Ain Shams University

Faculty of Engineering

Computer and Systems Engineering

4th Year

CSE481

Mancala Project

Students Data

Name	ID
سيف محمد طاهر فتحي	1600678
مصطفى محمد صديق	1601427
مصطفى حسن محمد	1601408
يوسف علاء عبد المحسن	1601737
محمد عصام عبد الحميد	1501259

Project Description

In this project we implemented an AI agent for Mancala Game using the Minimax algorithm with alpha-beta pruning. The game allows a human player to play against the AI agent. Our algorithm uses a depth-first strategy when exploring the game tree to ensure efficient memory usage. The project is implemented using python. The game allows either player to start first. The game is console based.

Mancala is a strategy based two player game which can be played with any anything resembling a stone. Each player has 6 pits with 4 stones in each and each have a target pit on their right, at the end of the game the player with the most stones in the target pit wins.

On each turn the player can choose to move the stones in any non-empty pit of his, the moving stones are removed from the chosen pit and each following pit gets one stone until they are finished if the stones end on the target pit, the player can play again.

There are two modes to the game and are also supported in our implementation:
- With Stealing: Which is a special case that the moving stones of the player end on an empty pit of his he then captures all the stones from his opponent's corresponding pit and adds them to his target pit.
- Without Stealing

Our implementation covers all the main features stated in the first paragraph as well as the following bonus features:
- We have multiple difficulty levels with all guaranteed to return a move in 30 seconds or less and they are based on different utility functions and tree depths explained in the following section.
    - Easy
    - Medium
    - Hard
- We Support game saving and loading.

Utility Functions Description

1. Easy Difficulty: The tree is built with depth 4 and the terminal nodes take values based on whether this move will add to the score difference in AI Favor and this value becomes the value of the node which is then used in the Minimax algorithm with alpha-beta pruning.
2. Medium Difficulty: The tree is built with depth 6 and the terminal nodes take values based on several factors different to the medium difficulty which are:
   - The score difference based on this move in the AI favor and the value is added to the stacked value of the node.
   - If this move grants me to play another move (the final stone lands on the target pit) it takes a positive 1 to its stacked value and a negative 1 if this node is on the player's turn.
   - For With-Stealing mode only: If this move grants to steal the stones from the player's stones in the corresponding non-empty pit (see With-Stealing description in the previous section) it takes a positive 1 to its stacked value and a negative 1 if this node is on the player's turn.
   - For With-Stealing mode only: If this move fills one or more of the player's empty pits (stops the player from stealing), the node value is stacked with no of pits it fills if the node is on the AI turn and negative if the filled zero pits are from the player's turn.
3. Hard Difficulty: Same as medium but with tree depth 8.

User Guide

- On the entry page the game asks if we want to play first or the AI plays first
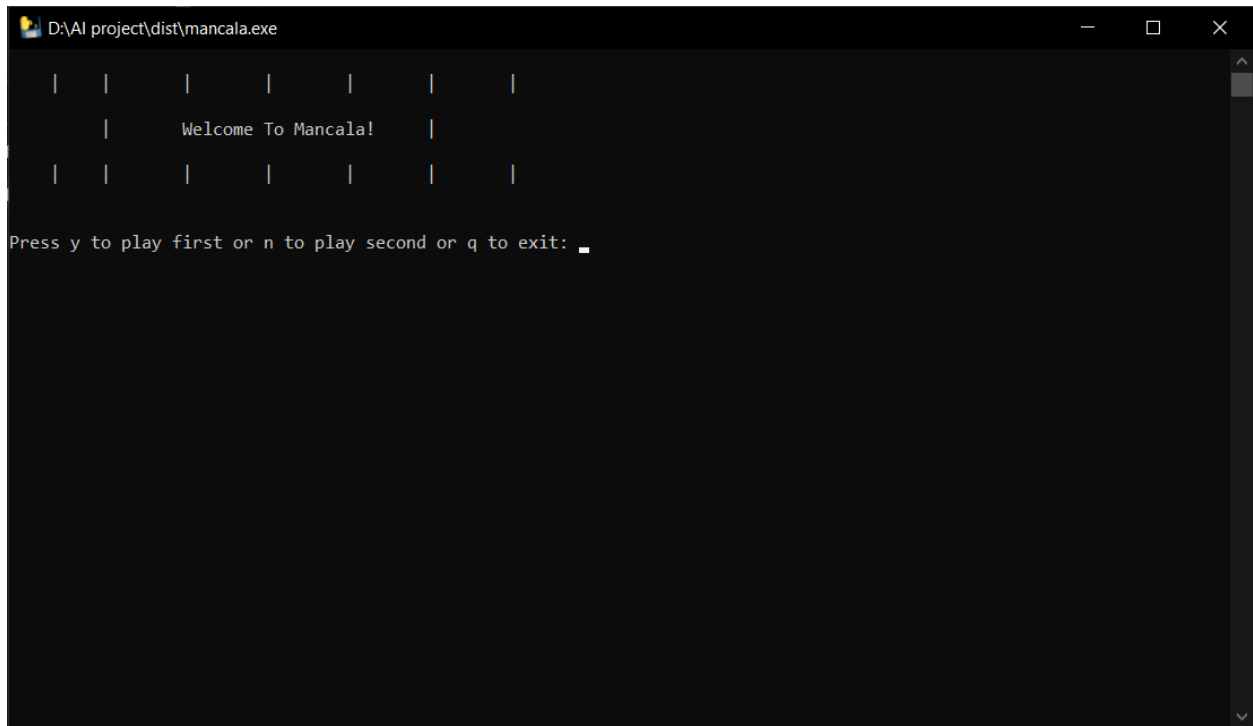
- It then asks for the difficulty of play

- It asks for game mode With-Stealing or Without-Stealing

- The game starts and your asked to input the pit you want to move.

