

Jogo da Velha em GUI

Mário Leite

...

Com certeza, você já deve ter ouvido falar no “Jogo da Velha”; e se é um programador já deve ter implementado (ou tentado implementar) esse famoso joguinho na sua linguagem preferida (antigamente eu fazia isto no Clipper, com muita dificuldade). Entretanto, com o passar dos anos, implementar esse jogo ficou bem mais fácil com as novas linguagens de programação; mas, essas implementações são, quase invariavelmente, com caracteres: no máximo com tracinhos para “desenhar” os quadrinhos do tabuleiro do jogo. Agora (aliás, desde 1991), com o aparecimento da linguagem Python tudo ficou mais fácil com o uso de poderosas bibliotecas, tais como: *matplotlib*, *turtle* e *tkinter*; esta última a minha preferida. Assim, criar uma interface gráfica interativa com o usuário (GUI) ficou muito fácil, embora em alguns casos o código fique um pouco trabalhoso e repetitivo, devido à exigência de marcar as coordenadas dos pontos para definir as posições dos elementos sobre ela. De qualquer forma, ficou muito fácil criar uma interface gráfica (com cores, inclusive) usando essas bibliotecas, que possuem funções específicas para diversas situações onde é necessário movimentar elementos na interface gráfica, podendo até criar efeitos de movimentos. O Jogo da Velha é um desses joguinhos bastante simples, mas que pode demonstrar o poder do Python. O programa “**JogoDaVelha**” mostra como é fácil criar um joguinho nessa linguagem usando a biblioteca *tkinter*. As **figuras 1a, 1b e 1c** mostram os três possíveis resultados do jogo, onde o jogador deve clicar com o ponteiro do *mouse* sobre os quadrinhos do tabuleiro para definir a sua jogada com “X”, enquanto o computador marca com “O”, criando um ambiente bem realista desse jogo.

