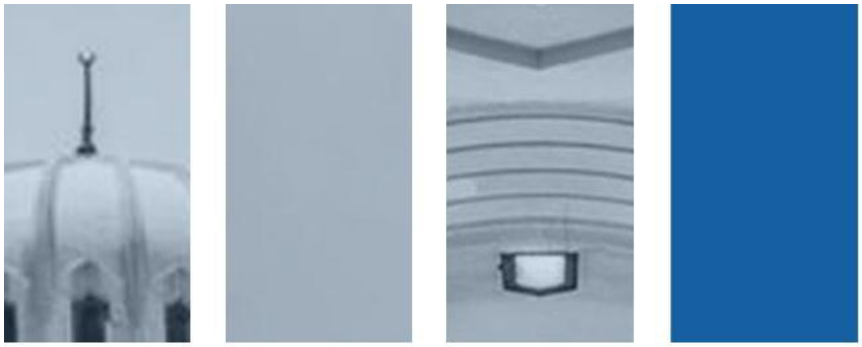


Mancala Game

Program: MainStream

Course: *CSE481*

Team:

- **Mohamed obada sarsar 16W0061**
- **Mariam Atef Shafik 1601372**
- **Mostafa Ashraf Mohamed 16T0107**

Submitted to:

Prof. Dr. Manual Morad Zaki
Ain Shams University
Faculty of Engineering
2021

SUMMER SEMESTER - YEAR 2020/2021

● Brief Description

This is an implementation of the classic 2-rank (2-players) Mancala where the object of the game is to capture the most stones.

Game Features:

☐ One of the players is an AI Agent.
☐ The game has 2 modes: with stealing and without stealing.
☐ The game supports 2 levels of hardness.
☐ As a player, you can choose either you or the bot to start first.
☐ The player can save the game status to continue later.

● Detailed Description of The Utility Functions

One recursive utility function is used.

The easy mode investigates 3 levels ahead and The hard mode uses a depth of 15.

Utility function detailed structure:

The first condition is the stopping conditions of the recursion which are:

- reaching the max depth of the mode
- one of the players finished their stones which means the game can’t go further

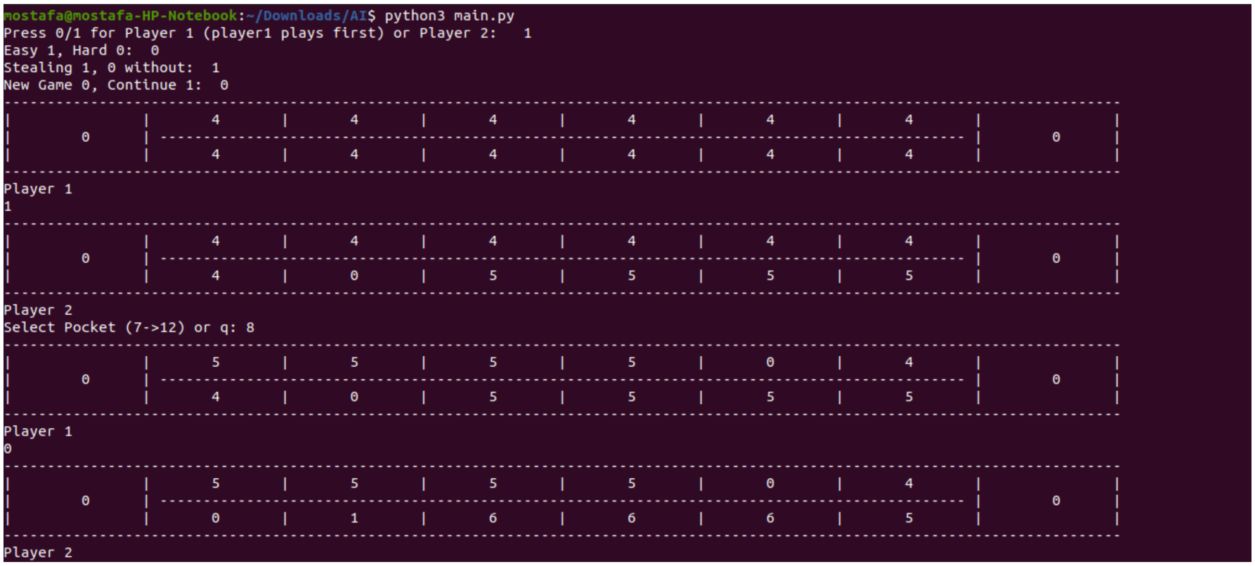
Here is where we assign the score (using the “cost” function) which is calculated by subtracting the number of stones in the AI’s mancala from the number of stones in the opponent’s mancala.

