

Group 3 Project

Generated by Doxygen 1.7.6.1

Fri Apr 7 2017 22:21:24

Contents

1	EECE 435L Games Project	1
2	Class 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	Class Documentation	11
5.1	Barn Class Reference	11
5.1.1	Constructor & Destructor Documentation	11
5.1.1.1	Barn	11
5.1.1.2	~Barn	12
5.1.2	Member Function Documentation	12
5.1.2.1	sheepIn	12
5.2	Box Class Reference	12
5.2.1	Constructor & Destructor Documentation	13
5.2.1.1	Box	13
5.2.1.2	~Box	13
5.2.2	Member Function Documentation	13
5.2.2.1	drawBitzer	13
5.2.2.2	drawShaun	13
5.2.2.3	isClosed	13
5.2.2.4	numberOfLinesDrawn	14

5.3	Cannon Class Reference	14
5.3.1	Constructor & Destructor Documentation	14
5.3.1.1	Cannon	14
5.3.1.2	~Cannon	14
5.3.2	Member Function Documentation	15
5.3.2.1	keyPressEvent	15
5.4	Dot Class Reference	15
5.4.1	Constructor & Destructor Documentation	15
5.4.1.1	Dot	15
5.4.1.2	~Dot	15
5.5	Game1 Class Reference	16
5.5.1	Constructor & Destructor Documentation	16
5.5.1.1	Game1	16
5.5.1.2	~Game1	17
5.5.2	Member Function Documentation	17
5.5.2.1	endGame	17
5.5.2.2	goToMainMenu	17
5.5.2.3	loadNewGame	17
5.5.2.4	next	17
5.5.2.5	replay	18
5.6	Game1Options Class Reference	18
5.6.1	Constructor & Destructor Documentation	18
5.6.1.1	Game1Options	18
5.6.1.2	~Game1Options	19
5.6.2	Member Function Documentation	19
5.6.2.1	getNumberOfUnlockedLevels	19
5.6.2.2	gotoGame1	19
5.6.2.3	gotoGameMainMenu	19
5.6.2.4	unlockExtraLevel	19
5.7	Game1Scene Class Reference	20
5.7.1	Constructor & Destructor Documentation	20
5.7.1.1	Game1Scene	21
5.7.1.2	~Game1Scene	21
5.7.2	Member Function Documentation	21

5.7.2.1	collidesWithSheepInLine	21
5.7.2.2	fireSheep	21
5.7.2.3	gameOver	21
5.7.2.4	getScore	22
5.7.2.5	mousePressEvent	22
5.7.2.6	move_line	22
5.7.2.7	moveCurrentSheep	22
5.8	Game2 Class Reference	22
5.8.1	Constructor & Destructor Documentation	23
5.8.1.1	Game2	23
5.8.1.2	~Game2	23
5.8.2	Member Function Documentation	23
5.8.2.1	endGame	23
5.8.2.2	goToMainMenu	24
5.8.2.3	replay	24
5.9	Game2Options Class Reference	24
5.9.1	Constructor & Destructor Documentation	24
5.9.1.1	Game2Options	24
5.9.1.2	~Game2Options	25
5.9.2	Member Function Documentation	25
5.9.2.1	gotoGame2	25
5.9.2.2	gotoGameMainMenu	25
5.10	Game2Scene Class Reference	25
5.10.1	Constructor & Destructor Documentation	27
5.10.1.1	Game2Scene	27
5.10.1.2	~Game2Scene	27
5.10.2	Member Function Documentation	27
5.10.2.1	computerTurn	27
5.10.2.2	decrementScore	27
5.10.2.3	gameOver	27
5.10.2.4	getBlockCount	27
5.10.2.5	getNeighbors	28
5.10.2.6	getNonBlockedBorders	28
5.10.2.7	getScoreDisplay	28

5.10.2.8	getSheep	28
5.10.2.9	getUserTurn	29
5.10.2.10	moveSheep	29
5.10.2.11	placeSheepInitial	29
5.10.2.12	placeTiles	29
5.10.2.13	resetDistances	29
5.10.2.14	resetPrevious	29
5.10.2.15	resetVisited	29
5.10.2.16	setUserTurn	30
5.10.2.17	tileAt	30
5.10.2.18	tilesToBlock	30
5.10.2.19	win	30
5.11	Game3 Class Reference	31
5.11.1	Constructor & Destructor Documentation	31
5.11.1.1	Game3	31
5.11.1.2	~Game3	32
5.11.2	Member Function Documentation	32
5.11.2.1	endGame	32
5.11.2.2	goToMainMenu	32
5.11.2.3	replay	32
5.12	Game3Options Class Reference	32
5.12.1	Constructor & Destructor Documentation	33
5.12.1.1	Game3Options	33
5.12.1.2	~Game3Options	33
5.12.2	Member Function Documentation	33
5.12.2.1	gotoGame	33
5.12.2.2	gotoGameMainMenu	34
5.12.2.3	setEasy	34
5.12.2.4	setHard	34
5.12.2.5	setModerate	34
5.12.2.6	setSizeEight	34
5.12.2.7	setSizeFour	34
5.12.2.8	setSizeSixteen	34
5.13	Game3Scene Class Reference	35

5.13.1	Constructor & Destructor Documentation	35
5.13.1.1	Game3Scene	36
5.13.1.2	~Game3Scene	36
5.13.2	Member Function Documentation	36
5.13.2.1	addNewlyDrawnLine	36
5.13.2.2	clearNewLines	36
5.13.2.3	computerMove	36
5.13.2.4	computerTurn	36
5.13.2.5	gameOver	37
5.13.2.6	getLineThatClosesBox	37
5.13.2.7	getSmartLine	37
5.13.2.8	isUserTurn	37
5.13.2.9	noMoreMoves	37
5.14	GameMainMenu Class Reference	38
5.14.1	Constructor & Destructor Documentation	38
5.14.1.1	GameMainMenu	38
5.14.1.2	~GameMainMenu	38
5.14.2	Member Function Documentation	39
5.14.2.1	gotoGameOptions	39
5.14.2.2	gotoGameSelection	39
5.15	GameOver Class Reference	39
5.15.1	Constructor & Destructor Documentation	39
5.15.1.1	GameOver	39
5.15.1.2	~GameOver	40
5.16	GameSelection Class Reference	40
5.16.1	Constructor & Destructor Documentation	40
5.16.1.1	GameSelection	41
5.16.1.2	~GameSelection	41
5.16.2	Member Function Documentation	41
5.16.2.1	goToGame1	41
5.16.2.2	goToGame2	41
5.16.2.3	goToGame3	41
5.16.2.4	goToMain	41
5.16.2.5	goToMyAccount	41

5.17	Helper Class Reference	42
5.17.1	Member Function Documentation	42
5.17.1.1	makeWidgetLarge	42
5.17.1.2	makeWidgetSmall	42
5.17.1.3	toRadians	43
5.18	HorizontalLine Class Reference	43
5.18.1	Constructor & Destructor Documentation	44
5.18.1.1	HorizontalLine	44
5.18.1.2	~HorizontalLine	44
5.18.2	Member Function Documentation	44
5.18.2.1	getAbove	44
5.18.2.2	getUnder	44
5.18.2.3	playTurn	45
5.18.2.4	turnGrey	45
5.19	Line Class Reference	45
5.19.1	Constructor & Destructor Documentation	46
5.19.1.1	Line	46
5.19.2	Member Function Documentation	46
5.19.2.1	isDrawn	46
5.19.2.2	isHorizontal	46
5.19.2.3	mousePressEvent	47
5.19.2.4	playTurn	47
5.20	MainWidget Class Reference	47
5.20.1	Constructor & Destructor Documentation	48
5.20.1.1	MainWidget	48
5.20.1.2	~MainWidget	48
5.21	MyAccount Class Reference	48
5.21.1	Constructor & Destructor Documentation	48
5.21.1.1	MyAccount	48
5.21.1.2	~MyAccount	49
5.21.2	Member Function Documentation	49
5.21.2.1	goToGames	49
5.22	ShaunGamesTest Class Reference	49
5.23	Sheep1 Class Reference	50

5.23.1	Constructor & Destructor Documentation	50
5.23.1.1	Sheep1	51
5.23.1.2	~Sheep1	51
5.23.2	Member Function Documentation	51
5.23.2.1	fire	51
5.23.2.2	firedMove	51
5.23.2.3	getAngle	51
5.23.2.4	getNumber	52
5.23.2.5	getRandomSheepNumber	52
5.23.2.6	inLineDistanceTo	52
5.23.2.7	isInLine	52
5.23.2.8	moveInLine	52
5.23.2.9	setAngle	53
5.23.2.10	setInLine	53
5.24	Sheep2 Class Reference	53
5.24.1	Constructor & Destructor Documentation	53
5.24.1.1	Sheep2	54
5.24.2	Member Function Documentation	54
5.24.2.1	getCurrent	54
5.24.2.2	setCurrent	54
5.25	Tile Class Reference	54
5.25.1	Constructor & Destructor Documentation	55
5.25.1.1	Tile	55
5.25.1.2	~Tile	55
5.25.2	Member Function Documentation	56
5.25.2.1	getCol	56
5.25.2.2	getDistance	56
5.25.2.3	getPrev	56
5.25.2.4	getRow	56
5.25.2.5	isBlocked	57
5.25.2.6	isBorder	57
5.25.2.7	isVisited	57
5.25.2.8	mousePressEvent	57
5.25.2.9	setBlock	57

5.25.2.10	setDistance	58
5.25.2.11	setHasSheep	58
5.25.2.12	setPrev	58
5.25.2.13	setVisited	58
5.26	VerticalLine Class Reference	59
5.26.1	Constructor & Destructor Documentation	59
5.26.1.1	VerticalLine	59
5.26.1.2	~VerticalLine	60
5.26.2	Member Function Documentation	60
5.26.2.1	getLeft	60
5.26.2.2	getRight	60
5.26.2.3	playTurn	60
5.26.2.4	turnGrey	61
6	File Documentation	63
6.1	difficulty.h File Reference	63
6.1.1	Detailed Description	63
6.2	game1/barn.cpp File Reference	63
6.2.1	Detailed Description	64
6.3	game1/barn.h File Reference	64
6.3.1	Detailed Description	64
6.4	game1/cannon.cpp File Reference	64
6.4.1	Detailed Description	64
6.5	game1/cannon.h File Reference	64
6.5.1	Detailed Description	65
6.6	game1/game1.cpp File Reference	65
6.6.1	Detailed Description	65
6.7	game1/game1.h File Reference	65
6.7.1	Detailed Description	65
6.8	game1/game1options.cpp File Reference	66
6.8.1	Detailed Description	66
6.9	game1/game1options.h File Reference	66
6.9.1	Detailed Description	66
6.10	game1/game1scene.cpp File Reference	66

