

My Project

Generated by Doxygen 1.8.6

Sat Nov 9 2019 12:36:37

Contents

1	Phase 2 Submission	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	birthdaywidget Class Reference	9
5.1.1	Member Function Documentation	9
5.1.1.1	confirmExit	9
5.2	gameScene_1 Class Reference	10
5.2.1	Member Function Documentation	11
5.2.1.1	backToMenu	11
5.2.1.2	diceRoll	11
5.2.1.3	generateBoard	11
5.2.1.4	loadGame	12
5.2.1.5	movePlayer	12
5.2.1.6	movePlayerOpp	12
5.2.1.7	selectLeft	12
5.2.1.8	selectRight	12
5.2.1.9	winCondition	12
5.3	gameScene_1_dice Class Reference	12
5.3.1	Member Function Documentation	13
5.3.1.1	setup	13
5.3.1.2	update	13
5.4	gamescene_1_ladderSnake Class Reference	13
5.4.1	Member Function Documentation	14
5.4.1.1	getXYcase	14

5.4.1.2	initLadder	15
5.5	gameScene_1_player Class Reference	15
5.5.1	Member Function Documentation	16
5.5.1.1	moveInstantaneous	16
5.5.1.2	setup	16
5.6	loginWidget Class Reference	16
5.6.1	Member Function Documentation	17
5.6.1.1	checkLogin	17
5.6.1.2	GoToMainAsGuest	17
5.6.1.3	GoToRegisterPage	17
5.7	menuWidget Class Reference	17
5.7.1	Member Function Documentation	19
5.7.1.1	loadGame	19
5.7.1.2	readFromJsonProfile	19
5.7.1.3	startGameCPU	19
5.7.1.4	startGameMultiplayer	20
5.8	registerWidget Class Reference	20
5.8.1	Member Function Documentation	21
5.8.1.1	cancelRegistration	21
5.8.1.2	checkConditions	22
5.8.1.3	confirmRegistration	22
5.8.1.4	passCheck	22
5.8.1.5	selectPicture	22
6	File Documentation	23
6.1	birthdaywidget.cpp File Reference	23
6.1.1	Detailed Description	23
6.2	birthdaywidget.h File Reference	23
6.2.1	Detailed Description	23
6.3	gamescene_1.h File Reference	23
6.3.1	Detailed Description	24
6.4	gamescene_1_dice.h File Reference	24
6.4.1	Detailed Description	24
6.5	menuwidget.cpp File Reference	24
6.5.1	Detailed Description	25
6.6	menuwidget.h File Reference	25
6.6.1	Detailed Description	25
6.7	registerwidget.cpp File Reference	25
6.7.1	Detailed Description	25

Chapter 1

Phase 2 Submission

Author

Mehieddine Zeidan
Hadi Sandid

Date

9-10-2019

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

QGraphicsPixmapItem	
gameScene_1_dice	12
gameScene_1_ladderSnake	13
gameScene_1_player	15
QGraphicsScene	
gameScene_1	10
QObject	
gameScene_1_dice	12
gameScene_1_ladderSnake	13
gameScene_1_player	15
QWidget	
birthdaywidget	9
loginWidget	16
menuWidget	17
registerWidget	20

Chapter 3

Class Index

3.1 Class List

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

birthdaywidget	9
gameScene_1	10
gameScene_1_dice	12
gamescene_1_ladderSnake	13
gameScene_1_player	15
loginWidget	16
menuWidget	17
registerWidget	20

Chapter 4

File Index

4.1 File List

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

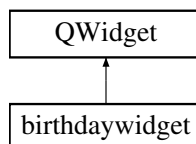
birthdaywidget.cpp	
Contains birthdaywidget class definition	23
birthdaywidget.h	
The birthdaywidget class	23
gamescene_1.h	
The gameScene_1 class	23
gamescene_1_dice.h	
The gameScene_1_dice class	24
gamescene_1_ladderSnake.h	??
gamescene_1_player.h	??
loginwidget.h	??
menuwidget.cpp	
Contains menuWidget class definition	24
menuwidget.h	
The menuWidget class	25
registerwidget.cpp	
Contains registerWidget class definition	25
registerwidget.h	??

