



# A DAY IN BİLKENT FINAL REPORT

CS319 / OBJECT-ORIENTED SOFTWARE ENGINEERING

ASAF KAĞAN BEZGİN	21402006
ALPER KILIÇASLAN	21502103
IMRAN HAJIYEV	21503442
MUHAMMAD AKSARI IQBAL	21504228
ENES VAROL	21604086

# 1.INTRODUCTION

After delivering the design report, implementation process started immediately by distributing the tasks among the group members. Division of the tasks is done according to the three-tier architecture. Two group members focused on game-object, one group member focused on game managers and game engine, the remaining two members focused on user interface. Façade Design Pattern is taken into consideration while distributing the tasks. For an efficient implementation process, every group member helped each other out and supported their teammates.

Distribution of the tasks started with deciding which parts are harder compared to the rest of the project and who are better at writing specific parts of the game. After the end of the discussion, specific tasks are distributed. During the implementation process, ideas are shared constantly to be on the same page all the time. GitHub is used for sharing code segments and with the help of the developed system of GitHub, every group member started writing their own parts simultaneously. Therefore, tasks are not finished one by one. IntelliJ is the preferred IDE for implementing the game since it works together with GitHub and it is one of the most powerful Java IDE on the market because of its helping on coding.

After every group member finish working on their individual parts, the code on the GitHub merged and debugged. After fixing the errors, several tests are made in order to check if the functions are working properly.

## 2.CHANGES IN DESIGN

Some several changes are made during the implementation. Game-Object is the first part things started to change. New classes were necessary because enum type is not preferred during the implementation. Therefore, different classes for each companion and character are added to the system. Other than that, mayfest, allnighter, double bullet, shield, rage mode classes are added. After the addition of the new classes, new variables also added to several classes and several classes also needed to have different methods. Except adding new classes, methods and variables, some classes, variables and methods are also changed by making them abstract or protected. The associations are also changed since lots of new classes are added to the system and with the implementation of new classes, associations of some of the existing classes are also changed.

Game Engine is also changed. Since A Day in Bilkent is a spawn-based shooting game, spawn enemies method is implemented and it is essential to implement waves. Also create enemy methods included to game engine again to be able to implement wave concept.

Game play did not meet every single requirement specified before due to the time constrains. Companions do not shoot with different patterns. There are no bosses appearing during the game play. There is music on the menu however there is no music while player plays the game.

### 3.LESSONS LEARNT

Communication is the most important skill someone needs to have if a project like A Day in Bilkent is going to be implemented. Since every group member has their own tasks to do, being on the same page is very difficult especially when the number of the contributors are increasing. Except that soft skill, doing proper analysis and design is extremely important before implementing the project because a good analysis helps doing a great design and a good design makes implementing much more easier and makes the developers more efficient. More requirements are satisfied with the help of analysis and design and it also helps developers in a way that problem could be approached in a more accurate way; boundaries become more clear.

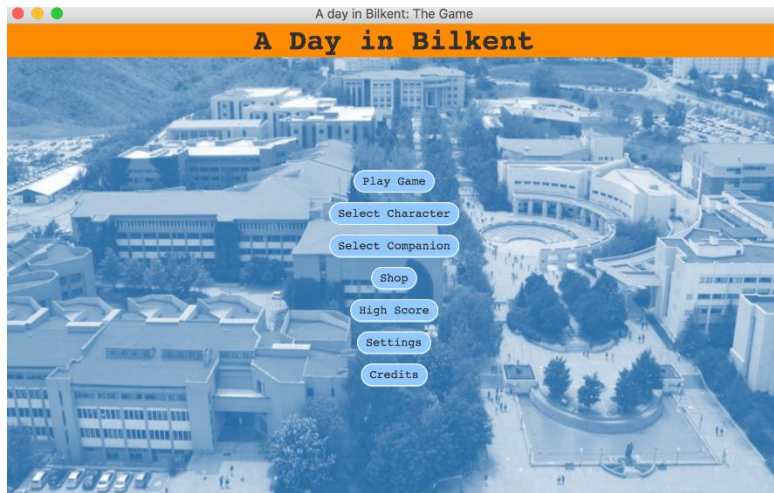
### 4.1 SYSTEM REQUIREMENTS & INSTALLATION

In order to be able to play A Day in Bilkent, every user must have at least

- Java 8
- Intel Graphics
- 4gb Ram
- 500 MB free disk space

The game could be played from an executable file.

## 4.2 HOW TO USE



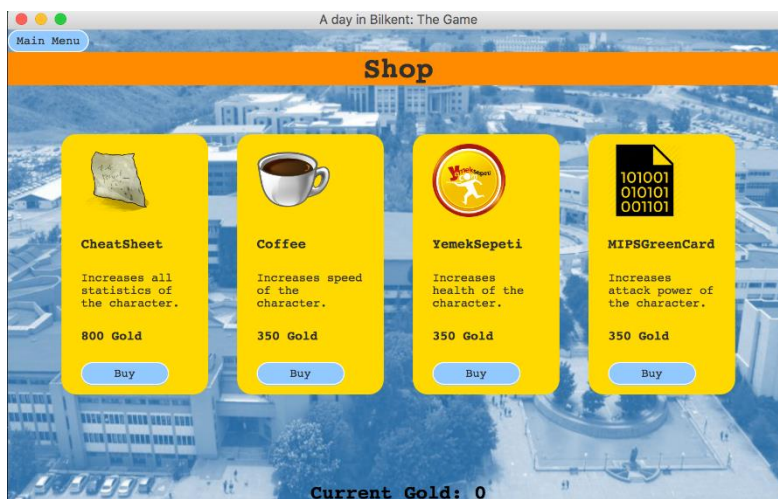
### Menu



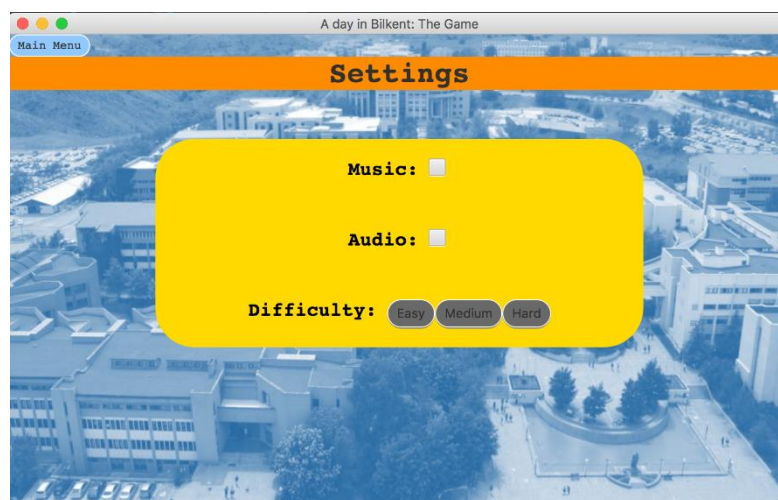
### Character Select



Companion Select



Shop



Settings