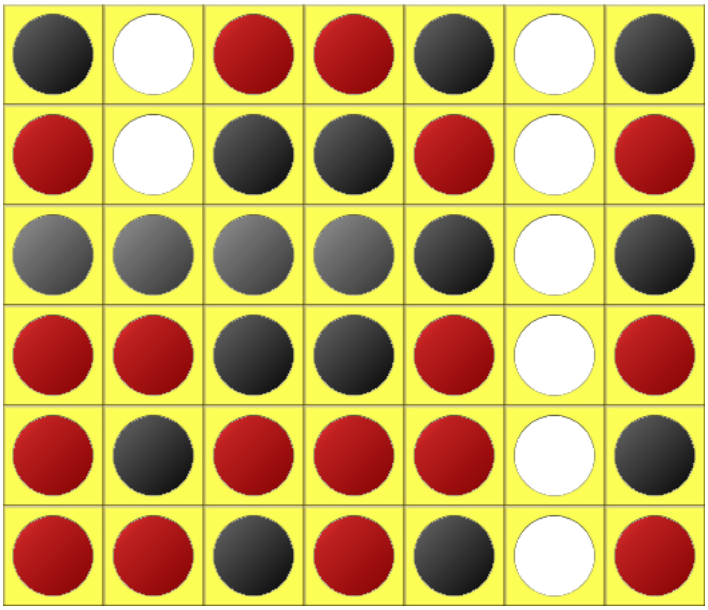
Computer wins.



# CONNECT FOUR

Accept the challenge!


TEAM MEMBERS

Eslam Alaa Zaki
Basma Saeed Ragab
Eslam Medhat Mahrous
Amira Mohamed Fareed


Artificial Intelligence Project


DR. MANAL
TAs. YOMNA AMIN AND AHMED FATHI

# Table of Contents

## Description

A human player plays Connect Four game against the computer.

### Code Implementation

- The code is written in python and it's console based
- The user chooses the difficulty level and who starts the game.
- Minimax algorithm with alpha-beta pruning is used.
- The board is represented by 2d array, the cells may have 1's, 0's, -1's; 1's for computer(us), -1's for user, and 0's for empty.
- User can undo the last move by pressing 'z' or 'Z'.

### Bonus Features

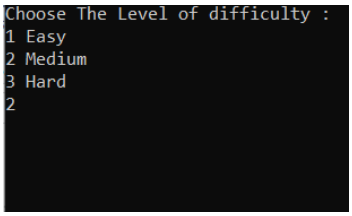
1. Support various difficulty levels corresponding to different game tree depths.
   There are 3 levels of depth difficulties;
   - The first level (easy) of depth=1 checks only the next level states,
   - The second level (medium) of depth=2 checks the next 2 levels states,
   - The third level (hard) of depth=4 checks 2 next levels of user and 2 next level of me(computer).
   - The default level is of depth=2.

## Utility Functions Description

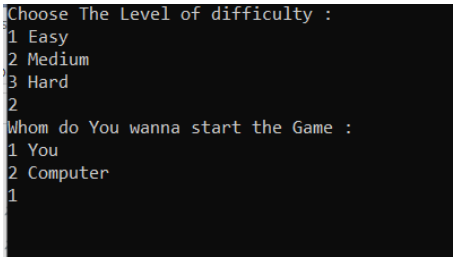
- Utility Function "utility(board)"
  Takes a state (board) and evaluates how good this state is by accumulating its weight.
  It sends to "utilityOfSlice" function slices of the **whole** state, each slice has 4 adjacent cells that may be horizontally, vertically, or diagonally.
  Then accumulates the weights coming from "utilityOfSlice" function to give the state the final weight.
- Utility of Slice Function "utilityOfSlice(slice)"
  Takes a slice of 4 cells and checks the cells if they are blank, have 1's (computer moves) or have any -1's (user moves).
  Then increase or decrease the state weights based on the numbers of 1's, 0's, and -1's as shown in the code.
  Then returns the weight of the slice to the utility function.

