TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS

Project Report
‘DUEL Game Package’

SUBMITTED BY:

Suraj Pathak (076BCT090)

Tikaharu Sharma (076BCT094)

SUBMITTED TO:

Department of Electronics and Computer Engineering

ACKNOWLEDGEMENT

We would like to take this opportunity to express our immense gratitude to the department of electronics and computer engineering for providing us with this opportunity to explore more in the field of programming via this project.

We would like to express our sincere thanks to our lecturer Daya Sagar Baral Sir for giving us the proper knowledge of object oriented programming in C++.

At last, a special thank you goes to all our friends who have helped us during the making of this project.

ABSTRACT

The project “Duel Game Package” is a part of our academic session B.E.(Computer) Second Year First part as prescribed in the syllabus of Object Oriented Programming course. The main objective of the project was to develop a practical program that would provide student better understanding of the concept of Object Oriented programming and its importance for the development of real-world applications.

For this purpose, we have made a multiplayer game that contains three mini-games within it and the player with most game wins, is declared the winner of the series. We have used the SFML -2.5.1 library for graphics. The game contains attractive background, animations and different sound effects to make it more interesting.

TABLE OF CONTENTS

OBJECTIVES

The main objectives of this project are:

1. To learn about the object oriented approach and enhance our skills in C++ programming.
2. To learn the basics of game development and implementation of real-world scenarios as a part of your game logic.
3. To understand the use of a graphics library, particularly in the implementation of SFML Modules to create a 2-D game.
4. To be familiar with resource re-usability by using multiply linked files and inheritable classes.
5. To learn to work in a team to make us able to work in major projects in the coming future.

INTRODUCTION

Duel Game Package

Duel Game Package is a modified variation of widely known 2 player games that are quite entertaining. This is the 2d animated game written in C++ using OOP concepts and SFML library for multimedia. This game package comes with three games. 2 players battle against each other in three rounds with different games and try to outscore their opponents. The player with most game wins will be the winner of the series.

Ping Pong:

Ping pong is a table-tennis themed arcade sports video game featuring simple two dimensional graphics. The game rules are very similar to the original Pong. The right player scores a point if the ball hits the left wall. The left player scores a point if the ball hits the right wall. If it hits the top or bottom wall then it bounces off. Each player will have a racket that can be moved up and down to hit back the ball.

Space Race:

Each player controls a rocket. The idea is to get to the top of the map having avoided the randomly generated space debris or asteroids to gain a point. If the player's rocket collides with any asteroids along the way the rocket has to restart its journey. The player who completes 3 laps fastest wins the game!

Jungle Run:

This game is an endless runner game. The character continuously runs across a jungle avoiding obstacles on its way by jumping or sliding. The player can shoot stars to kill flying angry birds. If the character fails to avoid any obstacle the game ends. The player with most points wins the game.

Besides the game, there will be menu, how to play instructions, scorecard, result window, etc. There will be background images, animations, background sounds throughout the game to give better user experience.

APPLICATION

Duel Game Package is a multiplayer game package with three rounds of mini games. Typically, you'll sit at the same computer and use different keys to control your characters separately. Both players will try to outscore other to win the game. So, it will serve entertainment and challenge to the players who are seeking to play against their friends or family members in real time in a single device. The project has potential to be made more advanced by adding more fun games, realistic graphics, more options, online multiplayer mode and so on. The ping pong game is the replication of the real life ping pong game. Jungle run is an endless runner game and such games have engaged millions of users in the world. Different mathematical concepts were also used to apply various logic in the game. People sometimes tend to get bored playing only one type of game, that is why we have developed three different mini games in a single game. We believe such multiplayer game package still has large prospects in the gaming industry.

LITERATURE SURVEY

We referred to various books and resources for understanding the concepts behind game development using C++ and SFML. Our main reference for the OOP using C++ was "The secrets of object oriented programming" by Daya Sagar Baral. Using just C++ for graphics was not sufficient to make the game interactive so, SFML library was used. Our reference for SFML was “SFML Essentials: A fast-paced, practical guide to building functionally enriched 2D games using the core concepts of SFML” by Milcho G. Milchev.

We also referred the official SFML documentation and various online forums and discussion groups to better understand the implementation of SFML in our game.

EXISTING SYSTEMS

As of the present time, we can find various games in different online platforms similar to our three games. Ping pong game is quite popular as a casual sports game in both android and desktop devices. Space Race game with similar concept but with different names were popular as an arcade game. There exists similar Android game Crossy Road that has over 100 million downloads. The jungle run game is very similar with Google Chrome's T-Rex Dinosaur game and many other endless runner games.

