EECE 435 Project

1.0.0

Generated by Doxygen 1.8.6

Tue Jan 16 2018 03:13:27

Contents

Chapter 1

EECE 435L Project

Qt application that includes three interactive games. This project has been done for a course at AUB: EECE 435L.

Libraries used:

`OAuth 2.0 for Qt` included in /OAuth

Setup:

Fonts:

Install `these` fonts on your system

Ressources:

Add `these` resources to your build folder.

Documentation:

Code Documentation:

Game Report:

Check `this` pdf file for detailed explanation.

Game Presentation:

Check `this` for the project presentation.

Screenshots:

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

Chapter 4

Class Index

4.1 Class List

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

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

Chapter 6

Namespace Documentation

6.1 QKeychain Namespace Reference

6.2 Ui Namespace Reference

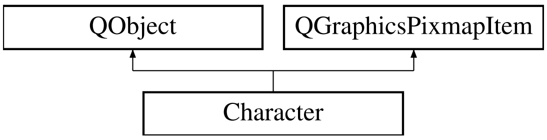
Chapter 7

Class Documentation

7.1 Character Class Reference

```
#include <character.h>
```

Inheritance diagram for Character:



Public Member Functions

- Character (QObject ∗parent=0)
- void setDifficulty (int diff)

  Character::setDifficulty.
- void keyPressEvent (QKeyEvent ∗event)

  Character::keyPressEvent.
- QStringList ∗ getValues ()
- QStringList ∗ getVices ()
- void checkCollisions ()

Public Attributes

- bool started

7.1.1 Constructor & Destructor Documentation

7.1.1.1 Character::Character (QObject ∗ *parent* = `0`) `[explicit]`

7.1.2 Member Function Documentation

7.1.2.1 void Character::checkCollisions ( )

item is an item from the colliding items list groupTemp is 0 if the item is not a QGraphicsTextItem pointer

7.1.2.2 QStringList ∗ Character::getValues ( )

7.1.2.3 QStringList ∗ Character::getVices ( )

7.1.2.4 void Character::keyPressEvent (QKeyEvent ∗ *event*)

Character::keyPressEvent.

Parameters

event	

Move up, down, right or left according to key press even.

Move up or down: only by increments of 65 (60 the first time).

Move right or left: continuously without exceeding bounds.

7.1.2.5 void Character::setDifficulty (int *diff*)

Character::setDifficulty.

Parameters

diff	

7.1.3 Member Data Documentation

7.1.3.1 bool Character::started

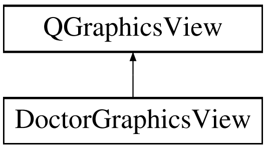
The documentation for this class was generated from the following files:

- Game1/character.h
- Game1/character.cpp

7.2 DoctorGraphicsView Class Reference

```
#include <doctorgraphicsview.h>
```

Inheritance diagram for DoctorGraphicsView:



Public Member Functions

- DoctorGraphicsView (QWidget ∗parent=0)
- void setStackedWidget (QStackedWidget ∗stack)

7.2.1 Constructor & Destructor Documentation

7.2.1.1 DoctorGraphicsView::DoctorGraphicsView (QWidget ∗ *parent* = 0) [explicit]

7.2.2 Member Function Documentation

7.2.2.1 void DoctorGraphicsView::setStackedWidget (QStackedWidget ∗ *stack*)

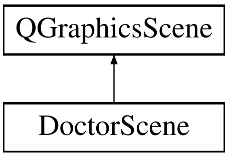
The documentation for this class was generated from the following files:

- Game2/Doctor/doctorgraphicsview.h
- Game2/Doctor/doctorgraphicsview.cpp

7.3 DoctorScene Class Reference

```
#include <doctorscene.h>
```

Inheritance diagram for DoctorScene:



Public Slots

- void updateTime ()

  DoctorScene::updateTime.

Public Member Functions

- DoctorScene (QObject ∗parent=0)
- void setStackedWidget (QStackedWidget ∗stack)

  DoctorScene::setStackedWidget.
- void mousePressEvent (QGraphicsSceneMouseEvent ∗event)

  DoctorScene::mousePressEvent.

7.3.1 Constructor & Destructor Documentation

7.3.1.1 DoctorScene::DoctorScene (QObject ∗ *parent* = 0) [explicit]

7.3.2 Member Function Documentation

7.3.2.1 void DoctorScene::mousePressEvent (QGraphicsSceneMouseEvent ∗ *event*)

DoctorScene::mousePressEvent.

Parameters

event	

If player clicks on arrow, this function takes him to next scenario.

After the last scenario, takes him to score ui.

7.3.2.2 void DoctorScene::setStackedWidget (QStackedWidget ∗ *stack*)

DoctorScene::setStackedWidget.

Parameters

stack	

7.3.2.3 void DoctorScene::updateTime ( ) [slot]

DoctorScene::updateTime.

Shows the time of day, updated every second.

Depending on value of stateOfEngineer, which is global, sets position of character on the map.

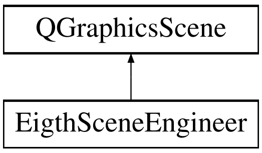
The documentation for this class was generated from the following files:

- Game2/Doctor/doctorscene.h
- Game2/Doctor/doctorscene.cpp

7.4 EigthSceneEngineer Class Reference

```
#include <eigthsceneengineer.h>
```

Inheritance diagram for EigthSceneEngineer:



Public Member Functions

- EigthSceneEngineer (QObject ∗parent=0)
- void keyPressEvent (QKeyEvent ∗event)

  EigthSceneEngineer::keyPressEvent.
- void mousePressEvent (QGraphicsSceneMouseEvent ∗event)

  EigthSceneEngineer::mousePressEvent.
- void changeScene ()

  EigthSceneEngineer::changeScene.
- void showResult ()

  EigthSceneEngineer::showResult.

7.4.1 Constructor & Destructor Documentation

7.4.1.1 EigthSceneEngineer::EigthSceneEngineer (QObject ∗ *parent* = 0) [explicit]

7.4.2 Member Function Documentation

7.4.2.1 void EigthSceneEngineer::changeScene ( )

EigthSceneEngineer::changeScene.

Gets the view that is showing the current scene.

Sets back the scene to the parent of the current scene, which is the main map.

7.4.2.2 void EigthSceneEngineer::keyPressEvent (QKeyEvent * *event*)

EigthSceneEngineer::keyPressEvent.

Parameters

event	

Changes the text of the upper box

enterState sets result of pressing enter key.

If it is 0, pressing enter shows the options.

After showing the options, enterState is set to 1.

If enterState is 1, pressing enter key has no effect.

If enterState is 2, this means the scene is showing the result of the player's choice, and pressing enter switches back to main map.

In this case we also have to stop the sound of the rocket

7.4.2.3 void EigthSceneEngineer::mousePressEvent (QGraphicsSceneMouseEvent ∗ *event*)

EigthSceneEngineer::mousePressEvent.

Parameters

event	

If player clicks on one option, values are updated and results of option choosed is shown.

7.4.2.4 void EigthSceneEngineer::showResult ( )

EigthSceneEngineer::showResult.

Removes unwanted items.

Shows result depending on value of response.

enterState is set to 2 so that enter key is disabled after this

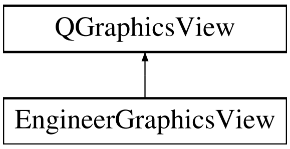
The documentation for this class was generated from the following files:

- Game2/Engineer/eigthsceneengineer.h
- Game2/Engineer/eigthsceneengineer.cpp

7.5 EngineerGraphicsView Class Reference

```
#include <engineergraphicsview.h>
```

Inheritance diagram for EngineerGraphicsView:



Public Member Functions

- EngineerGraphicsView (QWidget ∗parent=0)
- void setStackedWidget (QStackedWidget ∗stack)

7.5.1 Constructor & Destructor Documentation

7.5.1.1 EngineerGraphicsView::EngineerGraphicsView (QWidget ∗ *parent* = 0) [explicit]

7.5.2 Member Function Documentation

7.5.2.1 void EngineerGraphicsView::setStackedWidget (QStackedWidget ∗ *stack*)

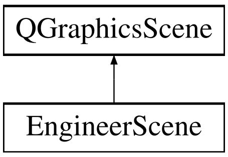
The documentation for this class was generated from the following files:

- Game2/Engineer/engineergraphicsview.h
- Game2/Engineer/engineergraphicsview.cpp

7.6 EngineerScene Class Reference

`#include <engineerscene.h>`

Inheritance diagram for EngineerScene:



Public Slots

- void updateTime ()

  EngineerScene::updateTime.

Public Member Functions

- EngineerScene (QObject ∗parent=0)
- void setStackedWidget (QStackedWidget ∗stack)

  EngineerScene::setStackedWidget.
- void mousePressEvent (QGraphicsSceneMouseEvent ∗event)

  EngineerScene::mousePressEvent.

7.6.1 Constructor & Destructor Documentation

7.6.1.1 EngineerScene::EngineerScene (QObject ∗ *parent* = 0) [explicit]

7.6.2 Member Function Documentation

7.6.2.1 void EngineerScene::mousePressEvent (QGraphicsSceneMouseEvent ∗ *event*)

EngineerScene::mousePressEvent.

Parameters

event	

If player clicks on arrow, this function takes him to next scenario.

After the last scenario, takes him to score ui.

7.6.2.2 void EngineerScene::setStackedWidget (QStackedWidget ∗ *stack*)

EngineerScene::setStackedWidget.

Parameters

stack	

7.6.2.3 void EngineerScene::updateTime ( ) `[slot]`

EngineerScene::updateTime.

Shows the time of day, updated every second.

Depending on value of stateOfEngineer, which is global, sets position of character on the map.

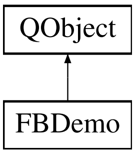
The documentation for this class was generated from the following files:

- Game2/Engineer/engineerscene.h
- Game2/Engineer/engineerscene.cpp

7.7 FBDemo Class Reference

`#include <fbdemo.h>`

Inheritance diagram for FBDemo:



Public Slots

- void doOAuth (O2::GrantFlow grantFlowType)
- void validateToken ()

Signals

- void extraTokensReady (const QVariantMap &extraTokens)
- void linkingFailed ()
- void linkingSucceeded ()
- void replied ()

Public Member Functions

- FBDemo (QObject ∗parent=0)
- void getInformation ()
- QJsonDocument getResponse ()

7.7.1 Constructor & Destructor Documentation

7.7.1.1 FBDemo::FBDemo (QObject ∗ *parent* = 0) [explicit]

7.7.2 Member Function Documentation

7.7.2.1 void FBDemo::doOAuth (O2::GrantFlow *grantFlowType*) [slot]

7.7.2.2 void FBDemo::extraTokensReady (const QVariantMap & *extraTokens*) [signal]

7.7.2.3 void FBDemo::getInformation ( )

7.7.2.4 QJsonDocument FBDemo::getResponse ( )

7.7.2.5 void FBDemo::linkingFailed ( ) [signal]

7.7.2.6 void FBDemo::linkingSucceeded ( ) [signal]

7.7.2.7 void FBDemo::replied ( ) [signal]

7.7.2.8 void FBDemo::validateToken ( ) [slot]

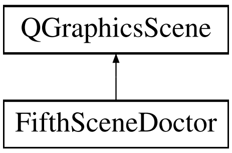
The documentation for this class was generated from the following files:

- OAuth/fbdemo.h
- OAuth/fbdemo.cpp

7.8 FifthSceneDoctor Class Reference

```
#include <fifthscenedoctor.h>
```

Inheritance diagram for FifthSceneDoctor:



Public Slots

- void updateScene ()

Public Member Functions

- FifthSceneDoctor (QObject ∗parent=0)
- void keyPressEvent (QKeyEvent ∗event)

- void mousePressEvent (QGraphicsSceneMouseEvent ∗event)
- void mouseMoveEvent (QGraphicsSceneMouseEvent ∗event)
- void changeScene ()
- void showResult ()

7.8.1 Constructor & Destructor Documentation

7.8.1.1 FifthSceneDoctor::FifthSceneDoctor (QObject ∗ *parent* = 0) [explicit]

7.8.2 Member Function Documentation

7.8.2.1 void FifthSceneDoctor::changeScene ( )

7.8.2.2 void FifthSceneDoctor::keyPressEvent (QKeyEvent ∗ *event*)

7.8.2.3 void FifthSceneDoctor::mouseMoveEvent (QGraphicsSceneMouseEvent ∗ *event*)

7.8.2.4 void FifthSceneDoctor::mousePressEvent (QGraphicsSceneMouseEvent ∗ *event*)

7.8.2.5 void FifthSceneDoctor::showResult ( )

7.8.2.6 void FifthSceneDoctor::updateScene ( ) [slot]

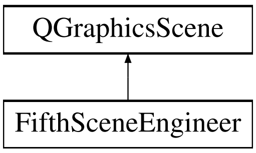
The documentation for this class was generated from the following files:

- Game2/Doctor/fifthscenedoctor.h
- Game2/Doctor/fifthscenedoctor.cpp

7.9 FifthSceneEngineer Class Reference

```
#include <fifthsceneengineer.h>
```

Inheritance diagram for FifthSceneEngineer:



Public Member Functions

- FifthSceneEngineer (QObject ∗parent=0)
- void keyPressEvent (QKeyEvent ∗event)

  FifthSceneEngineer::keyPressEvent.
- void mousePressEvent (QGraphicsSceneMouseEvent ∗event)

  FifthSceneEngineer::mousePressEvent.
- void mouseMoveEvent (QGraphicsSceneMouseEvent ∗event)

  FifthSceneEngineer::mouseMoveEvent.
- void changeScene ()

  FifthSceneEngineer::changeScene.

7.9.1 Constructor & Destructor Documentation

7.9.1.1 FifthSceneEngineer::FifthSceneEngineer (QObject ∗ *parent* = 0) [explicit]

7.9.2 Member Function Documentation

7.9.2.1 void FifthSceneEngineer::changeScene ( )

FifthSceneEngineer::changeScene.

Gets the view that is showing the current scene.

Sets back the scene to the parent of the current scene, which is the main map.

7.9.2.2 void FifthSceneEngineer::keyPressEvent (QKeyEvent ∗ *event*)

FifthSceneEngineer::keyPressEvent.

Parameters

event	

Changes the text of the upper box

enterState sets result of pressing enter key.

If it is 0, pressing enter shows the options.

After showing the options, enterState is set to 1.

If enterState is 1, pressing enter key has no effect.

clickState is set to 1 to allow player to click on an option.

7.9.2.3 void FifthSceneEngineer::mouseMoveEvent (QGraphicsSceneMouseEvent ∗ *event*)

FifthSceneEngineer::mouseMoveEvent.

Parameters

event	

If clickState is not yet set to 1, (options are not shown), moving mouse has no effect on images.

When clickState is 1, moving mouse over friend or colleagues changes their pixmap to make them glow.

7.9.2.4 void FifthSceneEngineer::mousePressEvent (QGraphicsSceneMouseEvent ∗ *event*)

FifthSceneEngineer::mousePressEvent.

Parameters

event	

Before clickState is set to 1, clicking on screen has no effect.

If player clicks on one option, values are updated and scene is switched back to the main map.

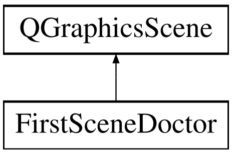
The documentation for this class was generated from the following files:

- Game2/Engineer/fifthsceneengineer.h
- Game2/Engineer/fifthsceneengineer.cpp

7.10 FirstSceneDoctor Class Reference

`#include <firstscenedoctor.h>`

Inheritance diagram for FirstSceneDoctor:



Public Member Functions

- FirstSceneDoctor (QObject ∗parent=0)
- void keyPressEvent (QKeyEvent ∗event)

  FirstSceneDoctor::keyPressEvent.
- void mousePressEvent (QGraphicsSceneMouseEvent ∗event)

  FirstSceneDoctor::mousePressEvent.
- void changeScene ()

  FirstSceneDoctor::changeScene.
- void addAliens ()

  FirstSceneDoctor::addAliens.

7.10.1 Constructor & Destructor Documentation

7.10.1.1 FirstSceneDoctor::FirstSceneDoctor (QObject ∗ *parent* = 0) [explicit]

7.10.2 Member Function Documentation

7.10.2.1 void FirstSceneDoctor::addAliens ( )

FirstSceneDoctor::addAliens.

Creates 3 aliens randomly at specific positions.

7.10.2.2 void FirstSceneDoctor::changeScene ( )

FirstSceneDoctor::changeScene.

Gets the view that is showing the current scene.

Sets back the scene to the parent of the current scene, which is the main map.

7.10.2.3 void FirstSceneDoctor::keyPressEvent (QKeyEvent ∗ *event*)

FirstSceneDoctor::keyPressEvent.

Parameters

event	

Changes the text of the upper box

enterState sets result of pressing enter key.

If it is 0, pressing enter shows the options.

After showing the options, enterState is set to 1.

If enterState is 1, pressing enter key has no effect.

7.10.2.4 void FirstSceneDoctor::mousePressEvent (QGraphicsSceneMouseEvent ∗ *event*)

FirstSceneDoctor::mousePressEvent.

Parameters

event	

If player clicks on one option, values are updated and scene is switched back to the main map.

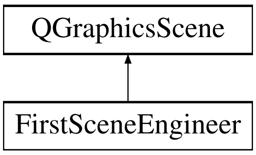
The documentation for this class was generated from the following files:

- Game2/Doctor/firstscenedoctor.h
- Game2/Doctor/firstscenedoctor.cpp

7.11 FirstSceneEngineer Class Reference

`#include <firstsceneengineer.h>`

Inheritance diagram for FirstSceneEngineer:



Public Member Functions

- FirstSceneEngineer (QObject ∗parent=0)
- void addAliens ()

  FirstSceneEngineer::addAliens.
- void keyPressEvent (QKeyEvent ∗event)

  FirstSceneEngineer::keyPressEvent.
- void mousePressEvent (QGraphicsSceneMouseEvent ∗event)

  FirstSceneEngineer::mousePressEvent.
- void changeScene ()

  FirstSceneEngineer::changeScene.

7.11.1 Constructor & Destructor Documentation

7.11.1.1 FirstSceneEngineer::FirstSceneEngineer (QObject ∗ *parent* = 0) [explicit]

7.11.2 Member Function Documentation

7.11.2.1 void FirstSceneEngineer::addAliens ( )

FirstSceneEngineer::addAliens.

Creates 3 aliens randomly at specific positions.

7.11.2.2 void FirstSceneEngineer::changeScene ( )

FirstSceneEngineer::changeScene.

Gets the view that is showing the current scene.

Sets back the scene to the parent of the current scene, which is the main map.

7.11.2.3 void FirstSceneEngineer::keyPressEvent (QKeyEvent ∗ *event*)

FirstSceneEngineer::keyPressEvent.

Parameters

event	

Changes the text of the upper box

enterState sets result of pressing enter key.

If it is 0, pressing enter shows the options.

After showing the options, enterState is set to 1.

If enterState is 1, pressing enter key has no effect.

7.11.2.4 void FirstSceneEngineer::mousePressEvent (QGraphicsSceneMouseEvent ∗ *event*)

FirstSceneEngineer::mousePressEvent.

Parameters

event	

If player clicks on one option, values are updated and scene is switched back to the main map.