Chapter 5

Class Documentation

5.1 birthdaywidget Class Reference

Inheritance diagram for birthdaywidget:



Public Slots

- void [confirmExit](#) ()
[birthdaywidget::confirmExit](#)

Public Member Functions

- **birthdaywidget** (QWidget *parent=0)

Public Attributes

- QVBoxLayout * [layout](#)
VBox Layout.
- QLabel * [bdtext](#)
Main text.
- QPushButton * [conf](#)
Main Button.

5.1.1 Member Function Documentation

5.1.1.1 void birthdaywidget::confirmExit () [slot]

[birthdaywidget::confirmExit](#)

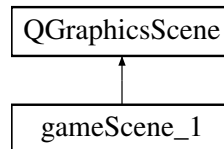
Exit from birthday message

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

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

5.2 gameScene_1 Class Reference

Inheritance diagram for gameScene_1:



Public Slots

- void [generateBoard](#) ()
gameScene_1::generateBoard
- void [selectLeft](#) ()
gameScene_1::selectLeft
- void [selectRight](#) ()
gameScene_1::selectRight
- void [diceRoll](#) ()
gameScene_1::diceRoll
- void **checkSnakeLadder** ()
- void [winCondition](#) ()
gameScene_1::winCondition
- void [movePlayer](#) ()
gameScene_1::movePlayer
- void [movePlayerOpp](#) ()
gameScene_1::movePlayerOpp
- void [backToMenu](#) ()
gameScene_1::backToMenu
- void [loadGame](#) (int posp1, int posp2, bool turn)
gameScene_1::loadGame

Public Member Functions

- **gameScene_1** (QObject *parent=0)

Public Attributes

- QGraphicsTextItem * [text](#)
Text element in GUI.
- QGraphicsTextItem * [text2](#)
Text element in GUI.
- QGraphicsTextItem * [text3](#)
Text element in GUI.
- QPushButton * [rolldice](#)
Button associated to rolling the dices.

- QPushButton * [pickdice1](#)
Button associated to pick a dice.
- QPushButton * [pickdice2](#)
Button associated to pick a dice.
- QPushButton * [backtomenu](#)
Button associated to going back to menu.
- [gameScene_1_player](#) * [player1](#)
Player 1 entity.
- [gameScene_1_player](#) * [player2](#)
Player 2 entity.
- [gameScene_1_dice](#) * [dice1](#)
Dice 1 entity.
- [gameScene_1_dice](#) * [dice2](#)
Dice 2 entity.
- [gamescene_1_ladderSnake](#) * [ladderSnake](#)
Ladder&Snake entity.
- QGraphicsRectItem * [square](#)
Boxes used to denote ladder/snakes.
- int * [ladderPos](#)
Array used to store Ladder start/end positions.
- int * [snakePos](#)
Array used to store Snake start/end positions.
- bool * [firstRoll](#)
Bool denoting first roll status.
- bool * [playerTurn](#)
Bool denoting player turn.
- bool * [sidePick](#)
Bool denoting left/right dice pick.
- bool * [winState](#)
Bool denoting win state.
- bool * [CPU](#)
Bool denoting CPU adversary / human adversary.

5.2.1 Member Function Documentation

5.2.1.1 void [gameScene_1::backToMenu](#) () [slot]

[gameScene_1::backToMenu](#)

Go back to main menu

5.2.1.2 void [gameScene_1::diceRoll](#) () [slot]

[gameScene_1::diceRoll](#)

Roll dice & implements game loop

5.2.1.3 void [gameScene_1::generateBoard](#) () [slot]

[gameScene_1::generateBoard](#)

Generate board dynamically (add snakes/ladders) from a text file

5.2.1.4 void gameScene_1::loadGame (int *posp1*, int *posp2*, bool *turn*) [slot]

[gameScene_1::loadGame](#)

Parameters

<i>posp1</i>	: player1 position
<i>posp2</i>	: player2 position
<i>turn</i>	: bool which denotes which player's turn it is

Loads existing game, if there is any

5.2.1.5 void gameScene_1::movePlayer () [slot]

