My Project

Generated by Doxygen 1.7.6.1

Tue May 12 2015 21:43:56

Contents

1 Class Index **1**

1.1 Class List 1

2 File Index **3**

2.1 File List 3

3 Class Documentation **5**

3.1 gamemanager5 Class Reference 5

3.1.1 Constructor & Destructor Documentation 7

3.1.1.1 gamemanager5 7

3.1.2 Member Function Documentation 7

3.1.2.1 updateTimer 7

3.1.3 Member Data Documentation 7

3.1.3.1 temp 7

3.2 mainmenu Class Reference 8

3.2.1 Constructor & Destructor Documentation 8

3.2.1.1 mainmenu 8

3.2.2 Member Function Documentation 9

3.2.2.1 chooseAdd 9

3.2.2.2 chooseMult 9

3.3 nextwindow Class Reference 9

3.3.1 Constructor & Destructor Documentation 10

3.3.1.1 nextwindow 10

3.3.2 Member Function Documentation 10

3.3.2.1 nextLevel 11

3.3.2.2 restartLevel 11

3.4 restartwindow Class Reference 11
3.4.1 Constructor & Destructor Documentation 12
3.4.1.1 restartwindow 12
3.4.2 Member Function Documentation 12
3.4.2.1 restartLevel 12
3.5 sbox Class Reference 12

4 File Documentation **15**
4.1 gamemanager5.cpp File Reference 15
4.1.1 Detailed Description 15
4.2 mainmenu.cpp File Reference 15
4.2.1 Detailed Description 15
4.3 nextwindow.cpp File Reference 15
4.3.1 Detailed Description 16
4.4 restartwindow.cpp File Reference 16
4.4.1 Detailed Description 16

Chapter 1

Class Index

1.1 Class List

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

gamemanager5 5
mainmenu 8
nextwindow 9
restartwindow 11
sbox 12

Chapter 2

File Index

2.1 File List

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

gamemanager5.cpp
Source file for gamemanager5.h 15
gamemanager5.h . . . . . **??**
mainmenu.cpp
Source file for mainmenu.h 15
mainmenu.h . . . . . **??**
nextwindow.cpp
Source file for nextwindow.h 15
nextwindow.h . . . . . **??**
restartwindow.cpp
Source file for restartwindow.h 16
restartwindow.h . . . . . **??**
sbox.h . . . . . **??**

Chapter 3

Class Documentation

3.1 gamemanager5 Class Reference

Public Slots

- void updateTimer ()

  slot function

Public Member Functions

- gamemanager5 (QObject *parent=0)

  default constructor
- gamemanager5 (QGraphicsScene *scene, sbox *box, int level, int score, int character)

  constructor of the gameManager class
- void **showNumbers** ()
- void **showpic** (int)

Public Attributes

- sbox * **box**
- QGraphicsTextItem * **inputTextItem**
- QString temp

  TextItem to show the input number.
- QGraphicsScene * scene

  the game scene
- QPixmap imagebackground

  background image
- QGraphicsPixmapItem * itembackground

  item for background image

- QPixmap imageframe

  square frame image
- QGraphicsPixmapItem * itemframe

  item for square frame
- QPixmap imagedoors

  square frame image
- QGraphicsPixmapItem * itemdoors

  item for square frame
- QGraphicsTextItem * **itemoption1**
- QGraphicsTextItem * **itemoption2**
- QGraphicsTextItem * **question**
- nextwindow * nextWindow1

  nextwindow that shows up after winning a level
- restartwindow * restartWindow1

  restartwindow that shows up after losing a level
- QString **state**
- QTimer * gameTimer

  timer for the game
- int **number1**
- int **number2**
- int **option1**
- int **option2**
- int **luck**
- int **counter**
- int character

  int determines if it's addition or multiplication
- int level

  int shows which level we're at
- int timer

  int timer
- int score

  int score
- int second

  int to check the time
- QGraphicsTextItem * clock

  textitem for the timer
- QGraphicsTextItem * scoreItem

  textitem for the score

3.1.1 Constructor & Destructor Documentation

3.1.1.1 gamemanager5::gamemanager5 (QGraphicsScene * *s*, sbox * *b*, int *l*, int *sc*, int *c*)

constructor of the gameManager class

It takes as parameters the scene of the game to add and remove items from it. It the level to determine the numbers and the array lengths. It takes the textItem "score" to change it with the next level logic. It takes the int "opreation" to check if the operation is addition or multiplication .

Parameters

scene	
level	
score	
operation	

<update function when timer is timeout

3.1.2 Member Function Documentation

3.1.2.1 void gamemanager5::updateTimer ( ) [slot]

slot function

This is where the timer and score are updated.