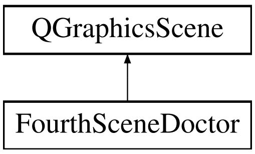
The documentation for this class was generated from the following files:

- Game2/Engineer/firstsceneengineer.h
- Game2/Engineer/firstsceneengineer.cpp

7.12 FourthSceneDoctor Class Reference

```
#include <fourthscenedoctor.h>
```

Inheritance diagram for FourthSceneDoctor:



Public Slots

- void updateScene ()

Public Member Functions

- FourthSceneDoctor (QObject ∗parent=0)
- void keyPressEvent (QKeyEvent ∗event)
- void mousePressEvent (QGraphicsSceneMouseEvent ∗event)
- void changeScene ()
- void showResult ()

7.12.1 Constructor & Destructor Documentation

7.12.1.1 FourthSceneDoctor::FourthSceneDoctor (QObject ∗ *parent* = 0) [explicit]

7.12.2 Member Function Documentation

7.12.2.1 void FourthSceneDoctor::changeScene ( )

7.12.2.2 void FourthSceneDoctor::keyPressEvent (QKeyEvent ∗ *event*)

7.12.2.3 void FourthSceneDoctor::mousePressEvent (QGraphicsSceneMouseEvent ∗ *event*)

7.12.2.4 void FourthSceneDoctor::showResult ( )

7.12.2.5 void FourthSceneDoctor::updateScene ( ) [slot]

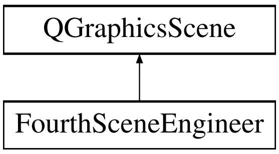
The documentation for this class was generated from the following files:

- Game2/Doctor/fourthscenedoctor.h
- Game2/Doctor/fourthscenedoctor.cpp

7.13 FourthSceneEngineer Class Reference

```
#include <fourthsceneengineer.h>
```

Inheritance diagram for FourthSceneEngineer:



Public Member Functions

- FourthSceneEngineer (QObject ∗parent=0)
- void keyPressEvent (QKeyEvent ∗event)

  FourthSceneEngineer::keyPressEvent.
- void mousePressEvent (QGraphicsSceneMouseEvent ∗event)

  FourthSceneEngineer::mousePressEvent.

- void changeScene ()

  FourthSceneEngineer::changeScene.
- void showResult ()

  FourthSceneEngineer::showResult.

7.13.1 Constructor & Destructor Documentation

7.13.1.1 FourthSceneEngineer::FourthSceneEngineer (QObject ∗ *parent* = 0) [explicit]

7.13.2 Member Function Documentation

7.13.2.1 void FourthSceneEngineer::changeScene ( )

FourthSceneEngineer::changeScene.

Gets the view that is showing the current scene.

Sets back the scene to the parent of the current scene, which is the main map.

7.13.2.2 void FourthSceneEngineer::keyPressEvent (QKeyEvent ∗ *event*)

FourthSceneEngineer::keyPressEvent.

Parameters

event	

Changes the text of the upper box

enterState sets result of pressing enter key.

If it is 0, pressing enter shows the options.

After showing the options, enterState is set to 1.

If enterState is 1, pressing enter key has no effect.

If enterState is 2, this means the scene is showing the result of the player's choice, and pressing enter switches back to main map.

7.13.2.3 void FourthSceneEngineer::mousePressEvent (QGraphicsSceneMouseEvent ∗ *event*)

FourthSceneEngineer::mousePressEvent.

Parameters

event	

If player clicks on one option, values are updated and results of option choosed is shown.

7.13.2.4 void FourthSceneEngineer::showResult ( )

FourthSceneEngineer::showResult.

Removes unwanted items.

Shows result depending on value of response.

enterState is set to 2 so that enter key is disabled after this

Updates moneyGlobal, which is global,depending on response.

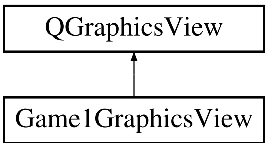
The documentation for this class was generated from the following files:

- Game2/Engineer/fourthsceneengineer.h
- Game2/Engineer/fourthsceneengineer.cpp

7.14 Game1GraphicsView Class Reference

```
#include <game1graphicsview.h>
```

Inheritance diagram for Game1GraphicsView:



Public Member Functions

- Game1GraphicsView (QWidget ∗parent=0)
- void setDifficulty (int diff)
- void setStackedWidget (QStackedWidget ∗stack)
- void resume ()
- Game1Scene ∗ getScene ()

7.14.1 Constructor & Destructor Documentation

7.14.1.1 Game1GraphicsView::Game1GraphicsView (QWidget ∗ *parent* = 0) [explicit]

7.14.2 Member Function Documentation

7.14.2.1 Game1Scene ∗ Game1GraphicsView::getScene ( )

7.14.2.2 void Game1GraphicsView::resume ( )

7.14.2.3 void Game1GraphicsView::setDifficulty (int *diff*)

7.14.2.4 void Game1GraphicsView::setStackedWidget (QStackedWidget ∗ *stack*)

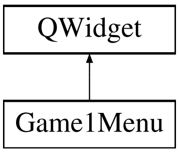
The documentation for this class was generated from the following files:

- Game1/game1graphicsview.h
- Game1/game1graphicsview.cpp

7.15 Game1Menu Class Reference

```
#include <game1menu.h>
```

Inheritance diagram for Game1Menu:



Signals

- void resumeOrder ()

Public Member Functions

- Game1Menu (QWidget ∗parent=0)
- ∼Game1Menu ()
- void setStackedWidget (QStackedWidget ∗stack)

7.15.1 Constructor & Destructor Documentation

7.15.1.1 Game1Menu::Game1Menu (QWidget ∗ *parent* = 0) [explicit]

7.15.1.2 Game1Menu::∼Game1Menu ( )

7.15.2 Member Function Documentation

7.15.2.1 void Game1Menu::resumeOrder ( ) [signal]

7.15.2.2 void Game1Menu::setStackedWidget (QStackedWidget ∗ *stack*)

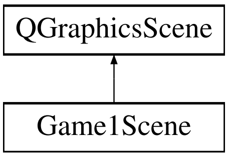
The documentation for this class was generated from the following files:

- Game1/game1menu.h
- Game1/game1menu.cpp

7.16 Game1Scene Class Reference

```
#include <game1scene.h>
```

Inheritance diagram for Game1Scene:



Public Slots

- void newObstacle ()

  Game1Scene::newObstacle.
- void updateTimer ()

  Game1Scene::updateTimer.
- void updateLives ()

  Game1Scene::updateLives.
- void start ()
- void resume ()

Public Member Functions

- Game1Scene (QObject ∗parent=0)
- void setDifficulty (int diff)
- void addAcquired (QString element)
- void endGame ()
- void updateAcquired ()
- void setStackedWidget (QStackedWidget ∗stack)
- void mousePressEvent (QGraphicsSceneMouseEvent ∗event)

7.16.1 Constructor & Destructor Documentation

7.16.1.1 Game1Scene::Game1Scene (QObject ∗ *parent* = 0) [explicit]

7.16.2 Member Function Documentation

7.16.2.1 void Game1Scene::addAcquired (QString *element*)

7.16.2.2 void Game1Scene::endGame ( )

7.16.2.3 void Game1Scene::mousePressEvent (QGraphicsSceneMouseEvent ∗ *event*)

7.16.2.4 void Game1Scene::newObstacle ( ) [slot]

Game1Scene::newObstacle.

Setting position of obstacle randomly.

id = 0 or 3 –> flying saucer, can be added left or right.

id = 1 –> space shuttle oriented to left, can be added only starting at right position.

id = 2 –> space shuttle oriented to right, can be added only starting at left position.

7.16.2.5 void Game1Scene::resume ( ) [slot]

7.16.2.6 void Game1Scene::setDifficulty (int *diff*)

7.16.2.7 void Game1Scene::setStackedWidget (QStackedWidget ∗ *stack*)

7.16.2.8 void Game1Scene::start ( ) [slot]

7.16.2.9 void Game1Scene::updateAcquired ( )

7.16.2.10 void Game1Scene::updateLives ( ) [slot]

Game1Scene::updateLives.

Updating the live items on scene.

lives is a List of live images.

When function is called, remove exisiting images and pritn new ones equal to number of current lives.

7.16.2.11 void Game1Scene::updateTimer ( ) [slot]

Game1Scene::updateTimer.

countTime counts the number of times timer has expired (initalized to 60).

When countTime reaches 0 or when character reaches bottom of page –> end game.

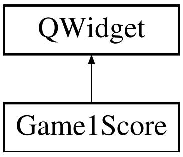
The documentation for this class was generated from the following files:

- Game1/game1scene.h
- Game1/game1scene.cpp

7.17 Game1Score Class Reference

```
#include <game1score.h>
```

Inheritance diagram for Game1Score:



Public Member Functions

- Game1Score (QWidget ∗parent=0)
- void setScore (int lives, int timeLeft, QStringList ∗finalValues, QStringList ∗finalVices)
- void setStackedWidget (QStackedWidget ∗stack)
- ∼Game1Score ()

7.17.1 Constructor & Destructor Documentation

7.17.1.1 Game1Score::Game1Score (QWidget ∗ *parent* = 0) [explicit]

7.17.1.2 Game1Score::∼Game1Score ( )

7.17.2 Member Function Documentation

7.17.2.1 void Game1Score::setScore (int *lives,* int *timeLeft,* QStringList ∗ *finalValues,* QStringList ∗ *finalVices*)

7.17.2.2 void Game1Score::setStackedWidget (QStackedWidget ∗ *stack*)

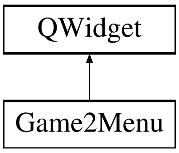
The documentation for this class was generated from the following files:

- Game1/game1score.h
- Game1/game1score.cpp

7.18 Game2Menu Class Reference

```
#include <game2menu.h>
```

Inheritance diagram for Game2Menu:



Public Member Functions

- Game2Menu (QWidget ∗parent=0)
- void setStackedWidget (QStackedWidget ∗stack)
- ∼Game2Menu ()

7.18.1 Constructor & Destructor Documentation

7.18.1.1 Game2Menu::Game2Menu (QWidget ∗ *parent* = 0) [explicit]

7.18.1.2 Game2Menu::∼Game2Menu ( )

7.18.2 Member Function Documentation

7.18.2.1 void Game2Menu::setStackedWidget (QStackedWidget ∗ *stack*)

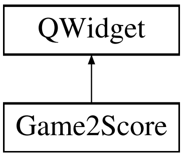
The documentation for this class was generated from the following files:

- Game2/game2menu.h
- Game2/game2menu.cpp

7.19 Game2Score Class Reference

```
#include <game2score.h>
```

Inheritance diagram for Game2Score:



Public Member Functions

- Game2Score (QWidget ∗parent=0)
- void setStackedWidget (QStackedWidget ∗stack)
- ∼Game2Score ()

7.19.1 Constructor & Destructor Documentation

7.19.1.1 Game2Score::Game2Score (QWidget ∗ *parent* = 0) [explicit]

7.19.1.2 Game2Score::∼Game2Score ( )

7.19.2 Member Function Documentation

7.19.2.1 void Game2Score::setStackedWidget (QStackedWidget ∗ *stack*)

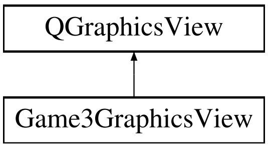
The documentation for this class was generated from the following files:

- Game2/game2score.h
- Game2/game2score.cpp

7.20 Game3GraphicsView Class Reference

```
#include <game3graphicsview.h>
```

Inheritance diagram for Game3GraphicsView:



Public Member Functions

- Game3GraphicsView (QWidget ∗parent=0)
- void setDifficulty (int diff)

  Game3GraphicsView::setDifficulty.
- void setStackedWidget (QStackedWidget ∗stack, int menuIndex)

7.20.1 Constructor & Destructor Documentation

7.20.1.1 Game3GraphicsView::Game3GraphicsView (QWidget ∗ *parent* = 0) [explicit]

7.20.2 Member Function Documentation

7.20.2.1 void Game3GraphicsView::setDifficulty (int *diff*)

Game3GraphicsView::setDifficulty.

Parameters

diff	

Passing difficulty to scene

Showing the scene

7.20.2.2 void Game3GraphicsView::setStackedWidget (QStackedWidget ∗ *stack,* int *menuIndex*)

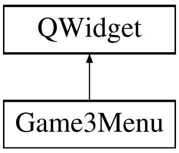
The documentation for this class was generated from the following files:

- Game3/game3graphicsview.h
- Game3/game3graphicsview.cpp

7.21 Game3Menu Class Reference

```
#include <game3menu.h>
```

Inheritance diagram for Game3Menu:



Public Member Functions

- Game3Menu (QWidget ∗parent=0)
- ∼Game3Menu ()
- void setStackedWidget (QStackedWidget ∗stack)

7.21.1 Constructor & Destructor Documentation

7.21.1.1 Game3Menu::Game3Menu (QWidget ∗ *parent* = 0) [explicit]

7.21.1.2 Game3Menu::∼Game3Menu ( )

7.21.2 Member Function Documentation

7.21.2.1 void Game3Menu::setStackedWidget (QStackedWidget ∗ *stack*)

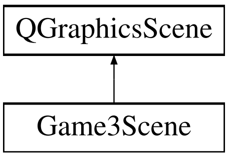
The documentation for this class was generated from the following files:

- Game3/game3menu.h
- Game3/game3menu.cpp

7.22 Game3Scene Class Reference

```
#include <game3scene.h>
```

Inheritance diagram for Game3Scene:



Public Slots

- void match ()

  Game3Scene::match.
- void notMatch ()

  Game3Scene::notMatch.
- void endGame ()

  Game3Scene::endGame.

Public Member Functions

- Game3Scene (QObject ∗parent=0)
- void setDifficulty (int diff)

  Game3Scene::setDifficulty.
- void setStackedWidget (QStackedWidget ∗stack, int menuIndex)
- void placeCards ()

  Game3Scene::placeCards.
- void setValues ()

Game3Scene::setValues.

- void keyPressEvent (QKeyEvent ∗event)

  Game3Scene::keyPressEvent.
- void mousePressEvent (QGraphicsSceneMouseEvent ∗event)

7.22.1 Constructor & Destructor Documentation

7.22.1.1 Game3Scene::Game3Scene (QObject ∗ *parent* = 0) [explicit]

7.22.2 Member Function Documentation

7.22.2.1 void Game3Scene::endGame ( ) [slot]

Game3Scene::endGame.

Called when there are no more values left or no lives left and display.

7.22.2.2 void Game3Scene::keyPressEvent (QKeyEvent ∗ *event*)

Game3Scene::keyPressEvent.

Parameters

event	

Detects keyPressEvents and act accordingly.

If the user pressed enter, it flips the card if it is open and if it is the second card opened, check if it is a match or not.

Else, the keyPressEvent is passed to the character.

7.22.2.3 void Game3Scene::match ( ) [slot]

Game3Scene::match.

Called when two cards match and checks wether it is a value or a vice.

Checks if card at the position entered is contained in list of vices or values.

If it is a value, we decrement the count.

If all values have been found, end the game and flip all remaining cards. If it is a vice, we remove a life.

If there are no lives left, end game and set endMessage accordingly.

For both cases we remove the corresponding cards and set their state to REMOVED.

7.22.2.4 void Game3Scene::mousePressEvent (QGraphicsSceneMouseEvent ∗ *event*)

7.22.2.5 void Game3Scene::notMatch ( ) [slot]

Game3Scene::notMatch.

Called when two cards do not match.

Flips cards back.

Resets the position of currently flipped cards to nothing (-1)

7.22.2.6 void Game3Scene::placeCards ( )

Game3Scene::placeCards.

Displays all the cards on the screen grid of 3 rows and 6 columns.

7.22.2.7 void Game3Scene::setDifficulty (int *diff*)

Game3Scene::setDifficulty.

Parameters

diff	

Takes the difficulty from the graphics view and the initializes the scene accordingly by showing the apropriate character and number of lives.

7.22.2.8 void Game3Scene::setStackedWidget (QStackedWidget ∗ *stack,* int *menuIndex*)

7.22.2.9 void Game3Scene::setValues ( )

Game3Scene::setValues.

Adds values and vices two by two to the values list.

The number of values and vices added depends on the difficulty.

Each value/vice is mapped to the corresponding card with the same index.

maxi, Number of values to be added.

maxj, Number of vices to be added.

done, checks if we added a value or a vice.

format, used to center the text.

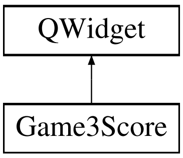
The documentation for this class was generated from the following files:

- Game3/game3scene.h
- Game3/game3scene.cpp

7.23 Game3Score Class Reference

```
#include <game3score.h>
```

Inheritance diagram for Game3Score:



Public Member Functions

- Game3Score (QWidget ∗parent=0)
- ∼Game3Score ()
- void setStackedWidget (QStackedWidget ∗stack, int menuIndex)
- void setScore (int livesCount, QStringList ∗values, QStringList ∗vices)

  Game3Score::setScore.

7.23.1 Constructor & Destructor Documentation

7.23.1.1 Game3Score::Game3Score (QWidget ∗ *parent* = 0) [explicit]

7.23.1.2 Game3Score::∼Game3Score ( )

7.23.2 Member Function Documentation

7.23.2.1 void Game3Score::setScore (int *lives,* QStringList ∗ *finalValues,* QStringList ∗ *finalVices*)

Game3Score::setScore.

Parameters

lives	Number of lives left
finalValues	QStringList of values gained
finalVices	QStrindList of vices gained

Adding the values to the score page

Adding the vices to the score page

Adding the lives to the score page

7.23.2.2 void Game3Score::setStackedWidget (QStackedWidget ∗ *stack,* int *menuIndex*)

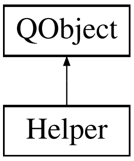
The documentation for this class was generated from the following files:

- Game3/game3score.h
- Game3/game3score.cpp

7.24 Helper Class Reference

```
#include <helper.h>
```

Inheritance diagram for Helper:



Public Slots

- void processArgs ()
- void onLinkingFailed ()
- void onLinkingSucceeded ()
- void validate ()

Signals

- void success ()
- void fail ()

Public Member Functions

- Helper ()
- QJsonDocument getResponse ()

  Helper::getResponse Get JsonDocument of fbdemo.

7.24.1 Constructor & Destructor Documentation

7.24.1.1 Helper::Helper ( ) [inline]

7.24.2 Member Function Documentation

7.24.2.1 void Helper::fail ( ) [signal]

7.24.2.2 QJsonDocument Helper::getResponse ( )

Helper::getResponse Get JsonDocument of fbdemo.

Returns

7.24.2.3 void Helper::onLinkingFailed ( ) [slot]

7.24.2.4 void Helper::onLinkingSucceeded ( ) [slot]

7.24.2.5 void Helper::processArgs ( ) [slot]

7.24.2.6 void Helper::success ( ) [signal]

7.24.2.7 void Helper::validate ( ) [slot]

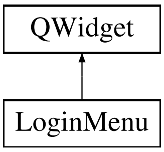
The documentation for this class was generated from the following files:

- OAuth/helper.h
- OAuth/helper.cpp

7.25 LoginMenu Class Reference

```
#include <loginmenu.h>
```

Inheritance diagram for LoginMenu:



Public Member Functions

- LoginMenu (QWidget ∗parent=0)
- ∼LoginMenu ()
- void setStackedWidget (QStackedWidget ∗stack)

7.25.1 Constructor & Destructor Documentation

7.25.1.1 LoginMenu::LoginMenu (QWidget ∗ *parent* = 0) [explicit]

7.25.1.2 LoginMenu::∼LoginMenu ( )

7.25.2 Member Function Documentation

7.25.2.1 void LoginMenu::setStackedWidget (QStackedWidget ∗ *stack*)

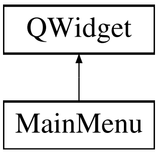
The documentation for this class was generated from the following files:

- loginmenu.h
- loginmenu.cpp

7.26 MainMenu Class Reference

```
#include <mainmenu.h>
```

Inheritance diagram for MainMenu:



Public Member Functions

- MainMenu (QWidget ∗parent=0)
- void setStackedWidget (QStackedWidget ∗stack)
- ∼MainMenu ()

7.26.1 Constructor & Destructor Documentation

7.26.1.1 MainMenu::MainMenu (QWidget ∗ *parent* = 0) [explicit]

7.26.1.2 MainMenu::∼MainMenu ( )

7.26.2 Member Function Documentation

7.26.2.1 void MainMenu::setStackedWidget (QStackedWidget ∗ *stack*)

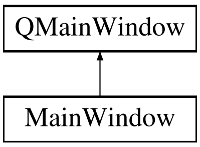
The documentation for this class was generated from the following files:

- mainmenu.h
- mainmenu.cpp

7.27 MainWindow Class Reference

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

Inheritance diagram for MainWindow:



Public Member Functions

- MainWindow (QWidget ∗parent=0)
- ∼MainWindow ()

7.27.1 Constructor & Destructor Documentation

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

7.27.1.2 MainWindow::∼MainWindow ( )

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

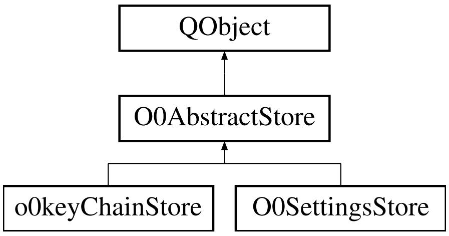
- mainwindow.h
- mainwindow.cpp

7.28 O0AbstractStore Class Reference

Storage for strings.

