<a href="#">element/Element.h File Reference</a>	30
6.7.1	<a href="#">Detailed Description</a>	31
6.8	<a href="#">element/Main.cpp File Reference</a>	31
6.8.1	<a href="#">Function Documentation</a>	31
6.8.1.1	<a href="#">main</a>	31
6.9	<a href="#">image/Main.cpp File Reference</a>	31
6.9.1	<a href="#">Function Documentation</a>	31
6.9.1.1	<a href="#">main</a>	31
6.10	<a href="#">togglebutton/Main.cpp File Reference</a>	31
6.10.1	<a href="#">Function Documentation</a>	32
6.10.1.1	<a href="#">main</a>	32
6.11	<a href="#">image/Image.cpp File Reference</a>	32
6.11.1	<a href="#">Function Documentation</a>	32
6.11.1.1	<a href="#">operator&lt;&lt;</a>	32
6.11.2	<a href="#">Variable Documentation</a>	32
6.11.2.1	<a href="#">ERRORPIXEL</a>	32
6.12	<a href="#">image/Image.h File Reference</a>	32
6.13	<a href="#">image/ImageElement.h File Reference</a>	33
6.14	<a href="#">image/Pixel.cpp File Reference</a>	33
6.14.1	<a href="#">Function Documentation</a>	33

6.14.1.1 operator<<	33
6.15 image/Pixel.h File Reference	33
6.16 main.h File Reference	33
6.16.1 Function Documentation	34
6.16.1.1 buttonClicked	34
6.16.1.2 draw	34
6.16.1.3 init	34
6.16.1.4 loadGuiTexture	34
6.16.1.5 mainLoop	34
6.16.1.6 mouseClicked	34
6.16.1.7 shutDown	34
6.16.1.8 windowResize	35
6.17 README.md File Reference	35
6.18 text/Text.cpp File Reference	35
6.19 text/Text.h File Reference	35
6.20 text/TextElement.cpp File Reference	35
6.21 text/TextElement.h File Reference	35
6.22 togglebutton/ToggleButton.h File Reference	36



## Chapter 1

# barbs-bandits

OpenGL Internal GUI Framework



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

Element . . . . .	11
Button . . . . .	9
ToggleButton . . . . .	24
CheckBox . . . . .	11
ImageElement . . . . .	18
TextElement . . . . .	23
Image . . . . .	16
Pixel . . . . .	20
Text . . . . .	21



## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">Button</a>	9
<a href="#">CheckBox</a>	11
<a href="#">Element</a>	
The base class that all GUI elements derive from	11
<a href="#">Image</a>	16
<a href="#">ImageElement</a>	18
<a href="#">Pixel</a>	20
<a href="#">Text</a>	21
<a href="#">TextElement</a>	23
<a href="#">ToggleButton</a>	24



## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

<a href="#">main.cpp</a>	28
<a href="#">main.h</a>	33
<a href="#">button/Button.h</a>	27
<a href="#">button/main.cpp</a>	27
<a href="#">checkbox/CheckBox.h</a>	30
<a href="#">element/Element.cpp</a>	30
<a href="#">element/Element.h</a>	30
<a href="#">element/Main.cpp</a>	31
<a href="#">image/Image.cpp</a>	32
<a href="#">image/Image.h</a>	32
<a href="#">image/ImageElement.h</a>	33
<a href="#">image/Main.cpp</a>	31
<a href="#">image/Pixel.cpp</a>	33
<a href="#">image/Pixel.h</a>	33
<a href="#">text/main.cpp</a>	29
<a href="#">text/Text.cpp</a>	35
<a href="#">text/Text.h</a>	35
<a href="#">text/TextElement.cpp</a>	35
<a href="#">text/TextElement.h</a>	35
<a href="#">togglebutton/Main.cpp</a>	31
<a href="#">togglebutton/ToggleButton.h</a>	36





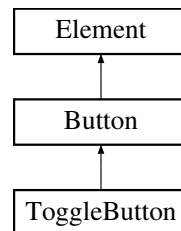
## Chapter 5

# Class Documentation

### 5.1 Button Class Reference

```
#include <Button.h>
```

Inheritance diagram for Button:



#### Public Member Functions

- [Button](#) ()
- [Button](#) (unsigned int x, unsigned int y)
- [Button](#) (unsigned int x, unsigned int y, unsigned int width, unsigned int height)
- [Button](#) (unsigned int x, unsigned int y, unsigned int width, unsigned int height, string txt)
- [Button](#) (unsigned int x, unsigned int y, unsigned int width, unsigned int height, [ImageElement](#) \*img)
- void [setBgImg](#) ([ImageElement](#) \*img)
- void [setText](#) (string txt)

#### Protected Attributes

- [TextElement](#) \* [\\_textE](#)
- [ImageElement](#) \* [\\_imageE](#)

#### 5.1.1 Detailed Description

Definition at line 13 of file Button.h.

### 5.1.2 Constructor & Destructor Documentation

#### 5.1.2.1 `Button::Button ( )` `[inline]`

Definition at line 15 of file `Button.h`.

#### 5.1.2.2 `Button::Button ( unsigned int x, unsigned int y )` `[inline]`

Definition at line 16 of file `Button.h`.

#### 5.1.2.3 `Button::Button ( unsigned int x, unsigned int y, unsigned int width, unsigned int height )` `[inline]`

Definition at line 18 of file `Button.h`.

