

Hole_in_one

Generated by Doxygen 1.8.11

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	Block Class Reference	7
4.1.1	Detailed Description	8
4.1.2	Constructor & Destructor Documentation	8
4.1.2.1	Block(b2World *world, QGraphicsScene *level, b2Vec2 center, qreal m_angle, qreal m_length, qreal m_width, b2BodyType type, qreal friction, QString mode)	8
4.1.3	Member Function Documentation	8
4.1.3.1	drawGraphics()	8
4.1.4	Member Data Documentation	8
4.1.4.1	angle	8
4.1.4.2	body	8
4.1.4.3	graphics	8
4.1.4.4	length	9
4.1.4.5	width	9
4.2	Circle Class Reference	9
4.2.1	Constructor & Destructor Documentation	9

4.2.1.1	Circle(b2World *world, QGraphicsScene *level, QPointF position, qreal angle, b2BodyType type, b2CircleShape &circle, QString mode)	9
4.2.2	Member Function Documentation	10
4.2.2.1	createCircle(b2World world, QGraphicsScene levelscene, QPointF pos, qreal angle, b2BodyType type, b2CircleShape &circle)	10
4.2.2.2	draw()	10
4.2.2.3	drawBall1()	10
4.2.2.4	drawGraphics()	10
4.2.3	Member Data Documentation	10
4.2.3.1	body	10
4.2.3.2	graphics	10
4.3	GUI Class Reference	11
4.3.1	Detailed Description	12
4.3.2	Constructor & Destructor Documentation	12
4.3.2.1	GUI(QWidget *parent=NULL)	12
4.3.3	Member Function Documentation	12
4.3.3.1	back	12
4.3.3.2	box	13
4.3.3.3	checkLevel()	13
4.3.3.4	circle	13
4.3.3.5	csnd	13
4.3.3.6	displayGUI()	13
4.3.3.7	help	13
4.3.3.8	highscore	13
4.3.3.9	levelMenu	14
4.3.3.10	mute	14
4.3.3.11	showGuiagain	14
4.3.3.12	showLevel1	14
4.3.3.13	showLevel2	14
4.3.3.14	showLevel3	14
4.3.3.15	showLevel4	14

4.3.3.16	triangle	14
4.3.4	Member Data Documentation	14
4.3.4.1	ismute	14
4.3.4.2	mutepicButton	15
4.3.4.3	scene	15
4.4	Level_1 Class Reference	15
4.4.1	Constructor & Destructor Documentation	16
4.4.1.1	Level_1()	16
4.4.2	Member Function Documentation	16
4.4.2.1	addCircle	16
4.4.2.2	addRectangle	16
4.4.2.3	closeLevel	16
4.4.2.4	getTime	16
4.4.2.5	highscoreCounter	17
4.4.2.6	levelcompleted	17
4.4.2.7	pauseLevel	17
4.4.2.8	reset	17
4.4.2.9	resumeLevel	17
4.4.2.10	rotateLeft	17
4.4.2.11	rotateRight	17
4.4.2.12	startLevel	17
4.4.2.13	update	17
4.4.3	Member Data Documentation	17
4.4.3.1	vectb	17
4.4.3.2	vectt	17
4.5	Level_2 Class Reference	18
4.5.1	Constructor & Destructor Documentation	19
4.5.1.1	Level_2()	19
4.5.2	Member Function Documentation	19
4.5.2.1	addCircle	19

4.5.2.2	addRectangle	19
4.5.2.3	addTriangle	19
4.5.2.4	closeLevel	19
4.5.2.5	getTime	19
4.5.2.6	highscoreCounter	19
4.5.2.7	levelcompleted	19
4.5.2.8	pauseLevel	19
4.5.2.9	reset	20
4.5.2.10	resumeLevel	20
4.5.2.11	rotateLeft	20
4.5.2.12	rotateRight	20
4.5.2.13	startLevel	20
4.5.2.14	update	20
4.6	Level_3 Class Reference	20
4.6.1	Constructor & Destructor Documentation	21
4.6.1.1	Level_3()	21
4.6.2	Member Function Documentation	22
4.6.2.1	addCircle	22
4.6.2.2	addRectangle	22
4.6.2.3	addTriangle	22
4.6.2.4	closeLevel	22
4.6.2.5	getTime	22
4.6.2.6	highscoreCounter	22
4.6.2.7	levelcompleted	22
4.6.2.8	pauseLevel	22
4.6.2.9	reset	22
4.6.2.10	resumeLevel	22
4.6.2.11	rotateLeft	23
4.6.2.12	rotateRight	23
4.6.2.13	startLevel	23

4.6.2.14	update	23
4.7	Level_4 Class Reference	23
4.7.1	Constructor & Destructor Documentation	24
4.7.1.1	Level_4()	24
4.7.2	Member Function Documentation	25
4.7.2.1	addCircle	25
4.7.2.2	addRectangle	25
4.7.2.3	addTriangle	25
4.7.2.4	closeLevel	25
4.7.2.5	getTime	25
4.7.2.6	highscoreCounter	25
4.7.2.7	levelcompleted	25
4.7.2.8	pauseLevel	25
4.7.2.9	reset	25
4.7.2.10	resumeLevel	25
4.7.2.11	rotateLeft	26
4.7.2.12	rotateRight	26
4.7.2.13	startLevel	26
4.7.2.14	update	26
4.8	MeinElement Class Reference	26
4.8.1	Constructor & Destructor Documentation	27
4.8.1.1	MeinElement(b2World *world, QGraphicsScene *level, b2Vec2 center, qreal length, qreal width, b2BodyType type, qreal friction)	27
4.8.1.2	MeinElement(QGraphicsScene *level, QPointF center, qreal length, qreal width)	27
4.8.2	Member Function Documentation	27
4.8.2.1	drawBottom()	27
4.8.3	Member Data Documentation	27
4.8.3.1	applause	28
4.8.3.2	body	28
4.8.3.3	graphics	28
4.8.3.4	white	28

4.9	Mover Class Reference	28
4.9.1	Constructor & Destructor Documentation	28
4.9.1.1	Mover(b2World *world, QGraphicsScene *level, b2Vec2 center, qreal m_angle, qreal m_length, qreal m_width, qreal friction, QString mode)	28
4.9.2	Member Data Documentation	29
4.9.2.1	angle	29
4.9.2.2	body	29
4.9.2.3	graphics	29
4.9.2.4	length	29
4.9.2.5	width	29
4.10	Paperball Class Reference	30
4.10.1	Detailed Description	30
4.10.2	Constructor & Destructor Documentation	30
4.10.2.1	Paperball(b2World *world, QGraphicsScene *level, QPointF position, qreal angle, b2BodyType type, b2CircleShape &circle)	30
4.10.3	Member Function Documentation	31
4.10.3.1	createPaper(b2World world, QGraphicsScene levelscene, QPointF pos, qreal angle, b2BodyType type, b2CircleShape &circle)	31
4.10.3.2	drawBall1()	31
4.10.4	Member Data Documentation	31
4.10.4.1	body	31
4.10.4.2	graphics	31
4.11	picButton Class Reference	31
4.11.1	Detailed Description	32
4.11.2	Constructor & Destructor Documentation	32
4.11.2.1	picButton(QPixmap _defaultpic, QPixmap _hoverpic)	32
4.11.2.2	picButton(QPixmap _defaultpic, QPixmap _hoverpic, bool hover)	32
4.11.3	Member Function Documentation	33
4.11.3.1	enterEvent(QEvent *event)	33
4.11.3.2	leaveEvent(QEvent *event)	33
4.11.3.3	setdefaultpic(QPixmap defaultpic)	33
4.11.3.4	sethoverpic(QPixmap hoverpic)	33

4.11.4	Member Data Documentation	34
4.11.4.1	defaultpic	34
4.11.4.2	hover	34
4.11.4.3	hoverpic	34
4.12	RecycleBin Class Reference	34
4.12.1	Detailed Description	34
4.12.2	Constructor & Destructor Documentation	34
4.12.2.1	RecycleBin(b2World *world, QGraphicsScene *level, QPointF a, QPointF b, Q↔ PointF c, QPointF d, qreal angle, b2BodyType type)	34
4.12.3	Member Function Documentation	35
4.12.3.1	drawGraphics()	35
4.12.4	Member Data Documentation	35
4.12.4.1	body	35
4.12.4.2	graphics	35
4.13	RecycleBinGraphics Class Reference	35
4.13.1	Detailed Description	36
4.13.2	Constructor & Destructor Documentation	36
4.13.2.1	RecycleBinGraphics(QGraphicsScene *level)	36
4.13.3	Member Data Documentation	36
4.13.3.1	graphics	36
4.14	Trampoline Class Reference	36
4.14.1	Constructor & Destructor Documentation	37
4.14.1.1	Trampoline(b2World *world, QGraphicsScene *level, b2Vec2 center, qreal m↔ _angle, qreal m_length, qreal m_width, b2BodyType type, qreal friction, QString mode)	37
4.14.2	Member Data Documentation	37
4.14.2.1	angle	37
4.14.2.2	body	37
4.14.2.3	graphics	38
4.14.2.4	length	38
4.14.2.5	width	38
4.15	Triangle Class Reference	38
4.15.1	Constructor & Destructor Documentation	38
4.15.1.1	Triangle(b2World *world, QGraphicsScene *level, QPointF a, QPointF b, Q↔ PointF c, qreal angle, b2BodyType type, qreal friction, QString mode)	38
4.15.2	Member Function Documentation	39
4.15.2.1	drawGraphics()	39
4.15.3	Member Data Documentation	39
4.15.3.1	body	39
4.15.3.2	graphics	39

