|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования**  **«Московский государственный технический университет имени Н.Э. Баумана**  **(национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)** |

**Факультет «Информатика и системы управления»**

**Кафедра «Системы обработки информации и управления»**

Лахин Никита ИУ5-35Б

Парадигмы и конструкции языков программирования

**ОТЧЁТ ПО**

**Домашнему заданию**

Москва

2023

Задание

1. Выберите язык программирования (который Вы ранее не изучали) и (1) напишите по нему реферат с примерами кода или (2) реализуйте на нем небольшой проект (с детальным текстовым описанием).
2. Реферат (проект) может быть посвящен отдельному аспекту (аспектам) языка или содержать решение какой-либо задачи на этом языке.
3. Необходимо установить на свой компьютер компилятор (интерпретатор, транспилятор) этого языка и произвольную среду разработки.
4. В случае написания реферата необходимо разработать и откомпилировать примеры кода (или модифицировать стандартные примеры).
5. В случае создания проекта необходимо детально комментировать код.
6. При написании реферата (создании проекта) необходимо изучить и корректно использовать особенности парадигмы языка и основных конструкций данного языка.
7. Приветствуется написание черновика статьи по результатам выполнения ДЗ. Черновик статьи может быть подготовлен группой студентов, которые исследовали один и тот же аспект в нескольких языках или решили одинаковую задачу на нескольких языках.

Задача

Задача заключается в разработке программы для автоматического обучения игры в крестики-нолики. Программа должна способна играть самостоятельно против себя, используя методы машинного обучения для улучшения своих стратегий и принятия оптимальных решений в игровом процессе.

Структура программы

* AI\_VS\_AI.py — программа с помощью которой AI учиться играть
* Art\_Intel.py — программа, которая запускает окно приложения для игры против обученной AI
* memory.txt, memory2.txt — файлы c оценкой ходов

Текст программы

## AI\_VS\_AI.py

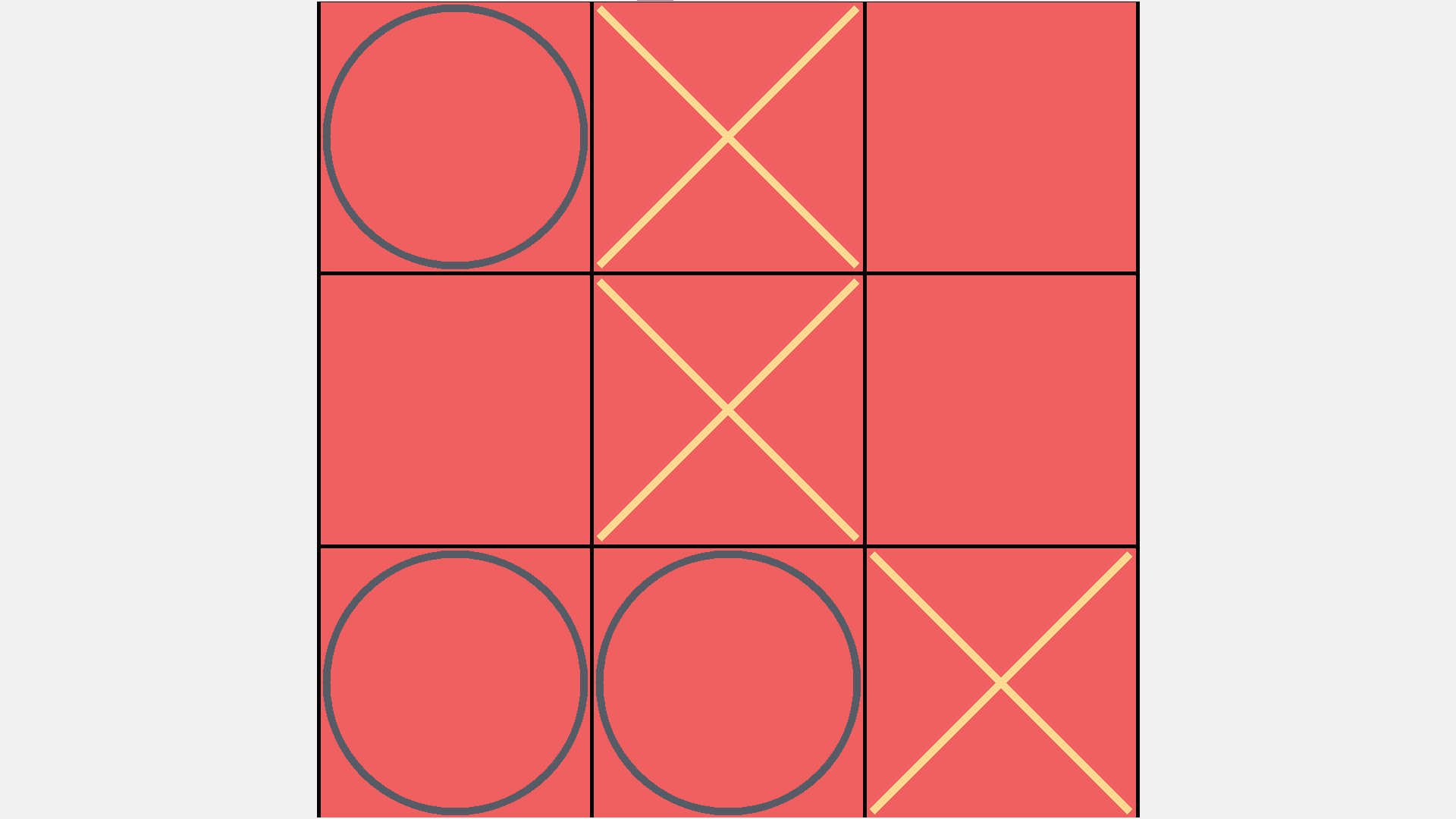
from random import randint  
import os  
#from time import time  
#open("Results.txt","w").close()  
results = [""]\*10  
while True:  
 #ntime = time()  
 # - open memory  
 try:  
 file = open("memory.txt","r")  
 lie = file.readlines()  
 file.close()  
 except:  
 open("memory.txt","w").close()  
 lie = []  
   
