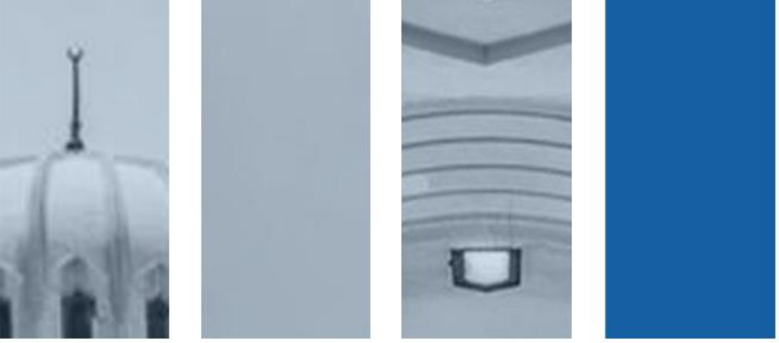


Mancala-AI-Project

Program:

Course Code: ***CSE 481***

Course Name: ***Artificial Intelligence***

Ain Shams University
Faculty of Engineering
Spring Semester – 2021

Groups Names:

ID	Name
1600505	خالد محمد السيد احمد خليل ابوالليل
1600753	عبدالرحمن محمد ايمن مصطفى
1600747	عبدالرحمن قدرى انور
1600719	عاصم كمال احمد
17x0040	محمد حماده محمد

A brief description the game:

Mancala is a generic name for a family of two-player turn-based strategy board games played with small stones, beans, or seeds and rows of holes or pits in the earth, a board or other playing surface. The objective is usually to capture all or some set of the opponent's pieces.

Most mancala games share a common general game play. Players begin by placing a certain number of seeds, prescribed for the particular game, in each of the pits on the game board. A player may count their stones to plot the game. A turn consists of removing all seeds from a pit, "sowing" the seeds (placing one in each of the following pits in sequence) and capturing based on the state of the board. The object of the game is to plant the most seeds in the bank. This leads to the English phrase "count and capture" sometimes used to describe the gameplay. Although the details differ greatly, this general sequence applies to all games.

If playing in capture mode, once a player ends their turn in an empty pit on their own side, they capture the opponent's pieces directly across. Once captured, the player gets to put the seeds in their own bank. After capturing, the opponent forfeits a turn.

Functions:

Name: get_possible_actions.

Parameters: s -> is a tuple of 2 elements first element is the game board as array, and second element is player turn.

Description: this function returns all possible action according to the current state of the game board.

Name: is_end_game.

Parameters: s -> is a tuple of 2 elements first element is the game board as array, and second element is player turn.

Description: this function returns true if the game reached the end state (when all the buckets for either player are zeros) and returns false else.

Name: get_state_score.

Parameters: s -> is a tuple of 2 elements first element is the game board as array, and second element is player turn.

Description: if we reached game end state, function would return float(“inf”) if first player has more score than the second player, float(“-inf”) if the second player has more score than first player, and 0 if both have equal scores. If it is not the game end state function will return player_one_score – player_two_score.

Name: max_value.

Parameters: s -> is a tuple of 2 elements first element is the game board as array, and second element is player turn.

depth-> is the depth of search tree (maximum number of recursion).

alpha-> the initial value of alpha.

beta-> the initial value of beta.

Stealing-> if it is true the game will be with stealing mode.

Description: this function returns the action with maximum heuristic value at certain depth using apha-beta algorithm.

Name: min_value.

Parameters: s -> is a tuple of 2 elements first element is the game board as array, and second element is player turn.

depth-> is the depth of search tree (maximum number of recursion).

alpha-> the initial value of alpha.

beta-> the initial value of beta.

Stealing-> if it is true the game will be with stealing mode.

Description: this function returns the action with minimum heuristic value at certain depth using apha-beta algorithm.

Name: get_result_state_without_stealing.

Parameters: s -> is a tuple of 2 elements first element is the game board as array, and second element is player turn.

a-> is the action that will be applied on game state.

Description: this function takes a state and action and return new state after the action is applied in without stealing mode.

Name: get_result_state_with_stealing.

Parameters: s -> is a tuple of 2 elements first element is the game board as array, and second element is player turn.

a-> is the action that will be applied on game state.

Description: this function takes a state and action and return new state after the action is applied in with stealing mode.

