robbit

4.0

Generated by Doxygen 1.5.6

Fri Sep 5 17:46:45 2008

Contents

Chapter 1

Robbit

1.1 What is Robbit ?

Robbit is a open-source software which provide 3D simulation environment for multiple robot system. Current version of Robbit is 4.0.0 (for both windows and linux). At the moment, Robbit supports modules for mobile robots only. However, in the future the scope should expand and, of course, contributions are most welcome.

Robbit has been developed thinking in researchers, students, roboticists and hobbyists who want to design, test and simulate mobile robots and research topics like autonomous navigation techniques, obstacle avoidance, artificial intelligence etc.

Robbit is open source and is distributed under the GNU General Public License, published by the Free Software Foundation (version 3 of the License, or any later version).

1.2 Contributors

(alphabetical)

Antonelli Gianluca, Dipartimento di Automazione, Elettromagnetismo, Ingegneria dell'Informazione e Matematica Industriale, Universita degli Studi di Cassino, Italy

Arrichiello Filippo, Laboratory of Industrial Automation-Robotics Research group, Dipartimento di Automazione, Elettromagnetismo, Universita degli Studi di Cassino, Italy

Bhushan Chitresh, Department of Electronics & Electrical Communication Engineering Indian Institute of Technology Kharagpur, India

Prakash Chander, Department of Computer Science Engineering Indian Institute of Technology Guwahati, India

Purkayasth Sayandeep, Department of Electronics & Electrical Communication Engineering Indian Institute of Technology Kharagpur, India

Chapter 2

Installation

2.1 Installation

All the dependencies must be fulfilled prior to installation. see Dependencies in Related Pages.

2.1.1 Linux

All the structures and classes are available in .h files, and the methods/functions are available in .c files. To complie Robbit go to source directory and run:

```
 $> cmake .
```

then

```
 $> make
```

To execute run:

```
 $> ./Robbit
```

Note: Configuration settings can be modified in the file "CMakeLists.txt" before starting the installation.

2.1.2 Windows 32-/64-Bit

Open Microsoft Visual Studio 2005 Solution robbit.sln, which can be found in vc2005 directory. Build the solution. **robbit.exe** will be created in the same directory.

The dependencies are included in the include/ folder which contain FLTK and PNG includes and libraries. These will be used directly by the build process. Make sure OpenGL libraries (static and dynamic) and include files are present your system.

The following libraries are to be used for linking.

```
opengl32.lib wsock32.lib comctl32.lib glaux.lib glu32.lib fltk.lib fltkgl.lib libpng.lib
```

While distributing, note that the project is built with FLTK and STL headers and libs, so only glut32.dll will be required apart from the executable, and optional log-file and/or Obstacle files.

Chapter 3

Dependencies

3.1 Dependencies

- C/C++ compiler: (`Windows`), (`Linux`)
- openGL v 3.7.6: (`Windows`), freeglut v 2.4.0 (`Linux`)
- FLTK with FLUID v 1.1.9: (`Windows & Linux`)
- libpng v 1.2.31: (`Windows & Linux`)
- make v 3.81: (`Linux`)
- cmake v 2.6.1: (`Linux`)
- Standard Template Library (`Windows & Linux`)
- openCV Library: (`Windows & Linux`) [NOT required for Robbit 4.0.0]
- Doxygen: (`Windows & Linux`) [For Documentation]

Chapter 4

Usage

4.1 Keyboard mappings

t/T	Toggle Top View
l/L	Toggle Trails
a/A	About Window
d/D	Advanced Settings
s/S	Take Sreenshot
Escape	Quit simulator
n/N	Toggle Bot Numbering
o/O	Toggle Obstacles
u/U	Toggle Auto-rotating view
1,2,... 6	Onboard view of respective bot
+/-	Zoom in/out

4.2 Mouse mappings

Pan, zoom and rotate features have been mapped to mouse keys followed by drag, as follows.

Left key press and drag	Panaromic rotate
Right key press and drag	Pan
Mouse scroll	Zoom in/out

4.3 Advanced settings

Options have been provided to modify the arena's dimension, number of trail points, radius of ball, number of frames till which a collision is marked colored after the collision has actually ended, and render quality.

4.3.1 Obstacles

The position of obstacles, size and dimensions are to be placed in the Obstacles.txt. The format to be followed for the different types of simplistic obstacles are given within the same. As a sample an example of each is also provided. Another file following the same format may be used instead of this file (at runtime, an option for file selection is provided).

4.4 Custom algorithms

One may write his own algorithm to generate coordinates of the objects, and other details. This must be written within the user function GetNextFrame.h.

Chapter 5

Features

Comprehensive feature list:

- 3D panaromic view through mouse control
- Lighting control - upto four lights
- Chessboard floor
- Numbering the robots
- Onboard View of robots
- Saving snapshots
- Animation speed control
- Centroid of the robots
- Motion trails
- Show/Hide trails
- Placing obstacle (by reading from ASCII file)
- Show/Hide obstacle
- Collision detection by changing colour
- Media Player style play/pause/stop/seek of animation
- 3D view of robots of different types including two built-in KheperaII and KheperaIII, and support for custom robot (limited).
- Customizable Robots
- Customizable Arena
- Pan view
- Auto-rotating view
- Simulating from the given motion algorithm (in form of code)

External Libraries used:

- GUI using FLTK
- libPNG for screenshots (earlier use of OpenCV has been discontinued)
- Standard Template Library for dynamic implementation

Chapter 6

Class Index

6.1 Class List

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

Chapter 7

File Index

7.1 File List

Here is a list of all files with brief descriptions:

Chapter 8

Class Documentation

8.1 File_Data Class Reference

Class to store file data.

```
#include <File_Data.h>
```

Public Member Functions

- int BuildFileIndex ()

  Builds index for quick reference within log file.

- int SetFileInput (const char ∗fname)

  Opens the log file and calls BuildFileIndex().

- int IndexSize () const
- int GetData (int i, Frame_Data &data)

  Get data from file index.

Protected Types

- typedef std::size_t pos_type

Protected Attributes

- std::vector< pos_type > file_index
- std::ifstream input

8.1.1 Detailed Description

Class to store file data.

This record may be one from a log-file. It includes methods to build an index, to get based on the index, etc.

Definition at line 28 of file File_Data.h.

8.1.2 Member Typedef Documentation

8.1.2.1 typedef std::size_t File_Data::pos_type `[protected]`

Definition at line 31 of file File_Data.h.

8.1.3 Member Function Documentation

8.1.3.1 int File_Data::BuildFileIndex ()

Builds index for quick reference within log file.

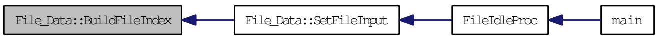
Stores the positions of get pointers to valid records in the log file, in a vector. This way, the file does not need to be traversed in serial fashion, and allows for quicker and random access to file data.

Definition at line 47 of file File_Data.c.

References file_index, input, max_x, max_y, min_x, min_y, and no_of_bots.

Referenced by SetFileInput().

Here is the caller graph for this function:



8.1.3.2 int File_Data::SetFileInput (const char $*$ *fname*)

Opens the log file and calls BuildFileIndex().

Parameters:

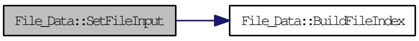
fname string : name of the log file to be opened

Definition at line 29 of file File_Data.c.

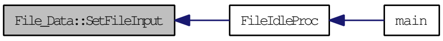
References BuildFileIndex(), and input.

Referenced by FileIdleProc().

Here is the call graph for this function:



Here is the caller graph for this function:



8.1.3.3 int File_Data::IndexSize () const `[inline]`

Definition at line 38 of file File_Data.h.

References file_index.

8.1.3.4 int File_Data::GetData (int *ind*, Frame_Data & *data*)

Get data from file index.

Reads each line pointed to by the get pointer locations stored in the index. Breaks each line into the required fields, and stores the fields into the object 'data' of class 'Frame_Data'.

Parameters:

ind current record number / index

data object 'data' of class 'Frame_Data'

Returns:

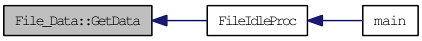
0 always

Definition at line 125 of file File_Data.c.

References Frame_Data::ball_x, Frame_Data::ball_y, Frame_Data::bot_hit, Frame_Data::bot_orient, Frame_Data::bot_vorient, Frame_Data::bot_vx, Frame_Data::bot_vy, Frame_Data::bot_x, Frame_Data::bot_y, file_index, input, no_of_bots, Frame_Data::time, and Frame_Data::time_step.

Referenced by FileIdleProc().

Here is the caller graph for this function:



8.1.4 Member Data Documentation

8.1.4.1 std::vector<pos_type> File_Data::file_index [protected]

Definition at line 32 of file File_Data.h.

Referenced by BuildFileIndex(), GetData(), and IndexSize().

8.1.4.2 std::ifstream File_Data::input [protected]

Definition at line 33 of file File_Data.h.

Referenced by BuildFileIndex(), GetData(), and SetFileInput().

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

- File_Data.h
- File_Data.c

8.2 Frame_Data Class Reference

Class to store frame data.

```
#include <Frame_Data.h>
```

Public Attributes

- FloatVec bot_x
- FloatVec bot_y
- FloatVec bot_vx
- FloatVec bot_vy
- FloatVec bot_orient
- FloatVec bot_vorient
- IntVec bot_hit [2]
- int bot_design
- float time
- float ball_x
- float ball_y
- float ball_vx
- float ball_vy
- float bot_center_x
- float bot_center_y
- float time_step

8.2.1 Detailed Description

Class to store frame data.

Stores data for a frame that is to be rendered. On extraction of a record from log-file, or when the custom algorithm updates the coordinate data, etc. the data here must be stored here.

Definition at line 30 of file Frame_Data.h.

8.2.2 Member Data Documentation

8.2.2.1 FloatVec Frame_Data::bot_x

Definition at line 33 of file Frame_Data.h.

Referenced by Frame_Display::BotHit(), Frame_Display::DetectObstacleCollision(), Frame_Display::draw(), Frame_Display::DrawTrails(), File_Data::GetData(), GetNextFrame(), Frame_Display::reshape(), and Frame_Display::setDefaults().

8.2.2.2 FloatVec Frame_Data::bot_y

Definition at line 34 of file Frame_Data.h.

Referenced by Frame_Display::BotHit(), Frame_Display::DetectObstacleCollision(), Frame_Display::draw(), Frame_Display::DrawTrails(), File_Data::GetData(), GetNextFrame(), Frame_Display::reshape(), and Frame_Display::setDefaults().

8.2.2.3 FloatVec Frame_Data::bot_vx

Definition at line 35 of file Frame_Data.h.

Referenced by File_Data::GetData(), GetNextFrame(), and Frame_Display::setDefaults().

8.2.2.4 FloatVec Frame_Data::bot_vy

Definition at line 36 of file Frame_Data.h.

Referenced by File_Data::GetData(), GetNextFrame(), and Frame_Display::setDefaults().

8.2.2.5 FloatVec Frame_Data::bot_orient

Definition at line 37 of file Frame_Data.h.

Referenced by Frame_Display::draw(), File_Data::GetData(), GetNextFrame(), Frame_Display::reshape(), and Frame_Display::setDefaults().

8.2.2.6 FloatVec Frame_Data::bot_vorient

Definition at line 38 of file Frame_Data.h.

Referenced by File_Data::GetData(), GetNextFrame(), and Frame_Display::setDefaults().

8.2.2.7 IntVec Frame_Data::bot_hit[2]

Definition at line 39 of file Frame_Data.h.

Referenced by Frame_Display::BotHit(), Frame_Display::DetectObstacleCollision(), Frame_Display::draw(), File_Data::GetData(), GetNextFrame(), and Frame_Display::setDefaults().

8.2.2.8 int Frame_Data::bot_design

Definition at line 41 of file Frame_Data.h.

8.2.2.9 float Frame_Data::time

Definition at line 42 of file Frame_Data.h.

Referenced by Frame_Display::CaptureScreenshot(), File_Data::GetData(), and GetNextFrame().

8.2.2.10 float Frame_Data::ball_x

Definition at line 44 of file Frame_Data.h.

Referenced by Frame_Display::BotHit(), Frame_Display::draw(), Frame_-Display::DrawTrails(), File_Data::GetData(), and GetNextFrame().

8.2.2.11 float Frame_Data::ball_y

Definition at line 45 of file Frame_Data.h.

Referenced by Frame_Display::BotHit(), Frame_Display::draw(), Frame_-Display::DrawTrails(), File_Data::GetData(), and GetNextFrame().

8.2.2.12 float Frame_Data::ball_vx

Definition at line 46 of file Frame_Data.h.

8.2.2.13 float Frame_Data::ball_vy

Definition at line 47 of file Frame_Data.h.

8.2.2.14 float Frame_Data::bot_center_x

Definition at line 48 of file Frame_Data.h.

Referenced by Frame_Display::draw().

8.2.2.15 float Frame_Data::bot_center_y

Definition at line 49 of file Frame_Data.h.

Referenced by Frame_Display::draw().

8.2.2.16 float Frame_Data::time_step

Definition at line 50 of file Frame_Data.h.

Referenced by FileIdleProc(), File_Data::GetData(), and GetNextFrame().

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

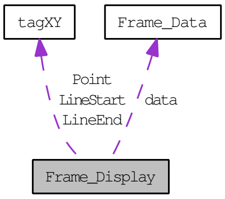
- Frame_Data.h

8.3 Frame_Display Class Reference

Class to store data required for running of simulation and GUI.

```
#include <Frame_Display.h>
```

Collaboration diagram for Frame_Display:



Public Member Functions

- Frame_Display (int x, int y, int w, int h)

    Creates GL rendering widget in main window.

- int handle (int event)

    Event handler.

- void UpdateFrame (Frame_Data _data)

    Updates the current frame data.

- void draw ()

    Renders the current frame.

- void DrawFloor ()

    Renders the floor.

- void ReadObstacle ()

    Reads obstacle data file.

- void DrawObstacle ()

    Renders obstacle.

- void DrawTrails ()

    Renders the trails.

- void BotHit ()

    Detects collisions between any of the bots and/or the ball and/or obstacles.

- void CaptureScreenshot ()

    Captures a screenshot as a PNG.

- void DetectObstacleCollision (int bot_no)

  Detects collision between an object and any obstacle.

- void position_khepera2 (float pos_x, float pos_y, double orient, int colour)

  Renders KheperaII bots.

- void position_khepera3 (float pos_x, float pos_y, double orient, int colour)

  Renders KheperaIII bots.

- void PositionCustomRobot (float pos_x, float pos_y, double orient, int colour)

  Renders custom bots.

- void AutoUpdateEye ()

  Changes view angle during autoview.

- void init (void)

  Initializes OpenGL simulation startup settings.

- void reshape (int w, int h)

  Resizes the simulation.

- void setIndex (int val)
- void outputCharacter (float x, float y, float z, char ∗string)

  Renders a character at a specific location.

- void setPlayPause ()

  Swaps Play and Pause symbols in GUI Play/Pause button, on click.

- void setExit ()

  Shows confirmation box before exit.

- void setStop ()

  Stops simulation.

- void setReset ()

  Resets all the settings to default.

- void setBotNumbering (bool val)

  Toggles the numbering of bot.

- void setTrails (bool val)

  Clears trail data.

- void setObstacles (bool val)

  Toggles the displaying of the Obstacles.

- void setTopView (bool val)

  Toggles the top view mode.

- void setAutoView (bool val)

  Toggles the Auto-rotate mode.

- void selectBot (int val)

  Sets bot structure variables.

- void setLights (int val)

  Set no of lights to be used.

- void setRenderSpeed (float val)

  Sets the rendering speed.

- void setDefaults ()

  Sets default values of variables.

- void setCustomRobot (float height, float radius)

  Sets dimension of Custom Robot (Cylindrical).

- void setArena (float maxx, float minx, float maxy, float miny)

  Sets Arena coordinates.

- void setBallRadius (float val)

  Sets radius of ball.

- void setTrailPoints (float val)

  Sets no of trail data to keep.

- void setColouredSteps (float val)

  Sets no of coloured steps (in case of any collision).

- void initAdvWindow ()

  Sets configuration of Advanced Settings menu elements.

- void setGraphicsQuality (int slices, int stacks)

  Sets graphic quality of objects rendered.

Public Attributes

- bool is_paused
- float render_speed
- int current_index

- float view_centerX
- float view_centerY
- float view_centerZ

Private Attributes

- int bot_selected
- bool number_bots
- bool show_obstacle
- bool show_trails
- bool topview
- bool autoview
- float obs_2D [20][2]
- float obs_WL [20][5]
- float obs_CB [20][3]
- float obs_SP [20][3]
- int obs_2D_counter
- int WL_counter
- int CB_counter
- int SP_counter
- float radius_ball
- int no_of_trail_data
- int coloured_steps
- std::deque< float > bot_trail [no_of_bots][2]
- std::deque< float > ball_trail [2]
- float radius_camera_movement
- float theta
- float phi
- float eyeX
- float eyeY
- float eyeZ
- float upX
- float upY
- float upZ
- float aspect
- float Near
- float Far
- int camera_on_bot
- int delta_zoom
- int no_of_lights
- float mouseX0
- float mouseY0
- bool mousePushValid
- float radius_of_robot
- float height_robot
- int bot_slices

- int bot_stacks
- int small_disk_slices
- int small_disk_stacks
- Frame_Data data
- XY LineStart
- XY LineEnd
- XY Point

8.3.1 Detailed Description

Class to store data required for running of simulation and GUI.

Stores data required to render a frame. It also has all the data for running and handling the events from GUI. On update of values of Frame_Data object, the data here must be updated.

Definition at line 27 of file Frame_Display.h.

8.3.2 Constructor & Destructor Documentation

8.3.2.1 Frame_Display::Frame_Display (int *x*, int *y*, int *w*, int *h*)

Creates GL rendering widget in main window.

Parameters:

x x coordinate of top-left pixel of widget

y y coordinate of top-left pixel of widget

w width of widget

h height of widget

Definition at line 91 of file Frame_Display.c.

8.3.3 Member Function Documentation

8.3.3.1 int Frame_Display::handle (int *event*)

Event handler.

Any event in the FLTK window called this event handler with a event handle. Handles both keystrokes (shortcuts), mouse gestures.

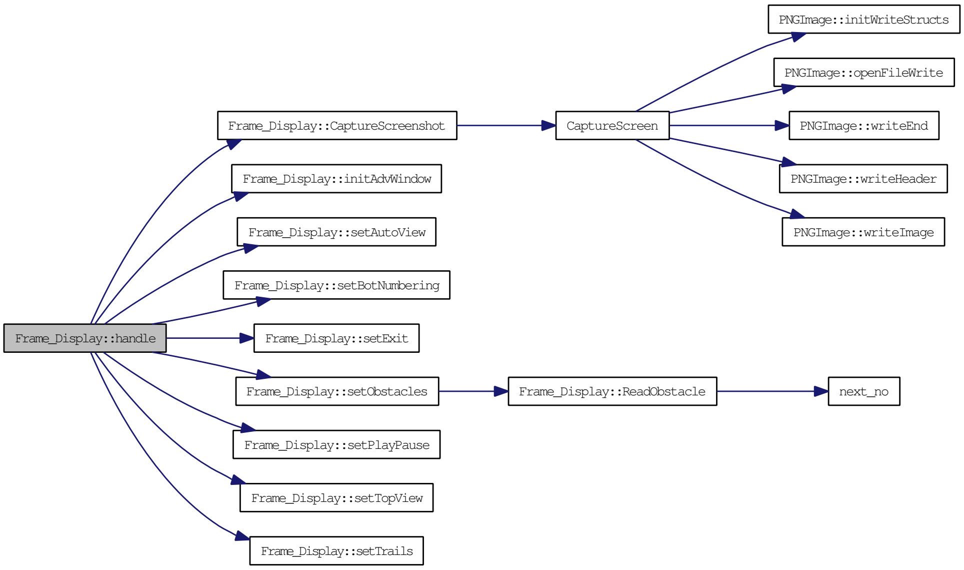
Parameters:

event event handle number

Definition at line 908 of file Frame_Display.c.

