Jungle Game - Development Manual

Setup

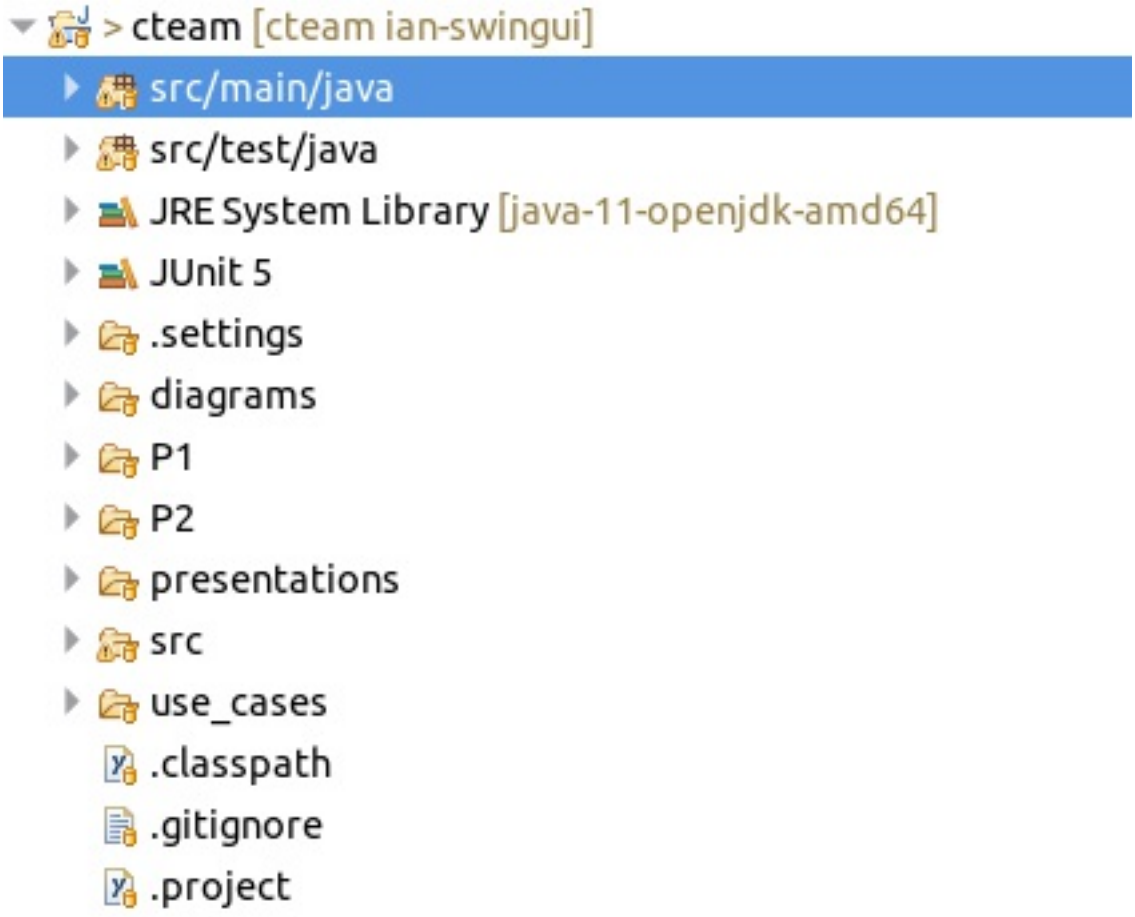
Before developing on this project, make sure you have installed the latest versions of:

- Eclipse for Java Enterprise Edition
- Java SDK 11
- JUnit 5

In Eclipse, open the Git Repositories View by going to Window -> Show View -> Other… -> Git -> Git Repositories.

