

# Bilkent University

## Department of Computer Engineering

---



### CS-319 Object Oriented Software Engineering Project

#### Sword & Shield: A Space Adventure

## Final Report

### Iteration II

#### Group 2D

Akın Berkay Bal

Eren Aslantürk

Mehmet Enes Keleş

Sadık Said Kasap

<b>1 - Introduction</b>	<b>3</b>
<b>2 - Design Changes</b>	<b>3</b>
<b>3- Lessons Learned</b>	<b>4</b>
<b>4 - User's Guide</b>	<b>5</b>
4.1 - System Requirements & Installation	6
<b>5 - User Interface</b>	<b>6</b>

## 1 - Introduction

We have designed and implemented a game called Sword & Shield: A Space Adventure. It is a top-down strategy/tower defense game which is turn-based and playable by two players, an attacker and a defender. It will be playable using only a mouse. The flow of the game goes like this: in every turn the defender will place some turrets until the his/her time ends. Then the attacker will place some factories, with the constraints of given gold and time. The attacker will try to penetrate the defense this way. If the attacker cannot breach through the defense, the turn will be won by the defender and vice versa. Every turn, the winner will gain a predetermined amount of gold. Unfortunately the balancing has not been completed yet; therefore, the stats of gameplay elements may still change for balancing. We believe more user feedback is required in order to achieve more enhanced gameplay experience. Also we could not be able to implement load/save feature but everything else is finished.

## 2 - Design Changes

We have made a lot of design changes throughout the implementation. Especially while working on design report. We have decided to work on a three layer architecture in first iteration. In second iteration we changed to MVC thanks to our Instructor. Also we decided to write a Collision Manager for collisions that will occur between projectiles and ships. We also moved methods of entities to their respected manager classes. We also decided to add load/save and pause functionalities to the game. Unfortunately load/save function had to be dropped throughout the project due to time constraints and our inexperience in that field. All changes are made to have a low coupling, high coherence system. We wrote a Drawable

interface but as project went on it became a burden rather than being an ease for us and it did not achieve the things we have wanted from it. Therefore we dropped it along the way as it was more efficient not to have it.

### 3- Lessons Learned

During the project, we have learned that planning ahead of time is very important. Thanks to our group meetings and simplified design of our system we easily wrote the code, even though there were a lot of changes that we have made throughout the project. We also learned that, earlier ideas about classes could be wrong/lacking and needed to be changed/updated. For example, we made some mistakes on various UML diagrams at the first report. We also learned working with IntelliJ and Github. IntelliJ was a very comprehensive and easy to use IDE that we needed as we used it synchronized to Github. We also learned the difficulties of game design. For example, we have faced a lot of difficulties about game balance. During first runs of our game the attacker side was more powerful than the defender side. We tried to fix this unbalanced situation. We believe there is still progress to be made in terms of game balancing. This can be achieved when we get more user feedback. We also learned the difficulties calculating geometric position of objects. We believe that this cause due to Java standard library does not have *vector* class which commonly uses in other major game development libraries.

One of the problems we have faced during the second iteration was group meetings. We met regularly until last two week. However, last 2 weeks because of exams and other projects we could not be able to meet regularly as 4 people. Because of that we weren't able to made the progress we planned. Unfortunately we couldn't implement save/load feature.

## 4 - User's Guide

### 4.1 - System Requirements & Installation

The game runs on any platform that has JRE 8. The game's performance depends heavily on the memory size and CPU. To run the game simply download the source code from github repository and click to the JAR file. We set mouse as the main input device for the game therefore you will need a mouse. Without complicated keyboard key combinations, it is aimed to make players enjoy through gameplay. Since the game is for two players you will also need a friend to play with.

Link: <https://github.com/Ulkudas/2D.Sword-Shield>

### 4.2 - How to Play

The game initiated by clicking to JAR file. When the game is started, there will be a menu with some some options. New players can use "Play" button to start a new game as a fresh startup. As next, the gameplay screen will appear. Here, they can see scores and status of both players. Due to turn based future of the game, attacker and defender status change. When the player is the attacker, he tries to win the turn by charging with offensive units through defender's base. If he has reactors on his base, he can gather gold. While he is the defender, he has to survive from assault of the attacker. He can gain profits from each unit destroyed and from reactors constantly. With the profits, factories and reactors can be purchased from the panel related with the player. As a general purpose, the player who wins the turn will get a score.

## 5 - User Interface

### Game Menu



We have designed a menu for the opening screen of our project S&S: A Space Adventure. We chose a nebula themed picture to emphasize the space adventure and to impress gamers who are interested in space themed games. With “Tron” like text alignment we also symbolize the futuristic content of the project. The menu includes 5 buttons: Play, load, credits, music and help. Play button is the access to main gameplay of the project. Load

is the get back a gameplay which is already started before by players. Credits shows the information and details about the project. Music is an option which allows people to activate theme song of the game to enhance gameplay taste. Help menu gives detailed information about the game objects.