References RobbitUI::AboutWindow, RobbitUI::AdvSettingsWindow, autoview, camera_on_bot, CaptureScreenshot(), delta_zoom, initAdvWindow(), max_x, max_y, min_x, min_y, mousePushValid, mouseX0, mouseY0, no_of_bots, number_bots, phi, radianFactor, radius_camera_movement, setAutoView(), setBotNumbering(), setExit(), setObstacles(), setPlayPause(), setTopView(), setTrails(), show_obstacle, show_trails, RobbitUI::textrobbit, theta, RobbitUI::ToggleAutoView, RobbitUI::ToggleBotNumbering, RobbitUI::ToggleObstacles, RobbitUI::ToggleTopView, RobbitUI::ToggleTrails, topview, view_centerX, view_centerY, and view_centerZ.

Here is the call graph for this function:


PNGImage::initWriteStructs
PNGImage::openFileWrite
Frame_Display::CaptureScreenshot
CaptureScreen
PNGImage::writeEnd
Frame_Display::initAdvWindow
PNGImage::writeHeader
Frame_Display::setAutoView
PNGImage::writeImage
Frame_Display::setBotNumbering
Frame_Display::handle
Frame_Display::setExit
Frame_Display::setObstacles
Frame_Display::ReadObstacle
next_no
Frame_Display::setPlayPause
Frame_Display::setTopView
Frame_Display::setTrails

8.3.3.2 void Frame_Display::UpdateFrame (Frame_Data *_data*)

Updates the current frame data.

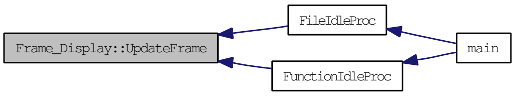
Updates the object data to be rendered to mirror those most recently read from the file/custom algorithm.

Definition at line 286 of file Frame_Display.c.

References current_index, and data.

Referenced by FileIdleProc(), and FunctionIdleProc().

Here is the caller graph for this function:



8.3.3.3 void Frame_Display::draw ()

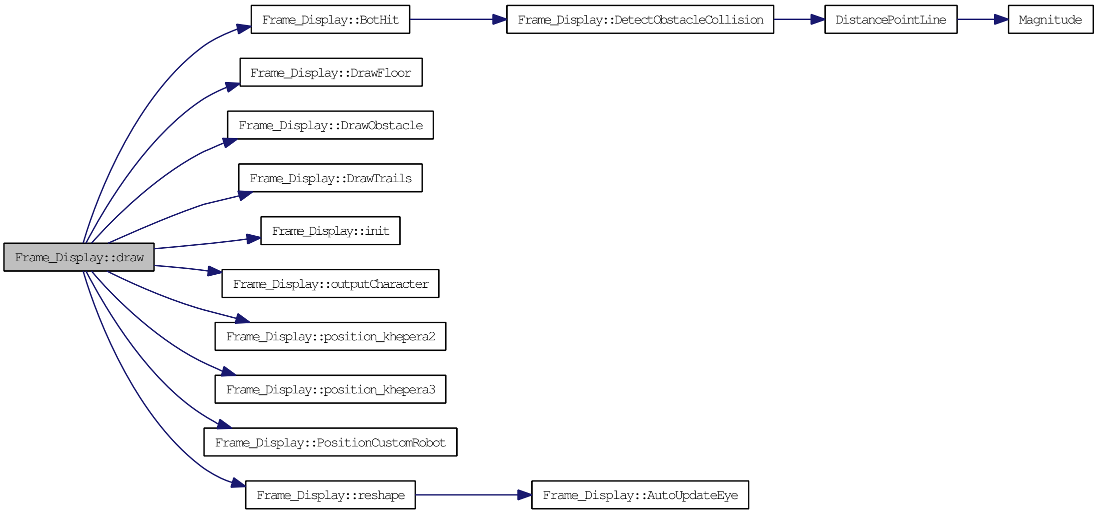
Renders the current frame.

Draws the centroid of the robots, the robots, bot numbering, trails, and the ball. Also, if the robots and/or the ball or obstacles collide, then it marks them and sets highlight colors as required. Calls functions to render the obstacles, and the floor.

Definition at line 212 of file Frame_Display.c.

References ball_shininess, ball_specular, Frame_Data::ball_x, Frame_Data::ball_y, Frame_Data::bot_center_x, Frame_Data::bot_center_y, Frame_Data::bot_hit, Frame_Data::bot_orient, bot_selected, bot_slices, bot_stacks, Frame_Data::bot_x, Frame_Data::bot_y, BotHit(), data, disk_center_specular, DrawFloor(), DrawObstacle(), DrawTrails(), height_robot, init(), no_of_bots, number_bots, outputCharacter(), position_khepera2(), position_khepera3(), PositionCustomRobot(), quad, radius_ball, reshape(), show_obstacle, show_trails, and text_specular.

Here is the call graph for this function:


Frame_Display::draw
Frame_Display::BotHit
Frame_Display::DetectObstacleCollision
DistancePointLine
Magnitude
Frame_Display::DrawFloor
Frame_Display::DrawObstacle
Frame_Display::DrawTrails
Frame_Display::init
Frame_Display::outputCharacter
Frame_Display::position_khepera2
Frame_Display::position_khepera3
Frame_Display::PositionCustomRobot
Frame_Display::reshape
Frame_Display::AutoUpdateEye

8.3.3.4 void Frame_Display::DrawFloor ()

Renders the floor.

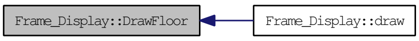
The floor is presently a checkerboard of blue and white.

Definition at line 510 of file Frame_Display.c.

References floor_specular, max_x, max_y, min_x, min_y, no_of_lights, and plane_specular.

Referenced by draw().

Here is the caller graph for this function:



8.3.3.5 void Frame_Display::ReadObstacle ()

Reads obstacle data file.

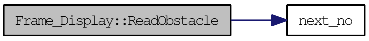
Reads and parses the obstacle data file and stores the obstacle details, provided they are given in a particular format, into an array. Presently, supported types of obstacles are Cube, Sphere, Wall and a Pole.

Definition at line 360 of file Frame_Display.c.

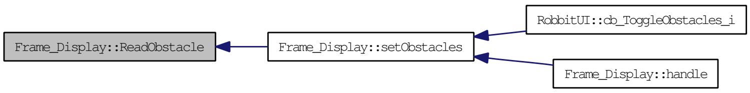
References CB_counter, next_no(), obs_2D, obs_2D_counter, obs_CB, obs_SP, obs_WL, show_obstacle, SP_counter, and WL_counter.

Referenced by setObstacles().

Here is the call graph for this function:



Here is the caller graph for this function:


Frame_Display::ReadObstacle
Frame_Display::setObstacles
RobbitUI::cb_ToggleObstacles_i
Frame_Display::handle

8.3.3.6 void Frame_Display::DrawObstacle ()

Renders obstacle.

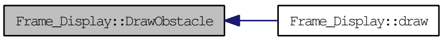
Takes details stored by ReadObstacle and renders some supported types of obstacles. Note that the pole is represented purely as a point on the floor. It has no rendering of its exected structure to prevent obstruction in viewing.

Definition at line 452 of file Frame_Display.c.

References bot_slices, CB_counter, disk_2D_obstacle_shininess, disk_2D_obstacle_specular, obs_2D, obs_2D_counter, obs_CB, obs_SP, obs_WL, quad, small_disk_slices, small_disk_stacks, SP_counter, and WL_counter.

Referenced by draw().

Here is the caller graph for this function:



8.3.3.7 void Frame_Display::DrawTrails ()

Renders the trails.

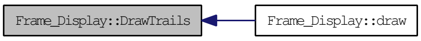
The trails are stored as a finite length vector. On change of frame data, the oldest points belonging to it are popped from the front and new points are pushed at the back.

Definition at line 545 of file Frame_Display.c.

References ball_trail, Frame_Data::ball_x, Frame_Data::ball_y, bot_trail, Frame_Data::bot_x, Frame_Data::bot_y, data, disk_center_specular, no_of_bots, and no_of_trail_data.

Referenced by draw().

Here is the caller graph for this function:



8.3.3.8 void Frame_Display::BotHit ()

Detects collisions between any of the bots and/or the ball and/or obstacles.

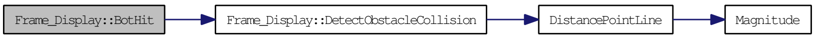
The collision of any bot with a obstacle is handled through DetectObstacleCollision(<bot number>).

Definition at line 867 of file Frame_Display.c.

References Frame_Data::ball_x, Frame_Data::ball_y, Frame_Data::bot_hit, Frame_Data::bot_x, Frame_Data::bot_y, coloured_steps, data, DetectObstacleCollision(), info_refresh_count, no_of_bots, radius_ball, radius_of_robot, and show_obstacle.

Referenced by draw().

Here is the call graph for this function:



Here is the caller graph for this function:

8.3.3.9 void Frame_Display::CaptureScreenshot ()

Captures a screenshot as a PNG.

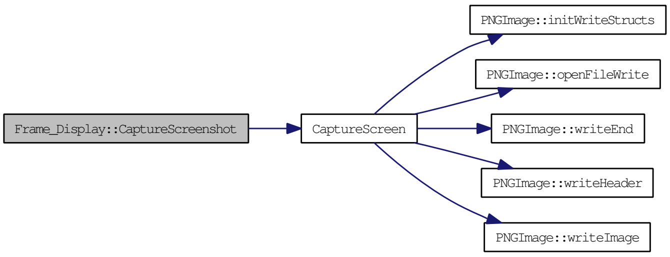
Saves a screenshot with a generated filename in Portable Network Graphics format. The filename contains details like the time, and view angles, θ and ϕ.

Definition at line 1097 of file Frame_Display.c.

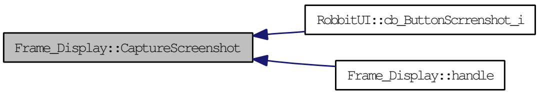
References CaptureScreen(), data, phi, theta, and Frame_Data::time.

Referenced by RobbitUI::cb_ButtonScrrenshot_i(), and handle().

Here is the call graph for this function:



Here is the caller graph for this function:


Frame_Display::CaptureScreenshot
RobbitUI::cb_ButtonScrrenshot_i
Frame_Display::handle

8.3.3.10 void Frame_Display::DetectObstacleCollision (int *bot_no*)

Detects collision between an object and any obstacle.

Parameters:

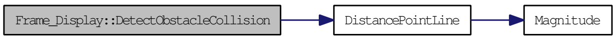
bot_no number of the bot whose collision is being evaluated

Definition at line 592 of file Frame_Display.c.

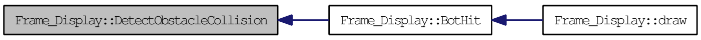
References Frame_Data::bot_hit, Frame_Data::bot_x, Frame_Data::bot_y, CB_-counter, data, DistancePointLine(), height_robot, info_refresh_count, LineEnd, LineStart, obs_2D, obs_2D_counter, obs_CB, obs_SP, obs_WL, Point, radius_of_robot, SP_counter, WL_counter, tagXY::X, and tagXY::Y.

Referenced by BotHit().

Here is the call graph for this function:



Here is the caller graph for this function:



8.3.3.11 void Frame_Display::position_khepera2 (float *pos_x*, float *pos_y*, double *orient*, int *colour*)

Renders KheperaII bots.

Renders the 3D structure of the KheperaII robots, which is contained herein.

Parameters:

pos_x x coordinate

pos_y y coordinate

orient orientation (direction it faces)

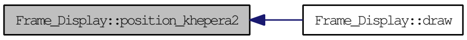
color non-default color when collision occurs

Definition at line 662 of file Frame_Display.c.

References bot_slices, bot_stacks, cyl_k2_specular, disk0_specular, disk1_specular, disk2_specular, height_robot, quad, radius_of_orient_disk, and radius_of_robot.

Referenced by draw().

Here is the caller graph for this function:



8.3.3.12 void Frame_Display::position_khepera3 (float *pos_x*, float *pos_y*, double *orient*, int *colour*)

Renders KheperaIII bots.

Renders the 3D structure of the KheperaIII robots, which is contained herein.

Parameters:

pos_x x coordinate

pos_y y coordinate

orient orientation (direction it faces)

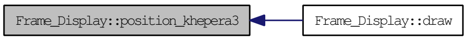
color non-default color when collision occurs

Definition at line 714 of file Frame_Display.c.

References bot_slices, bot_stacks, cyl_k3_specular, disk1_specular, disk2_specular, height_robot, no_of_lights, quad, radius_of_orient_disk, and radius_of_robot.

Referenced by draw().

Here is the caller graph for this function:



8.3.3.13 void Frame_Display::PositionCustomRobot (float *pos_x*, float *pos_y*, double *orient*, int *colour*)

Renders custom bots.

Renders the 3D structure of user customizable robots. Presently, this is a simplistic cylinder with a disk to show the direction it faces.

Parameters:

pos_x x coordinate

pos_y y coordinate

orient orientation (direction it faces)

color non-default color when collision occurs

Definition at line 776 of file Frame_Display.c.

References bot_slices, bot_stacks, cyl_k2_specular, disk1_specular, disk2_specular, height_robot, quad, radius_of_orient_disk, and radius_of_robot.

Referenced by draw().

Here is the caller graph for this function:

8.3.3.14 void Frame_Display::AutoUpdateEye () [inline]

Changes view angle during autoview.

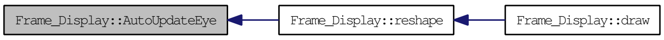
Decrements the angle made by the line connecting the camera location and the look-at position, with the X axis, projected on the floor.

Definition at line 30 of file Frame_Display.c.

References phi, and render_speed.

Referenced by reshape().

Here is the caller graph for this function:



8.3.3.15 void Frame_Display::init (void)

Initializes OpenGL simulation startup settings.

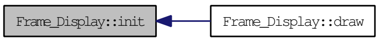
Contains settings for light sources (positions, etc.). Eliminates hidden surfaces. Also enables fog effect.

Definition at line 806 of file Frame_Display.c.

References cyl_shininess, Far, light0_ambient, light0_diffuse, light0_pos, light0_-specular, light1_pos, light1_specular, light2_pos, light2_specular, light3_pos, light3_-specular, no_of_lights, and quad.

Referenced by draw().

Here is the caller graph for this function:



8.3.3.16 void Frame_Display::reshape (int *w*, int *h*)

Resizes the simulation.

In case of resizing window, the OpenGL rendering is also resized to w times h pixel rectangle.

Parameters:

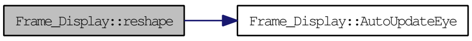
- ***w*** new width of rendering
- ***h*** new hight of rendering

Definition at line 297 of file Frame_Display.c.

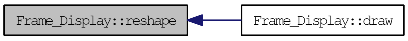
References aspect, AutoUpdateEye(), autoview, Frame_Data::bot_orient, Frame_-Data::bot_x, Frame_Data::bot_y, camera_on_bot, data, eyeX, eyeY, eyeZ, Far, height_robot, Near, phi, radianFactor, radius_camera_movement, radius_of_robot, theta, topview, upX, upY, upZ, view_centerX, view_centerY, and view_centerZ.

Referenced by draw().

Here is the call graph for this function:



Here is the caller graph for this function:



8.3.3.17 void Frame_Display::setIndex (int *val*) [inline]

Definition at line 115 of file Frame_Display.h.

References current_index.

Referenced by RobbitUI::cb_indexSlider_i().

Here is the caller graph for this function:

Frame_Display::setIndex
RobbitUI::cb_indexSlider_i

8.3.3.18 void Frame_Display::outputCharacter (float *x*, float *y*, float *z*, char ∗ *string*)

Renders a character at a specific location.

Parameters:

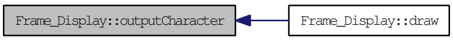
- ***x*** x coordinate of string to be rendered
- ***y*** y coordinate of string to be rendered
- ***z*** z coordinate of string to be rendered
- ***string*** string to be rendered

Definition at line 1084 of file Frame_Display.c.

References font.

Referenced by draw().

Here is the caller graph for this function:



8.3.3.19 void Frame_Display::setPlayPause ()

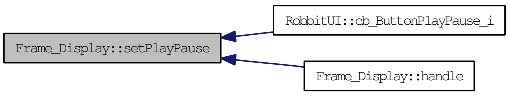
Swaps Play and Pause symbols in GUI Play/Pause button, on click.

Definition at line 77 of file Frame_Display.c.

References RobbitUI::ButtonPlayPause, and is_paused.

Referenced by RobbitUI::cb_ButtonPlayPause_i(), and handle().

Here is the caller graph for this function:



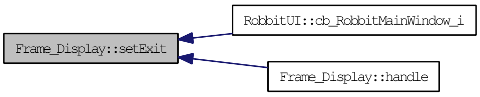
8.3.3.20 void Frame_Display::setExit () [inline]

Shows confirmation box before exit.

Definition at line 121 of file Frame_Display.h.

Referenced by RobbitUI::cb_RobbitMainWindow_i(), and handle().

Here is the caller graph for this function:



8.3.3.21 void Frame_Display::setStop ()

Stops simulation.

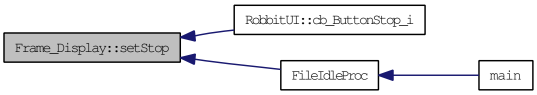
Clears file record index, repositions slider to 0, pauses the animation.

Definition at line 68 of file Frame_Display.c.

References RobbitUI::ButtonPlayPause, current_index, RobbitUI::indexSlider, and is_paused.

Referenced by RobbitUI::cb_ButtonStop_i(), and FileIdleProc().

Here is the caller graph for this function:



8.3.3.22 void Frame_Display::setReset () [inline]

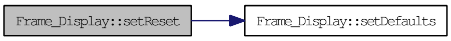
Resets all the settings to default.

Definition at line 125 of file Frame_Display.h.

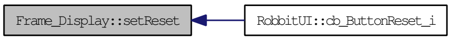
References setDefaults().

Referenced by RobbitUI::cb_ButtonReset_i().

Here is the call graph for this function:



Here is the caller graph for this function:



8.3.3.23 void Frame_Display::setBotNumbering (bool *val*) [inline]

Toggles the numbering of bot.

Parameters:

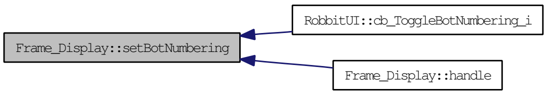
val value of the numbering bot (GUI)

Definition at line 131 of file Frame_Display.h.

References number_bots.

Referenced by RobbitUI::cb_ToggleBotNumbering_i(), and handle().

Here is the caller graph for this function:


Frame_Display::setBotNumbering
RobbitUI::cb_ToggleBotNumbering_i
Frame_Display::handle

8.3.3.24 void Frame_Display::setTrails (bool *val*)

Clears trail data.

