

My Project

Generated by Doxygen 1.8.17

1 Namespace Index	1
1.1 Namespace List	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	5
3.1 Class List	5
4 File Index	7
4.1 File List	7
5 Namespace Documentation	9
5.1 Ui Namespace Reference	9
6 Class Documentation	11
6.1 Button Class Reference	11
6.1.1 Detailed Description	13
6.1.2 Constructor & Destructor Documentation	13
6.1.2.1 Button()	13
6.1.2.2 ~Button()	13
6.1.3 Member Function Documentation	13
6.1.3.1 autoPilot	13
6.1.3.2 mousePressEvent()	14
6.1.3.3 mouseReleaseEvent()	15
6.1.3.4 pressed	15
6.1.3.5 released	15
6.1.3.6 setBorderWidth	15
6.2 Enemy Class Reference	16
6.2.1 Detailed Description	18
6.2.2 Constructor & Destructor Documentation	18
6.2.2.1 Enemy()	18
6.2.2.2 ~Enemy()	18
6.2.3 Member Function Documentation	18
6.2.3.1 checkStuck	19
6.2.3.2 getUnstuck	19
6.2.3.3 hit	19
6.2.3.4 resumeAllTimers	19
6.2.3.5 startAutonomousMovement	19
6.2.3.6 stopAllTimers	19
6.2.3.7 stopChasing	20
6.3 EnemyItem Class Reference	20
6.3.1 Detailed Description	22
6.3.2 Constructor & Destructor Documentation	22

6.3.2.1 EnemyItem()	22
6.3.2.2 ~EnemyItem()	22
6.3.3 Member Function Documentation	22
6.3.3.1 getDistance()	22
6.3.3.2 getOrientation()	22
6.3.3.3 getRotationAngle()	23
6.3.3.4 getVelocity()	23
6.3.3.5 setDistance()	23
6.3.3.6 setOrientation()	23
6.3.3.7 setRotationAngle()	24
6.3.3.8 setVelocity()	24
6.4 GameInfo Class Reference	24
6.4.1 Detailed Description	26
6.4.2 Constructor & Destructor Documentation	26
6.4.2.1 GameInfo()	26
6.4.2.2 ~GameInfo()	26
6.4.3 Member Function Documentation	27
6.4.3.1 damage	27
6.4.3.2 ouch	27
6.4.3.3 resetMessage	27
6.4.3.4 updateMessage()	27
6.5 GameMaster Class Reference	28
6.5.1 Detailed Description	29
6.5.2 Member Function Documentation	29
6.5.2.1 run()	29
6.6 Heart Class Reference	29
6.6.1 Detailed Description	30
6.6.2 Constructor & Destructor Documentation	30
6.6.2.1 Heart()	30
6.6.2.2 ~Heart()	31
6.6.3 Member Function Documentation	31
6.6.3.1 setDead	31
6.7 HorizontalLowerBar Class Reference	31
6.7.1 Detailed Description	32
6.7.2 Constructor & Destructor Documentation	32
6.7.2.1 HorizontalLowerBar()	32
6.7.2.2 ~HorizontalLowerBar()	33
6.7.3 Member Function Documentation	33
6.7.3.1 mouseReleaseEvent()	33
6.7.4 Member Data Documentation	33
6.7.4.1 Autopilot	33
6.8 HorizontalUpperBar Class Reference	34

6.8.1 Detailed Description	35
6.8.2 Constructor & Destructor Documentation	35
6.8.2.1 HorizontalUpperBar()	36
6.8.2.2 ~HorizontalUpperBar()	36
6.8.3 Member Data Documentation	36
6.8.3.1 pauseButton	36
6.8.3.2 timer	36
6.9 MainWindow Class Reference	37
6.9.1 Detailed Description	38
6.9.2 Constructor & Destructor Documentation	38
6.9.2.1 MainWindow()	38
6.9.2.2 ~MainWindow()	38
6.9.3 Member Function Documentation	38
6.9.3.1 applyGraphics()	39
6.9.3.2 createNewMapWindow	39
6.9.3.3 loadFile	39
6.9.3.4 loadGame()	39
6.9.3.5 quitApp()	39
6.10 MapWindow Class Reference	40
6.10.1 Detailed Description	42
6.10.2 Constructor & Destructor Documentation	42
6.10.2.1 MapWindow()	42
6.10.2.2 ~MapWindow()	42
6.10.3 Member Function Documentation	42
6.10.3.1 clearTableCell()	42
6.10.3.2 disableEditing()	43
6.10.3.3 fillFile()	43
6.10.3.4 setBarBottom()	43
6.10.3.5 setBarTop()	43
6.10.3.6 setTableCell()	44
6.10.3.7 startSession	44
6.10.3.8 updateEnemyCounter()	44
6.10.3.9 updateObstacleCounter()	44
6.10.3.10 updateRobotCounter()	44
6.10.3.11 updateTimer()	45
6.10.4 Member Data Documentation	45
6.10.4.1 mapData	45
6.11 Obstacle Class Reference	45
6.11.1 Detailed Description	46
6.11.2 Constructor & Destructor Documentation	46
6.11.2.1 Obstacle()	46
6.11.2.2 ~Obstacle()	47

6.12 ObstacleItem Class Reference	47
6.12.1 Detailed Description	48
6.12.2 Constructor & Destructor Documentation	48
6.12.2.1 ObstacleItem()	48
6.13 pixmapIcons Struct Reference	49
6.13.1 Detailed Description	49
6.13.2 Member Data Documentation	49
6.13.2.1 enemy	49
6.13.2.2 obstacle	50
6.13.2.3 robot	50
6.14 PopUp Class Reference	50
6.14.1 Detailed Description	52
6.14.2 Constructor & Destructor Documentation	52
6.14.2.1 PopUp()	52
6.14.2.2 ~PopUp()	52
6.14.3 Member Function Documentation	52
6.14.3.1 hide	53
6.14.3.2 pauseTimers	53
6.14.3.3 show	53
6.14.4 Member Data Documentation	53
6.14.4.1 exitButton	53
6.14.4.2 mainMenuButton	53
6.14.4.3 restartButton	53
6.14.4.4 resumeButton	54
6.15 QDialog Class Reference	54
6.16 QGraphicsEllipseItem Class Reference	55
6.17 QGraphicsRectItem Class Reference	56
6.18 QGraphicsTextItem Class Reference	57
6.19 QMainWindow Class Reference	58
6.20 QObject Class Reference	59
6.21 QTableWidgetItem Class Reference	60
6.22 QWidget Class Reference	61
6.23 RobotItem Class Reference	62
6.23.1 Detailed Description	63
6.23.2 Constructor & Destructor Documentation	63
6.23.2.1 RobotItem()	64
6.23.2.2 ~RobotItem()	64
6.23.3 Member Function Documentation	64
6.23.3.1 getOrientation()	64
6.23.3.2 getVelocity()	64
6.23.3.3 setOrientation()	64
6.23.3.4 setVelocity()	65

6.24 RobotParamDialog Class Reference	65
6.24.1 Detailed Description	67
6.24.2 Constructor & Destructor Documentation	67
6.24.2.1 RobotParamDialog()	67
6.24.2.2 ~RobotParamDialog()	67
6.24.3 Member Function Documentation	67
6.24.3.1 createLayout()	67
6.24.3.2 getDistance()	68
6.24.3.3 getOrientation()	68
6.24.3.4 getRotationAngle()	68
6.24.3.5 getVelocity()	68
6.25 Star Class Reference	69
6.25.1 Detailed Description	70
6.25.2 Constructor & Destructor Documentation	70
6.25.2.1 Star()	70
6.25.2.2 ~Star()	70
6.25.3 Member Function Documentation	70
6.25.3.1 move	71
6.26 Timer Class Reference	71
6.26.1 Detailed Description	72
6.26.2 Constructor & Destructor Documentation	73
6.26.2.1 Timer()	73
6.26.2.2 ~Timer()	73
6.26.3 Member Function Documentation	73
6.26.3.1 continueTimer	73
6.26.3.2 stopTimer	73
6.26.3.3 timersUp	74
6.26.3.4 updateTimer	74
6.27 User Class Reference	74
6.27.1 Detailed Description	76
6.27.2 Constructor & Destructor Documentation	76
6.27.2.1 User()	76
6.27.2.2 ~User()	77
6.27.3 Member Function Documentation	77
6.27.3.1 decreaseLives	77
6.27.3.2 deleteLife1	77
6.27.3.3 deleteLife2	77
6.27.3.4 deleteLife3	77
6.27.3.5 keyPressEvent()	77
6.27.3.6 resumeAllTimers	78
6.27.3.7 startMovingForward	78
6.27.3.8 startRotatingClockwise	78

6.27.3.9 startRotatingCounterClockwise	78
6.27.3.10 stopAllTimers	78
6.27.3.11 stopMoving	79
6.27.3.12 stopRotating	79
6.27.3.13 stunned	79
6.27.3.14 switchControl	79
6.27.4 Member Data Documentation	79
6.27.4.1 visionPoint	79
7 File Documentation	81
7.1 button.cpp File Reference	81
7.1.1 Detailed Description	81
7.2 button.h File Reference	82
7.2.1 Detailed Description	83
7.3 enemy.cpp File Reference	83
7.3.1 Detailed Description	83
7.4 enemy.h File Reference	84
7.4.1 Detailed Description	85
7.5 enemyitem.cpp File Reference	85
7.5.1 Detailed Description	85
7.6 enemyitem.h File Reference	86
7.6.1 Detailed Description	86
7.7 gameinfo.cpp File Reference	87
7.7.1 Detailed Description	87
7.8 gameinfo.h File Reference	87
7.8.1 Detailed Description	88
7.9 gamemaster.cpp File Reference	88
7.9.1 Detailed Description	89
7.10 gamemaster.h File Reference	89
7.10.1 Detailed Description	90
7.11 heart.cpp File Reference	90
7.11.1 Detailed Description	91
7.12 heart.h File Reference	91
7.12.1 Detailed Description	92
7.13 horizontalbar.cpp File Reference	92
7.13.1 Detailed Description	93
7.14 horizontalbar.h File Reference	93
7.14.1 Detailed Description	94
7.15 main.cpp File Reference	94
7.15.1 Detailed Description	95
7.15.2 Function Documentation	95
7.15.2.1 main()	95

7.16 mainwindow.cpp File Reference	95
7.16.1 Detailed Description	96
7.17 mainwindow.h File Reference	96
7.17.1 Detailed Description	97
7.18 mapwindow.cpp File Reference	97
7.18.1 Detailed Description	97
7.19 mapwindow.h File Reference	98
7.19.1 Detailed Description	99
7.19.2 Macro Definition Documentation	99
7.19.2.1 SQUARE_SIZE	99
7.19.3 Typedef Documentation	99
7.19.3.1 pixMapIcons	99
7.20 obstacle.cpp File Reference	100
7.20.1 Detailed Description	100
7.21 obstacle.h File Reference	100
7.21.1 Detailed Description	101
7.22 obstacleitem.h File Reference	101
7.22.1 Detailed Description	102
7.23 popup.cpp File Reference	103
7.23.1 Detailed Description	103
7.24 popup.h File Reference	103
7.24.1 Detailed Description	104
7.25 robotdialog.cpp File Reference	104
7.25.1 Detailed Description	105
7.26 robotdialog.h File Reference	105
7.26.1 Detailed Description	106
7.27 robotitem.cpp File Reference	106
7.27.1 Detailed Description	107
7.28 robotitem.h File Reference	107
7.28.1 Detailed Description	108
7.29 star.cpp File Reference	108
7.29.1 Detailed Description	109
7.30 star.h File Reference	109
7.30.1 Detailed Description	110
7.31 timer.cpp File Reference	111
7.31.1 Detailed Description	111
7.32 timer.h File Reference	111
7.32.1 Detailed Description	112
7.33 user.cpp File Reference	113
7.33.1 Detailed Description	113
7.34 user.h File Reference	113
7.34.1 Detailed Description	114

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Ui	9
----------	---

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

pixmapIcons	49
QDialog	54
RobotParamDialog	65
QGraphicsEllipseItem	55
Enemy	16
Star	69
User	74
QGraphicsRectItem	56
Button	11
Heart	29
HorizontalLowerBar	31
HorizontalUpperBar	34
Obstacle	45
PopUp	50
QGraphicsTextItem	57
GameInfo	24
Timer	71
QMainWindow	58
MainWindow	37
QObject	59
Button	11
Enemy	16
GameMaster	28
Heart	29
HorizontalLowerBar	31
HorizontalUpperBar	34
PopUp	50
Star	69
User	74
QTableWidgetItem	60
EnemyItem	20
ObstacleItem	47
RobotItem	62
QWidget	61
MapWindow	40

Chapter 3

Class Index

3.1 Class List

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

Button		
	Button class, inherits from QObject and QGraphicsRectItem	11
Enemy		
	Enemy class, inherits from QObject and QGraphicsEllipseItem	16
EnemyItem		
	The EnemyItem class	20
GameInfo		
	GameInfo class, inherits from QObject and QGraphicsTextItem	24
GameMaster		
	GameMaster class, inherits from QObject	28
Heart		
	Heart class, inherits from QObject and QGraphicsRectItem	29
HorizontalLowerBar		
	HorizontalLowerBar class, inherits from QObject and QGraphicsRectItem	31
HorizontalUpperBar		
	HorizontalUpperBar class, inherits from QObject and QGraphicsRectItem	34
MainWindow		
	The MainWindow class	37
MapWindow		
	Class for the Map Window	40
Obstacle		
	Obstacle class, inherits from QGraphicsRectItem	45
ObstacleItem		
	The ObstacleItem class	47
pixmapIcons		
	Struct to store the icons for the pixmap	49
PopUp		
	Popup class, inherits from QObject and QGraphicsRectItem	50
QDialog		54
QGraphicsEllipseItem		55
QGraphicsRectItem		56
QGraphicsTextItem		57
QMainWindow		58
QObject		59
QTableWidgetItem		60

QWidget	61
RobotItem	
The RobotItem class	62
RobotParamDialog	
The RobotParamDialog class	65
Star	
Star class, inherits from QObject and QGraphicsEllipseItem	69
Timer	
Timer class, inherits from QGraphicsTextItem	71
User	
User class, inherits from QObject and QGraphicsEllipseItem	74

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

button.cpp		
Button class implementation file	81	
button.h		
Button class header file	82	
enemy.cpp		
Enemy class implementation file	83	
enemy.h		
Enemy class header file	84	
enemyitem.cpp		
Enemy Item Class Implementation file	85	
enemyitem.h		
Enemy Item Class header file	86	
gameinfo.cpp		
GameInfo class implementation file	87	
gameinfo.h		
GameInfo class header file	87	
gamemaster.cpp		
GameMaster class implementation file	88	
gamemaster.h		
GameMaster class header file	89	
heart.cpp		
Hnemy class implementation file	90	
heart.h		
Heart class header file	91	
horizontalbar.cpp		
HorizontalBar classes (upper, lower) implementation file	92	
horizontalbar.h		
HorizontalBar classes (upper, lower) header file	93	
main.cpp		
Main file of the project	94	
mainwindow.cpp		
Main Window Class Implementation file	95	
mainwindow.h		
Main Window Class Header file	96	
mapwindow.cpp		
Map Window Class Implementation file	97	

mapwindow.h	Map Window Class Header file	98
obstacle.cpp	Obstacle class implementation file	100
obstacle.h	Obstacle class header file	100
obstacleitem.h	Obstacle Item Class Header file	101
popup.cpp	Popup class implementation file	103
popup.h	Popup class header file	103
robotdialog.cpp	Robot Dialog Class Implementation file	104
robotdialog.h	Robot Dialog Class Header file	105
robotitem.cpp	Robot Item Class Implementation file	106
robotitem.h	Robot Item Class header file	107
star.cpp	Star class implementation file	108
star.h	Star class header file	109
timer.cpp	Timer class implementation file	111
timer.h	Timer class header file	111
user.cpp	User class implementation file	113
user.h	User class header file	113

Chapter 5

Namespace Documentation

5.1 Ui Namespace Reference

Chapter 6

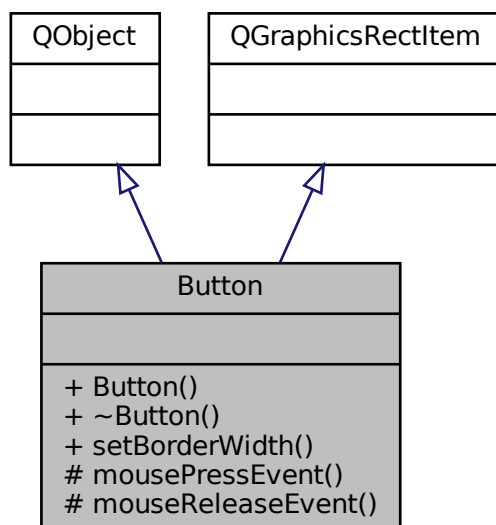
Class Documentation

6.1 Button Class Reference

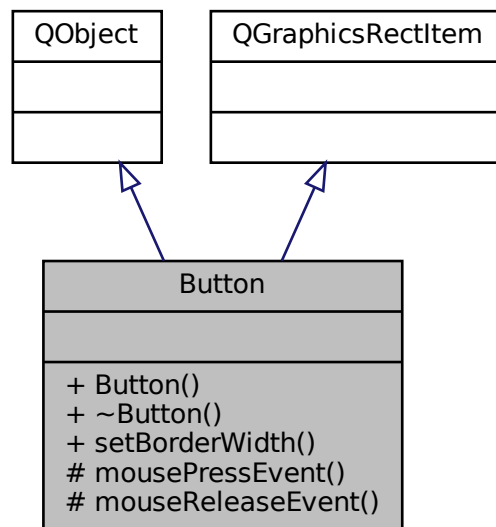
[Button](#) class, inherits from [QObject](#) and [QGraphicsRectItem](#).

```
#include <button.h>
```

Inheritance diagram for Button:



Collaboration diagram for Button:



Public Slots

- void [setBorderWidth](#) (int width)
Set the border width of the button.

