Bilkent University

CS319 - OBJECT-ORIENTED SOFTWARE ENGINEERING

Final Report

Group 1E : Tank Zone

Aybüke Ceren Duran

Hakan Türkmenoğlu

Mert Sezer

Supervisor: Eray Tüzün

Progress of Implementation

1. Status of the Project

We have successfully completed *'TankZone'* game as we mentioned in analysis and design reports except a few changes.

2. Implementation During Iteration 1

At the phase of Iteration-01, we designed our game in primitive state such as moving tanks, their barrels and angles. The game featured AI that could differentiate between the factions so that it avoids friendly fire. Additionally, collision system was added so the tanks could destroy each other with their bullets. Blackhole powerup was implemented, which creates a gravitational pull towards itself. When the tanks hit a blackhole, they get destroyed. We also designed GUI components of our game. Our main concern in Iteration-01 was to decide main ingredients of our game.

3. Changes in the Implementation During Iteration 2

At the phase of Iteration-02, we improved our design model of our game. There were major changes in the implementation compared to our design report. In the progress of Iteration-02, we added and removed some features of our game so that we satisfy design goals of our game.

3.1 Architectural Design Changes

Compared to our architectural design in iteration-01, we changed our architectural design from ground-up so all the diagrams had to be changed.

We went from traditional inheritance based design to Entity-Component-System which is a
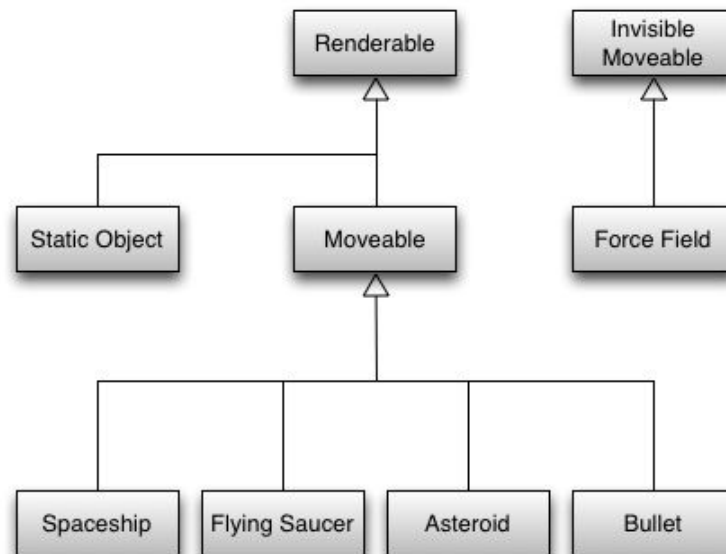

Figure 1: Inheritance based design [1]

composition based design. For instance, suppose we have a base GameObject that provides x,y coordinates and texture. Then we provide StaticObjects that don't move and DynamicObjects that move. Later on we discover that we want an invisible ForceField that should move but it's not clear where this entity should come from. Then we remove the texture property from GameObject. Then we discover we need similar kind of variations for things related to angles. The list goes on. This is illustrated in Figure 1.

What we can do is instead of a making an IS-A relationship we can have a HAS-A relationship. This will provide flexibility for us.

In Entity-Component-System,

- **Component:** Component just holds data like models and doesn't contain business logic. E.g, PositionComponent holds x and y coordinates.

- **Entity:** Adds or removes components and is a general purpose object. E.g Tank entity has components PositionComponent, VelocityComponent, TextureComponent.
- **System:** Runs through the list of entities in the program and applies business logic to the entities that has specific components. E.g MovementSystem handles the movement for entities that has PositionComponent and VelocityComponent.

