**Bilkent University**

**Department of Computer Engineering**

****

**CS-319 Object Oriented Software Engineering Project**

**Sword & Shield: A Space Adventure**

Final Report

Iteration I

**Group 2D**

Akın Berkay Bal

Eren Aslantürk

Mehmet Enes Keleş

Sadık Said Kasap

[**1 - Introduction**](#_fl4ksj5ctwxy) **2**

[**2 - Design Changes**](#_ujdzncywcll6) **3**

[**3- Lessons Learned**](#_6abyjrh3zwde) **4**

[**4 - User’s Guide**](#_e0ioes445pti) **5**

[4.1 - System Requirements & Installation](#_ko2igclbxjn7) 5

[4.2 - How to Play](#_1mg3uhi2hrbt) 5

**5 - User Interface 6**

## 1 - Introduction

We have designed and implemented the game called Sword & Shield: A Space Adventure. It is a top-down strategy/tower defense game which is turn-based and playable by 2 players, an attacker and a defender. It will be playable through a mouse. Firstly, the defender will place their turrets then the attacker will place the factories, with the constraints of given gold and time. As we believe, there should be a second chance for everyone, in case, attacker fails to win at its first attack, if the player has already built a reactor, more of factories can be built with the income of reactor. Every turn, the winner will gain a predetermined amount of gold. At this point, the balancing has not been completed yet; therefore, the gameplay elements may still change for balancing. We have worked hard until this point but it still has planned parts with some missing features. We will implement these at second iteration such as pause, save and load functions.

## 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 since then. We decided to add a Geometry Manager as we need to calculate the projectiles, how they hit and how they will behave. We need to take into account their ranges, their speed and where and when can they hit the passing ship. Also we decided to write a Collision Manager for collisions that will occur between projectiles and ships. Manager classes and factory classes are separated. We also moved methods of entities to their respected manager classes. Also render method of entities are moved into the Game Manager. We also decided to add load/save and pause functionalities to the game.

All changes are made to have a low coupling, high coherence system. All future changes will also take this approach into the account.

## 3- Lessons Learned

During the project, we have learned that planning ahead is very important. Thanks to planning classes and overall design of our system we would be able to easily write 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 the Github. We also learned the difficulties of game design. For example, we have faced a lot of difficulties about game balance. Attacker was more powerful than defender. We tried to fix this unbalanced situation. We believe there is still progress to be made in terms of game balancing. We plan to fix this problem in iteration 2 by testing the game several times.

One of the problems we have faced during to iteration 1 was meetings. We met regularly until last two week. However, last 2 week 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.

## 4 - User’s Guide

### 4.1 - System Requirements & Installation

In order to initiate the Sword and Shield, java environment is an obligatory requirement. This game is implemented for universal usage to serve a wide range of audience and spread the joy among its fans. Although it is playable on XP and better windows versions as well as on mac versions for the best game experience and graphics, it is suggested to use on most recent versions of mac, or windows 7 and above. We set mouse as the main input device for the game, especially for gameplay process and interactions with panels. Without complicated keyboard key combinations, it is aimed to make players enjoy through gameplay.

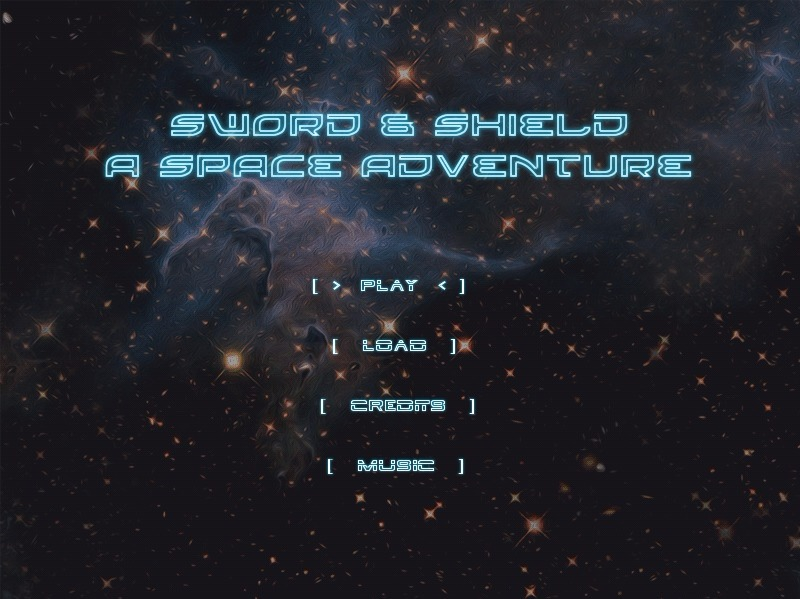
The game is downloadable from the github web link on Github of "Sword and Shield: A Space Adventure” where the game is ready to obtain. The gamers can download the game any directory on their computer as in their wish. In the file, the program code of the game is available and ready to use. The gamers can use any IDE to run the game using its code.

### 4.2 - How to Play

The game is initiated from the IDE using its code where the gamers downloaded. 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 gain more profit. 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. For the game balancing, we will consider some improvements for the game in next iterations.

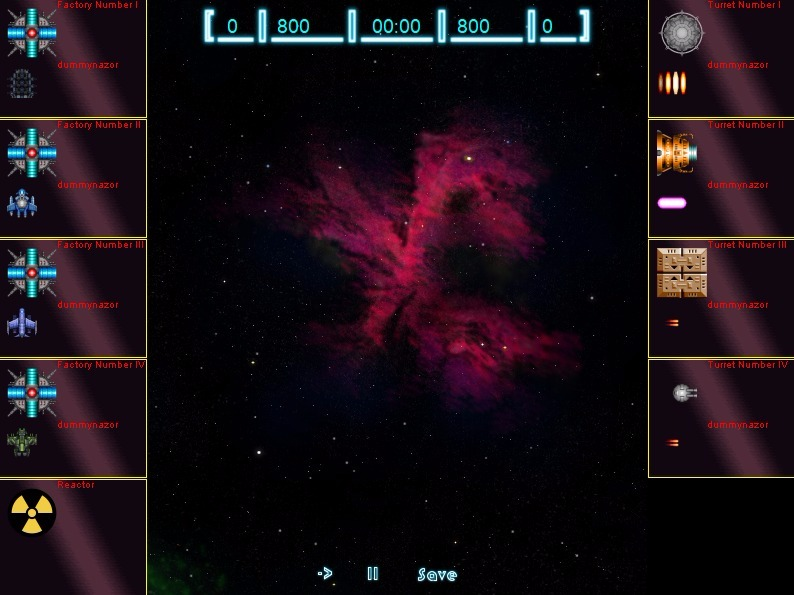
## 5 - User Interface

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 4 buttons: Play, load, credits and music. 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.

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



Credits window is the informative part of the project. Here, players can observe detailed information about the game. In future, more details may be added depending on the contents of the project. This section serves people to enlighten them about contributors, the game and make the application trustable. Eventually, customers will not a nameless or an anonymous brand. When players are affected by the delicacy of the joy of a game, they become more curious and willing to test more of the projects or games of same contributors. As a famous person said, knowledge is power.