Name : Moving function:

Parameters :

board ->list the contain number of the stone for each index in the board ,

num -> number of index that want to move ,

flag -> for moving with stealing or without stealing .

Description-> to moving just read the number of stones in this index then distribute stones on the next indexes . if it is stealing it check if the last stone just fill in empty index and on the other side for another player if there stone so it was staled .

____

Name : Is end function

Description: Checks if game is end by check the value of index[0:6] or [7:13] those are the field of stones if anyone of them empty so the game is end otherwise the game is not end

Name :End of game

Description : It call function Is end to see if the game is end or not . if it is end then print scores who won or it was draw

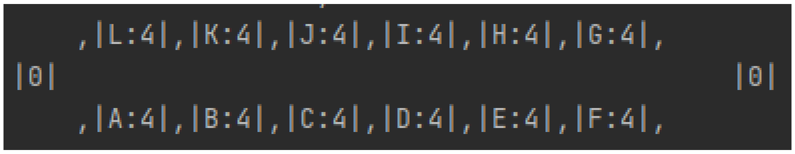
____

Name : Game_design

Parameter :

board -> list of the number of stones in every field of the table

Description : it takes the number of the stone in each filed and represent it like areal board



```
     ,|L:4|,|K:4|,|J:4|,|I:4|,|H:4|,|G:4|,
|0|                                           |0|
     ,|A:4|,|B:4|,|C:4|,|D:4|,|E:4|,|F:4|,
```

Name : convert

Parameter ->str

Description : to make easy for user not entering the indexes of the board just enter a char and the function convert it to the index like the above picture

____

Name : save

Parameter :

Borad-> list of number of stones in every field

Playerone -> just flag to know who play first player or bot

Mode -> depth of the tree alpha beta algrothim

Stealing -> flag to say if we play with stealing or not

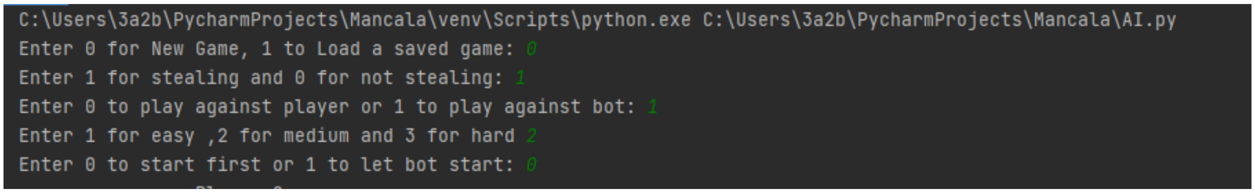
Description : just save in the file date , board ,player flag

,mode and stealing flag

Name : load

Description :load from a file the board and the game status like stealing flag and the mode

User guide



```
C:\Users\3a2b\PycharmProjects\Mancala\venv\Scripts\python.exe C:\Users\3a2b\PycharmProjects\Mancala\AI.py
Enter 0 for New Game, 1 to Load a saved game: 0
Enter 1 for stealing and 0 for not stealing: 1
Enter 0 to play against player or 1 to play against bot: 1
Enter 1 for easy ,2 for medium and 3 for hard 2
Enter 0 to start first or 1 to let bot start: 0
```

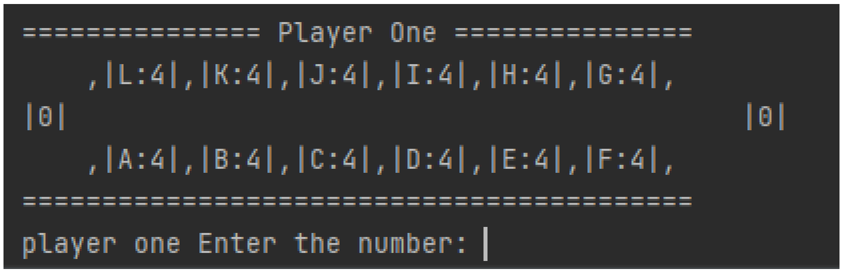
First will ask if I want to load old game or start new game