This way we will eliminate most of the exceptional codes for entities that don't map to the general idea of our object, which means adding specialized objects is easy. Figure 2 illustrates this composition based mechanism.
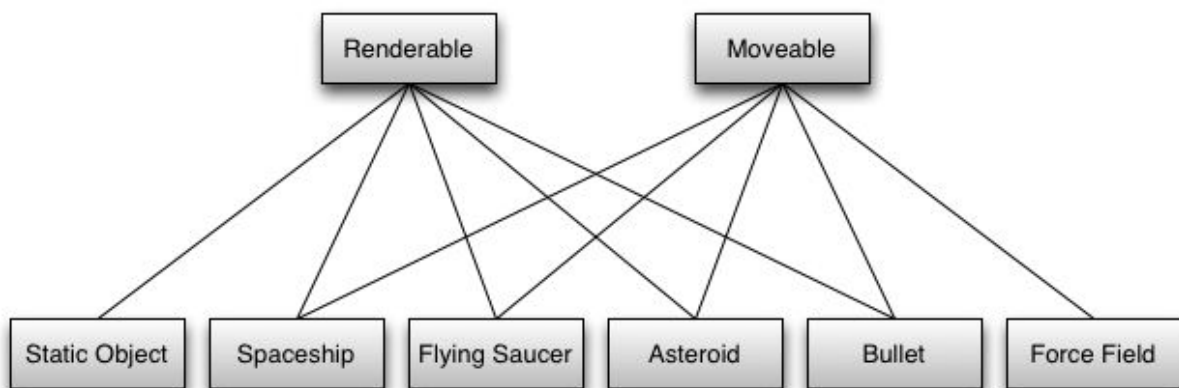

Figure 2: A composition based design [2]

3.2 Online Database Scores

We added the feature of online database score feature so that we increase the feeling of competition between different users of our game.

3.3 Removal of 'Achievements' Feature

We decided to remove the *'Achievements'* feature from our game because this feature increases the complexity a lot while not bringing a lot of game elements, and it contradicts the feature of maintainability of our game.

3.4 Remembering the Sound On/Off Feature and the Username

We added the feature of remembering the last state of the sound of the specific user in our game. With this feature, every time the user opens the game, s/he will not spend time on changing the sound his/her intended way. Similarly we also store the username for similar purposes.

3.5 Implementation of Map and Game Modes

Previously, the game didn't have a functioning background. We added a camera system that follows the player wherever they go. This way we can achieve a much more dynamic gameplay by allowing player to traverse the world that is bigger than the window itself. Moreover, this map features things like obstacles and tiles. The map looks differently depending on the climate. Each climate has its own specialties. The map changes depending on the gameplay mode, for instance Capture The Flag mode has castles while Free for All mode doesn't.
The map is generated via Cellular Automata Algorithm. Depending on the climate type, some parts of the map are covered with grass/snow/sand while the others are background tiles. In temperate climate for instance, the tanks can move with full speed in the dirt (background) and the grass (in the forest). However since forest features a lot of trees, rocks etc. (obstacles), the tank cannot move easily without taking hit. For winter climate, the snowy tiles do not introduce obstacles that much but they actually make the tanks slippery.
Similarly, easy, medium and hard modes are implemented.

3.6 Implementation of Powerups and Scores

Powerups and scoring are implemented on top of the existing system. The way the scores are calculated is as follows:

$$score = A * numberOfTanksDestroyed + B * numberOfPowerupsTaken$$
$$- C * healthPointsLost - D * gameTimeInSeconds$$

where A, B, C and D are constants.

3.7 Implementation of Upgrades and Levels

Player can upgrade the tank. Player's experience points are directly calculated from the score (except the time penalty). The experience point required for each level grows polynomially, $O(n^2)$ to stimulate interest in the game. Player can spend their upgrade points in the pause screen.

3.8 Improvement of Collision System

There was a collision system in iteration 1 but that it allowed tanks to collide with each other. This was fixed by adding another collision type that does not include damage, i.e when tanks collide they don't overlap with each other but this collision does not damage the tanks themselves unlike a bullet collision.

3.9 Implementation of Spawning System

For the Capture the Flag mode of our game, we added Spawning System so that tanks could be generated during the specific time frequencies, and the combat between enemy tanks can increase during our game.

3.10 Health Bars

Tanks and castles have a health bar underneath in this iteration which shows their current health compared to their maximum health.

4. Users's Guide

4.1 System Requirements

Since Tank zone will be implemented in Java 8, users will be have to install JRE (Java Runtime Environment) 8. Minimum requirements include:

- **Operating System:** Windows 7, macOS Sierra or Ubuntu 16.04 (or equivalent Linux desktop distribution).

- **RAM:** 1024 MB

- **GPU:** Any graphics card with at least 1024 MB VRAM

- **CPU:** Intel® Core™2 Duo Processor E4500

- **Resolution:** 1280x720

4.2 Installation

Currently the only way to play the game is to compile either with the help of an IDE (Eclipse, IntelliJ IDEA…) or gradle. Work directory from the project settings should be set to "src/desktop/assets" in order to let the game load the assets. In the future we plan to distribute the game with a JAR file.

