Bilkent University

Department of Computer Engineering

CS 319 - Object-Oriented Software Engineering

CS319-2F-DE: Defender

Term Project - Final Report

- Project Name: Defender
- Group No: 2F-DE
- Group Members: Büşra Ünver, Celal Bayraktar, Javid Haji-zada, Samir Suleymanli, Selen Uysal

1. Introduction

As the first iteration process came over, we started implementation of full capable game with new features. As a group we have used Intellij Idea for implementation and a GitHub repository for synchronous collaboration on the implementation. Now, by the end of the second iteration, we could manage to finish the fundamental gameplay of nostalgic Defender, plus some additional features that were presented in Analysis Report. Section 2 furtherly discuss implemented functionalities.

2. Implemented Functionalities

For the second iteration, additional to the Main Menu Screen, Single Player Endless Mode, How to Play Screen and Exit button, we have implemented Single Player Story Mode, Credits, Shop and Highest Scores Screens. We spent time for screen transitions and making our user interface user friendly both for the implemented parts at Iteration 1 and 2.

2.1 Shop Screen

We put 6 spaceships with different colors and they can be bought if the player has enough coins. It is a one-time purchase and any player is able to equip already bought spaceships.

2.2 Highest Score Screen

Here, we list the top 10 scores gained in Single Player Endless Mode.

2.3 Credits Screen

We implemented simple credits screen where contributors names are listed.

2.4 Single Player Story Mode

We implemented it from scratch and put enemies with different images and functionalities at each wave. For the first wave, the players are introduced with 5 aliens. The player is introduced with queen and there are 10 total enemies (7 alien and 3 queen) at the second wave. At the third wave, asteroids are new to the player and there are 12 aliens and 3 asteroids. At the fourth wave, the players try to rescue astronauts to gain more points and there are 4 aliens, 3 queens, 5 asteroids and 3 astronauts. At the fifth wave, there are 7 aliens, 5 queens, 3 darwins (introduced in this wave), 10 asteroids and 5 astronauts.

3. Design Changes

Implementation stage led us to notice that we are not able to fully implement our system in the exact way that we have planned. Even though these differences did not lead to any change in the general design structure, we still have some minor new additions.

3.1 Extended Dependencies

During the first and second design stages, we tried to structurize our system in a way that packages and subsystems would not be depended on each other in a restricted manner. We have seen that implementing independent classes would make our design ambiguous. Thus, for simplicity purposes of our design, several new dependencies were introduced. For example, our GameManager class was not planned to be dependent on Stage object, however, for screen transitions purposes we have introduced this dependency.

3.2 Game Constants

During implementation we have faced with several constants, which are used in different classes, such as screen size, image paths and so on, so we decided to implement Constants class to make our code cleaner and possibility to make easier changes in several classes at the same time.

3.3 Newly Introduced Sub-Packages

During implementation of our screens, we started to use “.fxml” files for UI designs. By this, we have introduced two new packages screen.fxml and screen.controller. Controller classes basically handles screen transitions and some minor UI design changes. This also led us to remove our previously designed ScreenManager class.

3.4 Private Utility Functions

During implementation of our screens, we have introduced new sub-functions to make our code cleaner. These sub-functions were not included in our design since we didn’t know the exact contents of algorithms.

4. Lessons Learnt

During the implementation phase, we could not properly manage task division. We tried to divide activities into tasks and tasks into subtasks. Some of our group members also learnt how to collaborate through VCS (Version Control System), in our case Git.

After implementation we had several outcomes:

- Planning is important.
- Whatever can go wrong will go wrong. (Yes, Murphy)
- Group needs to be responsible towards each other and manage each other.
- Beside team meetings, dialogs are also important in group work.
- Various design patterns helps project to be more readable and implementable.
- Diagrams are great way to explain how the project is going to be developed in a higher level and helps us to predict the end project.

5. Users Guide

Users should have JDK with Java 8 or higher versions. You may follow below instructions:

- Download git repository.
- Open src/src as Java Project.
- If necessary, set the main class of the project in the configurations as "defenders2FDE.Main"
- Run the project in the IDE to play the game.

5.1 System Requirement & Installation

The game can run any operating system that supports Java runtime environment. Game does not require any serious memory and storage to run game because the game uses a small amount of memory which is around 200 MB RAM and storage which is around 300KB HDD.

5.2 How to Use

In all game modes, the player should shoot and avoid enemies. Game Modes Pop Up will be shown on the Main Screen if the user click "Game Modes" button.

