

KATHMANDU UNIVERSITY

SCHOOL OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



A PROJECT REPORT ON

‘SARPA’: A Dual Player Snake Game

[Code No: 102]

(For the partial fulfillment of 1st year, 2nd semester in Computer Engineering)

Submitted To:

Mr. Sameer Tamrakar

Project Supervisor

Department of Computer Science & Engineering (DoCSE)

Submitted By:

Prajwol Lamichhane (26)

Arpan Mahatra (28)

Abhay Raut (43)

Computer Engineering

1st year, 2nd semester

Submission Date: 13th August, 2017

August 13, 2017

CERTIFICATION

FIRST YEAR SECOND SEMESTER PROJECT REPORT

On

‘SARPA: A Dual Player Snake Game’

Certified that this project report “SARPA : A Dual Player Snake Game” is the bona fide work of “Prajwol Lamichhane, Arpan Mahatra, Abhay Raut Chettri”, who carried out the project work under my supervision.

Approved by :

Mr. Sameer Tamrakar

SUPERVISOR

Mr. Nabin Ghimire

PROJECT COORDINATOR

Department of Computer Science and Engineering (DoCSE)
Kathmandu University
Dhulikhel, Kavre

ACKNOWLEDGEMENT

It is a delight to acknowledge those who have supported us over the last semester. We would like to thank our supervisor, Mr. Sameer Tamrakar, for his guidance and relaxed, thoughtful insight who gave us constant supervision and guidance for the betterment of our project.. Many people have played a great role for the completion of our project. We are very thankful for all the support that we got from all the teachers, coordinators, friends and many individuals.

Also, we would like to express our deepest appreciation towards Kathmandu University Department of Computer Science and Engineering (DoCSE) who gave us the golden opportunity to do this wonderful project. We would like to express our sincere thanks to all the teachers of DoCSE for their guidance and encouragement making this project possible. We would also like to thank Mr. Nabin Ghimire for his proper coordination of every project held in the semester.

Finally, we wish to thank our parents for their love and encouragement, without whom we would never have enjoyed so many opportunities. We are really thankful to our colleagues who helped us in developing the project. At-last, we would like to extend our sincere thanks to all of them and are really grateful to their kindness.

ABSTRACT

This project aims to bring the fun and simplicity of snake game with some new features. It will include two opponents whose aim will be to win over each other on the respective levels. This game will have three rounds of play on the three different locations. The rounds are named as green round ,blue round and the ultimatum round that will determine the wining or losing of a player. If the round is green the green snake will get benefited as the bugs will not harm the green snake and vice-versa for the red snake in the red round. While in the ultimatum round both snakes can get harmed by the bugs hovering on the screen.

This project explores a new dimension in the traditional snake game to make it more interesting and challenging. The simplicity of this game makes it an ideal candidate for a end semester project as we can focus on advanced topics like dual player functionality and implementation of graphics and sound.

LIST OF ILLUSTRATIONS

List of Abbreviations

SMFL: Simple and Fast Multimedia Layer

OS: Operating System

GUI: Graphical User Interface

PC: Personal Computer

List of Figures

Fig 1: CodeBlocks Platform

Fig 2: Opening Window Page

Fig 3: Green Round Window

Fig 4: Red Round Window

Fig 5: Ultimatum Window

Fig 6: Green Wins

Fig 7: Red Wins

Fig 8: Progress Logs

Table of Contents

1. INTRODUCTION	1
1.1 Purpose:	1
1.2 Concept of Snake Game:	1
1.3 Objectives:	1
2. LITERATURE REVIEW	2
3. SYSTEM	4
3.1 Tools Used:	4
4. SOFTWARE DEVELOPMENT	5
4.1 Program Flowchart	5
4.2 Screenshots	6
4.3 Structures and Functions	13
5. PROCESS LOGS	15
6. DISSCUSION	16
6.1 Features:	16
6.2 Limitations:	16
6.3 Further Development:	17
7. CONCLUSION	18

Chapter 1: INTRODUCTION

1.1 Purpose:

The main aim of our project is to develop a Dual Player Snake Game as a part of our course content ENGG1O2. This software will be basically for the purpose of entertainment, furthermore; it will help us develop some standard programming skills.

