# MP 2 Documentation

# “Simon Slays”

Jennie Ron S. Ablog
John Christian E. Sun

# Table of Contents

# Hierarchical Index

## Class Hierarchy

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

# Class Index

## Class List

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

# File Index

## File List

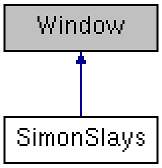
Here is a list of all files with brief descriptions:

# Class Documentation

## SimonSlays Class Reference

```
#include <simonslays.h>
```

Inheritance diagram for SimonSlays:



### Public Member Functions

- **SimonSlays** ()
  *Constructor of class* ***SimonSlays****.*

- **~SimonSlays** ()
  *Destructor of class* ***SimonSlays****.*

- void **resetGame** ()
  *Sets game conditions necessary to start a new game.*

- void **on_button_clicked** (int)
  *Signal handler for Gtk::Button gameButton[] and primary handler for the actual game.*

- bool **simonMove** (std::list< int > &)
  *Mimics Simon's (CPU) every move for each round. Scheduled to be called every unsigned int interval milliseconds.*

- void **slay** (int)
  *Animates the button images and schedules bool SimonSlays::on_time_out(int i, int j).*

- bool **on_timeout** (int, int)
  *Scheduled for one-time call with a duration of int timeout_value, 250 milliseconds. Changes button images from clicked version to default version.*

- void **show_entry_dialog** ()
  *Shows the dialog for name entry on new game.*

- void **on_button_quit** ()
  *Exits the game.*

- void **on_button_reset** ()
  *Resets the game.*

## Constructor & Destructor Documentation

### SimonSlays::SimonSlays ()

Constructor of class **SimonSlays**.

1 Calls the constructors of Gtk::Widgets on which it passes parameters necessary for: loading of images, initialization of Gtk::Labels, button labels, orientation of boxes, and variable values needed by the timeout functions used.

2 Sets the window's properties: default size, border width, initial position, icon and title.

3 Sets the margins of Gtk::Box statBox and Gtk::Box menuBox.

4 Pack starts other boxes to Gtk::Box bigBox and their children to themselves.

5 Connects Gtk::Button quitButton and Gtk::Button resetButton to signal handlers void **SimonSlays::on_button_quit()**; and void **SimonSlays::on_button_reset()**;

6 Calls void **SimonSlays::resetGame()**; to start a new game.

### SimonSlays::~SimonSlays ()

Destructor of class **SimonSlays**.

1 This is the destructor of class **SimonSlays**.

---

## Member Function Documentation

### void SimonSlays::on_button_clicked (int )

Signal handler for Gtk::Button gameButton[] and primary handler for the actual game.

1 Passes int n to void **SimonSlays::slay()** which animates the button's image to signify click.

2 [SIMON'S TURN] IF list<int> tempList is empty: Buttons are made insensitive to prevent user interaction during the flashing of latest sequence. Flashing of sequence is made possible by scheduling a timeout function bool **SimonSlays::simonMove()**.

[PLAYER'S TURN] ELSE: Tests if user's move is correct (or if parameter int n is equal to tempList.begin())

Correct move leads to list<int> simonsList.pop_front(); Wrong move leads to updating high score file (score.txt) if necessary, and running the local obj Gtk::MessageDialog endDialog();

**Parameters:**

| | |
|---|---|
| *n* | The integer pertaining to the index of the button clicked in Gtk::Button gameButton[] IF user clicked a button. -1 IF AND ONLY IF function is called within code i.e. done to start the game. |

### void SimonSlays::on_button_quit ()

Exits the game.

This function exits the game window!

### void SimonSlays::on_button_reset ()

Resets the game.

This function calls **resetGame()** to reset the game!

### bool SimonSlays::on_timeout (int , int )

Scheduled for one-time call with a duration of int timeout_value, 250 milliseconds. Changes button images from clicked version to default version.

1 On first call, sets button images to default and returns false thereby unscheduling self.

2 If not first call, return true.

**Parameters:**

| | |
|---|---|
| *n* | Index of the timer in std::map<int, int> counters. |
| *i* | Index of the button clicked in Gtk::Button gameButton[]. |

**Returns:**

Returns false on first call to unschedule self, and true otherwise.

### void SimonSlays::resetGame ()

Sets game conditions necessary to start a new game.

1 Initializes game variable int level to default value 0.

2 Reads the initial value for int hs when score.txt file exists.

3 Calls void **SimonSlays::show_entry_dialog()** to fetch player name and void SimonSlays::startDialog() to signify start of new game.

4 Calls void **SimonSlays::on_button_clicked()** with -1 as parameter to start game.

### void SimonSlays::show_entry_dialog ()

Shows the dialog for name entry on new game.

1 IF button OK is pressed, username will be set to player's input on entry; if player's input is empty, username will be 'stranger'

2 IF button CANCEL is pressed, the program will be terminated.

### bool SimonSlays::simonMove (std::list< int > & )

Mimics Simon's (CPU) every move for each round. Scheduled to be called every unsigned int interval milliseconds.

1 IF list<int>& someList is empty: Buttons are made sensitive to make way for user interaction. Returns false to unschedule self.

ELSE: Passes first element of list<int>& someList to void **SimonSlays::slay()** which animates the button's image to signify Simon's move. Removes first element of list<int>& someList.

**Parameters:**

| | |
|---|---|
| *someList* | A mutable list that contains the current sequence of Simon's moves. |

**Returns:**

Returns false if list<int>& someList is empty to unschedule self, and true if otherwise.

### void SimonSlays::slay (int )

Animates the button images and schedules bool SimonSlays::on_time_out(int i, int j)

1 Sets the image gameButton[n] to its corresponding clicked version then calls bool SimonSlays::on_time_out() with int timer_number and int n as parameter.

**Parameters:**

| | |
|---|---|
| *n* | Shall contain index of the button clicked in Gtk::Button gameButton[] IF user clicked a button. |

---

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

*simonslays.h*<br>*simonslays.cpp*

# File Documentation

## main.cpp File Reference

```
#include "simonslays.h"
```

### Functions

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

### Function Documentation

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

## simonslays.cpp File Reference

```
#include "simonslays.h"
#include <iostream>
#include <cstdlib>
#include <ctime>
#include <unistd.h>
#include <fstream>
```

## simonslays.h File Reference

```
#include <gtkmm.h>
#include <map>
#include <list>
#include <cstring>
```

### Classes

- class **SimonSlays**