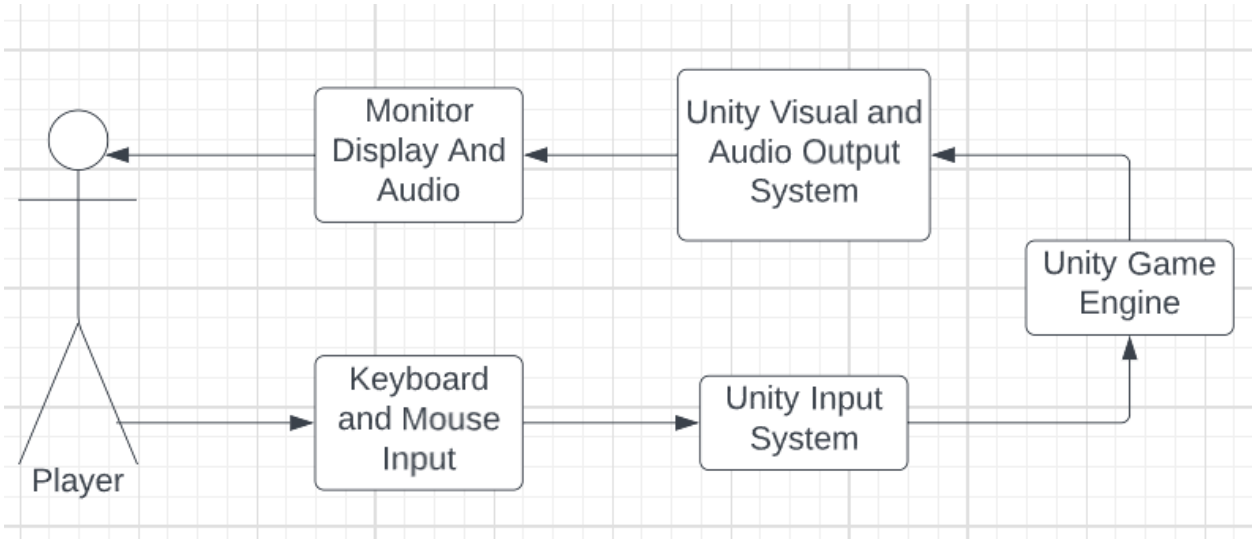
Zombie Game-FPS style (Unity based game)

URL(idea): Zombies_Game_idea Date: 17th September 2022

Team: Anant Singh, Adil Haijev , Calvin Zhuang
email : _anant@u.pacific.edu, a_hajiyev@u.pacific.edu , c_zhuang@u.pacific.edu

1. System Architecture



The software used to build this game is Unity, which is also the game engine of the application. With respect to unity, the system's(desktop) audio output will be accessed to create visual effects for the application. The User required a keyboard and a mouse to play the video game.

2. Hardware,Software and System Requirements:

Software & Hardware requirements for editors:

Minimum requirements	Windows	macOS	Linux (Support in Preview)
Operating system version	Windows 7 (SP1+) and Windows 10, 64-bit versions only	High Sierra 10.13+	Ubuntu 16.04, Ubuntu 18.04, and CentOS 7
CPU	X64 architecture with SSE2 instruction set support	X64 architecture with SSE2 instruction set support	X64 architecture with SSE2 instruction set support
Graphics API	DX10, DX11, and DX12-capable GPUs	Metal-capable Intel and AMD GPUs	OpenGL 3.2+ or Vulkan-capable, Nvidia and AMD GPUs.
Additional requirements	Hardware vendor officially supported drivers	Apple officially supported drivers	Gnome desktop environment running on top of X11 windowing system, Nvidia official proprietary graphics driver, or AMD Mesa graphics driver. Other configuration and user environment as provided by default with the

			supported distribution (Kernel, Compositor, etc.)

The applications would run on a single machine, so no network protocol is required.

The application would be made only for pc gaming, so It will not work for other devices.

