from tkinter import \*

from tkinter import messagebox

import sys

import os

import random

import tkinter.messagebox

root = Tk()

root.resizable(width=False, height=False)

root.geometry('1000x750')

root.configure(background='green')

root.title("jogo de damas")

logo = PhotoImage(file="caixa branca.gif")

logo2 = PhotoImage(file="lado vermelho.gif")

logo3 = PhotoImage(file="vermelho.gif")

logo4 = PhotoImage(file="lado azul.gif")

logo5 = PhotoImage(file="lado verde.gif")

logo6 = PhotoImage(file="lado amarelo.gif")

logo7 = PhotoImage(file="centro.gif")

logoxx = PhotoImage(file="teste.gif")

logog = PhotoImage(file="caixa verde.gif")

logogs = PhotoImage(file="parada verde.gif")

logoy = PhotoImage(file="caixa amarela.gif")

logoys = PhotoImage(file="parada amarela.gif")

logob = PhotoImage(file="caixa azul.gif")

logobs = PhotoImage(file="parada azul.gif")

logor = PhotoImage(file="caixa vermelho.gif")

logors = PhotoImage(file="parada vermelha.gif")

logoh = PhotoImage(file="cabeça.gif")

logot = PhotoImage(file="cauda.gif")

logoh1 = PhotoImage(file="cabeça1.gif")

logot1 = PhotoImage(file="cauda1.gif")

logoh2 = PhotoImage(file="cabeça2.gif")

logot2 = PhotoImage(file="cauda2.gif")

logoh3 = PhotoImage(file="cabeça 3.gif")

logot3 = PhotoImage(file="cauda3.gif")

logoab= PhotoImage(file="azul.gif")

logoay= PhotoImage(file="amarelo.gif")

logoag= PhotoImage(file="verde.gif")

Label(image=logo2, width=298, height=298).place(x=-1, y=-1)

Label(image=logo4, width=300, height=300).place(x=(-2), y=(448))

Label(image=logo5, width=296, height=296).place(x=(450), y=(0))

Label(image=logo6, width=294, height=294).place(x=(450), y=(450))

Label(image=logo7, width=150, height=150).place(x=(298), y=(298))

c = 0

lx = 0

bb =0

nc = 0

rollc = 0

rolls = []

RED = True

BLUE = False

GREEN = False

YELLOW = False

TURN = True

REDKILL = False

BLUEKILL = False

GREENKILL = False

YELLOWKILL = False

def board():

tkinter.messagebox.showinfo(title=None, message="PARA INICIAR O JOGO, PRESSIONE OK E PARA SAIR, PRESSIONE CROSS UP NA JANELA")

v = 0

z = 0

while (v != 300):

z = 0

while (z != 150):

Label(image=logo, width=46, height=46).place(x=(300 + z), y=(0 + v))

z = z + 50

v = v + 50

z = 0

v = 0

while (v != 300):

z = 0

while (z != 150):

Label(image=logo, width=46, height=46).place(x=(0 + v), y=(300 + z))

z = z + 50

v = v + 50

v = 0

z = 0

while (v != 300):

z = 0

while (z != 150):

Label(image=logo, width=46, height=46).place(x=(300 + z), y=(450 + v))

z = z + 50

v = v + 50

z = 0

v = 0

while (v != 300):

z = 0

while (z != 150):

Label(image=logo, width=46, height=46).place(x=(450 + v), y=(300 + z))

z = z + 50

v = v + 50

v = 0

while (v != 250):

Label(image=logog, width=46, height=46).place(x=(350), y=(50 + v))

v = v + 50

Label(image=logog, width=46, height=46).place(x=(300), y=(100))

Label(image=logogs, width=46, height=46).place(x=(400), y=(50))

v = 0

while (v != 250):

Label(image=logoy, width=46, height=46).place(x=(450 + v), y=(350))

v = v + 50

Label(image=logoy, width=46, height=46).place(x=(600), y=(300))

Label(image=logoys, width=46, height=46).place(x=(650), y=(400))

v = 0

while (v != 250):

Label(image=logor, width=46, height=46).place(x=(50 + v), y=(350))

v = v + 50

Label(image=logor, width=46, height=46).place(x=(100), y=(400))

