Q2: (40 minutes, 09.40-10.20)

1. Instruction

1) Create a new Java Project named "2110215_Final_Q2".
2) Copy folder "**application**" and "**data**" (in "toStudentQ2") into your project **src** folder.
3) Copy folder "**res**" (in "toStudentQ2") into your project and make it a resource folder.
4) Fix the code (detail shown below) inside.
5) To submit:
   - go to project folder that you actually do the coding for this question.
   - Zip this question's src and res folder into one file. Make sure you get all the .java files. Name the zipped file YOUR-ID_Q2.zip (for example, 6332112121_Q2.zip)
   - Submit the zipped file as an assignment on MyCourseville.

2. Problem Statement: **Octopus Between Us**

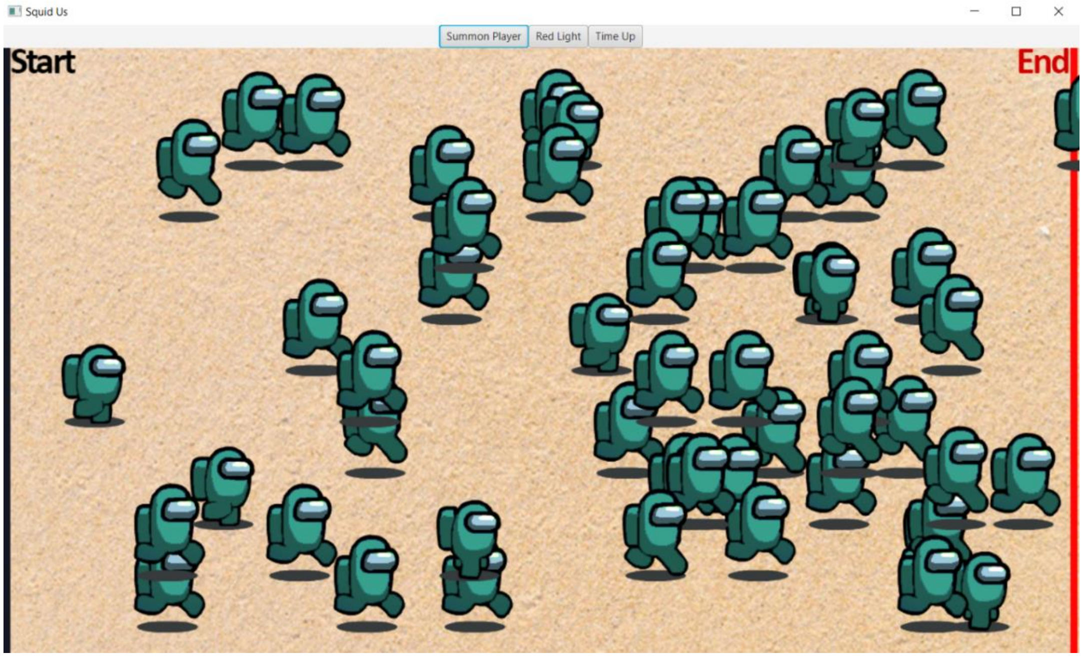
Based on a popular Korean online series that we can't mention by name without getting taken down by internet police, we recreate one of the game that appears in the show with character from a popular game that we also can't mention by name.

In this program, we are going to simulate this game that allows multiple running players to appear on the scene at the same time without slow down. **A video file is given to you to show how a finished application should look like.**

main.java is a javafx application that shows the room where players can run, stop, die, and disappear from existence.