Then will ask for stealing mode or non-stealing mode

Then will ask if I want to player against player or bot

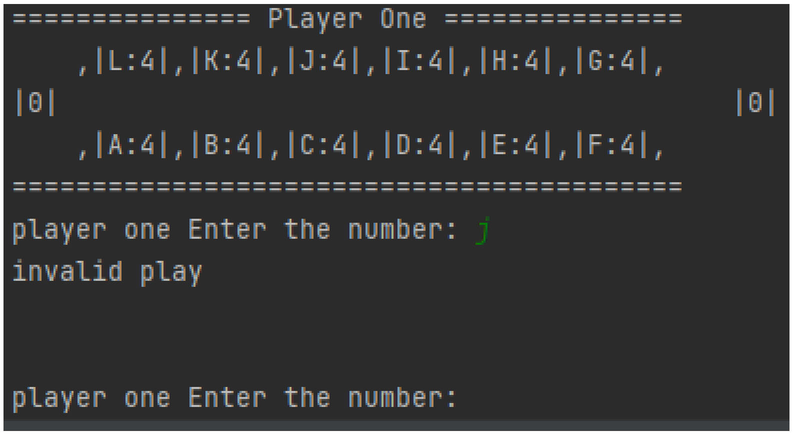
Then will ask for mode (1 for easy , 2 for medium and 3 for hard)

If we play against bot it will ask if I want to start or bot starts



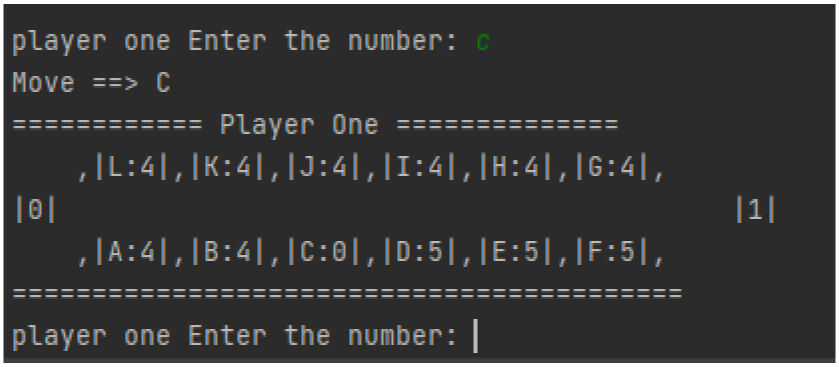
```
=============== Player One ===============
    ,|L:4|,|K:4|,|J:4|,|I:4|,|H:4|,|G:4|,
|0|                                          |0|
    ,|A:4|,|B:4|,|C:4|,|D:4|,|E:4|,|F:4|,
==========================================
player one Enter the number: 
```

This is board design for player one he choose from A to F if he chooses another char it will said invalid game and ask him to enter again like the below photo

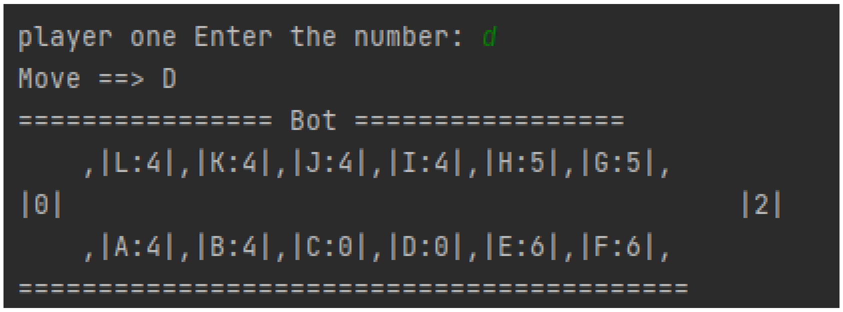


```
=============== Player One ===============
    ,|L:4|,|K:4|,|J:4|,|I:4|,|H:4|,|G:4|,
|0|                                          |0|
    ,|A:4|,|B:4|,|C:4|,|D:4|,|E:4|,|F:4|,
==========================================
player one Enter the number: j
invalid play



player one Enter the number: 
```

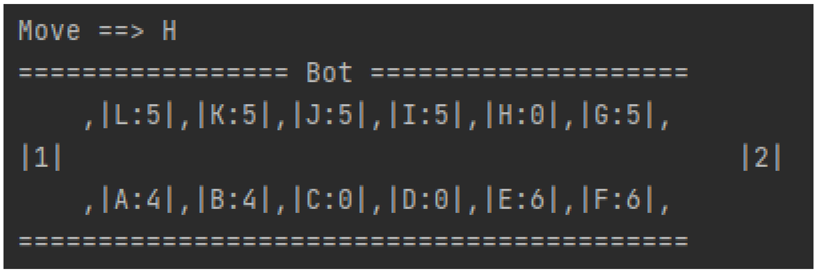
And continue the game between player and the bot or another player



