

Project435L

Generated by Doxygen 1.8.13

Contents

1	Project435L	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	File Index	7
4.1	File List	7
5	Class Documentation	9
5.1	Battleship Class Reference	9
5.1.1	Constructor & Destructor Documentation	10
5.1.1.1	Battleship()	10
5.1.2	Member Function Documentation	10
5.1.2.1	back	10
5.1.2.2	keyPressEvent()	11
5.1.2.3	startgame	11
5.1.3	Member Data Documentation	11
5.1.3.1	goback	11
5.1.3.2	grid1	11
5.1.3.3	start	12
5.1.3.4	uname	12
5.2	battleshipgame Class Reference	12

5.2.1	Constructor & Destructor Documentation	14
5.2.1.1	battleshipgame()	14
5.2.2	Member Function Documentation	14
5.2.2.1	exit	14
5.2.2.2	shoot	15
5.2.2.3	shoot10	15
5.2.2.4	shoot11	16
5.2.2.5	shoot12	16
5.2.2.6	shoot13	16
5.2.2.7	shoot14	17
5.2.2.8	shoot15	17
5.2.2.9	shoot16	17
5.2.2.10	shoot2	18
5.2.2.11	shoot3	18
5.2.2.12	shoot4	18
5.2.2.13	shoot5	19
5.2.2.14	shoot6	19
5.2.2.15	shoot7	19
5.2.2.16	shoot8	20
5.2.2.17	shoot9	20
5.2.2.18	test	20
5.2.2.19	test10	21
5.2.2.20	test11	22
5.2.2.21	test12	22
5.2.2.22	test13	22
5.2.2.23	test14	23
5.2.2.24	test15	23
5.2.2.25	test16	23
5.2.2.26	test2	24
5.2.2.27	test3	24

5.2.2.28	test4	24
5.2.2.29	test5	25
5.2.2.30	test6	25
5.2.2.31	test7	25
5.2.2.32	test8	26
5.2.2.33	test9	26
5.2.3	Member Data Documentation	26
5.2.3.1	Attempts	26
5.2.3.2	box	26
5.2.3.3	btn	27
5.2.3.4	btn1	27
5.2.3.5	btn10	27
5.2.3.6	btn11	27
5.2.3.7	btn12	27
5.2.3.8	btn13	27
5.2.3.9	btn14	27
5.2.3.10	btn15	27
5.2.3.11	btn2	28
5.2.3.12	btn3	28
5.2.3.13	btn4	28
5.2.3.14	btn5	28
5.2.3.15	btn6	28
5.2.3.16	btn7	28
5.2.3.17	btn8	28
5.2.3.18	btn9	28
5.2.3.19	goback	29
5.2.3.20	gr	29
5.2.3.21	nbtn	29
5.2.3.22	nbtn1	29
5.2.3.23	nbtn10	29

5.2.3.24	nbtn11	29
5.2.3.25	nbtn12	29
5.2.3.26	nbtn13	29
5.2.3.27	nbtn14	30
5.2.3.28	nbtn15	30
5.2.3.29	nbtn2	30
5.2.3.30	nbtn3	30
5.2.3.31	nbtn4	30
5.2.3.32	nbtn5	30
5.2.3.33	nbtn6	30
5.2.3.34	nbtn7	30
5.2.3.35	nbtn8	31
5.2.3.36	nbtn9	31
5.2.3.37	q	31
5.2.3.38	table	31
5.2.3.39	uname	31
5.3	bluedisk Class Reference	31
5.3.1	Constructor & Destructor Documentation	32
5.3.1.1	bluedisk()	32
5.3.2	Member Function Documentation	32
5.3.2.1	keyPressEvent()	32
5.4	disk Class Reference	32
5.4.1	Constructor & Destructor Documentation	33
5.4.1.1	disk()	33
5.4.2	Member Function Documentation	33
5.4.2.1	back	34
5.4.2.2	keyPressEvent()	34
5.4.2.3	startgame	34
5.4.3	Member Data Documentation	35
5.4.3.1	goback	35

5.4.3.2	grid1	35
5.4.3.3	start	35
5.4.3.4	uname	35
5.5	diskgame Class Reference	35
5.5.1	Constructor & Destructor Documentation	36
5.5.1.1	diskgame()	36
5.5.2	Member Data Documentation	37
5.5.2.1	uname	37
5.6	disks Class Reference	37
5.6.1	Constructor & Destructor Documentation	37
5.6.1.1	disks()	38
5.6.2	Member Function Documentation	38
5.6.2.1	move	38
5.6.3	Member Data Documentation	39
5.6.3.1	uname	39
5.7	greendisk Class Reference	39
5.7.1	Constructor & Destructor Documentation	39
5.7.1.1	greendisk()	39
5.7.2	Member Function Documentation	40
5.7.2.1	keyPressEvent()	40
5.8	history Class Reference	40
5.8.1	Constructor & Destructor Documentation	40
5.8.1.1	history()	41
5.8.2	Member Function Documentation	41
5.8.2.1	goback	41
5.8.3	Member Data Documentation	41
5.8.3.1	back	41
5.8.3.2	grid	42
5.8.3.3	hist	42
5.8.3.4	uname	42

5.9	history2 Class Reference	42
5.9.1	Constructor & Destructor Documentation	43
5.9.1.1	history2()	43
5.9.2	Member Function Documentation	43
5.9.2.1	goback	43
5.9.3	Member Data Documentation	44
5.9.3.1	back	44
5.9.3.2	grid	44
5.9.3.3	hist	44
5.9.3.4	uname	44
5.10	profile Class Reference	44
5.10.1	Constructor & Destructor Documentation	45
5.10.1.1	profile()	46
5.10.2	Member Function Documentation	47
5.10.2.1	goback	47
5.10.2.2	gohistory	47
5.10.2.3	gohistory2	48
5.10.2.4	playgame1	48
5.10.2.5	playgame2	49
5.10.3	Member Data Documentation	49
5.10.3.1	back	49
5.10.3.2	box	49
5.10.3.3	countryname	49
5.10.3.4	current	50
5.10.3.5	currenttime	50
5.10.3.6	flag	50
5.10.3.7	fullname	50
5.10.3.8	grid1	50
5.10.3.9	histor	50
5.10.3.10	histor2	50

5.10.3.11 name	50
5.10.3.12 play	51
5.10.3.13 play2	51
5.10.3.14 pp	51
5.10.3.15 s	51
5.10.3.16 uname	51
5.11 questions Class Reference	51
5.11.1 Constructor & Destructor Documentation	52
5.11.1.1 questions()	52
5.11.2 Member Function Documentation	52
5.11.2.1 sigemit	53
5.11.2.2 submit	53
5.11.3 Member Data Documentation	53
5.11.3.1 False	53
5.11.3.2 grid	54
5.11.3.3 ok	54
5.11.3.4 ques	54
5.11.3.5 True	54
5.12 reddisk Class Reference	54
5.12.1 Constructor & Destructor Documentation	55
5.12.1.1 reddisk()	55
5.12.2 Member Function Documentation	55
5.12.2.1 keyPressEvent()	55
5.12.2.2 spawn	56
5.12.3 Member Data Documentation	56
5.12.3.1 uname	56
5.13 Signin Class Reference	57
5.13.1 Constructor & Destructor Documentation	57
5.13.1.1 Signin()	58
5.13.2 Member Function Documentation	58

5.13.2.1	playasguest	58
5.13.2.2	signinvalidation	59
5.13.2.3	signuppage	59
5.13.3	Member Data Documentation	60
5.13.3.1	box	60
5.13.3.2	grid1	60
5.13.3.3	guest	60
5.13.3.4	password	60
5.13.3.5	passwordtxt	60
5.13.3.6	signup	60
5.13.3.7	submit	60
5.13.3.8	username	61
5.13.3.9	username.txt	61
5.14	SignUp Class Reference	61
5.14.1	Constructor & Destructor Documentation	62
5.14.1.1	SignUp()	62
5.14.2	Member Function Documentation	63
5.14.2.1	browse	63
5.14.2.2	goback	63
5.14.2.3	submitreg	64
5.14.3	Member Data Documentation	65
5.14.3.1	back	65
5.14.3.2	box	65
5.14.3.3	countrycode	65
5.14.3.4	dob	65
5.14.3.5	dobtxt	65
5.14.3.6	female	65
5.14.3.7	firstname	65
5.14.3.8	firstnametxt	66
5.14.3.9	fullname	66

5.14.3.10 gender	66
5.14.3.11 grid1	66
5.14.3.12 group	66
5.14.3.13 lastname	66
5.14.3.14 lastnametxt	66
5.14.3.15 male	66
5.14.3.16 password	67
5.14.3.17 passwordtxt	67
5.14.3.18 phonenummer	67
5.14.3.19 phonenumbertxt	67
5.14.3.20 profilepicture	67
5.14.3.21 profilepicturebr	67
5.14.3.22 s1	67
5.14.3.23 showpass	67
5.14.3.24 submit	68
5.14.3.25 username	68
5.14.3.26 usernametxt	68
6 File Documentation	69
6.1 Project435L/battleship.cpp File Reference	69
6.1.1 Detailed Description	69
6.2 Project435L/battleship.h File Reference	69
6.2.1 Detailed Description	70
6.3 Project435L/battleshipgame.cpp File Reference	70
6.3.1 Detailed Description	70
6.3.2 Variable Documentation	70
6.3.2.1 attempt	71
6.3.2.2 correct	71
6.3.2.3 wrong	71
6.4 Project435L/battleshipgame.h File Reference	71
6.4.1 Detailed Description	72

6.4.2	Variable Documentation	72
6.4.2.1	attempt	72
6.4.2.2	correct	72
6.4.2.3	wrong	72
6.5	Project435L/bluedisk.cpp File Reference	73
6.5.1	Detailed Description	73
6.6	Project435L/bluedisk.h File Reference	73
6.6.1	Detailed Description	74
6.7	Project435L/disk.cpp File Reference	74
6.7.1	Detailed Description	74
6.8	Project435L/disk.h File Reference	74
6.8.1	Detailed Description	75
6.9	Project435L/diskgame.cpp File Reference	75
6.9.1	Detailed Description	75
6.9.2	Variable Documentation	76
6.9.2.1	attempts	76
6.9.2.2	blue	76
6.9.2.3	green	76
6.9.2.4	red	76
6.9.2.5	totalscore	76
6.10	Project435L/diskgame.h File Reference	77
6.10.1	Detailed Description	77
6.10.2	Variable Documentation	77
6.10.2.1	attempts	77
6.10.2.2	blue	78
6.10.2.3	green	78
6.10.2.4	red	78
6.10.2.5	totalscore	78
6.11	Project435L/disks.cpp File Reference	78
6.11.1	Detailed Description	79

6.12	Project435L/disks.h File Reference	79
6.12.1	Detailed Description	79
6.13	Project435L/greendisk.cpp File Reference	79
6.13.1	Detailed Description	80
6.14	Project435L/greendisk.h File Reference	80
6.14.1	Detailed Description	80
6.15	Project435L/history.cpp File Reference	80
6.15.1	Detailed Description	81
6.16	Project435L/history.h File Reference	81
6.17	Project435L/history2.cpp File Reference	81
6.17.1	Detailed Description	81
6.18	Project435L/history2.h File Reference	82
6.18.1	Detailed Description	82
6.19	Project435L/main.cpp File Reference	82
6.19.1	Function Documentation	82
6.19.1.1	main()	83
6.20	Project435L/profile.cpp File Reference	83
6.20.1	Detailed Description	83
6.21	Project435L/profile.h File Reference	83
6.21.1	Detailed Description	84
6.22	Project435L/questions.cpp File Reference	84
6.22.1	Detailed Description	84
6.23	Project435L/questions.h File Reference	84
6.23.1	Detailed Description	85
6.24	Project435L/reddisk.cpp File Reference	85
6.24.1	Detailed Description	85
6.25	Project435L/reddisk.h File Reference	85
6.25.1	Detailed Description	86
6.26	Project435L/signin.cpp File Reference	86
6.26.1	Detailed Description	86
6.27	Project435L/signin.h File Reference	86
6.28	Project435L/signup.cpp File Reference	87
6.28.1	Detailed Description	87
6.29	Project435L/signup.h File Reference	87

Chapter 1

Project435L

sets the scale of the window, and shows the sign in page with buttons to navigate

Author

Mohamad Abou Salem
Hadi Al Karhani

Date

11-28-2021

Returns

application execution

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

QGraphicsPixmapItem	
bluedisk	31
disks	37
greendisk	39
reddisk	54
QGraphicsScene	
diskgame	35
QObject	
bluedisk	31
disks	37
greendisk	39
reddisk	54
QWidget	
Battleship	9
battleshipgame	12
disk	32
history	40
history2	42
profile	44
questions	51
Signin	57
SignUp	61

Chapter 3

Class Index

3.1 Class List

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

Battleship	9
battleshipgame	12
bluedisk	31
disk	32
diskgame	35
disks	37
greendisk	39
history	40
history2	42
profile	44
questions	51
reddisk	54
Signin	57
SignUp	61

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Project435L/ battleship.cpp	
Battleship class definition	69
Project435L/ battleship.h	
Battleship class	69
Project435L/ battleshipgame.cpp	
Battleshipgame class definition	70
Project435L/ battleshipgame.h	
Battleshipgame class	71
Project435L/ bluedisk.cpp	
Disk pad of blue disk class definition	73
Project435L/ bluedisk.h	
Bluedisk class	73
Project435L/ disk.cpp	
Disk class definition	74
Project435L/ disk.h	
Disk class	74
Project435L/ diskgame.cpp	
Diskgame class definition	75
Project435L/ diskgame.h	
Diskagame class	77
Project435L/ disks.cpp	
Disks class definition	78
Project435L/ disks.h	
Disks class	79
Project435L/ greendisk.cpp	
Disk pad of green disk class definition	79
Project435L/ greendisk.h	
Greendisk class	80
Project435L/ history.cpp	
User history of game1 which is the shooting discs game	80
Project435L/ history.h	
Project435L/ history2.cpp	
User history of game2 which is the shooting discs game	81
Project435L/ history2.h	
History2 class	82