[gameScene_1::movePlayer](#)

Move player after a dice roll

5.2.1.6 void gameScene_1::movePlayerOpp () [slot]

[gameScene_1::movePlayerOpp](#)

Move player after a dice roll

5.2.1.7 void gameScene_1::selectLeft () [slot]

[gameScene_1::selectLeft](#)

This function is executed when the player selects the left dice

5.2.1.8 void gameScene_1::selectRight () [slot]

[gameScene_1::selectRight](#)

This function is executed when the player selects the right dice

5.2.1.9 void gameScene_1::winCondition () [slot]

[gameScene_1::winCondition](#)

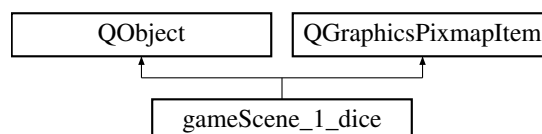
Check if the win condition (player reaches case 100) is satisfied

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

- [gamescene_1.h](#)
- [gamescene_1.cpp](#)

5.3 gameScene_1_dice Class Reference

Inheritance diagram for gameScene_1_dice:



Public Slots

- void [update](#) ()
[gameScene_1_dice::update](#)
- void [setup](#) (int value)
[gameScene_1_dice::setup](#)

Public Member Functions

- **gameScene_1_dice** (QObject *parent=0)

Public Attributes

- int * [id](#)
ID of the dice (dice1/dice2)
- int * [diceVal](#)
int denoting current value of dice

5.3.1 Member Function Documentation

5.3.1.1 void [gameScene_1_dice::setup](#) (int *value*) [slot]

[gameScene_1_dice::setup](#)

Parameters

<i>value</i>	: id of dice (either 1 or 2)
--------------	------------------------------

Setup a dice instance

5.3.1.2 void [gameScene_1_dice::update](#) () [slot]

[gameScene_1_dice::update](#)

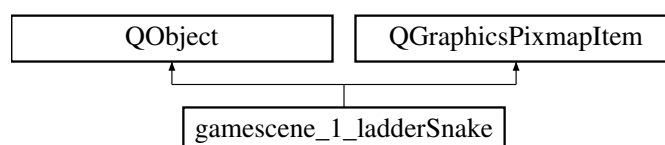
Roll the dice and update its value randomly

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

- [gamescene_1_dice.h](#)
- [gamescene_1_dice.cpp](#)

5.4 gamescene_1_ladderSnake Class Reference

Inheritance diagram for gamescene_1_ladderSnake:



Public Slots

- void [initLadder](#) (int seed, int start, int end)
[gamescene_1_ladderSnake::initLadder](#)
- int [getXYcase](#) (int caseNum, int xy)
[gamescene_1_ladderSnake::getXYcase](#)

Public Member Functions

- **gamescene_1_ladderSnake** (QObject *parent=0)

Public Attributes

- QPixmap * [image](#)
Image associated to ladder/snake.
- int * [id](#)
Id associated to ladder/snake.
- bool * [goForward](#)
Bool associated to movement logic.
- int * [posXstart](#)
Temp value used in calculations across functions.
- int * [posXend](#)
Temp value used in calculations across functions.
- int * [posYstart](#)
Temp value used in calculations across functions.
- int * [posYend](#)
Temp value used in calculations across functions.
- int * [posXtemp](#)
Temp value used in calculations across functions.
- int * [posYtemp](#)
Temp value used in calculations across functions.
- int * [posXtempFunc](#)
Temp value used in calculations across functions.
- int * [posYtempFunc](#)
Temp value used in calculations across functions.
- double * [size](#)
Temp value used in calculations across functions.
- double * [angle](#)
Temp value used in calculations across functions.
- double * [centerX](#)
Temp value used in calculations across functions.
- double * [centerY](#)
Temp value used in calculations across functions.
- double * [tempLength](#)
Temp value used in calculations across functions.

