cmput 355 2020 assignment4

Boyuan Dong

November 20, 2020

1. Boyuan Dong.
2. Boyuan Dong.
3. I coded for the game Amazon and I also write the report.
4. Actually, the code ttt_classic.py and go_player.py in class are very helpful for me starting the code.
   This is my GitHub repo `https://github.com/boyuandong/Amazon`.
   This is a video demo for play the game of Amazon: `https://youtu.be/e_YT5Qyas5A`.
   This is a video demo for end of the game: `https://youtu.be/bXfwnQo6gmg`.
   (These videos are just for the demo, not a serious play, please forgive my unreasonable moves)
5. The Game of the Amazons is a board game played by two players. White moves first, and the players alternate moves thereafter. Each move consists of two parts:
   (a) First, move one of one's own amazon stone as a queen move in chess (there are eight directions).
   (b) Second, after moving, the amazon shoots an arrow from its landing position to another position, using another queenlike move.

   An arrow, like an amazon, cannot cross or enter a square where another arrow has landed or an amazon of either color stands.
   The last player to be able to make a move wins. Draws are impossible. Here is the link from Wikipedia: `https://en.wikipedia.org/wiki/Game_of_the_Amazons`
6. Summary:
   - My original goal is write a player for game of Amazons, and this player will work properly.
   - I think I achieve the goal.
   - I am very proud of that I did a lot of error checks for the input, help instructions will prompt for most of situations. Besides of that, it will also prompt all suggest legal moves when player make an illegal move.
   - I tried to use numpy to print a grid like go game or ttt game as we seen in class, but failed. Then I just simply use the python instead. That's quite disappointing as I spent a lot of time on this part. Also, the player can not change to opponent automatically.
     The player is not be able to undo moves so far, that's what I need to keep working on.
   - If I will continue to work on this project later : I'll try to let the player check the opponent, and change to the opponent automatically after each move.
     Also, I will find a better way get all legal moves, maybe make a class for Position like go_player.py did, making the code looks more clean.
     I probably will implement the undo method for my player as well.
     Then get a resolver for this game.
7. Compete Amazons player with go_player.py and ttt_classic.py:

Compete performance	Amazon	Go Player	TTT Classic Player	Summary
Use numpy	No	Yes	Yes	
Used classes	Amazon	Position	TransposType Transpos Cell Position	Maybe I will have a Position class to make code more clean if I'll continue update my player.
Be able to undo	No	Yes	Yes	Maybe next time I will stored the board states to make it be able to undo moves.
The number of input positions each time	2	1	1	Unlike Go and TTT, Amazon need to move the stones already exist, there are 8 stones in total, need to enter the position of the stone, and the destination position each time want to make a move.
Number of steps each move	2	1	1	Amazon include two steps each move: first make a move, second place an arrow.
Be able to genmove (finds value of all moves using alphabeta search)	No	No	Yes	ttt_classic is also a solver for searching a solution.
Be able to modify the gameboard size	Yes	Yes	No	Amazon is able to modify the board size, but the size could only be even number.
Time of showboard function (seconds)	8e-05	2e-04	9e-05	amazon player is the fastest to draw the board.Mainly because I did not use numpy.
Time of find all possible moves (seconds)	3.5e-05	1.5e-04	2e-05	Amazon use the while loops to find all possible queen moves in 8 directions. Instead get all legal moves, go_player use request-move method to check if the move is legal.
There are help instructions	Yes	Yes	Yes	Besides the help menu, I also add the suggest moves/coords when player make a illegal move.

8. I think it's a player good for players to try and play. I am happy since it works properly. But there are some other updates I can do to make it even greater.

Working Diary
I choose this project because it's an interesting 2-player board game. And it's a good game for me to practice what I learned in CMPUT355.

- **November 2, 2020 3 hours:**
  I search for the board games for this project, and finally picked Game of Amazons.

- **November 9, 2020 6 hours:**
  I go over the ttt_classic.py and go_player.py and try to use numpy, but failed.
  Then I started to simply write a player for Amazon by using python.

- **November 16, 2020 8 hours:**
  Keep working on Amazon, I finished and updated player_amazon.py.
  Test and play around with the player.
  Working on the assignment report, also make a video record for this game.