Mancala
Artificial intelligence

Mohamed Ahmed Khalil :1601104

Mohamed Elsayed Mohamed:1601129

Mohamed Ahmed Hassan :1601100

Mohamed Ashraf Mohamed :1501156

Mohamed Osama Elshafei:1501147

Members participation in the project

Mohamed Ahmed Khalil:implementation of ***minimax_alpha_beta, minimax_alpha_beta _without_stealing,The_winner,heuristic_value***

Mohamed Elsayed Mohamed:implementation of ***minimax_alpha_beta, minimax_alpha_beta _without_stealing,The_winner,heuristic_value***

Mohamed Osama Elshafie: implementation of next_move , main function

Mohamed Ahmed Hassan: implementation of next_move_without_stealing

Mohamed Ashraf Mohamed: implementation of game_ended,Initialize_board

Brief description of the project:

We formed a board as a list with 14 separate cups each player has 6 cups and a mancala
player 1: have cups from index 0 to index 5, mancala in index 6
player 2 (AI): have cups from index 7 to index 12, mancala in index 13

The user can choose whether he wants to play in stealing or non-stealing mode.

In the stealing mode: if the player moves his last card into an empty cup he collects his card with the opposite side cards at the opposite index

We used the minimax algorithm with alpha-beta pruning in implementing this game.

The user can choose initially the depth of the tree he wants to play against, the larger the depth, the harder the game will be.

The game supports that AI or Human can player first.

Language used:

Python without using any external frameworks or libraries

Function implementation:

The_winner:

Calculates the number of stones inside each mancala , and determine which player is the winner.

minimax_alpha_beta:

let the Ai plays first, so the AI will be the maximizer.

Let depth=2

Let alpha and beta is -inf and +inf respectively

We start drawing the tree, the AI start playing one of the 6 plays (7,8,9,10,11,12) so in depth zero we have only one node.

The Ai tries to determine the best play according to alpha-beta values

In minimizer turn at depth 1 we have 6 nodes and for each play the maximizer can play the minimizer has 6 plays (0,1,2,3,4,5) and so on.

At node 7,8,10,11,12: the minimizer has 6 plays he can perform

At node 9: the minimizer will wait until the maximizer plays another turn.

And so on.

Say player one plays and his last stone fill into an empty cup, he will take his last stone and the opposite side stones and put them all together into his mancala so it's called "Stealing".

Say we arrived at the last depth the user mentioned, we will call the function heuristic_value that will return a heuristic value for the cost.

minimax_alpha_beta _without_stealing:

this function has the same algorithm like the last function but without stealing.

We also here call the function next_move_without_stealing.

Heuristic_value:

Subtraction of the summation of maximizer mancala and the summation of minimizer mancala.

This function is used when we arrived at the last depth the user mentioned.

Next move without stealing :

It takes the current board shape and the index of the selected cup

So if it’s the ai turn the selected cup will be empty from the stones the number of stones which was in it will be put in the next cups , one stone for each cup

If the number of the current cup ==6 we have to skip as it’s our mangala

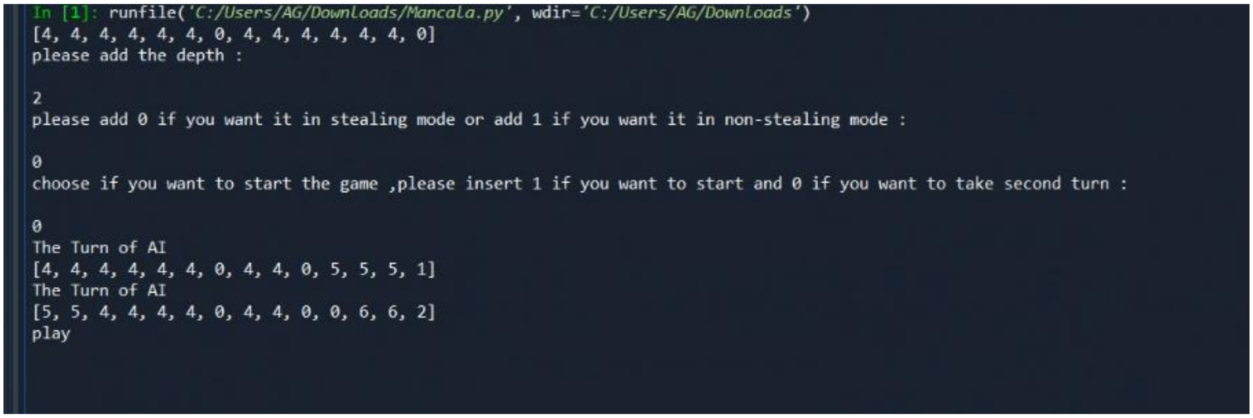
If the current cup ==14 then we repeat the count from 0 again