Signals

- void [pressed](#) ()
Signal emitted when the button is pressed.
- void [released](#) ()
Signal emitted when the button is released.
- void [autoPilot](#) ()
Signal emitted when the button is released and the autopilot rectangle contains the mouse position.

Public Member Functions

- [Button](#) (const QPixmap &pixmap, std::string name, QGraphicsItem *parent=0, qreal x=0, qreal y=0)
Button constructor.
- [~Button](#) ()=default
Button destructor.

Protected Member Functions

- void [mousePressEvent](#) (QGraphicsSceneMouseEvent *event) override
Mouse press event.
- void [mouseReleaseEvent](#) (QGraphicsSceneMouseEvent *event) override
Mouse release event.

6.1.1 Detailed Description

[Button](#) class, inherits from [QObject](#) and [QGraphicsRectItem](#).

6.1.2 Constructor & Destructor Documentation

6.1.2.1 Button()

```
Button::Button (
    const QPixmap & pixmap,
    std::string name,
    QGraphicsItem * parent = 0,
    qreal x = 0,
    qreal y = 0 ) [explicit]
```

[Button](#) constructor.

Parameters

<i>pixmap</i>	QPixmap object
<i>name</i>	Name of the button
<i>parent</i>	Parent QGraphicsItem
<i>x</i>	X coordinate
<i>y</i>	Y coordinate

6.1.2.2 ~Button()

```
Button::~~Button ( ) [default]
```

[Button](#) destructor.

6.1.3 Member Function Documentation

6.1.3.1 autoPilot

```
void Button::autoPilot ( ) [signal]
```

Signal emitted when the button is released and the autopilot rectangle contains the mouse position.

6.1.3.2 mousePressEvent()

```
void Button::mousePressEvent (  
    QGraphicsSceneMouseEvent * event )    [override], [protected]
```

Mouse press event.

Parameters

<i>event</i>	QGraphicsSceneMouseEvent object
--------------	---------------------------------

6.1.3.3 mouseReleaseEvent()

```
void Button::mouseReleaseEvent (
    QGraphicsSceneMouseEvent * event ) [override], [protected]
```

Mouse release event.

Parameters

<i>event</i>	QGraphicsSceneMouseEvent object
--------------	---------------------------------

6.1.3.4 pressed

```
void Button::pressed ( ) [signal]
```

Signal emitted when the button is pressed.

6.1.3.5 released

```
void Button::released ( ) [signal]
```

Signal emitted when the button is released.

6.1.3.6 setBorderWidth

```
void Button::setBorderWidth (
    int width ) [slot]
```

Set the border width of the button.

Parameters

<i>width</i>	Width of the border
--------------	---------------------

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

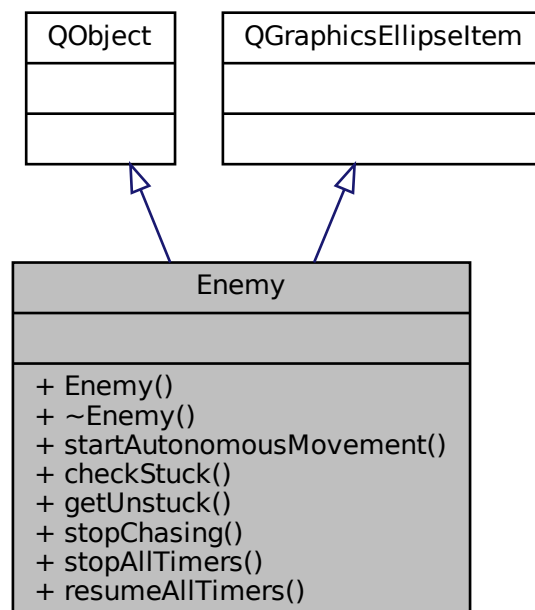
- [button.h](#)
- [button.cpp](#)

6.2 Enemy Class Reference

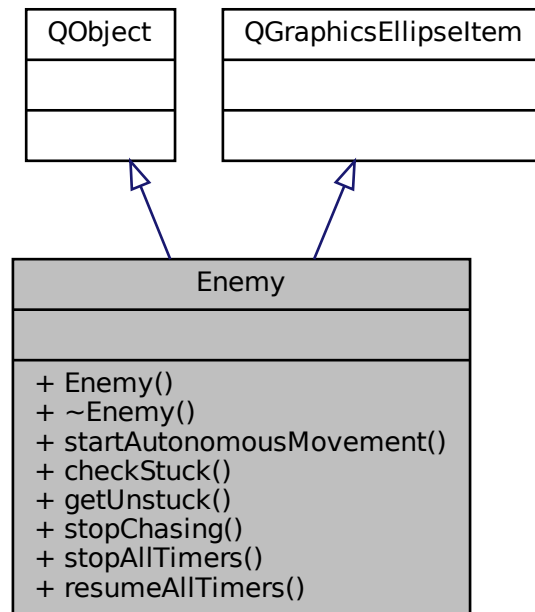
[Enemy](#) class, inherits from [QObject](#) and [QGraphicsEllipseItem](#).

```
#include <enemy.h>
```

Inheritance diagram for Enemy:



Collaboration diagram for Enemy:



Public Slots

- void [startAutonomousMovement](#) ()
Start the autonomous movement of the enemy.
- void [checkStuck](#) ()
After 5s check whether the last position equals (in a set range) the current position.
- void [getUnstuck](#) ()
Get the enemy unstuck (rotation and moving out of map boundaries)
- void [stopChasing](#) ()
Stop chasing the user.
- void [stopAllTimers](#) ()
Stop all the timers of the enemy, used for pausing the game.
- void [resumeAllTimers](#) ()
Resume all the timers of the enemy, used for resuming the game.

Signals

- void [hit](#) ()
Signal emitted when the enemy collides with the user.

Public Member Functions

- [Enemy](#) (QGraphicsItem *parent=0, [User](#) *user=nullptr, int distance=3, int orientation=0, int velocity=3, int rotationAngle=30)
[Enemy](#) constructor.
- [~Enemy](#) ()
[Enemy](#) destructor.

6.2.1 Detailed Description

[Enemy](#) class, inherits from [QObject](#) and [QGraphicsEllipseItem](#).

6.2.2 Constructor & Destructor Documentation

6.2.2.1 Enemy()

```
Enemy::Enemy (
    QGraphicsItem * parent = 0,
    User * user = nullptr,
    int distance = 3,
    int orientation = 0,
    int velocity = 3,
    int rotationAngle = 30 )
```

[Enemy](#) constructor.

Parameters

<i>parent</i>	Parent QGraphicsItem
<i>user</i>	User object
<i>distance</i>	Distance from the obstacles to turn
<i>orientation</i>	Orientation of the enemy
<i>velocity</i>	Velocity of the enemy
<i>rotationAngle</i>	Rotation angle of the enemy (default values shall never be needed)

6.2.2.2 ~Enemy()

```
Enemy::~Enemy ( )
```

[Enemy](#) destructor.

6.2.3 Member Function Documentation

6.2.3.1 checkStuck

```
void Enemy::checkStuck ( ) [slot]
```

After 5s check whether the last position equals (in a set range) the current position.

6.2.3.2 getUnstuck

```
void Enemy::getUnstuck ( ) [slot]
```

Get the enemy unstuck (rotation and moving out of map boundaries)

6.2.3.3 hit

```
void Enemy::hit ( ) [signal]
```

Signal emitted when the enemy collides with the user.

6.2.3.4 resumeAllTimers

```
void Enemy::resumeAllTimers ( ) [slot]
```

Resume all the timers of the enemy, used for resuming the game.

6.2.3.5 startAutonomousMovement

```
void Enemy::startAutonomousMovement ( ) [slot]
```

Start the autonomous movement of the enemy.

6.2.3.6 stopAllTimers

```
void Enemy::stopAllTimers ( ) [slot]
```

Stop all the timers of the enemy, used for pausing the game.

6.2.3.7 stopChasing

```
void Enemy::stopChasing ( ) [slot]
```

Stop chasing the user.

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

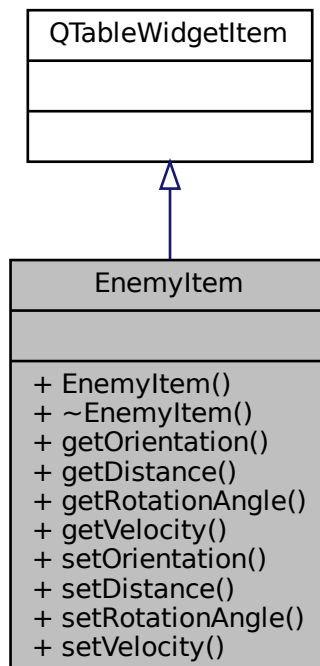
- [enemy.h](#)
- [enemy.cpp](#)

6.3 EnemyItem Class Reference

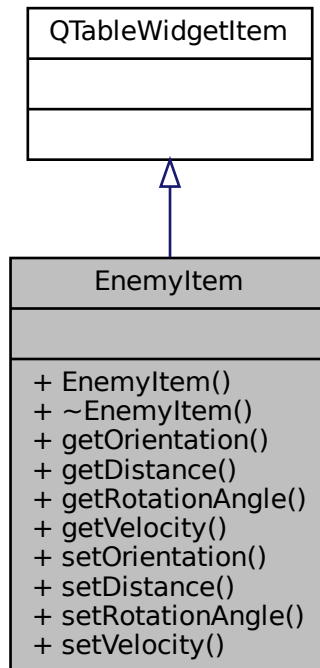
The [EnemyItem](#) class.

```
#include <enemyitem.h>
```

Inheritance diagram for EnemyItem:



Collaboration diagram for EnemyItem:



Public Member Functions

- [EnemyItem](#) (int orientation=0, int distance=0, int rotationAngle=0, int velocity=0)
- [~EnemyItem](#) ()
- int [getOrientation](#) ()
Gets the orientation of the enemy robot.
- int [getDistance](#) ()
Gets the distance of the enemy robot.
- int [getRotationAngle](#) ()
Gets the rotation angle of the enemy robot.
- int [getVelocity](#) ()
Gets the velocity of the enemy robot.
- void [setOrientation](#) (int orientation)
Sets the orientation of the enemy robot.
- void [setDistance](#) (int distance)
Sets the distance of the enemy robot.
- void [setRotationAngle](#) (int rotationAngle)
Sets the rotation angle of the enemy robot.
- void [setVelocity](#) (int velocity)
Sets the velocity of the enemy robot.

6.3.1 Detailed Description

The [EnemyItem](#) class.

This class is used to store the data of the enemy robot in the QWidgetTable

6.3.2 Constructor & Destructor Documentation

6.3.2.1 EnemyItem()

```
EnemyItem::EnemyItem (
    int orientation = 0,
    int distance = 0,
    int rotationAngle = 0,
    int velocity = 0 ) [inline]
```

6.3.2.2 ~EnemyItem()

```
EnemyItem::~EnemyItem ( )
```

6.3.3 Member Function Documentation

6.3.3.1 getDistance()

```
int EnemyItem::getDistance ( )
```

Gets the distance of the enemy robot.

Returns

int distance

6.3.3.2 getOrientation()

```
int EnemyItem::getOrientation ( )
```

Gets the orientation of the enemy robot.

Returns

int orientation

6.3.3.3 getRotationAngle()

```
int EnemyItem::getRotationAngle ( )
```

Gets the rotation angle of the enemy robot.

Returns

int angle

6.3.3.4 getVelocity()

```
int EnemyItem::getVelocity ( )
```

Gets the velocity of the enemy robot.

Returns

int velocity

6.3.3.5 setDistance()

```
void EnemyItem::setDistance (
    int distance )
```

Sets the distance of the enemy robot.

Parameters

<i>distance</i>	distance of the enemy robot
-----------------	-----------------------------

6.3.3.6 setOrientation()

```
void EnemyItem::setOrientation (
    int orientation )
```

Sets the orientation of the enemy robot.

Parameters

<i>orientation</i>	orientation of the enemy robot
--------------------	--------------------------------

6.3.3.7 setRotationAngle()

```
void EnemyItem::setRotationAngle (
    int rotationAngle )
```

Sets the rotation angle of the enemy robot.

Parameters

<i>rotationAngle</i>	rotation angle of the enemy robot
----------------------	-----------------------------------

6.3.3.8 setVelocity()

```
void EnemyItem::setVelocity (
    int velocity )
```

Sets the velocity of the enemy robot.

Parameters

<i>velocity</i>	velocity of the enemy robot
-----------------	-----------------------------

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

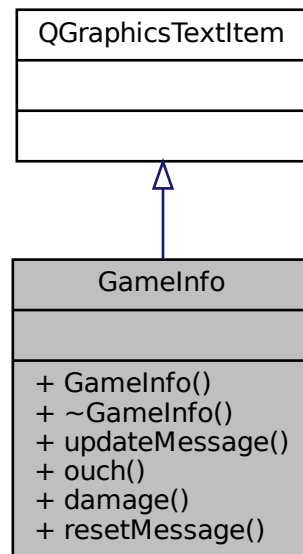
- [enemyitem.h](#)
- [enemyitem.cpp](#)

6.4 GameInfo Class Reference

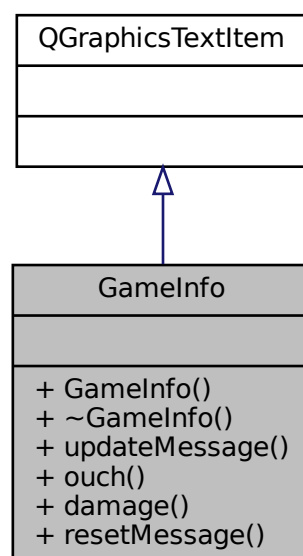
[GameInfo](#) class, inherits from [QObject](#) and [QGraphicsTextItem](#).

```
#include <gameinfo.h>
```

Inheritance diagram for GameInfo:



Collaboration diagram for GameInfo:



Public Slots

- void [ouch](#) ()
Display the "OUCH!" message.
- void [damage](#) ()
*Display the "SH*T!" message.*
- void [resetMessage](#) ()
Reset the message to the default text.

Public Member Functions

- [GameInfo](#) (QGraphicsItem *parent=0)
GameInfo constructor.
- [~GameInfo](#) ()
GameInfo destructor.
- void [updateMessage](#) (const QString &message, const QColor &color, int duration)
Update the message in the text box.

6.4.1 Detailed Description

[GameInfo](#) class, inherits from [QObject](#) and [QGraphicsTextItem](#).

6.4.2 Constructor & Destructor Documentation

6.4.2.1 GameInfo()

```
GameInfo::GameInfo (
    QGraphicsItem * parent = 0 )
```

[GameInfo](#) constructor.

Parameters

<i>parent</i>	Parent QGraphicsItem
---------------	----------------------

6.4.2.2 ~GameInfo()

```
GameInfo::~GameInfo ( )
```

[GameInfo](#) destructor.

6.4.3 Member Function Documentation

6.4.3.1 damage

```
void GameInfo::damage ( ) [slot]
```

Display the "SH*T!" message.

6.4.3.2 ouch

```
void GameInfo::ouch ( ) [slot]
```

Display the "OUCH!" message.

6.4.3.3 resetMessage

```
void GameInfo::resetMessage ( ) [slot]
```

Reset the message to the default text.

6.4.3.4 updateMessage()

```
void GameInfo::updateMessage (
    const QString & message,
    const QColor & color,
    int duration )
```

Update the message in the text box.

Parameters

<i>message</i>	Message to be displayed
<i>color</i>	Color of the message
<i>duration</i>	Duration of the message

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

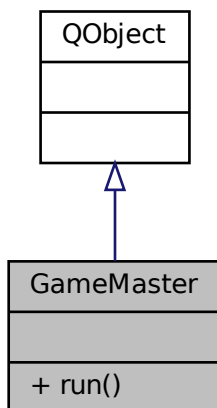
- [gameinfo.h](#)
- [gameinfo.cpp](#)

6.5 GameMaster Class Reference

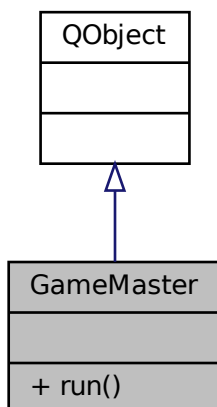
[GameMaster](#) class, inherits from [QObject](#).

```
#include <gamemaster.h>
```

Inheritance diagram for GameMaster:



Collaboration diagram for GameMaster:



Public Member Functions

- void [run](#) ()
Run the game - initialize the main window.

6.5.1 Detailed Description

[GameMaster](#) class, inherits from [QObject](#).

6.5.2 Member Function Documentation

6.5.2.1 run()

```
void GameMaster::run ( )
```

Run the game - initialize the main window.

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

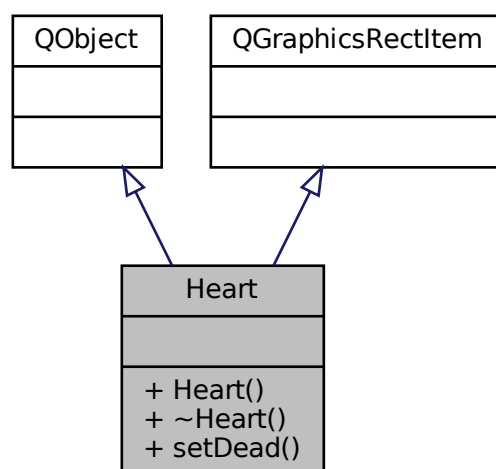
- [gamemaster.h](#)
- [gamemaster.cpp](#)

6.6 Heart Class Reference

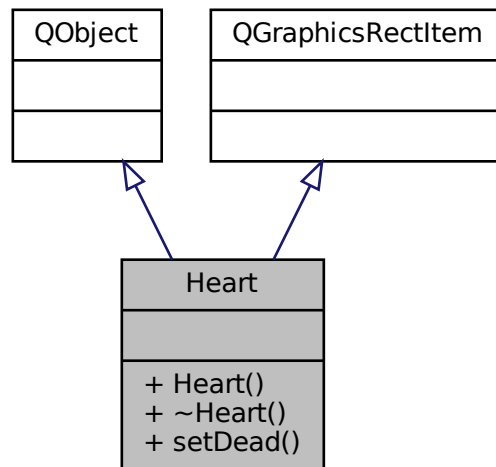
[Heart](#) class, inherits from [QObject](#) and [QGraphicsRectItem](#).

```
#include <heart.h>
```

Inheritance diagram for Heart:



Collaboration diagram for Heart:



Public Slots

- void [setDead](#) ()
Sets [Heart](#) as dead - turns it gray.

