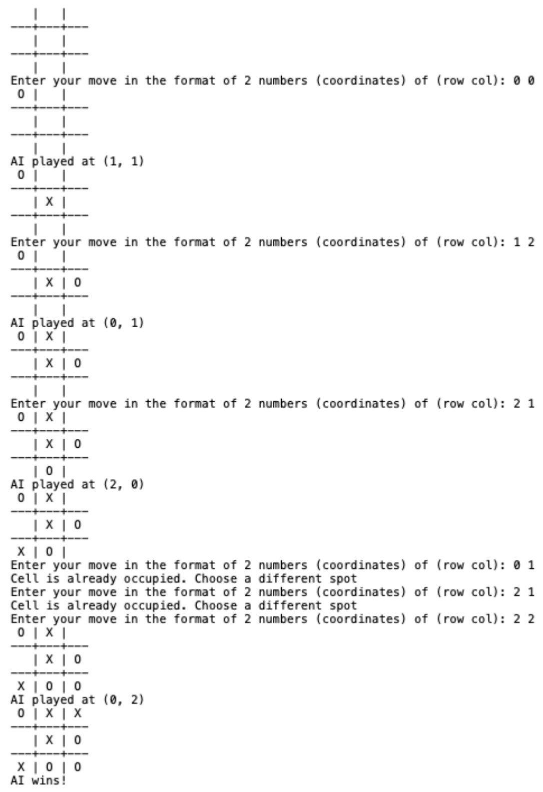
Kashyap Challapalli and Arnav Jain
CSCI 3202 Intermediate Results Write-up

Our game was a tic-tac-toe game where a user could play against an AI agent, and the main concept we used from a class other than Game theory and decision-making concepts is the Mini Max Algorithm, where we assign the scores of ‘X’ as 1, and ‘O’ as -1, and a tie is 0. The AI is assigned X and the human is assigned O to play with the minimax algorithm works where we have a boolean to track whether we are maximizing (for the AI agent) or minimizing (for us humans). We recursively call the minimax function at each next move (while changing the isMaximizing boolean to false or true and increasing the depth of the decision tree). Eventually, we check who our winner is by checking all of the rows, columns, and diagonals and seeing if the values are equal to each other in those directions and output whoever the winner is there, otherwise outputting a tie if all of those directions are exhausted and there’s no winner. This is it for the actual game theory and minimax concept implementations, and the rest is just the actual gameplay and user interface.

Right now, our user interface is text-based so you enter the coordinates of the move you want to make, where it is row, col, and then you keep going, and at each move, we check to see if there is a winner or not, and if there is we output who it is or a tie.

We didn’t have to use any extra libraries or frameworks so far to create the code, just pure python. There was no need to since we don’t have any functionalities that require it, as the actual algorithms and “thinking” that the code or agent does is taken care of by implementing algorithms in simple python.

In terms of testing our code, we have the minimax function working, as when you start off with O and the AI agent goes second as X, you have an interactive game working. One thing that may need some improvement is the interface, but overall it’s a solid 1st version, that has all the functionalities you need. This is an example of one of the tests we ran:



```
   |   |
---+---+---
   |   |
---+---+---
   |   |
Enter your move in the format of 2 numbers (coordinates) of (row col): 0 0
 O |   |
---+---+---
   |   |
---+---+---
   |   |
AI played at (1, 1)
 O |   |
---+---+---
   | X |
---+---+---
   |   |
Enter your move in the format of 2 numbers (coordinates) of (row col): 1 2
 O |   |
---+---+---
   | X | O
---+---+---
   |   |
AI played at (0, 1)
 O | X |
---+---+---
   | X | O
---+---+---
   |   |
Enter your move in the format of 2 numbers (coordinates) of (row col): 2 1
 O | X |
---+---+---
   | X | O
---+---+---
   | O |
AI played at (2, 0)
 O | X |
---+---+---
   | X | O
---+---+---
 X | O |
Enter your move in the format of 2 numbers (coordinates) of (row col): 0 1
Cell is already occupied. Choose a different spot
Enter your move in the format of 2 numbers (coordinates) of (row col): 2 1
Cell is already occupied. Choose a different spot
Enter your move in the format of 2 numbers (coordinates) of (row col): 2 2
 O | X |
---+---+---
   | X | O
---+---+---
 X | O | O
AI played at (0, 2)
 O | X | X
---+---+---
   | X | O
---+---+---
 X | O | O
AI wins!
```