 try:  
 file = open("memory2.txt","r")  
 lie2 = file.readlines()  
 file.close()  
 except:  
 open("memory2.txt","w").close()  
 lie2 = []  
   
 field = ["000","000","000"]  
 def in\_field(x,y,symbol):  
 global field  
 field[y] = field[y][:x]+symbol+field[y][x+1:]  
   
 turns = []  
 turns2 = []  
 turn = 0  
 end = 0  
   
 while end != 1:  
 if end == 0:  
 nlies = []  
 nbest = ["00",0]  
 best\_num = 0  
 for i in range(0,len(lie2),5):  
 find = 1  
 for c in range(3):  
 if lie2[i+c][:-1] != field[c]:  
 find = 0  
   
 if find == 1:  
 try:  
 if nbest[1] < int(lie2[i+4]):  
 nbest = [lie2[i+3][:-1],int(lie2[i+4])]  
 best\_num = i+4  
 nlies += [[lie2[i+3][:-1],int(lie2[i+4]),i+4]]  
 except:  
 if nbest[1] < int(lie2[i+4][:-1]):  
 nbest = [lie2[i+3][:-1],int(lie2[i+4][:-1])]  
 best\_num = i+4  
 nlies += [[lie2[i+3][:-1],int(lie2[i+4][:-1]),i+4]]  
   
 if nbest[1] > 50:  
 in\_field(int(nbest[0][0]),int(nbest[0][1]),"1")  
 turns2 += [best\_num]  
 else:  
 if len(nlies) < 9-turn\*2:  
 # - random "0"  
 num = randint(1,9-turn\*2)  
 nnum = 0  
 for i in range(3):  
 for c in range(3):  
 if field[i][c] == "0":  
 nnum += 1  
 if nnum == num:  
 nx = c  
 ny = i  
 nnum += 1  
 find = -1  
 for i in range(len(nlies)):  
 if nlies[i][0] == str(nx)+str(ny):  
 find = i  
   
