**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 2015A7PS0127P

Name SAMIP JASANI

1. Mention the names of Submitted files :
   1. alphabeta.py
   2. driver.py
   3. finalgui.py
   4. minimax.py
   5. myversion.py
   6. 2015A7PS0127P.docx
   7. <filename.ext>
2. Total number of submitted files: \_\_\_6\_\_\_\_\_\_\_\_
3. Name of the folder : 2015A7PS0127P
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? yes
6. Problem formulation
   1. State representation:

My State consists of board condition in form of a 2D Array and marker for which players turn it is.

* 1. Pseudo code of your successor function

My Successor function checks which actions are possible and return a list of states which are successor to current state.

* 1. Terminal states generation process

Instead of generating the terminal states. I have written a function to check for terminal state. Once A terminal state is formed it is added to hash table in the form of dict so that same need not be computed each time.

* 1. Data structure to store terminal states

Hash Table in form of dict

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

Check if state in hash table. If there return corresponding utility value

1. Minimax Technique details
   1. Node structure:

node is min\_value or max\_value function that contains state of the board and player whose turn it is

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

Terminal Test is correct as it looks at any possible combination of 3 coins to check if it is terminal case

* 1. Total number of nodes generated to play one game:

65752 nodes

* 1. Write the statistics here as asked

R1 = 65752 R2 = 1088 Bytes R3 = 16

R4 = 9.85 sec R5= 0.0067 nodes per microsec

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

Fully Implemented

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

alpha and beta which is best maximum and minimum utility value in the path to the node are passed to compare if value of max node is >beta or min node < alpha then that tree is pruned

* 1. Total number of nodes generated to play one game

2733 (approx)

* 1. Write the statistics here as asked

R6 = 2733 R7 = 0.9585 R8 = 0.42 sec

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 | 65752 | 2733 |
| Average time taken | 9.85 | 0.42 |
| 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

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

Yes

* 1. How are you showing the base line?

Top red line is base line

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

Green coin display

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

Blue coin display

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

python driver.py

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

It is working properly

1. Declaration: I, \_\_\_\_\_\_\_\_\_\_\_SAMIP JASANI\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (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\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2015A7PS0127P\_\_\_\_\_\_\_\_\_\_\_ Name:\_SAMIP JASANI\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_3/10/18\_\_\_\_\_\_\_\_\_\_

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