Public Member Functions

- [Heart](#) (QGraphicsItem *parent=0)
[Heart](#) constructor.
- [~Heart](#) ()=default
[Heart](#) destructor.

6.6.1 Detailed Description

[Heart](#) class, inherits from [QObject](#) and [QGraphicsRectItem](#).

6.6.2 Constructor & Destructor Documentation

6.6.2.1 Heart()

```
Heart::Heart (
    QGraphicsItem * parent = 0 ) [explicit]
```

[Heart](#) constructor.

Parameters

<i>parent</i>	Parent QGraphicsItem
---------------	----------------------

6.6.2.2 ~Heart()

Heart::~~Heart () [default]

[Heart](#) destructor.

6.6.3 Member Function Documentation

6.6.3.1 setDead

void Heart::setDead () [slot]

Sets [Heart](#) as dead - turns it gray.

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

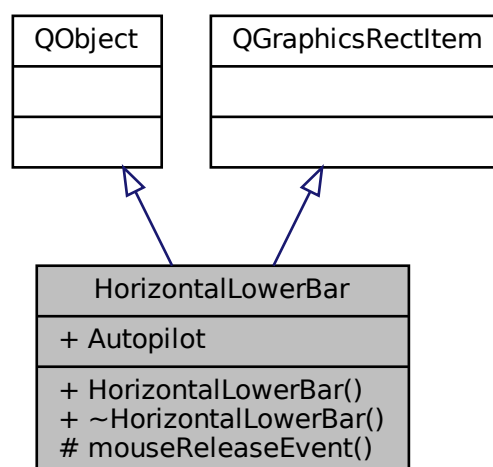
- [heart.h](#)
- [heart.cpp](#)

6.7 HorizontalLowerBar Class Reference

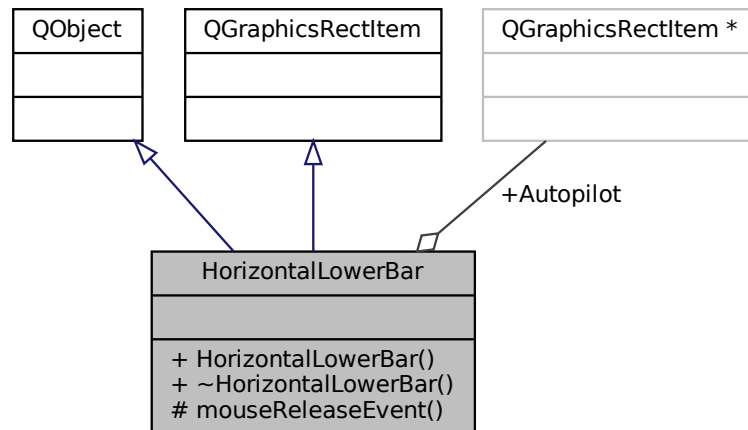
[HorizontalLowerBar](#) class, inherits from [QObject](#) and [QGraphicsRectItem](#).

```
#include <horizontalbar.h>
```

Inheritance diagram for HorizontalLowerBar:



Collaboration diagram for `HorizontalLowerBar`:



Public Member Functions

- [HorizontalLowerBar](#) ([User](#) *user)
HorizontalLowerBar constructor.
- [~HorizontalLowerBar](#) ()
HorizontalLowerBar destructor.

Public Attributes

- [QGraphicsRectItem](#) * [Autopilot](#)

Protected Member Functions

- void [mouseReleaseEvent](#) ([QGraphicsSceneMouseEvent](#) *event) override
Mouse release event.

6.7.1 Detailed Description

[HorizontalLowerBar](#) class, inherits from [QObject](#) and [QGraphicsRectItem](#).

6.7.2 Constructor & Destructor Documentation

6.7.2.1 HorizontalLowerBar()

```
HorizontalLowerBar::HorizontalLowerBar (
    User * user )
```

[HorizontalLowerBar](#) constructor.

Parameters

<i>user</i>	User object
-------------	-----------------------------

6.7.2.2 ~HorizontalLowerBar()

```
HorizontalLowerBar::~~HorizontalLowerBar ( )
```

[HorizontalLowerBar](#) destructor.

6.7.3 Member Function Documentation

6.7.3.1 mouseReleaseEvent()

```
void HorizontalLowerBar::mouseReleaseEvent (
    QGraphicsSceneMouseEvent * event ) [override], [protected]
```

Mouse release event.

Parameters

<i>event</i>	QGraphicsSceneMouseEvent
--------------	--------------------------

6.7.4 Member Data Documentation

6.7.4.1 Autopilot

```
QGraphicsRectItem* HorizontalLowerBar::Autopilot
```

Autopilot rectangle

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

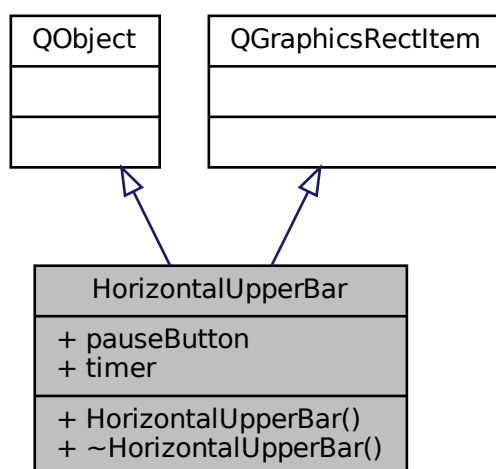
- [horizontalbar.h](#)
- [horizontalbar.cpp](#)

6.8 HorizontalUpperBar Class Reference

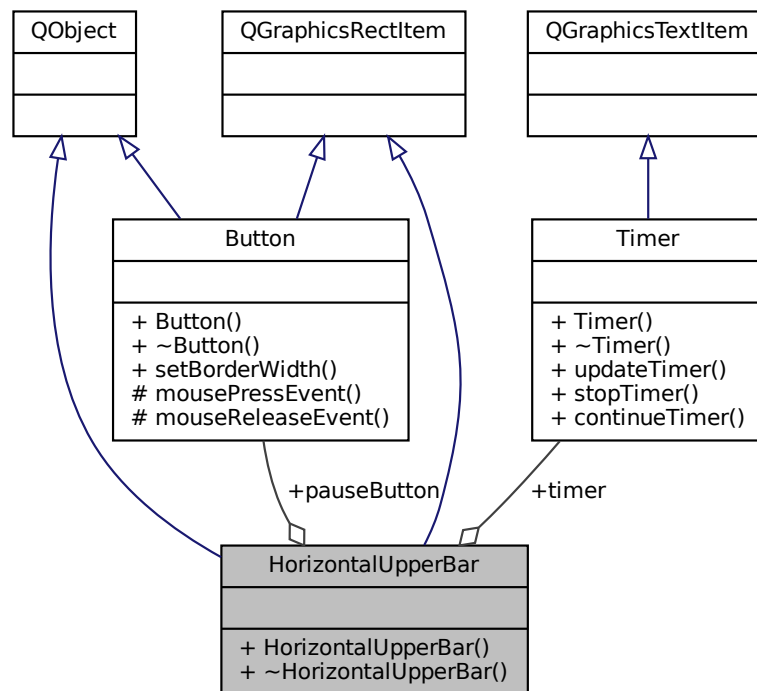
[HorizontalUpperBar](#) class, inherits from [QObject](#) and [QGraphicsRectItem](#).

```
#include <horizontalbar.h>
```

Inheritance diagram for HorizontalUpperBar:



Collaboration diagram for HorizontalUpperBar:



Public Member Functions

- [HorizontalUpperBar](#) ([User](#) *user, int timeLimit)
HorizontalUpperBar constructor.
- [~HorizontalUpperBar](#) ()
HorizontalUpperBar destructor.

Public Attributes

- [Button](#) * `pauseButton`
- [Timer](#) * `timer`

6.8.1 Detailed Description

[HorizontalUpperBar](#) class, inherits from [QObject](#) and [QGraphicsRectItem](#).

6.8.2 Constructor & Destructor Documentation

6.8.2.1 HorizontalUpperBar()

```
HorizontalUpperBar::HorizontalUpperBar (
    User * user,
    int timeLimit )
```

[HorizontalUpperBar](#) constructor.

Parameters

<i>user</i>	User object
<i>timeLimit</i>	Time limit for the game

6.8.2.2 ~HorizontalUpperBar()

```
HorizontalUpperBar::~~HorizontalUpperBar ( )
```

[HorizontalUpperBar](#) destructor.

6.8.3 Member Data Documentation

6.8.3.1 pauseButton

```
Button* HorizontalUpperBar::pauseButton
```

Pause button

6.8.3.2 timer

```
Timer* HorizontalUpperBar::timer
```

[Timer](#)

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

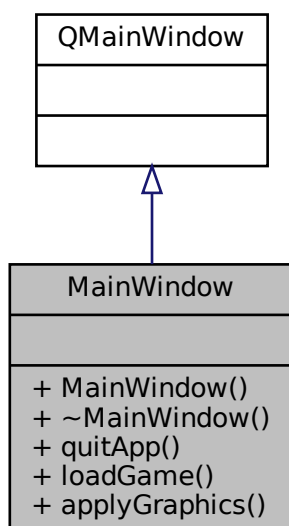
- [horizontalbar.h](#)
- [horizontalbar.cpp](#)

6.9 MainWindow Class Reference

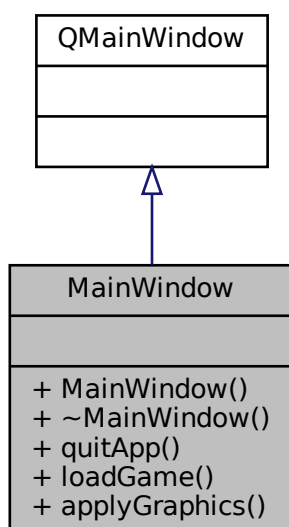
The [MainWindow](#) class.

```
#include <mainwindow.h>
```

Inheritance diagram for MainWindow:



Collaboration diagram for MainWindow:



Signals

- void [createNewMapWindow](#) ()
Signal to create a new map window.
- void [loadFile](#) ()
Signal to start loading a JSON file.

Public Member Functions

- [MainWindow](#) (QWidget *parent=nullptr)
- [~MainWindow](#) ()
- void [quitApp](#) ()
Quits the app when the exit button is pressed.
- void [loadGame](#) ()
Starts loading game from JSON file when the load button is pressed.
- void [applyGraphics](#) ()
Applies background and font to the main window.

6.9.1 Detailed Description

The [MainWindow](#) class.

This class is used to hold the main window of the application

6.9.2 Constructor & Destructor Documentation

6.9.2.1 MainWindow()

```
MainWindow::MainWindow (  
    QWidget * parent = nullptr )
```

6.9.2.2 ~MainWindow()

```
MainWindow::~MainWindow ( )
```

6.9.3 Member Function Documentation

6.9.3.1 applyGraphics()

```
void MainWindow::applyGraphics ( )
```

Applies background and font to the main window.

6.9.3.2 createNewMapWindow

```
void MainWindow::createNewMapWindow ( ) [signal]
```

Signal to create a new map window.

6.9.3.3 loadFile

```
void MainWindow::loadFile ( ) [signal]
```

Signal to start loading a JSON file.

6.9.3.4 loadGame()

```
void MainWindow::loadGame ( )
```

Starts loading game from JSON file when the load button is pressed.

6.9.3.5 quitApp()

```
void MainWindow::quitApp ( )
```

Quits the app when the exit button is pressed.

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

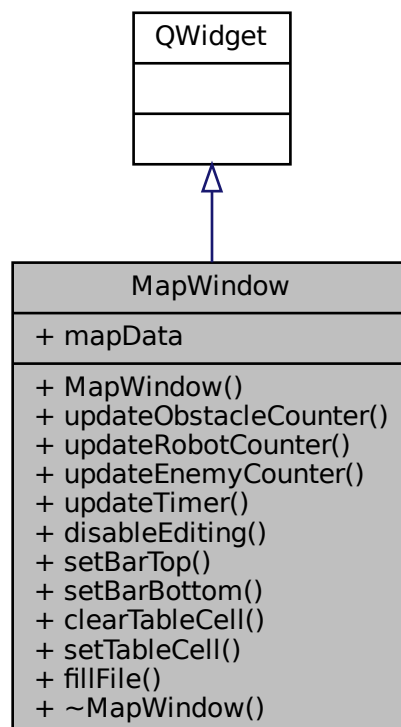
- [mainwindow.h](#)
- [mainwindow.cpp](#)

6.10 MapWindow Class Reference

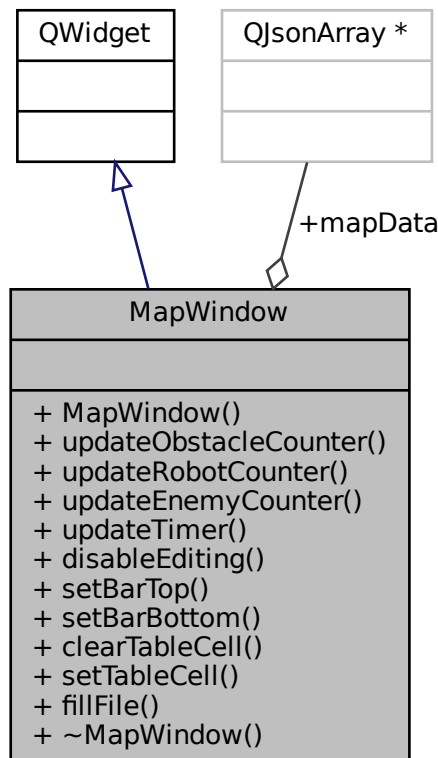
Class for the Map Window.

```
#include <mapwindow.h>
```

Inheritance diagram for MapWindow:



Collaboration diagram for MapWindow:



Signals

- void [startSession](#) ()
Signal to start the game.

Public Member Functions

- [MapWindow](#) ([QWidget](#) *parent=nullptr)
- void [updateObstacleCounter](#) ()
Updates the number of obstacles to be placed.
- void [updateRobotCounter](#) ()
Updates the number of robots to be placed.
- void [updateEnemyCounter](#) ()
Updates the number of enemy robots to be placed.
- void [updateTimer](#) ()
Updates the time limit for the game.
- void [disableEditing](#) ()
Disables editing of the map cells.
- void [setBarTop](#) ()

- Sets up top bar of the map editor.*
 - void [setBarBottom](#) ()
- Sets up bottom bar of the map editor.*
 - void [clearTableCell](#) ([QTableWidgetItem](#) *item, int row, int column)
- Clears the table cell.*
 - void [setTableCell](#) ([QTableWidgetItem](#) *item, int row, int column, int orientationAngle)
- Sets the table cell.*
 - [QJsonArray](#) * [fillFile](#) (bool *robotFound)
- Fills the JSON array with the map data.*
 - [~MapWindow](#) ()

Public Attributes

- [QJsonArray](#) * [mapData](#)

6.10.1 Detailed Description

Class for the Map Window.

6.10.2 Constructor & Destructor Documentation

6.10.2.1 MapWindow()

```
MapWindow::MapWindow (
    QWidget * parent = nullptr ) [explicit]
```

6.10.2.2 ~MapWindow()

```
MapWindow::~MapWindow ( )
```

6.10.3 Member Function Documentation

6.10.3.1 clearTableCell()

```
void MapWindow::clearTableCell (
    QTableWidgetItem * item,
    int row,
    int column )
```

Clears the table cell.

Parameters

<i>item</i>	pointer to the QTableWidgetItem to be cleared
<i>row</i>	number of the row
<i>column</i>	number of the column

6.10.3.2 disableEditing()

```
void MapWindow::disableEditing ( )
```

Disables editing of the map cells.

6.10.3.3 fillFile()

```
QJsonArray * MapWindow::fillFile (
    bool * robotFound )
```

Fills the JSON array with the map data.

Parameters

<i>robotFound</i>	pointer to the boolean variable to check if robot is found
-------------------	--

Returns

QJsonArray* pointer to the array with map data

6.10.3.4 setBarBottom()

```
void MapWindow::setBarBottom ( )
```

Sets up bottom bar of the map editor.

6.10.3.5 setBarTop()

```
void MapWindow::setBarTop ( )
```

Sets up top bar of the map editor.

6.10.3.6 setTableCell()

```
void MapWindow::setTableCell (
    QTableWidgetItem * item,
    int row,
    int column,
    int orientationAngle )
```

Sets the table cell.

Parameters

<i>item</i>	pointer to the QTableWidgetItem to be set
<i>row</i>	number of the row
<i>column</i>	number of the column
<i>orientationAngle</i>	angle of orientation in case of robots

6.10.3.7 startSession

```
void MapWindow::startSession ( ) [signal]
```

Signal to start the game.

6.10.3.8 updateEnemyCounter()

```
void MapWindow::updateEnemyCounter ( )
```

Updates the number of enemy robots to be placed.

6.10.3.9 updateObstacleCounter()

```
void MapWindow::updateObstacleCounter ( )
```

Updates the number of obstacles to be placed.

6.10.3.10 updateRobotCounter()

```
void MapWindow::updateRobotCounter ( )
```

Updates the number of robots to be placed.

6.10.3.11 updateTimer()

```
void MapWindow::updateTimer ( )
```

Updates the time limit for the game.

