Unity Game: Shoot Me If You Can

Fabiha Iffat 160041072
Novera Tasnuba Aura 160041041
Raisa Fariha 160041049

September 9, 2019

Contents

List of Figures

0.1 PROJECT OVERVIEW

The project we made is basically a shooting game. The player has to shoot the zombies that comes across in order to complete a particular level. The player can also collect bullets after a certain period of time by shooting. This is a game which is suitable for a particular age, for example children from age 8-10 years to adults anybody can play the game and enjoy it. Zombie shooting can be fun with all the fun environment, sounds and interesting zombie structures running across through a space.

0.2 MOTIVATION AND GOAL OF THE PROJECT

It is almost every person's dream to be a super-hero. Now this game can be a purely entertaining and a good time-passing medium where anybody can have fun and at the same time feel like a superhero by killing some zombies. Zombies are horrendous creatures so being able to kill those can make anyone feel heroic and also have a lot of fun and anyone can play it in their leisure time.

0.3 SCOPE OF THE WORK

This game is totally based upon unity. Unity is a game developing software where game assets, graphics, scenes can be used to build a particular game environment and ultimately deliver a proper game scene. It is a great platform for those who are interested in game development and designing. So we are using graphics, game sounds, environment etc. for creating a proper game scene in our zombie shooting game.

0.3.1 CONTEXT OF THE WORK

At startup, Unity 5 shows an empty scene. This scene already has a default lighting context available with ambient, sky dome-based reflections and a directional light. Any object placed in that scene should, by default, have all lighting it needs to look correct. A Skybox, baked or procedural, is an integral part of your lighting setup. It is used to control the ambient lighting and the reflections in your objects in addition to rendering the sky. Then we have imported assets from unity asset store like zombies, cube, palm tree and houses.

0.3.2 WORK PLANNING

In order to implement the project we are going through the following steps:

1. Create a storyline
2. Design the IUT map
3. Point out the places from which enemies will appear
4. Design the different places for different scenes
5. Collect assets like player, shooting gun, enemies and the environment

6. Start implementing level 1
7. Complete the scenes of the different levels
8. Build and run them, merge them together and present the game

0.4 PROJECT USERS

This game is mainly suitable for anybody above 16. This is a pass-time game and anyone can play it for entertainment and pleasure in their leisure time. We have targeted the students of IUT as the game scene is solely based on IUT.

0.5 PROJECT MODELS

We are using 'Agile' method as the project model which is a type of incremental model that is our project is developed in incremental rapid cycles. We have shown our teachers our progress in small incremental releases with each release building on previous functionality.

0.5.1 UML DIAGRAMS AND DESCRIPTIONS

The UML DIAGRAMS that we have chosen to manifest here are activity diagram, class diagram, use case diagram. The diagrams are shown below:

0.5.2 ACTIVITY DIAGRAM

To describe the activity diagram of our game, we have divided the whole diagram into three simlanes-

1. Game
2. First person or the player
3. Enemy

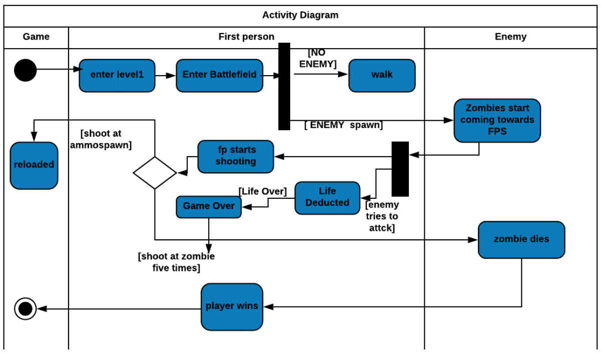

Figure 1: Activity Diagram

Now when the game starts the player enters his unlocked battlefield since there are going to be three levels in our game.When the player enters the battlefield zombies will start spawning and coming towards the player. The player has now two options. Either he can walk or he can shoot the zombies. Zombies will also attack the player and thus the life of the player will start decreasing and this life being over will result in game over. Now the players can shoot the zombies the moment they come to the eyesight and if a zombie is shot five bullets he will die. If the player can kill all the zombies then he will win and as a result he will get access to the next level.

0.5.3 CLASS DIAGRAM

In our environment of the game there is going to be some attributes like houses, trees. In environment there will be two more classes:

1. First person

2. Enemy

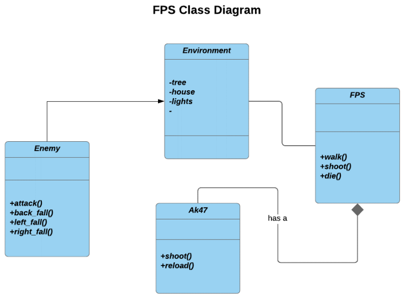

Figure 2: Class Diagram

Now the zombies can walk, attack the player, and also falls whenever he is shot 5 times which results in death of the player. Again first person can move and shoot the zombies and also dies if the zombie attacks. Now first person will have AK47 for defending himself from the zombies. This AK47 doesn't have its own identity if the first person does not exist in the game. So we have added AK47 and first person as composite relationship.

