OpenGLot

0.1

Generated by Doxygen 1.5.8

Fri Mar 27 17:52:25 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 List

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

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Namespace Documentation

4.1 glot Namespace Reference

Classes

- class **color**
- class **curve**
- class **function**
- class **grapher**
- class **point**

Enumerations

- enum **display_opt** {
  AXES_OFF = 0, **GRID_OFF** = 0, **X_LIN** = 0, **Y_LIN** = 0,
  AXES_ON = 1, **GRID_ON** = 2, **X_LOG** = 4, **Y_LOG** = 8 }
  Enumeration for display options.

- enum **keyboard_opt** {
  ZOOM_KEYS_OFF = 0, **AXES_KEYS_OFF** = 0, **GRID_KEYS_OFF** = 0,
  ZOOM_KEYS_ON = 1,
  AXES_KEYS_ON = 2, **GRID_KEYS_ON** = 4 }
  Enumeration for keyboard action options.

4.1.1 Enumeration Type Documentation

4.1.1.1 enum glot::display_opt

Enumeration for display options.

Bitwise or these to set the display options

Enumerator:

AXES_OFF

GRID_OFF
X_LIN
Y_LIN
AXES_ON
GRID_ON
X_LOG
Y_LOG

4.1.1.2 enum glot::keyboard_opt

Enumeration for keyboard action options.

Bitwise or these to set the keyboard action options

Enumerator:

ZOOM_KEYS_OFF
AXES_KEYS_OFF
GRID_KEYS_OFF
ZOOM_KEYS_ON
AXES_KEYS_ON
GRID_KEYS_ON

Chapter 5

Class Documentation

5.1 glot::color Class Reference

`#include <color.h>`

Public Member Functions

- **color** (double red=0, double green=0, double blue=0, double alpha=1)
  Constructor.

Public Attributes

- double **r**
  *The red component of the **color** (p. 9).*
- double **g**
  *The green component of the **color** (p. 9).*
- double **b**
  *The blue component of the **color** (p. 9).*
- double **a**
  *The transparency of the **color** (p. 9) (1 = opaque).*

5.1.1 Constructor & Destructor Documentation

5.1.1.1 glot::color::color (double *red* = 0, double *green* = 0, double *blue* = 0, double *alpha* = 1) [inline]

Constructor.

Parameters:

red - red component of **color** (p. 9)

green - green component of **color** (p. 9)
blue - blue component of **color** (p. 9)
alpha - the transparency of the **color** (p. 9)

5.1.2 Member Data Documentation

5.1.2.1 double glot::color::a

The transparency of the **color** (p. 9) (1 = opaque).

5.1.2.2 double glot::color::b

The blue component of the **color** (p. 9).

5.1.2.3 double glot::color::g

The green component of the **color** (p. 9).

5.1.2.4 double glot::color::r

The red component of the **color** (p. 9).

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

- **color.h**

5.2 glot::curve Class Reference

`#include <curve.h>`

Public Member Functions

- **curve** (const **function** &func, const **color** &col)

  Constructor.

- **curve** (**function::double_function** func, const **color** &col)

  Constructor.

- double **at** (double x)

  *Evaluate the underlying **function** (p. 13) at x.*

Public Attributes

- **color c**

  Color variable.

5.2.1 Constructor & Destructor Documentation

5.2.1.1 glot::curve::curve (const function & *func*, const color & *col*) [inline]

Constructor.

Parameters:

func - the **function** (p. 13) to render

col - the **color** (p. 9) of the **curve** (p. 11)

5.2.1.2 glot::curve::curve (function::double_function *func*, const color & *col*) [inline]

Constructor.

Parameters:

func - a **function** (p. 13) to plot

col - the **color** (p. 9) of the **curve** (p. 11)

Instead of declaring both a **function** (p. 13) and a **curve** (p. 11), we anticipate it being useful to just declare a **curve** (p. 11) with the **function** (p. 13) you've already defined

5.2.2 Member Function Documentation

5.2.2.1 double glot::curve::at (double *x*)

Evaluate the underlying **function** (p. 13) at x.

Parameters:

x - the x value at which to evaluate

For a given x, it returns the y value of the **function** (p. 13)

5.2.3 Member Data Documentation

5.2.3.1 color glot::curve::c

Color variable.

This is the **color** (p. 9) the **curve** (p. 11) is supposed to take on.

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

- **curve.h**

5.3 glot::function Class Reference

`#include <function.h>`

Public Types

- typedef double(∗ **double_function**)(double x)
  A double-to-double mapping.

Public Member Functions

- **function** (**double_function** f)
  Constructor.
- double **eval** (double x)
  *Evaluate the **function** (p. 13) at a **point** (p. 20).*

5.3.1 Member Typedef Documentation

5.3.1.1 typedef double(∗ glot::function::double_function)(double x)

A double-to-double mapping.

Parameters:

x - for a given x, return another double

Just a mapping of R1 onto R1, using any C++ code meeting that definition

5.3.2 Constructor & Destructor Documentation

5.3.2.1 glot::function::function (double_function *f*)

Constructor.

