10/23 - everyone

Starting block: <https://github.com/tarun018/AI-for-Ultimate-Tic-Tac-Toe>

Changes:

* Extend time - low priority
* 1. Modify way to win (1 of 9 smaller boards)
  + ***Have user decide what game they want to play***
* 2. Change block status to current block - Done
* Replace 0-9 to 0-2?
* 3. Add heuristics to AI bot
  + ***Need to add actual min/max alpha-beta search*** 
    - Refer to player.py in 10/23
* Add interactive GUI - probs last
  + Nope…

To run code from terminal:

python simulator1.py <number: 1, 2, or 3>

Python Syntax

* In: Evaluates to true if it finds a variable in the specified sequence and false otherwise.
* Xrange: similar to normal range, just stores it in memory slightly differently

Questions

* Do you think Dr. Garvey is looking for good coding practices? -SP
  + Some of these variables I have no clue what they mean!
  + Lots of duplicated code in simulator
  + We will need to do more documentation for the code I’m guessing.

10/29 - Some minor comments done in code to help understand it -SP

10/30 - Decided to use the code from UTTT folder. Here is the original link

<https://github.com/rohitsakala/ultimateTicTacToe>

Added comments - SP and SM

JF - uploaded version now disallows choosing from a cluster of 3 boards to play on.

11/1 - Deciphered most of Player2’s evaluation function - added comments. Figured out what some other functions do as well. Check comments.

11/2-Steph and Shannon: Got next block in play working for the normal block status. Takes player’s old move and calculates where the next move should be, without calculating cat’s games (where you can’t play in a certain block). Took out corner logic for the blocks because it didn’t use it and put it into it’s own method so we could access it easier. We made it more readable for human players and pretty.

If you need to force a draw by human vs human, here are the coordinates.

(0,0)

(0,1)

(0,3)

(0,2)  
(0,6)

(1,0)

(3,0)

(1,1)

(4,5)

(3,6)

(1,2)

(4,7)

(3,3)

(2,2,)

(8,6,)

(6,0)

(2,1)

(6,3)

(2,0)

(8,2)

(6,6)

11/5: Jeremy and Steph. Beginning of the game print out a sentence saying you can choose anywhere and not show the current block status, which used to show block 0 only. Draw is implemented as well for the user. Game can be won by winning 1 block now. Not yet implemented to give the user an option of winning 1 block or 3 blocks in a row. Looked at how to incorporate actual AI into our program.

11/6: Shannon and Sang. Changed the last block displayed to be the winning block by adding a new print\_winner function. We think the gui is fully functional now. We think that the next thing we have to do is set up the bot to actually be able to think. Right now, it blocks then plays in corners then plays randomly.

11/13 - Group. Integrated Player.py class into UTTT.py and had actually AI used min max search for winning 3 blocks. We change the team9.py name to UTTT.py and Player.py to PlayerAI.py

Player.py orignal link: <https://github.com/tarun018/AI-for-Ultimate-Tic-Tac-Toe>

AI vs Player9 played 30 times. AI won 22 and Player9 won 8

AI vs Player2 played 30 times. AI won 14 and Player 2 won 16

Player 9 vs Player 2 30 times. Player 9 won 18 and Player2 won 12. //Player9 supposed to be better than Player2

11/17-Jeremy and Steph:

Heuristic: if you can do a winning move, do it (wrote down psuedo code for this)

Else: count the number of opponent’s pieces in each block and calculate minimum. We assigned the minimum number a higher number so min/max alg would choose it

11/28-JF & SP:

Finished heuristic and added comments to PlayerAI. Started looking at final report requirements.

12/4/16 Group Presentation

* Introduce
* Explain rules of game-Jeremy
* Heuristics-Steph
  + Min max alpha beta pruning
  + Count the number of opponent’s pieces
  + Code on PP?
* Demo

Who wants to drive at the computer? Sang

How will we access our stuff?