Project435L/main.cpp	82
Project435L/profile.cpp	
User profile which displays current date & time, profile pic, full name, history of both games, buttons to play both games, and country flag depending on phone number inserted	83
Project435L/profile.h	
Profile class	83
Project435L/questions.cpp	
Questions to be asked at every turn of the battleship game	84
Project435L/questions.h	
Questions class	84
Project435L/reddisk.cpp	
Disk pad of red disk class definition	85
Project435L/reddisk.h	
Reddisk class	85
Project435L/signin.cpp	
Sign in class definition	86
Project435L/signin.h	86
Project435L/signup.cpp	
Sign up class definition	87
Project435L/signup.h	87

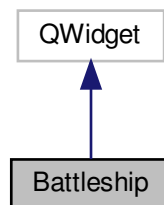
Chapter 5

Class Documentation

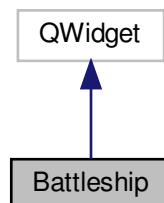
5.1 Battleship Class Reference

```
#include <battleship.h>
```

Inheritance diagram for Battleship:



Collaboration diagram for Battleship:



Public Slots

- void [back](#) ()
go back to the user profile page
- void [startgame](#) ()
enters game1 and starts playing

Public Member Functions

- [Battleship](#) (QString &username, QWidget *parent=nullptr)
- void [keyPressEvent](#) (QKeyEvent *event)
pres F1 to start the game

Private Attributes

- QPushButton * [start](#)
- QPushButton * [goback](#)
- QString [uname](#)
- QGridLayout * [grid1](#)

5.1.1 Constructor & Destructor Documentation

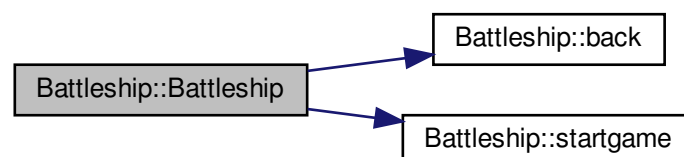
5.1.1.1 Battleship()

```
Battleship::Battleship (  
    QString & username,  
    QWidget * parent = nullptr ) [explicit]
```

go back to profile page

start playing the game

Here is the call graph for this function:



5.1.2 Member Function Documentation

5.1.2.1 back

```
void Battleship::back ( ) [slot]
```

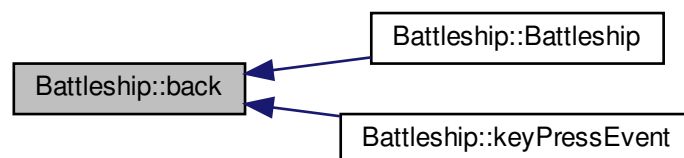
go back to the user profile page

its a function SLOT which is signaled by a press of a button when pressed the user profile page is opened and the game menu is closed

Author

Mohamad Abou Salem
Hadi Al Karhani

Here is the caller graph for this function:

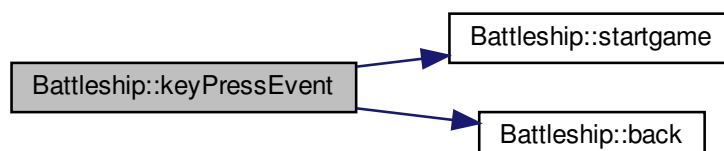


5.1.2.2 keyPressEvent()

```
void Battleship::keyPressEvent (
    QKeyEvent * event ) [inline]
```

pres F1 to start the game

Here is the call graph for this function:



5.1.2.3 startgame

```
void Battleship::startgame ( ) [slot]
```

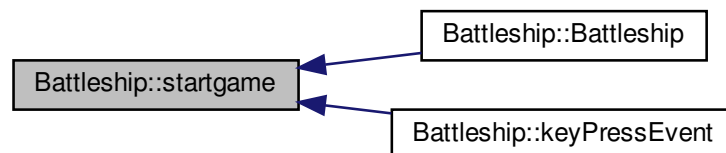
enters game1 and starts playing

its a function SLOT which is signaled by a press of a button when pressed the game menu page is closed and battleship game page is opened

Author

Mohamad Abou Salem
Hadi Al Karhani

Here is the caller graph for this function:



5.1.3 Member Data Documentation

5.1.3.1 goback

```
QPushButton* Battleship::goback [private]
```

5.1.3.2 grid1

```
QGridLayout* Battleship::grid1 [private]
```

5.1.3.3 start

```
QPushButton* Battleship::start [private]
```

5.1.3.4 uname

```
QString Battleship::uname [private]
```

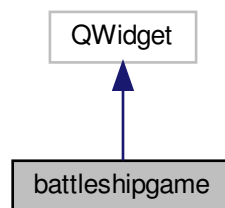
The documentation for this class was generated from the following files:

- Project435L/[battleship.h](#)
- Project435L/[battleship.cpp](#)

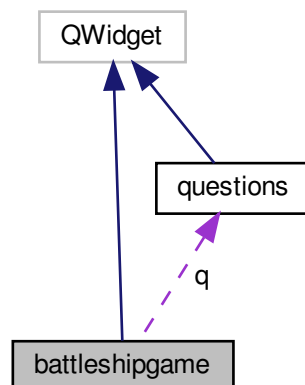
5.2 battleshipgame Class Reference

```
#include <battleshipgame.h>
```

Inheritance diagram for battleshipgame:



Collaboration diagram for battleshipgame:



Public Slots

- void [shoot](#) ()

shoot by clicking one of the right grid squares

- void [shoot2](#) ()
- void [shoot3](#) ()
- void [shoot4](#) ()
- void [shoot5](#) ()
- void [shoot6](#) ()
- void [shoot7](#) ()
- void [shoot8](#) ()
- void [shoot9](#) ()
- void [shoot10](#) ()
- void [shoot11](#) ()
- void [shoot12](#) ()
- void [shoot13](#) ()
- void [shoot14](#) ()
- void [shoot15](#) ()
- void [shoot16](#) ()
- void [test](#) (int x)

shoot by clicking one of the right grid squares

- void [test2](#) (int x)
- void [test3](#) (int x)
- void [test4](#) (int x)
- void [test5](#) (int x)
- void [test6](#) (int x)
- void [test7](#) (int x)
- void [test8](#) (int x)
- void [test9](#) (int x)
- void [test10](#) (int x)
- void [test11](#) (int x)
- void [test12](#) (int x)
- void [test13](#) (int x)
- void [test14](#) (int x)
- void [test15](#) (int x)
- void [test16](#) (int x)
- void [exit](#) ()

player exits when game is done and scores are saved

Public Member Functions

- [battleshipgame](#) (QString &username, QWidget *parent=nullptr)

incorrect answers initialized to 0

Private Attributes

- QPushButton * [goback](#)
- QString [uname](#)
- QVBoxLayout * [box](#)
- GridLayout * [gr](#)
- QTableWidgetItem * [table](#)
- QPushButton * [btn](#)
- QPushButton * [btn1](#)
- QPushButton * [btn2](#)
- QPushButton * [btn3](#)
- QPushButton * [btn4](#)
- QPushButton * [btn5](#)
- QPushButton * [btn6](#)
- QPushButton * [btn7](#)
- QPushButton * [btn8](#)
- QPushButton * [btn9](#)
- QPushButton * [btn10](#)
- QPushButton * [btn11](#)
- QPushButton * [btn12](#)
- QPushButton * [btn13](#)
- QPushButton * [btn14](#)
- QPushButton * [btn15](#)
- QPushButton * [nbtn](#)
- QPushButton * [nbtn1](#)
- QPushButton * [nbtn2](#)
- QPushButton * [nbtn3](#)
- QPushButton * [nbtn4](#)
- QPushButton * [nbtn5](#)
- QPushButton * [nbtn6](#)
- QPushButton * [nbtn7](#)
- QPushButton * [nbtn8](#)
- QPushButton * [nbtn9](#)
- QPushButton * [nbtn10](#)
- QPushButton * [nbtn11](#)
- QPushButton * [nbtn12](#)
- QPushButton * [nbtn13](#)
- QPushButton * [nbtn14](#)
- QPushButton * [nbtn15](#)
- [questions](#) * [q](#)
- QLabel * [Attempts](#)

5.2.1 Constructor & Destructor Documentation

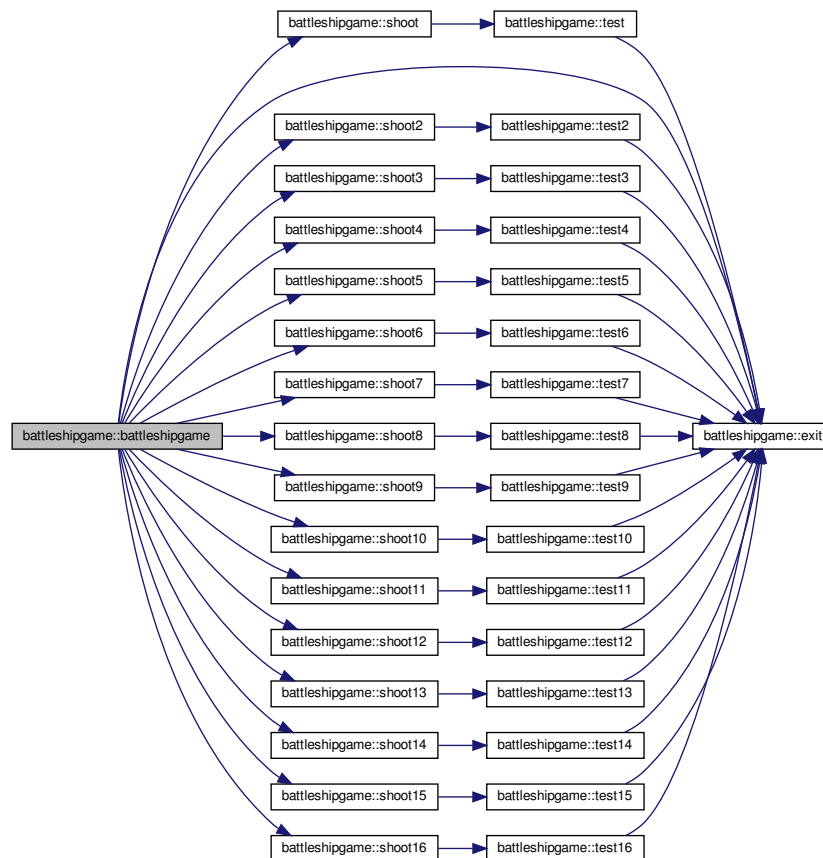
5.2.1.1 battleshipgame()

```
battleshipgame::battleshipgame (
    QString & username,
    QWidget * parent = nullptr )
```

incorrect answers initialized to 0

setting the 2 4x4 grids of the game

shoot by clicking on the right grid Here is the call graph for this function:



5.2.2 Member Function Documentation

5.2.2.1 exit

```
void battleshipgame::exit ( ) [slot]
```

player exits when game is done and scores are saved

when the player finishes the trials the scores are saved in Scores.txt file battleship game closes and opens the game menu

Author

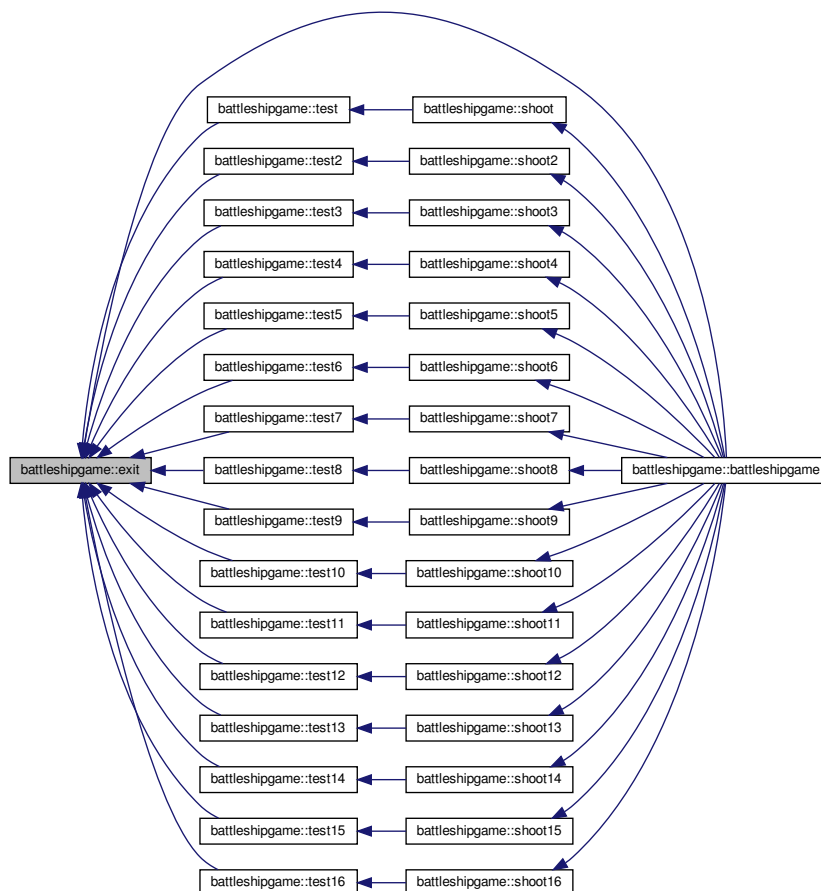
Mohamad Abou Salem
Hadi Al Karhani

opens text file

saves number of correct and wrong answers answered

closes game

Here is the caller graph for this function:



5.2.2.2 shoot

```
void battleshipgame::shoot ( ) [slot]
```

shoot by clicking one of the right grid squares

its a function SLOT which is signaled by a press of a button when pressed the questions page pops up and if answered correctly its a hit, if answered incorrectly its a miss if no ship is there then its an instant miss the number of attempts are decreased by 1 of every trial miss icon and hit icons are shown for the corresponding situation

Author

Mohamad Abou Salem
Hadi Al Karhani

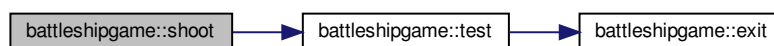
miss icon when ship is not there

randomly see if ship is there or not and do accordingly

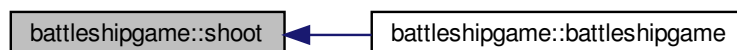
if missed then decrease attempt and declare as miss

if not missed then open up the question

Here is the call graph for this function:



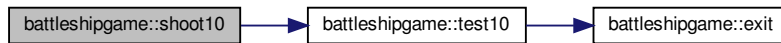
Here is the caller graph for this function:



5.2.2.3 shoot10

```
void battleshipgame::shoot10 ( ) [slot]
```

Here is the call graph for this function:



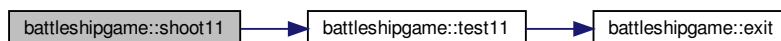
Here is the caller graph for this function:



5.2.2.4 shoot11

```
void battleshipgame::shoot11 ( ) [slot]
```

Here is the call graph for this function:



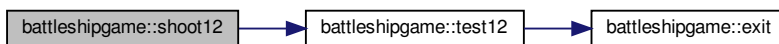
Here is the caller graph for this function:



5.2.2.5 shoot12

```
void battleshipgame::shoot12 ( ) [slot]
```

Here is the call graph for this function:



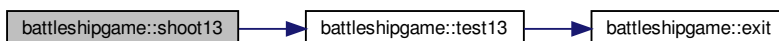
Here is the caller graph for this function:



5.2.2.6 shoot13

```
void battleshipgame::shoot13 ( ) [slot]
```

Here is the call graph for this function:



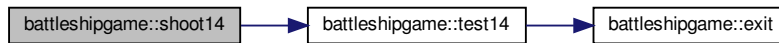
Here is the caller graph for this function:



5.2.2.7 shoot14

```
void battleshipgame::shoot14 ( ) [slot]
```

Here is the call graph for this function:



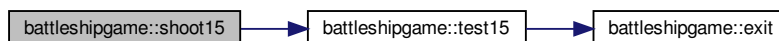
Here is the caller graph for this function:



5.2.2.8 shoot15

```
void battleshipgame::shoot15 ( ) [slot]
```

Here is the call graph for this function:



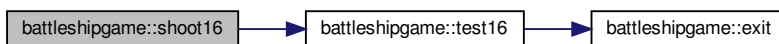
Here is the caller graph for this function:



5.2.2.9 shoot16

```
void battleshipgame::shoot16 ( ) [slot]
```

Here is the call graph for this function:



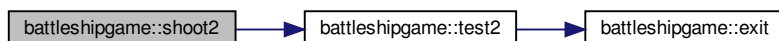
Here is the caller graph for this function:



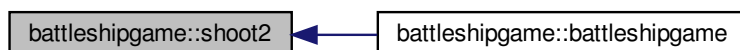
5.2.2.10 shoot2

```
void battleshipgame::shoot2 ( ) [slot]
```

Here is the call graph for this function:



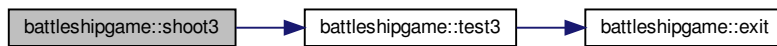
Here is the caller graph for this function:



5.2.2.11 shoot3

```
void battleshipgame::shoot3 ( ) [slot]
```

Here is the call graph for this function:



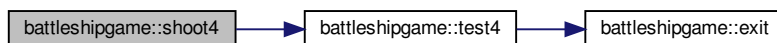
Here is the caller graph for this function:



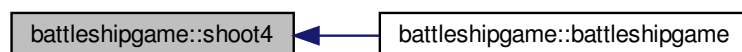
5.2.2.12 shoot4

```
void battleshipgame::shoot4 ( ) [slot]
```

Here is the call graph for this function:



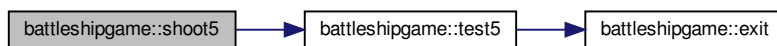
Here is the caller graph for this function:



5.2.2.13 shoot5

```
void battleshipgame::shoot5 ( ) [slot]
```

Here is the call graph for this function:



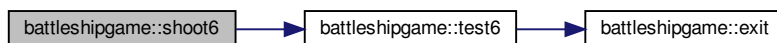
Here is the caller graph for this function:



5.2.2.14 shoot6

```
void battleshipgame::shoot6 ( ) [slot]
```

Here is the call graph for this function:



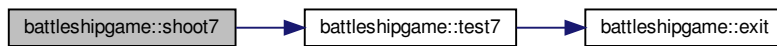
Here is the caller graph for this function:



5.2.2.15 shoot7

```
void battleshipgame::shoot7 ( ) [slot]
```

Here is the call graph for this function:



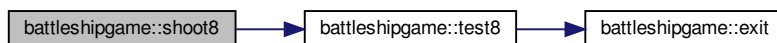
Here is the caller graph for this function:



5.2.2.16 shoot8

```
void battleshipgame::shoot8 ( ) [slot]
```

Here is the call graph for this function:



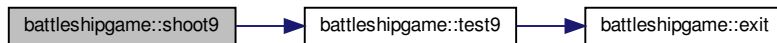
Here is the caller graph for this function:



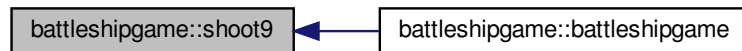
5.2.2.17 shoot9

```
void battleshipgame::shoot9 ( ) [slot]
```

Here is the call graph for this function:



Here is the caller graph for this function:



5.2.2.18 test

```
void battleshipgame::test (
    int x ) [slot]
```

shoot by clicking one of the right grid squares

its a function SLOT which is signaled by a press of a button when pressed the questions page pops up and if answered correctly its a hit, if answered incorrectly its a miss if no ship is there then its an instant miss the number of attempts are decreased by 1 of every trial miss icon and hit icons are shown for the corresponding situation when attempts = 0 the game is closed and declared if winner or loser prompts message to the user if won or lost

Author

Mohamad Abou Salem
Hadi Al Karhani

if hit then declare as hit and increase correct counter and decrease attempt counter

if wrong then declare as miss and increase wrong counter and decrease attempt counter

if wrong then randomly shoot a square from the grid from the user side

check number of wrongs if exceeded

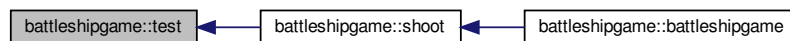
check number of corrects if exceeded

check number of attempts if reached

Here is the call graph for this function:



Here is the caller graph for this function:



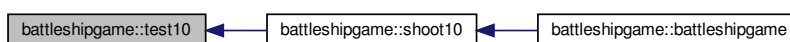
5.2.2.19 test10

```
void battleshipgame::test10 (  
    int x ) [slot]
```

Here is the call graph for this function:



Here is the caller graph for this function:



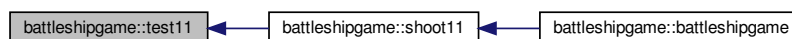
5.2.2.20 test11

```
void battleshipgame::test11 (  
    int x ) [slot]
```

Here is the call graph for this function:



Here is the caller graph for this function:



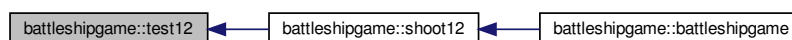
5.2.2.21 test12

```
void battleshipgame::test12 (  
    int x ) [slot]
```

Here is the call graph for this function:



Here is the caller graph for this function:



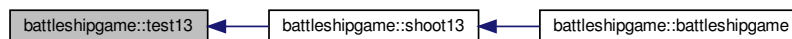
5.2.2.22 test13

```
void battleshipgame::test13 (  
    int x ) [slot]
```

Here is the call graph for this function:



Here is the caller graph for this function:



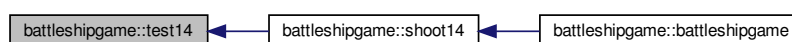
5.2.2.23 test14

```
void battleshipgame::test14 (  
    int x ) [slot]
```

Here is the call graph for this function:



Here is the caller graph for this function:



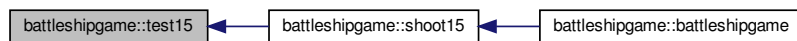
5.2.2.24 test15

```
void battleshipgame::test15 (  
    int x ) [slot]
```

Here is the call graph for this function:



Here is the caller graph for this function:



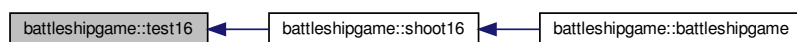
5.2.2.25 test16

```
void battleshipgame::test16 (  
    int x ) [slot]
```

Here is the call graph for this function:



Here is the caller graph for this function:



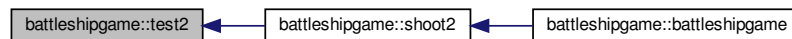
5.2.2.26 test2

```
void battleshipgame::test2 (  
    int x ) [slot]
```

Here is the call graph for this function:



Here is the caller graph for this function:



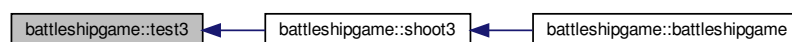
5.2.2.27 test3

```
void battleshipgame::test3 (  
    int x ) [slot]
```

Here is the call graph for this function:



Here is the caller graph for this function:



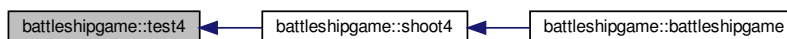
5.2.2.28 test4

```
void battleshipgame::test4 (  
    int x ) [slot]
```

Here is the call graph for this function:



Here is the caller graph for this function:



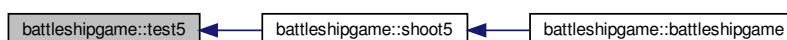
5.2.2.29 test5

```
void battleshipgame::test5 (  
    int x ) [slot]
```

Here is the call graph for this function:



Here is the caller graph for this function:



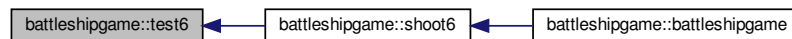
5.2.2.30 test6

```
void battleshipgame::test6 (  
    int x ) [slot]
```

Here is the call graph for this function:



Here is the caller graph for this function:



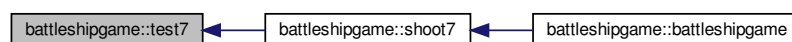
5.2.2.31 test7

```
void battleshipgame::test7 (  
    int x ) [slot]
```

Here is the call graph for this function:



Here is the caller graph for this function:



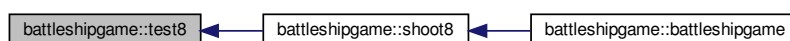
5.2.2.32 test8

```
void battleshipgame::test8 (  
    int x ) [slot]
```

Here is the call graph for this function:



Here is the caller graph for this function:



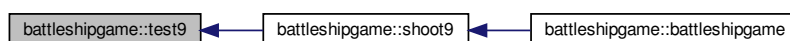
5.2.2.33 test9

```
void battleshipgame::test9 (  
    int x ) [slot]
```

Here is the call graph for this function:



Here is the caller graph for this function:



5.2.3 Member Data Documentation

5.2.3.1 Attempts

`QLabel* battleshipgame::Attempts` [private]

5.2.3.2 box

`QVBoxLayout* battleshipgame::box` [private]

5.2.3.3 btn

`QPushButton* battleshipgame::btn` [private]

5.2.3.4 btn1

`QPushButton* battleshipgame::btn1` [private]

5.2.3.5 btn10

`QPushButton* battleshipgame::btn10` [private]

5.2.3.6 btn11

`QPushButton* battleshipgame::btn11` [private]

5.2.3.7 btn12

`QPushButton* battleshipgame::btn12` [private]

5.2.3.8 btn13

`QPushButton* battleshipgame::btn13 [private]`

5.2.3.9 btn14

`QPushButton* battleshipgame::btn14 [private]`

5.2.3.10 btn15

`QPushButton* battleshipgame::btn15 [private]`

5.2.3.11 btn2

`QPushButton* battleshipgame::btn2 [private]`

5.2.3.12 btn3

`QPushButton* battleshipgame::btn3 [private]`

5.2.3.13 btn4

`QPushButton* battleshipgame::btn4 [private]`

5.2.3.14 btn5

`QPushButton* battleshipgame::btn5 [private]`

5.2.3.15 btn6

`QPushButton* battleshipgame::btn6 [private]`

5.2.3.16 btn7

QPushButton* battleshipgame::btn7 [private]

5.2.3.17 btn8

QPushButton* battleshipgame::btn8 [private]

5.2.3.18 btn9

QPushButton* battleshipgame::btn9 [private]

5.2.3.19 goback

QPushButton* battleshipgame::goback [private]

5.2.3.20 gr

QGridLayout* battleshipgame::gr [private]

5.2.3.21 nbtn

QPushButton* battleshipgame::nbtn [private]

5.2.3.22 nbtn1

QPushButton* battleshipgame::nbtn1 [private]

5.2.3.23 nbtn10

QPushButton* battleshipgame::nbtn10 [private]

5.2.3.24 nbtn11

`QPushButton* battleshipgame::nbtn11 [private]`

5.2.3.25 nbtn12

`QPushButton* battleshipgame::nbtn12 [private]`

5.2.3.26 nbtn13

`QPushButton* battleshipgame::nbtn13 [private]`

5.2.3.27 nbtn14

`QPushButton* battleshipgame::nbtn14 [private]`

5.2.3.28 nbtn15

`QPushButton* battleshipgame::nbtn15 [private]`

5.2.3.29 nbtn2

`QPushButton* battleshipgame::nbtn2 [private]`

5.2.3.30 nbtn3

`QPushButton* battleshipgame::nbtn3 [private]`

5.2.3.31 nbtn4

`QPushButton* battleshipgame::nbtn4 [private]`

5.2.3.32 nbtn5

`QPushButton* battleshipgame::nbtn5 [private]`

5.2.3.33 nbtn6

`QPushButton* battleshipgame::nbtn6 [private]`

5.2.3.34 nbtn7

`QPushButton* battleshipgame::nbtn7 [private]`

5.2.3.35 nbtn8

`QPushButton* battleshipgame::nbtn8 [private]`

5.2.3.36 nbtn9

`QPushButton* battleshipgame::nbtn9 [private]`

5.2.3.37 q

`questions* battleshipgame::q [private]`

5.2.3.38 table

`QTableWidget* battleshipgame::table [private]`

5.2.3.39 uname

```
QString battleshipgame::uname [private]
```

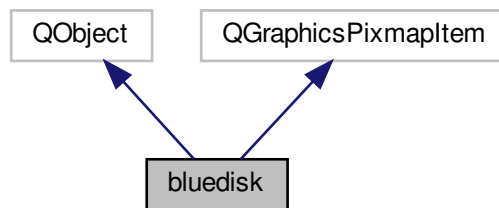
The documentation for this class was generated from the following files:

- [Project435L/battleshipgame.h](#)
- [Project435L/battleshipgame.cpp](#)