6.10.1 Detailed Description	66
6.11 game1/game1scene.h File Reference	67
6.11.1 Detailed Description	67
6.12 game1/sheep1.cpp File Reference	67
6.12.1 Detailed Description	67
6.13 game1/sheep1.h File Reference	67
6.13.1 Detailed Description	68
6.14 game2/game2.cpp File Reference	68
6.14.1 Detailed Description	68
6.15 game2/game2.h File Reference	68
6.15.1 Detailed Description	68
6.16 game2/game2options.cpp File Reference	69
6.16.1 Detailed Description	69
6.17 game2/game2options.h File Reference	69
6.17.1 Detailed Description	69
6.18 game2/game2scene.cpp File Reference	69
6.18.1 Detailed Description	69
6.19 game2/game2scene.h File Reference	70
6.19.1 Detailed Description	70
6.20 game2/sheep2.cpp File Reference	70
6.20.1 Detailed Description	70
6.21 game2/tile.cpp File Reference	70
6.21.1 Detailed Description	70
6.22 game2/tile.h File Reference	71
6.22.1 Detailed Description	71
6.23 game3/box.cpp File Reference	71
6.23.1 Detailed Description	71
6.24 game3/box.h File Reference	71
6.24.1 Detailed Description	72
6.25 game3/dot.cpp File Reference	72
6.25.1 Detailed Description	72
6.26 game3/dot.h File Reference	72
6.26.1 Detailed Description	72
6.27 game3/game3.cpp File Reference	73

6.27.1 Detailed Description	73
6.28 game3/game3.h File Reference	73
6.28.1 Detailed Description	73
6.29 game3/game3options.cpp File Reference	73
6.29.1 Detailed Description	73
6.30 game3/game3options.h File Reference	74
6.30.1 Detailed Description	74
6.31 game3/game3scene.cpp File Reference	74
6.31.1 Detailed Description	74
6.32 game3/game3scene.h File Reference	74
6.32.1 Detailed Description	75
6.33 game3/horizontalline.cpp File Reference	75
6.33.1 Detailed Description	75
6.34 game3/horizontalline.h File Reference	75
6.34.1 Detailed Description	75
6.35 game3/line.cpp File Reference	76
6.35.1 Detailed Description	76
6.36 game3/line.h File Reference	76
6.36.1 Detailed Description	76
6.37 game3/size.h File Reference	76
6.37.1 Detailed Description	77
6.38 game3/verticalline.cpp File Reference	77
6.38.1 Detailed Description	77
6.39 game3/verticalline.h File Reference	77
6.39.1 Detailed Description	77
6.40 gameover.cpp File Reference	78
6.40.1 Detailed Description	78
6.41 gameover.h File Reference	78
6.41.1 Detailed Description	78
6.42 gui/gamemainmenu.cpp File Reference	78
6.42.1 Detailed Description	78
6.43 gui/gamemainmenu.h File Reference	79
6.43.1 Detailed Description	79
6.44 gui/gameselection.cpp File Reference	79

6.44.1 Detailed Description	79
6.45 gui/gameselection.h File Reference	79
6.45.1 Detailed Description	80
6.46 gui/mainwidget.cpp File Reference	80
6.46.1 Detailed Description	80
6.47 gui/mainwidget.h File Reference	80
6.47.1 Detailed Description	80
6.48 helper.cpp File Reference	81
6.48.1 Detailed Description	81
6.49 helper.h File Reference	81
6.49.1 Detailed Description	81
6.50 myaccount.cpp File Reference	81
6.50.1 Detailed Description	81
6.51 myaccount.h File Reference	82
6.51.1 Detailed Description	82

Chapter 1

EECE 435L Games Project

Author

Rita Aoun
Rawan Moukalled

Date

07-04-2017

Runs the application.

Chapter 2

Class Index

2.1 Class Hierarchy

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

Barn	11
Box	12
Cannon	14
Dot	15
Game1	16
Game1Options	18
Game1Scene	20
Game2	22
Game2Options	24
Game2Scene	25
Game3	31
Game3Options	32
Game3Scene	35
GameMainMenu	38
GameOver	39
GameSelection	40
Helper	42
Line	45
HorizontalLine	43
VerticalLine	59
MainWidget	47
MyAccount	48
ShaunGamesTest	49
Sheep1	50
Sheep2	53
Tile	54

Chapter 3

Class Index

3.1 Class List

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

Barn	11
Box	12
Cannon	14
Dot	15
Game1	16
Game1Options	18
Game1Scene	20
Game2	22
Game2Options	24
Game2Scene	25
Game3	31
Game3Options	32
Game3Scene	35
GameMainMenu	38
GameOver	39
GameSelection	40
Helper	42
HorizontalLine	43
Line	45
MainWidget	47
MyAccount	48
ShaunGamesTest	49
Sheep1	50
Sheep2	53
Tile	54
VerticalLine	59

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

difficulty.h	Difficulty enum	63
gameover.cpp	Contains GameOver class definition	78
gameover.h	Game Over class	78
helper.cpp	Contains Helper class definition	81
helper.h	Helper class	81
myaccount.cpp	Contains MyAccount class definition	81
myaccount.h	Class representing the my account and performance history win- dows	82
game1/barn.cpp	Contains Barn class definition	63
game1/barn.h	Barn class	64
game1/cannon.cpp	Contains Cannon class definition	64
game1/cannon.h	Cannon class	64
game1/game1.cpp	Contains the Sheep Line	65
game1/game1.h	Sheep Line class	65
game1/game1options.cpp	Contains Game1Options class definition	66

game1/ game1options.h	
Game1Options class	66
game1/ game1scene.cpp	
Contains Game1Scene class definition	66
game1/ game1scene.h	
Sheep Line class	67
game1/ sheep1.cpp	
Contains Sheep1 class definition	67
game1/ sheep1.h	
Sheep1 class	67
game2/ game2.cpp	
Contains Game2 class definition	68
game2/ game2.h	
Trap the Sheep class	68
game2/ game2options.cpp	
Contains Game2Options class definition	69
game2/ game2options.h	
Game2Options class	69
game2/ game2scene.cpp	
Contains Game2Scene class definition	69
game2/ game2scene.h	
Trap the Sheep scene class	70
game2/ sheep2.cpp	
Contains Sheep class definition	70
game2/ sheep2.h	??
game2/ tile.cpp	
Contains Tile class definition	70
game2/ tile.h	
Class for the tiles of game 2	71
game3/ box.cpp	
Contains Box class definition	71
game3/ box.h	
Box class	71
game3/ dot.cpp	
Contains Dot class definition	72
game3/ dot.h	
Dot class	72
game3/ game3.cpp	
Contains the Dots and Lines game	73
game3/ game3.h	
Dots and Lines class	73
game3/ game3options.cpp	
Contains Game3Options class definition	73
game3/ game3options.h	
Game3Options class	74
game3/ game3scene.cpp	
Contains Game3Scene class definition	74
game3/ game3scene.h	
Game3Scene class	74

game3/horizontalline.cpp	
Contains HorizontalLine class definition	75
game3/horizontalline.h	
HorizontalLine class	75
game3/line.cpp	
Contains Line class definition	76
game3/line.h	
Line class	76
game3/size.h	
Size enum	76
game3/verticalline.cpp	
Contains VerticalLine class definition	77
game3/verticalline.h	
VerticalLine class	77
gui/gamemainmenu.cpp	
Contains GameMainMenu class definition	78
gui/gamemainmenu.h	
GameMainMenu class	79
gui/gameselection.cpp	
Contains GameSelection class definition	79
gui/gameselection.h	
Game selection menu class	79
gui/mainwidget.cpp	
Contains MainWidget class definition	80
gui/mainwidget.h	
MainWidget class	80
Tests/ ShaunGamesTest.h	??

Chapter 5

Class Documentation

5.1 Barn Class Reference

Public Slots

- void `sheepIn` ()

Triggers the end of the game once a sheep collides with the barn.

Public Member Functions

- `Barn` (QObject *parent=0)

Default constructor.

- virtual `~Barn` ()

Destructor.

5.1.1 Constructor & Destructor Documentation

5.1.1.1 `Barn::Barn (QObject * parent = 0) [explicit]`

Default constructor.

Sets the barn image and timer to check for collisions

5.1.1.2 `Barn::~~Barn () [virtual]`

Destructor.

Frees allocated memory.

5.1.2 Member Function Documentation

5.1.2.1 void Barn::sheepIn () [slot]

Triggers the end of the game once a sheep collides with the barn.

Called by the timer, checks if there are colliding items with the barn. If the sheep is part of the moving line, stop the game. Otherwise, the sheep was shot and the game proceeds normally.

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

- game1/[barn.h](#)
- game1/[barn.cpp](#)

5.2 Box Class Reference

Public Member Functions

- [Box](#) (QObject *parent=0)
Default constructor.
- virtual [~Box](#) ()
Destructor.
- void [drawShaun](#) ()
Sets pixmap to Shaun.
- void [drawBitzer](#) ()
Sets pixmap to Bitzer.
- void [setAbove](#) ()
Marks that the top of the box has been drawn.
- void [setLeft](#) ()
Marks that the left of the box has been drawn.
- void [setUnder](#) ()
Marks that the bottom of the box has been drawn.
- void [setRight](#) ()
Marks that the right of the box has been drawn.
- bool [isClosed](#) ()
Checks if the box has been closed.
- int [numberOfLinesDrawn](#) ()
Checks how many lines are drawn in the box.

5.2.1 Constructor & Destructor Documentation

5.2.1.1 Box::Box (QObject * parent = 0) [explicit]

Default constructor.

Sets [Box](#) properties.

5.2.1.2 `Box::~Box ()` [virtual]

Destructor.

Frees allocated memory.

5.2.2 Member Function Documentation

5.2.2.1 `void Box::drawBitzer ()`

Sets pixmap to Bitzer.

Draws Bitzer on the box.

5.2.2.2 `void Box::drawShaun ()`

Sets pixmap to Shaun.

Draws Shaun on the box.