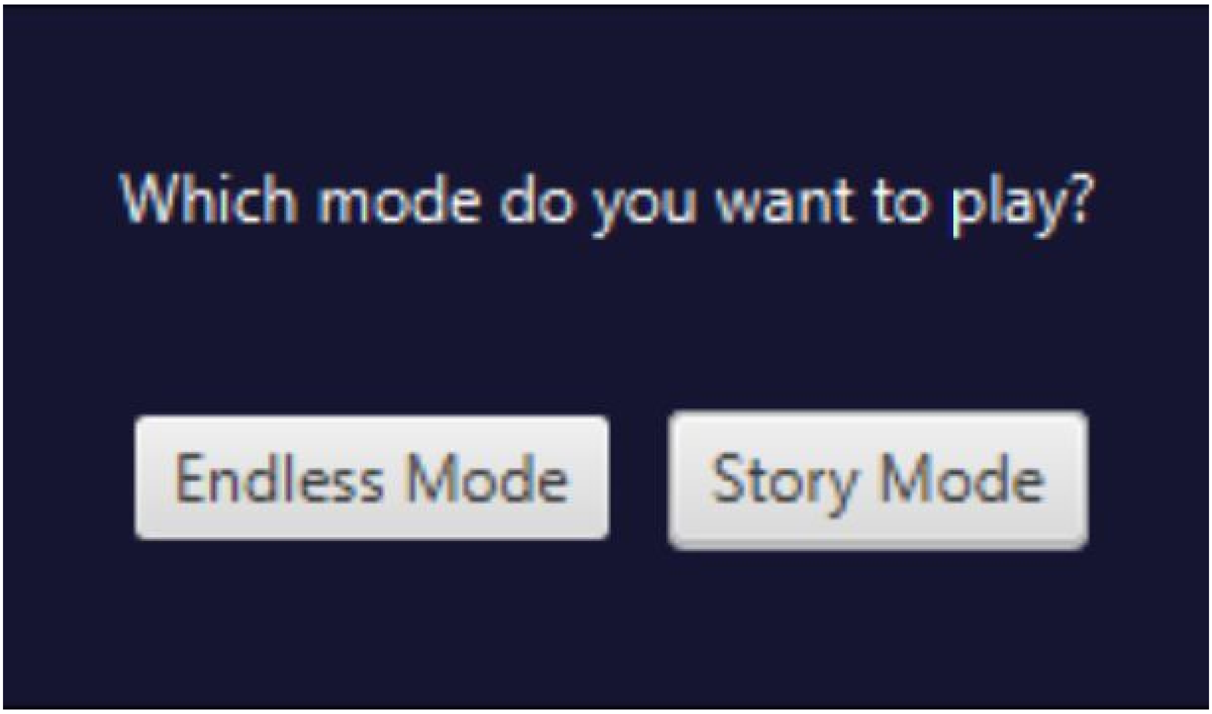


Figure 1 - Game Modes

5.2.1 Play Single Player Story Mode

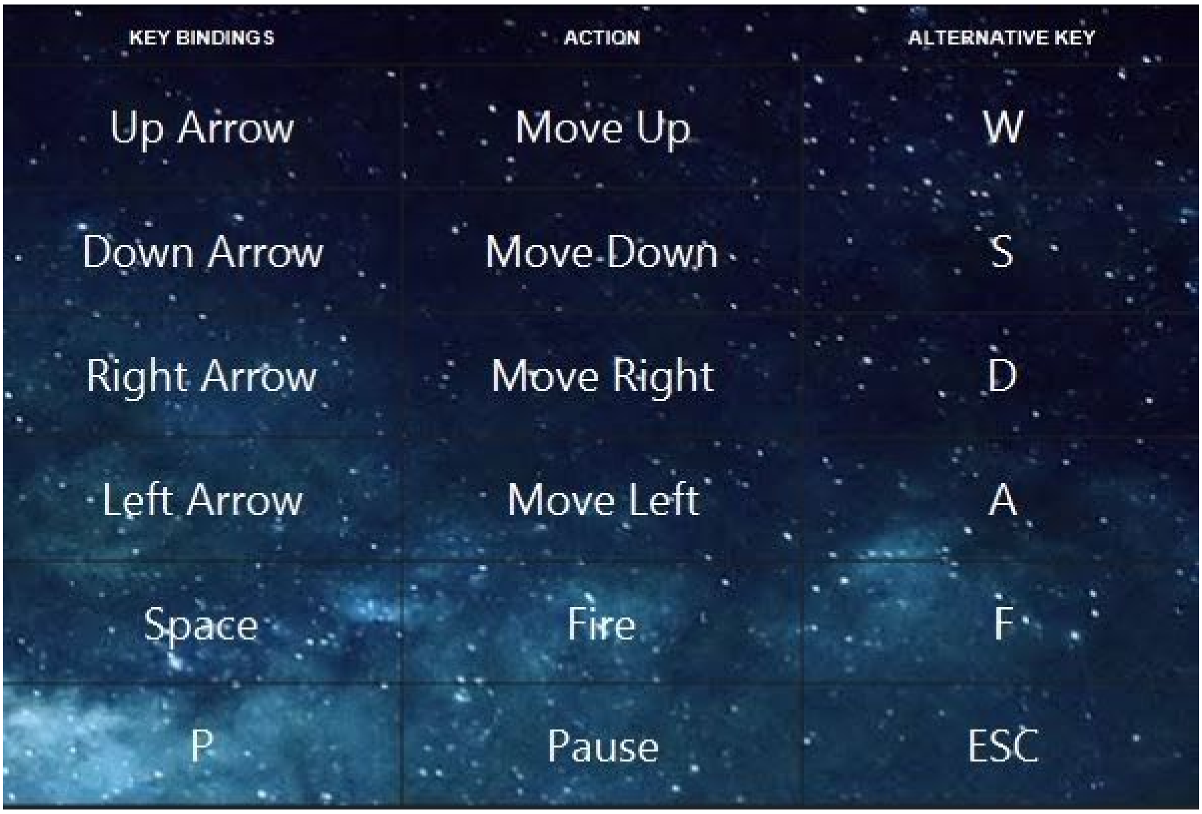
In order to start playing the single player story mode, the players need to click the “Game Modes” button in the main screen and then click “Single Player Story Mode” from the Game Modes pop-up.

5.2.2 Play Single Player Endless Mode

In order to play this mode, players should click the “Game Modes” button in the main menu and click “Single Player Endless Mode” from the “Game Modes” pop-up. However, the player can only play this mode if the single player story mode is completed. If it is not completed, there will be a lock sign and the player will not be able to open the mode.

5.2.3 How to Play

The players can learn instructions for playing this game by clicking the “How to Play” button from the main menu. They can go back to the main menu by the “Main Menu” button.