1.2 Concept of Snake Game:

As per the project given to us we thought of creating something for the purpose of entertainment. In this era of busy life style, people hardly get to spend some fun time. For such people this game would help them overcome their boredom. One can play with his/her friends and enjoy socializing. This is a strategy game where one player struggles with the other for surviving over the various levels which makes it even more interesting. The main concept of this game is to create a dual player snake game that would aim not only for the purpose of entertainment but also help one socializing with their friends.

1.3 Objectives

The objectives are:

- i. To create a snake game that will have all the functionality of traditional snake games.
- ii. Introduce dual player functionality in the game that will allow two players to play a game through the same key board.
- iii. It should give the experience of real time to the players.

This game aims to change the way people think of traditional snake game. It will offer the experience of commercial dual player game to the player retaining the simplicity of traditional snake game.

Chapter 2: LITERATURE REVIEW

- Traditional Snake game

Snake is the common name for a videogame concept where the player maneuvers a line which grows in length, with the line itself being a primary obstacle. The concept originated in the 1976 arcade game *Blockade*, and the ease of implementing *Snake* has led to hundreds of versions (some of which have the word *snake* or *worm* in the title) for many platforms. There are over 300 for iOS alone.

After a variant was preloaded on Nokia mobile phones in 1998, there was a resurgence of interest in the snake concept as it found a larger audience.

The gameplay consists of a player who controls a dot, square, or object on a bordered plane. As it moves forward, it leaves a trail behind, resembling a moving snake. In some games, the end of the trail is in a fixed position, so the snake continually gets longer as it moves. In another common scheme, the snake has a specific length, so there is a moving tail a fixed number of units away from the head. The player loses when the snake runs into the screen border, a trail or other obstacle, or itself.

- Slither.io

Slither.io is a massively multiplayer browser game developed by Steve Howse. Players control an avatar resembling a snake, which consumes multicolored pellets from other players, and ones that naturally spawn on the map in the game to grow in size. The objective of the game is to grow the longest snake in the server. *Slither.io* is similar in concept to the popular 2015 web game *Agar.io* and is reminiscent of the classic arcade game *Snake*.

The objective of the game is to control and move a snake around a colored area, eat pellets to gain mass, defeat and consume other players to grow the biggest and longest in the game. If the player's snake's head collides into a part of another snake, the player loses the game and must start over. The defeated avatar's body turns into bright, shining pellets for other players to consume. These pellets that remain from "death" of an avatar will correspond to the color of the avatar itself, and are both brighter and bigger than normal pellets.

Pellets also spawn from other snake avatars. By either pressing the space bar or clicking on the mouse or trackpad, the player can activate "boost mode", which causes the avatar to speed

up. When a player uses "boost mode", the snake loses some mass, causing the snake's size to shrink slightly, with the mass that is lost from the boost appearing as a line of dots where the boost was used. This feature is useful to outmaneuver and defeat opponents. Another strategy that players use to defeat opponents is coiling around them in a loop until the opponent, trapped in the loop, crashes into the player.

There is a border that confines avatars within the circular game board. If a snake hits the border, the player automatically dies without turning into the aforementioned pellets. As of 2016, the player with the biggest snake at the end of the day gets to share a "victory message" with the world