`#include <o0abstractstore.h>`

Inheritance diagram for O0AbstractStore:



Public Member Functions

- O0AbstractStore (QObject ∗parent=0)
- virtual QString value (const QString &key, const QString &defaultValue=QString())=0

  Retrieve a string value by key.
- virtual void setValue (const QString &key, const QString &value)=0

  Set a string value for a key.

7.28.1 Detailed Description

Storage for strings.

7.28.2 Constructor & Destructor Documentation

7.28.2.1 O0AbstractStore::O0AbstractStore (QObject * *parent* = 0) [inline], [explicit]

7.28.3 Member Function Documentation

7.28.3.1 virtual void O0AbstractStore::setValue (const QString & *key*, const QString & *value*) [pure virtual]

Set a string value for a key.

Implemented in o0keyChainStore, and O0SettingsStore.

7.28.3.2 virtual QString O0AbstractStore::value (const QString & *key*, const QString & *defaultValue* = QString()) [pure virtual]

Retrieve a string value by key.

Implemented in o0keyChainStore, and O0SettingsStore.

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

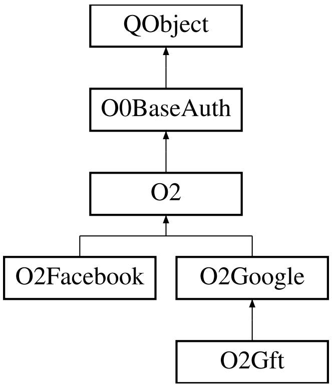
- OAuth/src/o0abstractstore.h

7.29 O0BaseAuth Class Reference

Base class of OAuth authenticators.

`#include <o0baseauth.h>`

Inheritance diagram for O0BaseAuth:



Public Slots

- virtual Q_INVOKABLE void link ()=0

  Authenticate.
- virtual Q_INVOKABLE void unlink ()=0

  De-authenticate.

Signals

- void openBrowser (const QUrl &url)

  Emitted when client needs to open a web browser window, with the given URL.

- void closeBrowser ()

  Emitted when client can close the browser window.
- void linkingSucceeded ()

  Emitted when authentication/deauthentication succeeded.
- void linkingFailed ()

  Emitted when authentication/deauthentication failed.
- void linkedChanged ()
- void clientIdChanged ()
- void clientSecretChanged ()
- void localPortChanged ()
- void tokenChanged ()
- void tokenSecretChanged ()
- void extraTokensChanged ()

Public Member Functions

- O0BaseAuth (QObject ∗parent=0, O0AbstractStore ∗store=0)
- bool linked ()
- QString token ()
- QString tokenSecret ()
- QVariantMap extraTokens ()
- QString clientId ()
- void setClientId (const QString &value)
- QString clientSecret ()
- void setClientSecret (const QString &value)
- QByteArray replyContent () const
- void setReplyContent (const QByteArray &value)
- int localPort ()
- void setLocalPort (int value)
- void setStore (O0AbstractStore ∗store)

  Sets the storage object to use for storing the OAuth tokens on a peristent medium.

Static Public Member Functions

- static QByteArray createQueryParameters (const QList< O0RequestParameter > &parameters)

  Construct query string from list of headers.

Protected Member Functions

- void setToken (const QString &v)

  Set authentication token.
- void setTokenSecret (const QString &v)

  Set authentication token secret.
- void setLinked (bool v)

  Set the linked state.
- void setExtraTokens (QVariantMap extraTokens)

  Set extra tokens found in OAuth response.

Protected Attributes

- QString clientId_
- QString clientSecret_
- QString redirectUri_
- QString requestToken_
- QString requestTokenSecret_
- QUrl requestTokenUrl_
- QUrl authorizeUrl_
- QUrl accessTokenUrl_
- quint16 localPort_
- O0AbstractStore ∗ store_
- QVariantMap extraTokens_
- O2ReplyServer ∗ replyServer_

Properties

- bool linked

  Are we authenticated?
- QString token

  Authentication token.
- QString tokenSecret

  Authentication token secret.
- QVariantMap extraTokens

  Provider-specific extra tokens, available after a successful authentication.
- QString clientId
- QString clientSecret
- QByteArray replyContent
- int localPort

7.29.1 Detailed Description

Base class of OAuth authenticators.

7.29.2 Constructor & Destructor Documentation

7.29.2.1 O0BaseAuth::O0BaseAuth (QObject ∗ *parent* = 0, O0AbstractStore ∗ *store* = 0) [explicit]

7.29.3 Member Function Documentation

7.29.3.1 QString O0BaseAuth::clientId ( )

7.29.3.2 void O0BaseAuth::clientIdChanged ( ) [signal]

7.29.3.3 QString O0BaseAuth::clientSecret ( )

7.29.3.4 void O0BaseAuth::clientSecretChanged ( ) [signal]

7.29.3.5 void O0BaseAuth::closeBrowser ( ) [signal]

Emitted when client can close the browser window.

7.29.3.6 QByteArray O0BaseAuth::createQueryParameters (const QList< O0RequestParameter > & *parameters*) [static]

Construct query string from list of headers.

7.29.3.7 QVariantMap O0BaseAuth::extraTokens ( )

7.29.3.8 void O0BaseAuth::extraTokensChanged ( ) [signal]

7.29.3.9 virtual Q_INVOKABLE void O0BaseAuth::link ( ) [pure virtual], [slot]

Authenticate.

7.29.3.10 bool O0BaseAuth::linked ( )

7.29.3.11 void O0BaseAuth::linkedChanged ( ) [signal]

7.29.3.12 void O0BaseAuth::linkingFailed ( ) [signal]

Emitted when authentication/deauthentication failed.

7.29.3.13 void O0BaseAuth::linkingSucceeded ( ) [signal]

Emitted when authentication/deauthentication succeeded.

7.29.3.14 int O0BaseAuth::localPort ( )

7.29.3.15 void O0BaseAuth::localPortChanged ( ) [signal]

7.29.3.16 void O0BaseAuth::openBrowser (const QUrl & *url*) [signal]

Emitted when client needs to open a web browser window, with the given URL.

7.29.3.17 QByteArray O0BaseAuth::replyContent ( ) const

7.29.3.18 void O0BaseAuth::setClientId (const QString & *value*)

7.29.3.19 void O0BaseAuth::setClientSecret (const QString & *value*)

7.29.3.20 void O0BaseAuth::setExtraTokens (QVariantMap *extraTokens*) [protected]

Set extra tokens found in OAuth response.

7.29.3.21 void O0BaseAuth::setLinked (bool *v*) [protected]

Set the linked state.

7.29.3.22 void O0BaseAuth::setLocalPort (int *value*)

7.29.3.23 void O0BaseAuth::setReplyContent (const QByteArray & *value*)

7.29.3.24 void O0BaseAuth::setStore (O0AbstractStore ∗ *store*)

Sets the storage object to use for storing the OAuth tokens on a peristent medium.

7.29.3.25 void O0BaseAuth::setToken (const QString & *v*) [protected]

Set authentication token.

7.29.3.26 void O0BaseAuth::setTokenSecret (const QString & *v*) [protected]

Set authentication token secret.

7.29.3.27 QString O0BaseAuth::token ( )

7.29.3.28 void O0BaseAuth::tokenChanged ( ) [signal]

7.29.3.29 QString O0BaseAuth::tokenSecret ( )

7.29.3.30 void O0BaseAuth::tokenSecretChanged ( ) [signal]

7.29.3.31 virtual Q_INVOKABLE void O0BaseAuth::unlink ( ) [pure virtual],[slot]

De-authenticate.

7.29.4 Member Data Documentation

7.29.4.1 QUrl O0BaseAuth::accessTokenUrl_ [protected]

7.29.4.2 QUrl O0BaseAuth::authorizeUrl_ [protected]

7.29.4.3 QString O0BaseAuth::clientId_ [protected]

7.29.4.4 QString O0BaseAuth::clientSecret_ [protected]

7.29.4.5 QVariantMap O0BaseAuth::extraTokens_ [protected]

7.29.4.6 quint16 O0BaseAuth::localPort_ [protected]

7.29.4.7 QString O0BaseAuth::redirectUri_ [protected]

7.29.4.8 O2ReplyServer∗ O0BaseAuth::replyServer_ [protected]

7.29.4.9 QString O0BaseAuth::requestToken_ [protected]

7.29.4.10 QString O0BaseAuth::requestTokenSecret_ [protected]

7.29.4.11 QUrl O0BaseAuth::requestTokenUrl_ [protected]

7.29.4.12 O0AbstractStore∗ O0BaseAuth::store_ [protected]

7.29.5 Property Documentation

7.29.5.1 QString O0BaseAuth::clientId [read], [write]

Client application ID. O1 instances with the same (client ID, client secret) share the same "linked", "token" and "tokenSecret" properties.

7.29.5.2 QString O0BaseAuth::clientSecret [read], [write]

Client application secret. O1 instances with the same (client ID, client secret) share the same "linked", "token" and "tokenSecret" properties.

7.29.5.3 QVariantMap O0BaseAuth::extraTokens [read]

Provider-specific extra tokens, available after a successful authentication.

7.29.5.4 bool O0BaseAuth::linked [read], [write]

Are we authenticated?

7.29.5.5 int O0BaseAuth::localPort [read], [write]

TCP port number to use in local redirections. The OAuth "redirect_uri" will be set to "http://localhost:<localPort>/". If localPort is set to 0 (default), O2 will replace it with a free one.

7.29.5.6 QByteArray O0BaseAuth::replyContent [read], [write]

Page content on local host after successful oauth. Provide it in case you do not want to close the browser, but display something

7.29.5.7 QString O0BaseAuth::token [read]

Authentication token.

7.29.5.8 QString O0BaseAuth::tokenSecret [read]

Authentication token secret.

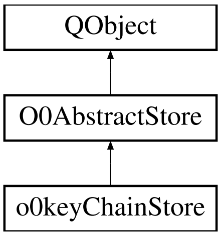
The documentation for this class was generated from the following files:

- OAuth/src/o0baseauth.h
- OAuth/src/o0baseauth.cpp

7.30 o0keyChainStore Class Reference

```
#include <o0keychainstore.h>
```

Inheritance diagram for o0keyChainStore:



Public Member Functions

- o0keyChainStore (const QString &app, const QString &name, QObject *parent=0)
- QString value (const QString &key, const QString &defaultValue=QString())

  Retrieve a string value by key.
- void setValue (const QString &key, const QString &value)

  Set a string value for a key.
- int persist ()
- int fetchFromKeychain ()
- int clearFromKeychain ()

Static Public Member Functions

- static bool isEntryNotFoundError (int errorCode)

7.30.1 Detailed Description

Calling persist(), fetchFromKeychain() and clearFromKeychain() member functions is the responsibility of the user of this class. This is important to minimize the number of keychain accesses (and potentially the number of user password prompts). For example: fetchFromKeychain() can be called immediately after creating a keychain store; persist() - after a successful authorization; clearFromKeychain() - when the user logs out from the service.

7.30.2 Constructor & Destructor Documentation

7.30.2.1 o0keyChainStore::o0keyChainStore (const QString & *app*, const QString & *name*, QObject ∗ *parent* = `0`) `[explicit]`

7.30.3 Member Function Documentation

7.30.3.1 int o0keyChainStore::clearFromKeychain ( )

7.30.3.2 int o0keyChainStore::fetchFromKeychain ( )

7.30.3.3 bool o0keyChainStore::isEntryNotFoundError (int *errorCode*) `[static]`

Returns

true if `errorCode` is equal to QKeychain::EntryNotFound.

Note

This function can be used to single out one type of an error returned from the functions above without including <keychain.h>. The EntryNotFound error type is special because it can be considered not an error if returned from clearFromKeychain().

7.30.3.4 int o0keyChainStore::persist ( )

7.30.3.5 void o0keyChainStore::setValue (const QString & *key,* const QString & *value*) [virtual]

Set a string value for a key.

Implements O0AbstractStore.

7.30.3.6 QString o0keyChainStore::value (const QString & *key,* const QString & *defaultValue* = QString()) [virtual]

Retrieve a string value by key.

Implements O0AbstractStore.

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

- OAuth/src/o0keychainstore.h
- OAuth/src/o0keychainstore.cpp

7.31 O0RequestParameter Struct Reference

Request parameter (name-value pair) participating in authentication.

`#include <o0requestparameter.h>`

Public Member Functions

- O0RequestParameter (const QByteArray &n, const QByteArray &v)
- bool operator< (const O0RequestParameter &other) const

Public Attributes

- QByteArray name
- QByteArray value

7.31.1 Detailed Description

Request parameter (name-value pair) participating in authentication.

7.31.2 Constructor & Destructor Documentation

7.31.2.1 O0RequestParameter::O0RequestParameter (const QByteArray & *n,* const QByteArray & *v*) [inline]

7.31.3 Member Function Documentation

7.31.3.1 bool O0RequestParameter::operator< (const O0RequestParameter & *other*) const [inline]

7.31.4 Member Data Documentation

7.31.4.1 QByteArray O0RequestParameter::name

7.31.4.2 QByteArray O0RequestParameter::value

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

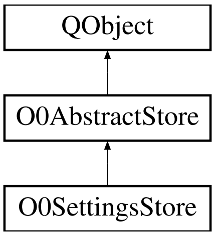
- OAuth/src/o0requestparameter.h

7.32 O0SettingsStore Class Reference

Persistent storage for authentication tokens, using QSettings.

```
#include <o0settingsstore.h>
```

Inheritance diagram for O0SettingsStore:



Signals

- void groupKeyChanged ()

Public Member Functions

- O0SettingsStore (const QString &encryptionKey, QObject ∗parent=0)

  Constructor.
- O0SettingsStore (QSettings ∗settings, const QString &encryptionKey, QObject ∗parent=0)

  Construct with an explicit QSettings instance.
- QString groupKey () const
- void setGroupKey (const QString &groupKey)
- QString value (const QString &key, const QString &defaultValue=QString())

  Get a string value for a key.
- void setValue (const QString &key, const QString &value)

  Set a string value for a key.

Protected Attributes

- QSettings ∗ settings_
- QString groupKey_
- O0SimpleCrypt crypt_

Properties

- QString groupKey

  Group key prefix.

7.32.1 Detailed Description

Persistent storage for authentication tokens, using QSettings.

7.32.2 Constructor & Destructor Documentation

7.32.2.1 O0SettingsStore::O0SettingsStore (const QString & *encryptionKey*, QObject ∗ *parent* = 0) [explicit]

Constructor.

7.32.2.2 O0SettingsStore::O0SettingsStore (QSettings ∗ *settings*, const QString & *encryptionKey*, QObject ∗ *parent* = 0) [explicit]

Construct with an explicit QSettings instance.

7.32.3 Member Function Documentation

7.32.3.1 QString O0SettingsStore::groupKey ( ) const

7.32.3.2 void O0SettingsStore::groupKeyChanged ( ) [signal]

7.32.3.3 void O0SettingsStore::setGroupKey (const QString & *groupKey*)

7.32.3.4 void O0SettingsStore::setValue (const QString & *key*, const QString & *value*) [virtual]

Set a string value for a key.

Implements O0AbstractStore.

7.32.3.5 QString O0SettingsStore::value (const QString & *key*, const QString & *defaultValue* = QString()) [virtual]

Get a string value for a key.

Implements O0AbstractStore.

7.32.4 Member Data Documentation

7.32.4.1 O0SimpleCrypt O0SettingsStore::crypt_ [protected]

7.32.4.2 QString O0SettingsStore::groupKey_ [protected]

7.32.4.3 QSettings∗ O0SettingsStore::settings_ [protected]

7.32.5 Property Documentation

7.32.5.1 QString O0SettingsStore::groupKey [read], [write]

Group key prefix.

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

- OAuth/src/o0settingsstore.h
- OAuth/src/o0settingsstore.cpp

7.33 O0SimpleCrypt Class Reference

Simple encryption and decryption of strings and byte arrays.

```
#include <o0simplecrypt.h>
```

Public Types

- enum CompressionMode { CompressionAuto, CompressionAlways, CompressionNever }
- enum IntegrityProtectionMode { ProtectionNone, ProtectionChecksum, ProtectionHash }
- enum Error { ErrorNoError, ErrorNoKeySet, ErrorUnknownVersion, ErrorIntegrityFailed }
- enum CryptoFlag { CryptoFlagNone = 0, CryptoFlagCompression = 0x01, CryptoFlagChecksum = 0x02, CryptoFlagHash = 0x04 }

Public Member Functions

- O0SimpleCrypt ()
- O0SimpleCrypt (quint64 key)
- void setKey (quint64 key)
- bool hasKey () const
- void setCompressionMode (CompressionMode mode)
- CompressionMode compressionMode () const
- void setIntegrityProtectionMode (IntegrityProtectionMode mode)
- IntegrityProtectionMode integrityProtectionMode () const
- Error lastError () const
- QString encryptToString (const QString &plaintext)
- QString encryptToString (QByteArray plaintext)
- QByteArray encryptToByteArray (const QString &plaintext)
- QByteArray encryptToByteArray (QByteArray plaintext)
- QString decryptToString (const QString &cyphertext)
- QByteArray decryptToByteArray (const QString &cyphertext)
- QString decryptToString (QByteArray cypher)
- QByteArray decryptToByteArray (QByteArray cypher)

7.33.1 Detailed Description

Simple encryption and decryption of strings and byte arrays.

This class provides a simple implementation of encryption and decryption of strings and byte arrays.

Warning

> The encryption provided by this class is NOT strong encryption. It may help to shield things from curious eyes, but it will NOT stand up to someone determined to break the encryption. Don't say you were not warned.

The class uses a 64 bit key. Simply create an instance of the class, set the key, and use the encryptToString() method to calculate an encrypted version of the input string. To decrypt that string again, use an instance of SimpleCrypt initialized with the same key, and call the decryptToString() method with the encrypted string. If the key matches, the decrypted version of the string will be returned again.

If you do not provide a key, or if something else is wrong, the encryption and decryption function will return an empty string or will return a string containing nonsense. lastError() will return a value indicating if the method was succesful, and if not, why not.

SimpleCrypt is prepared for the case that the encryption and decryption algorithm is changed in a later version, by prepending a version identifier to the cypertext.

7.33.2 Member Enumeration Documentation

7.33.2.1 enum O0SimpleCrypt::CompressionMode

CompressionMode describes if compression will be applied to the data to be encrypted.

Enumerator

CompressionAuto Only apply compression if that results in a shorter plaintext.

CompressionAlways Always apply compression. Note that for short inputs, a compression may result in longer data

CompressionNever Never apply compression.

7.33.2.2 enum O0SimpleCrypt::CryptoFlag

Enumerator

CryptoFlagNone

CryptoFlagCompression

CryptoFlagChecksum

CryptoFlagHash

7.33.2.3 enum O0SimpleCrypt::Error

Error describes the type of error that occured.

