

robbit  
4.0

Generated by Doxygen 1.5.6

Fri Sep 5 17:46:45 2008



# Contents

<b>1</b>	<b>Robbit</b>	<b>1</b>
1.1	What is Robbit ? . . . . .	1
1.2	Contributors . . . . .	1
<b>2</b>	<b>Installation</b>	<b>3</b>
2.1	Installation . . . . .	4
2.1.1	Linux . . . . .	4
2.1.2	Windows 32-/64-Bit . . . . .	4
<b>3</b>	<b>Dependencies</b>	<b>5</b>
3.1	Dependencies . . . . .	6
<b>4</b>	<b>Usage</b>	<b>7</b>
4.1	Keyboard mappings . . . . .	8
4.2	Mouse mappings . . . . .	8
4.3	Advanced settings . . . . .	8
4.3.1	Obstacles . . . . .	8
4.4	Custom algorithms . . . . .	8
<b>5</b>	<b>Features</b>	<b>9</b>
<b>6</b>	<b>Class Index</b>	<b>11</b>
6.1	Class List . . . . .	11
<b>7</b>	<b>File Index</b>	<b>13</b>
7.1	File List . . . . .	13
<b>8</b>	<b>Class Documentation</b>	<b>15</b>

8.1	File_Data Class Reference	15
8.1.1	Detailed Description	16
8.1.2	Member Typedef Documentation	16
8.1.2.1	pos_type	16
8.1.3	Member Function Documentation	16
8.1.3.1	BuildFileIndex	16
8.1.3.2	SetFileInput	16
8.1.3.3	IndexSize	17
8.1.3.4	GetData	17
8.1.4	Member Data Documentation	18
8.1.4.1	file_index	18
8.1.4.2	input	18
8.2	Frame_Data Class Reference	19
8.2.1	Detailed Description	19
8.2.2	Member Data Documentation	19
8.2.2.1	bot_x	19
8.2.2.2	bot_y	20
8.2.2.3	bot_vx	20
8.2.2.4	bot_vy	20
8.2.2.5	bot_orient	20
8.2.2.6	bot_vorient	20
8.2.2.7	bot_hit	20
8.2.2.8	bot_design	20
8.2.2.9	time	21
8.2.2.10	ball_x	21
8.2.2.11	ball_y	21
8.2.2.12	ball_vx	21
8.2.2.13	ball_vy	21
8.2.2.14	bot_center_x	21
8.2.2.15	bot_center_y	21
8.2.2.16	time_step	21
8.3	Frame_Display Class Reference	23
8.3.1	Detailed Description	27
8.3.2	Constructor & Destructor Documentation	27

8.3.2.1	Frame_Display . . . . .	27
8.3.3	Member Function Documentation . . . . .	27
8.3.3.1	handle . . . . .	27
8.3.3.2	UpdateFrame . . . . .	28
8.3.3.3	draw . . . . .	29
8.3.3.4	DrawFloor . . . . .	29
8.3.3.5	ReadObstacle . . . . .	30
8.3.3.6	DrawObstacle . . . . .	30
8.3.3.7	DrawTrails . . . . .	31
8.3.3.8	BotHit . . . . .	31
8.3.3.9	CaptureScreenshot . . . . .	32
8.3.3.10	DetectObstacleCollision . . . . .	32
8.3.3.11	position_khepera2 . . . . .	33
8.3.3.12	position_khepera3 . . . . .	33
8.3.3.13	PositionCustomRobot . . . . .	34
8.3.3.14	AutoUpdateEye . . . . .	34
8.3.3.15	init . . . . .	35
8.3.3.16	reshape . . . . .	35
8.3.3.17	setIndex . . . . .	36
8.3.3.18	outputCharacter . . . . .	36
8.3.3.19	setPlayPause . . . . .	36
8.3.3.20	setExit . . . . .	37
8.3.3.21	setStop . . . . .	37
8.3.3.22	setReset . . . . .	37
8.3.3.23	setBotNumbering . . . . .	38
8.3.3.24	setTrails . . . . .	38
8.3.3.25	setObstacles . . . . .	39
8.3.3.26	setTopView . . . . .	39
8.3.3.27	setAutoView . . . . .	40
8.3.3.28	selectBot . . . . .	40
8.3.3.29	setLights . . . . .	41
8.3.3.30	setRenderSpeed . . . . .	41
8.3.3.31	setDefaults . . . . .	42
8.3.3.32	setCustomRobot . . . . .	42

8.3.3.33	setArena	43
8.3.3.34	setBallRadius	44
8.3.3.35	setTrailPoints	44
8.3.3.36	setColouredSteps	44
8.3.3.37	initAdvWindow	45
8.3.3.38	setGraphicsQuality	45
8.3.4	Member Data Documentation	46
8.3.4.1	bot_selected	46
8.3.4.2	number_bots	46
8.3.4.3	show_obstacle	46
8.3.4.4	show_trails	46
8.3.4.5	topview	46
8.3.4.6	autoview	46
8.3.4.7	obs_2D	46
8.3.4.8	obs_WL	46
8.3.4.9	obs_CB	47
8.3.4.10	obs_SP	47
8.3.4.11	obs_2D_counter	47
8.3.4.12	WL_counter	47
8.3.4.13	CB_counter	47
8.3.4.14	SP_counter	47
8.3.4.15	radius_ball	47
8.3.4.16	no_of_trail_data	47
8.3.4.17	coloured_steps	48
8.3.4.18	bot_trail	48
8.3.4.19	ball_trail	48
8.3.4.20	radius_camera_movement	48
8.3.4.21	theta	48
8.3.4.22	phi	48
8.3.4.23	eyeX	48
8.3.4.24	eyeY	49
8.3.4.25	eyeZ	49
8.3.4.26	upX	49
8.3.4.27	upY	49

8.3.4.28	upZ . . . . .	49
8.3.4.29	aspect . . . . .	49
8.3.4.30	Near . . . . .	49
8.3.4.31	Far . . . . .	49
8.3.4.32	camera_on_bot . . . . .	50
8.3.4.33	delta_zoom . . . . .	50
8.3.4.34	no_of_lights . . . . .	50
8.3.4.35	mouseX0 . . . . .	50
8.3.4.36	mouseY0 . . . . .	50
8.3.4.37	mousePushValid . . . . .	50
8.3.4.38	radius_of_robot . . . . .	50
8.3.4.39	height_robot . . . . .	50
8.3.4.40	bot_slices . . . . .	51
8.3.4.41	bot_stacks . . . . .	51
8.3.4.42	small_disk_slices . . . . .	51
8.3.4.43	small_disk_stacks . . . . .	51
8.3.4.44	data . . . . .	51
8.3.4.45	LineStart . . . . .	51
8.3.4.46	LineEnd . . . . .	51
8.3.4.47	Point . . . . .	52
8.3.4.48	is_paused . . . . .	52
8.3.4.49	render_speed . . . . .	52
8.3.4.50	current_index . . . . .	52
8.3.4.51	view_centerX . . . . .	52
8.3.4.52	view_centerY . . . . .	52
8.3.4.53	view_centerZ . . . . .	52
8.4	PNGImage Class Reference . . . . .	54
8.4.1	Detailed Description . . . . .	55
8.4.2	Constructor & Destructor Documentation . . . . .	55
8.4.2.1	PNGImage . . . . .	55
8.4.2.2	PNGImage . . . . .	55
8.4.2.3	~PNGImage . . . . .	55
8.4.3	Member Function Documentation . . . . .	55
8.4.3.1	openFileRead . . . . .	55

8.4.3.2	<a href="#">openFileWrite</a>	56
8.4.3.3	<a href="#">isPNG</a>	56
8.4.3.4	<a href="#">initReadStructs</a>	56
8.4.3.5	<a href="#">initWriteStructs</a>	57
8.4.3.6	<a href="#">writeHeader</a>	57
8.4.3.7	<a href="#">writeImage</a>	57
8.4.3.8	<a href="#">writeEnd</a>	57
8.4.4	<a href="#">Member Data Documentation</a>	58
8.4.4.1	<a href="#">filename</a>	58
8.4.4.2	<a href="#">fp</a>	58
8.4.4.3	<a href="#">png_ptr</a>	58
8.4.4.4	<a href="#">info_ptr</a>	58
8.4.4.5	<a href="#">width</a>	58
8.4.4.6	<a href="#">height</a>	58
8.4.4.7	<a href="#">row</a>	59
8.4.4.8	<a href="#">bytes_per_pixel</a>	59
8.4.4.9	<a href="#">i</a>	59
8.4.4.10	<a href="#">j</a>	59
8.4.4.11	<a href="#">rgb</a>	59
8.4.4.12	<a href="#">width_para</a>	59
8.5	<a href="#">RobbitUI Class Reference</a>	60
8.5.1	<a href="#">Detailed Description</a>	63
8.5.2	<a href="#">Constructor &amp; Destructor Documentation</a>	63
8.5.2.1	<a href="#">RobbitUI</a>	63
8.5.3	<a href="#">Member Function Documentation</a>	65
8.5.3.1	<a href="#">cb_RobbitMainWindow_i</a>	65
8.5.3.2	<a href="#">cb_RobbitMainWindow</a>	65
8.5.3.3	<a href="#">cb_indexSlider_i</a>	65
8.5.3.4	<a href="#">cb_indexSlider</a>	65
8.5.3.5	<a href="#">cb_ButtonAbout_i</a>	66
8.5.3.6	<a href="#">cb_ButtonAbout</a>	66
8.5.3.7	<a href="#">cb_ToggleAutoView_i</a>	66
8.5.3.8	<a href="#">cb_ToggleAutoView</a>	66
8.5.3.9	<a href="#">cb_ToggleTopView_i</a>	67