6.10.4 Member Data Documentation

6.10.4.1 mapData

```
QJsonArray* MapWindow::mapData
```

Json array for storing mapa of the map

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

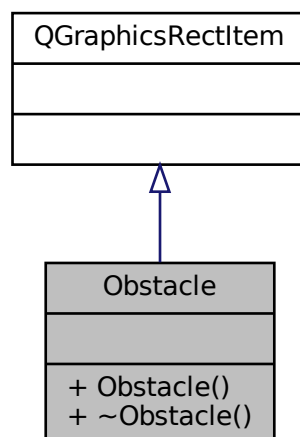
- [mapwindow.h](#)
- [mapwindow.cpp](#)

6.11 Obstacle Class Reference

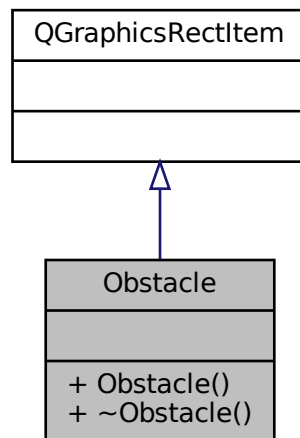
[Obstacle](#) class, inherits from [QGraphicsRectItem](#).

```
#include <obstacle.h>
```

Inheritance diagram for Obstacle:



Collaboration diagram for `Obstacle`:



Public Member Functions

- [Obstacle](#) (`QGraphicsItem *parent=0`)
[Obstacle](#) constructor.
- [~Obstacle](#) ()=default
[Obstacle](#) destructor.

6.11.1 Detailed Description

[Obstacle](#) class, inherits from [QGraphicsRectItem](#).

6.11.2 Constructor & Destructor Documentation

6.11.2.1 Obstacle()

```
Obstacle::Obstacle (
    QGraphicsItem * parent = 0 )
```

[Obstacle](#) constructor.

Parameters

<i>parent</i>	Parent <code>QGraphicsItem</code>
---------------	-----------------------------------

6.11.2.2 ~Obstacle()

```
Obstacle::~Obstacle ( ) [default]
```

[Obstacle](#) destructor.

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

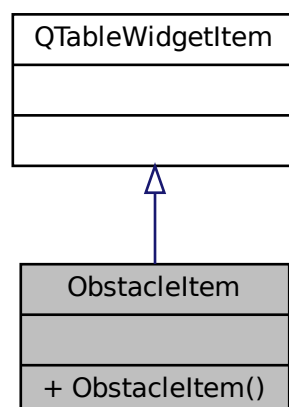
- [obstacle.h](#)
- [obstacle.cpp](#)

6.12 ObstacleItem Class Reference

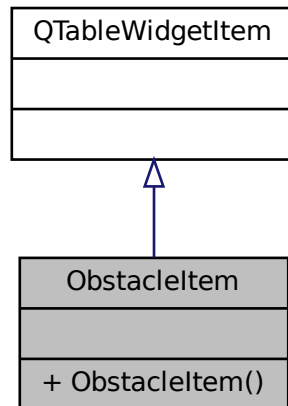
The [ObstacleItem](#) class.

```
#include <obstacleitem.h>
```

Inheritance diagram for ObstacleItem:



Collaboration diagram for `ObstacleItem`:



Public Member Functions

- [ObstacleItem](#) ()

6.12.1 Detailed Description

The [ObstacleItem](#) class.

This class is used to store the obstacle in the `QWidgetTable`

6.12.2 Constructor & Destructor Documentation

6.12.2.1 ObstacleItem()

```
ObstacleItem::ObstacleItem ( ) [inline]
```

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

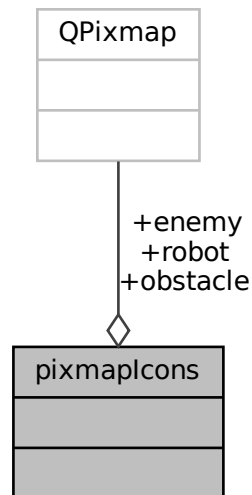
- [obstacleitem.h](#)

6.13 QPixmapIcons Struct Reference

Struct to store the icons for the pixmap.

```
#include <mapwindow.h>
```

Collaboration diagram for QPixmapIcons:



Public Attributes

- QPixmap [obstacle](#)
- QPixmap [robot](#)
- QPixmap [enemy](#)

6.13.1 Detailed Description

Struct to store the icons for the pixmap.

6.13.2 Member Data Documentation

6.13.2.1 enemy

```
QPixmap pixmapIcons::enemy
```

Icon of the enemy

6.13.2.2 obstacle

`QPixmap pixmapIcons::obstacle`

Icon of the obstacle

6.13.2.3 robot

`QPixmap pixmapIcons::robot`

Icon of the robot

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

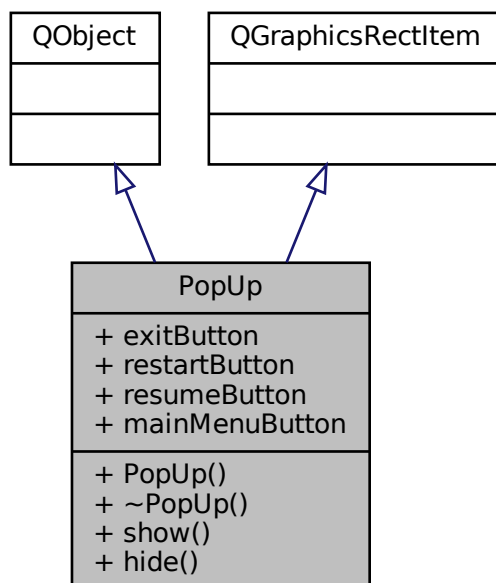
- [mapwindow.h](#)

6.14 PopUp Class Reference

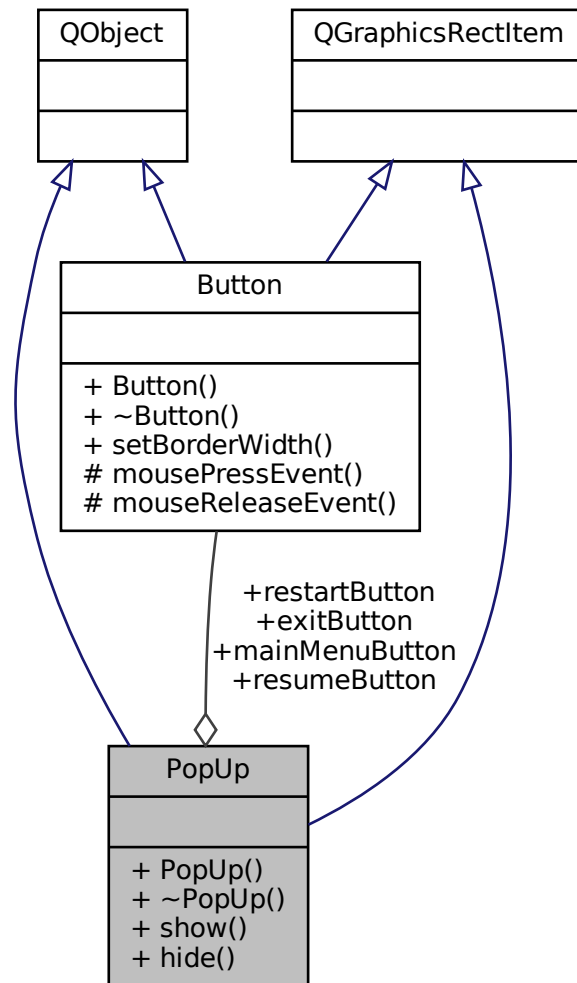
Popup class, inherits from [QObject](#) and [QGraphicsRectItem](#).

```
#include <popup.h>
```

Inheritance diagram for PopUp:



Collaboration diagram for PopUp:



Public Slots

- void `show` ()
Show the popup.
- void `hide` ()
Hide the popup.

Signals

- void `pauseTimers` ()
Pause the timers.

Public Member Functions

- [PopUp](#) (QGraphicsItem *parent=0, std::string name="")
Popup constructor.
- [~PopUp](#) ()
Popup destructor.

Public Attributes

- [Button](#) * [exitButton](#)
- [Button](#) * [restartButton](#)
- [Button](#) * [resumeButton](#)
- [Button](#) * [mainMenuButton](#)

6.14.1 Detailed Description

Popup class, inherits from [QObject](#) and [QGraphicsRectItem](#).

6.14.2 Constructor & Destructor Documentation

6.14.2.1 PopUp()

```
PopUp::PopUp (
    QGraphicsItem * parent = 0,
    std::string name = "" )
```

Popup constructor.

Parameters

<i>parent</i>	Parent QGraphicsItem
<i>name</i>	Name of the popup

6.14.2.2 ~PopUp()

```
PopUp::~~PopUp ( )
```

Popup destructor.

6.14.3 Member Function Documentation

6.14.3.1 hide

```
void PopUp::hide ( ) [slot]
```

Hide the popup.

6.14.3.2 pauseTimers

```
void PopUp::pauseTimers ( ) [signal]
```

Pause the timers.

6.14.3.3 show

```
void PopUp::show ( ) [slot]
```

Show the popup.

6.14.4 Member Data Documentation

6.14.4.1 exitButton

```
Button* PopUp::exitButton
```

Exit button

6.14.4.2 mainMenuButton

```
Button* PopUp::mainMenuButton
```

Main menu button

6.14.4.3 restartButton

```
Button* PopUp::restartButton
```

Restart button

6.14.4.4 resumeButton

```
Button* PopUp::resumeButton
```

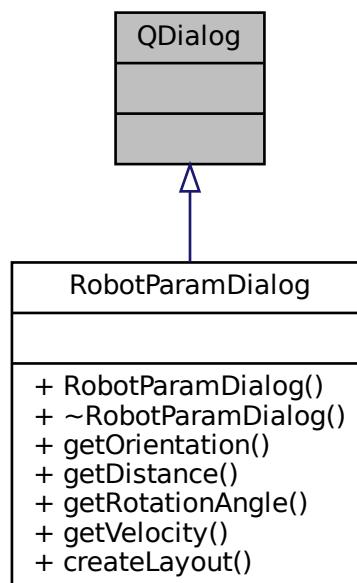
Resume button (in victory/ gameover screen does nothing)

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

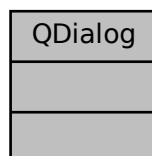
- [popup.h](#)
- [popup.cpp](#)

6.15 QDialog Class Reference

Inheritance diagram for QDialog:



Collaboration diagram for QDialog:

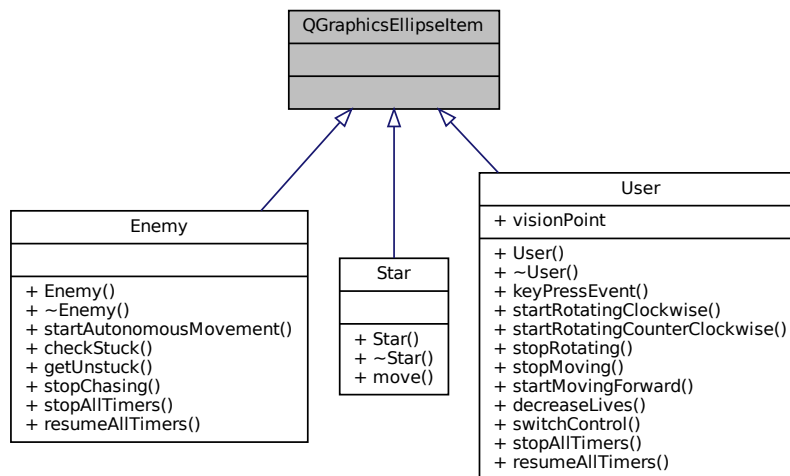


The documentation for this class was generated from the following file:

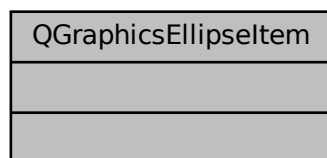
- [robotdialog.h](#)

6.16 QGraphicsEllipseItem Class Reference

Inheritance diagram for QGraphicsEllipseItem:



Collaboration diagram for QGraphicsEllipseItem:

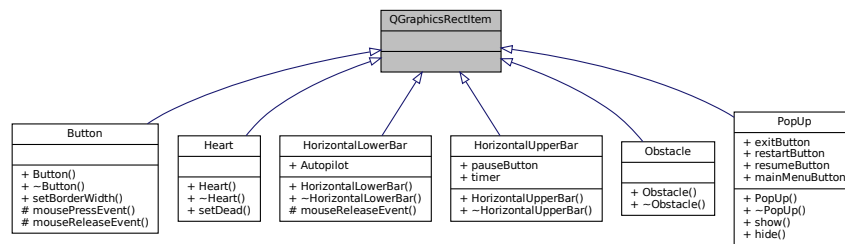


The documentation for this class was generated from the following file:

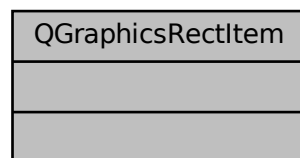
- [user.h](#)

6.17 QGraphicsRectItem Class Reference

Inheritance diagram for QGraphicsRectItem:



Collaboration diagram for QGraphicsRectItem:

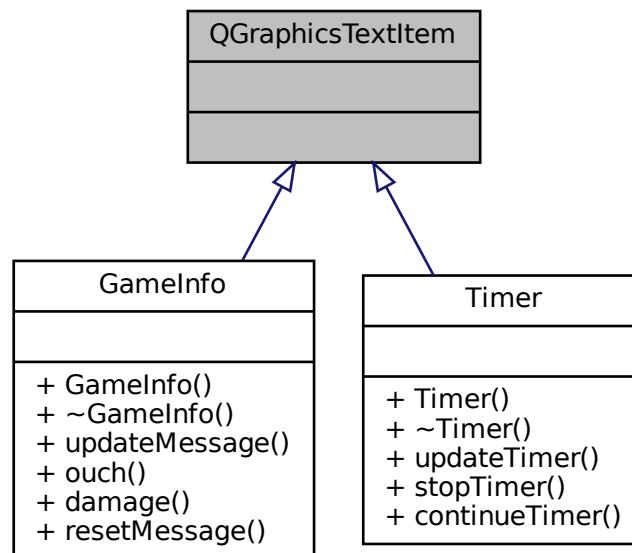


The documentation for this class was generated from the following file:

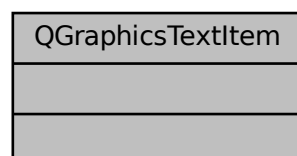
- [popup.h](#)

6.18 QGraphicsTextItem Class Reference

Inheritance diagram for QGraphicsTextItem:



Collaboration diagram for QGraphicsTextItem:

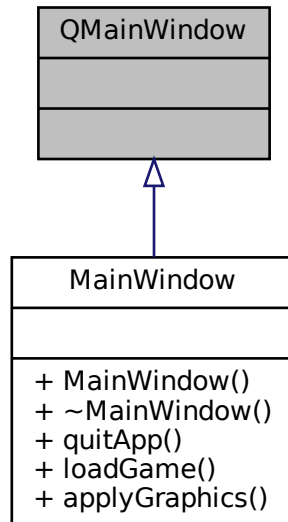


The documentation for this class was generated from the following file:

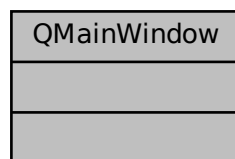
- [timer.h](#)

6.19 QMainWindow Class Reference

Inheritance diagram for QMainWindow:



Collaboration diagram for QMainWindow:

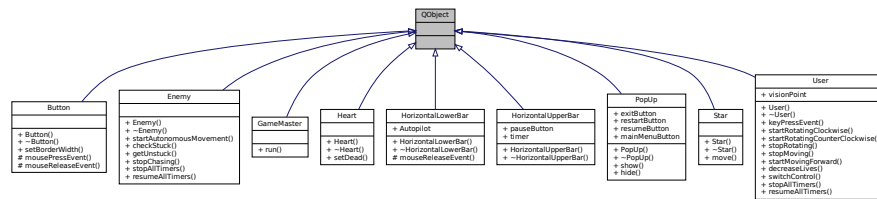


The documentation for this class was generated from the following file:

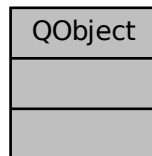
- [mainwindow.h](#)

6.20 QObject Class Reference

Inheritance diagram for QObject:



Collaboration diagram for QObject:

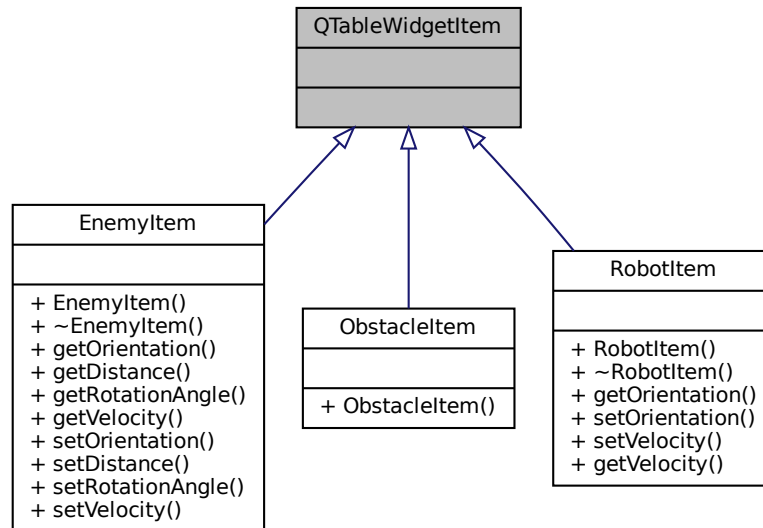


The documentation for this class was generated from the following file:

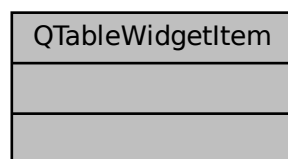
- [user.h](#)

6.21 QTableWidgetItem Class Reference

Inheritance diagram for QTableWidgetItem:



Collaboration diagram for QTableWidgetItem:

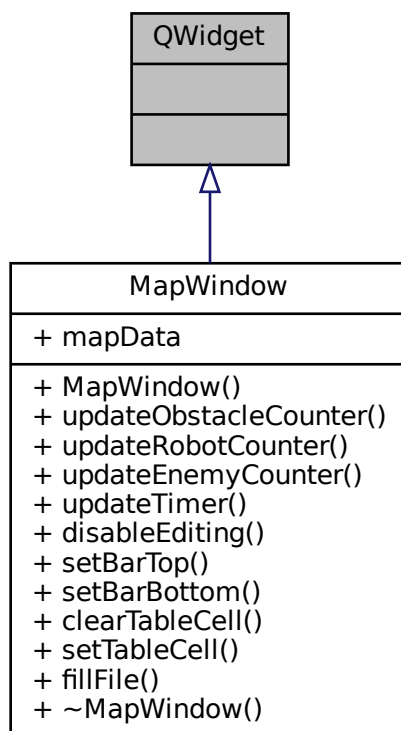


The documentation for this class was generated from the following file:

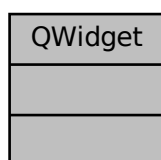
- [robotitem.h](#)

6.22 QWidget Class Reference

Inheritance diagram for QWidget:



Collaboration diagram for QWidget:



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

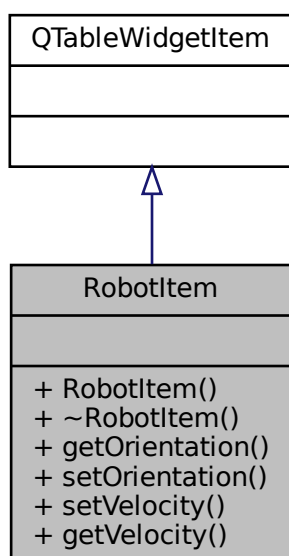
- [mapwindow.h](#)

6.23 RobotItem Class Reference

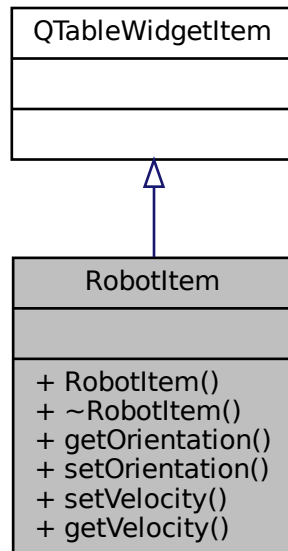
The [RobotItem](#) class.

```
#include <robotitem.h>
```

Inheritance diagram for RobotItem:



Collaboration diagram for RobotItem:



Public Member Functions

- [RobotItem](#) (int orientation=0, int velocity=0)
- [~RobotItem](#) ()
- int [getOrientation](#) ()
Gets the orientation of the robot.
- void [setOrientation](#) (int orientation)
Sets the orientation of the robot.
- void [setVelocity](#) (int velocity)
Sets the velocity of the robot.
- int [getVelocity](#) ()
Gets the velocity of the robot.

6.23.1 Detailed Description

The [RobotItem](#) class.

This class is used to store the data of the robot in the `QWidgetTable`

6.23.2 Constructor & Destructor Documentation

6.23.2.1 RobotItem()

```
RobotItem::RobotItem (
    int orientation = 0,
    int velocity = 0 ) [inline]
```

6.23.2.2 ~RobotItem()

```
RobotItem::~~RobotItem ( )
```

6.23.3 Member Function Documentation

6.23.3.1 getOrientation()

```
int RobotItem::getOrientation ( )
```

Gets the orientation of the robot.

Returns

int orientation

6.23.3.2 getVelocity()

```
int RobotItem::getVelocity ( )
```

Gets the velocity of the robot.

Returns

int velocity

6.23.3.3 setOrientation()

```
void RobotItem::setOrientation (
    int orientation )
```

Sets the orientation of the robot.

Parameters

<i>orientation</i>	orientation of the robot
--------------------	--------------------------

6.23.3.4 setVelocity()

```
void RobotItem::setVelocity (
    int velocity )
```

Sets the velocity of the robot.

Parameters

<i>velocity</i>	velocity of the robot
-----------------	-----------------------

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

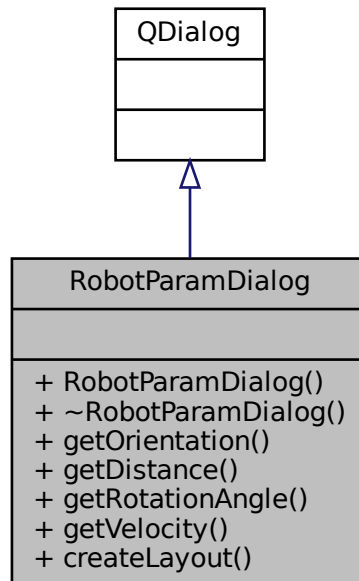
- [robotitem.h](#)
- [robotitem.cpp](#)

6.24 RobotParamDialog Class Reference

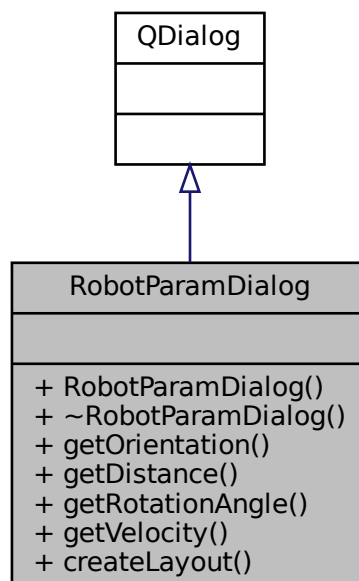
The [RobotParamDialog](#) class.

```
#include <robotdialog.h>
```

Inheritance diagram for RobotParamDialog:



Collaboration diagram for RobotParamDialog:



Public Member Functions

- [RobotParamDialog](#) ([QWidget](#) *parent=nullptr, bool robotExists=false)
- [~RobotParamDialog](#) ()=default
- int [getOrientation](#) ()
Gets the orientation of the controlled or enemy robot.
- int [getDistance](#) ()
Gets the distance of the controlled or enemy robot.
- int [getRotationAngle](#) ()
Gets the rotation angle of the controlled or enemy robot.
- int [getVelocity](#) ()
Gets the velocity of the controlled or enemy robot.
- void [createLayout](#) (bool robotExists)
Creates the layout of the dialog @bool robotExists if robot exists direction button is disabled.

6.24.1 Detailed Description

The [RobotParamDialog](#) class.

This class is used to create a dialog where the user can input the parameters of the robot

6.24.2 Constructor & Destructor Documentation

6.24.2.1 RobotParamDialog()

```
RobotParamDialog::RobotParamDialog (
    QWidget * parent = nullptr,
    bool robotExists = false ) [explicit]
```

6.24.2.2 ~RobotParamDialog()

```
RobotParamDialog::~RobotParamDialog ( ) [default]
```

6.24.3 Member Function Documentation

6.24.3.1 createLayout()

```
void RobotParamDialog::createLayout (
    bool robotExists )
```

Creates the layout of the dialog @bool robotExists if robot exists direction button is disabled.

6.24.3.2 `getDistance()`

```
int RobotParamDialog::getDistance ( )
```

Gets the distance of the controlled or enemy robot.

Returns

int distance

6.24.3.3 `getOrientation()`

```
int RobotParamDialog::getOrientation ( )
```

Gets the orientation of the controlled or enemy robot.

Returns

int orientation

6.24.3.4 `getRotationAngle()`

```
int RobotParamDialog::getRotationAngle ( )
```

Gets the rotation angle of the controlled or enemy robot.

Returns

int angle

6.24.3.5 `getVelocity()`

```
int RobotParamDialog::getVelocity ( )
```

Gets the velocity of the controlled or enemy robot.

Returns

int velocity

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

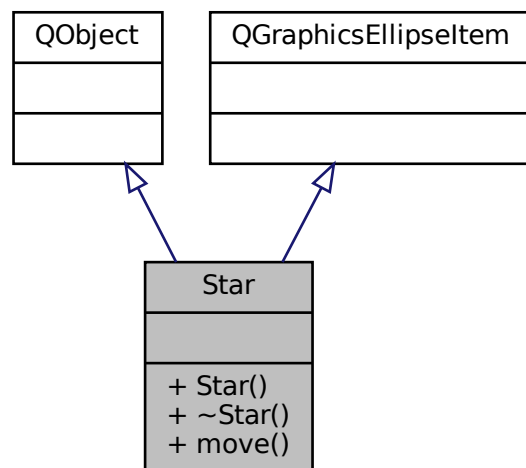
- [robotdialog.h](#)
- [robotdialog.cpp](#)

6.25 Star Class Reference

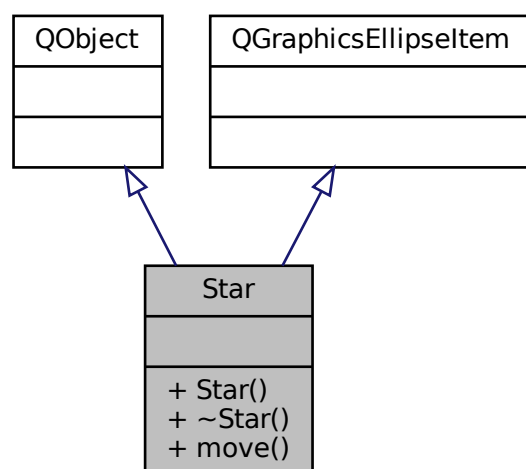
[Star](#) class, inherits from [QObject](#) and [QGraphicsEllipseItem](#).

```
#include <star.h>
```

Inheritance diagram for Star:



Collaboration diagram for Star:



Public Slots

- void [move](#) ()
Move the star.

Public Member Functions

- [Star](#) (QGraphicsItem *parent=0)
Star constructor.
- [~Star](#) ()=default
Star destructor.

6.25.1 Detailed Description

[Star](#) class, inherits from [QObject](#) and [QGraphicsEllipseItem](#).

6.25.2 Constructor & Destructor Documentation

6.25.2.1 [Star](#)()

```
Star::Star (  
    QGraphicsItem * parent = 0 )
```

[Star](#) constructor.

Parameters

<i>parent</i>	Parent QGraphicsItem
---------------	----------------------

6.25.2.2 [~Star](#)()

```
Star::~Star ( ) [default]
```

[Star](#) destructor.

6.25.3 Member Function Documentation

6.25.3.1 move

```
void Star::move ( ) [slot]
```

Move the star.

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

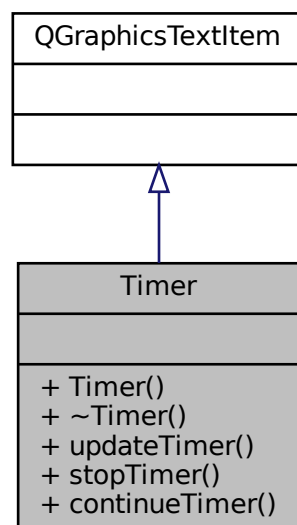
- [star.h](#)
- [star.cpp](#)

6.26 Timer Class Reference

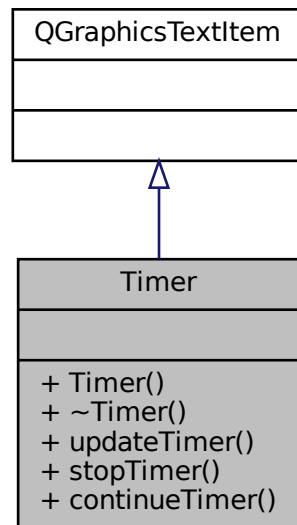
[Timer](#) class, inherits from [QGraphicsTextItem](#).

```
#include <timer.h>
```

Inheritance diagram for Timer:



Collaboration diagram for Timer:



Public Slots

- void [updateTimer](#) ()
Update the timer.
- void [stopTimer](#) ()
Stop the timer.
- void [continueTimer](#) ()
Continue the timer.

Signals

- void [timeIsUp](#) ()
Signal emitted when the time is up, victory popup is shown.

Public Member Functions

- [Timer](#) (QGraphicsItem *parent=0, int timeL=60)
[Timer](#) constructor.
- [~Timer](#) ()
[Timer](#) destructor.

6.26.1 Detailed Description

[Timer](#) class, inherits from [QGraphicsTextItem](#).

6.26.2 Constructor & Destructor Documentation

6.26.2.1 Timer()

```
Timer::Timer (
    QGraphicsItem * parent = 0,
    int timeL = 60 )
```

[Timer](#) constructor.

Parameters

<i>parent</i>	Parent QGraphicsItem
<i>timeL</i>	Time limit of the timer

6.26.2.2 ~Timer()

```
Timer::~~Timer ( )
```

[Timer](#) destructor.

6.26.3 Member Function Documentation

6.26.3.1 continueTimer

```
void Timer::continueTimer ( ) [slot]
```

Continue the timer.

6.26.3.2 stopTimer

```
void Timer::stopTimer ( ) [slot]
```

Stop the timer.

6.26.3.3 timeIsUp

```
void Timer::timeIsUp ( ) [signal]
```

Signal emitted when the time is up, victory popup is shown.

6.26.3.4 updateTimer

```
void Timer::updateTimer ( ) [slot]
```

Update the timer.

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

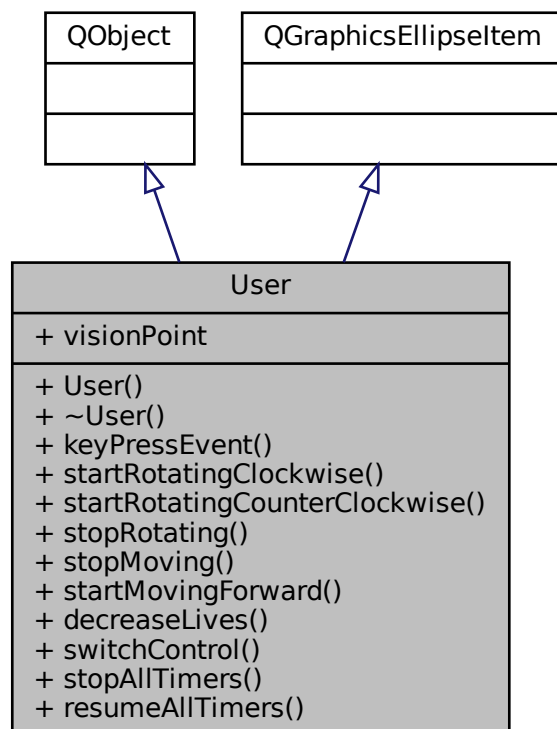
- [timer.h](#)
- [timer.cpp](#)

6.27 User Class Reference

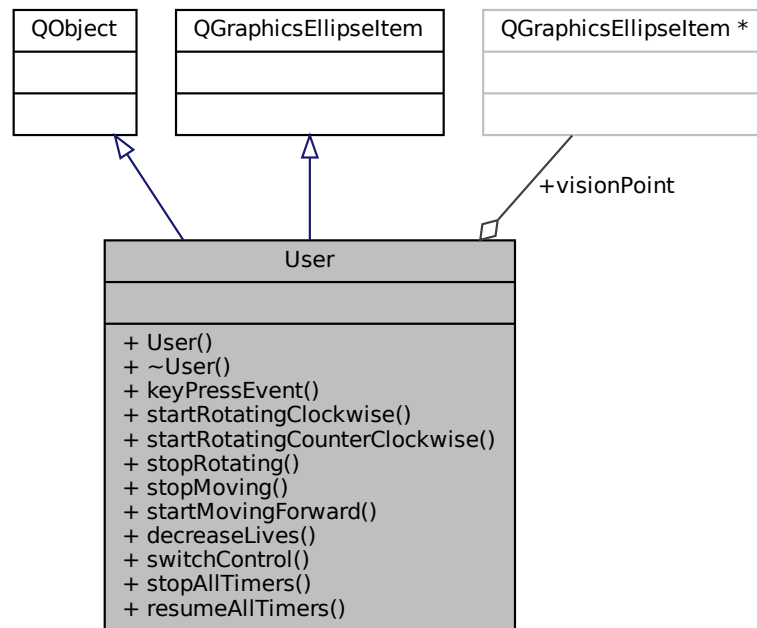
[User](#) class, inherits from [QObject](#) and [QGraphicsEllipseItem](#).

```
#include <user.h>
```

Inheritance diagram for User:



Collaboration diagram for User:



Public Slots

- void [startRotatingClockwise](#) ()
Start the clockwise rotation.
- void [startRotatingCounterClockwise](#) ()
Start the counter clockwise rotation.
- void [stopRotating](#) ()
Stop the rotation.
- void [stopMoving](#) ()
Stop the movement.
- void [startMovingForward](#) ()
Start the movement.
- void [decreaseLives](#) ()
Decrease the number of lives (3 default)
- void [switchControl](#) ()
Set the [User](#) as focusable (keyboard control)
- void [stopAllTimers](#) ()
Stop all timers.
- void [resumeAllTimers](#) ()
Resume all timers.

Signals

- void [stunned](#) ()
Signal emitted when the [User](#) is stunned.
- void [deleteLife3](#) ()
Signal emitted when the [User](#) is hit mapped to life3.
- void [deleteLife2](#) ()
Signal emitted when the [User](#) is hit mapped to life2.
- void [deleteLife1](#) ()
Signal emitted when the [User](#) is hit mapped to life1.

Public Member Functions

- [User](#) (QGraphicsItem *parent=0, int orientation=0, int velocity=5)
[User](#) constructor.
- [~User](#) ()
[User](#) destructor.
- void [keyPressEvent](#) (QKeyEvent *event)
Move the user using keyboard.

Public Attributes

- [QGraphicsEllipseItem](#) * [visionPoint](#)

6.27.1 Detailed Description

[User](#) class, inherits from [QObject](#) and [QGraphicsEllipseItem](#).

6.27.2 Constructor & Destructor Documentation

6.27.2.1 User()

```
User::User (
    QGraphicsItem * parent = 0,
    int orientation = 0,
    int velocity = 5 )
```

[User](#) constructor.

