

## **Chittagong University of Engineering & Technology**

Department of Computer Science and Engineering

Name of the Course: Software Development Project (Sessional)

Course Code: CSE-300

# ***Project Report***

Submitted by:

<b>ID</b>	<b>Name</b>
1704077	Zobair Hossain Fahim
1704087	Sudipta Saha Niloy
1704091	Jawad Bin Anwar
1704093	Sourav Saha
1704109	Yasin Arafat

Date of Submission: 03-01-2022

Table of Contents

1. Introduction.....3

2. Motivation of the project.....3

3. Background.....4

4. Possible Outcome.....4

5. Methodology.....4

6. Implementation.....5

7. Evaluation.....7

8. Conclusion.....8

## **1. Introduction:**

In this project, we are making an arcade shooter game using python. In the early 90s, arcade games were sensational and very popular. A lot of arcade games are still popular among many professional video games streamers. In the early times of video games era, the arcade games were mostly offline and role-playing based. But now since almost all of us has access to fast internet connection, multiplayer arcade games are becoming more and more popular. Since these games are not resource hungry, almost any personal computer is able to run them. Some of the successful multiplayer arcade games are Dota 2 (Developed by Valve Corporation) and League of Legends (Developed by riot games). So, a well develop arcade game still has appeal and attracts a lot of players. Motivated by this, we are designing an offline arcade shooter game which will challenge the player with various obstacles. For making this game, we mainly used the pygame library. Pygame is a cross-platform set of python modules designed for writing video games. It includes computer graphics and sound libraries designed to be used with the Python programming language. Pygame uses the Simple Direct Media Layer (SDL) library, with the intention of allowing real-time computer game development without the low-level mechanics of the C programming language and its derivatives. This is based on the assumption that the most expensive functions inside games can be abstracted from the game logic, making it possible to use a high-level programming language, such as Python, to structure the game.

## **2. Motivation of the project:**

A video game is only enjoyable when it is as much challenging as it is rewarding. The delicate balance between reward and challenge must be maintained perfectly. If the game is too challenging, an average player might lose interest in playing it further. If the game is too generously rewarding, the player will get bored easily. So, our main motivation is to keep the balance between challenge and reward. For that, we have to design the enemy AI with caution. Game development using python with AI to dynamically challenge the player with abstraction can be a great platform to work on software project, with a hint of Artificial Intelligence. It creates a great deal to work on these types of projects to enhance our programming skill and try to strengthen the base for future bigger projects. Therefore, making this project will not only be fun, but it will also build a strong basis for us in software development, implementing AI, algorithm implementation, solving problems, debugging codes and so on which are they very essential skill for any computer engineer in an ever-growing and demanding software industry.

### 3. Background:

Arcade games has a rich history. They are often called the first generation computer based video games. In these times of photorealistic graphics and RPG games, one may ask the feasibility of arcade games. But the truth is there are still a lot of fans for a good arcade game and they are willing to play them. Our game has the capability of fulfilling that need of a demanding gamer. Though it's a basic shooter game, its made very challenging and enjoyable.

### 4. Possible Outcomes:

Here in this project the intention was to do make a fully functional game with an interactive world and a touch of Artificial Intelligence. We are focused on the following output:

- To create a working game
- To gain experience about game developing
- To get introduced with AI implementation in games
- To get familiar with software designing and developing

### 5. Methodology:

The basic working principal of our project is described here, which includes work flow diagram, brief of working procedure and understanding of methods.