Enumerator

ErrorNoError No error occurred.

ErrorNoKeySet No key was set. You can not encrypt or decrypt without a valid key.

ErrorUnknownVersion The version of this data is unknown, or the data is otherwise not valid.

ErrorIntegrityFailed The integrity check of the data failed. Perhaps the wrong key was used.

7.33.2.4 enum O0SimpleCrypt::IntegrityProtectionMode

IntegrityProtectionMode describes measures taken to make it possible to detect problems with the data or wrong decryption keys.

Measures involve adding a checksum or a cryptograhpic hash to the data to be encrypted. This increases the length of the resulting cypertext, but makes it possible to check if the plaintext appears to be valid after decryption.

Enumerator

ProtectionNone The integerity of the encrypted data is not protected. It is not really possible to detect a wrong key, for instance.

ProtectionChecksum A simple checksum is used to verify that the data is in order. If not, an empty string is returned.

ProtectionHash A cryptographic hash is used to verify the integrity of the data. This method produces a much stronger, but longer check

7.33.3 Constructor & Destructor Documentation

7.33.3.1 O0SimpleCrypt::O0SimpleCrypt ( )

Constructor.

Constructs a SimpleCrypt instance without a valid key set on it.

7.33.3.2 O0SimpleCrypt::O0SimpleCrypt (quint64 *key*) [explicit]

Constructor.

Constructs a SimpleCrypt instance and initializes it with the given

- key.

7.33.4 Member Function Documentation

7.33.4.1 CompressionMode O0SimpleCrypt::compressionMode ( ) const [inline]

Returns the CompressionMode that is currently in use.

7.33.4.2 QByteArray O0SimpleCrypt::decryptToByteArray (const QString & *cyphertext*)

Decrypts a cyphertext string encrypted with this class with the set key back to the plain text version.

If an error occured, such as non-matching keys between encryption and decryption, an empty string or a string containing nonsense may be returned.

7.33.4.3 QByteArray O0SimpleCrypt::decryptToByteArray (QByteArray *cypher*)

Decrypts a cyphertext binary encrypted with this class with the set key back to the plain text version.

If an error occured, such as non-matching keys between encryption and decryption, an empty string or a string containing nonsense may be returned.

7.33.4.4 QString O0SimpleCrypt::decryptToString (const QString & *cyphertext*)

Decrypts a cyphertext string encrypted with this class with the set key back to the plain text version.

If an error occured, such as non-matching keys between encryption and decryption, an empty string or a string containing nonsense may be returned.

7.33.4.5 QString O0SimpleCrypt::decryptToString (QByteArray *cypher*)

Decrypts a cyphertext binary encrypted with this class with the set key back to the plain text version.

If an error occured, such as non-matching keys between encryption and decryption, an empty string or a string containing nonsense may be returned.

7.33.4.6 QByteArray O0SimpleCrypt::encryptToByteArray (const QString & *plaintext*)

Encrypts the

- plaintext string with the key the class was initialized with, and returns a binary cyphertext in a QByteArray the result.

This method returns a byte array, that is useable for storing a binary format. If you need a string you can store in a text file, use encryptToString() instead.

7.33.4.7 QByteArray O0SimpleCrypt::encryptToByteArray (QByteArray *plaintext*)

Encrypts the

- plaintext QByteArray with the key the class was initialized with, and returns a binary cyphertext in a QByteArray the result.

This method returns a byte array, that is useable for storing a binary format. If you need a string you can store in a text file, use encryptToString() instead.

7.33.4.8 QString O0SimpleCrypt::encryptToString (const QString & *plaintext*)

Encrypts the

- plaintext string with the key the class was initialized with, and returns a cyphertext the result. The result is a base64 encoded version of the binary array that is the actual result of the string, so it can be stored easily in a text format.

7.33.4.9 QString O0SimpleCrypt::encryptToString (QByteArray *plaintext*)

Encrypts the

- plaintext QByteArray with the key the class was initialized with, and returns a cyphertext the result. The result is a base64 encoded version of the binary array that is the actual result of the encryption, so it can be stored easily in a text format.

7.33.4.10 bool O0SimpleCrypt::hasKey ( ) const [inline]

Returns true if SimpleCrypt has been initialized with a key.

7.33.4.11 IntegrityProtectionMode O0SimpleCrypt::integrityProtectionMode ( ) const [inline]

Returns the IntegrityProtectionMode that is currently in use.

7.33.4.12 Error O0SimpleCrypt::lastError ( ) const [inline]

Returns the last error that occurred.

7.33.4.13 void O0SimpleCrypt::setCompressionMode (CompressionMode *mode*) [inline]

Sets the compression mode to use when encrypting data. The default mode is Auto.

Note that decryption is not influenced by this mode, as the decryption recognizes what mode was used when encrypting.

7.33.4.14 void O0SimpleCrypt::setIntegrityProtectionMode (IntegrityProtectionMode *mode*) [inline]

Sets the integrity mode to use when encrypting data. The default mode is Checksum.

Note that decryption is not influenced by this mode, as the decryption recognizes what mode was used when encrypting.

7.33.4.15 void O0SimpleCrypt::setKey (quint64 *key*)

(Re-) initializes the key with the given

- key.

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

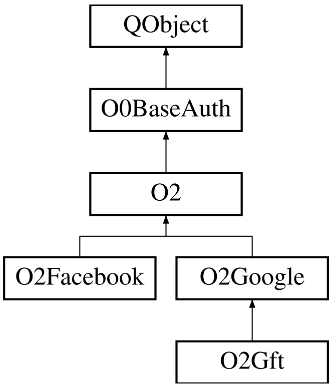
- OAuth/src/o0simplecrypt.h
- OAuth/src/o2simplecrypt.cpp

7.34 O2 Class Reference

Simple OAuth2 authenticator.

```
#include <o2.h>
```

Inheritance diagram for O2:



Public Types

- enum GrantFlow { GrantFlowAuthorizationCode, GrantFlowImplicit, GrantFlowResourceOwnerPasswordCredentials }

  Authorization flow types.

Public Slots

- virtual Q_INVOKABLE void link ()

  Authenticate.
- virtual Q_INVOKABLE void unlink ()

  De-authenticate.
- Q_INVOKABLE void refresh ()

  Refresh token.
- void serverHasClosed (bool paramsfound=false)

  Handle situation where reply server has opted to close its connection.

Signals

- void refreshFinished (QNetworkReply::NetworkError error)

  Emitted when a token refresh has been completed or failed.
- void grantFlowChanged ()
- void scopeChanged ()
- void usernameChanged ()
- void passwordChanged ()
- void requestUrlChanged ()
- void extraRequestParamsChanged ()
- void refreshTokenUrlChanged ()
- void tokenUrlChanged ()

Public Member Functions

- GrantFlow grantFlow ()
- void setGrantFlow (GrantFlow value)
- QString username ()
- void setUsername (const QString &value)
- QString password ()
- void setPassword (const QString &value)
- QString scope ()
- void setScope (const QString &value)
- QString localhostPolicy () const
- void setLocalhostPolicy (const QString &value)
- QString apiKey ()
- void setApiKey (const QString &value)
- bool ignoreSslErrors ()
- void setIgnoreSslErrors (bool ignoreSslErrors)
- QString requestUrl ()
- void setRequestUrl (const QString &value)
- QVariantMap extraRequestParams ()
- void setExtraRequestParams (const QVariantMap &value)
- QString tokenUrl ()
- void setTokenUrl (const QString &value)
- QString refreshTokenUrl ()
- void setRefreshTokenUrl (const QString &value)
- O2 (QObject ∗parent=0, QNetworkAccessManager ∗manager=0, O0AbstractStore ∗store=0)
- QString code ()

  Get authentication code.
- QString refreshToken ()

  Get refresh token.
- int expires ()

  Get token expiration time (seconds from Epoch).

Protected Slots

- virtual void onVerificationReceived (QMap< QString, QString >)

  Handle verification response.
- virtual void onTokenReplyFinished ()

  Handle completion of a token request.
- virtual void onTokenReplyError (QNetworkReply::NetworkError error)

  Handle failure of a token request.

- virtual void onRefreshFinished ()

  Handle completion of a refresh request.
- virtual void onRefreshError (QNetworkReply::NetworkError error)

  Handle failure of a refresh request.

Protected Member Functions

- QByteArray buildRequestBody (const QMap< QString, QString > &parameters)

  Build HTTP request body.
- void setCode (const QString &v)

  Set authentication code.
- void setRefreshToken (const QString &v)

  Set refresh token.
- void setExpires (int v)

  Set token expiration time.

Protected Attributes

- QString username_
- QString password_
- QUrl requestUrl_
- QVariantMap extraReqParams_
- QUrl tokenUrl_
- QUrl refreshTokenUrl_
- QString scope_
- QString code_
- QString localhostPolicy_
- QString apiKey_
- QNetworkAccessManager ∗ manager_
- O2ReplyList timedReplies_
- GrantFlow grantFlow_

Properties

- GrantFlow grantFlow

  Authorization flow.
- QString username
- QString password
- QString scope

  Scope of authentication.
- QString localhostPolicy
- QString apiKey

  API key.
- bool ignoreSslErrors
- QString requestUrl

  Request URL.
- QVariantMap extraRequestParams

  User-defined extra parameters to append to request URL.
- QString tokenUrl

  Token request URL.
- QString refreshTokenUrl

  Token refresh URL.

Additional Inherited Members

7.34.1 Detailed Description

Simple OAuth2 authenticator.

7.34.2 Member Enumeration Documentation

7.34.2.1 enum O2::GrantFlow

Authorization flow types.

Enumerator

GrantFlowAuthorizationCode **See Also**

`http://tools.ietf.org/html/draft-ietf-oauth-v2-15#section-4`
`1`

GrantFlowImplicit **See Also**

`http://tools.ietf.org/html/draft-ietf-oauth-v2-15#section-4.2`

GrantFlowResourceOwnerPasswordCredentials

7.34.3 Constructor & Destructor Documentation

7.34.3.1 O2::O2 (QObject ∗ *parent* = 0, QNetworkAccessManager ∗ *manager* = 0, O0AbstractStore ∗ *store* = 0) [explicit]

Constructor.

Parameters

parent	Parent object.

7.34.4 Member Function Documentation

7.34.4.1 QString O2::apiKey ( )

7.34.4.2 QByteArray O2::buildRequestBody (const QMap< QString, QString > & *parameters*) [protected]

Build HTTP request body.

7.34.4.3 QString O2::code ( )

Get authentication code.

7.34.4.4 int O2::expires ( )

Get token expiration time (seconds from Epoch).

7.34.4.5 QVariantMap O2::extraRequestParams ( )

7.34.4.6 void O2::extraRequestParamsChanged ( ) [signal]

7.34.4.7 GrantFlow O2::grantFlow ( )

7.34.4.8 void O2::grantFlowChanged ( ) [signal]

7.34.4.9 bool O2::ignoreSslErrors ( )

7.34.4.10 void O2::link ( ) [virtual], [slot]

Authenticate.

7.34.4.11 QString O2::localhostPolicy ( ) const

7.34.4.12 void O2::onRefreshError (QNetworkReply::NetworkError *error*) [protected], [virtual], [slot]

Handle failure of a refresh request.

7.34.4.13 void O2::onRefreshFinished ( ) [protected], [virtual], [slot]

Handle completion of a refresh request.

7.34.4.14 void O2::onTokenReplyError (QNetworkReply::NetworkError *error*) [protected], [virtual], [slot]

Handle failure of a token request.

7.34.4.15 void O2::onTokenReplyFinished ( ) [protected], [virtual], [slot]

Handle completion of a token request.

7.34.4.16 void O2::onVerificationReceived (QMap< QString, QString > *response*) [protected], [virtual], [slot]

Handle verification response.

7.34.4.17 QString O2::password ( )

7.34.4.18 void O2::passwordChanged ( ) [signal]

7.34.4.19 void O2::refresh ( ) [slot]

Refresh token.

7.34.4.20 void O2::refreshFinished (QNetworkReply::NetworkError *error*) [signal]

Emitted when a token refresh has been completed or failed.

7.34.4.21 QString O2::refreshToken ( )

Get refresh token.

7.34.4.22 QString O2::refreshTokenUrl ( )

7.34.4.23 void O2::refreshTokenUrlChanged ( ) [signal]

7.34.4.24 QString O2::requestUrl ( )

7.34.4.25 void O2::requestUrlChanged ( ) [signal]

7.34.4.26 QString O2::scope ( )

7.34.4.27 void O2::scopeChanged ( ) [signal]

7.34.4.28 void O2::serverHasClosed (bool *paramsfound* = false) [slot]

Handle situation where reply server has opted to close its connection.

7.34.4.29 void O2::setApiKey (const QString & *value*)

7.34.4.30 void O2::setCode (const QString & *v*) [protected]

Set authentication code.

7.34.4.31 void O2::setExpires (int *v*) [protected]

Set token expiration time.

7.34.4.32 void O2::setExtraRequestParams (const QVariantMap & *value*)

7.34.4.33 void O2::setGrantFlow (O2::GrantFlow *value*)

7.34.4.34 void O2::setIgnoreSslErrors (bool *ignoreSslErrors*)

7.34.4.35 void O2::setLocalhostPolicy (const QString & *value*)

7.34.4.36 void O2::setPassword (const QString & *value*)

7.34.4.37 void O2::setRefreshToken (const QString & *v*) [protected]

Set refresh token.

7.34.4.38 void O2::setRefreshTokenUrl (const QString & *value*)

7.34.4.39 void O2::setRequestUrl (const QString & *value*)

7.34.4.40 void O2::setScope (const QString & *value*)

7.34.4.41 void O2::setTokenUrl (const QString & *value*)

7.34.4.42 void O2::setUsername (const QString & *value*)

7.34.4.43 QString O2::tokenUrl ( )

7.34.4.44 void O2::tokenUrlChanged ( ) [signal]

7.34.4.45 void O2::unlink ( ) [virtual], [slot]

De-authenticate.

7.34.4.46 QString O2::username ( )

7.34.4.47 void O2::usernameChanged ( ) [signal]

7.34.5 Member Data Documentation

7.34.5.1 QString O2::apiKey_ [protected]

7.34.5.2 QString O2::code_ [protected]

7.34.5.3 QVariantMap O2::extraReqParams_ [protected]

7.34.5.4 GrantFlow O2::grantFlow_ [protected]

7.34.5.5 QString O2::localhostPolicy_ [protected]

7.34.5.6 QNetworkAccessManager∗ O2::manager_ [protected]

7.34.5.7 QString O2::password_ [protected]

7.34.5.8 QUrl O2::refreshTokenUrl_ [protected]

7.34.5.9 QUrl O2::requestUrl_ [protected]

7.34.5.10 QString O2::scope_ [protected]

7.34.5.11 O2ReplyList O2::timedReplies_ [protected]

7.34.5.12 QUrl O2::tokenUrl_ [protected]

7.34.5.13 QString O2::username_ [protected]

7.34.6 Property Documentation

7.34.6.1 QString O2::apiKey [read], [write]

API key.

7.34.6.2 QVariantMap O2::extraRequestParams [read], [write]

User-defined extra parameters to append to request URL.

7.34.6.3 O2::GrantFlow O2::grantFlow [read], [write]

Authorization flow.

7.34.6.4 bool O2::ignoreSslErrors [read], [write]

Allow ignoring SSL errors? E.g. SurveyMonkey fails on Mac due to SSL error. Ignoring the error circumvents the problem

7.34.6.5 QString O2::localhostPolicy [read], [write]

Localhost policy. By default it's value is `http://127.0.0.1:%1/`, however some services may require the use of `http://localhost:%1/` or any other value.

7.34.6.6 QString O2::password [read], [write]

Resource owner password. O2 instances with the same (username, password) share the same "linked" and "token" properties.

7.34.6.7 QString O2::refreshTokenUrl [read], [write]

Token refresh URL.

7.34.6.8 QString O2::requestUrl [read], [write]

Request URL.

7.34.6.9 QString O2::scope [read], [write]

Scope of authentication.

7.34.6.10 QString O2::tokenUrl [read], [write]

Token request URL.

7.34.6.11 QString O2::username [read], [write]

Resource owner username. O2 instances with the same (username, password) share the same "linked" and "token" properties.

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

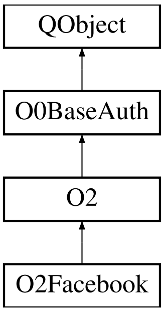
- OAuth/src/o2.h
- OAuth/src/o2.cpp

7.35 O2Facebook Class Reference

Facebook's dialect of OAuth 2.0.

```
#include <o2facebook.h>
```

Inheritance diagram for O2Facebook:



Public Member Functions

- O2Facebook (QObject ∗parent=0)

Protected Slots

- void onVerificationReceived (QMap< QString, QString >)
- virtual void onTokenReplyFinished ()

Additional Inherited Members

7.35.1 Detailed Description

Facebook's dialect of OAuth 2.0.

7.35.2 Constructor & Destructor Documentation

7.35.2.1 O2Facebook::O2Facebook (QObject ∗ *parent* = 0) [explicit]

7.35.3 Member Function Documentation

7.35.3.1 void O2Facebook::onTokenReplyFinished ( ) [protected], [virtual], [slot]

7.35.3.2 void O2Facebook::onVerificationReceived (QMap< QString, QString > *response*) [protected], [slot]

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

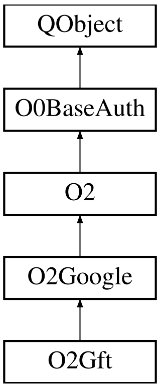
- OAuth/src/o2facebook.h
- OAuth/src/o2facebook.cpp

7.36 O2Gft Class Reference

Google Fusion Tables' dialect of OAuth 2.0.

```
#include <o2gft.h>
```

Inheritance diagram for O2Gft:



Public Member Functions

- O2Gft (QObject ∗parent=0)

Additional Inherited Members

7.36.1 Detailed Description

Google Fusion Tables' dialect of OAuth 2.0.

7.36.2 Constructor & Destructor Documentation

7.36.2.1 O2Gft::O2Gft (QObject ∗ *parent* = 0) [explicit]

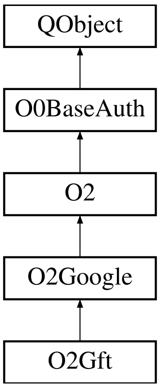
The documentation for this class was generated from the following files:

- OAuth/src/o2gft.h
- OAuth/src/o2gft.cpp

7.37 O2Google Class Reference

```
#include <o2google.h>
```

Inheritance diagram for O2Google:



Public Member Functions

- O2Google (QObject ∗parent=0)

Additional Inherited Members

7.37.1 Constructor & Destructor Documentation

7.37.1.1 O2Google::O2Google (QObject ∗ *parent* = 0) [explicit]

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

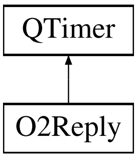
- OAuth/src/o2google.h
- OAuth/src/o2google.cpp

7.38 O2Reply Class Reference

A network request/reply pair that can time out.

`#include <o2reply.h>`

Inheritance diagram for O2Reply:



Public Slots

- void onTimeOut ()

    When time out occurs, the QNetworkReply's error() signal is triggered.

Signals

- void error (QNetworkReply::NetworkError)

Public Member Functions

- O2Reply (QNetworkReply ∗reply, int timeOut=60 ∗1000, QObject ∗parent=0)

Public Attributes

- QNetworkReply ∗ reply

7.38.1 Detailed Description

A network request/reply pair that can time out.

7.38.2 Constructor & Destructor Documentation

7.38.2.1 O2Reply::O2Reply (QNetworkReply ∗ *reply*, int *timeOut* = `60 * 1000`, QObject ∗ *parent* = `0`)

7.38.3 Member Function Documentation

7.38.3.1 void O2Reply::error (QNetworkReply::NetworkError) `[signal]`

7.38.3.2 void O2Reply::onTimeOut ( ) `[slot]`

When time out occurs, the QNetworkReply's error() signal is triggered.

