CS310: Advanced Data Structures and Algorithmns

Spring 2014 Programming Assignment 4

Due: Friday, April 11 2014 at midnight.
Question 1 should be submitted in class on Thursday April 10.

Goals

Introduction to the game project.

Questions

1. The Easy game. Start this problem by reading about games in games.pdf (given as handout and available for download from the course website). See the steps below to set up your environment and run games. Play Easy, and it will display the rules for this game. Play the game several times, to see how positions and moves are displayed. Draw the game tree for Easy on paper, showing all 3! = 6 leaves. Assume neither player gives up early. Label the nodes (each of which represents a possible position) with a label that contains all the relevant information for that position. Label the edges (each of which represents a move) with an appropriate label. Label each leaf with a 1, -1 or 0 indicating who has won the game.

   Keep a copy of this drawing to use for question 4.

2. Do the setup steps discussed at the end of this file.

3. Starting from TestGame.java, write a class DisplayTree1.java in package cs310 to display Easy's game tree. Write a recursive method "explore(Game game, int level)" that (unless the game is over) tries all moves from game position game (see the loop in TestGame.java) and calling explore on them with level + 1. For each game states found, print out the "toString" report on the game state and the level, one state per line. Include the first 12 lines in your homework paper, as well as DisplayTree1.java. Note that it is hard to see the game tree from this output. Many of the states involve early resignations by a player, because one of the allowed moves (the first on the move-list) is "quit", so there are many more leaves than you showed in your drawing.

4. Modify DisplayTree1 by numbering the different game states as you find them. To detect a new game state and reuse old game state numbers, build up a Map of Game to Integer as explore does its exploration, in DisplayTree2, also in package cs310. Note that Game is guaranteed to have good hashCode and equals. Use 0 for the root state number, and "G0" in the output line, 1 and "G1", 2 and "G2", etc. in the output, along with the level and the toString report. Note that it's still not easy to see the tree from this information. Include the first 12 lines of output in your homework paper, as well as DisplayTree2.java.

5. Correlate the full output of question 3 with your hand-drawn tree by adding the G0, G1, ... names to your hand-drawn tree and squeezing in the extra early-resignation nodes into your drawing (just little —o's). In your memo.txt file write: 1) how many different game states you found and 2) how they break down into not-over game states and game-over game states (should be 2-3 sentences...).

6. Deliver your DisplayTree sources back to cs310/pa4/games and make sure they work there too. Report on any problems.

Guidelines for getting started with games on our UNIX systems

This is needed for final delivery in any case.

Note that $cs310 means /data/htdocs/cs310 in our UNIX filesystem, as well as www.cs.umb.edu/cs310 on the web:

1. Make a pa4 subdirectory of your cs310 directory for this assignment.
2. Download the games.zip file from the course webpage and unzip it into your pa4 directory.
3. Try "java cs310.TestGame" after cd'ing to pa4/games/classes. You should see:

```
Using game Easy
at game start, pn = 1
at game start, game state:  [] [](1 next)
move:  0
to game state:  [] [](done)    (move 0 is quit)
move:  1
to game state:  [1] [](2 next)    (again from start state, move 1)
move:  2
to game state:  [2] [](2 next)    (again from start state, move 2)
move:  3
to game state:  [3] [](2 next)    ...
```

You can also run this by cd'ing to games and using ant run, using the provided ant build.xml.

Getting started with games on your Windows PC

All the files you need are available in the directory $cs310/games.zip.

1. Download and unpack games.zip to some convenient place we'll call the games directory.
2. Try "java cs310.TestGame" after cd'ing to the games/classes directory.
3. Set up a project in your IDE.

In your notes.txt also state whether you used late days.