METHODOLOGY

Our project is written in C++ programming language using object-oriented programming paradigm. We used Visual Studio Code as code editor and Mingw GCC compiler in Windows operating system. The program uses concept of "objects", which can contain data and code: data in the form of fields (often known as attributes or properties), and code, in the form of procedures (often known as methods). There is separate classes for bat, ball, player, enemy, star, plane, asteroids and other entities that are required in our game. The events for each object have been handled by the different member functions so that they form a final outlook working together simultaneously.

In order to create graphical interface, we have used Simple and Fast Multimedia Library (SFML). SFML, which is written in C++, is a cross-platform software development library designed to provide a simple application programming interface to various multimedia components in computers. We used SFML modules such as system, window, graphics, audio to implement our game.

The programming methods we've used can as summarized below:

- Discussion among team members and initial research. Scheduling the project and dividing the tasks among the group members.
- Learning required library (SFML)
- Initial coding for creating logic, including defining a clear-cut diagram of the classes required to fully implement OOP principles.

- Implementing the game logic with proper graphics and adding other necessary features such as menu, score system, result display, etc
- Finishing the game with animations and sound effects
- Execution, debugging and testing the entire application more thoroughly
- Documentation and preparing final report for the whole project

Classes Used In The Program:

- **Game**

  This is the class where the main game loop occurs. All the components of our game are drawn here. The textures, sound, players, font are loaded in the constructor of this class. The member function of game class run () is the main loop of our program.

- **Bat**

  This class defines the bat of ping pong. where the main game loop occurs. It contains different functions which creates the bat, controls the movement of bat, checks the boundary and draws the bat.

- **Ball**

  This class represents the ball of ping pong. It has various functions to move the ball, check boundary condition, check collision with the bat and to render the ball to window.

- **Plane**

  Plane class defines plane for Space Race game. It has data members and functions to initialize the plane, control movement, check boundary collision, render to window, get global bounds of the plane and reset its position on collision with asteroids.

- ## Asteroid

This class is for the asteroids of the space game. An array of this class is created which generates a lots of asteroids for hindering the movement of the plane. This class also contains several functions for the movement of the asteroids and checking for any collision with the plane

- ## Jungle

Thus is the main class for the jungle run game. This class contains the player and the enemy objects and controls the movement of the player and also sets the position of the enemy. The class also handles the scrolling background and updates the score of the player if there is any bird hit. Several other classes used in the jungle run game are:

1. Player
2. Enemy
3. Star
4. Bird
5. Saw
6. Cactus
7. Spike

- ## SpriteEntity, AnimationEntity, RectEntity

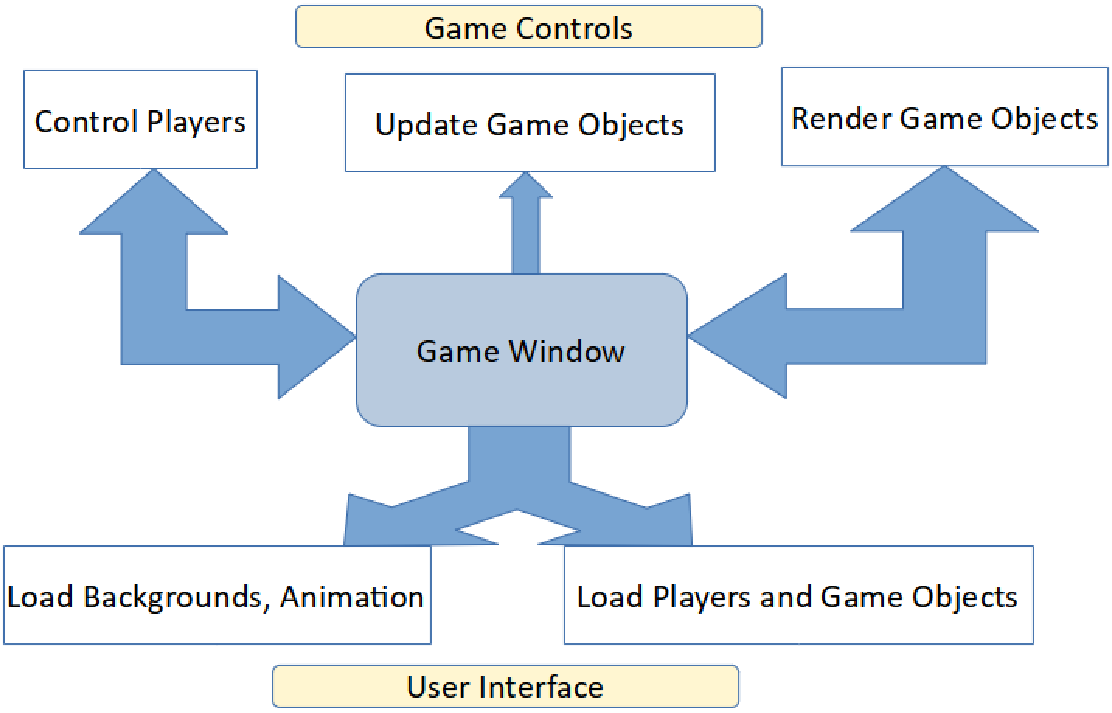
These are base classes which contains the framework required for a sprite or rectangle objects in the game. The contain data members for sprite, rectangle shape, texture, object width, object height, etc. There is also member function to render the object to the window. These are inherited by other classes such as bird, star, saw, etc.