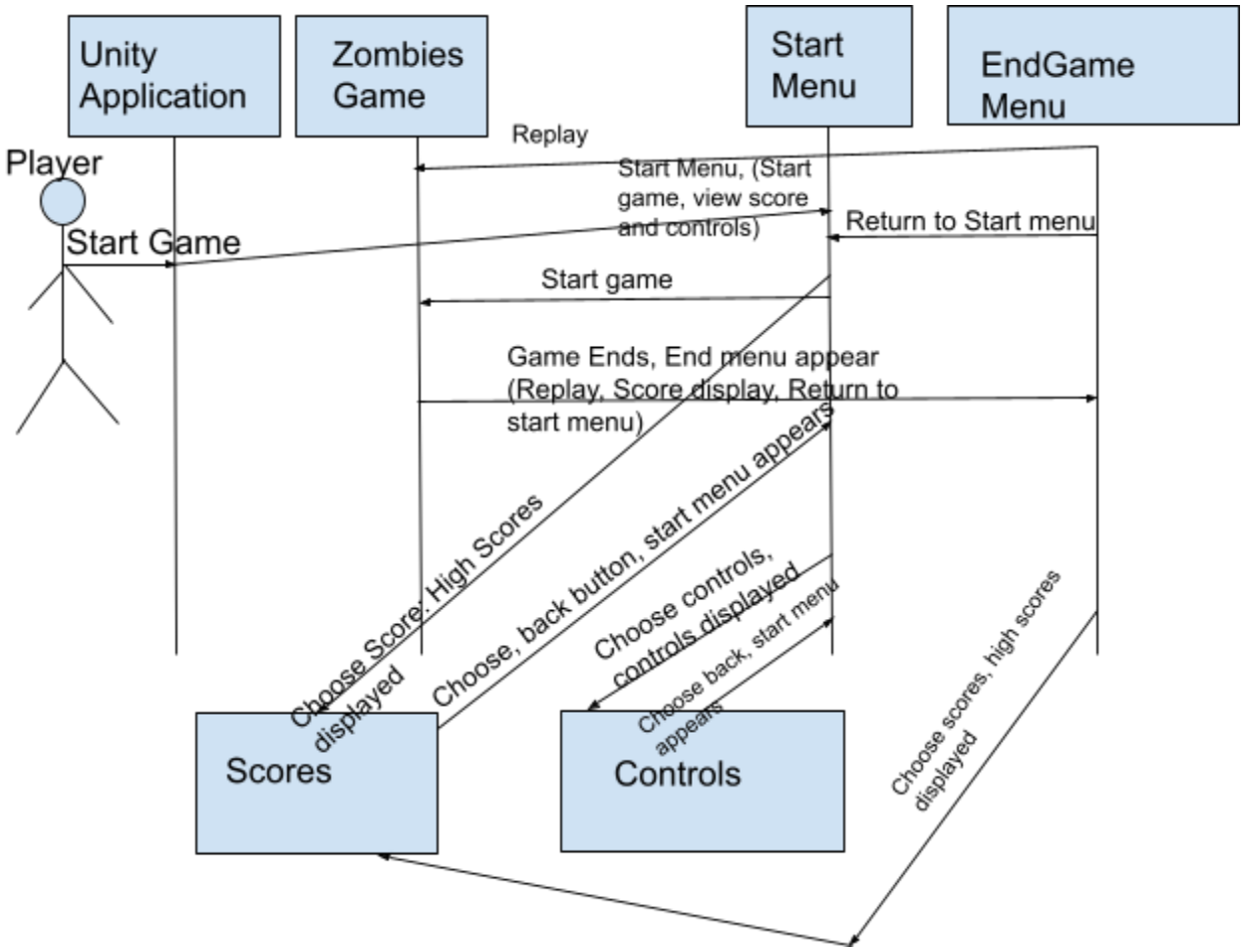
Software and Hardware requirements to run the application:

Operating system	**Windows**	**Universal Windows Platform**	**macOS**	**Linux**
Operating system version	Windows 7 (SP1+) and Windows 10	Windows 10, Xbox One, HoloLens	High Sierra 10.13+	Ubuntu 16.04 and Ubuntu 18.04
CPU	x86, x64 architecture with SSE2 instruction set support	x86, x64 architecture with SSE2 instruction set support, ARM, ARM64	x64 architecture with SSE2	x64 architecture with SSE2 instruction set support
Graphics API	DX10, DX11, DX12 capable	DX10, DX11, DX12 capable GPUs	Metal capable Intel and AMD GPUs	OpenGL 3.2+, Vulkan capable
Additional requirements	Officially supported hardware vendor drivers	Officially supported hardware vendor drivers For	Apple officially supported drivers	- Gnome desktop environment running on top of X11 windowing

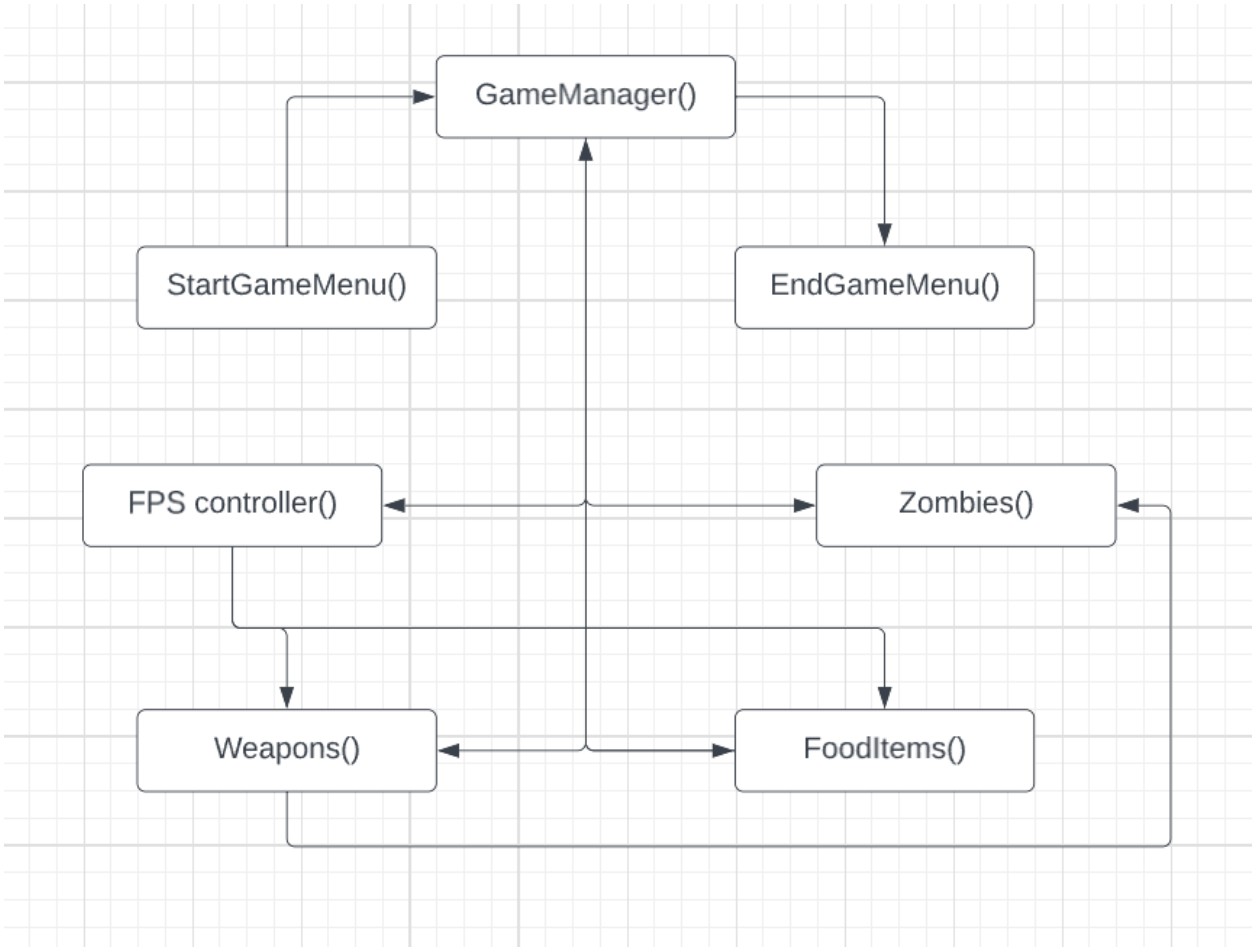
		development: Windows 10 (64-bit), Visual Studio 2015 with C++ Tools component or later and Windows 10 SDK		system - Other configuration and user environment as provided by default with the supported distribution (Kernel, Compositor, etc.) - Nvidia and AMD GPUs using Nvidia official proprietary graphics driver or AMD Mesa graphics driver