8.5.3.10	<a href="#">cb_ToggleTopView</a>	67
8.5.3.11	<a href="#">cb_ToggleTrails_i</a>	67
8.5.3.12	<a href="#">cb_ToggleTrails</a>	67
8.5.3.13	<a href="#">cb_ToggleObstacles_i</a>	68
8.5.3.14	<a href="#">cb_ToggleObstacles</a>	68
8.5.3.15	<a href="#">cb_ToggleBotNumbering_i</a>	68
8.5.3.16	<a href="#">cb_ToggleBotNumbering</a>	68
8.5.3.17	<a href="#">cb_ButtonAdvSettings_i</a>	69
8.5.3.18	<a href="#">cb_ButtonAdvSettings</a>	69
8.5.3.19	<a href="#">cb_ButtonScreenshot_i</a>	69
8.5.3.20	<a href="#">cb_ButtonScreenshot</a>	69
8.5.3.21	<a href="#">cb_0_i</a>	70
8.5.3.22	<a href="#">cb_0</a>	70
8.5.3.23	<a href="#">cb_1X_i</a>	70
8.5.3.24	<a href="#">cb_1X</a>	70
8.5.3.25	<a href="#">cb_2X_i</a>	70
8.5.3.26	<a href="#">cb_2X</a>	71
8.5.3.27	<a href="#">cb_5X_i</a>	71
8.5.3.28	<a href="#">cb_5X</a>	71
8.5.3.29	<a href="#">cb_10X_i</a>	71
8.5.3.30	<a href="#">cb_10X</a>	71
8.5.3.31	<a href="#">cb_MAX_i</a>	72
8.5.3.32	<a href="#">cb_MAX</a>	72
8.5.3.33	<a href="#">cb_KheperaII_i</a>	72
8.5.3.34	<a href="#">cb_KheperaII</a>	72
8.5.3.35	<a href="#">cb_KheperaIII_i</a>	72
8.5.3.36	<a href="#">cb_KheperaIII</a>	73
8.5.3.37	<a href="#">cb_Custom_i</a>	73
8.5.3.38	<a href="#">cb_Custom</a>	73
8.5.3.39	<a href="#">cb_Light1_i</a>	73
8.5.3.40	<a href="#">cb_Light1</a>	73
8.5.3.41	<a href="#">cb_Light2_i</a>	73
8.5.3.42	<a href="#">cb_Light2</a>	74
8.5.3.43	<a href="#">cb_Light3_i</a>	74

8.5.3.44	<a href="#">cb_Light3</a>	74
8.5.3.45	<a href="#">cb_Light4_i</a>	74
8.5.3.46	<a href="#">cb_Light4</a>	74
8.5.3.47	<a href="#">cb_ButtonReset_i</a>	74
8.5.3.48	<a href="#">cb_ButtonReset</a>	75
8.5.3.49	<a href="#">cb_ButtonPlayPause_i</a>	75
8.5.3.50	<a href="#">cb_ButtonPlayPause</a>	75
8.5.3.51	<a href="#">cb_ButtonStop_i</a>	75
8.5.3.52	<a href="#">cb_ButtonStop</a>	76
8.5.3.53	<a href="#">cb_About_i</a>	76
8.5.3.54	<a href="#">cb_About</a>	76
8.5.3.55	<a href="#">cb_Close_i</a>	76
8.5.3.56	<a href="#">cb_Close</a>	76
8.5.3.57	<a href="#">cb_OK_i</a>	77
8.5.3.58	<a href="#">cb_OK</a>	77
8.5.3.59	<a href="#">cb_maxy_i</a>	77
8.5.3.60	<a href="#">cb_maxy</a>	77
8.5.3.61	<a href="#">cb_miny_i</a>	78
8.5.3.62	<a href="#">cb_miny</a>	78
8.5.3.63	<a href="#">cb_maxx_i</a>	78
8.5.3.64	<a href="#">cb_maxx</a>	78
8.5.3.65	<a href="#">cb_minx_i</a>	79
8.5.3.66	<a href="#">cb_minx</a>	79
8.5.3.67	<a href="#">cb_ball_radius_i</a>	79
8.5.3.68	<a href="#">cb_ball_radius</a>	79
8.5.3.69	<a href="#">cb_trail_points_i</a>	80
8.5.3.70	<a href="#">cb_trail_points</a>	80
8.5.3.71	<a href="#">cb_coloured_steps_i</a>	80
8.5.3.72	<a href="#">cb_coloured_steps</a>	80
8.5.3.73	<a href="#">cb_graphics_quality_i</a>	81
8.5.3.74	<a href="#">cb_graphics_quality</a>	81
8.5.3.75	<a href="#">cb_OK1_i</a>	81
8.5.3.76	<a href="#">cb_OK1</a>	81
8.5.3.77	<a href="#">show</a>	81

8.5.4	Member Data Documentation	82
8.5.4.1	RobbitMainWindow	82
8.5.4.2	indexSlider	82
8.5.4.3	display	82
8.5.4.4	ButtonAbout	82
8.5.4.5	ToggleAutoView	82
8.5.4.6	ToggleTopView	83
8.5.4.7	ToggleTrails	83
8.5.4.8	ToggleObstacles	83
8.5.4.9	ToggleBotNumbering	83
8.5.4.10	ButtonAdvSettings	83
8.5.4.11	ButtonScreenshot	83
8.5.4.12	RenderSpeedMenu	83
8.5.4.13	menu_RenderSpeedMenu	83
8.5.4.14	BotMenu	84
8.5.4.15	menu_BotMenu	84
8.5.4.16	KheperaII	84
8.5.4.17	KheperaIII	84
8.5.4.18	Custom	84
8.5.4.19	LightMenu	85
8.5.4.20	menu_LightMenu	85
8.5.4.21	Light1	85
8.5.4.22	Light2	85
8.5.4.23	Light3	85
8.5.4.24	Light4	85
8.5.4.25	ButtonReset	85
8.5.4.26	ButtonPlayPause	86
8.5.4.27	ButtonStop	86
8.5.4.28	AboutWindow	86
8.5.4.29	textrobbit	86
8.5.4.30	SplashWindow	86
8.5.4.31	htmlSplash	86
8.5.4.32	custom_robot_window	86
8.5.4.33	height	86

8.5.4.34	radius	87
8.5.4.35	AdvSettingsWindow	87
8.5.4.36	maxy	87
8.5.4.37	miny	87
8.5.4.38	maxx	87
8.5.4.39	minx	87
8.5.4.40	ball_radius	87
8.5.4.41	trail_points	88
8.5.4.42	coloured_steps	88
8.5.4.43	graphics_quality	88
8.6	tagXY Struct Reference	89
8.6.1	Detailed Description	89
8.6.2	Member Data Documentation	89
8.6.2.1	X	89
8.6.2.2	Y	89
<b>9</b>	<b>File Documentation</b>	<b>91</b>
9.1	CaptureScreen.h File Reference	91
9.1.1	Function Documentation	91
9.1.1.1	CaptureScreen	91
9.2	Definitions.h File Reference	93
9.2.1	Define Documentation	95
9.2.1.1	height_obstacle	95
9.2.1.2	no_of_bots	95
9.2.1.3	PI	95
9.2.1.4	radius_of_orient_disk	96
9.2.2	Function Documentation	96
9.2.2.1	next_no	96
9.2.2.2	outputCharacter	96
9.2.3	Variable Documentation	96
9.2.3.1	ball_ambient	96
9.2.3.2	ball_diffuse	96
9.2.3.3	ball_shininess	96
9.2.3.4	ball_specular	97

9.2.3.5	cyl_ambient . . . . .	97
9.2.3.6	cyl_diffuse . . . . .	97
9.2.3.7	cyl_k2_specular . . . . .	97
9.2.3.8	cyl_k3_specular . . . . .	97
9.2.3.9	cyl_shininess . . . . .	97
9.2.3.10	default_sleep_time . . . . .	97
9.2.3.11	disk0_ambient . . . . .	97
9.2.3.12	disk0_diffuse . . . . .	98
9.2.3.13	disk0_shininess . . . . .	98
9.2.3.14	disk0_specular . . . . .	98
9.2.3.15	disk1_ambient . . . . .	98
9.2.3.16	disk1_diffuse . . . . .	98
9.2.3.17	disk1_shininess . . . . .	98
9.2.3.18	disk1_specular . . . . .	98
9.2.3.19	disk2_ambient . . . . .	98
9.2.3.20	disk2_diffuse . . . . .	98
9.2.3.21	disk2_shininess . . . . .	98
9.2.3.22	disk2_specular . . . . .	99
9.2.3.23	disk_2D_obstacle_shininess . . . . .	99
9.2.3.24	disk_2D_obstacle_specular . . . . .	99
9.2.3.25	disk_center_ambient . . . . .	99
9.2.3.26	disk_center_diffuse . . . . .	99
9.2.3.27	disk_center_shininess . . . . .	99
9.2.3.28	disk_center_specular . . . . .	99
9.2.3.29	floor_shininess . . . . .	99
9.2.3.30	floor_specular . . . . .	99
9.2.3.31	font . . . . .	100
9.2.3.32	info_refresh_count . . . . .	100
9.2.3.33	light0_ambient . . . . .	100
9.2.3.34	light0_diffuse . . . . .	100
9.2.3.35	light0_pos . . . . .	100
9.2.3.36	light0_specular . . . . .	100
9.2.3.37	light1_ambient . . . . .	100
9.2.3.38	light1_diffuse . . . . .	100

9.2.3.39	light1_pos	101
9.2.3.40	light1_specular	101
9.2.3.41	light2_ambient	101
9.2.3.42	light2_diffuse	101
9.2.3.43	light2_pos	101
9.2.3.44	light2_specular	101
9.2.3.45	light3_ambient	101
9.2.3.46	light3_diffuse	101
9.2.3.47	light3_pos	101
9.2.3.48	light3_specular	102
9.2.3.49	max_x	102
9.2.3.50	max_y	102
9.2.3.51	min_x	102
9.2.3.52	min_y	102
9.2.3.53	plane_shininess	102
9.2.3.54	plane_specular	103
9.2.3.55	quad	103
9.2.3.56	radianFactor	103
9.2.3.57	start_time	103
9.2.3.58	text_ambient	103
9.2.3.59	text_diffuse	103
9.2.3.60	text_shininess	103
9.2.3.61	text_specular	103
9.2.3.62	update_frame_mode	104
9.3	DistancePointLine.h File Reference	105
9.3.1	Typedef Documentation	105
9.3.1.1	XY	105
9.3.2	Function Documentation	105
9.3.2.1	DistancePointLine	105
9.3.2.2	Magnitude	106
9.4	File_Data.c File Reference	107
9.5	File_Data.h File Reference	108
9.6	Frame_Data.h File Reference	109
9.6.1	Typedef Documentation	109

