OpenGui

Generated by Doxygen 1.8.2

Thu Nov 1 2012 01:52:35

Contents

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Class Documentation

3.1 Element Class Reference

The base class that all GUI elements derive from.

`#include <Element.h>`

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)
- void addChild (Element ∗child)
- 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

3.1.1 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.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 Element::Element ( )

Default Constructor.

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

Definition at line 10 of file Element.cpp.

3.1.2.2 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.

3.1.2.3 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.

3.1.2.4 Element::∼Element ( ) `[virtual]`

Destructor.

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

Definition at line 54 of file Element.cpp.

3.1.3 Member Function Documentation

3.1.3.1 void Element::addChild (**Element** ∗ *child*)

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.

3.1.3.2 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.

Definition at line 65 of file Element.cpp.

3.1.3.3 unsigned int Element::getId ( ) `[inline]`

Retrieve the current element's unique id.

Definition at line 46 of file Element.h.

3.1.3.4 void Element::mouseInput (int *x,* int *y*)

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

Definition at line 70 of file Element.cpp.

3.1.3.5 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.

3.1.3.6 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.

3.1.3.7 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.

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

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

Definition at line 52 of file Element.h.

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

Set the element's height.

Definition at line 50 of file Element.h.

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

Set the element's width.

Definition at line 48 of file Element.h.

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

Set the x position of the element.

Definition at line 40 of file Element.h.

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

Set the y position of the element.

Definition at line 42 of file Element.h.

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

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

Definition at line 44 of file Element.h.

3.1.4 Member Data Documentation

3.1.4.1 unsigned int Element::_height [protected]

The element's height.

Definition at line 65 of file Element.h.

3.1.4.2 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.

3.1.4.3 unsigned int Element::_width [protected]

The element's width.

Definition at line 63 of file Element.h.

3.1.4.4 unsigned int Element::_xCoord [protected]

The x position of the element in the parent.

Definition at line 59 of file Element.h.

3.1.4.5 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.h
- Element.cpp

Chapter 4

File Documentation

4.1 Element.cpp File Reference

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

4.2 Element.h File Reference

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

Classes

- class Element

  The base class that all GUI elements derive from.

4.2.1 Detailed Description

This file contains the Element class.

Definition in file Element.h.

4.3 Main.cpp File Reference

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

Functions

- int main ()

4.3.1 Function Documentation

4.3.1.1 int main ( )

Definition at line 6 of file Main.cpp.

Index