Bilkent University

Department of Computer Engineering

CS 319 - Object Oriented Software Engineering Project

Bullet Drop

Analysis Report

Group 2F

Ulaş İş, Alperen Erkek, Ömer Faruk Ergün, Abdullah Seçkin Özdil

Course Instructor : Bora Güngören

Table of Contents

1. Introduction

1.1. Purpose

This report is for a description a Software Requirements Specification (SRS) for our game called “Bullet Drop”. This report explains the game by giving its summary of software, clarifying functional and nonfunctional requirements, showing its system models, which are called use case model, dynamic model and class model. Also, report includes user interface and some mock-ups for the game.

1.2. Content Of The Project

Our project is a basic arcade game called “Bullet Drop”. The audience of this game is both younger and adults who like to spend their spare times by playing these kinds of arcade games. This game isn’t complicated for its audience. If player wants to succeed in game, he/she needs some simple movement calculation. Even if he/she doesn’t succeed, game will be a funny challenge for who tries to achieve levels.

The game is basically a shooting game where the target is invisible at the beginning, that is, in the first try, the player will shoot the bullet without knowing where the target is. There will also be some outside powers that will affect the movement of the bullet. After firing the first bullet, the player will know where the target is and what outside effects change movement of the bullet and how. Then, the player will try to hit their next shot/s with this information. There will be different maps and different weapons available in the game.

1.3. Terms and Definitions

While the project is basically an arcade game, it has some specific terms and users may not use them before.

Arcade Game: An arcade game is a type of game which has short level system and requires basic control for the whole levels in game.

Gravity: Gravity is simply weight. In more detail perspective, gravity is a force of attraction from the objects to the center of earth.

Wind: Wind is a natural movement of the air from one direction to the other.

Weapon: Weapon is a general name of some defending or attacking tools such as gun, knife or just a stone. In this game users are using gun to shoot the target.

Bullet: Bullet is a metal tool which is using with a gun, riffle or a small firearm to fire.

Target: Target is changeable in itself. A target might be a person, a place or an object for attacking to it/him/her. In this game target is an object.

2. Proposed System

2.1. Overview

The aim of the player in Bullet Drop is basically to hit the target by arranging the position of the weapon and shoot. However, this won't be an easy task. First of all, there will be outside forces such as wind and gravity that will affect the bullet. Moreover, at first try, the target is invisible, therefore, the player is required to track the route of the bullet and see where the target is. Then, the player needs to shoot according to the location of the target and the outside effects.

There will also be levelling system as the player hits the target, there will be new maps with different backgrounds and outside effects change. There also be different types of weapons so that the player can choose the best weapon for different wind and gravity values. The player won't have limited shots, the will continue until the player hits the target and after that, new backgrounds with different effects will be available. The game will end after the player manages to hit the targets in all of the maps.

2.1.1. Game Play

The player will be able to play the game with keyboard. The position of the weapon can be moved in a vertical line. The player can move the weapon up and down by pressing up and down arrow keys. When the position of the weapon is arranged, the player can shoot the bullet by pressing space bar. For the menu, the player can select different options in menu by using mouse.

2.1.2. Leveling

As the player hits the target, there will be new levels available. In each level, the location of the bullet, the background, and the effects of outside forces will change. For example, as the player levels up, the gravity and the wind will be stronger. For the first shot, the target will be invisible as well. Therefore, hitting the target will be challenging in each level and the player won’t be able to hit the target by memorizing the target and the effects from previous levels. There won’t be any limit for shots in any level. The player will try until s/he hit the target and proceed to the next level.

2.1.3. Weapon and Bullet System

Player will be able to choose five different weapon, each will be different in terms of their qualities. For example, if current map’s weather conditions is really bad and gravity is high in that map, player would want to choose a weapon that has higher power so that bullet’s speed will be higher, in result the bullet would feel effects of these external forces less. However, more power a weapon has, more deviation it will have. So choosing a weapon with high power in a nice weather would bring unnecessary risk.

2.1.4. Force and Game Difficulty

The forces will be the main factor why player would miss their shots. As the bullet travelling, it will be under effect of these forces. In the easiest level, bullet won’t feel a force. However, as the level increases there will be more forces- that is gravity and wind in order. The placement of the wind forces will be random so that player will not be able to memorize it. Power factor of these forces will depend on the map. Additionally, wind forces will able to blow from the upper area and the lower area.

2.2. Functional Requirements

2.2.1. Start Game

Player can start the game by using “Start Game” button in the main menu. After he/she pushes button, weapon option page will immediately appear on the screen.

2.2.2. Pause Game

Player can stop game while he/she is playing by using ESC button on keyboard. In pause mode he/she can continues game, click options button or quit the game.

2.2.3. Options

In options menu, player can manage game sound and resolution. He/she can volume up and down or mute if it’s needed. Also game resolution can be selected as full screen or default.

2.2.4. Help

If player get confused about process of game, there will be help button for a quick tutorial or short information for game. It can be chosen from main menu.

2.2.5. Credits

For the credits button, small window will be appeared and there will be developer’s names.

2.2.6. Exit

Player can quit from game whenever he/she wants by using exit button in both main menu and pause mod.

2.2.7. Additional Requirements

Our functional requirements are same with the previous one. We didn't add additional requirements to our game.

2.3. Non-functional Requirements

2.3.1 Game Performance

Our main concern is keeping the game's performance stable for reasonable standards. Since our game will have lots of animations and sound effects, it will be important to manage these footages and use them in non-performance impact wise. Since we'll be using Java, it will be important to keep track of these footages. We also want to keep the input lag, i.e. the delay between a given input and the response to it, as minimal as possible.

2.3.2 Graphical Appealing

One of the most important aspects of a game is how it looks to the their players. We'll be using pictures created just for the our game through various programs and make sure they all integrated well with each other. We want our graphics to have certain color palette and has its unique way of showing. In addition, we may use particle effects in order to have more various in graphical showcase.

2.3.3 User-Friendly Interface

It's easy to get lost in today's modern games. We want to achive simple -yet effective user design. We want to achieve it having minimal of different screens but enough to difference between settings.

2.3.4 Extendibility

We want our game to be have open ended. This means other people can work on it and develop it further. There can be more forces, different maps and different weapons. All can be added in simple yet neat fashion.

3.2.5. Additional Requirements

For the nonfunctional requirements part, we are trying to improve our game performance by using JavaFx. We also make a little changes in user interface part.

2.4. Constraints

2.4.1 Implementation of The Game

The game will be implemented by using Java which is really suitable for object orianted software designing by its flexibility to use classes and create hierarchical order easily.

2.4.2 User Interface and Animations

a. Adobe After Effects

It is an efficient and user friendly tool to create the animations that we will use in the game. It has too many different options to make our interface more enjoyable and eye-catching.

b. Adobe Photoshop

Well-known tool to design the graphs and images in the game like buttons or some texts etc. This will be also really helpful for us to make the interface of the game more attractive.

2.5. System Models

2.5.1 Scenarios

Scenario 1) Player opens up the game, selects its favorite settings then clicks play.

Scenario 2) After selecting “play game”, player sees the map and chooses his favorite weapon through “left-right” arrow keys and presses “space” to select. Then he or she places his/hers desired weapon through “up-down” arrow keys and then presses “space” to fire. After some time seeing how forces affects the bullet, he/she sees where the target is and ultimately misses the target and clicks retry.

Scenario 3) Player memorizes where the target is and how the forces(if there any) affecting the bullet and re-adjusts his weapon through “up-down” arrow keys and presses “space” to fire again. Player hits the target and levels up.

2.5.2 Use Case Model

This section provides information about the main use case model of Bullet Drop game, detailed use case explanations are included below.

