CS 319 - Object-Oriented Software Engineering
Pre-draft Analysis Report

Battle City

Group 1-I

Mert Aytöre

Mert Özerdem

Muhammed Yusuf Satıcı

Mehmet Orçun Yalçın

TABLE OF CONTENTS

1. Introduction 3
2. Current System 3
3. Proposed System 4
    3.1. Overview 4
        3.1.1. Gameplay and Controls 4
        3.1.2. Obstacles and Objects 4
        3.1.3. Characters and NPCs 4
        3.1.4. Scoring Rules 5
        3.1.5. In-game Power-ups 5
        3.1.6. Time 5
        3.1.7. Sound 5
    3.2. Functional Requirements 5
        3.2.1. Play Game 5
        3.2.2. Settings 6
        3.2.3. Pause Game 6
        3.2.4. Help 6
        3.2.5. High Score 6
    3.3. Nonfunctional Requirements 7
        3.3.1. Usability 7
        3.3.2. Reliability 7
        3.3.3. Performance 7
        3.3.4. Supportability 7
    3.4. Pseudo Requirements 8
- Game will be implemented with Java. 8
    3.5. System Model 9
        3.5.1. Scenarios and Use Case Model 9
        3.5.2. Object Model 12
        3.5.3. Dynamic Models 14
            3.5.3.1. Sequence Diagrams 14
            3.5.3.2. Activity Diagrams 16
        3.5.4. User Interface 17
            3.5.4.1. Screen Mock-ups 17
                3.5.4.1.1. Main Menu Screen 17
                3.5.4.1.2. Play Game Screen 18
                3.5.4.1.3. Map & Difficulty Chooser Screen 19
                3.5.4.1.4. Highscores Screen 19
                3.5.4.1.5. Settings Screen 20
                3.5.4.1.6. Pause Screen 21
                3.5.4.1.7. Credits Screen 21
            3.5.4.2. Power-ups 22
            3.5.4.3. Tank Types 23
            3.5.4.4. Map Items 24
4. References 26

1. Introduction

Battle City is a basic 2-D strategy game that players aims to reach their opponent's base by destroying the bricks. It is a multiplayer game that two players can play against each other by using the same computer. The aim of developing this game is to provide a strategy game to the user that enables him to make rapid and correct decision while enjoying the time that they spend. Also, it is aimed that the game increase player's satisfaction by providing three different kind of maps with different difficulty levels and a variety of items that player can use during the game.

This report gives the basic explanation of the game by describing the features and components the game has. Also, it includes the functional and nonfunctional requirements that we need to satisfy for obtaining the system we propose. It explains the general design of the project from the client's point of view by providing use-case diagrams, scenarios, object models and dynamic models. In addition to the specified requirements, the report contains mock-ups of the user interface design of the game in order to represent the basic features and support the report with visuals.

2. Current System

The arcade game Battle City has several previous versions that were made for different kinds of platforms like Nintendo VS. System, Virtual Console Wii, Virtual Console 3DS and Virtual Console WiiU. Its initial version was developed by a Japanese arcade game company, Namco Limited (currently known as Bandai Namco Games) and they first released the game for Nintendo VS. System in 1985.

Since Battle City is a fairly aged game, in this project, Battle City's previous versions will be imitated and will be remade for desktop gameplay with additional power-ups, map creator and different scales of scoring in the game.

3. Proposed System

3.1. Overview

3.1.1. Gameplay and Controls

There are two control options for each player. First player controls the tank by w, a, s, d, e and space keys. Second player controls their tank by ↑, ↓, →, ←(arrow keys), enter and 0 keys, both players can pause the game via pause button on the screen. Space and enter keys are designed for firing weapon. e and 0 keys are designed for second weapon(dropping mine). w, a, s, d and ↑, ↓, →, ← keys will be used for navigation. The keyboard controls for each player can be changed before every game.

3.1.2. Obstacles and Objects

The game has two types of obstacles, ordinary brick and steel brick. These walls are destructible. Walls main purpose is to make game more interesting and more challenging.

Brick wall: Brick walls are the simplest wall type in the game. Brick walls can be breach by single shot from player's or AI's first weapon.

Steel wall: Steel walls are the most complex type of wall in the game. Steel walls are not destructible by bullet's fire unless it gets fire power up.

Player Base: Goal of this game is to reach the player base in order to win the game.

3.1.3. Characters and NPCs

The main actors of the game are the players and AI tanks and both actors can destroy ordinary bricks but player is also able to destroy steel bricks. At the starting stage of the both characters will have same damage and same speed. but player's character can bu upgraded via power ups in order to enhance the damage output. AI will not be able to destroy steel walls. There will be a single type of weapon for characters.

First weapon: basic type of weapon. This weapon's ammunition is infinite and is upgradable by power-ups. Player and AI can use first weapon. Does 1 damage per successful hit.

3.1.4.Scoring Rules

Player can gain score by destroying obstacles, AI opponents and also player can gain score by his or her remaining time. Player can get 100 point for destroying brick wall, 200 points for steel wall, 400 points for AI tank or Human opponent. For every remaining second player will get 20 points. Players' scores will display on the menu's high score window.

3.1.5.In-game Power-ups

Fire Power Up: This power up enhances the player's first weapon. Without this power up steel walls are indestructible, doubles the damage of the first weapon.
Time Power Up: Time power up adds 30 more seconds to the remaining time.
Health Power Up: Adds 3 more health to the player's health counter.

3.1.6.Time

The game will continue for a limited amount of time that is shown in the game screen. If none of the players can achieve to finish the game until the time is over the game will finish with a tie. If a player reaches his opponent's base or destroys his opponent's tank before the time is over, the remaining time will be added to the winning player's score. Every remaining second will increase the player's score by 20.

3.1.7.Sound

The sound effects will be used in the game after the players achieve to destroy or obtain some item. Also, the sounds will be used at the beginning and end of the game. The players will be able to close the sound effects from the settings menu or pause menu.

3.2. Functional Requirements

3.2.1.Play Game

This function initializes the game and allows player to direct the tank according to the control specified in "Gameplay and Control" section(3.1.1). The main purpose of game is

destroy ordinary and steel bricks by the helps of two types of weapons in order to reach the other player base and win the game.

After choosing to play the game, Users can choose their colours of their tanks and as noted above, there are different weapons and maps. These features make the game more compelling and enjoyable.

