Bilkent University

Department of Computer Engineering

CS319 - OBJECT-ORIENTED SOFTWARE ENGINEERING

Analysis Report

Group 1E : Tank Zone

Alperen Koca

Aybüke Ceren Duran

Hakan Türkmenoğlu

Mert Sezer

Supervisor: Eray Tüzün

1. Introduction

‘Tank Zone’ is a basic Java-based and single-player game we will develop. ‘Tank Zone’ will be the improved version of the game ‘diep.io’ with extra features. We consider that ‘diep.io’ game is appropriate for applying the concepts of object-oriented programming principles into designing a system because ‘diep.io’ game includes different tank classes, obstacles such as coloured squares, triangles and pentagons, bullets on moving (See Figure 1). We also aim to have fun while improving our object-oriented programming skills so ‘Tank Zone’ game is more enjoyable version of ‘diep.io’ game.

In ‘diep.io’ game, the player embodies himself/herself as a tank in the middle of the environment of the game. The goal of the player in the game is to demolish enemies which are also tanks and destroy the obstacles around his/her tank.

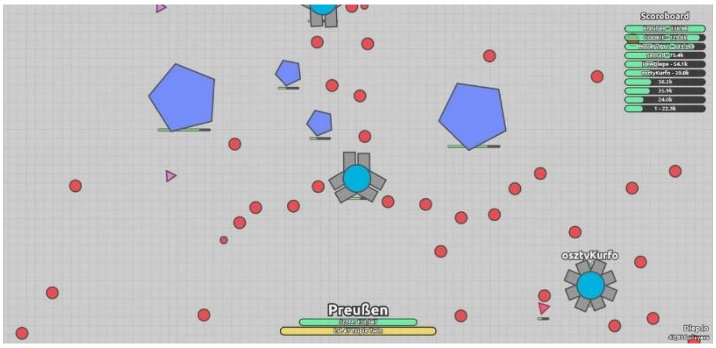


Figure 1: diep.io game

Detailed description of ‘diep.io’ game is following:

http://www.player.one/diepio-strategy-and-tips-tank-classes-upgrading-stats-and-progressing-through-agario-534450

The goal of ‘Tank Zone’ game is to attack and defeat enemies which are also tanks and upgrade the features of the tank as in ‘diep.io’ game. The player embodies himself/herself into tank. The differences of ‘Tank Zone’ game from ‘diep.io’ game is explained more detailed in the *overview* part.

‘Tank Zone’ game will be controlled by a mouse as in ‘diep.io’ game to destroy enemies. The movement of the user’s tank will be controlled by 4 keyboard keys.

2. Overview

‘Tank Zone’ game is an action and fast-paced game, and player needs to complete challenges by fighting with enemies. As having extra features, ‘Tank Zone’ provides opportunity/feature such as choosing game modes which are not in ‘diep.io’ game. By choosing game mode, gamer will also choose the type of having fun. One of the differences of ‘Tank Zone’ from ‘diep.io’ game is that ‘Tank Zone’ will have sound. We thought that a game with the sound represents more virtual reality compared to a game without sound. We thought that collision during the fight and climates such as rain will have sound.

2.1 Tanks

In 'diep.io' game, features of the tanks and how the tank is affected during the fight are reflected on the game screen during the war (See Figure-2). 'Tank Zone' will also include this feature because if the player is aware of how the tank is affected during the fight, game stream will change.

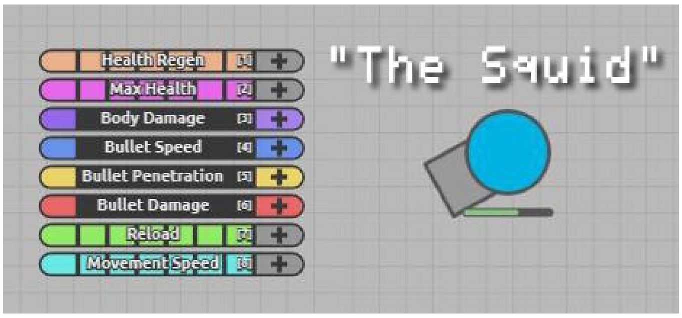


Figure 2: features of the tank and how the tank is affected during the fight

As the player passes level, player will have more efficient tanks in terms of max health, bullet speed, movement speed as in deep.io game (See Figure-3).

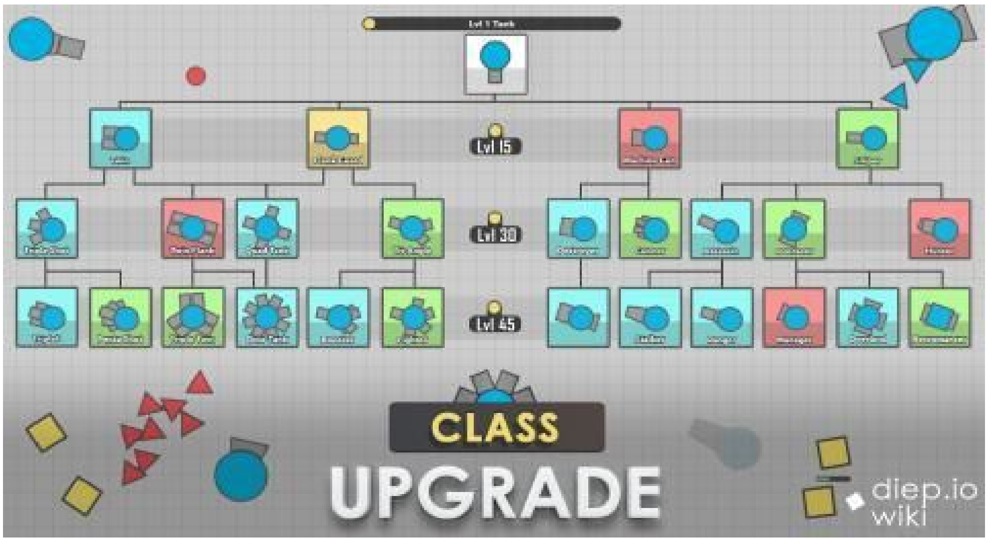


Figure 3: types of the tanks in 'diep.io' game

2.2 Game Modes

2.2.1 Capture The Flag mode

In our game, the player embodies as a tank and has his/her team. The player and his/her team will fight against 2 or more factions/teams. Therefore, 'Tank Zone' is also faction-type game. Compared to Free for All mode, Capture the Flag team mode increases the degree of competition and combat in our game. This game mode will include castles. Having castles in our game is one of the differences of our game from ‘diep.io’ game. Castles will represent checkpoint in our game which means that player’s tanks will get health pack there.