And if the last stone was at cup 13 the we open flag to make him take another turn

Next move *:*

we create a flag to indicate that if he will take another turn return this flag from function , we take the index of the cup that player want to play then empty this cup and take the stones in it and distribute them along the next cups , we check if we arrive to index 14 that mean we will take another turn , we check for stealing mode by last step if it’s in empty cup that meaning it will take the stones from opposite cup

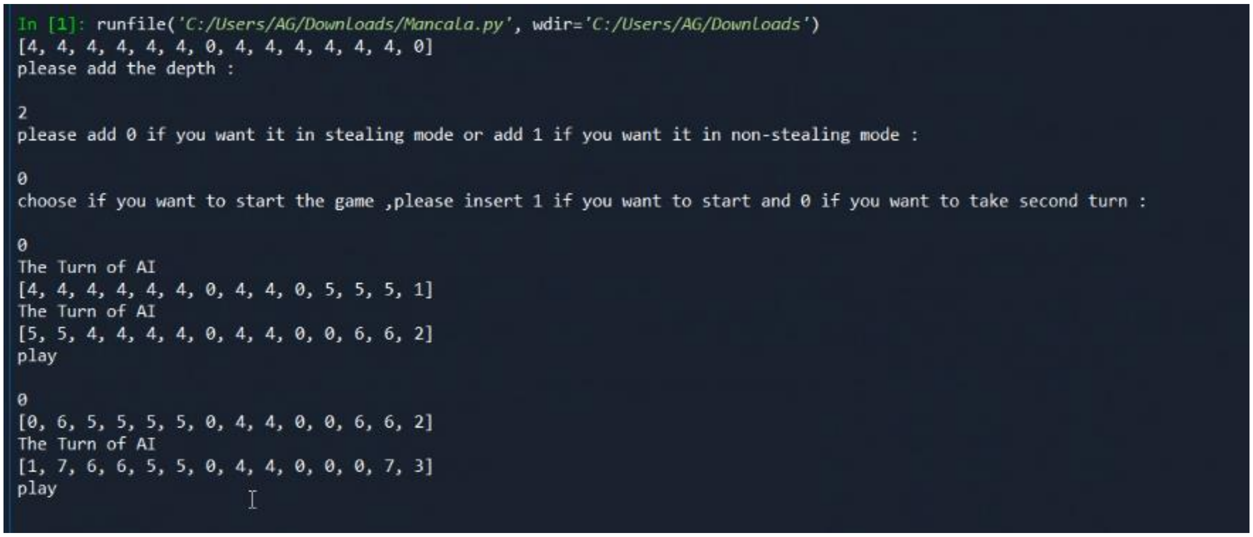
Snapshoots :