Second condition used for updating the ALPHA value and checking for cutting off (this condition is equivalent to a MAXIMUM taking turn)

Third condition used for updating the BETA value and checking for cutting off (this condition is equivalent to a MINIMUM taking turn)

- ## User Guide with Snapshots

  1. Choose which player will start the game.
  2. Choose the level of the game (Easy/Hard).
  3. Choose the mode of the game (Stealing/Without Stealing).
  4. Load the last game OR start a new game.



```
mostafa@mostafa-HP-Notebook:~/Downloads/AI$ python3 main.py
Press 0/1 for Player 1 (player1 plays first) or Player 2:   1
Easy 1, Hard 0:  0
Stealing 1, 0 without:  1
New Game 0, Continue 1:  0
-------------------------------------------------------------------------------------------------
|           |     4     |     4     |     4     |     4     |     4     |     4     |           |
|     0     | --------------------------------------------------------------------- |     0     |
|           |     4     |     4     |     4     |     4     |     4     |     4     |           |
-------------------------------------------------------------------------------------------------
Player 1
1
-------------------------------------------------------------------------------------------------
|           |     4     |     4     |     4     |     4     |     4     |     4     |           |
|     0     | --------------------------------------------------------------------- |     0     |
|           |     4     |     0     |     5     |     5     |     5     |     5     |           |
-------------------------------------------------------------------------------------------------
Player 2
Select Pocket (7->12) or q: 8
-------------------------------------------------------------------------------------------------
|           |     5     |     5     |     5     |     5     |     0     |     4     |           |
|     0     | --------------------------------------------------------------------- |     0     |
|           |     4     |     0     |     5     |     5     |     5     |     5     |           |
-------------------------------------------------------------------------------------------------
Player 1
0
-------------------------------------------------------------------------------------------------
|           |     5     |     5     |     5     |     5     |     0     |     4     |           |
|     0     | --------------------------------------------------------------------- |     0     |
|           |     0     |     1     |     6     |     6     |     6     |     5     |           |
-------------------------------------------------------------------------------------------------
Player 2
```

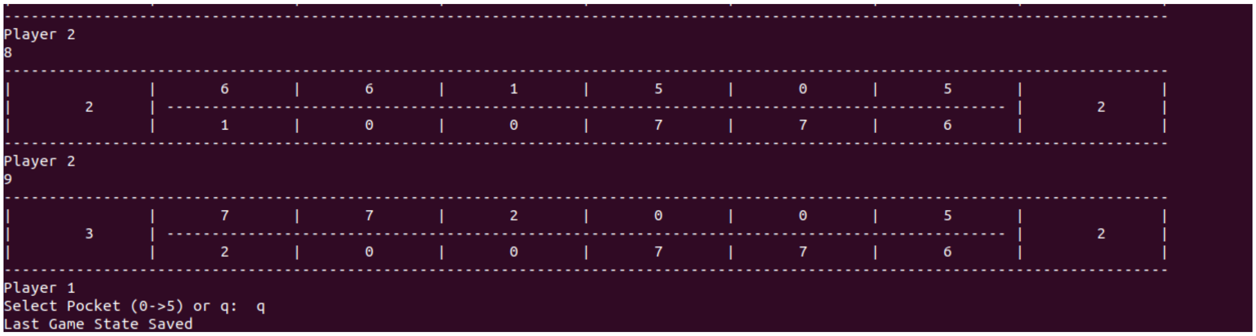
Fig.1

  5. Select packet

     (0:5) if The user is the first player.