Chapter 3: SYSTEM

3.1 Tools used

Following tools were used for the project building

- Code::Blocks 16.01.0.0
- Atom Text Editor
- SFML graphics library

Chapter 4: SOFTWARE DEVELOPMENT

4.1 Program Flowchart:

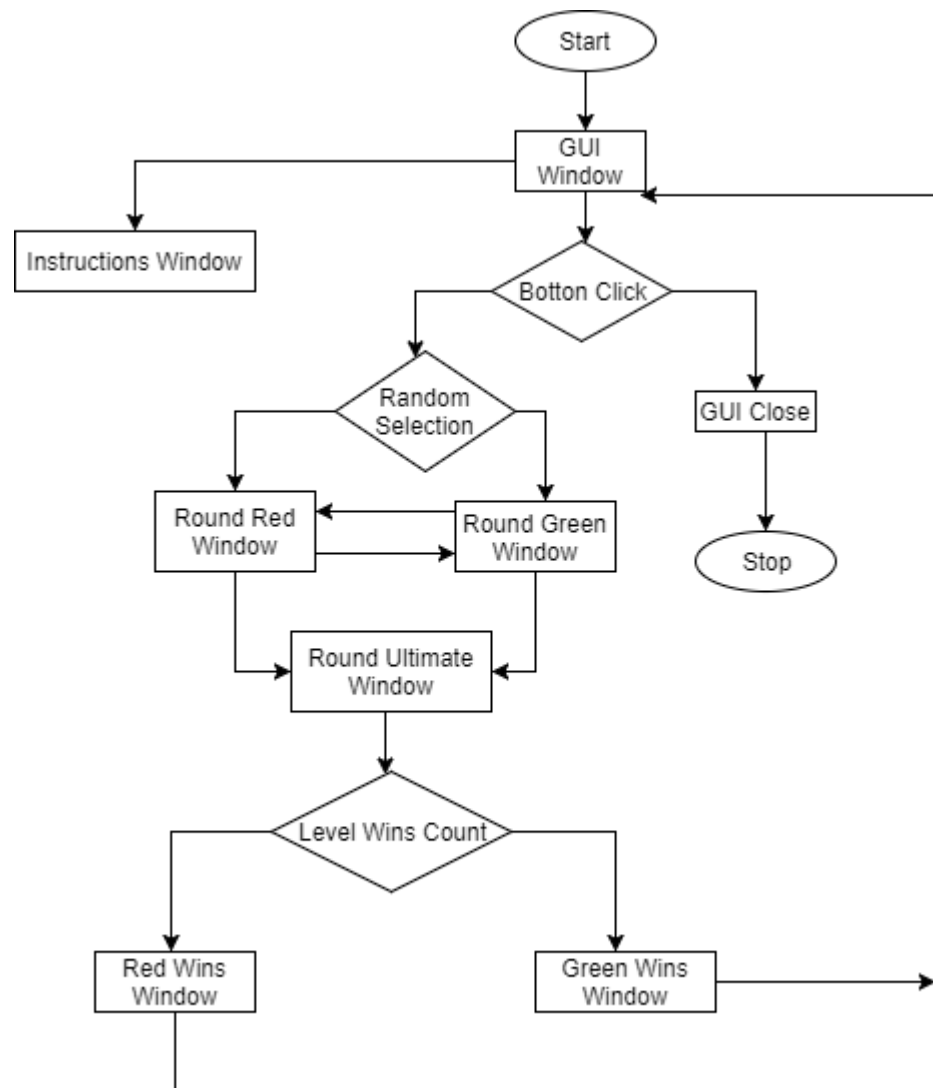


Figure: Program Flowchart

4.2 Screenshots:



Figure: GUI Window

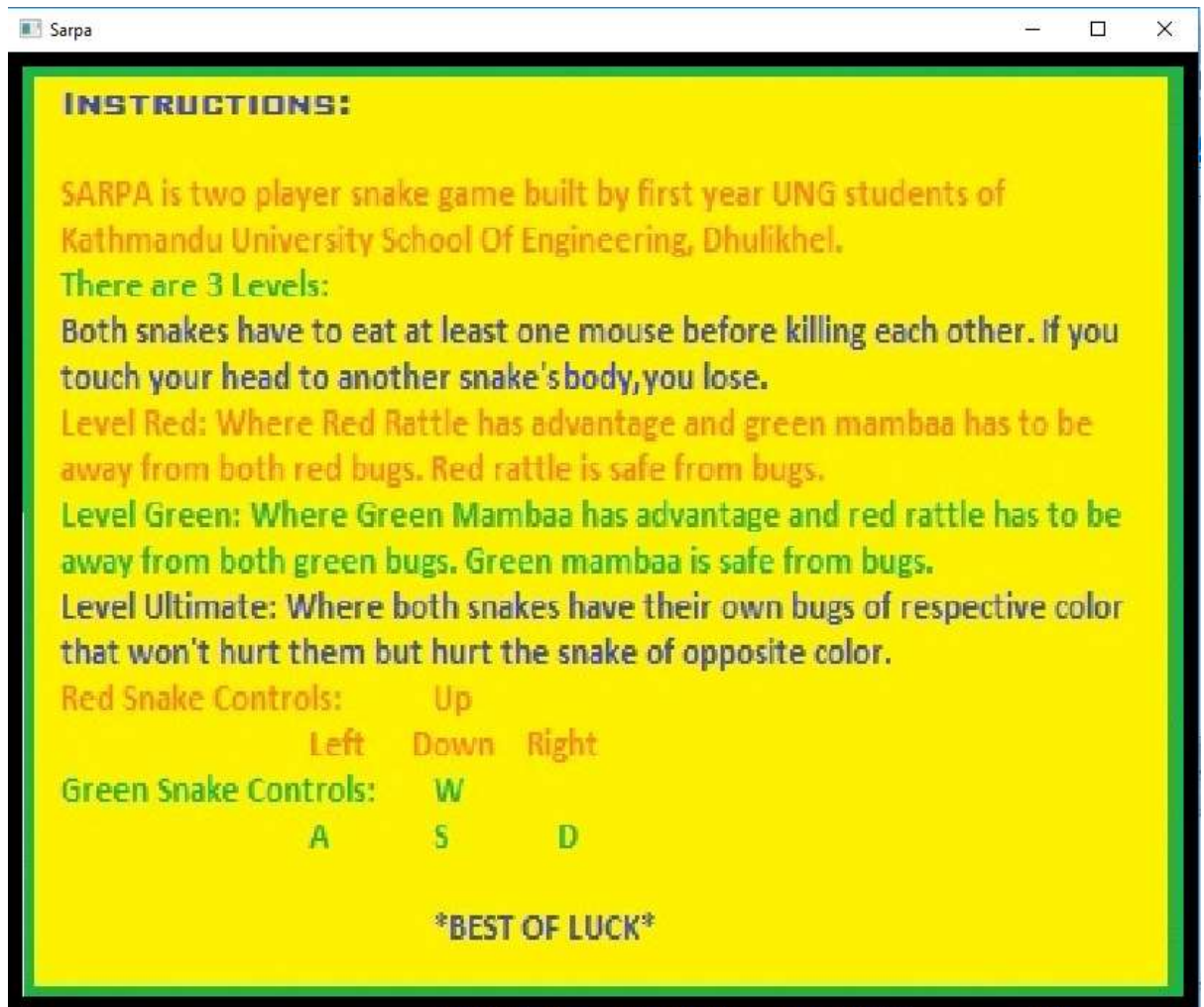


Figure: Instructions Window



Figure: Round Red Window (After GUI red)

