**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 \_2015A4PS0239P\_

Name \_SAI SHUBODH P\_

1. Mention the names of Submitted files :
   1. <shubodh\_alphabeta\_final.py>
   2. <shubodh\_minimax\_final.py>
   3. <coding details PA2 final.docx>
2. Total number of submitted files: 3
3. Name of the folder :2015A4PS0239P
4. Have you checked that all the files you are submitting have your name in the top?(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)
6. Problem formulation
   1. State representation: It is represented as array of tuples (2D matrix).

1 represents computer has won

2 represents user has won

0 represents the game has drawn

* 1. Pseudo code of your successor function:

function result(playerNum, action):

coln = action

row = finding\_row(coln)

state[row][coln] = player

* 1. Terminal states generation process: The function win\_state(A) takes the current state as input and determines whether the game is in any one of the winning patterns and returns 1 if Computer wins, -1 if user wins and 0 if it is drawn. (Utility values)
  2. Data structure to store terminal states: Array of tuples
  3. Method to access terminal states and corresponding utility values:

Mentioned in point c.

1. Minimax Technique details
   1. Node structure: Contains state.

* 1. Method to ensure the correctness of terminal test (describe in maximum 4 lines): if win\_state(A) function returns 1 or -1, I have defined the function to break at that point and print whoever has won as output.
  2. Total number of nodes generated to play one game: 721435
  3. Write the statistics here as asked

R1 = 721435 R2 = 64 bytes R3 = 16

R4 = 308.72

* 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)

Pruning basically checks if the minimum or maximum possible value is less/ more than a particular node, and if that turns out to be true, it prunes out the other nodes. This is because, for example, if first node is max and then, it is branching into min, we can prune out the remaining nodes because the min node is not going to pick values above that value anyways and the max node already has a value greater than that.

* 1. Total number of nodes generated to play one game 174560
  2. Write the statistics here as asked

R6 = 174560 R7 = 0.95 R8 = 4.50

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

1. Comparative analysis

Fill in the following information based of 10 independent games

|  |  |  |
| --- | --- | --- |
|  | Minimax Algorithm | Alpha Beta Pruning |
| Average number of nodes created | 721435 | 174560 |
| Average time taken | 308.72 | 4.50 |
| 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) YEs
   3. Which module of Python is used for creating graphics? Turtle
   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?Yes
   2. How have you calibrated the board and accepted human input to play the game?Yes
   3. How are you showing the base line? Red line below the board
   4. How are you showing the move of the machine? Blue circle
   5. How are you showing the move of the human player? Green circle
3. Compilation Details:
   1. Code Compiles (Yes/ No): yes
   2. Mention the .py files that do not compile: none
   3. Any specific function that does not compile: none
   4. Ensured the compatibility of your code with the specified Python version: yes
   5. 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)
4. Driver Details: Does it take care of the options specified earlier(yes/no): yes
5. Execution status (describe in maximum 2 lines): Everything gets executed perfectly.

1. Declaration: I, Sai Shubodh 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: 2015A4PS0239P Name: Sai Shubodh P

Date: 03-10-2017

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