COMP34120 — AI and Games

Project 1

Overview

Your task is to make a program (or ``bot'') which plays the game Kalah. In particular, your agent will play the 7,7 version of Kalah; the board contains 7 wells per side and each well starts with 7 seeds. Your bot will be entered in a tournament against all the other bots developed by your fellow students. Your goal is to make a bot which plays the game well (e.g. will beat other players), but bots will also be judged on speed of decision-making and the quality of the ideas used.

Schedule

- **Week of November 18 (Week 9):** Tell us what you plan to do, and how you plan to divide up the tasks among members of the group. Fill in the proforma on Blackboard and discuss with us during the week. This is done verbally and via a Proforma which you submit.
- **Week of December 9 (Week 12):** Group presentations telling us what you have done.
- **6pm, Friday, December 13:** Deadline to submit your bot.

Scheduled meeting times are Tuesday 12-1 and Wednesday 9-10 in Collab, but you are only required to attend in week 9 long enough to tell us your plan, and week 12 for the presentations. (We will schedule other presentation slots closer to the time.) Other booked times are for help sessions and to give you a place to work with your group.

The Game

Description of the game including the rules can be found here. We are using standard rules http://wikimanqala.org/wiki/Kalah, except we are also using the pie rule, which is a way to remove any first player advantage which might exist.

Resources

The game engine is found in /opt/info/courses/COMP34120/Project1_2019 and is called `ManKalah.jar`. The usage is:

```
java -jar ManKalah.jar <agent1> <agent2>
```

where <agent1> and <agent2> are programs which play the game. These could both be your bot, for example, or one could be yours and one could be one of your friends. We provide you with several agents, one is called `MKRefAgent.jar`. The agents are entered as strings, enclosed in quotes.

The game-playing agents interact with the game engine via a protocol which is described informally in `/opt/info/courses/COMP34120/Project1_2019/protocol/Protocol-info.txt`, and in Backus-Naur form in `/opt/info/courses/COMP34120/Project1_2019/doc/protocol.txt`.

One way to test your knowledge of the protocol is to play the game against the reference agent. You could do this as follows:

1. Open a window (e.g. using *xterm*) and type in this window
   ```
   nc -l localhost 12345
   ```
2. Start the game engine
   ```
   java -jar ManKalah.jar "nc localhost 12345" "java -jar MKRefAgent.jar"
   ```

You can play in the xterm against the provided agent (who you will probably be able to beat) and you will have to use the protocol commands to communicate with the game engine. Likewise, you can play two humans against each other by providing different ports (type *nc -l localhost 12345* in one window and *nc -l localhost 12346* in the other), or play *MKRefAgent* against itself in the obvious way.

The game engine contains a time-out mechanism which will end the game when one agent is taking too long (to avoid infinite loops). When this happens, the other agent is declared winner. The time-out period is one hour in total for each player. The game engine also ends the game when one agent attempts an illegal move or sends an illegal message.

For java programmers, we provide some relevant classes in `/opt/info/courses/COMP34120/Project1_2019/src` and the javadoc in in `/opt/info/courses/COMP34120/Project1_2019/doc`. However, you are free to do this in any language which you choose.

The Journal

Each group has been set up with a Journal on Blackboard. **You must use this to document your ideas and work.** We will use this as part of the assessment, to help gauge the quality of the approach and determine the extent of each group member's contribution to their project. For code sharing, you can use GitLab or whatever you choose.

Getting help

The best way to ask questions is to use the forum on Blackboard associated with this course. This way, both the questions and answers can be shared with all the students.

We will also be available to give help during the scheduled lab times, but it would be helpful to tell us in advanced what kind of help you need (game theory, algorithms, programming). You can do this by making a request on the forum.

Assessment

We remind you of the assessment we announced during the first lecture. More details are in this document, "Assessment for semester 1 project"

1. This project counts 30% of your final mark.
2. 15% comes from the content of the approach. How good was the idea, how well-informed by the content of the course, outside literature, etc.
3. 15% comes from the performance of the approach - how well does it play and how quickly does it play.
4. The group mark will be distributed to the members of the group using: self-assessment, contribution to the Journal and demonstration of knowledge during the presentation.

The presentation

During the last week of term, all groups will make a presentation to the lecturers. Other groups are free to attend. It is important that every member of the group speak and answer questions, so we can gauge the contributions of the members of the group. None of the marking will be based on presentation skills, so you need not make a polished presentation. The assessment is entirely on content: what your group did and what you learned from doing it. This and the information in the Journal is what constitues the approach mark.

(Note: if there is some "secret sauce" which you used in your agent, which you do not want other groups to hear, you may write it down and show it to the lecturers during the presentation. You must put it in your journal.)

Submission

Submission will be through Blackboard. You need to submit three files. 1) An executable file which can attach to the game engine. 2) A readme.txt file with instructions telling how to execute (e.g. what string to send to the game engine. 3) A zipped (or gzipped) archive of the source code.