If show_trails is set, it clears the trail data.

Parameters:

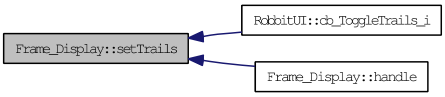
val value of the Show trails option (GUI)

Definition at line 51 of file Frame_Display.c.

References ball_trail, bot_trail, no_of_bots, and show_trails.

Referenced by RobbitUI::cb_ToggleTrails_i(), and handle().

Here is the caller graph for this function:



8.3.3.25 void Frame_Display::setObstacles (bool *val*) `[inline]`

Toggles the displaying of the Obstacles.

If set to display obstacle then reads obstacle from ascii file

Parameters:

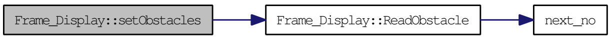
val value of the show obstacle option (GUI)

Definition at line 139 of file Frame_Display.h.

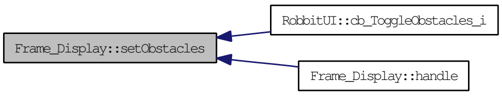
References ReadObstacle(), and show_obstacle.

Referenced by RobbitUI::cb_ToggleObstacles_i(), and handle().

Here is the call graph for this function:



Here is the caller graph for this function:



8.3.3.26 void Frame_Display::setTopView (bool *val*) `[inline]`

Toggles the top view mode.

Parameters:

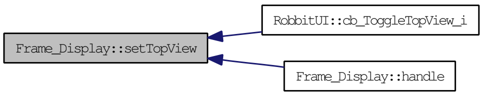
val value of the top view mode (GUI)

Definition at line 150 of file Frame_Display.h.

References topview.

Referenced by RobbitUI::cb_ToggleTopView_i(), and handle().

Here is the caller graph for this function:



8.3.3.27 void Frame_Display::setAutoView (bool *val*) [inline]

Toggles the Auto-rotate mode.

Parameters:

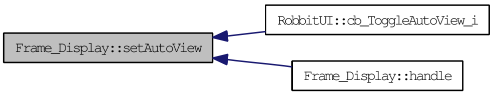
val value of the auto-rotate mode (GUI)

Definition at line 156 of file Frame_Display.h.

References autoview.

Referenced by RobbitUI::cb_ToggleAutoView_i(), and handle().

Here is the caller graph for this function:



8.3.3.28 void Frame_Display::selectBot (int *val*)

Sets bot structure variables.

Parameters:

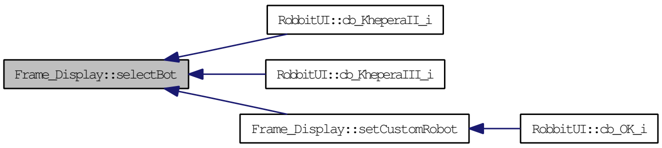
val type of bot: 1 = KheperaII; 2 = KheperaIII; 3 = Custom Robot

Definition at line 103 of file Frame_Display.c.

References bot_selected, bot_slices, height_robot, and radius_of_robot.

Referenced by RobbitUI::cb_KheperaII_i(), RobbitUI::cb_KheperaIII_i(), and setCustomRobot().

Here is the caller graph for this function:



8.3.3.29 void Frame_Display::setLights (int *val*) [inline]

Set no of lights to be used.

Parameters:

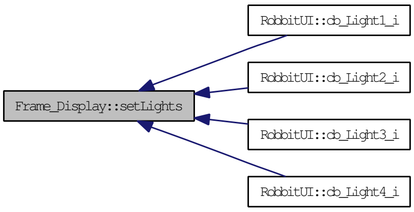
val no of lights (GUI)

Definition at line 163 of file Frame_Display.h.

References no_of_lights.

Referenced by RobbitUI::cb_Light1_i(), RobbitUI::cb_Light2_i(), RobbitUI::cb_Light3_i(), and RobbitUI::cb_Light4_i().

Here is the caller graph for this function:



8.3.3.30 void Frame_Display::setRenderSpeed (float *val*) [inline]

Sets the rendering speed.

Parameters:

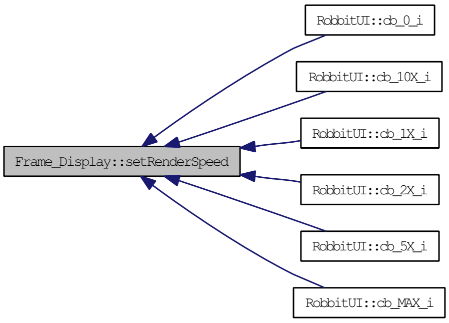
val value rendring speed in form of nX (GUI); 0: denotes maximum possible speed

Definition at line 169 of file Frame_Display.h.

References render_speed.

Referenced by RobbitUI::cb_0_i(), RobbitUI::cb_10X_i(), RobbitUI::cb_1X_i(), RobbitUI::cb_2X_i(), RobbitUI::cb_5X_i(), and RobbitUI::cb_MAX_i().

Here is the caller graph for this function:



8.3.3.31 void Frame_Display::setDefaults ()

Sets default values of variables.

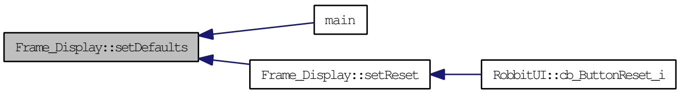
Developers may change them to their requirements.

Definition at line 122 of file Frame_Display.c.

References autoview, Frame_Data::bot_hit, Frame_Data::bot_orient, bot_selected, bot_slices, bot_stacks, Frame_Data::bot_vorient, Frame_Data::bot_vx, Frame_Data::bot_vy, Frame_Data::bot_x, Frame_Data::bot_y, RobbitUI::BotMenu, RobbitUI::ButtonPlayPause, camera_on_bot, coloured_steps, current_index, data, delta_zoom, Far, height_robot, is_paused, RobbitUI::LightMenu, max_x, max_y, min_x, min_y, Near, no_of_bots, no_of_lights, no_of_trail_data, number_bots, phi, radius_ball, radius_camera_movement, radius_of_robot, render_speed, RobbitUI::RenderSpeedMenu, show_obstacle, show_trails, small_disk_slices, small_disk_stacks, theta, RobbitUI::ToggleBotNumbering, RobbitUI::ToggleObstacles, RobbitUI::ToggleTopView, RobbitUI::ToggleTrails, topview, upX, upY, upZ, view_centerX, view_centerY, and view_centerZ.

Referenced by main(), and setReset().

Here is the caller graph for this function:



8.3.3.32 void Frame_Display::setCustomRobot (float *height*, float *radius*) [inline]

Sets dimension of Custom Robot (Cylindrical).

Also set the robot to be displayed to be Custom Robot

Parameters:

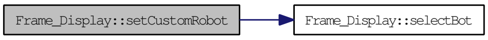
- ***height*** height of Custom Robot (GUI)
- ***radius*** radius of Custom Robot (GUI)

Definition at line 177 of file Frame_Display.h.

References height_robot, radius_of_robot, and selectBot().

Referenced by RobbitUI::cb_OK_i().

Here is the call graph for this function:



Here is the caller graph for this function:

Frame_Display::setCustomRobot
RobbitUI::cb_OK_i

8.3.3.33 void Frame_Display::setArena (float *maxx*, float *minx*, float *maxy*, float *miny*) [inline]

Sets Arena coordinates.

Parameters:

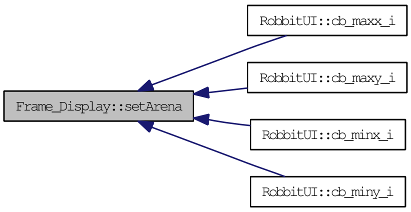
- ***maxx*** maximum x coordinate (GUI)
- ***minx*** minimum x coordinate (GUI)
- ***maxy*** maximum y coordinate (GUI)
- ***miny*** minimum y coordinate (GUI)

Definition at line 190 of file Frame_Display.h.

References max_x, max_y, min_x, and min_y.

Referenced by RobbitUI::cb_maxx_i(), RobbitUI::cb_maxy_i(), RobbitUI::cb_minx_i(), and RobbitUI::cb_miny_i().

Here is the caller graph for this function:



8.3.3.34 void Frame_Display::setBallRadius (float *val*) `[inline]`

Sets radius of ball.

Parameters:

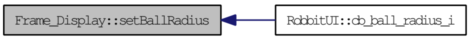
val radius of ball (GUI)

Definition at line 198 of file Frame_Display.h.

References radius_ball.

Referenced by RobbitUI::cb_ball_radius_i().

Here is the caller graph for this function:



8.3.3.35 void Frame_Display::setTrailPoints (float *val*) `[inline]`

Sets no of trail data to keep.

Parameters:

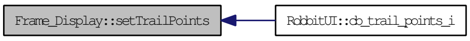
val number of trail data (GUI)

Definition at line 204 of file Frame_Display.h.

References no_of_trail_data.

Referenced by RobbitUI::cb_trail_points_i().

Here is the caller graph for this function:



8.3.3.36 void Frame_Display::setColouredSteps (float *val*) `[inline]`

Sets no of coloured steps (in case of any collision).

Parameters:

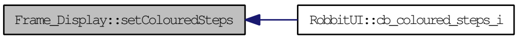
val number of coloured steps (GUI)

Definition at line 210 of file Frame_Display.h.

References coloured_steps.

Referenced by RobbitUI::cb_coloured_steps_i().

Here is the caller graph for this function:


Frame_Display::setColouredSteps
RobbitUI::cb_coloured_steps_i

8.3.3.37 void Frame_Display::initAdvWindow ()

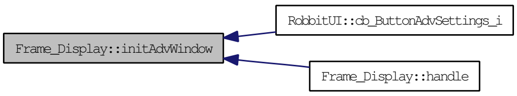
Sets configuration of Advanced Settings menu elements.

Definition at line 185 of file Frame_Display.c.

References RobbitUI::ball_radius, bot_slices, RobbitUI::coloured_steps, coloured_-steps, RobbitUI::graphics_quality, max_x, max_y, RobbitUI::maxx, RobbitUI::maxy, min_x, min_y, RobbitUI::minx, RobbitUI::miny, no_of_trail_data, radius_ball, and RobbitUI::trail_points.

Referenced by RobbitUI::cb_ButtonAdvSettings_i(), and handle().

Here is the caller graph for this function:


Frame_Display::initAdvWindow
RobbitUI::cb_ButtonAdvSettings_i
Frame_Display::handle

8.3.3.38 void Frame_Display::setGraphicsQuality (int *slices*, int *stacks*) [inline]

Sets graphic quality of objects rendered.

Parameters:

slices number of slices

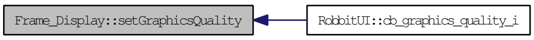
stacks number of stacks

Definition at line 218 of file Frame_Display.h.

References bot_slices, and bot_stacks.

Referenced by RobbitUI::cb_graphics_quality_i().

Here is the caller graph for this function:


Frame_Display::setGraphicsQuality
RobbitUI::cb_graphics_quality_i

8.3.4 Member Data Documentation

8.3.4.1 int Frame_Display::bot_selected `[private]`

Definition at line 29 of file Frame_Display.h.

Referenced by draw(), selectBot(), and setDefaults().

8.3.4.2 bool Frame_Display::number_bots `[private]`

Definition at line 31 of file Frame_Display.h.

Referenced by draw(), handle(), setBotNumbering(), and setDefaults().

8.3.4.3 bool Frame_Display::show_obstacle `[private]`

Definition at line 32 of file Frame_Display.h.

Referenced by BotHit(), draw(), handle(), ReadObstacle(), setDefaults(), and setObstacles().

8.3.4.4 bool Frame_Display::show_trails `[private]`

Definition at line 33 of file Frame_Display.h.

Referenced by draw(), handle(), setDefaults(), and setTrails().

8.3.4.5 bool Frame_Display::topview `[private]`

Definition at line 34 of file Frame_Display.h.

Referenced by handle(), reshape(), setDefaults(), and setTopView().

8.3.4.6 bool Frame_Display::autoview `[private]`

Definition at line 35 of file Frame_Display.h.

Referenced by handle(), reshape(), setAutoView(), and setDefaults().

8.3.4.7 float Frame_Display::obs_2D[20][2] `[private]`

Definition at line 37 of file Frame_Display.h.

Referenced by DetectObstacleCollision(), DrawObstacle(), and ReadObstacle().

8.3.4.8 float Frame_Display::obs_WL[20][5] `[private]`

Definition at line 38 of file Frame_Display.h.

Referenced by DetectObstacleCollision(), DrawObstacle(), and ReadObstacle().

8.3.4.9 float Frame_Display::obs_CB[20][3] `[private]`

Definition at line 39 of file Frame_Display.h.

Referenced by DetectObstacleCollision(), DrawObstacle(), and ReadObstacle().

8.3.4.10 float Frame_Display::obs_SP[20][3] `[private]`

Definition at line 40 of file Frame_Display.h.

Referenced by DetectObstacleCollision(), DrawObstacle(), and ReadObstacle().

8.3.4.11 int Frame_Display::obs_2D_counter `[private]`

Definition at line 42 of file Frame_Display.h.

Referenced by DetectObstacleCollision(), DrawObstacle(), and ReadObstacle().

8.3.4.12 int Frame_Display::WL_counter `[private]`

Definition at line 43 of file Frame_Display.h.

Referenced by DetectObstacleCollision(), DrawObstacle(), and ReadObstacle().

8.3.4.13 int Frame_Display::CB_counter `[private]`

Definition at line 44 of file Frame_Display.h.

Referenced by DetectObstacleCollision(), DrawObstacle(), and ReadObstacle().

8.3.4.14 int Frame_Display::SP_counter `[private]`

Definition at line 45 of file Frame_Display.h.

Referenced by DetectObstacleCollision(), DrawObstacle(), and ReadObstacle().

8.3.4.15 float Frame_Display::radius_ball `[private]`

Definition at line 46 of file Frame_Display.h.

Referenced by BotHit(), draw(), initAdvWindow(), setBallRadius(), and setDefaults().

8.3.4.16 int Frame_Display::no_of_trail_data `[private]`

Definition at line 47 of file Frame_Display.h.

Referenced by DrawTrails(), initAdvWindow(), setDefaults(), and setTrailPoints().

8.3.4.17 int Frame_Display::coloured_steps [private]

Definition at line 49 of file Frame_Display.h.

Referenced by BotHit(), initAdvWindow(), setColouredSteps(), and setDefaults().

8.3.4.18 std::deque<float> Frame_Display::bot_trail[no_of_bots][2] [private]

Definition at line 52 of file Frame_Display.h.

Referenced by DrawTrails(), and setTrails().

8.3.4.19 std::deque<float> Frame_Display::ball_trail[2] [private]

Definition at line 53 of file Frame_Display.h.

Referenced by DrawTrails(), and setTrails().

8.3.4.20 float Frame_Display::radius_camera_movement [private]

Definition at line 56 of file Frame_Display.h.

Referenced by handle(), reshape(), and setDefaults().

8.3.4.21 float Frame_Display::theta [private]

Definition at line 57 of file Frame_Display.h.

Referenced by CaptureScreenshot(), handle(), reshape(), and setDefaults().

8.3.4.22 float Frame_Display::phi [private]

Definition at line 58 of file Frame_Display.h.

Referenced by AutoUpdateEye(), CaptureScreenshot(), handle(), reshape(), and setDefaults().

8.3.4.23 float Frame_Display::eyeX [private]

Definition at line 59 of file Frame_Display.h.

Referenced by reshape().

8.3.4.24 float Frame_Display::eyeY `[private]`

Definition at line 59 of file Frame_Display.h.

Referenced by reshape().

8.3.4.25 float Frame_Display::eyeZ `[private]`

Definition at line 59 of file Frame_Display.h.

Referenced by reshape().

8.3.4.26 float Frame_Display::upX `[private]`

Definition at line 62 of file Frame_Display.h.

Referenced by reshape(), and setDefaults().

8.3.4.27 float Frame_Display::upY `[private]`

Definition at line 62 of file Frame_Display.h.

Referenced by reshape(), and setDefaults().

8.3.4.28 float Frame_Display::upZ `[private]`

Definition at line 62 of file Frame_Display.h.

Referenced by reshape(), and setDefaults().

8.3.4.29 float Frame_Display::aspect `[private]`

Definition at line 63 of file Frame_Display.h.

Referenced by reshape().

8.3.4.30 float Frame_Display::Near `[private]`

Definition at line 64 of file Frame_Display.h.

Referenced by reshape(), and setDefaults().

8.3.4.31 float Frame_Display::Far `[private]`

Definition at line 64 of file Frame_Display.h.

Referenced by init(), reshape(), and setDefaults().

8.3.4.32 int Frame_Display::camera_on_bot [private]

Definition at line 65 of file Frame_Display.h.

Referenced by handle(), reshape(), and setDefaults().

8.3.4.33 int Frame_Display::delta_zoom [private]

Definition at line 66 of file Frame_Display.h.

Referenced by handle(), and setDefaults().

8.3.4.34 int Frame_Display::no_of_lights [private]

Definition at line 67 of file Frame_Display.h.

Referenced by DrawFloor(), init(), position_khepera3(), setDefaults(), and setLights().

8.3.4.35 float Frame_Display::mouseX0 [private]

Definition at line 68 of file Frame_Display.h.

Referenced by handle().

8.3.4.36 float Frame_Display::mouseY0 [private]

Definition at line 68 of file Frame_Display.h.

Referenced by handle().

8.3.4.37 bool Frame_Display::mousePushValid [private]

Definition at line 69 of file Frame_Display.h.

Referenced by handle().

8.3.4.38 float Frame_Display::radius_of_robot [private]

Definition at line 71 of file Frame_Display.h.

Referenced by BotHit(), DetectObstacleCollision(), position_khepera2(), position_khepera3(), PositionCustomRobot(), reshape(), selectBot(), setCustomRobot(), and setDefaults().

8.3.4.39 float Frame_Display::height_robot [private]

Definition at line 72 of file Frame_Display.h.

Referenced by DetectObstacleCollision(), draw(), position_khepera2(), position_-khepera3(), PositionCustomRobot(), reshape(), selectBot(), setCustomRobot(), and setDefaults().

8.3.4.40 int Frame_Display::bot_slices [private]

Definition at line 73 of file Frame_Display.h.

Referenced by draw(), DrawObstacle(), initAdvWindow(), position_khepera2(), position_khepera3(), PositionCustomRobot(), selectBot(), setDefaults(), and setGraphicsQuality().

8.3.4.41 int Frame_Display::bot_stacks [private]

Definition at line 74 of file Frame_Display.h.

Referenced by draw(), position_khepera2(), position_khepera3(), PositionCustomRobot(), setDefaults(), and setGraphicsQuality().

8.3.4.42 int Frame_Display::small_disk_slices [private]

Definition at line 75 of file Frame_Display.h.

Referenced by DrawObstacle(), and setDefaults().

8.3.4.43 int Frame_Display::small_disk_stacks [private]

Definition at line 76 of file Frame_Display.h.

Referenced by DrawObstacle(), and setDefaults().

8.3.4.44 Frame_Data Frame_Display::data [private]

Definition at line 78 of file Frame_Display.h.

Referenced by BotHit(), CaptureScreenshot(), DetectObstacleCollision(), draw(), DrawTrails(), reshape(), setDefaults(), and UpdateFrame().