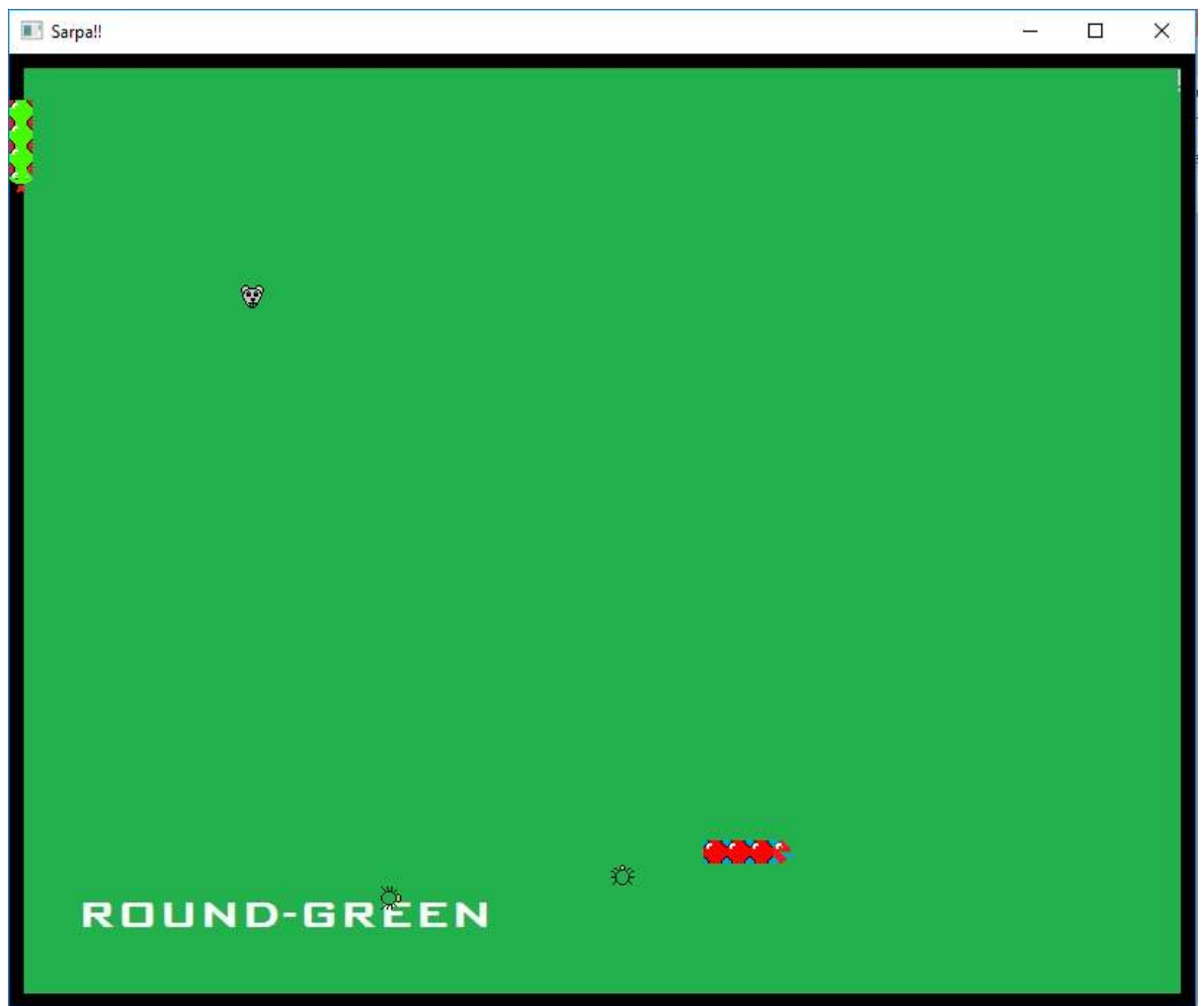


Figure: Round Green Window (After GUI green)