#### 5.1.2.4 `Button::Button ( unsigned int x, unsigned int y, unsigned int width, unsigned int height, string txt )` `[inline]`

Definition at line 20 of file `Button.h`.

#### 5.1.2.5 `Button::Button ( unsigned int x, unsigned int y, unsigned int width, unsigned int height, ImageElement * img )` `[inline]`

Definition at line 28 of file `Button.h`.

### 5.1.3 Member Function Documentation

#### 5.1.3.1 `void Button::setBgImg ( ImageElement * img )` `[inline]`

Definition at line 34 of file `Button.h`.

#### 5.1.3.2 `void Button::setText ( string txt )` `[inline]`

Definition at line 36 of file `Button.h`.

### 5.1.4 Member Data Documentation

#### 5.1.4.1 `ImageElement* Button::_imageE` `[protected]`

Definition at line 40 of file `Button.h`.

#### 5.1.4.2 `TextElement* Button::_textE` `[protected]`

Definition at line 39 of file `Button.h`.

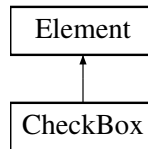
The documentation for this class was generated from the following file:

- `button/`[Button.h](#)

## 5.2 CheckBox Class Reference

```
#include <CheckBox.h>
```

Inheritance diagram for CheckBox:



### Public Member Functions

- [CheckBox](#) ()
- [CheckBox](#) (unsigned int x, unsigned int y)

### Additional Inherited Members

#### 5.2.1 Detailed Description

Definition at line 12 of file `CheckBox.h`.

#### 5.2.2 Constructor & Destructor Documentation

5.2.2.1 `CheckBox::CheckBox ( )` [`inline`]

Definition at line 14 of file `CheckBox.h`.

5.2.2.2 `CheckBox::CheckBox ( unsigned int x, unsigned int y )` [`inline`]

Definition at line 15 of file `CheckBox.h`.

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

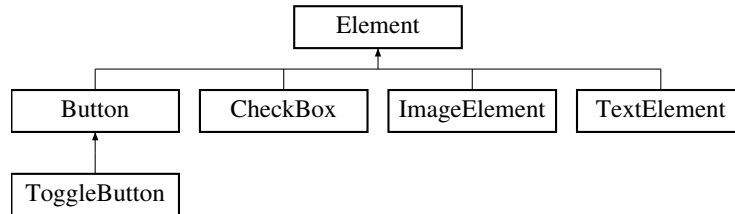
- checkbox/[CheckBox.h](#)

## 5.3 Element Class Reference

The base class that all GUI elements derive from.

```
#include <Element.h>
```

Inheritance diagram for Element:



## Public Member Functions

- [Element](#) ()  
*Default Constructor.*
- [Element](#) (int x, int y)  
*Construct with position.*
- [Element](#) (int x, int y, int xs, int ys)  
*Construct with position and size.*
- virtual [~Element](#) ()  
*Destructor.*
- virtual void [clearResult](#) ()  
*Clears the result image to a color (black is default).*
- [Image \\*](#) [render](#) ()  
*Renders the element and its children recursively.*
- void [registerCallback](#) (void(\*func)(void \*))  
*Registers a callback function for the element.*
- void [mouseInput](#) (int x, int y)  
*Test if element clicked by mouse.*
- void [addChild](#) ([Element](#) \*child)  
*Add a child element to the current element.*
- void [setX](#) (unsigned int x)  
*Set the x position of the element.*
- void [setY](#) (unsigned int y)  
*Set the y position of the element.*
- void [setZ](#) (float z)  
*Set the z position (z index) of the element.*
- unsigned int [getId](#) ()  
*Retrieve the current element's unique id.*
- void [setWidth](#) (unsigned int width)  
*Set the element's width.*
- void [setHeight](#) (unsigned int height)  
*Set the element's height.*
- void [setDirty](#) (bool dirty)  
*Set the dirty flag. Causes the element re-render.*
- bool [operator<](#) (const [Element](#) &other)  
*Less than operator so [Element](#) objects may be sorted.*

## Protected Attributes

- unsigned int `_xCoord`
- unsigned int `_yCoord`
- unsigned int `_width`
- unsigned int `_height`
- `Image * _result`

### 5.3.1 Detailed Description

The base class that all GUI elements derive from.

This class provides a standard interface that is required for element traversal, rendering, and events.

Definition at line 22 of file `Element.h`.

### 5.3.2 Constructor & Destructor Documentation

#### 5.3.2.1 `Element::Element ( )`

Default Constructor.

Creates an element positioned at (0,0) with dimensions (0,0).

Definition at line 10 of file `Element.cpp`.

#### 5.3.2.2 `Element::Element ( int x, int y )`

Construct with position.

Creates an element positioned at (x,y) with dimensions (0,0).

Definition at line 24 of file `Element.cpp`.

#### 5.3.2.3 `Element::Element ( int x, int y, int xs, int ys )`

Construct with position and size.

Creates an element positioned at (x, y) with dimensions (xs, ys).

Definition at line 39 of file `Element.cpp`.

#### 5.3.2.4 `Element::~Element ( )` `[virtual]`

Destructor.

Deletes the pointers for the result image and the clear image (background).

Definition at line 54 of file `Element.cpp`.

### 5.3.3 Member Function Documentation

#### 5.3.3.1 void Element::addChild ( Element \* child )

Add a child element to the current element.