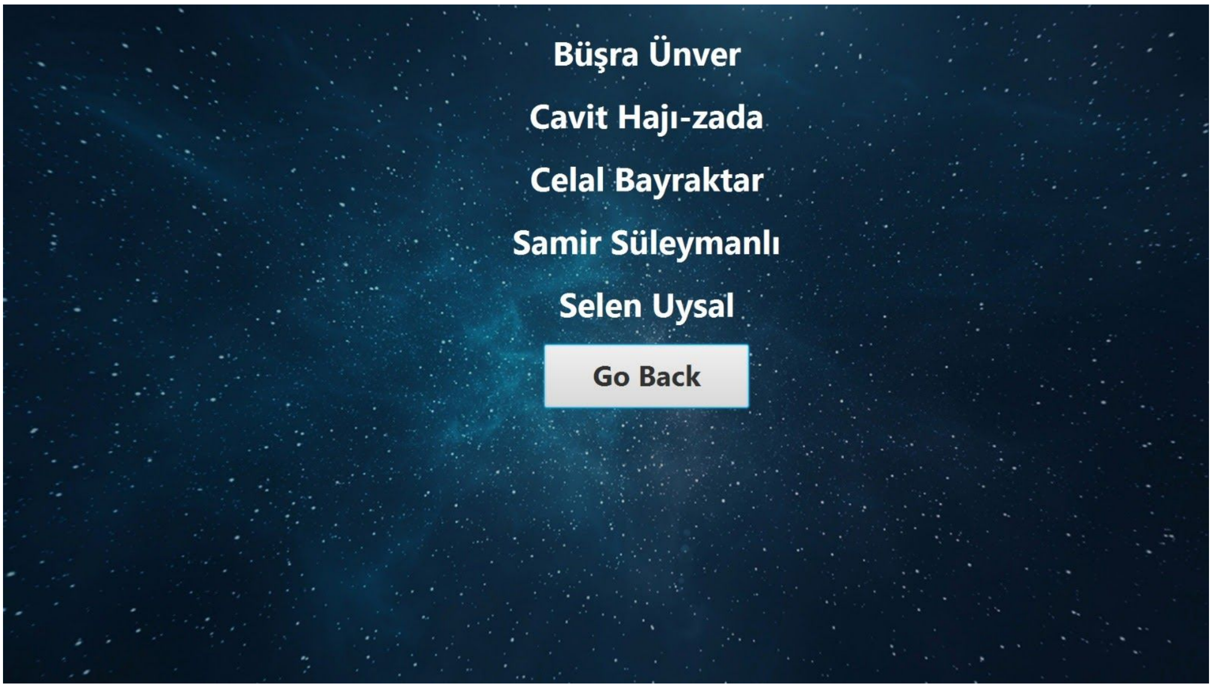


KEY BINDINGS	ACTION	ALTERNATIVE KEY
Up Arrow	Move Up	W
Down Arrow	Move Down	S
Right Arrow	Move Right	D
Left Arrow	Move Left	A
Space	Fire	F
P	Pause	ESC

Figure 2 - How to Play Pop-up Screen

5.2.4 Credits

The players can view credits by clicking the “Credits” button from the main screen. They can go back to the main menu by the “Main Menu” button.



Büşra Ünver

Cavit Hajı-zada

Celal Bayraktar

Samir Süleymanlı

Selen Uysal

Figure 3 - Credits Screen

5.2.5 Change Settings

The players can mute/unmute sounds and effects from the main menu on the top right by clicking on the symbols. The sounds and effects information will be saved after the player moves to another screen.

5.2.6 High Scores

The high scores of the player in the single player and two players endless modes can be seen by clicking the “High Scores” button from the main menu. The player can reset his/her high scores by clicking the “Reset” button at the top right. He/she can go back to the main menu by the “Main Menu” button.

6. Contributions

6.1 Büşra Ünver

- Iteration 1, Analysis Report: Created half of the mock-ups, written introduction and summary of the project.
- Iteration 2, Analysis Report: Improved class diagrams, stage diagrams, sequence diagrams, improved non functional requirements.
- Iteration 1, Design Report: I wrote the design goals and trade offs and applied design patterns.
- Iteration 2, Design Report: I improved deployment diagrams, manager classes.
- Game: Designing fxml views of the game screens, binding buttons to screens and assigning them action controllers. Creating various game components of game like pop-ups, labels, panes, screens' stages.diagrams.

6.2 Celal Bayraktar

- Iteration 1, Analysis Report: Introduction, Mockup UI Design
- Iteration 2, Analysis Report: Improvement Summary, Object and Class Model
- Iteration 1, Design Report: Subsystem Services
- Iteration 2, Design Report: Improvement Summary, Low Level Design
- Game: Shop Screen design and implementation, spaceship design
- Game Trailer

6.3 Javid Haji-zada

- Analysis Report: Object and Class Model, Functional/ Non-Functional Requirements, Revision of Overview.
- Design Report: Subsystem Services, Final Object Design, Packages, Applied Design Patterns, GameComponent Classes, Boundary Conditions
- Final Report: Introduction, Design Changes, Lessons Learned

6.4 Samir Suleymanli

- Iteration 1, Analysis Report: Use Case Diagrams, Activity Diagrams
- Iteration 2, Analysis Report: Revision of Use Case Diagrams, Activity Diagrams
- Iteration 1, Design Report: Final Object Design
- Iteration 2, Design Report: Revision of Final Object Design
- Game: Story Mode

6.5 Selen Uysal

- Iteration 1, Analysis Report: 6.2.2 Sequence Diagram
- Iteration 2, Analysis Report: 2.8 Game Modes, 3.1 Game Modes, 3.3 Highest Scores, 3.5 Settings, 6.2.2 Sequence Diagram (review), 6.3 Object and Class Model
- Iteration 1, Design Report: 4.2 Final Object Design (4.2.1, 4.2.2, 4.2.3, 4.2.4, 4.2.5)
- Iteration 2, Design Report: All of 4.2 Final Object Design review, design patterns
- Final Report: Backbone of the final report, Implemented Functionalities, User's Guide
- Game: Game sounds, enemies and spaceships' designs, game assets, designing scenes of the game, backgrounds, imageviews, pop-ups
- Presentation: Slides of Iteration 1 and Final Presentation