5.2.2.3 `bool Box::isClosed ()`

Checks if the box has been closed.

Returns

Whether the box has been closed

5.2.2.4 `int Box::numberOfLinesDrawn ()`

Checks how many lines are drawn in the box.

Returns

The number of lines drawn

Checks if the box is one line away from being closed.

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

- [game3/box.h](#)
- [game3/box.cpp](#)

5.3 Cannon Class Reference

Public Member Functions

- [Cannon](#) (QObject *parent=0)

Default constructor.

- virtual [~Cannon](#) ()

Destructor.

- void [keyPressEvent](#) (QKeyEvent *event)

Entrance point of triggered key events.

5.3.1 Constructor & Destructor Documentation

5.3.1.1 Cannon::Cannon (QObject *parent = 0) [explicit]

Default constructor.

Sets the cannonimage and initializes variables.

5.3.1.2 Cannon::~~Cannon () [virtual]

Destructor.

Frees allocated memory.

5.3.2 Member Function Documentation

5.3.2.1 void Cannon::keyPressEvent (QKeyEvent * event)

Entrance point of triggered key events.

Parameters

<i>event</i>	The event that has been triggered
--------------	-----------------------------------

Checks the key that triggered the event. If the key was a left or right arrow key, the cannon rotates left or right. If the key was a space, a sheep is thrown.

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

- game1/[cannon.h](#)
- game1/[cannon.cpp](#)

5.4 Dot Class Reference

Public Member Functions

- [Dot](#) (QObject *parent=0)

Default constructor.

- virtual [~Dot](#) ()

Destructor.

5.4.1 Constructor & Destructor Documentation

5.4.1.1 `Dot::Dot (QObject * parent = 0) [explicit]`

Default constructor.

Sets [Dot](#) properties.

5.4.1.2 `Dot::~Dot() [virtual]`

Destructor.

Frees allocated memory.

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

- [game3/dot.h](#)
- [game3/dot.cpp](#)

5.5 Game1 Class Reference

Public Slots

- void [goToMainMenu](#) ()
Slot to go back to the games main menu when pressing Exit.
- void [endGame](#) (bool win)
Slot to handle ending the game once it's over.
- void [replay](#) ()
Reloads the game with the same level.
- void [next](#) ()
Proceed to the next level.

Public Member Functions

- [Game1](#) (int level, QWidget *parent=0)
Constructor.
- virtual [~Game1](#) ()
Destructor.
- void [loadNewGame](#) (bool sameLevel)
Load new game.

5.5.1 Constructor & Destructor Documentation

5.5.1.1 **Game1::Game1** (int *level*, QWidget * *parent* = 0) [explicit]

Constructor.

Parameters

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

Sets the size of the window, initializes the graphic items, sets the layouts and connects buttons to their slots.

5.5.1.2 **Game1::~~Game1** () [virtual]

Destructor.

Frees allocated memory

5.5.2 Member Function Documentation

5.5.2.1 **void Game1::endGame** (bool *win*) [slot]

Slot to handle ending the game once it's over.

Parameters

<i>win</i>	Indicates if the user has won the game
------------	--

Removes the save and exit button and adds the go back and replay buttons along with their connections

5.5.2.2 **void Game1::goToMainMenu** () [slot]

Slot to go back to the games main menu when pressing Exit.

Goes to the main menu of Sheep [Line](#)

5.5.2.3 **void Game1::loadNewGame** (bool *sameLevel*)

Load new game.

Parameters

<i>sameLevel</i>	Indicates if the level is the same
------------------	------------------------------------

Loads a new game of either the same level or the next

5.5.2.4 void Game1::next () [slot]

Proceed to the next level.

Proceed to the next level

5.5.2.5 void Game1::replay () [slot]

Reloads the game with the same level.

Loads a new instance of the [Game1](#) Scene

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

- [game1/game1.h](#)
- [game1/game1.cpp](#)

5.6 Game1Options Class Reference

Public Slots

- void [gotoGameMainMenu](#) ()
Takes the user to the game main menu.
- void [gotoGame1](#) (int level)
Takes the user to game 1.

Public Member Functions

- [Game1Options](#) (QWidget *parent=0)
Default constructor.
- virtual [~Game1Options](#) ()
Destructor.

Static Public Member Functions

- static int [getNumberOfUnlockedLevels](#) ()
Returns the number of levels in game 1 that the user has unlocked.
- static void [unlockExtraLevel](#) (int currLevel)
Unlocks one new game 1 level if appropriate.

5.6.1 Constructor & Destructor Documentation

5.6.1.1 `Game1Options::Game1Options (QWidget * parent = 0)` `[explicit]`

Default constructor.

Initializes all buttons and text and shows them on the screen. Also initializes connections.

5.6.1.2 `Game1Options::~~Game1Options ()` `[virtual]`

Destructor.

Frees allocated memory

5.6.2 Member Function Documentation

5.6.2.1 `int Game1Options::getNumberOfUnlockedLevels ()` `[static]`

Returns the number of levels in game 1 that the user has unlocked.

Returns

Number of levels

Returns the number of levels that the user has unlocked.

5.6.2.2 `void Game1Options::gotoGame1 (int level)` `[slot]`

Takes the user to game 1.

Parameters

<i>level</i>	The level of the game
--------------	-----------------------

Takes the user to game 1. Called after clicking any level button.

5.6.2.3 `void Game1Options::gotoGameMainMenu ()` `[slot]`

Takes the user to the game main menu.

Takes the user to the game main menu that corresponds to game 1. Called after clicking the corresponding button.

5.6.2.4 `void Game1Options::unlockExtraLevel (int currLevel)` `[static]`

Unlocks one new game 1 level if appropriate.

Parameters

<i>currLevel</i>	Level that was just won
------------------	-------------------------

Unlocks a new level if the user has just won the last unlocked level. Called after a user wins a certain level.

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

- [game1/game1options.h](#)
- [game1/game1options.cpp](#)

5.7 Game1Scene Class Reference

Public Slots

- void [move_line](#) ()
Move sheep in the line according to a straight line then circle.

Signals

- void [Done](#) (bool)
Signals [Game1](#) that the game is over.

Public Member Functions

- [Game1Scene](#) (int level, QObject *parent=0)
Constructor.
- virtual [~Game1Scene](#) ()
Destructor.
- void [mousePressEvent](#) (QGraphicsSceneMouseEvent *)
Adjusts focus.
- void [moveCurrentSheep](#) (bool toRight)
Moves sheep with rotating cannon.
- void [gameOver](#) (bool win)
Stops movement of the sheep and triggers finishing the game.
- void [fireSheep](#) ()
Fires the sheep.
- bool [collidesWithSheepInLine](#) (QGraphicsItem *item)
Checks the collision list of the item for a sheep in the sheep list.
- int [getScore](#) () const
Returns the current player score.

5.7.1 Constructor & Destructor Documentation

5.7.1.1 `Game1Scene::Game1Scene (int level, QObject * parent = 0) [explicit]`

Constructor.

Parameters

<i>level</i>	Level of the game
--------------	-------------------

Initializes variables and connections.

5.7.1.2 `Game1Scene::~Game1Scene () [virtual]`

Destructor.

Frees allocated memory.

5.7.2 Member Function Documentation

5.7.2.1 `bool Game1Scene::collidesWithSheepInLine (QGraphicsItem * item)`

Checks the collision list of the item for a sheep in the sheep list.

Parameters

<i>item</i>	Pointer to item to check
-------------	--------------------------

Returns

Whether the given item collides with a sheep in the list

Returns whether the given item collides with a sheep in the line.

5.7.2.2 `void Game1Scene::fireSheep ()`

Fires the sheep.

Releases the sheep and makes it move in a straight line.

5.7.2.3 `void Game1Scene::gameOver (bool win)`

Stops movement of the sheep and triggers finishing the game.

Parameters

<i>win</i>	Indicates if the game has been won
------------	------------------------------------

Ends the game

5.7.2.4 int Game1Scene::getScore () const

Returns the current player score.

Returns

Current player score

Returns the player score for the current game. For each destroyed sheep, the player is awarded 10 points. When the player wins the game, they are awarded `m_scoreDisplay->display(m_score)`; 1 point per remaining sheep in-line move.

5.7.2.5 void Game1Scene::mousePressEvent (QGraphicsSceneMouseEvent *)

Adjusts focus.

Sets focus on the cannon

5.7.2.6 void Game1Scene::move_line () [slot]

Move sheep in the line according to a straight line then circle.

Moves the sheep in the line according to their position on the screen

5.7.2.7 void Game1Scene::moveCurrentSheep (bool toRight)

Moves sheep with rotating cannon.

Parameters

<i>toRight</i>	Indicates the direction of the cannon move
----------------	--

Moves current sheep. Called when the cannon rotates.

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

- [game1/game1scene.h](#)
- [game1/game1scene.cpp](#)

5.8 Game2 Class Reference

Public Slots

- void [goToMainMenu](#) ()

Slot to go back to the games main menu when pressing Exit.

- void `endGame` ()

removes save and exit button and adds replay and back buttons

- void `replay` ()

removes save and exit button and adds replay and back buttons

Public Member Functions

- `Game2` (Difficulty difficulty, QWidget *parent=0)

Default constructor.

- virtual `~Game2` ()

Destructor.

5.8.1 Constructor & Destructor Documentation

5.8.1.1 `Game2::Game2 (Difficulty difficulty, QWidget * parent = 0) [explicit]`

Default constructor.

Sets the size of the window, initializes the graphic items, sets the layouts and connects buttons to their slots.

5.8.1.2 `Game2::~~Game2 () [virtual]`

Destructor.

Frees allocated memory.

5.8.2 Member Function Documentation

5.8.2.1 `void Game2::endGame () [slot]`

removes save and exit button and adds replay and back buttons

Removes the save and exitbutton and adds the go back and replay buttons along with their connections.

5.8.2.2 `void Game2::goToMainMenu () [slot]`

Slot to go back to the games main menu when pressing Exit.

Goes to the main menu of Trap the Sheep.

5.8.2.3 void Game2::replay () [slot]

removes save and exit button and adds replay and back buttons

Reloads game.

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

- [game2/game2.h](#)
- [game2/game2.cpp](#)

5.9 Game2Options Class Reference

Public Slots

- void [gotoGameMainMenu](#) ()
Takes the user to the game main menu.
- void [gotoGame2](#) (int difficulty)
Takes the user to game 2.