2.2.2 Free for All mode

Compared to Capture the Flag team mode, in Free for All mode everybody fights with each other and there is no faction/team. The game also has additional powerups specific to this game mode.

2.3 Climate mode

Climate mode provides different environments for users. Climate mode represents not only a decoration of realistic virtual world in ‘Tank Zone’ game, climate mode is crucial for increasing the degree of difficulty in ‘Tank Zone’ game. Climate mode includes following features:

- Temperate
- Winter
- Desert

Climate mode is important for levels in our game. As player passes level, climate conditions will affect the fight against enemies more because climate conditions will impact on the movement of the tank. For instance, in the environment with winterweather, tanks in the game will skid.

2.4 Difficulty mode

‘Tank Zone’ game includes easy, medium and hard difficulty types. For recent years, the choice of difficulty level varies from players to players. Some players like having challenge more so ‘Tank Zone’ has difficult game mode. Some players prefer to focus on learning game and its strategies, and some players like the feeling of victory in the easiest way and want to reach high score without having hardness. Therefore, ‘Tank Zone’ game includes easy game mode.

2.5 Power-ups

Power-ups are essential to make the fight between tanks more tough.

The power-ups are following:

- ***Teleport:*** Teleports a tank into a random location in the map.
- ***Blackhole:*** Acts as a gravitational field that pulls tanks inside, and eventually destroying them.
- ***Mind-control:*** In the Capture the Flag mode, changes the closest enemy tank’s
- alliance to the friendly faction.
- ***Time-freezer:*** Freezes enemy tanks
- ***Laser:*** Instead for firing bullets, the player will fire lasers that destroy enemy tanks rapidly.

As stated in the *climate mode* part, climate mode has big impact on fight. Therefore, the player will get power-ups by climate conditions.

2.6 Levelling Up

There will be 15 levels in ‘Tank Zone’ game which are arranged according to the difficulty order. As the player gets higher levels, tanks in the game will have more speed limit, health pack, and power-ups. Player will encounter unfavourable climate conditions as the level gets more difficult. Therefore, the fight between tanks will be more tough.

3. Requirements

3.1 Functional Requirements

- Player will be able to move the tank in any direction with using 4 keyboard keys.

- Player will be able to decide where to take aim at while shooting by using mouse.

- If the player’s tank is destroyed or fuel and weapon does not remain for this tank, the system automatically will end the game.

- Player must choose one option for the game (free for all or capture the flag).

- In free for all game, the aim is to destroy all opponents’ tanks.

- In capture the flag game, every team has 4 fortresses and the aim is to destroy the opponent team’s or teams’(if there is more than 1 opponent team) fortresses while protecting your team’s fortresses from damage; which leads to demolition. If your team’s all 4 fortresses are demolished, your team loses and the system ends the game.

- Player will be able to give damage to the opponents’ tanks or to the opponents’ fortresses (in capture the flag game) if at least 1 weapon remains in player’s tank and if his/her shot hits to these tanks or fortresses.

- One factor that determines the amount of damage the player can give to other tanks or fortresses is the type of weapon this player uses that depends on power-ups.

- There are surprise power-ups and traps in the gameplay; the player’s tank takes damage if he/she isn’t careful about these traps. Also if the player wins a power-up, the system is asking the player whether he/she wants to use this power-up. If the player rejects that power-up, he/she is only be able to use the previous power-up or the first, single, small cannon ball (if the user did not win power-up before.)

- There are also 15 levels in the game; which depend on player’s skills. If the player is successful enough, his/her level automatically increases by 1 and he/she will be able to increase his/her health regen or max health or body damage or bullet speed or bullet penetration or bullet damage or reload or movement speed by 1 in this case.

- We, developers choose the paths, power-ups, traps and enemies randomly for each map; so players won’t be able to do any arrangement for maps and background colors of this maps for the gameplay.

- There are enough maps for the gameplay and the player will select the climate and choose the difficulty level; in other words, player can take initiative in deciding the probability of encountering with different types of power-ups, traps and enemies in different types of maps.

- Player will be able to pause the game and then decide quiting the game or going to In Game Menu.

- Player will be able to change the sound options(On-Off) from OPTIONS in the In Game Menu and then continue the game from where he/she left off.

- Player will be able to change the controller settings from SETTINGS in the Main Menu. Controller settings include tank's moving directions and shooting directions.

- Player will be able to reach to the help document through the "HELP" button in the Main Menu, if he/she need to understand better about the gameplay; purpose of game, player controls, power-ups, traps, etc⋯

- Player will be able to see top 50 scores all around the world by pressing "HIGHSCORES" button in the "Main Menu". Also system changes and updates its' HighScores list, if any player takes a score higher than the lowest score in this HighScores list. This will encourage the players to work harder to be in that list.

- Player will be able to see his/her own highest score by pressing "ACHIEVEMENTS" button in the "Main Menu". We want player to see which score he/she achieved in previous games and this will encourage the player to work harder to achieve more.

- Player will be able to find the developers and contact with them by accessing to their informations by pressing "CREDITS" button in the "Main Menu". "CREDITS" will include the developers' names and their contact informations like Facebook, Twitter, Gmail or WhatsApp.

3.2 Pseudo-Functional Requirements

- This game will be implemented in Java.

3.3 Non-Functional Requirements

Modifiability: Before the implementation part of this project, we give priority to determine what the game will be and what it will look like. Because of that, the tasks that our system should perform change easily with every new idea. Therefore our project must be modifiable and we should be able to make changes in the software readily for our future systems.

Maintability: Since we want our project to be modifiable and we want to be able to make changes quickly and cost-effectively to functionality of our system during this semester, our project should be maintainable for the long-term production of our product. To provide maintainability, we are doing use of standard parts to make components, services, functions and interfaces of our system flexible. Thus our product will allow safe, quick, easy replacement of its component parts.

Reliability: Since we are to develop an online game, we are aware of the consequences of software failure. Since our system will be complex, it may not theoretically be possible to test every combination of states and inputs for referring to software reliability. Because of that, we will specify how to protect from failure, a strategy for error detection and a strategy for correction.

Security: Different users can not have same username or password. Also all passwords should be at least 7 characters length and they must include at least 1 uppercase letter, 1 lowercase letter and 1 number. If multiple users are using the same device to play the TankZone game, they have to enter their own username and password for entering their own accounts. Also for information hiding, users are not allowed to see detailed informations about other users' accounts like these users' highest scores in ACHIEVEMENTS part.