0.5.4 USE CASE DIAGRAM

In our use case diagram the player is considered the primary actor and the zombies as secondary actor.

Now the player will enter battlefield and starts shooting zombies. He can reload the bullets by shooting at the ammo spawn. The zombies will ignore all the obstacles and walks towards the players. And if he is shot five bullets zombie dies. And player dies when his life is over. If all the zombies die then that will be considered a victory for the player and he will be moved to the new level. It is up-to the player if he wants to continue the game or not. That's why we have added an extend relationship between them.

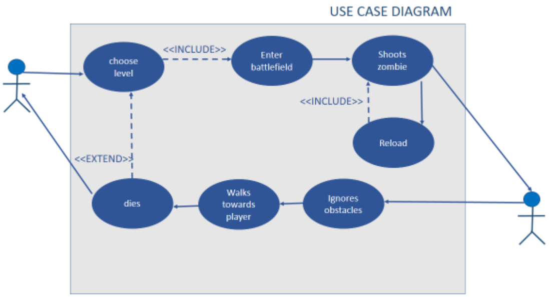

Figure 3: Use Case Diagram

0.6 REQUIREMENTS

0.6.1 FUNCTIONAL REQUIREMENTS

The functional requirements of our game are:

1. The system will allow user to be a player
2. The system will allow user to be an administrator
3. The system will allow user to be in one game at a time
4. The system shall display the number of deaths of enemies and also the death of player

0.6.2 SYSTEM REQUIREMENTS

The data requirements of our game are:

- Windows: Any computer having 32 bit/64 bit windows can play this game
- RAM: A computer must have a RAM of 16 GB and above to play the game
- Enough disk space: Any computer can store this game if it has the enough memory capacity
- Operating system: This game can be played using any operating system

0.6.3 USABILITY REQUIREMENTS

The usability requirements of our game are:

1. Learnability: Game is very easy to learn and takes very less time to understand all the features
2. Subjective first impression: First impression of our game will hopefully give positive and superhero vibe
3. Operability: User can play the game using mouse and directional keys on the keyboard
4. Understandability: The human computer interaction is very easy

0.6.4 NON-FUNCTIONAL REQUIREMENTS

The non-functional requirements of our game are:

1. Performance: We are ensuring that the game will not abruptly stop or create any difficulties while playing

2. Security: There will be no option for spamming in our proposed game

3. Compatibility: The game is very user friendly and interactive with the players. User can give inputs using the mouse and keyboard only

0.6.5 PERFORMANCE AND PORTABILITY

The game will be designed in such a way so that it can be played in any device such as mobile, laptop, computers, tabs etc.

0.7 PROTOTYPE AND DESIGN

0.7.1 SKETCH

Our proposal is a basic shooting game which will have the unique features of various scenes and a storyline. The main storyline of our game includes a superhero or a savior who will protect the people and the world from different enemies. By different enemies we mean, in different levels there will be different scenes and as the levels proceed the type of enemies will change and the player will build reputation the better he plays.

Our story line begins in IUT where suddenly one day a group of zombies attacks the students. They become afraid and ask the superhero to save their lives from the horrible zombies.

LEVEL 1: Now begins the first level of our game. The scene contains five pillars, cafeteria and the main field. The zombies will appear from different parts as shown in the map, the player will move around and kill them. The zombies have the ability to kill the player as well so he has to save himself from their attack. As the player kills all the zombies he gets a star for his reputation which is the token to proceeding to next level. The map of our game scene is given below:

LEVEL 2: The next level scenes will be towards the register building, auditorium and first academic building where again zombies will be attacking the people and the player has to repeat the same thing. The player has to kill all of them to earn the star and proceed to the next level **LEVEL 3:** Now the rest of IUT will be involved and thus the game will continue. So from level 1 and upcoming levels the player will gradually get access to every part of IUT. When he conquers all the places of IUT he will be considered a superhero.

0.7.2 UI DESIGN

The UI design of our game is shown below:

This is the opening scene of our game as shown in the picture above.

