|  |  |  |  |
| --- | --- | --- | --- |
| **CS102** | **Spring 2018/19** | Project Group | 1A |
| Instructor: | **David Davenport** |
| Assistant: | Çağlar Öksüz |

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

~ Droneer ~

BAZUKA

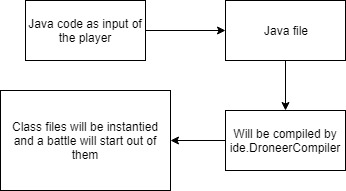
Baykam Say, Alp Üneri, Zübeyir Bodur, Uğur Erdem Seyfi, Ege Kaan Gürkan, Agil Aliyev

|  |
| --- |
| Detailed Design Report  ( version 2.0 )  12 May 2019 |

# Introduction

Learning programming can be a difficult and often boring task for beginners. In order to solve this problem, we have proposed a programming game called Droneer where the players code drones on the screen and make them fight. On the following pages, there will be a diagram of basic organization of the game and two UML diagrams about core design of the project.

# System Overview



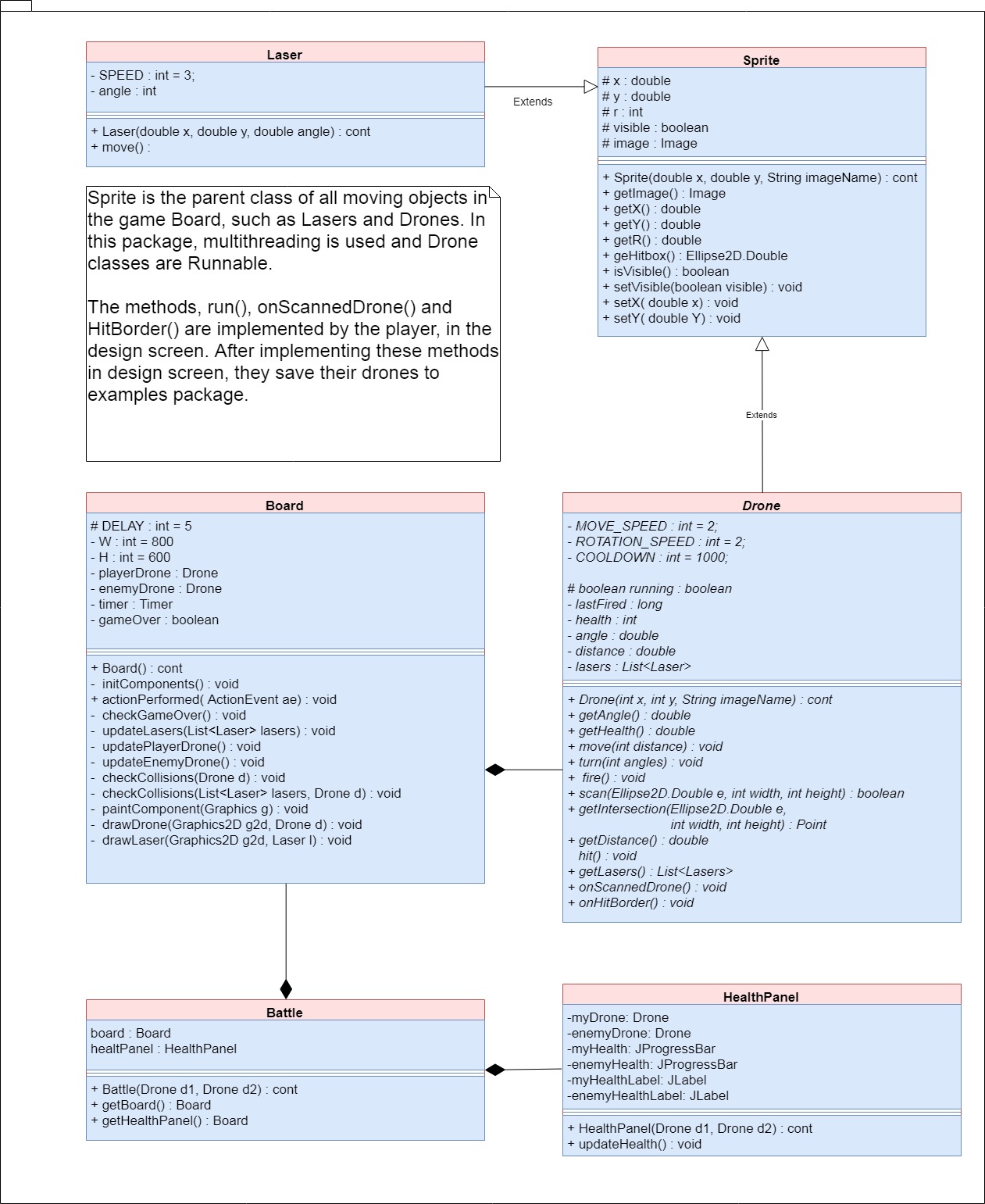
Droneer is offline desktop game. Our program mostly uses javax.swing and java.awt packages. Other classes (TextLineNumber), packages, techniques (e.g. Multi-Threading) libraries of java (e.g. JDK 1.8, JRE 1.8) are also used in this project.

Our program’s main idea is that player enters the code for drones instead of instructing them. For this, the design menu (IDE part) is on the program. Specific buttons on the code editor will call specific actions. Above, there is a diagram briefly explaining how this part will work.

