CS 319 - Object-Oriented Software Engineering
Project Final Report

Battle City

Group 1-I
Mert Aytöre
Muhammed Yusuf Satıcı
Mehmet Orçun Yalçın

1. Introduction

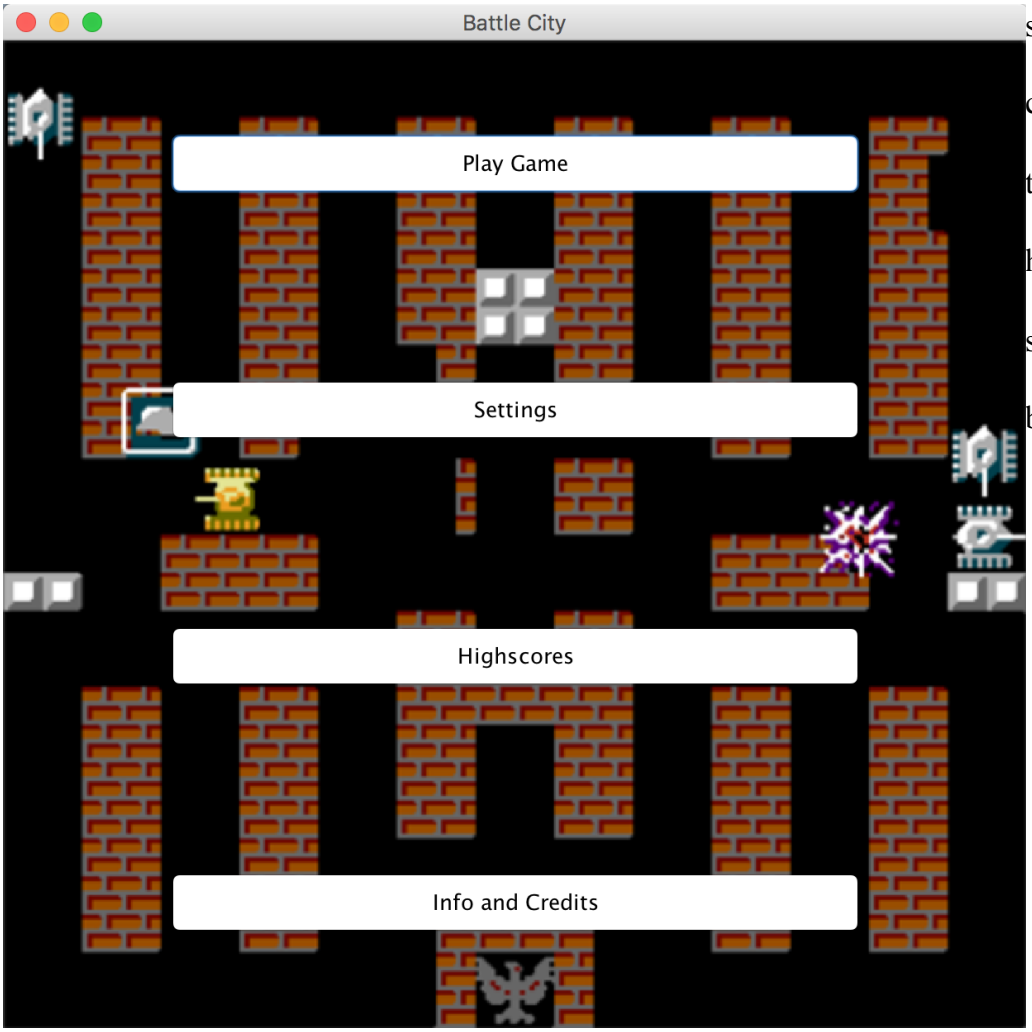
Battle City is a basic 2-D strategy game that players aim to destroy each other and get higher scores than each other. It is a multiplayer game that two players can play against each other by using the same computer. Also, if a player stops playing the game, NPC replaces that player and the game continues on. 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, with randomly generated maps and a variety of items that player can use during the game.

2. Application Setup

The user needs to have at least the Java standard edition runtime environment. The user can either use the source code or the executable jar file to play the game. To obtain the jar file, the user can download the game files (Play_BattleCity.zip) from the project repository by from GitHub (https://github.com/mertaytore/CS319_Group_1I) and simply run the jar file that is downloaded within (from terminal `java -jar Battle_City.jar`). Also, if the user downloads the source code, the user can run the code from a development environment or from a terminal to play the game.