3. Software Design:

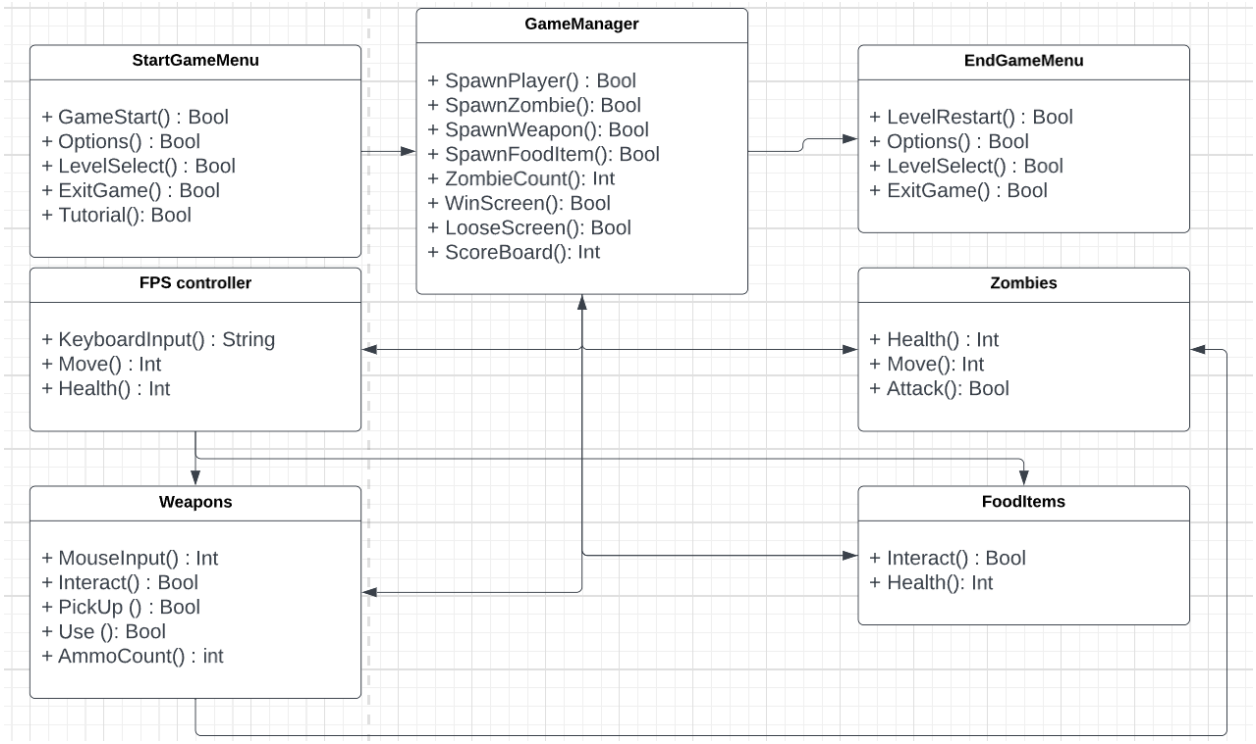
Interactions Diagram:



Class Diagram:



Class Specifications:



Design considerations:

The principle used to design the classes for the application is OOP (Object oriented Programming), to make the scripts readable and every complexity will be handled in terms of classes, similar to Java programming, to make the code reusable and avoid any forms of redundancy.

4. User Interface:

The initial prototype would only feature one wave of zombies and 2 weapons, one primary and the other one you can acquire in the wave, and food items that will spawn randomly, which would help the player to replenish its HP.

When the use starts the application, it will see a start menu which would have three options:

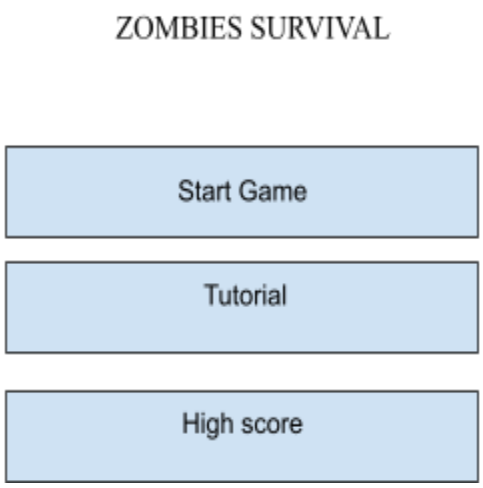
1. Start game: If selected the game will start
2. High scores: It will be a menu of all the highscores recorded, till the date, with the highest being on top
3. Controls/Tutorial: It will explain how the video game will be played

Once the game starts there will be first wave, with zombies attacking the player and player has to shoot them to survive and when the game ends the GameOver menu will appear with the following options:

1. Replay: To restart the game
2. Return to Menu: It will take the user to the start menu
3. High score: This will be displayed right below the game-over message.

The user interface will be very friendly and will not be very complex, a mouse and keyboard directions W,A,S and D or arrow keys may be used to play. The mouse will be used to aim and shoot and the directional keys (W A S D or the arrow keys) to move.

Start Menu Prototype



End Menu Prototype

High Score Table
Prototype

High Score

1. High score 1:
2. High score 2:
3. .
4. .
5. .
6. .
7. High Score 7:

Highest score:

5. Glossary of Terms

1. HP - Health Points
2. FPS - First Person Shooter
3. Int - Integer
4. Bool - Boolean
5. String - Combination of characters
6. OOP: Object Oriented Language
7. GameManager(): Class that manages the whole game
8. StartGameMenu(): Class that handles the start menu
9. EndGameMenu(): Class that handles and checks if the game has ended
10. FPSController(): Class that handles input from user to make the player move and shoot
11. Weapons(): Class that handles various weapons and their spawns in the game
12. FoodItems(): To handles the food and manage health of the player
13. Zombies(): Class to deal with enemy and their health points.

References:

1. https://docs.unity3d.com/2020.1/Documentation/Manual/system-requirements.html
2. https://en.wikipedia.org/wiki/Call_of_Duty:_World_at_War_%E2%80%93_Zombies
3. www.lucidchart.com