Object Oriented Programming Lab

### Project Report



## Instructor

Saniya Zafar/ Muhammad Asad

## Submitted by

Faraz Ahmad Qureshi

200901045

Saba Taha

200901038

Jabeen-e-Zahra Hamdani

200901047

Bint e Zainab Laraib

200901106

Muhammad Ahmad Siddique

200901077

**Department of Computer Science,**

IST, Islamabad.

[17th May 2021]

# **Please be precise in all sections. Also, avoid to write the same statements again and again.**

# Abstract

We designed a videogame using C++, which is the industry standard for developing games. While most modern game engines (2d and 3d) are written in that language, our 2d game was built using the SFML library. It stands for Simple and Fast Multimedia Library. It is an efficient and easy way to use C++ and develop graphic applications.

# Introduction of your project

We have designed an infinite run and jump game where the goal of the player is to stay on one side of the screen and dodge obstacles that are coming their way by jumping over them. The main goal of the game is to survive the longest and attain the highest score possible. The game mechanics are simple and easy to understand and people of all ages can learn to play the game in a very short amount of time. The overall theme is playful and entertaining. It is meant to be addictive in nature and can provide hours of fun if one so pleases.

# Methodology/ Plan development to solve the Problem

We used Object Oriented Programming to plan out our game. We also used the features of Model View Control to distribute the application code:

We divided the code into several modules (OOP approach). Each part of the game (player, obstacle, text) was separated into an object (a separate class) .

We also distributed the code into a Model group (which contains the Functionality and logic) a View group (The render engine which renders everything to the screen) and a Control group (Code that controls the running of the game and its logic and keeps everything in sync). This separation was done using the fundamental concept of MVC designs.

Because of the complexity of the application, we decided to develop our code by separating objects into different class (.h/.hpp header files) and function/logic blocks (.cpp files) and then we used both the Update function (to control the logic) and render() function to control display to the screen.

# Specifications

Because the code makes use of SFML, the code will not run properly if SFML and its release files are setup improperly in your compiler.

# Code

https://pern-my.sharepoint.com/:u:/g/personal/200901045\_ist\_edu\_pk/Ecu-xMiDhU5BlxgETR6lBbMBz1W5laKkNzuxmpptd153rA?e=5JD82e

# Results Discussion

Idle-   


When hit-



Debug-



# Conclusion

We couldn’t implement the parallax effect for the background using our current knowledge of C++. For future improvements we could add a main menu, title screen and a player select screen where the player can select between different playable characters. We could also add sound to the game. We could also improve the score system by setting up files to record all the high scores (or like the top 10 highest scores) in the game (currently it only records the highest score).

We could also work on adding the ability of having multiple players in the game. And eventually we should be able to port the game over to the web and use online multiplayer system to make the game playable over the World Wide Web.

# 8. References

* The sfml class used to display the game window-   
  <https://www.sfml-dev.org/documentation/2.5.1/classsf_1_1Window.php>
* tutorial to setup a window-  
  <https://www.sfml-dev.org/tutorials/2.5/window-window.php>
* Texture Class responsible for all the texture/images used in the game  
  <https://www.sfml-dev.org/documentation/2.5.1/classsf_1_1Texture.php>
* Setting up SFML with visual studio-   
  <https://www.sfml-dev.org/tutorials/2.5/start-vc.php>