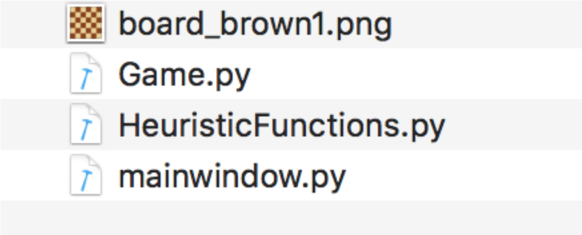
Name : Rohit Garla Net ID: rg3365
Section : CS-GY-6613

Artificial Intelligence
Checkers

INCLUDED FILES



board_brown1.png
Game.py
HeuristicFunctions.py
mainwindow.py

1) Board_brown : It is the .png file for the checkers board
2) Game.py : It is the main file which is to be executed.
3) Heuristic Functions.py : It is the files which is called by game.py, it consists of the heuristic functions which are used.
4) Mainwindow.py : It is used to create a dialog box for choosing whether to go first or second.

Types of pieces:

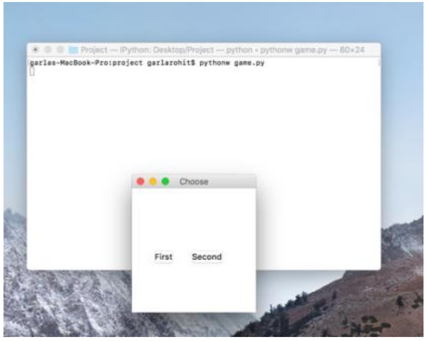
- Regular pieces : This pieces are the regular ones, which moves only upward-diagonal manner.
- King pieces : The regular pieces convert to king pieces upon reaching the last row from their respective position. These pieces can no longer move.

Minimum and Maximum Utility Values

- The minimum and maximum utility values are taken as 10000 and -10000
- The terminal states are defined based on who wins and who has more no of pieces, if black has more pieces then black wins, if white has more pieces then white wins. However If the no of pieces are same then its is declared as a draw.

INSTRUCTIONS

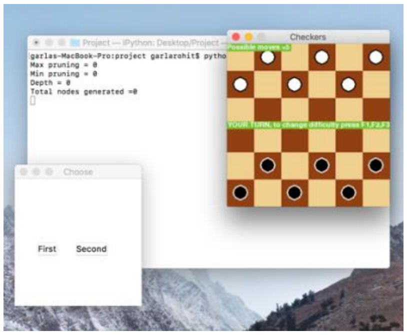
1) Run the Game.py file as shown(use python if running on osX),



```
garlas-MacBook-Pro:project garlarohit$ pythonw game.py
```

Project — IPython: Desktop/Project — python ◂ pythonw game.py — 80×24
Choose
First
Second

2) As seen in the picture the first dialog box appears, select first or Second. (NOTE THAT ON SELECTING THE OPTION THE BUTTON IS NOT HIGHLIGHTED BUT IT IS SELECTED)
3) After clicking on a button close the dialog box.
4) Now a new gaming box opens up. (Here the game is played)



```
garlas-MacBook-Pro:project garlarohit$ pytho
Max pruning = 0
Min pruning = 0
Depth = 0
Total nodes generated =0
```

Possible moves =5

YOUR TURN, to change difficulty press F1,F2,F3

5) At first the game begins with easy level difficulty, you can click on F1(easy),F2(medium),F3(Hard) anytime during the game to restart the game with higher difficulty.
6) If the computer or player is victorious the game begins again with the same settings (i.e if the user decided to go first, he will still go first in the second game with easy difficulty)
7) To go second the program must be exited and restarted.

HEURISTIC FUNCTIONS

In total I have used three heuristics: Simplescore, piecerank, edgeking

1) Simple_score : It calculates the score of the board and returns a value. More points are awarded to kings rather than normal pieces.
2) Piecerank : It calculates the rank of pieces based on the position(kings are not included here).
3) Edgeking : It awards more points if a pice is stopped at edge. Since it can no longer be captured.

Difficulty Implementation

The difficulty was implemented based on the cutoff limits of different node. Assuming the max player at root node is 0, my different levels are defined as

1) Easy mode : 2
2) Medium mode : 4
3) Hard mode : 8

Statistics

Max Pruning: It indicates the total no of times pruning occurred in max function.
Min Pruning: It indicates the total no of times pruning occurred in min function.
Depth: Indicates the total depth of the tree.
Total nodes Generated : It gives the total no of nodes generated.

Software Requirements

- Install python 3.x For a OSX machine.
- Install the 'pygame' and 'copy' modules for python. Please refer this link: https://docs.python.org/2/installing/index.html
- Place the image file '.png' in the same folder (project folder) as the python code files.