Label(image=logors, width=46, height=46).place(x=(50), y=(300))

v = 0

while (v != 250):

Label(image=logob, width=46, height=46).place(x=(350), y=(450 + v))

v = v + 50

Label(image=logobs, width=46, height=46).place(x=(300), y=(650))

Label(image=logob, width=46, height=46).place(x=(400), y=(600))

Label(image=logoh, width=46, height=46).place(x=250, y=400)

Label(image=logot, width=46, height=46).place(x=300, y=450)

Label(image=logoh1, width=46, height=46).place(x=400, y=450)

Label(image=logot1, width=46, height=46).place(x=450, y=400)

Label(image=logoh2, width=46, height=46).place(x=450, y=300)

Label(image=logot2, width=46, height=46).place(x=400, y=250)

Label(image=logoh3, width=46, height=46).place(x=300, y=250)

Label(image=logot3, width=46, height=46).place(x=250, y=300)

class YBox:

rap = None

def \_\_init\_\_(self, num=-1, x=0, y=0, x0=0, y0=0, double=False, ):

self.num = num

self.x = x

self.y = y

self.x0 = x0

self.y0 = y0

self.rap = Label(image=logoay, width=20, height=20)

self.double = double

def swap(self):

self.rap.place(x=self.x0 + 13, y=self.y0 + 14)

class GBox:

rap = None

def \_\_init\_\_(self, num=-1, x=0, y=0, x0=0, y0=0, double=False, ):

self.num = num

self.x = x

self.y = y

self.x0 = x0

self.y0 = y0

self.rap = Label(image=logoag, width=20, height=20)

self.double = double

def swap(self):

self.rap.place(x=self.x0 + 13, y=self.y0 + 14)

class BBox:

rap = None

def \_\_init\_\_(self, num=-1, x=0, y=0, x0=0, y0=0, double=False, ):

self.num = num

self.x = x

self.y = y

self.x0 = x0

self.y0 = y0

self.rap = Label(image=logoab, width=20, height=20)

self.double = double

def swap(self):

self.rap.place(x=self.x0 + 13, y=self.y0 + 14)

class Box:

rap = None

def \_\_init\_\_(self, num=-1, x=0, y=0, x0=0, y0=0, double=False, ):

self.num = num

self.x = x

self.y = y

self.x0 = x0

self.y0 = y0

self.rap = Label(image=logo3, width=20, height=20)

self.double = double

def swap(self):

self.rap.place(x=self.x0 + 13, y=self.y0 + 14)

def main():

global box, redbox, bluebox, greenbox, yellowbox, redhome, bluehome, yellowhome, greenhome

global red, blue, yellow, green, rap, RED, BLUE, GREEN, YELLOW, dice, nc, TURN, bb

if c == 0:

board()

box = [Box() for i in range(52)]

redbox = [Box() for i in range(57)]

bluebox = [Box() for i in range(57)]

greenbox = [Box() for i in range(57)]

yellowbox = [Box() for i in range(57)]

redhome = [Box() for i in range(4)]

bluehome = [Box() for i in range(4)]

greenhome = [Box() for i in range(4)]

yellowhome = [Box() for i in range(4)]

red = [Box() for i in range(4)]

blue = [BBox() for i in range(4)]

green = [GBox() for i in range(4)]

yellow = [YBox() for i in range(4)]

for i in range(2):

redhome[i].x = (100 + (100 \* i))

redhome[i].y = 100

red[i].x0 = redhome[i].x

red[i].y0 = redhome[i].y

red[i].x = (red[i].x0) + 25

red[i].y = (red[i].y0) + 25

bluehome[i].x = (100 + (100 \* i))

bluehome[i].y = (550)

blue[i].x0 = bluehome[i].x

blue[i].y0 = bluehome[i].y

blue[i].x = (blue[i].x0) + 25

blue[i].y = (blue[i].y0) + 25

yellowhome[i].x = (550 + (100 \* i))

yellowhome[i].y = (550)

yellow[i].x0 = yellowhome[i].x

yellow[i].y0 = yellowhome[i].y

yellow[i].x = (yellow[i].x0) + 25

yellow[i].y = (yellow[i].y0) + 25

greenhome[i].x = (550 + (100 \* i))

