

## My Project

Generated by Doxygen 1.7.6.1

Sun Apr 26 2015 23:31:40



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	gamemanager3 Class Reference . . . . .	5
3.1.1	Constructor & Destructor Documentation . . . . .	7
3.1.1.1	gamemanager3 . . . . .	7
3.1.2	Member Function Documentation . . . . .	7
3.1.2.1	fillArray . . . . .	8
3.1.2.2	showNumbers . . . . .	8
3.1.2.3	updateTimer . . . . .	8
3.1.3	Member Data Documentation . . . . .	8
3.1.3.1	temp . . . . .	8
3.2	mainmenu Class Reference . . . . .	8
3.2.1	Constructor & Destructor Documentation . . . . .	9
3.2.1.1	mainmenu . . . . .	9
3.2.2	Member Function Documentation . . . . .	10
3.2.2.1	chooseAdd . . . . .	10
3.2.2.2	chooseMult . . . . .	10
3.3	nextwindow Class Reference . . . . .	10
3.3.1	Constructor & Destructor Documentation . . . . .	11
3.3.1.1	nextwindow . . . . .	11
3.3.2	Member Function Documentation . . . . .	11

3.3.2.1	nextLevel . . . . .	11
3.3.2.2	restartLevel . . . . .	12
3.4	restartwindow Class Reference . . . . .	12
3.4.1	Constructor & Destructor Documentation . . . . .	13
3.4.1.1	restartwindow . . . . .	13
3.4.2	Member Function Documentation . . . . .	13
3.4.2.1	restartLevel . . . . .	13
3.5	sframe Class Reference . . . . .	13
3.5.1	Constructor & Destructor Documentation . . . . .	14
3.5.1.1	sframe . . . . .	14
3.5.2	Member Function Documentation . . . . .	14
3.5.2.1	keyPressEvent . . . . .	14
<b>4</b>	<b>File Documentation</b>	<b>15</b>
4.1	gamemanager3.cpp File Reference . . . . .	15
4.1.1	Detailed Description . . . . .	15
4.2	gamemanager3.h File Reference . . . . .	15
4.2.1	Detailed Description . . . . .	16
4.3	mainmenu.cpp File Reference . . . . .	16
4.3.1	Detailed Description . . . . .	16
4.4	nextwindow.cpp File Reference . . . . .	16
4.4.1	Detailed Description . . . . .	16
4.5	restartwindow.cpp File Reference . . . . .	16
4.5.1	Detailed Description . . . . .	16
4.6	sframe.cpp File Reference . . . . .	16
4.6.1	Detailed Description . . . . .	17
4.7	sframe.h File Reference . . . . .	17
4.7.1	Detailed Description . . . . .	17

# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">gamemanager3</a>	5
<a href="#">mainmenu</a>	8
<a href="#">nextwindow</a>	10
<a href="#">restartwindow</a>	12
<a href="#">sframe</a>	13



## Chapter 2

# File Index

### 2.1 File List

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

<a href="#">gamemanager3.cpp</a>	
Source file for <a href="#">gamemanager3.h</a>	15
<a href="#">gamemanager3.h</a>	
Class for the Game Manager	15
<a href="#">mainmenu.cpp</a>	
Source file for <a href="#">mainmenu.h</a>	16
<b>mainmenu.h</b>	??
<a href="#">nextwindow.cpp</a>	
Source file for <a href="#">nextwindow.h</a>	16
<b>nextwindow.h</b>	??
<a href="#">restartwindow.cpp</a>	
Source file for <a href="#">restartwindow.h</a>	16
<b>restartwindow.h</b>	??
<a href="#">sframe.cpp</a>	
Source file for <a href="#">sframe.h</a>	16
<a href="#">sframe.h</a>	
Class for the sframe	17





## Chapter 3

# Class Documentation

### 3.1 gamemanager3 Class Reference

#### Public Slots

- void [updateTimer](#) ()  
*slot function*

#### Public Member Functions

- [gamemanager3](#) (QObject \*parent=0)  
*default constructor*
- [gamemanager3](#) (QGraphicsScene \*scene, int level, int score, int operation)  
*constructor of the gameManager class*
- void [showNumbers](#) ()  
*function to show the numbers in the array*
- void [fillArray](#) (int)  
*function to fill the number arrays*