4.3 Controls

- W: Move upward

- A: Move left

- S: Move downward

- D: Move right

- User can also move with arrow keys:
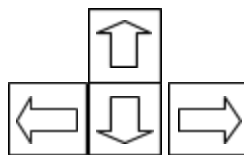
- Mouse: Aim

- Mouse Left Click: Shoot

4.4 Game Overview

There is a map lower right of the screen. There is only a player icon on the map; which is showing where the user's tank is in the playground. Player can move around the playground

with using W,A,S,D keys or arrow keys. Player has a tank and can take aim through moving the mouse, can shoot by left clicking the mouse. There are enemies in the playground; which also have tanks. There are also stationary objects with different shapes and sizes in the playground. Stationary objects and tanks' health are visible on their icons and decreases when a bullet hits them or when there occurs a collision between them and other tanks or stationary objects. If these stationary objects or tanks take damage that is equal or more than their health, they will disappear from the playground. If a tank destroys a stationary object or another tank; its' score increases depending on the size of the stationary object or another tank's level. If the tank can collect enough score in a level, its' level increases by 1.

Both scores and levels are in the lower side of the screen. There are some visible power-ups in upper left of the screen and when the tank's level increases by 1, the tank can use these power-ups by clicking and increase its' health regen, max health, body damage, bullet speed, bullet penetration, bullet damage, reload or movement speed by 1 by clicking "+"; which is in the right of these power-ups. Also there are some surprise power-ups in the game and in some moments, the game is showing some of these surprise power-ups to the tanks and ask them whether they want to choose a surprise power-up among the showed ones. If they choose one power-up, they continue to game with that. If they reject these showed surprise power-ups, they can't choose the same surprise power-ups again unless the game shows them these surprise power-ups again.

Also there are 2 type of game modes: Free For All, Capture The Flag. In Free For All game mode, the main purpose is collecting the highest score and killing all enemy tanks. Different from free for all game mode, in Capture The Flag game, there are 2 teams. Each team has a castle and the main purpose is destroying other team's castle before they destroy your team's castle. Castles have also health and their health decreases if they collide with enemy team's tanks, stationary objects, and if other team's bullet hits them. Also in Capture The Flag game mode, a tank, a stationary object and the castle can't take damage from the same team's tanks and stationary objects.

4.5 Menu Overview

Pause Menu: When player decides to pause the game, this menu will appear. Player can choose continuing the game or quitting the game and going to main menu.

Game Over Menu: If player completes the game successfully, or his/her tank is destroyed, this menu will appear. Player can play a new game or can go to main menu.

Main Menu: In Main Menu, there are 5 buttons. The user can press one of these buttons.

Start Game Menu: If player presses "START GAME" button in the main menu, this menu will appear. In this menu, the player must select the game mode(Free For All or Capture The Flag) and then the climate and then the difficulty level before starting to game.

Settings Menu: If player presses "SETTING" button in the main menu, this menu will appear.

The player can choose the controller settings and can arrange the sound options in that menu.

Help Menu: If player presses "HELP" button in the main menu, this menu will appear. In this menu, there is a document explaining purpose of game, player controls, power-ups, traps and the gameplay screen showing the amount of remaining health, weapon, etc..

Credits Menu: If player clicks "CREDITS" button in the main menu, this menu will appear. In this menu, there are the names of the developers of the game.