Add a child element to the set of children elements. The function accepts a pointer to an [Element](#), which must remain in scope as long as the parent. Calls STL sort on the children, organizing by z-index (z position).

Definition at line 95 of file Element.cpp.

#### 5.3.3.2 void Element::clearResult ( ) [virtual]

Clears the result image to a color (black is default).

Renders the background of the element, namely element contents. For generic Elements, it blits a solid color (black) image to the element's result image. For content elements ([TextElement](#) and [ImageElement](#)) it will blit the stored image (for image elements) or resulting image from rendering the text (for text elements) before rendering the children.

Reimplemented in [ImageElement](#), and [TextElement](#).

Definition at line 65 of file Element.cpp.

#### 5.3.3.3 unsigned int Element::getId ( ) [inline]

Retrieve the current element's unique id.

Definition at line 46 of file Element.h.

#### 5.3.3.4 void Element::mouseInput ( int x, int y )

Test if element clicked by mouse.

Tests if the mouse click at (x, y) is within the element.

Definition at line 70 of file Element.cpp.

#### 5.3.3.5 bool Element::operator< ( const Element & other )

Less than operator so [Element](#) objects may be sorted.

Less than operator which compares two elements based solely on their z-index (z position).

Definition at line 136 of file Element.cpp.

#### 5.3.3.6 void Element::registerCallback ( void(\*)(void \*) func )

Registers a callback function for the element.

Register a callback function, accepts a function pointer to a function which takes one argument of void\*.

Definition at line 87 of file Element.cpp.

#### 5.3.3.7 Image \* Element::render ( )

Renders the element and its children recursively.

Clears the result image of past renders with [clearResult\(\)](#), filling it with either a color or the element's content, then renders each child in order of z-index (z position). Once all of the children have been rendered, it is blitted to the result image. After all children are rendered and blitted, the result image is returned.

Definition at line 110 of file Element.cpp.

#### 5.3.3.8 void Element::setDirty ( bool *dirty* ) [inline]

Set the dirty flag. Causes the element re-render.

Definition at line 52 of file Element.h.

#### 5.3.3.9 void Element::setHeight ( unsigned int *height* ) [inline]

Set the element's height.

Definition at line 50 of file Element.h.

#### 5.3.3.10 void Element::setWidth ( unsigned int *width* ) [inline]

Set the element's width.

Definition at line 48 of file Element.h.

#### 5.3.3.11 void Element::setX ( unsigned int *x* ) [inline]

Set the x position of the element.

Definition at line 40 of file Element.h.

#### 5.3.3.12 void Element::setY ( unsigned int *y* ) [inline]

Set the y position of the element.

Definition at line 42 of file Element.h.

#### 5.3.3.13 void Element::setZ ( float *z* ) [inline]

Set the z position (z index) of the element.

Definition at line 44 of file Element.h.

### 5.3.4 Member Data Documentation

#### 5.3.4.1 unsigned int Element::\_height [protected]

The element's height.

Definition at line 65 of file Element.h.

#### 5.3.4.2 `Image* Element::_result` [protected]

The resulting image for the element to be blitted to a parent element or rendered on a surface

Definition at line 69 of file `Element.h`.

#### 5.3.4.3 `unsigned int Element::_width` [protected]

The element's width.

Definition at line 63 of file `Element.h`.

#### 5.3.4.4 `unsigned int Element::_xCoord` [protected]

The x position of the element in the parent.

Definition at line 59 of file `Element.h`.

#### 5.3.4.5 `unsigned int Element::_yCoord` [protected]

The y position of the element in the parent.

Definition at line 61 of file `Element.h`.

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

- element/[Element.h](#)
- element/[Element.cpp](#)

## 5.4 Image Class Reference

```
#include <Image.h>
```

### Public Member Functions

- [Image](#) ()
- [Image](#) (unsigned int [width](#), unsigned int [height](#))
- [Image](#) (unsigned int [width](#), unsigned int [height](#), const [Pixel](#) &p)
- [Image](#) (unsigned int [width](#), unsigned int [height](#), unsigned char \*data)
- [Image](#) ([Image](#) &img)
- [~Image](#) ()
- unsigned int [width](#) () const
- unsigned int [height](#) () const
- void [set](#) (unsigned int x, unsigned int y, const [Pixel](#) &color)
- void [set](#) (unsigned int x, unsigned int y, unsigned int [width](#), unsigned int [height](#), const [Pixel](#) &color)
- const [Pixel](#) & [get](#) (unsigned int x, unsigned int y) const
- [Pixel](#) \* [getPixels](#) ()
- void [blit](#) ([Image](#) &dest, unsigned int xSource, unsigned int ySource, unsigned int xDest, unsigned int yDest, unsigned int [width](#), unsigned int [height](#)) const



## Friends

- `std::ostream & operator<< (std::ostream &out, const Image &img)`

### 5.4.1 Detailed Description

Definition at line 8 of file Image.h.

### 5.4.2 Constructor & Destructor Documentation

#### 5.4.2.1 Image::Image ( )

Definition at line 8 of file Image.cpp.

#### 5.4.2.2 Image::Image ( unsigned int *width*, unsigned int *height* )

Definition at line 14 of file Image.cpp.

#### 5.4.2.3 Image::Image ( unsigned int *width*, unsigned int *height*, const Pixel & *p* )

Definition at line 22 of file Image.cpp.

#### 5.4.2.4 Image::Image ( unsigned int *width*, unsigned int *height*, unsigned char \* *data* )

Definition at line 41 of file Image.cpp.