- ## InitUI

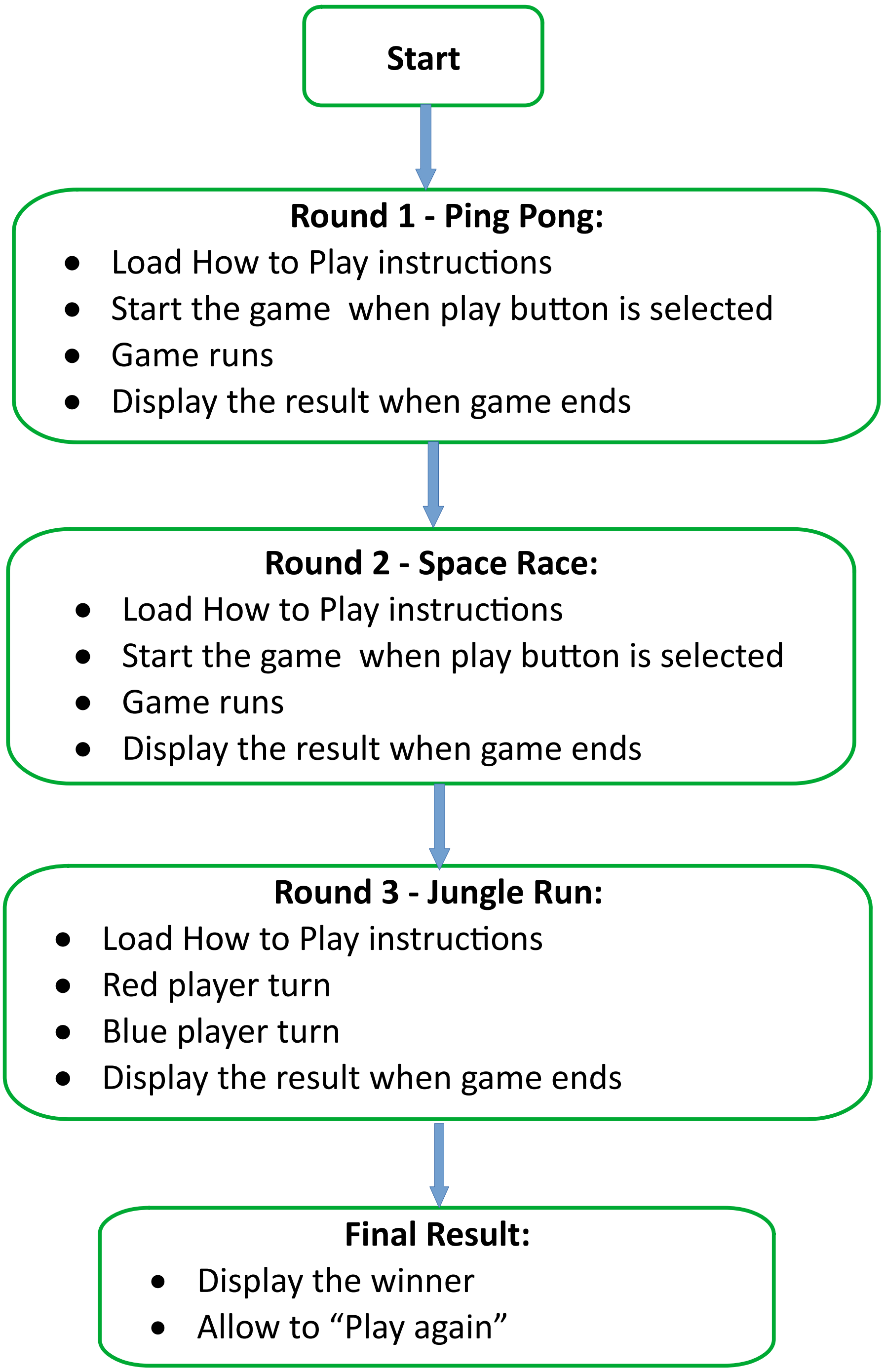
InitUi class contains methods to initialize text and texture with required size, color, position, etc. It is inherited by Game and Jungle class.

