My Project

Generated by Doxygen 1.8.3.1

Sun Jan 25 2015 23:20:17

Contents

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Chapter 2

Hierarchical 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 documented files with brief descriptions:

Chapter 5

Module Documentation

5.1 Constants

Macros

- #define MAX_NUM_BALLS 1000

  Maximum number of balls possible on screen.
- #define UPDATE_TIMER 20

  Minimum time before next update(int) function is called.
- #define DEFAULT_WINDOW_HEIGHT 1056

  Initial height of window.
- #define DEFAULT_WINDOW_WIDTH 1855

  Initial width of window.
- #define DEFAULT_WINDOW_DEPTH 600

  Initial Depth of box (in 3D view)
- #define TIME_LAG 50

  Time after which succesive balls enter the window.
- #define NUM_SEGMENTS 100

  Number of segment along and around z-axis of sphere.
- #define LIMIT_W 500

  Minimum width of window.
- #define LIMIT_H 500

  Minimum height of window.
- #define PI 3.1415926f

  Value of `PI`
- #define CHANGE_FACTOR 1.2f

  Factor by which speed is increased/decreased on a single click on increase/decrease buttons.
- #define MAX_SPLITS 7;

  Max number of smaller balls on splitting a ball.
- #define zDistance 15.0f

  Default view distance from the plane of drawing.
- #define DT 0.5f

  Time for which position of ball is changed in each update call.
- #define NUM_PARTICLES 1000

  Number of snow particles in the background.

5.1.1 Detailed Description

5.2 Typedefs/Enums

Enumerations

- enum **GameState** { **PLAY**, **PAUSE** }
- enum **Select** { **YES**, **NO** }

Variables

- GameState **gameState**
- Select **border**
- Select **showMenu**
- Select **enable3D**

5.2.1 Detailed Description

5.3 GLUI variables

Variables

- GLUI ∗ **glui**
- int **theme_group_item_id**
- int **theme_group_id**
- int **view_group_item_id**
- int **view_group_id**
- int **inc_id**
- int **dec_id**
- int **play_id**
- int **split_id**
- int **delete_id**
- int **add_id**
- int **window_id**
- GLUI_Rotation ∗ **cube_rotate**
- float **cube_rotation** [16]
- std::string **speed_str**
- GLUI_StaticText ∗ **speed_text**

5.3.1 Detailed Description

Chapter 6

Class Documentation

6.1 Ball Class Reference

Simple class for defining ball objects.

```
#include <Ball.h>
```

Public Member Functions

- Ball ()
- int getID ()

  Returns the ID.
- void setID (int)

  Sets the ID.
- ThreeD getCenter ()

  Returns the center.
- void setCenter (float, float, float)

  Sets the center at (x,y,z) co-ordinates.
- ThreeD getVelocity ()

  Returns velocity.
- void setVelocity (float vx, float vy, float vz)

  Sets velocity vector (vx, vy, vz)
- float getRadius ()

  Returns radius.
- void setRadius (float r)

  Sets radius.
- float getMass ()

  Returns Mass.
- void setMass (float)

  Sets Mass.
- Color getColor ()

  Returns Color.
- void setColor (float, float, float)

  Set Color.
- void Draw (int num_segments)

  Draws the Ball.
- void Move (float)

  changes the position of ball according to its velocity in a small interval dt

- bool clickListen (float, float)

  checks if the ball has been clicked
- pthread_mutex_t & getMutex ()

  Returns mutex for synchronization.
- float getSpeed ()

  Returns speed of the ball.

6.1.1 Detailed Description

Simple class for defining ball objects.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 Ball::Ball ( )

Constructor Creates a Ball with default values of parameters

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

- inc/Ball.h

6.2 Image Class Reference

Represents a BMP image.

`#include <imageLoad.h>`

Public Member Functions

- Image (char ∗, int, int)

  Constructor.
- ∼Image ()

  Destructor.

Public Attributes

- char ∗ data

  Pixel data of image.
- int width

  width and height of image
- int **height**

6.2.1 Detailed Description

Represents a BMP image.

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

- inc/imageLoad.h

6.3 MyColor Struct Reference

To store the color attribute in RGB format.

```
#include <global.h>
```

Public Member Functions

- MyColor (float x=1.0f, float y=1.0f, float z=1.0f)

  Default color : White (1,1,1)

Public Attributes

- float **r**
- float **g**
- float **b**

6.3.1 Detailed Description

To store the color attribute in RGB format.

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

- inc/global.h

6.4 MyStruct Struct Reference

To store three dimensional variables like position, velocity, etc.

```
#include <global.h>
```

Public Attributes

- float **x**
- float **y**
- float **z**

6.4.1 Detailed Description

To store three dimensional variables like position, velocity, etc.