#### 5.4.2.5 Image::Image ( Image & *img* )

Definition at line 31 of file Image.cpp.

#### 5.4.2.6 Image::~Image ( )

Definition at line 54 of file Image.cpp.

### 5.4.3 Member Function Documentation

#### 5.4.3.1 void Image::blit ( Image & *dest*, unsigned int *xSource*, unsigned int *ySource*, unsigned int *xDest*, unsigned int *yDest*, unsigned int *width*, unsigned int *height* ) const

Definition at line 115 of file Image.cpp.

#### 5.4.3.2 const Pixel & Image::get ( unsigned int *x*, unsigned int *y* ) const

Definition at line 102 of file Image.cpp.

#### 5.4.3.3 Pixel\* Image::getPixels ( ) [inline]

Definition at line 22 of file Image.h.

#### 5.4.3.4 unsigned int Image::height ( ) const

Definition at line 63 of file Image.cpp.

#### 5.4.3.5 void Image::set ( unsigned int x, unsigned int y, const Pixel & color )

Definition at line 66 of file Image.cpp.

#### 5.4.3.6 void Image::set ( unsigned int x, unsigned int y, unsigned int width, unsigned int height, const Pixel & color )

Definition at line 80 of file Image.cpp.

#### 5.4.3.7 unsigned int Image::width ( ) const

Definition at line 60 of file Image.cpp.

### 5.4.4 Friends And Related Function Documentation

#### 5.4.4.1 std::ostream& operator<< ( std::ostream & out, const Image & img ) [friend]

Definition at line 147 of file Image.cpp.

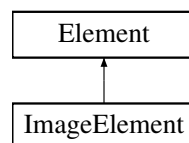
The documentation for this class was generated from the following files:

- [image/Image.h](#)
- [image/Image.cpp](#)

## 5.5 ImageElement Class Reference

```
#include <ImageElement.h>
```

Inheritance diagram for ImageElement:



### Public Member Functions

- [ImageElement \( \)](#)
- [ImageElement \(Image &img\)](#)

- [ImageElement](#) (unsigned int x, unsigned int y)
- [ImageElement](#) (unsigned int x, unsigned int y, unsigned int width, unsigned int height)
- [ImageElement](#) (unsigned int x, unsigned int y, unsigned int width, unsigned int height, [Image](#) &img)
- void [clearResult](#) ()

*Clears the result image to a color (black is default).*

## Additional Inherited Members

### 5.5.1 Detailed Description

Definition at line 9 of file ImageElement.h.

### 5.5.2 Constructor & Destructor Documentation

#### 5.5.2.1 ImageElement::ImageElement ( ) [inline]

Definition at line 12 of file ImageElement.h.

#### 5.5.2.2 ImageElement::ImageElement ( Image & img ) [inline]

Definition at line 13 of file ImageElement.h.

#### 5.5.2.3 ImageElement::ImageElement ( unsigned int x, unsigned int y ) [inline]

Definition at line 14 of file ImageElement.h.

#### 5.5.2.4 ImageElement::ImageElement ( unsigned int x, unsigned int y, unsigned int width, unsigned int height ) [inline]

Definition at line 17 of file ImageElement.h.

#### 5.5.2.5 ImageElement::ImageElement ( unsigned int x, unsigned int y, unsigned int width, unsigned int height, Image & img ) [inline]

Definition at line 19 of file ImageElement.h.

### 5.5.3 Member Function Documentation

#### 5.5.3.1 void ImageElement::clearResult ( ) [inline], [virtual]

Clears the result image to a color (black is default).

Renders the background of the element, namely element contents. For generic Elements, it blits a solid color (black) image to the element's result image. For content elements ([TextElement](#) and [ImageElement](#)) it will blit the stored image (for image elements) or resulting image from rendering the text (for text elements) before rendering the children.

Reimplemented from [Element](#).

Definition at line 22 of file ImageElement.h.

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

- [image/ImageElement.h](#)

## 5.6 Pixel Class Reference

```
#include <Pixel.h>
```

### Public Member Functions

- [Pixel](#) ()
- [Pixel](#) (int R, int G, int B)
- [Pixel](#) (int R, int G, int B, int A)
- [Pixel](#) ([Pixel](#) &p)
- void [setRGB](#) (int R, int G, int B)
- void [setRGBA](#) (int R, int G, int B, int A)
- int [getR](#) () const
- int [getG](#) () const
- int [getB](#) () const
- int [getA](#) () const

### Friends

- std::ostream & [operator<<](#) (std::ostream &out, const [Pixel](#) &p)

### 5.6.1 Detailed Description

Definition at line 6 of file Pixel.h.

### 5.6.2 Constructor & Destructor Documentation

#### 5.6.2.1 [Pixel::Pixel](#) ( )

Definition at line 3 of file Pixel.cpp.

#### 5.6.2.2 [Pixel::Pixel](#) ( int *R*, int *G*, int *B* )

Definition at line 4 of file Pixel.cpp.

#### 5.6.2.3 [Pixel::Pixel](#) ( int *R*, int *G*, int *B*, int *A* )

Definition at line 7 of file Pixel.cpp.

#### 5.6.2.4 [Pixel::Pixel](#) ( [Pixel](#) & *p* )

Definition at line 11 of file Pixel.cpp.

### 5.6.3 Member Function Documentation

#### 5.6.3.1 `int Pixel::getA ( ) const`

Definition at line 34 of file Pixel.cpp.