## User Guide with Snapshots

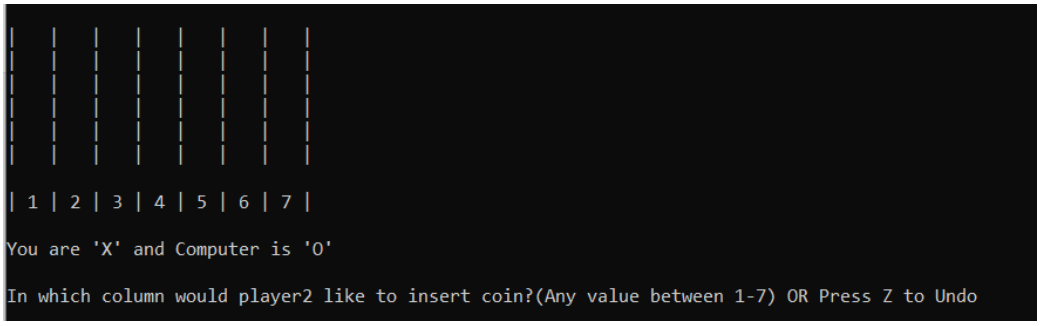
1. Open the “project.exe”.
2. Choose the level of difficulty; easy, medium, or hard.



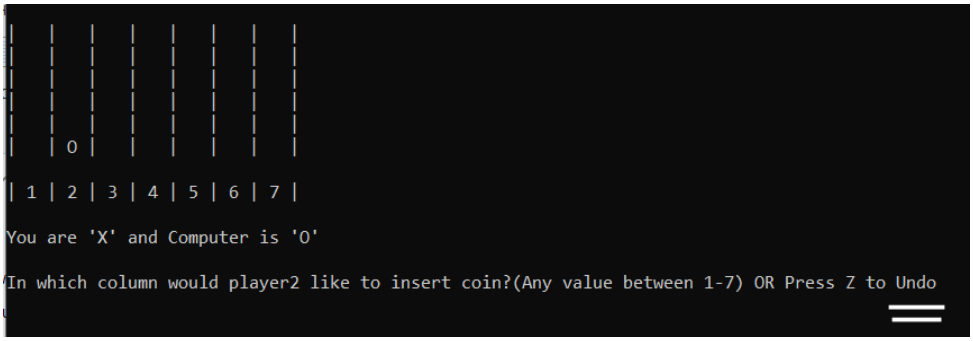

3. Choose who starts the game; you or the computer.



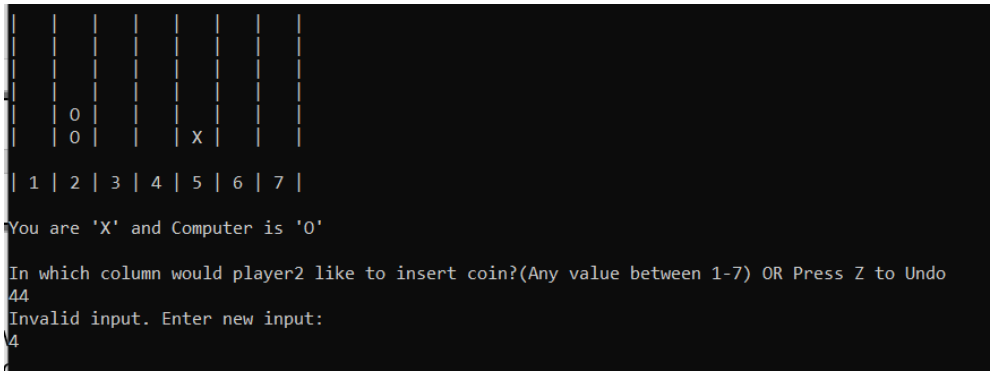

4. Write the number of column you want to play in as shown, you’re X and the computer is O.




5. You can UNDO by pressing ‘z’ or ‘Z’.

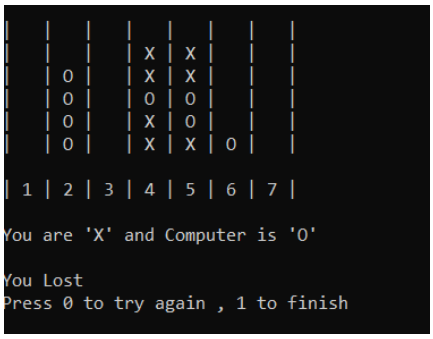


6. If an invalid input is entered “not a digit”, a message will be shown to try again.




7. Play you game and try to win.

8. After finishing press 1 to exit or 0 to play another game.




## Team Work

We didn't assign tasks or specific parts to each one we worked together locally, discussing each part together till we find a solution or the right way to write the function then:

- Eslam Alaa converted the python file to an executable.
- Basma Saeed wrote the Report Documentation.
- Eslam Medhat: wrote the comments in the code to explain the function.
- Amira Fareed recorded the video of the application running.

## Code Documentation

The code has detailed comments of all the functions.

Additional feature: the user can Undo his moves more than once at a time.