9.6.1.1	FloatVec	109
9.6.1.2	IntVec	109
9.7	Frame_Display.c File Reference	110
9.7.1	Variable Documentation	110
9.7.1.1	robbit_gui	110
9.8	Frame_Display.h File Reference	111
9.9	GetNextFrame.h File Reference	112
9.9.1	Function Documentation	112
9.9.1.1	GetNextFrame	112
9.10	NextNo.h File Reference	113
9.10.1	Function Documentation	113
9.10.1.1	next_no	113
9.11	Robbit.cpp File Reference	114
9.11.1	Function Documentation	115
9.11.1.1	FileIdleProc	115
9.11.1.2	FunctionIdleProc	115
9.11.1.3	main	116
9.11.1.4	WriteInputFile	117
9.11.2	Variable Documentation	118
9.11.2.1	robbit_gui	118
9.12	robbitGUL.cxx File Reference	119
9.13	robbitGUI.h File Reference	120
9.14	SleepMilli.h File Reference	121
9.14.1	Function Documentation	121
9.14.1.1	SleepMilli	121
9.15	WritePNG.h File Reference	122





# 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 compile 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 Screenshot
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 panoramic 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:

<a href="#">File_Data</a> (Class to store file data ) . . . . .	15
<a href="#">Frame_Data</a> (Class to store frame data ) . . . . .	19
<a href="#">Frame_Display</a> (Class to store data required for running of simulation and GUI ) . . . . .	23
<a href="#">PNGImage</a> (Class to store information of a PNG image ) . . . . .	54
<a href="#">RobbitUI</a> (CLass to build the whole GUI ) . . . . .	60
<a href="#">tagXY</a> (Contains x, y coordinates of a generic point ) . . . . .	89



## Chapter 7

# File Index

### 7.1 File List

Here is a list of all files with brief descriptions:

<a href="#">CaptureScreen.h</a>	91
<a href="#">Definitions.h</a>	93
<a href="#">DistancePointLine.h</a>	105
<a href="#">File_Data.c</a>	107
<a href="#">File_Data.h</a>	108
<a href="#">Frame_Data.h</a>	109
<a href="#">Frame_Display.c</a>	110
<a href="#">Frame_Display.h</a>	111
<a href="#">GetNextFrame.h</a>	112
<a href="#">NextNo.h</a>	113
<a href="#">Robbit.cpp</a>	114
<a href="#">robbitGUI.cxx</a>	119
<a href="#">robbitGUI.h</a>	120
<a href="#">SleepMilli.h</a>	121
<a href="#">WritePNG.h</a>	122



## 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.

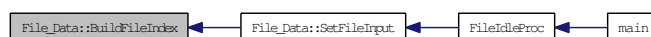
Stores the positions of get pointers to valid records in the log file, in a vector. This way, the file does not need to be traversed in serial fashion, and allows for quicker and random access to file data.

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:

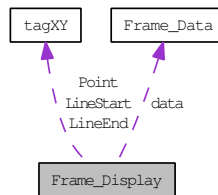
- [Frame\\_Data.h](#)

## 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.

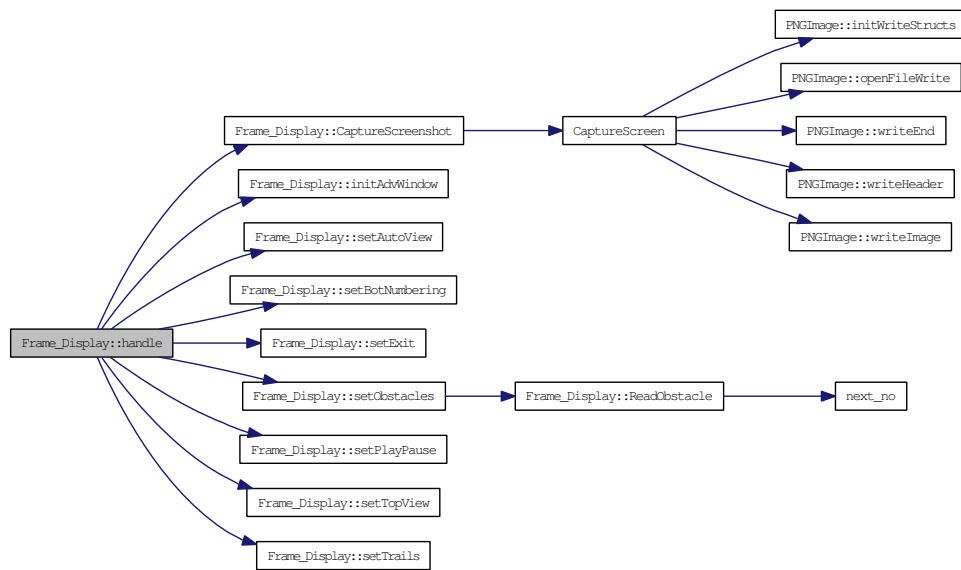
##### Parameters:

- 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:



### 8.3.3.2 void Frame\_Display::UpdateFrame (Frame\_Data \_data)

Updates the current frame data.

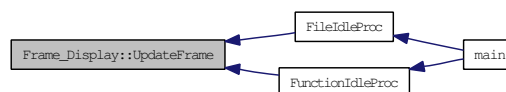
Updates the object data to be rendered to mirror those most recently read from the file/custom algorithm.

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 ()

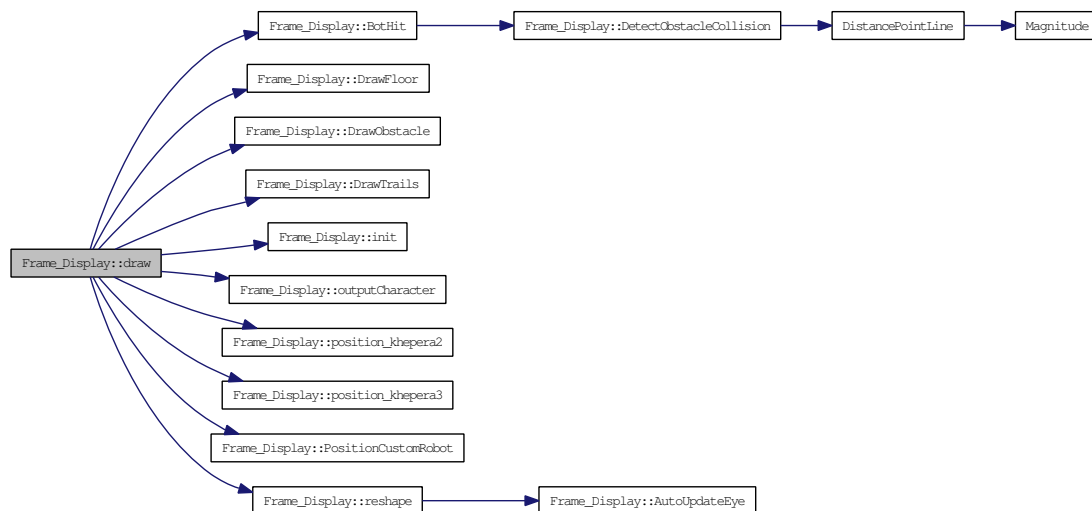
Renders the current frame.

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:



### 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:



### 8.3.3.6 void Frame\_Display::DrawObstacle ()

Renders obstacle.

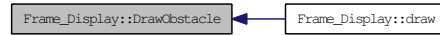
Takes details stored by `ReadObstacle` and renders some supported types of obstacles. Note that the pole is represented purely as a point on the floor. It has no rendering of its expected structure to prevent obstruction in viewing.

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.

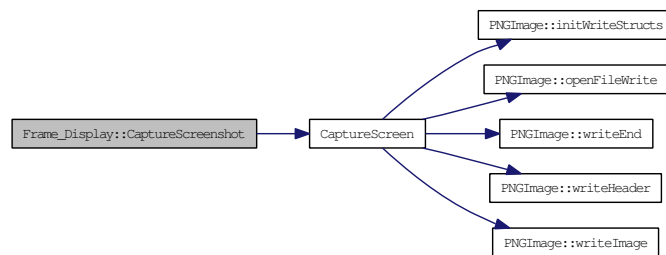
Saves a screenshot with a generated filename in Portable Network Graphics format. The filename contains details like the time, and view angles,  $\theta$  and  $\phi$ .

Definition at line 1097 of file Frame\_Display.c.

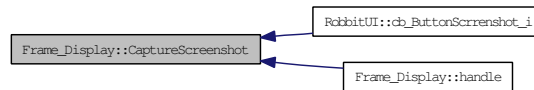
References CaptureScreen(), data, phi, theta, and Frame\_Data::time.

Referenced by RobbitUI::cb\_ButtonScreenshot\_i(), and handle().

Here is the call graph for this function:



Here is the caller graph for this function:



### 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)

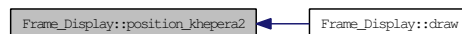
*color* non-default color when collision occurs

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.

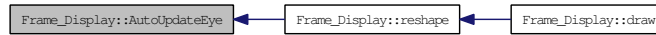
Decrements the angle made by the line connecting the camera location and the look-at position, with the X axis, projected on the floor.

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:



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

Renders a character at a specific location.

#### Parameters:

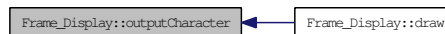
- x* x coordinate of string to be rendered
- y* y coordinate of string to be rendered
- z* z coordinate of string to be rendered
- string* string to be rendered

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 ()`

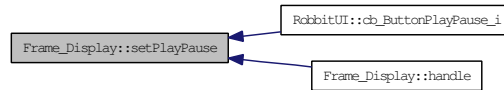
Swaps Play and Pause symbols in GUI Play/Pause button, on click.

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:



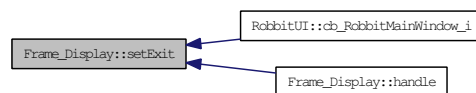
### 8.3.3.20 void Frame\_Display::setExit () [inline]

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.

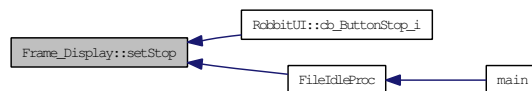
Clears file record index, repositions slider to 0, pauses the animation.

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:

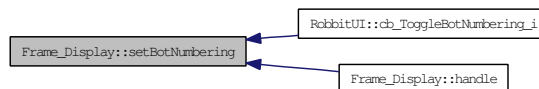
*val* value of the numbering bot (GUI)

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:



### 8.3.3.24 void Frame\_Display::setTrails (bool val)

Clears trail data.

If `show_trails` is set, it clears the trail data.