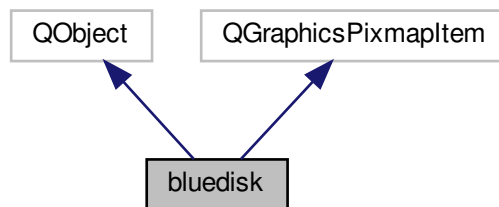
5.3 bluedisk Class Reference

```
#include <bluedisk.h>
```

Inheritance diagram for bluedisk:



Collaboration diagram for bluedisk:



Public Member Functions

- `bluedisk ()`
- `void keyPressEvent (QKeyEvent *event)`

5.3.1 Constructor & Destructor Documentation

5.3.1.1 `bluedisk()`

```
bluedisk::bluedisk ( )
```

set scale

set position

5.3.2 Member Function Documentation

5.3.2.1 `keyPressEvent()`

```
void bluedisk::keyPressEvent (
    QKeyEvent * event ) [inline]
```

check if pressed then check if disks collide

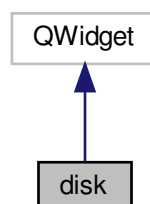
The documentation for this class was generated from the following files:

- [Project435L/bluedisk.h](#)
- [Project435L/bluedisk.cpp](#)

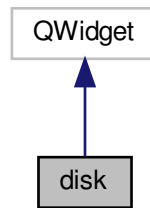
5.4 disk Class Reference

```
#include <disk.h>
```

Inheritance diagram for disk:



Collaboration diagram for disk:



Public Slots

- void [back](#) ()
go back to the user profile page
- void [startgame](#) ()
enters game2 and starts playing

Public Member Functions

- [disk](#) (QString &username, QWidget *parent=nullptr)
- void [keyPressEvent](#) (QKeyEvent *event)

Private Attributes

- QPushButton * [start](#)
- QPushButton * [goback](#)
- QString [uname](#)
- QGridLayout * [grid1](#)

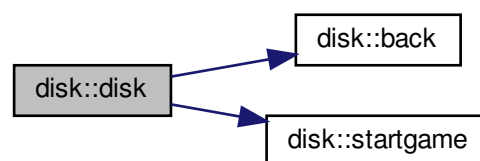
5.4.1 Constructor & Destructor Documentation

5.4.1.1 disk()

```
disk::disk (
    QString & username,
    QWidget * parent = nullptr ) [explicit]
```

set window background

Here is the call graph for this function:



5.4.2 Member Function Documentation

5.4.2.1 back

```
void disk::back ( ) [slot]
```

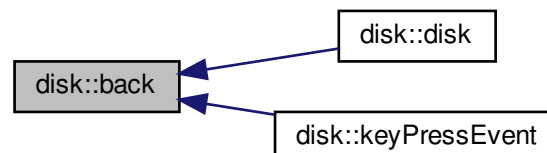
go back to the user profile page

its a function SLOT which is signaled by a press of a button when pressed the user profile page is opened and the game menu is closed

Author

Mohamad Abou Salem
Hadi Al Karhani

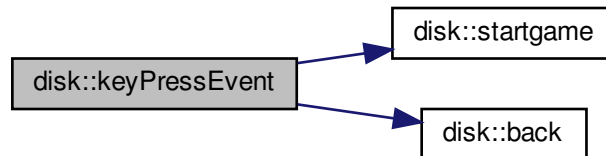
Here is the caller graph for this function:



5.4.2.2 keyPressEvent()

```
void disk::keyPressEvent (
    QKeyEvent * event ) [inline]
```

Here is the call graph for this function:



5.4.2.3 startgame

```
void disk::startgame ( ) [slot]
```

enters game2 and starts playing

its a function SLOT which is signaled by a press of a button when pressed the game menu page is closed and shooting discs game scene page is opened

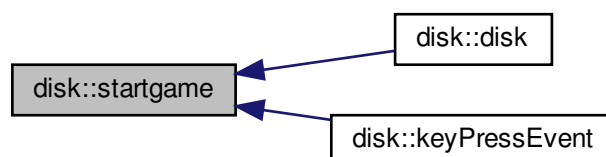
Author

Mohamad Abou Salem
Hadi Al Karhani

< turn off horizontal scrollbar

< turn off vertical scrollbar

< set window to fixed size 200x400 Here is the caller graph for this function:



5.4.3 Member Data Documentation

5.4.3.1 goback

```
QPushButton* disk::goback [private]
```

5.4.3.2 grid1

```
QGridLayout* disk::grid1 [private]
```

5.4.3.3 start

```
QPushButton* disk::start [private]
```

5.4.3.4 uname

```
QString disk::uname [private]
```

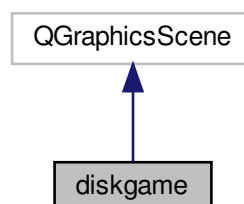
The documentation for this class was generated from the following files:

- [Project435L/disk.h](#)
- [Project435L/disk.cpp](#)

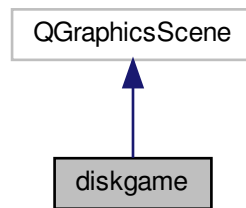
5.5 diskgame Class Reference

```
#include <diskgame.h>
```

Inheritance diagram for diskgame:



Collaboration diagram for diskgame:



Public Member Functions

- `diskgame` (QString &username)
total score initialized to 0

Private Attributes

- QString `uname`

5.5.1 Constructor & Destructor Documentation

5.5.1.1 `diskgame()`

```
diskgame::diskgame (  
    QString & username )
```

total score initialized to 0

get screen geometry

move to center of the screen

< add background image and scale it to the window size

add red disk to the scene

add green disk to the scene

add blue disk to the scene

add disk spawner to the scene

spawn a disk at the top every 1.5 sec

set focus on red pad every 1 sec

set focus on blue pad every 1.2 sec

set focus on green pad every 1.4 sec

5.5.2 Member Data Documentation

5.5.2.1 uname

```
QString diskgame::uname [private]
```

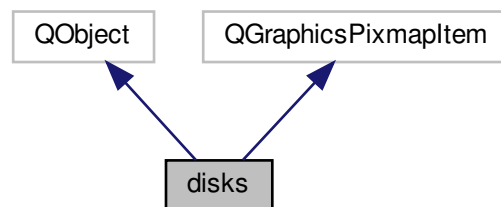
The documentation for this class was generated from the following files:

- [Project435L/diskgame.h](#)
- [Project435L/diskgame.cpp](#)

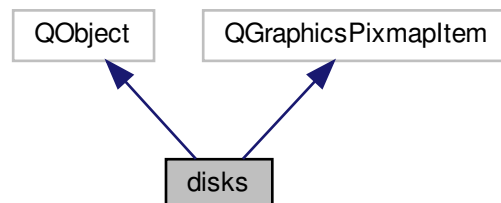
5.6 disks Class Reference

```
#include <disks.h>
```

Inheritance diagram for disks:



Collaboration diagram for disks:



Public Slots

- void `move` ()
move the disks downwards to a certain level then deletes from memory and scene

Public Member Functions

- `disks` (QString &username)

Private Attributes

- QString `uname`

5.6.1 Constructor & Destructor Documentation

5.6.1.1 `disks()`

```
disks::disks (  
    QString & username )
```

random number to choose between which disk to spawn

disks move at a certain speed depending on timer

speed doubles

speed doubles

speed doubles

speed doubles

speed doubles

Here is the call graph for this function:



5.6.2 Member Function Documentation

5.6.2.1 move

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

move the disks downwards to a certain level then deletes from memory and scene

the disks are moving at a fixed speed on the y axis downwards when reaching the bottom of the screen the disks are deleted from memory and removed from the scene when reaching the bottom and removed it is considered a miss so the user loses one attempt each time

Author

Mohamad Abou Salem
Hadi Al Karhani

disk is removed from memory and scene when it reaches the bottom of the screen to save memory

Here is the caller graph for this function:



5.6.3 Member Data Documentation

5.6.3.1 uname

```
QString disks::uname [private]
```

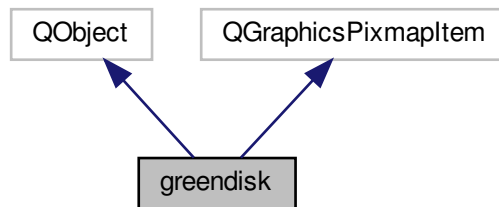
The documentation for this class was generated from the following files:

- [Project435L/disks.h](#)
- [Project435L/disks.cpp](#)

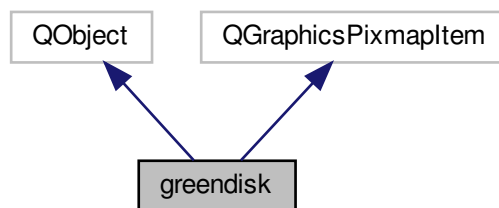
5.7 greendisk Class Reference

```
#include <greendisk.h>
```

Inheritance diagram for greendisk:



Collaboration diagram for greendisk:



Public Member Functions

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

5.7.1 Constructor & Destructor Documentation

5.7.1.1 greendisk()

```
greendisk::greendisk ( )
```

set scale

set position

5.7.2 Member Function Documentation

5.7.2.1 keyPressEvent()

```
void greendisk::keyPressEvent (
    QKeyEvent * event ) [inline]
```

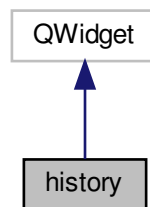
The documentation for this class was generated from the following files:

- [Project435L/greendisk.h](#)
- [Project435L/greendisk.cpp](#)

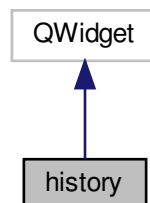
5.8 history Class Reference

```
#include <history.h>
```

Inheritance diagram for history:



Collaboration diagram for history:



Public Slots

- void [goback](#) ()
goes back to profile page

Public Member Functions

- [history](#) (QString &username, QWidget *parent=nullptr)

Private Attributes

- QString [uname](#)
- QLabel * [hist](#)
- QGridLayout * [grid](#)
- QPushButton * [back](#)

5.8.1 Constructor & Destructor Documentation

5.8.1.1 history()

```
history::history (
    QString & username,
    QWidget * parent = nullptr )
```

check Scores.txt file to display on screen

Here is the call graph for this function:



5.8.2 Member Function Documentation

5.8.2.1 goback

```
void history::goback ( ) [slot]
```

goes back to profile page

its a function SLOT which is signaled by a press of a button when pressed the user history of game1 page is closed and profile page is opened

Author

Mohamad Abou Salem
Hadi Al Karhani

Here is the caller graph for this function:



5.8.3 Member Data Documentation

5.8.3.1 back

```
QPushButton* history::back [private]
```

5.8.3.2 grid

```
QGridLayout* history::grid [private]
```

5.8.3.3 hist

```
QLabel* history::hist [private]
```

5.8.3.4 `uname`

```
QString history::uname [private]
```

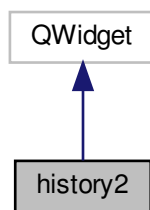
The documentation for this class was generated from the following files:

- [Project435L/history.h](#)
- [Project435L/history.cpp](#)

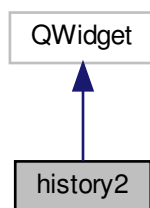
5.9 `history2` Class Reference

```
#include <history2.h>
```

Inheritance diagram for `history2`:



Collaboration diagram for `history2`:



Public Slots

- `void goback ()`
goes back to profile page

Public Member Functions

- [history2](#) (QString &username, QWidget *parent=nullptr)

Private Attributes

- QString [uname](#)
- QLabel * [hist](#)
- QGridLayout * [grid](#)
- QPushButton * [back](#)

5.9.1 Constructor & Destructor Documentation

5.9.1.1 history2()

```
history2::history2 (
    QString & username,
    QWidget * parent = nullptr )
```

check Scores2.txt file to display on screen

go back to user profile page

Here is the call graph for this function:



5.9.2 Member Function Documentation

5.9.2.1 goback

```
void history2::goback ( ) [slot]
```

goes back to profile page

its a function SLOT which is signaled by a press of a button when pressed the user history of game2 page is closed and profile page is opened

Author

Mohamad Abou Salem
Hadi Al Karhani

Here is the caller graph for this function:



5.9.3 Member Data Documentation

5.9.3.1 back

```
QPushButton* history2::back [private]
```

5.9.3.2 grid

```
QGridLayout* history2::grid [private]
```

5.9.3.3 hist

```
QLabel* history2::hist [private]
```

5.9.3.4 uname

```
QString history2::uname [private]
```

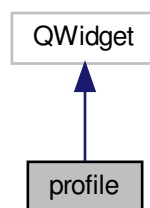
The documentation for this class was generated from the following files:

- [Project435L/history2.h](#)
- [Project435L/history2.cpp](#)

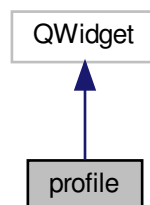
5.10 profile Class Reference

```
#include <profile.h>
```

Inheritance diagram for profile:



Collaboration diagram for profile:



Public Member Functions

- [profile](#) (QString &username, QWidget *parent=nullptr)

Public Attributes

- QLabel * [fullname](#)
- QLabel * [name](#)
- QLabel * [currenttime](#)
- QLabel * [current](#)
- QLabel * [countryname](#)
- QLabel * [pp](#)
- QLabel * [flag](#)
- QStringList [s](#)
- QString [uname](#)
- QPushButton * [back](#)
- QPushButton * [histor](#)
- QPushButton * [histor2](#)
- QPushButton * [play](#)
- QPushButton * [play2](#)
- QGridLayout * [grid1](#)
- QVBoxLayout * [box](#)

Private Slots

- void [goback](#) ()
signs out and goes to the sign in page
- void [playgame1](#) ()
enters game1
- void [playgame2](#) ()
enters game2
- void [gohistory](#) ()
enters game1 history page
- void [gohistory2](#) ()
enters game2 history page

