APOCALYPSE

SUBJECT: SYSTEM DESIGN PRACTICE

DHARAMSINH DESAI UNIVERSITY

FACULTY OF TECHNOLOGY, NADIAD

PREPARED BY:

Patel Hiral (CE-082)

Patel Nishtha (CE-085)

Pujara Khushal(CE-093)

GUIDED BY:

Dr. C.K. Bhensdadia

(Head,CE Department)

Dharmsinh Desai University

College Road, NADIAD – 387001. (GUJARAT)

CERTIFICATE

This is to certify that the term work carried out in the subject of SYSTEM DESIGN PRACTICE and recorded in this report is the bonafide work of
Miss. Hiral B. Patel (Roll No.: CE082, Identity Number: 15CEUON119),
Miss. Nishtha R. Patel (Roll No.: CE085, Identity Number: 15CEUOS115),
Mr. Khushal P. Pujara (Roll No.: CE093, Identity Number: 15CEUON024)
Of B.Tech Semester 6th in the branch of Computer Engineering during the academic year 2017-18.

Dr. C. K. Bhensdadia
(Project Guide and Head of CE Department)
Dharmsinh Desai University, NADIAD
Date :

INDEX

Abstract

"Apocalypse" is a name of the game that was developed for project in Computer Engineering Department at Dharmsinh Desai University. The game is a 3D single player first-person shooter game creating using C# in Unity Platform. The game contains different parts of university and in that zombies will attack player. The player can defend himself/ herself and can also attack zombies. The game goes on till the health-meter not reaches to zero.

Introduction

- The main purpose of the project is to create an entertaining and a unique game with a FPS(First Person Shooter) theme, which runs on the windows platforms.

- Apocalypse is an interactive computer game with a graphical interface.

- Players can explore some portions of university.

- While exploring university, player will face zombies and will fight with them.

- In this fight, if zombie attacks player then life will be reduced from health-meter and player can also attack zombies and save his/her health-meter.

- The set of rooms(in university) and zombies can be extended or replaced to give different game variations.

--Tools and Technology

Platform: Unity 2017.3.0f3(64 bit)

Unity is a cross-platform game engine developed by Unity Technologies, which is primarily used to develop both 3D and 2D video games and simulations for computers,consoles and mobile devices.

Language: C#

Diagram Tool: UMLet

UMLet is an open-source Java-based UML tool designed for teaching the Unified Modeling Language and for quickly creating UML diagrams. It is a drawing tool rather than a modelling tool as there is no underlying dictionary or directory of reusable design objects.

Other Tools: Adobe photoshop cs6

SOFTWARE REQUIREMENT SPECIFICATION

For

APOCALYPSE Game

3.1 Introduction

3.1.1 Purpose

The purpose of this documentation is to provide detailed documentation of the 'APOCALYPSE' game. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate. The purpose of the project is to design and implement a 3-dimensional game written in C# using Unity Platform. The game will be a single-player. The team members don't take aim at developing an instructive game; instead the aim is action, action and more action.

Because of the project, the team members will acquire experience in developing a game and increase our skills in different areas of game development project.

3.1.2 Product Scope

In Scope

The project will be based on creating an FPS game with the goal in mind of being fun.

- Single Player
- PC based
- 3D platform
- Action-based
- Science-fiction
- Testing for 0 bugs in game
- Written in C #

Out of Scope

- Will not be multiple player
- Will not be consoled based

3.1.3 Intended Audience and Reading Suggestions

This document can be referred by software developers, documentation writers and the users.

3.1.4 References

IEEE format to write Software Requirement Specification.

3.2 Overall Description

3.2.1 System Perspective:

- In this game, the user "roamed" through a series of interconnected rooms(Mainly University Map) with standard moving controls such as W,A,S,D and the arrow key as well as the mouse or joystick.

- During the game, the user would collect various artifacts, such as weapons, ammunition, and fight the zombies he or she encountered.

- This specification is essentially a re-implementation of the Adventure game. As with the original adventure game, the game specified here is single user only. The specification is, of course, open ended, in that an unlimited set of rooms, artifacts, and zombies can be added.

3.2.2 System Functions:

There are essentially three main functional areas;

- *Dialogue Management:* The system uses a simple command line interface. A limited vocabulary is used. Interaction proceeds between the user and the game proceeds as a dialogue, in which the user enters a command, and

the system describes a response to the command. Users are shown some sample commands, and may try creating further commands by guessing the vocabulary.

- *Artifact Management* : The player can find and collect a number of artifacts during a game. These include weapons, ammunition, HP and so on. Different artifacts can be used for different things. The artifact management functions include keeping track of descriptions of artifacts, and of what can be done with them.

- *Creature and Fight Management*: When a player encounters zombies during the game, a fight can ensue. Players can employ a number of different weapons to attack zombies, and zombies may attack back. If the zombies wounds the player, points are deducted from the player's current strength. The game ends if the players' strength reaches zero. Monsters may also pursue users as they move from room to room, and may attack one another.

3.2.3 User Characteristics:

The game should be useable by any users, via a graphical interface. No special knowledge or skills should be assumed on the part of the users. Users are should not be expected to learn a set of commands in order to start using the game.

3.2.4 Constraints:

No special constraints have been identified.

3.2.5 Assumptions and Dependencies

No special assumptions or dependencies have been identified.

3.3 System Features

- The gameplay is action-based with no strategic or role-playing elements. The only goal is to kill all zombies until even one is not left before they kill our character.
- The game actually consists of more than one level. However, the number of levels has not been determined yet, because we will work on the first level in the first episode.
- In all levels, the mission will be the same as in first level, killing all "Zombies". The only difference will be the locations where the levels take place.
- Every level will take place in different parts of university.
- The player plays these levels from a first person view in a 3D environment.
- The player experiences the game world through the eyes of a main character. This makes one feel one is a part of the game.
- The abilities of our character as a player are walking and running in all four directions, jumping, using-reloading-changing weapons, and collecting the pick-ups.
- In the later episodes, we will neither change the main character nor add any additional characters that can be controlled by the player.
- The game has a design with only single-player mode. The attack of enemies will reduce the health primarily. The effect of damage will vary depending on the type of attack.
- The enemies can bite; they do not have the ability to use gun and shoot. However, the player can pick up the power-ups that increase health. The player has only one life.
- When his/her health will reduce to zero or lower, one will die. When our main character will die, the player will lose the game.
- If the player kill all the enemies before the health reduce to zero or lower, one will win the game.