- If the final stone ends on the target pit you may play again.


```
D:\AI project\dist\mancala.exe                                    —    □    ✕

Press y to play first or n to play second or q to exit: y

Press e for easy difficulty or m for medium difficulty or h for hard difficulty or q to quit: m

Press y for steal mode or n for non-steal mode or q to quit: y

Your turn.

    | 4 | 4 | 4 | 4 | 4 | 4 |

  0 |                             | 0

    | 4 | 4 | 4 | 4 | 4 | 4 |

      a   b   c   d   e   f

Input any non-empty pit from a to f,input s to save game, input q to exit: c

Your turn.

    | 4 | 4 | 4 | 4 | 4 | 4 |

  0 |                             | 1

    | 4 | 4 | 0 | 5 | 5 | 5 |

      a   b   c   d   e   f

Input any non-empty pit from a to f,input s to save game, input q to exit:
```

- Then the AI plays


```
D:\AI project\dist\mancala.exe                                    —    □    ✕

  0 |                             | 1

    | 4 | 4 | 0 | 5 | 5 | 5 |

      a   b   c   d   e   f

Input any non-empty pit from a to f,input s to save game, input q to exit: a

AI's turn.

    | 4 | 4 | 4 | 4 | 4 | 4 |

  0 |                             | 1

    | 0 | 5 | 1 | 6 | 6 | 5 |

      a   b   c   d   e   f

Your turn.

    | 0 | 4 | 4 | 4 | 4 | 4 |

  1 |                             | 1

    | 1 | 6 | 2 | 6 | 6 | 5 |

      a   b   c   d   e   f

Input any non-empty pit from a to f,input s to save game, input q to exit:
```

- If you input s at any time the game is saved

```
D:\AI project\dist\mancala.exe                              —    □    ×

    | 0 |  0 |  3 |  7 |  7 |  6 |

      a    b    c    d    e    f

Input any non-empty pit from a to f,input s to save game, input q to exit: c

AI's turn.

    | 0 |  1 |  5 |  6 |  6 |  0 |

 5 |                                 | 2

    | 0 |  0 |  0 |  8 |  8 |  7 |

      a    b    c    d    e    f

Your turn.

    | 0 |  0 |  5 |  6 |  6 |  0 |

 6 |                                 | 2

    | 0 |  0 |  0 |  8 |  8 |  7 |

      a    b    c    d    e    f

Input any non-empty pit from a to f,input s to save game, input q to exit: s

Thank you for playing.▄
```

- Now if you restart the game different state is found where you can input l to load the saved game.

```
D:\AI project\dist\mancala.exe                              —    □    ×

   |    |     |     |     |     |     |

         |       Welcome To Mancala!      |

   |    |     |     |     |     |     |


Press y to play first or n to play second or q to exit or l to load game: l

Your turn.

    | 0 |  0 |  5 |  6 |  6 |  0 |

 6 |                                 | 2

    | 0 |  0 |  0 |  8 |  8 |  7 |

      a    b    c    d    e    f

Input any non-empty pit from a to f,input s to save game, input q to exit: ▄
```

- This demonstrates the With-Stealing case occurring.

- This demonstrates the Without-Stealing case occurring.

- This screenshot shows if the AI plays first

```
D:\AI project\dist\mancala.exe                                              —    □    ✕
   |   |       |       |       |       |       |
Press y to play first or n to play second or q to exit or l to load game: n
Press e for easy difficulty or m for medium difficulty or h for hard difficulty or q to quit: h
Press y for steal mode or n for non-steal mode or q to quit: y
AI's turn.
    | 4 | 4 | 4 | 4 | 4 | 4 |
  0 |                             | 0
    | 4 | 4 | 4 | 4 | 4 | 4 |
      a   b   c   d   e   f
Your turn.
    | 0 | 4 | 4 | 4 | 4 | 4 |
  1 |                             | 0
    | 5 | 5 | 5 | 4 | 4 | 4 |
      a   b   c   d   e   f
Input any non-empty pit from a to f,input s to save game, input q to exit: _
```

- The game ends with a win or a draw.

```
D:\AI project\dist\mancala.exe                                              —    □    ✕
    | 0 | 0 | 5 | 2 | 1 | 2 |
 25 |                             | 11
    | 2 | 0 | 0 | 0 | 0 | 0 |
      a   b   c   d   e   f
Input any non-empty pit from a to f,input s to save game, input q to exit: a
AI's turn.
    | 0 | 0 | 0 | 2 | 1 | 2 |
 25 |                             | 17
    | 0 | 1 | 0 | 0 | 0 | 0 |
      a   b   c   d   e   f

    | 0 | 0 | 0 | 0 | 0 | 0 |
 31 |                             | 17
    | 0 | 0 | 0 | 0 | 0 | 0 |
      a   b   c   d   e   f
AI won!
```

Roles and Work Summary:

Name	Role
سيف محمد طاهر فتحي	Implementation of Game Structure and Class used in building the tree of the AI and Report.
مصطفى محمد صديق	Implementation of Game Structure and Report.
مصطفى حسن محمد	Implementation of Utility Functions
يوسف علاء عبد المحسن	Implementation of all bonus features implemented as well as the Class used in building the tree of the AI.
محمد عصام عبد الحميد	Implementation of Utility Functions