4.6 Game Screenshots
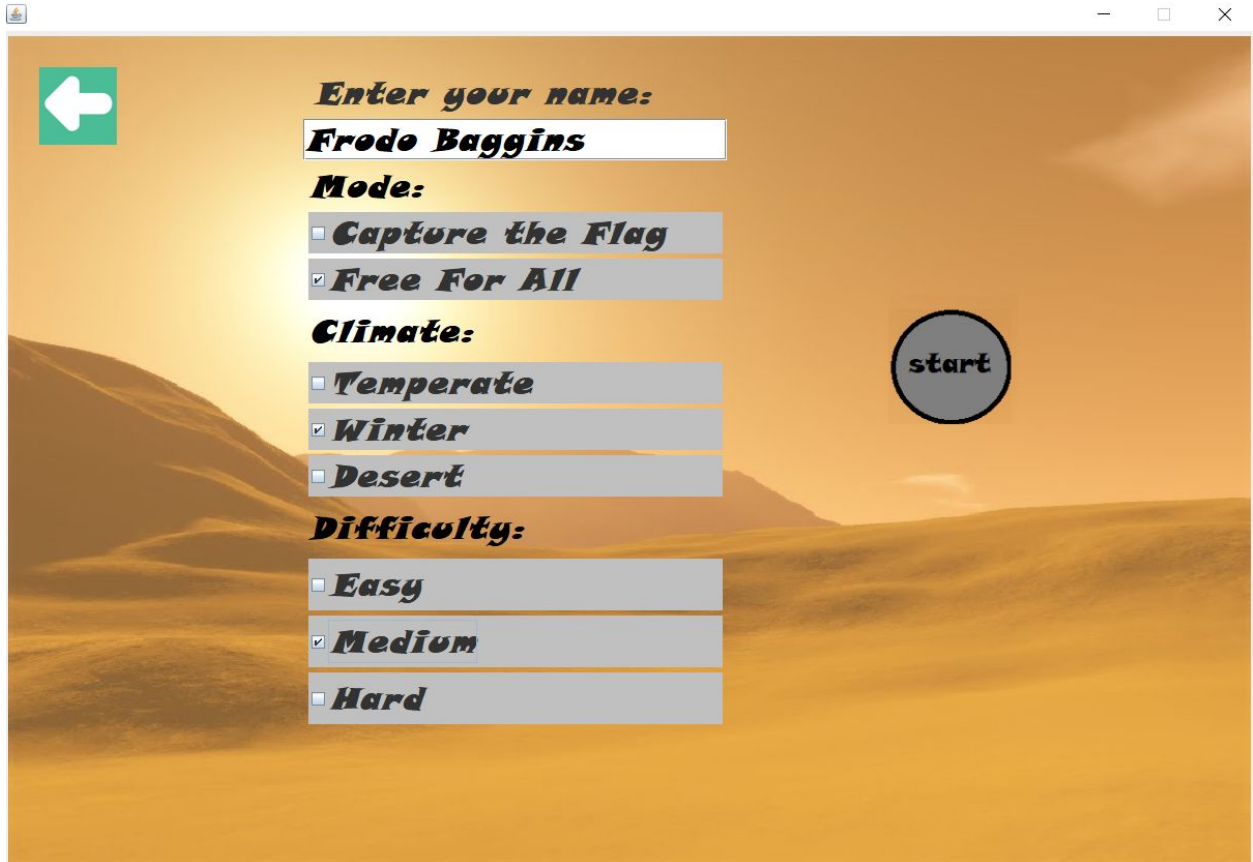
Figure-01: Menu screen of TankZone Game