5.10.1 Constructor & Destructor Documentation

5.10.1.1 profile()

```
profile::profile (
    QString & username,
    QWidget * parent = nullptr ) [explicit]
```

change the profile background and text color

scales window to 350x300

moves window to the center of the screen when opened

initializing all QLabels and QPushButton to display on the screen

check username to display that user's information

if user plays as guest, username would not be found

set default profile pic

set name as guest

set current date and time

set user's country flag based on phone number

set user's country flag based on phone number

set user's country flag based on phone number

set user's country flag based on phone number

set user's country flag based on phone number

go back to sign in page by signing out

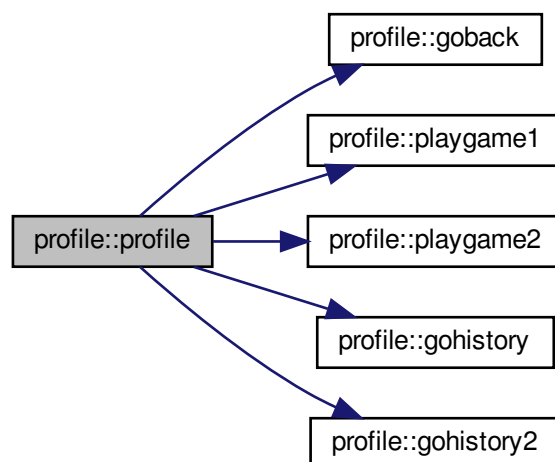
play game1, battleship game

play game2, shooting discs game

check game1 history

check game2 history

Here is the call graph for this function:



5.10.2 Member Function Documentation

5.10.2.1 goback

```
void profile::goback ( ) [private], [slot]
```

signs out and goes to the sign in page

its a function SLOT which is signaled by a press of a button when pressed the user profile page is closed and a sign in page is opened

Author

Mohamad Abou Salem
Hadi Al Karhani

open this instance of sign in page

close this instance of profile page

Here is the caller graph for this function:



5.10.2.2 gohistory

```
void profile::gohistory ( ) [private], [slot]
```

enters game1 history page

its a function SLOT which is signaled by a press of a button when pressed the user profile page is closed and battleship game history page is opened

Author

Mohamad Abou Salem
Hadi Al Karhani

open this instance of battleship game history page

close this instance of profile page

Here is the caller graph for this function:

**5.10.2.3 gohistory2**

```
void profile::gohistory2 ( ) [private], [slot]
```

enters game2 history page

its a function SLOT which is signaled by a press of a button when pressed the user profile page is closed and shooting discs game history page is opened

Author

Mohamad Abou Salem
Hadi Al Karhani

open this instance of shooting discs game history page

close this instance of profile page

Here is the caller graph for this function:



5.10.2.4 playgame1

```
void profile::playgame1 ( ) [private], [slot]
```

enters game1

its a function SLOT which is signaled by a press of a button when pressed the user profile page is closed and battleship game page is opened

Author

Mohamad Abou Salem
Hadi Al Karhani

open this instance of battleship game page

close this instance of profile page

Here is the caller graph for this function:



5.10.2.5 playgame2

```
void profile::playgame2 ( ) [private], [slot]
```

enters game2

its a function SLOT which is signaled by a press of a button when pressed the user profile page is closed and shooting discs game page is opened

Author

Mohamad Abou Salem
Hadi Al Karhani

open this instance of shooting discs game page

close this instance of profile page

Here is the caller graph for this function:



5.10.3 Member Data Documentation

5.10.3.1 back

`QPushButton* profile::back`

5.10.3.2 box

`QVBoxLayout* profile::box`

5.10.3.3 countryname

`QLabel* profile::countryname`

5.10.3.4 current

`QLabel* profile::current`

5.10.3.5 currenttime

`QLabel* profile::currenttime`

5.10.3.6 flag

`QLabel* profile::flag`

5.10.3.7 fullname

`QLabel* profile::fullname`

5.10.3.8 grid1

`QGridLayout* profile::grid1`

5.10.3.9 histor

`QPushButton* profile::histor`

5.10.3.10 histor2

`QPushButton* profile::histor2`

5.10.3.11 name

`QLabel* profile::name`

5.10.3.12 play

`QPushButton* profile::play`

5.10.3.13 play2

`QPushButton* profile::play2`

5.10.3.14 pp

`QLabel* profile::pp`

5.10.3.15 s

`QStringList profile::s`

5.10.3.16 uname

```
QString profile::uname
```

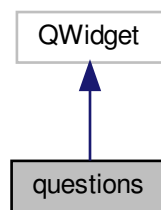
The documentation for this class was generated from the following files:

- [Project435L/profile.h](#)
- [Project435L/profile.cpp](#)

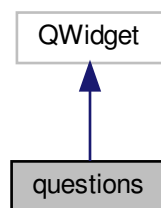
5.11 questions Class Reference

```
#include <questions.h>
```

Inheritance diagram for questions:



Collaboration diagram for questions:



Public Slots

- `int submit ()`

Submits the input provided from the user and checks if answered correctly.

Signals

- int `sigemit` (int x)

Public Member Functions

- `questions` (QWidget *parent=nullptr)

Private Attributes

- QRadioButton * `True`
- QRadioButton * `False`
- QGridLayout * `grid`
- QLabel * `ques`
- QPushButton * `ok`

5.11.1 Constructor & Destructor Documentation

5.11.1.1 `questions()`

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

initializing all QLabels, QRadioButtons, and QPushButtons to display on the screen

move window to center of the screen

read questions from txt file

when submitted it checks if answered correctly or not

Here is the call graph for this function:

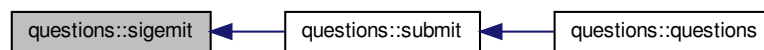


5.11.2 Member Function Documentation

5.11.2.1 sigemit

```
int questions::sigemit (  
    int x ) [signal]
```

Here is the caller graph for this function:



5.11.2.2 submit

```
int questions::submit ( ) [slot]
```

Submits the input provided from the user and checks if answered correctly.

its a function SLOT which is signaled by a press of a button when pressed the window closes and returns to the battleship game the questions are stored in txt file called Questions.txt

Author

Mohamad Abou Salem
Hadi Al Karhani

open text file

checks answer if user inputs True

checks answer if user inputs False

Here is the caller graph for this function:



5.11.3 Member Data Documentation

5.11.3.1 False

```
QRadioButton* questions::False [private]
```

5.11.3.2 grid

```
QGridLayout* questions::grid [private]
```

5.11.3.3 ok

```
QPushButton* questions::ok [private]
```

5.11.3.4 ques

```
QLabel* questions::ques [private]
```

5.11.3.5 True

```
QRadioButton* questions::True [private]
```

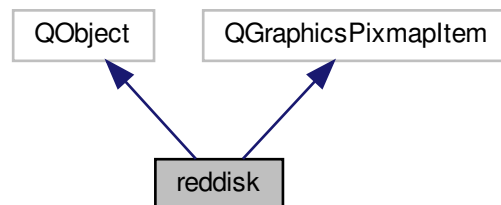
The documentation for this class was generated from the following files:

- [Project435L/questions.h](#)
- [Project435L/questions.cpp](#)

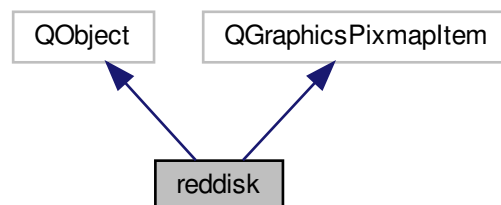
5.12 reddisk Class Reference

```
#include <reddisk.h>
```

Inheritance diagram for reddisk:



Collaboration diagram for reddisk:



Public Slots

- void [spawn](#) ()
spawn disks and add them to the scene for them to fall from above and checks number of attempts

Public Member Functions

- [reddisk](#) (QString &username)
- void [keyPressEvent](#) (QKeyEvent *event)

Private Attributes

- QString [uname](#)

5.12.1 Constructor & Destructor Documentation

5.12.1.1 reddisk()

```
reddisk::reddisk (
    QString & username )
```

set scale

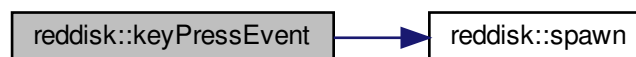
set position

5.12.2 Member Function Documentation

5.12.2.1 keyPressEvent()

```
void reddisk::keyPressEvent (
    QKeyEvent * event ) [inline]
```

Here is the call graph for this function:



5.12.2.2 spawn

```
void reddisk::spawn ( ) [slot]
```

spawn disks and add them to the scene for them to fall from above and checks number of attempts

checks for number of attempts if attempts reach zero the player wins or loses depending on the score the score is saved in Scores2.txt file

spawns disk depending on timer used in [diskgame.cpp](#) which is set to 1500 ms

Author

Mohamad Abou Salem
Hadi Al Karhani

when attempts reach zero it checks if the player wins or loses

if less than 150 points player loses and his/her score is saved in Scores2.txt

open text file to save score

if greater than 150 points player wins and his/her score is saved in Scores2.txt along with the bonus points

open text file to save score

spawns a new disk each timer which is set to 1.5 seconds

adds disk to the scene

Here is the caller graph for this function:



5.12.3 Member Data Documentation

5.12.3.1 `uname`

```
QString reddisk::uname [private]
```

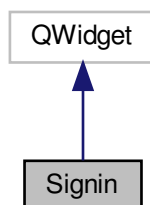
The documentation for this class was generated from the following files:

- Project435L/[reddisk.h](#)
- Project435L/[reddisk.cpp](#)

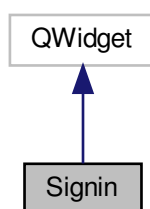
5.13 Signin Class Reference

```
#include <signin.h>
```

Inheritance diagram for Signin:



Collaboration diagram for Signin:



Public Member Functions

- [Signin](#) (QWidget *parent=nullptr)

Public Attributes

- QLabel * [username](#)
- QLineEdit * [usernameetxt](#)
- QLabel * [password](#)
- QLineEdit * [passwordtxt](#)
- QPushButton * [signup](#)
- QPushButton * [guest](#)
- QPushButton * [submit](#)
- QGridLayout * [grid1](#)
- QVBoxLayout * [box](#)

Private Slots

- void `signuppage` ()
goes to the sign up page
- void `signinvalidation` ()
checks whether the input credentials are correct or invalid
- void `playasguest` ()
user can play as a guest without signing up

5.13.1 Constructor & Destructor Documentation

5.13.1.1 Signin()

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

change the sign in background and text color

scales window to 400x200

moves window to the center of the screen when opened

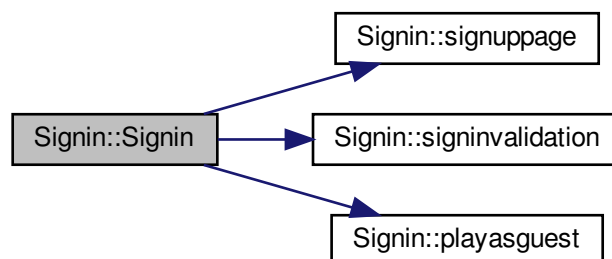
initializing all QLabels, QLineEdit, QRadioButtons, and QPushButton to display on the screen

navigate to the sign up page

checks if the input credentials are correct

user can play as a guest if they don't want to sign up

Here is the call graph for this function:



5.13.2 Member Function Documentation

5.13.2.1 playasguest

```
void Signin::playasguest ( ) [private], [slot]
```

user can play as a guest without signing up

its a function SLOT which is signaled by a press of a button when pressed the sign in page is closed and a profile page is opened

Author

Mohamad Abou Salem
Hadi Al Karhani

sign in instance is closed

profile instance is opened

Here is the caller graph for this function:



5.13.2.2 signinvalidation

```
void Signin::signinvalidation ( ) [private], [slot]
```

checks whether the input credentials are correct or invalid

its a function SLOT which is signaled by a press of a button when pressed the Database.txt file is checked if username and password match of the user input checks if all fields are filled and not empty

Author

Mohamad Abou Salem

Hadi Al Karhani

check username field if empty

check password field if empty

open text file

check if the username and password are correct

if correct sign in and go to the profile page

if incorrect, prompt the user with a message

Here is the caller graph for this function:

**5.13.2.3 signupage**

```
void Signin::signupage ( ) [private], [slot]
```

goes to the sign up page

its a function SLOT which is signaled by a press of a button when pressed the sign in page is closed and a sign up page is opened

Author

Mohamad Abou Salem

Hadi Al Karhani

close this instance of sign in page

open new instance of sign up page

Here is the caller graph for this function:



5.13.3 Member Data Documentation

5.13.3.1 box

`QVBoxLayout* Signin::box`

5.13.3.2 grid1

`QGridLayout* Signin::grid1`

5.13.3.3 guest

`QPushButton* Signin::guest`

5.13.3.4 password

`QLabel* Signin::password`

5.13.3.5 passwordtxt

`QLineEdit* Signin::passwordtxt`

5.13.3.6 signup

`QPushButton* Signin::signup`

5.13.3.7 submit

`QPushButton* Signin::submit`

5.13.3.8 username

```
QLabel* Signin::username
```

5.13.3.9 usernametxt

```
QLineEdit* Signin::usernametxt
```

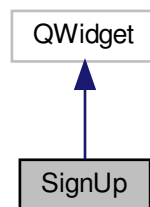
The documentation for this class was generated from the following files:

- [Project435L/signin.h](#)
- [Project435L/signin.cpp](#)

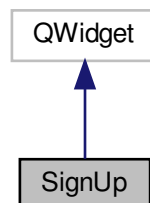
5.14 SignUp Class Reference

```
#include <signup.h>
```

Inheritance diagram for SignUp:



Collaboration diagram for SignUp:



Public Member Functions

- [SignUp](#) (QWidget *parent=nullptr)

Public Attributes

- QLabel * [firstname](#)
- QLineEdit * [firstnametxt](#)
- QLabel * [lastname](#)
- QLineEdit * [lastnametxt](#)
- QLabel * [username](#)
- QLineEdit * [usernameetxt](#)
- QLabel * [password](#)
- QLineEdit * [passwordtxt](#)
- QLabel * [gender](#)
- QRadioButton * [male](#)
- QRadioButton * [female](#)
- QLabel * [dob](#)
- QDateTimeEdit * [dobtxt](#)
- QLabel * [profilepicture](#)
- QPushButton * [profilepicturebr](#)
- QGroupBox * [group](#)
- QLabel * [phonenumber](#)
- QLineEdit * [phonenumbertxt](#)
- QPushButton * [submit](#)
- QString [s1](#)
- QPushButton * [back](#)
- QGridLayout * [grid1](#)
- QVBoxLayout * [box](#)
- QCheckBox * [showpass](#)
- QLineEdit * [fullname](#)
- QComboBox * [countrycode](#)

Private Slots

- void [goback](#) ()
goes back to the sign in page which is the main page
- void [submitreg](#) ()
Submits the data provided from the user to sign up and checks if username is taken or not.
- void [browse](#) ()
browses the computer to insert a profile picture