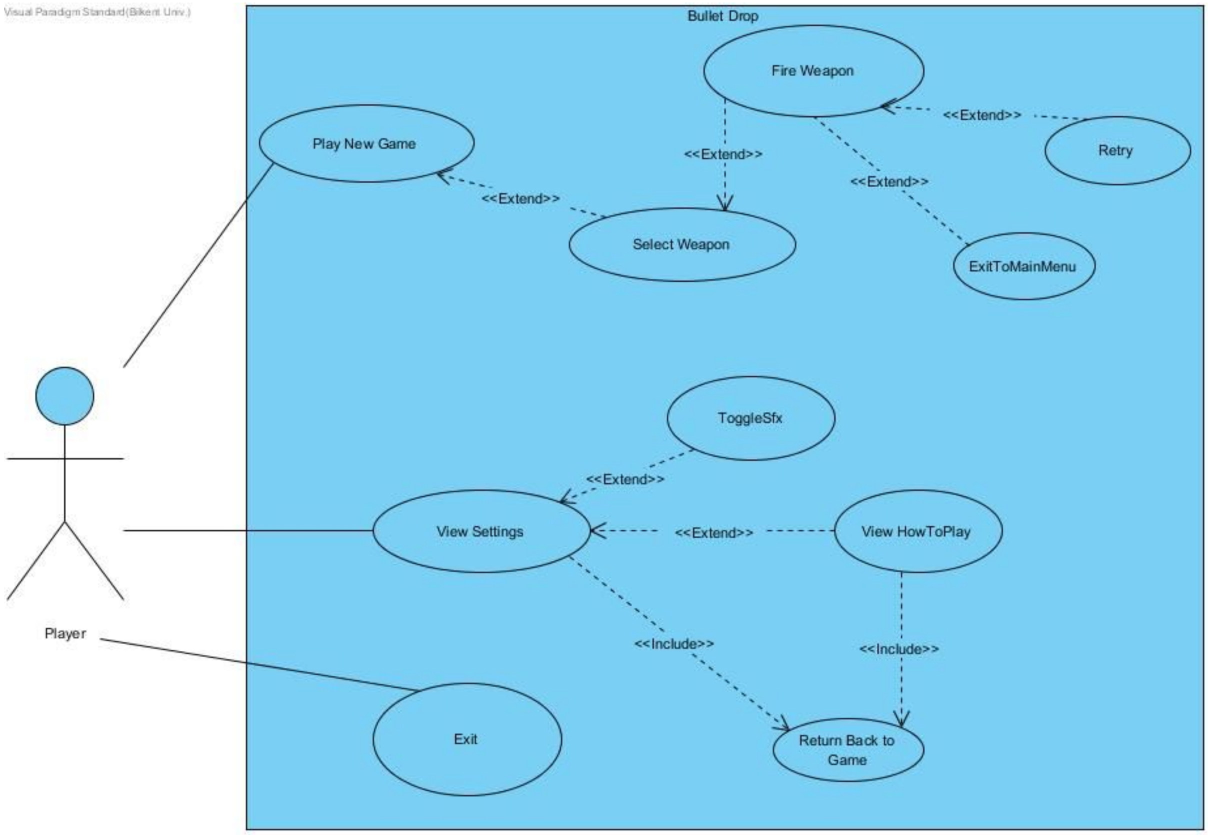


Figure 2.5.2 – Illustrates the use case model of Bullet Drop

2.5.2.1 Exit

Use Case Name: Exit

Primary Actor: Player

Entry Condition: Player selects “Exit” from Main Menu.
Exit Condition: -

Main Flow of Events:

1.Player selects “Exit” in Main Menu

2.Game closes the game succesfully

Alternative Flows:

2.5.2.2 Settings

Use Case Name: Settings

Primary Actor: Player

Entry Condition: Player selects “Settings” from Main Menu.
Exit Condition: Player selects “Back” in Settings.

Main Flow of Events:

1.Player selects “Settings” in Main Menu
2. Game switches to Settings Screen and shows the settings.
3. Player changes settings for his/her desire
4. Player clicks “Back” button to return Main Menu

Alternative Flows:
A. If player desires to return main menu at any time:
A.1. Player selects “Back” button to return main menu.
A.2. System displays Main Menu.

2.5.2.3 Play Game

Use Case Name: Play Game

Primary Actor: Player

Entry Condition: Player selects “Play Game” from Main Menu.

Exit Condition:

- Player presses“Escape” key in Weapon Select Screen
- Player selects “Exit to Main Menu” in Result Screen
- Player finishes all levels.

Main Flow of Events:

1.Player selects “Play Game” in Main Menu

2.Game switches to Weapon Selection Screen.

3. Player chooses his/her weapon and then clicks “Next”.

Alternative Flows:

2.5.2.4 Select Weapon

Use Case Name: Select Weapon

Primary Actor: Player

Entry Condition:

- The player starts a new game by clicking “New Game”.
- The player proceeds to next level by hitting a target.

Exit Condition:

- Player selects “Exit to Main Menu” in Weapon Selection screen.
- Player finishes all levels.

Main Flow of Events:

1.Player presses left or right arrow to go through different weapons

2.Player selects a weapon by clicking “Select”.

3. The weapon placement screen opens

Alternative Flows:

A. If player clicks “Exit to Main Menu”, game switches back to Main Menu.

2.5.2.5 Fire Weapon

Use Case Name: Fire Weapon

Primary Actor: Player

Entry Condition:

- The player presses “Select” in Weapon Selection Screen.
- The player presses “Retry” to go back to Weapon Selection Screen.

Exit Condition:

- Player selects “Exit to Main Menu” in Result Screen
- Player finishes all levels.

Main Flow of Events:

1.Player presses “Select” in Weapon Selection Screen

2.Game switches to Weapon Placement Screen.