In a multiplayer game, if a player does not provide any input for a specific amount of time, the system will assume that player has left the game. In this situation, the AI shall take over that player's tank and continue to the game without any other change. After these changes, the game will continue as a single player game. After the AI takes over the player's tank, if the system receives an input from that player's keyboard, the system assume that the player has returned and AI will be terminated. After these changes, the game will continue as a multiplayer game.

3.2.2.Settings

Settings function is the basic part of the main menu screen where users can make adjustments on the game such as:

- Users are able to adjust the volume.
- Users are able to mute the game's sounds
- Users are able to change their tank control keys.

3.2.3.Pause Game

Function allows player to pause the game whenever s/he wants it.This option makes game more functional due to external interferences.

3.2.4.Help

This function allows new and veteran players to learn how to play the game. Function of each key damage output, power up types and weapons types of the attacks can be learned from here.

3.2.5.High Score

After every finished game this function will save them and display them from high to low. This function will not save more than 10 scores but it will replace scores as higher scores

earned by player. This feature makes game more enjoyable by increasing competition among players.

3.3. Nonfunctional Requirements

3.3.1. Usability

Our project aims to develop a game that can be played by anyone regardless of their knowledge about software systems. Therefore, it is crucial to provide simple and easy to use user interfaces to the user. Also, the game shall not contain sophisticated connections between screens that can confuse the user. Since Battle city has actions and inputs that are similar to any other strategy tank game, it is expected that user will be able to play the game without having to complete any tutorial. Therefore, even though our project has new features, the logic of the game should be understandable by the players without needing any assistance.

3.3.2. Reliability

The system shall not be affected by the invalid inputs of the user. Also, the game shall continue without any failure when the user chooses invalid actions. The user shall not be able to change any information related to the previous games such as highest scores of the players.

3.3.3. Performance

The game shall respond to the inputs of the player within one second in order to protect the flow of the game. The player shall not lose the access to system throughout the whole game. Even if one of the players leave the game, the computer shall take the players' place and game shall continue with a delay of no more than 5 seconds. In this case, some amount of delay is necessary to ensure that there is only one player left. The sounds of the game shall not affect the game performance.

3.3.4. Supportability

Since the variety of the items and maps is important to increase the choices that the user can make during the game, the items and maps in the game shall be modifiable and open to addition of new features. Also, the game should be platform independent in order to enable users to play the game from any software or hardware environment. The game has a default

keyboard setting. However, the player can modify the keyboard settings to change the key used in the game play. Therefore, the keyboard settings should be modifiable.

3.4. Pseudo Requirements

- Game will be implemented with Java.
- It will support only desktop devices.
- Swing framework will be used to handle graphics.
- Audio files that have .wav and .mp3 extensions will be used due to their compatibility of usage in Java.
- Maps in the game will be stored in a .txt file. Depending on the text files' content, the map will be created with whatever is embedded as a map inside the file. For map elements, unique identifiers (that are separated with spaces) will be used. B will stand for Ordinary Brick wall, S will stand for Steel Brick wall, P will stand for Players' Base, G will stand for Grass Bush and lastly to represent empty blocks in the map, "." (dots) will be used. So for a map that is 7x7 big, an example in .txt file will be in following:

  . . B P B . .
  . . B B B . .
  G G G G G G G
  S S . B B . .
  B G G G G G B
  . . B B B . .
  . . B P B . .