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

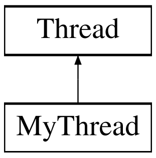
- inc/global.h

6.5 MyThread Class Reference

Extends the functionality of Class Thread.

```
#include <MyThread.h>
```

Inheritance diagram for MyThread:



Public Member Functions

- MyThread (queue< Ball ∗ > ∗)

  Constructor.
- void setQueue (queue< Ball ∗ > ∗)

  sets queue
- queue< Ball ∗ > ∗ getQueue ()

  return queue
- void setBall (Ball ∗)

  sets the ball
- Ball ∗ getBall ()

  return the ball
- void ∗ run ()

  function which runs when start() method of parent thread is called

6.5.1 Detailed Description

Extends the functionality of Class Thread.

Extends Thread Class publicly

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

- inc/MyThread.h

6.6 Particle Class Reference

snow particles in background.

`#include <Particle.h>`

Public Member Functions

- Particle ()

  Constructor.
- float getRadius ()

  returns radius
- void setRadius (float r)

  sets the radius
- ThreeD getCenter ()

  returns center
- void setCenter (float, float, float)

  sets center
- ThreeD getVelocity ()

  returns velocity

- void setVelocity (float, float, float)

    sets velocity
- void drawP ()

    drawing utility function
- void moveP ()

    changes the position of snow particle
- void reset ()

    resets the position of snow particle

6.6.1 Detailed Description

snow particles in background.

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

- inc/Particle.h

6.7 Theme Class Reference

To store features of theme.

```
#include <Theme.h>
```

Public Member Functions

- Theme ()

    Theme with default settings.

Public Attributes

- Color background

    Background color.
- Color clr [3]

    Lighting color.
- ThreeD pos [3]

    Lights position.
- bool isLight [4]

    Enable/Disable Light.
- string image

    Background Image.

6.7.1 Detailed Description

To store features of theme.

Changes in this and should be in sync

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

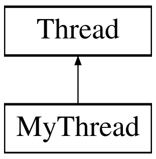
- inc/Theme.h

6.8 Thread Class Reference

Class to be inherited by MyThread.

`#include <thr.h>`

Inheritance diagram for Thread:



Public Member Functions

- Thread ()

    Constructor.
- ∼Thread ()

    Destructor.
- int start ()

    creates a thread and calls run() method
- int join ()

    joins the thread
- int detach ()

    detaches the thread
- pthread_t self ()

    return thread Id
- virtual void ∗ **run** ()=0

6.8.1 Detailed Description

Class to be inherited by MyThread.

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

- inc/thr.h

6.9 Wall Class Reference

Wall object to define boundaries of window/box.

`#include <Wall.h>`

Public Member Functions

- Wall ()

    Constructor.
- Wall (PlaneType, float)

    Creates wall with at given positon of given type.
- PlaneType getPlane () const

    Returns type of the plane.

- float getPosition () const

    Returns the position of the plane.
- void setPosition (float)

    Sets the position of the plane.

6.9.1 Detailed Description

Wall object to define boundaries of window/box.

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

- inc/Wall.h

Chapter 7

File Documentation

7.1 inc/Collision.h File Reference

```
#include "Wall.h"
#include "Ball.h"
```

Functions

- void check_Collision_With_Wall (Ball &, Wall &)

  Checks for a possible collision of a ball with a wall and accordingly update the velocity.
- void check_Collision_With_Ball (Ball &, Ball &)

  Checks for a possible collision of two balls with each other and accordingly update the velocities of both the balls using their respective mutexes.

7.1.1 Detailed Description

Functions for maintaining the physics of the system.

All the collisions are elastic.

7.2 inc/global.h File Reference

Classes

- struct MyStruct

  To store three dimensional variables like position, velocity, etc.
- struct MyColor

  To store the color attribute in RGB format.

Typedefs

- typedef struct MyStruct ThreeD

  To store three dimensional variables like position, velocity, etc.
- typedef struct MyColor Color

  To store the color attribute in RGB format.

7.2.1 Detailed Description

Useful Structures are defined here.

7.3 inc/GUI.h File Reference

```
#include <GL/glut.h>
#include <GL/glui.h>
#include "imageLoad.h"
```

Functions

- void handleMouse (int, int, int, int)

  glutMouseFunc
- void handleResize (int, int)

  glutReshapeFunc
- void handleKeypress (unsigned char, int, int)

  glutKeyboardFunc
- GLuint loadTexture (Image ∗)

  loads texture from an Image
- void drawBox ()

  draws a 3D box at boundaries of balls i.e. walls
- void drawScene ()

  glutDrawFunc
- void initRendering ()

  initialize the graphics rendering
- void update (int)

  glutTimerFunc

7.3.1 Detailed Description

openGL functions are maintained in this seperate file.

Any changes regarding graphics rendering should be made here.