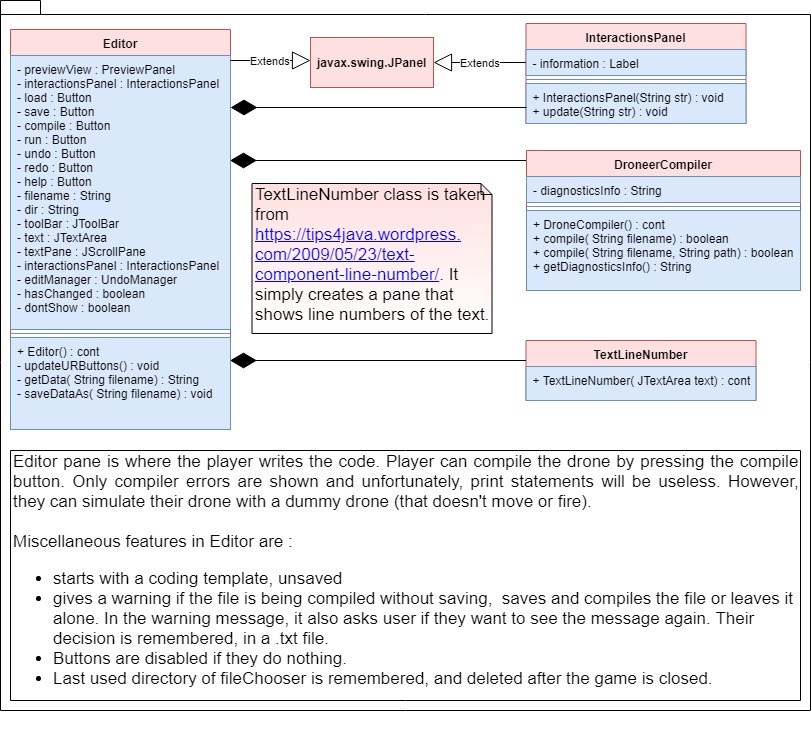
In that specific part, the code player enters will be taken as input from the UI. Then, it will be saved as a java file. A compiler class that uses javax.tools.JavaCompiler will compile this class and the class file generated will be instantiated, possibly to draw and move them in quick battle. If the compilation has errors, they will be shown in InteractionsPanel.

# Core Design Details

## drones Package



## ide package



# Task Assignment

## Baykam Say:

* Worked on the Drones ackage
  + Created the classes within the Drone package
  + Worked on the core logic of the drones
  + Fixed bugs regarding drone detection and lasers
  + Used threads and timers to get the game working
  + Helped with several bug fixes

## Alp Üneri:

* Created the DroneerMaster class to integrate the components of the Project
  + Worked on using the core logic Ege came up with to allow for transitions between screens with button presses
* Worked on the Menus package
  + Worked on the CreditsMenu class
  + Worked on the DesignMenu class
  + Worked on the EscapeMenu class
  + Worked on the WinMenu class
  + Worked on the LoseMenu class

## Zübeyir Bodur:

* InteractionsPanel and rest of the ide package that Uğur made.
  + Bug fixes, simplification of the code.
  + More user friendly features,
    - Undo-Redo feature to the Editor.
    - Hot-keys (e.g Ctrl + S for saving, F5 for compiling, Ctrl + O for opening, Ctrl + Z for undo, Ctrl + Y for redo )
    - Warning message when user tries to compile the file without saving. User can choose whether or not to see the message.
* Improvements to menu UI design.
  + Font sizes and styles in the UI, made all of them bold, and bigger.
  + Worked on EscapeMenu, win-lose screens, MainMenu and buttons much more better (e.g. BoxLayout instead of GridLayout, upper case buttons).
  + Worked DroneSelectMenu class.

## Uğur Erdem Seyfi:

* Editor in ide package, core of the editor (open, save, compile, save as features). In other words, the compile feature.
* DroneCompiler class, that makes compiling much more easier to compile a .java file. This class is used in several places of the project, such as ide.Editor, drones.UpdatableBoard, drones.BoardUpdater.
* Written two classes calles UpdatableBoard and BoardUpdater in order to solve the problem that comes when simulating a drone (Which is actually a .java file). However, after finding a simpler solution, these classes excluded from the project.
* Other contributions to drones package.
  + Contributed the scan part of the Drone class, in other words, fixed the bug that occures in collision detection between radar and target drone.
  + Written a code that allows to simulate Drones, that creates instance of the Drone class by taking a string (classname).

## Ege Kaan Gürkan:

* Worked on the Menus package
  + Worked on the HelpMenu class
  + Worked on the DroneSelectMenu class
  + Worked on the DroneerMenuButton class
  + Worked on the MainMenu class
* Worked on the DroneerMaster class
  + Came up with the core logic used to transition between the screens through toggling multiple JFrames’ visibility

## Agil Aliyev:

* Worked on the Menus package
  + Worked on the HelpMenu class
    - Wrote the text used within the HelpMenu class
  + Worked on the Settings screen
    - The Settings screen was not implemented on the final version of the code due to time constraints

# 5. References

TextLineNumber class in ide package:

Camick, Rob. “Text Component Line Number.” *Java Tips Weblog*, WordPress, 13 Sept. 2009, <http://tips4java.wordpress.com/2009/05/23/text-component-line-number/>. Accessed 5 May 2019.