import socket, requests, re, sys

#checks if it is the Server side turn - can go twice occasionally, but cannot fire more than the opponent

def my\_turn():

if check\_whole\_board(my\_board) > check\_whole\_board(your\_board):

return True

if check\_whole\_board(my\_board)< check\_whole\_board(your\_board):

return False

#prints board with specified file

def print\_board(file):

board= open(file, "r")

for i in board:

print(" ".join(i))

board.close()

#processes a shot from a remote client

def shot(x,y):

x=int(x)

y=int(y)

print("fired on "+str(x)+" "+str(y))

board=open(my\_board,"r")

s=board.readlines()

board.close()

if s[y][x]!='\_' and s[y][x]!='X' and s[y][x]!='O':

header="Hit=1"

if s[y][x]=='D':

header=header+("&Sunk=D")

elif s[y][x]=='X':

pass

check=0

for i in range(10):

for j in range(10):

if s[i][j]==s[y][x] and i!=y and j!=x:

check=1

if check==1:

header=header+("&Sunk="+s[y][x])

s[y]=s[y][0:x]+"X"+s[y][x+1:]

board=open(my\_board,"w")

for i in s:

board.write(i)

board.close()

else:

header=("Hit=0")

s[y]=s[y][0:x]+"O"+s[y][x+1:]

board=open(my\_board,"w")

for i in s:

board.write(i)

board.close()

return header

#returns the character of a x-y coordinate in file

def check\_board(file,x,y):

board= open(file, "r")

s=board.readlines()

board.close()

return s[y][x]

#counts total number of fires on a board

def check\_whole\_board(file): #17 hits triggers end game

hit\_count=0

for i in range(10):

for j in range(10):

hit=check\_board(file,i,j)

if hit=='X' or hit=='O':

hit\_count+=1

return hit\_count

#test code

'''HTTP/1.1 201 OK

Location: http://127.0.0.1:8080?hit=0

Date: Sat, 22 Sept 2018

Content-Type: text/html''' #first try at HTTP format

''' using ports to relay info

s=socket.socket(socket.AF\_INET,socket.SOCK\_DGRAM)

s.bind(('127.0.0.1',8080))

data,addr=s.recvfrom(64)

print("Received message: "+data.decode('utf-8')+"'")

print()

'''

#Game Flow

#--------------------

#default file paths - dont need to be used

my\_board = r"C:\Users\Justin\Documents\MSU\_Classes\Networks\own\_board.txt"

your\_board= r"C:\Users\Justin\Documents\MSU\_Classes\Networks\opponent\_board.txt"

#creates empty opponent board ---

your\_b = open(your\_board,"w")

for i in range(10):

your\_b.write("\_\_\_\_\_\_\_\_\_\_\n")

your\_b.close()

def main():

port=int(sys.argv[1])

board=str(sys.argv[2])

my\_board=board

#--------------------------------

s=socket.socket(socket.AF\_INET,socket.SOCK\_STREAM)

s.bind(('192.168.1.4',port))

#while someone hasn't won

while(check\_whole\_board(my\_board)!=17 and check\_whole\_board(your\_board)!=17):

s.listen(1)

conn, addr = s.accept()

data=conn.recv(192)

#gets post arguments

coordinates = re.findall(r'\d+', data.decode('utf-8'))

#print(coordinates)

#print(int(coordinates[0]))

#print(int(coordinates[1]))

x=int(coordinates[0])

y=int(coordinates[1])

#print("Recieved request: "+ data.decode('utf-8')+"'")

board=open(my\_board,"r")

board1=board.readlines()

board.close()

#sends different codes based on errors or successful hits

if(x>9 or y>9 or x<0 or y<0):

respond= 'HTTP/1.1 404 Out Of Bounds'

elif(board1[y][x]=='X' or board1[y][x]=='O'):

respond= 'HTTP/1.1 410 Already Fired'

elif(re.search(r'x=\d&y=\d', data.decode('utf-8'))==None):

respond='HTTP/1.1 400 Bad Request'

elif my\_turn():

respond='''HTTP/1.1 403 OK

not your turn'''

else:

#sends a successful shot

header=shot(coordinates[0],coordinates[1])

response\_proto = 'HTTP/1.1 '

response\_status = '200 '

response\_status\_text = 'OK' # this can be random

respond='''HTTP/1.1 200 OK

'''+(header)

#print(header)

conn.send(respond.encode('utf-8'))

conn.close()

#prints board between every move

print("My Board:")

print\_board(my\_board)

print("My Opponent's Board:")

print\_board(your\_board)

print("My Board:")

print\_board(my\_board)

print("My Opponent's Board:")

print\_board(your\_board)

main()

if(check\_whole\_board(my\_board)!=17):

print('Your Remote Opponent Won :(')

else:

print('You Won :)')