5.14.1 Constructor & Destructor Documentation

5.14.1.1 SignUp()

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

change the sign up background and text color

scales window to 400x300

moves window to the center of the screen when opened

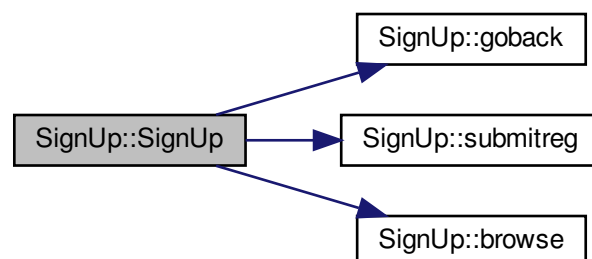
initializing all QLabels, QLineEdit, QRadioButtons, and QPushButton to display on the screen

navigate back to the sign in page which is the main page of this program

user submits the provided data

browse the computer to insert a picture

Here is the call graph for this function:



5.14.2 Member Function Documentation

5.14.2.1 browse

```
void SignUp::browse ( ) [private], [slot]
```

browses the computer to insert a profile picture

its a function SLOT which is signaled by a press of a button when pressed the user is prompted a file explorer window to navigate to the desired picture of choice if none are given, a default profile picture is set

Author

Mohamad Abou Salem
Hadi Al Karhani

Here is the caller graph for this function:



5.14.2.2 goback

```
void SignUp::goback ( ) [private], [slot]
```

goes back to the sign in page which is the main page

its a function SLOT which is signaled by a press of a button when pressed the sign up page is closed and a sign in page is opened

Author

Mohamad Abou Salem
Hadi Al Karhani

show instance of the sign in page

close this instance of sign up page

Here is the caller graph for this function:



5.14.2.3 submitreg

```
void SignUp::submitreg ( ) [private], [slot]
```

Submits the data provided from the user to sign up and checks if username is taken or not.

its a function SLOT which is signaled by a press of a button when pressed the sign up page is closed and a sign in page is opened the data is stored in txt file called Database.txt it checks whether the username is taken since the username is a unique key it checks if all of the fields are filled, and for valid phone number and password of at least 8 characters and contains at least one number, upper, and lower case letters

Author

Mohamad Abou Salem
Hadi Al Karhani

check if firstname field is empty

check if lastname field is empty

check if username field is empty

check if password field is empty

check if phonenumber field is empty

check if phonenumber field is valid for country specified, here it is Lebanon

check if phonenumber field is valid for country specified, here it is USA

check if phonenumber field is valid for country specified, here it is Saudi Arabia

check if phonenumber field is valid for country specified, here it is Belgium

check if phonenumber field is valid for country specified, here it is France

check if date of birth field is empty

check if password requirements are met

text file is open to save the data input

check if username already taken

if username take prompt user with message

if not taken then save all the data in the text file

Here is the caller graph for this function:



5.14.3 Member Data Documentation

5.14.3.1 back

`QPushButton* SignUp::back`

5.14.3.2 box

`QVBoxLayout* SignUp::box`

5.14.3.3 countrycode

`QComboBox* SignUp::countrycode`

5.14.3.4 dob

`QLabel* SignUp::dob`

5.14.3.5 dobtxt

`QDateTimeEdit* SignUp::dobtxt`

5.14.3.6 female

`QRadioButton* SignUp::female`

5.14.3.7 firstname

`QLabel* SignUp::firstname`

5.14.3.8 firstnametxt

`QLineEdit* SignUp::firstnametxt`

5.14.3.9 fullname

`QLineEdit* SignUp::fullname`

5.14.3.10 gender

`QLabel* SignUp::gender`

5.14.3.11 grid1

`QGridLayout* SignUp::grid1`

5.14.3.12 group

`QGroupBox* SignUp::group`

5.14.3.13 lastname

`QLabel* SignUp::lastname`

5.14.3.14 lastnametxt

`QLineEdit* SignUp::lastnametxt`

5.14.3.15 male

`QRadioButton* SignUp::male`

5.14.3.16 password

```
QLabel* SignUp::password
```

5.14.3.17 passwordtxt

```
QLineEdit* SignUp::passwordtxt
```

5.14.3.18 phonenumber

```
QLabel* SignUp::onenumber
```

5.14.3.19 phonenumbertxt

```
QLineEdit* SignUp::phonenumbertxt
```

5.14.3.20 profilepicture

```
QLabel* SignUp::profilepicture
```

5.14.3.21 profilepicturebr

```
QPushButton* SignUp::profilepicturebr
```

5.14.3.22 s1

```
QString SignUp::s1
```

5.14.3.23 showpass

```
QCheckBox* SignUp::showpass
```

5.14.3.24 submit

```
QPushButton* SignUp::submit
```

5.14.3.25 username

```
QLabel* SignUp::username
```

5.14.3.26 usernametxt

```
QLineEdit* SignUp::usernetxt
```

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

- Project435L/[signup.h](#)
- Project435L/[signup.cpp](#)

Chapter 6

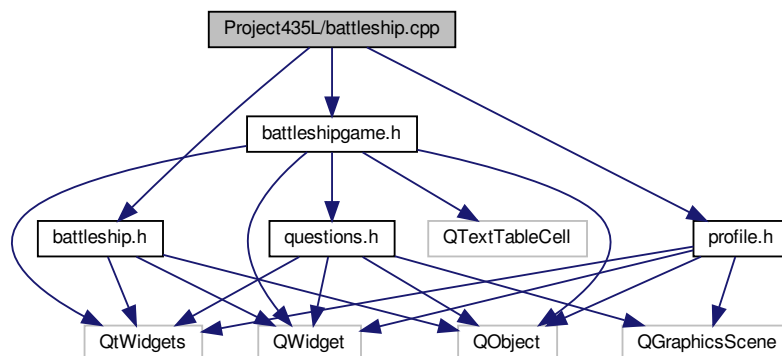
File Documentation

6.1 Project435L/battleship.cpp File Reference

contains battleship class definition

```
#include "battleship.h"  
#include "battleshipgame.h"  
#include "profile.h"
```

Include dependency graph for battleship.cpp:



6.1.1 Detailed Description

contains battleship class definition

it is the game menu of the battleship game allows player to start game or go back to the profile page

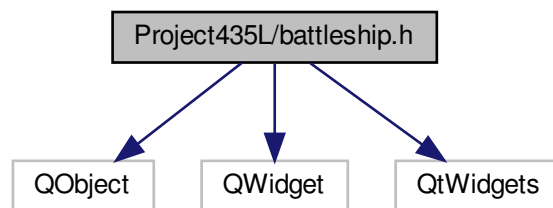
Author

Mohamad Abou Salem
Hadi Al Karhani

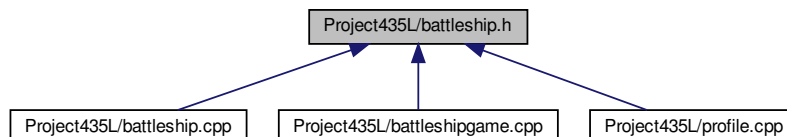
6.2 Project435L/battleship.h File Reference

battleship class

```
#include <QObject>
#include <QWidget>
#include <QtWidgets>
Include dependency graph for battleship.h:
```



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



Classes

- class [Battleship](#)

6.2.1 Detailed Description

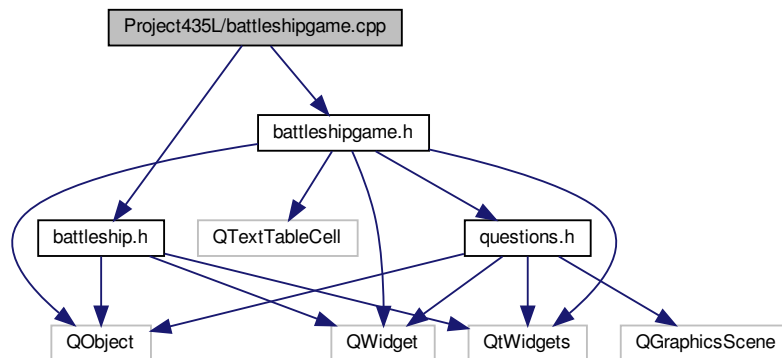
battleship class

initialize Keypressevent, a public slot called `back()` which is a void, a public slot called `startgame()` which is a void

6.3 Project435L/battleshipgame.cpp File Reference

contains battleshipgame class definition

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



Variables

- int `attempt` = 16
number of attempts is max 16
- int `correct` = 0
correct answers initialized to 0
- int `wrong` = 0
correct answers initialized to 0

6.3.1 Detailed Description

contains battleshipgame class definition

it is the battleship game with 2 4x4 grids the right 4x4 grid is the enemy, the left 4x4 grid is the player allows player to play a strategy and thinking game by answering questions provided if the player answers correctly, the player shoots enemy ship if player answers incorrectly, the player misses the shot and the enemy hits back randomly if the player clicks on a box where an enemy ship is not present, it is a direct miss and attempts are lost every click the player fails if 4 questions are answered incorrectly the player wins if 7 out of 10 question are answered correctly if attempts = 0, the player wins if number of questions correct are more than the incorrect questions answered

Author

Mohamad Abou Salem
Hadi Al Karhani

6.3.2 Variable Documentation

6.3.2.1 attempt

```
int attempt = 16
```

6.3.2.2 correct

```
int correct = 0
```

number of attempts is max 16

global variable number of attempts

6.3.2.3 wrong

```
int wrong = 0
```

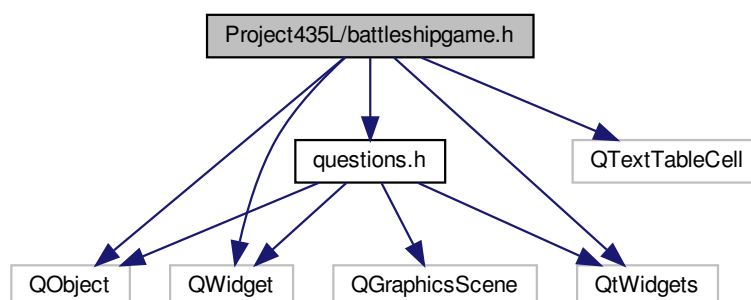
correct answers initialized to 0

global variable number of correct answers

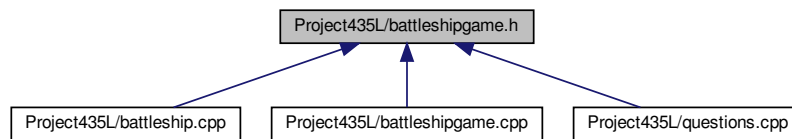
6.4 Project435L/battleshipgame.h File Reference

battleshipgame class

```
#include <QObject>
#include <QWidget>
#include <QtWidgets>
#include <QTextTableCell>
#include "questions.h"
Include dependency graph for battleshipgame.h:
```



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



Classes

- class [battleshipgame](#)

Variables

- int [attempt](#)
global variable number of attempts
- int [correct](#)
global variable number of correct answers
- int [wrong](#)
global variable number of correct answers

6.4.1 Detailed Description

battleshipgame class

initialize global variables attempts, wrong, correct, a public slot called exit() which is a void, public slot called shoot() which is a void, public slots called test() which is a void

6.4.2 Variable Documentation

6.4.2.1 attempt

```
int attempt
```

6.4.2.2 correct

```
int correct
```

global variable number of attempts

global variable number of attempts

6.4.2.3 wrong

```
int wrong
```

global variable number of correct answers

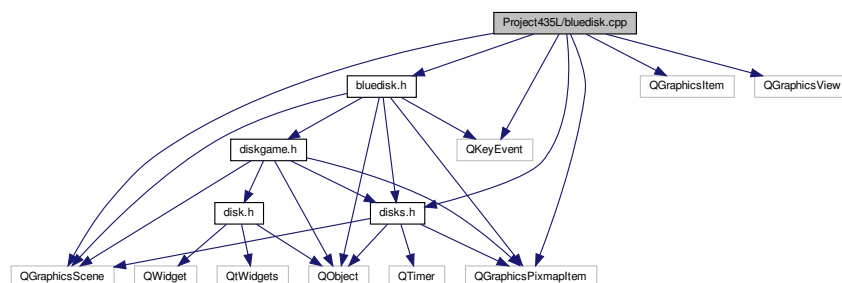
global variable number of correct answers

6.5 Project435L/bluedisk.cpp File Reference

contains disk pad of blue disk class definition

```
#include "bluedisk.h"
#include "disks.h"
#include <QGraphicsItem>
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QGraphicsPixmapItem>
#include <QKeyEvent>
```

Include dependency graph for bluedisk.cpp:



6.5.1 Detailed Description

contains disk pad of blue disk class definition

scales blue disk to a 40x40 scale sets position of the blue disk pad at 160, 340

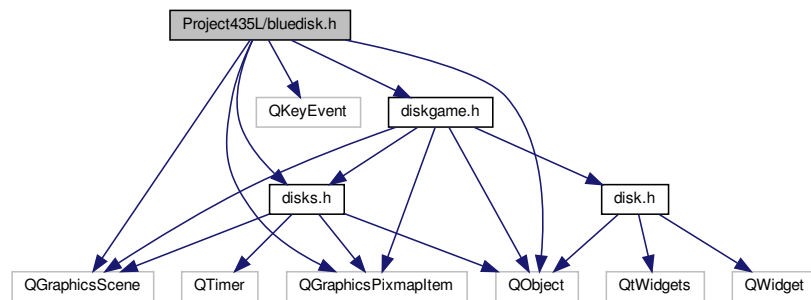
Author

Mohamad Abou Salem
Hadi Al Karhani

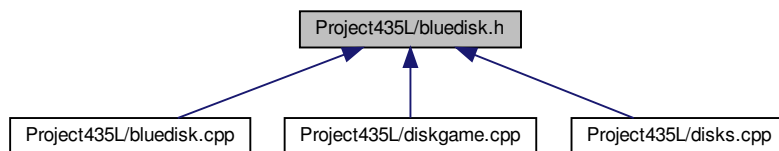
6.6 Project435L/bluedisk.h File Reference

bluedisk class

```
#include <QObject>
#include <QGraphicsScene>
#include <QGraphicsPixmapItem>
#include <QKeyEvent>
#include "disks.h"
#include "diskgame.h"
Include dependency graph for bluedisk.h:
```



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



Classes

- class [bluedisk](#)