greenhome[i].y = (100)

green[i].x0 = greenhome[i].x

green[i].y0 = greenhome[i].y

green[i].x = (green[i].x0) + 25

green[i].y = (green[i].y0) + 25

for i in range(2, 4):

redhome[i].x = (100 + (100 \* (i - 2)))

redhome[i].y = 200

red[i].x0 = redhome[i].x

red[i].y0 = redhome[i].y

red[i].x = (red[i].x0) + 25

red[i].y = (red[i].y0) + 25

bluehome[i].x = (100 + (100 \* (i - 2)))

bluehome[i].y = (650)

blue[i].x0 = bluehome[i].x

blue[i].y0 = bluehome[i].y

blue[i].x = (blue[i].x0) + 25

blue[i].y = (blue[i].y0) + 25

yellowhome[i].x = (550 + (100 \* (i - 2)))

yellowhome[i].y = (650)

yellow[i].x0 = yellowhome[i].x

yellow[i].y0 = yellowhome[i].y

yellow[i].x = (yellow[i].x0) + 25

yellow[i].y = (yellow[i].y0) + 25

greenhome[i].x = (550 + (100 \* (i - 2)))

greenhome[i].y = 200

green[i].x0 = greenhome[i].x

green[i].y0 = greenhome[i].y

green[i].x = (green[i].x0) + 25

green[i].y = (green[i].y0) + 25

for i in range(6):

box[i].x = 300

box[i].y = (700 - (50 \* i))

for i in range(6, 12):

box[i].x = (250 - (50 \* (i - 6)))

box[i].y = (400)

box[12].x = 0

box[12].y = 350

for i in range(13, 19):

box[i].x = (0 + (50 \* (i - 13)))

box[i].y = (300)

for i in range(19, 25):

box[i].x = (300)

box[i].y = (250 - (50 \* (i - 19)))

box[25].x = 350

box[25].y = 0

for i in range(26, 32):

box[i].x = (400)

box[i].y = (0 + (50 \* (i - 26)))

for i in range(32, 38):

box[i].x = (450 + (50 \* (i - 32)))

box[i].y = (300)

box[38].x = 700

box[38].y = 350

for i in range(39, 45):

box[i].x = (700 - (50 \* (i - 39)))

box[i].y = (400)

for i in range(45, 51):

box[i].x = (400)

box[i].y = (450 + (50 \* (i - 45)))

box[51].x = 350

box[51].y = 700

lx = 14

for i in range(52):

redbox[i].x = box[lx].x

redbox[i].y = box[lx].y

lx = lx + 1

if lx > 51:

lx = 0

lx = 50

for i in range(7):

redbox[lx].x = (0 + (50 \* i))

redbox[lx].y = 350

lx = lx + 1

lx = 1

for i in range(52):

bluebox[i].x = box[lx].x

bluebox[i].y = box[lx].y

lx = lx + 1

if lx > 51:

lx = 0

lx = 50

for i in range(7):

bluebox[lx].x = 350

bluebox[lx].y = (700 - (50 \* i))

lx = lx + 1

lx = 40

for i in range(52):

yellowbox[i].x = box[lx].x

yellowbox[i].y = box[lx].y

lx = lx + 1

if lx > 51:

lx = 0

lx = 50

for i in range(7):

yellowbox[lx].x = (700 - (50 \* i))

yellowbox[lx].y = (350)

lx = lx + 1

lx = 27

for i in range(52):

greenbox[i].x = box[lx].x

greenbox[i].y = box[lx].y

lx = lx + 1

if lx > 51:

lx = 0

lx = 50

for i in range(7):

greenbox[lx].x = 350

greenbox[lx].y = (0 + (50 \* i))

lx = lx + 1

for i in range(4):

red[i].swap()

blue[i].swap()

green[i].swap()

yellow[i].swap()

else:

if c >= 1:

if RED == True and TURN == False:

print("a vez do vermelho")

print("movimentos disponíveis: ", rolls)

la = "VERMELHO"

if (movecheck(red, redhome, redbox, la)) == False:

BLUE = True

RED = False

clear()

if RED == True:

for i in range(len(red)):

if ((((cx > red[i].x0 + 13) and (cx < red[i].x + 13)) and (

(cy > red[i].y0 + 14) and (cy < red[i].y + 14)))

and (red[i].x0 == redhome[i].x) and (red[i].y0 == redhome[i].y)):

print("woila ")

if rolls[0 + nc] == 6:

red[i].x0 = redbox[0].x

red[i].y0 = redbox[0].y

red[i].x = redbox[0].x + 25

red[i].y = redbox[0].y + 25

red[i].num = 0

red[i].swap()

nc = nc + 1

if nc > len(rolls) - 1:

BLUE = True

RED = False

clear()

break

if ((((cx > red[i].x0 + 13) and (cx < red[i].x + 13)) and (

(cy > red[i].y0 + 14) and (cy < red[i].y + 14)))

and ((red[i].x0 > 270) or (red[i].y0 > 270))):

print("woila ")

bb = ((red[i].num) + rolls[0 + nc])

if bb > 57:

break

kill(redbox,blue,yellow,green,bluehome,yellowhome,greenhome)

red[i].x0 = redbox[bb].x

red[i].y0 = redbox[bb].y

red[i].x = redbox[bb].x + 25

red[i].y = redbox[bb].y + 25

red[i].swap()

red[i].num = bb

doublecheck(red)

nc = nc + 1

if bb == 57:

red.remove(red[i]);

if nc > len(rolls) - 1:

BLUE = True

RED = False

clear()

break

if BLUE == True and TURN == False:

print("a vez do azul")

print("movimentos disponíveis: ", rolls)

la= "AZUL"

if (movecheck(blue, bluehome, bluebox, la)) == False:

print("SEM MOVIMENTOS")

BLUE = False

YELLOW = True

clear()

if BLUE == True:

for i in range(len(blue)):

if ((((cx > blue[i].x0 + 13) and (cx < blue[i].x + 13)) and (

(cy > blue[i].y0 + 14) and (cy < blue[i].y + 14)))

and (blue[i].x0 == bluehome[i].x) and (blue[i].y0 == bluehome[i].y)):

print("woila ")

if rolls[0 + nc] == 6:

blue[i].x0 = bluebox[0].x

blue[i].y0 = bluebox[0].y

blue[i].x = bluebox[0].x + 25

blue[i].y = bluebox[0].y + 25

blue[i].num = 0

blue[i].swap()

nc = nc + 1

if nc > len(rolls) - 1:

YELLOW = True

BLUE = False

clear()

break

if ((((cx > blue[i].x0 + 13) and (cx < blue[i].x + 13)) and (

(cy > blue[i].y0 + 14) and (cy < blue[i].y + 14)))

and ((blue[i].x0 > 270) or (blue[i].y0 < 470))):

print("woila ")

bb = ((blue[i].num) + rolls[0 + nc])

if bb > 57:

break

kill(bluebox,red,yellow,green,redhome,yellowhome,greenhome)

blue[i].x0 = bluebox[bb].x

blue[i].y0 = bluebox[bb].y

blue[i].x = bluebox[bb].x + 25

blue[i].y = bluebox[bb].y + 25

blue[i].swap()

blue[i].num = bb

doublecheck(blue)

nc = nc + 1

if bb == 57:

blue.remove(blue[i]);

if nc > len(rolls) - 1:

YELLOW = True

BLUE = False

clear()

break

if YELLOW == True and TURN == False:

print("A vez do amarelo")

print("movimentos disponíveis: ", rolls)

la="AMARELO"

if (movecheck(yellow, yellowhome, yellowbox,la)) == False:

print("SEM MOVIMETOS")

YELLOW = False

GREEN = True

clear()

if YELLOW == True:

for i in range(len(yellow)):

if ((((cx > yellow[i].x0 + 13) and (cx < yellow[i].x + 13)) and (

(cy > yellow[i].y0 + 14) and (cy < yellow[i].y + 14)))

and (yellow[i].x0 == yellowhome[i].x) and (yellow[i].y0 == yellowhome[i].y)):

print("woila ")

if rolls[0 + nc] == 6:

yellow[i].x0 = yellowbox[0].x

yellow[i].y0 = yellowbox[0].y

yellow[i].x = yellowbox[0].x + 25

yellow[i].y = yellowbox[0].y + 25

yellow[i].num = 0

yellow[i].swap()

nc = nc + 1

if nc > len(rolls) - 1:

YELLOW = False

GREEN = True

clear()

break

if ((((cx > yellow[i].x0 + 13) and (cx < yellow[i].x + 13)) and (

(cy > yellow[i].y0 + 14) and (cy < yellow[i].y + 14)))

and ((yellow[i].x0 < 470) or (yellow[i].y0 < 470))):

print("woila ")

bb = ((yellow[i].num) + rolls[0 + nc])

if bb > 57:

break

kill(yellowbox,blue,red,green,bluehome,redhome,greenhome)

yellow[i].x0 = yellowbox[bb].x

yellow[i].y0 = yellowbox[bb].y

yellow[i].x = yellowbox[bb].x + 25

yellow[i].y = yellowbox[bb].y + 25

yellow[i].swap()

yellow[i].num = bb

doublecheck(yellow)

nc = nc + 1

if bb == 57:

yellow.remove(yellow[i]);

if nc > len(rolls) - 1:

YELLOW = False

GREEN = True

clear()

break

if GREEN == True and TURN == False:

print("a vez do verde")

print("movimentos disponíveis: ", rolls)

la="VERDE"

if (movecheck(green, greenhome, greenbox,la)) == False:

print("SEM MOVIMENTOS")

GREEN = False

RED = True

clear()

if GREEN == True:

for i in range(len(green)):

if ((((cx > green[i].x0 + 13) and (cx < green[i].x + 13)) and (

(cy > green[i].y0 + 14) and (cy < green[i].y + 14)))

and (green[i].x0 == greenhome[i].x) and (green[i].y0 == greenhome[i].y)):

print("woila ")

if rolls[0 + nc] == 6:

green[i].x0 = greenbox[0].x

green[i].y0 = greenbox[0].y

green[i].x = greenbox[0].x + 25

green[i].y = greenbox[0].y + 25

green[i].num = 0

green[i].swap()

nc = nc + 1

print("verde x.y: ", green[i].x0, green[i].y0)

if nc > len(rolls) - 1:

GREEN = False

RED = True

clear()

break

if ((((cx > green[i].x0 + 13) and (cx < green[i].x + 13)) and (

(cy > green[i].y0 + 14) and (cy < green[i].y + 14)))

and ((green[i].x0 < 470) or (green[i].y0 < 470))):

print("woila ")

bb = ((green[i].num) + rolls[0 + nc])

if bb > 57:

break

kill(greenbox,blue,yellow,red,bluehome,yellowhome,redhome)

green[i].x0 = greenbox[bb].x

green[i].y0 = greenbox[bb].y

green[i].x = greenbox[bb].x + 25

green[i].y = greenbox[bb].y + 25

green[i].swap()

green[i].num = bb

nc = nc + 1

doublecheck(green)

if bb == 57:

green.remove(green[i]);

if nc > len(rolls) - 1:

GREEN = False

RED = True

clear()

break

main()

def leftClick(event):

global c, cx, cy, RED, YELLOW

c = c + 1

cx = root.winfo\_pointerx() - root.winfo\_rootx()

cy = root.winfo\_pointery() - root.winfo\_rooty()

print("Clique em: ", cx, cy)

main()

root.bind("<Butão-1>", leftClick)

def turn():

if RED == True:

L2 = Label(root, text=" A vez do vermelho ", fg='Black', background='green', font=("Arial", 24, "bold"))

L2.place(x=770, y=50)

if BLUE == True:

L2 = Label(root, text=" ", fg='Black', background='green', font=("Arial", 24, "negrito"))

L2.place(x=770, y=50)

if GREEN == True:

L2 = Label(root, text="A vez do verde", fg='Black', background='green', font=("Arial", 24, "negrito"))

L2.place(x=770, y=50)

if YELLOW == True:

L2 = Label(root, text="A vez do verde", fg='Black', background='green', font=("Arial", 24, "negrito"))

L2.place(x=770, y=50)

def roll():

global rollc, dice, dice1, dice2, TURN, rolls

if TURN == True:

rollc = rollc + 1

print("lista: ", rollc)

if rollc == 1:

dice = random.randint(1, 6)