#### Public Attributes

- 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*
- [nextwindow](#) \* [nextWindow1](#)
  - nextwindow that shows up after winning a level*
- [restartwindow](#) \* [restartWindow1](#)
  - restartwindow that shows up after losing a level*
- QTimer \* [gameTimer](#)
  - timer for the game*
- int [inputNumber](#)
  - int number input by user*
- int [nextNumber](#)
  - int should be the next answer*
- int [operation](#)
  - int determines if it's addition or multiplication*
- int \* [branch1](#)
  - int array to store answers of branch 1*
- int \* [branch2](#)
  - int array to store answers of branch 2*
- int \* [branch3](#)
  - int array to store answers of branch 3*
- int \* [branch4](#)
  - int array to store answers of branch 4*
- int [stage](#)
  - int shows which stage the user's in*
- int [step1](#)
  - int shows which step is the user in branch 1*
- int [step2](#)
  - int shows which step is the user in branch 2*
- int [step3](#)
  - int shows which step is the user in branch 3*
- int [step4](#)
  - int shows which step is the user in branch 4*
- 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*
- QGraphicsTextItem \* [inputLabel](#)  
*textitem for the input number*
- QGraphicsTextItem \* [numberArray1](#)  
*textitem array to show the numbers of branch 1*
- QGraphicsTextItem \* [numberArray2](#)  
*textitem array to show the numbers of branch 2*
- QGraphicsTextItem \* [numberArray3](#)  
*textitem array to show the numbers of branch 3*
- QGraphicsTextItem \* [numberArray4](#)  
*textitem array to show the numbers of branch 4*

### 3.1.1 Constructor & Destructor Documentation

#### 3.1.1.1 gamemanager3::gamemanager3 ( QGraphicsScene \* s, int l, int sc, int o )

constructor of the gameManager class

It takes as parameters the scene of the game to add and remove items from it. It takes 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 "operation" to check if the operation is addition or multiplication .

##### Parameters

<i>scene</i>	
<i>level</i>	
<i>score</i>	
<i>operation</i>	

<update function when timer is timeout

### 3.1.2 Member Function Documentation

#### 3.1.2.1 void gamemanager3::fillArray ( int op )

function to fill the number arrays

fill function

Fills the 4 arrays with numbers. Takes operation to fill with addition or multiplication.

## Parameters

<i>operation</i>	
------------------	--

**3.1.2.2 void gamemanager3::showNumbers ( )**

function to show the numbers in the array

show function

Shows the numbers on the screen.

**3.1.2.3 void gamemanager3::updateTimer ( )** [slot]

slot function

This is where the timer and score are updated.

**3.1.3 Member Data Documentation****3.1.3.1 QString gamemanager3::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:

- [gamemanager3.h](#)
- [gamemanager3.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](#) (gamemanager3 \*game)  
*constructor for the mainmenu class*

## Public Attributes

- QLabel \* **choose**
- QPushButton \* **addition**  
*button "Addition"*
- QPushButton \* **multiplication**  
*button "Multiplication"*
- QVBoxLayout \* **vBox**
- gamemanager3 \* **game**  
*gameManager class that it will use and edit*
- int **operation**  
*int to determine addition or multiplication*

## 3.2.1 Constructor & Destructor Documentation

### 3.2.1.1 mainmenu::mainmenu ( gamemanager3 \* game )

constructor for the mainmenu class

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

#### Parameters

<i>game</i>	
-------------	--

<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](#) ([gamemanager3](#) \*, [sframe](#) \*)  
*constructor of the nextwindow class*

#### Public Attributes

- [gamemanager3](#) \* [game](#)  
*gameManager class that we will use and edit*
- [sframe](#) \* **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 ( gamemanager3 \* game, sframe \* frame )

constructor of the nextwindow class

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

##### Parameters

<i>game</i>	
<i>sframe</i>	

<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](#) (gamemanager3 \*, [sframe](#) \*)  
*constructor of the restartwindow class*