#### Parameters:

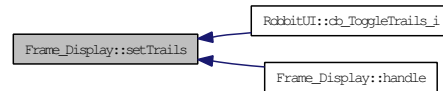
*val* value of the Show trails option (GUI)

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:

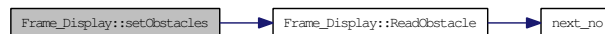
*val* value of the show obstacle option (GUI)

Definition at line 139 of file Frame\_Display.h.

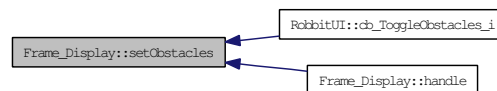
References ReadObstacle(), and show\_obstacle.

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:

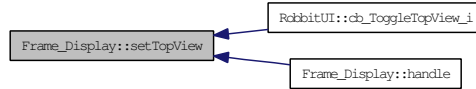
*val* value of the top view mode (GUI)

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:

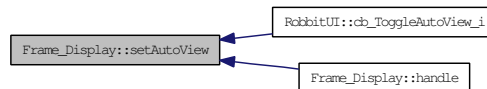
*val* value of the auto-rotate mode (GUI)

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:

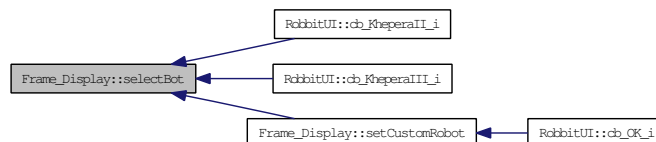
*val* type of bot: 1 = KheperaII; 2 = KheperaIII; 3 = Custom Robot

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:**

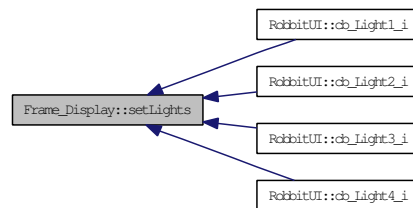
*val* no of lights (GUI)

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:**

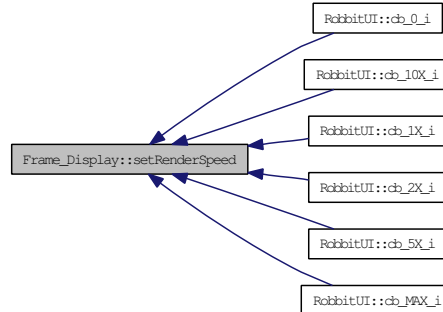
*val* value rendering speed in form of nX (GUI); 0: denotes maximum possible speed

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.

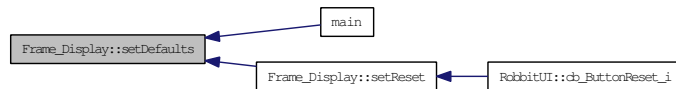
Developers may change them to their requirements.

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:



### 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)

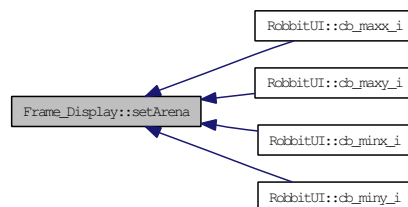
***miny*** minimum y coordinate (GUI)

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:



### 8.3.3.37 void Frame\_Display::initAdvWindow ()

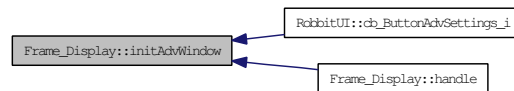
Sets configuration of Advanced Settings menu elements.

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:



### 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:



## 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:



### 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.



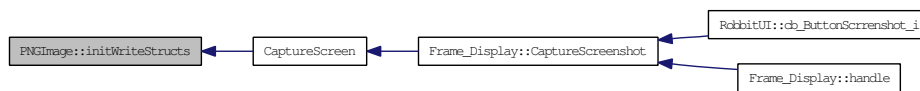
**8.4.3.5 bool PNGImage::initWriteStructs ()**

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:

**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:

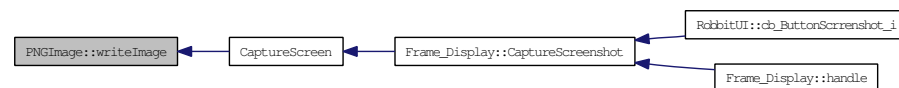
**8.4.3.7 bool PNGImage::writeImage (void \* bits)**

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:

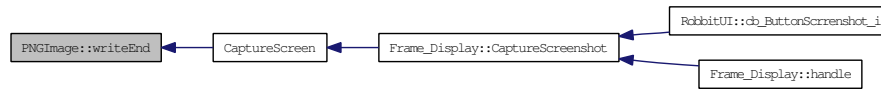
**8.4.3.8 bool PNGImage::writeEnd ()**

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:



## 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_info 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:

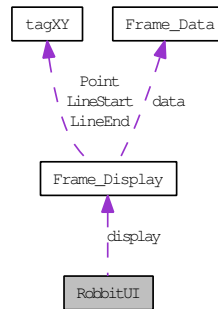
- [WritePNG.h](#)

## 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 *` [ButtonScreenshot](#)
- `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\\_ButtonScreenshot\\_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\\_ButtonScreenshot](#) (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`.

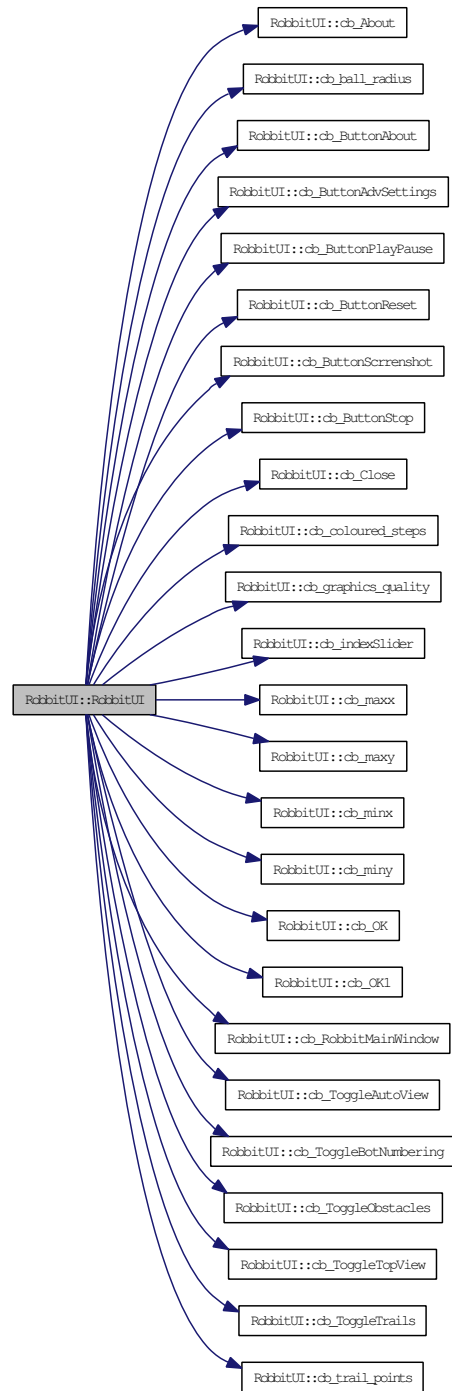
### 8.5.2 Constructor & Destructor Documentation

#### 8.5.2.1 RobbitUI::RobbitUI ()

Definition at line 321 of file `robbitGUI.cxx`.

References `AboutWindow`, `AdvSettingsWindow`, `ball_radius`, `BotMenu`, `ButtonAbout`, `ButtonAdvSettings`, `ButtonPlayPause`, `ButtonReset`, `ButtonScreenshot`, `ButtonStop`, `cb_About()`, `cb_ball_radius()`, `cb_ButtonAbout()`, `cb_ButtonAdvSettings()`, `cb_ButtonPlayPause()`, `cb_ButtonReset()`, `cb_ButtonScreenshot()`, `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:





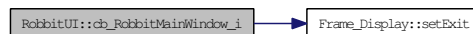
### 8.5.3 Member Function Documentation

#### 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:



#### 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:



#### 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:

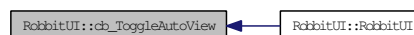


#### 8.5.3.8 void RobbitUI::cb\_ToggleAutoView (Fl\_Light\_Button \* o, void \* v) [static, private]

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:

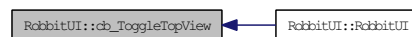


### 8.5.3.10 void RobbitUI::cb\_ToggleTopView (Fl\_Light\_Button \* o, void \* v) [static, private]

Definition at line 45 of file robbitGUI.cxx.

Referenced by RobbitUI().

Here is the caller graph for this function:



### 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:

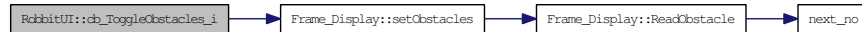


### 8.5.3.13 void RobbitUI::cb\_ToggleObstacles\_i (Fl\_Light\_Button \* o, void \*) [private]

Definition at line 56 of file robbitGUI.cxx.

References display, and Frame\_Display::setObstacles().

Here is the call graph for this function:

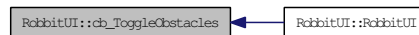


### 8.5.3.14 void RobbitUI::cb\_ToggleObstacles (Fl\_Light\_Button \* o, void \* v) [static, private]

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:

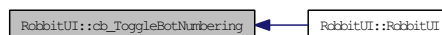


### 8.5.3.16 void RobbitUI::cb\_ToggleBotNumbering (Fl\_Light\_Button \* o, void \* v) [static, private]

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:



### 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:

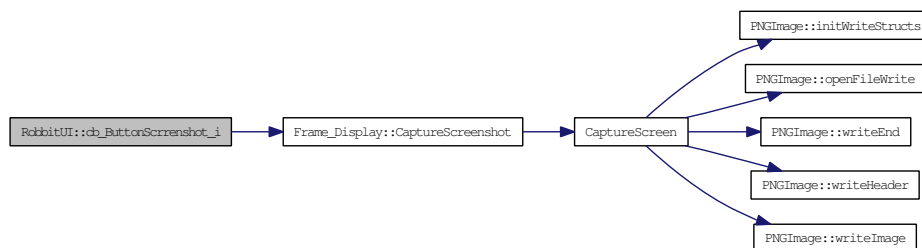


### 8.5.3.19 void RobbitUI::cb\_ButtonScreenshot\_i (Fl\_Button \*, void \*) [private]

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\_ButtonScreenshot (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:

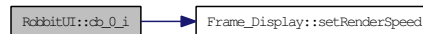


#### 8.5.3.21 `void RobbitUI::cb_0_i (Fl_Menu_ *, void *)` [private]

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().