Implementation

System Block Diagram



DETAILED USER INTERFACE BLOCK DIAGRAM



Start

Round 1 - Ping Pong:

- Load How to Play instructions
- Start the game when play button is selected
- Game runs
- Display the result when game ends

Round 2 - Space Race:

- Load How to Play instructions
- Start the game when play button is selected
- Game runs
- Display the result when game ends

Round 3 - Jungle Run:

- Load How to Play instructions
- Red player turn
- Blue player turn
- Display the result when game ends

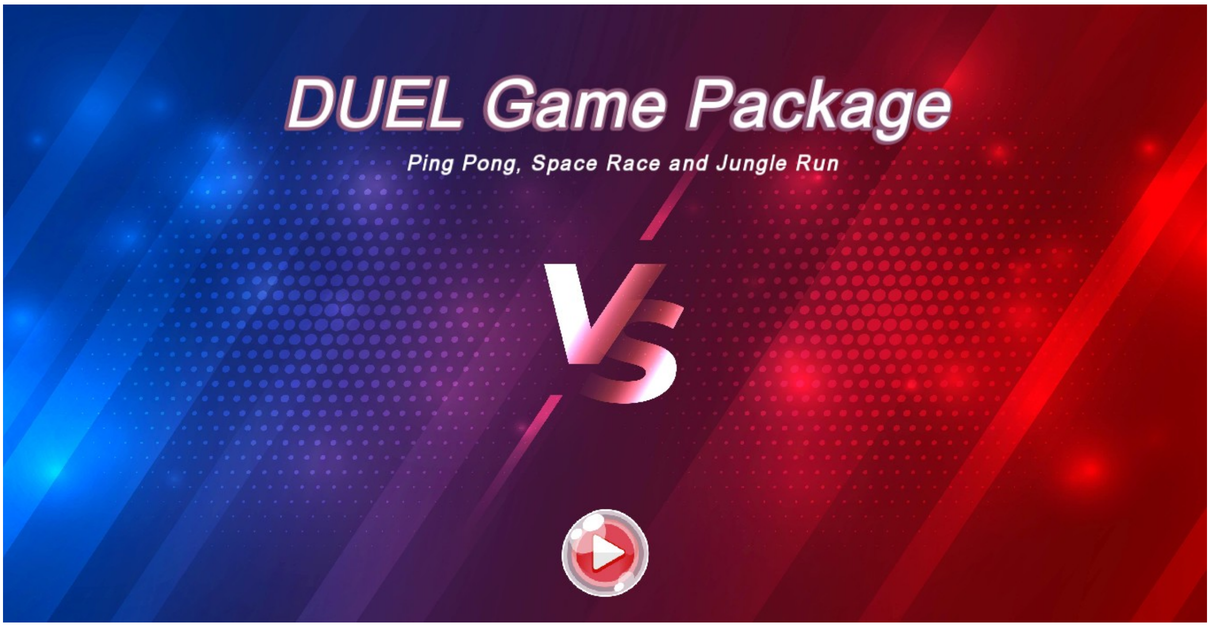
Final Result:

- Display the winner
- Allow to “Play again”