Public Member Functions

- [Game2Options](#) (QWidget *parent=0)
Constructor.
- virtual [~Game2Options](#) ()
Destructor.

5.9.1 Constructor & Destructor Documentation

5.9.1.1 Game2Options::Game2Options (QWidget * *parent* = 0) [explicit]

Constructor.

Initializes all buttons and text and shows them on the screen. Also initializes connections.

5.9.1.2 Game2Options::~~Game2Options () [virtual]

Destructor.

Frees allocated memory.

5.9.2 Member Function Documentation

5.9.2.1 void Game2Options::gotoGame2 (int *difficulty*) [slot]

Takes the user to game 2.

Takes the user to game 2. Called after clicking any level button.

5.9.2.2 void Game2Options::gotoGameMainMenu () [slot]

Takes the user to the game main menu.

Takes the user to the game main menu that corresponds to game 2. Called after clicking the corresponding button.

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

- [game2/game2options.h](#)
- [game2/game2options.cpp](#)

5.10 Game2Scene Class Reference

Public Slots

- void [moveSheep](#) ()
moves the sheep according to the difficulty of the game

Signals

- void [Done](#) ()
Signal sent to game2 to show that the game has ended.

Public Member Functions

- [Game2Scene](#) (Difficulty difficulty, QObject *parent=0)
Default constructor.
- virtual [~Game2Scene](#) ()
- QVector< int > [tilesToBlock](#) ()
Sets the layout of the level buttons.
- void [placeTiles](#) ()
Places the tiles to create the full grid.
- void [placeSheepInitial](#) ()
places the sheep on a random unblocked tile at the beginning of the game
- QVector< [Tile](#) * > * [getNeighbors](#) ([Tile](#) *center)
Gets the left, right, upper and lower non-blocked neighbors of the given tile.

- void [resetVisited](#) ()
Sets all the blocks of the grid as having been unvisited.
- void [resetDistances](#) ()
Sets all the blocks of the grid as having infinite distance to the sheep.
- void [resetPrevious](#) ()
Sets all the blocks of the grid as having the previous node null.
- [Tile](#) * [tileAt](#) (int i, int j)
retrieves the tile from the grid at the indices given
- bool [win](#) ([Tile](#) *tile)
determines if the user successfully trapped the sheep
- [Sheep2](#) * [getSheep](#) ()
gets the sheep of the game
- bool [getUserTurn](#) ()
retrieves whether or not it is the user's turn
- void [setUserTurn](#) (bool userTurn)
sets the turn of the user
- void [computerTurn](#) ()
delays the computer turn
- void [gameOver](#) (bool [win](#))
Ends the game.
- void [decrementScore](#) ()
Increments the number of blocks on click of a tile.
- [QLCDNumber](#) * [getScoreDisplay](#) ()
retrieves the lcd display
- int [getBlockCount](#) ()
retrieves the number of blocked tiles
- [Tile](#) * [findNextTile](#) ()
Finds the shortest path and returns the next tile accordingly.
- void [computeDistances](#) ([Tile](#) *current)
Computes the distances from the sheep to every other tile.
- [QVector](#)< [Tile](#) * > * [getNonBlockedBorders](#) ()
Retrieves the non blocked border tiles.

5.10.1 Constructor & Destructor Documentation

5.10.1.1 [Game2Scene::Game2Scene](#) ([Difficulty](#) difficulty, [QObject](#) * parent = 0) [explicit]

Default constructor.

Places the items on the scene and sets the user turn.

5.10.1.2 `Game2Scene::~Game2Scene ()` [virtual]

Destructor

Frees allocated memory.

5.10.2 Member Function Documentation

5.10.2.1 `void Game2Scene::computerTurn ()`

delays the computer turn

Starts the timer to delay the computer move.

5.10.2.2 `void Game2Scene::decrementScore ()`

Increments the number of blocks on click of a tile.

Increments the number of blocks by one on click of a tile.

5.10.2.3 `void Game2Scene::gameOver (bool win)`

Ends the game.

Parameters

<i>win</i>	whether or not the user won
------------	-----------------------------

Ends the game and displays the [GameOver](#) item.

5.10.2.4 `int Game2Scene::getBlockCount ()`

retrieves the number of blocked tiles

Returns

number of blocked tiles

Returns the number of blocked tiles.

5.10.2.5 `QVector< Tile * > * Game2Scene::getNeighbors (Tile * center)`

Gets the left, right, upper and lower non-blocked neighbors of the given tile.

Parameters

<i>center</i>	The tile that we're getting the neighbors of
---------------	--

Returns

Pointer to a vector of pointers to the neighboring tiles

Gets the left, right, upper and bottom neighbors for the given tile. Memory is allocated in this function and should be freed by the caller.

5.10.2.6 QVector< Tile * > * Game2Scene::getNonBlockedBorders ()

Retrieves the non blocked border tiles.

Retrieves the non blocked border tiles. Memory is allocated in this function and should be freed by the caller.

5.10.2.7 QLCDNumber * Game2Scene::getScoreDisplay ()

retrieves the lcd display

Returns

the lcd display

Returns the score lcd.

5.10.2.8 Sheep2 * Game2Scene::getSheep ()

gets the sheep of the game

Returns

the sheep of the game

Retrieves the sheep.

5.10.2.9 bool Game2Scene::getUserTurn ()

retrieves whether or not it is the user's turn

Returns

boolean indicating if it's the user's turn

Retrieves the value that indicates if it's the user's turn.

5.10.2.10 void Game2Scene::moveSheep () [slot]

moves the sheep according to the difficulty of the game

Moves the sheep according to the difficulty when it's the computer's turn.

5.10.2.11 void Game2Scene::placeSheepInitial ()

places the sheep on a random unblocked tile at the beginning of the game
 Chooses a random tile to place the sheep on at the beginning of the game.

5.10.2.12 void Game2Scene::placeTiles ()

Places the tiles to create the full grid.
 Places the tiles on the grid of the game.

5.10.2.13 void Game2Scene::resetDistances ()

Sets all the blocks of the grid as having infinite distance to the sheep.
 Resets the status of the grid tiles as having infinite distance to the sheep.

5.10.2.14 void Game2Scene::resetPrevious ()

Sets all the blocks of the grid as having the previous node null.
 Resets the status of the grid tiles as having their previous tile NULL.

5.10.2.15 void Game2Scene::resetVisited ()

Sets all the blocks of the grid as having been unvisited.
 Resets the status of the grid tiles as not visited.

5.10.2.16 void Game2Scene::setUserTurn (bool *userTurn*)

sets the turn of the user

Parameters

<i>userTurn</i>	whether or not it's the user's turn
-----------------	-------------------------------------

Sets the user's turn as true or false.

5.10.2.17 Tile * Game2Scene::tileAt (int *i*, int *j*)

retrieves the tile from the grid at the indices given

Parameters

<i>i</i>	the row of the tile
<i>j</i>	the column of the tile

Returns

pointer to the tile at the indices

Gets the tile at the given indices.

5.10.2.18 QVector< int > Game2Scene::tilesToBlock ()

Sets the layout of the level buttons.

Returns

The indices of the tiles to block initially

chooses distinct random tiles to flag for blocking initially at the start of the game.

5.10.2.19 bool Game2Scene::win (Tile * tile)

determines if the user successfully trapped the sheep

Parameters

<i>tile</i>	the tile to check the neighbors of
-------------	------------------------------------

Returns

the state of the game if win or loss

Determines a win by checking if from the current sheep position it's possible to get to the border without encountering a blocked tile.

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

- [game2/game2scene.h](#)
- [game2/game2scene.cpp](#)

5.11 Game3 Class Reference

Public Slots

- void [endGame](#) ()
Removes save and exit button and adds replay and back buttons.
- void [replay](#) ()
Removes save and exit button and adds replay and back buttons.
- void [goToMainMenu](#) ()
Slot to go back to the games main menu when pressing Exit.

Public Member Functions

- [Game3](#) (Difficulty *difficulty*, Size *size*, QWidget **parent*=0)
Default constructor.
- virtual [~Game3](#) ()
Destructor.

5.11.1 Constructor & Destructor Documentation

5.11.1.1 **Game3::Game3** (*Difficulty difficulty*, *Size size*, *QWidget * parent = 0*)
[explicit]

Default constructor.

Sets the size of the window, initializes the graphic items, sets the layouts and connects buttons to their slots.

5.11.1.2 **Game3::~~Game3** () [virtual]

Destructor.

Frees allocated memory.

5.11.2 Member Function Documentation

5.11.2.1 **void Game3::endGame** () [slot]

Removes save and exit button and adds replay and back buttons.

Removes the save and exitbutton and adds the go back and replay buttons along with their connections.

5.11.2.2 **void Game3::goToMainMenu** () [slot]

Slot to go back to the games main menu when pressing Exit.

Goes back to the Main meny of Dots and Lines.

5.11.2.3 **void Game3::replay** () [slot]

Removes save and exit button and adds replay and back buttons.

Reloads game.

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

- [game3/game3.h](#)
- [game3/game3.cpp](#)

5.12 Game3Options Class Reference

Public Slots

- void [gotoGameMainMenu](#) ()
Takes the user to the game main menu.
- void [gotoGame](#) ()
Takes the user to game 3.
- void [setEasy](#) ()
Sets the game difficulty to Easy.
- void [setModerate](#) ()
Sets the game difficulty to Moderate.
- void [setHard](#) ()
Sets the game difficulty to Hard.
- void [setSizeFour](#) ()
Sets the game size to 4x4.
- void [setSizeEight](#) ()
Sets the game size to 8x8.
- void [setSizeSixteen](#) ()
Sets the game size to 16x16.

Public Member Functions

- [Game3Options](#) (QWidget *parent=0)
Constructor.
- virtual [~Game3Options](#) ()
Destructor.

5.12.1 Constructor & Destructor Documentation

5.12.1.1 [Game3Options::Game3Options \(QWidget * *parent* = 0 \)](#) [explicit]

Constructor.

Initializes all buttons and text and shows them on the screen. Also initializes connections.

5.12.1.2 [Game3Options::~~Game3Options \(\)](#) [virtual]

Destructor.

Frees allocated memory.

5.12.2 Member Function Documentation

5.12.2.1 void Game3Options::gotoGame () [slot]

Takes the user to game 3.

Takes the user to game 3. Called after clicking any level button.

5.12.2.2 void Game3Options::gotoGameMainMenu () [slot]

Takes the user to the game main menu.