L1 = Label(root, text=dice, fg='black', background='green', font=("Arial", 24,"negrito"))

L1.place(x=800, y=200)

print("dados: ", dice)

rolls.append(dice)

if dice != 6:

rollc = 0

TURN = False

if rollc == 2:

if dice == 6:

dice1 = random.randint(1, 6)

L3 = Label(root, text=dice1, fg='Preto', background='green', font=("Arial", 24, "negrito"))

L3.place(x=800, y=250)

rolls.append(dice1)

if dice1 != 6:

rollc = 0

TURN = False

if rollc == 3:

if dice1 == 6:

dice2 = random.randint(1, 6)

L4 = Label(root, text=dice2, fg='Preto', background='green', font=("Arial", 24, "negrito"))

L4.place(x=800, y=300)

rolls.append(dice2)

rollc = 0

TURN = False

def clear():

global nc, rolls, TURN, L1, L3, L4

nc = 0

del rolls[:]

TURN = True

L1 = Label(root, text=" ", fg='Preto', background='green', font=("Arial", 24, "negrito"))

L1.place(x=800, y=200)

L3 = Label(root, text=" ", fg='Preto', background='green', font=("Arial", 24, "negrito"))

L3.place(x=800, y=250)

L4 = Label(root, text=" ", fg='Preto', background='green', font=("Arial", 24, "negrito"))

L4.place(x=800, y=300)

print("limpo")

turn()

def movecheck(r, rh, rb, la):

if (dice == 6 and dice1 == 6 and dice2 == 6):

return False

win=True

for j in range(4):

if (r[j].x0 != rb[56].x) and (r[j].y0 != rb[56].y):

win=False

if win == True:

print("VOCÊ VENCEU")

L2 = Label(root, text=(la + "Vitórias"), fg='Preto', background='green', font=("Arial", 24, "negrito"))

L2.place(x=770, y=500)

return False

if win == False and dice != 6:

for i in range(len(r)):

if(r[i].num != -1):

(print("good hai"))

return True

print("jani all in")

return False

def kill(a,b,c,d,bh,ch,dh):

if ((a[bb].x0 != box[1].x and a[bb].y0 != box[1].y) and (a[bb].x0 != box[14].x and a[bb].y0 != box[14].y) and

(a[bb].x0 != box[9].x and a[bb].y0 != box[9].y) and (a[bb].x0 != box[22].x and a[bb].y0 != box[22].y) and

(a[bb].x0 != box[27].x and a[bb].y0 != box[27].y) and (a[bb].x0 != box[35].x and a[bb].y0 != box[35].y) and

(a[bb].x0 != box[40].x and a[bb].y0 != box[40].y) and (a[bb].x0 != box[48].x and a[bb].y0 != box[48].y)):

for i in range (len(b)):

if (b[i].x0 == a[bb].x and b[i].y0 == a[bb].y and (b[i].double == False)):

b[i].x0 = bh[i].x

b[i].y0 = bh[i].y

b[i].x = bh[i].x + 25

b[i].y = bh[i].y + 25

b[i].num=-1

b[i].swap()

break

for i in range (len(c)):

if (c[i].x0 == a[bb].x and c[i].y0 == a[bb].y and (c[i].double == False)):

c[i].x0 = ch[i].x

c[i].y0 = ch[i].y

c[i].x = ch[i].x + 25

c[i].y = ch[i].y + 25

c[i].num=-1

c[i].swap()

break

for i in range (len(d)):

if (d[i].x0 == a[bb].x and d[i].y0 == a[bb].y and (d[i].double == False)):

d[i].x0 = dh[i].x

d[i].y0 = dh[i].y

d[i].x = dh[i].x + 25

d[i].y = dh[i].y + 25

d[i].num=-1

d[i].swap()

break

def doublecheck(a):

for k in range (len(a)):

a[k].double = False

for i in range (len(a)):

for j in range (len(a)):

if (a[i].num == a[j].num) and (i != j):

a[j].double = True

a[i].double = True

turn()

button = Button(root, text=" LISTA ", relief="criado", font=("Arial", 20),

command=roll)

root.mainloop()

