COSC1204/2048 - Agent-oriented Programming and Design
Warming up for AOPD'12!

Deadline: Monday 23th July

Semester 2, 2012

1. Get and read the *project description* document. This document should give you an idea of what the project is about and what will be expected during the term, particularly for Assignments 1 and 3. You should also read and familiarize with the *Scenario Description* and the *Communication Protocol Specification* in order to fully understand the domain we will be working on. All this can be downloaded from the course web page http://goo.gl/xRqzS, under the *Course Material* section.

2. First, let us setup the Gold Mining game server from the Multi-Agent Contest:

   2.1. Get the *game server* (file `game-server-AOPD-GoldMining.zip`) and unpack it somewhere (e.g., your linux box or `yallara`). Remember that while your agent system may run in any platform, the game server only runs in a Unix-based system.

   2.2. Configure the game server connection and rmi ports using script `setmyports.sh`. If you run it in your local machine, then you want to use the default values (game connection port `12300`), but if you will run it in `yallara` you need to make sure you are using unique ports that nobody else is using.

   2.3. Check the server sub-directory `output/`, it should be empty. This is where the server will put all images corresponding to each game played (each game in its own subdirectory).

   2.4. Have a quick look at file `conf/serverconfig.xml`. This is the configuration file for the game server, which sates what kind of games will be played and who are the teams (including their connection credentials). For example, you can modify this file to alter the length of a game (i.e., how many steps) or the size of the grid. You will notice that in that configuration file a match compromises just one simulatation game (called "Park") of 150 steps.

   2.5. Next, change the Park game to just 10 steps.

   2.6. Start the game server using the `startServer.sh` script. The server will just wait for agent connections. However, here we will not connect any team.

   2.7. Start the game by hitting the `ENTER` in the game server.

   2.8. The game should start. The server will wait up to 4 seconds for teams to send their actions. Because no team is connected, you will notice the server makes a step every 4 seconds.

   2.9. After 10 iterations, the game finishes. You should be able to see the game pictures inside `output/` as svg files (can be opened with Firefox, for example). You will observe that because no team was connected, all agents are still, nobody has moved.

   2.10. You can terminate the game server doing Control-C.

3. Next, we will set up your JACK mining agent for development:

   3.1. Get the JACK *core agent system* (file `JACKMINER-AOPD12.zip`) and import it as an ECLIPSE project:

      - Use a regular project, not a Java one, as this is not Java per se and otherwise you will get lots of errors.
      - In ECLIPSE, do: File ⇒ Import ⇒ General ⇒ Existing Projects into Workspace ⇒ Select Archive File, and pick the zip file you just obtained.
      - Read files `FILES.TXT` and `README.TXT` in the core agent distribution.
      - You need to obtain and copy `jack.jar` into `lib/`. This is the library providing the JACK compile and runtime engine. Remember your agent is not Java code, but JACK code.
      - You can install file `build-jack.xml` as an Ant script to be able to compile the JACK code within ECLIPSE itself if you want (you can always compile from the shell).

   3.2. Compile your agent system by running `make all` or the `compile` script on the shell, or running the Ant script from ECLIPSE.

      - You should now see the JACK compilation messsages in the console and, when finshed, all `.class` files for your agent application within subdirectory `bin/`.

   3.3. Setup the corresponding host and port number in the agent main script `run-team.sh` that matches the host and game connection port where your game server will be running. If the serer is running locally in your machine and you did not do any changes in the server, you will have `localhost` at port `12300`.

4. Start the game server using the `startServer.sh` script. Again, the server will just wait for agent connections.

5. Run the agent system by running script `run-team.sh`. Your six agents should successfully connect to the game server and you should see that in the console.

6. Finally, start the game by hitting the `ENTER` in the game server.

7. The game should start and now your agent system should show a GUI screen displaying the behavior of your agent team. Your agents should be sending their actions to the game server.

8. When the game finishes, you should be able to see the game pictures in the `output/` subdirectory of the game server, reflective the movements of your team. You will notice that the other team is still stuck, because no other team was connected. You can use the demo agent (available on the course web page) to connect a second team.

9. Finally, you will prepare your ECLIPSE environment to be able to *develop* further the agent platform given. To that end you need to install the PDT plugin for ECLIPSe. See FAQ section on the course web page or Section 2 (“Mini How-To”) in the *JACK core agent system manual*.

If you get stuck or have problems, please report this and participate actively in the course forum (see *“Discussion Forum”* Section in the web page). By now you should have posting access to such Google Groups using your RMIT account.

It is *very important* that you accomplish all the above steps **during week 1 and no later than the start of week 2, that is, Monday July 23**. Starting from week 2, you are supposed to be actively working on assignment 1, which assumes you have successfully completed the warm-up task.