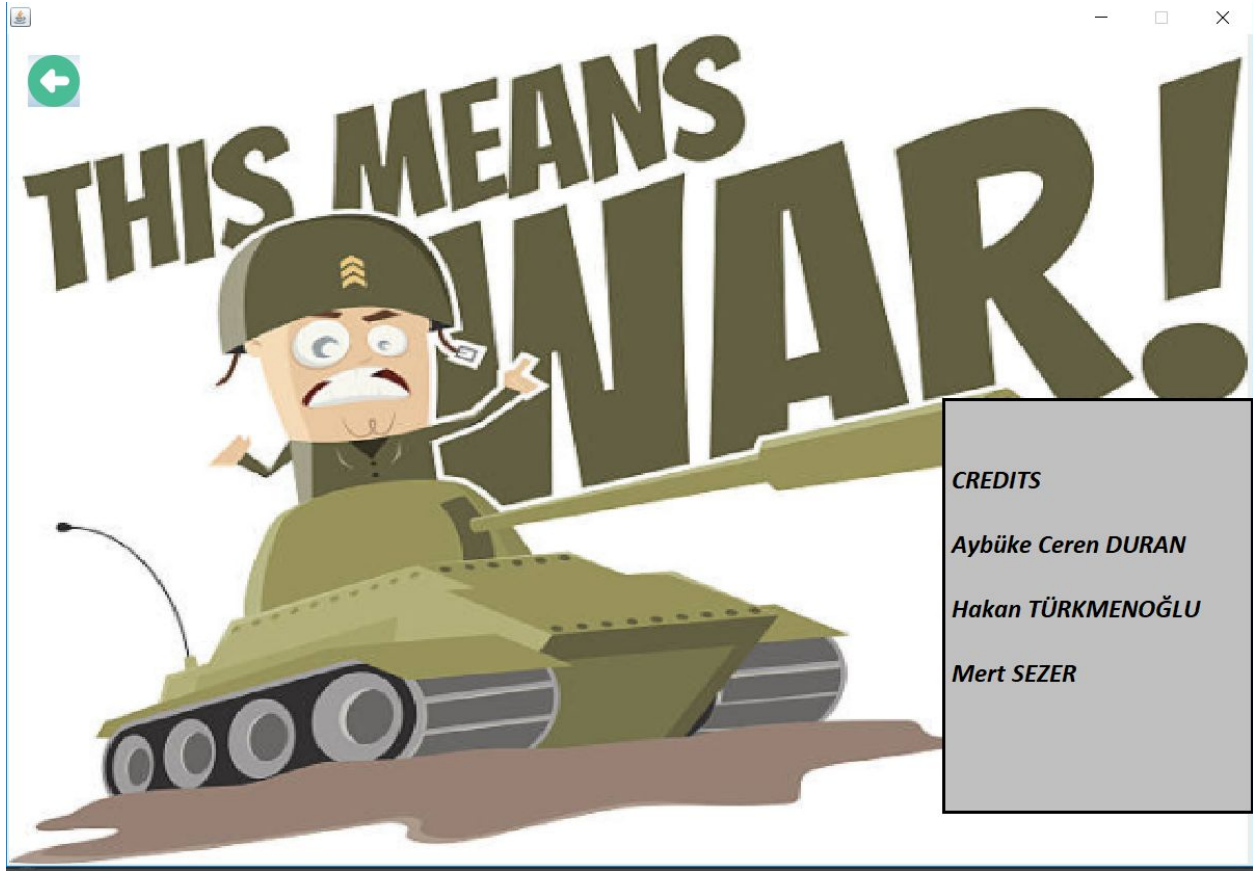Figure-02: Start Menu screen of TankZone Game