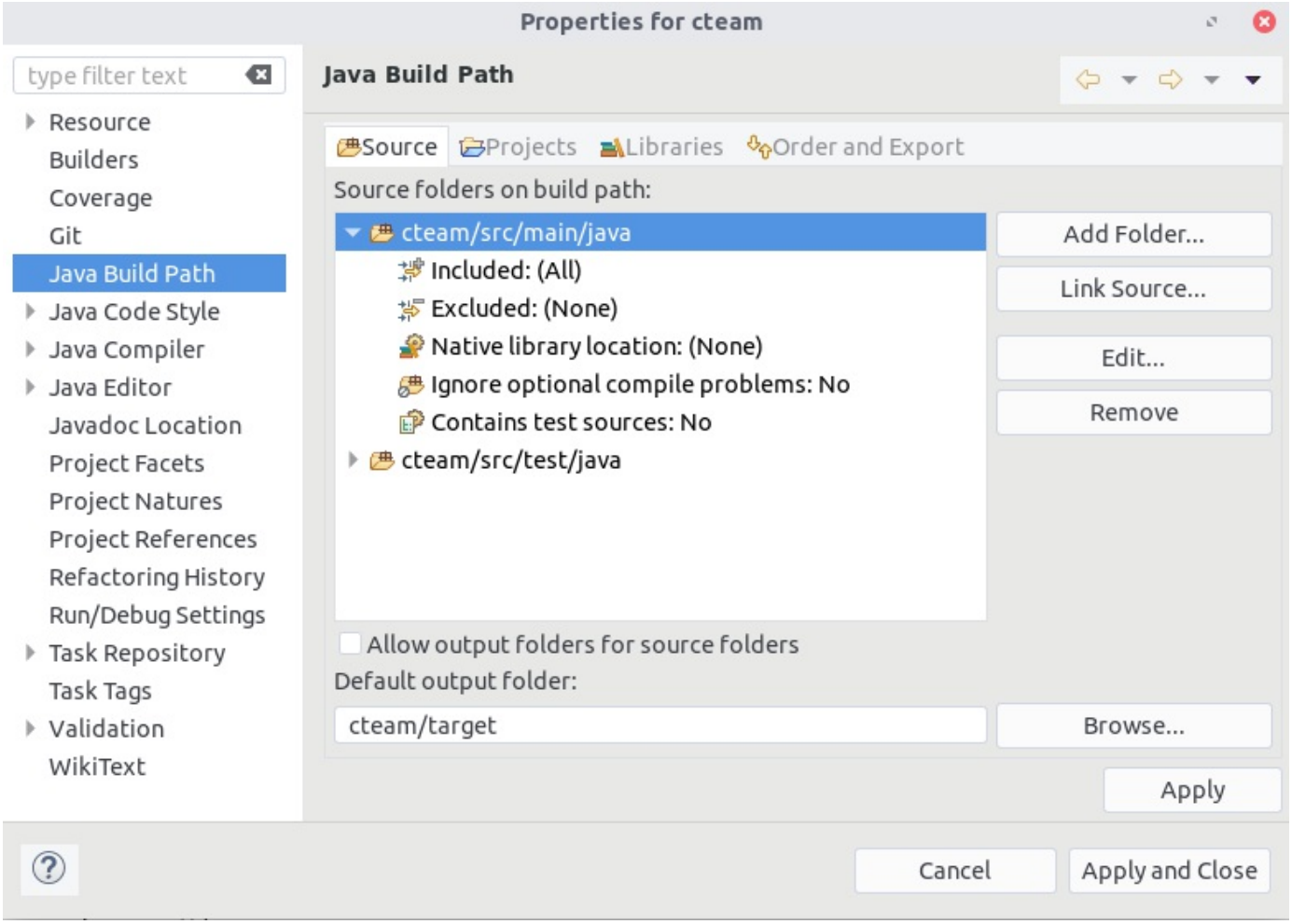
Press the “Clone a New Repository” Button  and select GitHub.

Clone the repository through its URL https://github.com/cs414-C-Team/cs414-f18-001-C-Team.git or by searching “cs414-f18-001-C-Team”. Choose a directory to store the project and Finish.



In the Package Explorer, ensure that Eclipse has placed the `main` and `test` packages on the project's build path. To do this, right click on any directory and select Build Path -> Configure Build Path.