3. Player places his weapon through “up-down” arrow keys.

4. Player presses “Space” to fire weapon.

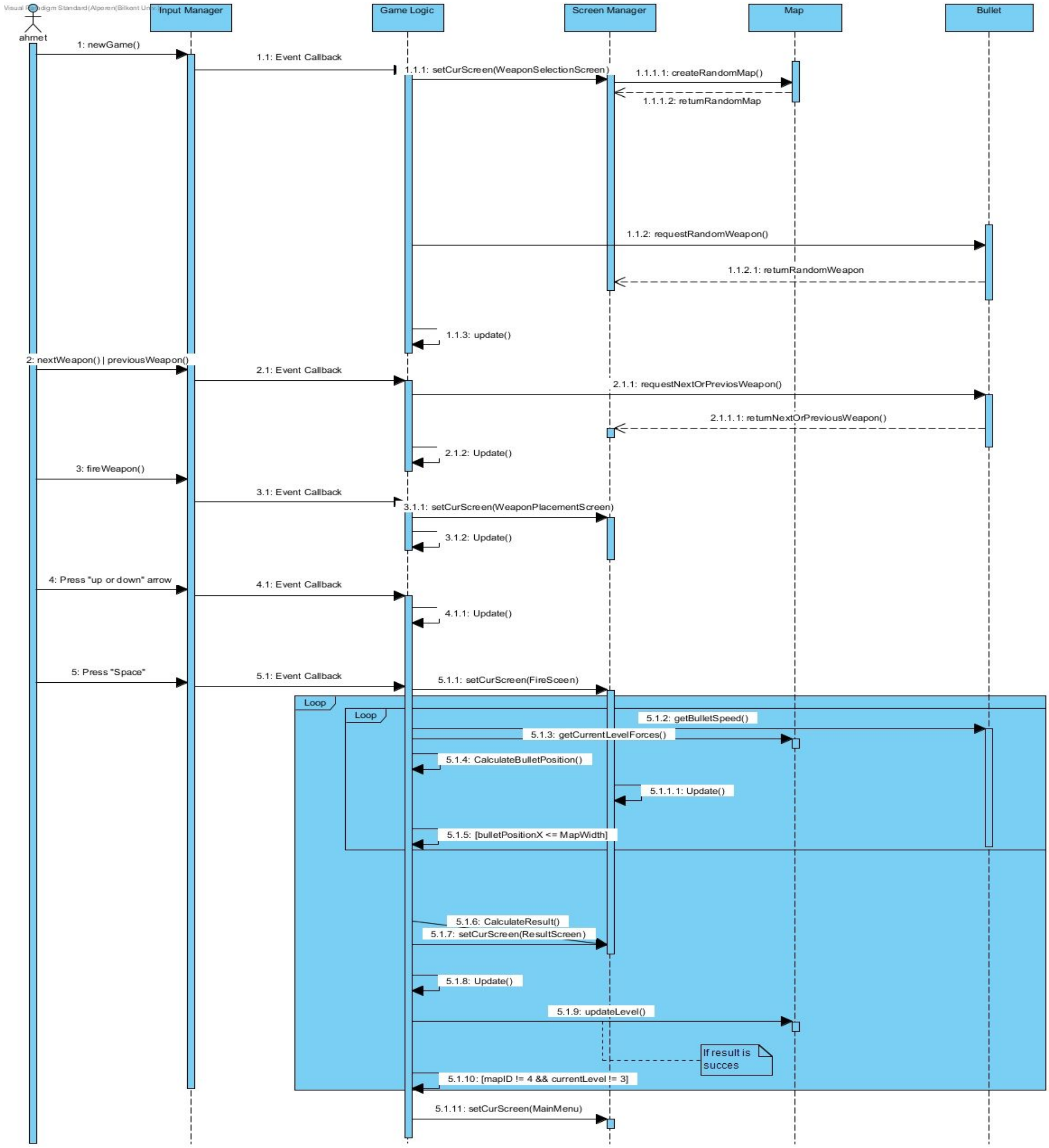
2.5.3 Dynamic Models

2.5.3.1 Sequence Diagram

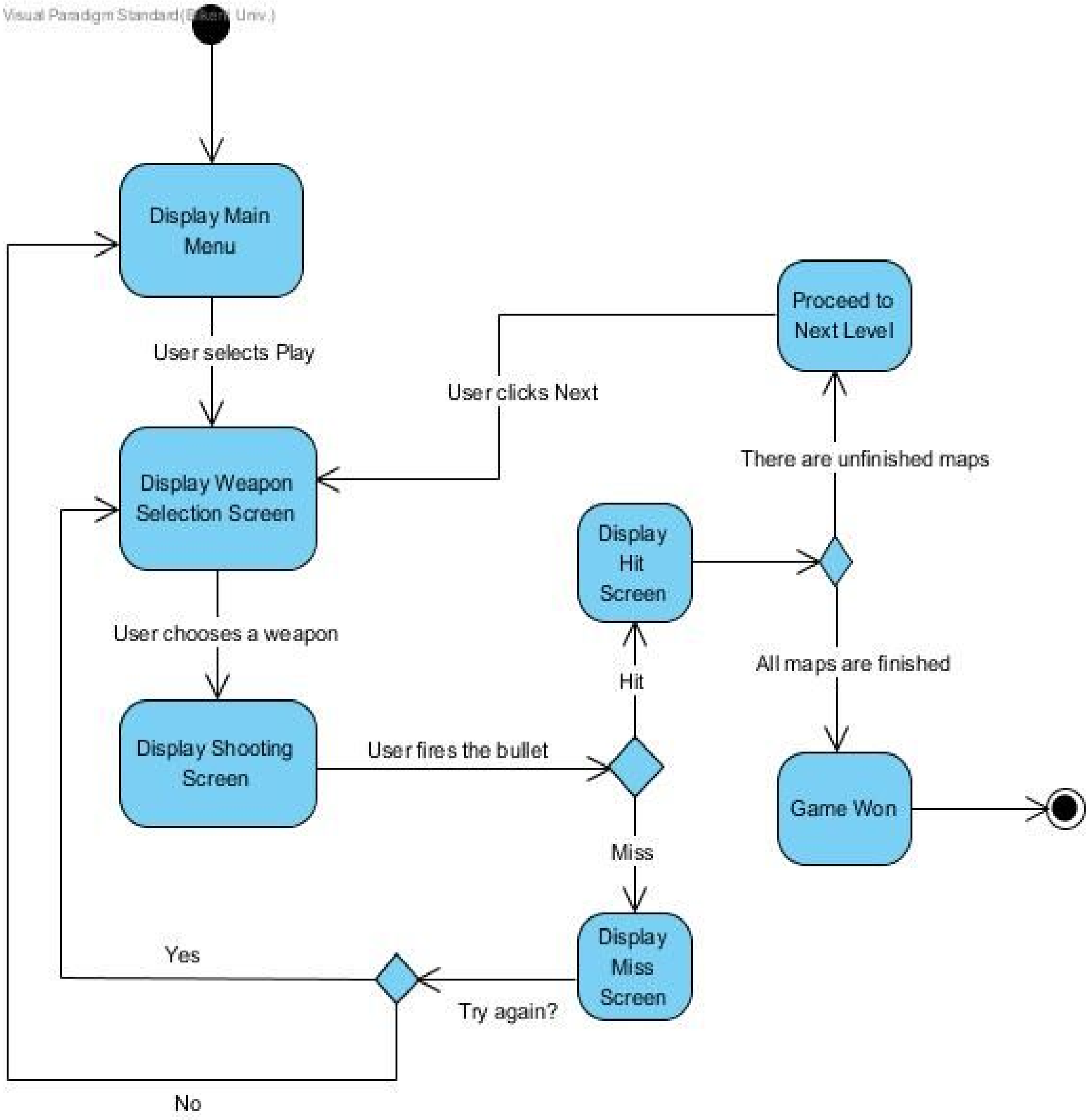
Below sequence diagram shows the sequence of the game from the user starts the game to the point where user finishes the game and returns the main menu. Recall that user has to click “retry” or “exit to main menu” after each missed shot. For simplicity, we did not show this interaction in diagram.

Scenerio : Start Game

In main menu, Ahmet clicks “Start Game”, after that he selects weapon in weapon selection screen by pressing left or right arrow and press space.. Then, he arranges the position of the bullet and fires the bullet. If the shot is a miss, Ahmet tries again if he chooses to.. If it’s a hit, then he proceeds to the next level with different force levels until he hits the target in all of the maps. In that case, he returns to the main menu.

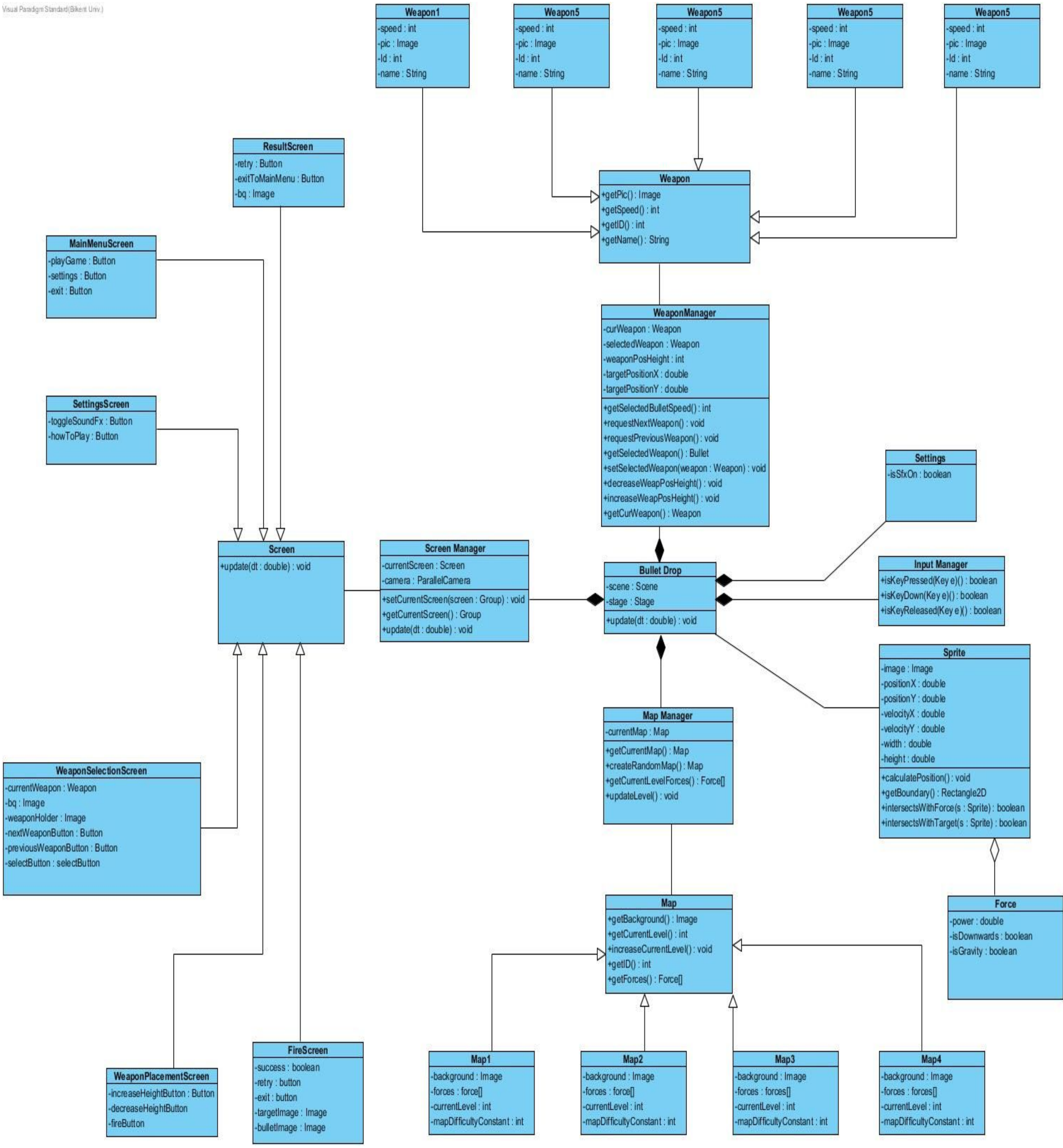

Visual Paradigm Standard(Alperen(Bilkent Univ.))
ahmet
Input Manager
Game Logic
Screen Manager
Map
Bullet
1: newGame()
1.1: Event Callback
1.1.1: setCurScreen(WeaponSelectionScreen)
1.1.1.1: createRandomMap()
1.1.1.2: returnRandomMap
1.1.2: requestRandomWeapon()
1.1.2.1: returnRandomWeapon
1.1.3: update()
2: nextWeapon() | previousWeapon()
2.1: Event Callback
2.1.1: requestNextOrPreviosWeapon()
2.1.1.1: returnNextOrPreviousWeapon()
2.1.2: Update()
3: fireWeapon()
3.1: Event Callback
3.1.1: setCurScreen(WeaponPlacementScreen)
3.1.2: Update()
4: Press "up or down" arrow
4.1: Event Callback
4.1.1: Update()
5: Press "Space"
5.1: Event Callback
5.1.1: setCurScreen(FireSceen)
Loop
Loop
5.1.2: getBulletSpeed()
5.1.3: getCurrentLevelForces()
5.1.4: CalculateBulletPosition()
5.1.1.1: Update()
5.1.5: [bulletPositionX <= MapWidth]
5.1.6: CalculateResult()
5.1.7: setCurScreen(ResultScreen)
5.1.8: Update()
5.1.9: updateLevel()
If result is succes
5.1.10: [mapID != 4 && currentLevel != 3]
5.1.11: setCurScreen(MainMenu)

2.5.3.2 Activity Diagram



2.5.4 Object and Class Model

Below diagram explains the basic class diagram and its relation



In the design of the program GameLogic is the main class that will proceed the combinations and interactions between other classes. The Screeen Manager class chooses which screen to displayed in a current time according to user's choices and result of actions like missing the target in the playing room of the game etc. These screen options are MainMenuScreen which is the main menu that opens when the game start, SettingsScreen a screen that sound and resolution can be selected, CreditsScreen, WeaponSelectionScreen the room that weapon will be selected by user, WeaponPlacementScreen which is the first screen when the game starts and user choose place of weapon, FireScreen that is the screen where the game will be played and fire action will done and ResultScreen that provides retry option to user if he/she misses the target or provides an option to pass to next level if he/she succeed.