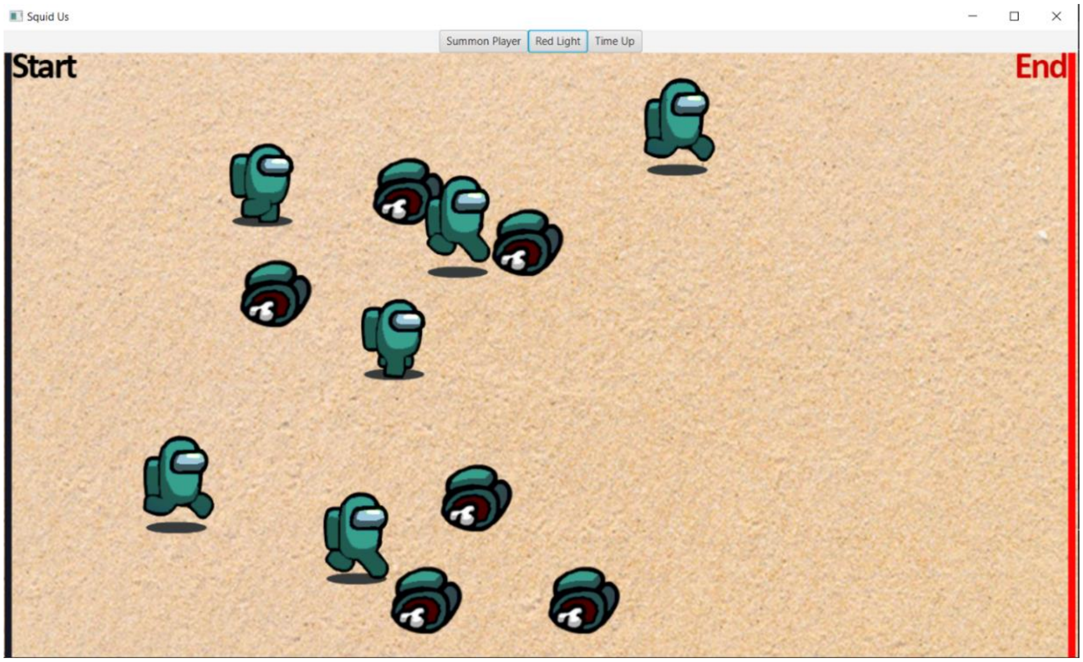
Click on "Summon Player" button will create a player that will run on the field (pane) below from starting line on the left, and pass through finish line on the right.

Player will cycle between running and stopping in place, and each player will have a random movement speed, run time, and stop time.



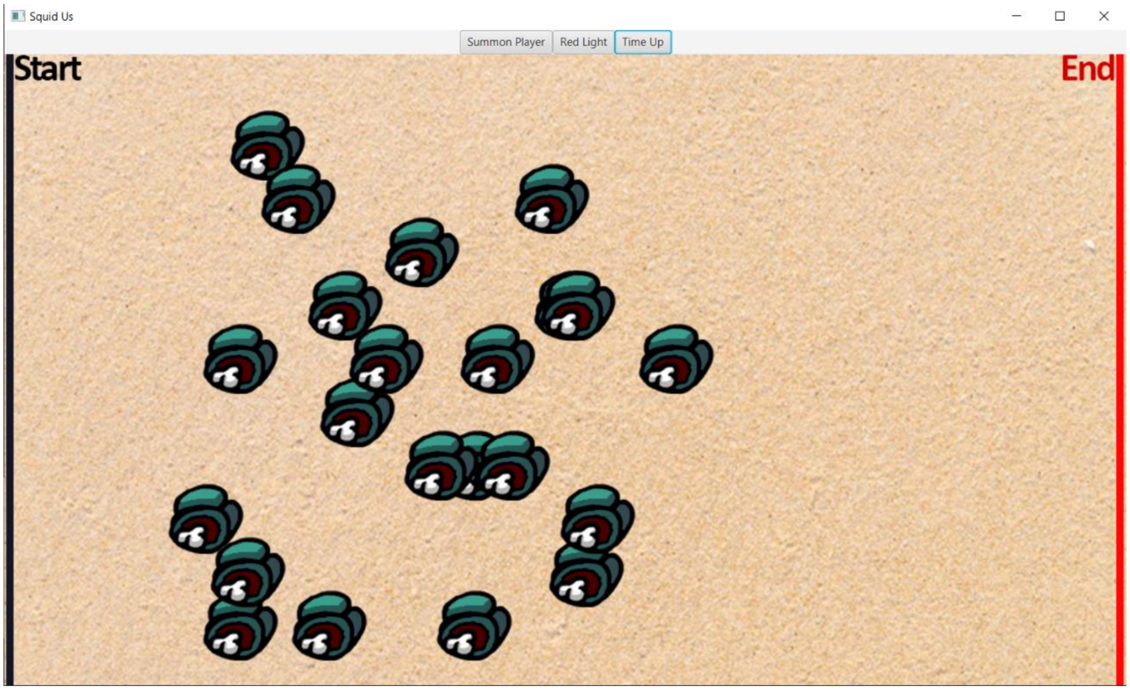
Click “Summon Player” screenshot.

Click on “Red Light” button will kill every player that is currently moving. Each player’s body will remain for a bit and then disappear one by one.



Click “Red Light” screenshot.

Click on “Time Up” button will kill every player (whether they’re moving or not) and like “Red Light” button, the body will remain for a bit and then disappear one by one.