 if find == -1: # - if turn is unique  
 # - write in memory  
 file = open("memory2.txt","a")  
 file.write(field[0]+"\n"+field[1]+"\n"+field[2]+"\n"+str(nx)+str(ny)+"\n50\n")  
 file.close()  
 file = open("memory2.txt","r")  
 lie2 = file.readlines()  
 file.close()  
 turns2 += [len(lie2)-1]  
 else:  
 turns2 += [nlies[find][2]]  
 in\_field(nx,ny,"1")  
 else:  
 best\_nlies = []  
 for i in range(len(nlies)):  
 if nlies[i][1] == nbest[1]:  
 best\_nlies += [nlies[i]]  
 if len(best\_nlies)-1 != 0:  
 num = randint(0,len(best\_nlies)-1)  
 else:  
 num = 0  
 in\_field(int(best\_nlies[num][0][0]),int(best\_nlies[num][0][1]),"1")  
 turns2 += [best\_nlies[num][2]]  
   
 if field[0] == "111" or field[1] == "111" or field[2] == "111" or (field[0][0] == "1" and field[1][0] == "1" and field[2][0] == "1") or (field[0][1] == "1" and field[1][1] == "1" and field[2][1] == "1") or (field[0][2] == "1" and field[1][2] == "1" and field[2][2] == "1") or (field[0][0] == "1" and field[1][1] == "1" and field[2][2] == "1") or (field[0][2] == "1" and field[1][1] == "1" and field[2][0] == "1"):  
 file = open("memory.txt","r")  
 lie = file.readlines()  
 file.close()  
 for i in range(len(turns)):  
 if i != len(turns)-1:  
 if int(lie[turns[i]]) != 1:  
 lie[turns[i]] = str(int(lie[turns[i]])-1)+"\n"  
 else:  
 lie[turns[i]] = "0\n"  
 file = open("memory.txt","w")  
 for i in range(len(lie)):  
 file.write(lie[i])  
 file.close()  
 turns = []  
   
 file = open("memory2.txt","r")  
 lie2 = file.readlines()  
 file.close()  
 for i in range(len(turns2)):  
 if i != len(turns2)-1:  
 if int(lie2[turns2[i]]) != 99:  
 lie2[turns2[i]] = str(int(lie2[turns2[i]])+1)+"\n"  
 else:  
 lie2[turns2[i]] = "100\n"  
 file = open("memory2.txt","w")  
 for i in range(len(lie2)):  
 file.write(lie2[i])  
 file.close()  
 turns2 = []  
 turn = -1  
 results = ["FIRST WIN"]+results[:-1]  
 #file = open("Results.txt","a")  
 #file.write("FIRST\n")  
 #file.close()  
 end = 1  
   
 if turn == 4 and end == 0:  
 file = open("memory.txt","r")  
 lie = file.readlines()  
 file.close()  
 for i in range(len(turns)):  
 if int(lie[turns[i]]) != 99:  
 lie[turns[i]] = str(int(lie[turns[i]])+1)+"\n"  
   
 file = open("memory.txt","w")  
 for i in range(len(lie)):  
 file.write(lie[i])  
 file.close()  
 turns = []  
   
 file = open("memory2.txt","r")  
 lie2 = file.readlines()  
 file.close()  
 for i in range(len(turns2)):  
 if i != len(turns2)-1:  
 if int(lie2[turns2[i]]) != 1:  
 lie2[turns2[i]] = str(int(lie2[turns2[i]])-1)+"\n"  
   
 file = open("memory2.txt","w")  
 for i in range(len(lie2)):  
 file.write(lie2[i])  
 file.close()  
 turns2 = []   
 turn = -1  
 results = ["DRAFT"]+results[:-1]  
 #file = open("Results.txt","a")  
 #file.write("DRAFT\n")  
 #file.close()  
 end = 1  
   
 if end == 0:  
 nlies = []  
 nbest = ["00",0]  
 best\_num = 0  
 for i in range(0,len(lie),5):  
 find = 1  
 for c in range(3):  
 if lie[i+c][:-1] != field[c]:  
 find = 0  
   
 if find == 1:  
 try:  
 if nbest[1] < int(lie[i+4]):  
 nbest = [lie[i+3][:-1],int(lie[i+4])]  
 best\_num = i+4  
 nlies += [[lie[i+3][:-1],int(lie[i+4]),i+4]]  
 except:  
 if nbest[1] < int(lie[i+4][:-1]):  
 nbest = [lie[i+3][:-1],int(lie[i+4][:-1])]  
 best\_num = i+4  
 nlies += [[lie[i+3][:-1],int(lie[i+4][:-1]),i+4]]  
   