Usability: To make the game delightful and unputdownable for the user, we are putting some good and bad surprises in front of the user. These surprises like diiferent type of weapons, power-ups, traps and enemies are a bit confusing; but the main purpose of the game is easy to understand. Also the system is easy to learn and HELP part of the project will explain how to achieve common tasks in the game like achieving a different level and finishing the game successfully.

4. System Models

4.1 Use Case Model

Below is the use case diagram that describes the main flow of the game.

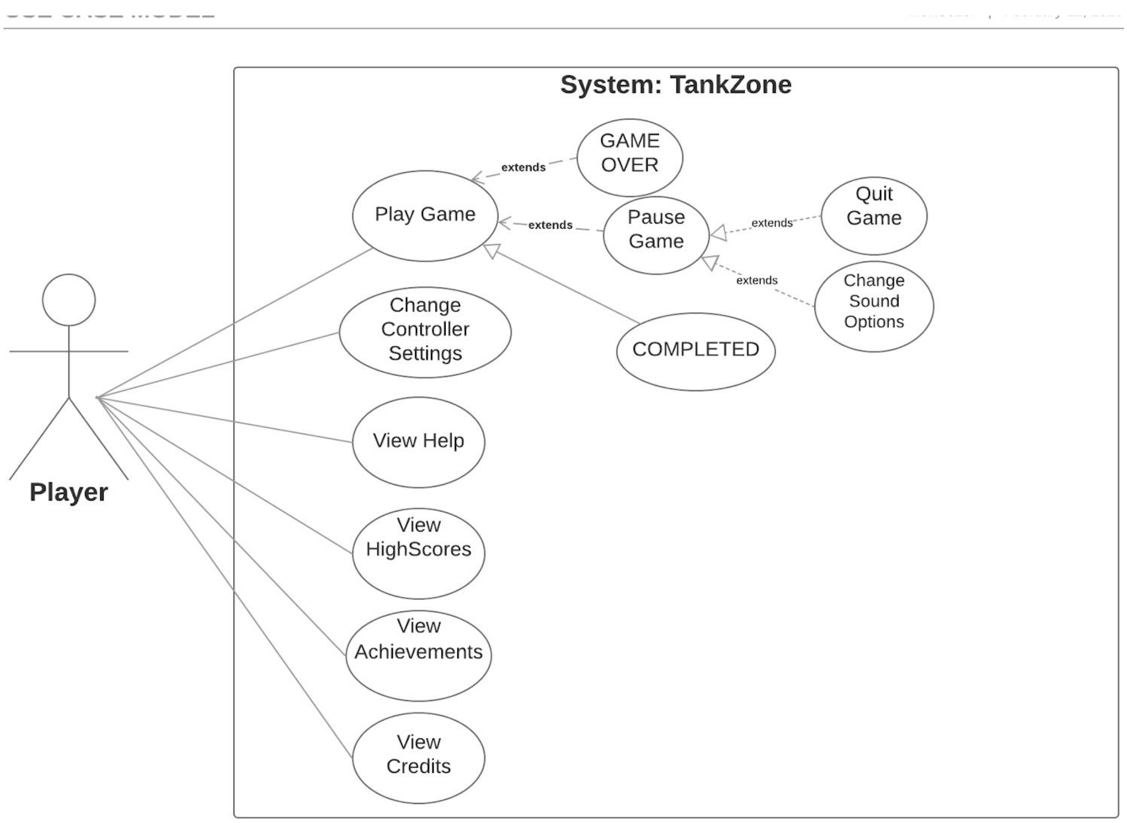


Figure 4: The use case model of TankZone

4.1.1 Play Game

Use Case Name: Play Game

Participating Actors: Player

Entry conditions: Player has already opened the game and is on "Main Menu" of the game.

Exit conditions:

-Player and his/her team clears off opponent team's fortresses if player is in capture the flag game, player defeats all his/her opponents' tanks if player is in free for all game OR

- Player's tank fails and system prints "GAME OVER" OR

- Player pauses the game and then exit game.

Main Flow of Events:

1.Player presses "START" button to start the game.

2. The system asks player to choose 1 of the 2 options(free for all or capture the flag).

3. Player chooses 1 of these 2 options.

4. System asks player to choose the climate(continental, dry, tropical, dessert, polar).

5. Player chooses 1 climate for gameplay.

6. System asks player to choose the difficulty level(easy, medium, hard).

7. Player chooses 1 difficulty level.

8. The system constructs the game and the game starts.

9. Player plays until all enemies(other tanks)are destroyed if game is free for all or player plays until the opponent team's fortresses are destroyed if game is capture the flag.

10. If player achieves a score higher than the lowest score in the HIGHSCORES list; which contains top 50 scores, system updates HIGHSCORES list accordingly.

11. If player achieves a score higher than his/her previous highest score, the highest score in the ACHIEVEMENTS list will be updated by the system accordingly.

Alternative Flow of Events:

Part 1, 2, 3, 4, 5, 6, 7, 8 is same as main flow of events.

9. People can't complete the game; because player's tank is destroyed or there is no remaining fuel and weapon.

10. System displays "GAME OVER".

11. If player has achieved a score higher than the lowest score in the HIGHSCORES list; which contains top 50 scores, system updates HIGHSCORES list accordingly.

12. If player has achieved a score higher than his/her previous highest score, the highest score in the ACHIEVEMENTS list will be updated by the system accordingly.

The Other Alternative Event Flow:

Part 1, 2, 3, 4, 5, 6, 7, 8 is same as main flow of events.

9. Player presses “||” button to pause the game.

10. System asks whether the player wants to quit game.

11. Player selects “YES” and goes to “Main Menu”.

12. If player has achieved a score higher than the lowest score in the HIGHSCORES list; which contains top 50 scores, system updates HIGHSCORES list accordingly.

13. If player has achieved a score higher than his/her previous highest score, the highest score in the ACHIEVEMENTS list will be updated by the system accordingly.

4.1.2 Change Sound Options

Use Case Name: Change Sound Options

Participating Actors: Player

Entry conditions: Player has already started a new game.

Exit conditions: Player presses “BACK” button to return to “In Game Menu”again.

Main Flow of Events:

1. Player presses “||” button to pause the game.

2. System asks whether the player wants to quit game.

3. Player selects “NO”.

4. System opens “In Game Menu”.

5.Player selects “OPTIONS” from “In Game Menu”.

6. Player chooses one option(Sound On or Sound Off).

7. System decides using or not using sound for the remaining gameplay according to player’s choice.

8. Player returns to “In Game Menu”.

4.1.3 Change Controller Settings

Use Case Name: Change Controller Settings

Participating Actors: Player

Entry conditions: Player is on “Main Menu” of the game.