3.5. System Model

3.5.1. Scenarios and Use Case Model

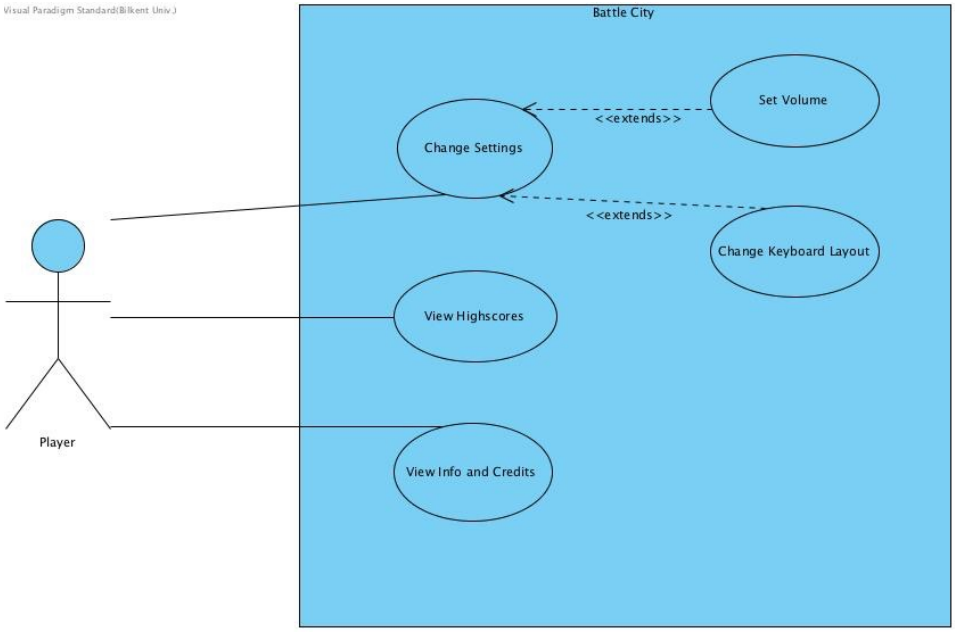


Figure 1: Use Case Diagram of Battle City - Without PlayGame cases

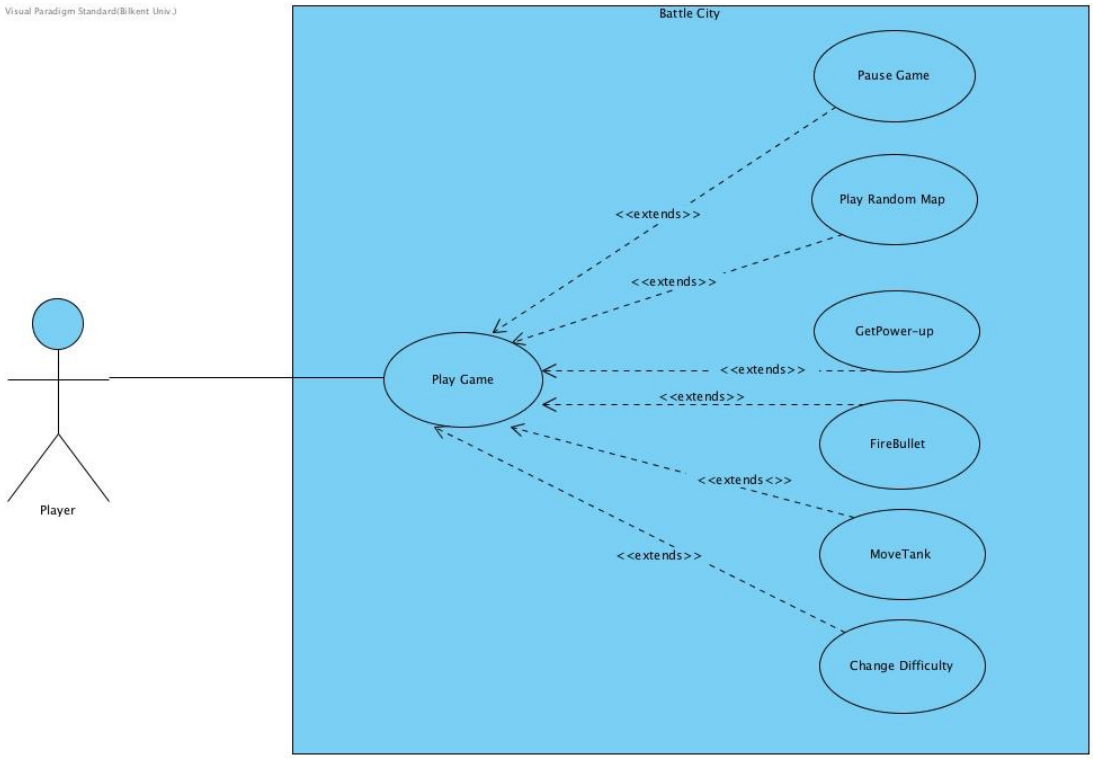


Figure 2: Use Case Diagram of Battle City - PlayGame cases

Scenario #1

Scenario name: playMultiPlayer

Participating actors: Muhittin:Player Nurettin:Player

Flow of events: Muhittin starts the multiplayer game by clicking the proper button on the main menu and choosing a predefined map. Both players see the game screen with the initial arrangements of the items. Muhittin moves his tank and fires bullets from the tank by using the keyboard. Both players sees the changes in the position of the tank and bricks at the game screen. Nurettin moves his tank and takes a power-up by using different keys from the keyboard. Both players see the effect of the power-up on the screen. Muhittin hides under a bush and fires bullet. Players cannot see the bullet until it leaves the bushes. Muhittin lands a mine. Players cannot see the mine until Nurettin passes over the mine. Players see the change in the health of Nurettin after Nurettin passes over the mine. Muhittin reaches the base of the Nurettin and the game finishes. Players see their scores on the screen.

Scenario #2

Scenario name: playSinglePlayer

Participating actors: Muhittin:Player Nurettin:Player

Flow of events: Muhittin starts the multiplayer game by clicking the proper button on the main menu and choosing a predefined map. Nurettin does not start playing the game. After a short amount of time, the AI starts to play the game instead of Nurettin and Muhittin sees the actions of the computer on the game screen. Player and computer play the game with the actions similar to the first scenario. The game finishes when one of the players reaches the opponent's base.

Scenario #3

Scenario name: changeSettings

Participating actors: Kamil:Player

Flow of events: Kamil clicks the proper button to open the change settings screen. Kamil turns the sounds off. Kamil changes the input keys of the keyboard from the keyboard settings. Kamil returns to the main screen. Kamil start a game similar to the previous scenarios. The game starts with new settings.

Scenario #4

Scenario name: viewHighScore

Participating actors: Kamil:Player

Flow of events: Kamil opens the high score screen by clicking right buttons. Kamil sees the score information on the screen along with the names of the players. Kamil returns to the main screen.

Scenario #5

Scenario name: playRandomMap

Participating actors: Muhittin:Player Nurettin:Player

Flow of events: Muhittin starts the multiplayer game by clicking the proper button on the main menu and choosing a random map. System generates a map randomly. The game starts with the newly generated map. They follow actions similar to the first scenario. System removes the randomly generated map at the end of the game.