Takes the user to the game main menu that corresponds to game 3. Called after clicking the corresponding button.

5.12.2.3 void Game3Options::setEasy () [slot]

Sets the game difficulty to Easy.

Sets the game difficulty to Easy. Called after clicking the corresponding button.

5.12.2.4 void Game3Options::setHard () [slot]

Sets the game difficulty to Hard.

Sets the game difficulty to Hard. Called after clicking the corresponding button.

5.12.2.5 void Game3Options::setModerate () [slot]

Sets the game difficulty to Moderate.

Sets the game difficulty to Moderate. Called after clicking the corresponding button.

5.12.2.6 void Game3Options::setSizeEight () [slot]

Sets the game size to 8x8.

Sets the game size to 8x8. Called after clicking the corresponding button.

5.12.2.7 void Game3Options::setSizeFour () [slot]

Sets the game size to 4x4.

Sets the game size to 4x4. Called after clicking the corresponding button.

5.12.2.8 void Game3Options::setSizeSixteen () [slot]

Sets the game size to 16x16.

Sets the game size to 16x16. Called after clicking the corresponding button.

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

- [game3/game3options.h](#)
- [game3/game3options.cpp](#)

5.13 Game3Scene Class Reference

Public Slots

- void [computerMove](#) ()
Computer move.

Signals

- void [done](#) ()
Signals [Game3](#) that the game is over.

Public Member Functions

- [Game3Scene](#) (Difficulty difficulty, Size size, QObject *parent=0)
Constructor.
- virtual [~Game3Scene](#) ()
Destructor.
- void [computerTurn](#) ()
Computer turn.
- bool [isUserTurn](#) ()
Returns whose turn it is to play.
- void [addNewlyDrawnLine](#) ([Line](#) *line)
Remembers newly drawn line.
- void [clearNewLines](#) ()
Clears new lines by turning them grey.
- void [gameOver](#) ()
Triggers finishing the game.
- bool [noMoreMoves](#) ()
Returns whether there are any moves left.
- void [closeBoxByUser](#) ()
Declares one more box as closed by user.
- void [closeBoxByComputer](#) ()
Declares one more box as closed by computer.
- [Line](#) * [getLineThatClosesBox](#) ()
Finds and returns a non-clicked line that closes at least one box.
- [Line](#) * [getSmartLine](#) ()
Finds and returns a non-clicked line that does not let the user close a box.

5.13.1 Constructor & Destructor Documentation

5.13.1.1 `Game3Scene::Game3Scene (Difficulty difficulty, Size size, QObject * parent = 0) [explicit]`

Constructor.

Initializes the difficulty, size, dots, lines and boxes of the game.

5.13.1.2 `Game3Scene::~~Game3Scene () [virtual]`

Destructor.

Frees allocated memory.

5.13.2 Member Function Documentation

5.13.2.1 `void Game3Scene::addNewlyDrawnLine (Line * line)`

Remembers newly drawn line.

Parameters

<i>line</i>	Newly drawn line
-------------	------------------

Remembers newly drawn line so it can be turned grey later. Also removes the line from the list of unmarked lines.

5.13.2.2 `void Game3Scene::clearNewLines ()`

Clears new lines by turning them grey.

Clears new lines by turning them grey and removing them from the vector.

5.13.2.3 `void Game3Scene::computerMove () [slot]`

Computer move.

Picks a line to select according to difficulty, and plays the turn.

5.13.2.4 `void Game3Scene::computerTurn ()`

Computer turn.

Starts a delay to call [computerMove\(\)](#).

5.13.2.5 void Game3Scene::gameOver ()

Triggers finishing the game.

Ends the game.

5.13.2.6 Line * Game3Scene::getLineThatClosesBox ()

Finds and returns a non-clicked line that closes at least one box.

Returns

Non-clicked line that closes a box

Finds and returns a non-clicked line that closes at least one box. Returns NULL if not found.

5.13.2.7 Line * Game3Scene::getSmartLine ()

Finds and returns a non-clicked line that does not let the user close a box.

Returns

Non-clicked line that does not let the user close a box next

Finds and returns a non-clicked line that does not let the user close a box. It does so by checking that the returned line is not the third line to be drawn around any box. If such a line is not found, it returns NULL.

5.13.2.8 bool Game3Scene::isUserTurn ()

Returns whose turn it is to play.

Returns

Whether it is the user's turn to play

Returns whether it is the user's turn to play.

5.13.2.9 bool Game3Scene::noMoreMoves ()

Returns whether there are any moves left.

Returns

Whether there are any unmarked lines left

Returns whether there are any unmarked lines left.

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

- [game3/game3scene.h](#)
- [game3/game3scene.cpp](#)

5.14 GameMainMenu Class Reference

Public Slots

- void [gotoGameOptions](#) ()
Takes the user to the game options.
- void [gotoGameSelection](#) ()
Takes the user to the game selection menu.

Public Member Functions

- [GameMainMenu](#) (int gameChoice, QWidget *parent=0)
Constructor.
- virtual [~GameMainMenu](#) ()
Destructor.

5.14.1 Constructor & Destructor Documentation

5.14.1.1 `GameMainMenu::GameMainMenu (int gameChoice, QWidget * parent = 0)` `[explicit]`

Constructor.

Initializes all buttons and the game title and instructions and shows them on the screen.
Also initializes connections.

5.14.1.2 `GameMainMenu::~~GameMainMenu ()` `[virtual]`

Destructor.

Frees allocated memory.

5.14.2 Member Function Documentation

5.14.2.1 void GameMainMenu::gotoGameOptions () [slot]

Takes the user to the game options.

Takes the user to the game option widget that corresponds to the game choice. Called after clicking "Play".

5.14.2.2 void GameMainMenu::gotoGameSelection () [slot]

Takes the user to the game selection menu.

Takes the user back to the game selection menu. Called after clicking the corresponding button.

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

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

5.15 GameOver Class Reference

Public Member Functions

- [GameOver](#) (bool win, QObject *parent=0)
Default constructor.
- virtual [~GameOver](#) ()
Destructor.

5.15.1 Constructor & Destructor Documentation

5.15.1.1 GameOver::GameOver (bool win, QObject * parent = 0) [explicit]

Default constructor.

Parameters

<i>win</i>	Indicates if the game has been won
------------	------------------------------------

Sets origin of the image depending on win state.

5.15.1.2 GameOver::~~GameOver () [virtual]

Destructor.

Frees allocated memory.

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

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

5.16 GameSelection Class Reference

Public Slots

- void [goToMyAccount](#) ()
Takes the user from the game selection menu to the My Account page.
- void [goToMain](#) ()
Takes the user from the game selection menu to the main welcome window.
- void [goToGame1](#) ()
Takes the user from the game selection menu to the Sheep [Line](#) game.
- void [goToGame2](#) ()
Takes the user from the game selection menu to the Trap the Sheep game.
- void [goToGame3](#) ()
Takes the user from the game selection menu to the Dots and Lines game.

Public Member Functions

- [GameSelection](#) (QWidget *parent=0)
Default constructor.
- virtual [~GameSelection](#) ()
Destructor.

5.16.1 Constructor & Destructor Documentation

5.16.1.1 `GameSelection::GameSelection (QWidget * parent = 0) [explicit]`

Default constructor.

Initializes all buttons and labels and shows them on the game selection menu.

5.16.1.2 `GameSelection::~~GameSelection () [virtual]`

Destructor.

Frees allocated memory.

5.16.2 Member Function Documentation

5.16.2.1 void `GameSelection::goToGame1` () [slot]

Takes the user from the game selection menu to the Sheep [Line](#) game.

Takes the user to game 1 when the appropriate button is clicked.

5.16.2.2 void `GameSelection::goToGame2` () [slot]

Takes the user from the game selection menu to the Trap the Sheep game.

Takes the user to game 2 when the appropriate button is clicked.

5.16.2.3 void `GameSelection::goToGame3` () [slot]

Takes the user from the game selection menu to the Dots and Lines game.

Takes the user to game 3 when the appropriate button is clicked.

5.16.2.4 void `GameSelection::goToMain` () [slot]

Takes the user from the game selection menu to the main welcome window.

Takes the user back to the login widget. Called when the logout button is clicked.

5.16.2.5 void `GameSelection::goToMyAccount` () [slot]

Takes the user from the game selection menu to the My Account page.

Takes the user to his/her account page. Called when the "Account" button is clicked.

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

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

5.17 Helper Class Reference

Static Public Member Functions

- static void [makeWidgetSmall](#) (QWidget *widget)
Makes given widget small.
- static void [makeWidgetLarge](#) (QWidget *widget)
Makes given widget large.
- static double [toRadians](#) (double degrees)
Takes an angle in degrees, turns it into radians.

Static Public Attributes

- static const double [PI](#) = 3.14159265
PI.

5.17.1 Member Function Documentation

5.17.1.1 void `Helper::makeWidgetLarge (QWidget * widget)` `[static]`

Makes given widget large.

Parameters

<i>widget</i>	Widget whose size to change
---------------	-----------------------------

Makes the given widget large.

5.17.1.2 void `Helper::makeWidgetSmall (QWidget * widget)` `[static]`

Makes given widget small.

Parameters

<i>widget</i>	Widget whose size to change
---------------	-----------------------------

Makes the given widget small.

5.17.1.3 double `Helper::toRadians (double degrees)` `[static]`

Takes an angle in degrees, turns it into radians.

Parameters

<i>degrees</i>	Angle in degrees
----------------	------------------

Returns

Angle in radians

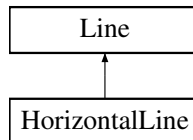
Converts angle from degrees to radians.

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

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

5.18 HorizontalLine Class Reference

Inheritance diagram for HorizontalLine:



Public Member Functions

- `HorizontalLine (Box *above, Box *under, QObject *parent=0)`
Default constructor.
- `virtual ~HorizontalLine ()`
Destructor.
- `void turnGrey ()`
Makes the line grey.
- `bool playTurn (bool userTurn)`
Called when a line is selected.
- `Box * getAbove () const`
Gets the box above the line.
- `Box * getUnder () const`
Gets the box under the line.

5.18.1 Constructor & Destructor Documentation

5.18.1.1 `HorizontalLine::HorizontalLine (Box * above, Box * under, QObject * parent = 0) [explicit]`

Default constructor.

Parameters

<i>above</i>	Box above line
<i>under</i>	Box under line

Sets `HorizontalLine` properties.

