

Project 0
Computer Science 4013: Artificial Intelligence
Due 12:01AM Feb 1, 2011 (Note: NOT the beginning of class)

Introduction

This semester, you will be using a variety of simulators for your projects. All of the simulators use the same framework and java packages, making it easy to go from one to the other. For this project, you will focus on the pacman simulator. The goal of this project is for you to have the code checked out to your machine and to submit a valid (but random) agent to the competition ladder.

Project details

Each of the projects (except project 0) will have a class-wide competition ladder. The ladder will generally start 2 weeks before the project is due. The first place player will receive one extra point for each night that that player wins the ladder up to a maximum of 5 points. The second place player will receive 1/2 extra point for each night that the player is in second place. No player can receive more than 5 extra points from the ladder and points will be distributed down the ladder accordingly should the maximum be reached.

For this project, you must turn in an agent that runs in the ladder. There will be NO extra credit for the ladder so performance does not matter so long as your agent does not crash.

Project 0

This project has two main components. Although this project is only worth a few points, it is absolutely required for testing the ladder. For this assignment, you need to do the following:

1. Download and install the aiprojects code.

I strongly recommend that you use an IDE for these projects! You can pick your favorite IDE but the instructions below are for eclipse, which is a very good IDE available for free for any platform. You can download eclipse from:

<http://www.eclipse.org/>

svn should be installed on the cs machines but if you want to install it on your own machine, visit the subversion homepage at:

<http://subversion.tigris.org/>

subclipse (the svn plug-in for eclipse) can be found at:

<http://subclipse.tigris.org/>

To install subclipse in eclipse, go to "Help → Install New Software" in eclipse and then use the latest subclipse URL for eclipse available in the "Download and Install" tab above.

You should download the latest code from our subversion repository (URL below). The advantage of the subversion repository is that you will have access to bug fixes. For all projects, both finding and fixing bugs will get you extra credit.

You can check the code out using eclipse and subclipse or you can check it out at the command line. eclipse instructions are given below. If you are using linux or mac os x, you should type the following. Be sure to replace the "4x4" with YOUR actual 4x4. If you are using windows, you can use tortoise svn to check out the code. Be sure you are running in administrator mode as tortoise gives odd errors for non-administrative users.

```
svn checkout svn+ssh://4x4@csnsvn.cs.ou.edu/home/svn/idea/aiprojects
```

Your username is your 4x4 and your default password is cs#4x4. **Change your default password!**

Using subclipse, to check your code out into eclipse, use the following instructions. Note that subclipse can be odd sometimes. If it refuses to work, use the command line instructions above.

- Chose File → New → Other
- Expand SVN and choose "Checkout Projects from SVN"

- Add the aiprojects SVN URL
`svn+ssh://4x4@csnsvn.cs.ou.edu/home/svn/idea/aiprojects`
Be sure to replace 4x4 with your username!
 - Checkout the directory and all its subdirectories (it prompts which directory to check out; don't just check out 'src' without 'configs' and everything else).
 - Choose to check out as a project configured by using the New Project Wizard (and the head revision)
 - In the new project wizard, choose Java project
 - Give the project a name, and select to create separate folders for sources and class files.
 - On the Source folders tab, select the folder 'src' as the source folder rather than the root folder of the project.
 - On the libraries tab, press the "Add JARs" button and add all of the jars in the lib folder to the build path.
2. You can test your download of the code in two ways. First, you can go to the directory where you downloaded the code and type:

```
ant hello
```

If this does not work, you do not have java or ant installed properly. Second, you can also run it from eclipse by clicking on the build.xml file in eclipse and choosing the second 'run as ant build' option. Then choose 'hello'.

3. Create a controllable agent (e.g. something that controls an agent) that makes legal moves. For this project, it is perfectly acceptable to reuse the sample random agent. The point of this project is to ensure you can get the mechanics of your player working, not that you have made the world's smartest player. Note that although the ladder will run once for this project, it will NOT count for extra credit. You can find the sample random agent at:

```
edu.ou.pacman.client.RandomClient
```

Your agent needs to run in its own package. If you copy the random agent, be sure to change the package header to be:

```
package your4x4;
```

All of your code must run in a single package based on your 4x4. To compile and run your code with the pacman code create a directory under “pacman/src” with your 4x4 such as “pacman/src/name1234”. Place your java source files in this directory.

To make the simulation know about your new agent, you also need a `pacmaninit.xml` file. There is a sample `pacmaninit.xml` file in the `examples.hello` subdirectory. You will need to change this file to point to your new agent at `name1234.MyAgent` (or whatever you call your agent class). If you name it something other than `pacmaninit.xml`, you will also need to edit the `worldconfig.xml` in `examples.hello` to point to your new file.

Now you can execute the program by typing `ant hello` or by using the IDE (recommended).

Note that this method of compilation requires the “ant” program, which can be obtained at:

```
http://ant.apache.org/
```

It should also be installed on most CS machines.

4. Ensure that your player runs on linux on the CSN class machines. You can ssh into these machines or go into the lab personally to verify this. Java should be in your path by default. If it is not, java lives in:

```
/opt/java/bin/java
```

5. Submit your project on `codd.cs.ou.edu` using the submit script as described below.
 - (a) Log into `codd.cs.ou.edu` using the account that was created for you for this class. Your username is your 4x4 and your default password is `cs#4x4`. **Change your default password!**
 - (b) Make sure your working directory contains all the files you want to turn in. For example, my directory contained the following files:

```
AmyRandomAgent.java amypacmaninit.xml
```

- (c) Submit your file using the following command:

```
/opt/ai4013/bin/submit CS4013 Project0 AmyRandomAgent.java amypacmaninit.xml
```

- (d) After the project deadline, the above command will not accept submissions. If you want to turn in your project late, use:

```
/opt/ai4013/bin/submit CS4013 Project0Late AmysRandomAgent.java amypacmaninit.x
```

At 12:02AM on Feb 1, I will run a sample ladder. This ladder will not count for extra-credit points as the focus of this part of the assignment is to make sure that your accounts are setup correctly and that the ladder is able to run each of your players.

Point distribution

Since the point of this assignment is to ensure that your accounts are setup correctly and that your program runs without crashing, the rubric is pretty simple.

- 5 points for an agent that correctly runs in the ladder
- 0 points for anyone who has NOT changed their password from the default