CSCI 1440/2440 Lab Installation/Setup Guide

1 Introduction

CSCI 1440/2440 labs consist of you, the student, coding autonomous agents to compete in various games and auctions, against either separate autonomous agents or your classmates.

Below is a guide to installing and running the **Java** labs in CSCI 1440/2440, and we will be using **Eclipse** during labs in order to run the competitions.

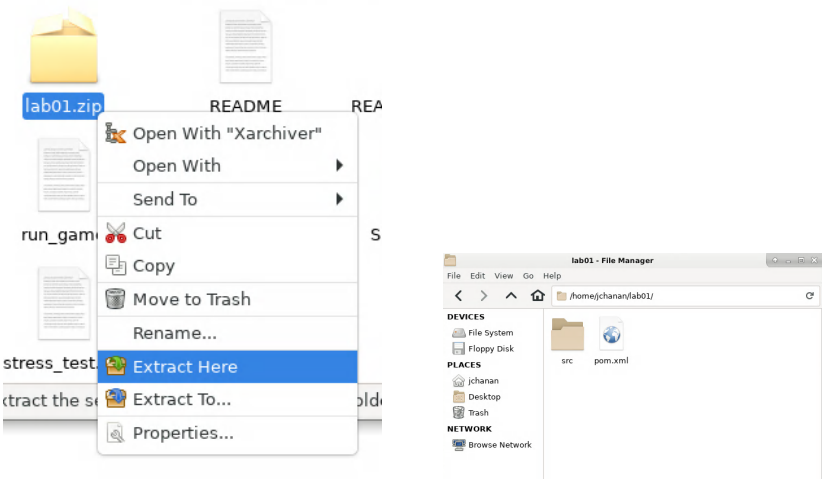
Each lab will contain one or both of the following components: **Simulations** and **Competitions**. In simulations, your agent will execute your strategy against an offline agent implemented by the TAs. Competitions will put your strategy to the test, pitting your agent against those of your classmates. While simulations can be run locally, competitions are run synchronously via a central server on the CS department machines, and therefore your lab code will need to reside there when we run the competitions.

During lab, you will have the option to work either on a department computer directly or on your own personal machine running a remote department machine. If you choose to do the latter, we recommend that you install FastX, a program which allows you to work on the department machines from your own computer. If that is the case, then click here for CS Department instructions on how to download, install, and launch FastX 3. We recommend you doing so before the first lab, and feel free to post questions on Ed if you are having problems with this.

2 Lab Setup Guide

Below are the steps to run the labs:

1. Visit the course website (specifically, the Labs page) and download the lab stencil code, which will be a ZIP file.
2. Extract the contents of the ZIP file. The extracted folder should have a file called `pom.xml` at the top-level. Remember the location of this file.



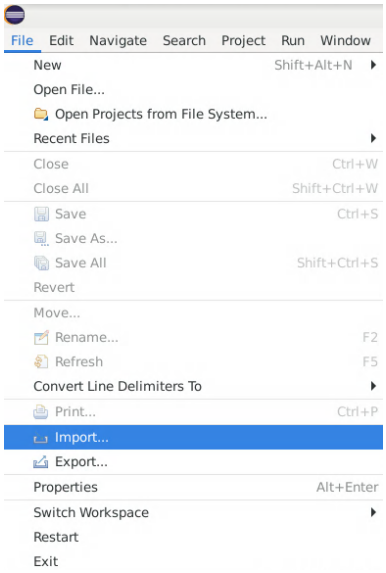
3. Open **Eclipse** by opening a terminal in FastX and entering the command:

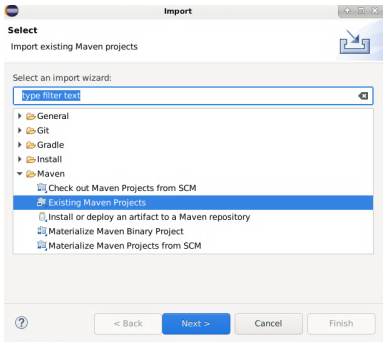
```
eclipse -vm /usr/lib/jvm/java-17-openjdk-amd64/bin/ &
```

If this is your first time using Eclipse, do not worry about the settings – using the default workspace is perfectly fine.

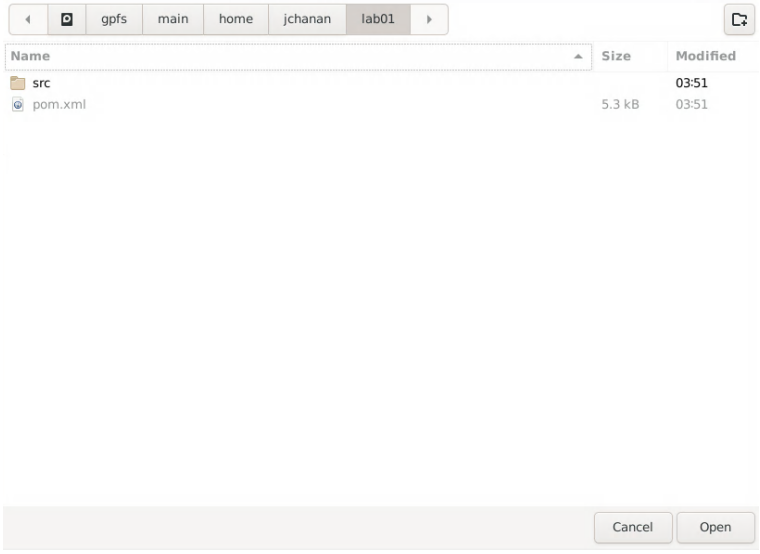
4. Import the Lab code.