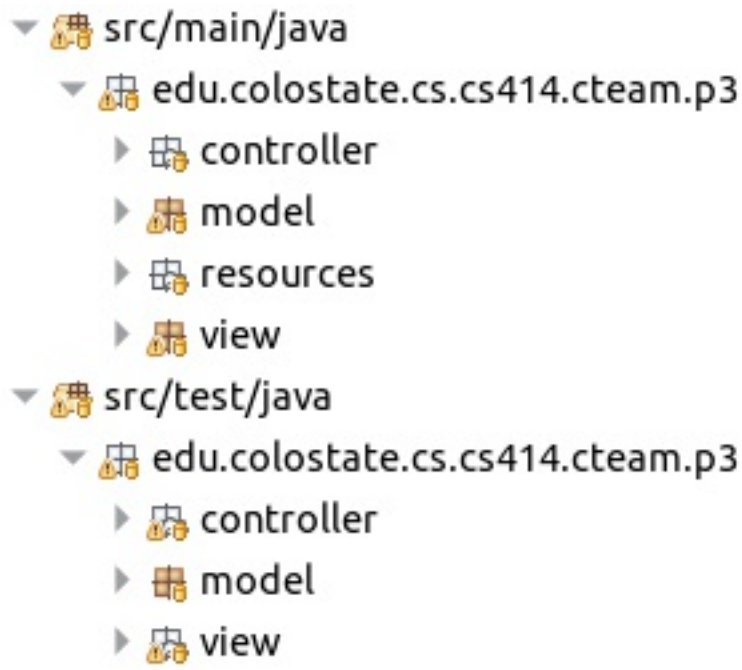
Also ensure that the my-sql-connector-java-5.1.47-bin.jar file is include the project libraries.



Add folders `cteam/src/main/java` and `cteam/test/main/java` to the build path if they aren't included already. Under "Libraries", also ensure that java-11-openjdk is selected as the project's JRE.

Packages

Inside the `src` directory are our `main` and `test` folders. In main is the actual code for the game, while test contains test cases for the main code. In each, the package structure for our project is `edu.colostate.cs.cs414.cteam.p3`. The project has 3 packages `model`, `view`, and `controller`. Model contains code representing data, View contains code for displaying the data to the user, and Controller contains code that links the system and the user.



Notice there are two modules in the project - a Client and a Server. Repeat the package steps above for both modules.

Running the Game

There are two elements to the game: a central server that handles database connections, and a serve that handles all client side events and match logic. These need to be run separately; the best way is to use Maven to package the client, and Eclipse to run the server.

Starting the Server:

- Find the ServerRunner class, and run this as a Java Application.
- Note the IP address logged by the console.

Packaging the Client:

- In Eclipse, right click the Client module and select Run As

maven build…

- In the targets option of the Maven build view, write “package”.
- This will generate a client-1.0.jar file in the target folder of the repo.