8.3.4.45 XY Frame_Display::LineStart [private]

Definition at line 80 of file Frame_Display.h.

Referenced by DetectObstacleCollision().

8.3.4.46 XY Frame_Display::LineEnd [private]

Definition at line 80 of file Frame_Display.h.

Referenced by DetectObstacleCollision().

8.3.4.47 XY Frame_Display::Point [private]

Definition at line 80 of file Frame_Display.h.

Referenced by DetectObstacleCollision().

8.3.4.48 bool Frame_Display::is_paused

Definition at line 86 of file Frame_Display.h.

Referenced by FileIdleProc(), FunctionIdleProc(), setDefaults(), setPlayPause(), and setStop().

8.3.4.49 float Frame_Display::render_speed

Definition at line 87 of file Frame_Display.h.

Referenced by AutoUpdateEye(), FileIdleProc(), setDefaults(), and setRenderSpeed().

8.3.4.50 int Frame_Display::current_index

Definition at line 88 of file Frame_Display.h.

Referenced by FileIdleProc(), setDefaults(), setIndex(), setStop(), and UpdateFrame().

8.3.4.51 float Frame_Display::view_centerX

Definition at line 89 of file Frame_Display.h.

Referenced by FileIdleProc(), FunctionIdleProc(), handle(), reshape(), and setDefaults().

8.3.4.52 float Frame_Display::view_centerY

Definition at line 89 of file Frame_Display.h.

Referenced by FileIdleProc(), FunctionIdleProc(), handle(), reshape(), and setDefaults().

8.3.4.53 float Frame_Display::view_centerZ

Definition at line 89 of file Frame_Display.h.

Referenced by FileIdleProc(), FunctionIdleProc(), handle(), reshape(), and setDefaults().

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

- Frame_Display.h

- Frame_Display.c

8.4 PNGImage Class Reference

Class to store information of a PNG image.

```
#include <WritePNG.h>
```

Public Member Functions

- PNGImage ()
- PNGImage (char ∗_filename, int _width, int _height)
- ∼PNGImage ()
- bool openFileRead ()

  Open PNG file for read.
- bool openFileWrite ()

  Open PNG file for write.
- bool isPNG (int bytesToCheck)

  Query if file is open and PNG.
- bool initReadStructs ()
- bool initWriteStructs ()
- bool writeHeader ()
- bool writeImage (void ∗bits)
- bool writeEnd ()

Private Attributes

- char ∗ filename
- FILE ∗ fp
- png_structp png_ptr
- png_infop info_ptr
- long width
- long height
- png_byte ∗ row
- int bytes_per_pixel
- long i
- long j
- GLubyte ∗ rgb
- long width_para

8.4.1 Detailed Description

Class to store information of a PNG image.

It contains operations for opening/closing PNG files, reading/writing structs, headers, writing image data, and other miscellaneous operations.

Definition at line 9 of file WritePNG.h.

8.4.2 Constructor & Destructor Documentation

8.4.2.1 PNGImage::PNGImage () `[inline]`

Definition at line 11 of file WritePNG.h.

8.4.2.2 PNGImage::PNGImage (char ∗ *_filename*, int *_width*, int *_height*) `[inline]`

Definition at line 12 of file WritePNG.h.

8.4.2.3 PNGImage::∼PNGImage ()

Definition at line 42 of file WritePNG.h.

References fp.

8.4.3 Member Function Documentation

8.4.3.1 bool PNGImage::openFileRead ()

Open PNG file for read.

Returns:

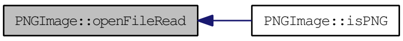
false, if already open, or due to other error; true, if file opened successfully

Definition at line 51 of file WritePNG.h.

References filename, and fp.

Referenced by isPNG().

Here is the caller graph for this function:



8.4.3.2 bool PNGImage::openFileWrite ()

Open PNG file for write.

Returns:

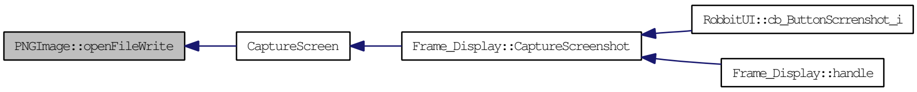
false, if already open, or due to other error; true, if file opened successfully

Definition at line 70 of file WritePNG.h.

References filename, and fp.

Referenced by CaptureScreen().

Here is the caller graph for this function:


PNGImage::openFileWrite
CaptureScreen
Frame_Display::CaptureScreenshot
RobbitUI::cb_ButtonScrrenshot_i
Frame_Display::handle

8.4.3.3 bool PNGImage::isPNG (int *bytesToCheck* = 8)

Query if file is open and PNG.

Parameters:

bytesToCheck = 8, initial checkbytes for valid PNG file.

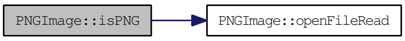
Returns:

true, if file is a valid PNG file; false, if invalid, or due to other error;

Definition at line 90 of file WritePNG.h.

References fp, and openFileRead().

Here is the call graph for this function:



8.4.3.4 bool PNGImage::initReadStructs ()

Definition at line 121 of file WritePNG.h.

References info_ptr, and png_ptr.

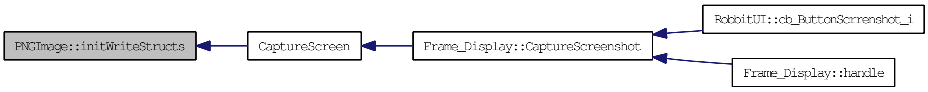
8.4.3.5 bool PNGImage::initWriteStructs ()

Definition at line 151 of file WritePNG.h.

References fp, info_ptr, and png_ptr.

Referenced by CaptureScreen().

Here is the caller graph for this function:


PNGImage::initWriteStructs
CaptureScreen
Frame_Display::CaptureScreenshot
RobbitUI::cb_ButtonScrrenshot_i
Frame_Display::handle

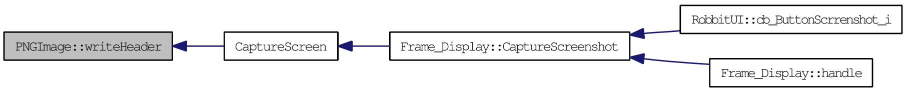
8.4.3.6 bool PNGImage::writeHeader ()

Definition at line 183 of file WritePNG.h.

References height, info_ptr, png_ptr, and width.

Referenced by CaptureScreen().

Here is the caller graph for this function:


PNGImage::writeHeader
CaptureScreen
Frame_Display::CaptureScreenshot
RobbitUI::cb_ButtonScrrenshot_i
Frame_Display::handle

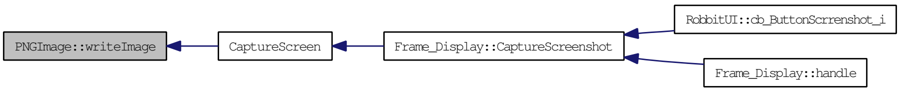
8.4.3.7 bool PNGImage::writeImage (void ∗ *bits*)

Definition at line 199 of file WritePNG.h.

References bytes_per_pixel, height, i, info_ptr, j, png_ptr, rgb, row, width, and width_-para.

Referenced by CaptureScreen().

Here is the caller graph for this function:


PNGImage::writeImage
CaptureScreen
Frame_Display::CaptureScreenshot
RobbitUI::cb_ButtonScrrenshot_i
Frame_Display::handle

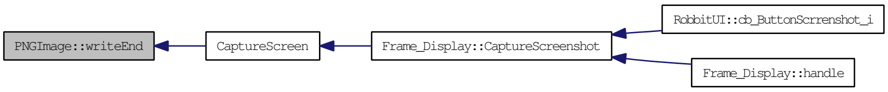
8.4.3.8 bool PNGImage::writeEnd ()

Definition at line 226 of file WritePNG.h.

References info_ptr, and png_ptr.

Referenced by CaptureScreen().

Here is the caller graph for this function:


PNGImage::writeEnd
CaptureScreen
Frame_Display::CaptureScreenshot
RobbitUI::cb_ButtonScrrenshot_i
Frame_Display::handle

8.4.4 Member Data Documentation

8.4.4.1 char∗ PNGImage::filename [private]

Definition at line 29 of file WritePNG.h.

Referenced by openFileRead(), and openFileWrite().

8.4.4.2 FILE∗ PNGImage::fp [private]

Definition at line 30 of file WritePNG.h.

Referenced by initWriteStructs(), isPNG(), openFileRead(), openFileWrite(), and ∼PNGImage().

8.4.4.3 png_structp PNGImage::png_ptr [private]

Definition at line 31 of file WritePNG.h.

Referenced by initReadStructs(), initWriteStructs(), writeEnd(), writeHeader(), and writeImage().

8.4.4.4 png_infop PNGImage::info_ptr [private]

Definition at line 32 of file WritePNG.h.

Referenced by initReadStructs(), initWriteStructs(), writeEnd(), writeHeader(), and writeImage().

8.4.4.5 long PNGImage::width [private]

Definition at line 33 of file WritePNG.h.

Referenced by writeHeader(), and writeImage().

8.4.4.6 long PNGImage::height [private]

Definition at line 33 of file WritePNG.h.

Referenced by writeHeader(), and writeImage().

8.4.4.7 png_byte* PNGImage::row [private]

Definition at line 34 of file WritePNG.h.

Referenced by writeImage().

8.4.4.8 int PNGImage::bytes_per_pixel [private]

Definition at line 35 of file WritePNG.h.

Referenced by writeImage().

8.4.4.9 long PNGImage::i [private]

Definition at line 36 of file WritePNG.h.

Referenced by writeImage().

8.4.4.10 long PNGImage::j [private]

Definition at line 36 of file WritePNG.h.

Referenced by writeImage().

8.4.4.11 GLubyte* PNGImage::rgb [private]

Definition at line 37 of file WritePNG.h.

Referenced by writeImage().

8.4.4.12 long PNGImage::width_para [private]

Definition at line 38 of file WritePNG.h.

Referenced by writeImage().

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

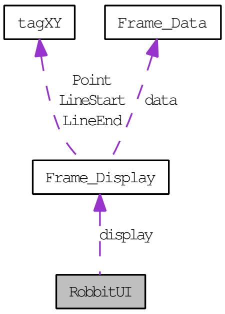
- WritePNG.h

8.5 RobbitUI Class Reference

CLass to build the whole GUI.

`#include <robbitGUI.h>`

Collaboration diagram for RobbitUI:



Public Member Functions

- RobbitUI ()
- void show ()

Public Attributes

- Fl_Double_Window ∗ RobbitMainWindow
- Fl_Value_Slider ∗ indexSlider
- Frame_Display ∗ display
- Fl_Button ∗ ButtonAbout
- Fl_Light_Button ∗ ToggleAutoView
- Fl_Light_Button ∗ ToggleTopView
- Fl_Light_Button ∗ ToggleTrails
- Fl_Light_Button ∗ ToggleObstacles
- Fl_Light_Button ∗ ToggleBotNumbering
- Fl_Button ∗ ButtonAdvSettings
- Fl_Button ∗ ButtonScrrenshot
- Fl_Choice ∗ RenderSpeedMenu
- Fl_Choice ∗ BotMenu
- Fl_Choice ∗ LightMenu
- Fl_Button ∗ ButtonReset
- Fl_Button ∗ ButtonPlayPause
- Fl_Button ∗ ButtonStop
- Fl_Double_Window ∗ AboutWindow
- Fl_Help_View ∗ textrobbit
- Fl_Double_Window ∗ SplashWindow

- Fl_Help_View ∗ htmlSplash
- Fl_Double_Window ∗ custom_robot_window
- Fl_Input ∗ height
- Fl_Input ∗ radius
- Fl_Double_Window ∗ AdvSettingsWindow
- Fl_Input ∗ maxy
- Fl_Input ∗ miny
- Fl_Input ∗ maxx
- Fl_Input ∗ minx
- Fl_Input ∗ ball_radius
- Fl_Input ∗ trail_points
- Fl_Input ∗ coloured_steps
- Fl_Slider ∗ graphics_quality

Static Public Attributes

- static Fl_Menu_Item menu_RenderSpeedMenu [ ]
- static Fl_Menu_Item menu_BotMenu [ ]
- static Fl_Menu_Item ∗ KheperaII = RobbitUI::menu_BotMenu + 0
- static Fl_Menu_Item ∗ KheperaIII = RobbitUI::menu_BotMenu + 1
- static Fl_Menu_Item ∗ Custom = RobbitUI::menu_BotMenu + 2
- static Fl_Menu_Item menu_LightMenu [ ]
- static Fl_Menu_Item ∗ Light1 = RobbitUI::menu_LightMenu + 0
- static Fl_Menu_Item ∗ Light2 = RobbitUI::menu_LightMenu + 1
- static Fl_Menu_Item ∗ Light3 = RobbitUI::menu_LightMenu + 2
- static Fl_Menu_Item ∗ Light4 = RobbitUI::menu_LightMenu + 3

Private Member Functions

- void cb_RobbitMainWindow_i (Fl_Double_Window ∗, void ∗)
- void cb_indexSlider_i (Fl_Value_Slider ∗, void ∗)
- void cb_ButtonAbout_i (Fl_Button ∗, void ∗)
- void cb_ToggleAutoView_i (Fl_Light_Button ∗, void ∗)
- void cb_ToggleTopView_i (Fl_Light_Button ∗, void ∗)
- void cb_ToggleTrails_i (Fl_Light_Button ∗, void ∗)
- void cb_ToggleObstacles_i (Fl_Light_Button ∗, void ∗)
- void cb_ToggleBotNumbering_i (Fl_Light_Button ∗, void ∗)
- void cb_ButtonAdvSettings_i (Fl_Button ∗, void ∗)
- void cb_ButtonScrrenshot_i (Fl_Button ∗, void ∗)
- void cb_0_i (Fl_Menu_ ∗, void ∗)
- void cb_1X_i (Fl_Menu_ ∗, void ∗)
- void cb_2X_i (Fl_Menu_ ∗, void ∗)
- void cb_5X_i (Fl_Menu_ ∗, void ∗)
- void cb_10X_i (Fl_Menu_ ∗, void ∗)
- void cb_MAX_i (Fl_Menu_ ∗, void ∗)

- void cb_KheperaII_i (Fl_Menu_ ∗, void ∗)
- void cb_KheperaIII_i (Fl_Menu_ ∗, void ∗)
- void cb_Custom_i (Fl_Menu_ ∗, void ∗)
- void cb_Light1_i (Fl_Menu_ ∗, void ∗)
- void cb_Light2_i (Fl_Menu_ ∗, void ∗)
- void cb_Light3_i (Fl_Menu_ ∗, void ∗)
- void cb_Light4_i (Fl_Menu_ ∗, void ∗)
- void cb_ButtonReset_i (Fl_Button ∗, void ∗)
- void cb_ButtonPlayPause_i (Fl_Button ∗, void ∗)
- void cb_ButtonStop_i (Fl_Button ∗, void ∗)
- void cb_About_i (Fl_Button ∗, void ∗)
- void cb_Close_i (Fl_Return_Button ∗, void ∗)
- void cb_OK_i (Fl_Return_Button ∗, void ∗)
- void cb_maxy_i (Fl_Input ∗, void ∗)
- void cb_miny_i (Fl_Input ∗, void ∗)
- void cb_maxx_i (Fl_Input ∗, void ∗)
- void cb_minx_i (Fl_Input ∗, void ∗)
- void cb_ball_radius_i (Fl_Input ∗, void ∗)
- void cb_trail_points_i (Fl_Input ∗, void ∗)
- void cb_coloured_steps_i (Fl_Input ∗, void ∗)
- void cb_graphics_quality_i (Fl_Slider ∗, void ∗)
- void cb_OK1_i (Fl_Return_Button ∗, void ∗)

Static Private Member Functions

- static void cb_RobbitMainWindow (Fl_Double_Window ∗, void ∗)
- static void cb_indexSlider (Fl_Value_Slider ∗, void ∗)
- static void cb_ButtonAbout (Fl_Button ∗, void ∗)
- static void cb_ToggleAutoView (Fl_Light_Button ∗, void ∗)
- static void cb_ToggleTopView (Fl_Light_Button ∗, void ∗)
- static void cb_ToggleTrails (Fl_Light_Button ∗, void ∗)
- static void cb_ToggleObstacles (Fl_Light_Button ∗, void ∗)
- static void cb_ToggleBotNumbering (Fl_Light_Button ∗, void ∗)
- static void cb_ButtonAdvSettings (Fl_Button ∗, void ∗)
- static void cb_ButtonScrrenshot (Fl_Button ∗, void ∗)
- static void cb_0 (Fl_Menu_ ∗, void ∗)
- static void cb_1X (Fl_Menu_ ∗, void ∗)
- static void cb_2X (Fl_Menu_ ∗, void ∗)
- static void cb_5X (Fl_Menu_ ∗, void ∗)
- static void cb_10X (Fl_Menu_ ∗, void ∗)
- static void cb_MAX (Fl_Menu_ ∗, void ∗)
- static void cb_KheperaII (Fl_Menu_ ∗, void ∗)
- static void cb_KheperaIII (Fl_Menu_ ∗, void ∗)
- static void cb_Custom (Fl_Menu_ ∗, void ∗)
- static void cb_Light1 (Fl_Menu_ ∗, void ∗)
- static void cb_Light2 (Fl_Menu_ ∗, void ∗)

- static void cb_Light3 (Fl_Menu_ ∗, void ∗)
- static void cb_Light4 (Fl_Menu_ ∗, void ∗)
- static void cb_ButtonReset (Fl_Button ∗, void ∗)
- static void cb_ButtonPlayPause (Fl_Button ∗, void ∗)
- static void cb_ButtonStop (Fl_Button ∗, void ∗)
- static void cb_About (Fl_Button ∗, void ∗)
- static void cb_Close (Fl_Return_Button ∗, void ∗)
- static void cb_OK (Fl_Return_Button ∗, void ∗)
- static void cb_maxy (Fl_Input ∗, void ∗)
- static void cb_miny (Fl_Input ∗, void ∗)
- static void cb_maxx (Fl_Input ∗, void ∗)
- static void cb_minx (Fl_Input ∗, void ∗)
- static void cb_ball_radius (Fl_Input ∗, void ∗)
- static void cb_trail_points (Fl_Input ∗, void ∗)
- static void cb_coloured_steps (Fl_Input ∗, void ∗)
- static void cb_graphics_quality (Fl_Slider ∗, void ∗)
- static void cb_OK1 (Fl_Return_Button ∗, void ∗)

8.5.1 Detailed Description

CLass to build the whole GUI.

This class is created with FLUID (Fast Light User-Interface Designer). It also monitors all the events and callback

Definition at line 45 of file robbitGUI.h.