 if nbest[1] > 50:  
 in\_field(int(nbest[0][0]),int(nbest[0][1]),"2")  
 turns += [best\_num]  
 else:  
 if len(nlies) < 8-turn\*2:  
 # - random "0"  
 num = randint(1,8-turn\*2)  
 nnum = 0  
 for i in range(3):  
 for c in range(3):  
 if field[i][c] == "0":  
 nnum += 1  
 if nnum == num:  
 nx = c  
 ny = i  
 nnum += 1  
 find = -1  
 for i in range(len(nlies)):  
 if nlies[i][0] == str(nx)+str(ny):  
 find = i  
   
 if find == -1: # - if turn is unique  
 # - write in memory  
 file = open("memory.txt","a")  
 file.write(field[0]+"\n"+field[1]+"\n"+field[2]+"\n"+str(nx)+str(ny)+"\n50\n")  
 file.close()  
 file = open("memory.txt","r")  
 lie = file.readlines()  
 file.close()  
 turns += [len(lie)-1]  
 else:  
 turns += [nlies[find][2]]  
 in\_field(nx,ny,"2")  
 else:  
 best\_nlies = []  
 for i in range(len(nlies)):  
 if nlies[i][1] == nbest[1]:  
 best\_nlies += [nlies[i]]  
 if len(best\_nlies)-1 != 0:  
 num = randint(0,len(best\_nlies)-1)  
 else:  
 num = 0  
 in\_field(int(best\_nlies[num][0][0]),int(best\_nlies[num][0][1]),"2")  
 turns += [best\_nlies[num][2]]  
   
 if field[0] == "222" or field[1] == "222" or field[2] == "222" or (field[0][0] == "2" and field[1][0] == "2" and field[2][0] == "2") or (field[0][1] == "2" and field[1][1] == "2" and field[2][1] == "2") or (field[0][2] == "2" and field[1][2] == "2" and field[2][2] == "2") or (field[0][0] == "2" and field[1][1] == "2" and field[2][2] == "2") or (field[0][2] == "2" and field[1][1] == "2" and field[2][0] == "2"):  
 file = open("memory.txt","r")  
 lie = file.readlines()  
 file.close()  
 for i in range(len(turns)):  
 if i != len(turns)-1:  
 if int(lie[turns[i]]) != 99:  
 lie[turns[i]] = str(int(lie[turns[i]])+1)+"\n"  
 else:  
 lie[turns[i]] = "100\n"  
 file = open("memory.txt","w")  
 for i in range(len(lie)):  
 file.write(lie[i])  
 file.close()  
   
 file = open("memory2.txt","r")  
 lie2 = file.readlines()  
 file.close()  
 for i in range(len(turns2)):  
 if i != len(turns2)-1:  
 if int(lie2[turns2[i]]) != 1:  
 lie2[turns2[i]] = str(int(lie2[turns2[i]])-1)+"\n"  
 else:  
 lie2[turns2[i]] = "0\n"  
 file = open("memory2.txt","w")  
 for i in range(len(lie2)):  
 file.write(lie2[i])  
 file.close()  
 turns2 = []   
 turns = []  
 turn = -1  
 results = ["SECOND WIN"]+results[:-1]  
 end = 1  
 #file = open("Results.txt","a")  
 #file.write("SECOND\n")  
 #file.close()  
 turn += 1  
 os.system("cls")  
 print("\_\_\_\_\_\_\_\_\_\_")  
 for i in range(len(results)):  
 print(results[i])  
 #print(str((time()-ntime)//0.01\*0.01))