7.4 inc/mainw.h File Reference

```
#include <GL/glut.h>
#include <GL/glui.h>
#include <queue>
#include <vector>
#include <utility>
#include "Ball.h"
#include "MyThread.h"
#include "Wall.h"
#include "Theme.h"
#include "imageLoad.h"
#include "Particle.h"
```

Variables

- float **ALPHA**
- float **rotate**
- int **number_of_balls**
- vector< MyThread * > **threads**
- float **wideAngle**
- float **ratio**
- int **window_height**
- int **window_width**
- float **angleX**
- float **angleY**
- float **angleZ**
- int **borderNumber**
- float **viewDistance**
- GLuint **t1**
- GLuint **tex3d**
- GLuint **wall_tex**
- GLUquadric * **quad**
- GLuint **_textureId**
- vector< Theme * > **themes**
- Theme * **curTheme**
- Wall **wall_x**
- Wall **wall_y**
- Wall **wall_z**
- Particle * **particles**

7.4.1 Detailed Description

File containing global variables to be used in other files/classes

7.5 inc/MyDefines.h File Reference

Macros

- #define MAX_NUM_BALLS 1000

  Maximum number of balls possible on screen.
- #define UPDATE_TIMER 20

  Minimum time before next update(int) function is called.
- #define DEFAULT_WINDOW_HEIGHT 1056

  Initial height of window.
- #define DEFAULT_WINDOW_WIDTH 1855

  Initial width of window.
- #define DEFAULT_WINDOW_DEPTH 600

  Initial Depth of box (in 3D view)
- #define TIME_LAG 50

  Time after which succesive balls enter the window.
- #define NUM_SEGMENTS 100

  Number of segment along and around z-axis of sphere.
- #define LIMIT_W 500

  Minimum width of window.
- #define LIMIT_H 500

Minimum height of window.
- #define PI 3.1415926f

  Value of `PI`
- #define CHANGE_FACTOR 1.2f

  Factor by which speed is increased/decreased on a single click on increase/decrease buttons.
- #define MAX_SPLITS 7;

  Max number of smaller balls on splitting a ball.
- #define zDistance 15.0f

  Default view distance from the plane of drawing.
- #define DT 0.5f

  Time for which position of ball is changed in each update call.
- #define NUM_PARTICLES 1000

  Number of snow particles in the background.

7.5.1 Detailed Description

All the constants are defined here in seperate file.

7.6 inc/MyEnums.h File Reference

Enumerations

- enum **GameState** { **PLAY**, **PAUSE** }
- enum **Select** { **YES**, **NO** }

Variables

- GameState **gameState**
- Select **border**
- Select **showMenu**
- Select **enable3D**

7.6.1 Detailed Description

Enums are defined in this seperate file.

7.7 inc/subMenu.h File Reference

```
#include <GL/glut.h>
#include <GL/glui.h>
#include <string.h>
```

Functions

- void myGlutIdle (void)

  glutIdleFunc
- void glui_callback (int)

  Common call back function for all the menu buttons.

- void initMenu ()

  Initializes the menu. Adds panels, buttons and other items.

  .

Variables

- GLUI ∗ **glui**
- int **theme_group_item_id**
- int **theme_group_id**
- int **view_group_item_id**
- int **view_group_id**
- int **inc_id**
- int **dec_id**
- int **play_id**
- int **split_id**
- int **delete_id**
- int **add_id**
- int **window_id**
- GLUI_Rotation ∗ **cube_rotate**
- float **cube_rotation** [16]
- std::string **speed_str**
- GLUI_StaticText ∗ **speed_text**

7.7.1 Detailed Description

Interface using GL user interface library, glui.h

All the changes to the menu should be added in this file only.

7.8 inc/themeReader.h File Reference

```
#include <vector>
#include <fstream>
#include <iostream>
#include "Theme.h"
```

Functions

- vector< Theme ∗ > readThemes (const char ∗filename)

  Loads different theme from themeFile.txt into a vector .
- bool readBool (ifstream &fin)

  utility to read a bool from file
- float readFloat (ifstream &fin)

  utility to read a float from file

7.8.1 Detailed Description

Changes in this and class should be in sync

7.9 inc/UtilityFunctions.h File Reference

```
#include "Ball.h"
#include "Theme.h"
```

Functions

- void **preProcessTheme** ()
- void **setTheme** (Theme &)
- pair$<$ float, float $>$ **convPixel** (int, int)
- void **changeVelocity** (Ball &, float)
- int **findClickedBall** (float, float)
- Ball $*$ **createNewBall** ()
- void **splitBall** (Ball &, int)
- void **addBall** ()
- void **deleteBall** (int)
- void **setSpeedText** (Ball &)
- void **resetSpeedText** ()
- void **addNewThread** ()
- void **addWorkItems** ()
- float **getRandomFloat** ()

7.9.1 Detailed Description

Functions required from main function and other classes. To add extra features/functionality define in this file only.

Index