COMP30024 Artificial Intelligence

Project Part B: Submission Instructions

Chris Leckie and Sarah Erfani
Modified by: Matt Farrugia
Last updated: 12 May 2017

SliderPlayer implementation

The core part of your submission will be a set of `.java` files, comprising all the code required to compile and run your game playing program. This should include the source code for your class implementing the SliderPlayer interface, and any additional Java classes your implementation makes use of. Additionally, you must provide the `.jar` files for any third-party libraries which your program requires, not including the AIMA textbook libraries or the standard Java library (we will provide these libraries).

Agent file

To enable us to run your program, you must tell us the name of your Java class implementing the SliderPlayer interface. This information must be in the format of an 'agent file'. This is simply a text file called `agent.txt` (case-sensitive) containing a single line with the **fully-qualified name** of your game playing agent (your program). A Java class's fully-qualified name is made from its package name and its class name together, connected by a dot ('`.`'). For example, if your player implementation class looks like this:

```
package aiproj.slider.farrugiam.agents;
// ...
public class AgentAlphaBeta implements SliderPlayer { /* ... */ }
```

Then your fully-qualified class name is aiproj.slider.farrugiam.agents.AgentAlphaBeta, and your `agent.txt` file should contain this name on a single line:

```
aiproj.slider.farrugiam.agents.AgentAlphaBeta
```

If your Java class is *not* part of a package (i.e. it is in the default package, and has no 'package' line at the top), then the fully qualified name is just the class name:

```
public class AgentFarrugiam implements SliderPlayer {/* ... */}
```

For the above class, `agent.txt` would simply contain the line '`AgentFarrugiam`'.

The name of your class and package is up to you, however as per the 'Naming' section of the Part B specification, it is important that the name is unique to your group. Therefore, we recommend that you use a package name that somehow incorporates either (or both) of your student usernames.

While you may have created multiple implementations of the SliderPlayer class, you may only submit *one* player for testing. Therefore you should choose your best implementation to put into your `agent.txt` file. However, you should still submit any alternative designs you have created (see 'Additional files' section for more on this point).

Comments file

You must write and submit a text file `comments.txt` (case-sensitive) describing your game playing program. In your comments file, you should:

- Briefly describe the structure of your solution, in terms of the major packages and classes you have created.
- Note any third party Java libraries or code you have used.
- Describe the approach taken by your game playing program for deciding on which moves to make in terms of
    - your search strategy,
    - your evaluation function, and
    - any creative techniques that you have applied, for example; machine learning, search strategy optimisations, game-specific heuristics, specialised data structures, other code optimisations, or any search algorithms not discussed in lectures.
- If you have applied Machine Learning, you should discuss the learning techniques and methodology you followed for training your agent.
- Add any additional comments you want to be considered by the markers, including about any other creative aspects of your solution.

Additional files

While working on your project, you may have built alternative player implementations, modified the provided Referee, created additional tools or programs to test your player or its strategy, or programs to create training data for machine learning, or programs for any other purpose not directly related to implementing the SliderPlayer interface.
All of these files are worth including in your submission, and should be included when you use the `submit` system as described below. If you have any large files or other special materials you would like to submit that would not be suitable for submission through the `submit` system, please contact Matt Farrugia (matt.farrugia@unimelb.edu.au) to discuss their submission (note: you should do so at least one day in advance of the due date to arrange this).

Submission process

The submission process is similar to Part A. One submission is required for each group. That is, one group member is responsible for submitting all of the necessary files that make up your group's submission. You should include both your name and login id and your partner's name and login id in a comment at the top of each source code file, and in your comments file.

You must submit all parts of the project using the submit command on one of the MSE Student Unix machines (e.g. `dimefox.eng.unimelb.edu.au`):

```
submit COMP30024 ProjB <list of all files>
```

Where `<list of all files>` is a list of files, separated by spaces, containing the names of all of your `.java` files, your `agent.txt` file, your `comments.txt` file, and any additional files as described in 'Additional files' section above.

Please note that you should **list all files in a single command**, as each time you `submit`, any previous files are overwritten. Also note that your files should all be in the currently directory; there is no need to submit them in a directory structure such as the one Java uses for storing class files according to their package name. If you have many additional files not needed for directly testing your program, you may like to submit them in an archive format (e.g. `.zip` or `.tar.gz`).

Your submission will then enter the queue to be verified. During verification, your project will be compiled, and assuming there are no errors, three games will be played:

1. A 5x5 game with your player playing against itself
2. A 6x6 game with your player as Horizontal against a player making random moves.
3. A 7x7 game with your player as Vertical against a player making only forward moves.

After waiting a short period of time*, you will be able to see the result of these games by running the verify command:

```
verify -t COMP30024 ProjB > verify.txt
```

Omit the '`-t`' option if you would like to see the files you submitted included in the verification output. Even with this option set, the output will be long as it contains the output for three full games. So, it is recommended to redirect this output to a file (here, `verify.txt`) and to view it afterwards for example by using the command `less verify.txt`, or by copying the file to your own machine and viewing it in an editor.

In most cases the output will show the result of each game. In some cases, a game result will be an unfinished board. This indicates that the time for this game has expired and the game was terminated early. This could be due to an infinite loop, or to one player taking too long to make a decision.

*The maximum time necessary to verify your project is about 1.5 minutes of CPU time, because the time limit constraints are enforced on the test games (and in the second two games, the provided players make very fast decisions). Of course, if your project doesn't compile or the system can't find your player because of a problem with your `agent.txt` file, then the verification will be very fast, because the games will not be played.

However, in either case, the time may be increased due to other groups using the `submit` system, as only one program can be verified at a time. So, you should test that your project compiles and plays a legal game **before you submit**, and avoid submitting more often than necessary. You should also **submit early**, as system load near the deadline may cause you not to be able to view your verification output from your submission in time to make changes.

Submission issues

If you experience problems with the submit system, you should contact Matt Farrugia by email (matt.farrugia@unimelb.edu.au). If it is close to the deadline (within one day), please also send a copy of the files you are trying to submit, in case there isn't time to sort any problems before the final deadline.

Teamwork reflection

If you would like to make any comments on your experience as part of your group in relation to the relative contribution of each group member, you may submit a document with these comments via the LMS before the end of week 11 (11.59pm, Sunday 21st May). Note that this reflection should be completed individually, and is optional (you may choose not to submit a reflection).