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

Batch No. :

**DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION SYSTEMS**

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

**I Semester 2018-19**

**Programming Assignment-3**

**Coding Details**

**(October 20, 2018)**

*Instruction: Type the details precisely and neatly*

1. ID \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2017H1030130130P\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_Santosh Desai\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Mention the names of Submitted files :
   1. State.py
   2. Helper.py
   3. Gui.py
   4. Driver.py
   5. Api.py
2. Total number of submitted files: \_\_\_\_\_5\_\_\_\_\_\_
3. Name of the folder :\_\_\_\_\_\_\_2017H1030130P\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
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:

It is a Spatial Matrix. 0 indicates empty place, 1 indicates player 1’s coin, 2 indicates player 2’s coin.

-1 are used are guards to indicate forbidden places and lines to travel by. It is interpreted accordingly by the GUI module.

* 1. Pseudo code of your successor function

Successor(current node):

For all coins:

For possible places in all 6 directions:

For both attack move and shift move:

If the obtained node wasn’t encountered during current decision making session:

Generate a new node and add to successors list.

If no new nodes were created during search

Duplicate current node and make it child of current node (Voluntary forfeit/pass step)

Return all successors

* 1. Terminal states generation process (manual/ automated). Also describe if it is one time generation of terminal states or you are generating the terminal states every time you reach next state.

Terminal States are given by a criterion and is automated. It is per decision.

If there are only three coins on the board, end the game.

If one of the player has no coins, end the game.

If current state has no children, end the game.

* 1. Data structure to store terminal states (hash table or any other?)

A Transposition Table is used to keep track of terminal states.

* 1. Method to access terminal states and corresponding utility values

Each terminal state is hashed in the following manner:

[<board\_values\_hash>+<turn\_of\_the\_player>] 🡪 Node.

1. Minimax Technique details
   1. Node structure:

Variables :

state, action, parent, depth, utility, child\_nodes\_list, number\_of\_children, number of coins of player 1, number of coins of player 2, playerId of current node.

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

It is evaluated in terms of gameStatus function which tells how many coins of each player are present.

* 1. Are you limiting the depth using any heuristic to evaluate the approximate value of the state? At which depth are you deciding to return back?

There are three modes of game play. Easy, Medium and Hard. Easy mode is hardwired in the given code and there is no GUI or CLI provided to alter it. The only way is through changing values in code. Easy corresponds to depth 4, Medium : depth 5 and Hard : depth 6. Please refer line 123 of driver.py.

However, when total number of coins equals the diameter of the board, the number of successors maximizes. The given simple heuristic isn’t powerful enough to make decision effectively. Hence, the search depth is boosted to 7 to quickly finish off the game.

* 1. Total number of nodes generated to play one game: 34290782
  2. Write the statistics here as asked

R1 = 34290782 R2 = 64 bytes R3 = 6

R4 = 3947 seconds (including user decision delays) R5= 0.35 per microsecond

* 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)

At any node if Alpha > Beta, the entire subtree is not considered for valuation.

If the node has been previously examined, it won’t be examined again.

If a node has been encountered in transposition table, it won’t be evaluated again.

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

R6 = 19109057 R7 = 0.55 R8 = 1748 seconds (including user decision delays)

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

1. Comparative analysis

Fill in the following information based of 10 independent games

|  |  |  |
| --- | --- | --- |
|  | Minimax Algorithm | Alpha Beta Pruning |
| Average number of nodes created | 36691136.57 | 21805760.193 |
| Average time taken | 3582.6 seconds | 1863.45 seconds |
| Number of times machine wins (player M) | 3 | 5 |

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? 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? Yes
   2. How have you calibrated the board and accepted human input to play the game?

It interprets the state and draws a GUI representation of the state. Mouse clicks are indicators of game play by user.

* 1. How are you showing the board?

Yellow circles indicate empty places, Red coins are for Player 1 and Green coins are for Player 2. Coins can move only along blue lines connecting the yellow circles.

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

Once the decision is made, it selects a coin, highlighted by dark blue surround highlight, it moves the coin to one of the purple highlighted destinations. Then all highlights are removed.

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

If the user clicks on their coin, the possible places are highlighted where he can drop the coin. The user needs to double click on the destination. Once a move is done, they need to press enter on console. This indicates the end of move and turn is given to computer/AI.

Note : There is a possible mischief here, player 1 could make multiple moves which may lead to crash of the game. As this game was for evaluation purposes only, much time wasn’t spent on bugs. The user needs to be honest in game play.

1. Compilation Details:
   1. Code Compiles (Yes/ No):\_\_\_\_\_\_\_Yes\_\_\_\_\_\_\_
   2. Mention the .py files that do not compile:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_N/A\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   3. Any specific function that does not compile:\_\_\_\_\_\_\_\_\_\_\_\_\_\_N/A\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   4. Ensured the compatibility of your code with the specified Python version(yes/no)

\_\_\_Yes (Py 2.7)\_\_\_\_\_\_\_\_\_

* 1. 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)
     1. Run Driver.py
     2. Keep the console open on the side to press enter indicating end of turn by user.

1. Driver Details: Does it take care of the options specified earlier(yes/no):\_\_\_\_\_Yes\_\_\_\_\_\_
2. Execution status (describe in maximum 2 lines)

Unless there is a mischievous move performed as mentioned above, the code runs satisfactorily

1. Declaration: I, \_\_\_\_\_\_\_\_\_\_\_Santosh Desai\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (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\_\_\_\_\_\_\_\_\_\_2017H1030130P\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name:\_\_\_\_\_\_\_\_Santosh Desai\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_17-10-2018\_\_\_\_\_\_\_\_\_\_\_

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Should not exceed four pages