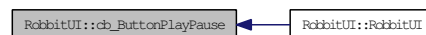
Here is the call graph for this function:

**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:

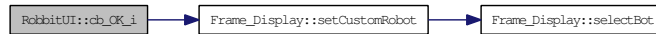


**8.5.3.57 void RobbitUI::cb\_OK\_i (Fl\_Return\_Button \*, void \*)** [private]

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:

**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().

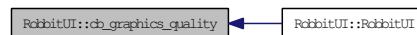
Here is the call graph for this function:

**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:

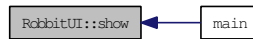
**8.5.3.77 void RobbitUI::show ()**

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_ButtonScreenshot_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::ButtonScreenshot`

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](#)

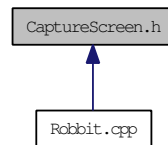


## Chapter 9

# 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.

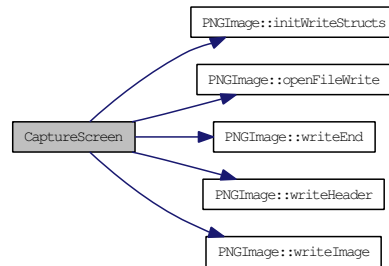
Captures OpenGL render buffer. Creates a PNG file, writes header info, image data, and closes it.

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:



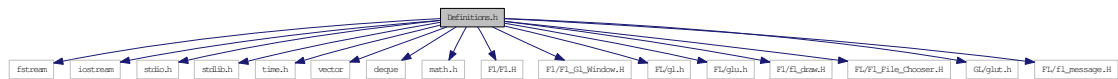
Here is the caller graph for this function:



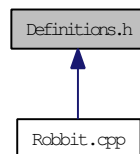
## 9.2 Definitions.h File Reference

```
#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::setDefault()`, 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 separated by space or comma. It is used to read the numbers from the Obstacle ASCII file. It also updates the string pointer position

#### Parameters:

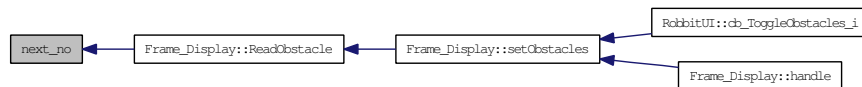
\**string* Pointer to the string

*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:



### 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**

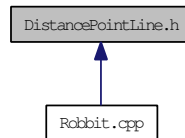
Definition at line 155 of file Definitions.h.

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:**

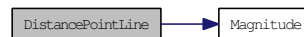
0, if bot is not on the line; 1, if it is.

Definition at line 53 of file DistancePointLine.h.

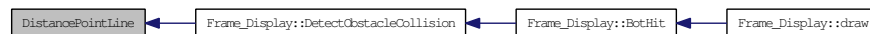
References `Magnitude()`, `tagXY::X`, and `tagXY::Y`.

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:**

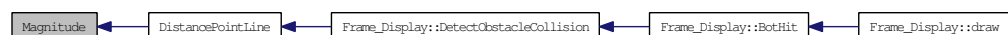
Distance between two points as float

Definition at line 38 of file DistancePointLine.h.

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

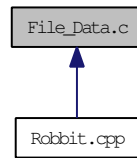
Referenced by `DistancePointLine()`.

Here is the caller graph for this function:



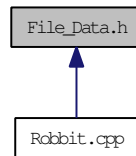
## 9.4 File\_Data.c File Reference

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:

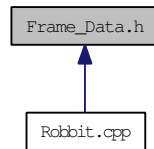


### Classes

- 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.

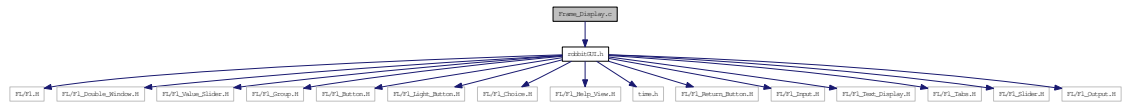
##### 9.6.1.2 typedef std::vector<int> IntVec

Definition at line 24 of file Frame\_Data.h.

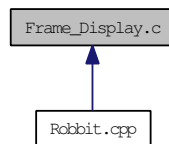
## 9.7 Frame\_Display.c File Reference

```
#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](#)

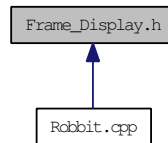
#### 9.7.1 Variable Documentation

##### 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:



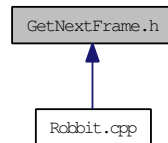
### Classes

- 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.

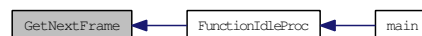
A user may define his algorithm to generate the data in Frame\_data here.

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()`.

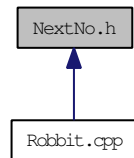
Here is the caller graph for this function:





## 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 separated by space or comma. It is used to read the numbers from the Obstacle ASCII file. It also updates the string pointer position

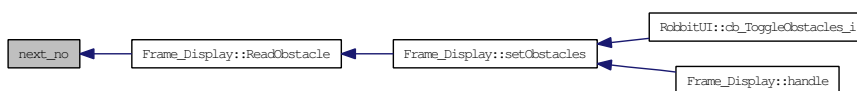
#### Parameters:

- \*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:



## 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.

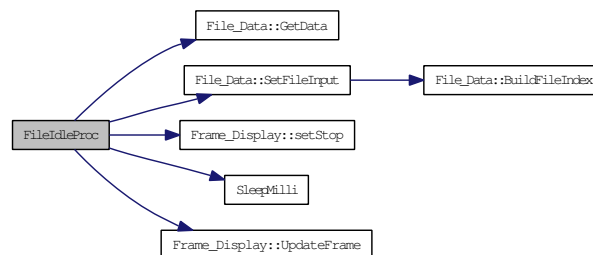
When the simulation is run by reading data from pre-existing log file, this function handles the data handling and updating. It also sets the positions of the various light sources, numbered 0 through 3. It sets the floor size by reading through the data file once, and getting the maximum and minimum 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.

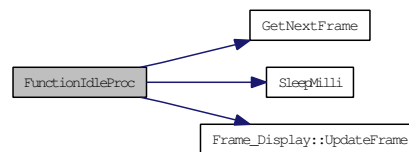
When the simulation is run by creating coordinate data using a custom algorithm, this function is used. Algorithm writers may use the above function, `void FileIdleProc (void *)` as reference while writing this function.

Definition at line 169 of file Robbit.cpp.

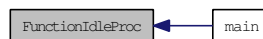
References RobbitUI::display, GetNextFrame(), Frame\_Display::is\_paused, light0\_pos, light1\_pos, light2\_pos, light3\_pos, max\_x, max\_y, min\_x, min\_y, Sleep\_Milli(), Frame\_Display::UpdateFrame(), Frame\_Display::view\_centerX, Frame\_Display::view\_centerY, and Frame\_Display::view\_centerZ.

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

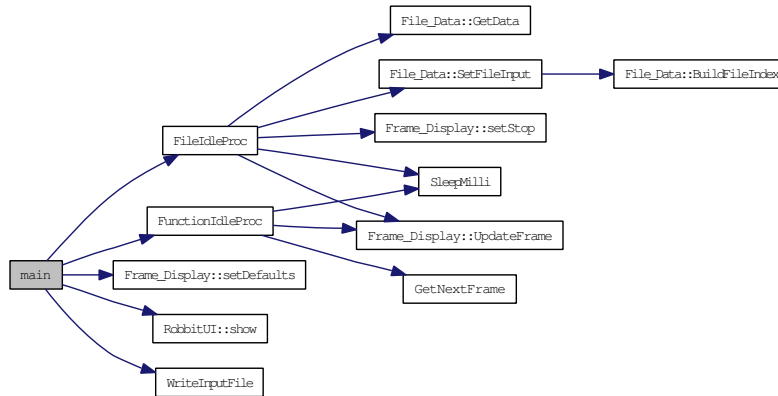
#### Returns:

0, except in case of error

Definition at line 59 of file Robbit.cpp.

References RobbitUI::display, FileIdleProc(), FunctionIdleProc(), RobbitUI::htmlSplash, Frame\_Display::setDefault(), 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 ( $\theta$ ), 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 psuedo code explains 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
}

for i=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

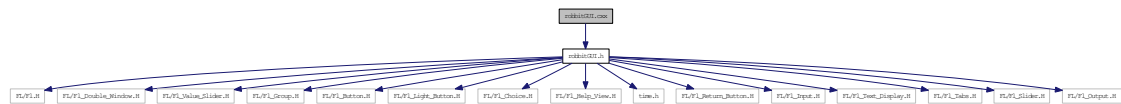
### 9.11.2.1 RobbitUI robbit\_gui

Definition at line 45 of file Robbit.cpp.