3. The User's Manual

- When the application starts running, the main menu will be displayed on the screen. The main menu provides a variety of options to the user. If the user wants to play the game, he/she should click to the play game button. If the user wants to change the settings of the game for the keyboard control and sound adjustment, he/she should click to the settings button. If the user wants to see the current high scores, he/she



should click to the high scores button.

Finally, to learn about the developers of the game, user should click to the info and credits button.

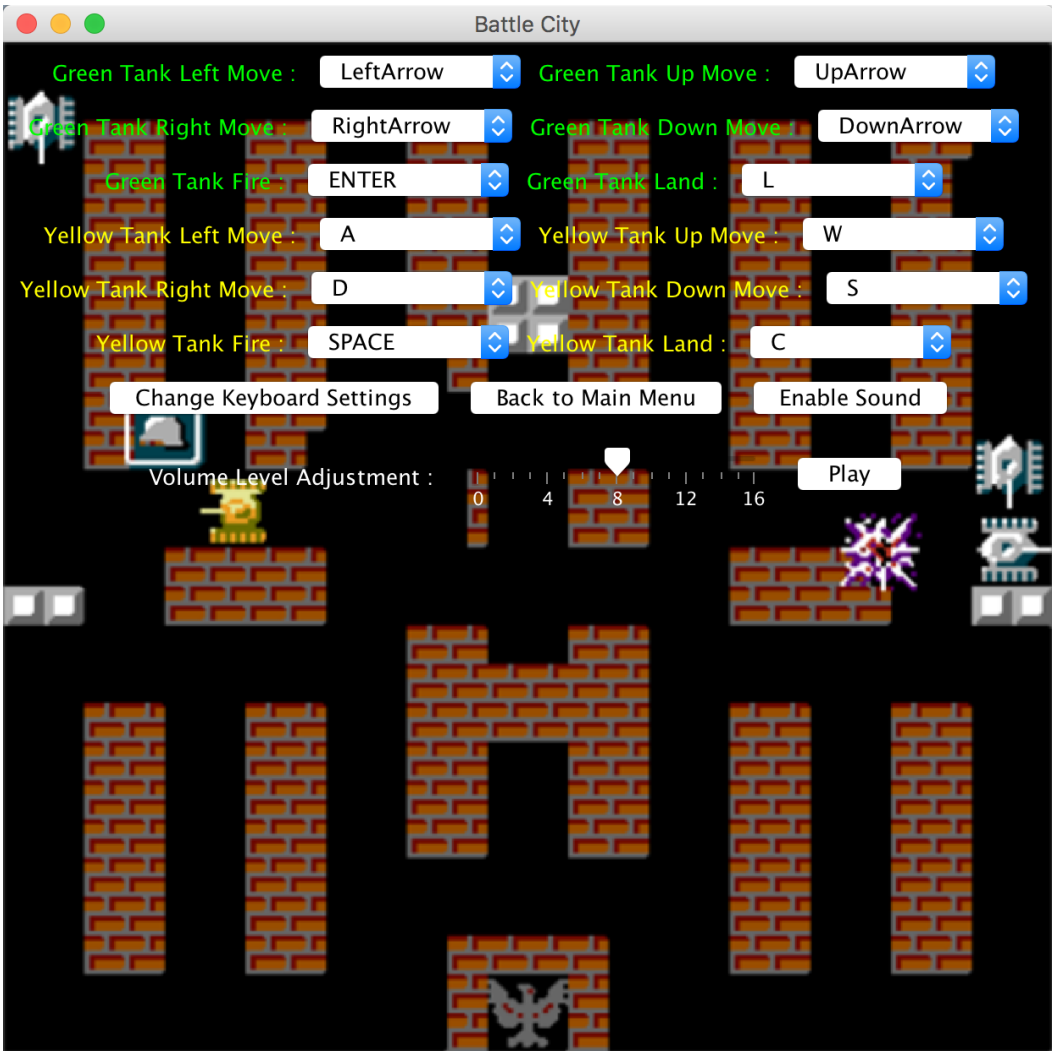


Figure 1: Main Menu of Battle City

❑ When the user clicks to the settings button, the settings menu is displayed. The comboboxes in this menu shows the selected keyboard settings for the game and the labels before each combobox indicates which input of the game is controlled by the settings provided inside the comboboxes. Also, the labels show the color of the tank that will be controlled by the given keyboard settings. The user can select any settings

given in the combobox and click to the change settings button to apply the keyboard changes he made in this screen. Besides, the user can enable or disable the sounds of the game by clicking the enable sound toggle button on this screen. After the enable sound button is set, the user can adjust the sound volume by using the slide bar in this screen. If the sound is not enabled, the changes made in the slide bar does not affect the game since the sound is mute. The user can use the back to main menu and play game buttons to navigate to the main menu and options menu of the game respectively.