Parameters:

f - a **function** (p. 13)

5.3.3 Member Function Documentation

5.3.3.1 double glot::function::eval (double *x*)

Evaluate the **function** (p. 13) at a **point** (p. 20).

Parameters:

x - the x value at which to evaluate

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

- **function.h**

5.4 glot::grapher Class Reference

`#include <grapher.h>`

Public Types

- typedef void(∗ **keyboard_function**)(unsigned char key, GLint x, GLint y)
  Keyboard event handler typedef.
- typedef void(∗ **click_function**)(GLint button, GLint x, GLint y)
  Click even handler typedef.

Static Public Member Functions

- static int **initialize** (int argc, char ∗∗argv, short int options=AXES_ON|GRID_ON|X_LIN|Y_LIN, short int k_options=ZOOM_KEYS_ON|AXES_KEYS_ON|GRID_KEYS_ON)
  *Initialize the **grapher** (p. 15).*
- static void **run** ()
  Enter the OpenGL main loop after initialization.
- static void **redraw** ()
  User-requested redraw.
- static void **add_curve** (**curve** &c)
  *Add a **curve** (p. 11) to the plot.*
- static void **delete_curve** (**curve** &c)
  *Delete a **curve** (p. 11) from the plot.*
- static void **add_point** (**point** &p)
  *Add a **point** (p. 20) to the plot.*
- static void **delete_point** (**point** &p)
  *Delete a **point** (p. 20) from the plot.*
- static void **set_keyboard_function** (**keyboard_function** k)
  Set up a keyboard event handler.
- static void **set_click_function** (**click_function** c)
  Set up a click event handler.
- static void **zoom** (double scale)
  Zoom in / out by a scale.
- static double **get_x_coord** (GLint x)
  Transform a screen x coordinate to a world one.

- static double **get_y_coord** (GLint y)

  Transform a screen y coordinate to a world one.

5.4.1 Member Typedef Documentation

5.4.1.1 typedef void(* glot::grapher::click_function)(GLint button, GLint x, GLint y)

Click even handler typedef.

Parameters:

button - GLint button pressed

x - GLint x coordinate

y - GLint y coordinate

See set_click_function(...) for more details

5.4.1.2 typedef void(* glot::grapher::keyboard_function)(unsigned char key, GLint x, GLint y)

Keyboard event handler typedef.

Parameters:

key - unsigned char

x - GLint x coordinate

y - GLint y coordinate

A keyboard event handler accepts a key and x, y coordinates

5.4.2 Member Function Documentation

5.4.2.1 static void glot::grapher::add_curve (curve & *c*) [static]

Add a **curve** (p. 11) to the plot.

Parameters:

c - The **curve** (p. 11) you wish to add to the plot

If you instantiate a **curve** (p. 11), you can add it to the plot with this **function** (p. 13). NOTE: This does not automatically request a redisplay. The idea here is that you may be adding several curves at once, and so after adding all of your curves, etc., you should then call **grapher::redraw()** (p. 18)

5.4.2.2 static void glot::grapher::add_point (point & p) [static]

Add a **point** (p. 20) to the plot.

Parameters:

p - The Point you wish to add to the plot

Instantiate a **point** (p. 20) and then add it to the plot with this **function** (p. 13). NOTE: Does not automatically request a redisplay

5.4.2.3 static void glot::grapher::delete_curve (curve & c) [static]

Delete a **curve** (p. 11) from the plot.

Parameters:

c - The **curve** (p. 11) you wish to delete from the plot

If you've plotted a **curve** (p. 11), but wish to remove it, you can do so with this **function** (p. 13). As with add_curve(...), this does not automatically request a redisplay.

5.4.2.4 static void glot::grapher::delete_point (point & p) [static]

Delete a **point** (p. 20) from the plot.

Parameters:

p - The **point** (p. 20) you wish to delete from the plot

If you've plotted a **point** (p. 20), you can remove it from the plot with this **function** (p. 13). NOTE: Does not automatically request a redisplay.

5.4.2.5 static double glot::grapher::get_x_coord (GLint x) [static]

Transform a screen x coordinate to a world one.

Parameters:

x - the screen x coordinate to transform

It's the grapher's responsibility to know how to transform a screen coordinate into a world coordinate with this **function** (p. 13).

5.4.2.6 static double glot::grapher::get_y_coord (GLint y) [static]

Transform a screen y coordinate to a world one.

Parameters:

y - the screen y coordinate to transform

It's the grapher's repsonsibility to know how to transform a screen coordinate into a world coordinate with this **function** (p. 13).

5.4.2.7 static int glot::grapher::initialize (int *argc*, char ** *argv*, short int *options* = AXES_ON|GRID_ON|X_LIN|Y_LIN, short int *k_ options* = ZOOM_KEYS_ON|AXES_KEYS_ON|GRID_KEYS_ON) [static]

Initialize the **grapher** (p. 15).