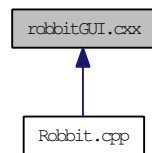
## 9.12 robbitGUI.cxx File Reference

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

Include dependency graph for robbitGUI.cxx:



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



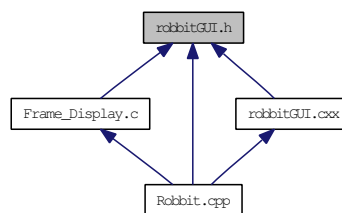
## 9.13 robbitGUI.h File Reference

```
#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:



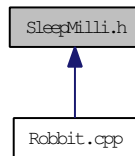
## Classes

- 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)

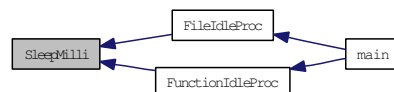
#### 9.14.1 Function Documentation

##### 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:

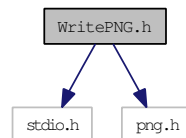


## 9.15 WritePNG.h File Reference

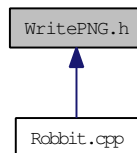
```
#include <stdio.h>
```

```
#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

- ~PNGImage
  - PNGImage, [55](#)
- AboutWindow
  - RobbitUI, [86](#)
- AdvSettingsWindow
  - RobbitUI, [87](#)
- aspect
  - Frame\_Display, [49](#)
- AutoUpdateEye
  - Frame\_Display, [34](#)
- autoview
  - Frame\_Display, [46](#)
- ball\_ambient
  - Definitions.h, [96](#)
- ball\_diffuse
  - Definitions.h, [96](#)
- ball\_radius
  - RobbitUI, [87](#)
- ball\_shininess
  - Definitions.h, [96](#)
- ball\_specular
  - Definitions.h, [96](#)
- ball\_trail
  - Frame\_Display, [48](#)
- ball\_vx
  - Frame\_Data, [21](#)
- ball\_vy
  - Frame\_Data, [21](#)
- ball\_x
  - Frame\_Data, [21](#)
- ball\_y
  - Frame\_Data, [21](#)
- bot\_center\_x
  - Frame\_Data, [21](#)
- bot\_center\_y
  - Frame\_Data, [21](#)
- bot\_design
  - Frame\_Data, [20](#)
- bot\_hit
  - Frame\_Data, [20](#)
- bot\_orient
  - Frame\_Data, [20](#)
- bot\_selected
  - Frame\_Display, [46](#)
- bot\_slices
  - Frame\_Display, [51](#)
- bot\_stacks
  - Frame\_Display, [51](#)
- bot\_trail
  - Frame\_Display, [48](#)
- bot\_vorient
  - Frame\_Data, [20](#)
- bot\_vx
  - Frame\_Data, [20](#)
- bot\_vy
  - Frame\_Data, [20](#)
- bot\_x
  - Frame\_Data, [19](#)
- bot\_y
  - Frame\_Data, [19](#)
- BotHit
  - Frame\_Display, [31](#)
- BotMenu
  - RobbitUI, [84](#)
- BuildFileIndex
  - File\_Data, [16](#)
- ButtonAbout
  - RobbitUI, [82](#)
- ButtonAdvSettings
  - RobbitUI, [83](#)
- ButtonPlayPause
  - RobbitUI, [85](#)
- ButtonReset
  - RobbitUI, [85](#)
- ButtonScreenshot
  - RobbitUI, [83](#)
- ButtonStop
  - RobbitUI, [86](#)
- bytes\_per\_pixel
  - PNGImage, [59](#)

- camera\_on\_bot
  - Frame\_Display, [49](#)
- CaptureScreen
  - CaptureScreen.h, [91](#)
- CaptureScreen.h, [91](#)
  - CaptureScreen, [91](#)
- CaptureScreenshot
  - Frame\_Display, [31](#)
- cb\_0
  - RobbitUI, [70](#)
- cb\_0\_i
  - RobbitUI, [70](#)
- cb\_10X
  - RobbitUI, [71](#)
- cb\_10X\_i
  - RobbitUI, [71](#)
- cb\_1X
  - RobbitUI, [70](#)
- cb\_1X\_i
  - RobbitUI, [70](#)
- cb\_2X
  - RobbitUI, [71](#)
- cb\_2X\_i
  - RobbitUI, [70](#)
- cb\_5X
  - RobbitUI, [71](#)
- cb\_5X\_i
  - RobbitUI, [71](#)
- cb\_About
  - RobbitUI, [76](#)
- cb\_About\_i
  - RobbitUI, [76](#)
- cb\_ball\_radius
  - RobbitUI, [79](#)
- cb\_ball\_radius\_i
  - RobbitUI, [79](#)
- cb\_ButtonAbout
  - RobbitUI, [66](#)
- cb\_ButtonAbout\_i
  - RobbitUI, [66](#)
- cb\_ButtonAdvSettings
  - RobbitUI, [69](#)
- cb\_ButtonAdvSettings\_i
  - RobbitUI, [68](#)
- cb\_ButtonPlayPause
  - RobbitUI, [75](#)
- cb\_ButtonPlayPause\_i
  - RobbitUI, [75](#)
- cb\_ButtonReset
  - RobbitUI, [75](#)
- cb\_ButtonReset\_i
  - RobbitUI, [74](#)
- cb\_ButtonScreenshot
  - RobbitUI, [69](#)
- cb\_ButtonScreenshot\_i
  - RobbitUI, [69](#)
- cb\_ButtonStop
  - RobbitUI, [75](#)
- cb\_ButtonStop\_i
  - RobbitUI, [75](#)
- cb\_Close
  - RobbitUI, [76](#)
- cb\_Close\_i
  - RobbitUI, [76](#)
- cb\_coloured\_steps
  - RobbitUI, [80](#)
- cb\_coloured\_steps\_i
  - RobbitUI, [80](#)
- CB\_counter
  - Frame\_Display, [47](#)
- cb\_Custom
  - RobbitUI, [73](#)
- cb\_Custom\_i
  - RobbitUI, [73](#)
- cb\_graphics\_quality
  - RobbitUI, [81](#)
- cb\_graphics\_quality\_i
  - RobbitUI, [80](#)
- cb\_indexSlider
  - RobbitUI, [65](#)
- cb\_indexSlider\_i
  - RobbitUI, [65](#)
- cb\_KheperaII
  - RobbitUI, [72](#)
- cb\_KheperaII\_i
  - RobbitUI, [72](#)
- cb\_KheperaIII
  - RobbitUI, [72](#)
- cb\_KheperaIII\_i
  - RobbitUI, [72](#)
- cb\_Light1
  - RobbitUI, [73](#)
- cb\_Light1\_i
  - RobbitUI, [73](#)
- cb\_Light2
  - RobbitUI, [73](#)
- cb\_Light2\_i
  - RobbitUI, [73](#)
- cb\_Light3
  - RobbitUI, [74](#)

- cb\_Light3\_i
  - RobbitUI, [74](#)
- cb\_Light4
  - RobbitUI, [74](#)
- cb\_Light4\_i
  - RobbitUI, [74](#)
- cb\_MAX
  - RobbitUI, [72](#)
- cb\_MAX\_i
  - RobbitUI, [71](#)
- cb\_maxx
  - RobbitUI, [78](#)
- cb\_maxx\_i
  - RobbitUI, [78](#)
- cb\_maxy
  - RobbitUI, [77](#)
- cb\_maxy\_i
  - RobbitUI, [77](#)
- cb\_minx
  - RobbitUI, [79](#)
- cb\_minx\_i
  - RobbitUI, [78](#)
- cb\_miny
  - RobbitUI, [78](#)
- cb\_miny\_i
  - RobbitUI, [77](#)
- cb\_OK
  - RobbitUI, [77](#)
- cb\_OK1
  - RobbitUI, [81](#)
- cb\_OK1\_i
  - RobbitUI, [81](#)
- cb\_OK\_i
  - RobbitUI, [76](#)
- cb\_RobbitMainWindow
  - RobbitUI, [65](#)
- cb\_RobbitMainWindow\_i
  - RobbitUI, [65](#)
- cb\_ToggleAutoView
  - RobbitUI, [66](#)
- cb\_ToggleAutoView\_i
  - RobbitUI, [66](#)
- cb\_ToggleBotNumbering
  - RobbitUI, [68](#)
- cb\_ToggleBotNumbering\_i
  - RobbitUI, [68](#)
- cb\_ToggleObstacles
  - RobbitUI, [68](#)
- cb\_ToggleObstacles\_i
  - RobbitUI, [67](#)
- cb\_ToggleTopView
  - RobbitUI, [67](#)
- cb\_ToggleTopView\_i
  - RobbitUI, [66](#)
- cb\_ToggleTrails
  - RobbitUI, [67](#)
- cb\_ToggleTrails\_i
  - RobbitUI, [67](#)
- cb\_trail\_points
  - RobbitUI, [80](#)
- cb\_trail\_points\_i
  - RobbitUI, [79](#)
- coloured\_steps
  - Frame\_Display, [48](#)
  - RobbitUI, [88](#)
- current\_index
  - Frame\_Display, [52](#)
- Custom
  - RobbitUI, [84](#)
- custom\_robot\_window
  - RobbitUI, [86](#)
- cyl\_ambient
  - Definitions.h, [97](#)
- cyl\_diffuse
  - Definitions.h, [97](#)
- cyl\_k2\_specular
  - Definitions.h, [97](#)
- cyl\_k3\_specular
  - Definitions.h, [97](#)
- cyl\_shininess
  - Definitions.h, [97](#)
- data
  - Frame\_Display, [51](#)
- default\_sleep\_time
  - Definitions.h, [97](#)
- Definitions.h, [93](#)
  - ball\_ambient, [96](#)
  - ball\_diffuse, [96](#)
  - ball\_shininess, [96](#)
  - ball\_specular, [96](#)
  - cyl\_ambient, [97](#)
  - cyl\_diffuse, [97](#)
  - cyl\_k2\_specular, [97](#)
  - cyl\_k3\_specular, [97](#)
  - cyl\_shininess, [97](#)
  - default\_sleep\_time, [97](#)
  - disk0\_ambient, [97](#)
  - disk0\_diffuse, [97](#)
  - disk0\_shininess, [98](#)