5.18.1.2 `HorizontalLine::~~HorizontalLine () [virtual]`

Destructor.

Frees allocated memory.

5.18.2 Member Function Documentation

5.18.2.1 **Box * HorizontalLine::getAbove () const**

Gets the box above the line.

Returns

box above the line

Returns the box above the line.

5.18.2.2 **Box * HorizontalLine::getUnder () const**

Gets the box under the line.

Returns

box under the line

Returns the box under the line.

5.18.2.3 **bool HorizontalLine::playTurn (bool *userTurn*) [virtual]**

Called when a line is selected.

Parameters

<i>userTurn</i>	Whether it is the user's turn
-----------------	-------------------------------

Returns

Whether it is still the player's turn

Called when a line is drawn. Returns whether it is still the same player's turn.

Implements [Line](#).

5.18.2.4 **void HorizontalLine::turnGrey () [virtual]**

Makes the line grey.

Changes the object image to make it grey.

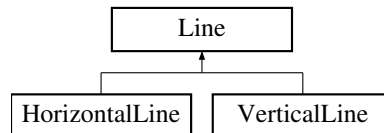
Implements [Line](#).

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

- [game3/horizontalline.h](#)
- [game3/horizontalline.cpp](#)

5.19 Line Class Reference

Inheritance diagram for Line:



Public Member Functions

- [Line](#) (bool [isHorizontal](#), QObject *parent=0)
Default constructor.
- virtual void [turnGrey](#) ()=0
Makes the line grey.
- void [mousePressEvent](#) (QGraphicsSceneMouseEvent *event)
Called when the user clicks on the line.
- virtual bool [playTurn](#) (bool userTurn)=0
Called when a line is selected.
- bool [isDrawn](#) () const
Returns whether the line has been drawn already.
- void [draw](#) ()
Sets the line as drawn.
- bool [isHorizontal](#) () const
Returns whether the line is horizontal or vertical.

5.19.1 Constructor & Destructor Documentation

5.19.1.1 [Line::Line \(bool *isHorizontal*, QObject * *parent* = 0 \)](#) `[explicit]`

Default constructor.

Parameters

isHorizontal	Whether the line is horizontal or not (vertical)
------------------------------	--

Initializes the object and marks it as not drawn.

5.19.2 Member Function Documentation

5.19.2.1 [bool Line::isDrawn \(\)](#) const

Returns whether the line has been drawn already.

Returns

Whether the line has been drawn already

5.19.2.2 bool Line::isHorizontal () const

Returns whether the line is horizontal or vertical.

Returns

Whether the line is horizontal

Returns whether the line is horizontal or vertical

5.19.2.3 void Line::mousePressEvent (QGraphicsSceneMouseEvent * event)

Called when the user clicks on the line.

Called when the user clicks on the line. The function changes the states of corresponding lines and boxes. It then checks for a win.

5.19.2.4 virtual bool Line::playTurn (bool userTurn) [pure virtual]

Called when a line is selected.

Parameters

<i>userTurn</i>	Whether it is the user's turn
-----------------	-------------------------------

Returns

Whether it is still the player's turn

Implemented in [HorizontalLine](#), and [VerticalLine](#).

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

- [game3/line.h](#)
- [game3/line.cpp](#)

5.20 MainWidget Class Reference**Public Slots**

- void [goToGameSelection](#) ()
Slot that closes widget and opens the game selection menu.

Public Member Functions

- [MainWindow](#) (QWidget *parent=0)
Default constructor.
- virtual [~MainWindow](#) ()
Destructor.

5.20.1 Constructor & Destructor Documentation

5.20.1.1 MainWindow::MainWindow (QWidget * *parent* = 0) [explicit]

Default constructor.

Initializes all buttons, input fields and labels and shows them on the screen.

5.20.1.2 MainWindow::~~MainWindow () [virtual]

Destructor.

Frees allocated memory.

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

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

5.21 MyAccount Class Reference

Public Slots

- void [goToGames](#) ()
Goes back to the games selection menu.

Public Member Functions

- [MyAccount](#) (QWidget *parent=0)
Default constructor.
- virtual [~MyAccount](#) ()
Destructor.

5.21.1 Constructor & Destructor Documentation

5.21.1.1 MyAccount::MyAccount (QWidget * *parent* = 0) [explicit]

Default constructor.

Initializes all buttons and labels and shows them on the game selection menu.

5.21.1.2 **MyAccount::~~MyAccount ()** [virtual]

Destructor.

Frees allocated memory.

5.21.2 Member Function Documentation

5.21.2.1 **void MyAccount::goToGames ()** [slot]

Goes back to the games selection menu.

Takes the user back to the game selection menu. Called when the user clicks the corresponding button.

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

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

5.22 ShaunGamesTest Class Reference

Public Member Functions

- void **setUp** ()
- void **tearDown** ()

Protected Member Functions

- void **toRadiansTest** ()
- void **getRandomSheepNumberTest** ()

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

- Tests/ShاونGamesTest.h
- Tests/ShاونGamesTest.cpp

5.23 Sheep1 Class Reference

Public Slots

- void [firedMove](#) ()
Moves the sheep in the direction of the firing.

Public Member Functions

- [Sheep1](#) (int number, bool inLine, QObject *parent=0)
Constructor.
- virtual [~Sheep1](#) ()
Destructor.
- double [getAngle](#) () const
Gets the angle of the sheep in degrees.
- void [setAngle](#) (double angle)
Sets the angle of the sheep in degrees.
- bool [isInLine](#) () const
Checks whether the sheep is part of the moving line.
- void [setInLine](#) (bool inLine)
Changes the status of the sheep as in or out of the moving line.
- void [fire](#) (double angle)
Fires the sheep in a straight line.
- void [moveInLine](#) (double distance)
Moves the sheep in line by given distance.
- double [inLineDistanceTo](#) (const [Sheep1](#) *other) const
Returns the in-line distance between object and given sheep.
- int [getNumber](#) () const
Returns the sheep number.

Static Public Member Functions

- static int [getRandomSheepNumber](#) ()
Returns a number between 1 and 9.

5.23.1 Constructor & Destructor Documentation

5.23.1.1 [Sheep1::Sheep1](#) (int number, bool inLine, QObject * parent = 0) [explicit]

Constructor.

Parameters

<i>number</i>	Sheep number
<i>inLine</i>	Whether sheep is in line

Sets the properties of the sheep.

5.23.1.2 **Sheep1::~~Sheep1** () [virtual]

Destructor.

Frees allocated memory.

5.23.2 Member Function Documentation

5.23.2.1 **void Sheep1::fire** (double *angle*)

Fires the sheep in a straight line.

Parameters

<i>angle</i>	Angle at which to fire the sheep
--------------	----------------------------------

Fires sheep at an angle. Called when the user fires the cannon.

5.23.2.2 **void Sheep1::firedMove** () [slot]

Moves the sheep in the direction of the firing.

Moves the sheep in the distance of the firing of the cannon.

5.23.2.3 **double Sheep1::getAngle** () const

Gets the angle of the sheep in degrees.

Returns

The angle of the sheep in the circle

5.23.2.4 **int Sheep1::getNumber** () const

Returns the sheep number.

Returns

Sheep number

5.23.2.5 **int Sheep1::getRandomSheepNumber** () [static]

Returns a number between 1 and 9.

Returns

Number between 1 and 9

Returns a random sheep number between 1 and 9.

5.23.2.6 double Sheep1::inLineDistanceTo (const Sheep1 * *other*) const

Returns the in-line distance between object and given sheep.

Parameters

<i>other</i>	Other sheep
--------------	-------------

Returns

Distance in pixels

Calculates the distance between the two in-line sheep.

5.23.2.7 bool Sheep1::isInLine () const

Checks whether the sheep is part of the moving line.

Returns

Whether sheep is in line

5.23.2.8 void Sheep1::moveInLine (double *distance*)

Moves the sheep in line by given distance.

Parameters

<i>distance</i>	Distance by which to move the sheep in pixels
-----------------	---

Moves sheep in line by given distance.

5.23.2.9 void Sheep1::setAngle (double *angle*)

Sets the angle of the sheep in degrees.

Parameters

<i>angle</i>	The angle of the sheep in the circle
--------------	--------------------------------------

5.23.2.10 void Sheep1::setInLine (bool *inLine*)

Changes the status of the sheep as in or out of the moving line.

Parameters

<i>inLine</i>	Status of the sheep (inside or outside the line)
---------------	--

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

- [game1/sheep1.h](#)
- [game1/sheep1.cpp](#)

5.24 Sheep2 Class Reference

Public Member Functions

- [Sheep2](#) ([Tile](#) *tile, [QObject](#) *parent=0)
default constructor
- void [setCurrent](#) ([Tile](#) *tile)
sets the current tile of the sheep
- [Tile](#) * [getCurrent](#) ()
gets the current tile of the sheep

5.24.1 Constructor & Destructor Documentation

5.24.1.1 [Sheep2::Sheep2](#) ([Tile](#) * *tile*, [QObject](#) * *parent* = 0) [explicit]

default constructor

Initializes the sheep position and picture.

5.24.2 Member Function Documentation

5.24.2.1 [Tile](#) * [Sheep2::getCurrent](#) ()

gets the current tile of the sheep

Returns

the current tile

Gets the current tile of the sheep.

5.24.2.2 void [Sheep2::setCurrent](#) ([Tile](#) * *tile*)

sets the current tile of the sheep

Parameters

<i>tile</i>	the tile to be set as current
-------------	-------------------------------

Unsets current tile and sets the argument tile as current.

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

- game2/sheep2.h
- game2/[sheep2.cpp](#)

5.25 Tile Class Reference

Public Member Functions

- [Tile](#) (bool block, int row, int col, QObject *parent=0)
Default constructor.
- void [setBlock](#) (bool block)
set the status of the tile as blocked or not
- bool [isBlocked](#) ()
retrieves the blocked status of the tile
- void [mousePressEvent](#) (QGraphicsSceneMouseEvent *event)
what to do on the mouse press event
- virtual [~Tile](#) ()
Destructor.
- void [setHasSheep](#) (bool placed)
sets the status of the tile as having a sheep on it or not
- int [getRow](#) ()
retrieves the row index of the current tile
- int [getCol](#) ()
retrieves the column index of the current tile
- bool [isBorder](#) ()
checks if the current tile is on the border of the grid
- void [setVisited](#) (bool visit)
sets the status of the tile as visited or not
- bool [isVisited](#) ()
checks if the current tile has been visited
- int [getDistance](#) ()
Retrieves the distance to the sheep so far.
- void [setDistance](#) (int distance)
Sets the distance to the sheep.
- [Tile](#) * [getPrev](#) ()
Retrieves the previous tile.
- void [setPrev](#) ([Tile](#) *tile)
Sets the previous tile.