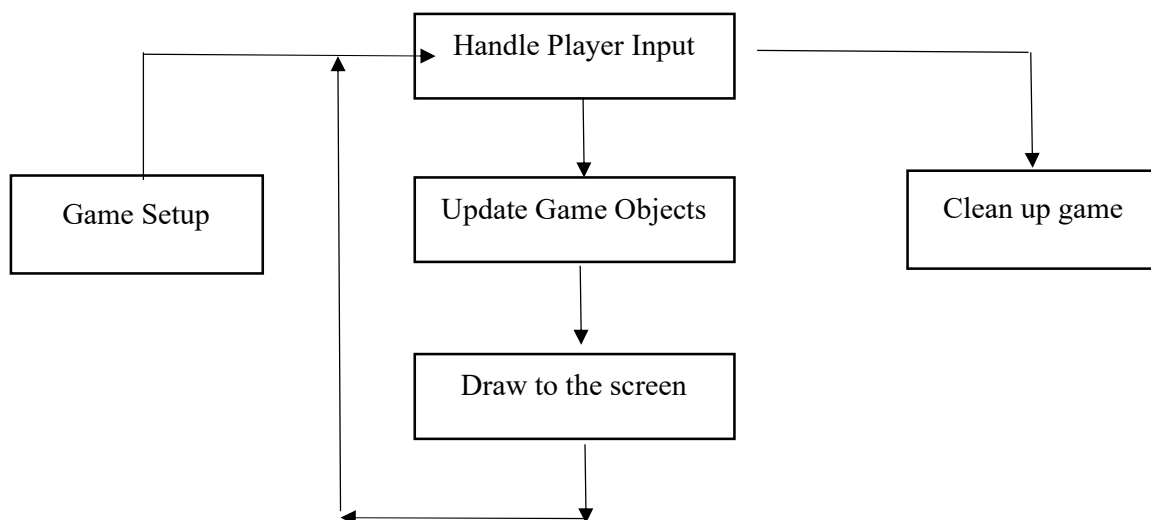


Fig. 1: Workflow diagram of the game

We have made the concept of the game simple at first, on which, we can add complexity as we develop the project. The basic idea is, there will be a player (Human) who will play the role of a soldier. Also, there will be multiple enemy soldiers (Computer). The player will have to eliminate all the soldiers in a level with limited resources and health to get to the next level. Defeating the enemies will be challenging as the player will be constantly challenged to find a solution for a tricky situation. Once the player has passed a certain level, he or she will be taken to the next level which will be more challenging than the previous one. So, in brief, our procedure can be summarized in the following steps:

- Taking input from the user (Movement of character)
- Moving the character as per user command (Up, Down, Left, Right)
- Defeating enemies with limited resources (Bullet, Health)
- Increasing total number of kills
- Implementing basic AI skill to enemies for detecting player and trying to defeat the player

## **6. Implementation:**

We have extensively utilized the pygame library to implement every aspect of this game. Pygame is a vast library with a lot of resources for implementing any games. Therefore, the whole source code is written in python. The game is vastly object oriented. Meaning we have utilized classes and instances to implement various features of the game. In total we have implemented 11 classes. All of them are explained briefly below:

### **1. Soldier Class:**

This class implements the player and enemies. It also implements how the player can control the character, run, jump and shoot and how the enemy AI will work. This class inherits properties from Healthbar, Ammo and Grenade classes. This is the biggest and most complex class in the whole code

### **2. World Class:**

The gaming environment is defined by some rectangles. These data are preloaded into .csv files. We have created this class to create the gaming world.

### **3. Decoration Class:**

This class decorates the world by adding still images, colors and background variations and music to the game.

4. Water Class:

Water was initially under decoration class but when the player is in contact with water, the player will die. So to implement this logic particularly, this class is used.

5. Exit Class:

This class is implemented to execute termination of the game

6. Itembox Class:

Various refill items such as bullets, grenades and ammos will be found across the world. This class implements that.

7. Health Bar Class:

This class keeps track of the player health and enemies health. Also it updates the player health according to damage and health refills.

8. Bullet Class:

This class keeps track of the bullet, its refill and how much damage it will inflict upon soldier or enemies. Also it includes the bullet trajectory animation.

9. Grenade Class:

This class keeps track of grenades. It also includes the animation of grenade throwing trajectory.

10. Explosion Class:

This is the simplest class of the game. It just replaces the image of a grenade with an image of an explosion.

11. Screenfade Class:

Screen Fading effect is implemented in this class. It is used to exit to the game or move to the next level.