Parameters

<i>parent</i>	Parent QGraphicsItem
<i>orientation</i>	Orientation of the user
<i>velocity</i>	Velocity of the user

6.27.2.2 ~User()

```
User::~~User ( )
```

[User](#) destructor.

6.27.3 Member Function Documentation

6.27.3.1 decreaseLives

```
void User::decreaseLives ( ) [slot]
```

Decrease the number of lives (3 default)

6.27.3.2 deleteLife1

```
void User::deleteLife1 ( ) [signal]
```

Signal emitted when the [User](#) is hit mapped to life1.

6.27.3.3 deleteLife2

```
void User::deleteLife2 ( ) [signal]
```

Signal emitted when the [User](#) is hit mapped to life2.

6.27.3.4 deleteLife3

```
void User::deleteLife3 ( ) [signal]
```

Signal emitted when the [User](#) is hit mapped to life3.

6.27.3.5 keyPressEvent()

```
void User::keyPressEvent (
    QKeyEvent * event )
```

Move the user using keyboard.

Parameters

<i>event</i>	
--------------	--

6.27.3.6 resumeAllTimers

```
void User::resumeAllTimers ( ) [slot]
```

Resume all timers.

6.27.3.7 startMovingForward

```
void User::startMovingForward ( ) [slot]
```

Start the movement.

6.27.3.8 startRotatingClockwise

```
void User::startRotatingClockwise ( ) [slot]
```

Start the clockwise rotation.

6.27.3.9 startRotatingCounterClockwise

```
void User::startRotatingCounterClockwise ( ) [slot]
```

Start the counter clockwise rotation.

6.27.3.10 stopAllTimers

```
void User::stopAllTimers ( ) [slot]
```

Stop all timers.

6.27.3.11 stopMoving

```
void User::stopMoving ( ) [slot]
```

Stop the movement.

6.27.3.12 stopRotating

```
void User::stopRotating ( ) [slot]
```

Stop the rotation.

6.27.3.13 stunned

```
void User::stunned ( ) [signal]
```

Signal emitted when the [User](#) is stunned.

6.27.3.14 switchControl

```
void User::switchControl ( ) [slot]
```

Set the [User](#) as focusable (keyboard control)

6.27.4 Member Data Documentation

6.27.4.1 visionPoint

```
QGraphicsEllipseItem* User::visionPoint
```

A little dot representing the vision point

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

- [user.h](#)
- [user.cpp](#)

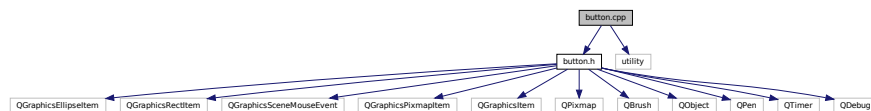
Chapter 7

File Documentation

7.1 button.cpp File Reference

[Button](#) class implementation file.

```
#include "button.h"
#include <utility>
Include dependency graph for button.cpp:
```



7.1.1 Detailed Description

[Button](#) class implementation file.

Version

1.0

This class is responsible for creating buttons in the game @project ICP Project - Epic Robot Survival

Author

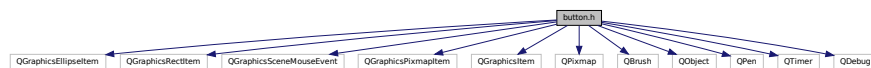
Marek Effenberger

7.2 button.h File Reference

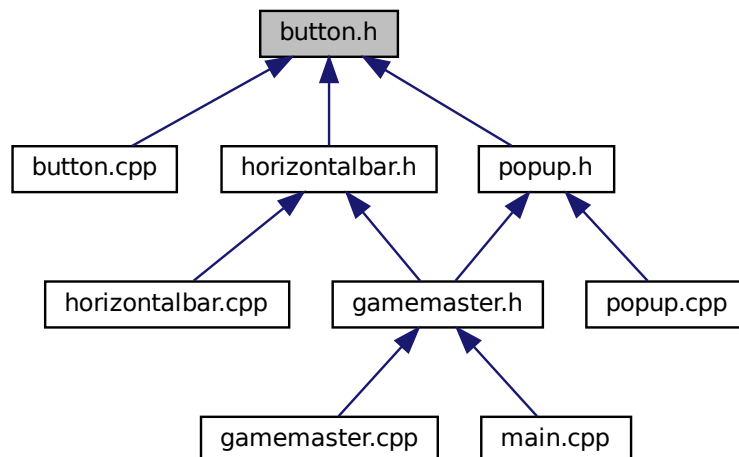
[Button](#) class header file.

```
#include <QGraphicsEllipseItem>
#include <QGraphicsRectItem>
#include <QGraphicsSceneMouseEvent>
#include <QGraphicsPixmapItem>
#include <QGraphicsItem>
#include <QPixmap>
#include <QBrush>
#include <QObject>
#include <QPen>
#include <QTimer>
#include <QDebug>
```

Include dependency graph for button.h:



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



Classes

- class [Button](#)

[Button](#) class, inherits from [QObject](#) and [QGraphicsRectItem](#).

7.2.1 Detailed Description

[Button](#) class header file.

Version

1.0

This class is responsible for creating buttons in the game @project ICP Project - Epic Robot Survival

Author

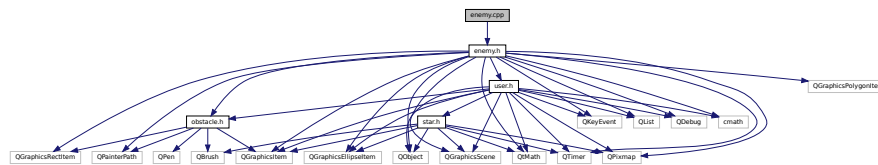
Marek Effenberger

7.3 enemy.cpp File Reference

[Enemy](#) class implementation file.

```
#include "enemy.h"
```

Include dependency graph for enemy.cpp:



7.3.1 Detailed Description

[Enemy](#) class implementation file.

Version

1.0

This class is responsible for creating autonomous robots (in our implementation - the enemies) in the game @project ICP Project - Epic Robot Survival

Author

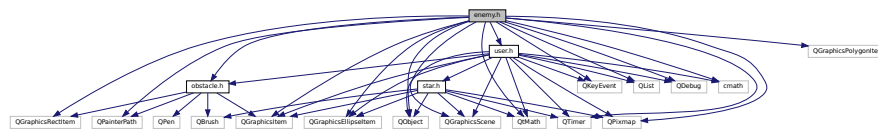
Marek Effenberger

7.4 enemy.h File Reference

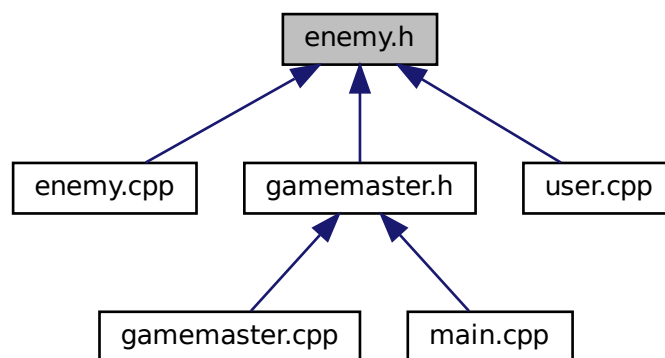
[Enemy](#) class header file.

```
#include <QGraphicsEllipseItem>
#include <QGraphicsRectItem>
#include <QObject>
#include <QKeyEvent>
#include <QList>
#include <QDebug>
#include <QGraphicsScene>
#include "obstacle.h"
#include <cmath>
#include <QGraphicsItem>
#include <QtMath>
#include <QTimer>
#include <QPixmap>
#include <QPainterPath>
#include "user.h"
#include <QGraphicsPolygonItem>
```

Include dependency graph for enemy.h:



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



Classes

- class [Enemy](#)
Enemy class, inherits from [QObject](#) and [QGraphicsEllipseItem](#).

7.4.1 Detailed Description

[Enemy](#) class header file.

Version

1.0

This class is responsible for creating autonomous robots (in our implementation - the enemies) in the game @project ICP Project - Epic Robot Survival

Author

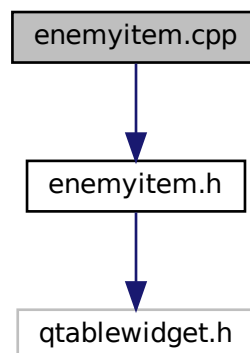
Marek Effenberger

7.5 enemyitem.cpp File Reference

[Enemy](#) Item Class Implementation file.

```
#include "enemyitem.h"
```

Include dependency graph for enemyitem.cpp:



7.5.1 Detailed Description

[Enemy](#) Item Class Implementation file.

Version

1.0

This class is responsible for holding the enemy item data in the game @project ICP Project - Epic Robot Survival

Author

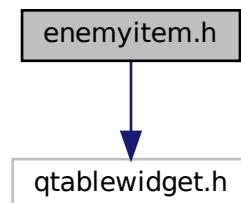
Samuel Hejnicek

7.6 enemyitem.h File Reference

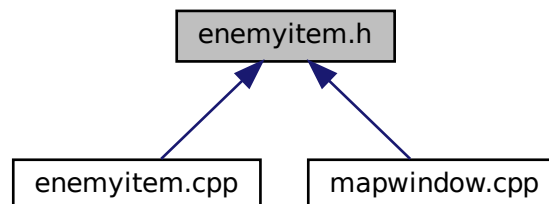
[Enemy](#) Item Class header file.

```
#include "qtablewidget.h"
```

Include dependency graph for enemyitem.h:



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



Classes

- class [EnemyItem](#)
The [EnemyItem](#) class.

7.6.1 Detailed Description

[Enemy](#) Item Class header file.

Version

1.0

This class is responsible for holding the enemy cell item data in the game @project ICP Project - Epic Robot Survival

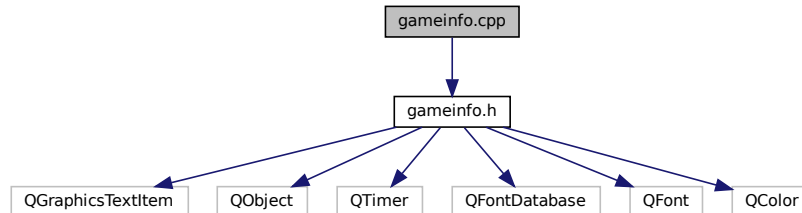
Author

Samuel Hejnicek

7.7 gameinfo.cpp File Reference

[GameInfo](#) class implementation file.

```
#include "gameinfo.h"
Include dependency graph for gameinfo.cpp:
```



7.7.1 Detailed Description

[GameInfo](#) class implementation file.

Version

1.0

This class is responsible for creating a little text box in the game @project ICP Project - Epic Robot Survival

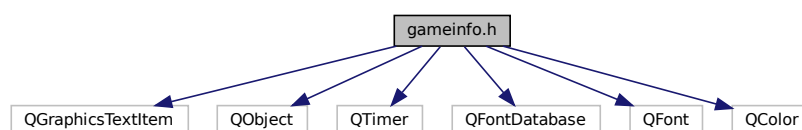
Author

Marek Effenberger

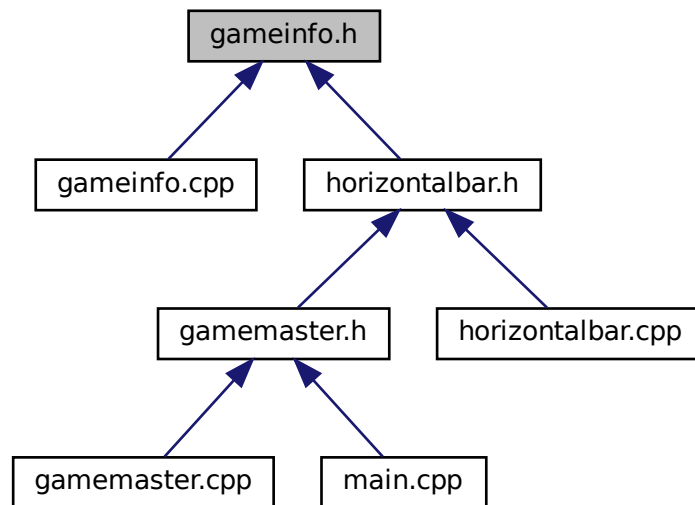
7.8 gameinfo.h File Reference

[GameInfo](#) class header file.

```
#include <QGraphicsTextItem>
#include <QObject>
#include <QTimer>
#include <QFontDatabase>
#include <QFont>
#include <QColor>
Include dependency graph for gameinfo.h:
```



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



Classes

- class [GameInfo](#)
[GameInfo](#) class, inherits from [QObject](#) and [QGraphicsTextItem](#).

7.8.1 Detailed Description

[GameInfo](#) class header file.

Version

1.0

This class is responsible for creating a little text box in the game @project ICP Project - Epic Robot Survival

Author

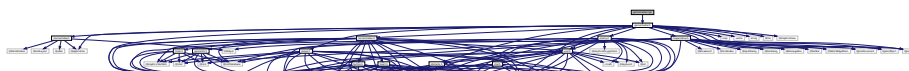
Marek Effenberger

7.9 gamemaster.cpp File Reference

[GameMaster](#) class implementation file.

```
#include "gamemaster.h"
```

Include dependency graph for gamemaster.cpp:



7.9.1 Detailed Description

[GameMaster](#) class implementation file.

Version

1.0

This class is responsible for correct switching between the game windows pausing the game, setting up the main menu and the game itself based on the parsed JSON file The scene creation and setting up the main game was inspired by [@https://www.youtube.com/watch?v=8ntEQpg7gck&list=PLyb40eoxkel0a5x4CB9fvGrkoBf8JzEwtV&ab_channel=Abdullah](https://www.youtube.com/watch?v=8ntEQpg7gck&list=PLyb40eoxkel0a5x4CB9fvGrkoBf8JzEwtV&ab_channel=Abdullah) @project ICP Project - Epic Robot Survival

Authors

Marek Effenberger, Samuel Hejníček

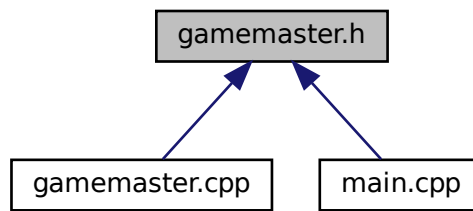
7.10 gamemaster.h File Reference

[GameMaster](#) class header file.

```
#include <QObject>
#include <map>
#include <vector>
#include <string>
#include "mainwindow.h"
#include "mapwindow.h"
#include <QJsonArray>
#include <QJsonObject>
#include <QJsonDocument>
#include <QFile>
#include "user.h"
#include "obstacle.h"
#include "enemy.h"
#include "horizontalbar.h"
#include "popup.h"
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QApplication>
Include dependency graph for gamemaster.h:
```



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



Classes

- class [GameMaster](#)
[GameMaster](#) class, inherits from [QObject](#).

7.10.1 Detailed Description

[GameMaster](#) class header file.

Version

1.0

This class is responsible for correct switching between the game windows pausing the game, setting up the main menu and the game itself based on the parsed JSON file @project ICP Project - Epic Robot Survival

Authors

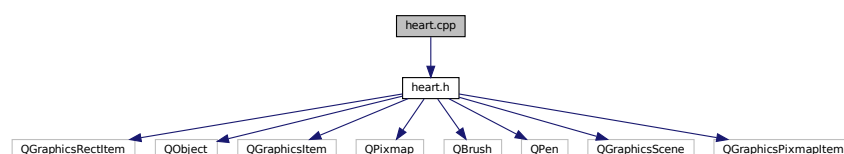
Marek Effenberger, Samuel Hejníček

7.11 heart.cpp File Reference

Hnemy class implementation file.

```
#include "heart.h"
```

Include dependency graph for heart.cpp:



7.11.1 Detailed Description

Hnemy class implementation file.

Version

1.0

This class is responsible for creating [Heart](#) Icons in the game @project ICP Project - Epic Robot Survival

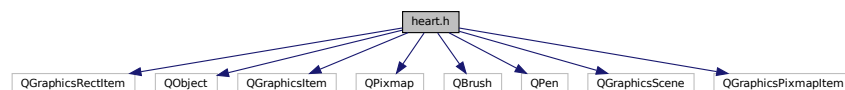
Author

Marek Effenberger

7.12 heart.h File Reference

[Heart](#) class header file.

```
#include <QGraphicsRectItem>
#include <QObject>
#include <QGraphicsItem>
#include <QPixmap>
#include <QBrush>
#include <QPen>
#include <QGraphicsScene>
#include <QGraphicsPixmapItem>
Include dependency graph for heart.h:
```



7.13.1 Detailed Description

HorizontalBar classes (upper, lower) implementation file.

Version

1.0

This class is responsible for creating the horizontal bars in the game @project ICP Project - Epic Robot Survival

Author

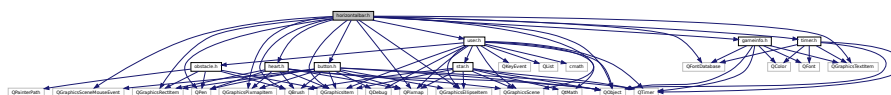
Marek Effenberger

7.14 horizontalbar.h File Reference

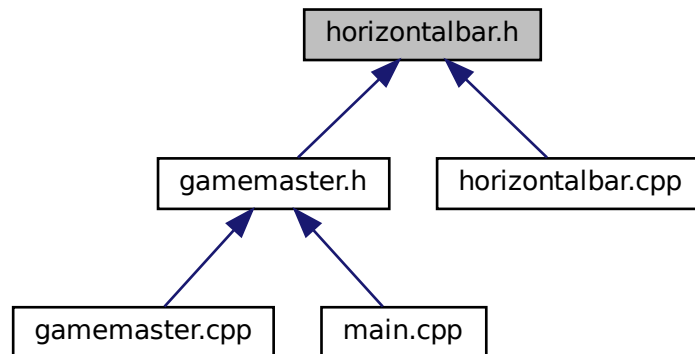
HorizontalBar classes (upper, lower) header file.

```
#include <QGraphicsRectItem>
#include <QGraphicsScene>
#include <QGraphicsEllipseItem>
#include <QGraphicsPixmapItem>
#include <QPixmap>
#include <QBrush>
#include <QGraphicsSceneMouseEvent>
#include "button.h"
#include "user.h"
#include <QGraphicsItem>
#include <QObject>
#include <QPen>
#include <QGraphicsTextItem>
#include <QFontDatabase>
#include "heart.h"
#include "gameinfo.h"
#include "timer.h"
```

Include dependency graph for horizontalbar.h:



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



Classes

- class [HorizontalUpperBar](#)
HorizontalUpperBar class, inherits from [QObject](#) and [QGraphicsRectItem](#).
- class [HorizontalLowerBar](#)
HorizontalLowerBar class, inherits from [QObject](#) and [QGraphicsRectItem](#).