3.5.2.Object Model

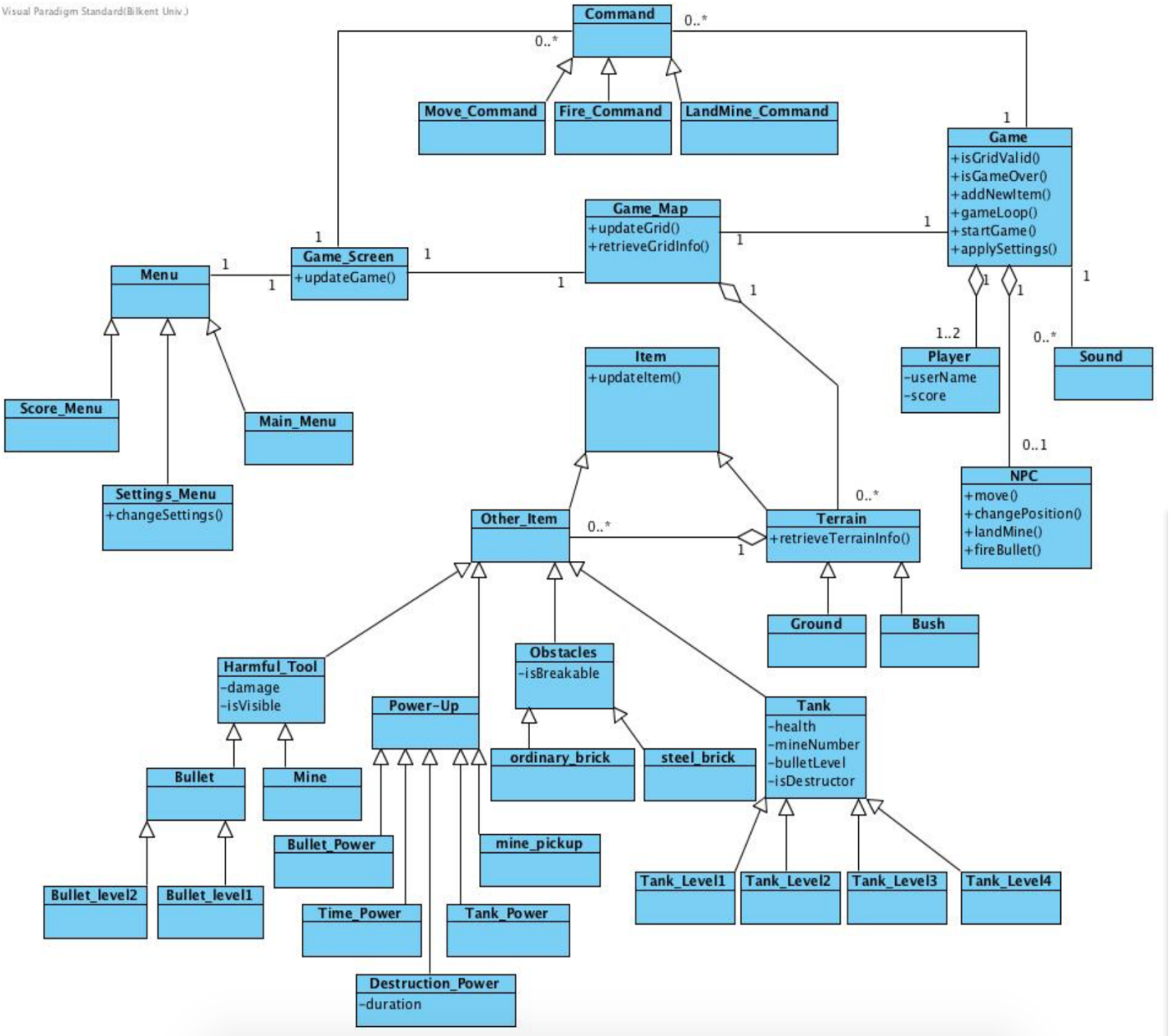


Figure 3: Object Model Diagram of Battle City

The object model of the Battle City game is shown in the diagram above. Class diagram of Battle City game includes 23 classes.

Game_Screen class is the main user interface class that represent the game on the screen. It is the boundary class that interacts with the user throughout the game play.

Command takes the events from the boundary class and checks the validity of the action. It is the controller class that controls the flow of the input in the game. It has subclasses for different kinds of command that user can give from the keyboard..

Game class checks the conditions for updating the game map and ending the game. It is the control class that controls the flow of the events and changes in the game. It also controls the changes that is made to the game map or the items inside the game.

Game_Map contains the information about the position of the items in the map and updates the grids and items of the map. It is the entity class that provides the positions of the items at any instant in the game.

Item entity class has two subclasses Other_Item and terrain. Terrain contains other items that could appear in the game and it is responsible for the update of those items. Other_Item has four subclasses which are obstacles, harmful_tool, power-up and tank. Each instances of these classes appear in one of the terrains in the game map. Tanks and harmful tools can change their position inside the game throughout the game whereas power-up and obstacles remain in the same terrain until they are removed from the map.

Sound class handles the usage of the audio files inside the game. It changes the audio when it receive an information from the game or settings screen.

NPC class handles the decisions of the tanks controlled by the computer during the single player game. It interacts with the Game class to check the validity of the decisions and transmit the new positions of the tanks, which AI controls, to the Game_Map. It also associates with the Game_Map to retrieve information about the items inside the map.