## Public Attributes

- [gamemanager3](#) \* [game](#)  
*gameManager class that we will use and edit*
- [sframe](#) \* **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](#) ( [gamemanager3](#) \* *game*, [sframe](#) \* *frame* )

constructor of the restartwindow class

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

#### Parameters

<i>game</i>	
<i>sframe</i>	

<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 sframe Class Reference

### Public Member Functions

- [sframe](#) ([gamemanager3](#) \*)  
*default constructor for the sframe class*
- void [keyPressEvent](#) ([QKeyEvent](#) \*event)  
*keyPressEvent function of the sframe*

### Public Attributes

- [gamemanager3](#) \* [game](#)  
*[gamemanager3](#) to get all the game information*

### 3.5.1 Constructor & Destructor Documentation

#### 3.5.1.1 [sframe::sframe](#) ( [gamemanager3](#) \* *game* )

default constructor for the sframe class

It instantiates all the components of the class. It takes all the game information from [gamemanager3](#)

### 3.5.2 Member Function Documentation

#### 3.5.2.1 void [sframe::keyPressEvent](#) ( [QKeyEvent](#) \* *event* )

[keyPressEvent](#) function of the sframe

This function is what makes the sframe move after checking the answer. It is used whenever a number is pressed or the Enter, Right Arrow or Backslash keys. The sframe is set to focus in the main function so that it can work. The parameter "event" detects key presses. So we use if statements to implement the logic of the movement of the sframe.

#### Parameters

<i>event</i>	
--------------	--

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

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

## Chapter 4

# File Documentation

### 4.1 gamemanager3.cpp File Reference

source file for [gamemanager3.h](#)

```
#include "gamemanager3.h"    #include "sframe.h"    #include  
"mainmenu.h" #include "nextwindow.h" #include "restartwindow.-  
h"
```

#### 4.1.1 Detailed Description

source file for [gamemanager3.h](#)

### 4.2 gamemanager3.h File Reference

class for the Game Manager

```
#include <QGraphicsScene>      #include <QGraphicsPixmap-  
Item> #include <QKeyEvent> #include <QLayout> #include <-  
QObject> #include <QTimer> #include "sframe.h"
```

#### Classes

- class [gamemanager3](#)

#### 4.2.1 Detailed Description

class for the Game Manager This class basically takes care of everything that is happening in the game. It makes the square move according to the answers. It shows the elapsed time according to the gameTimer along with the score.

## 4.3 mainmenu.cpp File Reference

source file for [mainmenu.h](#)

```
#include "mainmenu.h" #include "sframe.h" #include "gamemanager3.-  
h"
```

### 4.3.1 Detailed Description

source file for [mainmenu.h](#)

## 4.4 nextwindow.cpp File Reference

source file for [nextwindow.h](#)

```
#include "nextwindow.h" #include "gamemanager3.h" #include  
"sframe.h"
```

### 4.4.1 Detailed Description

source file for [nextwindow.h](#)

## 4.5 restartwindow.cpp File Reference

source file for [restartwindow.h](#)

```
#include "restartwindow.h" #include "gamemanager3.h" ×  
#include "sframe.h"
```

### 4.5.1 Detailed Description

source file for [restartwindow.h](#)

## 4.6 sframe.cpp File Reference

source file for [sframe.h](#)

```
#include "sframe.h" #include "gamemanager3.h" #include  
"nextwindow.h" #include "restartwindow.h"
```

### 4.6.1 Detailed Description

source file for [sframe.h](#)

## 4.7 sframe.h File Reference

class for the sframe

```
#include <QGraphicsScene>          #include <QGraphicsPixmap-  
Item> #include <QKeyEvent> #include <QLayout> #include <-  
QGraphicsRectItem>
```

### Classes

- class [sframe](#)

#### 4.7.1 Detailed Description

class for the sframe This sframe is a rectItem. It is moved up,right or down with the correct/wrong answer or pressing the right arrow. With it the player has to reach a castle by answering correctly.

#### Author

Ali Naji, Alexis Haddad