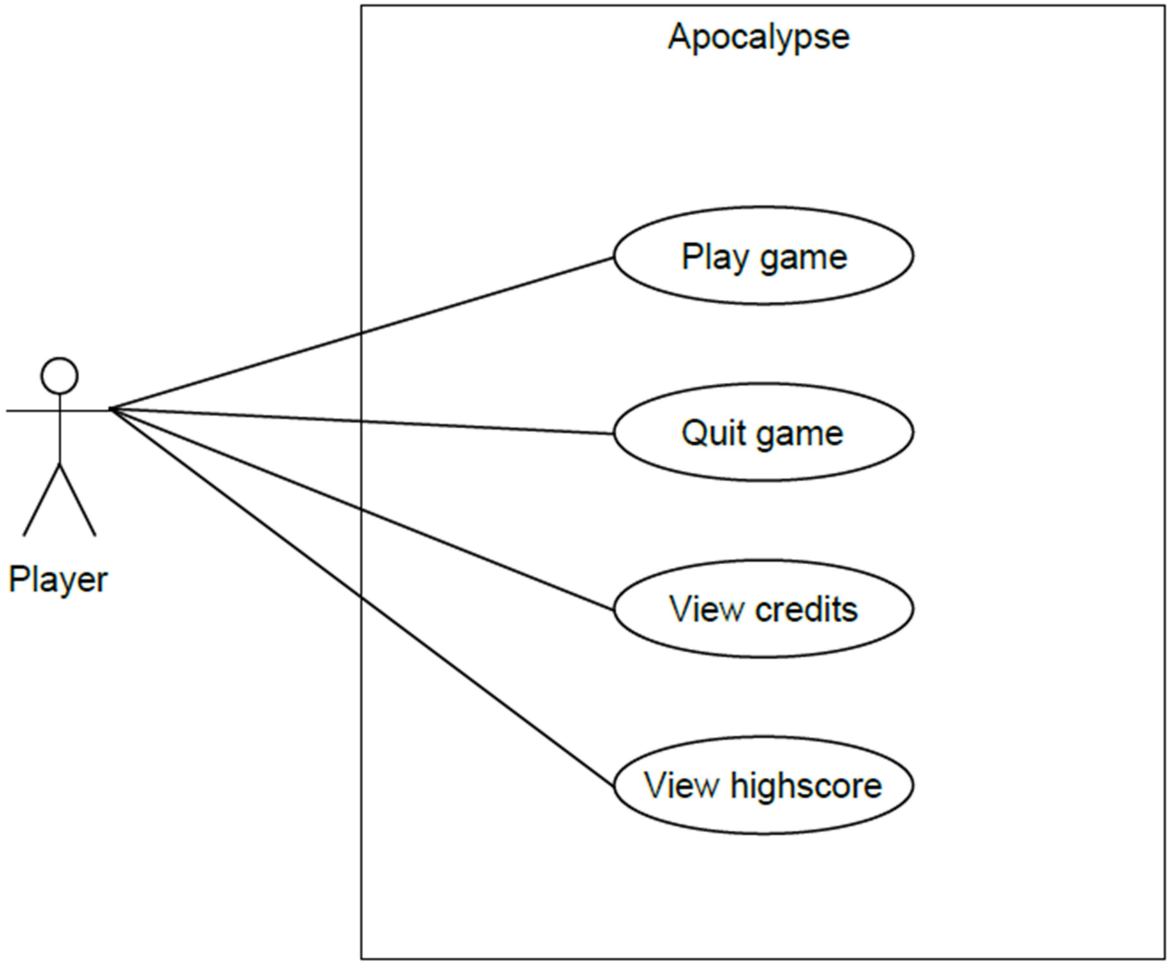
3.4 Non-functional Requirements

The application is developed in Unity game engine, where we will be importing the assets like characters and 3D model of building and a point source of light which will act like sun.

Design

1.Use-Case Diagram

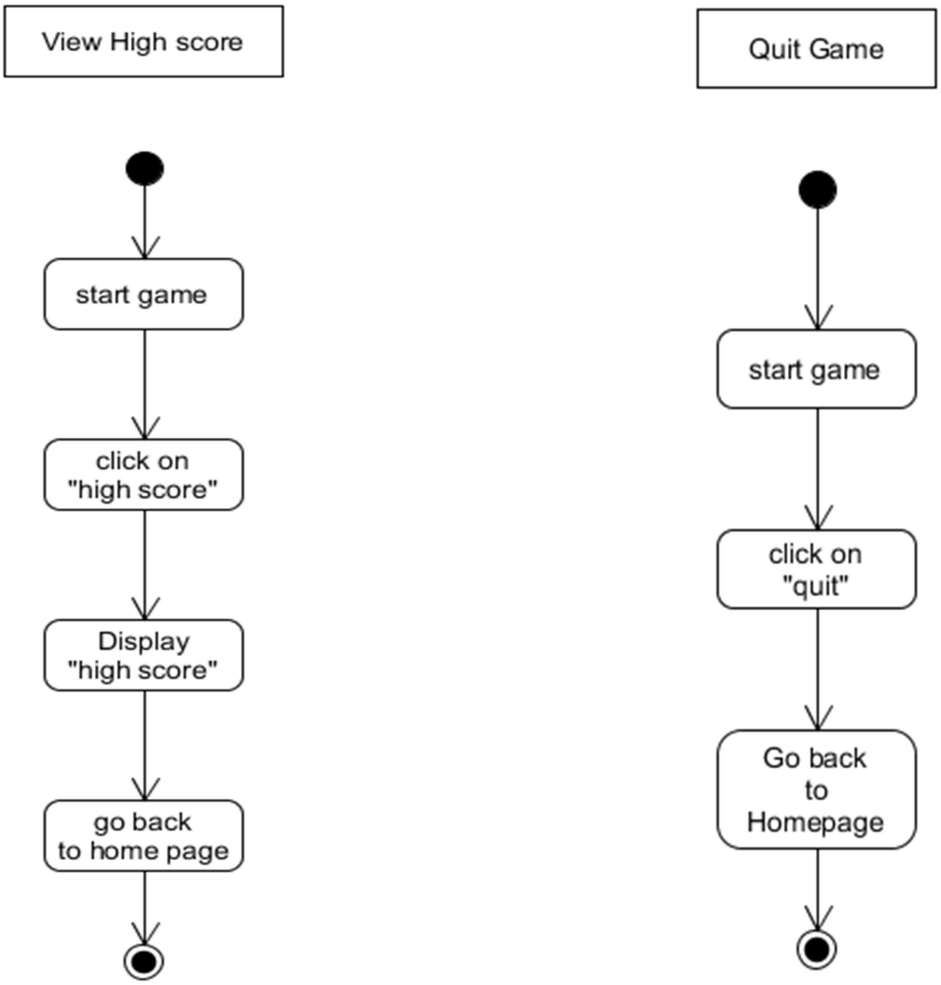
A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements.



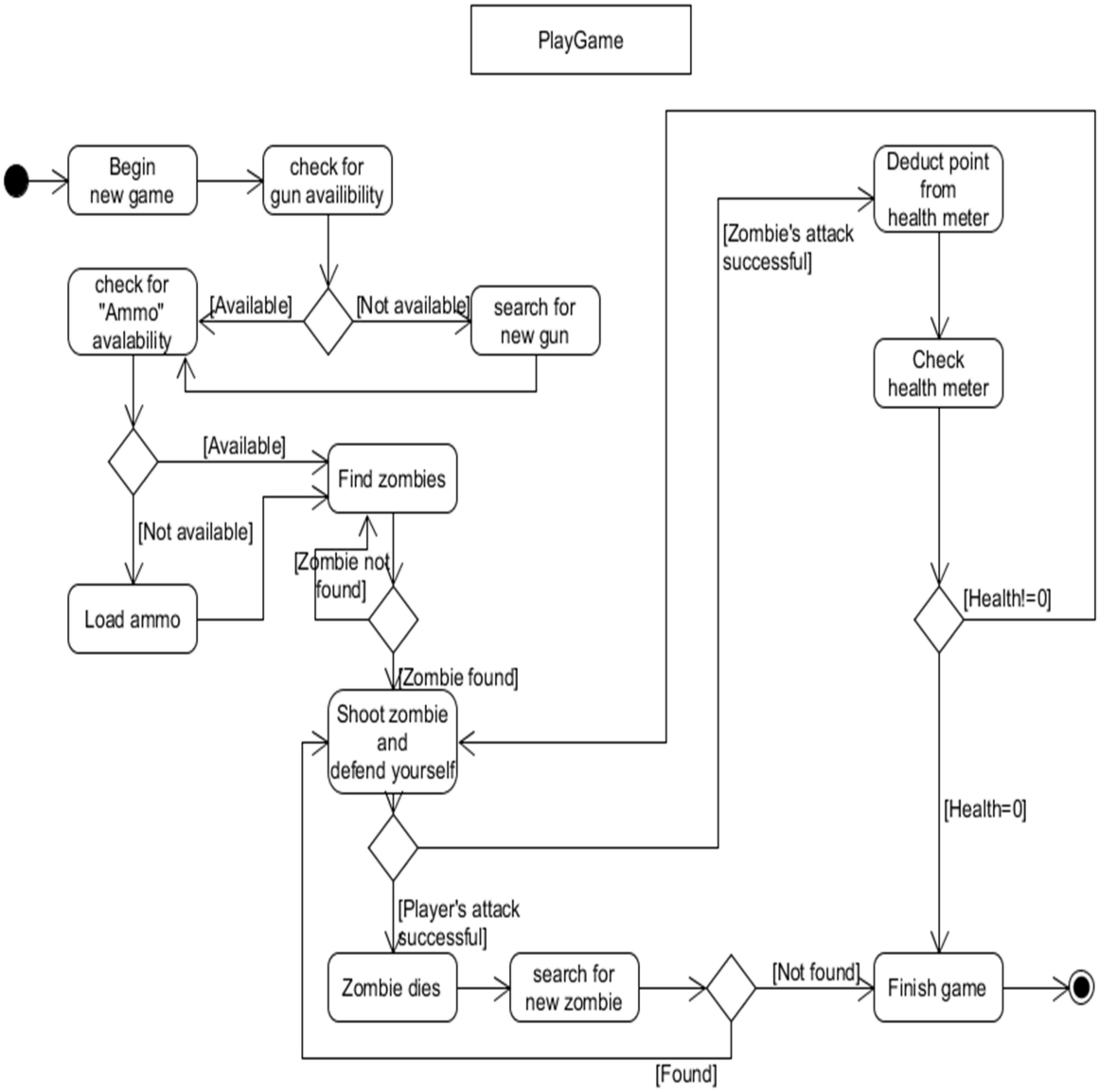
-This usecase diagram shows 4 usecases present in ‘APOCALYPSE’