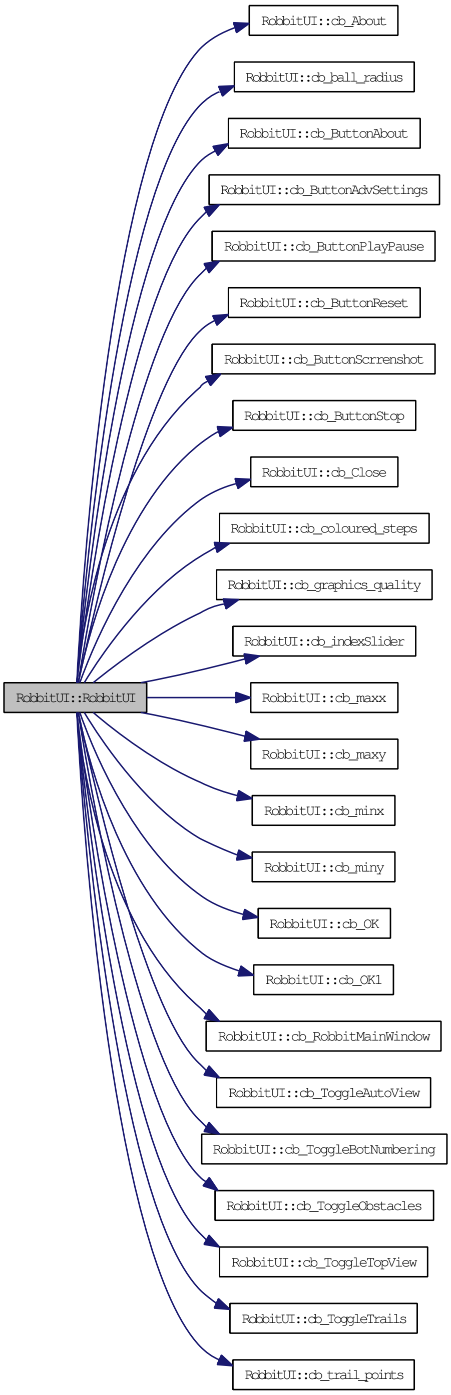
8.5.2 Constructor & Destructor Documentation

8.5.2.1 RobbitUI::RobbitUI ()

Definition at line 321 of file robbitGUI.cxx.

References AboutWindow, AdvSettingsWindow, ball_radius, BotMenu, ButtonAbout, ButtonAdvSettings, ButtonPlayPause, ButtonReset, ButtonScrrenshot, ButtonStop, cb_About(), cb_ball_radius(), cb_ButtonAbout(), cb_ButtonAdvSettings(), cb_ButtonPlayPause(), cb_ButtonReset(), cb_ButtonScrrenshot(), cb_ButtonStop(), cb_Close(), cb_coloured_steps(), cb_graphics_quality(), cb_indexSlider(), cb_maxx(), cb_maxy(), cb_minx(), cb_miny(), cb_OK(), cb_OK1(), cb_RobbitMainWindow(), cb_ToggleAutoView(), cb_ToggleBotNumbering(), cb_ToggleObstacles(), cb_ToggleTopView(), cb_ToggleTrails(), cb_trail_points(), coloured_steps, custom_robot_window, display, graphics_quality, height, htmlSplash, indexSlider, LightMenu, maxx, maxy, menu_BotMenu, menu_LightMenu, menu_RenderSpeedMenu, minx, miny, radius, RenderSpeedMenu, RobbitMainWindow, SplashWindow, textrobbit, ToggleAutoView, ToggleBotNumbering, ToggleObstacles, ToggleTopView, ToggleTrails, and trail_points.

Here is the call graph for this function:


RobbitUI::RobbitUI
RobbitUI::cb_About
RobbitUI::cb_ball_radius
RobbitUI::cb_ButtonAbout
RobbitUI::cb_ButtonAdvSettings
RobbitUI::cb_ButtonPlayPause
RobbitUI::cb_ButtonReset
RobbitUI::cb_ButtonScrrenshot
RobbitUI::cb_ButtonStop
RobbitUI::cb_Close
RobbitUI::cb_coloured_steps
RobbitUI::cb_graphics_quality
RobbitUI::cb_indexSlider
RobbitUI::cb_maxx
RobbitUI::cb_maxy
RobbitUI::cb_minx
RobbitUI::cb_miny
RobbitUI::cb_OK
RobbitUI::cb_OK1
RobbitUI::cb_RobbitMainWindow
RobbitUI::cb_ToggleAutoView
RobbitUI::cb_ToggleBotNumbering
RobbitUI::cb_ToggleObstacles
RobbitUI::cb_ToggleTopView
RobbitUI::cb_ToggleTrails
RobbitUI::cb_trail_points

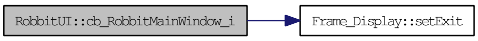
8.5.3 Member Function Documentation

8.5.3.1 void RobbitUI::cb_RobbitMainWindow_i (Fl_Double_Window ∗, void ∗) [private]

Definition at line 5 of file robbitGUI.cxx.

References display, and Frame_Display::setExit().

Here is the call graph for this function:



8.5.3.2 void RobbitUI::cb_RobbitMainWindow (Fl_Double_Window ∗ *o*, void ∗ *v*) [static, private]

Definition at line 8 of file robbitGUI.cxx.

Referenced by RobbitUI().

Here is the caller graph for this function:

RobbitUI::cb_RobbitMainWindow
RobbitUI::RobbitUI

8.5.3.3 void RobbitUI::cb_indexSlider_i (Fl_Value_Slider ∗ *o*, void ∗) [private]

Definition at line 12 of file robbitGUI.cxx.

References display, and Frame_Display::setIndex().

Here is the call graph for this function:

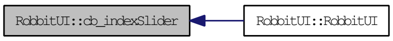
RobbitUI::cb_indexSlider_i
Frame_Display::setIndex

8.5.3.4 void RobbitUI::cb_indexSlider (Fl_Value_Slider ∗ *o*, void ∗ *v*) [static, private]

Definition at line 15 of file robbitGUI.cxx.

Referenced by RobbitUI().

Here is the caller graph for this function:



8.5.3.5 void RobbitUI::cb_ButtonAbout_i (Fl_Button ∗, void ∗) [private]

Definition at line 19 of file robbitGUI.cxx.

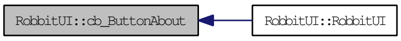
References AboutWindow, and textrobbit.

8.5.3.6 void RobbitUI::cb_ButtonAbout (Fl_Button ∗ *o*, void ∗ *v*) [static, private]

Definition at line 23 of file robbitGUI.cxx.

Referenced by RobbitUI().

Here is the caller graph for this function:

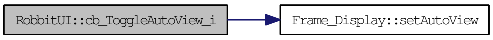


8.5.3.7 void RobbitUI::cb_ToggleAutoView_i (Fl_Light_Button ∗ *o*, void ∗) [private]

Definition at line 27 of file robbitGUI.cxx.

References display, Frame_Display::setAutoView(), and ToggleTopView.

Here is the call graph for this function:

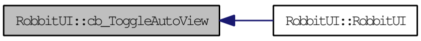


8.5.3.8 void RobbitUI::cb_ToggleAutoView (Fl_Light_Button ∗ *o*, void ∗ *v*) [static, private]

Definition at line 34 of file robbitGUI.cxx.

Referenced by RobbitUI().

Here is the caller graph for this function:

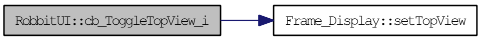


8.5.3.9 void RobbitUI::cb_ToggleTopView_i (Fl_Light_Button ∗ *o*, void ∗) `[private]`

Definition at line 38 of file robbitGUI.cxx.

References display, Frame_Display::setTopView(), and ToggleAutoView.

Here is the call graph for this function:



8.5.3.10 void RobbitUI::cb_ToggleTopView (Fl_Light_Button ∗ *o*, void ∗ *v*) `[static, private]`

Definition at line 45 of file robbitGUI.cxx.

Referenced by RobbitUI().

Here is the caller graph for this function:

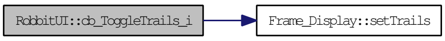
RobbitUI::cb_ToggleTopView
RobbitUI::RobbitUI

8.5.3.11 void RobbitUI::cb_ToggleTrails_i (Fl_Light_Button ∗ *o*, void ∗) `[private]`

Definition at line 49 of file robbitGUI.cxx.

References display, and Frame_Display::setTrails().

Here is the call graph for this function:

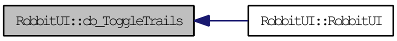


8.5.3.12 void RobbitUI::cb_ToggleTrails (Fl_Light_Button ∗ *o*, void ∗ *v*) `[static, private]`

Definition at line 52 of file robbitGUI.cxx.

Referenced by RobbitUI().

Here is the caller graph for this function:

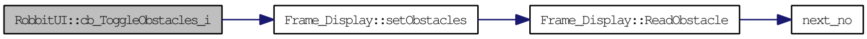


8.5.3.13 void RobbitUI::cb_ToggleObstacles_i (Fl_Light_Button ∗ *o*, void ∗) [private]

Definition at line 56 of file robbitGUI.cxx.

References display, and Frame_Display::setObstacles().

Here is the call graph for this function:

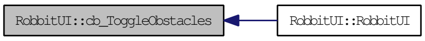

RobbitUI::cb_ToggleObstacles_i
Frame_Display::setObstacles
Frame_Display::ReadObstacle
next_no

8.5.3.14 void RobbitUI::cb_ToggleObstacles (Fl_Light_Button ∗ *o*, void ∗ *v*) [static, private]

Definition at line 59 of file robbitGUI.cxx.

Referenced by RobbitUI().

Here is the caller graph for this function:

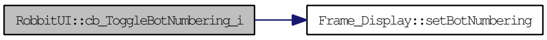


8.5.3.15 void RobbitUI::cb_ToggleBotNumbering_i (Fl_Light_Button ∗ *o*, void ∗) [private]

Definition at line 63 of file robbitGUI.cxx.

References display, and Frame_Display::setBotNumbering().

Here is the call graph for this function:

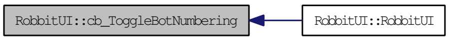

RobbitUI::cb_ToggleBotNumbering_i
Frame_Display::setBotNumbering

8.5.3.16 void RobbitUI::cb_ToggleBotNumbering (Fl_Light_Button ∗ *o*, void ∗ *v*) [static, private]

Definition at line 66 of file robbitGUI.cxx.

Referenced by RobbitUI().

Here is the caller graph for this function:

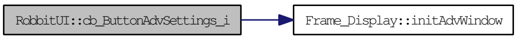


8.5.3.17 void RobbitUI::cb_ButtonAdvSettings_i (Fl_Button *, void *) [private]

Definition at line 70 of file robbitGUI.cxx.

References AdvSettingsWindow, display, and Frame_Display::initAdvWindow().

Here is the call graph for this function:

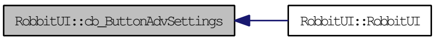

RobbitUI::cb_ButtonAdvSettings_i
Frame_Display::initAdvWindow

8.5.3.18 void RobbitUI::cb_ButtonAdvSettings (Fl_Button * *o*, void * *v*) [static, private]

Definition at line 74 of file robbitGUI.cxx.

Referenced by RobbitUI().

Here is the caller graph for this function:

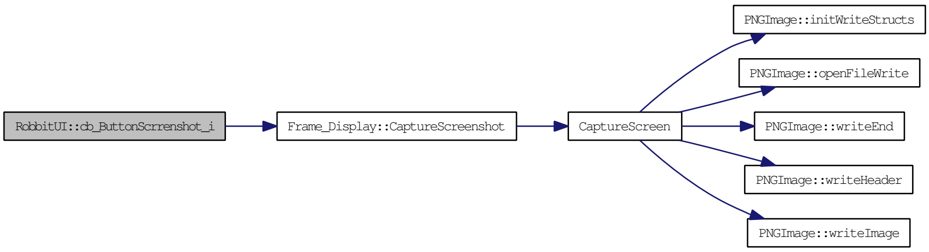


8.5.3.19 void RobbitUI::cb_ButtonScrrenshot_i (Fl_Button *, void *) [private]

Definition at line 78 of file robbitGUI.cxx.

References Frame_Display::CaptureScreenshot(), and display.

Here is the call graph for this function:



8.5.3.20 void RobbitUI::cb_ButtonScrrenshot (Fl_Button * *o*, void * *v*) [static, private]

Definition at line 81 of file robbitGUI.cxx.

Referenced by RobbitUI().

Here is the caller graph for this function:

8.5.3.21 void RobbitUI::cb_0_i (Fl_Menu_ $*$, void $*$) [private]

Definition at line 85 of file robbitGUI.cxx.

References display, and Frame_Display::setRenderSpeed().

Here is the call graph for this function:

8.5.3.22 void RobbitUI::cb_0 (Fl_Menu_ $*$ *o*, void $*$ *v*) [static, private]

Definition at line 88 of file robbitGUI.cxx.

8.5.3.23 void RobbitUI::cb_1X_i (Fl_Menu_ $*$, void $*$) [private]

Definition at line 92 of file robbitGUI.cxx.

References display, and Frame_Display::setRenderSpeed().

Here is the call graph for this function:

8.5.3.24 void RobbitUI::cb_1X (Fl_Menu_ $*$ *o*, void $*$ *v*) [static, private]

Definition at line 95 of file robbitGUI.cxx.

8.5.3.25 void RobbitUI::cb_2X_i (Fl_Menu_ $*$, void $*$) [private]

Definition at line 99 of file robbitGUI.cxx.

References display, and Frame_Display::setRenderSpeed().

Here is the call graph for this function:

8.5.3.26 void RobbitUI::cb_2X (Fl_Menu_ ∗ *o*, void ∗ *v*) [static, private]

Definition at line 102 of file robbitGUI.cxx.

8.5.3.27 void RobbitUI::cb_5X_i (Fl_Menu_ ∗, void ∗) [private]

Definition at line 106 of file robbitGUI.cxx.

References display, and Frame_Display::setRenderSpeed().

Here is the call graph for this function:

8.5.3.28 void RobbitUI::cb_5X (Fl_Menu_ ∗ *o*, void ∗ *v*) [static, private]

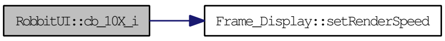
Definition at line 109 of file robbitGUI.cxx.

8.5.3.29 void RobbitUI::cb_10X_i (Fl_Menu_ ∗, void ∗) [private]

Definition at line 113 of file robbitGUI.cxx.

References display, and Frame_Display::setRenderSpeed().

Here is the call graph for this function:



8.5.3.30 void RobbitUI::cb_10X (Fl_Menu_ ∗ *o*, void ∗ *v*) [static, private]

Definition at line 116 of file robbitGUI.cxx.

8.5.3.31 void RobbitUI::cb_MAX_i (Fl_Menu_ $*$, void $*$) [private]

Definition at line 120 of file robbitGUI.cxx.

References display, and Frame_Display::setRenderSpeed().

Here is the call graph for this function:

8.5.3.32 void RobbitUI::cb_MAX (Fl_Menu_ $*$ *o*, void $*$ *v*) [static, private]

Definition at line 123 of file robbitGUI.cxx.

8.5.3.33 void RobbitUI::cb_KheperaII_i (Fl_Menu_ $*$, void $*$) [private]

Definition at line 137 of file robbitGUI.cxx.

References display, and Frame_Display::selectBot().

Here is the call graph for this function:

8.5.3.34 void RobbitUI::cb_KheperaII (Fl_Menu_ $*$ *o*, void $*$ *v*) [static, private]

Definition at line 140 of file robbitGUI.cxx.

8.5.3.35 void RobbitUI::cb_KheperaIII_i (Fl_Menu_ $*$, void $*$) [private]

Definition at line 144 of file robbitGUI.cxx.

References display, and Frame_Display::selectBot().

Here is the call graph for this function:

8.5.3.36 void RobbitUI::cb_KheperaIII (Fl_Menu_ ∗ *o*, void ∗ *v*) [static, private]

Definition at line 147 of file robbitGUI.cxx.

8.5.3.37 void RobbitUI::cb_Custom_i (Fl_Menu_ ∗, void ∗) [private]

Definition at line 151 of file robbitGUI.cxx.

References custom_robot_window.

8.5.3.38 void RobbitUI::cb_Custom (Fl_Menu_ ∗ *o*, void ∗ *v*) [static, private]

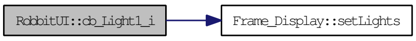
Definition at line 154 of file robbitGUI.cxx.

8.5.3.39 void RobbitUI::cb_Light1_i (Fl_Menu_ ∗, void ∗) [private]

Definition at line 168 of file robbitGUI.cxx.

References display, and Frame_Display::setLights().

Here is the call graph for this function:



8.5.3.40 void RobbitUI::cb_Light1 (Fl_Menu_ ∗ *o*, void ∗ *v*) [static, private]

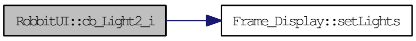
Definition at line 171 of file robbitGUI.cxx.

8.5.3.41 void RobbitUI::cb_Light2_i (Fl_Menu_ ∗, void ∗) [private]

Definition at line 175 of file robbitGUI.cxx.

References display, and Frame_Display::setLights().

Here is the call graph for this function:



8.5.3.42 void RobbitUI::cb_Light2 (Fl_Menu_ ∗ *o*, void ∗ *v*) [static, private]

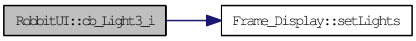
Definition at line 178 of file robbitGUI.cxx.

8.5.3.43 void RobbitUI::cb_Light3_i (Fl_Menu_ ∗, void ∗) [private]

Definition at line 182 of file robbitGUI.cxx.

References display, and Frame_Display::setLights().

Here is the call graph for this function:



8.5.3.44 void RobbitUI::cb_Light3 (Fl_Menu_ ∗ *o*, void ∗ *v*) [static, private]

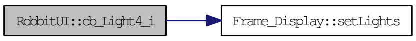
Definition at line 185 of file robbitGUI.cxx.

8.5.3.45 void RobbitUI::cb_Light4_i (Fl_Menu_ ∗, void ∗) [private]

Definition at line 189 of file robbitGUI.cxx.

References display, and Frame_Display::setLights().

Here is the call graph for this function:



8.5.3.46 void RobbitUI::cb_Light4 (Fl_Menu_ ∗ *o*, void ∗ *v*) [static, private]

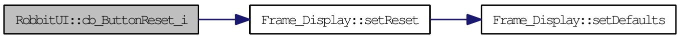
Definition at line 192 of file robbitGUI.cxx.

8.5.3.47 void RobbitUI::cb_ButtonReset_i (Fl_Button ∗, void ∗) [private]

Definition at line 208 of file robbitGUI.cxx.

References display, and Frame_Display::setReset().

Here is the call graph for this function:

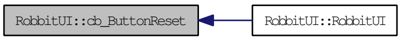


8.5.3.48 void RobbitUI::cb_ButtonReset (Fl_Button * *o*, void * *v*) [static, private]

Definition at line 211 of file robbitGUI.cxx.

Referenced by RobbitUI().

Here is the caller graph for this function:

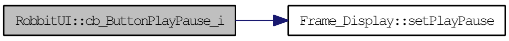


8.5.3.49 void RobbitUI::cb_ButtonPlayPause_i (Fl_Button *, void *) [private]

Definition at line 215 of file robbitGUI.cxx.

References display, and Frame_Display::setPlayPause().

Here is the call graph for this function:



8.5.3.50 void RobbitUI::cb_ButtonPlayPause (Fl_Button * *o*, void * *v*) [static, private]

Definition at line 218 of file robbitGUI.cxx.

Referenced by RobbitUI().

Here is the caller graph for this function:

8.5.3.51 void RobbitUI::cb_ButtonStop_i (Fl_Button *, void *) [private]

Definition at line 222 of file robbitGUI.cxx.

References display, and Frame_Display::setStop().

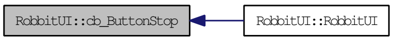
Here is the call graph for this function:

8.5.3.52 void RobbitUI::cb_ButtonStop (Fl_Button ∗ *o*, void ∗ *v*) [static, private]

Definition at line 225 of file robbitGUI.cxx.

Referenced by RobbitUI().