7.14.1 Detailed Description

HorizontalBar classes (upper, lower) header file.

Version

1.0

This class is responsible for creating the horizontal bars in the game @project ICP Project - Epic Robot Survival

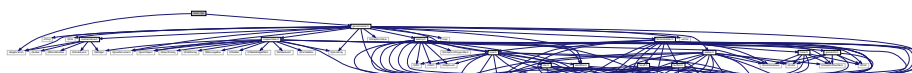
Author

Marek Effenberger

7.15 main.cpp File Reference

Main file of the project.

```
#include "gamemaster.h"
#include <QApplication>
Include dependency graph for main.cpp:
```



Functions

- `int main (int argc, char *argv[])`

7.15.1 Detailed Description

Main file of the project.

Version

1.0

Instantiates the `GameMaster` class and runs the game

Authors

Marek Effenberger, Samuel Hejníček

7.15.2 Function Documentation

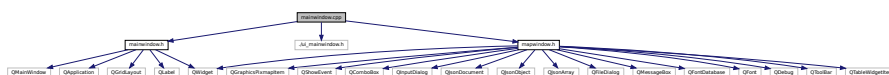
7.15.2.1 main()

```
int main (
    int argc,
    char * argv[ ] )
```

7.16 mainwindow.cpp File Reference

Main Window Class Implementation file.

```
#include "mainwindow.h"
#include "../ui_mainwindow.h"
#include "mapwindow.h"
Include dependency graph for mainwindow.cpp:
```



7.16.1 Detailed Description

Main Window Class Implementation file.

Version

1.0

This class is responsible for creating the main window of the robot game @project ICP Project - Epic Robot Survival

Author

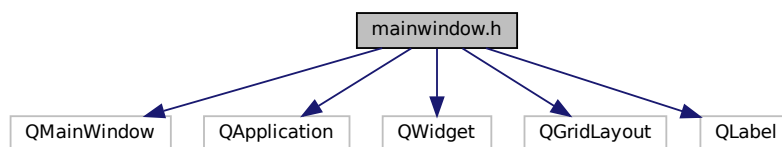
Samuel Hejnicek

7.17 mainwindow.h File Reference

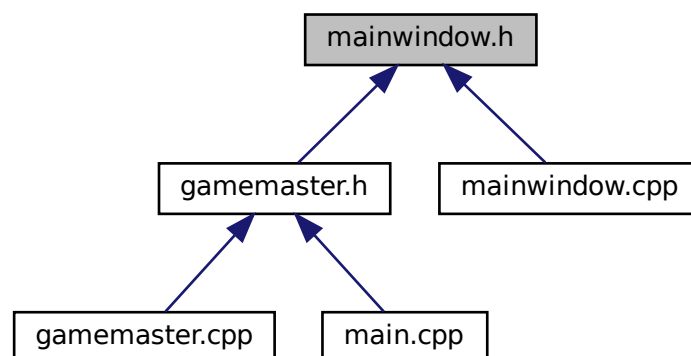
Main Window Class Header file.

```
#include <QMainWindow>
#include <QApplication>
#include <QWidget>
#include <QGridLayout>
#include <QLabel>
```

Include dependency graph for mainwindow.h:



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

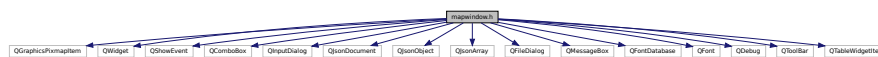


7.19 mapwindow.h File Reference

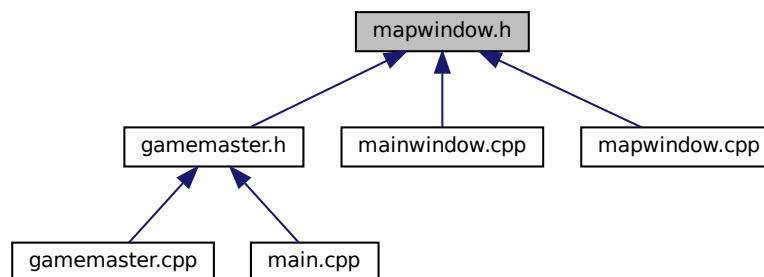
Map Window Class Header file.

```
#include <QGraphicsPixmapItem>
#include <QWidget>
#include <QShowEvent>
#include <QComboBox>
#include <QInputDialog>
#include <QJsonDocument>
#include <QJsonObject>
#include <QJsonArray>
#include <QFileDialog>
#include <QMessageBox>
#include <QFontDatabase>
#include <QFont>
#include <QDebug>
#include <QToolBar>
#include <QTableWidgetItem>
```

Include dependency graph for mapwindow.h:



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



Classes

- struct [pixmapIcons](#)
Struct to store the icons for the pixmap.
- class [MapWindow](#)
Class for the Map Window.

Namespaces

- [Ui](#)

Macros

- `#define SQUARE_SIZE 75`

Typedefs

- `typedef struct pixmapIcons pixMapIcons`
Struct to store the icons for the pixmap.

7.19.1 Detailed Description

Map Window Class Header file.

Version

1.0

This class is responsible for creating the map editor window of the game @project ICP Project - Epic Robot Survival

Author

Samuel Hejnicek

7.19.2 Macro Definition Documentation

7.19.2.1 SQUARE_SIZE

```
#define SQUARE_SIZE 75
```

7.19.3 Typedef Documentation

7.19.3.1 pixMapIcons

```
typedef struct pixmapIcons pixMapIcons
```

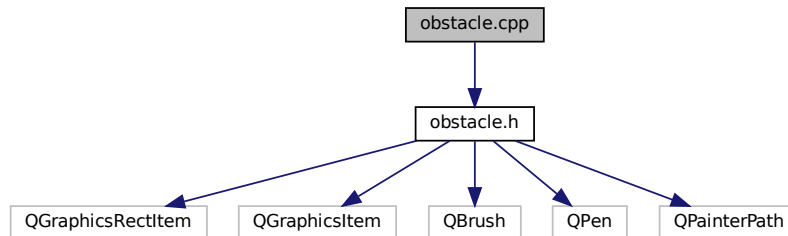
Struct to store the icons for the pixmap.

7.20 obstacle.cpp File Reference

[Obstacle](#) class implementation file.

```
#include "obstacle.h"
```

Include dependency graph for obstacle.cpp:



7.20.1 Detailed Description

[Obstacle](#) class implementation file.

Version

1.0

This class is responsible for creating the obstacles in the game @project ICP Project - Epic Robot Survival

Author

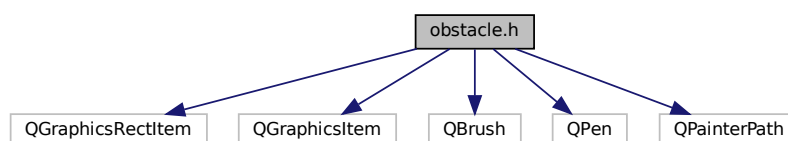
Marek Effenberger

7.21 obstacle.h File Reference

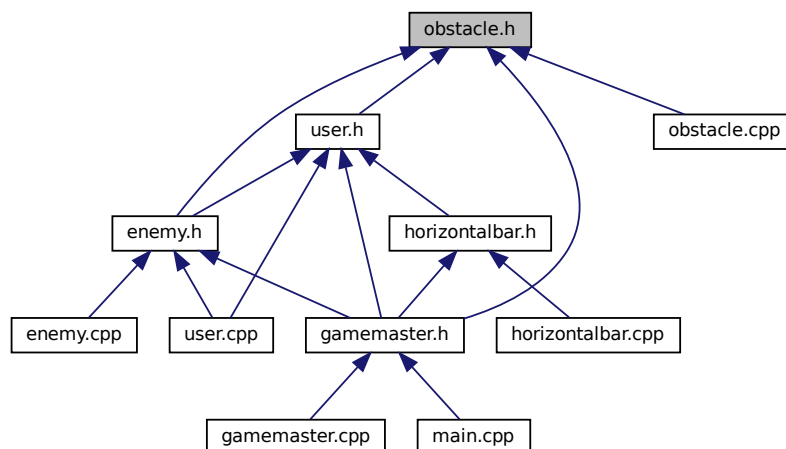
[Obstacle](#) class header file.

```
#include <QGraphicsRectItem>
#include <QGraphicsItem>
#include <QBrush>
#include <QPen>
#include <QPainterPath>
```

Include dependency graph for obstacle.h:



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



Classes

- class [Obstacle](#)

[Obstacle](#) class, inherits from [QGraphicsRectItem](#).

7.21.1 Detailed Description

[Obstacle](#) class header file.

Version

1.0

This class is responsible for creating the obstacles in the game @project ICP Project - Epic Robot Survival

Author

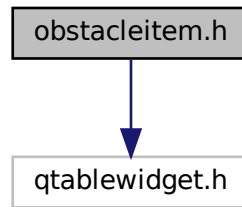
Marek Effenberger

7.22 obstacleitem.h File Reference

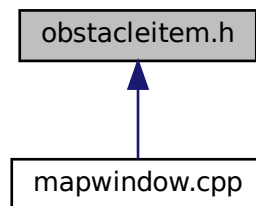
[Obstacle](#) Item Class Header file.

```
#include "qtablewidget.h"
```

Include dependency graph for obstacleitem.h:



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



Classes

- class [ObstacleItem](#)
The [ObstacleItem](#) class.

7.22.1 Detailed Description

[Obstacle](#) Item Class Header file.

Version

1.0

This class is responsible for holding the obstacle cell item data in the game @project ICP Project - Epic Robot Survival

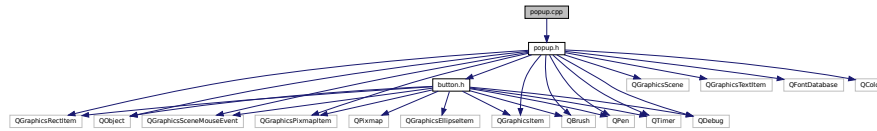
Author

Samuel Hejnicek

7.23 popup.cpp File Reference

Popup class implementation file.

```
#include "popup.h"
Include dependency graph for popup.cpp:
```



7.23.1 Detailed Description

Popup class implementation file.

Version

1.0

This class is responsible for creating the three popups in the game @project ICP Project - Epic Robot Survival

Author

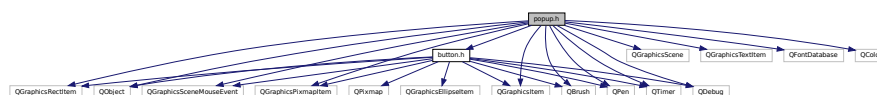
Marek Effenberger

7.24 popup.h File Reference

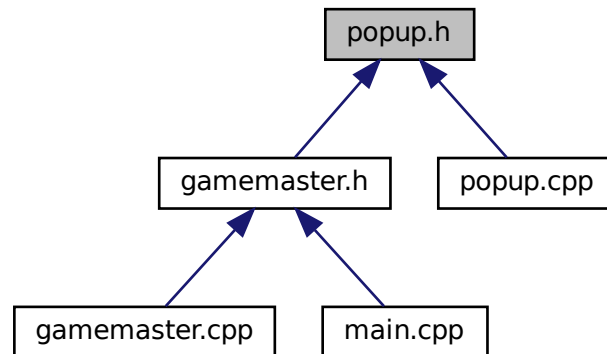
Popup class header file.

```
#include <QGraphicsRectItem>
#include <QObject>
#include "button.h"
#include <QGraphicsScene>
#include <QGraphicsPixmapItem>
#include <QGraphicsItem>
#include <QBrush>
#include <QPen>
#include <QGraphicsSceneMouseEvent>
#include <QGraphicsTextItem>
#include <QFontDatabase>
#include <QTimer>
#include <QDebug>
#include <QColor>
```

Include dependency graph for popup.h:



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



Classes

- class [PopUp](#)

Popup class, inherits from [QObject](#) and [QGraphicsRectItem](#).

7.24.1 Detailed Description

Popup class header file.

Version

1.0

This class is responsible for creating the three popups in the game @project ICP Project - Epic Robot Survival

Author

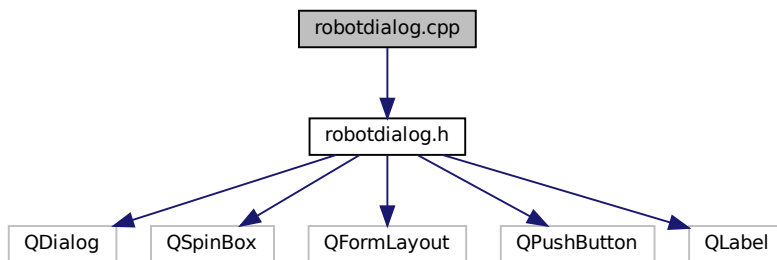
Marek Effenberger

7.25 robotdialog.cpp File Reference

Robot Dialog Class Implementation file.

```
#include "robotdialog.h"
```

Include dependency graph for robotdialog.cpp:



7.25.1 Detailed Description

Robot Dialog Class Implementation file.

Version

1.0

This class is responsible for creating a dialog where user can input the parameters of the robot/enemy @project ICP Project - Epic Robot Survival

Author

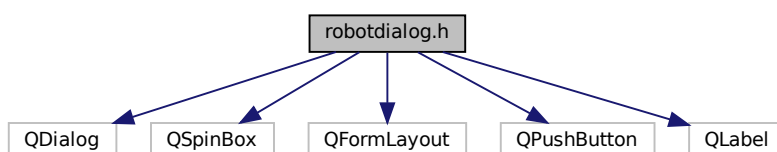
Samuel Hejnicek

7.26 robotdialog.h File Reference

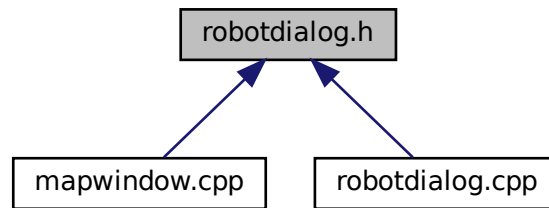
Robot Dialog Class Header file.

```
#include <QDialog>
#include <QSpinBox>
#include <QFormLayout>
#include <QPushButton>
#include <QLabel>
```

Include dependency graph for robotdialog.h:



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



Classes

- class [RobotParamDialog](#)
The [RobotParamDialog](#) class.

7.26.1 Detailed Description

Robot Dialog Class Header file.

Version

1.0

This class is responsible for creating a dialog where user can input the parameters of the robot/enemy @project ICP Project - Epic Robot Survival

Author

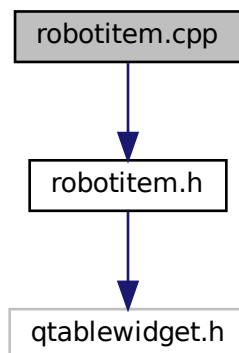
Samuel Hejnicek

7.27 robotitem.cpp File Reference

Robot Item Class Implementation file.

```
#include "robotitem.h"
```

Include dependency graph for robotitem.cpp:



7.27.1 Detailed Description

Robot Item Class Implementation file.

Version

1.0

This class is responsible for holding the controlled robot cell item data in the game @project ICP Project - Epic Robot Survival

Author

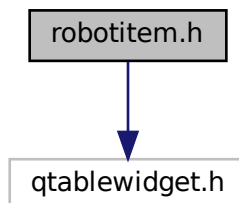
Samuel Hejnicek

7.28 robotitem.h File Reference

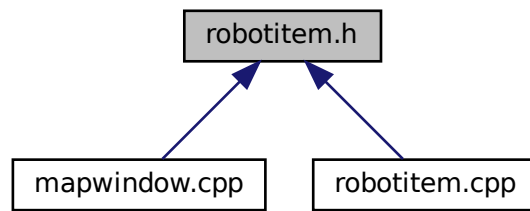
Robot Item Class header file.

```
#include "qtablewidget.h"
```

Include dependency graph for robotitem.h:



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



Classes

- class [RobotItem](#)
The [RobotItem](#) class.

7.28.1 Detailed Description

Robot Item Class header file.

Version

1.0

This class is responsible for holding the controlled robot cell item data in the game @project ICP Project - Epic Robot Survival

Author

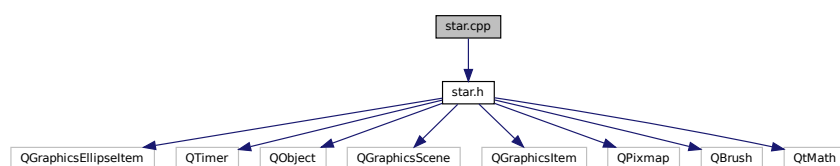
Samuel Hejnicek

7.29 star.cpp File Reference

[Star](#) class implementation file.

```
#include "star.h"
```

Include dependency graph for `star.cpp`:



7.29.1 Detailed Description

[Star](#) class implementation file.