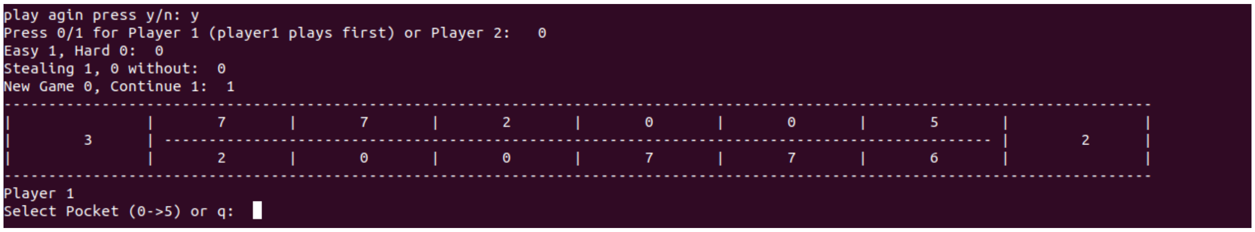
     (7:12) if The user is the second player.

  6. Any turn the user can save the game by entering (q).



```
-------------------------------------------------------------------------------------------------
Player 2
8
-------------------------------------------------------------------------------------------------
|           |     6     |     6     |     1     |     5     |     0     |     5     |           |
|     2     | --------------------------------------------------------------------- |     2     |
|           |     1     |     0     |     0     |     7     |     7     |     6     |           |
-------------------------------------------------------------------------------------------------
Player 2
9
-------------------------------------------------------------------------------------------------
|           |     7     |     7     |     2     |     0     |     0     |     5     |           |
|     3     | --------------------------------------------------------------------- |     2     |
|           |     2     |     0     |     0     |     7     |     7     |     6     |           |
-------------------------------------------------------------------------------------------------
Player 1
Select Pocket (0->5) or q:  q
Last Game State Saved
```

7. If the user chooses to load the last game by inter (1), the Saved board game will be shown.



```
play agin press y/n: y
Press 0/1 for Player 1 (player1 plays first) or Player 2:   0
Easy 1, Hard 0:  0
Stealing 1, 0 without:  0
New Game 0, Continue 1:  1
---------------------------------------------------------------------------------------------------------------------------------
|               |       7       |       7       |       2       |       0       |       0       |       5       |               |
|        3      | --------------------------------------------------------------------------------------------- |        2      |
|               |       2       |       0       |       0       |       7       |       7       |       6       |               |
---------------------------------------------------------------------------------------------------------------------------------
Player 1
Select Pocket (0->5) or q:  █
```

- ## Summary of How The Work Was Split Among The Team

Mohamed obada sarsar	Documentation + Codebase
Mariam Atef Shafik	Documentation + Codebase
Mostafa Ashraf Mohamed	Documentation + Codebase

- **Any additional documentation you might find useful (including code documentation, descriptions of difficulties encountered, tricks used, etc.)**

We divide our code into multiple functions as shown below.

1. Initialization Function

   1.1. Initialize our main list by the default starting game.

   1.2. Let the player make the first turn.

   1.3. Assuming the game is stealing mode.

2. End Function

   2.1. checks if the game is finished or not.

   2.2. summation of all packets of one player must be zero to end the game.

3. Switch Players Function:

   3.1. take input from the user to let the player or AI play first.

4. Cycle Function:

   4.1. The turns of the game.

   4.2. The player and AI play against each other by choosing which packet will be empty.

   4.3. Users can at any turn save a game on their PC and restore it anytime.

5. Cost Function:

   5.1. Take the status of the list

   5.2. Calculate the score of each player.

5.3. Decide which player is the winner.

6. Move Function:

6.1. Move balls of the packet which the player had chosen to the next packets.

7. Print Board Function:

7.1. Show the game board after each turn of each player.

8. Steal:

8.1. Only in Stealing Mode

8.2. Steal the balls from the other player if we put a ball in an empty packet.

9. Minimax Function:

9.1. Taking Alpha & Beta of the tree.

9.2. Taking the depth of the tree depends on the level of the game.