Parameters:

argc - same as argc used for OpenGL initialization

argv - same as argv used for OpenGL initialization

options - startup options

k_ options - the default keyboard actions options

Use a bitwise or to select startup options: AXES_ON, AXES_OFF GRID_ON, GRID_OFF X_LIN, X_LOG (linear x scale or logarithmic) Y_LIN, Y_LOG (linear y scale or logarithmic)

5.4.2.8 static void glot::grapher::redraw () [static]

User-requested redraw.

If you create an event handler that will make some changes to the graph, you can make those changes and then request a redraw with this **function** (p. 13).

5.4.2.9 static void glot::grapher::run () [static]

Enter the OpenGL main loop after initialization.

In general, you will set up your event handlers, main code, etc., and then when you've gotten everything in place, you call **grapher::run()** (p. 18) to start the program's OpenGL portion.

5.4.2.10 static void glot::grapher::set_click_function (click_function *c*) [static]

Set up a click event handler.

Parameters:

c - the **function** (p. 13) you'd like to handle click events

If you'd like to set up a **function** (p. 13) to handle click events (which are when a user presses down and then releases a button at the same spot), you can register it with this **function** (p. 13). NOTE: this is different from a motion **function** (p. 13).

5.4.2.11 static void glot::grapher::set_keyboard_function (keyboard_function *k*) [static]

Set up a keyboard event handler.

Parameters:

k - the **function** (p. 13) you'd like to handle key events

If you'd like to set up a **function** (p. 13) to handle key events, you can register them with this **function** (p. 13). NOTE: there are default behaviors for certain keys that are not overridden here. For example, '+' zooms in, but if you use that key as well, both a zoom and your operation will take place.

The idea behind this is that these default behaviors are not the programmer's responsibility to code up as well. They are interface freebies.

Bitwise or options together from the set: ZOOM_KEYS_ON / ZOOM_KEYS_OFF AXES_-KEYS_ON / AXES_KEYS_OFF GRID_KEYS_ON / GRID_KEYS_OFF

5.4.2.12 static void glot::grapher::zoom (double *scale*) [static]

Zoom in / out by a scale.

Parameters:

scale - the scale by which to zoom

This determines the center of the plot as it is now, and scales in / out by a factor of scale outward / inward from that center **point** (p. 20).

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

- **grapher.h**

5.5 glot::point Class Reference

`#include <point.h>`

Public Member Functions

- **point** (double i, double j, double k, const **color** &col)

  Constructor.

Public Attributes

- **color c**

  *The **color** (p. 9) of the the **point** (p. 20).*

- double **x**

  *The x coordinate of the **point** (p. 20).*

- double **y**

  *The y coordinate of the **point** (p. 20).*

- double **z**

  *The z coordinate of the **point** (p. 20).*

5.5.1 Constructor & Destructor Documentation

5.5.1.1 glot::point::point (double *i*, double *j*, double *k*, const color & *col*) [inline]

Constructor.

Parameters:

i - the x coordinate of the **point** (p. 20)

j - the y coordinate of the **point** (p. 20)

k - the z coordinate of the **point** (p. 20)

col - the **color** (p. 9) with which to draw

5.5.2 Member Data Documentation

5.5.2.1 color glot::point::c

The **color** (p. 9) of the the **point** (p. 20).

5.5.2.2 double glot::point::x

The x coordinate of the **point** (p. 20).

5.5.2.3 double glot::point::y

The y coordinate of the **point** (p. 20).

5.5.2.4 double glot::point::z

The z coordinate of the **point** (p. 20).

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

- **point.h**

Chapter 6

File Documentation

6.1 color.h File Reference

Classes

- class **glot::color**

Namespaces

- namespace **glot**

6.2 curve.h File Reference

```
#include "function.h"
#include "color.h"
```

Classes

- class **glot::curve**

Namespaces

- namespace **glot**

6.3 function.h File Reference

Classes

- class **glot::function**

Namespaces

- namespace **glot**

6.4 grapher.h File Reference

```
#include <OpenGL/glu.h>
#include <OpenGL/gl.h>
#include <GLUT/glut.h>
#include <list>
#include <map>
#include "curve.h"
#include "point.h"
```

Classes

- class **glot::grapher**

Namespaces

- namespace **glot**

Enumerations

- enum **glot::display_opt** {
  glot::AXES_OFF = 0, **glot::GRID_OFF** = 0, **glot::X_LIN** = 0, **glot::Y_LIN** = 0,
  glot::AXES_ON = 1, **glot::GRID_ON** = 2, **glot::X_LOG** = 4, **glot::Y_LOG** = 8
  }

  Enumeration for display options.

- enum **glot::keyboard_opt** {
  glot::ZOOM_KEYS_OFF = 0, **glot::AXES_KEYS_OFF** = 0, **glot::GRID_KEYS_OFF** = 0, **glot::ZOOM_KEYS_ON** = 1,
  glot::AXES_KEYS_ON = 2, **glot::GRID_KEYS_ON** = 4 }

  Enumeration for keyboard action options.

6.5 point.h File Reference

```
#include "color.h"
```

Classes

- class **glot::point**

Namespaces

- namespace **glot**

Index