7.38.4 Member Data Documentation

7.38.4.1 QNetworkReply∗ O2Reply::reply

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

- OAuth/src/o2reply.h
- OAuth/src/o2reply.cpp

7.39 O2ReplyList Class Reference

List of O2Replies.

```
#include <o2reply.h>
```

Public Member Functions

- O2ReplyList ()
- virtual ∼O2ReplyList ()
- void add (QNetworkReply ∗reply)

  Create a new O2Reply from a QNetworkReply, and add it to this list.
- void add (O2Reply ∗reply)

  Add an O2Reply to the list, while taking ownership of it.
- void remove (QNetworkReply ∗reply)

  Remove item from the list that corresponds to a QNetworkReply.
- O2Reply ∗ find (QNetworkReply ∗reply)
- bool ignoreSslErrors ()
- void setIgnoreSslErrors (bool ignoreSslErrors)

Protected Attributes

- QList< O2Reply ∗ > replies_
- bool ignoreSslErrors_

7.39.1 Detailed Description

List of O2Replies.

7.39.2 Constructor & Destructor Documentation

7.39.2.1 O2ReplyList::O2ReplyList ( ) [inline]

7.39.2.2 O2ReplyList::∼O2ReplyList ( ) [virtual]

Destructor. Deletes all O2Reply instances in the list.

7.39.3 Member Function Documentation

7.39.3.1 void O2ReplyList::add (QNetworkReply ∗ *reply*)

Create a new O2Reply from a QNetworkReply, and add it to this list.

7.39.3.2 void O2ReplyList::add (O2Reply ∗ *reply*)

Add an O2Reply to the list, while taking ownership of it.

7.39.3.3 O2Reply ∗ O2ReplyList::find (QNetworkReply ∗ *reply*)

Find an O2Reply in the list, corresponding to a QNetworkReply.

Returns

Matching O2Reply or NULL.

7.39.3.4 bool O2ReplyList::ignoreSslErrors ( )

7.39.3.5 void O2ReplyList::remove (QNetworkReply ∗ *reply*)

Remove item from the list that corresponds to a QNetworkReply.

7.39.3.6 void O2ReplyList::setIgnoreSslErrors (bool *ignoreSslErrors*)

7.39.4 Member Data Documentation

7.39.4.1 bool O2ReplyList::ignoreSslErrors_ [protected]

7.39.4.2 QList<O2Reply ∗> O2ReplyList::replies_ [protected]

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

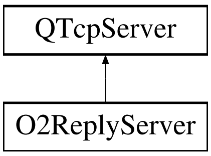
- OAuth/src/o2reply.h
- OAuth/src/o2reply.cpp

7.40 O2ReplyServer Class Reference

HTTP server to process authentication response.

```
#include <o2replyserver.h>
```

Inheritance diagram for O2ReplyServer:



Public Slots

- void onIncomingConnection ()
- void onBytesReady ()
- QMap< QString, QString > parseQueryParams (QByteArray ∗data)
- void closeServer (QTcpSocket ∗socket=0, bool hasparameters=false)

Signals

- void verificationReceived (QMap< QString, QString >)
- void serverClosed (bool)

Public Member Functions

- O2ReplyServer (QObject ∗parent=0)
- QByteArray replyContent ()
- void setReplyContent (const QByteArray &value)
- int timeout ()
- void setTimeout (int timeout)
- int callbackTries ()
- void setCallbackTries (int maxtries)

Protected Attributes

- QByteArray replyContent_
- int timeout_
- int maxtries_
- int tries_

Properties

- QByteArray replyContent

  Page content on local host after successful oauth - in case you do not want to close the browser, but display something.
- int timeout

  Seconds to keep listening after first response for a callback with token content.
- int callbackTries

  Maximum number of callback tries to accept, in case some don't have token content (favicons, etc.)

7.40.1 Detailed Description

HTTP server to process authentication response.

7.40.2 Constructor & Destructor Documentation

7.40.2.1 O2ReplyServer::O2ReplyServer (QObject * *parent* = 0) [explicit]

7.40.3 Member Function Documentation

7.40.3.1 int O2ReplyServer::callbackTries ( )

7.40.3.2 void O2ReplyServer::closeServer (QTcpSocket * *socket* = 0, bool *hasparameters* = false) [slot]

7.40.3.3 void O2ReplyServer::onBytesReady ( ) [slot]

7.40.3.4 void O2ReplyServer::onIncomingConnection ( ) [slot]

7.40.3.5 QMap< QString, QString > O2ReplyServer::parseQueryParams (QByteArray * *data*) [slot]

7.40.3.6 QByteArray O2ReplyServer::replyContent ( )

7.40.3.7 void O2ReplyServer::serverClosed (bool) [signal]

7.40.3.8 void O2ReplyServer::setCallbackTries (int *maxtries*)

7.40.3.9 void O2ReplyServer::setReplyContent (const QByteArray & *value*)

7.40.3.10 void O2ReplyServer::setTimeout (int *timeout*)

7.40.3.11 int O2ReplyServer::timeout ( )

7.40.3.12 void O2ReplyServer::verificationReceived (QMap< QString, QString >) [signal]

7.40.4 Member Data Documentation

7.40.4.1 int O2ReplyServer::maxtries_ [protected]

7.40.4.2 QByteArray O2ReplyServer::replyContent_ [protected]

7.40.4.3 int O2ReplyServer::timeout_ [protected]

7.40.4.4 int O2ReplyServer::tries_ [protected]

7.40.5 Property Documentation

7.40.5.1 int O2ReplyServer::callbackTries [read], [write]

Maximum number of callback tries to accept, in case some don't have token content (favicons, etc.)

7.40.5.2 QByteArray O2ReplyServer::replyContent [read], [write]

Page content on local host after successful oauth - in case you do not want to close the browser, but display something.

7.40.5.3 int O2ReplyServer::timeout [read], [write]

Seconds to keep listening *after* first response for a callback with token content.

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

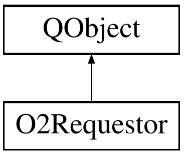
- OAuth/src/o2replyserver.h
- OAuth/src/o2replyserver.cpp

7.41 O2Requestor Class Reference

Makes authenticated requests.

```
#include <o2requestor.h>
```

Inheritance diagram for O2Requestor:



Public Slots

- int get (const QNetworkRequest &req)
- int post (const QNetworkRequest &req, const QByteArray &data)
- int put (const QNetworkRequest &req, const QByteArray &data)

Signals

- void finished (int id, QNetworkReply::NetworkError error, QByteArray data)

  Emitted when a request has been completed or failed.
- void uploadProgress (int id, qint64 bytesSent, qint64 bytesTotal)

  Emitted when an upload has progressed.

Public Member Functions

- O2Requestor (QNetworkAccessManager ∗manager, O2 ∗authenticator, QObject ∗parent=0)
- ∼O2Requestor ()

Protected Types

- enum Status { Idle, Requesting, ReRequesting }

Protected Slots

- void onRefreshFinished (QNetworkReply::NetworkError error)

  Handle refresh completion.
- void onRequestFinished ()

  Handle request finished.
- void onRequestError (QNetworkReply::NetworkError error)

  Handle request error.
- void retry ()

  Re-try request (after successful token refresh).

- void finish ()

  Finish the request, Q_EMIT finished() signal.
- void onUploadProgress (qint64 uploaded, qint64 total)

  Handle upload progress.

Protected Member Functions

- int setup (const QNetworkRequest &request, QNetworkAccessManager::Operation operation)

Protected Attributes

- QNetworkAccessManager ∗ manager_
- O2 ∗ authenticator_
- QNetworkRequest request_
- QByteArray data_
- QNetworkReply ∗ reply_
- Status status_
- int id_
- QNetworkAccessManager::Operation operation_
- QUrl url_
- O2ReplyList timedReplies_
- QNetworkReply::NetworkError error_

7.41.1 Detailed Description

Makes authenticated requests.

7.41.2 Member Enumeration Documentation

7.41.2.1 enum **O2Requestor::Status** `[protected]`

Enumerator

Idle

Requesting

ReRequesting

7.41.3 Constructor & Destructor Documentation

7.41.3.1 O2Requestor::O2Requestor (QNetworkAccessManager ∗ *manager,* O2 ∗ *authenticator,* QObject ∗ *parent =* `0`) `[explicit]`

7.41.3.2 O2Requestor::∼O2Requestor ( )

7.41.4 Member Function Documentation

7.41.4.1 void O2Requestor::finish ( ) `[protected]`, `[slot]`

Finish the request, Q_EMIT finished() signal.

7.41.4.2 void O2Requestor::finished (int *id*, QNetworkReply::NetworkError *error*, QByteArray *data*) [signal]

Emitted when a request has been completed or failed.

7.41.4.3 int O2Requestor::get (const QNetworkRequest & *req*) [slot]

Make a GET request.

Returns

Request ID or -1 if there are too many requests in the queue.

7.41.4.4 void O2Requestor::onRefreshFinished (QNetworkReply::NetworkError *error*) [protected], [slot]

Handle refresh completion.

7.41.4.5 void O2Requestor::onRequestError (QNetworkReply::NetworkError *error*) [protected], [slot]

Handle request error.

7.41.4.6 void O2Requestor::onRequestFinished ( ) [protected], [slot]

Handle request finished.

7.41.4.7 void O2Requestor::onUploadProgress (qint64 *uploaded*, qint64 *total*) [protected], [slot]

Handle upload progress.

7.41.4.8 int O2Requestor::post (const QNetworkRequest & *req*, const QByteArray & *data*) [slot]

Make a POST request.

Returns

Request ID or -1 if there are too many requests in the queue.

7.41.4.9 int O2Requestor::put (const QNetworkRequest & *req*, const QByteArray & *data*) [slot]

Make a PUT request.

Returns

Request ID or -1 if there are too many requests in the queue.

7.41.4.10 void O2Requestor::retry ( ) [protected], [slot]

Re-try request (after successful token refresh).

7.41.4.11 int O2Requestor::setup (const QNetworkRequest & *request*, QNetworkAccessManager::Operation *operation*) [protected]

7.41.4.12 void O2Requestor::uploadProgress (int *id*, qint64 *bytesSent*, qint64 *bytesTotal*) [signal]

Emitted when an upload has progressed.

7.41.5 Member Data Documentation

7.41.5.1 O2∗ O2Requestor::authenticator_ [protected]

7.41.5.2 QByteArray O2Requestor::data_ [protected]

7.41.5.3 QNetworkReply::NetworkError O2Requestor::error_ [protected]

7.41.5.4 int O2Requestor::id_ [protected]

7.41.5.5 QNetworkAccessManager∗ O2Requestor::manager_ [protected]

7.41.5.6 QNetworkAccessManager::Operation O2Requestor::operation_ [protected]

7.41.5.7 QNetworkReply∗ O2Requestor::reply_ [protected]

7.41.5.8 QNetworkRequest O2Requestor::request_ [protected]

7.41.5.9 Status O2Requestor::status_ [protected]

7.41.5.10 O2ReplyList O2Requestor::timedReplies_ [protected]

7.41.5.11 QUrl O2Requestor::url_ [protected]

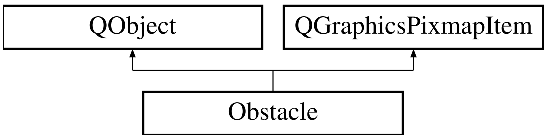
The documentation for this class was generated from the following files:

- OAuth/src/o2requestor.h
- OAuth/src/o2requestor.cpp

7.42 Obstacle Class Reference

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

Inheritance diagram for Obstacle:



Public Member Functions

- Obstacle (QObject ∗parent=0)
- void setIdentity (int id)
- int getIdentity ()

7.42.1 Constructor & Destructor Documentation

7.42.1.1 Obstacle::Obstacle (QObject ∗ *parent* = 0) [explicit]

7.42.2 Member Function Documentation

7.42.2.1 int Obstacle::getIdentity ( )

7.42.2.2 void Obstacle::setIdentity (int *id*)

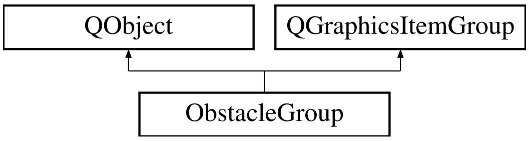
The documentation for this class was generated from the following files:

- Game1/obstacle.h
- Game1/obstacle.cpp

7.43 ObstacleGroup Class Reference

```
#include <obstaclegroup.h>
```

Inheritance diagram for ObstacleGroup:



Public Slots

- void move ()

    ObstacleGroup::move.

Public Member Functions

- ObstacleGroup (QObject ∗parent=0)
- QString getText ()
- bool getType ()
- void setIdentity (int id)
- int getIdentity ()
- QGraphicsTextItem ∗ getLabel ()
- void setRand (int random)
- void setDifficulty (int difficulty)

7.43.1 Constructor & Destructor Documentation

7.43.1.1 ObstacleGroup::ObstacleGroup (QObject ∗ *parent* = 0) [explicit]

Generating a random value from 2 lists: values and vices.

Goruping the value (label) with an obstacle.

7.43.2 Member Function Documentation

7.43.2.1 int ObstacleGroup::getIdentity ( )

7.43.2.2 QGraphicsTextItem ∗ ObstacleGroup::getLabel ( )

7.43.2.3 QString ObstacleGroup::getText ( )

7.43.2.4 bool ObstacleGroup::getType ( )

7.43.2.5 void ObstacleGroup::move ( ) [slot]

ObstacleGroup::move.

Speed increments as difficulty increases.

Move group left or right according to lane it is in.

Detecting collisions and adding catched values in output files: values and vices.

Removing groups when they exceed bounds.

7.43.2.6 void ObstacleGroup::setDifficulty (int *difficulty*)

7.43.2.7 void ObstacleGroup::setIdentity (int *id*)

7.43.2.8 void ObstacleGroup::setRand (int *random*)

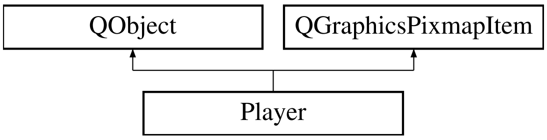
The documentation for this class was generated from the following files:

- Game1/obstaclegroup.h
- Game1/obstaclegroup.cpp

7.44 Player Class Reference

`#include <player.h>`

Inheritance diagram for Player:



Public Member Functions

- Player (QObject ∗parent=0)
- void setDifficulty (int diff)

  Character::setDifficulty.
- void move (QKeyEvent ∗event)

7.44.1 Constructor & Destructor Documentation

7.44.1.1 Player::Player (QObject ∗ *parent* = 0) [explicit]

7.44.2 Member Function Documentation

7.44.2.1 void Player::move (QKeyEvent ∗ *event*)

7.44.2.2 void Player::setDifficulty (int *diff*)

Character::setDifficulty.

Parameters

diff	

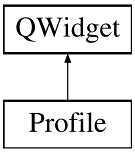
The documentation for this class was generated from the following files:

- Game3/player.h
- Game3/player.cpp

7.45 Profile Class Reference

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

Inheritance diagram for Profile:



Public Member Functions

- Profile (QWidget ∗parent=0)
- ∼Profile ()

7.45.1 Constructor & Destructor Documentation

7.45.1.1 Profile::Profile (QWidget ∗ *parent* = 0) [explicit]

7.45.1.2 Profile::∼Profile ( )

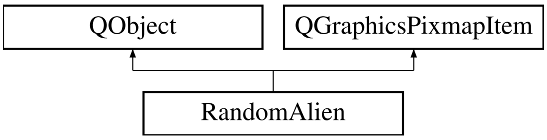
The documentation for this class was generated from the following files:

- profile.h
- profile.cpp

7.46 RandomAlien Class Reference

```
#include <randomalien.h>
```

Inheritance diagram for RandomAlien:



Public Member Functions

- RandomAlien (QObject ∗parent=0)

7.46.1 Constructor & Destructor Documentation

7.46.1.1 RandomAlien::RandomAlien (QObject ∗ *parent* = 0) [explicit]

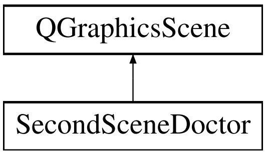
The documentation for this class was generated from the following files:

- Game2/randomalien.h
- Game2/randomalien.cpp

7.47 SecondSceneDoctor Class Reference

```
#include <secondscenedoctor.h>
```

Inheritance diagram for SecondSceneDoctor:



Public Member Functions

- SecondSceneDoctor (QObject ∗parent=0)
- void keyPressEvent (QKeyEvent ∗event)

  SecondSceneDoctor::keyPressEvent.
- void mousePressEvent (QGraphicsSceneMouseEvent ∗event)

  SecondSceneDoctor::mousePressEvent.
- void mouseMoveEvent (QGraphicsSceneMouseEvent ∗event)

  SecondSceneDoctor::mouseMoveEvent.
- void changeScene ()

  SecondSceneDoctor::changeScene.
- void showResult ()

  SecondSceneDoctor::showResult.

7.47.1 Constructor & Destructor Documentation

7.47.1.1 SecondSceneDoctor::SecondSceneDoctor (QObject ∗ *parent* = 0) [explicit]

7.47.2 Member Function Documentation

7.47.2.1 void SecondSceneDoctor::changeScene ( )

SecondSceneDoctor::changeScene.

Gets the view that is showing the current scene.

Sets back the scene to the parent of the current scene, which is the main map.

7.47.2.2 void SecondSceneDoctor::keyPressEvent (QKeyEvent ∗ *event*)

SecondSceneDoctor::keyPressEvent.

Parameters

event	

Changes the text of the upper box

enterState sets result of pressing enter key.

If it is 0, pressing enter shows the options.

After showing the options, enterState is set to 1.

If enterState is 1, pressing enter key has no effect.

If enterState is 2, this means the scene is showing the result of the player's choice, and pressing enter switches back to main map.

7.47.2.3 void SecondSceneDoctor::mouseMoveEvent (QGraphicsSceneMouseEvent ∗ *event*)

SecondSceneDoctor::mouseMoveEvent.

Parameters

event	

If clickState is not yet set to 1, (options are not shown), moving mouse has no effect on images.

When clickState is 1, moving mouse over brother or stranger changes their pixmap to make them glow.

7.47.2.4 void SecondSceneDoctor::mousePressEvent (QGraphicsSceneMouseEvent ∗ *event*)

SecondSceneDoctor::mousePressEvent.

Parameters

event	

If player clicks on one option, values are updated and results of option choosed is shown.

7.47.2.5 void SecondSceneDoctor::showResult ( )

SecondSceneDoctor::showResult.

Removes unwanted items.

Shows result depending on value of response.

enterState is set to 2 so that enter key is disabled after this

Removes character that was not chosen.

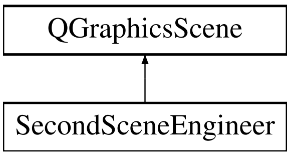
The documentation for this class was generated from the following files:

- Game2/Doctor/secondscenedoctor.h
- Game2/Doctor/secondscenedoctor.cpp

7.48 SecondSceneEngineer Class Reference

```
#include <secondsceneengineer.h>
```

Inheritance diagram for SecondSceneEngineer:



Public Member Functions

- SecondSceneEngineer (QObject ∗parent=0)
- void addAliens ()

  SecondSceneEngineer::addAliens.
- void keyPressEvent (QKeyEvent ∗event)

  SecondSceneEngineer::keyPressEvent.
- void mousePressEvent (QGraphicsSceneMouseEvent ∗event)

  SecondSceneEngineer::mousePressEvent.
- void changeScene ()

  SecondSceneEngineer::changeScene.

7.48.1 Constructor & Destructor Documentation

7.48.1.1 SecondSceneEngineer::SecondSceneEngineer (QObject ∗ *parent* = 0) [explicit]

7.48.2 Member Function Documentation

7.48.2.1 void SecondSceneEngineer::addAliens ( )

SecondSceneEngineer::addAliens.