6.6.1 Detailed Description

bluedisk class

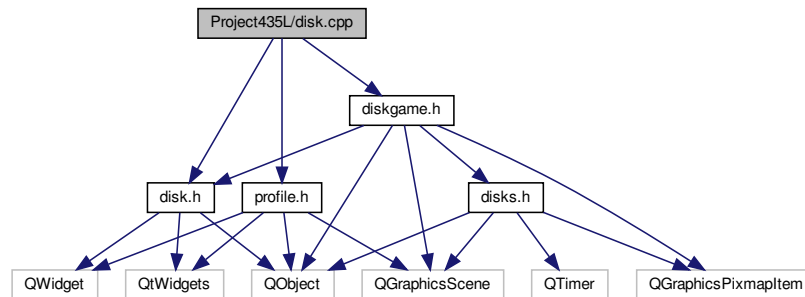
initialize keypressevent to know when disks collide with each other of the same colors using the right key and updates counters totalscore and blue

6.7 Project435L/disk.cpp File Reference

contains disk class definition

```
#include "disk.h"
#include "profile.h"
#include "diskgame.h"
```

Include dependency graph for disk.cpp:



6.7.1 Detailed Description

contains disk class definition

it is the game menu of the shooting discs allows player to start game or go back to the profile page

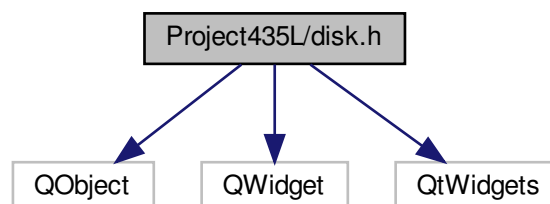
Author

Mohamad Abou Salem
Hadi Al Karhani

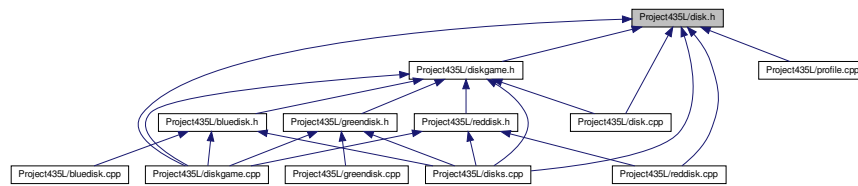
6.8 Project435L/disk.h File Reference

disk class

```
#include <QObject>
#include <QWidget>
#include <QtWidgets>
Include dependency graph for disk.h:
```



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



Classes

- class [disk](#)

6.8.1 Detailed Description

disk class

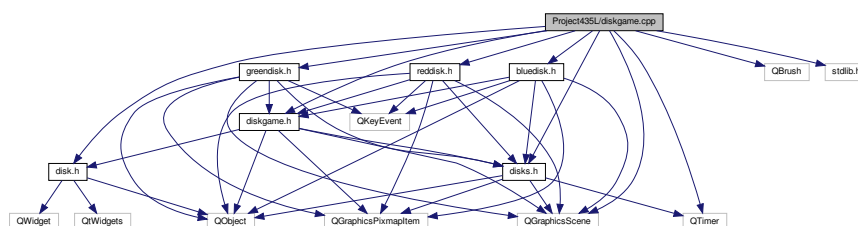
initialize keypressevent to start game using F1 key initializes QPushButton initializes two slots, back() and startgame() both are void

6.9 Project435L/diskgame.cpp File Reference

contains diskgame class definition

```
#include "diskgame.h"
#include "disks.h"
#include "reddisk.h"
#include <QGraphicsScene>
#include <QBrush>
#include <QTimer>
#include <stdlib.h>
#include "disk.h"
#include "greendisk.h"
#include "bluedisk.h"
```

Include dependency graph for diskgame.cpp:



Variables

- int `attempts` = 3
- int `red` = 0
max number of attempts is 3
- int `blue` = 0
red score initialized to 0
- int `green` = 0
blue score initialized to 0
- int `totalscore` = 0
green score initialized to 0

6.9.1 Detailed Description

contains diskgame class definition

sets disk pad at the bottom of the screen disks spawn and move downwards every 1.5 sec press right key to activate blue disk pad press down key to activate green disk pad press left key to activate red disk pad

Author

Mohamad Abou Salem
Hadi Al Karhani

6.9.2 Variable Documentation

6.9.2.1 attempts

```
int attempts = 3
```

6.9.2.2 blue

```
int blue = 0
```

red score initialized to 0

6.9.2.3 green

```
int green = 0
```

blue score initialized to 0

6.9.2.4 red

```
int red = 0
```

max number of attempts is 3

6.9.2.5 totalscore

```
int totalscore = 0
```

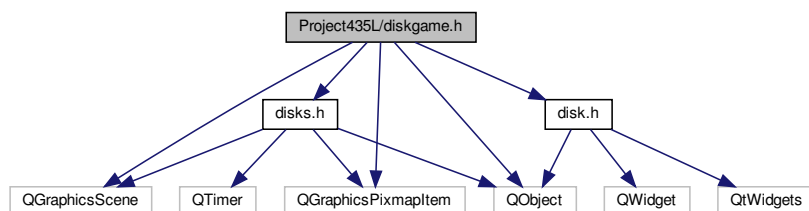
green score initialized to 0

6.10 Project435L/diskgame.h File Reference

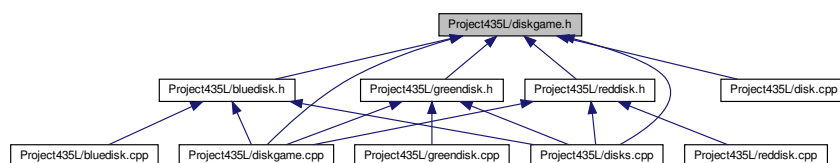
diskgame class

```
#include <QObject>
#include <QGraphicsScene>
#include <QGraphicsPixmapItem>
#include "disks.h"
#include "disk.h"
```

Include dependency graph for diskgame.h:



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



Classes

- class `diskgame`

Variables

- int `attempts`
- int `red`
max number of attempts is 3
- int `blue`
red score initialized to 0
- int `green`
blue score initialized to 0
- int `totalscore`
green score initialized to 0

6.10.1 Detailed Description

`diskgame` class

initialize global variables `attempts`, `red`, `blue`, `green`, and `totalscore`

6.10.2 Variable Documentation

6.10.2.1 `attempts`

```
int attempts
```

6.10.2.2 `blue`

```
int blue
```

red score initialized to 0

6.10.2.3 `green`

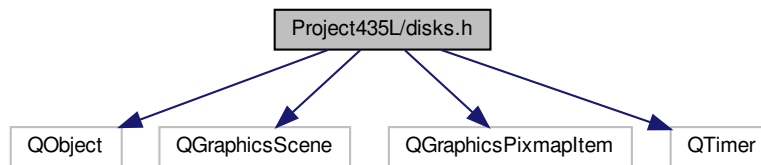
```
int green
```

blue score initialized to 0

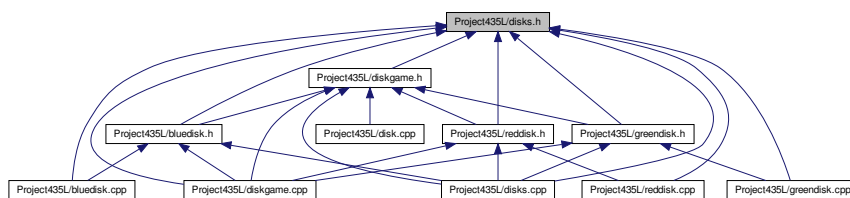
6.12 Project435L/disks.h File Reference

disks class

```
#include <QObject>
#include <QGraphicsScene>
#include <QGraphicsPixmapItem>
#include <QTimer>
Include dependency graph for disks.h:
```



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



Classes

- class [disks](#)

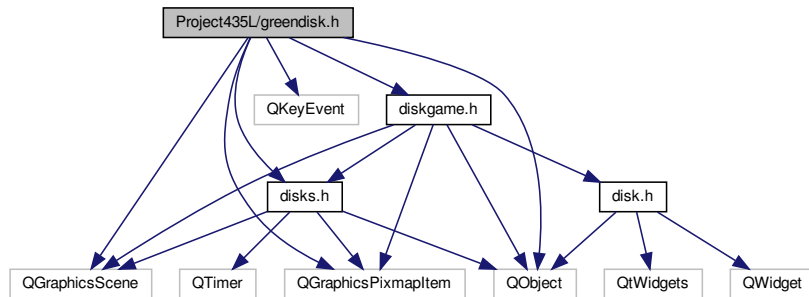
6.12.1 Detailed Description

disks class

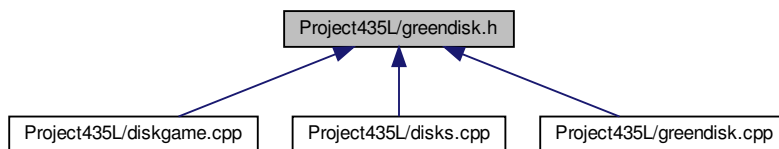
initialize slot called move() which is a void initialize QString to save username in


```
#include "diskgame.h"
```

Include dependency graph for greendisk.h:



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



Classes

- class [greendisk](#)

6.14.1 Detailed Description

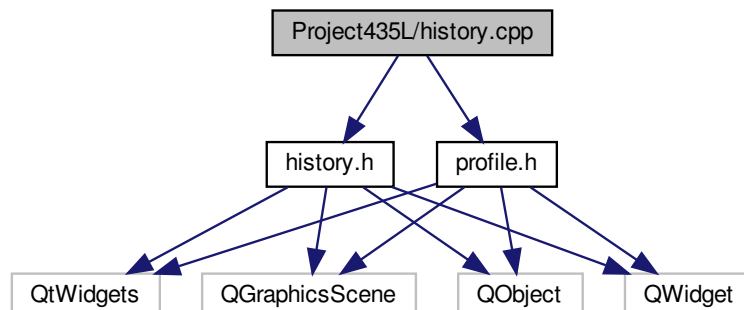
greendisk class

initialize keypressevent to know when disks collide with each other of the same colors using the down key

6.15 Project435L/history.cpp File Reference

contains user history of game1 which is the shooting discs game

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



6.15.1 Detailed Description

contains user history of game1 which is the shooting discs game

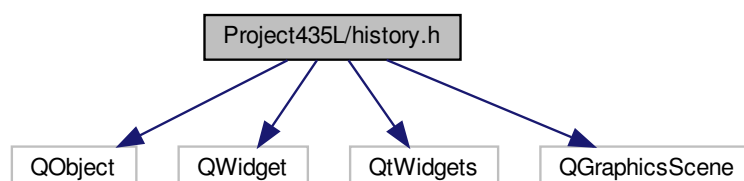
keeps track of user scores which are saved in Scores.txt file

Author

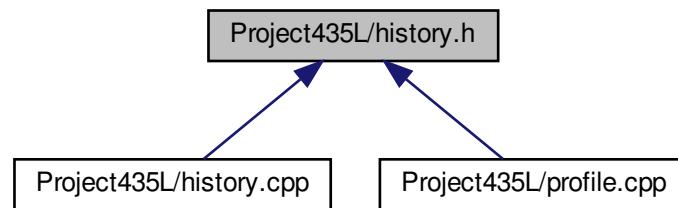
Mohamad Abou Salem
Hadi Al Karhani

6.16 Project435L/history.h File Reference

```
#include <QObject>
#include <QWidget>
#include <QtWidgets>
#include <QGraphicsScene>
Include dependency graph for history.h:
```



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



Classes

- class [history](#)

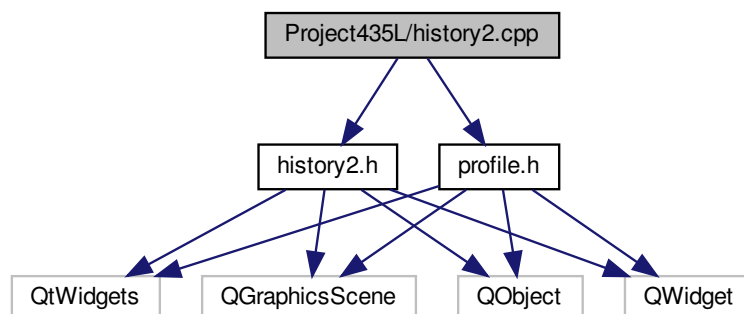
6.17 Project435L/history2.cpp File Reference

contains user history of game2 which is the shooting discs game

```
#include "history2.h"
```

```
#include "profile.h"
```

Include dependency graph for history2.cpp:



6.17.1 Detailed Description

contains user history of game2 which is the shooting discs game

keeps track of user scores which are saved in Scores2.txt file

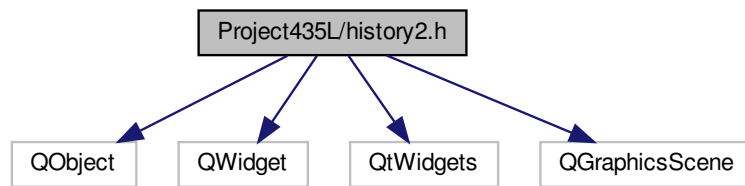
Author

Mohamad Abou Salem
Hadi Al Karhani

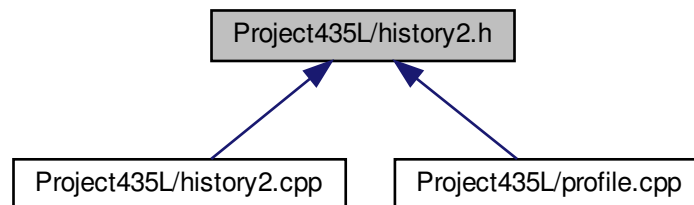
6.18 Project435L/history2.h File Reference

`history2` class

```
#include <QObject>
#include <QWidget>
#include <QtWidgets>
#include <QGraphicsScene>
Include dependency graph for history2.h:
```



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



Classes

- class `history2`

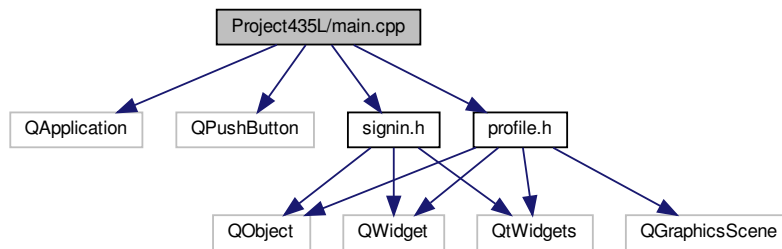
6.18.1 Detailed Description

`history2` class

initialize slot called `goback()` which is a void initialize `QStrings`, `QLabels`, `QPushButtons`