Exit conditions: Player presses “BACK” button to return to main menuagain.

Main Flow of Events:

1.Player selects “SETTINGS” from the “Main Menu”.

2. Player selects “Controller Settings” from “SETTINGS” menu.

3. Player adjusts the settings according to his/her comfort.

4. System saves and updates corresponding changes for future gameplays.

5. Player returns to “Main Menu”.

Alternative Flow of Events:

1.Player selects “SETTINGS” from the “Main Menu”.

2. Player selects “Controller Settings” from “SETTINGS” menu.

3. Player selects “Use Default Settings” instead of adjusting them.

4. System saves and updates its’ own adjusted settings; which are adjusted for the first run for future gameplays.

5. Player returns to “Main Menu”.

4.1.4 View Help

Use Case Name: View Help

Participating Actors: Player

Entry conditions: Player is on “Main Menu” of the game.

Exit conditions: Player presses “BACK” button to return to main menuagain.

Main Flow of Events:

1.Player selects “HELP” from the “Main Menu”.

2. System displays a document explaining purpose of game, player controls, power-ups, traps and thegameplay screen showing the amount of remaining power-ups, other weapons, fuel.

3. Player reads that document to understand the gameplay and returns to “Main Menu”.

4.1.5 View HighScores

Use Case Name: View HighScores

Participating Actors: Player

Entry conditions: Player is on “Main Menu” of the game.

Exit conditions: Player presses “BACK” button to return to main menu again.

Main Flow of Events:

1.Player selects “HIGHSCORES” from the “Main Menu”.

2. System displays top 50 scores and corresponding player names from all world.

3. After seeing top 50 scores, player returns to “Main Menu”.

4.1.6 View Achievements

Use Case Name: View Achievements

Participating Actors: Player

Entry conditions: Player is on “Main Menu” of the game.

Exit conditions: Player presses “BACK” button to return to main menuagain.

Main Flow of Events:

1.Player selects “ACHIEVEMENTS” from the “Main Menu”.

2. System displays 1 player name; whose account is being used at that time and his/her highest score in the game.

3. After seeing his/her highest score, player returns to “Main Menu”.

4.1.7 View Credits

Use Case Name: View Credits

Participating Actors: Player

Entry conditions: Player is on “Main Menu” of the game.

Exit conditions: Player presses “BACK” button to return to main menuagain.

Main Flow of Events:

1.Player selects “CREDITS” from the “Main Menu”.

2. System displays names and contact informations (Facebook, Twitter, Gmail or WhatsApp) of the developers of the game.

3. After seeing these informations, player returns to “Main Menu”.

4.2 Dynamic Model

4.2.1 Sequence Diagrams

4.2.1.1 Starting a game

Scenario: Player Mert starts the game and is welcomed with the MainMenu screen. The menu asks Mert’s preference for nickname, game mode, climate and faction. After that Mert clicks “Start Game” button. This causes the game to generate a random procedural map from scratch using a Pseudo Random Number Generator to obtain seed. Using this, Map creates game objects such as Player, Enemy, Obstacle, Powerup and Castle. These objects are placed into the game engine and then the game engine starts running the game loop.

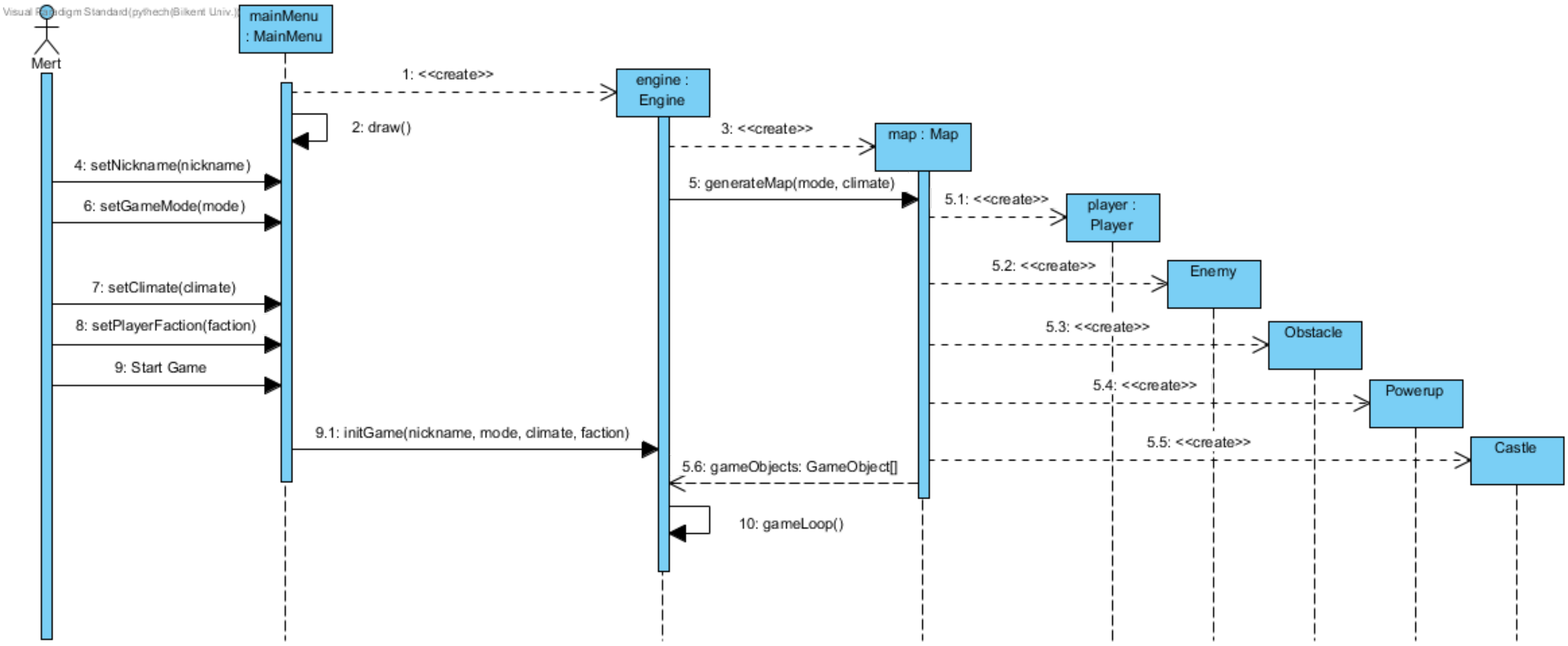


Figure 5: Starting Tank Zone

4.2.1.2 Upgrade Management

Scenario: Player Mert wants to check if he has enough points for an upgrade for his tank. For this purpose Mert clicks the Upgrade button, this triggers PowerupManager to get current upgrades and available points from the Player object. Using these information the player is shown which upgrades are available. Mert clicks “Quantum Accelerator” upgrade which increases velocity by certain amount. The PowerupManager checks if he has enough points to buy this item. After it’s confirmed that Mert indeed has enough points, said upgrade is installed to the Player which will cause the Player to have additional velocity.