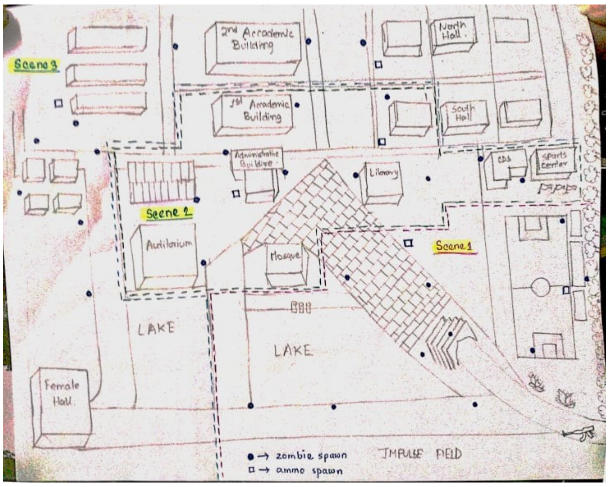

Figure 4: Sketch

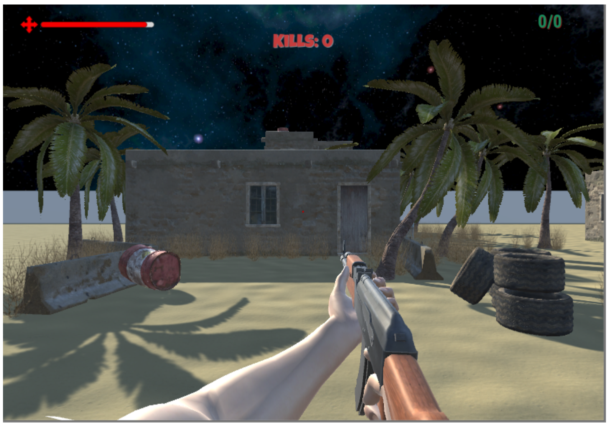

Figure 5: UI design

0.8 PROPOSED SYSTEM ARCHITECTURE

0.8.1 ARCHITECTURAL BLOCK DIAGRAM

Our game is implemented using ‘Incremental Process Model’. This model is basically a combination of both linear and parallel model. In this model , software features are described in the Y axis and time is described in the X axis.

This game was shown to our teachers three times on an incremental basis. On the first week we presented our proposed idea and on the second week we showed our prototype to our teachers and on the third incremental week we presented our final game to our teachers.

0.9 SOFTWARE TEST PLANNING AND EVALUATION

Game testing is the most important part in a game development process. This is the final component that analyses whether the game is ready for launch or not. Following game testing techniques will be followed:

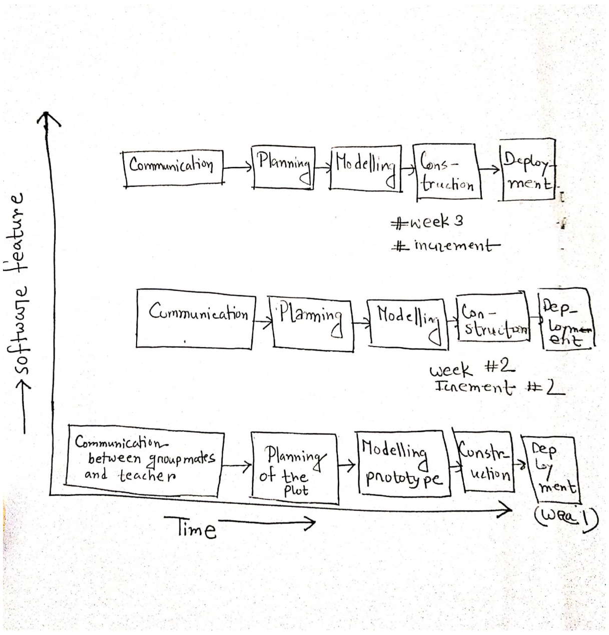

Figure 6: Incremental Process Model

0.9.1 Combinatorial Testing

This is a method of experimental design that is used for commercial software testing and to generate test cases. Applying combinatorial testing to game testing increases test execution efficiency, provide better quality, reduce cost and better phase containment.

0.9.2 Functionality Testing

This literally means the method to identify bugs or errors in a game that may affect the user-experience.

0.9.3 Compatibility Testing

This is used to find whether a game is functioning properly or not with respect to the hardware, graphics and software configuration that the device is built with. It is one of the essential mobile app testing services that checks if a game title is able to run on specific devices.

0.9.4 Play Testing

Play testing is the method of game-testing by playing the game to analyze non-functional features like fun factors, difficulty levels, balance, etc. Here a selected group of users plays the unfinished versions of the game to check the work flow.

0.10 CONCLUSION AND FUTURE WORK

So this is what our proposal is about our game "SHOOT ME IF YOU CAN". We have tried to present the game attractive to the users and have plans to develop it further in the future. We have plans to implement an online version of the game where multiple players will be able to interact and play at the same time. We intend to add more and more features to the game so that it becomes more improved and more attractive. We intend to release the game in play store where it will be available for both android and iPhone users.

1. https://www.youtube.com/watch?v=a1Na540p4gUt=187s

2. https://www.gamedesigning.org/learn/game-plot/

3. https://www.youtube.com/watch?v=SuldrhZ6qUM

4. https://www.red-gate.com/simple-talk/dotnet/c-programming/creating-a-simple-ai-with-unity-and-c/