INDEX

THE CODE

The code is divided into two files, one containing the game logic and one conatining the UI

alg.py

#the logic for the game

from numpy import array

#from pprint import pprint

from colorama import Style, Fore

from random import randint, choice

LN=3

board=array([[0 for i in range(LN)] for i in range(LN)])

values=array([[0 for i in range(LN)] for i in range(LN)],dtype="float64")

TOLERANCE=0

player\_goes\_first = False

def win(\_board,LN=LN):

d1,d2=0,0

for i in range(LN):

s=sum(\_board[:,i])

if abs(s)==LN:return s//LN

s=sum(\_board[i,:])

if abs(s)==LN:return s//LN

d1+=\_board[i,i]

d2+=\_board[LN-i-1,i]

else:

if abs(d1)==LN:return d1//LN

if abs(d2)==LN:return d2//LN

return 0

def aremovesleft(\_board):

return 0 in board.reshape(-1)

def printb(\_board,v=[" ",f"{Fore.LIGHTGREEN\_EX}O{Fore.RESET}",f"{Fore.LIGHTRED\_EX}X{Fore.RESET}"]):

for i in range(LN):

line=v[\_board[i][0]]

for j in range(1,LN):

line+="|"+v[\_board[i][j]]

print(line)

print("-"\*(2\*LN-1)) if i<LN-1 else 0

def set(x,y,p,LN=LN):

board[x][y]=p

s=TOLERANCE if p==-1 else 1

values[x][y]=0

iv=values.copy()

for j in range(LN):

if (not j==y) and board[x][j]==0:

values[x][j]+=s#s\*(abs(y-j)+1)

for i in range(LN):

if (not i==x) and board[i][y]==0:

values[i][y]+=s#s\*(abs(x-i)+1)

if x==y:

for i in range(LN):

if (not x==i) and board[i][i]==0:

values[i][i]+=s#s\*(abs(x+y-i-j)+1)

if x+y==LN-1:

for i in range(LN):

if (not y==i) and board[LN-i-1][i]==0:

values[LN-i-1][i]+=s#s\*(abs(x-y-i+j)+1)

def findbest(\_values,LN=LN):

#best,bi,bj=-float('inf'),-1,-1

if sum(sum(values))==0:

return randint(0,LN-1),randint(0,LN-1)

mask = \_values == \_values.max()

coords = []

for i in range(LN):

for j in range(LN):

if mask[i][j]:

coords.append((i,j))

return choice(coords)

def main():

moves = 0

if player\_goes\_first:

c=int(input(">> "))

x,y=c//2, c%2

set(x,y,1)

moves += 1

while aremovesleft(board):

i,j=findbest(values)

set(i,j,-1)

if win(board,LN):

printb(board)

print("i win")

return -1, moves

printb(board)

c=int(input(">> "))

x,y=c//3, c%3

if board[x][y]==0:

set(x,y,1)

moves += 1

else:

print("That place is already filled")

continue

if win(board,LN):

printb(board)

print("you win")

return 1, moves

else:

print("draw")

printb(board)

return 0, moves

def reset():

global board, values

board=array([[0 for i in range(LN)] for i in range(LN)])

values=array([[0 for i in range(LN)] for i in range(LN)],dtype="float64")

if \_\_name\_\_ == '\_\_main\_\_':

main()

main.py

#the game user-interface

import alg

from colorama import init

from math import ceil

import mysql.connector as msc

init()

conn = msc.connect(

host="localhost",

user="root",

passwd="password",

database="TicTacToe"

)

if not conn.is\_connected():

conn.connect()

print("reconnecting")

curs = conn.cursor()

curs.execute(

"""create table if not exists scores(

name varchar(50) not null,

score int(3) not null,

win\_time timestamp default now()

)"""

)

def get\_scores(PAGE=1):

curs.execute("select \* from scores order by score DESC, win\_time")

all=curs.fetchall()

print("-"\*81)

print(f"||{'name':^50}|{'score':^5}|{'date and time (UTC)':^20}||")

print("-"\*81)

for name, score, date in all[(PAGE-1)\*10:PAGE\*10]:

print(f"||{name:<50}|{score:>5}|{str(date):^20}||")

print("-"\*81)

print("|{:>79}|".format(f"PAGE {PAGE} of {ceil(len(all)/10)}"))

print("-"\*81)

def insert(score, name=None):

name = name or input("Enter name (less than 50 characters): ")

if name == "":

print("Name cant be empty")

insert(score)

curs.execute(f"insert into scores(name, score) values('{name}', {score})")

conn.commit()

def game():

res, nmoves = alg.main()

input("(press any key to continue)")

if res == 1:

insert(64-nmoves)

elif res == -1:

insert(64-nmoves, "computer")

def main():

alg.player\_goes\_first =eval(open("settings.txt").read())

print("-"\*62)

print("|{:^60}|\n|{:^60}|".format("Welcome to TicTacToe!","(press any key to continue)"))

input("-"\*62+"\n")

while True:

choice = int(input(

"""

Press:-

1. To play the Game

2. To see the HighScores

3. To Quit

4. How to Play

5. Settings

>>> """

))

if choice == 1:

game()

alg.reset()

elif choice == 2:

get\_scores()

if input("Do you want to see another page(Yes/No): ").lower() == "yes":

get\_scores(

int(input("Enter page: "))

)

elif choice == 3:

print("THANKS FOR PLAYING 😄")

break

elif choice == 4:

print(

"""

the computer would play X and you would play O

in front of you,

a board would be printed like this

O| |O

-----

X|X|

-----

X| |

>>>

you have to type the indices of the matrix where you want to place your O

the indices are as follows

0|1|2

-----

3|4|5

-----

6|7|8

like typing

>>> 1

would result in

O|O|O

-----

X|X|

-----

X| |

"""

)

elif choice == 5:

print(f"There is a single setting\nPlayer goes first: {alg.player\_goes\_first}")

if input("Do you wish to flip it(Yes/No)? ").lower() == "yes":

alg.player\_goes\_first = not alg.player\_goes\_first

open("settings.txt","w").write(str(alg.player\_goes\_first))

else:

print("INVALID CHOICE")

input("(press any key to continue)")

input("(press any key to continue)")

if \_\_name\_\_ == '\_\_main\_\_':

main()

bibliography

* NCERT(www.ncert.nic.in)
* Google([www.google.co.in](http://www.google.co.in))