## Art\_Intel.py

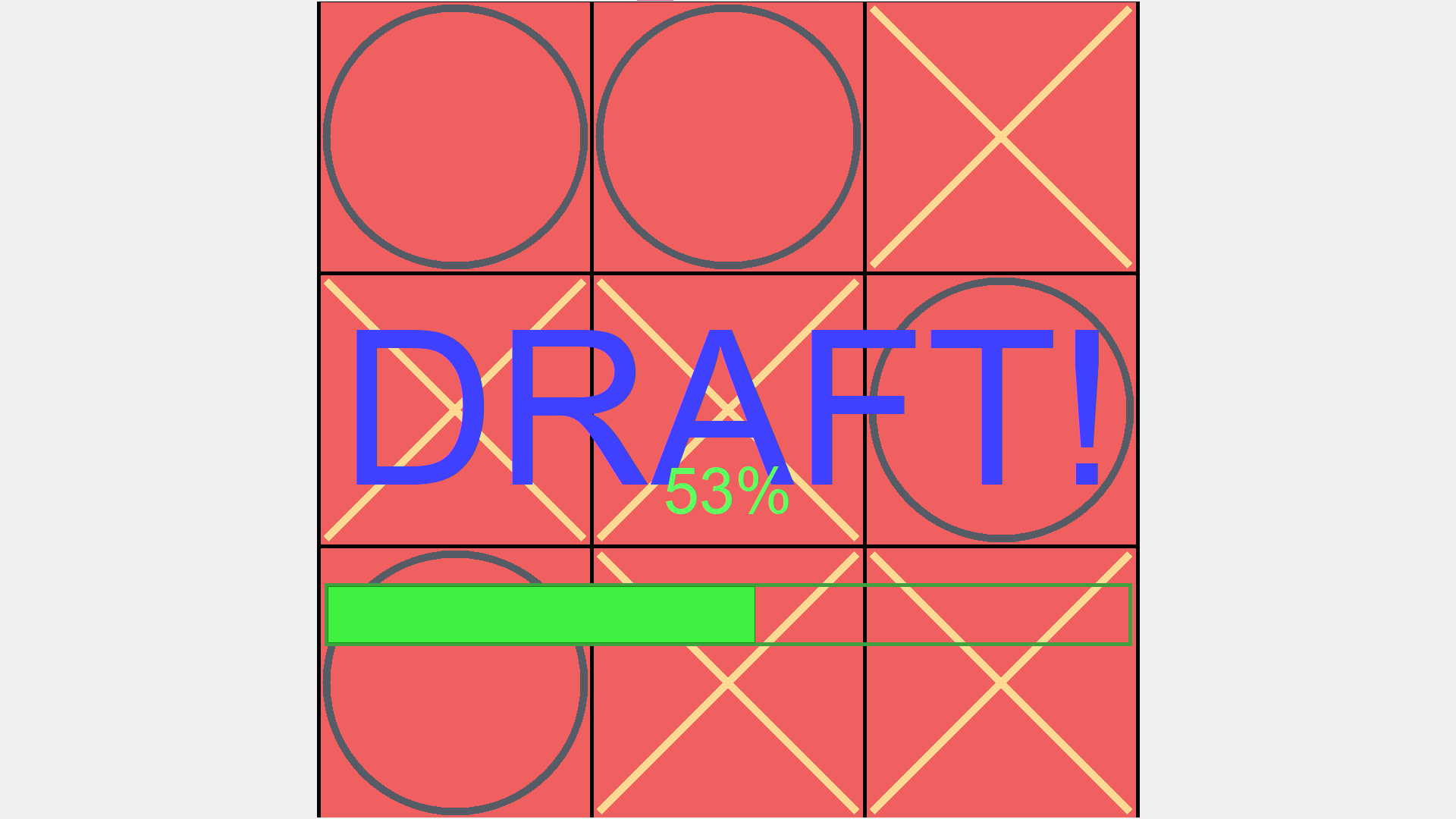
from random import randint  
from tkinter import \*  
import time  
root = Tk()  
root.geometry("300x4100")  
root.attributes('-fullscreen', True)  
xw = root.winfo\_screenwidth()  
yw = root.winfo\_screenheight()  
print("Screen width:", xw)  
print("Screen height:", yw)  
canvas = Canvas(root, width=xw, height=yw)  
canvas.pack()  
bg = "#F06060"  
turn\_time = 30  
turns = []  
turn = 0  
your\_turn = 0  
x,y = "",""  
result = ""  
end = 0  
  