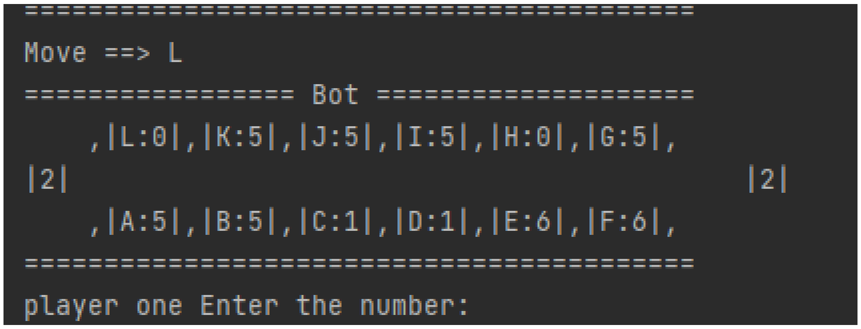
```
player one Enter the number: c
Move ==> C
============ Player One ==============
    ,|L:4|,|K:4|,|J:4|,|I:4|,|H:4|,|G:4|,
|0|                                          |1|
    ,|A:4|,|B:4|,|C:0|,|D:5|,|E:5|,|F:5|,
==========================================
player one Enter the number: 
```



```
player one Enter the number: d
Move ==> D
================ Bot =================
    ,|L:4|,|K:4|,|J:4|,|I:4|,|H:5|,|G:5|,
|0|                                          |2|
    ,|A:4|,|B:4|,|C:0|,|D:0|,|E:6|,|F:6|,
==========================================
```

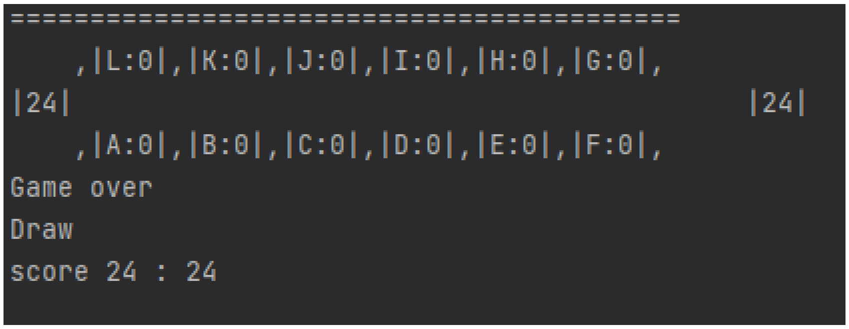


```
Move ==> H
================= Bot ====================
    ,|L:5|,|K:5|,|J:5|,|I:5|,|H:0|,|G:5|,
|1|                                          |2|
    ,|A:4|,|B:4|,|C:0|,|D:0|,|E:6|,|F:6|,
==========================================
```



```
==========================================
Move ==> L
================= Bot ====================
    ,|L:0|,|K:5|,|J:5|,|I:5|,|H:0|,|G:5|,
|2|                                          |2|
    ,|A:5|,|B:5|,|C:1|,|D:1|,|E:6|,|F:6|,
==========================================
player one Enter the number:
```

Until the game is finished and print scores



```
==========================================
    ,|L:0|,|K:0|,|J:0|,|I:0|,|H:0|,|G:0|,
|24|                                          |24|
    ,|A:0|,|B:0|,|C:0|,|D:0|,|E:0|,|F:0|,
Game over
Draw
score 24 : 24
```

Participation:

خالد محمد السيد احمد خليل ابوالليل : Board design code and moving code with or without stealing and code debugging.

عبدالرحمن محمد ايمن مصطفى : save and load and integrating code and debugging and testing stealing and without stealing algorithms.

عبدالرحمن قدرى انور : implementing AI code (alpha beta algorithms)

عاصم كمال احمد : helped in implementing AI code

محمد حماده محمد : implemented integrating code and moving code