   (a) In Eclipse, navigate to **File** → **Import**, and then select **Maven** → **Existing Maven Projects**.

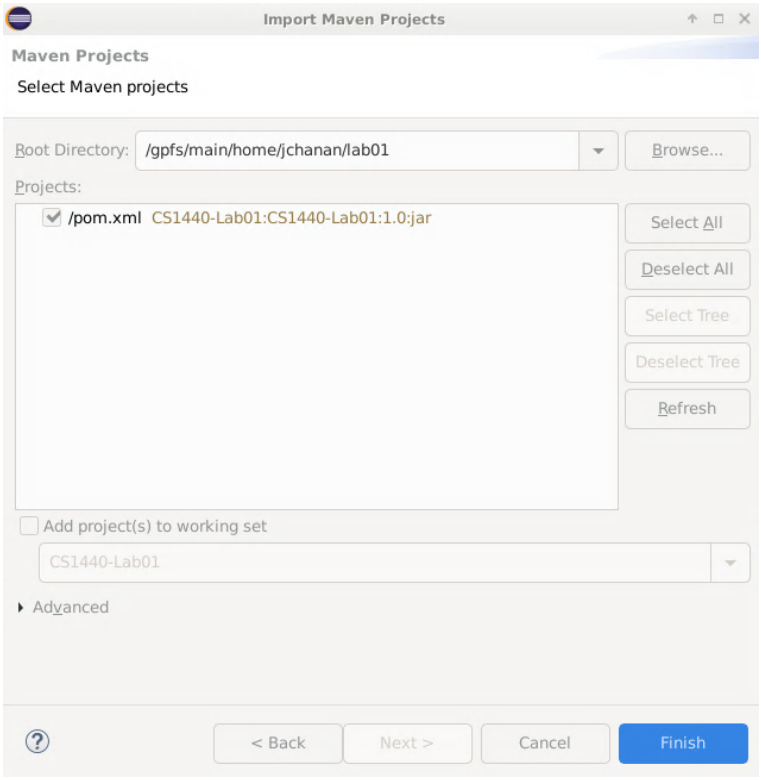




   (b) **Browse** for the root directory, and choose the lab stencil code directory containing `pom.xml`.



   (c) Ensure that the checkbox next to `pom.xml` is checked in the Projects section, and then click **Finish**. This will import the lab code, along with all dependencies, into a new Eclipse project.



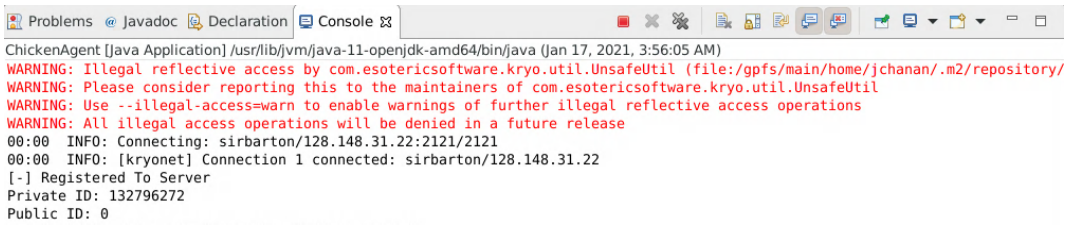
5. Work on the lab – fill in the stencil code as directed on the lab handout, and don't hesitate to ask the TAs if you have any questions! During the lab, you will come across **simulations** and **competitions**, both of which are detailed below.

   - A **simulation** is a locally-run version of a game or auction in which you test out a strategy against a bot written by the TAs. When the handout indicates that a certain file is meant to be a simulation (i.e. "Fill in the code in `AuctionAgent.java` and run the simulation"), all you need to do is fill in the code and use Eclipse's **Run** button to launch the program from that file. You should get a steady stream of output in the console – once the simulation ends, the program will output a score for you and your opponent(s) so you can see how you did. Feel free to iterate on your strategy and repeat these as many times as you want!
   - A **competition** is a game or auction run by the TAs on a central server, in which your agent will compete against those of your classmates. Competitions will be run live during section.

     If the handout indicates that a file is for the competition (i.e. "Fill in the code in `GameAgent.java` for the classwide competition"), just fill in the code with your strategy and follow the handin instructions in the following steps to submit your code. You will need to **name** your agent (for the scoreboard) by filling in the `NAME` variable.

     The TA will explain it in depth, but just know that you will need to name your agent, input the correct host, and then use Eclipse's **Run** button to launch your agent right after the TA has launched the server.

     Below is an example output from a successful launch of competition agent. The warnings are completely fine – as long as you get the **Registered to Server** message, it worked. This should not be an issue if you filled out the correct host, named your agent, and ran your program at the correct time.



6. Once you have finished the lab (so, you have run all simulations and written your agent for the competition), the TAs will check you off and mark you as present for the lab. Please do not leave until you are sure the TAs have checked you off.

3 Other Tips

- Running out of space on the department machines? Try the `ncdu` command.
- If you are working locally and want to make file transferring more efficient, try looking into SFTP alternatives or writing a script. If you think of a good idea, definitely share it with the class and the TAs, so we can add it to this guide.
- Any suggestions or improvements to this guide are welcome.