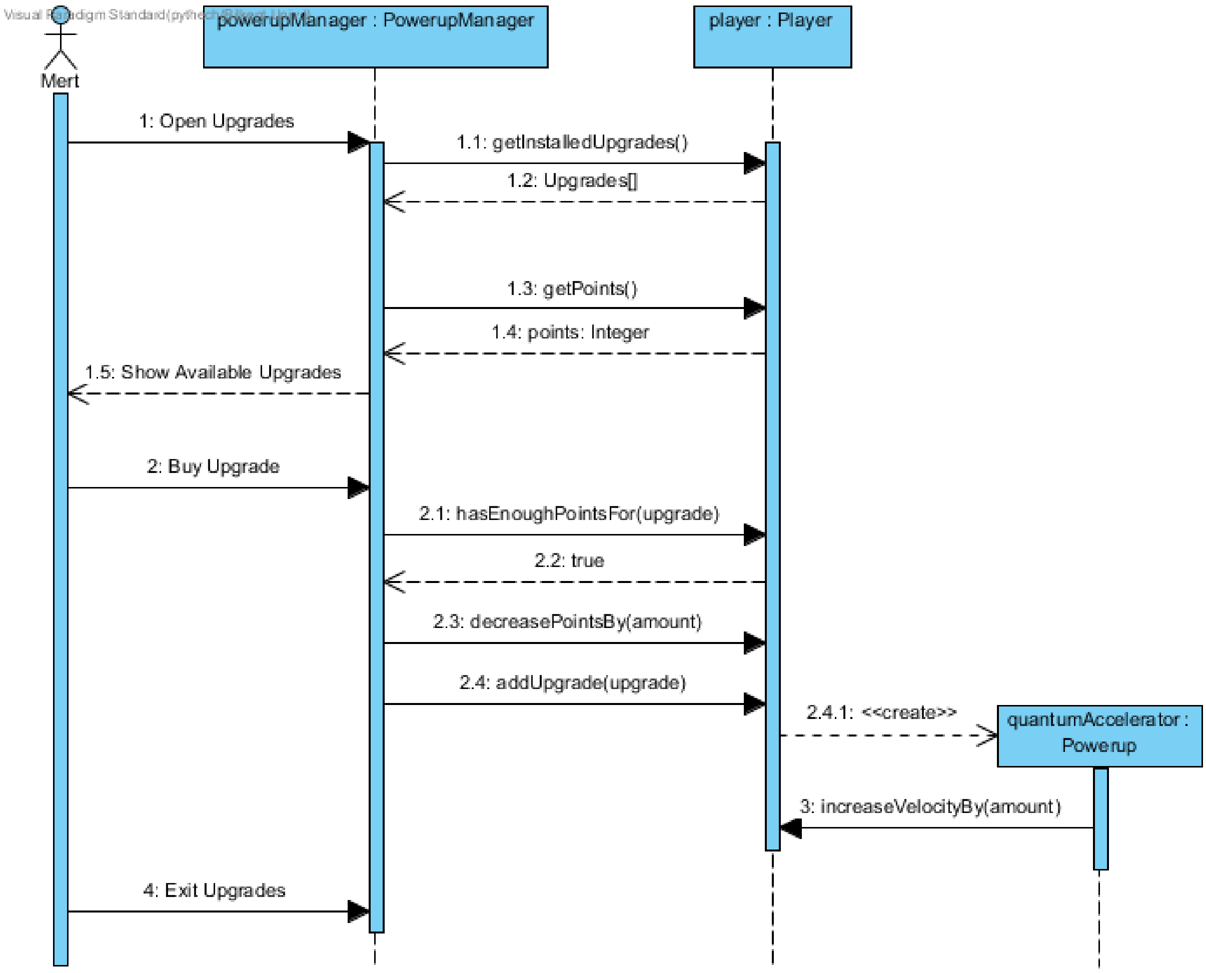

Visual Paradigm Standard(python(Bilkent Univ.))
Mert
powerupManager : PowerupManager
player : Player
1: Open Upgrades
1.1: getInstalledUpgrades()
1.2: Upgrades[]
1.3: getPoints()
1.4: points: Integer
1.5: Show Available Upgrades
2: Buy Upgrade
2.1: hasEnoughPointsFor(upgrade)
2.2: true
2.3: decreasePointsBy(amount)
2.4: addUpgrade(upgrade)
2.4.1: <<create>>
quantumAccelerator : Powerup
3: increaseVelocityBy(amount)
4: Exit Upgrades

Figure 6: Upgrade management

4.2.2 Activity Diagram

An activity diagrams that describes the main gameplay is the following:

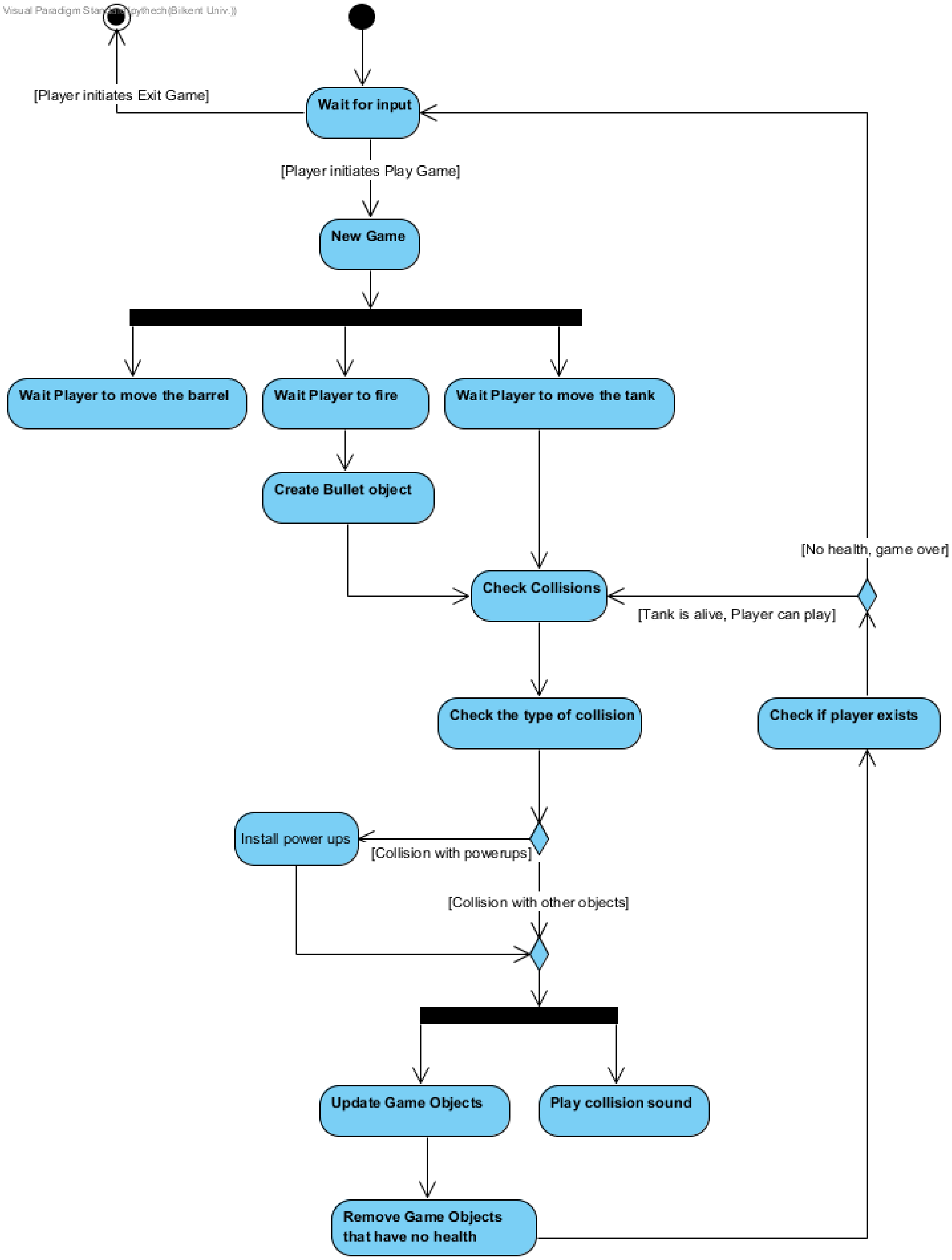

Visual Paradigm Standard(...ythech(Bilkent Univ.))
[Player initiates Exit Game]
Wait for input
[Player initiates Play Game]
New Game
Wait Player to move the barrel
Wait Player to fire
Wait Player to move the tank
Create Bullet object
[No health, game over]
Check Collisions
[Tank is alive, Player can play]
Check the type of collision
Check if player exists
Install power ups
[Collision with powerups]
[Collision with other objects]
Update Game Objects
Play collision sound
Remove Game Objects that have no health

Figure 7: Gameplay diagram

4.3 Object and Class Model

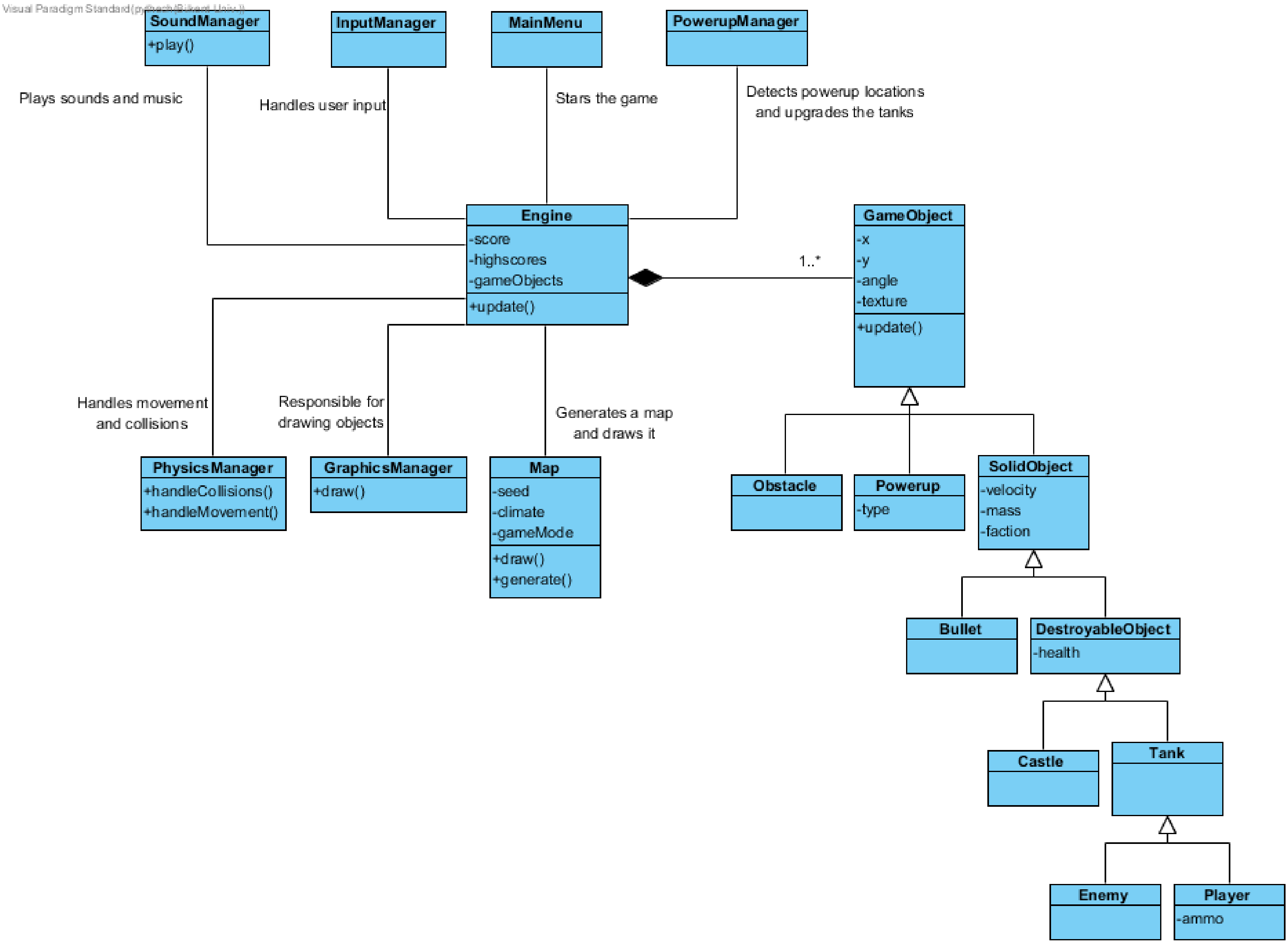


Figure 8: Class diagram of Tank Zone

Above is the class diagram of Tank Zone. Classes are explained as follows:

- **Engine** class is central to our business logic. It organizes the game by making use of manager classes, by itself it doesn’t do much.
- **GraphicsManager** draws each **GameObject** according to their properties. Note that **Map** is excluded here.
- **Map** generates a procedural map with a given seed, climate and game mode options. This will create the **GameObject**s in the game and insert them to the **Engine**. This class is also responsible for drawing the background tiles etc.
- **SoundManager** plays sounds and music.
- **InputManager** handles user input to make Player object is movable etc.