#### 5.6.3.2 `int Pixel::getB ( ) const`

Definition at line 31 of file Pixel.cpp.

#### 5.6.3.3 `int Pixel::getG ( ) const`

Definition at line 28 of file Pixel.cpp.

#### 5.6.3.4 `int Pixel::getR ( ) const`

Definition at line 25 of file Pixel.cpp.

#### 5.6.3.5 `void Pixel::setRGB ( int R, int G, int B )`

Definition at line 19 of file Pixel.cpp.

#### 5.6.3.6 `void Pixel::setRGBA ( int R, int G, int B, int A )`

Definition at line 22 of file Pixel.cpp.

### 5.6.4 Friends And Related Function Documentation

#### 5.6.4.1 `std::ostream& operator<< ( std::ostream & out, const Pixel & p )` [*friend*]

Definition at line 37 of file Pixel.cpp.

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

- [image/Pixel.h](#)
- [image/Pixel.cpp](#)

## 5.7 Text Class Reference

```
#include <Text.h>
```

### Public Member Functions

- [Text](#) ()
- [Text](#) ([Text](#) &txt)
- [Text](#) (int w, int h, int size)
- [Text](#) (int w, int h, int size, string c)

- [~Text](#) ()
- void [setText](#) (string c)
- [Image](#) \* [getImage](#) ()
- string [getText](#) (void)

### 5.7.1 Detailed Description

Definition at line 12 of file Text.h.

### 5.7.2 Constructor & Destructor Documentation

#### 5.7.2.1 Text::Text ( )

Definition at line 14 of file Text.cpp.

#### 5.7.2.2 Text::Text ( Text & txt )

Definition at line 41 of file Text.cpp.

#### 5.7.2.3 Text::Text ( int w, int h, int size )

Definition at line 19 of file Text.cpp.

#### 5.7.2.4 Text::Text ( int w, int h, int size, string c )

Definition at line 24 of file Text.cpp.

#### 5.7.2.5 Text::~~Text ( )

Definition at line 46 of file Text.cpp.

### 5.7.3 Member Function Documentation

#### 5.7.3.1 Image \* Text::getImage ( )

Definition at line 62 of file Text.cpp.

#### 5.7.3.2 string Text::getText ( void ) `[inline]`

Definition at line 22 of file Text.h.

#### 5.7.3.3 void Text::setText ( string c )

Definition at line 56 of file Text.cpp.

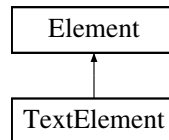
The documentation for this class was generated from the following files:

- [text/Text.h](#)
- [text/Text.cpp](#)

## 5.8 TextElement Class Reference

```
#include <TextElement.h>
```

Inheritance diagram for TextElement:



### Public Member Functions

- [TextElement](#) (unsigned int x, unsigned int y)
- [TextElement](#) (unsigned int x, unsigned int y, unsigned int width, unsigned int height)
- [TextElement](#) (unsigned int x, unsigned int y, unsigned int width, unsigned int height, int size, string txt)
- [~TextElement](#) ()
- void [setText](#) (string txt)
- void [clearResult](#) ()

*Clears the result image to a color (black is default).*

### Additional Inherited Members

#### 5.8.1 Detailed Description

Definition at line 9 of file TextElement.h.

#### 5.8.2 Constructor & Destructor Documentation

##### 5.8.2.1 TextElement::TextElement ( unsigned int x, unsigned int y )

Definition at line 10 of file TextElement.cpp.

##### 5.8.2.2 TextElement::TextElement ( unsigned int x, unsigned int y, unsigned int width, unsigned int height )

Definition at line 15 of file TextElement.cpp.

##### 5.8.2.3 TextElement::TextElement ( unsigned int x, unsigned int y, unsigned int width, unsigned int height, int size, string txt )

Definition at line 20 of file TextElement.cpp.

#### 5.8.2.4 `TextElement::~~TextElement ( )`

Definition at line 28 of file `TextElement.cpp`.

### 5.8.3 Member Function Documentation

#### 5.8.3.1 `void TextElement::clearResult ( )` `[virtual]`

Clears the result image to a color (black is default).

Renders the background of the element, namely element contents. For generic Elements, it blits a solid color (black) image to the element's result image. For content elements ([TextElement](#) and [ImageElement](#)) it will blit the stored image (for image elements) or resulting image from rendering the text (for text elements) before rendering the children.

Reimplemented from [Element](#).

Definition at line 40 of file `TextElement.cpp`.

#### 5.8.3.2 `void TextElement::setText ( string txt )`

Definition at line 35 of file `TextElement.cpp`.

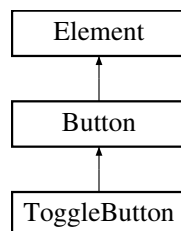
The documentation for this class was generated from the following files:

- [text/TextElement.h](#)
- [text/TextElement.cpp](#)

## 5.9 ToggleButton Class Reference

```
#include <ToggleButton.h>
```

Inheritance diagram for `ToggleButton`:



### Public Member Functions

- [ToggleButton \( \)](#)
- [ToggleButton \(int x, int y\)](#)
- [ToggleButton \(int x, int y, int w, int h\)](#)
- [ToggleButton \(int x, int y, int w, int h, string txt\)](#)
- [ToggleButton \(int x, int y, int w, int h, string content, int size, \[ImageElement\]\(#\) \\*i\)](#)



## Additional Inherited Members

### 5.9.1 Detailed Description

Definition at line 12 of file `ToggleButton.h`.