# - open memory  
try:  
 file = open("memory.txt","r")  
 lie = file.readlines()  
 file.close()  
except:  
 open("memory.txt","w").close()  
 lie = []  
   
field = ["000","000","000"]  
  
def in\_field(x,y,symbol):  
 global field  
 field[y] = field[y][:x]+symbol+field[y][x+1:]  
   
def tic(event):  
 global your\_turn,x,y  
 if your\_turn == 1:  
 gx = event.x\_root  
 gy = event.y\_root  
 if xw/2-yw/2 < gx < xw/2+yw/2:  
 x = str(int((gx-xw/2+yw/2)//(yw/3)))  
 y = str(int(gy//(yw/3)))  
 your\_turn = 0  
  
def krest(x1,y1,x2,y2):  
 canvas.create\_line(x1+10,y1+10,x2-10,y2-10, width = 10, fill = "#FFDA93")  
 canvas.create\_line(x2-10,y1+10,x1+10,y2-10, width = 10, fill = "#FFDA93")  
  
def zero(x1,y1,x2,y2):  
 canvas.create\_oval(x1+10,y1+10,x2-10,y2-10, width = 10, outline = "#555C66")  
  
root.bind("<Button-1>", tic)  
while True:  
 if end != 0:  
 end += 1  
 x,y = "",""  
 your\_turn = 1  
 if end > 0:  
 your\_turn = 0  
 if end == turn\_time:  
 field = ["000","000","000"]  
 end = 0   
 canvas.delete("all")  
 canvas.create\_rectangle(xw/2-yw/2,0, xw/2+yw/2,yw, fill = bg, width = 5)  
 canvas.create\_line(xw/2-yw/2+yw/3,0, xw/2-yw/2+yw/3,yw, width = 5)  
 canvas.create\_line(xw/2-yw/2+yw/3\*2,0, xw/2-yw/2+yw/3\*2,yw, width = 5)  
 canvas.create\_line(xw/2-yw/2,yw/3, xw/2+yw/2,yw/3, width = 5)  
 canvas.create\_line(xw/2-yw/2,yw/3\*2, xw/2+yw/2,yw/3\*2, width = 5)  
 for i in range(3):  
 for c in range(3):  
 if field[c][i] == "1":  
 krest(xw/2-yw/2+i\*yw/3, c\*yw/3, xw/2-yw/2+(i+1)\*yw/3, (c+1)\*yw/3)  
 elif field[c][i] == "2":  
 zero(xw/2-yw/2+i\*yw/3, c\*yw/3, xw/2-yw/2+(i+1)\*yw/3, (c+1)\*yw/3)  
 if end > 0:  
 canvas.create\_text(xw/2,yw/2, text = result, fill = "#4040FF", font = ("Areal",xw//9))  
 canvas.create\_text(xw/2,yw/5\*3, text = str(round(end/turn\_time\*100))+"%", fill = "#60FF60", font = ("Areal",xw//30))  
 canvas.create\_rectangle(xw/2-yw/2+10,yw/7\*5, xw/2+yw/2-10,yw/7\*5.5, width = 5, outline = "#40A040")  
 canvas.create\_rectangle(xw/2-yw/2+12,yw/7\*5+2, xw/2-yw/2+12+(yw-24)/turn\_time\*end,yw/7\*5.5-2, width = 1, outline = "#109010", fill = "#40F040")  
 canvas.update()  
  
 if x != "" and y != "" and x.isalnum() and y.isalnum() and 0<=int(x)<=2 and 0<=int(y)<=2 and field[int(y)][int(x)] == "0":  
 your\_turn = 0  
 x = int(x)  
 y = int(y)   
 in\_field(x,y,"1")  
 if field[0] == "111" or field[1] == "111" or field[2] == "111" or (field[0][0] == "1" and field[1][0] == "1" and field[2][0] == "1") or (field[0][1] == "1" and field[1][1] == "1" and field[2][1] == "1") or (field[0][2] == "1" and field[1][2] == "1" and field[2][2] == "1") or (field[0][0] == "1" and field[1][1] == "1" and field[2][2] == "1") or (field[0][2] == "1" and field[1][1] == "1" and field[2][0] == "1"):  
 file = open("memory.txt","r")  
 lie = file.readlines()  
 file.close()  
 for i in range(len(turns)):  
 if i != len(turns)-1:  
 if int(lie[turns[i]]) != 1:  
 lie[turns[i]] = str(int(lie[turns[i]])-1)+"\n"  
 else:  
 lie[turns[i]] = "0\n"  
 file = open("memory.txt","w")  
 for i in range(len(lie)):  
 file.write(lie[i])  
 file.close()  
 turns = []  
 turn = -1  
 result = "You WIN!"  
 end = 1  
   