- **PowerupManager** handles issues related to the power up management like checking if the player is eligible for an upgrade etc.
- **PhysicsManager** handles cases where two **SolidObject**s collide and dictates the physical rules of the game such as movement.
- **GameObject** is a basic game object that has properties shared with all kinds of entities in the game.
- **SolidObject** is a game object which interacts with the physics of the game, these objects can collide with each other and move.
- **DestroyableObject** is a game object which has the health property added to **SolidObject**s.

4.4 User Interface and Navigational Paths and Screen Mockups

4.4.1 Navigational Path

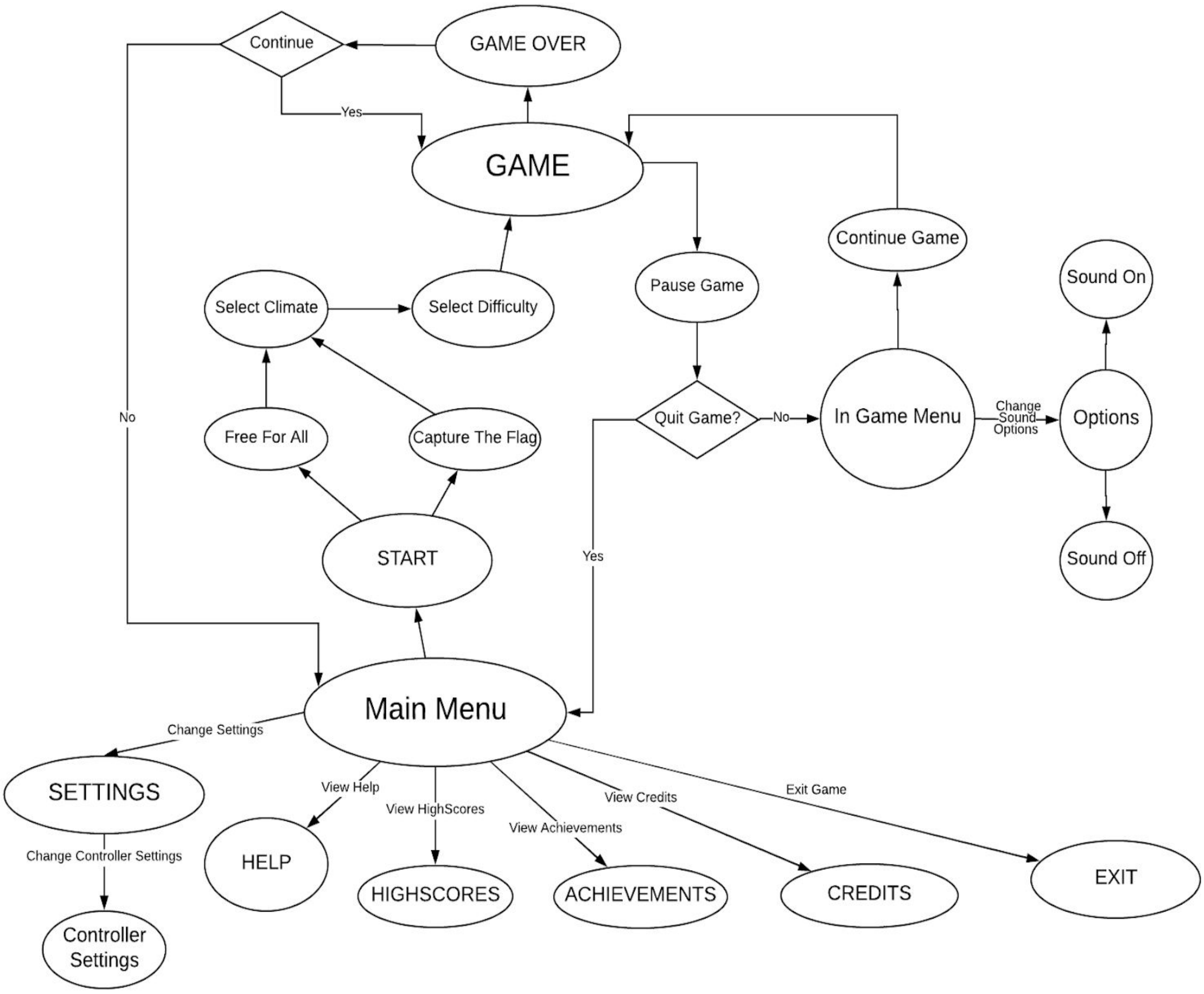


Figure 9: Navigational path for Tank Zone

4.4.2 Screen Mockups

4.4.2.1 Main Menu

This is the screen the player will encounter the first time game opens. It shows 7 options: New Game, Settings, High Scores, Achievements, Help, Credits and Exit.