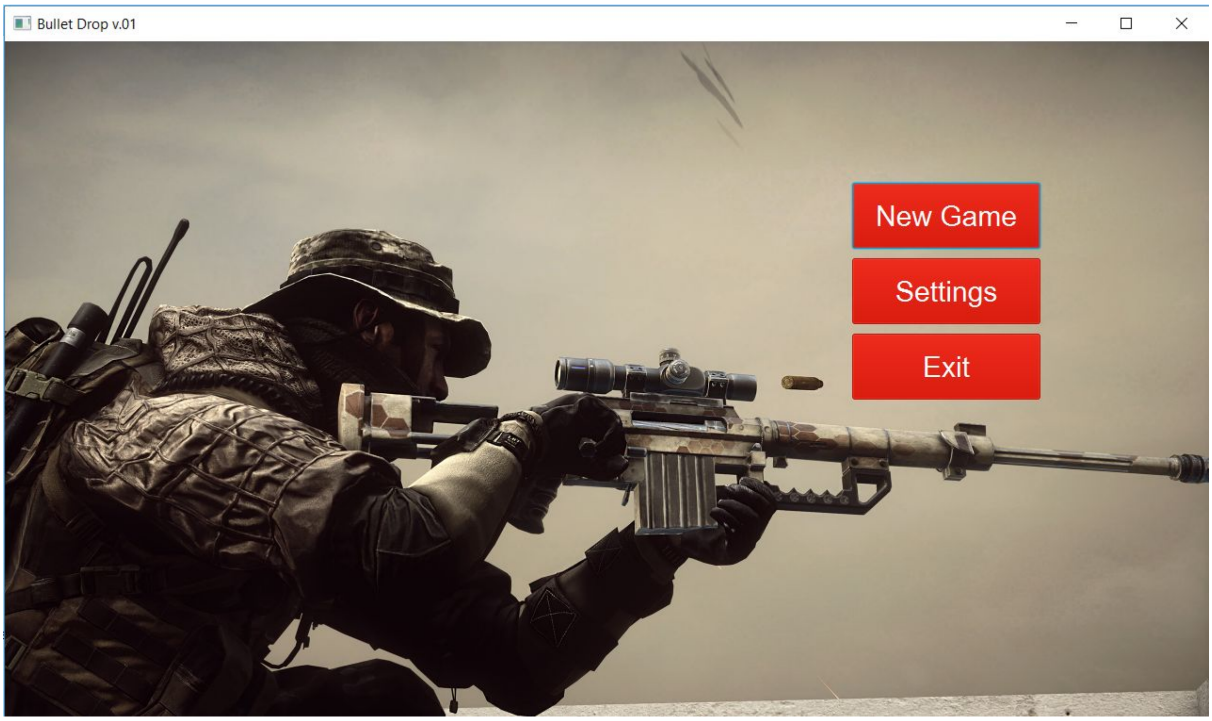
Bullet class keeps the main options for the weapons and how bullet will act according to this weapon unders the effect of forces like gravity or wind. Weapon1, Weapon2, Weapon3 and Weapon4 extends this class which are the weapon options that will be opened by user in a proceeding time that he/she passes the levels.

InputManager is the class that will take and check user instructions from user and inform game logic about user's choices.

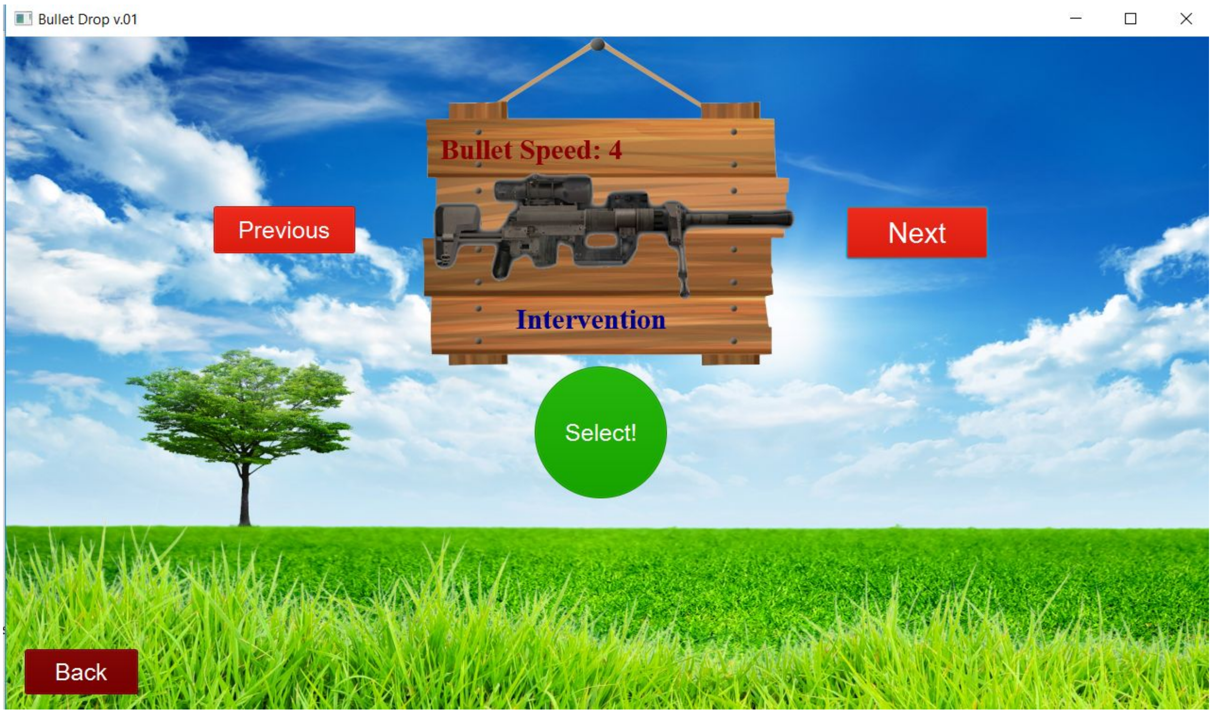
Map is the class that keeps the wind and gravity forces according to different maps which will have different effects on bullet. Map1, Map2, Map3 and Map4 classes extend Map class. They all have identical background images which will make the game more optional and enjoyable. Their difficulty level will be determined by forces and they will be opened in a time by user as which is the result of competetive game. For instance; when the game first start user will play on Map1, when he passes three levels he will be able to see Map2. This will continue like this process.

2.5.5 User Interface

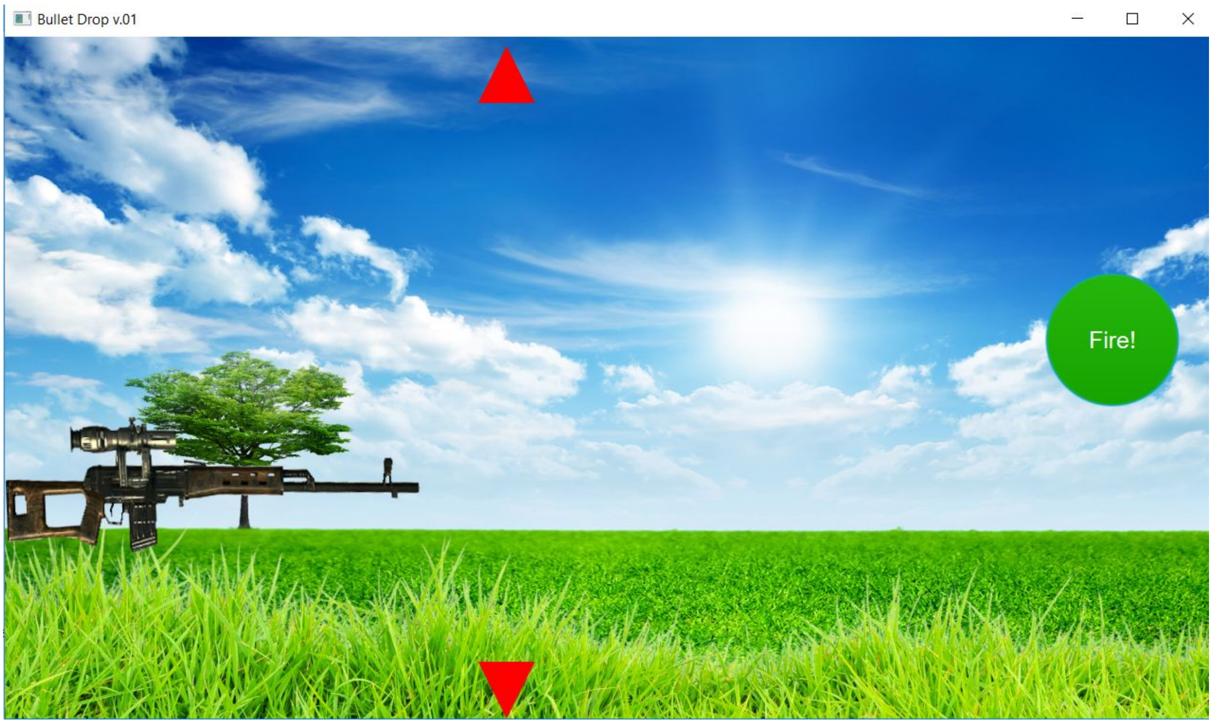
When player opens the game, the first thing that comes out is a main menu, which contains 4 options: Play, Settings, Credits and Exit. Player can start playing the game, change the settings of the game, read credits or exit the game from this menu