3.5.3.Dynamic Models

3.5.3.1.Sequence Diagrams

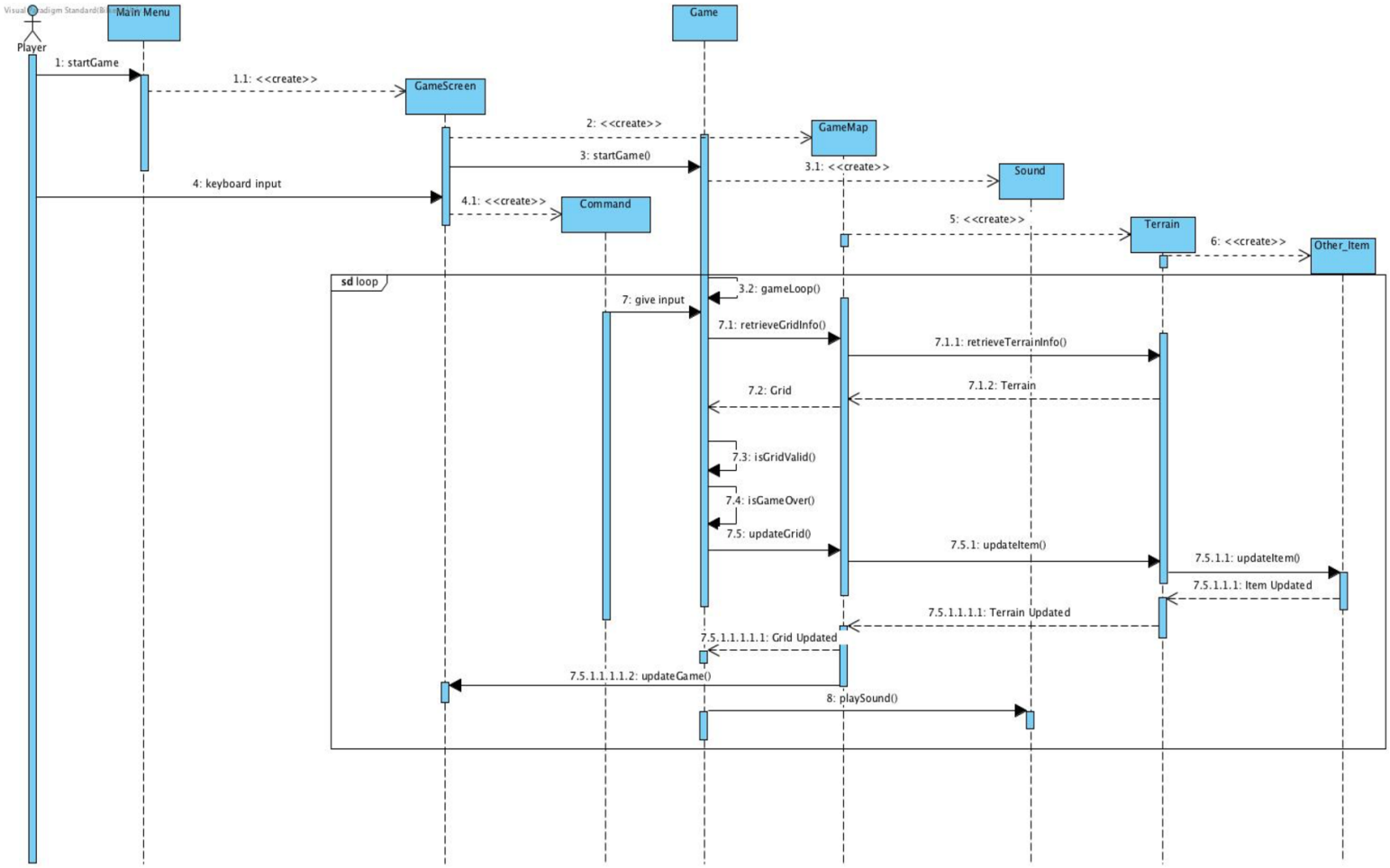


Figure 4: Sequence Diagram of Battle City for multiplayer game

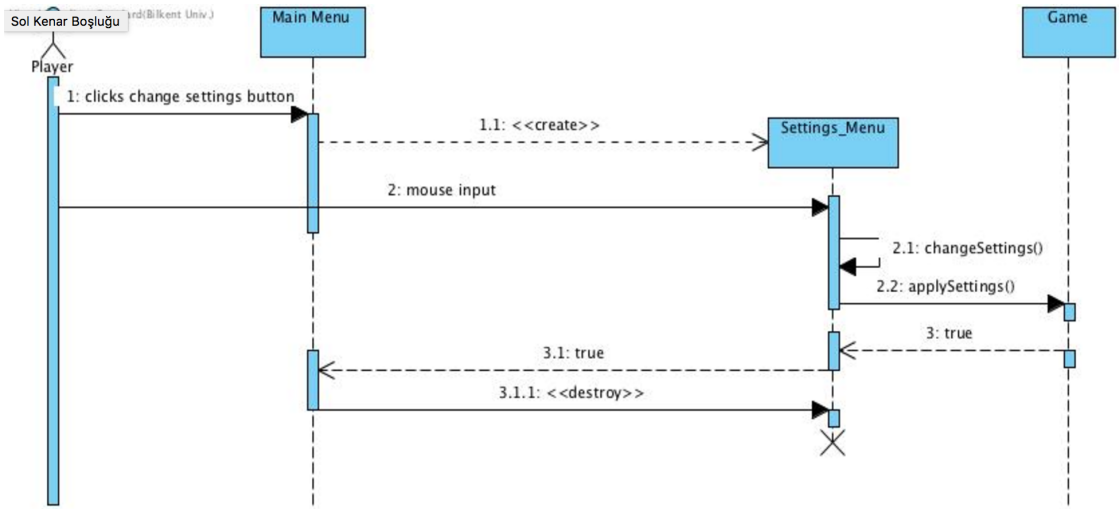


Figure 5: Sequence Diagram of Battle City for changing settings

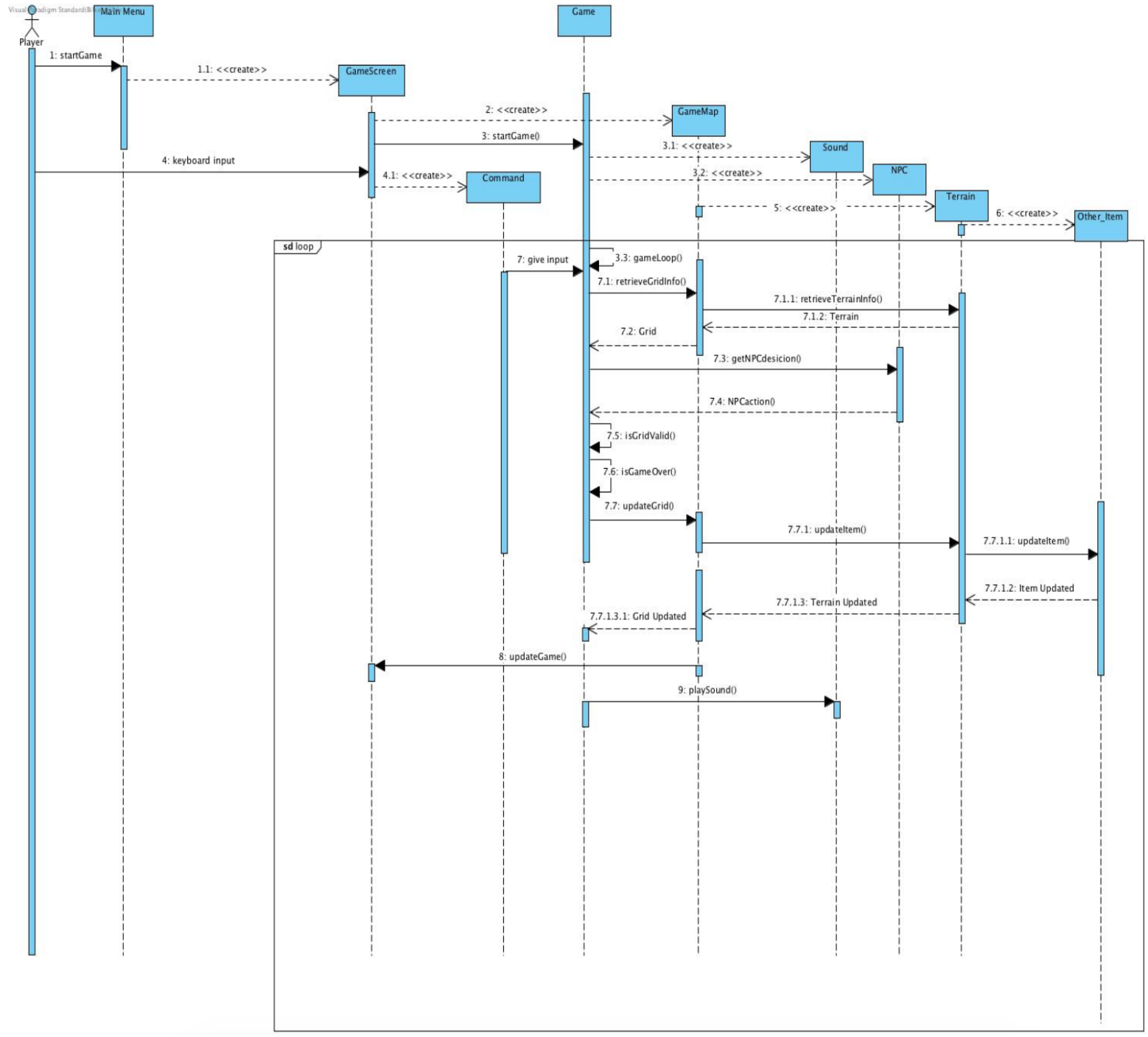


Figure 6: Sequence Diagram of Battle City for single player game

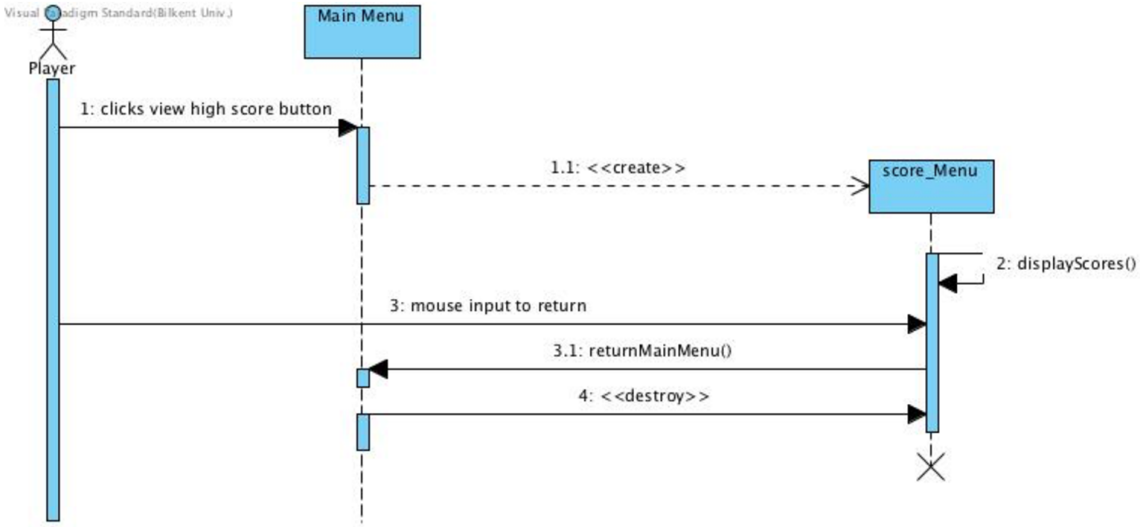


Figure 7: Sequence Diagram of Battle City for viewing scores

3.5.3.2.Activity Diagrams

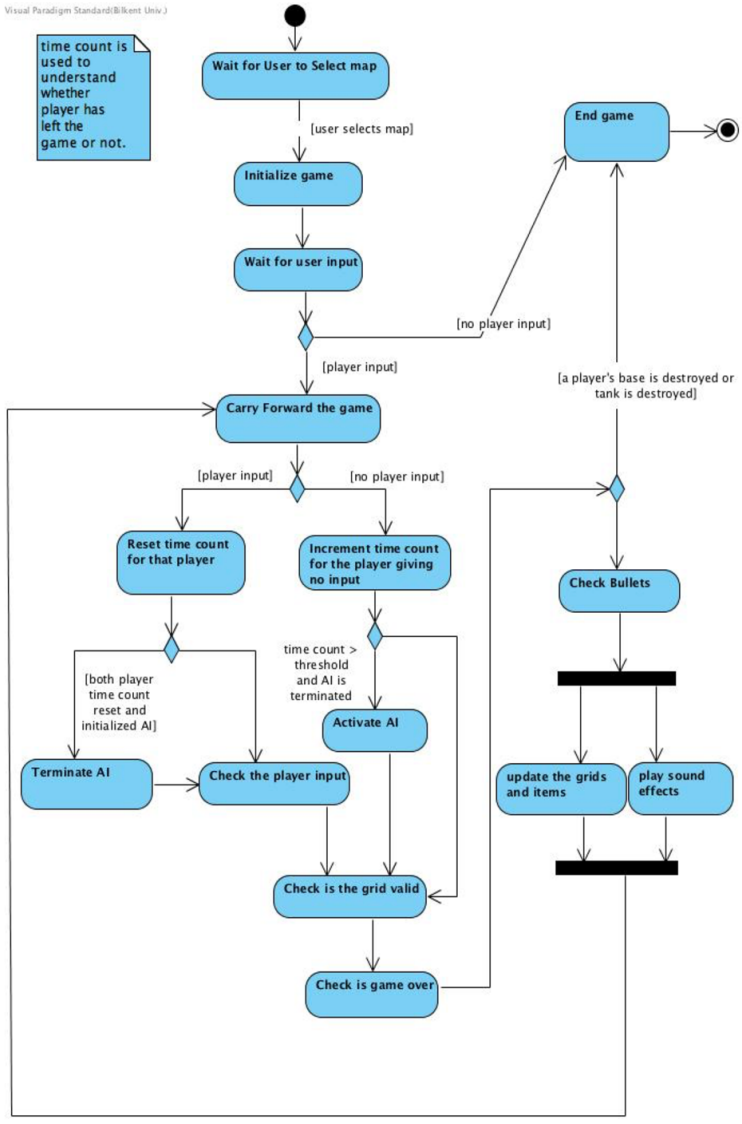


Figure 8: Activity Diagram of Battle City

The system waits for the user to select a predefined or random map at the initial state. When user selects the map, the system initializes the game by creating the map and the components inside the map. After the initialization, system waits for the user input in order to start the game. If there is no user input after some amount of time, the game finishes without starting. If there is valid user inputs from the keyboard, the system starts a game.

After the game starts, the system carries forward the game. In order to carry the game forward, the system decides the new positions of the moving objects and the new items that will be added to the map at that particular instant of the game. Also, the system decides the AI decisions of the single player game in the carry the game forward state. If there is no user input from the keyboard, the system increases the time count which will be used for deciding whether the user has left the game. If the time count for a particular user is more than a threshold time, the computer will decide that the user has left the game and the AI will be activated in order to continue with a single player game. If there is user input from the keyboard, the system will reset the time count for the player that gave the user input. If both player provides a user input and the AI was initialized before, the system will terminate the AI and continue with multiplayer game. Also, if there is a user input, the system will check the validity of the user input. The system checks the validity of the changes in the map and whether the game is over or not. If a player's base or tank is destroyed, the game finishes after showing the scores in the end game state. Otherwise the system checks the change in the position of currently moving bullets in the map. The system updates the grids and items inside the map and plays the proper sound effects for that instant of the game. After updating the game map, the system goes back to the carry forward state to continue with the game.

3.5.4.User Interface

3.5.4.1.Screen Mock-ups

3.5.4.1.1.Main Menu Screen

When the game is launched, the players will first see the main menu screen. From this screen, players will be able to start the game, change their game settings, access and see the best 10 scores and lastly they will be able to reach more information about the game.

Figure 9: Screenshot of Main Menu Screen

3.5.4.1.2.Play Game Screen

After choosing to play the game from the main menu screen, the players will see the game options chooser screen. This screen allows players to choose their colors of their tanks and choose the map that they're going to play in. The button by the "Difficulty:" label leads to a new screen which allows players to choose the map.