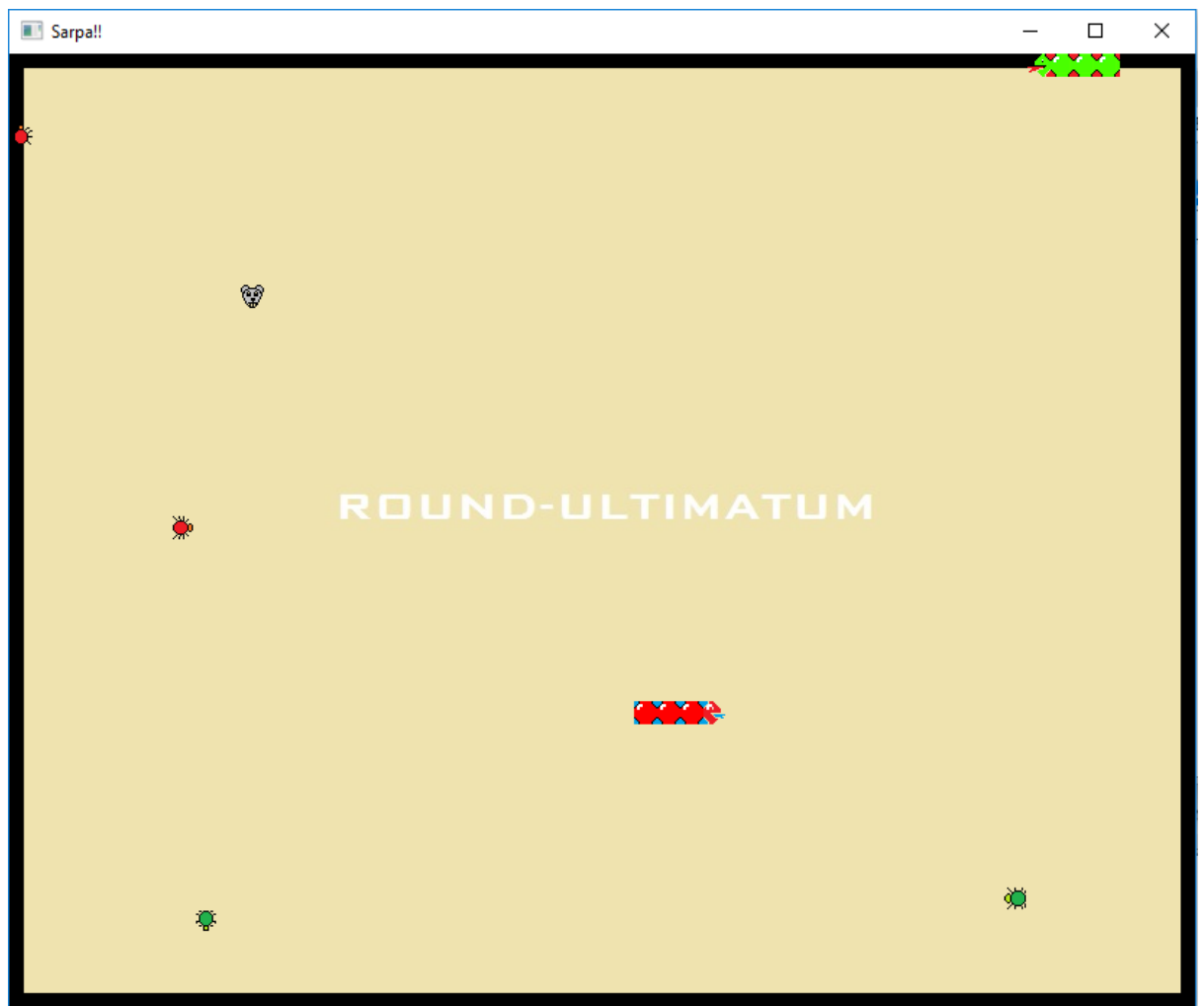


Figure: Round Ultimate Window (After GUI)