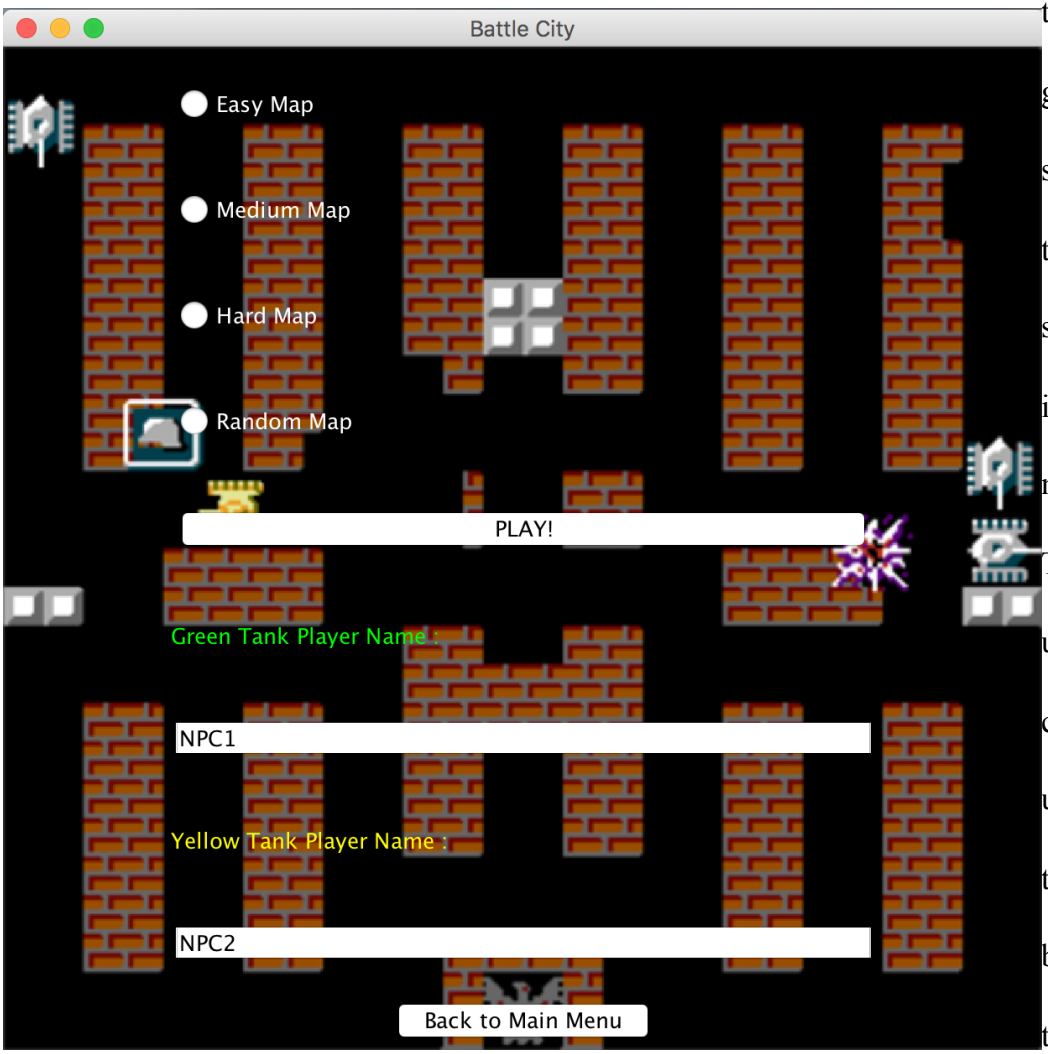


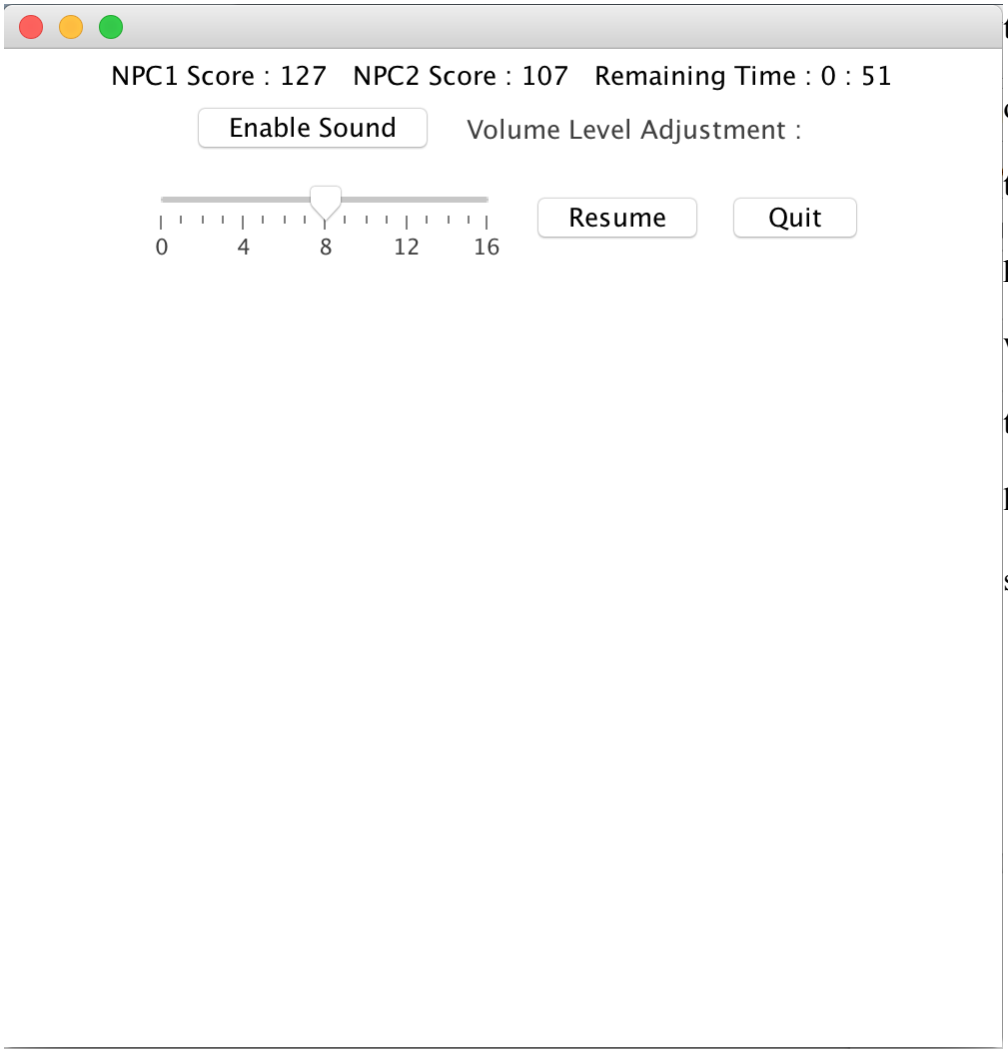
Figure 2: Settings Menu of Battle City

❏ The play game button in the main menu and the play button in the settings menu opens the screen for the options of the game. In this screen, the user can choose the

difficulty of the map by clicking the radio buttons on the screen. It is also possible to start a game with a randomly generated map by choosing the random map from the radio buttons. The user can write his player name under the color of the tank that he wants to use in the game. However, the user should be aware that the keyboard settings of that tank is set in the settings menu and he needs to go to the settings menu in order to customize these settings. There is no need for specifying whether the game is a singleplayer or multiplayer game. The user can click the play button to start the game.

Figure 3: Options Menu of Battle City

- The game uses sound clips when a specific event in the game occurs. Therefore, there is no guarantee that an audio clip will be played or not during a particular game and the user does not choose the audio clip that will be played in the game. In addition to this aspect, the same events can play different sound clips for different color of tanks.
- In the game screen the necessary information for the game including the players' name, score and tank health along with the remaining time is displayed. Also, the game screen shows the map of the game with the game components in it. The user can take three different actions in the game, which are move, fire bullet and land mine. The aim of the game is to destroy the opponent's tank before the time is over. If the time is over, the player with the higher score is



assumed to win the game. The time limit for finishing the game is one minute at the beginning of the game.