Figure 10: Screenshot of Play Game Screen

3.5.4.1.3.Map & Difficulty Chooser Screen

In this screen, players are able to pick the difficulty of their map. They can either choose the difficulty or play in a randomly generated map that is built by the computer. Players must click the “BACK” button on the left bottom corner of the screen in order to start their game.

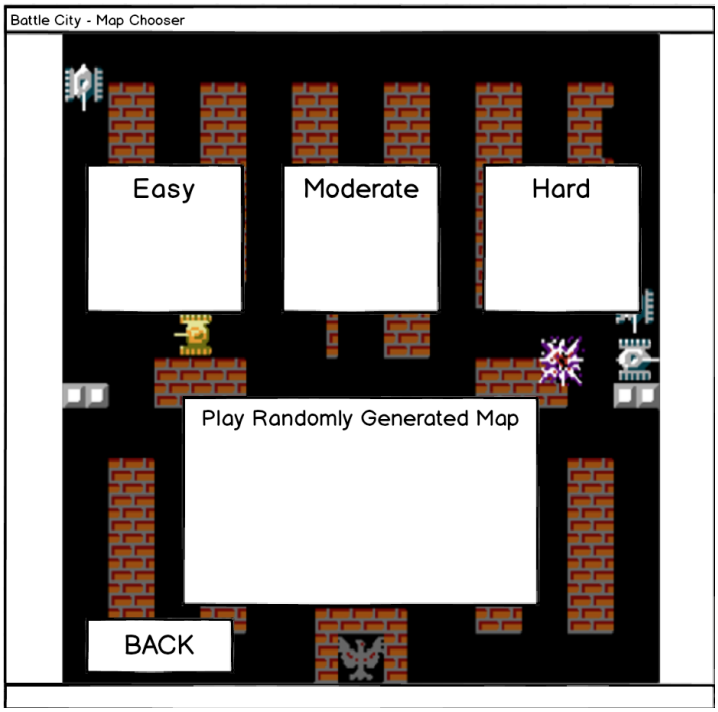


Figure 11: Screenshot of Difficulty Chooser Screen

3.5.4.1.4.Highscores Screen

Highscores screen is accessible via main menu screen and it contains the highest ten scores that were made in the game, since the time it has been installed. Players can click the “BACK” button on the left bottom corner of the screen to return back to the main menu screen.

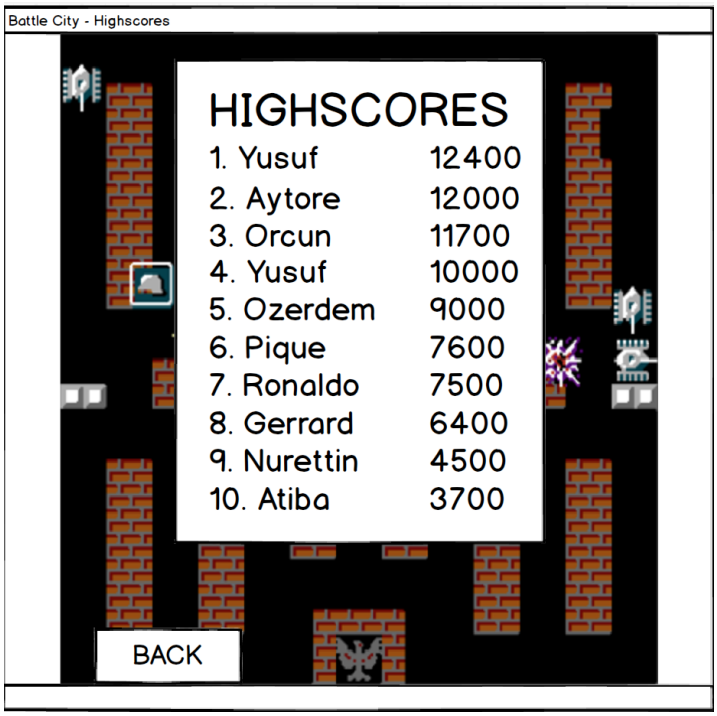


Figure 12: Screenshot of Highscores Screen

3.5.4.1.5.Settings Screen

In this screen, players are able to adjust the volume level of the game, and can easily mute the game's sounds by clicking the button that is located by the "Mute:" label. Players can also change their tank control keys by clicking the button by the "Set Keyboard Layout:" label. Players can click the "BACK" button on the left bottom corner of the screen to return back to the main menu screen.

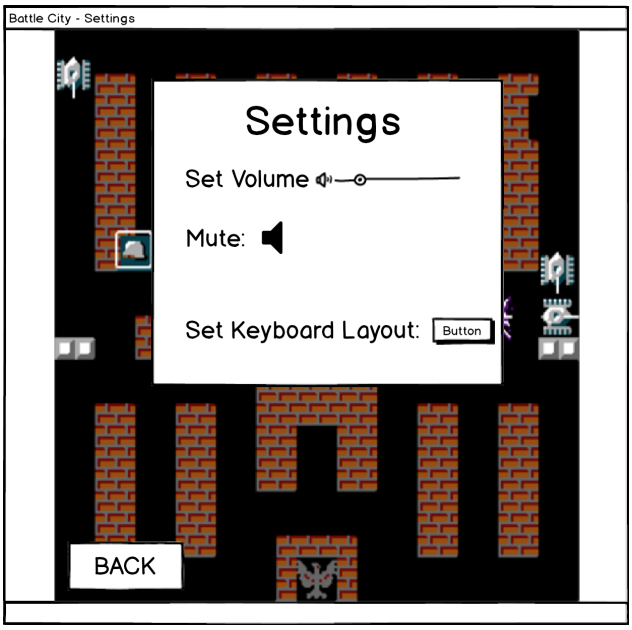


Figure 13: Screenshot of Settings Screen

3.5.4.1.6.Pause Screen

Players see the pause screen when they click the pause button that will be placed on the gameplay screen. Whole game flow stops when the game is paused and players see their scores on the screen, depending on the player count. Players will also see the remaining time until the game ends on the screen. Players must click the “Return to game” button on the left bottom corner of the screen in order to resume playing their game.