Creates 3 aliens randomly at specific positions.

7.48.2.2 void SecondSceneEngineer::changeScene ( )

SecondSceneEngineer::changeScene.

Gets the view that is showing the current scene.

Sets back the scene to the parent of the current scene, which is the main map.

7.48.2.3 void SecondSceneEngineer::keyPressEvent (QKeyEvent ∗ *event*)

SecondSceneEngineer::keyPressEvent.

Parameters

event	

Changes the text of the upper box

enterState sets result of pressing enter key.

If it is 0, pressing enter shows the options.

After showing the options, enterState is set to 1.

If enterState is 1, pressing enter key has no effect.

7.48.2.4 void SecondSceneEngineer::mousePressEvent (QGraphicsSceneMouseEvent ∗ *event*)

SecondSceneEngineer::mousePressEvent.

Parameters

event	

If player clicks on one option, values are updated and scene is switched back to the main map.

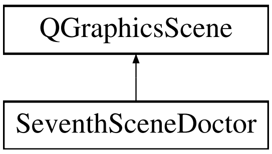
The documentation for this class was generated from the following files:

- Game2/Engineer/secondsceneengineer.h
- Game2/Engineer/secondsceneengineer.cpp

7.49 SeventhSceneDoctor Class Reference

```
#include <seventhscenedoctor.h>
```

Inheritance diagram for SeventhSceneDoctor:



Public Member Functions

- SeventhSceneDoctor (QObject ∗parent=0)
- void keyPressEvent (QKeyEvent ∗event)

  SeventhSceneDoctor::keyPressEvent.
- void mousePressEvent (QGraphicsSceneMouseEvent ∗event)

  SeventhSceneDoctor::mousePressEvent.
- void changeScene ()

  SeventhSceneDoctor::changeScene.
- void secondScenario ()

  SeventhSceneDoctor::secondScenario.

7.49.1 Constructor & Destructor Documentation

7.49.1.1 SeventhSceneDoctor::SeventhSceneDoctor (QObject ∗ *parent* = 0) [explicit]

7.49.2 Member Function Documentation

7.49.2.1 void SeventhSceneDoctor::changeScene ( )

SeventhSceneDoctor::changeScene.

Gets the view that is showing the current scene.

Sets back the scene to the parent of the current scene, which is the main map.

7.49.2.2 void SeventhSceneDoctor::keyPressEvent (QKeyEvent ∗ *event*)

SeventhSceneDoctor::keyPressEvent.

Parameters

event	

Changes the text of the upper box

enterState sets result of pressing enter key.

If it is 0, pressing enter shows the options.

After showing the options, enterState is set to 1.

If enterState is 1, pressing enter key has no effect.

clickState is set to 1 to allow player to click on an option in first scenario.

If enterState is 2, pressing enter shows options in second scenario.

If enterState is 3, pressing enter has no effect.

7.49.2.3 void SeventhSceneDoctor::mousePressEvent (QGraphicsSceneMouseEvent ∗ *event*)

SeventhSceneDoctor::mousePressEvent.

Parameters

event	

Before clickState is set to 1, clicking on screen has no effect.

If player clicks on one option, values are updated and scene is switched to second scenario.

If clickState is 2, player can click on option of second scenario.

After this, values are updated and scene is switched back to main map.

7.49.2.4 void SeventhSceneDoctor::secondScenario ( )

SeventhSceneDoctor::secondScenario.

Removes unwanted items.

Adds graphics for second scenario.

enterState is set to 2 so that enter key shows options of second scenario.

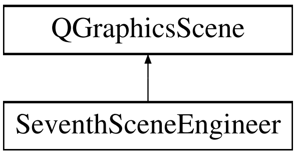
The documentation for this class was generated from the following files:

- Game2/Doctor/seventhscenedoctor.h
- Game2/Doctor/seventhscenedoctor.cpp

7.50 SeventhSceneEngineer Class Reference

```
#include <seventhsceneengineer.h>
```

Inheritance diagram for SeventhSceneEngineer:



Public Slots

- void updateScene ()

  SeventhSceneEngineer::updateScene.

Public Member Functions

- SeventhSceneEngineer (QObject ∗parent=0)
- void keyPressEvent (QKeyEvent ∗event)

  SeventhSceneEngineer::keyPressEvent.
- void mousePressEvent (QGraphicsSceneMouseEvent ∗event)

  SeventhSceneEngineer::mousePressEvent.
- void changeScene ()

  SeventhSceneEngineer::changeScene.
- void showResult ()

  SeventhSceneEngineer::showResult.

7.50.1 Constructor & Destructor Documentation

7.50.1.1 SeventhSceneEngineer::SeventhSceneEngineer (QObject ∗ *parent* = 0) [explicit]

7.50.2 Member Function Documentation

7.50.2.1 void SeventhSceneEngineer::changeScene ( )

SeventhSceneEngineer::changeScene.

Gets the view that is showing the current scene.

Sets back the scene to the parent of the current scene, which is the main map.

7.50.2.2 void SeventhSceneEngineer::keyPressEvent (QKeyEvent ∗ *event*)

SeventhSceneEngineer::keyPressEvent.

Parameters

event	

Changes the text of the upper box

enterState sets result of pressing enter key.

If it is 0, pressing enter shows the options.

After showing the options, enterState is set to 1.

If enterState is 1, pressing enter key has no effect.

If enterState is 2, this means the scene is showing the result of the player's choice, and pressing enter switches back to main map.

7.50.2.3 void SeventhSceneEngineer::mousePressEvent (QGraphicsSceneMouseEvent ∗ *event*)

SeventhSceneEngineer::mousePressEvent.

Parameters

event	

If player clicks on one option, values are updated and results of option choosed is shown.

7.50.2.4 void SeventhSceneEngineer::showResult ( )

SeventhSceneEngineer::showResult.

Removes unwanted items.

Shows result depending on value of response.

enterState is set to 2 so that enter key is disabled after this

7.50.2.5 void SeventhSceneEngineer::updateScene ( ) [slot]

SeventhSceneEngineer::updateScene.

Function called every second.

Updates size of fire image by scaling it.

After time has passed, scene is switched to show the result.

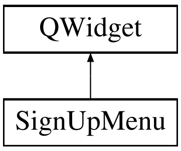
The documentation for this class was generated from the following files:

- Game2/Engineer/seventhsceneengineer.h
- Game2/Engineer/seventhsceneengineer.cpp

7.51 SignUpMenu Class Reference

```
#include <signupmenu.h>
```

Inheritance diagram for SignUpMenu:



Public Member Functions

- SignUpMenu (QWidget ∗parent=0)
- ∼SignUpMenu ()
- bool validateEmail (QString email)

  SignUpMenu::validateEmail Checks if user email is valid.
- void setStackedWidget (QStackedWidget ∗stack)

  SignUpMenu::setSackedWidget.
- void mousePressEvent (QMouseEvent ∗event)

  SignUpMenu::mousePressEvent check if user presses on sign up with facebook.

7.51.1 Constructor & Destructor Documentation

7.51.1.1 SignUpMenu::SignUpMenu (QWidget ∗ *parent* = 0) [explicit]

7.51.1.2 SignUpMenu::∼SignUpMenu ( )

7.51.2 Member Function Documentation

7.51.2.1 void SignUpMenu::mousePressEvent (QMouseEvent ∗ *event*)

SignUpMenu::mousePressEvent check if user presses on sign up with facebook.

Parameters

event	

7.51.2.2 void SignUpMenu::setStackedWidget (QStackedWidget ∗ *stack*)

SignUpMenu::setSackedWidget.

Parameters

stack	

7.51.2.3 bool SignUpMenu::validateEmail (QString *email*)

SignUpMenu::validateEmail Checks if user email is valid.

Parameters

email	

Returns

true if email is valid or false if it is not valid

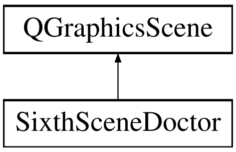
The documentation for this class was generated from the following files:

- signupmenu.h
- signupmenu.cpp

7.52 SixthSceneDoctor Class Reference

`#include <sixthscenedoctor.h>`

Inheritance diagram for SixthSceneDoctor:



Public Slots

- void rotatePhone ()

Public Member Functions

- SixthSceneDoctor (QObject ∗parent=0)
- void keyPressEvent (QKeyEvent ∗event)
- void mousePressEvent (QGraphicsSceneMouseEvent ∗event)
- void changeScene ()
- void showResult ()

7.52.1 Constructor & Destructor Documentation

7.52.1.1 SixthSceneDoctor::SixthSceneDoctor (QObject ∗ *parent* = 0) [explicit]

7.52.2 Member Function Documentation

7.52.2.1 void SixthSceneDoctor::changeScene ( )

7.52.2.2 void SixthSceneDoctor::keyPressEvent (QKeyEvent ∗ *event*)

7.52.2.3 void SixthSceneDoctor::mousePressEvent (QGraphicsSceneMouseEvent ∗ *event*)

7.52.2.4 void SixthSceneDoctor::rotatePhone ( ) [slot]

7.52.2.5 void SixthSceneDoctor::showResult ( )

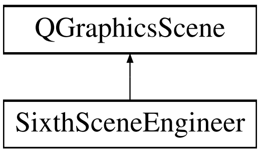
The documentation for this class was generated from the following files:

- Game2/Doctor/sixthscenedoctor.h
- Game2/Doctor/sixthscenedoctor.cpp

7.53 SixthSceneEngineer Class Reference

```
#include <sixthsceneengineer.h>
```

Inheritance diagram for SixthSceneEngineer:



Public Member Functions

- SixthSceneEngineer (QObject ∗parent=0)
- void keyPressEvent (QKeyEvent ∗event)

  SixthSceneEngineer::keyPressEvent.
- void mousePressEvent (QGraphicsSceneMouseEvent ∗event)

  SixthSceneEngineer::mousePressEvent.
- void changeScene ()

  SixthSceneEngineer::changeScene.
- void showResult ()

  SixthSceneEngineer::showResult.

7.53.1 Constructor & Destructor Documentation

7.53.1.1 SixthSceneEngineer::SixthSceneEngineer (QObject ∗ *parent* = 0) [explicit]

7.53.2 Member Function Documentation

7.53.2.1 void SixthSceneEngineer::changeScene ( )

SixthSceneEngineer::changeScene.

Gets the view that is showing the current scene.

Sets back the scene to the parent of the current scene, which is the main map.

7.53.2.2 void SixthSceneEngineer::keyPressEvent (QKeyEvent ∗ *event*)

SixthSceneEngineer::keyPressEvent.

Parameters

event	

Changes the text of the upper box

enterState sets result of pressing enter key.

If it is 0, pressing enter shows the options.

After showing the options, enterState is set to 1.

If enterState is 1, pressing enter key has no effect.

If enterState is 2, this means the scene is showing the result of the player's choice, and pressing enter switches back to main map.

7.53.2.3 void SixthSceneEngineer::mousePressEvent (QGraphicsSceneMouseEvent ∗ *event*)

SixthSceneEngineer::mousePressEvent.

Parameters

event	

If player clicks on one option, values are updated and results of option choosed is shown.

7.53.2.4 void SixthSceneEngineer::showResult ( )

SixthSceneEngineer::showResult.

Removes unwanted items.

Shows result depending on value of response.

enterState is set to 2 so that enter key is disabled after this

Updates moneyGlobal, which is global,depending on response.

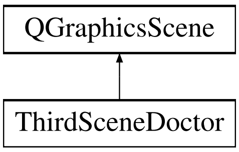
The documentation for this class was generated from the following files:

- Game2/Engineer/sixthsceneengineer.h
- Game2/Engineer/sixthsceneengineer.cpp

7.54 ThirdSceneDoctor Class Reference

```
#include <thirdscenedoctor.h>
```

Inheritance diagram for ThirdSceneDoctor:



Public Member Functions

- ThirdSceneDoctor (QObject ∗parent=0)
- void keyPressEvent (QKeyEvent ∗event)
- void mousePressEvent (QGraphicsSceneMouseEvent ∗event)
- void mouseMoveEvent (QGraphicsSceneMouseEvent ∗event)
- void changeScene ()
- void showResult ()

7.54.1 Constructor & Destructor Documentation

7.54.1.1 ThirdSceneDoctor::ThirdSceneDoctor (QObject ∗ *parent* = 0) [explicit]

7.54.2 Member Function Documentation

7.54.2.1 void ThirdSceneDoctor::changeScene ( )

7.54.2.2 void ThirdSceneDoctor::keyPressEvent (QKeyEvent ∗ *event*)

7.54.2.3 void ThirdSceneDoctor::mouseMoveEvent (QGraphicsSceneMouseEvent ∗ *event*)

7.54.2.4 void ThirdSceneDoctor::mousePressEvent (QGraphicsSceneMouseEvent ∗ *event*)

7.54.2.5 void ThirdSceneDoctor::showResult ( )

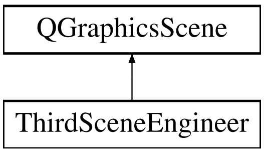
The documentation for this class was generated from the following files:

- Game2/Doctor/thirdscenedoctor.h
- Game2/Doctor/thirdscenedoctor.cpp

7.55 ThirdSceneEngineer Class Reference

```
#include <thirdsceneengineer.h>
```

Inheritance diagram for ThirdSceneEngineer:



Public Slots

- void rotatePhone ()
- void updateSeconds ()

  ThirdSceneEngineer::updateSeconds.

Public Member Functions

- ThirdSceneEngineer (QObject ∗parent=0)
- void keyPressEvent (QKeyEvent ∗event)

  ThirdSceneEngineer::keyPressEvent.
- void mousePressEvent (QGraphicsSceneMouseEvent ∗event)

  ThirdSceneEngineer::mousePressEvent.
- void changeScene ()

  ThirdSceneEngineer::changeScene.
- void showResult ()

  ThirdSceneEngineer::showResult.

7.55.1 Constructor & Destructor Documentation

7.55.1.1 ThirdSceneEngineer::ThirdSceneEngineer (QObject ∗ *parent* = 0) [explicit]

7.55.2 Member Function Documentation

7.55.2.1 void ThirdSceneEngineer::changeScene ( )

ThirdSceneEngineer::changeScene.

7.55.2.2 void ThirdSceneEngineer::keyPressEvent (QKeyEvent ∗ *event*)

ThirdSceneEngineer::keyPressEvent.

Parameters

event	

Changes the text of the upper box

enterState sets result of pressing enter key.

If it is 0, pressing enter shows the options.

After showing the options, enterState is set to 1.

If enterState is 1, pressing enter key has no effect.

If enterState is 2, this means the scene is showing the result of the player's choice, and pressing enter switches back to main map.

7.55.2.3 void ThirdSceneEngineer::mousePressEvent (QGraphicsSceneMouseEvent ∗ *event*)

ThirdSceneEngineer::mousePressEvent.

Parameters

event	

If player clicks on one option, values are updated and results of option choosed is shown.

7.55.2.4 void ThirdSceneEngineer::rotatePhone ( ) [slot]

7.55.2.5 void ThirdSceneEngineer::showResult ( )

ThirdSceneEngineer::showResult.

Removes unwanted items.

Shows result depending on value of response.

enterState is set to 2 so that enter key is disabled after this

7.55.2.6 void ThirdSceneEngineer::updateSeconds ( ) [slot]

ThirdSceneEngineer::updateSeconds.

countSeconds counts the number of times this function has been called.

When countSeconds becomes negative, counter stops and displays the result.

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

- Game2/Engineer/thirdsceneengineer.h
- Game2/Engineer/thirdsceneengineer.cpp

7.56 User Class Reference

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

Public Member Functions

- User ()
- bool signUp (QString firstName, QString lastName, QString email, QString username, QString password, QString age, QString gender)

  User::signUp Signs up the user and save his information in txt file.
- bool login (QString username, QString password)

  User::login Logs in the user and loads his information.
- bool exists (QString email, QString username)

  User::exists Checks if the user exists or username taken.
- void addGameScore (int game, int score)
- QStringList getGame1Scores ()
- QStringList getGame2Scores ()
- QStringList getGame3Scores ()
- int averageGlobalScore (int game)

7.56.1 Constructor & Destructor Documentation

7.56.1.1 User::User ( ) [explicit]

7.56.2 Member Function Documentation

7.56.2.1 void User::addGameScore (int *game*, int *score*)

7.56.2.2 int User::averageGlobalScore (int *game*)

7.56.2.3 bool User::exists (QString *email*, QString *username*)

User::exists Checks if the user exists or username taken.

Parameters

email	If email is "any" then checking if username taken
username	

Returns

7.56.2.4 QStringList User::getGame1Scores ( )

7.56.2.5 QStringList User::getGame2Scores ( )

7.56.2.6 QStringList User::getGame3Scores ( )

7.56.2.7 bool User::login (QString *username*, QString *password*)

User::login Logs in the user and loads his information.

Parameters

username	
password	

Returns

7.56.2.8 bool User::signUp (QString *firstName*, QString *lastName*, QString *email*, QString *username*, QString *password*, QString *age*, QString *gender*)

User::signUp Signs up the user and save his information in txt file.

Parameters

firstName	
lastName	
email	
username	
password	
age	
gender	

Returns

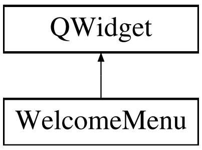
The documentation for this class was generated from the following files:

- user.h
- user.cpp

7.57 WelcomeMenu Class Reference

```
#include <welcomemenu.h>
```

Inheritance diagram for WelcomeMenu:



Public Member Functions

- WelcomeMenu (QWidget ∗parent=0)
- ∼WelcomeMenu ()
- void setStackedWidget (QStackedWidget ∗stack)

7.57.1 Constructor & Destructor Documentation

7.57.1.1 WelcomeMenu::WelcomeMenu (QWidget ∗ *parent* = 0) `[explicit]`

7.57.1.2 WelcomeMenu::∼WelcomeMenu ( )

7.57.2 Member Function Documentation

7.57.2.1 void WelcomeMenu::setStackedWidget (QStackedWidget ∗ *stack*)

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

- welcomemenu.h
- welcomemenu.cpp

Chapter 8

File Documentation

8.1 Game1/character.cpp File Reference

Character Functionalities.

```
#include "character.h"
```

8.1.1 Detailed Description

Character Functionalities. Setting the image of the character according to level of difficulty.

Implementing movement of the character.

8.2 Game1/character.h File Reference

```
#include <QObject>
#include <QGraphicsRectItem>
#include <QGraphicsItem>
#include <QKeyEvent>
#include <QGraphicsPixmapItem>
#include <QStringList>
#include <QList>
#include "obstaclegroup.h"
#include "global.h"
```

Classes

- class Character

8.3 Game1/game1graphicsview.cpp File Reference

```
#include "game1graphicsview.h"
```

8.4 Game1/game1graphicsview.h File Reference

```
#include <QGraphicsView>
#include <QWidget>
#include <QtWidgets>
#include "game1scene.h"
```

Classes

- class Game1GraphicsView

8.5 Game1/game1menu.cpp File Reference

Main Menu of the Game1.

```
#include "game1menu.h"
#include "ui_game1menu.h"
```

8.5.1 Detailed Description

Main Menu of the Game1. Choosing level of difficulty.

Starting game.

8.6 Game1/game1menu.h File Reference

```
#include <QWidget>
#include <QtWidgets>
#include "game1graphicsview.h"
#include "user.h"
#include "../globalindices.h"
```

Classes

- class Game1Menu

Namespaces

- Ui

8.7 Game1/game1scene.cpp File Reference

Adding and managing scene items.

```
#include "game1scene.h"
#include "character.h"
#include "obstaclegroup.h"
#include <QGraphicsItem>
#include <QGraphicsScene>
#include <QGraphicsRectItem>
#include <QGraphicsView>
#include <QLabel>
#include <QGraphicsLinearLayout>
#include <QTextDocument>
```

8.7.1 Detailed Description

Adding and managing scene items. Setting the background, adding the obstacles and updating the number of lives.

