EECS2030: LAB 4 Due: Feb 20th , 2021- 11:59 pm

The purpose of this lab is to ensure that you practice

1- creating a correct aggregation and composition relationship between objects.

2- implementing static factory methods.

1. Setup

Please download `Lab4.zip` that is attached to this description.

- Open `eclipse`.
- Click on *`File`* and select *`Import`*.
- Choose *`Existing Projects into Workspace`* and click *`Next`*.
- Click on *`Select Archive File`* and then *`Browse`*. Find `Lab4.zip` and click *`Finish`*.
- Please make sure that you do not already have a project called `EECS2030_Lab4`, otherwise eclipse cannot import it for you.

You should see two files, one is called `Game.java` and one `GameTest.java`.

2. JavaDoc generation

The javaDoc has been written for you. All you need to do is to generate it as an HTML file to make it easier for navigation. For this, right click on `Game.java ->` select `export -> javaDoc -> Next`. It will ask you for the location in which you want to store the documentation. Enter the path and then click `Finish`.

If you look at the location in which you stored the documentation, you'll see there is a file called index.html. Clicking on this file, shows the documentation of the project in your browser.

3. Programming Task

One of the most popular multiplayer video games in 2020 was "among us". In this lab you are going to implement (a small part of) the backbone of this game, which is the objects and their relationships.

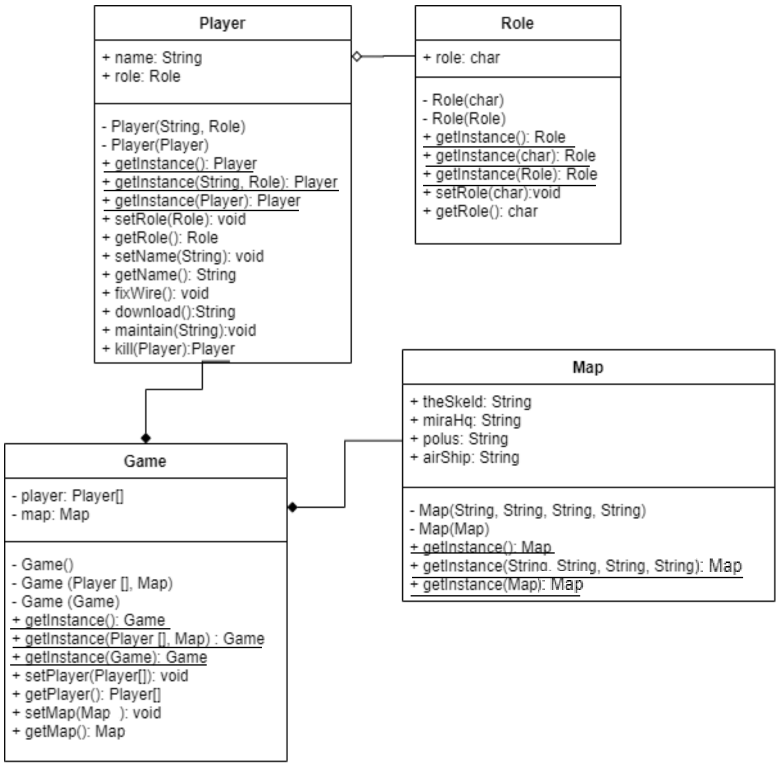
Four to ten people can play this game, and each takes a role. A player can either be an imposter or a crewmate. The goal of the crewmates is to complete a set of tasks, while they identify the imposters and eliminate them. The goal of the imposters is to covertly kill the crewmate before they complete their tasks. Through a voting system, a player may be voted as an imposter and fired from the game. Crewmates will be the winner either if they complete all the tasks or eliminate all the imposters. Imposters will be the winner if the number of the crewmates in the game is the same as the imposters.

The game has a science fiction theme with four areas in which players play the game. These areas are called "The Skeld", "MIRA HQ", "Polus" and "Airship".

For this lab, you are going to implement the constructors, static factory methods, setter and getter methods for the classes that are shown in the UML below. It is important that the relationships (i.e. aggregations and composition) are implemented correctly. Please note that there are some methods in this UML that we are not asking you to implement. They are only there to make the objects more understandable and the relationship clearer.

Please make sure that you read the UML carefully for private/public features and aggregation/composition relationships.

Static features are underlined in UML.



Task 1: Implementing Role class

For this task, you are required to implement the overloaded and copy constructor in addition to the static factory methods and setter and getter methods. The full documentation of class and the methods can be found in the starter code.

Task 2: Implementing Map class

For this class you only need to implement two constructors; the overloaded and copy constructor in addition to the static factory methods.

Task 3: Implementing Player class

Two types of the constructors (overloaded and copy) and all the setter and getter methods is what we ask you to implement for this task. Also, you should implement the static factory methods.

Task 4: Implementing Game class

Three types of the constructors (default, overloaded and copy) and all the setter and getter methods plus the static factory methods is what we ask you to implement for this task.

Please note that the instance variables in this class are all private, as the UML shows.

4. Submit

You only submit one file that is called `Game.java` via eClass by clicking on the lab link.

You do not need to submit het tester or HTML files.