|  |  |  |  |
| --- | --- | --- | --- |
| **CS102** | **Spring 2017/18** | Project Group | 1A |
| Instructor: | **David Davenport** |  |  |
| Assistant: | Çağlar Öksüz |  |  |

|  |  |  |
| --- | --- | --- |
| **Criteria** | **TA/Grader** | **Instructor** |
| Presentation |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Overall |  |  |

~ Droneer ~

BAZUKA

Baykam Say, Alp Üneri, Zübeyir Bodur, Uğur Erdem Seyfi, Ege Kaan Gürkan, Agil Aliyev

|  |
| --- |
| Project Requirements Report  ( version 2 )  12 March 2019 |

# Introduction

Learning to program using a programming language such as Java can be a daunting task because of its complexity and difficulty. This is due to several factors, mainly the fact that a programming language comes off as too abstract for novice programmers because it is not possible to immediately see the results of the code that has been written or make changes to it on the fly. This makes it difficult for beginners to form connections as to which piece of code results in which part of the program they are attempting to create. In order to mitigate this problem, we propose a programming oriented game which will provide a more fun and engaging alternative to simply learning a programming language by trying to build sample programming assignments. We aim for our game to be an effective learning platform for novice programmers while providing veteran programmers with fun and captivating gameplay.

# Details

## Features

Our programming oriented game will feature a two-dimensional gameplay zone in which the player pilots a drone by providing it with a set of instructions written in a separate IDE-like section of the GUI beforehand. The player will have to write a complete set of instructions for the drone in the form of a modified version of Java and will not be able to make changes to the code during an encounter.

The encounters will consist of an enemy drone that can be dealt with in several ways. Each drone will have different parts of it (such as its engines, turrets et cetera) be individual objects with a diverse set of properties. The game will also feature an in-game text editor that will be using the JDK to function.

## Drones

The drones will have five core stats: health, shield, energy, ammo, and repair power. They will unlock throughout the game. Health is the condition of the drone and the drone gets destroyed if its health is 0. Shield is the virtual bubble around the drone which protects against physical impacts and bullets. Energy is the total energy generated by the generator in the drone. Every system in the drone (shields, lasers, thrusters etc.) uses energy. The ammo is the ammunition present in the ship. Rockets and bombs use ammunition while lasers don’t. The repair power is the amount the drone can repair in a specified time. Throughout the combat some systems in the ship get damaged. In that case the drone can repair the systems but it has to choose which systems are the most essential in order to give priority to those systems. This 5 stat concept teaches the player optimization while making the game more complex and challenging.

The drone will also have multiple subsystems: the nuclear generator, the shield generator, the engine, and weapons (lasers, rockets, bombs).

## The Code

All the drones will extend an abstract base drone class which will provide the basic methods for the drones to function. There will be a run() method that will start when the game starts. There will also be other methods such as onHitByBullet() and onEnemyDetect() that will be automatically triggered when a certain condition is satisfied. The class will also have proper-ties that are the subsystems of the drone. For example, if the player wants to fire the lasers they will need to write “lasers.fire()”.

## Learning Opportunities

Our programming oriented game will allow the player to increase their knowledge of if/else statements, several types of loops including the while, for, and do/while loop, Boolean logic, the concept of methods, object oriented programming, as well as basic Java syntax, algorithmic thinking ability, and problem solving/critical thinking.

## Differences from Competitors

There exist other programming oriented competitor games with similar aims or gameplay as our proposed programming oriented game, such as Robocode [1], Human Resource Machine [2], and CodinGame.com [3]. Robocode has a similar concept of combat, but with tanks. The problem with Robocode is that it’s designed for people who already know Java and it’s too complicated for inexperienced programmers. Human Resource Machine on the other hand, is for people who don’t know programming but it’s designed to teach algorithmic thinking, not actual programming. It only has basic functions such as addition and simple loops. Finally although CodinGame.com is similar to Droneer in concept, because of the poor implementation, it fails to achieve its aim. Droneer will have a user-friendly interface that will be easy to understand with intuitive controls. Thanks to its fun and instructive story, it will keep the player’s interest while teaching them coding.

## Future Plans

A full-fledged campaign mode drawing inspiration from the critically acclaimed game Faster Than Light [4]. The campaign will start off with the mothership where the player used to be stationed coming under fire by a massive hostile ship, sustaining heavy damage and losing its ability to move or use long range communications to call for help. The crew of the mothership therefore sends out emergency drones to ask for reinforcements. The main objective of the campaign mode will be to pilot one of these emergency drones safely to civilization on the other side of the galaxy. The drone will be able to travel to different markers on a linear map consisting of multiple branches (similar to the sector maps on Faster Than Light). At each location the drone will be faced with different challenges, such as a pirate or alien ship, a space wreck, or a space station. At these locations the player will be able to make choices (such as whether to investigate the wreck or simply not approach it, or whether to approach an alien ship in a friendly manner i.e. with the shields down or not) that will alter the rest of the campaign. At the locations the player may engage in combat, in which case the programming of the drone (provided by the player beforehand) will kick in. After each successful event the player will receive an upgrade in the form of a better piece of equipment or a new snippet of code. In between each travel the player will have the chance to modify their code or the equipment of their drone. Should the player’s drone fail to eliminate the threats at any location and get destroyed as a result, the player will have to start over with a new drone, but will retain the code snippets that they have acquired.

We are also planning to implement an online multiplayer mode in which the players can upload their ships (with their code) to a server where they can later fight against other ships uploaded by other players.

# Summary & Conclusions

This programming oriented game will allow for players without any background in programming to learn and master its basics in a fun and stimulating environment, as well as allow for more seasoned programmers to improve upon their existing knowledge whilst also having a good time. While the concept of such a game is not a novel one, the implementations of our competitors have been lacking in providing for a wide range of players with differing coding abilities and have not been user-friendly enough. We aim to fix these problems with Droneer.

# References

1. Robocode. <https://robocode.sourceforge.io/>. Latest Version: 5 January 2019. Last Visited: 22 February 2019.
2. Human Resource Machine, Tomorrow Corporation. <https://tomorrowcorporation.com/humanresourcemachine>. Last Visited: 22 February 2019.
3. CodinGame. <https://www.codingame.com/start>. Last Visited: 22 February 2019.
4. Faster than Light: Advanced Edition, Subset Games. <https://subsetgames.com/ftl.html>. Last Visited: 22 February 2019