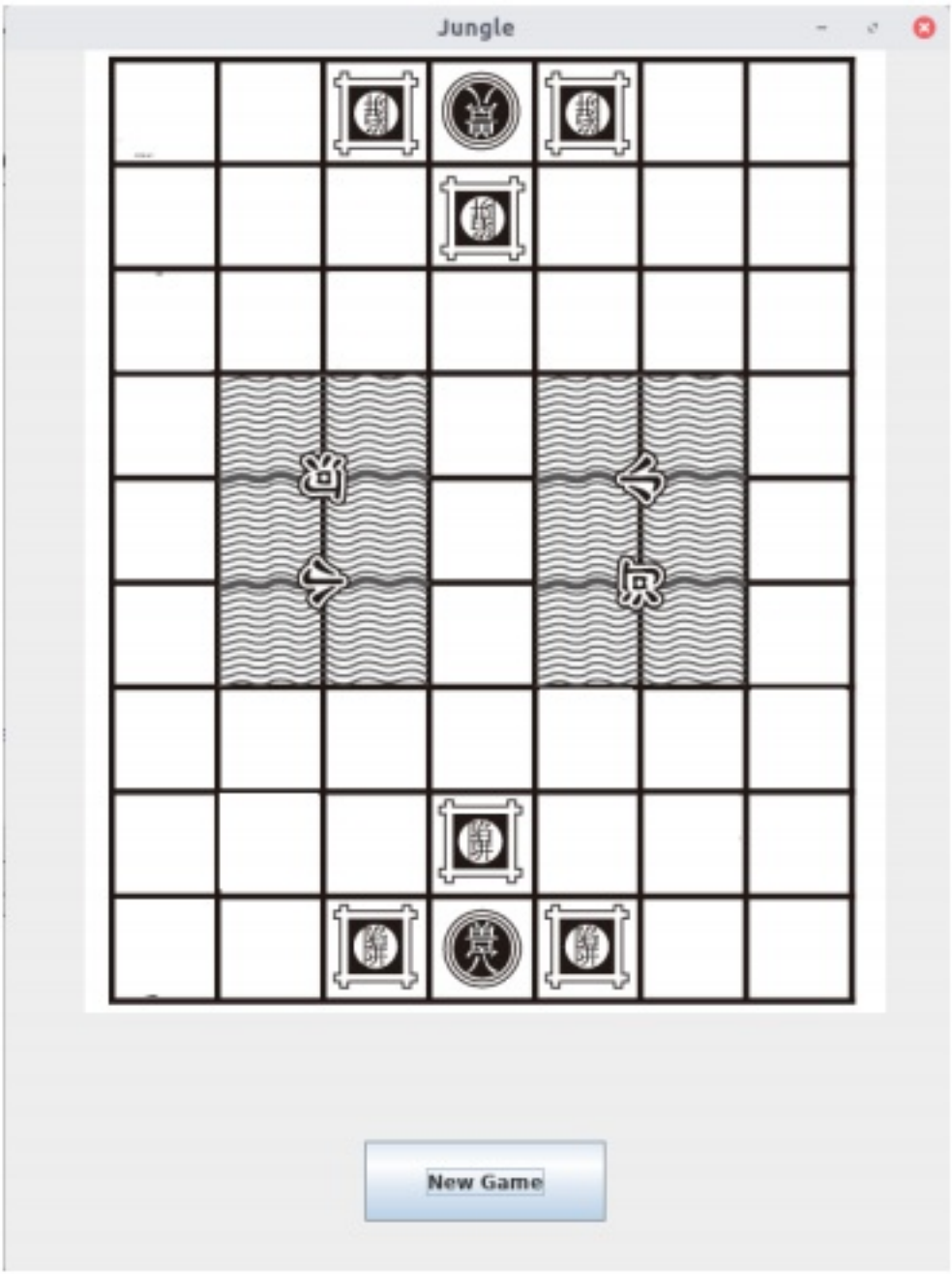
Running the Client:

- Run the client-1.0.jar file from the previous step
- If you want to see the logs of the jar file, run it on the command line
- The jar opens JFrame window to log in and specify the IP address. Enter the IP address the server output
- Login/register an account
- You can also use the “test” user or “local” user (both without password) to login. Test is a normal user, while local does not require a server connection and can be used to play local games

Using the Interface:

- After you’ve logged in, you can send an invite to play a game with a player. Click search to view the list of players in the bottom box, select a player, and click the Send Invitation button.
- The player will recieve an invite. You will be notified if they accept the invite by a new match appearing in the top box.
- If you recieve an invite, it will appear in the top box. Upon selecting, you can accept or deny the invite.
- If you accept the invite, a game will be created between you and the sender. The game will be displayed in the top box. where the invite was located.

- The sender’s turn is first. If you were the sender, selecting the game will give the the option to click the Launch Game button. This will open up a match screen with the current state of the board, where you can make a move by selecting one of your pieces and a spot it can move to. The game will move the piece if it is valid.
- After making a move, click Submit Turn to finalize your turn and send the current match state to your opponent.
- If it is not your turn, you will be presented with the option to view the board, but you cannot make any moves.



Testing

To run all current tests for all packages, simply run the test suite `AllTests` in `test.java.edu.colostate.cs.cs414.cteam.p3` in Eclipse. Access the test cases for various classes in the `model`, `view`, and `controller` packages.