Figure: Red Rattle Wins

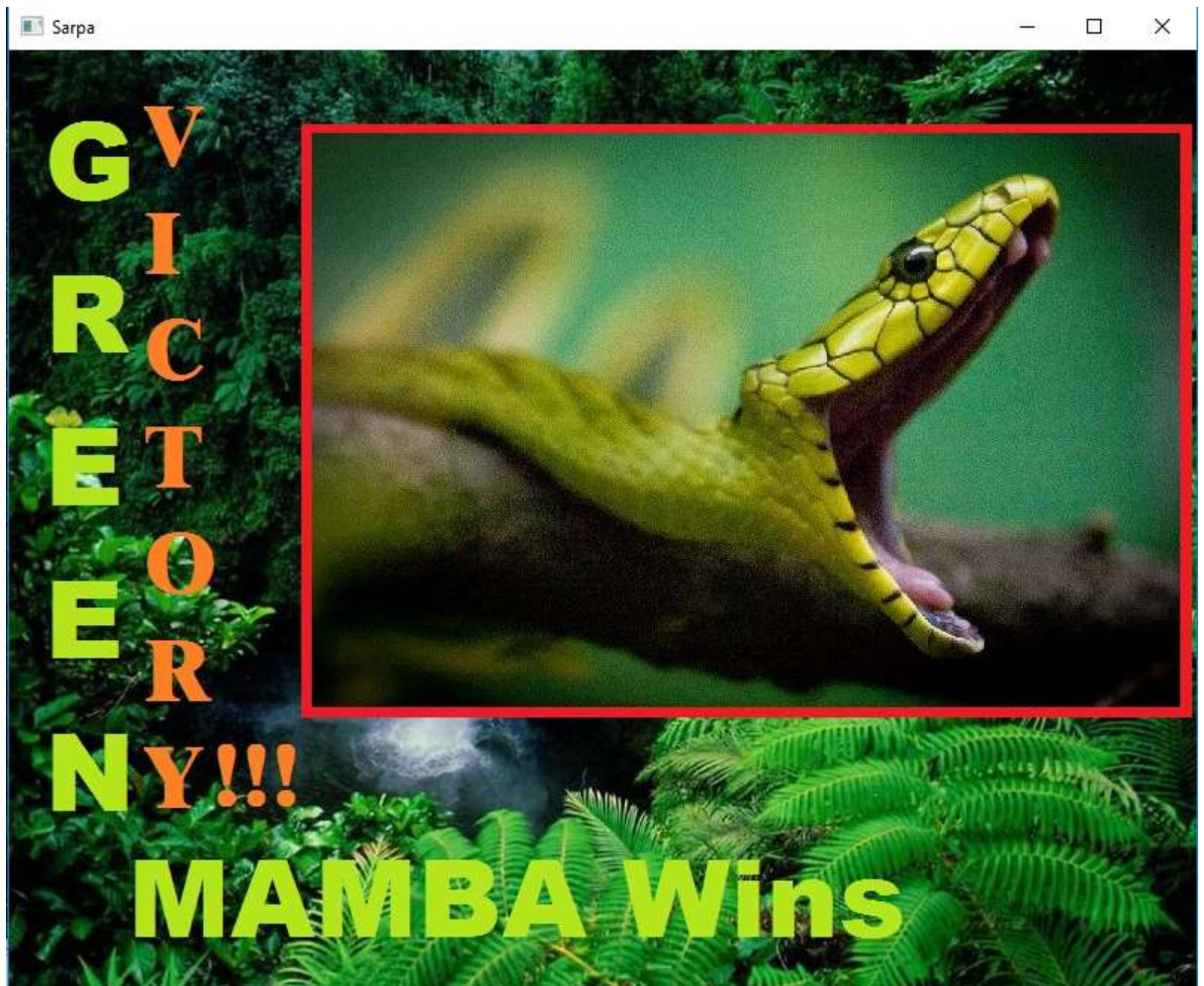


Figure: Green Mambaa Wins

4.3 Structures and Functions:

Predefined:

- `OpenFromFile()`:
It opens the music from respective file.
- `setLoop(boolean)`:
It takes true or false as an argument and loops the music file or not.
- `setVolume(int)`:
It provides the volume to the music.
- `play()`:
It plays the music audio.
- `Sprite name(&Texture)`:
It initializes the texture background to the sprite.
- `isOpen()`:
It opens the user-defined window.
- `pollEvent(&Eventname)`:
It processes the user-defined event in the window.
- `close()`:
It closes the window after all the events are over.
- `setPosition(int,int)`:
It initializes the position of the sprite taking position numbers in pixels as arguments.
- `draw()`:
It draws the sprite in the window.
- `getGlobalBounds()`:
It takes the position of the movement.
- `contains(event.x-coordinate,event.y-coordinate)`:
It calculates if the cursor position is under `getGlobalBounds`.
- `isButtonPressed()`:
It detects the button pressed on mouse.
- `isKeyPressed()`:
It detects the key pressed on keyboard.

- `display()`:
It displays the window.

