Assignment 4 - Part B: Spring Boot

- **This part of the assignment is to be done individually.** Do not show other students your code, do not copy code found online, and do not post questions about the assignment online. Do not use solutions from previous semesters, courses, or offerings. Please direct all questions to the instructor or TA.
- See the marking guide for marking details.
- Assignment must be done in **IntelliJ;** submission generated **using Zipper addon**

1. REST API for Assignment 3's Game

In this part of the assignment you will be creating a Spring Boot REST API for your assignment 3!

The course website provides a fully implemented web client front end that you can add to your project. Create a new Spring Boot project and extract the provided web client files into a `/public` folder in the root of your project.

When you run your project, Spring Boot will automatically serve up files in the `/public` folder to the web browser. So you should be able to browse to it at http://localhost:8080

The course website provides a number of resources to help you develop a Spring Boot server implementing the necessary REST API to support the web client:

- REST API document detailing the end points and data
- Data classes that the web client expects to be sent to it. These are data transport objects (DTO): nearly empty classes that just expose data. You will need to implement the static factory methods in each to fully populate an instance of a class before sending it to the web client. You should not modify any of the field names or types in the DTOs because this may cause problems for the web client.
- Video of full game being played, plus some tips on debugging.
- POSTMAN file with many of the HTTP requests configured.
- Full source code to the web client.

You should not need to modify the web client; however, you can if you desire. It should function in the same way it does in the video.

You may use either your assignment 3 solution (either from your individual work, or if you worked with a partner), or the posted sample solution. No change in marks either way. If your assignment 3 had bugs that prevented it from working correctly, then you should likely use the sample solution.

Tips:

- Create a Spring Boot project and copy in your model from assignment 3 (or sample solution).
- Create a `/public` folder for the web client; copy in the provided web files.
- Create a new Java package to store the provided API DTO classes.

- Have your controller class store a list of games (i.e., instances of your model). Have your REST API look up games in this list whenever needed.
- You may assume that the server processes one HTTP request at a time, but must support multiple games going on at once (i.e., in different browser tabs…)
- You may need to edit the model to support any new fetaures this version of the game requires.
- Don't use the `MakeSpringPrettyPrintJSON.java` file: it messes up serving the *`public/`* folder (causes a "Whitelabel Error Page" to show); just delete the file if present.

2. Deliverables for Part B

Submit a ZIP file to CourSys: https://coursys.sfu.ca

See directions on course website.

Please remember that all submissions will automatically be compared for unexpected similarities.