from tkinter import \*

import random

root= Tk()

canvas = Canvas(width = 1000, height = 800, bg = 'amarelo')

root.resizable(width=False, height=False)

canvas.pack(expand = YES, fill = BOTH)

gif1 = PhotoImage(file = 'ludo board.gif')

canvas.create\_image(50, 10, image = gif1, anchor = NW)

g3 = canvas.create\_oval(50,290,80,320, outline="verde", fill="verde", tags="oval")

g4 = canvas.create\_oval(50,390,80,420, outline="verde", fill="verde", tags="oval")

drag\_data = {"x": 0, "y": 0, "item": None}

init\_data = {"x": 0, "y": 0, "item": None}

final\_coordinate = [0, 0]

def OnTokenButtonPress(event):

drag\_data["item"] = canvas.find\_closest(event.x, event.y)[0]

drag\_data["x"] = event.x

drag\_data["y"] = event.y

init\_data["item"] = drag\_data["item"]

init\_data["x"] = drag\_data["x"]

init\_data["y"] = drag\_data["y"]

item\_below = canvas.find\_overlapping(event.x, event.y, event.x, event.y)[0]

def OnTokenButtonRelease(event):

drag\_data["item"] = None

drag\_data["x"] = 0

drag\_data["y"] = 0

def OnTokenMotion(event):

moved\_x = event.x - drag\_data["x"]

moved\_y = event.y - drag\_data["y"]

canvas.move(drag\_data["item"], moved\_x, moved\_y)

drag\_data["x"] = event.x

drag\_data["y"] = event.y

if drag\_data["x"]>=444 and drag\_data["x"]<=582 and drag\_data["y"]>=330 and drag\_data["y"]<462:

print ('pug gayi')

canvas.tag\_bind("oval", "<ButtonPress-1>", OnTokenButtonPress)

canvas.tag\_bind("oval", "<B1-Motion>", OnTokenMotion)

class RollTheDice:

def \_\_init\_\_(self, parent):

self.dieParent = parent

self.dieContainer = Frame(parent).pack()

self.dieLabel = Label(self.dieContainer, text="Número de dados que você vai lançar:")

self.dieLabel.pack(side=TOP)

self.dieEntry = Entry(self.dieContainer)

self.dieEntry.pack(side=TOP)

self.sideLabel = Label(self.dieContainer, text="Número de lados por matriz:")

self.sideLabel.pack(side=TOP)

self.sideEntry = Entry(self.dieContainer)

self.sideEntry.pack(side=TOP)

global rolldisp

rolldisp = StringVar()

self.rollResult = Label(self.dieContainer, textvariable=rolldisp)

self.rollResult.pack(side=TOP)

self.diceButton = Button(self.dieContainer)

self.diceButton.configure(text="Lance os dados!", background="vermelholaranja1")

self.diceButton.pack(side=LEFT)

self.diceButton.bind("<Butão-1>", self.diceButtonClick)

self.diceButton.bind("<Retornar>", self.diceButtonClick)

self.quitButton = Button(self.dieContainer)

self.quitButton.configure(text="Sair", background="azul")

self.quitButton.pack(side=RIGHT)

self.quitButton.bind("<Butão-1>", self.quitButtonClick)

self.quitButton.bind("<Retornar>", self.quitButtonClick)

def diceButtonClick(self, event):

die = int(self.dieEntry.get())

side = int(self.sideEntry.get())

DieRoll(die, side)

def quitButtonClick(self, event):

self.dieParent.destroy()

def DieRoll(dice, sides):

import random

rollnumber = 1

runningtotal = 0

endresult = ""

while rollnumber <= dice:

roll = random.randint(1, sides)

endresult += "Lista #"

endresult += str(rollnumber)

endresult += ": "

endresult += str(roll)

endresult += "\n"

runningtotal += roll

rollnumber += 1

finalresult = "Sua Lista:\n"

finalresult += endresult

rolldisp.set(finalresult)

def leftClick(event):

x = root.winfo\_pointerx()

y = root.winfo\_pointery()

print("Clique em: ",x,y)

root.bind("<Butão-1>", leftClick)

root = Tk()

root.title("Die Roller")

myapp = RollTheDice(root)

root.mainloop()