User-Defined:

- `Clear()`:
It is a void function that initializes all the variables required during different stages of program back to zero.
- `Snake{}:`
It creates x and y coordinate for each block for snake for initializing position.
- `Food{}:`
It creates x and y coordinate for each mouse aka food.
- `SnakeCollision()`:
It detects collision of snake among each other.
- `Move()`:
It provides a movement for each cube of snake and detects self collision.
- `red()`:
It opens a new window that displays “Red Rattle Wins”.
- `green()`:
It opens a new window that displays “Green Mambaa Wins”.
- `bugs()`:
It creates the bug with x and y coordinates for initializing position.
- `preys()`:
It provides the prey movement.
- `preycollision()`, `preycollisionred()` and `preycollisiongreen()`:
They detect when snakes collide with bugs and return integer to count which player wins.
- `newwindow()`, `newwindowred()` and `newwindowgreen()`:
They open different windows for different game levels of ultimate, red and green rounds respectively.
- `Instructions()`:
It opens new window for providing help and instructions before game.

Chapter 5: PROCESS LOGS

Date	Problem Description	Action Taken	Comments
2017-05-25	Wasn't able to set the loop that described the movement of the snake.	Took help from 16 games in C++ of YouTube to learn the way.	Problem solved.
2017-06-08	Wasn't able to find the outcome of self-collision in snake.	Defined the function for reducing the length.	Problem solved.
2017-06-22	Wasn't able to start a new game after the three levels are over.	Defined a function to initialize various variables back.	Problem solved.
2017-07-06	Wasn't able to read audio.	Initialized music file with the help of classmates.	Problem solved.

Chapter 6: DISCUSSION

The results were very positive and we were able to achieve what we had thought of. We found the game very entertaining when we introduced it to our friends. We came to know the real power of coding through teamwork. We got the opportunity to know different practical application of programming language and its real life approaches in the world. Various factors in the game like background design, textures, sprites, collision, design and movement which were successfully done and implemented. Whereas, the various features of C++ like creation of classes and objects, inheritance, polymorphism were not implemented in the program.

6.1 Features

We believe the game “SARPA” is superior to the traditional snake game due to its following features:

- It includes all the basic features of the traditional snake game.
- The basic rules and concept are easy to understand so the game is compatible with people of all ages.
- It makes an involvement of two players at an instant thus helps in socializing.
- Since the game is a survival between two which makes it exciting to play.
- The background music triggers the mood for the players.
- The additional obstacle feature makes the game more challenging.

6.2 Limitations:

- User can't choose their snake's avatar.
- User can't change control functionalities.
- There is no provision of playing multiplayer by the use of internet or hotspot.
- Also, the idea of recording history records were unsuccessful.

6.3 Further Development

We can further develop our project by updating the game play, by adding some extra features like file handling to save the history records, adding pause feature in the game while also increasing some graphical elements. We can also add some many snake avatars as per the likeness of the user. We have also thought of establishing the online playing through internet hence making our project perfectly suitable to the real gaming world.

Chapter 7: CONCLUSION

This project game successfully provides a two player experience never found in previous 'Snake' games. It also provides three levels with advantages on either sides. However, the project can be improved upon with multiplayer mode, high-score management, controls and choosing different avatars for snakes to add on the user experience. Any recommendations from the users to make the game more entertaining are highly encouraged.

CONTACT INFORMATION:

Names: Prajwol Lamichhane, Arpan Mahatra and Abhay Raut Chhettri

Address: 28 Kilo, Kavre

Phone Numbers: 9841544244, 9843436289, and 9845792748

Emails: prajwollamichhane11@gmail.com

arpanmahatra@gmail.com

abhayraut712@gmail.com

BIBLIOGRAPHY:

- www.sfml-dev.org
- www.github.com
- www.gamecodeschool.com
- [www.wikipedia.org/wiki/Snake_\(video_game\)](http://www.wikipedia.org/wiki/Snake_(video_game))
- www.stackoverflow.com
- www.cplusplus.com

REFERENCE

1. Moreira, Artur, & Haller, Jan (2013), SFML Game Development, Birmingham -Mumbai: PACKT publishing ltd.
2. . Deitel, Paul, & Deitel, Harvey (2010), C++ How To Program, New Jersey, USA: PEARSON publication
3. . Lafore, Robert (1998), Object Oriented Programming In C++,
4. Balaguruswamy, E. (2013), Object Oriented Programming with C++, New Delhi:
5. McGraw Hill Education Private Limited