CS 6601: Assignment 1

Due September 15 by 9:35 AM

Abstract

You will write game playing agents for a game called Isolation.

1 The Game

The rules of Isolation are simple. Two players take turns placing their own game piece on different squares of a 5 by 5 grid. After the first two moves (in which each player puts their piece on an unoccupied square), the pieces move like queens in chess: diagonally, horizontally and vertically up to the edge of the board. Each time a player moves their piece, the square that they were previously occupying is blocked and cannot be moved to or moved through, for the remainder of the game. In addition, the players may not move through one another or occupy the same square. The first player who is unable to move loses.

(Note: this game is almost identical to "Isola" except that it's on a 5 by 5 grid, the first two moves are unconstrained, and each turn the moving player blocks the square from which their piece has moved, rather than any free square of their choosing.)

2 Your Assignment

Your task is to create an AI that can play and win a game of Isolation. Your AI will be tested against several pre-baked AIs as well as your peers' AI systems. You will implement your AI in Python (from a provided iPython notebook), using our provided code as a starting point. See the attached iPython notebook for a full reference of useful methods you might need. In an attached .zip file, we provide:

- a class for representing the game state
- a function for printing the game board
- a function for generating legal game states
- a class for running unit tests
- a random AI (baseline test)

Your goal is to implement the following parts of the AI in the class CustomPlayer:

- evaluation function (OpenMoveEvalFn())
- the minimax algorithm (minimax())
- alpha-beta pruning (alphabeta())

Your agent will have a limited amount of time to act each turn (500 ms). Note that we only time each turn, but if your agent takes more than a few minutes at construction time, for example because you're loading the entire set of possible board states from memory, you will be penalized.

These are the bare minimum requirements for your AI, and the rest is up to you. You will be scored according to how well your AI performs against some baseline AIs that we provide (see "Grading"). If you want to improve over the base performance, here are a few suggestions:

- storing the evaluation scores for past moves
- modifying your evaluation function to account for "killer moves"
- ordering terminal nodes to maximize pruning

3 Grading

The grade you receive for the assignment will be determined as follows:

10%: you write an evaluation function that scores based on the maximum number of moves that the AI can make, and your evaluation function performs correctly on some sample boards we provide.

75%: your AI defeats a random player $\geq 50\%$ of the time, plus above.

85%: your AI defeats an agent using OpenMoveEvalFn that searches 4 levels deep $\geq 50\%$ of the time, plus all the above.

90%: your AI defeats an an agent using OpenMoveEvalFn that uses iterative deepening and alpha-beta pruning $\geq 50\%$ of the time, plus all the above.

95%: your AI defeats an AI that uses Thad's secret evaluation function, iterative deepening, and alpha-beta pruning (a.k.a. Thad 2.0) $\geq 50\%$ of the time, plus all the above.

100%: your AI defeats Thad 2.0 $\geq 90\%$ of the time, plus all the above.

4 Botfight!

In addition to the basic assignment, you will have the option to compete against your peers for the glory of being the 2015 AI champ. We'll set up a system in a round-robin style to pit your AI against others, and we'll be giving out prizes for the top players. May the odds be ever in your favor.

If you wish to compete in the tournament, simply include a plaintext file with a description of your agent, entitled 'AI.txt', and your CustomPlayer instance will be enlisted.

If you compete in the AI tournament and earn 1st, 2nd or 3rd place, you will receive a bonus:

- 1st place: a letter of recommendation from Thad, plus 3 bonus points on your final grade.
- 2nd place: 2 bonus points on your final grade.
- 3rd place: 1 bonus point on your final grade.

5 Due date

This assignment is due on T-Square Tuesday September 15th by the start of class (9:35 AM). The deliverables for the assignment are:

- A filled out version of the iPython notebook provided. (player_notebook.ipynb)
- A brief plaintext description of the inner workings of your agent, if you are competing in the tournament (AI.txt).

6 Testing

We have uploaded the grading tests to a server for you to test on. Navigate *here* and upload the Python version of player_notebook.ipynb (player_notebook.py) to have it tested for time and for performance against our AIs. Please let us know ASAP if you run into any unexpected errors while testing and we'll fix the server as soon as we're able.

7 Resources

Links for installing Python on your computer:

- OS X (http://docs.python-guide.org/en/latest/starting/install/osx/)
- Windows (http://docs.python-guide.org/en/latest/starting/install/win/)
- Linux (http://docs.python-guide.org/en/latest/starting/install/linux/)

Installing iPython notebook (we recommend using pip):

- http://ipython.org/install.html (get version 3 or above)

Tutorial for iPython:

- http://ipython.org/ipython-doc/stable/interactive/tutorial.html

TA's will hold office hours Monday, Tuesday, Thursday and Friday from 2:00 to 4:00 PM outside TSRB 241.