 if turn == 4 and end == 0:  
 file = open("memory.txt","r")  
 lie = file.readlines()  
 file.close()  
 for i in range(len(turns)):  
 if int(lie[turns[i]]) != 99:  
 lie[turns[i]] = str(int(lie[turns[i]])+1)+"\n"  
   
 file = open("memory.txt","w")  
 for i in range(len(lie)):  
 file.write(lie[i])  
 file.close()  
 turns = []  
 turn = -1  
 result = "DRAFT!"  
 end = 1  
   
 if end == 0:  
 nlies = []  
 nbest = ["00",0]  
 best\_num = 0  
 for i in range(0,len(lie),5):  
 find = 1  
 for c in range(3):  
 if lie[i+c][:-1] != field[c]:  
 find = 0  
   
 if find == 1:  
 try:  
 if nbest[1] < int(lie[i+4]):  
 nbest = [lie[i+3][:-1],int(lie[i+4])]  
 best\_num = i+4  
 nlies += [[lie[i+3][:-1],int(lie[i+4]),i+4]]  
 except:  
 if nbest[1] < int(lie[i+4][:-1]):  
 nbest = [lie[i+3][:-1],int(lie[i+4][:-1])]  
 best\_num = i+4  
 nlies += [[lie[i+3][:-1],int(lie[i+4][:-1]),i+4]]  
   
 if nbest[1] > 50:  
 in\_field(int(nbest[0][0]),int(nbest[0][1]),"2")  
 turns += [best\_num]  
 else:  
 if len(nlies) < 8-turn\*2:  
 # - random "0"  
 num = randint(1,8-turn\*2)  
 nnum = 0  
 for i in range(3):  
 for c in range(3):  
 if field[i][c] == "0":  
 nnum += 1  
 if nnum == num:  
 nx = c  
 ny = i  
 nnum += 1  
 find = -1  
 for i in range(len(nlies)):  
 if nlies[i][0] == str(nx)+str(ny):  
 find = i  
   
 if find == -1: # - if turn is unique  
 # - write in memory  
 file = open("memory.txt","a")  
 file.write(field[0]+"\n"+field[1]+"\n"+field[2]+"\n"+str(nx)+str(ny)+"\n50\n")  
 file.close()  
 file = open("memory.txt","r")  
 lie = file.readlines()  
 file.close()  
 turns += [len(lie)-1]  
 else:  
 turns += [nlies[find][2]]  
 in\_field(nx,ny,"2")  
 else:  
 best\_nlies = []  
 for i in range(len(nlies)):  
 if nlies[i][1] == nbest[1]:  
 best\_nlies += [nlies[i]]  
 if len(best\_nlies)-1 != 0:  
 num = randint(0,len(best\_nlies)-1)  
 else:  
 num = 0  
 in\_field(int(best\_nlies[num][0][0]),int(best\_nlies[num][0][1]),"2")  
 turns += [best\_nlies[num][2]]  
   
 if field[0] == "222" or field[1] == "222" or field[2] == "222" or (field[0][0] == "2" and field[1][0] == "2" and field[2][0] == "2") or (field[0][1] == "2" and field[1][1] == "2" and field[2][1] == "2") or (field[0][2] == "2" and field[1][2] == "2" and field[2][2] == "2") or (field[0][0] == "2" and field[1][1] == "2" and field[2][2] == "2") or (field[0][2] == "2" and field[1][1] == "2" and field[2][0] == "2"):  
 file = open("memory.txt","r")  
 lie = file.readlines()  
 file.close()  
 for i in range(len(turns)):  
 if i != len(turns)-1:  
 if int(lie[turns[i]]) != 99:  
 lie[turns[i]] = str(int(lie[turns[i]])+1)+"\n"  
 else:  
 lie[turns[i]] = "100\n"  
 file = open("memory.txt","w")  
 for i in range(len(lie)):  
 file.write(lie[i])  
 file.close()  
 turns = []  
 turn = -1  
 result = "You LOSE!"  
 end = 1  
 ## - print field  
 #for i in range(3):  
 #print(field[i])  
 turn += 1  
   
 time.sleep(0.01)  
   
root.mainloop()