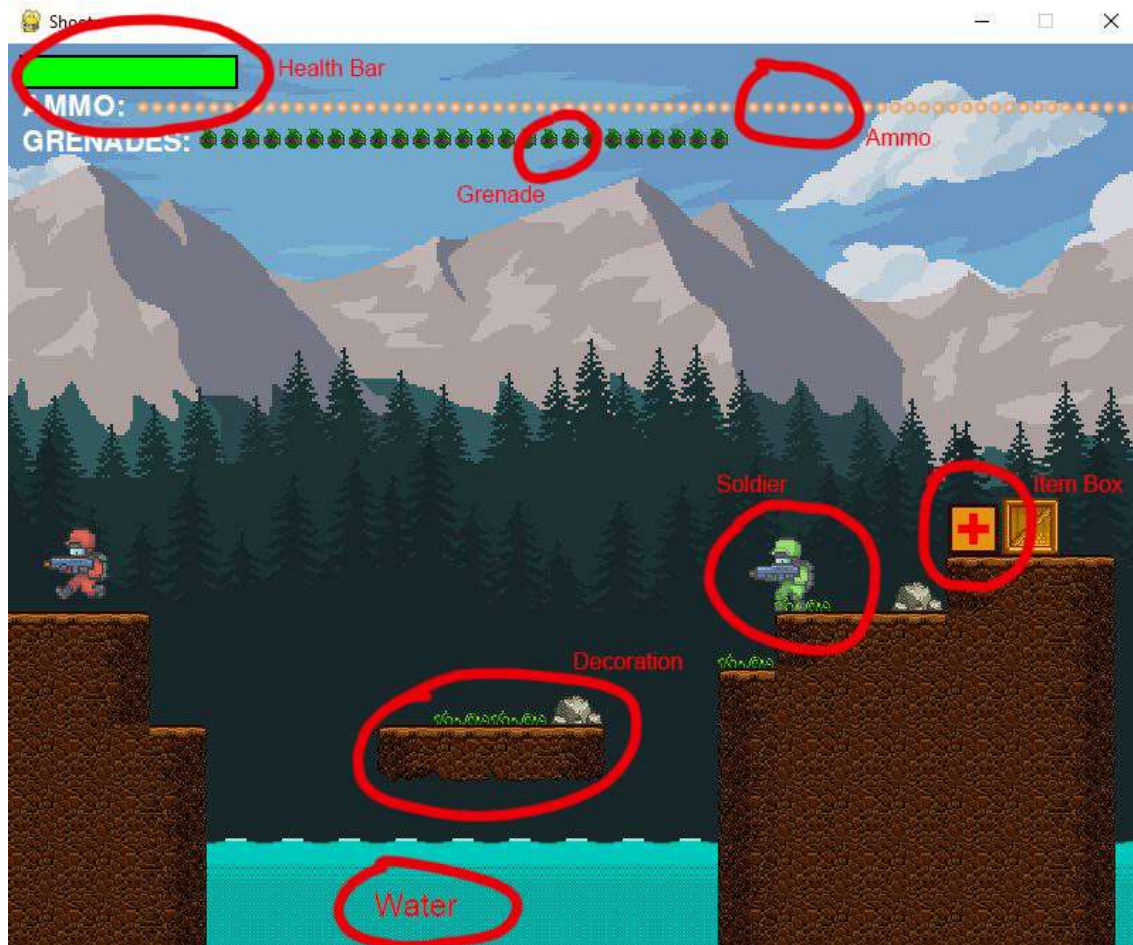


Fig. 2: How various classes contribute to the game

## 7. Evaluation:

After implementing the code in Visual studio code and running it, we have found out that the code ran successfully. The game was playable. We also found a lot bugs in the game but since each function was implemented by a particular class, it was not very hard to pinpoint the bugs in the program and fix them accordingly. We faced many challenges while making the animations for the game, as this game do not have any “Real” animations. We just looped through the images very fast using the internal clock to give the player a feel of animation. The player control is working correctly, it’s a typical “WASD” control type game. The enemy AI is also working correctly. It can detect the player and shoot him when the player is in range. Overall, the implementation of the game is successful. This game consists of only 2 levels as a test version.

## **8. Conclusion:**

Initially we had only 8 classes in the game. But as we progressed farther, the necessity of several more classes we seen and we have implemented those classes. In the end, we had in total 11 classes. The animation was a challenge but we have overcome that. Next stage of this game will be to make it playable between two human players. Once that is achieved, we can move on to the next level and work on the game to introduce an online multiplayer mode.