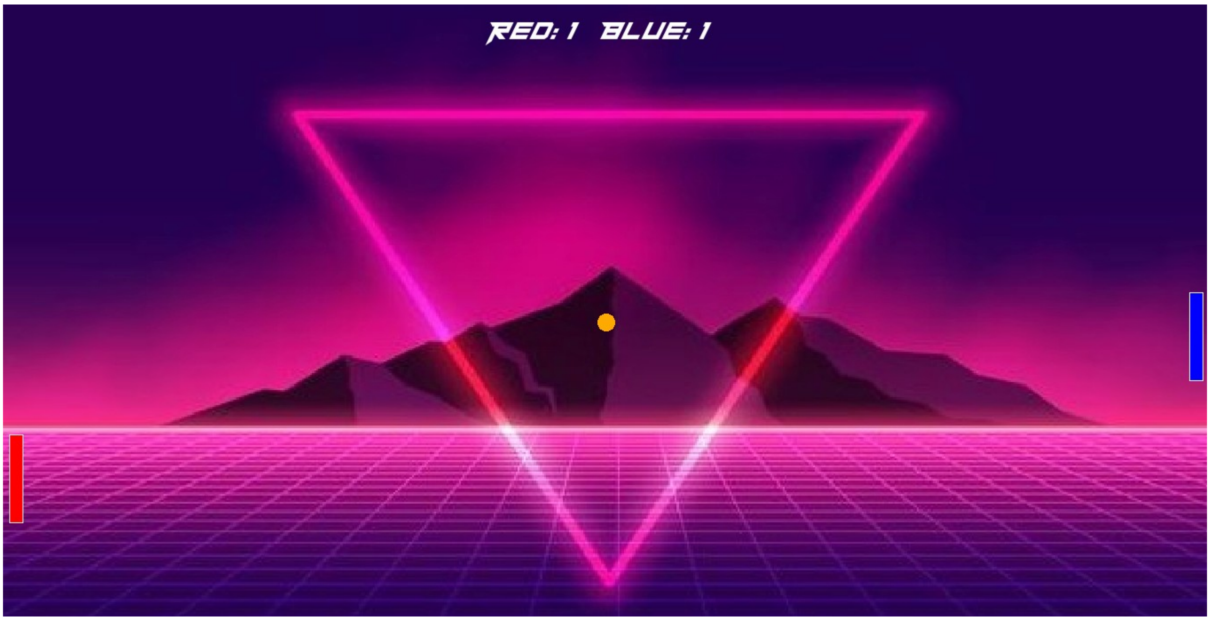
Results

We were able to complete the project by implementing the concept of OOP and Game development with SFML. We were nearly able to achieve almost all major objectives that we set during the proposal. It was challenging to implement the game logic, adding sprite sheet animation, integrate all components of the game and building the system for execution. We could have added more features and better user interface to the game but, were set-backed by our limited knowledge and lack of time. Finally, the project has been a success and this project has helped us gain a proper understanding of the concept of OOP and Game development with SFML.

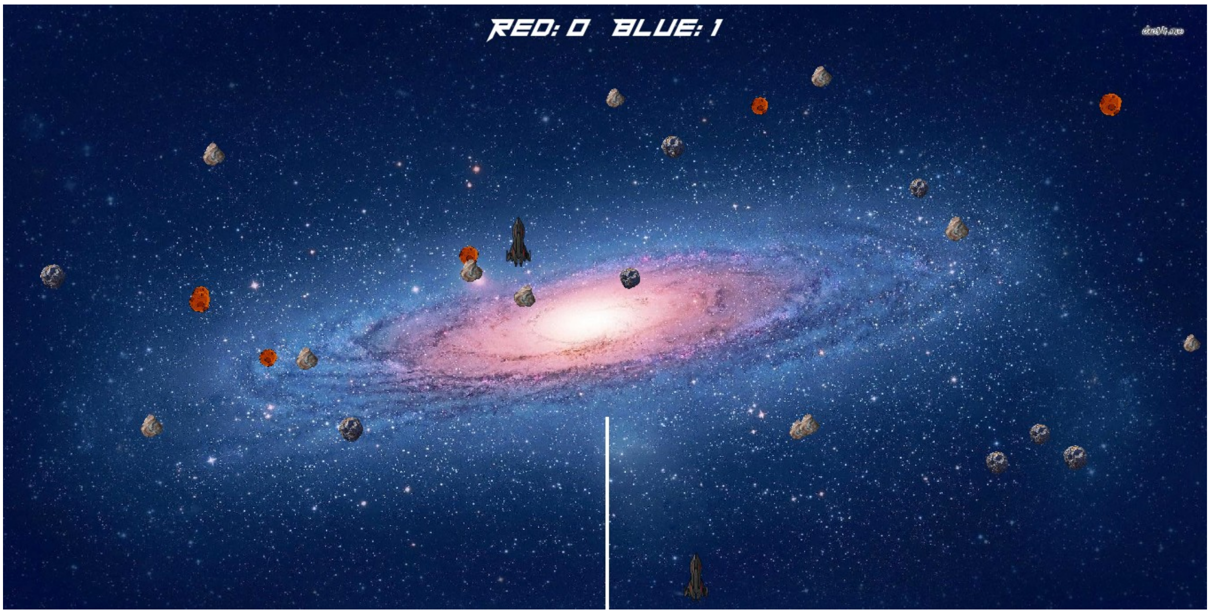
The Game Package has been developed with three variety of mini-games and decent user interface. The final result of the game can be illustrated by the following screenshots taken from different states throughout the game.