5.4.1 Member Function Documentation

5.4.1.1 int [gamescene_1_ladderSnake::getXYcase](#) (int *caseNum*, int *xy*) [slot]

[gamescene_1_ladderSnake::getXYcase](#)

Parameters

<i>caseNum</i>	: value of the case to search
<i>xy</i>	: xy=0 return x value , xy=1 return y value

Returns

x coordinate of the case

When assigning a case to a ladder/snake, we search for its coordinates using this function

5.4.1.2 void gameScene_1_ladderSnake::initLadder (int *seed*, int *start*, int *end*) [slot]

[gameScene_1_ladderSnake::initLadder](#)

Parameters

<i>seed</i>	: 0-1 are ladder instance seeds, 2-3-4 are snake instance seeds
<i>start</i>	: where is the instance start position
<i>end</i>	: where is the instance end position

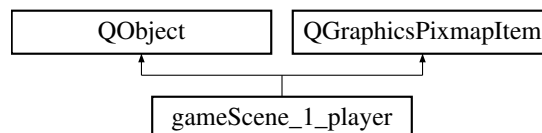
Initialize ladder or snake instance with given parameters

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

- gameScene_1_ladderSnake.h
- gameScene_1_ladderSnake.cpp

5.5 gameScene_1_player Class Reference

Inheritance diagram for gameScene_1_player:



Public Slots

- void [moveInstantaneous](#) ()
[gameScene_1_player::moveInstantaneous](#)
- void [setup](#) (int playerId)
[gameScene_1_player::setup](#)

Public Member Functions

- **gameScene_1_player** (QObject *parent=0)

Public Attributes

- QGraphicsScene * **parentScene**
- bool * [goForward](#)
Bool used in movement logic.

- int * [id](#)
ID used for player1/player2.
- int * [position](#)
Used to store player position.
- int * [newPosition](#)
Used to store player position.

5.5.1 Member Function Documentation

5.5.1.1 void [gameScene_1_player::moveInstantaneous](#) () [[slot](#)]

[gameScene_1_player::moveInstantaneous](#)

Move the player to its newly assigned position

5.5.1.2 void [gameScene_1_player::setup](#) (int *playerid*) [[slot](#)]

[gameScene_1_player::setup](#)

Parameters

<i>playerid</i>	: id for player
-----------------	-----------------

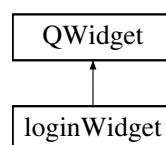
Initializing the player instance

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

- [gamescene_1_player.h](#)
- [gamescene_1_player.cpp](#)

5.6 loginWidget Class Reference

Inheritance diagram for loginWidget:



Public Slots

- void [GoToRegisterPage](#) ()
[loginWidget::GoToRegisterPage](#)
- void [checkLogin](#) ()
[loginWidget::checkLogin](#)
- void [GoToMainAsGuest](#) ()
[loginWidget::GoToMainAsGuest](#)

Public Member Functions

- **loginWidget** (QWidget *parent=0)

Public Attributes

- QLabel * [test_screen](#)
Text label.
- QPushButton * [LogInbutton](#)
Menu button to login.
- QPushButton * [RegisterButton](#)
Menu button to register.
- QGridLayout * [test_layout](#)
Main Layout.
- QLabel * [UserName](#)
Text label.
- QLabel * [PassWord](#)
Text label.
- QLineEdit * [UserNameLine](#)
Line edit field.
- QLineEdit * [PassWordLine](#)
Line edit field.
- QLabel * [ErrorMessage](#)
Text label.
- QPushButton * [PlayAsGuestButton](#)
Menu button to play as guest.

5.6.1 Member Function Documentation

5.6.1.1 void loginWidget::checkLogin () [slot]

[loginWidget::checkLogin](#)

Check if login conditions are met, and login with a particular user

5.6.1.2 void loginWidget::GoToMainAsGuest () [slot]

[loginWidget::GoToMainAsGuest](#)

Open main menu as a guest

5.6.1.3 void loginWidget::GoToRegisterPage () [slot]

[loginWidget::GoToRegisterPage](#)

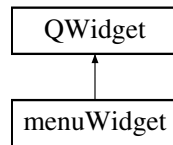
Load up register page

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

- loginwidget.h
- loginwidget.cpp

5.7 menuWidget Class Reference

Inheritance diagram for menuWidget:



Public Slots

- void [readFromJsonProfile](#) ()
menuWidget::readFromJsonProfile
- void **logout** ()
- void [startGameCPU](#) ()
menuWidget::startGameCPU
- void [startGameMultiplayer](#) ()
menuWidget::startGameMultiplayer
- void [loadGame](#) ()
menuWidget::loadGame

Public Member Functions

- **menuWidget** (QWidget *parent=0)

Public Attributes

- bool * [isGuest](#)
Bool which denotes whether the logged in user is a guest.
- QGridLayout * [test_layout](#)
Main Layout.
- QGridLayout * [play_layout](#)
Tab Layout.
- QGridLayout * [profile_layout](#)
Tab Layout.
- QGridLayout * [history_layout](#)
Tab Layout.
- QWidget * [play](#)
Tab Widget.
- QWidget * [profile](#)
Tab Widget.
- QWidget * [history](#)
Tab Widget.
- QTabWidget * [tabWidget](#)
Main Tab Widget.
- QLabel * [Welcome](#)
Text label.
- QLabel * [Game1Name](#)
Text label.
- QLabel * [Game2Name](#)
Text label.
- QPushButton * [Play1](#)
Button which opens Game1 VS CPU.

- QPushButton * [Play1x](#)
Button which opens Game1 Multiplayer.
- QPushButton * [Play2](#)
Button which opens Game2.
- QPushButton * [Play2x](#)
Button which opens Game2.
- QLabel * [image1](#)
Image in menu.
- QLabel * [image2](#)
Image in menu.
- QPushButton * [Load1](#)
Button which loads Game1 save.
- QPushButton * [Load2](#)
Button which loads Game2 save.
- QPushButton * [Logout](#)
Button which returns to logon menu.
- QLabel * [usernameDisplay](#)
Personnal Info display (username)
- QLabel * [ppDisplay](#)
Personnal Info display (profile picture)
- QLabel * [fnameDisplay](#)
Personnal Info display (fname)
- QLabel * [lnameDisplay](#)
Personnal Info display (lname)
- QLabel * [dobDisplay](#)
Personnal Info display (date of birth)
- QLabel * [genderDisplay](#)
Personnal Info display (gender)

5.7.1 Member Function Documentation

5.7.1.1 void menuWidget::loadGame () [slot]

[menuWidget::loadGame](#)

Check if a loadable game exists and allow the user to load a game from the main menu

5.7.1.2 void menuWidget::readFromJsonProfile () [slot]

[menuWidget::readFromJsonProfile](#)

Read user data from the userdata.json file

5.7.1.3 void menuWidget::startGameCPU () [slot]

[menuWidget::startGameCPU](#)

Start a new instance of Game1 vs CPU

5.7.1.4 void menuWidget::startGameMultiplayer () [slot]

[menuWidget::startGameMultiplayer](#)

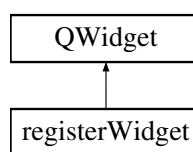
Start a new instance of Game1 multiplayer

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

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

5.8 registerWidget Class Reference

Inheritance diagram for registerWidget:



Public Slots

- void [selectPicture](#) ()
registerWidget::selectPicture
- void [cancelRegistration](#) ()
registerWidget::cancelRegistration
- void [confirmRegistration](#) ()
registerWidget::confirmRegistration
- void [checkConditions](#) ()
registerWidget::checkConditions
- bool [passCheck](#) ()
registerWidget::passCheck

Public Member Functions

- **registerWidget** (QWidget *parent=0)

Public Attributes

- bool [isSelected](#)
bool associated to profile picture selected
- QLabel * [welcome](#)
text label
- QLabel * [fname](#)
text label
- QLineEdit * [fname_edit](#)
personal info input (fname)
- QLabel * [lname](#)
text label
- QLineEdit * [lname_edit](#)