Here is the caller graph for this function:



8.5.3.53 void RobbitUI::cb_About_i (Fl_Button ∗, void ∗) [private]

Definition at line 229 of file robbitGUI.cxx.

References AboutWindow, SplashWindow, start_time, and textrobbit.

8.5.3.54 void RobbitUI::cb_About (Fl_Button ∗ *o*, void ∗ *v*) [static, private]

Definition at line 236 of file robbitGUI.cxx.

Referenced by RobbitUI().

Here is the caller graph for this function:

8.5.3.55 void RobbitUI::cb_Close_i (Fl_Return_Button ∗, void ∗) [private]

Definition at line 240 of file robbitGUI.cxx.

References SplashWindow, and start_time.

8.5.3.56 void RobbitUI::cb_Close (Fl_Return_Button ∗ *o*, void ∗ *v*) [static, private]

Definition at line 245 of file robbitGUI.cxx.

Referenced by RobbitUI().

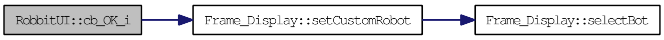
Here is the caller graph for this function:

8.5.3.57 void RobbitUI::cb_OK_i (Fl_Return_Button *, void *) [private]

Definition at line 249 of file robbitGUI.cxx.

References custom_robot_window, display, height, radius, and Frame_Display::setCustomRobot().

Here is the call graph for this function:

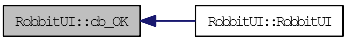


8.5.3.58 void RobbitUI::cb_OK (Fl_Return_Button * *o*, void * *v*) [static, private]

Definition at line 254 of file robbitGUI.cxx.

Referenced by RobbitUI().

Here is the caller graph for this function:

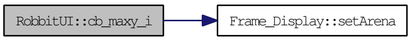


8.5.3.59 void RobbitUI::cb_maxy_i (Fl_Input *, void *) [private]

Definition at line 258 of file robbitGUI.cxx.

References display, maxx, maxy, minx, miny, and Frame_Display::setArena().

Here is the call graph for this function:

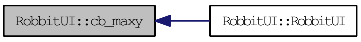


8.5.3.60 void RobbitUI::cb_maxy (Fl_Input * *o*, void * *v*) [static, private]

Definition at line 261 of file robbitGUI.cxx.

Referenced by RobbitUI().

Here is the caller graph for this function:

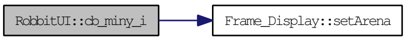


8.5.3.61 void RobbitUI::cb_miny_i (Fl_Input ∗, void ∗) [private]

Definition at line 265 of file robbitGUI.cxx.

References display, maxx, maxy, minx, miny, and Frame_Display::setArena().

Here is the call graph for this function:

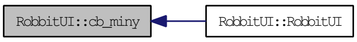


8.5.3.62 void RobbitUI::cb_miny (Fl_Input ∗ *o*, void ∗ *v*) [static, private]

Definition at line 268 of file robbitGUI.cxx.

Referenced by RobbitUI().

Here is the caller graph for this function:

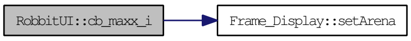


8.5.3.63 void RobbitUI::cb_maxx_i (Fl_Input ∗, void ∗) [private]

Definition at line 272 of file robbitGUI.cxx.

References display, maxx, maxy, minx, miny, and Frame_Display::setArena().

Here is the call graph for this function:

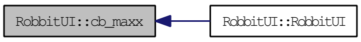


8.5.3.64 void RobbitUI::cb_maxx (Fl_Input ∗ *o*, void ∗ *v*) [static, private]

Definition at line 275 of file robbitGUI.cxx.

Referenced by RobbitUI().

Here is the caller graph for this function:

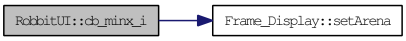


8.5.3.65 void RobbitUI::cb_minx_i (Fl_Input ∗, void ∗) [private]

Definition at line 279 of file robbitGUI.cxx.

References display, maxx, maxy, minx, miny, and Frame_Display::setArena().

Here is the call graph for this function:

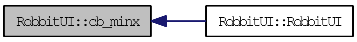


8.5.3.66 void RobbitUI::cb_minx (Fl_Input ∗ *o*, void ∗ *v*) [static, private]

Definition at line 282 of file robbitGUI.cxx.

Referenced by RobbitUI().

Here is the caller graph for this function:

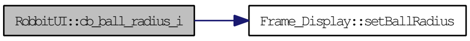


8.5.3.67 void RobbitUI::cb_ball_radius_i (Fl_Input ∗ *o*, void ∗) [private]

Definition at line 286 of file robbitGUI.cxx.

References display, and Frame_Display::setBallRadius().

Here is the call graph for this function:

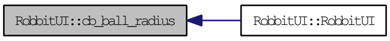


8.5.3.68 void RobbitUI::cb_ball_radius (Fl_Input ∗ *o*, void ∗ *v*) [static, private]

Definition at line 289 of file robbitGUI.cxx.

Referenced by RobbitUI().

Here is the caller graph for this function:

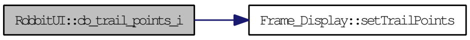


8.5.3.69 void RobbitUI::cb_trail_points_i (Fl_Input ∗ *o*, void ∗) `[private]`

Definition at line 293 of file robbitGUI.cxx.

References display, and Frame_Display::setTrailPoints().

Here is the call graph for this function:

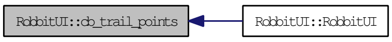


8.5.3.70 void RobbitUI::cb_trail_points (Fl_Input ∗ *o*, void ∗ *v*) `[static, private]`

Definition at line 296 of file robbitGUI.cxx.

Referenced by RobbitUI().

Here is the caller graph for this function:

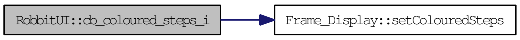


8.5.3.71 void RobbitUI::cb_coloured_steps_i (Fl_Input ∗ *o*, void ∗) `[private]`

Definition at line 300 of file robbitGUI.cxx.

References display, and Frame_Display::setColouredSteps().

Here is the call graph for this function:


RobbitUI::cb_coloured_steps_i
Frame_Display::setColouredSteps

8.5.3.72 void RobbitUI::cb_coloured_steps (Fl_Input ∗ *o*, void ∗ *v*) `[static, private]`

Definition at line 303 of file robbitGUI.cxx.

Referenced by RobbitUI().

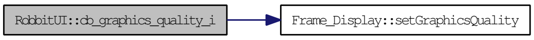
Here is the caller graph for this function:

8.5.3.73 void RobbitUI::cb_graphics_quality_i (Fl_Slider ∗ *o*, void ∗) [private]

Definition at line 307 of file robbitGUI.cxx.

References display, and Frame_Display::setGraphicsQuality().

Here is the call graph for this function:

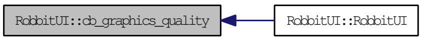

RobbitUI::cb_graphics_quality_i
Frame_Display::setGraphicsQuality

8.5.3.74 void RobbitUI::cb_graphics_quality (Fl_Slider ∗ *o*, void ∗ *v*) [static, private]

Definition at line 310 of file robbitGUI.cxx.

Referenced by RobbitUI().

Here is the caller graph for this function:



8.5.3.75 void RobbitUI::cb_OK1_i (Fl_Return_Button ∗, void ∗) [private]

Definition at line 314 of file robbitGUI.cxx.

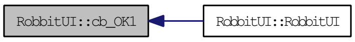
References AdvSettingsWindow.

8.5.3.76 void RobbitUI::cb_OK1 (Fl_Return_Button ∗ *o*, void ∗ *v*) [static, private]

Definition at line 317 of file robbitGUI.cxx.

Referenced by RobbitUI().

Here is the caller graph for this function:



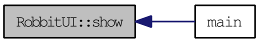
8.5.3.77 void RobbitUI::show ()

Definition at line 575 of file robbitGUI.cxx.

References RobbitMainWindow.

Referenced by main().

Here is the caller graph for this function:



8.5.4 Member Data Documentation

8.5.4.1 Fl_Double_Window∗ RobbitUI::RobbitMainWindow

Definition at line 48 of file robbitGUI.h.

Referenced by RobbitUI(), and show().

8.5.4.2 Fl_Value_Slider∗ RobbitUI::indexSlider

Definition at line 53 of file robbitGUI.h.

Referenced by FileIdleProc(), RobbitUI(), and Frame_Display::setStop().

8.5.4.3 Frame_Display∗ RobbitUI::display

Definition at line 58 of file robbitGUI.h.

Referenced by cb_0_i(), cb_10X_i(), cb_1X_i(), cb_2X_i(), cb_5X_i(), cb_ball_-radius_i(), cb_ButtonAdvSettings_i(), cb_ButtonPlayPause_i(), cb_ButtonReset_i(), cb_ButtonScrrenshot_i(), cb_ButtonStop_i(), cb_coloured_steps_i(), cb_graphics_-quality_i(), cb_indexSlider_i(), cb_KheperaII_i(), cb_KheperaIII_i(), cb_Light1_-i(), cb_Light2_i(), cb_Light3_i(), cb_Light4_i(), cb_MAX_i(), cb_maxx_i(), cb_-maxy_i(), cb_minx_i(), cb_miny_i(), cb_OK_i(), cb_RobbitMainWindow_i(), cb_-ToggleAutoView_i(), cb_ToggleBotNumbering_i(), cb_ToggleObstacles_i(), cb_-ToggleTopView_i(), cb_ToggleTrails_i(), cb_trail_points_i(), FileIdleProc(), FunctionIdleProc(), main(), and RobbitUI().

8.5.4.4 Fl_Button∗ RobbitUI::ButtonAbout

Definition at line 59 of file robbitGUI.h.

Referenced by RobbitUI().

8.5.4.5 Fl_Light_Button∗ RobbitUI::ToggleAutoView

Definition at line 64 of file robbitGUI.h.

Referenced by cb_ToggleTopView_i(), Frame_Display::handle(), and RobbitUI().

8.5.4.6 Fl_Light_Button∗ RobbitUI::ToggleTopView

Definition at line 69 of file robbitGUI.h.

Referenced by cb_ToggleAutoView_i(), Frame_Display::handle(), RobbitUI(), and Frame_Display::setDefaults().

8.5.4.7 Fl_Light_Button∗ RobbitUI::ToggleTrails

Definition at line 74 of file robbitGUI.h.

Referenced by Frame_Display::handle(), RobbitUI(), and Frame_Display::setDefaults().

8.5.4.8 Fl_Light_Button∗ RobbitUI::ToggleObstacles

Definition at line 79 of file robbitGUI.h.

Referenced by Frame_Display::handle(), RobbitUI(), and Frame_Display::setDefaults().

8.5.4.9 Fl_Light_Button∗ RobbitUI::ToggleBotNumbering

Definition at line 84 of file robbitGUI.h.

Referenced by Frame_Display::handle(), RobbitUI(), and Frame_Display::setDefaults().

8.5.4.10 Fl_Button∗ RobbitUI::ButtonAdvSettings

Definition at line 89 of file robbitGUI.h.

Referenced by RobbitUI().

8.5.4.11 Fl_Button∗ RobbitUI::ButtonScrrenshot

Definition at line 94 of file robbitGUI.h.

Referenced by RobbitUI().

8.5.4.12 Fl_Choice∗ RobbitUI::RenderSpeedMenu

Definition at line 99 of file robbitGUI.h.

Referenced by FileIdleProc(), RobbitUI(), and Frame_Display::setDefaults().

8.5.4.13 Fl_Menu_Item RobbitUI::menu_RenderSpeedMenu `[static]`

Initial value:

```
 {
 {"0.5X", 0,  (Fl_Callback*)RobbitUI::cb_0, 0, 0, FL_NORMAL_LABEL, 0, 14, 0},
 {"1X", 0,  (Fl_Callback*)RobbitUI::cb_1X, 0, 0, FL_NORMAL_LABEL, 0, 14, 0},
 {"2X", 0,  (Fl_Callback*)RobbitUI::cb_2X, 0, 0, FL_NORMAL_LABEL, 0, 14, 0},
 {"5X", 0,  (Fl_Callback*)RobbitUI::cb_5X, 0, 0, FL_NORMAL_LABEL, 0, 14, 0},
 {"10X", 0,  (Fl_Callback*)RobbitUI::cb_10X, 0, 0, FL_NORMAL_LABEL, 0, 14, 0},
 {"MAX", 0,  (Fl_Callback*)RobbitUI::cb_MAX, 0, 0, FL_NORMAL_LABEL, 0, 14, 0},
 {0,0,0,0,0,0,0,0,0}
}
```

Definition at line 100 of file robbitGUI.h.

Referenced by RobbitUI().

8.5.4.14 Fl_Choice∗ RobbitUI::BotMenu

Definition at line 115 of file robbitGUI.h.

Referenced by RobbitUI(), and Frame_Display::setDefaults().

8.5.4.15 Fl_Menu_Item RobbitUI::menu_BotMenu [static]

Initial value:

```
 {
 {"Khepera II", 0,  (Fl_Callback*)RobbitUI::cb_KheperaII, 0, 0, FL_NORMAL_LABEL, 0, 14, 0},
 {"Khepera III", 0,  (Fl_Callback*)RobbitUI::cb_KheperaIII, 0, 0, FL_NORMAL_LABEL, 0, 14, 0},
 {"Custom", 0,  (Fl_Callback*)RobbitUI::cb_Custom, 0, 0, FL_NORMAL_LABEL, 0, 14, 0},
 {0,0,0,0,0,0,0,0,0}
}
```

Definition at line 116 of file robbitGUI.h.

Referenced by RobbitUI().

8.5.4.16 Fl_Menu_Item ∗ RobbitUI::KheperaII = RobbitUI::menu_BotMenu + 0 [static]

Definition at line 117 of file robbitGUI.h.

8.5.4.17 Fl_Menu_Item ∗ RobbitUI::KheperaIII = RobbitUI::menu_BotMenu + 1 [static]

Definition at line 122 of file robbitGUI.h.

8.5.4.18 Fl_Menu_Item ∗ RobbitUI::Custom = RobbitUI::menu_BotMenu + 2 [static]

Definition at line 127 of file robbitGUI.h.

8.5.4.19 Fl_Choice∗ RobbitUI::LightMenu

Definition at line 132 of file robbitGUI.h.

Referenced by RobbitUI(), and Frame_Display::setDefaults().

8.5.4.20 Fl_Menu_Item RobbitUI::menu_LightMenu [static]

Initial value:

```
{
{"1", 0,  (Fl_Callback*)RobbitUI::cb_Light1, 0, 0, FL_NORMAL_LABEL, 0, 14, 0},
{"2", 0,  (Fl_Callback*)RobbitUI::cb_Light2, 0, 0, FL_NORMAL_LABEL, 0, 14, 0},
{"3", 0,  (Fl_Callback*)RobbitUI::cb_Light3, 0, 0, FL_NORMAL_LABEL, 0, 14, 0},
{"4", 0,  (Fl_Callback*)RobbitUI::cb_Light4, 0, 0, FL_NORMAL_LABEL, 0, 14, 0},
{0,0,0,0,0,0,0,0,0}
}
```

Definition at line 133 of file robbitGUI.h.

Referenced by RobbitUI().

8.5.4.21 Fl_Menu_Item ∗ RobbitUI::Light1 = RobbitUI::menu_LightMenu + 0 [static]

Definition at line 134 of file robbitGUI.h.

8.5.4.22 Fl_Menu_Item ∗ RobbitUI::Light2 = RobbitUI::menu_LightMenu + 1 [static]

Definition at line 139 of file robbitGUI.h.

8.5.4.23 Fl_Menu_Item ∗ RobbitUI::Light3 = RobbitUI::menu_LightMenu + 2 [static]

Definition at line 144 of file robbitGUI.h.

8.5.4.24 Fl_Menu_Item ∗ RobbitUI::Light4 = RobbitUI::menu_LightMenu + 3 [static]

Definition at line 149 of file robbitGUI.h.

8.5.4.25 Fl_Button∗ RobbitUI::ButtonReset

Definition at line 154 of file robbitGUI.h.

Referenced by RobbitUI().

8.5.4.26 Fl_Button∗ RobbitUI::ButtonPlayPause

Definition at line 159 of file robbitGUI.h.

Referenced by RobbitUI(), Frame_Display::setDefaults(), Frame_Display::setPlayPause(), and Frame_Display::setStop().

8.5.4.27 Fl_Button∗ RobbitUI::ButtonStop

Definition at line 164 of file robbitGUI.h.

Referenced by RobbitUI().

8.5.4.28 Fl_Double_Window∗ RobbitUI::AboutWindow

Definition at line 169 of file robbitGUI.h.

Referenced by cb_About_i(), cb_ButtonAbout_i(), Frame_Display::handle(), and RobbitUI().

8.5.4.29 Fl_Help_View∗ RobbitUI::textrobbit

Definition at line 170 of file robbitGUI.h.

Referenced by cb_About_i(), cb_ButtonAbout_i(), Frame_Display::handle(), and RobbitUI().

8.5.4.30 Fl_Double_Window∗ RobbitUI::SplashWindow

Definition at line 171 of file robbitGUI.h.

Referenced by cb_About_i(), cb_Close_i(), main(), and RobbitUI().

8.5.4.31 Fl_Help_View∗ RobbitUI::htmlSplash

Definition at line 172 of file robbitGUI.h.

Referenced by main(), and RobbitUI().

8.5.4.32 Fl_Double_Window∗ RobbitUI::custom_robot_window

Definition at line 179 of file robbitGUI.h.

Referenced by cb_Custom_i(), cb_OK_i(), and RobbitUI().

8.5.4.33 Fl_Input∗ RobbitUI::height

Definition at line 180 of file robbitGUI.h.

Referenced by cb_OK_i(), and RobbitUI().

8.5.4.34 Fl_Input∗ RobbitUI::radius

Definition at line 181 of file robbitGUI.h.

Referenced by cb_OK_i(), and RobbitUI().

8.5.4.35 Fl_Double_Window∗ RobbitUI::AdvSettingsWindow

Definition at line 186 of file robbitGUI.h.

Referenced by cb_ButtonAdvSettings_i(), cb_OK1_i(), Frame_Display::handle(), and RobbitUI().

8.5.4.36 Fl_Input∗ RobbitUI::maxy

Definition at line 187 of file robbitGUI.h.

Referenced by cb_maxx_i(), cb_maxy_i(), cb_minx_i(), cb_miny_i(), Frame_Display::initAdvWindow(), and RobbitUI().

8.5.4.37 Fl_Input∗ RobbitUI::miny

Definition at line 192 of file robbitGUI.h.

Referenced by cb_maxx_i(), cb_maxy_i(), cb_minx_i(), cb_miny_i(), Frame_Display::initAdvWindow(), and RobbitUI().

8.5.4.38 Fl_Input∗ RobbitUI::maxx

Definition at line 197 of file robbitGUI.h.

Referenced by cb_maxx_i(), cb_maxy_i(), cb_minx_i(), cb_miny_i(), Frame_Display::initAdvWindow(), and RobbitUI().

8.5.4.39 Fl_Input∗ RobbitUI::minx

Definition at line 202 of file robbitGUI.h.

Referenced by cb_maxx_i(), cb_maxy_i(), cb_minx_i(), cb_miny_i(), Frame_Display::initAdvWindow(), and RobbitUI().

8.5.4.40 Fl_Input∗ RobbitUI::ball_radius

Definition at line 207 of file robbitGUI.h.

Referenced by Frame_Display::initAdvWindow(), and RobbitUI().