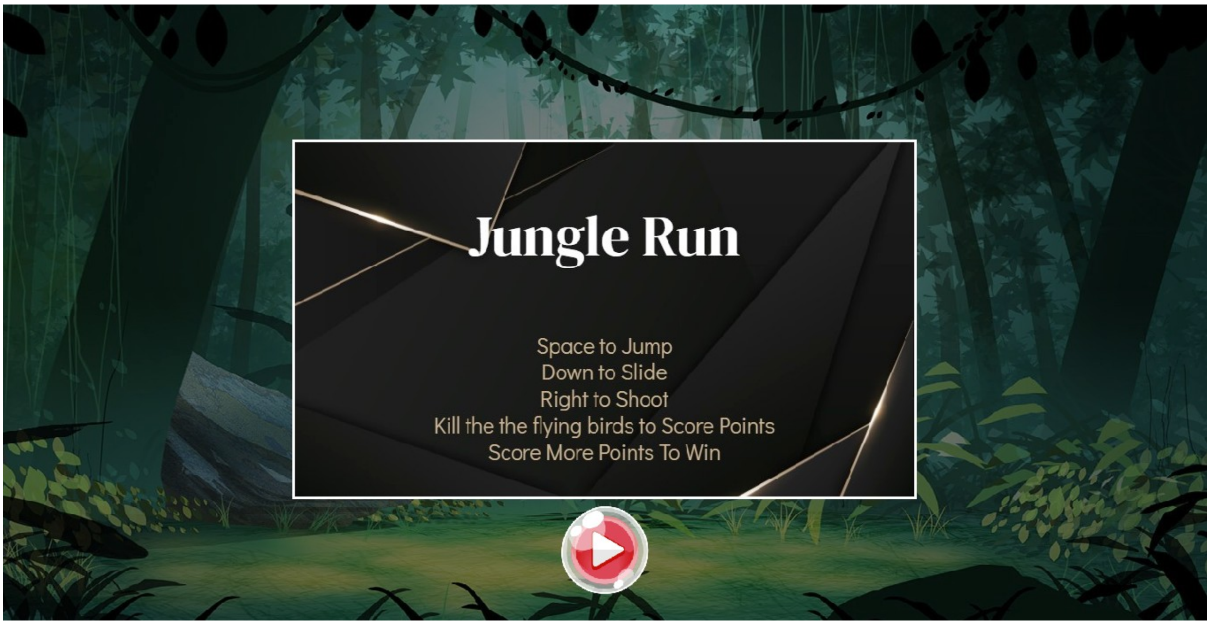
Home Screen



Ping Pong



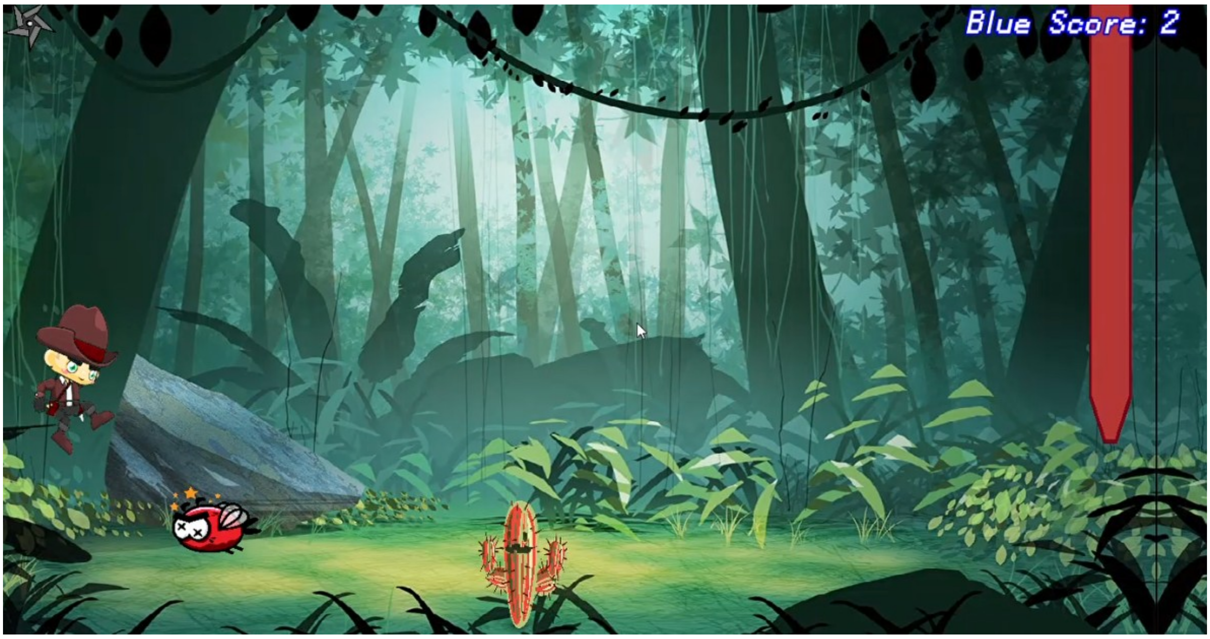
Space Race