- personal info input (lname)*
- QLabel * [username](#)
text label
- QLineEdit * [username_edit](#)
personal info input (username)
- QLabel * [password](#)
text label
- QLineEdit * [password_edit](#)
personal info input (password)
- QLabel * [passwordconf](#)
text label
- QLineEdit * [passwordconf_edit](#)
personal info input (password confirmation)
- QLabel * [date_text](#)
text label
- QDateEdit * [date_picker](#)
personal info input (date picker)
- QLabel * [error_message](#)
text label
- QLabel * [gender_text](#)
text label
- QVBoxLayout * [gender_radio](#)
Layout element.
- QGroupBox * [gender_container](#)
Layout element.
- QRadioButton * [male_button](#)
Radio button.
- QRadioButton * [female_button](#)
Radio button.
- QLabel * [pppicture_text](#)
text label
- QPushButton * [add_pppicture](#)
Button to open prompt to add picture.
- QLabel * [pppicture](#)
Picture display.
- QPushButton * [cancel](#)
Cancel button.
- QPushButton * [confirm](#)
Confirm button.
- QString * [filename](#)
Selected profile picture path.
- QFile * [tempfile](#)
Loaded profile picture.
- QGridLayout * [register_layout](#)
Main layout.

5.8.1 Member Function Documentation

5.8.1.1 void registerWidget::cancelRegistration () [slot]

[registerWidget::cancelRegistration](#)

Cancel user registration and go back to login screen

5.8.1.2 void registerWidget::checkConditions () [slot]

[registerWidget::checkConditions](#)

Check if all entries are filled and respect the criterias

5.8.1.3 void registerWidget::confirmRegistration () [slot]

[registerWidget::confirmRegistration](#)

Confirm user registration and go back to login menu

5.8.1.4 bool registerWidget::passCheck () [slot]

[registerWidget::passCheck](#)

Returns

Returns whether the password follows the criterias and matches the confirmation password

5.8.1.5 void registerWidget::selectPicture () [slot]

[registerWidget::selectPicture](#)

Open prompt to let the user select a profile picture

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

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

Chapter 6

File Documentation

6.1 birthdaywidget.cpp File Reference

Contains birthdaywidget class definition.

```
#include "birthdaywidget.h"
```

6.1.1 Detailed Description

Contains birthdaywidget class definition. This QWidget is shown whenever the user logs-in on his birthdate.

6.2 birthdaywidget.h File Reference

The birthdaywidget class.

```
#include <QWidget>
#include <QtWidgets>
```

Classes

- class [birthdaywidget](#)

6.2.1 Detailed Description

The birthdaywidget class.

6.3 gamescene_1.h File Reference

The [gameScene_1](#) class.

```
#include "gamescene_1_player.h"
#include "gamescene_1_dice.h"
#include "gamescene_1_ladderSnake.h"
#include <QGraphicsScene>
#include <QWidget>
#include <QtWidgets>
#include <QSound>
#include <QFuture>
#include <QThread>
#include <QtConcurrent/QtConcurrent>
#include <stdlib.h>
#include <time.h>
```

Classes

- class [gameScene_1](#)

6.3.1 Detailed Description

The [gameScene_1](#) class.

6.4 gamescene_1_dice.h File Reference

The [gameScene_1_dice](#) class.

```
#include <QObject>
#include <QGraphicsPixmapItem>
#include <QGraphicsScene>
#include <QTimer>
#include <stdlib.h>
#include <time.h>
```

Classes

- class [gameScene_1_dice](#)

6.4.1 Detailed Description

The [gameScene_1_dice](#) class.

6.5 menuwidget.cpp File Reference

Contains [menuWidget](#) class definition.

```
#include "menuwidget.h"
#include "loginwidget.h"
#include "birthdaywidget.h"
#include "gamescene_1.h"
```

6.5.1 Detailed Description

Contains [menuWidget](#) class definition. Main menu of the application. The user can see his history, profile and access Game 1 & Game 2 (New game / Load Game)

6.6 menuwidget.h File Reference

The [menuWidget](#) class.

```
#include <QWidget>
#include <QtWidgets>
```

Classes

- class [menuWidget](#)

6.6.1 Detailed Description

The [menuWidget](#) class.

6.7 registerwidget.cpp File Reference

Contains [registerWidget](#) class definition.

```
#include "registerwidget.h"
#include "loginwidget.h"
#include <QSpacerItem>
```

6.7.1 Detailed Description

Contains [registerWidget](#) class definition. This page allows the user to register a new account, for use within the application