6.19 Project435L/main.cpp File Reference

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



Functions

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

6.19.1 Function Documentation

6.19.1.1 main()

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

< set window to fixed size 400x200

6.20 Project435L/profile.cpp File Reference

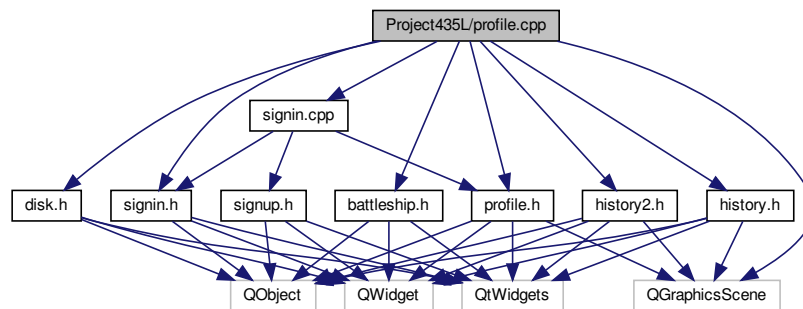
contains user profile which displays current date & time, profile pic, full name, history of both games, buttons to play both games, and country flag depending on phone number inserted

```
#include "profile.h"
#include "signin.h"
#include "signin.cpp"
#include "history.h"
#include "history2.h"
#include "battleship.h"
#include "disk.h"
```



```
#include <QGraphicsScene>
```

Include dependency graph for profile.cpp:



6.20.1 Detailed Description

contains user profile which displays current date & time, profile pic, full name, history of both games, buttons to play both games, and country flag depending on phone number inserted

Author

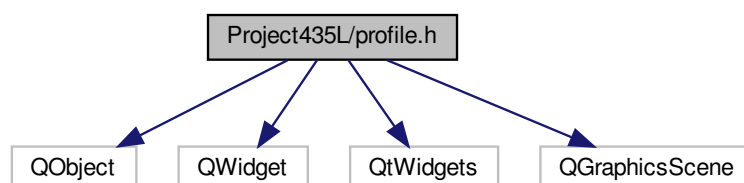
Mohamad Abou Salem
Hadi Al Karhani

6.21 Project435L/profile.h File Reference

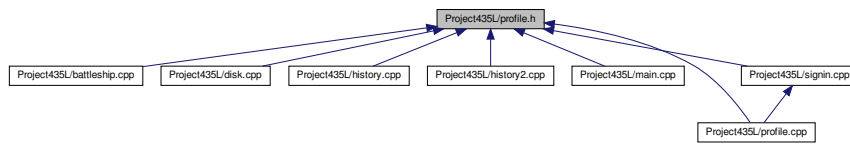
profile class

```
#include <QObject>
#include <QWidget>
#include <QtWidgets>
#include <QGraphicsScene>
```

Include dependency graph for profile.h:



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



Classes

- class [profile](#)

6.21.1 Detailed Description

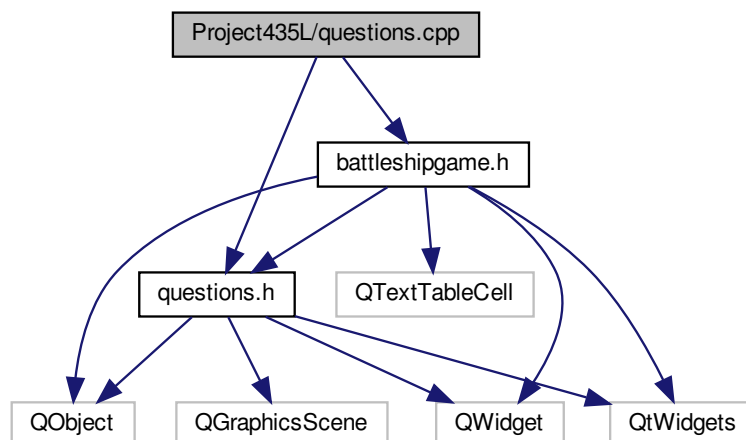
profile class

initialize slots called goback(), playgame1(), playgame2(), gohistory(), gohistory2() which are all void initialize Q←Strings, QLabels, QPushbuttons

6.22 Project435L/questions.cpp File Reference

contains questions to be asked at every turn of the battleship game

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



6.22.1 Detailed Description

contains questions to be asked at every turn of the battleship game

provides questions at random saved in Questions.txt

Author

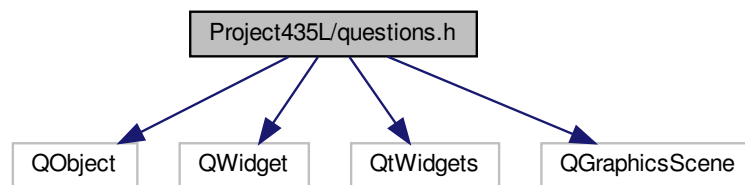
Mohamad Abou Salem
Hadi Al Karhani

6.23 Project435L/questions.h File Reference

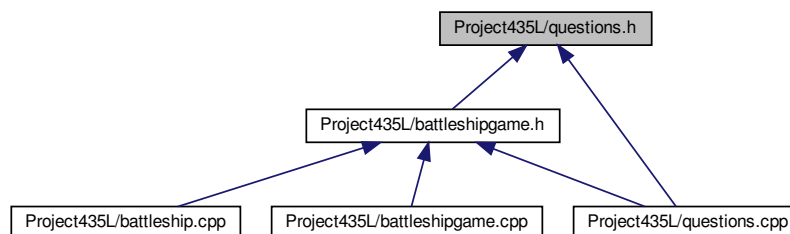
questions class

```
#include <QObject>
#include <QWidget>
#include <QtWidgets>
#include <QGraphicsScene>
```

Include dependency graph for questions.h:



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



Classes

- class [questions](#)

6.23.1 Detailed Description

questions class

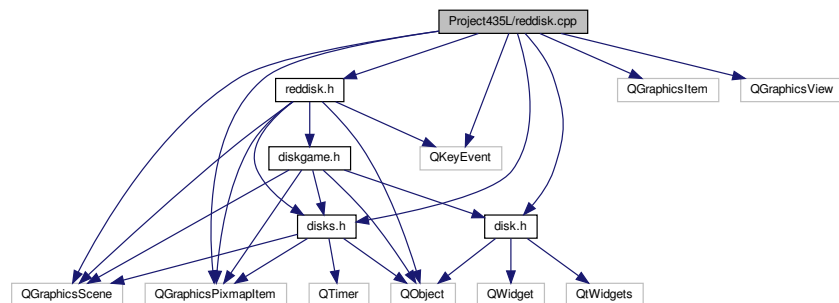
initialize slots called submit() which is an int initialize signal called sigemit() which is an int initialize QRadioButton, QLabels, QPushbuttons

6.24 Project435L/reddisk.cpp File Reference

contains disk pad of red disk class definition

```
#include "reddisk.h"
#include "disks.h"
#include <QGraphicsItem>
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QGraphicsPixmapItem>
#include <QKeyEvent>
#include "disk.h"
```

Include dependency graph for reddisk.cpp:



6.24.1 Detailed Description

contains disk pad of red disk class definition

scales red disk to a 40x40 scale sets position of the red disk pad at 0, 340

Author

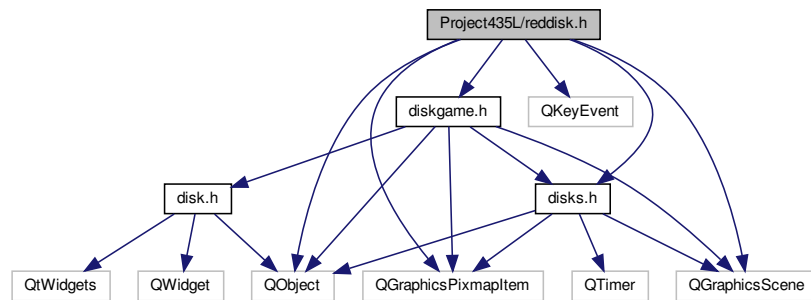
Mohamad Abou Salem
Hadi Al Karhani

6.25 Project435L/reddisk.h File Reference

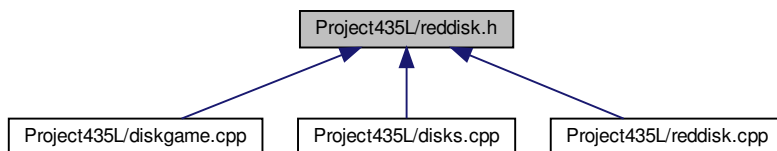
reddisk class

```
#include <QObject>
#include <QGraphicsScene>
#include <QGraphicsPixmapItem>
#include <QKeyEvent>
#include "diskgame.h"
#include "disks.h"
```

Include dependency graph for reddisk.h:



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



Classes

- class [reddisk](#)

6.25.1 Detailed Description

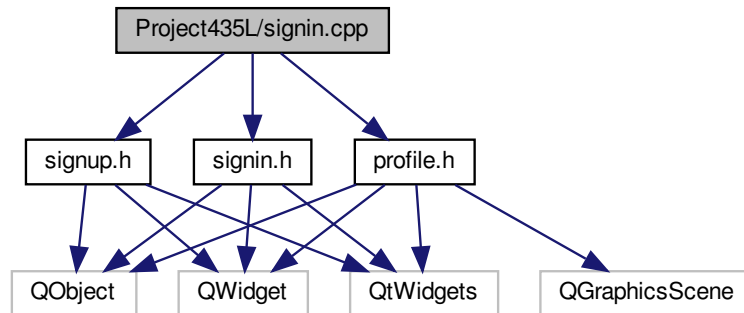
reddisk class

initialize keypressevent to know when disks collide with each other of the same colors using the left key and updates counters totalscore and red initialize slot called spawn() which is a void

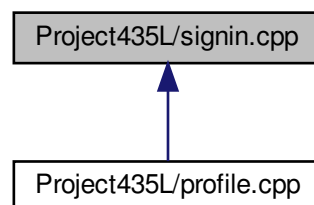
6.26 Project435L/signin.cpp File Reference

contains sign in class definition

```
#include "signin.h"
#include "signup.h"
#include "profile.h"
Include dependency graph for signin.cpp:
```



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



6.26.1 Detailed Description

contains sign in class definition

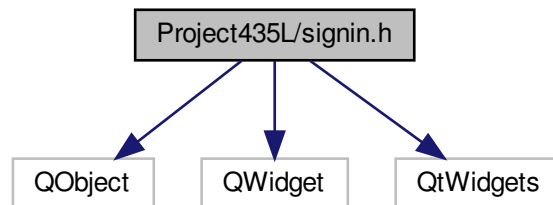
allows user to sign in and checks username and password in a txt file called Database.txt if the credentials are correct or invalid user can play as a guest if they don't want to sign up user can navigate to the sign up page

Author

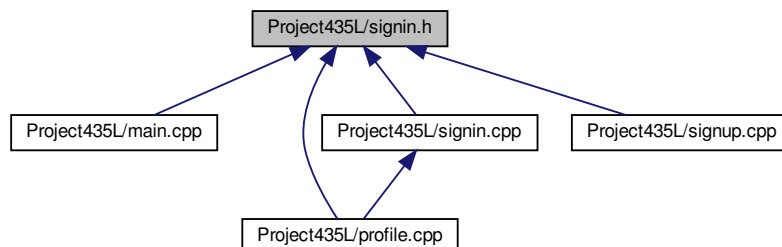
Mohamad Abou Salem
Hadi Al Karhani

6.27 Project435L/signin.h File Reference

```
#include <QObject>
#include <QWidget>
#include <QtWidgets>
Include dependency graph for signin.h:
```



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



Classes

- class [Signin](#)

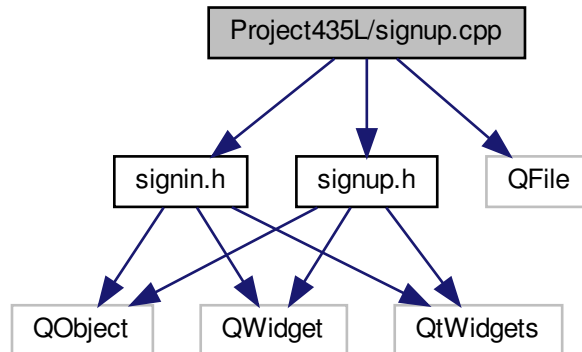
6.28 Project435L/signup.cpp File Reference

contains sign up class definition

```
#include "signup.h"
#include "signin.h"
```

```
#include <QFile>
```

Include dependency graph for signup.cpp:



6.28.1 Detailed Description

contains sign up class definition

allows user to sign up and saves the data provided in a txt file called Database.txt

Author

Mohamad Abou Salem
Hadi Al Karhani

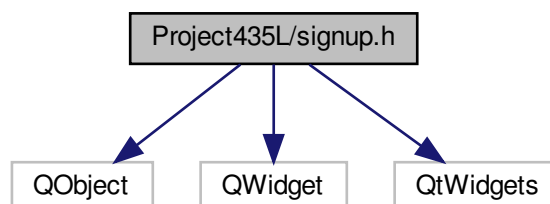
6.29 Project435L/signup.h File Reference

```
#include <QObject>
```

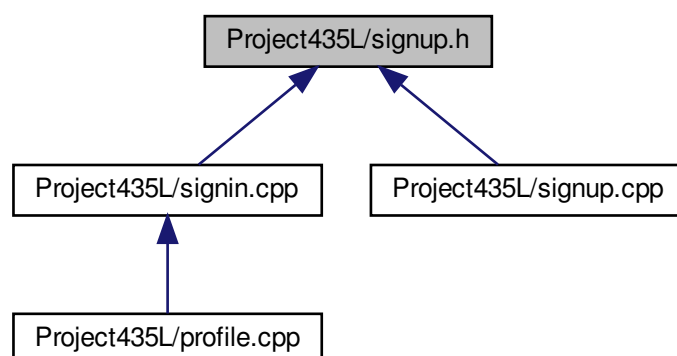
```
#include <QWidget>
```

```
#include <QtWidgets>
```

Include dependency graph for signup.h:



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



Classes

- class [SignUp](#)