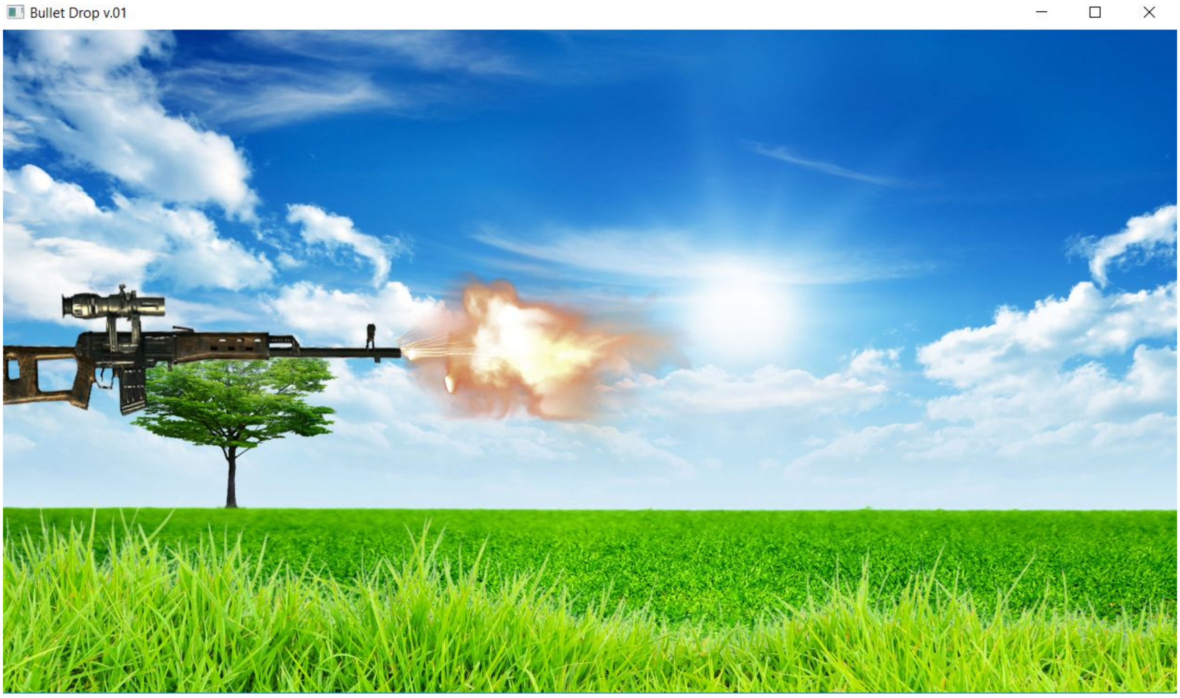
When player clicks Play option, the game starts with a weapon selection screen. In this screen, player can see different weapons by using right and left arrow keys and then select the weapon they want by pressing space bar.



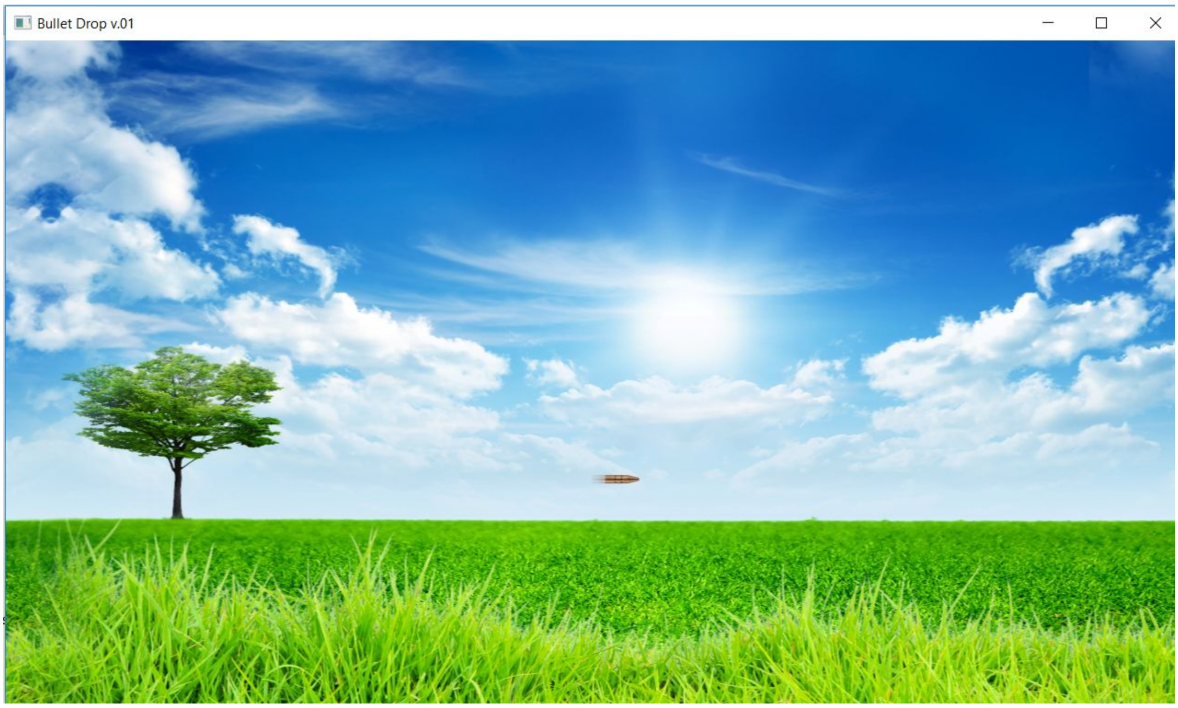
After choosing the weapon, player can arrange the position of the weapon with up and down arrow keys.



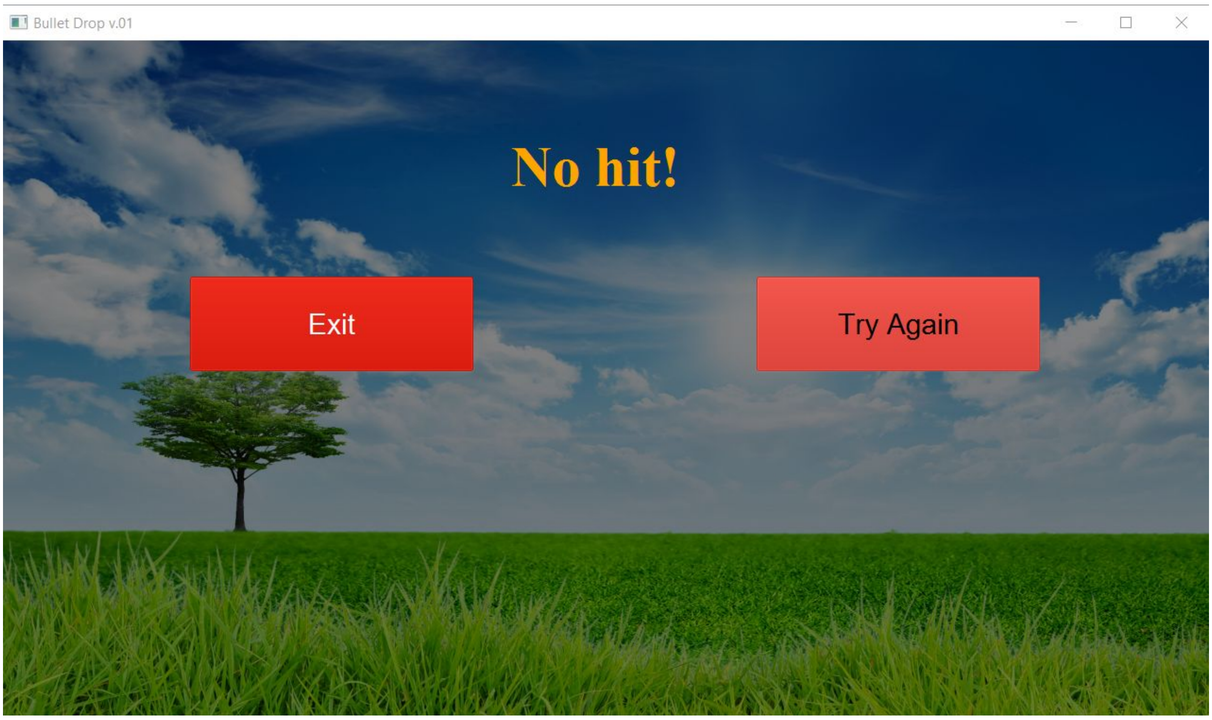
Then, player can shoot the bullet by pressing space bar. This is how it looks when the bullet is fired.

