There are 2 parts in the Team Project. We are going to start with an individual project (IP) and to finish as a team project (TP)

**IP Description (Due by end of Module 6)**

Each student is required to write a Tic-Tac-Toe (3x3) program. The student is going to build the classic game tic-tac-toe in Python. In this game, one player is X by a user and the other player (the computer) is O. You can imagine that the game will be played between you and the computer. Your program has the following requirements:

1. Selection of the first player

2. Assignment of "O" or "X"

3. Computer can select a position based on either random position.

4. If computer can select a position based on an algorithm (e.g. MiniMax algorithm or rule-based) approach, you will earn 20 extra points.

5. At each run, the program should display the board (3x3) as shown below. Placement is made based on the pre-assigned positional number.

6. Your program prints appropriate message if there is a winner and then quits.

7. Your program saves all the moves between playersin a file called tictactoe.txt (X:5 O:2 X:1 O:9 etc...)

8. Your program should handle incorrect inputs (e.g. input validation) and continue to play without an error.

(example of 3x3 board)

|  |  |
| --- | --- |
| ***Criteria*** | ***% of Grades*** |
| Execution without an error, input validation and coding comments | 70% |
| File Logging | 30% |
| **TOTAL** | 100% |

(Resource of rule-based minimax algorithm for Tic-Tac-Toe) <https://towardsdatascience.com/tic-tac-toe-creating-unbeatable-ai-with-minimax-algorithm-8af9e52c1e7d>

Please note that copying a segment of code from the Internet and submitting it as your work is considered as plagiarism.

<https://learning.oreilly.com/library/view/python-crash-course/9781492071266/xhtml/ch13.xhtml#ch13lev1sec2>

<https://medium.com/byte-tales/the-classic-tic-tac-toe-game-in-python-3-1427c68b8874>

<https://gsurma.medium.com/tic-tac-toe-creating-unbeatable-ai-with-minimax-algorithm-8af9e52c1e7d>

<https://www.youtube.com/watch?v=fT3YWCKvuQE>

**TP Description (Due by end of Module 10)**

Upon finishing up the IP project, each student now selects a partner (could be 3 in total depending on the total number of students in the class). Each group is going to expand the 3x3 tic-tac-toe game into 5x5 tic-tac-toe game using object-oriented programming approach. The same guidelines and requirements applies to TP as described in IP above including the extra point for using an AI based algorithm. If there are 4 consecutive positions occupied by the same label ("X" or "O"), then you can declare as a winner.

1. Selection of the first player

2. Assignment of "O" or "X"

3. Computer can select a position based on either random position.

4. If computer can select a position based on an algorithm (e.g. MiniMax algorithm or rule-based) approach, you will earn 20 extra points.

5. At each run, the program should display the board (5x5) as shown below. Placement is made based on the pre-assigned positional number.

6. Your program prints appropriate message if there is a winner and then quits.

7. Your program saves all the moves between players in a file called tictactoe.txt (X:5 O:2 X:1 O:9 etc...)

8. Your program should handle incorrect inputs (e.g. input validation) and continue to play without an error.

9. Your program should be designed using object oriented design and you should presents a class diagram as shown below.

|  |  |  |
| --- | --- | --- |
|  | or |  |

(example of 5x5 board)

A picture containing timeline

Description automatically generated

(example of a class diagram)

(Resource of rule-based minimax algorithm for Tic-Tac-Toe) <https://towardsdatascience.com/tic-tac-toe-creating-unbeatable-ai-with-minimax-algorithm-8af9e52c1e7d>

Please note that copying a segment of code from the Internet and submitting it as your work is considered as plagiarism.

|  |  |
| --- | --- |
| ***Criteria*** | ***% of Grades*** |
| Class diagram (OOD required) | 20% |
| Execution without an error, input validation and coding comments | 50% |
| File Logging | 30% |
| **TOTAL** | 100% |

**TP Presentation (10-minute Video Clip)**

In the video clip, one person will first present their object-oriented design including class diagrams and then the other person runs a demo while executing the code. Each presenter will have 5 minutes for their portion.

|  |  |
| --- | --- |
| ***Criteria*** | ***% of Grades*** |
| Structure | 20% |
| Visual Presentation | 30% |
| Verbal Quality & Engagement | 30% |
| Collaboration | 20% |
| **TOTAL** | 100% |