8.5.4.41 Fl_Input* RobbitUI::trail_points

Definition at line 212 of file robbitGUI.h.

Referenced by Frame_Display::initAdvWindow(), and RobbitUI().

8.5.4.42 Fl_Input* RobbitUI::coloured_steps

Definition at line 217 of file robbitGUI.h.

Referenced by Frame_Display::initAdvWindow(), and RobbitUI().

8.5.4.43 Fl_Slider* RobbitUI::graphics_quality

Definition at line 222 of file robbitGUI.h.

Referenced by Frame_Display::initAdvWindow(), and RobbitUI().

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

- robbitGUI.h
- robbitGUI.cxx

8.6 tagXY Struct Reference

Contains x, y coordinates of a generic point.

```
#include <DistancePointLine.h>
```

Public Attributes

- float X
- float Y

8.6.1 Detailed Description

Contains x, y coordinates of a generic point.

Definition at line 26 of file DistancePointLine.h.

8.6.2 Member Data Documentation

8.6.2.1 float tagXY::X

Definition at line 28 of file DistancePointLine.h.

Referenced by Frame_Display::DetectObstacleCollision(), DistancePointLine(), and Magnitude().

8.6.2.2 float tagXY::Y

Definition at line 28 of file DistancePointLine.h.

Referenced by Frame_Display::DetectObstacleCollision(), DistancePointLine(), and Magnitude().

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

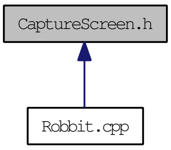
- DistancePointLine.h

Chapter 9

File Documentation

9.1 CaptureScreen.h File Reference

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



Functions

- void CaptureScreen (char ∗filename)

  Actual Capture screen routine.

9.1.1 Function Documentation

9.1.1.1 void CaptureScreen (char ∗ *filename*)

Actual Capture screen routine.

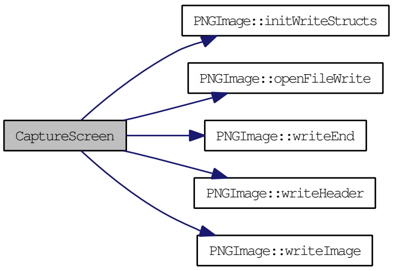
Captures OpenGL render buffer. Creates a PNG file, writes header info, image data, and closes it.

Definition at line 26 of file CaptureScreen.h.

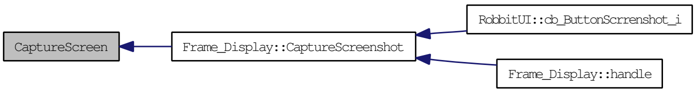
References PNGImage::initWriteStructs(), PNGImage::openFileWrite(), PNGImage::writeEnd(), PNGImage::writeHeader(), and PNGImage::writeImage().

Referenced by Frame_Display::CaptureScreenshot().

Here is the call graph for this function:



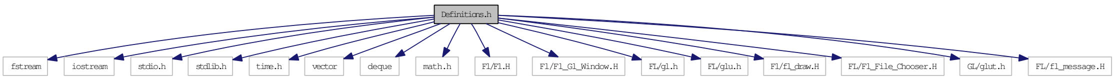
Here is the caller graph for this function:


RobbitUI::cb_ButtonScrrenshot_i
CaptureScreen
Frame_Display::CaptureScreenshot
Frame_Display::handle

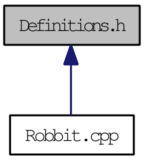
9.2 Definitions.h File Reference

```
#include <fstream>
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <vector>
#include <deque>
#include <math.h>
#include "Fl/Fl.H"
#include "Fl/Fl_Gl_Window.H"
#include "FL/gl.h"
#include "FL/glu.h"
#include "Fl/fl_draw.H"
#include <FL/Fl_File_Chooser.H>
#include <GL/glut.h>
#include <FL/fl_message.H>
```

Include dependency graph for Definitions.h:



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



Defines

- #define no_of_bots 6
- #define radius_of_orient_disk 1.0
- #define height_obstacle 10.0
- #define PI 3.1415926535

Functions

- float next_no (char ∗string, int &st_pos)

    Returns next number reading from a string.

- void outputCharacter (float x, float y, float z, char ∗string)

Variables

- const GLfloat light0_ambient [ ] = {0.7, 0.8, 0.7, 1.0}
- const GLfloat light0_diffuse [ ] = {1.0, 1.0, 1.0, 1.0}
- const GLfloat light0_specular [ ] = {1.0, 1.0, 1.0, 1.0}
- const GLfloat light1_ambient [ ] = {0.7, 0.8, 0.7, 1.0}
- const GLfloat light1_diffuse [ ] = {1.0, 1.0, 1.0, 1.0}
- const GLfloat light1_specular [ ] = {1.0, 1.0, 1.0, 1.0}
- const GLfloat light2_ambient [ ] = {0.7, 0.8, 0.7, 1.0}
- const GLfloat light2_diffuse [ ] = {1.0, 1.0, 1.0, 1.0}
- const GLfloat light2_specular [ ] = {1.0, 1.0, 1.0, 1.0}
- const GLfloat light3_ambient [ ] = {0.7, 0.8, 0.7, 1.0}
- const GLfloat light3_diffuse [ ] = {1.0, 1.0, 1.0, 1.0}
- const GLfloat light3_specular [ ] = {1.0, 1.0, 1.0, 1.0}
- const GLfloat cyl_shininess = {1}
- const GLfloat cyl_ambient [ ] = {1.0, 0.0, 0.0, 1.0}
- const GLfloat cyl_diffuse [ ] = {1.0, 0.0, 0.0, 1.0}
- const GLfloat cyl_k2_specular [4][4] = {{0.5, 0.5, 0.5, 1.0},{1.0, 0.0, 0.0, 1.0},{0.0, 1.0, 0.0, 1.0},{0.0, 1.0, 0.0, 1.0}}
- const GLfloat cyl_k3_specular [4][4] = {{0.9, 0.9, 0.9, 1.0},{1.0, 0.0, 0.0, 1.0},{0.0, 1.0, 0.0, 1.0},{0.0, 1.0, 0.0, 1.0}}
- const GLfloat disk0_shininess = {1}
- const GLfloat disk0_ambient [ ] = {1.0, 0.0, 0.0, 1.0}
- const GLfloat disk0_diffuse [ ] = {1.0, 0.0, 0.0, 1.0}
- const GLfloat disk0_specular [ ] = {0.18, 0.41, 0.18, 1.0}
- const GLfloat disk1_shininess = {50}
- const GLfloat disk1_ambient [ ] = {1.0, 0.0, 0.0, 1.0}
- const GLfloat disk1_diffuse [ ] = {1.0, 0.0, 0.0, 1.0}
- const GLfloat disk1_specular [3][4] = {{0.0, 0.0, 0.0, 1.0},{0.8, 0.0, 0.0, 1.0},{0.0, 0.8, 0.0, 1.0}}
- const GLfloat disk2_shininess = {1}
- const GLfloat disk2_ambient [ ] = {1.0, 0.0, 0.0, 1.0}
- const GLfloat disk2_diffuse [ ] = {1.0, 0.0, 0.0, 1.0}
- const GLfloat disk2_specular [ ] = {1.0, 1.0, 0.0, 1.0}
- const GLfloat disk_center_shininess = {1}
- const GLfloat disk_center_ambient [ ] = {1.0, 0.0, 0.0, 1.0}
- const GLfloat disk_center_diffuse [ ] = {1.0, 0.0, 0.0, 1.0}
- const GLfloat disk_center_specular [ ] = {0.0, 0.0, 0.0, 1.0}
- const GLfloat text_shininess = {1}

- const GLfloat text_ambient [ ] = {1.0, 0.0, 0.0, 1.0}
- const GLfloat text_diffuse [ ] = {1.0, 0.0, 0.0, 1.0}
- const GLfloat text_specular [ ] = {1.0, 0.0, 0.0, 1.0}
- const GLfloat ball_shininess = {1}
- const GLfloat ball_ambient [ ] = {1.0, 0.0, 0.0, 1.0}
- const GLfloat ball_diffuse [ ] = {1.0, 0.0, 0.0, 1.0}
- const GLfloat ball_specular [ ] = {1.0, 1.0, 0.0, 1.0}
- const GLfloat plane_shininess = {1}
- const GLfloat plane_specular [ ] = {0.3, 0.3, 0.9, 1.0}
- const GLfloat floor_shininess = {1}
- const GLfloat floor_specular [ ] = {0.8, 0.8, 0.8, 1.0}
- const GLfloat disk_2D_obstacle_shininess = {1}
- const GLfloat disk_2D_obstacle_specular [ ] = {1.0, 0.0, 0.0, 1.0}
- GLfloat light0_pos [4]
- GLfloat light1_pos [4]
- GLfloat light2_pos [4]
- GLfloat light3_pos [4]
- const double radianFactor = 2 ∗ PI / 360
- void ∗ font = GLUT_BITMAP_8_BY_13
- float max_x = -20000
- float max_y = -20000
- float min_x = 20000
- float min_y = 20000
- int default_sleep_time = 1000
- static GLUquadric ∗ quad
- int info_refresh_count = 100
- int update_frame_mode = -1
- clock_t start_time

9.2.1 Define Documentation

9.2.1.1 #define height_obstacle 10.0

Definition at line 136 of file Definitions.h.

9.2.1.2 #define no_of_bots 6

Definition at line 134 of file Definitions.h.

Referenced by Frame_Display::BotHit(), File_Data::BuildFileIndex(), Frame_Display::draw(), Frame_Display::DrawTrails(), File_Data::GetData(), GetNextFrame(), Frame_Display::handle(), Frame_Display::setDefaults(), and Frame_Display::setTrails().

9.2.1.3 #define PI 3.1415926535

Definition at line 137 of file Definitions.h.

9.2.1.4 #define radius_of_orient_disk 1.0

Definition at line 135 of file Definitions.h.

Referenced by Frame_Display::position_khepera2(), Frame_Display::position_khepera3(), and Frame_Display::PositionCustomRobot().

9.2.2 Function Documentation

9.2.2.1 float next_no (char ∗ *string*, int & *st_pos*)

Returns next number reading from a string.

Numbers should be seperated by space or comma. It is used to read the numbers form the Obstacle ASCII file. It also updates the string pointer position

Parameters:

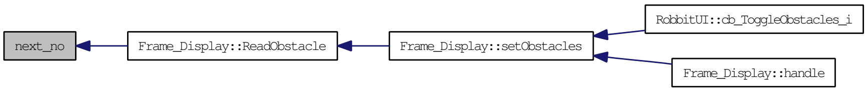
∗***string*** Pointer to the string

st_pos position of current string pointer (passed by reference)

Definition at line 41 of file NextNo.h.

Referenced by Frame_Display::ReadObstacle().

Here is the caller graph for this function:


next_no
Frame_Display::ReadObstacle
Frame_Display::setObstacles
RobbitUI::cb_ToggleObstacles_i
Frame_Display::handle

9.2.2.2 void outputCharacter (float *x*, float *y*, float *z*, char ∗ *string*)

9.2.3 Variable Documentation

9.2.3.1 const GLfloat ball_ambient[ ] = {1.0, 0.0, 0.0, 1.0}

Definition at line 111 of file Definitions.h.

9.2.3.2 const GLfloat ball_diffuse[ ] = {1.0, 0.0, 0.0, 1.0}

Definition at line 112 of file Definitions.h.

9.2.3.3 const GLfloat ball_shininess = {1}

Definition at line 110 of file Definitions.h.

Referenced by Frame_Display::draw().

9.2.3.4 const GLfloat ball_specular[ ] = {1.0, 1.0, 0.0, 1.0}

Definition at line 113 of file Definitions.h.

Referenced by Frame_Display::draw().

9.2.3.5 const GLfloat cyl_ambient[ ] = {1.0, 0.0, 0.0, 1.0}

Definition at line 74 of file Definitions.h.

9.2.3.6 const GLfloat cyl_diffuse[ ] = {1.0, 0.0, 0.0, 1.0}

Definition at line 75 of file Definitions.h.

9.2.3.7 const GLfloat cyl_k2_specular[4][4] = {{0.5, 0.5, 0.5, 1.0},{1.0, 0.0, 0.0, 1.0},{0.0, 1.0, 0.0, 1.0},{0.0, 1.0, 0.0, 1.0}}

Definition at line 76 of file Definitions.h.

Referenced by Frame_Display::position_khepera2(), and Frame_Display::PositionCustomRobot().

9.2.3.8 const GLfloat cyl_k3_specular[4][4] = {{0.9, 0.9, 0.9, 1.0},{1.0, 0.0, 0.0, 1.0},{0.0, 1.0, 0.0, 1.0},{0.0, 1.0, 0.0, 1.0}}

Definition at line 77 of file Definitions.h.

Referenced by Frame_Display::position_khepera3().

9.2.3.9 const GLfloat cyl_shininess = {1}

Definition at line 73 of file Definitions.h.

Referenced by Frame_Display::init().

9.2.3.10 int default_sleep_time = 1000

Definition at line 150 of file Definitions.h.

Referenced by FileIdleProc().

9.2.3.11 const GLfloat disk0_ambient[ ] = {1.0, 0.0, 0.0, 1.0}

Definition at line 81 of file Definitions.h.

9.2.3.12 const GLfloat disk0_diffuse[ ] = {1.0, 0.0, 0.0, 1.0}

Definition at line 82 of file Definitions.h.

9.2.3.13 const GLfloat disk0_shininess = {1}

Definition at line 80 of file Definitions.h.

9.2.3.14 const GLfloat disk0_specular[ ] = {0.18, 0.41, 0.18, 1.0}

Definition at line 83 of file Definitions.h.

Referenced by Frame_Display::position_khepera2().

9.2.3.15 const GLfloat disk1_ambient[ ] = {1.0, 0.0, 0.0, 1.0}

Definition at line 87 of file Definitions.h.

9.2.3.16 const GLfloat disk1_diffuse[ ] = {1.0, 0.0, 0.0, 1.0}

Definition at line 88 of file Definitions.h.

9.2.3.17 const GLfloat disk1_shininess = {50}

Definition at line 86 of file Definitions.h.

9.2.3.18 const GLfloat disk1_specular[3][4] = {{0.0, 0.0, 0.0, 1.0},{0.8, 0.0, 0.0, 1.0},{0.0, 0.8, 0.0, 1.0}}

Definition at line 89 of file Definitions.h.

Referenced by Frame_Display::position_khepera2(), Frame_Display::position_khepera3(), and Frame_Display::PositionCustomRobot().

9.2.3.19 const GLfloat disk2_ambient[ ] = {1.0, 0.0, 0.0, 1.0}

Definition at line 93 of file Definitions.h.

9.2.3.20 const GLfloat disk2_diffuse[ ] = {1.0, 0.0, 0.0, 1.0}

Definition at line 94 of file Definitions.h.

9.2.3.21 const GLfloat disk2_shininess = {1}

Definition at line 92 of file Definitions.h.

9.2.3.22 const GLfloat disk2_specular[ ] = {1.0, 1.0, 0.0, 1.0}

Definition at line 95 of file Definitions.h.

Referenced by Frame_Display::position_khepera2(), Frame_Display::position_khepera3(), and Frame_Display::PositionCustomRobot().

9.2.3.23 const GLfloat disk_2D_obstacle_shininess = {1}

Definition at line 123 of file Definitions.h.

Referenced by Frame_Display::DrawObstacle().

9.2.3.24 const GLfloat disk_2D_obstacle_specular[ ] = {1.0, 0.0, 0.0, 1.0}

Definition at line 124 of file Definitions.h.

Referenced by Frame_Display::DrawObstacle().

9.2.3.25 const GLfloat disk_center_ambient[ ] = {1.0, 0.0, 0.0, 1.0}

Definition at line 99 of file Definitions.h.

9.2.3.26 const GLfloat disk_center_diffuse[ ] = {1.0, 0.0, 0.0, 1.0}

Definition at line 100 of file Definitions.h.

9.2.3.27 const GLfloat disk_center_shininess = {1}

Definition at line 98 of file Definitions.h.

9.2.3.28 const GLfloat disk_center_specular[ ] = {0.0, 0.0, 0.0, 1.0}

Definition at line 101 of file Definitions.h.

Referenced by Frame_Display::draw(), and Frame_Display::DrawTrails().

9.2.3.29 const GLfloat floor_shininess = {1}

Definition at line 119 of file Definitions.h.

9.2.3.30 const GLfloat floor_specular[ ] = {0.8, 0.8, 0.8, 1.0}

Definition at line 120 of file Definitions.h.

Referenced by Frame_Display::DrawFloor().

9.2.3.31 void∗ font = GLUT_BITMAP_8_BY_13

Definition at line 140 of file Definitions.h.

Referenced by Frame_Display::outputCharacter().

9.2.3.32 int info_refresh_count = 100

Definition at line 153 of file Definitions.h.

Referenced by Frame_Display::BotHit(), and Frame_Display::DetectObstacleCollision().

9.2.3.33 const GLfloat light0_ambient[ ] = {0.7, 0.8, 0.7, 1.0}

Definition at line 53 of file Definitions.h.

Referenced by Frame_Display::init().

9.2.3.34 const GLfloat light0_diffuse[ ] = {1.0, 1.0, 1.0, 1.0}

Definition at line 54 of file Definitions.h.

Referenced by Frame_Display::init().

9.2.3.35 GLfloat light0_pos[4]

Definition at line 126 of file Definitions.h.

Referenced by FileIdleProc(), FunctionIdleProc(), and Frame_Display::init().

9.2.3.36 const GLfloat light0_specular[ ] = {1.0, 1.0, 1.0, 1.0}

Definition at line 55 of file Definitions.h.

Referenced by Frame_Display::init().

9.2.3.37 const GLfloat light1_ambient[ ] = {0.7, 0.8, 0.7, 1.0}

Definition at line 58 of file Definitions.h.

9.2.3.38 const GLfloat light1_diffuse[ ] = {1.0, 1.0, 1.0, 1.0}

Definition at line 59 of file Definitions.h.

9.2.3.39 GLfloat light1_pos[4]

Definition at line 127 of file Definitions.h.

Referenced by FileIdleProc(), FunctionIdleProc(), and Frame_Display::init().

9.2.3.40 const GLfloat light1_specular[ ] = {1.0, 1.0, 1.0, 1.0}

Definition at line 60 of file Definitions.h.

Referenced by Frame_Display::init().

9.2.3.41 const GLfloat light2_ambient[ ] = {0.7, 0.8, 0.7, 1.0}

Definition at line 63 of file Definitions.h.

9.2.3.42 const GLfloat light2_diffuse[ ] = {1.0, 1.0, 1.0, 1.0}

Definition at line 64 of file Definitions.h.

9.2.3.43 GLfloat light2_pos[4]

Definition at line 128 of file Definitions.h.

Referenced by FileIdleProc(), FunctionIdleProc(), and Frame_Display::init().

9.2.3.44 const GLfloat light2_specular[ ] = {1.0, 1.0, 1.0, 1.0}

Definition at line 65 of file Definitions.h.

Referenced by Frame_Display::init().

9.2.3.45 const GLfloat light3_ambient[ ] = {0.7, 0.8, 0.7, 1.0}

Definition at line 68 of file Definitions.h.

9.2.3.46 const GLfloat light3_diffuse[ ] = {1.0, 1.0, 1.0, 1.0}

Definition at line 69 of file Definitions.h.

9.2.3.47 GLfloat light3_pos[4]

Definition at line 129 of file Definitions.h.