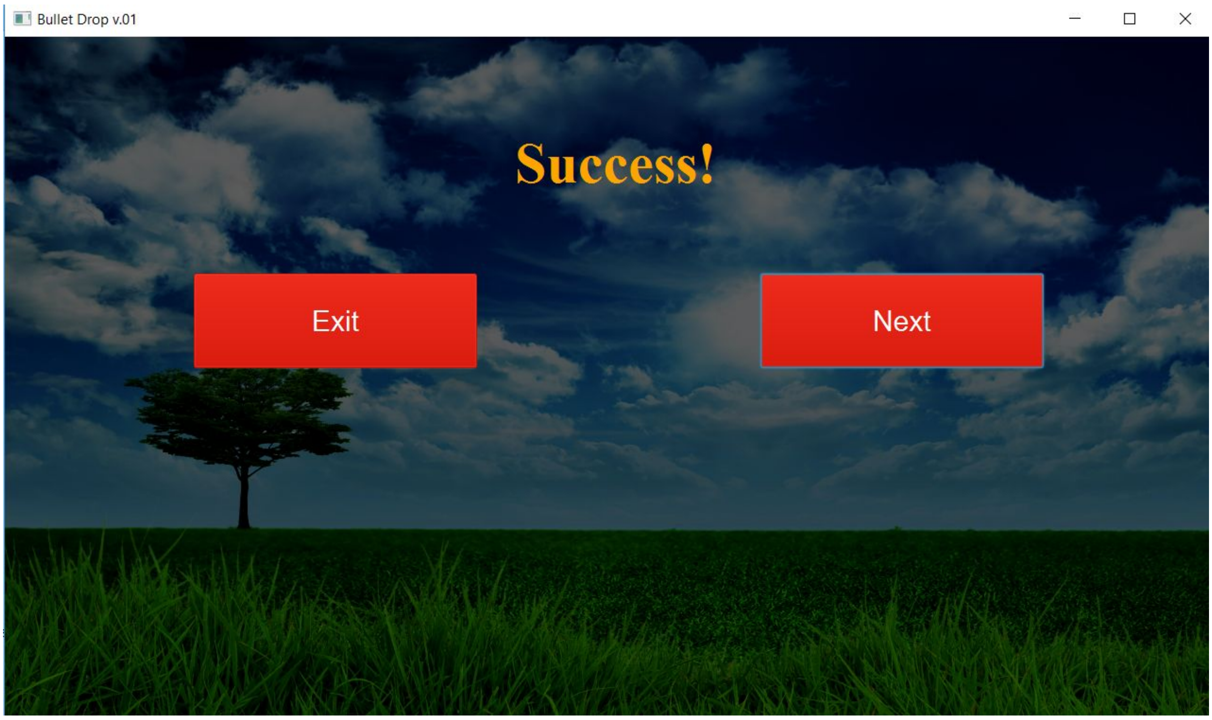


Then, the bullet will follow a route. However, the route will change due to the outside forces. The player needs to track this route and determine the direction and the power of these forces.



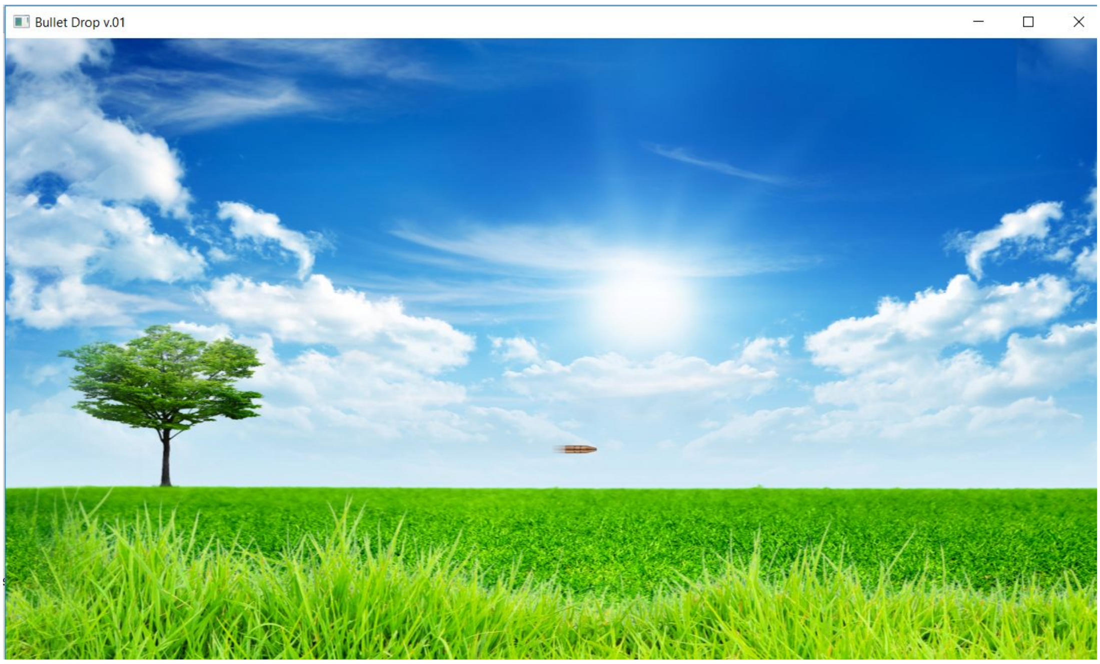
If player misses the shot, in the miss shot screen, player will be shown the location of the target and given two options; Exit to main menu or try again. If player hits the target, the next map will be loaded and the player will select weapon again.



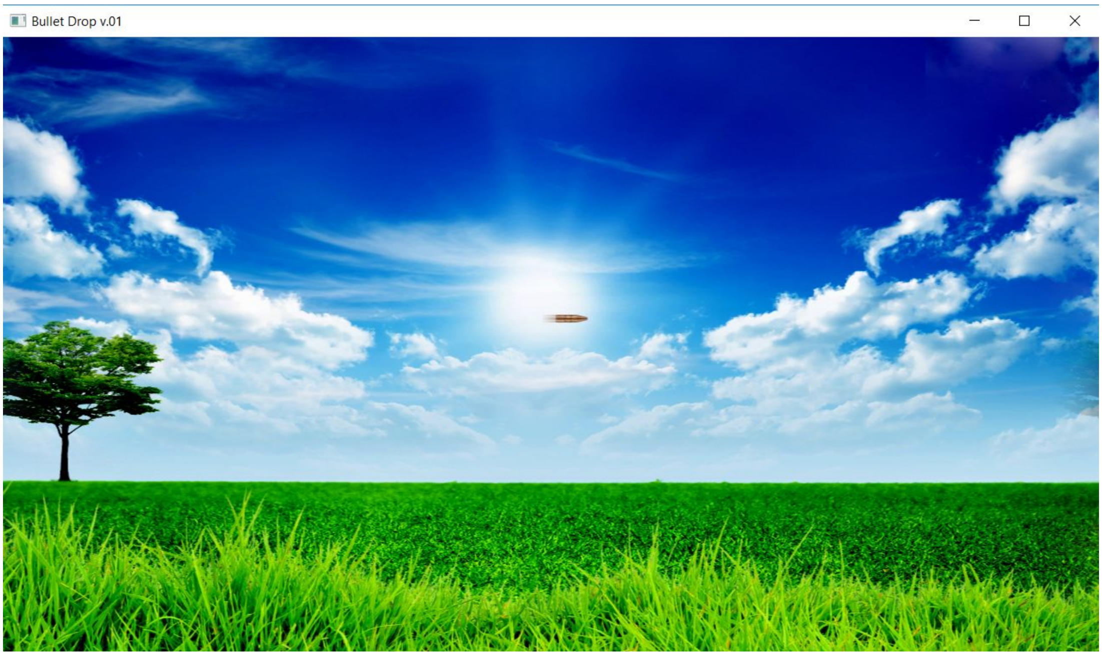


2.5.5.1 Maps

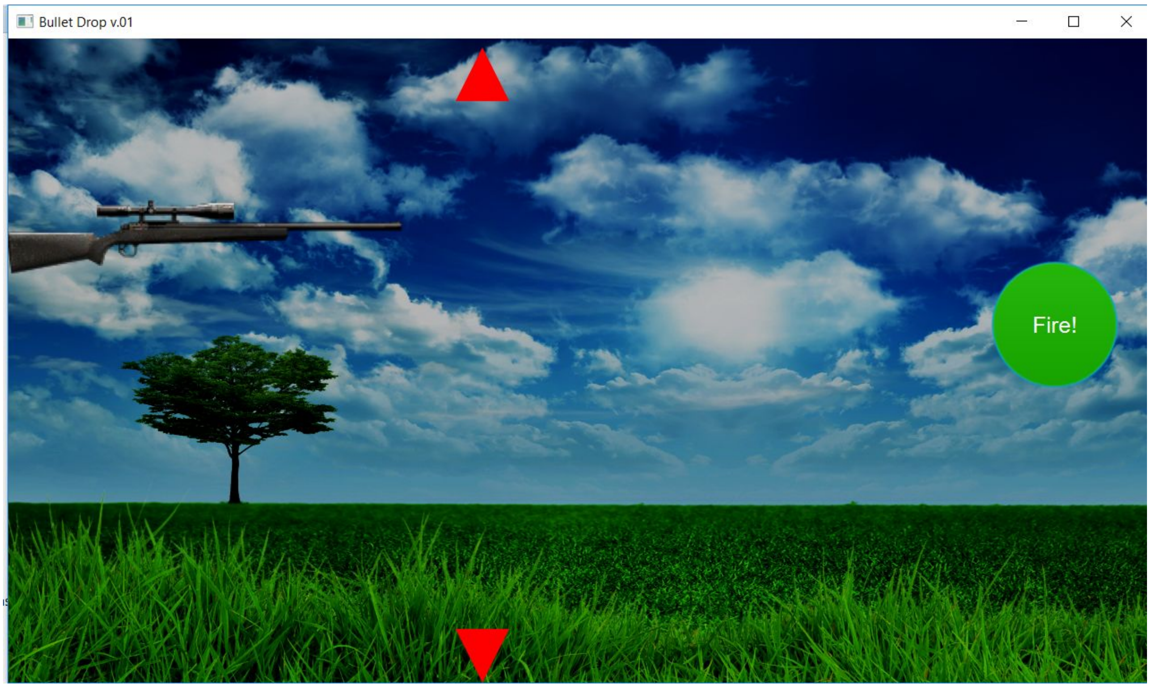
In this game, there are four different maps and each map has its own special force. In each map, we have 2 different forces which are called wind effect and attraction of gravity but these two forces have different abilities to make challange for player.