### 5.9.2 Constructor & Destructor Documentation

#### 5.9.2.1 `ToggleButton::ToggleButton ( )` [inline]

Definition at line 14 of file `ToggleButton.h`.

#### 5.9.2.2 `ToggleButton::ToggleButton ( int x, int y )` [inline]

Definition at line 15 of file `ToggleButton.h`.

#### 5.9.2.3 `ToggleButton::ToggleButton ( int x, int y, int w, int h )` [inline]

Definition at line 16 of file `ToggleButton.h`.

#### 5.9.2.4 `ToggleButton::ToggleButton ( int x, int y, int w, int h, string txt )` [inline]

Definition at line 19 of file `ToggleButton.h`.

#### 5.9.2.5 `ToggleButton::ToggleButton ( int x, int y, int w, int h, string content, int size, ImageElement * i )` [inline]

Definition at line 23 of file `ToggleButton.h`.

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

- [togglebutton/ToggleButton.h](#)



## Chapter 6

# File Documentation

### 6.1 button/Button.h File Reference

```
#include <string>
#include "../image/Image.h"
#include "../text/Text.h"
#include "../element/Element.h"
#include "../image/ImageElement.h"
#include "../text/TextElement.h"
```

#### Classes

- class [Button](#)

### 6.2 button/main.cpp File Reference

```
#include <string>
#include "../image/Image.h"
#include "../image/Pixel.h"
#include "../text/Text.h"
#include "Button.h"
```

#### Functions

- int [main](#) ()

#### 6.2.1 Function Documentation

##### 6.2.1.1 int main ( )

Definition at line 9 of file main.cpp.

## 6.3 main.cpp File Reference

```
#include <GL/glwf.h>
#include <FreeImage.h>
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include "main.h"
#include "element/Element.h"
#include "text/TextElement.h"
#include "image/ImageElement.h"
#include "togglebutton/ToggleButton.h"
#include "button/Button.h"
```

### Functions

- int [main](#) ()
- int [loadGuiTexture](#) (string textureString)
- void [shutDown](#) (int returnCode)
- int [closeWindow](#) (void)
- void [init](#) (void)
- void [mainLoop](#) (void)
- void [draw](#) (void)
- void GLFWCALL [windowResize](#) (int width, int height)
- void GLFWCALL [mouseClicked](#) (int mButton, int clicked)
- void [buttonClicked](#) (void \*e)

### Variables

- GLuint [texture](#)

#### 6.3.1 Function Documentation

##### 6.3.1.1 void buttonClicked ( void \* e )

Definition at line 209 of file main.cpp.

##### 6.3.1.2 int closeWindow ( void )

Definition at line 110 of file main.cpp.

##### 6.3.1.3 void draw ( void )

Definition at line 169 of file main.cpp.

#### 6.3.1.4 void init ( void )

Initializes a glfw window for use in the demo

Definition at line 115 of file main.cpp.

#### 6.3.1.5 int loadGuiTexture ( string *textureString* )

Definition at line 30 of file main.cpp.

#### 6.3.1.6 int main ( )

Definition at line 23 of file main.cpp.

#### 6.3.1.7 void mainLoop ( void )

the main event loop for the demo

Definition at line 153 of file main.cpp.

#### 6.3.1.8 void GLFWCALL mouseClicked ( int *mButton*, int *clicked* )

Definition at line 198 of file main.cpp.

#### 6.3.1.9 void shutDown ( int *returnCode* )

shuts down glfw and exits the program with a return code

Definition at line 101 of file main.cpp.

#### 6.3.1.10 void GLFWCALL windowResize ( int *width*, int *height* )

Definition at line 193 of file main.cpp.

### 6.3.2 Variable Documentation

#### 6.3.2.1 GLuint texture

Definition at line 19 of file main.cpp.

## 6.4 text/main.cpp File Reference

```
#include <iostream>
#include <string>
#include "Text.h"
#include "../image/Image.h"
```

## Functions

- int [main](#) ()

### 6.4.1 Function Documentation

#### 6.4.1.1 int main ( )

Definition at line 7 of file main.cpp.

## 6.5 checkbox/CheckBox.h File Reference

```
#include "Element.h"
#include "Text.h"
#include "Image.h"
#include "ImageElement.h"
#include "TextElement.h"
```

## Classes

- class [CheckBox](#)

## 6.6 element/Element.cpp File Reference

```
#include "Element.h"
#include <iostream>
#include <algorithm>
#include <stdio.h>
#include "../image/Image.h"
```

## 6.7 element/Element.h File Reference

```
#include <vector>
#include <algorithm>
#include "../image/Image.h"
```

## Classes

- class [Element](#)

*The base class that all GUI elements derive from.*

### 6.7.1 Detailed Description

This file contains the [Element](#) class.

Definition in file [Element.h](#).

## 6.8 element/Main.cpp File Reference

```
#include <iostream>
#include <vector>
#include "Element.h"
```

### Functions

- [int main\(\)](#)

### 6.8.1 Function Documentation

#### 6.8.1.1 [int main\(\)](#)

Definition at line 6 of file Main.cpp.

## 6.9 image/Main.cpp File Reference

```
#include <iostream>
#include "Image.h"
```

### Functions

- [int main\(\)](#)

### 6.9.1 Function Documentation

#### 6.9.1.1 [int main\(\)](#)

Definition at line 5 of file Main.cpp.