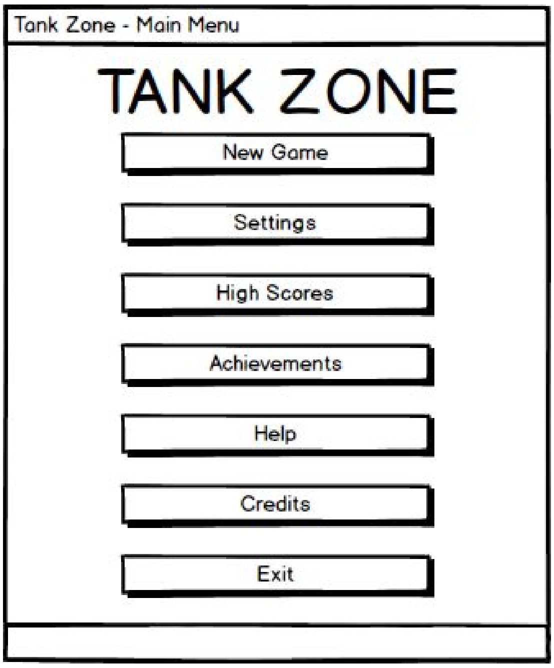
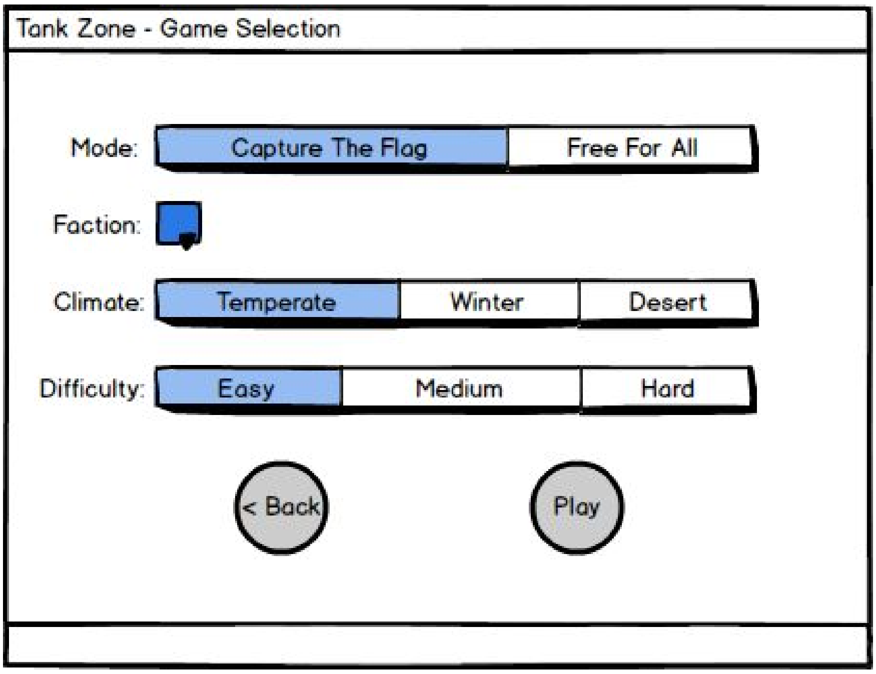


Figure 10: Main menu mockup

4.4.2.2 Game Selection

After clicking New Game button, the user will configure settings for the new match. These are: Game Mode, Faction (only if CTF mode is selected), Climate and Difficulty.



Mode: [✓ Capture The Flag] Free For All

Faction: 🟦

Climate: [✓ Temperate] Winter Desert

Difficulty: [✓ Easy] Medium Hard

Figure 11: Game selection mockup

4.4.2.3 Gameplay

Below illustrates how main game screen will look. In this example the game mode is Capture the Flag and the player Mert chooses blue faction. The tanks can fire cannons multiple times in a row. The player can move the tank barrel independently of the tank body. Player health, ammo and score are shown on the bottom-left of the screen. Player receives health and ammo packs when he/she gets to the checkpoint.

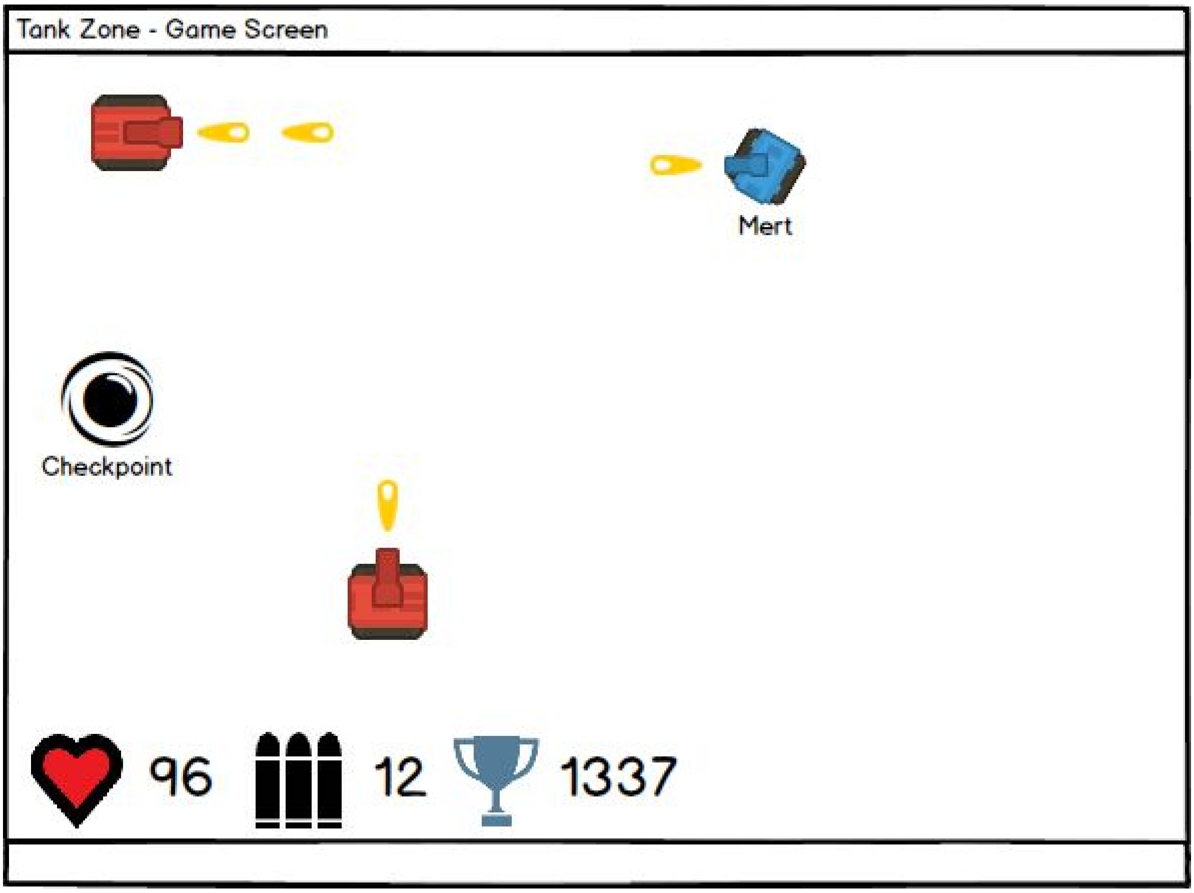


*Figure 12: Game screen mockup**

5. References

Diep.io game: http://diep.io/
* Top-down Tanks Redux assets by Kenney Vleugels (Kenney.nl)