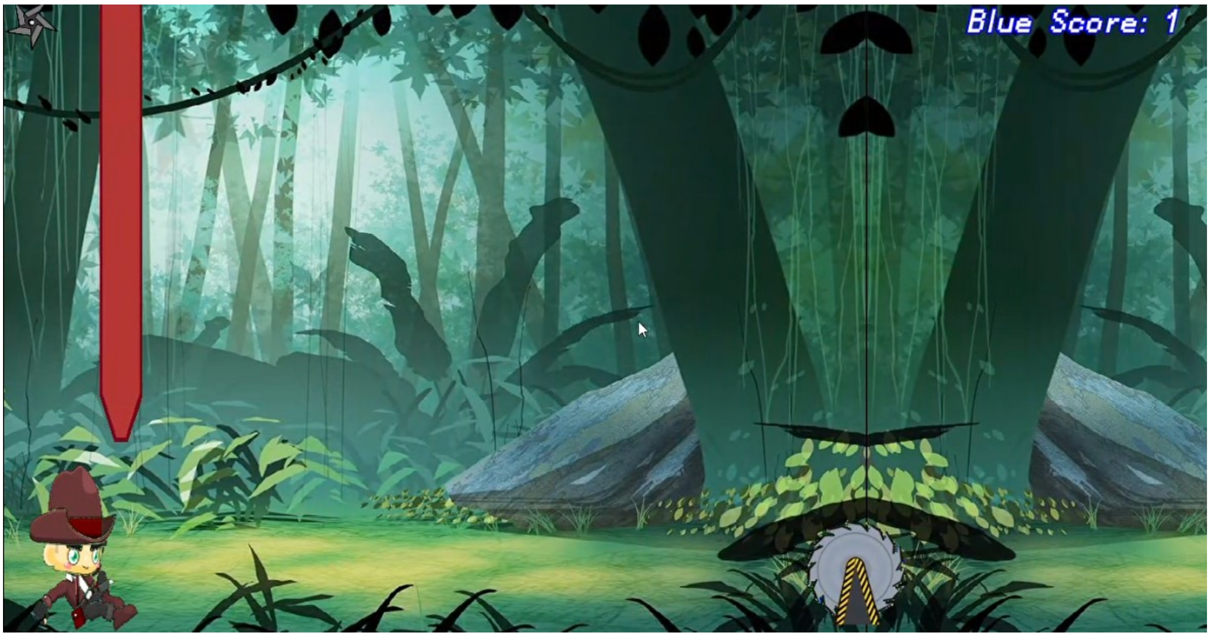
Jungle Run

Space to Jump
Down to Slide
Right to Shoot
Kill the the flying birds to Score Points
Score More Points To Win