## 6.10 togglebutton/Main.cpp File Reference

```
#include <string>
#include "../image/Image.h"
#include "../image/Pixel.h"
#include "../text/Text.h"
#include "Button.h"
```

## Functions

- int [main](#) ()

### 6.10.1 Function Documentation

#### 6.10.1.1 int main ( )

Definition at line 9 of file Main.cpp.

## 6.11 image/Image.cpp File Reference

```
#include "Image.h"  
#include <algorithm>
```

## Functions

- std::ostream & [operator<<](#) (std::ostream &out, const [Image](#) &img)

## Variables

- [Pixel ERRORPIXEL](#)

### 6.11.1 Function Documentation

#### 6.11.1.1 std::ostream& operator<< ( std::ostream & *out*, const Image & *img* )

Definition at line 147 of file Image.cpp.

### 6.11.2 Variable Documentation

#### 6.11.2.1 Pixel ERRORPIXEL

Definition at line 101 of file Image.cpp.

## 6.12 image/Image.h File Reference

```
#include <iostream>  
#include "Pixel.h"  
#include "Image.h"
```



## Classes

- class [Image](#)

## 6.13 image/ImageElement.h File Reference

```
#include <iostream>
#include "Pixel.h"
#include "Image.h"
#include "../element/Element.h"
```

## Classes

- class [ImageElement](#)

## 6.14 image/Pixel.cpp File Reference

```
#include "Pixel.h"
```

## Functions

- `std::ostream & operator<< (std::ostream &out, const Pixel &p)`

### 6.14.1 Function Documentation

6.14.1.1 `std::ostream& operator<< ( std::ostream & out, const Pixel & p )`

Definition at line 37 of file Pixel.cpp.

## 6.15 image/Pixel.h File Reference

```
#include <iostream>
```

## Classes

- class [Pixel](#)

## 6.16 main.h File Reference

```
#include "element/Element.h"
```

## Functions

- void `shutDown` (int `returnCode`)
- void `init` (void)
- void `mainLoop` (void)
- void `draw` (void)
- void GLFWCALL `windowResize` (int `width`, int `height`)
- int `loadGuiTexture` (string `texture`)
- void GLFWCALL `mouseClicked` (int `mButton`, int `clicked`)
- void `buttonClicked` (void \*`e`)

### 6.16.1 Function Documentation

#### 6.16.1.1 void `buttonClicked` ( void \* *e* )

Definition at line 209 of file `main.cpp`.

#### 6.16.1.2 void `draw` ( void )

Definition at line 169 of file `main.cpp`.

#### 6.16.1.3 void `init` ( void )

Initializes a glfw window for use in the demo

Definition at line 115 of file `main.cpp`.

#### 6.16.1.4 int `loadGuiTexture` ( string *texture* )

Definition at line 30 of file `main.cpp`.

#### 6.16.1.5 void `mainLoop` ( void )

the main event loop for the demo

Definition at line 153 of file `main.cpp`.

#### 6.16.1.6 void GLFWCALL `mouseClicked` ( int *mButton*, int *clicked* )

Definition at line 198 of file `main.cpp`.

#### 6.16.1.7 void `shutDown` ( int *returnCode* )

shuts down glfw and exits the program with a return code

Definition at line 101 of file `main.cpp`.

#### 6.16.1.8 void GLFWCALL windowResize ( int *width*, int *height* )

Definition at line 193 of file main.cpp.

## 6.17 README.md File Reference

## 6.18 text/Text.cpp File Reference

```
#include "Text.h"
#include <string>
#include <ft2build.h>
#include <iostream>
#include "../image/Image.h"
#include "../image/Pixel.h"
#include <stdio.h>
#include <math.h>
```

## 6.19 text/Text.h File Reference

```
#include <string>
#include <ft2build.h>
#include "../image/Image.h"
```

### Classes

- class [Text](#)

## 6.20 text/TextElement.cpp File Reference

```
#include <string>
#include "Text.h"
#include "TextElement.h"
#include "../image/Image.h"
#include "../element/Element.h"
```

## 6.21 text/TextElement.h File Reference

```
#include <string>
#include "Text.h"
#include "../image/Image.h"
#include "../element/Element.h"
```

## Classes

- class [TextElement](#)

## 6.22 togglebutton/ToggleButton.h File Reference

```
#include <string>
#include "../image/ImageElement.h"
#include "../text/TextElement.h"
#include "../text/Text.h"
#include "../button/Button.h"
```

## Classes

- class [ToggleButton](#)

