**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI**

Batch No. :

**DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION SYSTEMS**

**Artificial Intelligence (BITS F444/ CS F407)**

**I Semester 2017-18**

**Programming Assignment-2**

**Coding Details**

**(October 3, 2017)**

*Instruction: Type the details precisely and neatly*

1. ID \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2014B2A70220P\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Maitri\_Shastri\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Mention the names of Submitted files :
   1. ->assign2.py
   2. ->gmclick.py
2. Total number of submitted files: \_\_\_\_2\_\_\_\_\_\_\_
3. Name of the folder :\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2014B2A7220PAIAssign2\_\_\_\_\_
4. Have you checked that all the files you are submitting have your name in the top?(yes/no) yes
5. Have you checked that all the files you are submitting are in the folder as specified in 4 (and no subfolder exists)?(yes/no)yes
6. Problem formulation
   1. State representation: The state is represented using a matrix of 0s for blank spaces,1s for moves by human and 2s for moves by the agent. to\_play attribute denotes whose turn to play 'min player' or the 'max player'
   2. Pseudo code of your successor function

List Getmoves(state):

For y in range(0,4)

For x in range(0,4):

If(state.matrix[x][y]==0){moves.append([x,y]); break;}

Return moves

State result(state,move)

If(move is not in getmoves(state)) return state

Newstate=state

if state.to\_play=min

{Newstate.matrix[move[0]][move[1]]=1 ; newstate.to\_play=max;}

Else {Newstate.matrix[move[0]][move[1]]=2 ; newstate.to\_play=min;}

Return newstate

Listof states getsuccessors(state)

For move in getmoves(state)

Succ.append(move,result(state,move))

Return succ

* 1. Terminal states generation process-Terminal state condition is checked using TerminalTest function that returns boolean value if state is terminal one. Checking is done according to logic below:-

For each position in the matrix -4 lists containing coordinates of 2 diagonal wins, horizontal win and vertical wins are generated and checked whether they are within bounds. If state has all 1s or 2s in any of the list coordinates, we conclude that the state is a winning terminal state and returns corresponding utility value. If the state is a non-terminal state or if it is a draw this function returns 0. To distinguish between draw and non terminal state, the number of tokens on the board is counted.

* 1. Data structure to store terminal states

Terminal states are not generated and stored permanently. Each time we need to check whether a state is a terminal state or not a function is called which checks whether a state is terminal using above logic

* 1. Method to access terminal states and corresponding utility values-First we use the function TerminalTest() function to check if the state is terminal state and get the corresponding utility value using getTerminalvalue() function returns +1 for agent win and –1 for human win and 0 for a draw.

1. Minimax Technique details
   1. Node structure: A matrix of 0s,1s and 2s depicting the game board and to\_play ='min' or 'max' showing whether it is a min node or max node

* 1. Method to ensure the correctness of terminal test (describe in maximum 4 lines)

For each position in the matrix -4 lists containing coordinates of 2 diagonal wins, horizontal win and vertical wins are generated and checked whether they are within bounds. If state has all 1s or 2s in any of the list coordinates, we conclude that the state is a winning terminal state. For non-winning states if the number of coins=16 we conclude that it is a draw.

* 1. Total number of nodes generated to play one game:369909
  2. Write the statistics here as asked

R1 = 369909 R2 = 72 R3 = N.A.(height of implicit stack was not calculated correctly)

R4 = 52.98 s(Only time taken by agent to calculate move is taken into account)

R5=(Number of nodes created per sec since value in microsec was less than 1)=6979

* 1. Code status (implemented fully/ partially/ not done) implemented fully

1. Alpha Beta technique details:
   1. Explain the logic used for pruning (in maximum four lines)

Space complexity of Minimax algorithm can be reduced if we don’t expand trees pertaining to those moves that are worse than the best we have. The track of best move is kept using alpha and beta which are inItialized to –infinity and +infinity respectively. When max node is examining it's children and we find that a min node's child's value is less than the best max has so far we don't furthur expand the siblings of this min node's child.

* 1. Total number of nodes generated to play one game 2625 nodes
  2. Write the statistics here as asked

R6 = 2625 R7 time saving= 0.991 R8 =0.435s

1. Code status (implemented fully/ partially/ not done)implemented fully

1. Comparative analysis

Fill in the following information based of 10 independent games

|  |  |  |
| --- | --- | --- |
|  | Minimax Algorithm | Alpha Beta Pruning |
| Average number of nodes created | 4,45,202 | 3459 |
| Average time taken | 67 | 0.54 |
| Number of times machine wins (player M) | 10 | 10 |

1. GUI details
   1. Created the GUI (yes/ No):yes
   2. Have created it according to the specifications?(yes/No)No
   3. Which module of Python is used for creating graphics? Tkinter
   4. Is this under the standard Python library or not?yes
   5. If not, why?
2. Graphics details:
   1. Is turtle graphics working fine for displaying the board and coins?

Tkinter graphics Is working fine and displaying board and coins correctly.

* 1. How have you calibrated the board and accepted human input to play the game?

Calibration is not implemented in the main module and human player has to enter x-index and y-index of move through keyboard. However, calibration has been done in gmclick.py which displays coins according to click but this module could not be integrated with assign2.py.

* 1. How are you showing the base line?

Using a red line on top of the board

* 1. How are you showing the move of the machine?

Using green coin on the board and matrix displayed on the terminal shows 2 for machine move.

* 1. How are you showing the move of the human player?

Using blue coin on the board and matrix displayed on the terminal shows 1 for human move.

1. Compilation Details:
   1. Code Compiles (Yes/ No):\_\_\_\_\_yes\_\_\_\_\_\_\_\_\_
   2. Ensured the compatibility of your code with the specified Python version(yes/no)\_\_\_\_\_yes\_\_\_\_\_\_\_
   3. Instructions for compilation of your files mentioning the multi file compilation process used by you (We may use the replica of these for compiling your files while evaluating your code)-single file submitted
2. Driver Details: Does it take care of the options specified earlier(yes/no):\_\_\_yes\_\_\_\_\_\_\_\_
3. Execution status (describe in maximum 2 lines) Code compiling, executing and giving satisfactory results with display function working and showing status of gameboard. User input should be given through keyboard as calibration has been implemented in a separate module that could not be integrated in the main module.
4. Declaration: I, Maitri Shastri\_\_\_\_\_\_\_ (name) declare that I have put my genuine efforts in creating the python code for the given programming assignment and have submitted only the code developed by me. I have not copied any piece of code from any source. If the code is found plagiarized in any form or degree, I understand that a disciplinary action as per the institute rules will be taken against me and I will accept the penalty as decided by the department of Computer Science and Information Systems, BITS, Pilani.

ID\_\_\_\_\_\_\_\_\_\_\_\_\_2014B2A70220P\_\_\_ Name:\_\_ Maitri Shastri\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_3/10/2017\_\_\_\_\_

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Should not exceed three pages