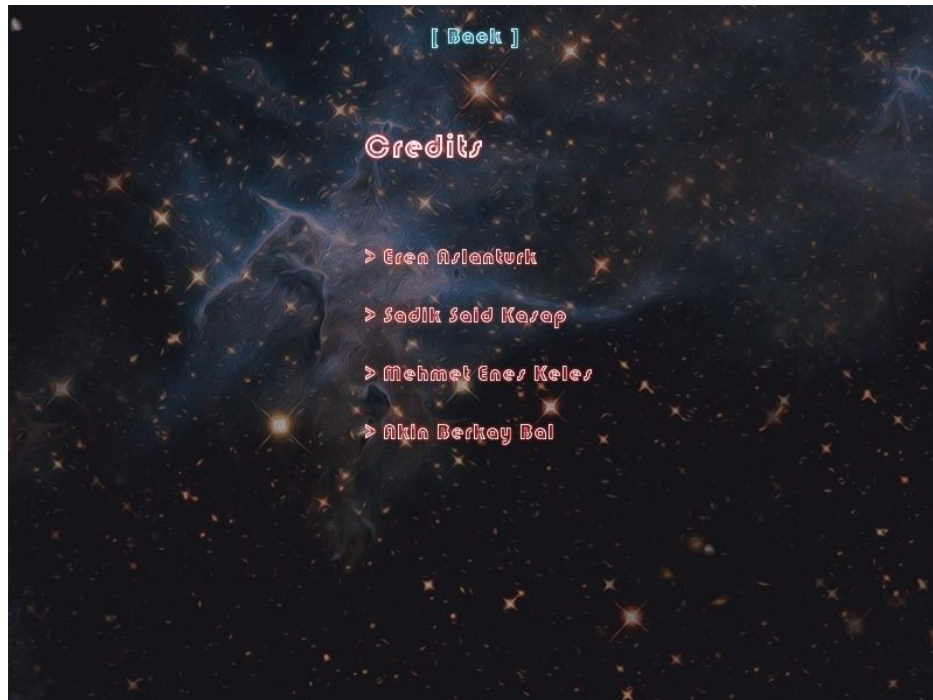
### Game Field



Gameplay is the main content of the game. Here, players, also known as, the attacker and defender can battle against each other as a pvp battle where both sides have own base with accessible player menu contents, i.e turret factories, offensive unit factories and reactors. On the game field attackers can use factories to train offensive units to assault on defender's base. On the top of window, players have own status information, i.e balance,

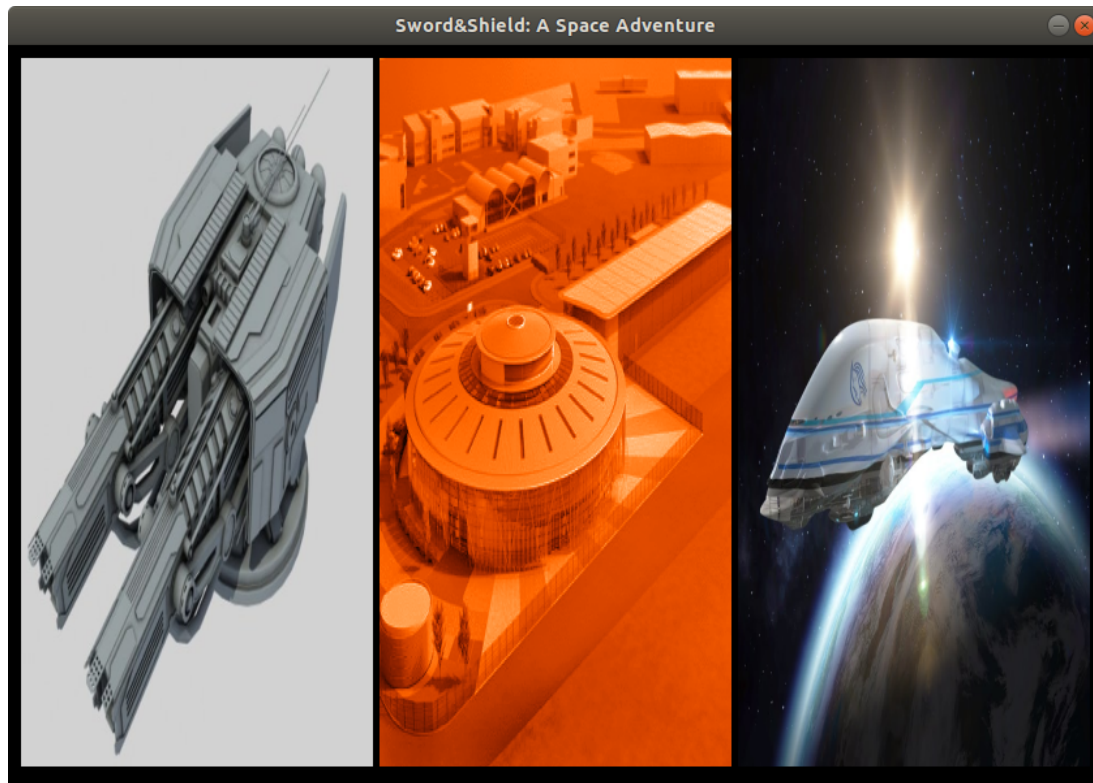
score and timer. On the bottom, there are play, pause and save buttons. Players can pause the gameplay anytime and save if they wish to continue later or to ensure the progress.

### Credits Screen





## Help Screen



### Athletic Spaceship

These units are fast but have light armor and average health

#### Stats

Speed:3  
Armor:2.5  
HP:234.0  
Reward:30

Spaceship1 Spaceship2 **Spaceship3** Spaceship4

## Load Screen