3.1.3 Member Data Documentation

3.1.3.1 QString gamemanager5::temp

TextItem to show the input number.

string in which the input number is stored

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

- gamemanager5.h
- gamemanager5.cpp

3.2 mainmenu Class Reference

Public Slots

- void chooseAdd ()

  function when choosing addition

- void chooseMult ()

  function when choosing addition

Public Member Functions

- mainmenu (QWidget *parent=0)

  default constructor
- mainmenu (gamemanager5 *game, QGraphicsView *view)

  constructor for the mainmenu class

Public Attributes

- QLabel * **choose**
- QPushButton * addition

  button "Addition"
- QPushButton * multiplication

  button "Multiplication"
- QVBoxLayout * **vBox**
- gamemanager5 * game

  gameManager class that it will use and edit
- QGraphicsView * **view**
- int operation

  int to determine addition or multiplication

3.2.1 Constructor & Destructor Documentation

3.2.1.1 mainmenu::mainmenu (gamemanager5 * *game,* QGraphicsView * *view*)

constructor for the mainmenu class

It takes as parameter gameManager "game" to control the game accordingly

Parameters

game	

<chooseAdd function when addition is clicked

<chooseMult function when multiplication is clicked

3.2.2 Member Function Documentation

3.2.2.1 void mainmenu::chooseAdd ( ) [slot]

function when choosing addition

slot function

This function sets the operation to 0 since the player chose to get the addition. It then starts the game by filling the arrays accordingly, starting the timers and closing the commandWin

3.2.2.2 void mainmenu::chooseMult ( ) [slot]

function when choosing addition

slot function

This function sets the operation to 1 since the player chose to get the multiplication. It then starts the game by filling the arrays accordingly, starting the timers and closing the commandWin

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

- mainmenu.h
- mainmenu.cpp

3.3 nextwindow Class Reference

Public Slots

- void nextLevel ()
  slot function
- void restartLevel ()
  slot function

Public Member Functions

- nextwindow (QWidget *parent=0)
  default constructor
- nextwindow (gamemanager5 *, sbox *)
  constructor of the nextwindow class

Public Attributes

- gamemanager5 * game
  gameManager class that we will use and edit
- sbox * **frame**
- int **score**
- int **level**
- int **operation**
- QGraphicsScene * **scene**

- QLabel * **scoreLabel**
- QLabel * **sug**
- QPushButton * nextLev

  button "Next Level"
- QPushButton * restartLev

  button "Restart Level"
- QGridLayout * **gridBox**

3.3.1 Constructor & Destructor Documentation

3.3.1.1 nextwindow::nextwindow (gamemanager5 * *game*, sbox * *frame*)

constructor of the nextwindow class

This is where everything is instantiated. The buttons and labels are added to the grid layout.

Parameters

game	
sbox	

<nextLevel function when nextLev is clicked

<restartLevel function when restartLev is clicked

3.3.2 Member Function Documentation

3.3.2.1 void nextwindow::nextLevel ( ) [slot]

slot function

This is where the level parameter is incremented and and everything is re-initialized with the new score and level.

3.3.2.2 void nextwindow::restartLevel ( ) [slot]

slot function

If the player chooses to restart the level, the score is set to zero, the level parameter is set to its initial value and the game is re-initialized.

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

- nextwindow.h
- nextwindow.cpp

3.4 restartwindow Class Reference

Public Slots

- void restartLevel ()

  slot function

Public Member Functions

- restartwindow (QWidget *parent=0)

  default constructor
- restartwindow (gamemanager5 *, sbox *)

  constructor of the restartwindow class

Public Attributes

- gamemanager5 * game

  gameManager class that we will use and edit
- sbox * **frame**
- int **score**
- int **level**
- int **operation**
- QGraphicsScene * **scene**
- QLabel * **scoreLabel**
- QLabel * **sug**
- QPushButton * restartLev

  button "Restart Level"
- QGridLayout * **gridBox**

3.4.1 Constructor & Destructor Documentation

3.4.1.1 restartwindow::restartwindow (gamemanager5 * *game*, sbox * *frame*)

constructor of the restartwindow class

This is where everything is instantiated. The buttons and labels are added to the grid layout.

Parameters

game	
sbox	

<restartLevel function when restartLev is clicked

3.4.2 Member Function Documentation

3.4.2.1 void restartwindow::restartLevel ( ) [slot]

slot function

If the player chooses to restart the level, the score is set to zero, the level parameter is set to its initial value and the game is re-initialized.

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

- restartwindow.h
- restartwindow.cpp

3.5 sbox Class Reference

Public Member Functions

- void **keyPressEvent** (QKeyEvent ∗event)

Public Attributes

- int **move**

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

- sbox.h
- sbox.cpp

Chapter 4

File Documentation

4.1 gamemanager5.cpp File Reference

source file for gamemanager5.h

```
#include "gamemanager5.h" #include "sbox.h" #include "mainmenu.-
h" #include "nextwindow.h" #include "restartwindow.h"
```

4.1.1 Detailed Description

source file for gamemanager5.h

4.2 mainmenu.cpp File Reference

source file for mainmenu.h

```
#include "mainmenu.h" #include "sbox.h" #include "gamemanager5.-
h"
```

4.2.1 Detailed Description

source file for mainmenu.h

4.3 nextwindow.cpp File Reference

source file for nextwindow.h

```
#include "nextwindow.h" #include "gamemanager5.h" #include
"sbox.h"
```

4.3.1 Detailed Description

source file for nextwindow.h

4.4 restartwindow.cpp File Reference

source file for restartwindow.h

```
#include "restartwindow.h"      #include "gamemanager5.h" ×
#include "sbox.h"
```

4.4.1 Detailed Description

source file for restartwindow.h