- disk0\_specular, [98](#)
- disk1\_ambient, [98](#)
- disk1\_diffuse, [98](#)
- disk1 shininess, [98](#)
- disk1\_specular, [98](#)
- disk2\_ambient, [98](#)
- disk2\_diffuse, [98](#)
- disk2 shininess, [98](#)
- disk2\_specular, [98](#)
- disk\_2D\_obstacle shininess, [99](#)
- disk\_2D\_obstacle\_specular, [99](#)
- disk\_center\_ambient, [99](#)
- disk\_center\_diffuse, [99](#)
- disk\_center shininess, [99](#)
- disk\_center\_specular, [99](#)
- floor shininess, [99](#)
- floor\_specular, [99](#)
- font, [99](#)
- height\_obstacle, [95](#)
- info\_refresh\_count, [100](#)
- light0\_ambient, [100](#)
- light0\_diffuse, [100](#)
- light0\_pos, [100](#)
- light0\_specular, [100](#)
- light1\_ambient, [100](#)
- light1\_diffuse, [100](#)
- light1\_pos, [100](#)
- light1\_specular, [101](#)
- light2\_ambient, [101](#)
- light2\_diffuse, [101](#)
- light2\_pos, [101](#)
- light2\_specular, [101](#)
- light3\_ambient, [101](#)
- light3\_diffuse, [101](#)
- light3\_pos, [101](#)
- light3\_specular, [101](#)
- max\_x, [102](#)
- max\_y, [102](#)
- min\_x, [102](#)
- min\_y, [102](#)
- next\_no, [96](#)
- no\_of\_bots, [95](#)
- outputCharacter, [96](#)
- PI, [95](#)
- plane shininess, [102](#)
- plane\_specular, [102](#)
- quad, [103](#)
- radianFactor, [103](#)
- radius\_of\_orient\_disk, [95](#)
- start\_time, [103](#)
- text\_ambient, [103](#)
- text\_diffuse, [103](#)
- text shininess, [103](#)
- text\_specular, [103](#)
- update\_frame\_mode, [103](#)
- delta\_zoom
  - Frame\_Display, [50](#)
- DetectObstacleCollision
  - Frame\_Display, [32](#)
- disk0\_ambient
  - Definitions.h, [97](#)
- disk0\_diffuse
  - Definitions.h, [97](#)
- disk0 shininess
  - Definitions.h, [98](#)
- disk0\_specular
  - Definitions.h, [98](#)
- disk1\_ambient
  - Definitions.h, [98](#)
- disk1\_diffuse
  - Definitions.h, [98](#)
- disk1 shininess
  - Definitions.h, [98](#)
- disk1\_specular
  - Definitions.h, [98](#)
- disk2\_ambient
  - Definitions.h, [98](#)
- disk2\_diffuse
  - Definitions.h, [98](#)
- disk2 shininess
  - Definitions.h, [98](#)
- disk2\_specular
  - Definitions.h, [98](#)
- disk\_2D\_obstacle shininess
  - Definitions.h, [99](#)
- disk\_2D\_obstacle\_specular
  - Definitions.h, [99](#)
- disk\_center\_ambient
  - Definitions.h, [99](#)
- disk\_center\_diffuse
  - Definitions.h, [99](#)
- disk\_center shininess
  - Definitions.h, [99](#)
- disk\_center\_specular
  - Definitions.h, [99](#)
- display
  - RobbitUI, [82](#)
- DistancePointLine
  - DistancePointLine.h, [105](#)
- DistancePointLine.h, [105](#)

- DistancePointLine, 105
- Magnitude, 106
- XY, 105
- draw
  - Frame\_Display, 29
- DrawFloor
  - Frame\_Display, 29
- DrawObstacle
  - Frame\_Display, 30
- DrawTrails
  - Frame\_Display, 31
- eyeX
  - Frame\_Display, 48
- eyeY
  - Frame\_Display, 48
- eyeZ
  - Frame\_Display, 49
- Far
  - Frame\_Display, 49
- File\_Data, 15
  - BuildFileIndex, 16
  - file\_index, 18
  - GetData, 17
  - IndexSize, 17
  - input, 18
  - pos\_type, 16
  - SetFileInput, 16
- File\_Data.c, 107
- File\_Data.h, 108
- file\_index
  - File\_Data, 18
- FileIdleProc
  - Robbit.cpp, 115
- filename
  - PNGImage, 58
- FloatVec
  - Frame\_Data.h, 109
- floor\_shininess
  - Definitions.h, 99
- floor\_specular
  - Definitions.h, 99
- font
  - Definitions.h, 99
- fp
  - PNGImage, 58
- Frame\_Data, 19
  - ball\_vx, 21
  - ball\_vy, 21
  - ball\_x, 21
  - ball\_y, 21
  - bot\_center\_x, 21
  - bot\_center\_y, 21
  - bot\_design, 20
  - bot\_hit, 20
  - bot\_orient, 20
  - bot\_vorient, 20
  - bot\_vx, 20
  - bot\_vy, 20
  - bot\_x, 19
  - bot\_y, 19
  - time, 20
  - time\_step, 21
- Frame\_Data.h, 109
  - FloatVec, 109
  - IntVec, 109
- Frame\_Display, 23
  - aspect, 49
  - AutoUpdateEye, 34
  - autoview, 46
  - ball\_trail, 48
  - bot\_selected, 46
  - bot\_slices, 51
  - bot\_stacks, 51
  - bot\_trail, 48
  - BotHit, 31
  - camera\_on\_bot, 49
  - CaptureScreenshot, 31
  - CB\_counter, 47
  - coloured\_steps, 48
  - current\_index, 52
  - data, 51
  - delta\_zoom, 50
  - DetectObstacleCollision, 32
  - draw, 29
  - DrawFloor, 29
  - DrawObstacle, 30
  - DrawTrails, 31
  - eyeX, 48
  - eyeY, 48
  - eyeZ, 49
  - Far, 49
  - Frame\_Display, 27
  - Frame\_Display, 27
  - handle, 27
  - height\_robot, 50
  - init, 35
  - initAdvWindow, 45
  - is\_paused, 52

- LineEnd, [51](#)
- LineStart, [51](#)
- mousePushValid, [50](#)
- mouseX0, [50](#)
- mouseY0, [50](#)
- Near, [49](#)
- no\_of\_lights, [50](#)
- no\_of\_trail\_data, [47](#)
- number\_bots, [46](#)
- obs\_2D, [46](#)
- obs\_2D\_counter, [47](#)
- obs\_CB, [47](#)
- obs\_SP, [47](#)
- obs\_WL, [46](#)
- outputCharacter, [36](#)
- phi, [48](#)
- Point, [51](#)
- position\_khepera2, [33](#)
- position\_khepera3, [33](#)
- PositionCustomRobot, [34](#)
- radius\_ball, [47](#)
- radius\_camera\_movement, [48](#)
- radius\_of\_robot, [50](#)
- ReadObstacle, [30](#)
- render\_speed, [52](#)
- reshape, [35](#)
- selectBot, [40](#)
- setArena, [43](#)
- setAutoView, [40](#)
- setBallRadius, [43](#)
- setBotNumbering, [38](#)
- setColouredSteps, [44](#)
- setCustomRobot, [42](#)
- setDefaults, [42](#)
- setExit, [37](#)
- setGraphicsQuality, [45](#)
- setIndex, [36](#)
- setLights, [40](#)
- setObstacles, [39](#)
- setPlayPause, [36](#)
- setRenderSpeed, [41](#)
- setReset, [37](#)
- setStop, [37](#)
- setTopView, [39](#)
- setTrailPoints, [44](#)
- setTrails, [38](#)
- show\_obstacle, [46](#)
- show\_trails, [46](#)
- small\_disk\_slices, [51](#)
- small\_disk\_stacks, [51](#)
- SP\_counter, [47](#)
- theta, [48](#)
- topview, [46](#)
- UpdateFrame, [28](#)
- upX, [49](#)
- upY, [49](#)
- upZ, [49](#)
- view\_centerX, [52](#)
- view\_centerY, [52](#)
- view\_centerZ, [52](#)
- WL\_counter, [47](#)
- Frame\_Display.c, [110](#)
- robbit\_gui, [110](#)
- Frame\_Display.h, [111](#)
- FunctionIdleProc
- Robbit.cpp, [115](#)
- GetData
- File\_Data, [17](#)
- GetNextFrame
- GetNextFrame.h, [112](#)
- GetNextFrame.h, [112](#)
- GetNextFrame, [112](#)
- graphics\_quality
- RobbitUI, [88](#)
- handle
- Frame\_Display, [27](#)
- height
- PNGImage, [58](#)
- RobbitUI, [86](#)
- height\_obstacle
- Definitions.h, [95](#)
- height\_robot
- Frame\_Display, [50](#)
- htmlSplash
- RobbitUI, [86](#)
- i
- PNGImage, [59](#)
- IndexSize
- File\_Data, [17](#)
- indexSlider
- RobbitUI, [82](#)
- info\_ptr
- PNGImage, [58](#)
- info\_refresh\_count
- Definitions.h, [100](#)
- init
- Frame\_Display, [35](#)



- initAdvWindow
  - Frame\_Display, 45
- initReadStructs
  - PNGImage, 56
- initWriteStructs
  - PNGImage, 56
- input
  - File\_Data, 18
- IntVec
  - Frame\_Data.h, 109
- is\_paused
  - Frame\_Display, 52
- isPNG
  - PNGImage, 56
- j
  - PNGImage, 59
- KheperaII
  - RobbitUI, 84
- KheperaIII
  - RobbitUI, 84
- light0\_ambient
  - Definitions.h, 100
- light0\_diffuse
  - Definitions.h, 100
- light0\_pos
  - Definitions.h, 100
- light0\_specular
  - Definitions.h, 100
- Light1
  - RobbitUI, 85
- light1\_ambient
  - Definitions.h, 100
- light1\_diffuse
  - Definitions.h, 100
- light1\_pos
  - Definitions.h, 100
- light1\_specular
  - Definitions.h, 101
- Light2
  - RobbitUI, 85
- light2\_ambient
  - Definitions.h, 101
- light2\_diffuse
  - Definitions.h, 101
- light2\_pos
  - Definitions.h, 101
- light2\_specular
  - Definitions.h, 101
- Light3
  - RobbitUI, 85
- light3\_ambient
  - Definitions.h, 101
- light3\_diffuse
  - Definitions.h, 101
- light3\_pos
  - Definitions.h, 101
- light3\_specular
  - Definitions.h, 101
- Light4
  - RobbitUI, 85
- LightMenu
  - RobbitUI, 84
- LineEnd
  - Frame\_Display, 51
- LineStart
  - Frame\_Display, 51
- Magnitude
  - DistancePointLine.h, 106
- main
  - Robbit.cpp, 116
- max\_x
  - Definitions.h, 102
- max\_y
  - Definitions.h, 102
- maxx
  - RobbitUI, 87
- maxy
  - RobbitUI, 87
- menu\_BotMenu
  - RobbitUI, 84
- menu\_LightMenu
  - RobbitUI, 85
- menu\_RenderSpeedMenu
  - RobbitUI, 83
- min\_x
  - Definitions.h, 102
- min\_y
  - Definitions.h, 102
- minx
  - RobbitUI, 87
- miny
  - RobbitUI, 87
- mousePushValid
  - Frame\_Display, 50
- mouseX0
  - Frame\_Display, 50