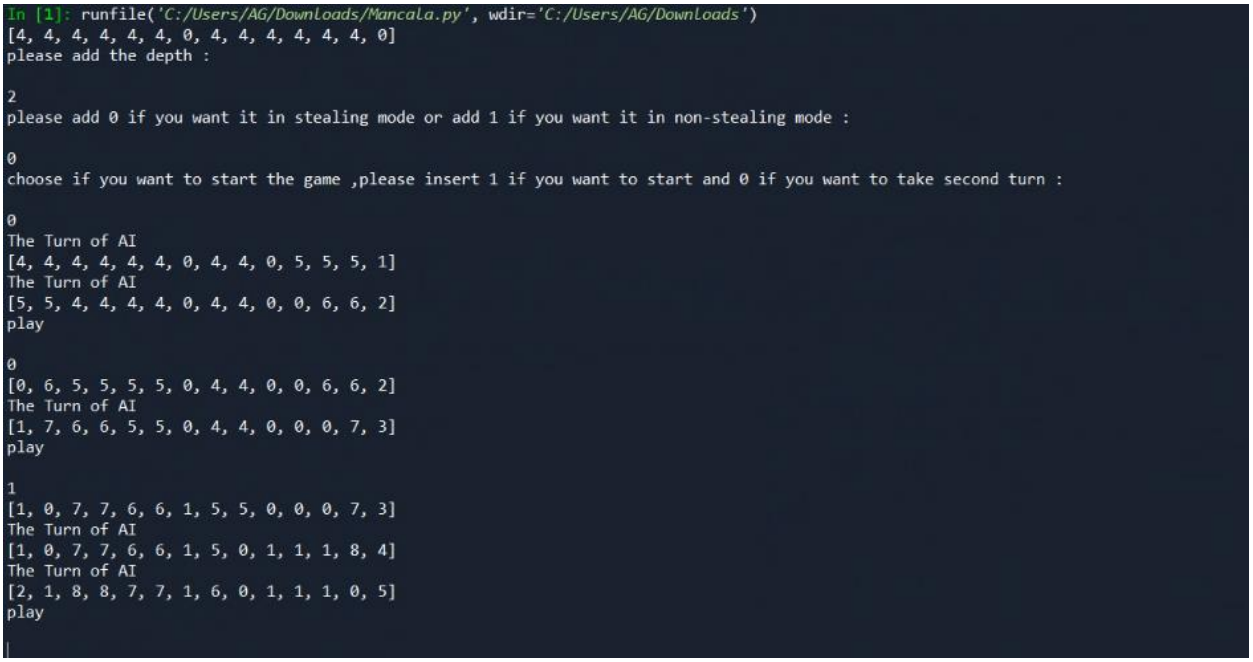
```
In [1]: runfile('C:/Users/AG/Downloads/Mancala.py', wdir='C:/Users/AG/Downloads')
[4, 4, 4, 4, 4, 4, 0, 4, 4, 4, 4, 4, 4, 0]
please add the depth :

2
please add 0 if you want it in stealing mode or add 1 if you want it in non-stealing mode :

0
choose if you want to start the game ,please insert 1 if you want to start and 0 if you want to take second turn :

0
The Turn of AI
[4, 4, 4, 4, 4, 4, 0, 4, 4, 0, 5, 5, 5, 1]
The Turn of AI
[5, 5, 4, 4, 4, 4, 0, 4, 4, 0, 0, 6, 6, 2]
play
```

The game asks the user to choose the 1)depth , 2) Mode ,3)Player or Ai

Here the Ai starts the game first, when the player see “play” he should make a play!

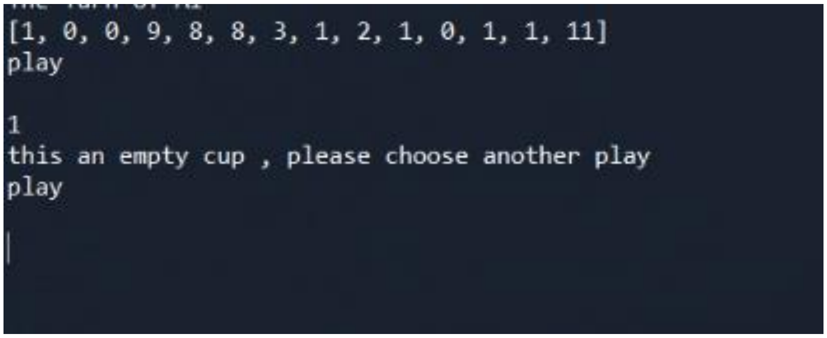


```
In [1]: runfile('C:/Users/AG/Downloads/Mancala.py', wdir='C:/Users/AG/Downloads')
[4, 4, 4, 4, 4, 4, 0, 4, 4, 4, 4, 4, 4, 0]
please add the depth :

2
please add 0 if you want it in stealing mode or add 1 if you want it in non-stealing mode :

0
choose if you want to start the game ,please insert 1 if you want to start and 0 if you want to take second turn :

0
The Turn of AI
[4, 4, 4, 4, 4, 4, 0, 4, 4, 0, 5, 5, 5, 1]
The Turn of AI
[5, 5, 4, 4, 4, 4, 0, 4, 4, 0, 0, 6, 6, 2]
play

0
[0, 6, 5, 5, 5, 5, 0, 4, 4, 0, 0, 6, 6, 2]
The Turn of AI
[1, 7, 6, 6, 5, 5, 0, 4, 4, 0, 0, 0, 7, 3]
play
```