2.Activity Diagram

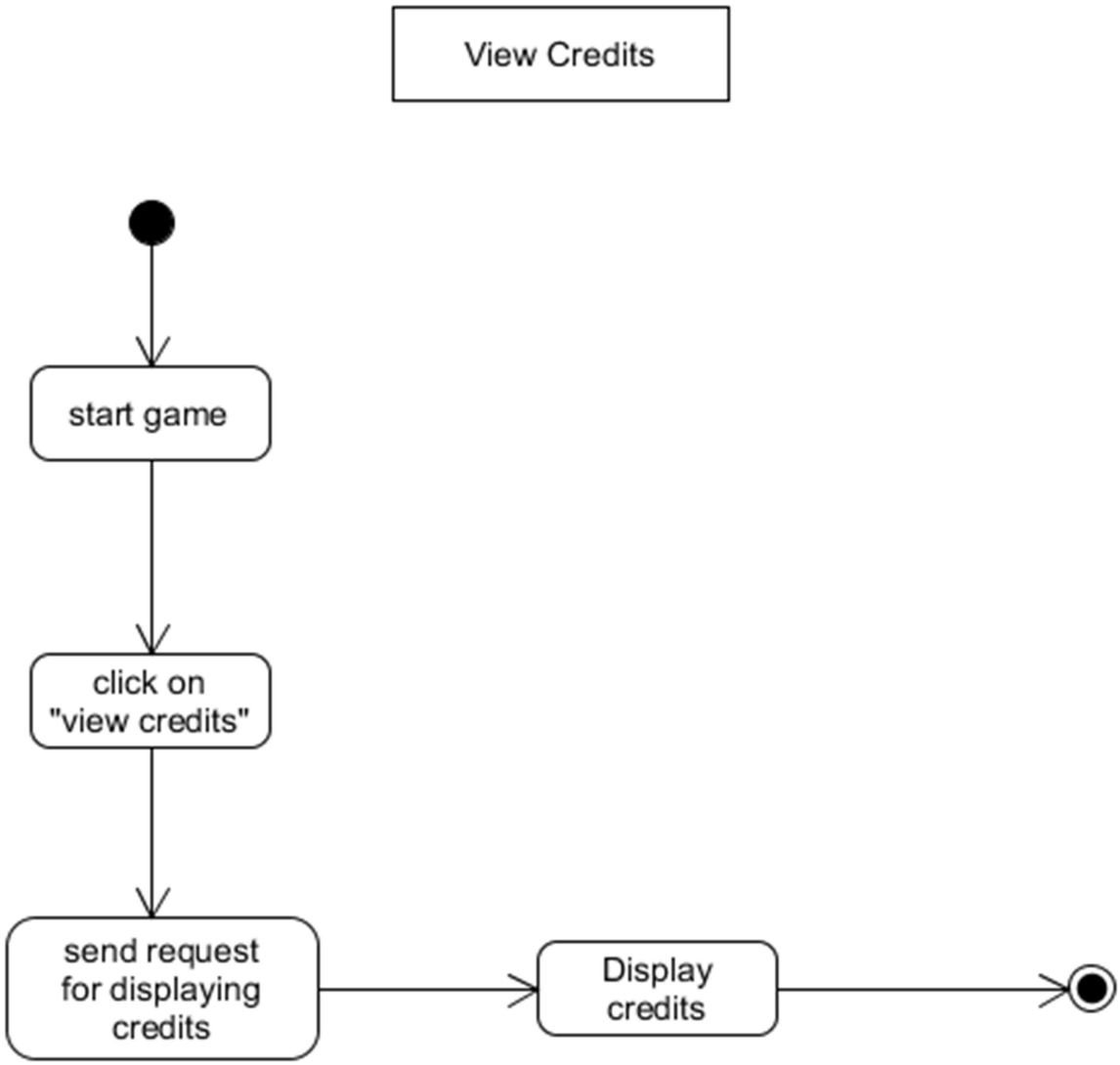
Activity Diagram is basically a flowchart to represent the flow from one activity to another activity. Following Activity diagrams are for ‘view high-score’ , ’view credits’ , ’play-game ’, ’quit-game’.



-This two activity diagrams represent how to view high score and how to quit game.

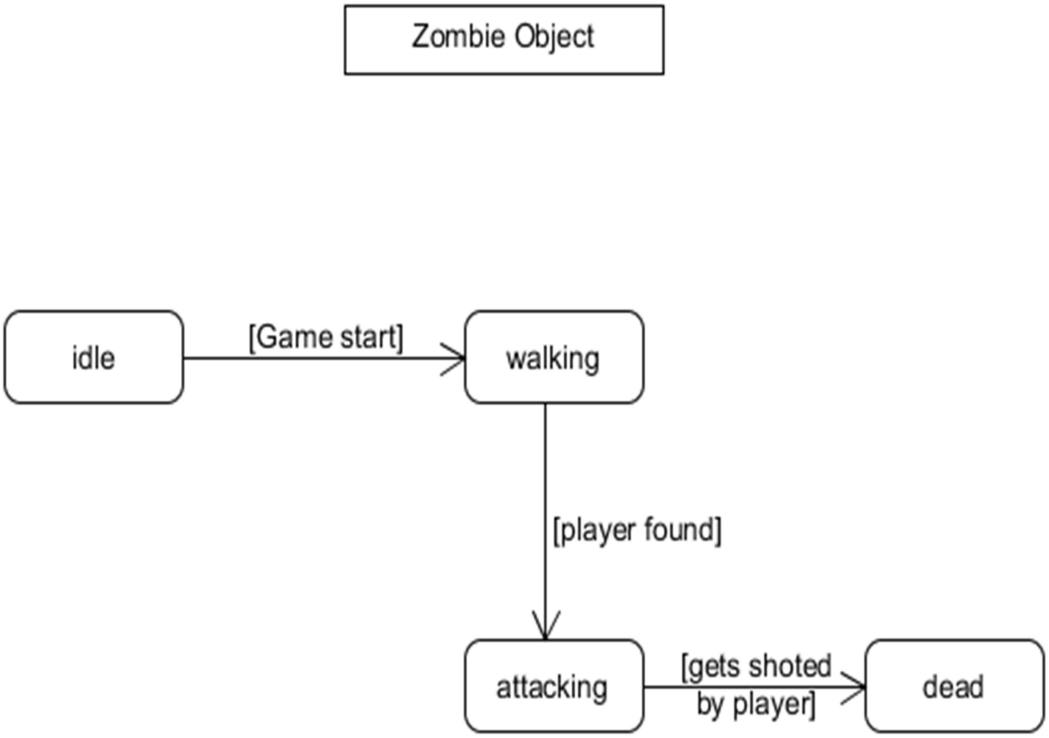


-This diagram represents the steps of playing game and how 'APOCALYPSE' behave in some conditions. The exact behavior, That how the game will behave is shown here.

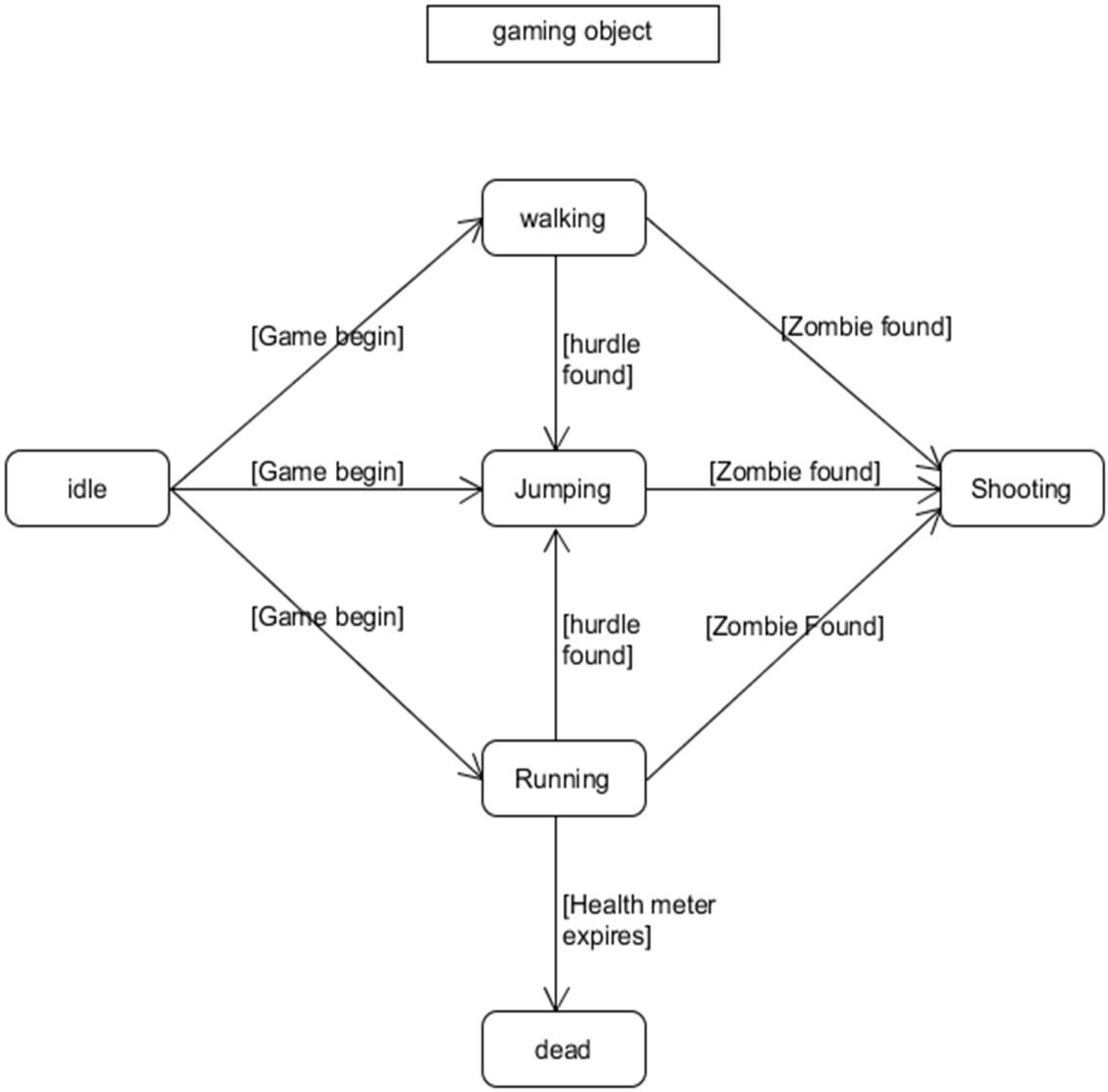


-This diagram shows the steps that how player can see credits.

3.State Diagram



-This diagram shows different states of zombie and how it changes.



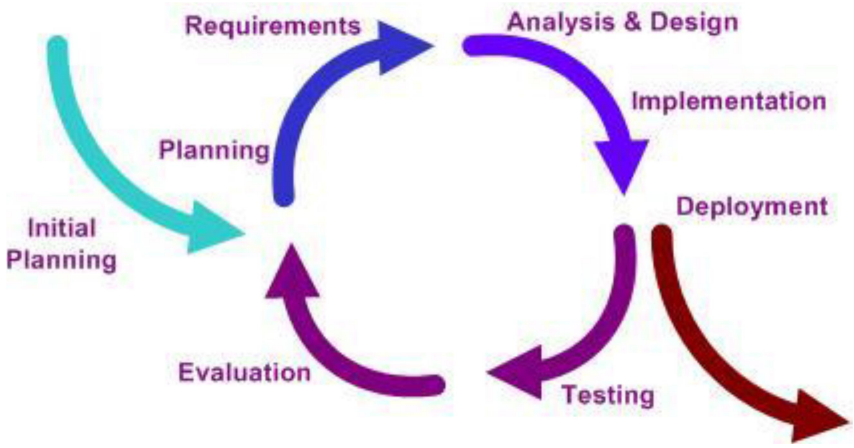
-This diagram represents different states of game object and how the

state changes.

Implementation Detail

--Methodology

- Even if one can prepare a well-thought-out design document, some features do not give the same effect in gameplay as on the paper.
- During implementation phase many features are added, removed or modified. Accordingly, one needs to make some modifications on game design and requirement analysis. Thus, one of the most suitable development methodologies for game development is iterative development process.
- Most of game developers have the same idea about the advantages of iterative development process for game developers.
- We prefer the iterative development process with specified milestones and deliverables for our project.



--Development Process

- We planned the project over a period of 10 weeks and divided it into four iterations. We planned the first iteration for planning, second iteration for game design, third iteration for coding and the final iteration for testing and finalizing the product.

- In the first iteration, we focused on Project Plan determined as the first planned milestone of the project as we mentioned in Project Plan document. Planning is essential for starting of upcoming milestones and delivering a finished project on time. Successful completion of a project is heavily dependent on effective planning. Iteration 1 was set to approximately one week (8 days) to figure out the detailed activities in order to minimize risk to the final result and delivery.

- The second iteration started by brainstorming among group members on what the game would be. Each group member denoted the attributes or properties of the game that one dreamed to implement. We gathered suggestions together and chose the ones that was possible to be implemented within a 10-week-project time. As soon as the game concept became clear, we made some early decisions on basic requirements of the project in order to more easily reach the development goals. GDD(Game design document) was also the main deliverable in this

iteration. Game design document was meant to be a living document. In other words, throughout the production process the document was updated, if needed.

- As we had GDD in hands, it was used as baseline to start implementation. In the third iteration coding in C# using visual studio was under way. None of us had experience in game programming. Therefore, most of time in this iteration was dedicated for internal training sessions.

- Last iteration was planned for testing and finalizing the product. The testing process is an iterative process. The successful testing process of software requires a good plan. Therefore, after the requirements of the project are confirmed, the future testing of the system and the code were planned. The test plan provided information on how and when the testing will be executed. In the second iteration, test cases were designed for the planned tests. In iteration three, the designed test cases were executed alongside the module testing and usability testing. During the last iteration, according to the result of the tests, the test reports were documented properly and the bugs were reported after the testing is completed.

--Texture and Structure

- The structure of the whole map is made by use of different 3d objects such as cube, sphere, capsules, cylinders, planes etc. On those structures, various kinds of material of various textures are applied for giving the real time effect.
- Most of the textures are taken directly from the real object using high definition camera using Photoshop.

- The Gun object is taken as a prefab. It is attached to the FPS controller. Whenever the player sets the trigger active, gun object gets activated. The game has two types of guns
  -5mm pistol
  -Sub machine gun

- Zombie and spider objects is taken as a prefab due to less time. The object has four types of animations Idle, Attack, Walking and Die. When the player will enter the attack territory of the enemy, the enemy will start acting the a particular way using particular animation

- Bullet holes and Blood particles are the results of particle system. Whenever the player shoots at some object, depending upon the tag of the object different types of particle are generated. The zombie will emit red, where as spider will emit green blood particles. The untagged object will have bullet holes on them

Testing

The test approach consists of a series of different tests. The primary goal of these tests is to ensure that APOCALYPSE is an error free game.

1. Main Menu

Test Case ID	Test Case	Actions	Expected Result	Test Status	Comment
I	Main Menu	1-User wants to play game 2-User runs game	A main menu is displayed.	Pass	
I.A	New Game Menu	1-User runs game 2-Main Menu pops up 3-User chooses “New Game” menu item	”New Menu” submenu is displayed.	Pass	
I.B	Back to Main Menu	1-User runs game 2-Main Menu pops up 3-User chooses “New Game” menu item 4-Submenu pops up. 5-User selects “Back to Main Menu” menu item	Main menu is loaded.	Pass	

2.The Game

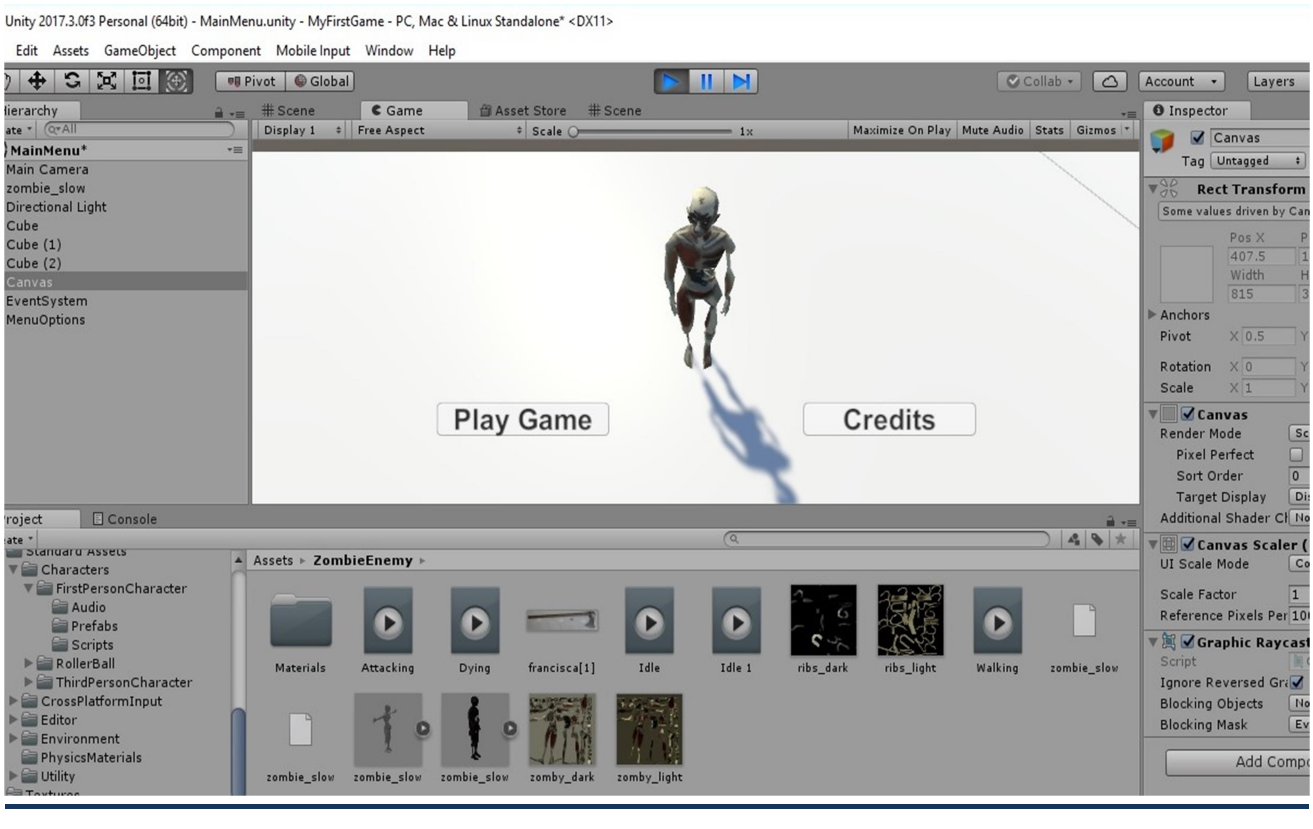
Test Case ID	Test Case	Actions	Expected Result	Test Status	Comment
I I-1	BeginGame	1- User runs game 2- Main Menu pops up 3-User enters a game	Game begins	Pass	
I I-2	EndGame	1-When player kills all enemies or is killed by enemies 2-User exits game via in-game menu 3-User returns to Main Menu.	Game ends	Pass	
II.A	Graphics	1-User runs game 2-Main Menu pops up 3-The main menu graphics are displayed 4-User starts game	The graphics requested are displayed.	Pass	
II.A.1	Enemies	1- User encounters enemies 2- The game requests the graphic model for enemies	Enemies are displayed.	Pass	

Test Case ID	Test Case	Actions	Expected Result	Test Status	Comment
II.A.2	Terrain	1-The game determines user direction 2-The game displays the appropriate graphic models.	A 3D terrain is displayed.	Pass	
II.A.3	Fire Effect	1- User fires the weapon 2- The fire particle effect is	A fire particle effect is displayed.	Pass	
II.A.4	Bullet-Hit Effect	1- User fires the weapon 2- The bullet hits the object 3- The bullet-hit effect is requested from the graphics engine	The game displays the bullet-hit effect where the bullet hits.	Pass	
II.A.5	Weapons	1- User presses key to change weapon 2- The game needs to display the new weapon 3- The weapon model is requested from the graphics engine 4- The game displays the new weapon.	A weapon is displayed.	Pass	

Screen-shots

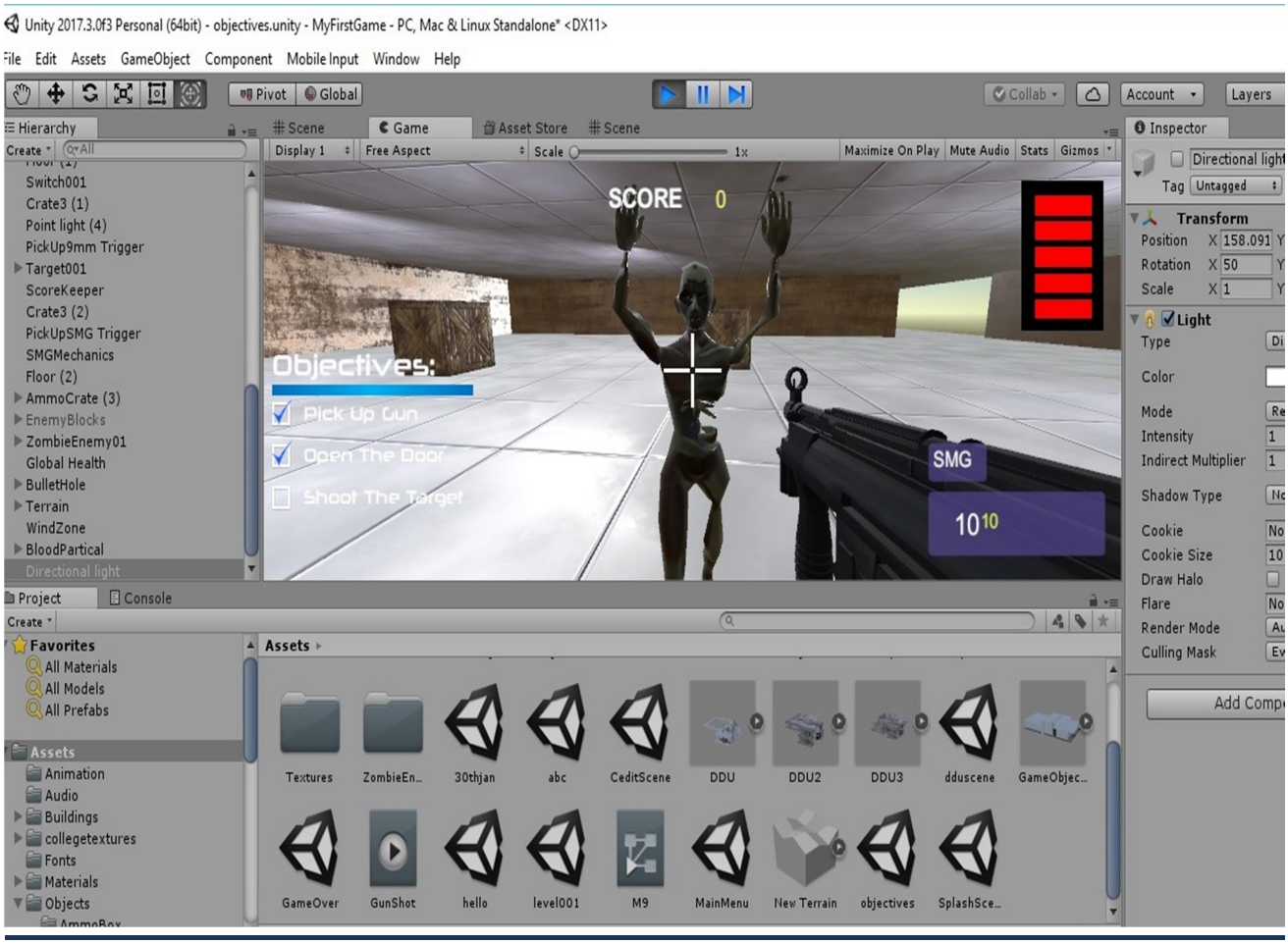
1. Main menu

User can click on play game button and game starts.



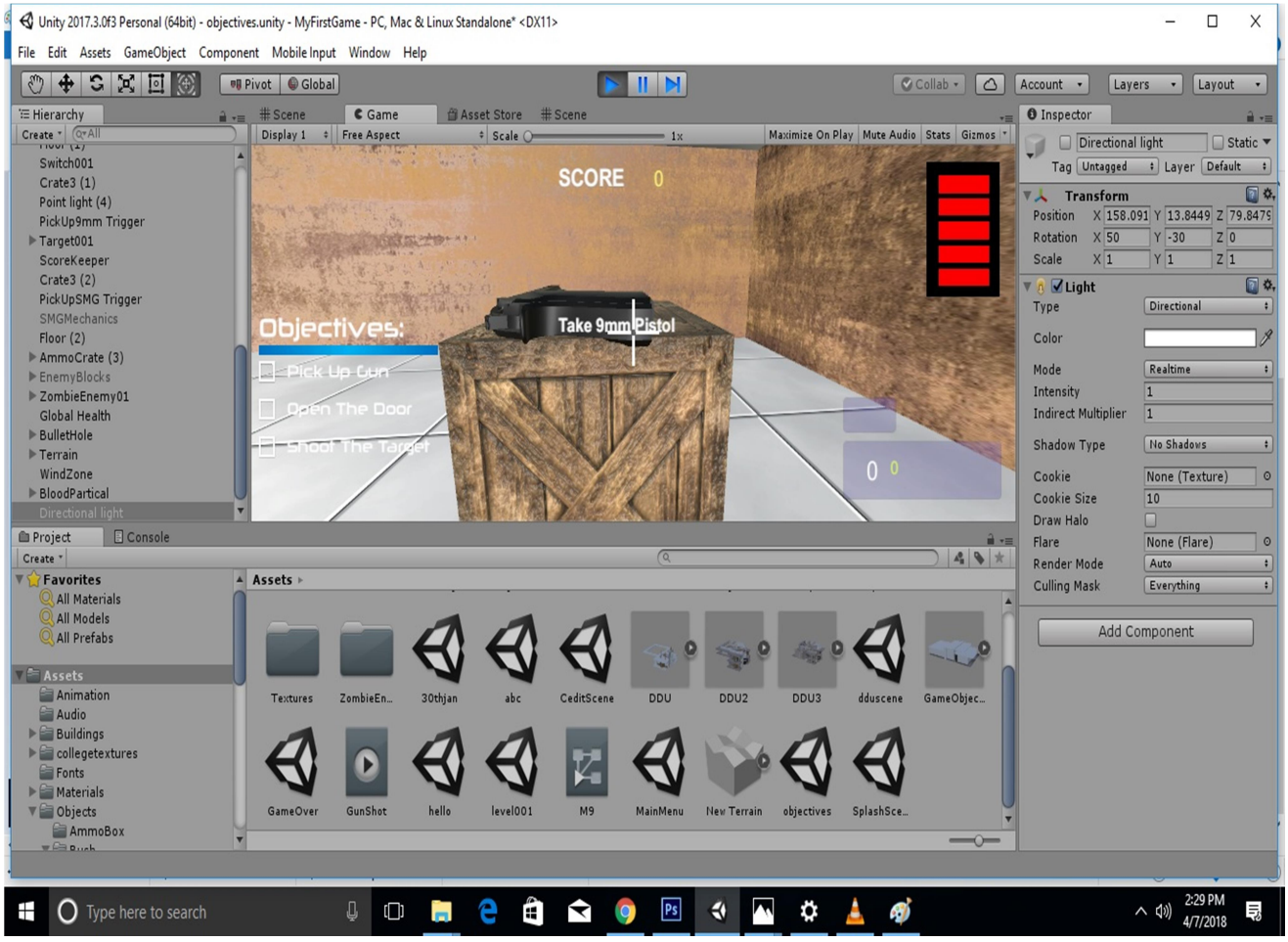
2.Shoot enemies

User can shoot enemies/zombies with guns. Zombies can also attack on player. Whenever zombie attacks the player, health-meter decreases. If health-meter reaches to 0 then game ends otherwise game continues.



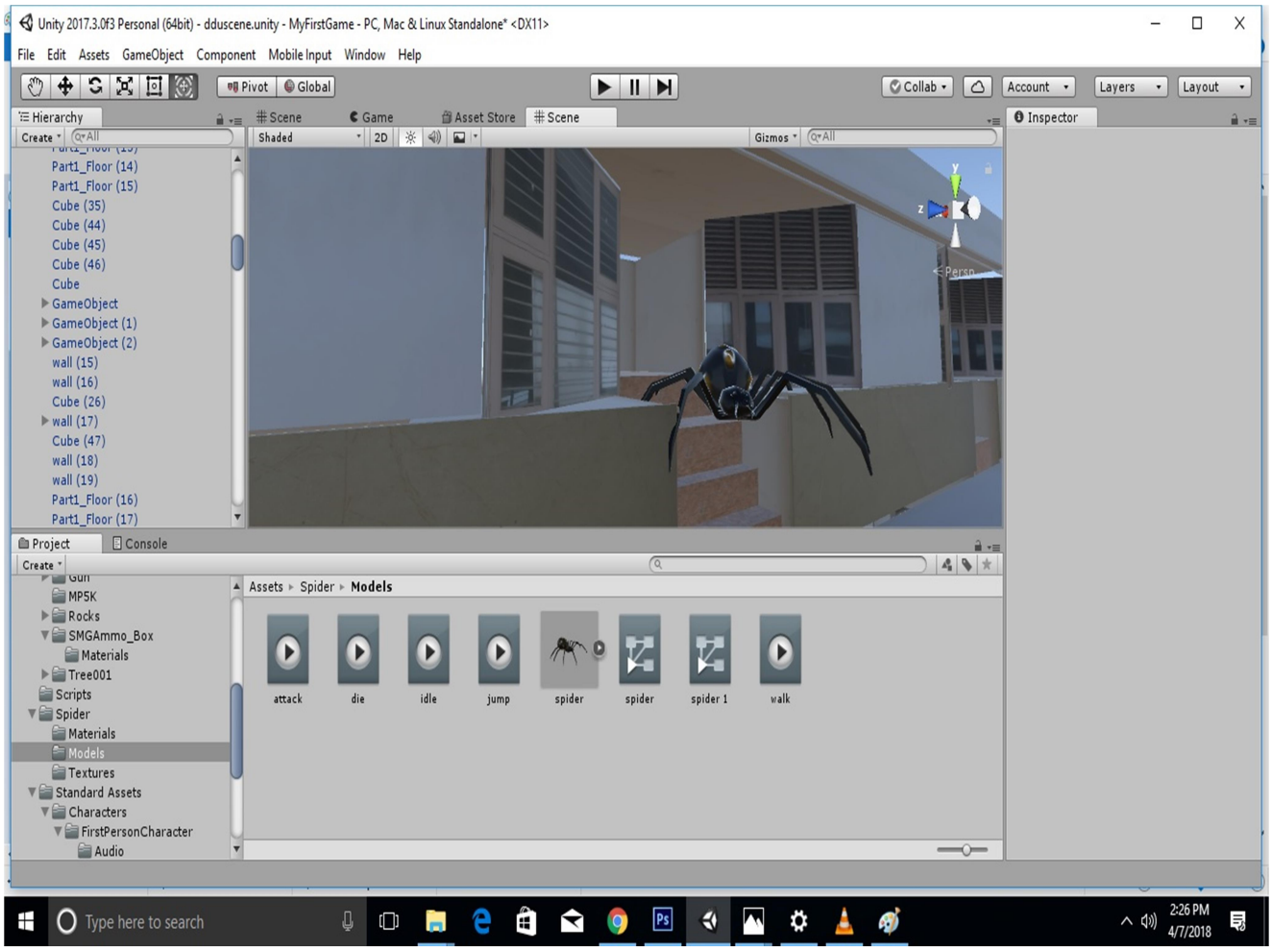
3.Pick up the gun

User can pick up the gun. User can also change the gun. After picking up the gun, user/player can collect ammos i.e. bullets. After finishing of ammos, user can not fire.



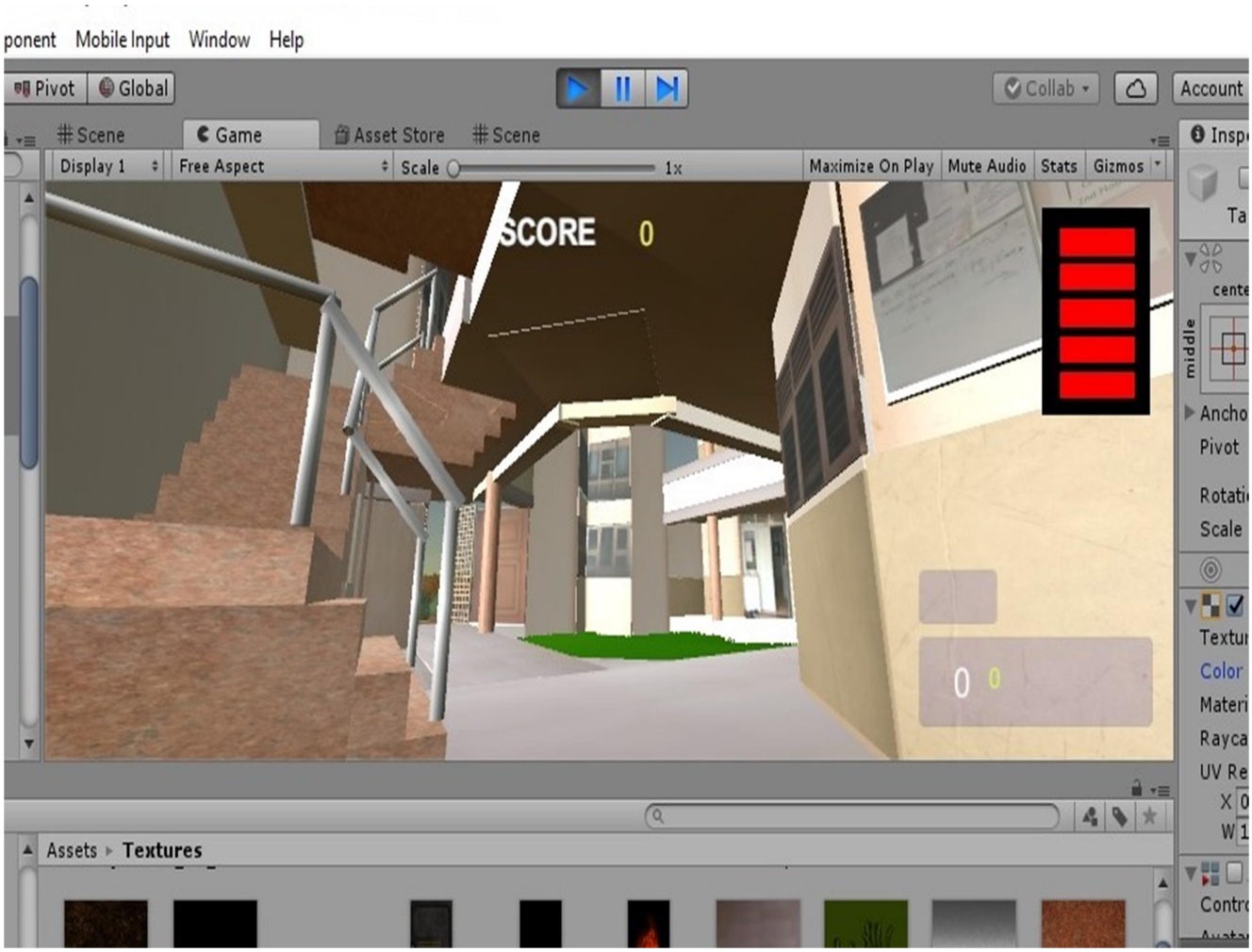
4.The second attacker-‘Spider’

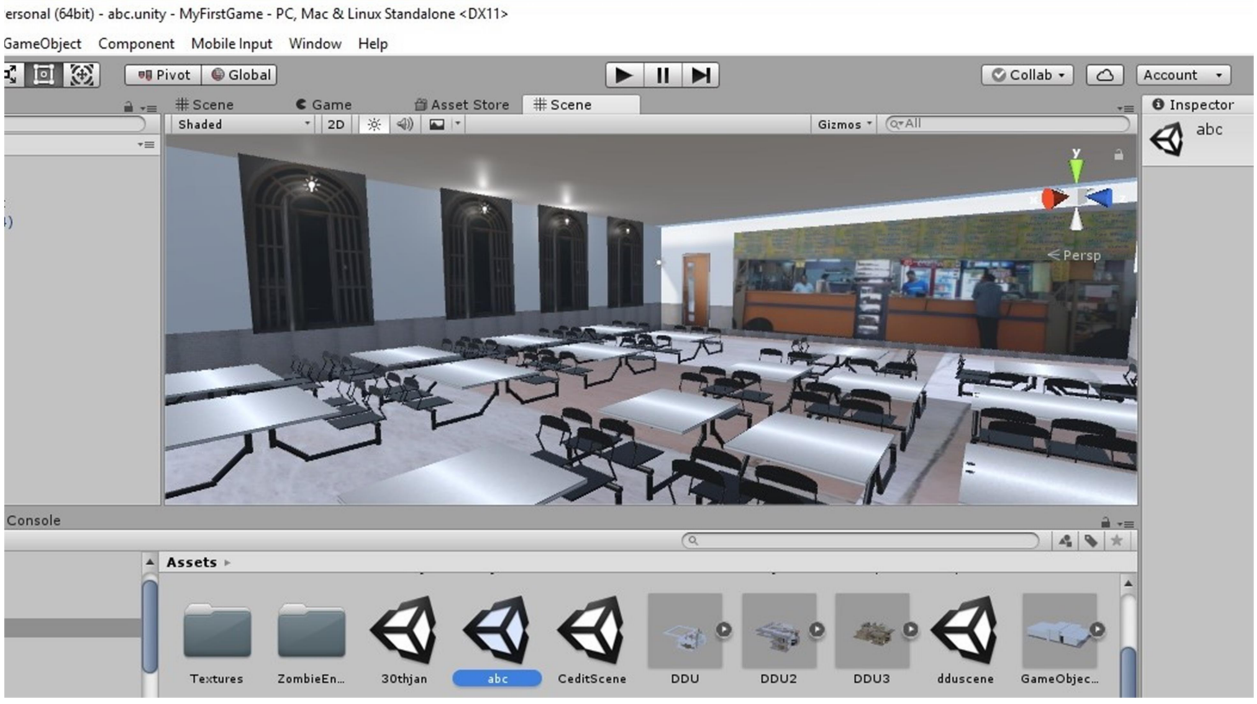
Not only zombies but spiders are also enemies of player. They can also attack player and can decrease user’s life.

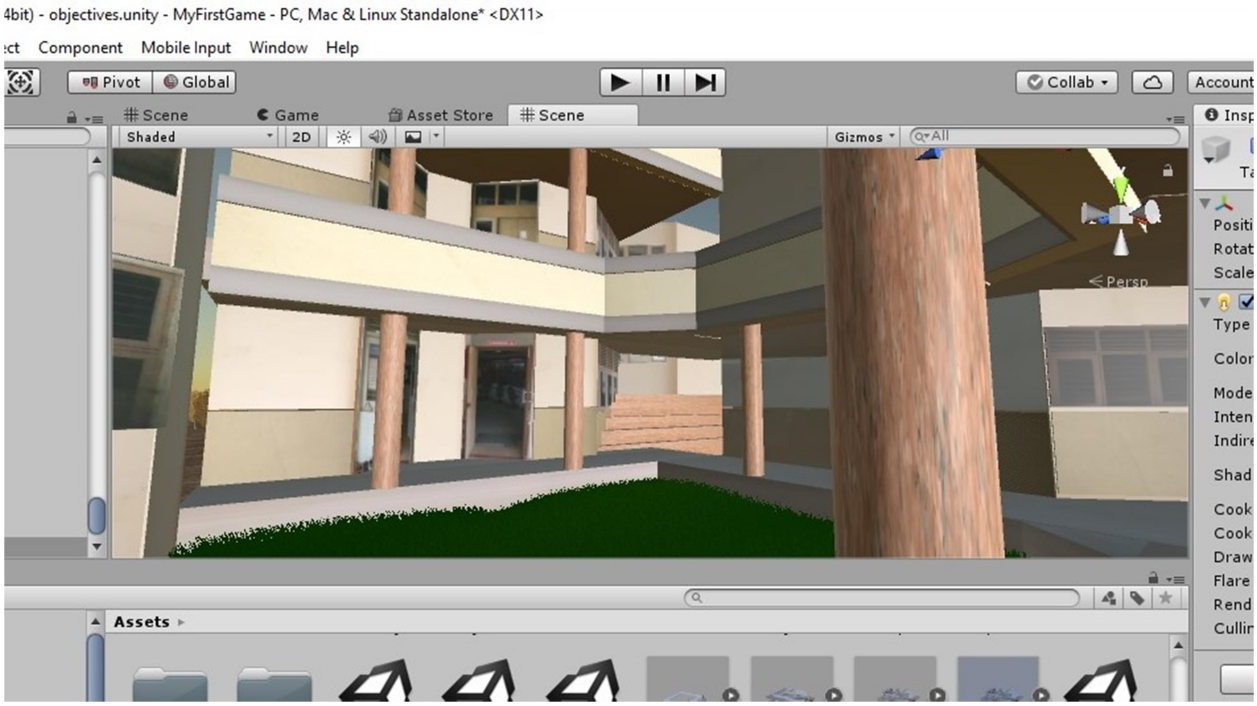


5. Glimpse of University

Following photos are of university in which game takes place.




ersonal (64bit) - abc.unity - MyFirstGame - PC, Mac & Linux Standalone <DX11>
GameObject Component Mobile Input Window Help
Pivot Global
Collab
Account
Scene
Game
Asset Store
Scene
Inspector
abc
Shaded
2D
Gizmos
All
Persp
Console
Assets
Textures
ZombieEn...
30thjan
abc
CeditScene
DDU
DDU2
DDU3
dduscene
GameObjec...


4bit) - objectives.unity - MyFirstGame - PC, Mac & Linux Standalone* <DX11>
ct Component Mobile Input Window Help
Pivot Global
Collab
Account
Scene
Game
Asset Store
Scene
Shaded
2D
Gizmos
All
Persp
Assets

Conclusion

This project is our first game development project. At the end of the project, the lesson that we learned and think to apply to our future projects is making a good project plan, remaining true to this plan as possible and documenting everything we do. By doing this, whenever we have a problem we can go back and see what we did the last time we had a similar problem.

Throughout the project, the biggest problem was limited time. Many times during the project we had the devil of a time because of time shortage. The fact that we had examinations addition to our thesis work caused us to take breaks, albeit short ones.

We, who developed a game for the first time, think that the decision on game genre is one of the most suitable decisions that we have made throughout the project. There exist so many alternatives to examine in the game market because of intense interest of the people in FPS games. Therefore, it was a good start for us. Our thought is “the more examples one has near at hands, the easier the implementation will be.”

As a result, 10 weeks were full of new great experiences and these experiences that we have gained will help us in our upcoming projects. Finally, the idea of developing game that was a dream for us since our bachelor education became real and we have experienced the pride of doing such a thing at the end of such an education.

Limitations And Future Enhancement

Limitations

- Timer is not set for the game.
- Only single user allowed to play game at a time.
- The game can be played within restricted area only.

Future Enhancement

- The game can be developed in such a way that multiuser can play it.
- New areas of university can be developed.
- Different times can be set for different levels.

BIBLIOGRAPHY

[1] Unity UI. (n.d.). Retrieved November 28, 2014, from http://unity3d.com/

[2] Assets : https://jvunity.weebly.com/downloads--assets.html

[3]Video Tutorials by Jimmy Vegas: https://www.youtube.com/playlist?list=PLZ1b66Z1KFKh-_auyvzNkOBFLWf6dH6iEH