# Index

- ~Element
  - Element, [13](#)
- ~Image
  - Image, [17](#)
- ~Text
  - Text, [22](#)
- ~TextElement
  - TextElement, [23](#)
- \_height
  - Element, [15](#)
- \_imageE
  - Button, [10](#)
- \_result
  - Element, [15](#)
- \_textE
  - Button, [10](#)
- \_width
  - Element, [16](#)
- \_xCoord
  - Element, [16](#)
- \_yCoord
  - Element, [16](#)
- addChild
  - Element, [13](#)
- blit
  - Image, [17](#)
- Button, [9](#)
  - \_imageE, [10](#)
  - \_textE, [10](#)
  - Button, [10](#)
  - setBgImg, [10](#)
  - setText, [10](#)
- button/Button.h, [27](#)
- button/main.cpp, [27](#)
  - main, [27](#)
- buttonClicked
  - main.cpp, [28](#)
  - main.h, [34](#)
- CheckBox, [11](#)
  - CheckBox, [11](#)
  - CheckBox, [11](#)
- checkbox/CheckBox.h, [30](#)
- clearResult
  - Element, [14](#)
  - ImageElement, [19](#)
  - TextElement, [24](#)
- closeWindow
  - main.cpp, [28](#)
- draw
  - main.cpp, [28](#)
  - main.h, [34](#)
- ERRORPIXEL
  - Image.cpp, [32](#)
- Element, [11](#)
  - ~Element, [13](#)
  - \_height, [15](#)
  - \_result, [15](#)
  - \_width, [16](#)
  - \_xCoord, [16](#)
  - \_yCoord, [16](#)
  - addChild, [13](#)
  - clearResult, [14](#)
  - Element, [13](#)
  - getId, [14](#)
  - mouseInput, [14](#)
  - operator<, [14](#)
  - registerCallback, [14](#)
  - render, [14](#)
  - setDirty, [15](#)
  - setHeight, [15](#)
  - setWidth, [15](#)
  - setX, [15](#)
  - setY, [15](#)
  - setZ, [15](#)
- element/Element.cpp, [30](#)
- element/Element.h, [30](#)
- element/Main.cpp, [31](#)
  - main, [31](#)
- get
  - Image, [17](#)
- getA
  - Pixel, [21](#)
- getB
  - Pixel, [21](#)
- getG
  - Pixel, [21](#)

- getId
  - Element, 14
- getImage
  - Text, 22
- getPixels
  - Image, 17
- getR
  - Pixel, 21
- getText
  - Text, 22
- height
  - Image, 18
- Image, 16
  - ~Image, 17
  - blit, 17
  - get, 17
  - getPixels, 17
  - height, 18
  - Image, 17
  - operator<<, 18
  - set, 18
  - width, 18
- Image.cpp
  - ERRORPIXEL, 32
  - operator<<, 32
- image/Image.cpp, 32
- image/Image.h, 32
- image/ImageElement.h, 33
- image/Main.cpp, 31
  - main, 31
- image/Pixel.cpp, 33
- image/Pixel.h, 33
- ImageElement, 18
  - clearResult, 19
  - ImageElement, 19
  - ImageElement, 19
- init
  - main.cpp, 28
  - main.h, 34
- loadGuiTexture
  - main.cpp, 29
  - main.h, 34
- main
  - button/main.cpp, 27
  - element/Main.cpp, 31
  - image/Main.cpp, 31
  - main.cpp, 29
  - text/main.cpp, 30
  - togglebutton/Main.cpp, 32
- main.cpp, 28
  - buttonClicked, 28
  - closeWindow, 28
  - draw, 28
  - init, 28
  - loadGuiTexture, 29
  - main, 29
  - mainLoop, 29
  - mouseClicked, 29
  - shutDown, 29
  - texture, 29
  - windowResize, 29
- main.h, 33
  - buttonClicked, 34
  - draw, 34
  - init, 34
  - loadGuiTexture, 34
  - mainLoop, 34
  - mouseClicked, 34
  - shutDown, 34
  - windowResize, 34
- mainLoop
  - main.cpp, 29
  - main.h, 34
- mouseClicked
  - main.cpp, 29
  - main.h, 34
- mouseInput
  - Element, 14
- operator<
  - Element, 14
- operator<<
  - Image, 18
  - Image.cpp, 32
  - Pixel, 21
  - Pixel.cpp, 33
- Pixel, 20
  - getA, 21
  - getB, 21
  - getG, 21
  - getR, 21
  - operator<<, 21
  - Pixel, 20
  - setRGB, 21
  - setRGBA, 21
- Pixel.cpp
  - operator<<, 33
- README.md, 35
- registerCallback
  - Element, 14
- render
  - Element, 14
- set

- Image, [18](#)
- setBgImg
  - Button, [10](#)
- setDirty
  - Element, [15](#)
- setHeight
  - Element, [15](#)
- setRGB
  - Pixel, [21](#)
- setRGBA
  - Pixel, [21](#)
- setText
  - Button, [10](#)
  - Text, [22](#)
  - TextElement, [24](#)
- setWidth
  - Element, [15](#)
- setX
  - Element, [15](#)
- setY
  - Element, [15](#)
- setZ
  - Element, [15](#)
- shutDown
  - main.cpp, [29](#)
  - main.h, [34](#)
- Text, [21](#)
  - ~Text, [22](#)
  - getImage, [22](#)
  - getText, [22](#)
  - setText, [22](#)
  - Text, [22](#)
- text/Text.cpp, [35](#)
- text/Text.h, [35](#)
- text/TextElement.cpp, [35](#)
- text/TextElement.h, [35](#)
- text/main.cpp, [29](#)
  - main, [30](#)
- TextElement, [23](#)
  - ~TextElement, [23](#)
  - clearResult, [24](#)
  - setText, [24](#)
  - TextElement, [23](#)
  - TextElement, [23](#)
- texture
  - main.cpp, [29](#)
- ToggleButton, [24](#)
  - ToggleButton, [25](#)
  - ToggleButton, [25](#)
- togglebutton/Main.cpp, [31](#)
  - main, [32](#)
- togglebutton/ToggleButton.h, [36](#)
- width
  - Image, [18](#)
  - windowResize
    - main.cpp, [29](#)
    - main.h, [34](#)