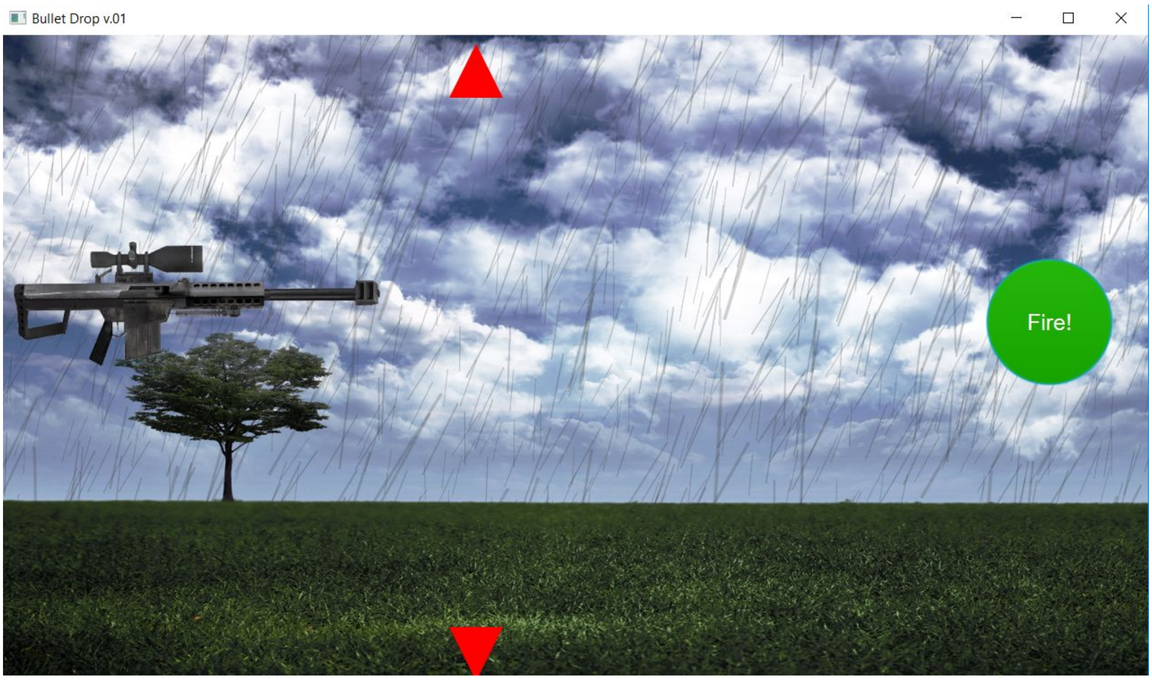
1st Map: In this map player will try to shoot target without any effect.



2nd Map: 2nd Map will include wind and gravity.



3rd Map: For this map wind and gravity will be more powerful.

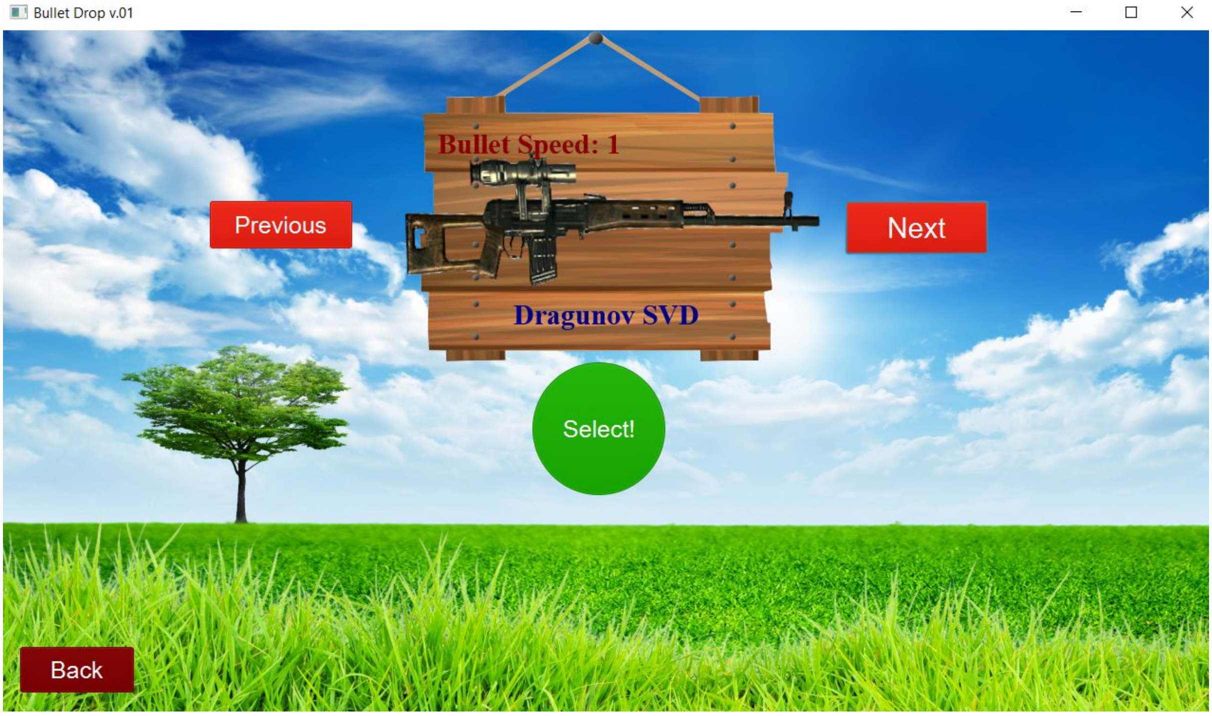


4th Map: Finally for the last map, wind and gravity will be in maximum level.

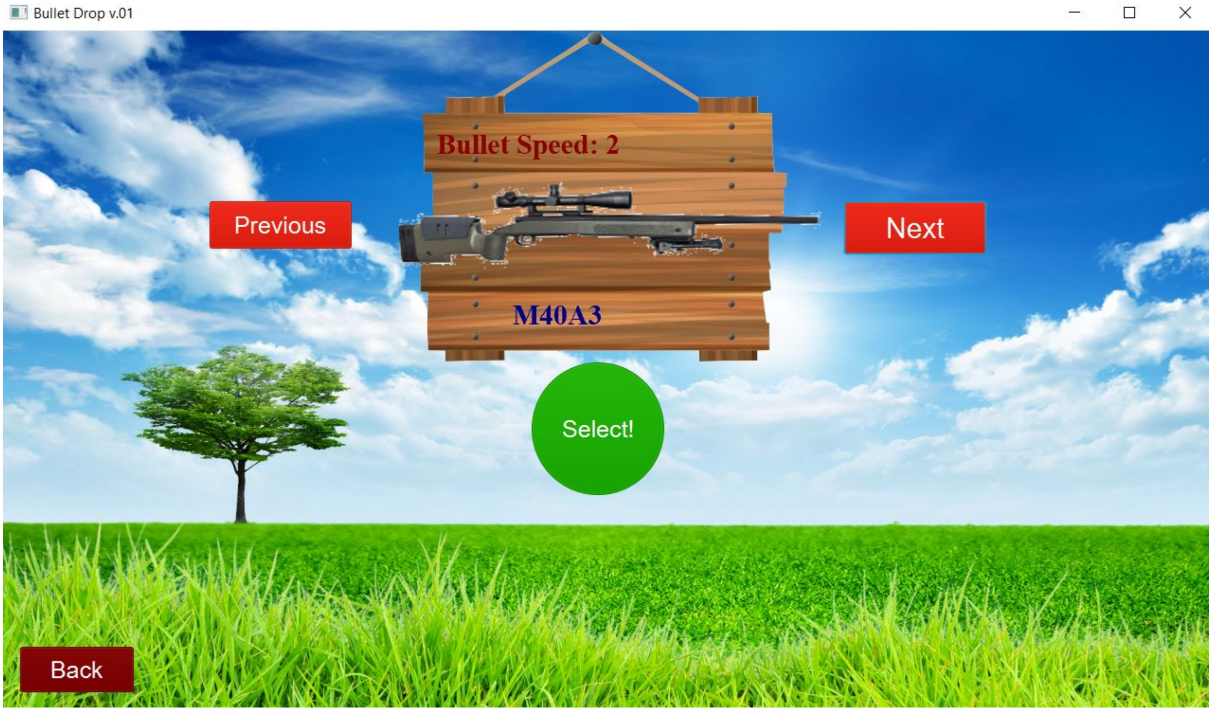
2.5.5.2 Weapons

In “Bullet Drop” we will have five different weapons and each have different visuality and range power. Here is their pictures according to their range and power. As we explained before, player can choose what weapon he/she needs for each map.

