Lab 1

Due: Tuesday 01/23/2024 @ 11:59pm EST

The purpose of labs is to practice the concepts that we learn in class. To that end you will be writing java code that uses a game engine called Sepia to develop agents that solve specific problems.

Task 0: Setup

You will hear it in lecture, but I encourage you to avoid using an IDE in this class, and instead compile and run java code from your terminal/command line.

Task 1: Installing java 8 on your machine

In this class we will be using java 8. There are a few ways to install it depending on what your operating system is. I have included several other pdfs in the `java8-install` directory. Plese choose the one appropriate for your OS and complete those instructions to ensure java8 is installed on your machine.

The tl;dr of this is that we need two commands to work: `javac` and `java`. `javac` is the java **compiler** while `java` is the java `runtime`.

After the Install

After following whichever instructions you needed to, please pop open a terminal (on windows called a command prompt: do **not** use powershell). You will need to execute the following command:
`$: java -version`
You should see output in your terminal/command prompt that looks like the following:

```
andrew@blue:~/labs/lab1$ java -version
openjdk version "1.8.0_382"
OpenJDK Runtime Environment (build 1.8.0_382-8u382-ga-1~22.04.1-b05)
OpenJDK 64-Bit Server VM (build 25.382-b05, mixed mode)
```

If the version is "java 1.8" then you're ready to move on.

Now we need to check that the java compiler is working (the java compiler is called `javac`). Please execute the followig command: `$: javac -version`
You should see output in your terminal/command prompt that looks like the following:

```
andrew@blue:~/labs/lab1$ javac -version
javac 1.8.0_382
```

The version of javac should match the version of java and they should both be java 17!

Task 1: Downloading Sepia

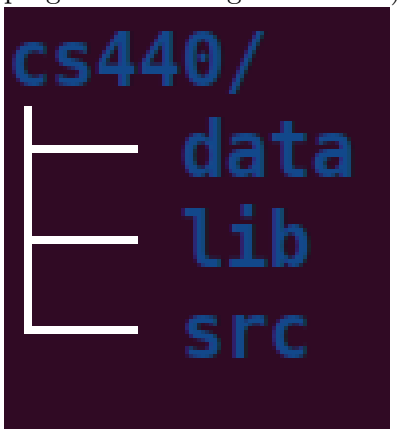
On Piazza, under the "Resources" tab there should be a section called "General Resources". In this section is a file called `Sepia.jar`. Please download this file.

Task 2: Creating the Lab/HW directory structure

To make life easier, we are going to create and maintain a directory structure for our labs and homeworks. Please trust me, this will make everything much easier to manage.
Please create a directory called `cs440`. In this directory you should create a few sub-directories:

- directory `lib/`. Please copy `Sepia.jar` that you downloaded earlier into this directory.
- directory `data/`
- directory `src/`

When you are done, your directory structure should look something like this (I am using the linux program `tree` to generate this):