Referenced by FileIdleProc(), FunctionIdleProc(), and Frame_Display::init().

9.2.3.48 const GLfloat light3_specular[ ] = {1.0, 1.0, 1.0, 1.0}

Definition at line 70 of file Definitions.h.

Referenced by Frame_Display::init().

9.2.3.49 float max_x = -20000

Definition at line 145 of file Definitions.h.

Referenced by File_Data::BuildFileIndex(), Frame_Display::DrawFloor(), FileIdleProc(), FunctionIdleProc(), Frame_Display::handle(), Frame_Display::initAdvWindow(), Frame_Display::setArena(), and Frame_Display::setDefaults().

9.2.3.50 float max_y = -20000

Definition at line 146 of file Definitions.h.

Referenced by File_Data::BuildFileIndex(), Frame_Display::DrawFloor(), FileIdleProc(), FunctionIdleProc(), Frame_Display::handle(), Frame_Display::initAdvWindow(), Frame_Display::setArena(), and Frame_Display::setDefaults().

9.2.3.51 float min_x = 20000

Definition at line 147 of file Definitions.h.

Referenced by File_Data::BuildFileIndex(), Frame_Display::DrawFloor(), FileIdleProc(), FunctionIdleProc(), Frame_Display::handle(), Frame_Display::initAdvWindow(), Frame_Display::setArena(), and Frame_Display::setDefaults().

9.2.3.52 float min_y = 20000

Definition at line 148 of file Definitions.h.

Referenced by File_Data::BuildFileIndex(), Frame_Display::DrawFloor(), FileIdleProc(), FunctionIdleProc(), Frame_Display::handle(), Frame_Display::initAdvWindow(), Frame_Display::setArena(), and Frame_Display::setDefaults().

9.2.3.53 const GLfloat plane_shininess = {1}

Definition at line 116 of file Definitions.h.

9.2.3.54 const GLfloat plane_specular[ ] = {0.3, 0.3, 0.9, 1.0}

Definition at line 117 of file Definitions.h.

Referenced by Frame_Display::DrawFloor().

9.2.3.55 GLUquadric∗ quad [static]

Definition at line 152 of file Definitions.h.

Referenced by Frame_Display::draw(), Frame_Display::DrawObstacle(), Frame_Display::init(), Frame_Display::position_khepera2(), Frame_Display::position_khepera3(), and Frame_Display::PositionCustomRobot().

9.2.3.56 const double radianFactor = 2 ∗ PI / 360

Definition at line 139 of file Definitions.h.

Referenced by Frame_Display::handle(), and Frame_Display::reshape().

9.2.3.57 clock_t start_time

Definition at line 159 of file Definitions.h.

Referenced by RobbitUI::cb_About_i(), RobbitUI::cb_Close_i(), and main().

9.2.3.58 const GLfloat text_ambient[ ] = {1.0, 0.0, 0.0, 1.0}

Definition at line 105 of file Definitions.h.

9.2.3.59 const GLfloat text_diffuse[ ] = {1.0, 0.0, 0.0, 1.0}

Definition at line 106 of file Definitions.h.

9.2.3.60 const GLfloat text_shininess = {1}

Definition at line 104 of file Definitions.h.

9.2.3.61 const GLfloat text_specular[ ] = {1.0, 0.0, 0.0, 1.0}

Definition at line 107 of file Definitions.h.

Referenced by Frame_Display::draw().

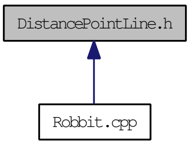
9.2.3.62 int update_frame_mode = -1

Definition at line 155 of file Definitions.h.

Referenced by main().

9.3 DistancePointLine.h File Reference

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



Classes

- struct tagXY

  Contains x, y coordinates of a generic point.

Typedefs

- typedef struct tagXY XY

  Contains x, y coordinates of a generic point.

Functions

- float Magnitude (XY ∗Point1, XY ∗Point2)

  Distance between two points.

- int DistancePointLine (XY ∗Point, XY ∗LineStart, XY ∗LineEnd, float radius_of_robot)

  Evaluates if a bot lies on a line given by two points.

9.3.1 Typedef Documentation

9.3.1.1 typedef struct tagXY XY

Contains x, y coordinates of a generic point.

9.3.2 Function Documentation

9.3.2.1 int DistancePointLine (XY ∗ *Point*, XY ∗ *LineStart*, XY ∗ *LineEnd*, float *radius_of_robot*)

Evaluates if a bot lies on a line given by two points.

Parameters:

- ***Point*** XY point, center of robot
- ***LineStart*** XY point, marks a point on the line
- ***LineEnd*** XY point, marks a second point on the line

Returns:

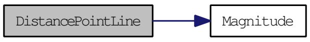
0, if bot is not on the line; 1, if it is.

Definition at line 53 of file DistancePointLine.h.

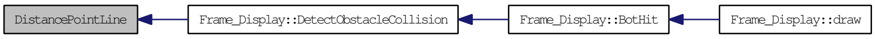
References Magnitude(), tagXY::X, and tagXY::Y.

Referenced by Frame_Display::DetectObstacleCollision().

Here is the call graph for this function:



Here is the caller graph for this function:



9.3.2.2 float Magnitude (XY * *Point1*, XY * *Point2*)

Distance between two points.

Parameters:

- ***Point1*** XY point
- ***Point2*** XY point

Returns:

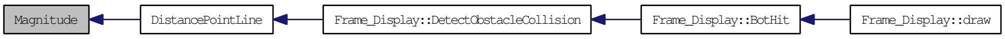
Distance between two points as float

Definition at line 38 of file DistancePointLine.h.

References tagXY::X, and tagXY::Y.

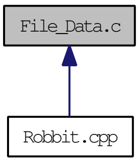
Referenced by DistancePointLine().

Here is the caller graph for this function:



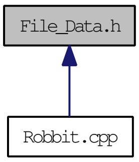
9.4 File_Data.c File Reference

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



9.5 File_Data.h File Reference

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



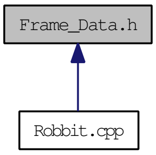
Classes

- class File_Data

  Class to store file data.

9.6 Frame_Data.h File Reference

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



Classes

- class Frame_Data

  Class to store frame data.

Typedefs

- typedef std::vector< float > FloatVec
- typedef std::vector< int > IntVec

9.6.1 Typedef Documentation

9.6.1.1 typedef std::vector<float> FloatVec

Definition at line 23 of file Frame_Data.h.

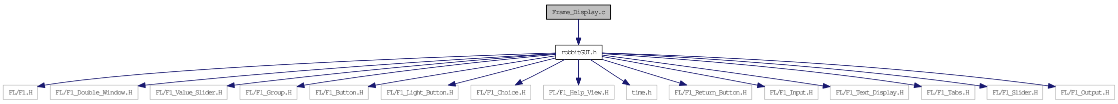
9.6.1.2 typedef std::vector<int> IntVec

Definition at line 24 of file Frame_Data.h.

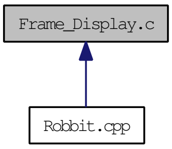
9.7 Frame_Display.c File Reference

```
#include "robbitGUI.h"
```

Include dependency graph for Frame_Display.c:



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



Variables

- RobbitUI robbit_gui

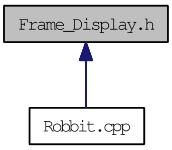
9.7.1 Variable Documentation

9.7.1.1 RobbitUI robbit_gui

Definition at line 45 of file Robbit.cpp.

9.8 Frame_Display.h File Reference

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



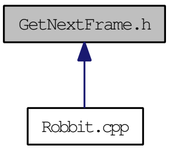
Classes

- class Frame_Display

    Class to store data required for running of simulation and GUI.

9.9 GetNextFrame.h File Reference

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



Functions

- Frame_Data GetNextFrame ()

  User defined function to generate next frame data.

9.9.1 Function Documentation

9.9.1.1 Frame_Data GetNextFrame ()

User defined function to generate next frame data.

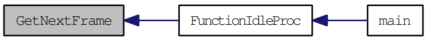
A user may define his algorithm to generate the data in Frame_data here.

Definition at line 26 of file GetNextFrame.h.

References Frame_Data::ball_x, Frame_Data::ball_y, Frame_Data::bot_hit, Frame_Data::bot_orient, Frame_Data::bot_vorient, Frame_Data::bot_vx, Frame_Data::bot_vy, Frame_Data::bot_x, Frame_Data::bot_y, no_of_bots, Frame_Data::time, and Frame_Data::time_step.

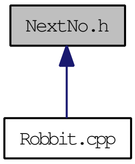
Referenced by FunctionIdleProc().

Here is the caller graph for this function:



9.10 NextNo.h File Reference

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



Functions

- float next_no (char ∗string, int &st_pos)

  Returns next number reading from a string.

9.10.1 Function Documentation

9.10.1.1 float next_no (char ∗ *string*, int & *st_pos*)

Returns next number reading from a string.

Numbers should be seperated by space or comma. It is used to read the numbers form the Obstacle ASCII file. It also updates the string pointer position

Parameters:

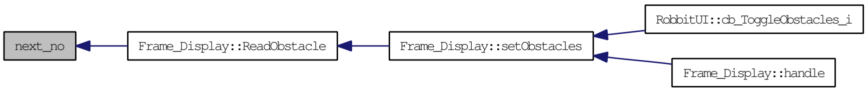
∗string Pointer to the string

st_pos position of current string pointer (passed by reference)

Definition at line 41 of file NextNo.h.

Referenced by Frame_Display::ReadObstacle().

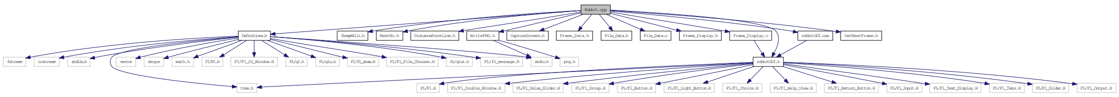
Here is the caller graph for this function:


next_no
Frame_Display::ReadObstacle
Frame_Display::setObstacles
RobbitUI::cb_ToggleObstacles_i
Frame_Display::handle

9.11 Robbit.cpp File Reference

```
#include "Definitions.h"
#include "SleepMilli.h"
#include "NextNo.h"
#include "DistancePointLine.h"
#include "WritePNG.h"
#include "CaptureScreen.h"
#include "Frame_Data.h"
#include "File_Data.h"
#include "File_Data.c"
#include "Frame_Display.h"
#include "Frame_Display.c"
#include "robbitGUI.h"
#include "robbitGUI.cxx"
#include "GetNextFrame.h"
```

Include dependency graph for Robbit.cpp:



Functions

- void FileIdleProc (void ∗)

  Idle function operating in log file mode.

- void FunctionIdleProc (void ∗)

  Idle function operating in custom algorithm mode.

- void WriteInputFile ()

  Writes log file in specified format.

- int main (int argc, char ∗∗argv)

  Main function.

Variables

- RobbitUI robbit_gui

9.11.1 Function Documentation

9.11.1.1 void FileIdleProc (void ∗)

Idle function operating in log file mode.

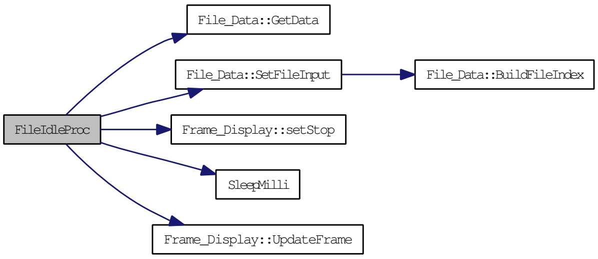
When the simlation is run by reading data from pre-existing log file, this function handles the data handling and updating. It also sets the positions of the various light sources, numbered 0 through 3. It sets the floor size by reading through the data file once, and getting the maximum and minimum coordinates of the objects present in the arena. Sets the initial look-at coordinate. Updates the sleep time between frame renders. Also controls play/pause actions.

Definition at line 115 of file Robbit.cpp.

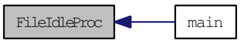
References Frame_Display::current_index, default_sleep_time, RobbitUI::display, File_Data::GetData(), RobbitUI::indexSlider, Frame_Display::is_paused, light0_pos, light1_pos, light2_pos, light3_pos, max_x, max_y, min_x, min_y, Frame_Display::render_speed, RobbitUI::RenderSpeedMenu, File_Data::SetFileInput(), Frame_Display::setStop(), SleepMilli(), Frame_Data::time_step, Frame_Display::UpdateFrame(), Frame_Display::view_centerX, Frame_Display::view_centerY, and Frame_Display::view_centerZ.

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:



9.11.1.2 void FunctionIdleProc (void ∗)

Idle function operating in custom algorithm mode.

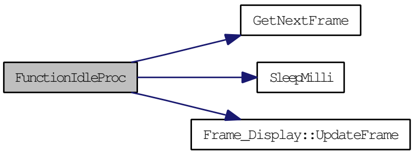
When the simlation is run by creating coordinate data using a custom algorithm, this function is used. Algorithm writers may use the above function, void FileIdleProc (void ∗) as reference while writing this function.

Definition at line 169 of file Robbit.cpp.

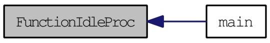
References RobbitUI::display, GetNextFrame(), Frame_Display::is_paused, light0_pos, light1_pos, light2_pos, light3_pos, max_x, max_y, min_x, min_y, SleepMilli(), Frame_Display::UpdateFrame(), Frame_Display::view_centerX, Frame_Display::view_centerY, and Frame_Display::view_centerZ.

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:



9.11.1.3 int main (int *argc*, char ∗∗ *argv*)

Main function.

Displays splash screen, asks mode of operation (log file, custom algorithm, etc., starts simulation)

Parameters:

argc number of commandline arguments

argv array of commandline arguments

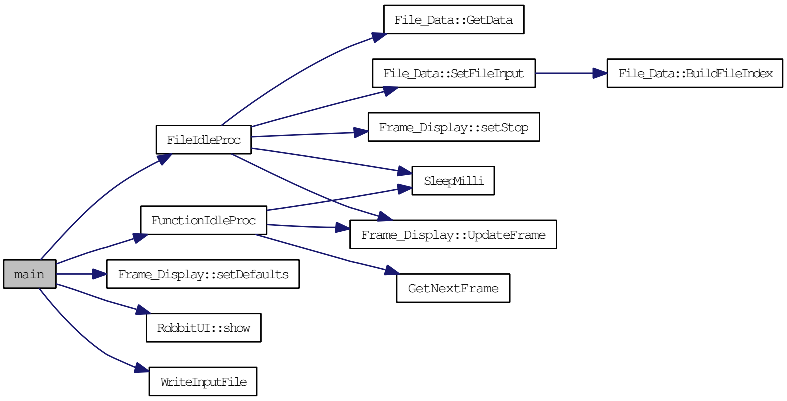
Returns:

0, except in case of error

Definition at line 59 of file Robbit.cpp.

References RobbitUI::display, FileIdleProc(), FunctionIdleProc(), RobbitUI::htmlSplash, Frame_Display::setDefaults(), RobbitUI::show(), RobbitUI::SplashWindow, start_time, update_frame_mode, and WriteInputFile().

Here is the call graph for this function:



9.11.1.4 void WriteInputFile ()

Writes log file in specified format.

Writes frame data including x & y coordinates, x & y velocities, orientation (θ), direction of velocity and other details for each object in the arena in a given format.

Any line starting with ” is treated as comment and is skipped while reading the log.

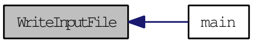
Each line contains the data for all the robots at a particular time. Following pseduo code expains the format.

```
fprintf(LogFile,"%4d %3.1f  ",iteration_number,tfly); // tfly: time between successive iterations
fprintf(LogFile,"%7.3f",current_time);

for i=0 to i<no_of_robots {
  fprintf(LogFile,"%7.2f %7.2f %6.3f %5.1f %5.1f ",x_pos_Robot[i],y_pos_Robot[i],orientation_Robot[i],Li
}

fori=0 to i<NoofRobots {
  fprintf(LogFile,"%5.1f %5.1f ",x_velocity_Robots[i],y_velocity_Robots[i]);
}

fprintf(LogFile,"%5.1f %5.1f ",x_pos_ball,y_pos_ball);
fprintf(LogFile,"\n"); // end of current line
```

Definition at line 225 of file Robbit.cpp.

Referenced by main().

Here is the caller graph for this function:



9.11.2 Variable Documentation

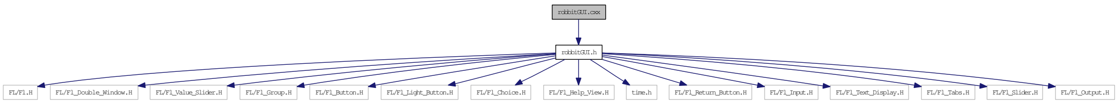
9.11.2.1 RobbitUI robbit_gui

Definition at line 45 of file Robbit.cpp.

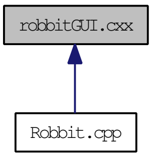
9.12 robbitGUI.cxx File Reference

```
#include "robbitGUI.h"
```

Include dependency graph for robbitGUI.cxx:



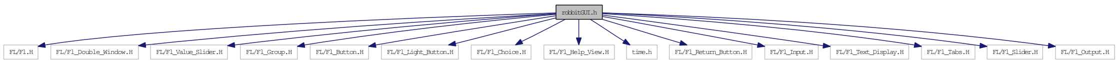
This graph shows which files directly or indirectly include this file:



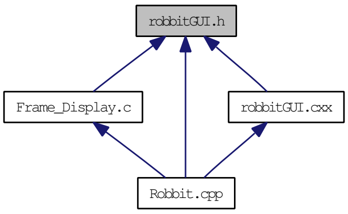
9.13 robbitGUI.h File Reference

```
#include <FL/Fl.H>
#include <FL/Fl_Double_Window.H>
#include <FL/Fl_Value_Slider.H>
#include <FL/Fl_Group.H>
#include <FL/Fl_Button.H>
#include <FL/Fl_Light_Button.H>
#include <FL/Fl_Choice.H>
#include <FL/Fl_Help_View.H>
#include <time.h>
#include <FL/Fl_Return_Button.H>
#include <FL/Fl_Input.H>
#include <FL/Fl_Text_Display.H>
#include <FL/Fl_Tabs.H>
#include <FL/Fl_Slider.H>
#include <FL/Fl_Output.H>
```

Include dependency graph for robbitGUI.h:



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



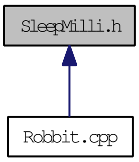
Classes

- class RobbitUI

  CLass to build the whole GUI.

9.14 SleepMilli.h File Reference

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



Functions

- void SleepMilli (long ms)

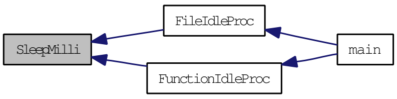
9.14.1 Function Documentation

9.14.1.1 void SleepMilli (long *ms*)

Definition at line 31 of file SleepMilli.h.

Referenced by FileIdleProc(), and FunctionIdleProc().

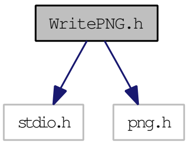
Here is the caller graph for this function:



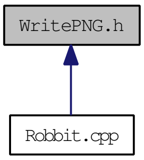
9.15 WritePNG.h File Reference

```
#include <stdio.h>
#include <png.h>
```

Include dependency graph for WritePNG.h:



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



Classes

- class PNGImage

  Class to store information of a PNG image.

Index