8.8 Game1/game1scene.h File Reference

```
#include <QGraphicsScene>
#include "character.h"
#include <stdlib.h>
#include <QObject>
#include <QImage>
#include <QBrush>
#include "obstacle.h"
#include <time.h>
#include <QStringList>
#include "obstaclegroup.h"
#include "game1score.h"
#include <QWidget>
#include <QtWidgets>
#include <QGraphicsSceneMouseEvent>
#include "../globalindices.h"
```

Classes

- class Game1Scene

8.9 Game1/game1score.cpp File Reference

```
#include "game1score.h"
#include "ui_game1score.h"
#include <QTextEdit>
```

8.10 Game1/game1score.h File Reference

```
#include <QWidget>
#include <QtWidgets>
#include <QStringList>
#include <string>
#include "../globalindices.h"
```

Classes

- class Game1Score

Namespaces

- Ui

8.11 Game1/global.cpp File Reference

```
#include "global.h"
```

Variables

- QStringList globalValues
- QStringList globalVices

8.11.1 Variable Documentation

8.11.1.1 QStringList globalValues

Initial value:

```
= QStringList()<<"CARING"<<"COMMITMENT"<<"COMPASSION"<<"COURTESY"<<"HONESTY"<<"INTEGRITY"<<"OPTIMISIM"
            <<"KINDNESS"<<"OPENNESS"<<"WISDOM"<<"PEACE"<<"FORGVING"
```

8.11.1.2 QStringList globalVices

Initial value:

```
= QStringList()<<"JEALOUS"<<"COWARDICE"<<"ENVY"<<"SHYNESS"<<"ANGER"<<"INSENSIBILITY"<<"MEANNESS"<<
           "ARROGANCE"<<"BRAGGING"<<"WEAKNESS"<<"GREED"<<"DISLOYALTY"<<"RECKLESNESS"<<"EXTREMESIM"
           <<"DELUSION"<<"HOSTILITY"<<"LAZINESS"<<"RACISM"<<"STUPIDITY"<<"EGOISM"
```

8.12 Game1/global.h File Reference

```
#include <QStringList>
```

Variables

- QStringList globalValues
- QStringList globalVices

8.12.1 Variable Documentation

8.12.1.1 QStringList globalValues

8.12.1.2 QStringList globalVices

8.13 Game1/obstacle.cpp File Reference

Creating obstacles.

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

8.13.1 Detailed Description

Creating obstacles. Determing type of obstacle (space shuttle or flying saucer) randomly.

8.14 Game1/obstacle.h File Reference

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

Classes

- class Obstacle

8.15 Game1/obstaclegroup.cpp File Reference

Groups a value with an obstacle.

```
#include "obstaclegroup.h"
```

8.15.1 Detailed Description

Groups a value with an obstacle. Creating lists of values and vices.

Grouping values with obstacles.

Moving groups.

8.16 Game1/obstaclegroup.h File Reference

```
#include <QGraphicsItemGroup>
#include "obstacle.h"
#include <QGraphicsTextItem>
#include <QString>
#include <time.h>
#include "stdlib.h"
#include <fstream>
#include <iostream>
#include "global.h"
```

Classes

- class ObstacleGroup

8.17 Game2/Doctor/doctorgraphicsview.cpp File Reference

```
#include "doctorgraphicsview.h"
```

8.18 Game2/Doctor/doctorgraphicsview.h File Reference

```
#include <QGraphicsView>
#include <QWidget>
#include <QtWidgets>
#include "doctorscene.h"
```

Classes

- class DoctorGraphicsView

8.19 Game2/Doctor/doctorscene.cpp File Reference

Main scene that shows progress of character.

```
#include "doctorscene.h"
```

8.19.1 Detailed Description

Main scene that shows progress of character. Shows the progress of the character on map.

Shows current amount of money.

Allows character to pass from scenario to scenario.

When all scenarios completed, allows to switch to score menu.

8.20 Game2/Doctor/doctorscene.h File Reference

```
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QGraphicsItem>
#include <QGraphicsTextItem>
#include <QWidget>
#include <QtWidgets>
#include <QString>
#include <QList>
#include <QSound>
#include <QTime>
#include <stdlib.h>
#include "../global2.h"
#include "../game2score.h"
#include "firstscenedoctor.h"
#include "secondscenedoctor.h"
#include "thirdscenedoctor.h"
#include "fourthscenedoctor.h"
#include "fifthscenedoctor.h"
#include "sixthscenedoctor.h"
#include "seventhscenedoctor.h"
```

Classes

- class DoctorScene

8.21 Game2/Doctor/fifthscenedoctor.cpp File Reference

```
#include "fifthscenedoctor.h"
```

8.22 Game2/Doctor/fifthscenedoctor.h File Reference

```
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QWidget>
#include <QtWidgets>
#include <QObject>
#include <QSound>
#include "../global2.h"
```

Classes

- class FifthSceneDoctor

8.23 Game2/Doctor/firstscenedoctor.cpp File Reference

First doctor scenario.

```
#include "firstscenedoctor.h"
```

8.23.1 Detailed Description

First doctor scenario.

8.24 Game2/Doctor/firstscenedoctor.h File Reference

```
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QWidget>
#include <QtWidgets>
#include <QObject>
#include <QSound>
#include "../global2.h"
#include "../randomalien.h"
```

Classes

- class FirstSceneDoctor

8.25 Game2/Doctor/fourthscenedoctor.cpp File Reference

```
#include "fourthscenedoctor.h"
```

8.26 Game2/Doctor/fourthscenedoctor.h File Reference

```
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QWidget>
#include <QtWidgets>
#include <QObject>
#include <QSound>
#include "../global2.h"
```

Classes

- class FourthSceneDoctor

8.27 Game2/Doctor/secondscenedoctor.cpp File Reference

Second doctor scenario.

```
#include "secondscenedoctor.h"
```

8.27.1 Detailed Description

Second doctor scenario.

8.28 Game2/Doctor/secondscenedoctor.h File Reference

```
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QWidget>
#include <QtWidgets>
#include <QObject>
#include <QSound>
#include "../global2.h"
```

Classes

- class SecondSceneDoctor

8.29 Game2/Doctor/seventhscenedoctor.cpp File Reference

Seventh doctor scenario.

```
#include "seventhscenedoctor.h"
```

8.29.1 Detailed Description

Seventh doctor scenario.

8.30 Game2/Doctor/seventhscenedoctor.h File Reference

```
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QWidget>
#include <QtWidgets>
#include <QObject>
#include <QSound>
#include "../global2.h"
```

Classes

- class SeventhSceneDoctor

8.31 Game2/Doctor/sixthscenedoctor.cpp File Reference

```
#include "sixthscenedoctor.h"
```

8.32 Game2/Doctor/sixthscenedoctor.h File Reference

```
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QWidget>
#include <QtWidgets>
#include <QObject>
#include <QSound>
#include "../global2.h"
```

Classes

- class SixthSceneDoctor

8.33 Game2/Doctor/thirdscenedoctor.cpp File Reference

Doctor third scenario.

```
#include "thirdscenedoctor.h"
```

8.33.1 Detailed Description

Doctor third scenario.

8.34 Game2/Doctor/thirdscenedoctor.h File Reference

```
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QWidget>
#include <QtWidgets>
#include <QObject>
#include <QSound>
#include "../global2.h"
```

Classes

- class ThirdSceneDoctor

8.35 Game2/Engineer/eigthsceneengineer.cpp File Reference

```
#include "eigthsceneengineer.h"
```

8.36 Game2/Engineer/eigthsceneengineer.h File Reference

```
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QWidget>
#include <QtWidgets>
#include <QObject>
#include <QSound>
#include "../global2.h"
```

Classes

- class EigthSceneEngineer

8.37 Game2/Engineer/engineergraphicsview.cpp File Reference

```
#include "engineergraphicsview.h"
```

8.38 Game2/Engineer/engineergraphicsview.h File Reference

```
#include <QGraphicsView>
#include <QWidget>
#include <QtWidgets>
#include "engineerscene.h"
#include "user.h"
```

Classes

- class EngineerGraphicsView

8.39 Game2/Engineer/engineerscene.cpp File Reference

Main scene that shows progress of character.

```
#include "engineerscene.h"
```

8.39.1 Detailed Description

Main scene that shows progress of character. Shows the progress of the character on map.

Shows current amount of money.

Allows character to pass from scenario to scenario.

When all scenarios completed, allows to switch to score menu.

8.40 Game2/Engineer/engineerscene.h File Reference

```
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QGraphicsItem>
#include <QGraphicsTextItem>
#include <QWidget>
#include <QtWidgets>
#include <QString>
#include <QList>
#include <QSound>
#include <QTime>
#include <stdlib.h>
#include "../global2.h"
#include "../game2score.h"
#include "firstsceneengineer.h"
#include "secondsceneengineer.h"
#include "thirdsceneengineer.h"
#include "fourthsceneengineer.h"
#include "fifthsceneengineer.h"
#include "sixthsceneengineer.h"
#include "seventhsceneengineer.h"
#include "eigthsceneengineer.h"
#include "../../globalindices.h"
```

Classes

- class EngineerScene

8.41 Game2/Engineer/fifthsceneengineer.cpp File Reference

Fifth engineer scenario.

```
#include "fifthsceneengineer.h"
```

8.41.1 Detailed Description

Fifth engineer scenario.

8.42 Game2/Engineer/fifthsceneengineer.h File Reference

```
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QWidget>
#include <QtWidgets>
#include <QObject>
#include <QSound>
#include "../global2.h"
```

Classes

- class FifthSceneEngineer

8.43 Game2/Engineer/firstsceneengineer.cpp File Reference

First engineer scenario.

```
#include "firstsceneengineer.h"
```

8.43.1 Detailed Description

First engineer scenario.

8.44 Game2/Engineer/firstsceneengineer.h File Reference

```
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QWidget>
#include <QtWidgets>
#include <QObject>
#include <QSound>
#include "../global2.h"
#include "../randomalien.h"
```

Classes

- class FirstSceneEngineer

8.45 Game2/Engineer/fourthsceneengineer.cpp File Reference

Fourth engineer scenario.

```
#include "fourthsceneengineer.h"
```

8.45.1 Detailed Description

Fourth engineer scenario.

8.46 Game2/Engineer/fourthsceneengineer.h File Reference

```
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QWidget>
#include <QtWidgets>
#include <QObject>
#include <QSound>
#include "../global2.h"
```

Classes

- class FourthSceneEngineer

8.47 Game2/Engineer/secondsceneengineer.cpp File Reference

Second engineer scenario.

```
#include "secondsceneengineer.h"
```

8.47.1 Detailed Description

Second engineer scenario.

8.48 Game2/Engineer/secondsceneengineer.h File Reference

```
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QWidget>
#include <QtWidgets>
#include <QObject>
#include <QSound>
#include "../global2.h"
#include "../randomalien.h"
```

Classes

- class SecondSceneEngineer

8.49 Game2/Engineer/seventhsceneengineer.cpp File Reference

Seventh engineer scenario.

```
#include "seventhsceneengineer.h"
```

8.49.1 Detailed Description

Seventh engineer scenario.

8.50 Game2/Engineer/seventhsceneengineer.h File Reference

```
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QWidget>
#include <QtWidgets>
#include <QObject>
#include <QSound>
#include "../global2.h"
```

Classes

- class SeventhSceneEngineer

8.51 Game2/Engineer/sixthsceneengineer.cpp File Reference

Sixth engineer scenario.

```
#include "sixthsceneengineer.h"
```

8.51.1 Detailed Description

Sixth engineer scenario.

8.52 Game2/Engineer/sixthsceneengineer.h File Reference

```
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QWidget>
#include <QtWidgets>
#include <QObject>
#include <QSound>
#include "../global2.h"
```

Classes

- class SixthSceneEngineer

8.53 Game2/Engineer/thirdsceneengineer.cpp File Reference

Third engineer scenario.

```
#include "thirdsceneengineer.h"
```

8.53.1 Detailed Description

Third engineer scenario.

8.54 Game2/Engineer/thirdsceneengineer.h File Reference

```
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QWidget>
#include <QtWidgets>
#include <QObject>
#include <QSound>
#include "../global2.h"
```

Classes

- class ThirdSceneEngineer

8.55 Game2/game2menu.cpp File Reference

```
#include "game2menu.h"
#include "ui_game2menu.h"
```

8.56 Game2/game2menu.h File Reference

```
#include <QWidget>
#include <QtWidgets>
#include "Engineer/engineergraphicsview.h"
#include "Doctor/doctorgraphicsview.h"
#include "user.h"
#include "../globalindices.h"
```

Classes

- class Game2Menu

Namespaces

- Ui

8.57 Game2/game2score.cpp File Reference

```
#include "game2score.h"
#include "ui_game2score.h"
```

8.58 Game2/game2score.h File Reference

```
#include <QWidget>
#include <QtWidgets>
#include "global2.h"
#include "../globalindices.h"
```

Classes

- class Game2Score

Namespaces

- Ui

8.59 Game2/global2.cpp File Reference

```
#include "global2.h"
```

Variables

- int stateOfEngineer = 0
- int stateOfDoctor = 0
- int moneyGlobal = 10000
- int friendship
- int compliance
- int honesty
- int forgivingness
- int helping
- int family
- int courage

8.59.1 Variable Documentation

8.59.1.1 int compliance

8.59.1.2 int courage

8.59.1.3 int family

8.59.1.4 int forgivingness

8.59.1.5 int friendship

8.59.1.6 int helping

8.59.1.7 int honesty

8.59.1.8 int moneyGlobal = 10000

8.59.1.9 int stateOfDoctor = 0

8.59.1.10 int stateOfEngineer = 0

8.60 Game2/global2.h File Reference

```
#include <QStringList>
```

Variables

- int stateOfEngineer
- int stateOfDoctor
- int moneyGlobal
- int friendship
- int compliance
- int honesty
- int forgivingness
- int helping
- int family
- int courage

8.60.1 Variable Documentation

8.60.1.1 int compliance

8.60.1.2 int courage

8.60.1.3 int family

8.60.1.4 int forgivingness

8.60.1.5 int friendship

8.60.1.6 int helping

8.60.1.7 int honesty

8.60.1.8 int moneyGlobal

8.60.1.9 int stateOfDoctor

8.60.1.10 int stateOfEngineer

8.61 Game2/randomalien.cpp File Reference

```
#include "randomalien.h"
```

8.62 Game2/randomalien.h File Reference

```
#include <QObject>
#include <QGraphicsItem>
#include <QGraphicsRectItem>
```

Classes

- class RandomAlien

8.63 Game3/game3graphicsview.cpp File Reference

```
#include "game3graphicsview.h"
```

8.64 Game3/game3graphicsview.h File Reference

```
#include <QGraphicsView>
#include <QWidget>
#include <QtWidgets>
#include "game3scene.h"
```

Classes

- class Game3GraphicsView

8.65 Game3/game3menu.cpp File Reference

```
#include "game3menu.h"
#include "ui_game3menu.h"
```

8.66 Game3/game3menu.h File Reference

```
#include <QWidget>
#include <QtWidgets>
#include "game3graphicsview.h"
#include "../globalindices.h"
```

Classes

- class Game3Menu

Namespaces

- Ui

8.67 Game3/game3scene.cpp File Reference

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

8.68 Game3/game3scene.h File Reference

```
#include <QGraphicsScene>
#include <QList>
#include <QKeyEvent>
#include <QWidget>
#include <QtWidgets>
#include <stdlib.h>
#include <QGraphicsRectItem>
#include <QSound>
#include <QMouseEvent>
#include "player.h"
#include "global3.h"
#include "game3score.h"
#include "../globalindices.h"
```

Classes

- class Game3Scene

Macros

- #define OPEN 1
- #define CLOSED 2
- #define REMOVED 3

8.68.1 Macro Definition Documentation

8.68.1.1 #define CLOSED 2

8.68.1.2 #define OPEN 1

8.68.1.3 #define REMOVED 3

8.69 Game3/game3score.cpp File Reference

```
#include "game3score.h"
#include "ui_game3score.h"
```

8.70 Game3/game3score.h File Reference

```
#include <QWidget>
#include <QtWidgets>
#include <QStringList>
#include <string>
```

Classes

- class Game3Score

Namespaces

- Ui

8.71 Game3/global3.cpp File Reference

```
#include "global3.h"
```

Variables

- QStringList globalValues3

  globalValues3, reference list of values to check against.
- QStringList globalVices3

  globalValues3, reference list of vices to check against.

8.71.1 Variable Documentation

8.71.1.1 QStringList globalValues3

Initial value:

```
= QStringList()<<"CARING"<<"COMMITMENT"<<"COMPASSION"<<"COURTESY"<<"HONESTY"<<"INTEGRITY"<<"OPTIMISIM"
        <<"KINDNESS"<<"OPENNESS"<<"WISDOM"<<"PEACE"<<"FORGVING"
```

globalValues3, reference list of values to check against.

8.71.1.2 QStringList globalVices3

Initial value:

```
= QStringList()<<"JEALOUS"<<"COWARDICE"<<"ENVY"<<"SHYNESS"<<"ANGER"<<"INSENSIBILITY"<<"MEANNESS"<<
       "ARROGANCE"<<"BRAGGING"<<"WEAKNESS"<<"GREED"<<"DISLOYALTY"<<"RECKLESNESS"<<"EXTREMESIM"
       <<"DELUSION"<<"HOSTILITY"<<"LAZINESS"<<"RACISM"<<"STUPIDITY"<<"EGOISM"
```

globalValues3, reference list of vices to check against.

8.72 Game3/global3.h File Reference

```
#include <QStringList>
```

Variables

- QStringList globalValues3

  globalValues3, reference list of values to check against.
- QStringList globalVices3

  globalValues3, reference list of vices to check against.

8.72.1 Variable Documentation

8.72.1.1 QStringList globalValues3

globalValues3, reference list of values to check against.

8.72.1.2 QStringList globalVices3

globalValues3, reference list of vices to check against.

8.73 Game3/player.cpp File Reference

Defining functionalities of the player.

```
#include "player.h"
```

8.73.1 Detailed Description

Defining functionalities of the player.