Figure-03: Credits screen of TankZone Game

Figure-03: Help screen of TankZone Game

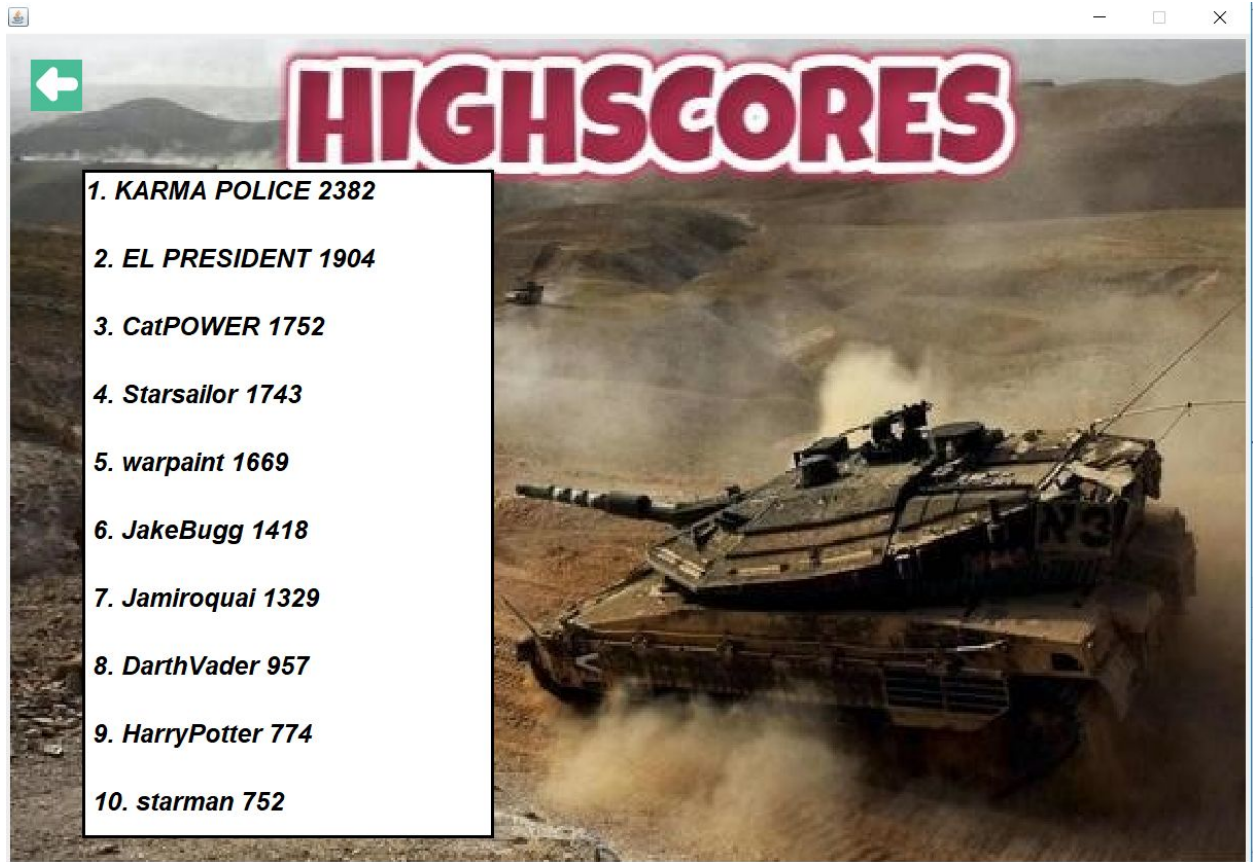Figure-04: Settings screen of TankZone Game
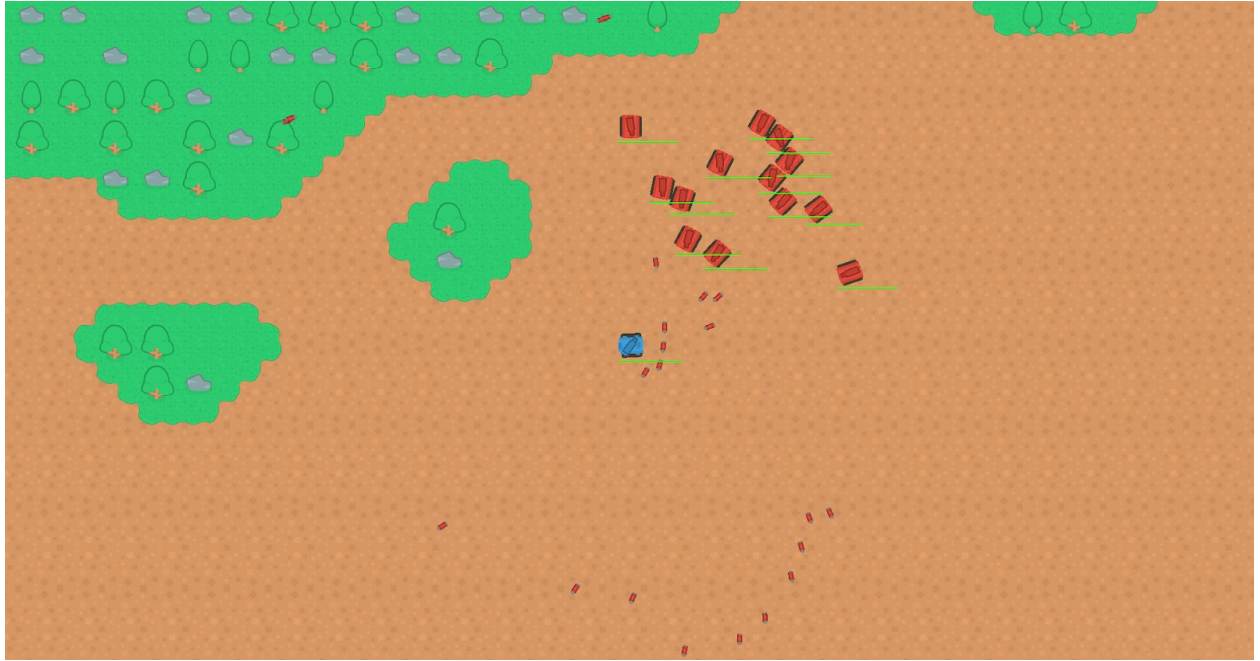
Figure-05: Help screen of TankZone Game

Figure-06: Temperate climate of TankZone Game

Figure-07: Winter climate of TankZone Game

Figure-08: Desert climate of TankZone Game

5. Experiences

In this project, we learned to get used to the real life projects, working as a team. We had an experience of how to work with other people that we do not before being assigned randomly for this project so we gained the experience of how real life projects proceed.