- mouseY0
  - Frame\_Display, 50
- Near
  - Frame\_Display, 49
- next\_no
  - Definitions.h, 96
  - NextNo.h, 113
- NextNo.h, 113
  - next\_no, 113
- no\_of\_bots
  - Definitions.h, 95
- no\_of\_lights
  - Frame\_Display, 50
- no\_of\_trail\_data
  - Frame\_Display, 47
- number\_bots
  - Frame\_Display, 46
- obs\_2D
  - Frame\_Display, 46
- obs\_2D\_counter
  - Frame\_Display, 47
- obs\_CB
  - Frame\_Display, 47
- obs\_SP
  - Frame\_Display, 47
- obs\_WL
  - Frame\_Display, 46
- openFileRead
  - PNGImage, 55
- openFileWrite
  - PNGImage, 55
- outputCharacter
  - Definitions.h, 96
  - Frame\_Display, 36
- phi
  - Frame\_Display, 48
- PI
  - Definitions.h, 95
- plane\_shininess
  - Definitions.h, 102
- plane\_specular
  - Definitions.h, 102
- png\_ptr
  - PNGImage, 58
- PNGImage, 54
  - ~PNGImage, 55
  - bytes\_per\_pixel, 59
  - filename, 58
  - fp, 58
  - height, 58
  - i, 59
  - info\_ptr, 58
  - initReadStructs, 56
  - initWriteStructs, 56
  - isPNG, 56
  - j, 59
  - openFileRead, 55
  - openFileWrite, 55
  - png\_ptr, 58
  - PNGImage, 55
  - rgb, 59
  - row, 59
  - width, 58
  - width\_para, 59
  - writeEnd, 57
  - writeHeader, 57
  - writeImage, 57
- Point
  - Frame\_Display, 51
- pos\_type
  - File\_Data, 16
- position\_khepera2
  - Frame\_Display, 33
- position\_khepera3
  - Frame\_Display, 33
- PositionCustomRobot
  - Frame\_Display, 34
- quad
  - Definitions.h, 103
- radianFactor
  - Definitions.h, 103
- radius
  - RobbitUI, 87
- radius\_ball
  - Frame\_Display, 47
- radius\_camera\_movement
  - Frame\_Display, 48
- radius\_of\_orient\_disk
  - Definitions.h, 95
- radius\_of\_robot
  - Frame\_Display, 50
- ReadObstacle
  - Frame\_Display, 30
- render\_speed
  - Frame\_Display, 52

- RenderSpeedMenu
  - RobbitUI, 83
- reshape
  - Frame\_Display, 35
- rgb
  - PNGImage, 59
- Robbit.cpp, 114
  - FileIdleProc, 115
  - FunctionIdleProc, 115
  - main, 116
  - robbit\_gui, 118
  - WriteInputFile, 117
- robbit\_gui
  - Frame\_Display.c, 110
  - Robbit.cpp, 118
- robbitGUI.cxx, 119
- robbitGUI.h, 120
- RobbitMainWindow
  - RobbitUI, 82
- RobbitUI, 60
  - AboutWindow, 86
  - AdvSettingsWindow, 87
  - ball\_radius, 87
  - BotMenu, 84
  - ButtonAbout, 82
  - ButtonAdvSettings, 83
  - ButtonPlayPause, 85
  - ButtonReset, 85
  - ButtonScreenshot, 83
  - ButtonStop, 86
  - cb\_0, 70
  - cb\_0\_i, 70
  - cb\_10X, 71
  - cb\_10X\_i, 71
  - cb\_1X, 70
  - cb\_1X\_i, 70
  - cb\_2X, 71
  - cb\_2X\_i, 70
  - cb\_5X, 71
  - cb\_5X\_i, 71
  - cb\_About, 76
  - cb\_About\_i, 76
  - cb\_ball\_radius, 79
  - cb\_ball\_radius\_i, 79
  - cb\_ButtonAbout, 66
  - cb\_ButtonAbout\_i, 66
  - cb\_ButtonAdvSettings, 69
  - cb\_ButtonAdvSettings\_i, 68
  - cb\_ButtonPlayPause, 75
  - cb\_ButtonPlayPause\_i, 75
  - cb\_ButtonReset, 75
  - cb\_ButtonReset\_i, 74
  - cb\_ButtonScreenshot, 69
  - cb\_ButtonScreenshot\_i, 69
  - cb\_ButtonStop, 75
  - cb\_ButtonStop\_i, 75
  - cb\_Close, 76
  - cb\_Close\_i, 76
  - cb\_coloured\_steps, 80
  - cb\_coloured\_steps\_i, 80
  - cb\_Custom, 73
  - cb\_Custom\_i, 73
  - cb\_graphics\_quality, 81
  - cb\_graphics\_quality\_i, 80
  - cb\_indexSlider, 65
  - cb\_indexSlider\_i, 65
  - cb\_KheperaII, 72
  - cb\_KheperaII\_i, 72
  - cb\_KheperaIII, 72
  - cb\_KheperaIII\_i, 72
  - cb\_Light1, 73
  - cb\_Light1\_i, 73
  - cb\_Light2, 73
  - cb\_Light2\_i, 73
  - cb\_Light3, 74
  - cb\_Light3\_i, 74
  - cb\_Light4, 74
  - cb\_Light4\_i, 74
  - cb\_MAX, 72
  - cb\_MAX\_i, 71
  - cb\_maxx, 78
  - cb\_maxx\_i, 78
  - cb\_maxy, 77
  - cb\_maxy\_i, 77
  - cb\_minx, 79
  - cb\_minx\_i, 78
  - cb\_miny, 78
  - cb\_miny\_i, 77
  - cb\_OK, 77
  - cb\_OK1, 81
  - cb\_OK1\_i, 81
  - cb\_OK\_i, 76
  - cb\_RobbitMainWindow, 65
  - cb\_RobbitMainWindow\_i, 65
  - cb\_ToggleAutoView, 66
  - cb\_ToggleAutoView\_i, 66
  - cb\_ToggleBotNumbering, 68
  - cb\_ToggleBotNumbering\_i, 68
  - cb\_ToggleObstacles, 68
  - cb\_ToggleObstacles\_i, 67

- cb\_ToggleTopView, 67
- cb\_ToggleTopView\_i, 66
- cb\_ToggleTrails, 67
- cb\_ToggleTrails\_i, 67
- cb\_trail\_points, 80
- cb\_trail\_points\_i, 79
- coloured\_steps, 88
- Custom, 84
- custom\_robot\_window, 86
- display, 82
- graphics\_quality, 88
- height, 86
- htmlSplash, 86
- indexSlider, 82
- KheperaII, 84
- KheperaIII, 84
- Light1, 85
- Light2, 85
- Light3, 85
- Light4, 85
- LightMenu, 84
- maxx, 87
- maxy, 87
- menu\_BotMenu, 84
- menu\_LightMenu, 85
- menu\_RenderSpeedMenu, 83
- minx, 87
- miny, 87
- radius, 87
- RenderSpeedMenu, 83
- RobbitMainWindow, 82
- RobbitUI, 63
- show, 81
- SplashWindow, 86
- textrobbit, 86
- ToggleAutoView, 82
- ToggleBotNumbering, 83
- ToggleObstacles, 83
- ToggleTopView, 82
- ToggleTrails, 83
- trail\_points, 87
- row
  - PNGImage, 59
- selectBot
  - Frame\_Display, 40
- setArena
  - Frame\_Display, 43
- setAutoView
  - Frame\_Display, 40
- setBallRadius
  - Frame\_Display, 43
- setBotNumbering
  - Frame\_Display, 38
- setColouredSteps
  - Frame\_Display, 44
- setCustomRobot
  - Frame\_Display, 42
- setDefaults
  - Frame\_Display, 42
- setExit
  - Frame\_Display, 37
- SetFileInput
  - File\_Data, 16
- setGraphicsQuality
  - Frame\_Display, 45
- setIndex
  - Frame\_Display, 36
- setLights
  - Frame\_Display, 40
- setObstacles
  - Frame\_Display, 39
- setPlayPause
  - Frame\_Display, 36
- setRenderSpeed
  - Frame\_Display, 41
- setReset
  - Frame\_Display, 37
- setStop
  - Frame\_Display, 37
- setTopView
  - Frame\_Display, 39
- setTrailPoints
  - Frame\_Display, 44
- setTrails
  - Frame\_Display, 38
- show
  - RobbitUI, 81
- show\_obstacle
  - Frame\_Display, 46
- show\_trails
  - Frame\_Display, 46
- SleepMilli
  - SleepMilli.h, 121
- SleepMilli.h, 121
  - SleepMilli, 121
- small\_disk\_slices
  - Frame\_Display, 51
- small\_disk\_stacks
  - Frame\_Display, 51

- SP\_counter
  - Frame\_Display, [47](#)
- SplashWindow
  - RobbitUI, [86](#)
- start\_time
  - Definitions.h, [103](#)
- tagXY, [89](#)
  - X, [89](#)
  - Y, [89](#)
- text\_ambient
  - Definitions.h, [103](#)
- text\_diffuse
  - Definitions.h, [103](#)
- text\_shininess
  - Definitions.h, [103](#)
- text\_specular
  - Definitions.h, [103](#)
- textrobbit
  - RobbitUI, [86](#)
- theta
  - Frame\_Display, [48](#)
- time
  - Frame\_Data, [20](#)
- time\_step
  - Frame\_Data, [21](#)
- ToggleAutoView
  - RobbitUI, [82](#)
- ToggleBotNumbering
  - RobbitUI, [83](#)
- ToggleObstacles
  - RobbitUI, [83](#)
- ToggleTopView
  - RobbitUI, [82](#)
- ToggleTrails
  - RobbitUI, [83](#)
- topview
  - Frame\_Display, [46](#)
- trail\_points
  - RobbitUI, [87](#)
- update\_frame\_mode
  - Definitions.h, [103](#)
- UpdateFrame
  - Frame\_Display, [28](#)
- upX
  - Frame\_Display, [49](#)
- upY
  - Frame\_Display, [49](#)
- upZ
  - Frame\_Display, [49](#)
- view\_centerX
  - Frame\_Display, [52](#)
- view\_centerY
  - Frame\_Display, [52](#)
- view\_centerZ
  - Frame\_Display, [52](#)
- width
  - PNGImage, [58](#)
- width\_para
  - PNGImage, [59](#)
- WL\_counter
  - Frame\_Display, [47](#)
- writeEnd
  - PNGImage, [57](#)
- writeHeader
  - PNGImage, [57](#)
- writeImage
  - PNGImage, [57](#)
- WriteInputFile
  - Robbit.cpp, [117](#)
- WritePNG.h, [122](#)
- X
  - tagXY, [89](#)
- XY
  - DistancePointLine.h, [105](#)
- Y
  - tagXY, [89](#)