8.74 Game3/player.h File Reference

```
#include <QObject>
#include <QGraphicsPixmapItem>
#include <QKeyEvent>
```

Classes

- class Player

8.75 globalindices.cpp File Reference

```
#include "globalindices.h"
```

Variables

- int mainIndex = 100
- int game1Index = 100
- int game2Index = 100
- int game3Index = 100
- User theUser

8.75.1 Variable Documentation

8.75.1.1 int game1Index = 100

8.75.1.2 int game2Index = 100

8.75.1.3 int game3Index = 100

8.75.1.4 int mainIndex = 100

8.75.1.5 User theUser

8.76 globalindices.h File Reference

```
#include "user.h"
```

Variables

- int mainIndex
- int game1Index
- int game2Index
- int game3Index
- User theUser

8.76.1 Variable Documentation

8.76.1.1 int game1Index

8.76.1.2 int game2Index

8.76.1.3 int game3Index

8.76.1.4 int mainIndex

8.76.1.5 User theUser

8.77 loginmenu.cpp File Reference

```
#include "loginmenu.h"
#include "ui_loginmenu.h"
```

8.78 loginmenu.h File Reference

```
#include <QWidget>
#include <QtWidgets>
#include "welcomemenu.h"
#include "mainmenu.h"
#include <cstring>
#include "user.h"
#include "globalindices.h"
```

Classes

- class LoginMenu

Namespaces

- Ui

8.79 main.cpp File Reference

```
#include "mainwindow.h"
#include <QApplication>
#include <cstdlib>
#include <ctime>
```

Functions

- int main (int argc, char ∗argv[])

8.79.1 Function Documentation

8.79.1.1 int main (int *argc*, char ∗ *argv[]*)

8.80 mainmenu.cpp File Reference

```
#include "mainmenu.h"
#include "ui_mainmenu.h"
```

8.81 mainmenu.h File Reference

```
#include <QWidget>
#include <QtWidgets>
#include "welcomemenu.h"
#include "Game1/game1menu.h"
#include "Game2/game2menu.h"
#include "Game3/game3menu.h"
#include "user.h"
#include "profile.h"
#include <cstring>
```

Classes

- class MainMenu

Namespaces

- Ui

8.82 mainwindow.cpp File Reference

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include "welcomemenu.h"
#include <cstring>
#include <QString>
```

8.83 mainwindow.h File Reference

```
#include <QMainWindow>
```

Classes

- class MainWindow

Namespaces

- Ui

8.84 OAuth/fbdemo.cpp File Reference

```
#include "fbdemo.h"
#include "o0globals.h"
#include "o0settingsstore.h"
```

Macros

- #define QENUM_NAME(o, e, v) (o::staticMetaObject.enumerator(o::staticMetaObject.indexOfEnumerator(#e)).valueToKey((v)))
- #define GRANTFLOW_STR(v) QString(QENUM_NAME(O2, GrantFlow, v))

Variables

- const char FB_APP_KEY [] = "571523936513739"
- const char FB_APP_SECRET [] = "e600cd4fcfd8b3015e3f38543785c987"
- const char FB_REQUEST_URL [] = "https://www.facebook.com/dialog/oauth"
- const char FB_DEBUG_TOKEN [] = "https://graph.facebook.com/me?fields=id&access_token=%1"
- const char FB_REQUEST_SCOPE [] = "email,user_birthday"
- const char FbGraph [] = "https://graph.facebook.com/me?fields=first_name,last_name,email,birthday,gender"
- const int localPort = 8000

8.84.1 Macro Definition Documentation

8.84.1.1 #define GRANTFLOW_STR(*v*) QString(QENUM_NAME(O2, GrantFlow, v))

8.84.1.2 #define QENUM_NAME(*o,* *e,* *v*) (o::staticMetaObject.enumerator(o::staticMetaObject.indexOfEnumerator(#e)).valueToKey((v)))

8.84.2 Variable Documentation

8.84.2.1 const char FB_APP_KEY[ ] = "571523936513739"

8.84.2.2 const char FB_APP_SECRET[ ] = "e600cd4fcfd8b3015e3f38543785c987"

8.84.2.3 const char FB_DEBUG_TOKEN[ ] = "https://graph.facebook.com/me?fields=id&access_token=%1"

8.84.2.4 const char FB_REQUEST_SCOPE[ ] = "email,user_birthday"

8.84.2.5 const char FB_REQUEST_URL[ ] = "https://www.facebook.com/dialog/oauth"

8.84.2.6 const char FbGraph[ ] = "https://graph.facebook.com/me?fields=first_name,last_name,email,birthday,gender"

8.84.2.7 const int localPort = 8000

8.85 OAuth/fbdemo.h File Reference

```
#include <QObject>
#include <QNetworkAccessManager>
#include <QNetworkRequest>
#include <QNetworkReply>
#include <QDesktopServices>
#include <QMetaEnum>
#include <QDebug>
#include <QJsonDocument>
#include <QJsonObject>
#include <QMap>
#include <QString>
#include <QStringList>
#include <QUrl>
#include "o2facebook.h"
```

Classes

- class FBDemo

8.86 OAuth/helper.cpp File Reference

```
#include "helper.h"
```

8.87 OAuth/helper.h File Reference

```
#include <QObject>
#include <QStringList>
#include <QTimer>
#include <QDebug>
#include "fbdemo.h"
```

Classes

- class Helper

Variables

- const char OPT_OAUTH_CODE [] = "-o"
- const char OPT_VALIDATE_TOKEN [] = "-v"
- const char USAGE []

8.87.1 Variable Documentation

8.87.1.1 const char OPT_OAUTH_CODE[] = "-o"

8.87.1.2 const char OPT_VALIDATE_TOKEN[] = "-v"

8.87.1.3 const char USAGE[]

Initial value:

```
= "\n"
                      "Usage: facebookdemo [OPTION]...\n"
                      "Get OAuth2 access tokens from Facebook's OAuth service\n"
                      "\nOptions:\n"
                      "  %1\t\tLink with Facebook OAuth2 service using Authorization Code\n"
                      "  %2\t\tValidate Access Token\n"
```

8.88 OAuth/src/o0abstractstore.h File Reference

```
#include <QObject>
#include <QString>
#include "o0export.h"
```

Classes

- class O0AbstractStore

  Storage for strings.

8.89 OAuth/src/o0baseauth.cpp File Reference

```
#include <QDataStream>
#include <QDebug>
#include "o0baseauth.h"
#include "o0globals.h"
#include "o0settingsstore.h"
#include "o2replyserver.h"
```

8.90 OAuth/src/o0baseauth.h File Reference

```
#include <QByteArray>
#include <QObject>
#include <QMap>
#include <QString>
#include <QUrl>
#include <QVariantMap>
#include "o0export.h"
#include "o0abstractstore.h"
#include "o0requestparameter.h"
```

Classes

- class O0BaseAuth

  Base class of OAuth authenticators.

8.91 OAuth/src/o0export.h File Reference

Macros

- #define O0_EXPORT

8.91.1 Macro Definition Documentation

8.91.1.1 #define O0_EXPORT

8.92 OAuth/src/o0globals.h File Reference

Variables

- const char O2_ENCRYPTION_KEY [] = "12345678"

- const char O2_CALLBACK_URL [] = "http://localhost:%1/"
- const char O2_MIME_TYPE_XFORM [] = "application/x-www-form-urlencoded"
- const char O2_MIME_TYPE_JSON [] = "application/json"
- const char O2_KEY_TOKEN [] = "token.%1"
- const char O2_KEY_TOKEN_SECRET [] = "tokensecret.%1"
- const char O2_KEY_CODE [] = "code.%1"
- const char O2_KEY_EXPIRES [] = "expires.%1"
- const char O2_KEY_REFRESH_TOKEN [] = "refreshtoken.%1"
- const char O2_KEY_LINKED [] = "linked.%1"
- const char O2_KEY_EXTRA_TOKENS [] = "extratokens.%1"
- const char O2_OAUTH_CALLBACK [] = "oauth_callback"
- const char O2_OAUTH_CONSUMER_KEY [] = "oauth_consumer_key"
- const char O2_OAUTH_NONCE [] = "oauth_nonce"
- const char O2_OAUTH_SIGNATURE [] = "oauth_signature"
- const char O2_OAUTH_SIGNATURE_METHOD [] = "oauth_signature_method"
- const char O2_OAUTH_TIMESTAMP [] = "oauth_timestamp"
- const char O2_OAUTH_VERSION [] = "oauth_version"
- const char O2_OAUTH_TOKEN [] = "oauth_token"
- const char O2_OAUTH_TOKEN_SECRET [] = "oauth_token_secret"
- const char O2_OAUTH_CALLBACK_CONFIRMED [] = "oauth_callback_confirmed"
- const char O2_OAUTH_VERFIER [] = "oauth_verifier"
- const char O2_OAUTH2_RESPONSE_TYPE [] = "response_type"
- const char O2_OAUTH2_CLIENT_ID [] = "client_id"
- const char O2_OAUTH2_CLIENT_SECRET [] = "client_secret"
- const char O2_OAUTH2_USERNAME [] = "username"
- const char O2_OAUTH2_PASSWORD [] = "password"
- const char O2_OAUTH2_REDIRECT_URI [] = "redirect_uri"
- const char O2_OAUTH2_SCOPE [] = "scope"
- const char O2_OAUTH2_GRANT_TYPE_CODE [] = "code"
- const char O2_OAUTH2_GRANT_TYPE_TOKEN [] = "token"
- const char O2_OAUTH2_GRANT_TYPE_PASSWORD [] = "password"
- const char O2_OAUTH2_GRANT_TYPE [] = "grant_type"
- const char O2_OAUTH2_API_KEY [] = "api_key"
- const char O2_OAUTH2_ACCESS_TOKEN [] = "access_token"
- const char O2_OAUTH2_REFRESH_TOKEN [] = "refresh_token"
- const char O2_OAUTH2_EXPIRES_IN [] = "expires_in"
- const char O2_SIGNATURE_TYPE_HMAC_SHA1 [] = "HMAC-SHA1"
- const char O2_SIGNATURE_TYPE_PLAINTEXT [] = "PLAINTEXT"
- const char O2_AUTHORIZATION_CODE [] = "authorization_code"
- const char O2_HTTP_AUTHORIZATION_HEADER [] = "Authorization"

8.92.1 Variable Documentation

8.92.1.1 const char O2_AUTHORIZATION_CODE[] = "authorization_code"

8.92.1.2 const char O2_CALLBACK_URL[] = "http://localhost:%1/"

8.92.1.3 const char O2_ENCRYPTION_KEY[] = "12345678"

8.92.1.4 const char O2_HTTP_AUTHORIZATION_HEADER[] = "Authorization"

8.92.1.5 const char O2_KEY_CODE[] = "code.%1"

8.92.1.6 const char O2_KEY_EXPIRES[] = "expires.%1"

8.92.1.7 const char O2_KEY_EXTRA_TOKENS[ ] = "extratokens.%1"

8.92.1.8 const char O2_KEY_LINKED[ ] = "linked.%1"

8.92.1.9 const char O2_KEY_REFRESH_TOKEN[ ] = "refreshtoken.%1"

8.92.1.10 const char O2_KEY_TOKEN[ ] = "token.%1"

8.92.1.11 const char O2_KEY_TOKEN_SECRET[ ] = "tokensecret.%1"

8.92.1.12 const char O2_MIME_TYPE_JSON[ ] = "application/json"

8.92.1.13 const char O2_MIME_TYPE_XFORM[ ] = "application/x-www-form-urlencoded"

8.92.1.14 const char O2_OAUTH2_ACCESS_TOKEN[ ] = "access_token"

8.92.1.15 const char O2_OAUTH2_API_KEY[ ] = "api_key"

8.92.1.16 const char O2_OAUTH2_CLIENT_ID[ ] = "client_id"

8.92.1.17 const char O2_OAUTH2_CLIENT_SECRET[ ] = "client_secret"

8.92.1.18 const char O2_OAUTH2_EXPIRES_IN[ ] = "expires_in"

8.92.1.19 const char O2_OAUTH2_GRANT_TYPE[ ] = "grant_type"

8.92.1.20 const char O2_OAUTH2_GRANT_TYPE_CODE[ ] = "code"

8.92.1.21 const char O2_OAUTH2_GRANT_TYPE_PASSWORD[ ] = "password"

8.92.1.22 const char O2_OAUTH2_GRANT_TYPE_TOKEN[ ] = "token"

8.92.1.23 const char O2_OAUTH2_PASSWORD[ ] = "password"

8.92.1.24 const char O2_OAUTH2_REDIRECT_URI[ ] = "redirect_uri"

8.92.1.25 const char O2_OAUTH2_REFRESH_TOKEN[ ] = "refresh_token"

8.92.1.26 const char O2_OAUTH2_RESPONSE_TYPE[ ] = "response_type"

8.92.1.27 const char O2_OAUTH2_SCOPE[ ] = "scope"

8.92.1.28 const char O2_OAUTH2_USERNAME[ ] = "username"

8.92.1.29 const char O2_OAUTH_CALLBACK[ ] = "oauth_callback"

8.92.1.30 const char O2_OAUTH_CALLBACK_CONFIRMED[ ] = "oauth_callback_confirmed"

8.92.1.31 const char O2_OAUTH_CONSUMER_KEY[ ] = "oauth_consumer_key"

8.92.1.32 const char O2_OAUTH_NONCE[ ] = "oauth_nonce"

8.92.1.33 const char O2_OAUTH_SIGNATURE[ ] = "oauth_signature"

8.92.1.34 const char O2_OAUTH_SIGNATURE_METHOD[ ] = "oauth_signature_method"

8.92.1.35 const char O2_OAUTH_TIMESTAMP[ ] = "oauth_timestamp"

8.92.1.36 const char O2_OAUTH_TOKEN[ ] = "oauth_token"

8.92.1.37 const char O2_OAUTH_TOKEN_SECRET[ ] = "oauth_token_secret"

8.92.1.38 const char O2_OAUTH_VERFIER[ ] = "oauth_verifier"

8.92.1.39 const char O2_OAUTH_VERSION[ ] = "oauth_version"

8.92.1.40 const char O2_SIGNATURE_TYPE_HMAC_SHA1[ ] = "HMAC-SHA1"

8.92.1.41 const char O2_SIGNATURE_TYPE_PLAINTEXT[ ] = "PLAINTEXT"

8.93 OAuth/src/o0keychainstore.cpp File Reference

```
#include "o0keychainstore.h"
#include <QDebug>
#include <keychain.h>
#include <QtCore/QDataStream>
#include <QtCore/QBuffer>
#include <QtCore/QEventLoop>
```

8.94 OAuth/src/o0keychainstore.h File Reference

```
#include <QtCore/QMap>
#include "o0abstractstore.h"
#include <QString>
```

Classes

- class o0keyChainStore

Namespaces

- QKeychain

8.95 OAuth/src/o0requestparameter.h File Reference

```
#include "o0export.h"
```

Classes

- struct O0RequestParameter

  Request parameter (name-value pair) participating in authentication.

8.96 OAuth/src/o0settingsstore.cpp File Reference

```
#include <QCryptographicHash>
#include <QByteArray>
#include <QDebug>
#include "o0settingsstore.h"
```

8.97 OAuth/src/o0settingsstore.h File Reference

```
#include <QSettings>
#include <QString>
#include "o0baseauth.h"
#include "o0abstractstore.h"
#include "o0simplecrypt.h"
```

Classes

- class O0SettingsStore

  Persistent storage for authentication tokens, using QSettings.

8.98 OAuth/src/o0simplecrypt.h File Reference

```
#include <QString>
#include <QVector>
#include <QFlags>
#include "o0baseauth.h"
```

Classes

- class O0SimpleCrypt

  Simple encryption and decryption of strings and byte arrays.

8.99 OAuth/src/o2.cpp File Reference

```
#include <QList>
#include <QPair>
#include <QDebug>
#include <QTcpServer>
#include <QMap>
#include <QNetworkRequest>
#include <QNetworkReply>
#include <QNetworkAccessManager>
#include <QDateTime>
#include <QCryptographicHash>
#include <QTimer>
#include <QVariantMap>
#include <QScriptEngine>
#include <QScriptValueIterator>
#include "o2.h"
#include "o2replyserver.h"
#include "o0globals.h"
#include "o0settingsstore.h"
```

8.100 OAuth/src/o2.h File Reference

```
#include <QNetworkAccessManager>
#include <QNetworkRequest>
#include <QNetworkReply>
#include <QPair>
#include "o0export.h"
#include "o0baseauth.h"
#include "o2reply.h"
#include "o0abstractstore.h"
```

Classes

- class O2

  Simple OAuth2 authenticator.

8.101 OAuth/src/o2facebook.cpp File Reference

```
#include <QDebug>
#include <QJsonDocument>
#include <QJsonObject>
#include <QMap>
#include <QNetworkReply>
#include <QString>
#include <QStringList>
#include <QUrl>
#include "o2facebook.h"
#include "o0globals.h"
```

8.102 OAuth/src/o2facebook.h File Reference

```
#include "o0export.h"
#include "o2.h"
```

Classes

- class O2Facebook

  Facebook's dialect of OAuth 2.0.

8.103 OAuth/src/o2gft.cpp File Reference

```
#include "o2gft.h"
#include "o2google.h"
```

8.104 OAuth/src/o2gft.h File Reference

```
#include "o0export.h"
#include "o2google.h"
```

Classes

- class O2Gft

  Google Fusion Tables' dialect of OAuth 2.0.

8.105 OAuth/src/o2google.cpp File Reference

```
#include "o2google.h"
```

8.106 OAuth/src/o2google.h File Reference

```
#include "o2.h"
```

Classes

- class O2Google

8.107 OAuth/src/o2reply.cpp File Reference

```
#include <QTimer>
#include <QNetworkReply>
#include "o2reply.h"
```

8.108 OAuth/src/o2reply.h File Reference

```
#include <QList>
#include <QTimer>
#include <QNetworkRequest>
#include <QNetworkReply>
#include <QNetworkAccessManager>
#include <QByteArray>
#include "o0export.h"
```

Classes

- class O2Reply

  A network request/reply pair that can time out.

- class O2ReplyList

  List of O2Replies.

8.109 OAuth/src/o2replyserver.cpp File Reference

```
#include <QTcpServer>
#include <QTcpSocket>
#include <QByteArray>
#include <QString>
#include <QMap>
#include <QPair>
#include <QTimer>
#include <QStringList>
#include <QUrl>
#include <QDebug>
#include "o2replyserver.h"
```

8.110 OAuth/src/o2replyserver.h File Reference

```
#include <QTcpServer>
#include <QMap>
#include <QByteArray>
#include <QString>
#include "o0export.h"
```

Classes

- class O2ReplyServer

  HTTP server to process authentication response.

8.111 OAuth/src/o2requestor.cpp File Reference

```
#include <QDebug>
#include <QTimer>
#include "o2requestor.h"
#include "o2.h"
#include "o0globals.h"
```

8.112 OAuth/src/o2requestor.h File Reference

```
#include <QObject>
#include <QNetworkRequest>
#include <QNetworkReply>
#include <QNetworkAccessManager>
#include <QUrl>
#include <QByteArray>
#include "o0export.h"
#include "o2reply.h"
```

Classes

- class O2Requestor

  Makes authenticated requests.

8.113 OAuth/src/o2simplecrypt.cpp File Reference

```
#include "o0simplecrypt.h"
#include <QByteArray>
#include <QtDebug>
#include <QtGlobal>
#include <QDateTime>
#include <QCryptographicHash>
#include <QDataStream>
```

8.114 profile.cpp File Reference

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

8.115 profile.h File Reference

```
#include <QWidget>
#include <QtWidgets>
#include "globalindices.h"
```

Classes

- class Profile

Namespaces

- Ui

8.116 README.md File Reference

8.117 signupmenu.cpp File Reference

Sign up code linked to the sign up ui.

```
#include "signupmenu.h"
#include "ui_signupmenu.h"
```

8.117.1 Detailed Description

Sign up code linked to the sign up ui.

8.118 signupmenu.h File Reference

```
#include <cstring>
#include <QWidget>
#include <QtWidgets>
#include <QKeyEvent>
#include <regex>
#include <QString>
#include <QTimer>
#include "user.h"
#include "welcomemenu.h"
#include "mainmenu.h"
#include "globalindices.h"
#include "OAuth/helper.h"
#include "OAuth/fbdemo.h"
#include <QJsonObject>
#include <QVariantMap>
```

Classes

- class SignUpMenu

Namespaces

- Ui

8.119 user.cpp File Reference

User class that performs operations on user data in txt file.

```
#include "user.h"
```

Variables

- User ∗ user = new User

8.119.1 Detailed Description

User class that performs operations on user data in txt file.

8.119.2 Variable Documentation

8.119.2.1 User∗ user = new User

8.120 user.h File Reference

```
#include <QString>
#include <QStringList>
#include <QFile>
#include <fstream>
#include <QTextStream>
#include <string>
#include <QDebug>
```

Classes

- class User

Variables

- User ∗ user

8.120.1 Variable Documentation

8.120.1.1 User∗ user

8.121 welcomemenu.cpp File Reference

```
#include "welcomemenu.h"
#include "ui_welcomemenu.h"
#include "loginmenu.h"
#include "mainwindow.h"
#include "signupmenu.h"
#include "mainmenu.h"
```

8.122 welcomemenu.h File Reference

```
#include <QWidget>
#include <QtWidgets>
#include "user.h"
#include "globalindices.h"
```

Classes

- class WelcomeMenu

Namespaces

- Ui