5	File Documentation	41
5.1	Game/block.cpp File Reference	41
5.2	Game/block.h File Reference	41
5.3	Game/circle.cpp File Reference	41
5.4	Game/circle.h File Reference	42
5.5	Game/gui.cpp File Reference	42
5.6	Game/gui.h File Reference	42
5.7	Game/level_1.cpp File Reference	43
5.8	Game/level_1.h File Reference	43
5.8.1	Macro Definition Documentation	44
5.8.1.1	framerate	44
5.9	Game/level_2.cpp File Reference	44
5.10	Game/level_2.h File Reference	44
5.10.1	Macro Definition Documentation	45
5.10.1.1	framerate	45
5.11	Game/level_3.cpp File Reference	45
5.12	Game/level_3.h File Reference	45
5.12.1	Macro Definition Documentation	46
5.12.1.1	framerate	46
5.13	Game/level_4.cpp File Reference	46
5.14	Game/level_4.h File Reference	46
5.14.1	Macro Definition Documentation	47
5.14.1.1	framerate	47
5.15	Game/main.cpp File Reference	47
5.15.1	Function Documentation	47
5.15.1.1	main(int argc, char *argv[])	47
5.16	Game/meinelement.cpp File Reference	47
5.17	Game/meinelement.h File Reference	47
5.18	Game/mover.cpp File Reference	47
5.19	Game/mover.h File Reference	48
5.20	Game/paperball.cpp File Reference	48
5.21	Game/paperball.h File Reference	48
5.22	Game/picbutton.cpp File Reference	48
5.23	Game/picbutton.h File Reference	48
5.24	Game/recyclebin.cpp File Reference	49
5.25	Game/recyclebin.h File Reference	49
5.26	Game/recyclebingraphics.cpp File Reference	49
5.27	Game/recyclebingraphics.h File Reference	49
5.28	Game/trampoline.cpp File Reference	50
5.29	Game/trampoline.h File Reference	50
5.30	Game/triangle.cpp File Reference	50
5.31	Game/triangle.h File Reference	50

Index	51
-----------------------	----

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Mover	28
QGraphicsView	
GUI	11
Level_1	15
Level_2	18
Level_3	20
Level_4	23
QObject	
Block	7
Circle	9
MeinElement	26
Paperball	30
Trampoline	36
Triangle	38
QPushButton	
picButton	31
RecycleBin	34
RecycleBinGraphics	35

Chapter 2

Class Index

2.1 Class List

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

Block		
	The Block class	7
Circle	9
GUI		
	The GUI class	11
Level_1	15
Level_2	18
Level_3	20
Level_4	23
MeinElement	26
Mover	28
Paperball		
	The Paperball class	30
picButton		
	The picButton class	31
RecycleBin		
	The RecycleBin class	34
RecycleBinGraphics		
	The RecycleBinGraphics class	35
Trampoline	36
Triangle	38

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Game/block.cpp	41
Game/block.h	41
Game/circle.cpp	41
Game/circle.h	42
Game/gui.cpp	42
Game/gui.h	42
Game/level_1.cpp	43
Game/level_1.h	43
Game/level_2.cpp	44
Game/level_2.h	44
Game/level_3.cpp	45
Game/level_3.h	45
Game/level_4.cpp	46
Game/level_4.h	46
Game/main.cpp	47
Game/meinelement.cpp	47
Game/meinelement.h	47
Game/mover.cpp	47
Game/mover.h	48
Game/paperball.cpp	48
Game/paperball.h	48
Game/picbutton.cpp	48
Game/picbutton.h	48
Game/recyclebin.cpp	49
Game/recyclebin.h	49
Game/recyclebingraphics.cpp	49
Game/recyclebingraphics.h	49
Game/trampoline.cpp	50
Game/trampoline.h	50
Game/triangle.cpp	50
Game/triangle.h	50

Chapter 4

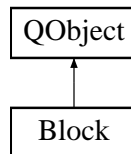
Class Documentation

4.1 Block Class Reference

The [Block](#) class.

```
#include <block.h>
```

Inheritance diagram for Block:



Public Member Functions

- [Block](#) (b2World *world, QGraphicsScene *level, b2Vec2 center, qreal m_angle, qreal m_length, qreal m_↔width, b2BodyType type, qreal friction, QString mode)
[Block::Block](#).
- void [drawGraphics](#) ()
[Block::drawGraphics](#) connects the Box2D-Object to the Graphics after relocation.

Public Attributes

- qreal [length](#)
lenth of the object
- qreal [width](#)
width of the object
- qreal [angle](#)
angle of the object
- b2Body * [body](#)
Box2D Body of object.
- QGraphicsItem * [graphics](#)
graphic of object

4.1.1 Detailed Description

The [Block](#) class.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 **Block::Block** (*b2World * world*, *QGraphicsScene * level*, *b2Vec2 center*, *qreal m_angle*, *qreal m_length*, *qreal m_width*, *b2BodyType type*, *qreal friction*, *QString mode*)

[Block::Block](#).

Parameters

<i>world</i>	: Box2D world for physic engine
<i>level</i>	: Scene for the game
<i>center</i>	: is the centerposition of the block
<i>m_angle</i>	angle for the block
<i>m_length</i>	: length bock
<i>m_width</i>	: width block
<i>type</i>	: Box2D type of the Bblock(if it's static or dynamic)
<i>friction</i>	: friction for the Block
<i>mode</i>	: is it a obstacle or a tool

4.1.3 Member Function Documentation

4.1.3.1 **void Block::drawGraphics** ()

[Block::drawGraphics](#) connects the Box2D-Object to the Graphics after relocation.

4.1.4 Member Data Documentation

4.1.4.1 **qreal Block::angle**

angle of the object

4.1.4.2 **b2Body* Block::body**

Box2D Body of object.

4.1.4.3 **QGraphicsItem* Block::graphics**

graphic of object

4.1.4.4 qreal Block::length

length of the object

4.1.4.5 qreal Block::width

width of the object

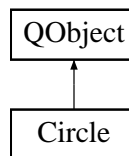
The documentation for this class was generated from the following files:

- [Game/block.h](#)
- [Game/block.cpp](#)

4.2 Circle Class Reference

```
#include <circle.h>
```

Inheritance diagram for Circle:



Public Member Functions

- [Circle](#) (b2World *world, QGraphicsScene *level, QPointF position, qreal angle, b2BodyType type, b2CircleShape &circle, QString mode)
Circle::Circle.
- void [createCircle](#) (b2World world, QGraphicsScene levelscene, QPointF pos, qreal angle, b2BodyType type, b2CircleShape &circle)
- void [draw](#) ()
Circle::draw connects the Graphics to the Box2D-Object.
- bool [drawBall1](#) ()
Circle::drawBall1 connects the Graphics to the Box2D-Object.
- void [drawGraphics](#) ()
Circle::drawGraphics connects the Box2D-Object to the Graphics after relocation.

Public Attributes

- b2Body * [body](#)
Box2D Body of object.
- QGraphicsItem * [graphics](#)
graphic of object

4.2.1 Constructor & Destructor Documentation

- 4.2.1.1 [Circle::Circle](#) (b2World * world, QGraphicsScene * level, QPointF position, qreal angle, b2BodyType type, b2CircleShape & circle, QString mode)

[Circle::Circle.](#)

Parameters

<i>world</i>	: Box2D world for physic engine
<i>level</i>	: Scene for the game
<i>position</i>	: left upper corner
<i>angle</i>	: angle for the circle
<i>type</i>	: Box2D type of the circle(if it's static or dynamic)
<i>circle</i>	: Box2D knows that it is a circle
<i>mode</i>	: is it a obstacle or a tool

4.2.2 Member Function Documentation

4.2.2.1 void Circle::createCircle (b2World *world*, QGraphicsScene *levelscene*, QPointF *pos*, qreal *angle*, b2BodyType *type*, b2CircleShape & *circle*)

4.2.2.2 void Circle::draw ()

[Circle::draw](#) connects the Graphics to the Box2D-Object.

4.2.2.3 bool Circle::drawBall1 ()

[Circle::drawBall1](#) connects the Graphics to the Box2D-Object.

Returns

4.2.2.4 void Circle::drawGraphics ()

[Circle::drawGraphics](#) connects the Box2D-Object to the Graphics after relocation.

4.2.3 Member Data Documentation

4.2.3.1 b2Body* Circle::body

Box2D Body of object.

4.2.3.2 QGraphicsItem* Circle::graphics

graphic of object

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

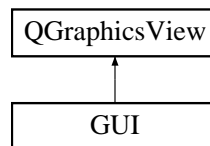
- [Game/circle.h](#)
- [Game/circle.cpp](#)

4.3 GUI Class Reference

The [GUI](#) class.

```
#include <gui.h>
```

Inheritance diagram for GUI:



Public Slots

- void [levelMenu](#) ()
[GUI::levelMenu](#) opens the Levelmenu shows the Levels which can be selected(if they are enabled) opens the selected Level.
- void [highscore](#) ()
[GUI::highscore](#) opens the Highscoretable and fill up the table for the level which are finished.
- void [back](#) ()
[GUI::back](#).
- void [showLevel1](#) ()
[GUI::showLevel1](#) starts [Level_1](#) and hide gui.
- void [showLevel2](#) ()
[GUI::showLevel2](#) starts [Level_2](#) and hide gui.
- void [showLevel3](#) ()
[GUI::showLevel3](#) starts [Level_3](#) and hide gui.
- void [showLevel4](#) ()
[GUI::showLevel4](#) starts [Level_4](#) and hide gui.
- void [showGuiagain](#) ()
[GUI::showGuiagain](#) reopen [GUI](#) after finish Level or close the level.
- void [help](#) ()
[GUI::help](#) opens the Helpmenu.
- void [box](#) ()
[GUI::box](#) box=rectangle helpmenu.
- void [circle](#) ()
[GUI::circle](#) circle helpmenu.
- void [triangle](#) ()
[GUI::triangle](#) triangle helpmenu.
- void [mute](#) ()
[GUI::mute](#) enables and disables the backgroundsound.
- void [csnd](#) ()
[GUI::csnd](#) paly sound if it is not muted.

Public Member Functions

- [GUI](#) (QWidget *parent=NULL)
[GUI::GUI.](#)
- void [displayGUI](#) ()
[GUI::displayGUI](#) opens the Startmenu creates the needed Buttons and connects them.
- void [checkLevel](#) ()
[GUI::checkLevel](#) read out level.txt and fill the content in levelenab.

Public Attributes

- QGraphicsScene * [scene](#)
Scene for [GUI](#).
- [picButton](#) * [mutepicButton](#)
Mutepicbutton for Sound.
- bool [ismute](#) = false
Check if Sound is on or off.

4.3.1 Detailed Description

The [GUI](#) class.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 [GUI::GUI](#) (QWidget * *parent* = NULL)

[GUI::GUI.](#)

Parameters

<i>parent</i>	Create QGraphicsView and enter scene
---------------	--------------------------------------

Screen setup. No scroll bar available

Set Application-Name

Scene setup

ismute false by default

Sound

4.3.3 Member Function Documentation

4.3.3.1 void [GUI::back](#) () [slot]

[GUI::back.](#)

4.3.3.2 void GUI::box () [slot]

[GUI::box](#) box=rectangle helpmenu.

create title text

4.3.3.3 void GUI::checkLevel ()

[GUI::checkLevel](#) read out level.txt and fill the content in levelenab.

4.3.3.4 void GUI::circle () [slot]

[GUI::circle](#) circle helpmenu.

create title text

4.3.3.5 void GUI::csnd () [slot]

[GUI::csnd](#) paly sound if it is not muted.

4.3.3.6 void GUI::displayGUI ()

[GUI::displayGUI](#) opens the Startmenu creates the needed Buttons and connects them.

create title text

create level menu button

create highscore button

create help button

create quit button

create sound button

4.3.3.7 void GUI::help () [slot]

[GUI::help](#) opens the Helpmenu.

create title text

4.3.3.8 void GUI::highscore () [slot]

[GUI::highscore](#) opens the Highscoretable and fill up the table for the level which are finished.

create title text

4.3.3.9 void GUI::levelMenu () [slot]

[GUI::levelMenu](#) opens the Levelmenu shows the Levels which can be selected(if they are enabled) opens the selected Level.

create level menu button

4.3.3.10 void GUI::mute () [slot]

[GUI::mute](#) enables and disables the backgroundsound.

4.3.3.11 void GUI::showGuiagain () [slot]

[GUI::showGuiagain](#) reopen [GUI](#) after finish Level or close the level.

4.3.3.12 void GUI::showLevel1 () [slot]

[GUI::showLevel1](#) starts [Level_1](#) and hide gui.

4.3.3.13 void GUI::showLevel2 () [slot]

[GUI::showLevel2](#) starts [Level_2](#) and hide gui.

4.3.3.14 void GUI::showLevel3 () [slot]

[GUI::showLevel3](#) starts [Level_3](#) and hide gui.

4.3.3.15 void GUI::showLevel4 () [slot]

[GUI::showLevel4](#) starts [Level_4](#) and hide gui.

4.3.3.16 void GUI::triangle () [slot]

[GUI::triangle](#) triangle helpmenu.

create title text

4.3.4 Member Data Documentation

4.3.4.1 bool GUI::ismute = false

Check if Sound is on or off.

4.3.4.2 `picButton*` `GUI::mutepicButton`

Mutepicbutton for Sound.

4.3.4.3 `QGraphicsScene*` `GUI::scene`

Scene for [GUI](#).

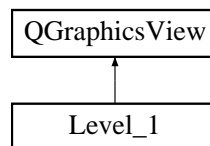
The documentation for this class was generated from the following files:

- [Game/gui.h](#)
- [Game/gui.cpp](#)

4.4 Level_1 Class Reference

```
#include <level_1.h>
```

Inheritance diagram for `Level_1`:



Public Slots

- void [update](#) ()
[Level_1::update](#) update function for moveable objects like our ball - sets the graphics of the ball to the position of the box2D body.
- void [startLevel](#) ()
[Level_1::startLevel](#) Set the flag of the QGraphicsItem, after start was clicked. draw the graphics if the body was moved before start was clicked.
- void [pauseLevel](#) ()
[Level_1::pauseLevel](#) pauses game when button pause is clicked.
- void [resumeLevel](#) ()
[Level_1::resumeLevel](#) resumes game when button resume is clicked.
- void [addRectangle](#) ()
[Level_1::addRectangle](#) Create new rectangle and count the rectangle items. limited to number.
- void [addCircle](#) ()
[Level_1::addCircle](#) create new circle and count the circle items. limited to number.
- void [reset](#) ()
[Level_1::reset](#) Clear scene and load Level again.
- void [closeLevel](#) ()
[Level_1::closeLevel](#) if QGraphicsView is closed emit Signal.
- void [rotateLeft](#) ()
[Level_1::rotateLeft](#) possibility to rotate objects to the left.
- void [rotateRight](#) ()
[Level_1::rotateRight](#) possibility to rotate right.
- void [getTime](#) ()
[Level_1::getTime](#) Stop time and convert it to ms.
- void [highscoreCounter](#) ()
[Level_1::highscoreCounter](#) Calculate the highscore.

Signals

- void [levelcompleted](#) ()

Public Member Functions

- [Level_1](#) ()
[Level_1::Level_1](#).

Public Attributes

- std::vector< [Block](#) * > [vectb](#)
- std::vector< [Triangle](#) * > [vectt](#)

4.4.1 Constructor & Destructor Documentation

4.4.1.1 [Level_1::Level_1](#) ()

[Level_1::Level_1](#).

Parameters

<i>parent</i>	Initialize Level1 - Screen/Scene Setup...
---------------	---

Set Application-Name

4.4.2 Member Function Documentation

4.4.2.1 void [Level_1::addCircle](#) () [slot]

[Level_1::addCircle](#) create new circle and count the circle items. limited to number.

4.4.2.2 void [Level_1::addRectangle](#) () [slot]

[Level_1::addRectangle](#) Create new rectangle and count the rectangle items. limited to number.

4.4.2.3 void [Level_1::closeLevel](#) () [slot]

[Level_1::closeLevel](#) if QGraphicsView is closed emit Signal.

4.4.2.4 void [Level_1::getTime](#) () [slot]

[Level_1::getTime](#) Stop time and convert it to ms.

4.4.2.5 void Level_1::highscoreCounter () [slot]

[Level_1::highscoreCounter](#) Calculate the highscore.

4.4.2.6 void Level_1::levelcompleted () [signal]

4.4.2.7 void Level_1::pauseLevel () [slot]

[Level_1::pauseLevel](#) pauses game when button pause is clicked.

4.4.2.8 void Level_1::reset () [slot]

[Level_1::reset](#) Clear scene and load Level again.

4.4.2.9 void Level_1::resumeLevel () [slot]

[Level_1::resumeLevel](#) resumes game when button resume is clicked.

4.4.2.10 void Level_1::rotateLeft () [slot]

[Level_1::rotateLeft](#) possibility to rotate objects to the left.

4.4.2.11 void Level_1::rotateRight () [slot]

[Level_1::rotateRight](#) possibility to rotate right.

4.4.2.12 void Level_1::startLevel () [slot]

[Level_1::startLevel](#) Set the flag of the QGraphicsItem, after start was clicked. draw the graphics if the body was moved before start was clicked.

4.4.2.13 void Level_1::update () [slot]

[Level_1::update](#) update function for moveable objects like our ball - sets the graphics of the ball to the position of the box2D body.

4.4.3 Member Data Documentation

4.4.3.1 std::vector<Block*> Level_1::vectb

4.4.3.2 std::vector<Triangle*> Level_1::vectt

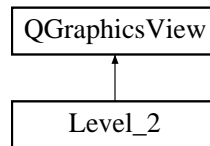
The documentation for this class was generated from the following files:

- [Game/level_1.h](#)
- [Game/level_1.cpp](#)

4.5 Level_2 Class Reference

```
#include <level_2.h>
```

Inheritance diagram for Level_2:



Public Slots

- void [update](#) ()
[Level_2::update](#) update function for moveable objects like our ball - sets the graphics of the ball to the position of the box2D body.
- void [startLevel](#) ()
[Level_2::startLevel](#) Set the flag of the QGraphicsItem, after start was clicked. draw the graphics if the body was moved before start was clicked.
- void [pauseLevel](#) ()
[Level_2::pauseLevel](#) pauses game when button pause is clicked.
- void [resumeLevel](#) ()
[Level_2::resumeLevel](#) resumes game when button resume is clicked.
- void [addRectangle](#) ()
[Level_2::addRectangle](#) Create new rectangle and count the rectangle items. limited to number.
- void [addCircle](#) ()
[Level_2::addCircle](#) Create new [Circle](#) and count the circle items. Limited to number.
- void [addTriangle](#) ()
[Level_2::addTriangle](#) Create new [Triangle](#) and count the triangle items. Limited to number.
- void [reset](#) ()
[Level_2::reset](#) Clear scene and load Level again.
- void [closeLevel](#) ()
[Level_2::closeLevel](#) if QGraphicsView is closed emit Signal.
- void [rotateLeft](#) ()
[Level_2::rotateLeft](#) possibility to rotate objects to the left.
- void [rotateRight](#) ()
[Level_2::rotateRight](#) possibility to rotate right.
- void [getTime](#) ()
[Level_2::getTime](#) Stop time and convert it to ms.
- void [highscoreCounter](#) ()
[Level_2::highscoreCounter](#) Calculate the highscore.

Signals

- void [levelcompleted](#) ()

Public Member Functions

- [Level_2](#) ()
[Level_2::Level_2](#) Initialize Level1 - Screen/Scene Setup...

4.5.1 Constructor & Destructor Documentation

4.5.1.1 [Level_2::Level_2](#) ()

[Level_2::Level_2](#) Initialize Level1 - Screen/Scene Setup...

Set Application-Name

4.5.2 Member Function Documentation

4.5.2.1 [void Level_2::addCircle](#) () [slot]

[Level_2::addCircle](#) Create new [Circle](#) and count the circle items. Limited to number.

4.5.2.2 [void Level_2::addRectangle](#) () [slot]

[Level_2::addRectangle](#) Create new rectangle and count the rectangle items. limited to number.

4.5.2.3 [void Level_2::addTriangle](#) () [slot]

[Level_2::addTriangle](#) Create new [Triangle](#) and count the triangle items. Limited to number.

4.5.2.4 [void Level_2::closeLevel](#) () [slot]

[Level_2::closeLevel](#) if QGraphicsView is closed emit Signal.

4.5.2.5 [void Level_2::getTime](#) () [slot]

[Level_2::getTime](#) Stop time and convert it to ms.

4.5.2.6 [void Level_2::highscoreCounter](#) () [slot]

[Level_2::highscoreCounter](#) Calculate the highscore.

4.5.2.7 [void Level_2::levelcompleted](#) () [signal]

4.5.2.8 [void Level_2::pauseLevel](#) () [slot]

[Level_2::pauseLevel](#) pauses game when button pause is clicked.

4.5.2.9 void Level_2::reset () [slot]

[Level_2::reset](#) Clear scene and load Level again.

4.5.2.10 void Level_2::resumeLevel () [slot]

[Level_2::resumeLevel](#) resumes game when button resume is clicked.

4.5.2.11 void Level_2::rotateLeft () [slot]

[Level_2::rotateLeft](#) possibility to rotate objects to the left.

4.5.2.12 void Level_2::rotateRight () [slot]

[Level_2::rotateRight](#) possibility to rotate right.

4.5.2.13 void Level_2::startLevel () [slot]

[Level_2::startLevel](#) Set the flag of the QGraphicsItem, after start was clicked. draw the graphics if the body was moved before start was clicked.

4.5.2.14 void Level_2::update () [slot]

[Level_2::update](#) update function for moveable objects like our ball - sets the graphics of the ball to the position of the box2D body.

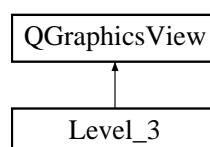
The documentation for this class was generated from the following files:

- [Game/level_2.h](#)
- [Game/level_2.cpp](#)

4.6 Level_3 Class Reference

```
#include <level_3.h>
```

Inheritance diagram for Level_3:



Public Slots

- void [update](#) ()
[Level_3::update](#) update function for moveable objects like our ball - sets the graphics of the ball to the position of the box2D body.
- void [startLevel](#) ()
[Level_3::startLevel](#) Set the flag of the QGraphicsItem, after start was clicked. draw the graphics if the body was moved before start was clicked.
- void [pauseLevel](#) ()
[Level_3::pauseLevel](#) pauses game when button pause is clicked.
- void [resumeLevel](#) ()
[Level_3::resumeLevel](#) resumes game when button resume is clicked.
- void [addRectangle](#) ()
[Level_3::addRectangle](#) Create new rectangle and count the rectangle items. limited to number.
- void [addCircle](#) ()
[Level_3::addCircle](#) Create new [Circle](#) and count the circle items. Limited to number.
- void [addTriangle](#) ()
[Level_3::addTriangle](#) Create new [Triangle](#) and count the triangle items. Limited to number.
- void [reset](#) ()
[Level_3::reset](#) Clear scene and load Level again.
- void [closeLevel](#) ()
[Level_3::closeLevel](#) if QGraphicsView is closed emit Signal.
- void [rotateLeft](#) ()
[Level_3::rotateLeft](#) possibility to rotate objects to the left.
- void [rotateRight](#) ()
[Level_3::rotateRight](#) possibility to rotate right.
- void [getTime](#) ()
[Level_3::getTime](#) Stop time and convert it to ms.
- void [highscoreCounter](#) ()
[Level_3::highscoreCounter](#) Calculate the highscore.

Signals

- void [levelcompleted](#) ()

Public Member Functions

- [Level_3](#) ()
[Level_3::Level_3](#) Initialize Level1 - Screen/Scene Setup...

4.6.1 Constructor & Destructor Documentation

4.6.1.1 [Level_3::Level_3](#) ()

[Level_3::Level_3](#) Initialize Level1 - Screen/Scene Setup...

Set Application-Name

4.6.2 Member Function Documentation

4.6.2.1 void Level_3::addCircle () [slot]

[Level_3::addCircle](#) Create new [Circle](#) and count the circle items. Limited to number.

4.6.2.2 void Level_3::addRectangle () [slot]

[Level_3::addRectangle](#) Create new rectangle and count the rectangle items. limited to number.

4.6.2.3 void Level_3::addTriangle () [slot]

[Level_3::addTriangle](#) Create new [Triangle](#) and count the triangle items. Limited to number.

4.6.2.4 void Level_3::closeLevel () [slot]

[Level_3::closeLevel](#) if QGraphicsView is closed emit Signal.

4.6.2.5 void Level_3::getTime () [slot]

[Level_3::getTime](#) Stop time and convert it to ms.

4.6.2.6 void Level_3::highscoreCounter () [slot]

[Level_3::highscoreCounter](#) Calculate the highscore.

4.6.2.7 void Level_3::levelcompleted () [signal]

4.6.2.8 void Level_3::pauseLevel () [slot]

[Level_3::pauseLevel](#) pauses game when button pause is clicked.

4.6.2.9 void Level_3::reset () [slot]

[Level_3::reset](#) Clear scene and load Level again.

4.6.2.10 void Level_3::resumeLevel () [slot]

[Level_3::resumeLevel](#) resumes game when button resume is clicked.

4.6.2.11 void Level_3::rotateLeft () [slot]

[Level_3::rotateLeft](#) possibility to rotate objects to the left.

4.6.2.12 void Level_3::rotateRight () [slot]

[Level_3::rotateRight](#) possibility to rotate right.

4.6.2.13 void Level_3::startLevel () [slot]

[Level_3::startLevel](#) Set the flag of the QGraphicsItem, after start was clicked. draw the graphics if the body was moved before start was clicked.

4.6.2.14 void Level_3::update () [slot]

[Level_3::update](#) update function for moveable objects like our ball - sets the graphics of the ball to the position of the box2D body.

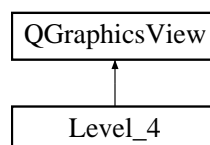
The documentation for this class was generated from the following files:

- [Game/level_3.h](#)
- [Game/level_3.cpp](#)

4.7 Level_4 Class Reference

```
#include <level_4.h>
```

Inheritance diagram for Level_4:



Public Slots

- void [update](#) ()
[Level_4::update](#) update function for moveable objects like our ball - sets the graphics of the ball to the position of the box2D body.
- void [startLevel](#) ()
[Level_4::startLevel](#) Set the flag of the QGraphicsItem, after start was clicked. draw the graphics if the body was moved before start was clicked.
- void [pauseLevel](#) ()
[Level_4::pauseLevel](#) pauses game when button pause is clicked.
- void [resumeLevel](#) ()
[Level_4::resumeLevel](#) resumes game when button resume is clicked.
- void [addRectangle](#) ()
[Level_4::addRectangle](#) Create new rectangle and count the rectangle items. limited to number.
- void [addCircle](#) ()
[Level_4::addCircle](#) Create new [Circle](#) and count the circle items. Limited to number.
- void [addTriangle](#) ()
[Level_4::addTriangle](#) Create new [Triangle](#) and count the triangle items. Limited to number.
- void [reset](#) ()
[Level_4::reset](#) Clear scene and load Level again.
- void [closeLevel](#) ()
[Level_4::closeLevel](#) if QGraphicsView is closed emit Signal.
- void [rotateLeft](#) ()
[Level_4::rotateLeft](#) possibility to rotate objects to the left.
- void [rotateRight](#) ()
[Level_4::rotateRight](#) possibility to rotate right.
- void [getTime](#) ()
[Level_4::getTime](#) Stop time and convert it to ms.
- void [highscoreCounter](#) ()
[Level_4::highscoreCounter](#) Calculate the highscore.

Signals

- void [levelcompleted](#) ()

Public Member Functions

- [Level_4](#) ()
[Level_4::Level_4](#) Initialize Level1 - Screen/Scene Setup...

4.7.1 Constructor & Destructor Documentation

4.7.1.1 [Level_4::Level_4](#) ()

[Level_4::Level_4](#) Initialize Level1 - Screen/Scene Setup...

Set Application-Name

4.7.2 Member Function Documentation

4.7.2.1 void Level_4::addCircle () [slot]

[Level_4::addCircle](#) Create new [Circle](#) and count the circle items. Limited to number.

4.7.2.2 void Level_4::addRectangle () [slot]

[Level_4::addRectangle](#) Create new rectangle and count the rectangle items. limited to number.

4.7.2.3 void Level_4::addTriangle () [slot]

[Level_4::addTriangle](#) Create new [Triangle](#) and count the triangle items. Limited to number.

4.7.2.4 void Level_4::closeLevel () [slot]

[Level_4::closeLevel](#) if QGraphicsView is closed emit Signal.

4.7.2.5 void Level_4::getTime () [slot]

[Level_4::getTime](#) Stop time and convert it to ms.

4.7.2.6 void Level_4::highscoreCounter () [slot]

[Level_4::highscoreCounter](#) Calculate the highscore.

4.7.2.7 void Level_4::levelcompleted () [signal]

4.7.2.8 void Level_4::pauseLevel () [slot]

[Level_4::pauseLevel](#) pauses game when button pause is clicked.

4.7.2.9 void Level_4::reset () [slot]

[Level_4::reset](#) Clear scene and load Level again.

4.7.2.10 void Level_4::resumeLevel () [slot]

[Level_4::resumeLevel](#) resumes game when button resume is clicked.

4.7.2.11 `void Level_4::rotateLeft () [slot]`

[Level_4::rotateLeft](#) possibility to rotate objects to the left.

4.7.2.12 `void Level_4::rotateRight () [slot]`

[Level_4::rotateRight](#) possibility to rotate right.

4.7.2.13 `void Level_4::startLevel () [slot]`

[Level_4::startLevel](#) Set the flag of the QGraphicsItem, after start was clicked. draw the graphics if the body was moved before start was clicked.

4.7.2.14 `void Level_4::update () [slot]`

[Level_4::update](#) update function for moveable objects like our ball - sets the graphics of the ball to the position of the box2D body.

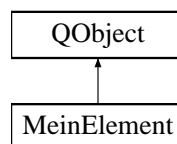
The documentation for this class was generated from the following files:

- [Game/level_4.h](#)
- [Game/level_4.cpp](#)

4.8 MeinElement Class Reference

```
#include <meinelement.h>
```

Inheritance diagram for MeinElement:



Public Member Functions

- [MeinElement](#) (b2World *world, QGraphicsScene *level, b2Vec2 center, qreal length, qreal width, b2BodyType type, qreal friction)
[MeinElement::MeinElement.](#)
- [MeinElement](#) (QGraphicsScene *level, QPointF center, qreal length, qreal width)
[MeinElement::MeinElement.](#)
- void [drawBottom](#) ()

Public Attributes

- b2Body * [body](#)
Box2D Body of object.
- QGraphicsItem * [graphics](#)
graphic of object
- QGraphicsItem * [white](#)
graphic of box, where 'Finished' message is written.
- QMediaPlayer * [applause](#)
Soundplayer for successfully finished level.

4.8.1 Constructor & Destructor Documentation

4.8.1.1 **MeinElement::MeinElement** (b2World * *world*, QGraphicsScene * *level*, b2Vec2 *center*, qreal *length*, qreal *width*, b2BodyType *type*, qreal *friction*)

[MeinElement::MeinElement.](#)

Parameters

<i>world</i>	: Box2D world for physic engine
<i>level</i>	: Scene for the game
<i>center</i>	: center of the object
<i>length</i>	: length of the object
<i>width</i>	: Breite of the object
<i>type</i>	: Box2D type of the Bblock(if it's static or dynamic)
<i>friction</i>	: friction for the Block

4.8.1.2 **MeinElement::MeinElement** (QGraphicsScene * *level*, QPointF *center*, qreal *length*, qreal *width*)

[MeinElement::MeinElement.](#)

Parameters

<i>level</i>	: Scene for the game
<i>center</i>	: Center of object (coordinates)
<i>length</i>	: length of object
<i>width</i>	: length of object Message Box Background for 'Finished Level'

4.8.2 Member Function Documentation

4.8.2.1 **void MeinElement::drawBottom** ()

4.8.3 Member Data Documentation

4.8.3.1 QMediaPlayer* MeinElement::applause

Soundplayer for successfully finished level.

4.8.3.2 b2Body* MeinElement::body

Box2D Body of object.

4.8.3.3 QGraphicsItem* MeinElement::graphics

graphic of object

4.8.3.4 QGraphicsItem* MeinElement::white

graphic of box, where 'Finished' message is written.

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

- [Game/meinelement.h](#)
- [Game/meinelement.cpp](#)

4.9 Mover Class Reference

```
#include <mover.h>
```

Public Member Functions

- [Mover](#) (b2World *world, QGraphicsScene *level, b2Vec2 center, qreal m_angle, qreal m_length, qreal m_width, qreal friction, QString mode)
[Mover::Mover.](#)

Public Attributes

- qreal [length](#)
length of object
- qreal [width](#)
width of object
- qreal [angle](#)
angle of object
- b2Body * [body](#)
body of object
- QGraphicsItem * [graphics](#)
graphics of object

4.9.1 Constructor & Destructor Documentation

- 4.9.1.1 [Mover::Mover](#) (b2World * world, QGraphicsScene * level, b2Vec2 center, qreal m_angle, qreal m_length, qreal m_width, qreal friction, QString mode)

[Mover::Mover.](#)

Parameters

<i>world</i>	
<i>level</i>	
<i>center</i>	
<i>m_angle</i>	
<i>m_length</i>	
<i>m_width</i>	
<i>type</i>	
<i>friction</i>	
<i>mode</i>	

4.9.2 Member Data Documentation

4.9.2.1 qreal Mover::angle

angle of object

4.9.2.2 b2Body* Mover::body

body of object

4.9.2.3 QGraphicsItem* Mover::graphics

graphics of object

4.9.2.4 qreal Mover::length

length of object

4.9.2.5 qreal Mover::width

width of object

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

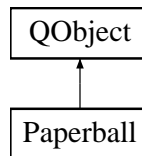
- Game/[mover.h](#)
- Game/[mover.cpp](#)

4.10 Paperball Class Reference

The [Paperball](#) class.

```
#include <paperball.h>
```

Inheritance diagram for Paperball:



Public Member Functions

- [Paperball](#) (b2World *world, QGraphicsScene *level, QPointF position, qreal angle, b2BodyType type, b2CircleShape &circle)
[Paperball::Paperball](#).
- void [createPaper](#) (b2World world, QGraphicsScene levelscene, QPointF pos, qreal angle, b2BodyType type, b2CircleShape &circle)
- bool [drawBall1](#) ()
[Paperball::drawBall1](#) connects the Graphics to the Box2D-Object checkout if paperball fell into the recyclebin.

Public Attributes

- b2Body * [body](#)
Box2D Body of Object.
- QGraphicsItem * [graphics](#)
Graphic of Object.

4.10.1 Detailed Description

The [Paperball](#) class.

4.10.2 Constructor & Destructor Documentation

- 4.10.2.1 [Paperball::Paperball](#) (b2World * *world*, QGraphicsScene * *level*, QPointF *position*, qreal *angle*, b2BodyType *type*, b2CircleShape & *circle*)

[Paperball::Paperball](#).

Parameters

<i>world</i>	
<i>level</i>	
<i>position</i>	
<i>angle</i>	
<i>type</i>	
<i>circle</i>	

4.10.3 Member Function Documentation

4.10.3.1 void Paperball::createPaper (b2World *world*, QGraphicsScene *levelscene*, QPointF *pos*, qreal *angle*, b2BodyType *type*, b2CircleShape & *circle*)

4.10.3.2 bool Paperball::drawBall1 ()

[Paperball::drawBall1](#) connects the Graphics to the Box2D-Object checkout if paperball fell into the recyclebin.

Returns

4.10.4 Member Data Documentation

4.10.4.1 b2Body* Paperball::body

Box2D Body of Object.

4.10.4.2 QGraphicsItem* Paperball::graphics

Graphic of Object.

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

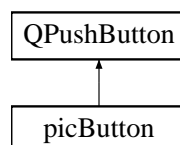
- [Game/paperball.h](#)
- [Game/paperball.cpp](#)

4.11 picButton Class Reference

The [picButton](#) class.

```
#include <picbutton.h>
```

Inheritance diagram for picButton:



Public Member Functions

- [picButton](#) (QPixmap _defaultpic, QPixmap _hoverpic)
[picButton::picButton](#)
- [picButton](#) (QPixmap _defaultpic, QPixmap _hoverpic, bool [hover](#))
[picButton::picButton](#)
- void [enterEvent](#) (QEvent *event)
[picButton::enterEvent](#)
- void [leaveEvent](#) (QEvent *event)
[picButton::leaveEvent](#)
- void [setdefaultpic](#) (QPixmap [defaultpic](#))
[picButton::setdefaultpic](#)
- void [sethoverpic](#) (QPixmap [hoverpic](#))
[picButton::sethoverpic](#)

Public Attributes

- QPixmap [defaultpic](#)
Picture of Button.
- QPixmap [hoverpic](#)
Picture when Button is selected.
- bool [hover](#)

4.11.1 Detailed Description

The [picButton](#) class.

4.11.2 Constructor & Destructor Documentation

4.11.2.1 [picButton::picButton](#) (QPixmap _defaultpic, QPixmap _hoverpic)

[picButton::picButton](#)

Parameters

<i>_defaultpic</i>	
<i>_hoverpic</i>	

Hovering mouse

4.11.2.2 [picButton::picButton](#) (QPixmap _defaultpic, QPixmap _hoverpic, bool _hover)

[picButton::picButton](#)

Parameters

<i>_defaultpic</i>	picture of button
<i>_hoverpic</i>	picture if you hover over button
<i>_hover</i>	does mouse hover

Hovering mouse

4.11.3 Member Function Documentation

4.11.3.1 void picButton::enterEvent (QEvent * *event*)

[picButton::enterEvent](#)

Parameters

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

when hovering -> change pic

4.11.3.2 void picButton::leaveEvent (QEvent * *event*)

[picButton::leaveEvent](#)

Parameters

<i>event</i>	mouse leave butto area
--------------	------------------------

Change pic back

4.11.3.3 void picButton::setDefaultpic (QPixmap *_defaultpic*)

[picButton::setDefaultpic](#)

Parameters

<i>_defaultpic</i>	insert picutre
--------------------	----------------

4.11.3.4 void picButton::sethoverpic (QPixmap *_hoverpic*)

[picButton::sethoverpic](#)

Parameters

<i>_hoverpic</i>	insert picture
------------------	----------------

4.11.4 Member Data Documentation

4.11.4.1 QPixmap picButton::defaultpic

Picture of Button.

4.11.4.2 bool picButton::hover

4.11.4.3 QPixmap picButton::hoverpic

Picture when Button is selected.

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

- [Game/picbutton.h](#)
- [Game/picbutton.cpp](#)

4.12 RecycleBin Class Reference

The [RecycleBin](#) class.

```
#include <recyclebin.h>
```

Public Member Functions

- [RecycleBin](#) (b2World *world, QGraphicsScene *level, QPointF a, QPointF b, QPointF c, QPointF d, qreal angle, b2BodyType type)
[RecycleBin::RecycleBin](#).
- void [drawGraphics](#) ()
[RecycleBin::drawGraphics](#) connects the Box2D-Object to the Graphics after relocation.

Public Attributes

- b2Body * [body](#)
Box2D Body of Object.
- QGraphicsItem * [graphics](#)
Graphic of Object.

4.12.1 Detailed Description

The [RecycleBin](#) class.

4.12.2 Constructor & Destructor Documentation

- ##### 4.12.2.1 RecycleBin::RecycleBin (b2World * world, QGraphicsScene * level, QPointF a, QPointF b, QPointF c, QPointF d, qreal angle, b2BodyType type)

[RecycleBin::RecycleBin](#).

Parameters

<i>world</i>	
<i>level</i>	
<i>a</i>	
<i>b</i>	
<i>c</i>	
<i>d</i>	
<i>angle</i>	
<i>type</i>	

4.12.3 Member Function Documentation

4.12.3.1 void RecycleBin::drawGraphics ()

[RecycleBin::drawGraphics](#) connects the Box2D-Object to the Graphics after relocation.

4.12.4 Member Data Documentation

4.12.4.1 b2Body* RecycleBin::body

Box2D Body of Object.

4.12.4.2 QGraphicsItem* RecycleBin::graphics

Graphic of Object.

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

- Game/[recyclebin.h](#)
- Game/[recyclebin.cpp](#)

4.13 RecycleBinGraphics Class Reference

The [RecycleBinGraphics](#) class.

```
#include <recyclebingraphics.h>
```

Public Member Functions

- [RecycleBinGraphics](#) (QGraphicsScene *level)
[RecycleBinGraphics::RecycleBinGraphics.](#)

Public Attributes

- `QGraphicsItem *` [graphics](#)
Graphic of Recyclebinobject.

4.13.1 Detailed Description

The [RecycleBinGraphics](#) class.

4.13.2 Constructor & Destructor Documentation

4.13.2.1 `RecycleBinGraphics::RecycleBinGraphics (QGraphicsScene * level)`

[RecycleBinGraphics::RecycleBinGraphics.](#)

Parameters

<i>level</i>	
--------------	--

4.13.3 Member Data Documentation

4.13.3.1 `QGraphicsItem* RecycleBinGraphics::graphics`

Graphic of Recyclebinobject.

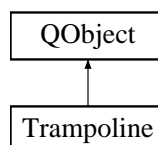
The documentation for this class was generated from the following files:

- `Game/recyclebingraphics.h`
- `Game/recyclebingraphics.cpp`

4.14 Trampoline Class Reference

```
#include <trampoline.h>
```

Inheritance diagram for Trampoline:



Public Member Functions

- [Trampoline](#) (b2World *world, QGraphicsScene *level, b2Vec2 center, qreal m_angle, qreal m_length, qreal m_width, b2BodyType type, qreal friction, QString mode)

[Trampoline::Trampoline.](#)

Public Attributes

- qreal [length](#)
length of object
- qreal [width](#)
width of object
- qreal [angle](#)
angle of object
- b2Body * [body](#)
body of object
- QGraphicsItem * [graphics](#)
graphics of object

4.14.1 Constructor & Destructor Documentation

- 4.14.1.1 [Trampoline::Trampoline](#) (b2World * *world*, QGraphicsScene * *level*, b2Vec2 *center*, qreal *m_angle*, qreal *m_length*, qreal *m_width*, b2BodyType *type*, qreal *friction*, QString *mode*)

[Trampoline::Trampoline.](#)

Parameters

<i>world</i>	
<i>level</i>	
<i>center</i>	
<i>m_angle</i>	
<i>m_length</i>	
<i>m_width</i>	
<i>type</i>	
<i>friction</i>	
<i>mode</i>	

4.14.2 Member Data Documentation

- 4.14.2.1 qreal [Trampoline::angle](#)

angle of object

- 4.14.2.2 b2Body* [Trampoline::body](#)

body of object

4.14.2.3 QGraphicsItem* Trampoline::graphics

graphics of object

4.14.2.4 qreal Trampoline::length

length of object

4.14.2.5 qreal Trampoline::width

width of object

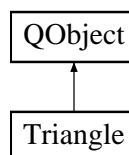
The documentation for this class was generated from the following files:

- [Game/trampoline.h](#)
- [Game/trampoline.cpp](#)

4.15 Triangle Class Reference

```
#include <triangle.h>
```

Inheritance diagram for Triangle:



Public Member Functions

- [Triangle](#) (b2World *world, QGraphicsScene *level, QPointF a, QPointF b, QPointF c, qreal angle, b2BodyType type, qreal friction, QString mode)
[Triangle::Triangle.](#)
- void [drawGraphics](#) ()
[Triangle::drawGraphics](#) connects the Box2D-Object to the Graphics after relocation.

Public Attributes

- b2Body * [body](#)
body of triangle
- QGraphicsItem * [graphics](#)
graphic of triangle

4.15.1 Constructor & Destructor Documentation

- 4.15.1.1 [Triangle::Triangle](#) (b2World * *world*, QGraphicsScene * *level*, QPointF *a*, QPointF *b*, QPointF *c*, qreal *angle*, b2BodyType *type*, qreal *friction*, QString *mode*)

[Triangle::Triangle.](#)

Parameters

<i>world</i>	box2d world
<i>level</i>	scene of qt
<i>a</i>	left upper corner, default setting if rotate calculate new position of a
<i>b</i>	right upper corner, default setting if rotate calculate new position of b
<i>c</i>	right lower corner, default setting if rotate calculate new position of c
<i>angle</i>	box2d angle
<i>type</i>	static or dynamic object
<i>friction</i>	box2d friction

4.15.2 Member Function Documentation

4.15.2.1 void Triangle::drawGraphics ()

[Triangle::drawGraphics](#) connects the Box2D-Object to the Graphics after relocation.

4.15.3 Member Data Documentation

4.15.3.1 b2Body* Triangle::body

body of triangle

4.15.3.2 QGraphicsItem* Triangle::graphics

graphic of triangle

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

- [Game/triangle.h](#)
- [Game/triangle.cpp](#)

Chapter 5

File Documentation

5.1 Game/block.cpp File Reference

```
#include "block.h"  
#include <QGraphicsScene>  
#include <QPoint>  
#include <QSize>  
#include <qdebug.h>
```

5.2 Game/block.h File Reference

```
#include "Box2D/Box2D.h"  
#include "QGraphicsItem"  
#include <QPointF>  
#include "meinelement.h"
```

Classes

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

5.3 Game/circle.cpp File Reference

```
#include "circle.h"  
#include "Box2D/Box2D.h"  
#include <QGraphicsScene>  
#include <QDebug>
```

5.4 Game/circle.h File Reference

```
#include "Box2D/Box2D.h"  
#include "QGraphicsItem"  
#include <QPointF>  
#include "meinelement.h"
```

Classes

- class [Circle](#)

5.5 Game/gui.cpp File Reference

```
#include "gui.h"  
#include <QFile>  
#include <QGraphicsItem>  
#include <QPushButton>  
#include <QSound>  
#include <QRect>  
#include "level_1.h"  
#include "level_2.h"  
#include "level_3.h"  
#include "level_4.h"  
#include "qdebug.h"
```

5.6 Game/gui.h File Reference

```
#include <QGraphicsScene>  
#include <QGraphicsView>  
#include <QMediaPlayer>  
#include <QMediaPlaylist>  
#include <picbutton.h>
```

Classes

- class [GUI](#)

The [GUI](#) class.

5.7 Game/level_1.cpp File Reference

```
#include "level_1.h"
#include <iostream>
#include <QTime>
#include <QTimer>
#include <QElapsedTimer>
#include <qdebug.h>
#include <QFile>
#include <QTextStream>
#include "string"
#include "gui.h"
#include <QtWidgets>
```

5.8 Game/level_1.h File Reference

```
#include "Box2D/Box2D.h"
#include <QMainWindow>
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QTimer>
#include "meinelement.h"
#include "triangle.h"
#include <QPushButton>
#include <QGraphicsSceneMouseEvent>
#include <QElapsedTimer>
#include <QTime>
#include "recyclebin.h"
#include "recyclebingraphics.h"
#include "circle.h"
#include "gui.h"
#include <QItemSelection>
#include "paperball.h"
#include <QMediaPlayer>
#include "block.h"
```

Classes

- class [Level_1](#)

Macros

- #define [framerate](#) 1.0/35.0

5.8.1 Macro Definition Documentation

5.8.1.1 #define framerate 1.0/35.0

5.9 Game/level_2.cpp File Reference

```
#include "level_2.h"
#include <iostream>
#include <QTime>
#include <QTimer>
#include <QElapsedTimer>
#include <qdebug.h>
#include <QFile>
#include <QTextStream>
#include "string"
#include "trampoline.h"
```

5.10 Game/level_2.h File Reference

```
#include "Box2D/Box2D.h"
#include <QMainWindow>
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QTimer>
#include "meinelement.h"
#include "triangle.h"
#include <QPushButton>
#include <QGraphicsSceneMouseEvent>
#include <QElapsedTimer>
#include <QTime>
#include "recyclebin.h"
#include "recyclebingraphics.h"
#include "circle.h"
#include "gui.h"
#include <QItemSelection>
#include <paperball.h>
#include "picbutton.h"
#include "block.h"
#include "trampoline.h"
```

Classes

- class [Level_2](#)

Macros

- #define [framerate](#) 1.0/35.0

5.10.1 Macro Definition Documentation

5.10.1.1 #define framerate 1.0/35.0

5.11 Game/level_3.cpp File Reference

```
#include "level_3.h"
#include <iostream>
#include <QTime>
#include <QTimer>
#include <QElapsedTimer>
#include <qdebug.h>
#include <QFile>
#include <QTextStream>
#include "string"
```

5.12 Game/level_3.h File Reference

```
#include "Box2D/Box2D.h"
#include <QMainWindow>
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QTimer>
#include "meinelement.h"
#include "triangle.h"
#include <QPushButton>
#include <QGraphicsSceneMouseEvent>
#include <QElapsedTimer>
#include <QTime>
#include "recyclebin.h"
#include "recyclebingraphics.h"
#include "circle.h"
#include "gui.h"
#include <QItemSelection>
#include <paperball.h>
#include "picbutton.h"
#include "trampoline.h"
#include "block.h"
```

Classes

- class [Level_3](#)

Macros

- #define [framerate](#) 1.0/35.0

5.12.1 Macro Definition Documentation

5.12.1.1 #define framerate 1.0/35.0

5.13 Game/level_4.cpp File Reference

```
#include "level_4.h"
#include <iostream>
#include <QTime>
#include <QTimer>
#include <QElapsedTimer>
#include <qdebug.h>
#include <QFile>
#include <QTextStream>
#include "string"
```

5.14 Game/level_4.h File Reference

```
#include "Box2D/Box2D.h"
#include <QMainWindow>
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QTimer>
#include "meinelement.h"
#include "triangle.h"
#include <QPushButton>
#include <QGraphicsSceneMouseEvent>
#include <QElapsedTimer>
#include <QTime>
#include "recyclebin.h"
#include "recyclebingraphics.h"
#include "circle.h"
#include "gui.h"
#include <QItemSelection>
#include <paperball.h>
#include "picbutton.h"
#include "trampoline.h"
#include "block.h"
```

Classes

- class [Level_4](#)

Macros

- #define [framerate](#) 1.0/35.0

5.14.1 Macro Definition Documentation

5.14.1.1 #define framerate 1.0/35.0

5.15 Game/main.cpp File Reference

```
#include <QApplication>
#include "gui.h"
```

Functions

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

5.15.1 Function Documentation

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

5.16 Game/meinelement.cpp File Reference

```
#include "meinelement.h"
#include <QGraphicsScene>
#include <QPoint>
#include <QSize>
#include <qdebug.h>
```

5.17 Game/meinelement.h File Reference

```
#include "Box2D/Box2D.h"
#include "QGraphicsItem"
#include <QMediaPlayer>
#include <QPointF>
```

Classes

- class [MeinElement](#)

5.18 Game/mover.cpp File Reference

```
#include "mover.h"
#include <QGraphicsScene>
#include <QPoint>
#include <QSize>
#include <qdebug.h>
```

5.19 Game/mover.h File Reference

```
#include "Box2D/Box2D.h"  
#include "QGraphicsItem"  
#include <QPointF>  
#include "meinelement.h"
```

Classes

- class [Mover](#)

5.20 Game/paperball.cpp File Reference

```
#include "paperball.h"  
#include "Box2D/Box2D.h"  
#include <QGraphicsScene>
```

5.21 Game/paperball.h File Reference

```
#include "Box2D/Box2D.h"  
#include "QGraphicsItem"  
#include <QPointF>  
#include "meinelement.h"
```

Classes

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

5.22 Game/picbutton.cpp File Reference

```
#include "picbutton.h"
```

5.23 Game/picbutton.h File Reference

```
#include <QObject>  
#include <QPushButton>  
#include <QGraphicsItem>  
#include <QPixmap>  
#include <QGraphicsSceneMouseEvent>
```

Classes

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

5.24 Game/recyclebin.cpp File Reference

```
#include "recyclebin.h"
#include <QGraphicsScene>
#include <QPoint>
#include <QSize>
#include <qdebug.h>
```

5.25 Game/recyclebin.h File Reference

```
#include "Box2D/Box2D.h"
#include "QGraphicsItem"
#include <QPointF>
```

Classes

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

5.26 Game/recyclebingraphics.cpp File Reference

```
#include "recyclebingraphics.h"
#include <QGraphicsScene>
#include <QPoint>
#include <QSize>
#include <qdebug.h>
```

5.27 Game/recyclebingraphics.h File Reference

```
#include "Box2D/Box2D.h"
#include "QGraphicsItem"
#include <QPointF>
```

Classes

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

5.28 Game/trampoline.cpp File Reference

```
#include "trampoline.h"  
#include <QGraphicsScene>  
#include <QPoint>  
#include <QSize>  
#include <qdebug.h>
```

5.29 Game/trampoline.h File Reference

```
#include "Box2D/Box2D.h"  
#include "QGraphicsItem"  
#include <QPointF>  
#include "meinelement.h"
```

Classes

- class [Trampoline](#)

5.30 Game/triangle.cpp File Reference

```
#include "triangle.h"  
#include <QGraphicsScene>  
#include <QPoint>  
#include <QSize>  
#include <qdebug.h>
```

5.31 Game/triangle.h File Reference

```
#include "Box2D/Box2D.h"  
#include "QGraphicsItem"  
#include <QPointF>  
#include "meinelement.h"
```

Classes

- class [Triangle](#)

Index

- addCircle
 - Level_1, [16](#)
 - Level_2, [19](#)
 - Level_3, [22](#)
 - Level_4, [25](#)
- addRectangle
 - Level_1, [16](#)
 - Level_2, [19](#)
 - Level_3, [22](#)
 - Level_4, [25](#)
- addTriangle
 - Level_2, [19](#)
 - Level_3, [22](#)
 - Level_4, [25](#)
- angle
 - Block, [8](#)
 - Mover, [29](#)
 - Trampoline, [37](#)
- applause
 - MeinElement, [27](#)
- back
 - GUI, [12](#)
- Block, [7](#)
 - angle, [8](#)
 - Block, [8](#)
 - body, [8](#)
 - drawGraphics, [8](#)
 - graphics, [8](#)
 - length, [8](#)
 - width, [9](#)
- body
 - Block, [8](#)
 - Circle, [10](#)
 - MeinElement, [28](#)
 - Mover, [29](#)
 - Paperball, [31](#)
 - RecycleBin, [35](#)
 - Trampoline, [37](#)
 - Triangle, [39](#)
- box
 - GUI, [12](#)
- checkLevel
 - GUI, [13](#)
- Circle, [9](#)
 - body, [10](#)
 - Circle, [9](#)
 - createCircle, [10](#)
 - draw, [10](#)
 - drawBall1, [10](#)
 - drawGraphics, [10](#)
 - graphics, [10](#)
- circle
 - GUI, [13](#)
- closeLevel
 - Level_1, [16](#)
 - Level_2, [19](#)
 - Level_3, [22](#)
 - Level_4, [25](#)
- createCircle
 - Circle, [10](#)
- createPaper
 - Paperball, [31](#)
- csnd
 - GUI, [13](#)
- defaultpic
 - picButton, [34](#)
- displayGUI
 - GUI, [13](#)
- draw
 - Circle, [10](#)
- drawBall1
 - Circle, [10](#)
 - Paperball, [31](#)
- drawBottom
 - MeinElement, [27](#)
- drawGraphics
 - Block, [8](#)
 - Circle, [10](#)
 - RecycleBin, [35](#)
 - Triangle, [39](#)
- enterEvent
 - picButton, [33](#)
- framerate
 - level_1.h, [44](#)
 - level_2.h, [45](#)
 - level_3.h, [46](#)
 - level_4.h, [47](#)
- GUI, [11](#)
 - back, [12](#)
 - box, [12](#)
 - checkLevel, [13](#)
 - circle, [13](#)
 - csnd, [13](#)
 - displayGUI, [13](#)

- GUI, 12
- help, 13
- highscore, 13
- ismute, 14
- levelMenu, 13
- mute, 14
- mutepicButton, 14
- scene, 15
- showGuiagain, 14
- showLevel1, 14
- showLevel2, 14
- showLevel3, 14
- showLevel4, 14
- triangle, 14
- Game/block.cpp, 41
- Game/block.h, 41
- Game/circle.cpp, 41
- Game/circle.h, 42
- Game/gui.cpp, 42
- Game/gui.h, 42
- Game/level_1.cpp, 43
- Game/level_1.h, 43
- Game/level_2.cpp, 44
- Game/level_2.h, 44
- Game/level_3.cpp, 45
- Game/level_3.h, 45
- Game/level_4.cpp, 46
- Game/level_4.h, 46
- Game/main.cpp, 47
- Game/meinelement.cpp, 47
- Game/meinelement.h, 47
- Game/mover.cpp, 47
- Game/mover.h, 48
- Game/paperball.cpp, 48
- Game/paperball.h, 48
- Game/picbutton.cpp, 48
- Game/picbutton.h, 48
- Game/recyclebin.cpp, 49
- Game/recyclebin.h, 49
- Game/recyclebingraphics.cpp, 49
- Game/recyclebingraphics.h, 49
- Game/trampoline.cpp, 50
- Game/trampoline.h, 50
- Game/triangle.cpp, 50
- Game/triangle.h, 50
- getTime
 - Level_1, 16
 - Level_2, 19
 - Level_3, 22
 - Level_4, 25
- graphics
 - Block, 8
 - Circle, 10
 - MeinElement, 28
 - Mover, 29
 - Paperball, 31
 - RecycleBin, 35
 - RecycleBinGraphics, 36
- Trampoline, 37
- Triangle, 39
- help
 - GUI, 13
- highscore
 - GUI, 13
- highscoreCounter
 - Level_1, 16
 - Level_2, 19
 - Level_3, 22
 - Level_4, 25
- hover
 - picButton, 34
- hoverpic
 - picButton, 34
- ismute
 - GUI, 14
- leaveEvent
 - picButton, 33
- length
 - Block, 8
 - Mover, 29
 - Trampoline, 38
- Level_1, 15
 - addCircle, 16
 - addRectangle, 16
 - closeLevel, 16
 - getTime, 16
 - highscoreCounter, 16
 - Level_1, 16
 - levelcompleted, 17
 - pauseLevel, 17
 - reset, 17
 - resumeLevel, 17
 - rotateLeft, 17
 - rotateRight, 17
 - startLevel, 17
 - update, 17
 - vectb, 17
 - vectt, 17
- level_1.h
 - framerate, 44
- Level_2, 18
 - addCircle, 19
 - addRectangle, 19
 - addTriangle, 19
 - closeLevel, 19
 - getTime, 19
 - highscoreCounter, 19
 - Level_2, 19
 - levelcompleted, 19
 - pauseLevel, 19
 - reset, 19
 - resumeLevel, 20
 - rotateLeft, 20
 - rotateRight, 20

- startLevel, 20
- update, 20
- level_2.h
 - framerate, 45
- Level_3, 20
 - addCircle, 22
 - addRectangle, 22
 - addTriangle, 22
 - closeLevel, 22
 - getTime, 22
 - highscoreCounter, 22
 - Level_3, 21
 - levelcompleted, 22
 - pauseLevel, 22
 - reset, 22
 - resumeLevel, 22
 - rotateLeft, 22
 - rotateRight, 23
 - startLevel, 23
 - update, 23
- level_3.h
 - framerate, 46
- Level_4, 23
 - addCircle, 25
 - addRectangle, 25
 - addTriangle, 25
 - closeLevel, 25
 - getTime, 25
 - highscoreCounter, 25
 - Level_4, 24
 - levelcompleted, 25
 - pauseLevel, 25
 - reset, 25
 - resumeLevel, 25
 - rotateLeft, 25
 - rotateRight, 26
 - startLevel, 26
 - update, 26
- level_4.h
 - framerate, 47
- levelMenu
 - GUI, 13
- levelcompleted
 - Level_1, 17
 - Level_2, 19
 - Level_3, 22
 - Level_4, 25
- main
 - main.cpp, 47
- main.cpp
 - main, 47
- MeinElement, 26
 - applause, 27
 - body, 28
 - drawBottom, 27
 - graphics, 28
 - MeinElement, 27
 - white, 28
- Mover, 28
 - angle, 29
 - body, 29
 - graphics, 29
 - length, 29
 - Mover, 28
 - width, 29
- mute
 - GUI, 14
- mutepicButton
 - GUI, 14
- Paperball, 30
 - body, 31
 - createPaper, 31
 - drawBall1, 31
 - graphics, 31
 - Paperball, 30
- pauseLevel
 - Level_1, 17
 - Level_2, 19
 - Level_3, 22
 - Level_4, 25
- picButton, 31
 - defaultpic, 34
 - enterEvent, 33
 - hover, 34
 - hoverpic, 34
 - leaveEvent, 33
 - picButton, 32
 - setDefaultpic, 33
 - sethoverpic, 33
- RecycleBin, 34
 - body, 35
 - drawGraphics, 35
 - graphics, 35
 - RecycleBin, 34
- RecycleBinGraphics, 35
 - graphics, 36
 - RecycleBinGraphics, 36
- reset
 - Level_1, 17
 - Level_2, 19
 - Level_3, 22
 - Level_4, 25
- resumeLevel
 - Level_1, 17
 - Level_2, 20
 - Level_3, 22
 - Level_4, 25
- rotateLeft
 - Level_1, 17
 - Level_2, 20
 - Level_3, 22
 - Level_4, 25
- rotateRight
 - Level_1, 17
 - Level_2, 20

- Level_3, [23](#)
- Level_4, [26](#)
- scene
 - GUI, [15](#)
- setDefaultpic
 - picButton, [33](#)
- sethoverpic
 - picButton, [33](#)
- showGuiagain
 - GUI, [14](#)
- showLevel1
 - GUI, [14](#)
- showLevel2
 - GUI, [14](#)
- showLevel3
 - GUI, [14](#)
- showLevel4
 - GUI, [14](#)
- startLevel
 - Level_1, [17](#)
 - Level_2, [20](#)
 - Level_3, [23](#)
 - Level_4, [26](#)
- Trampoline, [36](#)
 - angle, [37](#)
 - body, [37](#)
 - graphics, [37](#)
 - length, [38](#)
 - Trampoline, [37](#)
 - width, [38](#)
- Triangle, [38](#)
 - body, [39](#)
 - drawGraphics, [39](#)
 - graphics, [39](#)
 - Triangle, [38](#)
- triangle
 - GUI, [14](#)
- update
 - Level_1, [17](#)
 - Level_2, [20](#)
 - Level_3, [23](#)
 - Level_4, [26](#)
- vectb
 - Level_1, [17](#)
- vectt
 - Level_1, [17](#)
- white
 - MeinElement, [28](#)
- width
 - Block, [9](#)
 - Mover, [29](#)
 - Trampoline, [38](#)