Abdullah Belosokar

H00478897

Game project

**INTRODUCTION**

In the following report, the development and explanation of the game project shall be explained in detail. The github link for the game is linked here: https://github.com/BigBeloski/GAMEPROJECTDEMO

The project will cover the following:

The design, game mechanics, game physics, code development, player controls, character sprites, scoring mechanics, design elements and final product.

The game is a 2-d style, top-down arena shooter game like Vampire Survivors, HELLDIVERS, Metal Gear Solid, Ruiner etc.

The game is supposed to be simple proof of concept only, it is not in any way finished, or to be released officially. It was created as a template for a 2-d top-down arena shooter, the game was made by making modifications to the template. The template used and the final product will be shown later in the report.

The mechanics of the game are simple and easy to understand. You control your character which is represented by a blue spherical sprite, using w, a, s and d to move forward, strafe left, strafe right and backward like in a typical game. To shoot with a weapon, you are required to click or hold left click on your mouse. You have the option of choosing between three different weapons, a pistol, shotgun and submachine gun, which can be equipped by pressing 1,2 or 3 respectively. The weapons all fire differently and imitate the way a real weapon would work. The goal of the game is to score as many points and survive as long as possible. The enemies are represented by green spherical sprites, they inflict 10 damage to the players health when they collide with the player. If the player loses all their health points, the game over screen appears and the player loses the game.

**DEVELOPMENT**

The game was developed in the C language as instructed and made using an external graphical library like Raylib to develop the game. To code the game, the provided source code editor Notepad++ for raylib was used as it comes with the raylib library preconfigured and makes it easier to replicate and run the code without much hassle on other systems. The game was made using generative AI to create the base template, this way we were able to get a rough idea of using raylib libraries and we were able to modify the template to create a better more finished product with added features like added obstacles, scoring system, health points, player collision, projectile collision with enemies, weapon projectile collision system and more graphical improvements.

**GAME MECHANICS**

Using simple and easy to implement game mechanics, we were able to implement collision physics and game logic like bullet collision and damage to the enemies and players. The bullet projectile physics is handled by the HandleShooting() argument where you input a float value in the ShootBullet() argument to control the speed of the projectiles fired by the weapon. The rest of the game mechanics are handled by similar built-in functions and arguments included in the raylib library. The function of each argument and code is commented

**CONCLUSION**

Developing games in different programming languages requires knowledge of using different external libraries as we are limited by the lack of using these external graphical libraries to just text-based games or ASCII code-based graphics. Hence, it is important to consider using external libraries when developing code to ensure ease of implementation and maximum compatibility with the program.