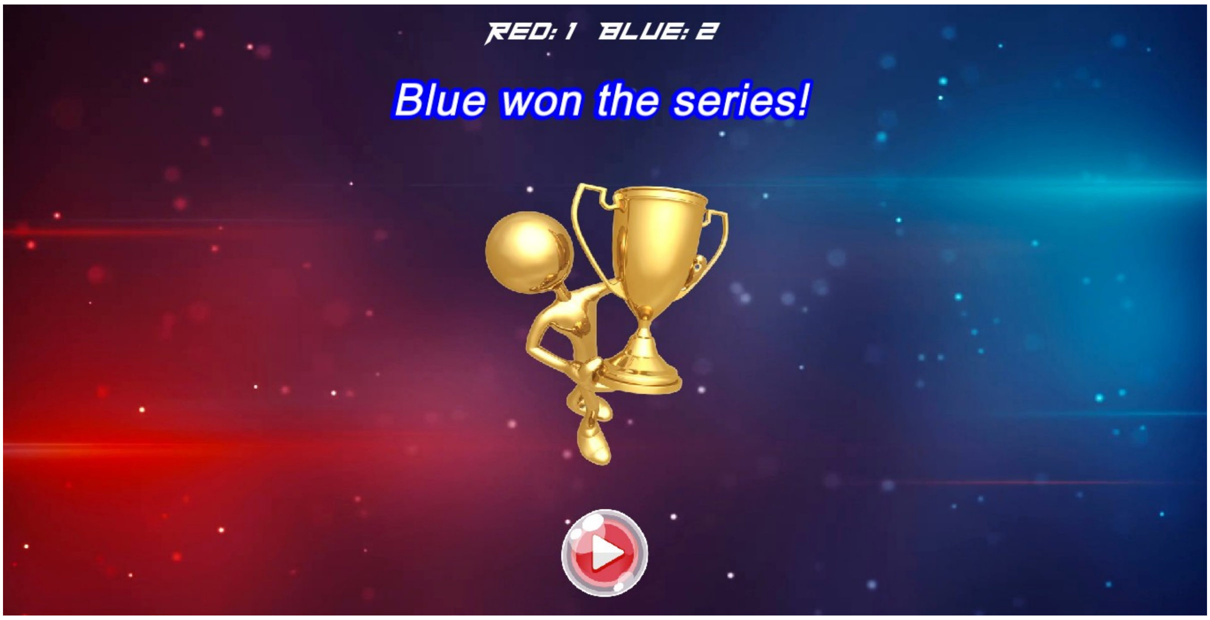
Jungle Run – How To Play



Jungle Run – Jumping Action



Jungle Run – Sliding Action



Game Result

Problems faced and Solutions

As we had no experience in game development and using graphics libraries in C++, we faced various problems as listed below:

- At first, we faced difficulty to configure the SFML library. There were many issues while setting up static libraries, dynamic libraries, to configure linkers and compilers.
- It was hard to find the right sprite sheets for the games and we had to manually edit some of these sprites to create proper animations.
- Problems occurred while implementing the game logic and phsyics for jumping, sliding, running and shooting actions. The animations looked unnatural and faulty and we had hard times correcting them.
- We faced many problems while merging the code from different members and putting it all together.

In order to tackle these problems, we searched in Google and found solutions to most of our problems in Stack Overflow. Besides, we sought help from the friends, discussion forums, official documentation, books and YouTube tutorials. Here are some solutions we applied to the above-mentioned problems:

- To setup the SFML library and linking multiple source files, we learned Makefiles, which helps building a software from its source files, and its compilation and linking.
- As two of us were working on this project, we used Git version control system and GitHub to track our progress and merge our codes together.
- After a lot of test, debug and hit-and-trial we found proper values for our game object’s co-ordinates, their speeds, animation switch times, etc.
- We had to manually edit the image size, backgrounds, color of the sprite sheets to generate the desired animation effect.

Limitation and Future Enhancements

There is always room for improvements in the game. Some of the limitations and the possible future enhancements are discussed below:

- SFML doesn't support 3D so we could choose other high class graphics like OpenGL, Direct3d, Vulkan to make the game 3d and more realistic.
- The game only supports multiplayer mode in a single device. The game can be enhanced by using networking to add online multiplayer mode. Likewise, single player mode can be added where single user competes against the computer.
- Currently, the game has 3 mini-games, so we can always add more games to the game package to give even more variety of game choices to the users.
- We can offer more control to the user to choose the game difficulty, speed, option to add player's name.
- The endless Jungle Run game has limited obstacles and less randomness in obstacles generation. This can be enhanced by adding more obstacles, including power-ups, and more weapons other than stars.

Conclusion and Recommendation

This project has helped us to understand the concept of object oriented programming and game development. The OOP paradigm allows to break the program into the bit-sized problems that can be solved easily (one object at a time).

Doing this project, we could realize the actual potential of object oriented approach as it provides re-usability, better security, productivity, code maintenance, easy troubleshooting, polymorphism flexibility and many more.

This project was a great learning experience for us as we successfully implemented theoretical knowledge into practice and succeeded in creating a real-world application. We learned graphics library (SFML) which is used to provide a simple application programming interface (API) to various multimedia components in computers. Additionally, we learned to work in a team to solve a common problem. Git version control system and GitHub proved to be quite essential to keep track of our projects and do collaborative work.

This project has taught us that there is more to software development than just coding. Only after proper planning, analysis, design, development and implementation, testing and maintenance, we can create a successful application.

References

- https://www.sfml-dev.org/
- “SFML For Beginners” – Hilze Vonck

  https://www.youtube.com/playlist?list=PL21OsoBLPpMOO6zyVlxZ4S4hwkY_SLRW9
- https://www.tutorialspoint.com/cplusplus/index.htm
- https://stackoverflow.com/
- SFML Essentials: A fast-paced, practical guide to building functionally enriched 2D games using the core concepts of SFML by Milcho G. Milchev (z-lib.org)
- “The Secrets of Object Oriented Programming in C++”- Daya Sagar Baral and Diwakar Baral, Bhundipuran Prakasan