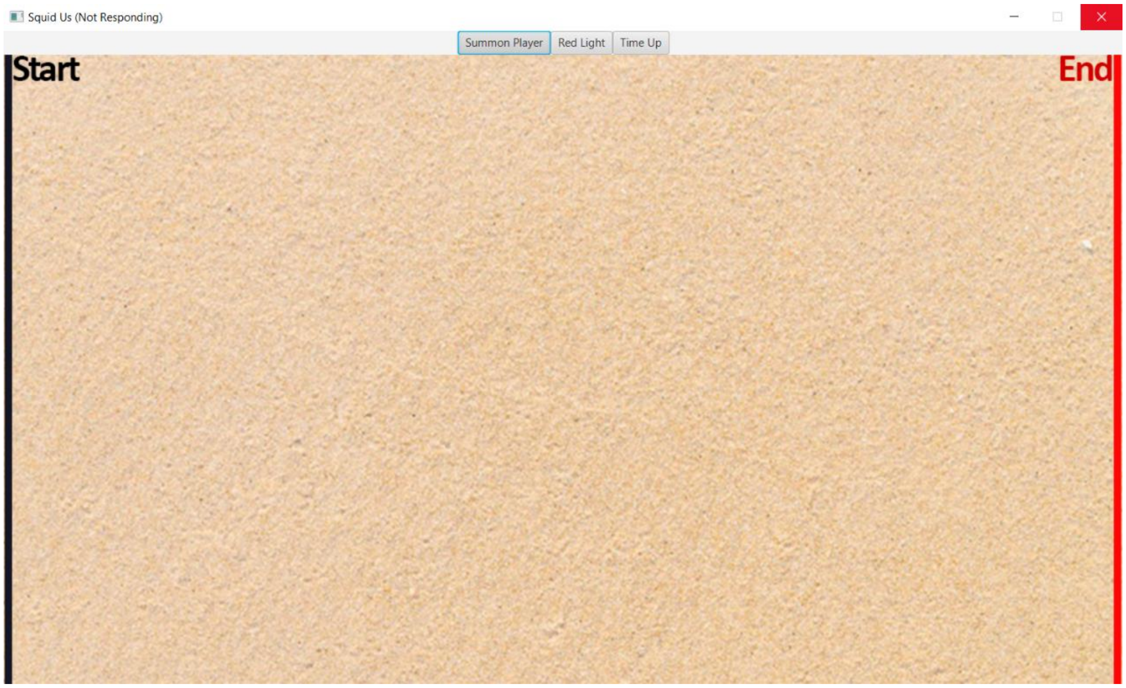
Click “Time Up” screenshot.

Note: Certain bug may cause an “java.lang.IndexOutOfBoundsException: Index -1....” exception error when summoning multiples players and the application will crash.

```
    Exception in thread "JavaFX Application Thread" java.lang.IndexOutOfBoundsException: Index -
1 out of bounds for length 2
```

This error will not affect your score as it requires Thread command that is outside of this class to fix. You only need an application to run with at least 20 players, with “Red Light” and “Time Up” button working as intended.

Without writing any code, try out main.java and click on “Summon Player” button. You will see that the application will not respond at all. This is because the application tries to draw a player running animation in a loop. But without thread implemented, other functions like button and pane don’t get a chance to run.



Non-respond screenshot.

Your task is to use threads to allow all players' animation to be shown on the pane. **For a player, it always has the same speed no matter how many other things are on the pane.** The application might be slower with high number of players, we will only look at 1-20 players cases.

3. Implementation Detail

The code is given in **ThreadMain.java**. You do not need to know the details of the user interface.

There are a total of **2 methods you must implement in ThreadMain.java.**

1. **initalizeNewPlayer**

2. **updatePlayerMovement**

Hint: Player in Main.java use ImageView, which is related to JavaFX**.**

- You can add your own methods and variables.
- You **MUST NOT change the line "Thread.sleep(50);"** If you do, you get **0 point**.
- If no threads are used to solve this question, you get **0 point**.

Main.java, Alarm.java, **Player.java**, **PlayerField.java**, and **SpriteAnimation.java** classes are given, **You MUST NOT change anything in These 5 classes.** If you do, you get **0 point**.

4. Scoring Criteria: **(5 points)**

4.1. Click "Summon Player" button once - create a player and show it moving on the pane while the application is still responding **(2.5 points).**

4.2. After creating more players, the players don’t slow down and not causing any exception aside from the mentioned “java.lang.IndexOutOfBoundsException” (from 2 to 20 players) **(2.5 points).**