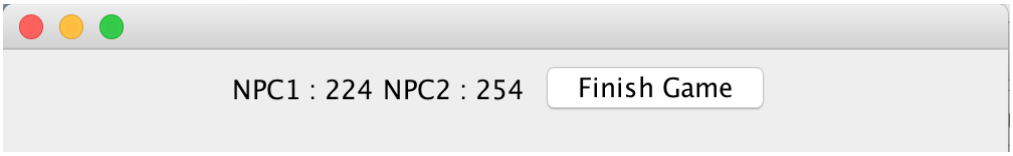


Figure 4: In-game Screen of Battle City, 2 NPCs playing

- The computation of the score is as follows. Score = health * tank level + remaining time in terms of second * 2 + number of bricks destroyed by the tank * 10 + number of steels destroyed by the tank * 20 + 40 (40 is added only if the player destroys his opponent's tank). Therefore, the score starts with a constant value and decreases or increases as the game progresses.
- The game has five power ups. The bullet power up increases the bullet damage from one to two. Also, the bullet power up enables the user to destroy the steel obstacles as

well as the bricks whereas the user can only destroy the brick without the bullet power up. The mine power up increases the number of mines the tank has. At the beginning of the game, the tank has five mines having the damage value one. The number of bullets tank has is unlimited. the time power up adds fifteen seconds to the remaining time. The tank upgrade changes the image of the tank and doubles the health of the tank. The brick rider power up enables the tank to go over the bricks and destroy them. The duration for the brick rider is five seconds. Also, the mines that are landed becomes invisible after three seconds and they do not become visible after becoming invisible. The power ups appear in random places at random times.

- In addition to these features of the game, the game has bushes that hides the power ups, tanks, bullets and mines. There are three different colors for the tanks. The white color is used for the npc tanks and the yellow and green colors are used for the player's tanks.
- The player does not need to worry about the number of the players in the game. If there are inputs from both players' keyboard settings, the game continues as a multiplayer game. If there is no input from any one of the players, the computer takes the place of the player and game continues as a single player game. After the single player game is started, an input from the player can change the game back to a multiplayer game as well. It is also possible to have a computer versus computer game without any players if both of the players does not provide any input. Five seconds is the threshold for determining that a player does not exist and computer should take his place.
- The user can pause the game by clicking the pause button in the game screen. After the game is paused, the pause menu displays the current scores of the players and the

remaining time of the game. Also, the players can change the sound settings from the pause menu similar to the settings menu. The resume button in this screen returns to the current game whereas the quit button finishes the game and displays the end game screen.

Figure 5: Pause Menu of Battle City

- At the end of the game, the end game screen shows the scores of the players. The finish game button in this screen goes to the main menu screen and finishes the current game.

Figure 6: End of Game Screen of Battle City

4. The Changes Made During The Implementation Stage

- We made changes on the computation of the scores and the effects of the power ups to the tanks in order to make the game more logical.
- In the design the game screen is notified by the game map. However, since the game screen needs to know about the player scores and names, the game screen is notified by the game and game map in the implementation. Therefore, the closed architecture of the subsystems is changed to open architecture.
- The game map and item classes were in the same subsystem in the design stage. However, the items did not call the game map methods or know about the details of the game map in the implementation. Also, after the game map notifies the game screen, the game screen retrieved information from the item classes. Therefore, the hierarchy of classes changed from the partitioning of game map and items and coupling with the game screen to the partitioning of the game screen and items and coupling with the game map.
- The high scores class added for reading and writing the high scores. Also, the game options screen added to the user interface for displaying the options of the game. Besides, the map generator class is added to separate the map generation from the game map in the random map case.
- Some redundant methods and attributes removed and some new methods and attributes added. For instance, in addition to the update items method of the terrain class, the update obstacle, update tank and update power up methods, which are called by the update items method when necessary, are added. For another example, the

bullet level attribute is removed from the bullet class since the bullet level can be obtained from the damage value and the tank class keeps the level of bullets as well.

5. Incomplete Parts Of The Project

- The player base is not implemented.
- The end game conditions concerning the player base is not implemented.

6. References

Bruegge, Bernd and Allen Dutoit. *Object-Oriented Software Engineering Using UML, Patterns, and Java*. 3rd ed. Edinburgh: Pearson, 2014. Print.

Audio Clips are taken from:

http://www.moviewavs.com/

Images are taken from:

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