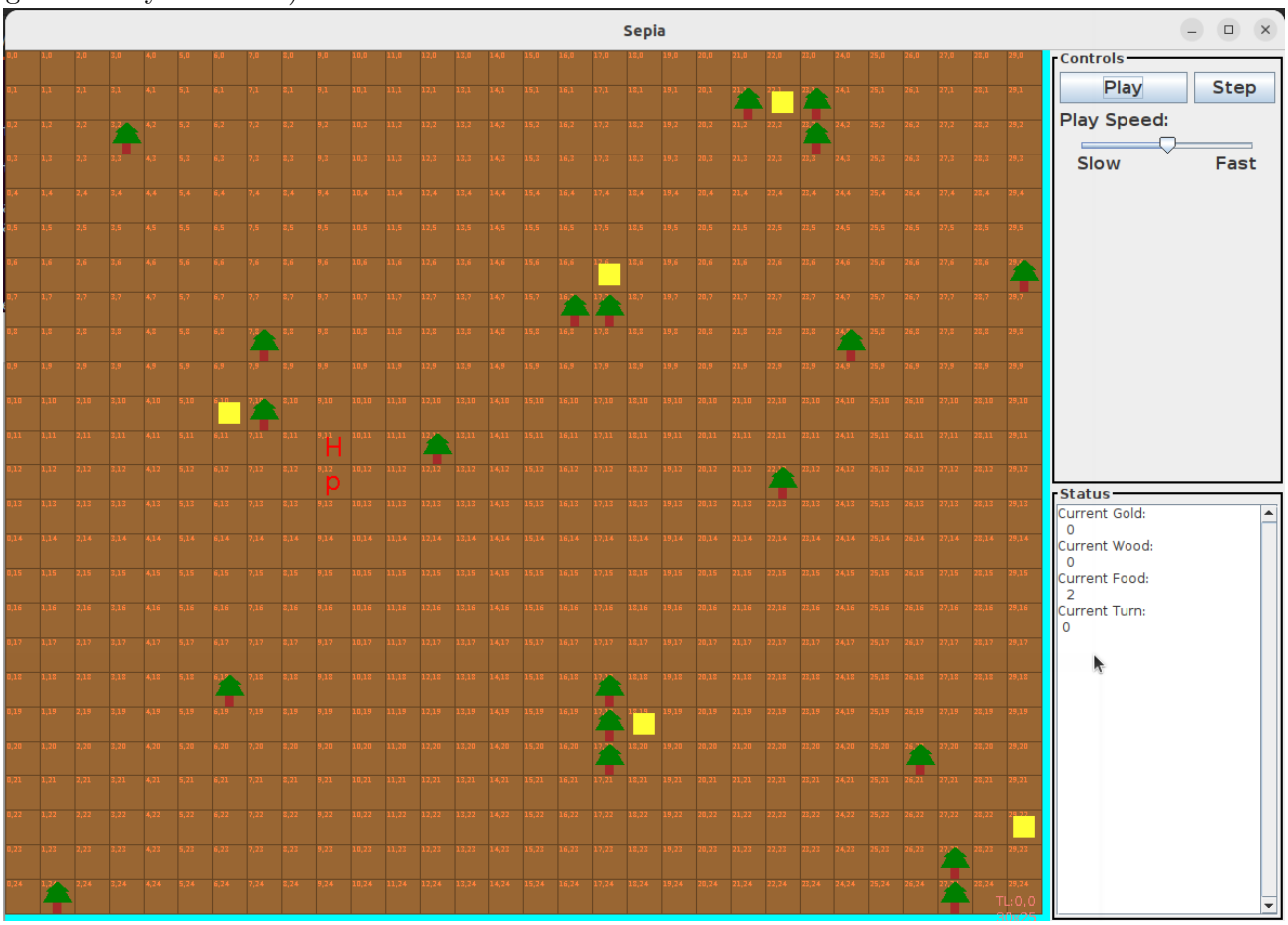
Sepia games are configuration files represented using the xml language. Traditionally there are two types of xml files, a *map* configuration file (which specifies how big the map is, what units are on what teams and their initial locations, any resources such as gold/trees, etc.). The second type is a *game* file which links to the map file for the game but also specifies which java agents control which team, should the game be rendered, etc. Don't worry, I will provide these for you, but if you wish feel free to create games of your own! All of these config files traditionally go in the `data` directory that you just created (inside their own sub-folders to keep everything nice and organized).
All source files will go under `src`. For example, code you write in lab 2 will go in `src/lab2/`, and code you write in pa1 will go in `src/pa1/`, etc.
Finally all .jar files will go in `lib/`. In the future, I will give you .jars that I write, and when I do I want you to put those in `lib/` as well.

Task 3: Testing your Sepia install

Now go ahead and download from Piazza's Resources tab (under "General Resources") a .zip file called `example_map.zip`. Please extract the contents (three .xml files) into the `data` directory.
To test whether or not your code is working, you need to compile and run your code. Since there is no code to write for this lab, we need to run the already built code (Sepia has some default agents inside it). To run java from the terminal/command line, we need to invoke the `java` command followed by setting the class path (with the `-cp` argument so java is aware of any jarfiles we wish it to use). Finally, we need to call the main class (i.e. the class to run) followed by any arguments that the main class needs. We will always be running Sepia's main class which is `edu.cwru.sepia.Main2`, and this class needs to be given the path to the config .xml file that specifies the game, players, etc. So, to test our code, we need to run the following command: `$: java -cp lib/Sepia.jar edu.cwru.sepia.Main2 data/ResourceCollectionConfig.xml`
which can be seen in the following screenshot:

```
andrew@blue:~/labs$ java -cp lib/Sepia.jar edu.cwru.sepia.Main2 data/ResourceCollectionConfig.xml
```

What should pop up is the following gui (you may have to resize it / it may not pop into the foreground on your screen):



Move the slidy bar for "Play Speed" to the "Fast" setting and hit the "Play button". You should see the current turn increase (quite quickly) and the red "p" on the screen should move! If this happens, your install is working.

Please now complete `your-first-agent.pdf`