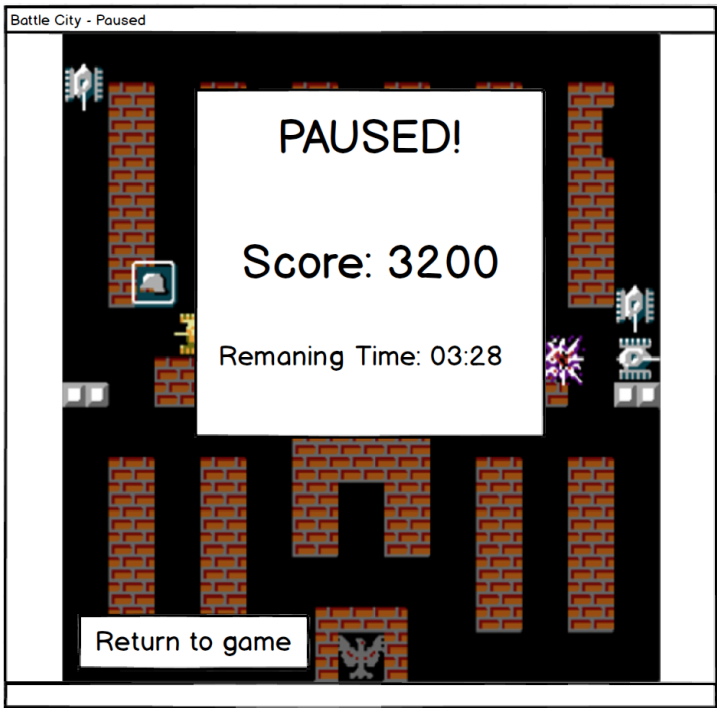


Figure 14: Screenshot of Pause Screen

3.5.4.1.7.Credits Screen

In credits screen, information on developers can be found. Also when the game is done, the date of release will also be available on the screen.

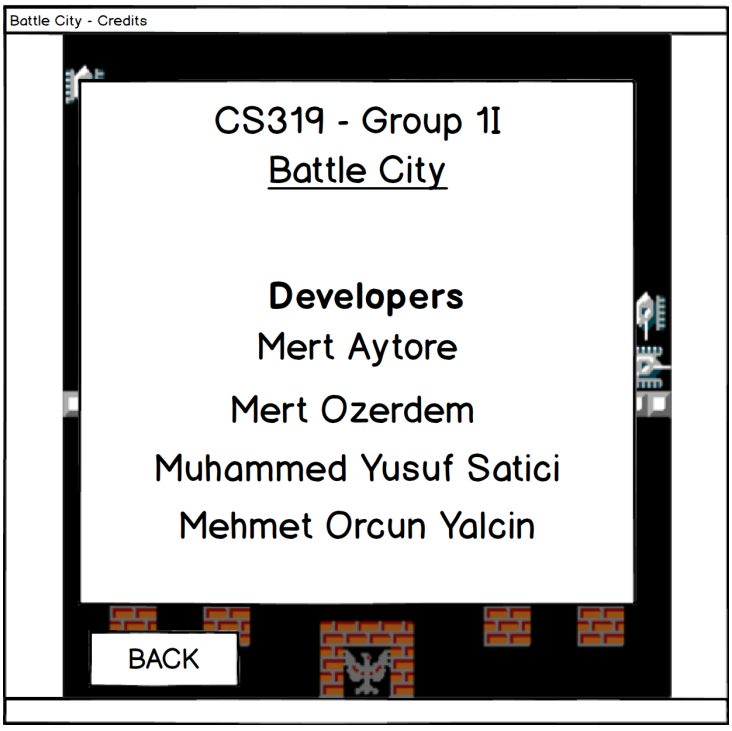


Figure 15: Screenshot of Credits Screen

3.5.4.2.Power-ups

The list of power-ups that are planned to be implemented in the game are listed below with their in-game images and explanations.

Brick Rider: Brick Rider enables the player's tank to ride over ordinary bricks to destroy and pass over them for a certain amount of time, 15 seconds.

Bullet Upgrade: Having this power-up picked up will upgrade the player's bullets and enable them to destroy steel bricks. This power-up can only be picked up once.

Mine Pick-up: Mine Pick-up gives the player a mine to be placed to anywhere of that player's choice. The mine is planned to damage tanks, destroy both ordinary and steel bricks.

Tank Upgrade: This power-up upgrades the player's or NPC's tanks' levels. With this upgrade, they become more durable and gain more health.

Extra Time: When this power-up is picked up, 30 seconds of extra time will be added to the current game's countdown timer. This power-up is crucial for players because if a player is playing against NPCs and the player runs out of time before destroying their base, then the player loses. If there are two players and time runs out, then the player with higher score wins the game.

 Brick Rider*

 Bullet Upgrade*

 Mine Pick-up*

 Tank Upgrade*

 Extra Time*

3.5.4.3.Tank Types

It is planned to have three different colors of tanks, for two different purposes. Yellow and Green tanks belong to human players and it is planned to have them controllable via keyboard inputs. Grey tanks belong to NPC players in the game. Those tanks' movement will not be controlled via keyboard, they will depend on game's logic that will be implemented. There are four levels of tanks and each of them will be implemented to upgrade when a Tank Upgrade power-up is picked up. When a tank power-ups, its health will increase and it will be more durable for incoming bullet attacks from NPCs/other player.

  Player Tank Level 1*  NPC Tank Level 1*

  Player Tank Level 2*  NPC Tank Level 2*



Player Tank Level 3*

NPC Tank Level 3*



Player Tank Level 4*

NPC Tank Level 4*

3.5.4.4.Map Items

The list of map items that are planned to be implemented in the game are listed below with their in-game images and their explanations.

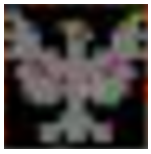
Players' Base: Castle-like map item that should be protected by players in order to win.

Ordinary Brick: One of the main obstacles in the game. Can be destroyed with a single non-upgraded/upgraded tank bullet hit.

Steel Brick: The obstacle in the game which cannot be damaged with a regular/non-upgraded tank bullet. It can be damaged and destroyed only with an upgraded tank bullet.

Bush: Bushes in the game can affect tanks' and their bullets' visibility when they are passing beneath them. Bushes cannot be damaged or destroyed. They serve the purpose of giving the players the advantage of concealing their location in the map.

Tank Bullet: Bullets are used in the game to damage map items, tanks and player bases. Tank bullets can only be upgraded once, with a Bullet Upgrade power-up.


Players' Base*


Ordinary Brick*


Steel Brick*

Bush*

Tank Bullet*

Images are taken from the game “Tank 1990”

4. References

https://www.spriters-resource.com/nes/batcity/sheet/60016/

https://en.wikipedia.org/wiki/Battle_City_(video_game)