Version

1.0

This class is responsible for creating and moving little stars when the user is stunned @project ICP Project - Epic Robot Survival

Author

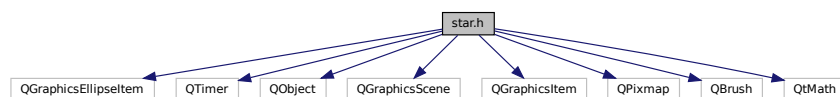
Marek Effenberger

7.30 star.h File Reference

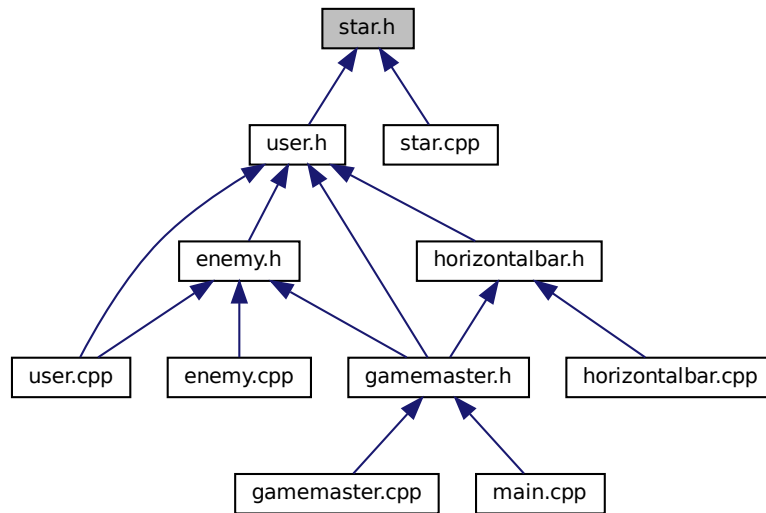
[Star](#) class header file.

```
#include <QGraphicsEllipseItem>
#include <QTimer>
#include <QObject>
#include <QGraphicsScene>
#include <QGraphicsItem>
#include <QPixmap>
#include <QBrush>
#include <QtMath>
```

Include dependency graph for star.h:



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



Classes

- class [Star](#)

[Star](#) class, inherits from [QObject](#) and [QGraphicsEllipseItem](#).

7.30.1 Detailed Description

[Star](#) class header file.

Version

1.0

This class is responsible for creating and moving little stars when the user is stunned @project ICP Project - Epic Robot Survival

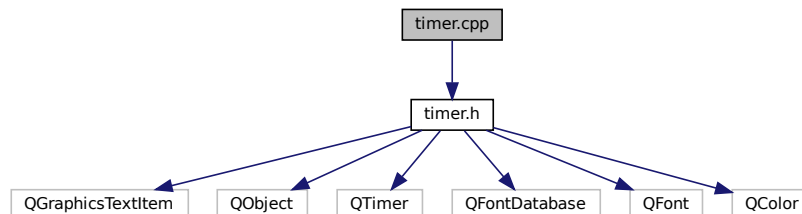
Author

Marek Effenberger

7.31 timer.cpp File Reference

[Timer](#) class implementation file.

```
#include "timer.h"
Include dependency graph for timer.cpp:
```



7.31.1 Detailed Description

[Timer](#) class implementation file.

Version

1.0

This class is responsible for a timer that counts up to a certain limit @project ICP Project - Epic Robot Survival

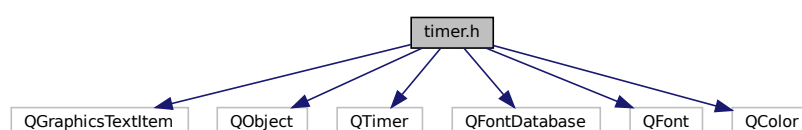
Author

Marek Effenberger

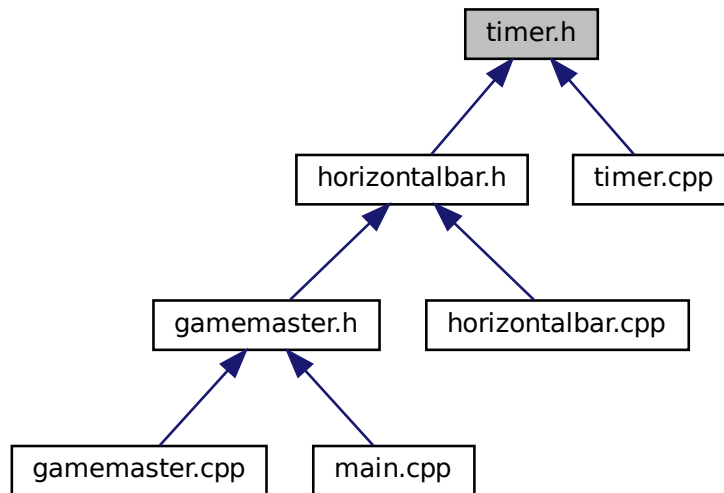
7.32 timer.h File Reference

[Timer](#) class header file.

```
#include <QGraphicsTextItem>
#include <QObject>
#include <QTimer>
#include <QFontDatabase>
#include <QFont>
#include <QColor>
Include dependency graph for timer.h:
```



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



Classes

- class [Timer](#)

[Timer](#) class, inherits from [QGraphicsTextItem](#).

7.32.1 Detailed Description

[Timer](#) class header file.

Version

1.0

This class is responsible for creating a timer that counts up to a certain limit @project ICP Project - Epic Robot Survival

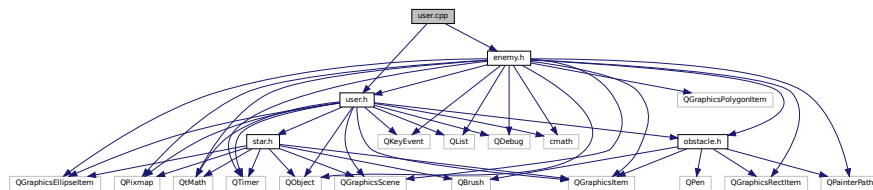
Author

Marek Effenberger

7.33 user.cpp File Reference

[User](#) class implementation file.

```
#include "user.h"
#include "enemy.h"
Include dependency graph for user.cpp:
```



7.33.1 Detailed Description

[User](#) class implementation file.

Version

1.0

This class is responsible for creating and moving the [User](#) robot @project ICP Project - Epic Robot Survival

Author

Marek Effenberger

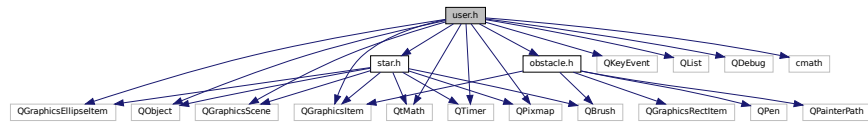
7.34 user.h File Reference

[User](#) class header file.

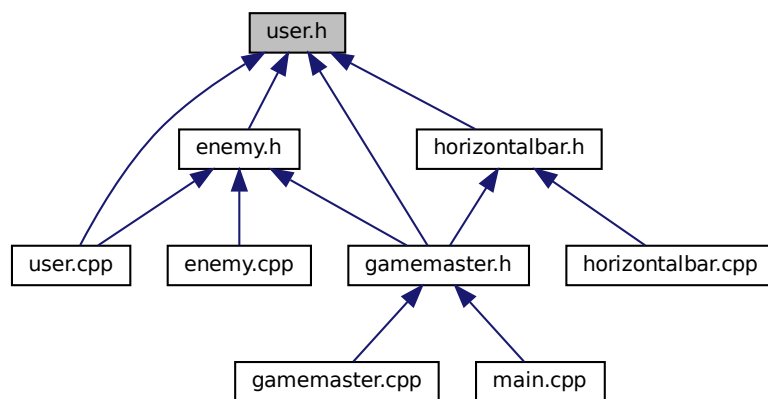
```
#include <QGraphicsEllipseItem>
#include <QObject>
#include <QKeyEvent>
#include <QList>
#include <QDebug>
#include <QGraphicsScene>
#include "obstacle.h"
#include <cmath>
#include <QGraphicsItem>
#include <QtMath>
#include <QTimer>
#include <QPixmap>
```

```
#include "star.h"
```

Include dependency graph for user.h:



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



Classes

- class [User](#)
User class, inherits from [QObject](#) and [QGraphicsEllipseItem](#).

7.34.1 Detailed Description

[User](#) class header file.

Version

1.0

This class is responsible for creating and moving the [User](#) robot @project ICP Project - Epic Robot Survival

Author

Marek Effenberger

Index

- ~Button
 - Button, [13](#)
- ~Enemy
 - Enemy, [18](#)
- ~EnemyItem
 - EnemyItem, [22](#)
- ~GameInfo
 - GameInfo, [26](#)
- ~Heart
 - Heart, [31](#)
- ~HorizontalLowerBar
 - HorizontalLowerBar, [33](#)
- ~HorizontalUpperBar
 - HorizontalUpperBar, [36](#)
- ~MainWindow
 - MainWindow, [38](#)
- ~MapWindow
 - MapWindow, [42](#)
- ~Obstacle
 - Obstacle, [47](#)
- ~PopUp
 - PopUp, [52](#)
- ~RobotItem
 - RobotItem, [64](#)
- ~RobotParamDialog
 - RobotParamDialog, [67](#)
- ~Star
 - Star, [70](#)
- ~Timer
 - Timer, [73](#)
- ~User
 - User, [77](#)
- applyGraphics
 - MainWindow, [38](#)
- Autopilot
 - HorizontalLowerBar, [33](#)
- autoPilot
 - Button, [13](#)
- Button, [11](#)
 - ~Button, [13](#)
 - autoPilot, [13](#)
 - Button, [13](#)
 - mousePressEvent, [13](#)
 - mouseReleaseEvent, [15](#)
 - pressed, [15](#)
 - released, [15](#)
 - setBorderWidth, [15](#)
- button.cpp, [81](#)
- button.h, [82](#)
- checkStuck
 - Enemy, [18](#)
- clearTableCell
 - MapWindow, [42](#)
- continueTimer
 - Timer, [73](#)
- createLayout
 - RobotParamDialog, [67](#)
- createNewMapWindow
 - MainWindow, [39](#)
- damage
 - GameInfo, [27](#)
- decreaseLives
 - User, [77](#)
- deleteLife1
 - User, [77](#)
- deleteLife2
 - User, [77](#)
- deleteLife3
 - User, [77](#)
- disableEditing
 - MapWindow, [43](#)
- Enemy, [16](#)
 - ~Enemy, [18](#)
 - checkStuck, [18](#)
 - Enemy, [18](#)
 - getUnstuck, [19](#)
 - hit, [19](#)
 - resumeAllTimers, [19](#)
 - startAutonomousMovement, [19](#)
 - stopAllTimers, [19](#)
 - stopChasing, [19](#)
- enemy
 - pixmapIcons, [49](#)
- enemy.cpp, [83](#)
- enemy.h, [84](#)
- EnemyItem, [20](#)
 - ~EnemyItem, [22](#)
 - EnemyItem, [22](#)
 - getDistance, [22](#)
 - getOrientation, [22](#)
 - getRotationAngle, [22](#)
 - getVelocity, [23](#)
 - setDistance, [23](#)
 - setOrientation, [23](#)
 - setRotationAngle, [24](#)

- setVelocity, 24
- enemyitem.cpp, 85
- enemyitem.h, 86
- exitButton
 - PopUp, 53
- fillFile
 - MapWindow, 43
- GameInfo, 24
 - ~GameInfo, 26
 - damage, 27
 - GameInfo, 26
 - ouch, 27
 - resetMessage, 27
 - updateMessage, 27
- gameinfo.cpp, 87
- gameinfo.h, 87
- GameMaster, 28
 - run, 29
- gamemaster.cpp, 88
- gamemaster.h, 89
- getDistance
 - EnemyItem, 22
 - RobotParamDialog, 67
- getOrientation
 - EnemyItem, 22
 - RobotItem, 64
 - RobotParamDialog, 68
- getRotationAngle
 - EnemyItem, 22
 - RobotParamDialog, 68
- getUnstuck
 - Enemy, 19
- getVelocity
 - EnemyItem, 23
 - RobotItem, 64
 - RobotParamDialog, 68
- Heart, 29
 - ~Heart, 31
 - Heart, 30
 - setDead, 31
- heart.cpp, 90
- heart.h, 91
- hide
 - PopUp, 52
- hit
 - Enemy, 19
- horizontalbar.cpp, 92
- horizontalbar.h, 93
- HorizontalLowerBar, 31
 - ~HorizontalLowerBar, 33
 - Autopilot, 33
 - HorizontalLowerBar, 32
 - mouseReleaseEvent, 33
- HorizontalUpperBar, 34
 - ~HorizontalUpperBar, 36
 - HorizontalUpperBar, 35
- pauseButton, 36
- timer, 36
- keyPressEvent
 - User, 77
- loadFile
 - MainWindow, 39
- loadGame
 - MainWindow, 39
- main
 - main.cpp, 95
- main.cpp, 94
 - main, 95
- mainMenuButton
 - PopUp, 53
- MainWindow, 37
 - ~MainWindow, 38
 - applyGraphics, 38
 - createNewMapWindow, 39
 - loadFile, 39
 - loadGame, 39
 - MainWindow, 38
 - quitApp, 39
- mainwindow.cpp, 95
- mainwindow.h, 96
- mapData
 - MapWindow, 45
- MapWindow, 40
 - ~MapWindow, 42
 - clearTableCell, 42
 - disableEditing, 43
 - fillFile, 43
 - mapData, 45
 - MapWindow, 42
 - setBarBottom, 43
 - setBarTop, 43
 - setTableCell, 43
 - startSession, 44
 - updateEnemyCounter, 44
 - updateObstacleCounter, 44
 - updateRobotCounter, 44
 - updateTimer, 44
- mapwindow.cpp, 97
- mapwindow.h, 98
 - pixMapIcons, 99
 - SQUARE_SIZE, 99
- mousePressEvent
 - Button, 13
- mouseReleaseEvent
 - Button, 15
 - HorizontalLowerBar, 33
- move
 - Star, 70
- Obstacle, 45
 - ~Obstacle, 47
 - Obstacle, 46

- obstacle
 - pixmapIcons, 49
- obstacle.cpp, 100
- obstacle.h, 100
- ObstacleItem, 47
 - ObstacleItem, 48
- obstacleitem.h, 101
- ouch
 - GameInfo, 27
- pauseButton
 - HorizontalUpperBar, 36
- pauseTimers
 - PopUp, 53
- pixmapIcons
 - mapwindow.h, 99
- pixmapIcons, 49
 - enemy, 49
 - obstacle, 49
 - robot, 50
- PopUp, 50
 - ~PopUp, 52
 - exitButton, 53
 - hide, 52
 - mainMenuButton, 53
 - pauseTimers, 53
 - PopUp, 52
 - restartButton, 53
 - resumeButton, 53
 - show, 53
- popup.cpp, 103
- popup.h, 103
- pressed
 - Button, 15
- QDialog, 54
- QGraphicsEllipseItem, 55
- QGraphicsRectItem, 56
- QGraphicsTextItem, 57
- QMainWindow, 58
- QObject, 59
- QTableWidgetItem, 60
- quitApp
 - MainWindow, 39
- QWidget, 61
- released
 - Button, 15
- resetMessage
 - GameInfo, 27
- restartButton
 - PopUp, 53
- resumeAllTimers
 - Enemy, 19
 - User, 78
- resumeButton
 - PopUp, 53
- robot
 - pixmapIcons, 50
 - robotdialog.cpp, 104
 - robotdialog.h, 105
 - RobotItem, 62
 - ~RobotItem, 64
 - getOrientation, 64
 - getVelocity, 64
 - RobotItem, 63
 - setOrientation, 64
 - setVelocity, 65
 - robotitem.cpp, 106
 - robotitem.h, 107
 - RobotParamDialog, 65
 - ~RobotParamDialog, 67
 - createLayout, 67
 - getDistance, 67
 - getOrientation, 68
 - getRotationAngle, 68
 - getVelocity, 68
 - RobotParamDialog, 67
- run
 - GameMaster, 29
- setBarBottom
 - MapWindow, 43
- setBarTop
 - MapWindow, 43
- setBorderWidth
 - Button, 15
- setDead
 - Heart, 31
- setDistance
 - EnemyItem, 23
- setOrientation
 - EnemyItem, 23
 - RobotItem, 64
- setRotationAngle
 - EnemyItem, 24
- setTableCell
 - MapWindow, 43
- setVelocity
 - EnemyItem, 24
 - RobotItem, 65
- show
 - PopUp, 53
- SQUARE_SIZE
 - mapwindow.h, 99
- Star, 69
 - ~Star, 70
 - move, 70
 - Star, 70
- star.cpp, 108
- star.h, 109
- startAutonomousMovement
 - Enemy, 19
- startMovingForward
 - User, 78
- startRotatingClockwise
 - User, 78
- startRotatingCounterClockwise

- User, 78
- startSession
 - MapWindow, 44
- stopAllTimers
 - Enemy, 19
 - User, 78
- stopChasing
 - Enemy, 19
- stopMoving
 - User, 78
- stopRotating
 - User, 79
- stopTimer
 - Timer, 73
- stunned
 - User, 79
- switchControl
 - User, 79
- timelsUp
 - Timer, 73
- Timer, 71
 - ~Timer, 73
 - continueTimer, 73
 - stopTimer, 73
 - timelsUp, 73
 - Timer, 73
 - updateTimer, 74
- timer
 - HorizontalUpperBar, 36
- timer.cpp, 111
- timer.h, 111
- Ui, 9
- updateEnemyCounter
 - MapWindow, 44
- updateMessage
 - GameInfo, 27
- updateObstacleCounter
 - MapWindow, 44
- updateRobotCounter
 - MapWindow, 44
- updateTimer
 - MapWindow, 44
 - Timer, 74
- User, 74
 - ~User, 77
 - decreaseLives, 77
 - deleteLife1, 77
 - deleteLife2, 77
 - deleteLife3, 77
 - keyPressEvent, 77
 - resumeAllTimers, 78
 - startMovingForward, 78
 - startRotatingClockwise, 78
 - startRotatingCounterClockwise, 78
 - stopAllTimers, 78
 - stopMoving, 78
 - stopRotating, 79
 - stunned, 79
 - switchControl, 79
 - User, 76
 - visionPoint, 79
- user.cpp, 113
- user.h, 113
- visionPoint
 - User, 79