Экранные формы

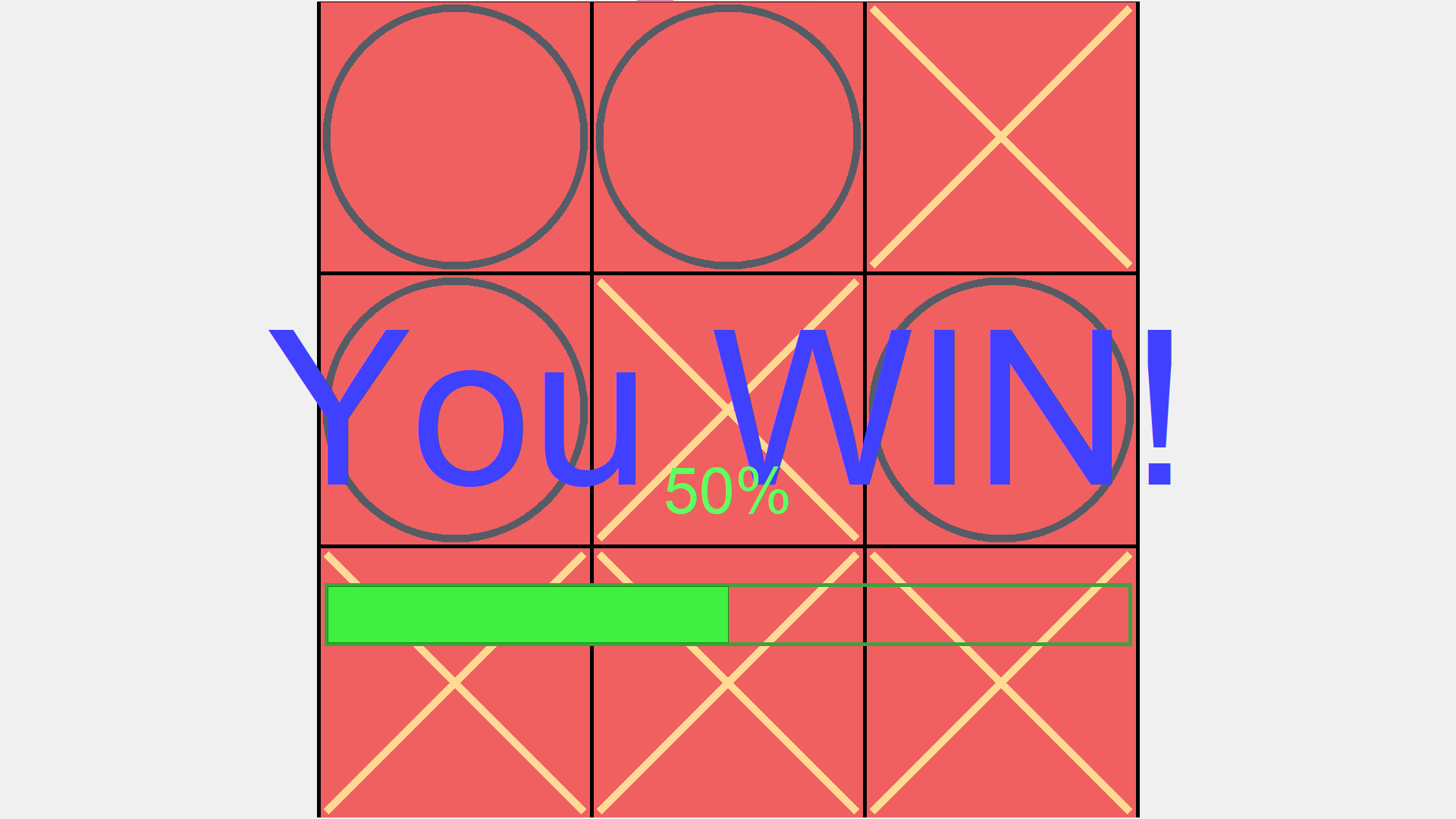
Окно приложения:



Результат игры ничья:



Результат игры победа:



Результат игры поражения:

