International Journal of Recent Trends in Technology and Engineering
ISSN [O]:2832-4277 Volume-II Issue-II April-June-2023

The Zombie World

Mithun Selvan J[1], Mahalakshmi L[2]

[1]Student, Department of Computer Applications, Hindusthan College of Engineering and Technology, Coimbatore

[2]Assistant Professor, Department of Computer Applications, Hindusthan College Of Engineering and Technology, Coimbatore. mahalogu@gmail.com

Abstract

This game project is the story mode of a survival game which is a kind of Third Person Shooting Game. The third person is a view mode of the game. The game is named: "THE ZOMBIE WORLD". Here the player can play the game using the character of a Human being who secures a girl from the zombie. Here the zombies are described as an enemy. We used Unity 3D game engine for developing a game. We have chosen C-sharp as a programming language, It was one of the scripting languages. Moreover, this project was developed using the platform Unity 3D 2022. The new generation of people give so much attention to playing games for entertainment and there is a remarkable tremble in all types of generation to play Action game, most young people expend a lot of time playing a game. We tried to do something more attractive to gamers.

Keyword: 3D Game, Zombie Game, Unity, Shooting Game, Rescuing Game.

Objective

The project objective is to develop a game to entertain people. The game is about how to save a girl from a zombie. It has several levels. On each level, there will be an obstacle. The level of complexity increases with the game level. It was alsodeveloped to motivate others to make their own game to get skilled, Make interest in game development, and Make resources of game development assets.

Existing system

2D games use flat graphics, called sprites, and don't have three-dimensional geometry. They're drawn to the screen as flat images, and the camera (orthographic camera) has no perspective. Platformers have dominated the 2D scene since the introduction of the 1981 arcade classic, Donkey Kong - the first platformer ever made. Platformers are all about jumping and climbing to explore the environment. While this is the oldest form ofanimation, many gamers still enjoy the cartoonish nature of 2D as 2D

animationdoes not try to imitate real life.

Disadvantage

- **IT LOOKS BORING:** Traditional animation can sometimes seem to be boring.
- **LESS DEMAND**: With the growth of 3d animation most people prefer to watch 3d animation movies as compared to 2d.
- **ECONOMIC REASONS**: In certain cases, you can create 3d animation with less money and me due to modern technology.
- **TIME IS MONEY**: It is time-consuming to create 2-dimensional based animation templates.

Proposed system

3D games focus on three-dimensional graphics and gameplay. This makes themthe most complex in terms of depth and graphical power. They can cover virtually any genre, most notably racing, and first-person shooters. Of all the online games, 3Dgames come the closest to big titles typically found on Playstation, Xbox, or PC. Gamers enjoy 3D animation as it gives an experience of reality. Players feel that whatever happens in the game is real as the 3D design looks, behaves, and feels realistic. The gamers get to enjoy more realistic action in 3D.

Advantage

- **MOTION COMMUNICATION**— 3d animation has a greater and superior ability to portray movement.
- **VISUAL APPEAL**—3danimation is much moreappealing and realistic.
- **TIME IS MONEY**— You can use 3d models made for a particular project in future. it helps to lower the cost of production.
- **GOOD QUALITY**—3d giveshigh quality and moregameplay compared to 2d.
- **IN DEMAND**—Most people now prefer 3d rather than 2d.

System specification

Hardware specification

- 1.4 GHz Processor
- 500 MB RAM
- Intel Motherboard
- 6 GB HDD
- 104 Key Standard Key Board

System specification

- Operating System: Windows 11
- RAM: 8 GB
- Tools: UNITY Engine
- Client: PC
- Code: C# scripts

About front end

UNITY Engine

Unity is a 2D & 3D game engine that has been around since 2005. Developed by Unity Technologies, it was made to provide more developers access to game development tools, which in those days was a novel venture. For its long life, the engine has changed and expanded dramatically, managing to keep up with the latest practices and technologies. Even today, the game engine's main focus is to both provide the most robust set of tools possible for the game development industry, as well as make it as easy as possible for game developers of any skill level to use the engine (yes, including beginner developers). They've also expanded their reach into other industries as well with a huge focus on real-time 3D development, making it one of the most powerful engines available. The only thing limiting what you can make in Unity is performance and yourimagination. Do you want to make RPGs, survival games, and platformers? Unity can do that. Do you want to make an animated short film? Unity can do that. Do you want to make the next VR hit to help schools by providing kids with new ways to learn? Unity can do that. This is one of those times where the sky is the limit, as there are just endless opportunities and tools that have made Unity a popular engine for every industry. There are a ton of popular games you probably don't realize were madewith the engine. Unity gives users the ability to create games and experiences in both 2D

and 3D, and the engine offers a primary scripting API in C# using Mono, for both theUnity editor in the form of plugins and games themselves, as well as drag and drop functionality. Before C# being the primary programming language used for the engine,it previously 9 supported Boo, which was removed with the release of Unity 5, and a Boo-based implementation of JavaScript called Unity Script, which was deprecated in August 2017, after the release of Unity 2017.1, in favour of C#.2D world renderer. For 3D games, Unity allows specification of texture compression, mipmaps, andresolution settings for each platform that the game engine supports, and provides support for bump mapping, reflection mapping, parallax mapping, screen space ambientocclusion (SSAO), dynamic shadows using shadow maps, render- to-texture and full- screen post-processing effects. Two separate render pipelines are available, High Definition Render Pipeline (HDRP) and Universal Render Pipeline (URP, previously LWRP), in addition to the legacy built-inpipeline. All three render pipelines are incompatible with each other. Unity offers a tool to upgrade shaders using the legacy renderer to URP or HDRP.

Supported platforms

Unity is a cross-platform engine. The Unity editor is supported on Windows, macOS, and the Linux platform, while the engine itself currently supports building games for more than 19 different platforms, including mobile, desktop, consoles, and virtual reality. Unity 2020 LTS officially supports the following platforms:

- Mobile platforms iOS, Android[55](Android TV), tvOS;
- Desktop platforms Windows (Universal Windows Platform[58]), Mac,[10] Linux; Web platform WebGL;
- Console platforms PlayStation (PS4, PS5), Xbox (Xbox One, Xbox Series X/S), Nintendo Switch, Stadia; Virtual/Extended reality platforms Oculus, PlayStation VR, Google's ARCore, Apple's ARKit, Windows Mixed Reality (HoloLens), Magic Leap, and via Unity XR SDK SteamVR, Google Cardboard.

About back end

C# with must be used to develop the system because most web-based systems are developed with this combination. Following are the reasons for using C# in our project:

Why C#?

Rapid Application Development

In the year 1970 to 1990 computer technology was prevalent only in research centres and universities where programmers struggled to reduce 1KB of memory usage for months. Now the wheel of technology has taken up a turn, personal computers have brought a revolution computer in every home. But

the number of programmers required to meet the ever-increasing number of applications isinsufficient and their development time isvery important as compared to computer resources like 1GB of ram which is so cheap nowadays. C# is a dynamic interpreted language which thus reduces the programmer's time to compile and deploy.

Cloud Ready

As C# was designed for the web and cloud isthe metaphor for the Internet needless to say

C# is a client-server-based Cloud Language. Many big Cloud Service Providers provide C# hosting in their environment.

Loosely Typed

Learning C# is very easy unlike complex languages and it doesn't mean without complexity you cannot make large applications without it. The basics of software engineering lie in presenting simplicity. When learning C# you don't need to take care of it or float it is handled by C# Dynamic typing that automates the allocation of data types to your data structure.

Open Source

C# itself is open source, no organization can claim C# is theirs. The complete code of C# is open and anyone can use it free of cost thus. Many beginners ask me if Security would be a problem then, to them this is the same thing as you or any developer doing your work alone in a room and an open environment watched by thousands of others. Platform Independent C# is true.

Platform Independent

Where developers can write their programs on a windows machine and run them on a Linux machine. The power of the Platform independence also does not come at a cost of performance.

Requirements Gathering

Interviewing the company employee will be used as a method to gather information and requirements about the project routines and activities.

PROJECT DESCRIPTION

MODULE DESCRIPTION

Scene 1:

In this scene, a main menu will display and it has two options play and quit. When the user clicks the play button it goes to scene 2. When the user clicks the quit button it closes the application.

Scene 2:

When a user clicks the play button, it shows some scenarios of the game. The scenario explains the testing of a T-Virus testing the virus, reacting with the DNA, and the testing went wrong and it shows how the human being's behaviour changed to look like a Zombie. It also has a skip button and ifthe user clicks the skip button it goes to the next scene.

Scene 3:

Scene three displays Welcome to our game in this scene the user is the main character ofthe game and so the user plays a main role. This game has many obstructions to saving a girl from the zombies and it defines how to escape from the forest.

Scene 4:

This is the last scene in which he escaped from the zombies and saved the girl. They use a helicopter to fly over from the zombie world and safely handed the girl to their parents.

Flow diagram

A flow diagram is a visualization of a sequence of actions, movements within a system, and/or decision points. They're a detailed explanation of each step in a process, no matter the level of complexity of that process. Flow diagrams, also known as flowcharts, are powerful tools for optimizingthe paths - or flow - of people, objects, or information through a system or procedure. Flow diagram meaning comes from the connectors and symbols working together to create a visual representation of the direction of movement and what's needed to make that movement happen. The flow diagram shown in Figure 1 explains the process of the system.

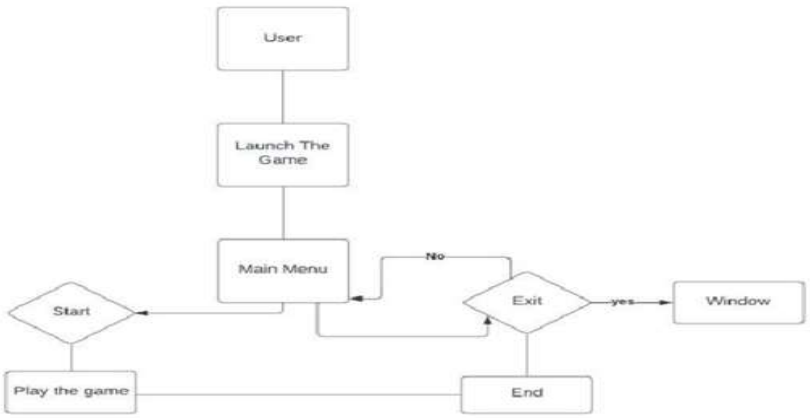

Figure 1: Flow Diagram

Sequence diagram

This is a simple sequence diagram that shows a game flow. The scenario begins when the player chooses to start a new round in the UI. The UI asks whether any new players want to join the round; if so, the new players are added using the UI. The project flow is explained in Figure 2,3,4, 5 and 6.

Main Menu:

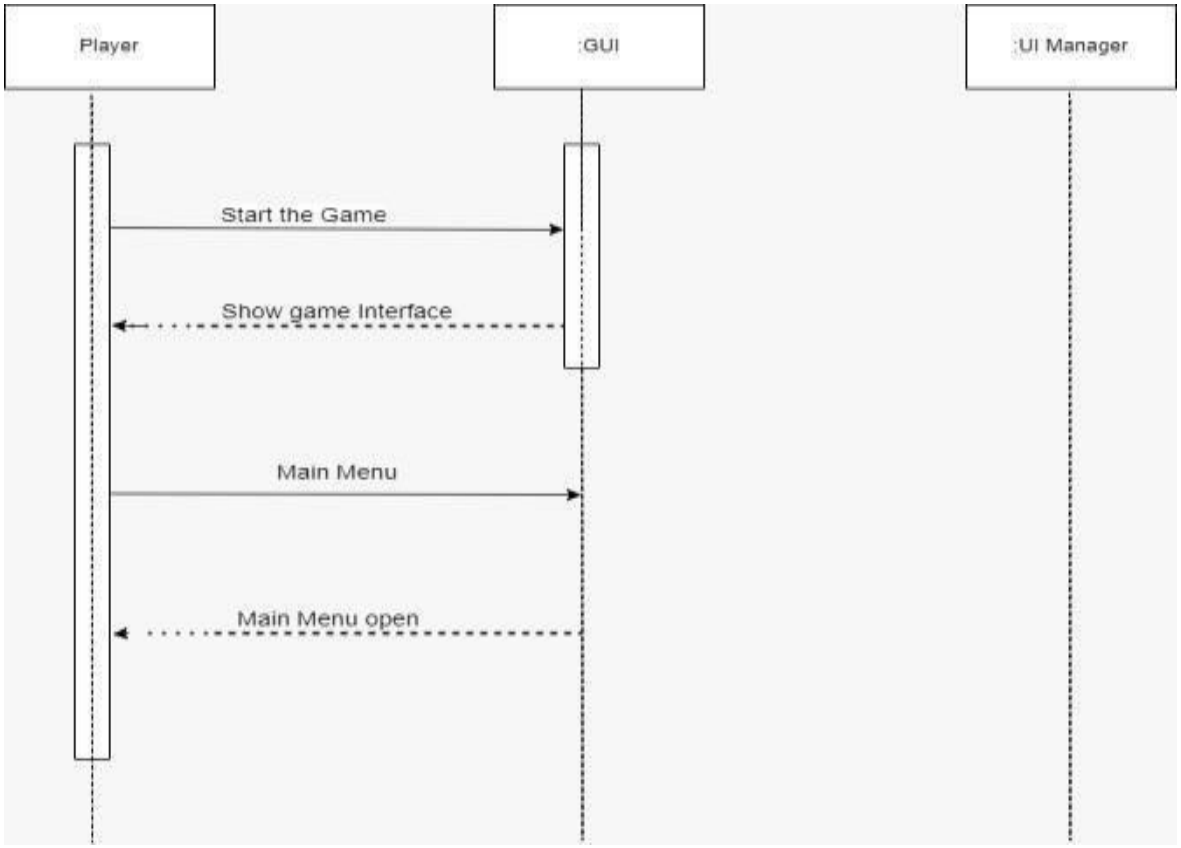

Figure 2: MainMenu Sequence Diagram

Game Over:

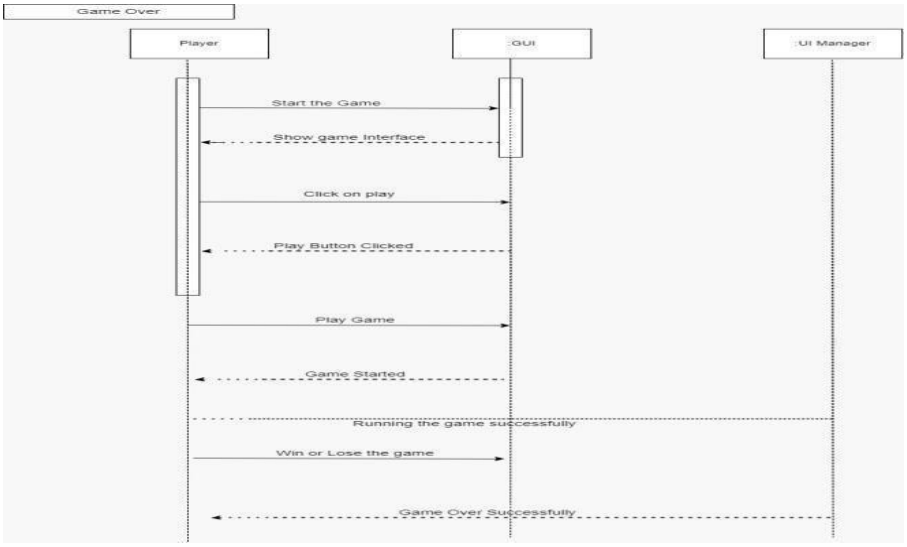

Figure 3: Game over Sequence Diagram

Activity diagram

Main Menu:

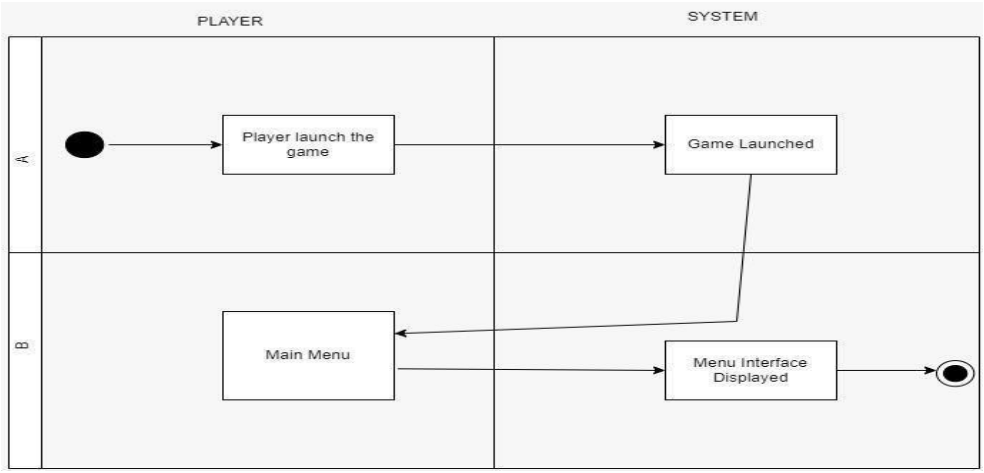

Figure 4: Main Menu Activity Diagram

Play Game:

Figure 5: Play Game Activity Diagram

Exit Game:

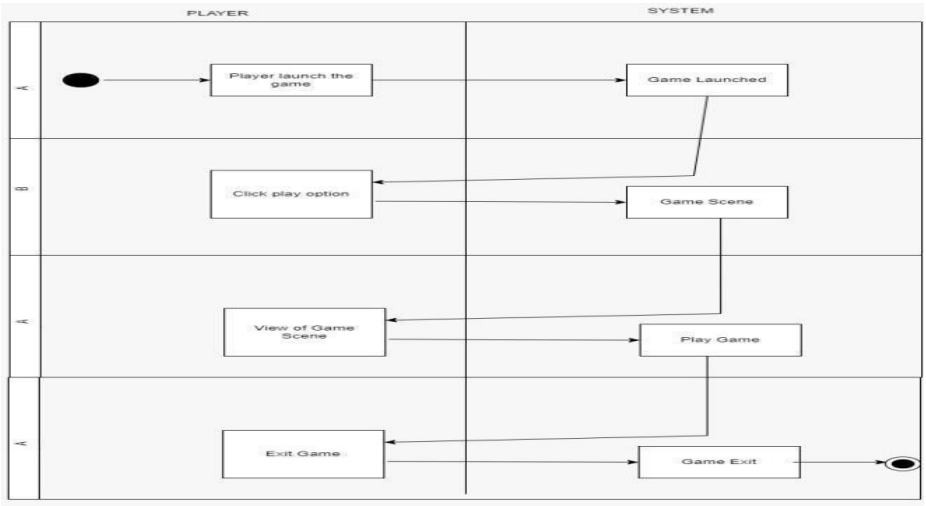

Figure 6: Play Game Activity Diagram

Software Requirement Specification

Functional Requirements

This section shows the basic needs for the game to have the game up and running in the time that is required.

Main Menu: The game should have a main menu where the player can decide what option they would like to invoke.

Terrain: The game should have a terrain where the characters can exist. It should have realistic physics including gravity, wind, and borders.

Characters: The game must have a set of characters who will either act as the Artificial intelligence or the player character

Weapon: The player must be provided with a weapon to kill and damage an enemy.

Play Game: The player should be able to select the play game option from the main menu. The play game button will give a short background story (which the player can skip if they wish before the game will start.

Game Completed: The player should be able to complete the game and receive recognition from the game that it is over. The player should complete the game once they have collected their soul after defeating the boss. When the player walks through the door and into the soul it will load the game over the screen which will then bring them back to the main menu where they can decide to play again or quit.

Pause Game: The player should be able to pause the game. The player will press the ESC button or press the pause button in the corner of the screen which will pause the game and give the player the option to continue, return to the main menu, or quit the game.

Quit Game: The player should be able to quit the game and stop playing. To prevent accidental exit, the game will ask if the user was sure. They can click yes to exit the game or no to continue.

Fight/Defeat Enemies: The player should be able to shoot and kill enemies in the game. These enemies will make the game more challenging as the player will need to pass them. The player can fight them or attempt to run away from them.

Player Movement: The player should be able to move the controllable character. The character can be moved using the movement keys and they can walk and shoot.

Aim And Sight: The player should be able to aim the gun and look down at the sight of the gun. This is achieved by pressing the right mouse click.

Intelligent Enemy AI: The game requires multiple enemies which attack the player. When the user

has defeated these enemies, a boss character will be introduced which will be stronger and harder to defeat. This boss character will also be more intelligent and it will provide the player with a challenge.

Graphics and animation

The system will constantly render textures and models. Graphics will need to be realistic to provide the user with a believable world. The game should also have suitable animations to complement the realism of the graphics including walking, running, attacking, etc. I feel that audio will help to complete this 3D environment.

User requirements

The user requirements are used to describe what the user will need to run and play the game. The user must have a computer or laptop capable of running a modern version of the Windows operating system and the graphics card requirements will vary depending on the type of game. As this project consists of simple graphics, the standard graphics cards in the user's PC should be able to run the game. Internet access will also be required for the user to be able to download the application. When the game is compiled, it is built into an executable file that is separate from Unity which will benefit the user as they will not be required to download and install Unity if they want to play the game. The game can also be compiled to run on web browsers. Users will need to install Unity Web Player which is supported by all web browsers. The user must also have a computer, keyboard, mouse, and speakers/headphones to play the game.

Environmental requirements

To play the game, the users will need a Windows, Linux, or Mac operating system environment but the game was designed to use on the Windows platform. This application will run on a computer or laptop capable of running a modern version of the Windows operating system (must beWindows XP or higher)

Usability requirements

The menu for the game should be easily accessible and visible and laid out to provide a clear understanding of the game menu. It should be easy for the user to accomplish a task and navigate through the menu. The game should not overwhelm the users and the UI should be clearly laid out and visually appealing.

Nonfunctional Requirements Performance/Response time requirement

It is important to ensure that the game does not suffer any graphical or performance problems as this can lead to frustrating gameplay. The game needs to run at a consistent frame rate to ensure that the graphics and animations remain smooth. The response time of the game should be immediate so that when the player initiates an action in the game, the results will be immediate. The player will need

precision for running and attacking, so the response time for input methods must be very quick.

Availability requirement

Once the user has downloaded and installed the game onto their machine it should be available to play whenever they want. The game will be built for Windows which will make the game easily available.

Security requirement

The player will need to ensure that they have the proper security privileges on their machine to allow them to install and run the game.

Reliability requirement

Once the user has downloaded and installed the game onto their machine it should be available to play whenever they want.

Maintainability requirement

Games can be easily maintained after their release through online patches. These patches can be used to fix any problems in the game. If there are several major bugs then a patch will be released to address and fix these. Players also can report any bugs to developers allowing them to fix the problem.

Portability requirement

Users can play the game on their PC or laptop. They can also copy it onto a storage device such as a memory stick and take the game with them if they wish. As I am using Unity3D, the game can be built to run using the Unity web player. The web player would allow users to play the game over the internet so they could play on any device connected to the internet.

Extendibility requirement

The option to expand the game with additional content, even after it has been released is always there. Development for the game does not need to end and this will be released in the form of an expansion which will aim to improve the overall game play experience.

System implementation

The previous chapters dealt with the basic theoretical background of the 3D game, and Unity game engine, and its development environment. To demonstrate how a 3D game can be developed, a case study called Third-Person Shooter (TPS) game was specifically created. This chapter demonstrates and discusses how the game and its main features were implemented and it deals with design, methods, and algorithms. The player plays a gun to find out all enemies and slay them on a terrain. When the player shoots at enemies and the enemy dies. The player must kill all enemies to finish the game.

Collection and Design of GameAssets

A completed game is normally made with the required game assets. The game assets consist of art assets and script assets. Art assets contain 2D/3D models, textures, pictures, music, etc. They are usually used to make game objects and Graphic User Interfaces (GUI). Game objects areobjects which can be in sight and arranged in the game scene. It can be said that game objects are the essential elements that constitute a whole game. And script assets are used to make all game objects to be performed. They are programmed with codes. Every script must be bonded with a suitable object. In this game, a few textures were collected and drawn to make the Graphic User Interfaces. And the standard assets packages were imported from Unity Assets Package. They are CharacterController Package, Terrain Assets, Skyboxes, and Tree Creators.

Character Controller Package

This package contains a 3rd character modelwhich was used to make a prefab with enemy game objects, a First-Person Controller which can simulate the animation of the player, and a few scripts which were used to make the model and controller work. It is men- 28tionable that the animations with “idle”, “walk”, “run” and “jump” is achieved for the models.

Terrain Assets

The package contains the tree model and a few textures and rendering materials, such as grass and palm. The designer can make the terrain and render all elements on the terrain with the Terrain Editor which was referred to in the previous section. Certainly, an outstanding art designer can make the assets more wonderful.

Skyboxes

Skyboxes were introduced in the previous section. The main function of skyboxes is to fill colourful dyestuff to the sky space in the game. In this case, the basic blue sky and white clouds were filled. Figure 17 shows the effect of Skyboxes in the game. The effect of Skyboxes in the game. The blue sky is the background and the white clouds are immobile. The effect is the basic effect of default Skyboxes in Unity. The designer can make more effect with 3D animation software and then imports these assets to Unity. The outcome of the project is shown in the figure 6,7,8,9 and 10.

Figure 7 Main Menu

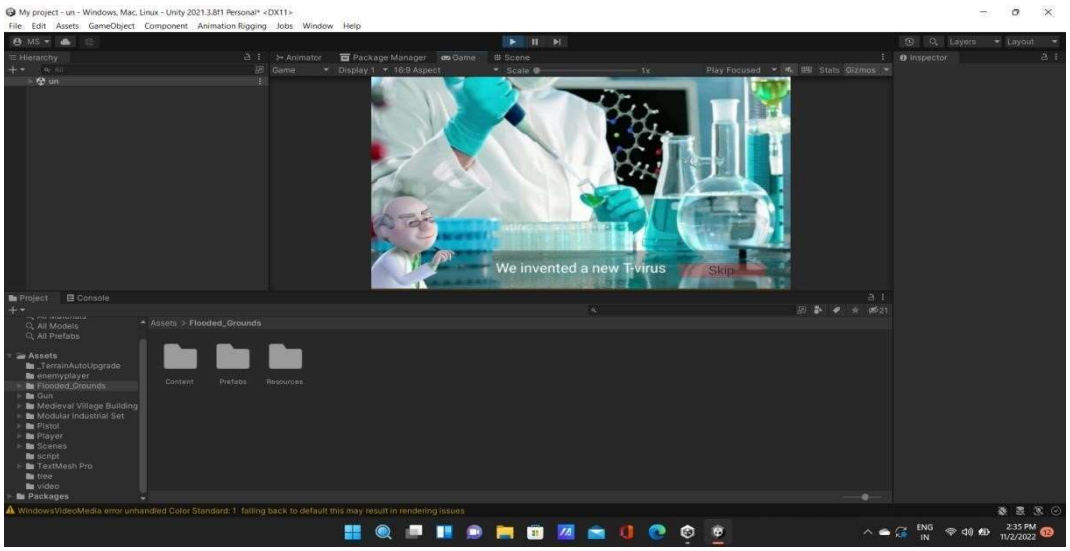

Figure 8 Game Intro

Figure 9 Game Scene

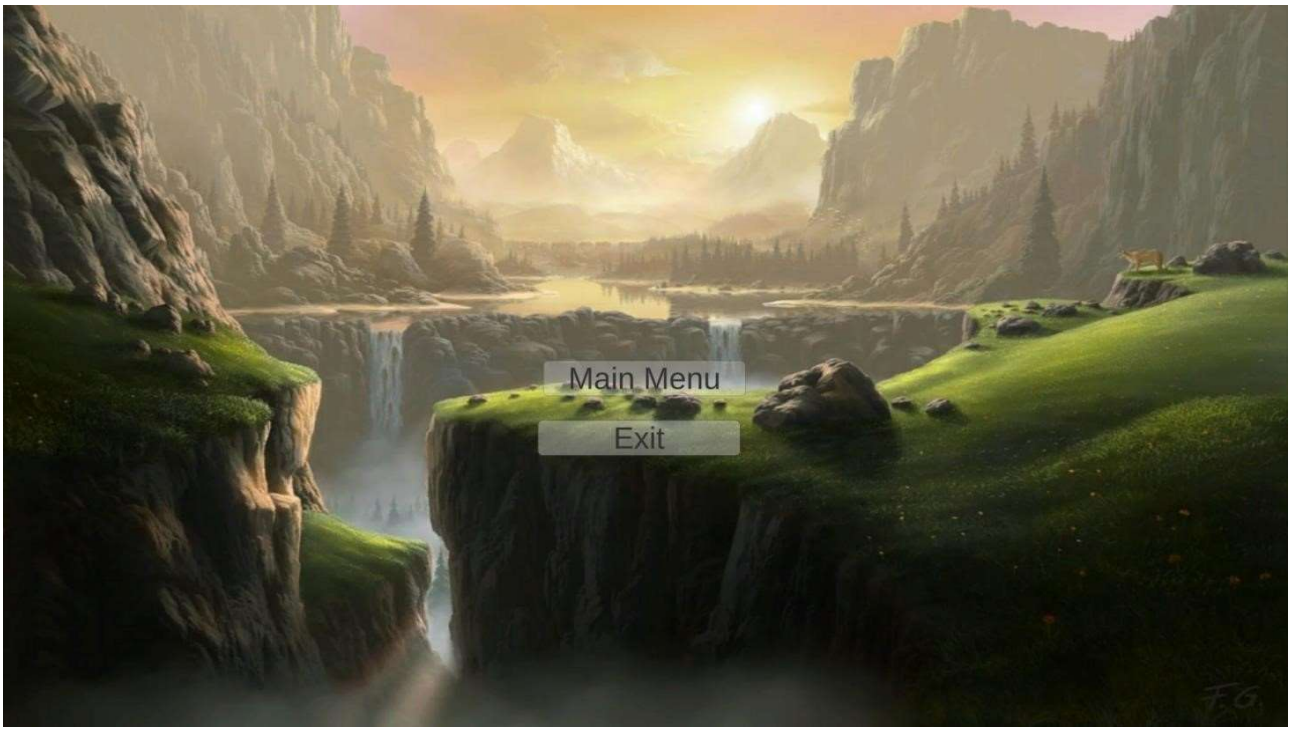

Figure 10 Pause Menu

Conclusion And Future Enhancement

This project aimed to create a 3D game which was a Third-Person Shooter game with the Unity game engine. This provides fundamental knowledge about the Unity game engine and programming. A game engine is a core of creating a game. The integration of model design, level design, and script design is the game engine, which is complex and powerful. The Unity game engine supports visualized design, thus it is a strong game engine that is suitable for a beginner. However, it is not very easy to learn the Unity game engine. There are various functions to be realized. To make the game to be an integrated game, four scenes were designed. One was the game's main menu scene and the other onewas the game intro video scene the other was a game scene and the last one was the ending game video. All of the game features were achieved as a Third-Person Shooter game. More game features can be looked forward to in the future. The goal of the report was to demonstrate how to create a 3D game with the Unity game engine and discuss the implementation of the scripts. The usages of the Skyboxes and the terrain design were the characteristics of the game. The most important aspect of game design is thinking about how to create new playabilities. This game is a product of a beginner who favours creating a virtual world. There might be more possibilities for the game in the future.

Future enhancements

- Introduce new environments andscenes
- Introduce a new game feature
- Adding new Characters
- Improve user interface
- Level Extension

References

Bibliography

[1] https://www.researchgate.net/publication/2956388_From_Visual_Simulation_to_Virtual_Reality_to_Games

[2] https://www.scirp.org/reference/Referencescapers. asp?ReferenceID=2543403

[3] https://www.researchgate.net/publication/344195486_Ibrahim_et_al
https://www.researchgate.net/publication/357399729_Nikhil_et_al

[4] Penny de Byl. Holistic Game Development with Unity. Focal Press: 1 edition November 15, 2011. 2 Xavier Borg.Understanding 2-dimensional spaces [Online].

[5] Blaze labs research. URL: http://www.blazelabs.com/f-u-hds.asp. Accessed 10 September 2013. 3 Wikipedia.Three-dimensional space [Online].

[6] Wikipedia, the free encyclopedia. URL: http://en.wikipedia.org/wiki/Three- dimensional space. Accessed 11 September2013. Accessed 11 September 2013

Webliography

[1] http://unity3d.com

[2] http://fourm.unity3d.com

[3]https://youtube.com/@AjayAJGameDev

[4]https://youtube.com/@CodingAnna

[5]https://youtube.com/@snapoutstudio2405

[6] https://www.instructables.com/How-to- make-a-simple-game-in-Unity-3D