5.25.1 Constructor & Destructor Documentation

5.25.1.1 `Tile::Tile (bool block, int row, int col, QObject * parent = 0)` [explicit]

Default constructor.

Sets the block status, scale and initializes indices.

5.25.1.2 `Tile::~~Tile ()` [virtual]

Destructor.

Frees allocated memory.

5.25.2 Member Function Documentation

5.25.2.1 `int Tile::getCol ()`

retrieves the column index of the current tile

Returns

the column index

Retrieves the column of the tile.

5.25.2.2 `int Tile::getDistance ()`

Retrieves the distance to the sheep so far.

Returns

The distance to the sheep

5.25.2.3 `Tile * Tile::getPrev ()`

Retrieves the previous tile.

Returns

The previous tile

5.25.2.4 `int Tile::getRow ()`

retrieves the row index of the current tile

Returns

the row index

Retrieves the row of the tile.

5.25.2.5 bool Tile::isBlocked ()

retrieves the blocked status of the tile

Returns

whether or not the tile is blocked

Retrieves the blocked status.

5.25.2.6 bool Tile::isBorder ()

checks if the current tile is on the border of the grid

Returns

the border status of the tile

Checks if the column is a border.

5.25.2.7 bool Tile::isVisited ()

checks if the current tile has been visited

Returns

the visited status of the tile

Returns the visited status of the tile.

5.25.2.8 void Tile::mousePressEvent (QGraphicsSceneMouseEvent * *event*)

what to do on the mouse press event

Parameters

<i>event</i>	the mouse press event
--------------	-----------------------

On click, places a block on the tile and checks for win status, then gives the turn to the computer.

5.25.2.9 void Tile::setBlock (bool *block*)

set the status of the tile as blocked or not

Parameters

<i>block</i>	decides if the tile is blocked or not
--------------	---------------------------------------

Marks tile as selected and adds border to it.

5.25.2.10 void Tile::setDistance (int *distance*)

Sets the distance to the sheep.

Parameters

<i>The</i>	distance to the sheep
------------	-----------------------

5.25.2.11 void Tile::setHasSheep (bool *placed*)

sets the status of the tile as having a sheep on it or not

Parameters

<i>placed</i>	boolean status of having a sheep placed on it or not
---------------	--

Sets the tile as having a sheep or not.

5.25.2.12 void Tile::setPrev (Tile * *tile*)

Sets the previous tile.

Parameters

<i>tile</i>	The tile to be set as previous
-------------	--------------------------------

5.25.2.13 void Tile::setVisited (bool *visit*)

sets the status of the tile as visited or not

Parameters

<i>visit</i>	visited status of the tile
--------------	----------------------------

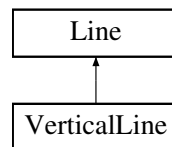
Sets the visited status of the tile.

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

- [game2/tile.h](#)
- [game2/tile.cpp](#)

5.26 VerticalLine Class Reference

Inheritance diagram for VerticalLine:



Public Member Functions

- [VerticalLine](#) ([Box](#) *left, [Box](#) *right, [QObject](#) *parent=0)
Default constructor.
- virtual [~VerticalLine](#) ()
Destructor.
- void [turnGrey](#) ()
Makes the line grey.
- bool [playTurn](#) (bool userTurn)
Called when a line is selected.
- [Box](#) * [getLeft](#) () const
Gets the box to the left of the line.
- [Box](#) * [getRight](#) () const
Gets the box to the right of the line.

5.26.1 Constructor & Destructor Documentation

5.26.1.1 [VerticalLine::VerticalLine](#) ([Box](#) * left, [Box](#) * right, [QObject](#) * parent = 0)
[explicit]

Default constructor.

Parameters

<i>left</i>	Box on left of line
<i>right</i>	Box on right of line

Sets [VerticalLine](#) properties.

5.26.1.2 `VerticalLine::~~VerticalLine ()` [virtual]

Destructor.

Frees allocated memory.

5.26.2 Member Function Documentation

5.26.2.1 `Box * VerticalLine::getLeft ()` const

Gets the box to the left of the line.

Returns

box to the left of the line

Returns the box to the left of the line.

5.26.2.2 `Box * VerticalLine::getRight ()` const

Gets the box to the right of the line.

Returns

box to the right of the line

Returns the box to the right of the line.

5.26.2.3 `bool VerticalLine::playTurn (bool userTurn)` [virtual]

Called when a line is selected.

Parameters

<i>userTurn</i>	Whether it is the user's turn
-----------------	-------------------------------

Returns

Whether it is still the player's turn

Called when a line is drawn. Returns whether it is still the same player's turn.

Implements [Line](#).

5.26.2.4 `void VerticalLine::turnGrey ()` [virtual]

Makes the line grey.

Changes the object image to make it grey.

Implements [Line](#).

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

- [game3/verticalline.h](#)
- [game3/verticalline.cpp](#)

Chapter 6

File Documentation

6.1 difficulty.h File Reference

Difficulty enum.

Enumerations

- enum **Difficulty** { **NO_DIFFICULTY**, **EASY**, **MODERATE**, **HARD**, **DIFFICULT-Y_END** }

6.1.1 Detailed Description

Difficulty enum. This enum lists the different possible difficulties for games 2 and 3.

Author

Rita Aoun
Rawan Moukalled

6.2 game1/barn.cpp File Reference

Contains [Barn](#) class definition.

```
#include "barn.h"    #include "game1scene.h"    #include <Q-  
List>
```

6.2.1 Detailed Description

Contains [Barn](#) class definition.

6.3 game1/barn.h File Reference

[Barn](#) class.

```
#include <QtGui> #include <QTimer>
```

Classes

- class [Barn](#)

6.3.1 Detailed Description

[Barn](#) class. [Barn](#) that terminates the game once a sheep from the line reaches it

Author

Rita Aoun
Rawan Moukalled

6.4 game1/cannon.cpp File Reference

Contains [Cannon](#) class definition.

```
#include "cannon.h" #include "game1scene.h"
```

6.4.1 Detailed Description

Contains [Cannon](#) class definition.

6.5 game1/cannon.h File Reference

[Cannon](#) class.

```
#include <QtGui>
```

Classes

- class [Cannon](#)

6.5.1 Detailed Description

[Cannon](#) class. [Cannon](#) objects rotate with mouse movements, and fire sheep on click.

Author

Rita Aoun
Rawan Moukalled

6.6 game1/game1.cpp File Reference

Contains the Sheep [Line](#).

```
#include "game1/game1.h"    #include "helper.h"    #include  
"gui/gamemainmenu.h"
```

6.6.1 Detailed Description

Contains the Sheep [Line](#).

6.7 game1/game1.h File Reference

Sheep [Line](#) class.

```
#include <QtGui> #include "game1/game1scene.h"
```

Classes

- class [Game1](#)

6.7.1 Detailed Description

Sheep [Line](#) class. This is the class for the gameplay of the Sheep [Line](#) game.

Author

Rita Aoun
Rawan Moukalled

6.8 game1/game1options.cpp File Reference

Contains [Game1Options](#) class definition.

```
#include "game1/game1options.h" #include "helper.h" #include  
"gui/gamemainmenu.h" #include "game1/game1.h"
```

6.8.1 Detailed Description

Contains [Game1Options](#) class definition.

6.9 game1/game1options.h File Reference

[Game1Options](#) class.

```
#include <QtGui>
```

Classes

- class [Game1Options](#)

6.9.1 Detailed Description

[Game1Options](#) class. This is the options page for game 1, where the user can choose the level with which to start the game. Only unlocked levels can be accessed.

Author

Rita Aoun
Rawan Moukalled

6.10 game1/game1scene.cpp File Reference

Contains [Game1Scene](#) class definition.

```
#include "game1scene.h" #include "helper.h" #include <Q-  
Vector> #include <QSet> #include "game1options.h"
```

6.10.1 Detailed Description

Contains [Game1Scene](#) class definition.

6.11 game1/game1scene.h File Reference

Sheep [Line](#) class.

```
#include <QtGui> #include <QLinkedList> #include <Q-  
Timer> #include "game1/cannon.h" #include "game1/sheep1.-  
h" #include "game1/barn.h" #include "gameover.h"
```

Classes

- class [Game1Scene](#)

6.11.1 Detailed Description

Sheep [Line](#) class. Implements the scene of Game 1: Sheep [Line](#)

Author

Rita Aoun
Rawan Moukalled

6.12 game1/sheep1.cpp File Reference

Contains [Sheep1](#) class definition.

```
#include "sheep1.h" #include "helper.h" #include "game1scene.h" #include <QString>
```

6.12.1 Detailed Description

Contains [Sheep1](#) class definition.

6.13 game1/sheep1.h File Reference

[Sheep1](#) class.

```
#include <QtGui> #include <QTimer>
```

Classes

- class [Sheep1](#)

6.13.1 Detailed Description

[Sheep1](#) class. Randomly numbered sheep that are used for Game 1: Sheep [Line](#)

Author

Rita Aoun
Rawan Moukalled

6.14 game2/game2.cpp File Reference

Contains [Game2](#) class definition.

```
#include "game2/game2.h"    #include "helper.h"    #include  
"gui/gamemainmenu.h"
```

6.14.1 Detailed Description

Contains [Game2](#) class definition.

6.15 game2/game2.h File Reference

Trap the Sheep class.

```
#include <QtGui> #include "difficulty.h" #include "game2/game2scene.-  
h"
```

Classes

- class [Game2](#)

6.15.1 Detailed Description

Trap the Sheep class. This is the class for the gameplay of the Trap the Sheep game.

Author

Rita Aoun
Rawan Moukalled

6.16 game2/game2options.cpp File Reference

Contains [Game2Options](#) class definition.

```
#include "game2/game2options.h" #include "helper.h" #include  
"game2/game2.h" #include "gui/gamemainmenu.h"
```

6.16.1 Detailed Description

Contains [Game2Options](#) class definition.

6.17 game2/game2options.h File Reference

[Game2Options](#) class.

```
#include <QtGui> #include "difficulty.h"
```

Classes

- class [Game2Options](#)

6.17.1 Detailed Description

[Game2Options](#) class. This is the options page for game 2, where the user can choose the level with which to start the game. Levels are: Easy, Moderate and Hard.

Author

Rita Aoun
Rawan Moukalled

6.18 game2/game2scene.cpp File Reference

Contains [Game2Scene](#) class definition.

```
#include "game2scene.h" #include <climits>
```

6.18.1 Detailed Description

Contains [Game2Scene](#) class definition.

6.19 game2/game2scene.h File Reference

Trap the Sheep scene class.

```
#include <QGraphicsScene> #include <QtGui> #include "difficulty.-  
h" #include "game2/tile.h" #include "game2/sheep2.h" ×  
#include "gameover.h"
```

Classes

- class [Game2Scene](#)

6.19.1 Detailed Description

Trap the Sheep scene class. This is the scene class for the gameplay of the Trap the Sheep game.

Author

Rita Aoun
Rawan Moukalled

6.20 game2/sheep2.cpp File Reference

Contains Sheep class definition.

```
#include "sheep2.h"
```

6.20.1 Detailed Description

Contains Sheep class definition.

6.21 game2/tile.cpp File Reference

Contains [Tile](#) class definition.

```
#include "tile.h" #include "game2scene.h" #include "sheep2.-  
h"
```

6.21.1 Detailed Description

Contains [Tile](#) class definition.

6.22 game2/tile.h File Reference

class for the tiles of game 2

```
#include <QtGui> #include <QMouseEvent>
```

Classes

- class [Tile](#)

6.22.1 Detailed Description

class for the tiles of game 2 This is class for the tiles of the grid in game 2

Author

Rita Aoun
Rawan Moukalled

6.23 game3/box.cpp File Reference

Contains [Box](#) class definition.

```
#include "box.h"
```

6.23.1 Detailed Description

Contains [Box](#) class definition.

6.24 game3/box.h File Reference

[Box](#) class.

```
#include <QtGui>
```

Classes

- class [Box](#)

6.24.1 Detailed Description

[Box](#) class. [Box](#) objects need to be bounded by a player for them to win points.

Author

Rita Aoun
Rawan Moukalled

6.25 game3/dot.cpp File Reference

Contains [Dot](#) class definition.

```
#include "dot.h"
```

6.25.1 Detailed Description

Contains [Dot](#) class definition.

6.26 game3/dot.h File Reference

[Dot](#) class.

```
#include <QtGui>
```

Classes

- class [Dot](#)

6.26.1 Detailed Description

[Dot](#) class. [Dot](#) objects delimit the game lines.

Author

Rita Aoun
Rawan Moukalled

6.27 game3/game3.cpp File Reference

Contains the Dots and Lines game.

```
#include "game3/game3.h"    #include "helper.h"    #include  
"gui/gamemainmenu.h"
```

6.27.1 Detailed Description

Contains the Dots and Lines game.

6.28 game3/game3.h File Reference

Dots and Lines class.

```
#include <QtGui> #include "difficulty.h" #include "game3/size.-  
h" #include "game3/game3scene.h"
```

Classes

- class [Game3](#)

6.28.1 Detailed Description

Dots and Lines class. This is the class for the gameplay of the Dots and Lines game.

Author

Rita Aoun
Rawan Moukalled

6.29 game3/game3options.cpp File Reference

Contains [Game3Options](#) class definition.

```
#include "game3options.h"    #include "helper.h"    #include  
"game3/game3.h" #include "gui/gamemainmenu.h"
```

6.29.1 Detailed Description

Contains [Game3Options](#) class definition.

6.30 game3/game3options.h File Reference

[Game3Options](#) class.

```
#include <QtGui> #include "difficulty.h" #include "game3/size.-  
h"
```

Classes

- class [Game3Options](#)

6.30.1 Detailed Description

[Game3Options](#) class. This is the options page for game 3, where the user can choose the level and size with which to start the game. Levels are: Easy, Moderate and Hard. Sizes are: 4x4, 8x8, 16x16.

Author

Rita Aoun
Rawan Moukalled

6.31 game3/game3scene.cpp File Reference

Contains [Game3Scene](#) class definition.

```
#include "game3scene.h"
```

6.31.1 Detailed Description

Contains [Game3Scene](#) class definition.

6.32 game3/game3scene.h File Reference

[Game3Scene](#) class.

```
#include <QtGui> #include "difficulty.h" #include "game3/size.-  
h" #include "game3/dot.h" #include "game3/horizontalline.-  
h" #include "game3/verticalline.h" #include "gameover.h"
```

Classes

- class [Game3Scene](#)

6.32.1 Detailed Description

[Game3Scene](#) class. This is the scene for game 3, Dots and Lines.

Author

Rita Aoun
Rawan Moukalled

6.33 game3/horizontalline.cpp File Reference

Contains [HorizontalLine](#) class definition.

```
#include "horizontalline.h" #include "game3/game3scene.-  
h"
```

6.33.1 Detailed Description

Contains [HorizontalLine](#) class definition.

6.34 game3/horizontalline.h File Reference

[HorizontalLine](#) class.

```
#include "game3/line.h" #include "game3/box.h"
```

Classes

- class [HorizontalLine](#)

6.34.1 Detailed Description

[HorizontalLine](#) class. Horizontal lines that delimit boxes from the top and bottom.

Author

Rita Aoun
Rawan Moukalled

6.35 game3/line.cpp File Reference

Contains [Line](#) class definition.

```
#include "line.h" #include "game3/game3scene.h"
```

6.35.1 Detailed Description

Contains [Line](#) class definition.

6.36 game3/line.h File Reference

[Line](#) class.

```
#include <QtGui>
```

Classes

- class [Line](#)

6.36.1 Detailed Description

[Line](#) class. [Line](#) is an interface for vertical and horizontal lines. It implements the on-click reaction of lines and it remembers whether a line has been clicked before.

Author

Rita Aoun
Rawan Moukalled

6.37 game3/size.h File Reference

Size enum.

Enumerations

- enum **Size** { **NO_SIZE**, **FOURBYFOUR** = 4, **EIGHTBYEIGHT** = 8, **SIXTEENBYSIXTEEN** = 16, **SIZE_END** }

6.37.1 Detailed Description

Size enum. This enum lists the different possible sizes of the game 3 grid.

Author

Rita Aoun
Rawan Moukalled

6.38 game3/verticalline.cpp File Reference

Contains [VerticalLine](#) class definition.

```
#include "verticalline.h" #include "game3/game3scene.h"
```

6.38.1 Detailed Description

Contains [VerticalLine](#) class definition.

6.39 game3/verticalline.h File Reference

[VerticalLine](#) class.

```
#include <game3/line.h> #include "game3/box.h"
```

Classes

- class [VerticalLine](#)

6.39.1 Detailed Description

[VerticalLine](#) class. Vertical lines that delimit boxes from the left and right.

Author

Rita Aoun
Rawan Moukalled

6.40 gameover.cpp File Reference

Contains [GameOver](#) class definition.

```
#include "gameover.h"
```

6.40.1 Detailed Description

Contains [GameOver](#) class definition.

6.41 gameover.h File Reference

Game Over class.

```
#include <QtGui>
```

Classes

- class [GameOver](#)

6.41.1 Detailed Description

Game Over class. Image overlayed on the screen when game is over

Author

Rita Aoun
Rawan Moukalled

6.42 gui/gamemainmenu.cpp File Reference

Contains [GameMainMenu](#) class definition.

```
#include "gui/gamemainmenu.h" #include "helper.h" #include  
"game1/game1options.h" #include "game2/game2options.h"
```

```
#include "game3/game3options.h" #include "gui/gameselection.-  
h"
```

6.42.1 Detailed Description

Contains [GameMainMenu](#) class definition.

6.43 gui/gamemainmenu.h File Reference

[GameMainMenu](#) class.

```
#include <QtGui>
```

Classes

- class [GameMainMenu](#)

6.43.1 Detailed Description

[GameMainMenu](#) class. This is the main game menu, where the user choose between resuming a previous game or starting a new one. The instructions are also shown on this menu.

Author

Rita Aoun
Rawan Moukalled

6.44 gui/gameselection.cpp File Reference

Contains [GameSelection](#) class definition.

```
#include "gui/gameselection.h" #include "helper.h" #include  
"gui/mainwidget.h" #include "gui/gamemainmenu.h" #include  
"myaccount.h"
```

6.44.1 Detailed Description

Contains [GameSelection](#) class definition.

6.45 gui/gameselection.h File Reference

Game selection menu class.


```
#include <QtGui>
```

Classes

- class [GameSelection](#)

6.45.1 Detailed Description

Game selection menu class. Game selection menu, where the user can select one of the three games available.

Author

Rita Aoun
Rawan Moukalled

6.46 gui/mainwidget.cpp File Reference

Contains [MainWidget](#) class definition.

```
#include "gui/mainwidget.h" #include "gui/gameselection.-  
h"
```

6.46.1 Detailed Description

Contains [MainWidget](#) class definition.

6.47 gui/mainwidget.h File Reference

[MainWidget](#) class.

```
#include <QtGui>
```

Classes

- class [MainWidget](#)

6.47.1 Detailed Description

[MainWidget](#) class. This is the main sign in window, where the user is given the chance to go on as a guest or to login/sign up.

Author

Rita Aoun
Rawan Moukalled

6.48 helper.cpp File Reference

Contains [Helper](#) class definition.

```
#include "helper.h" #include <ctime> #include <QTime>
```

6.48.1 Detailed Description

Contains [Helper](#) class definition.

6.49 helper.h File Reference

[Helper](#) class.

```
#include <QtGui>
```

Classes

- class [Helper](#)

6.49.1 Detailed Description

[Helper](#) class. This class provides various helper functions that are needed across windows.

Author

Rita Aoun
Rawan Moukalled

6.50 myaccount.cpp File Reference

Contains [MyAccount](#) class definition.

```
#include "myaccount.h" #include "helper.h" #include "gui/gameselection.-h"
```

6.50.1 Detailed Description

Contains [MyAccount](#) class definition.

6.51 myaccount.h File Reference

Class representing the my account and performance history windows.

```
#include <QtGui>
```

Classes

- class [MyAccount](#)

6.51.1 Detailed Description

Class representing the my account and performance history windows. Menu that presents to the signed in user his history and performance statistics

Author

Rita Aoun
Rawan Moukalled