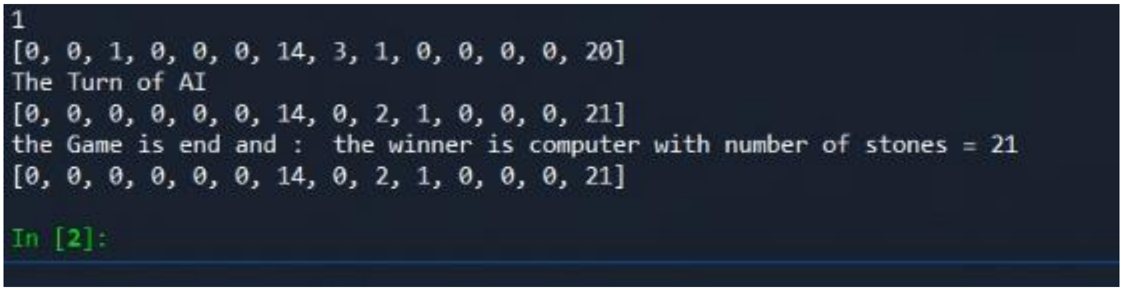
```
In [1]: runfile('C:/Users/AG/Downloads/Mancala.py', wdir='C:/Users/AG/Downloads')
[4, 4, 4, 4, 4, 4, 0, 4, 4, 4, 4, 4, 4, 0]
please add the depth :

2
please add 0 if you want it in stealing mode or add 1 if you want it in non-stealing mode :

0
choose if you want to start the game ,please insert 1 if you want to start and 0 if you want to take second turn :

0
The Turn of AI
[4, 4, 4, 4, 4, 4, 0, 4, 4, 0, 5, 5, 5, 1]
The Turn of AI
[5, 5, 4, 4, 4, 4, 0, 4, 4, 0, 0, 6, 6, 2]
play

0
[0, 6, 5, 5, 5, 5, 0, 4, 4, 0, 0, 6, 6, 2]
The Turn of AI
[1, 7, 6, 6, 5, 5, 0, 4, 4, 0, 0, 0, 7, 3]
play

1
[1, 0, 7, 7, 6, 6, 1, 5, 5, 0, 0, 0, 7, 3]
The Turn of AI
[1, 0, 7, 7, 6, 6, 1, 5, 0, 1, 1, 1, 8, 4]
The Turn of AI
[2, 1, 8, 8, 7, 7, 1, 6, 0, 1, 1, 1, 0, 5]
play

|
```

Here the AI played twice in a row because he fell his last stone into the mancala



```
The Turn of AI
[1, 0, 0, 9, 8, 8, 3, 1, 2, 1, 0, 1, 1, 11]
play

1
this an empty cup , please choose another play
play

|
```

Here the player tries to play from an empty cup!!



```
1
[0, 0, 1, 0, 0, 0, 14, 3, 1, 0, 0, 0, 0, 20]
The Turn of AI
[0, 0, 0, 0, 0, 0, 14, 0, 2, 1, 0, 0, 0, 21]
the Game is end and :  the winner is computer with number of stones = 21
[0, 0, 0, 0, 0, 0, 14, 0, 2, 1, 0, 0, 0, 21]

In [2]:
```

Sadly, the Ai wins ☹