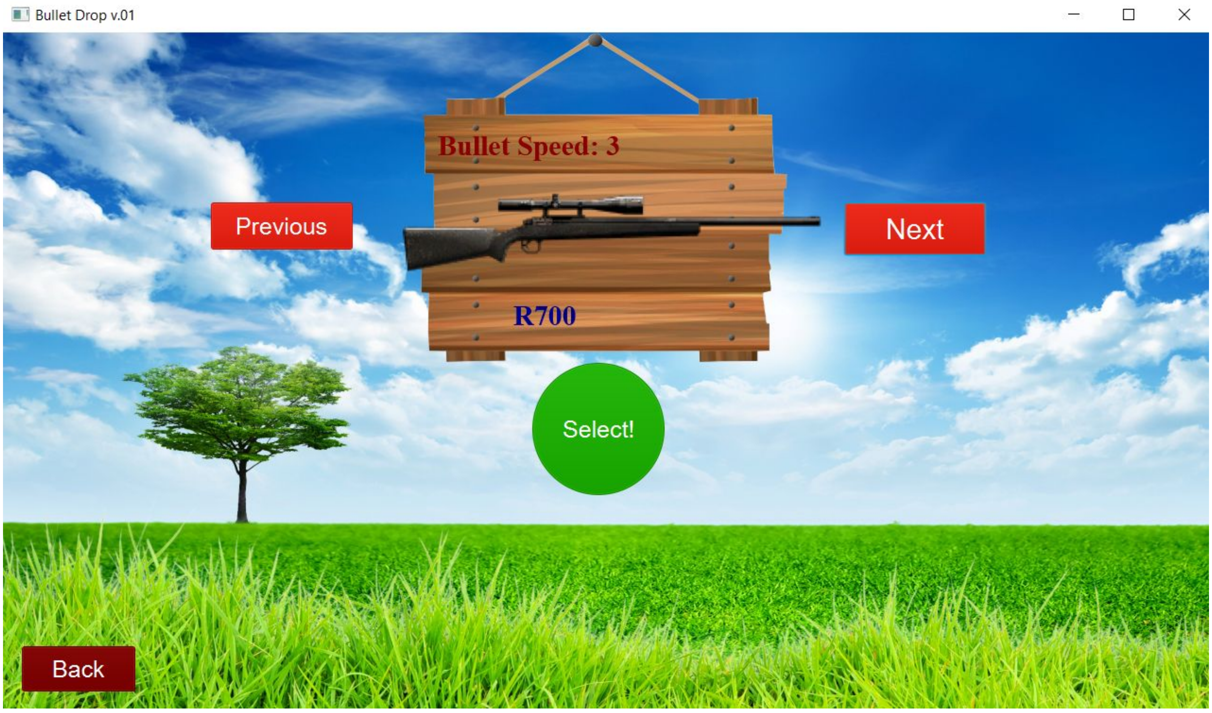
1st Weapon:



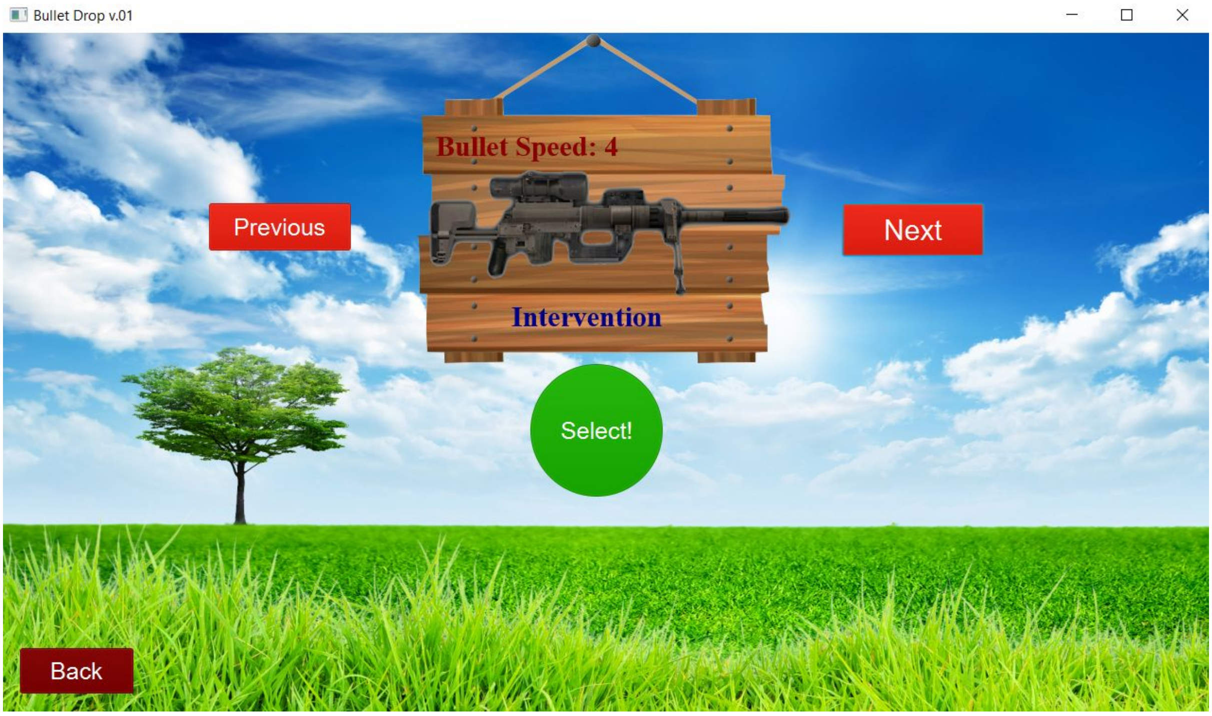
2nd Weapon:



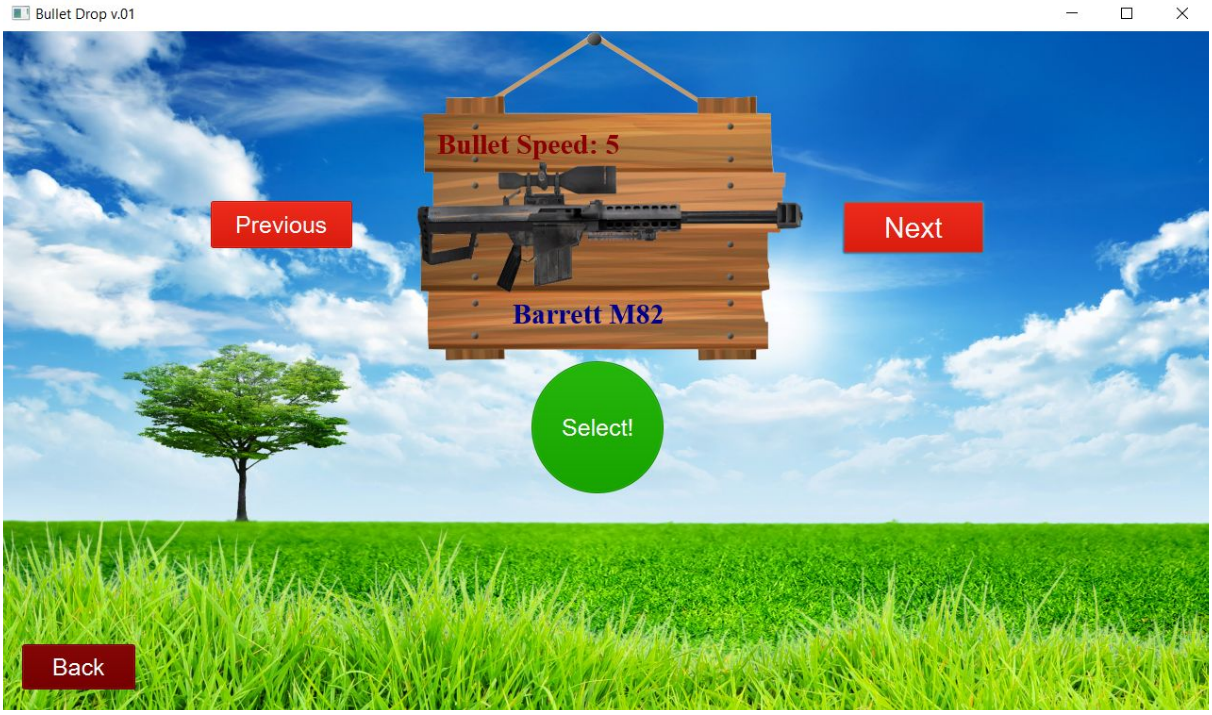
3rd Weapon:



4th Weapon:



5th Weapon:



3. Improvement summary

As we discussed before, we decided to use JavaFx instead of Swing. The reason behind this was it was easier for us to implement a main "tick" method that will update every screen and game objects once it gets called. Another reason behind this was, it's easier to organize screen elements and implement animations through JavaFx.

We have implemented interfaces for our Screen,Map,Weapon objects. The reason behind it is simple, every user that wants to add a new one will do it under proper rules so that it will be future proof. Besides, it gave us enough flexibility to implement our managers.