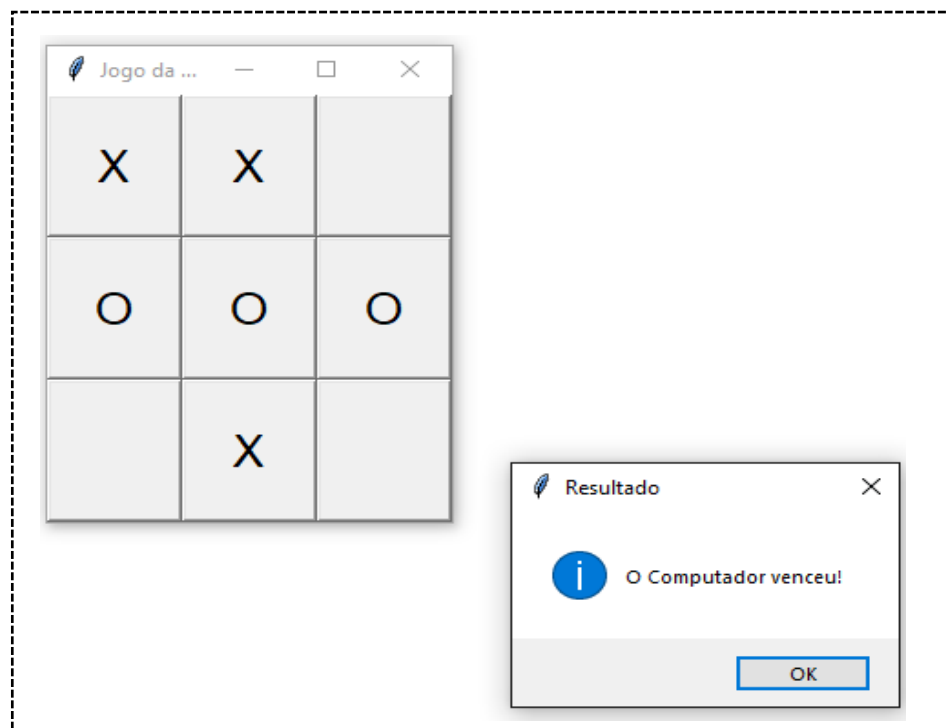


Figura 1a - Saída do programa “JogoDaVelha” onde o Computador venceu

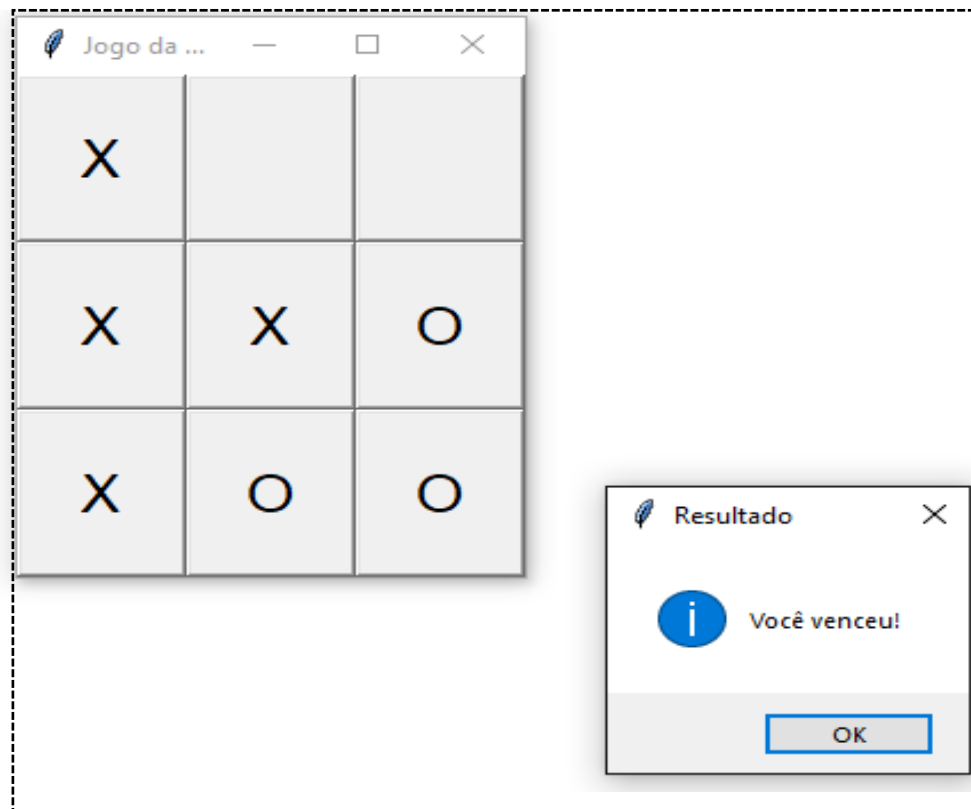


Figura 1b - Saída do programa “JogoDaVelha” onde o Jogador venceu

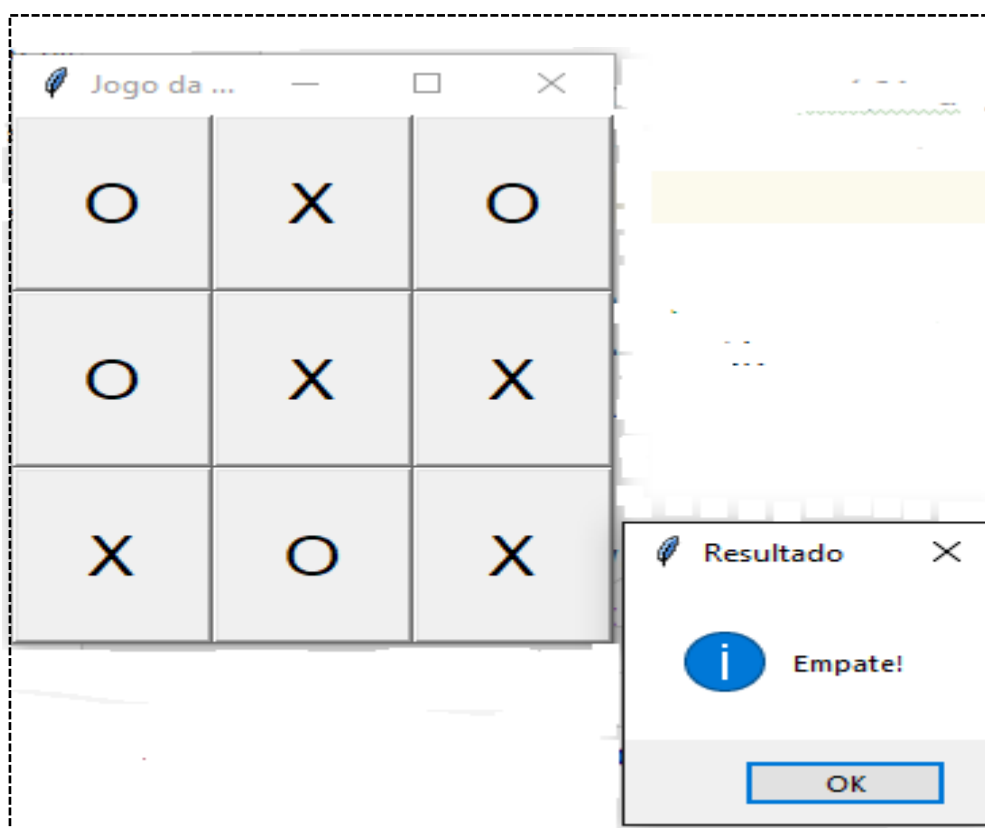


Figura 1c - Saída do programa “JogoDaVelha” onde houve Empate

```
'''
JogoDaVelha.py
-----
Simula uma interface gráfica para o clássico "Jogo da Velha".
-----
'''

import tkinter as tk
from tkinter import messagebox
import random

class ClsJogoVelha:
    def __init__(self, base_gui):
        self.base_gui = base_gui
        self.base_gui.title("Jogo da Velha")
        self.jogadorAtual = "X"
        self.tabuleiro = [[" " for _ in range(3)] for _ in range(3)]
        self.CriarInterface()

    def CriarInterface(self):
        self.buttons = [[None for _ in range(3)] for _ in range(3)]
        for i in range(3):
            for j in range(3):
                self.buttons[i][j] = tk.Button(self.base_gui, text="",
                                                font=("Arial", 20), width=4, height=2,
                                                command=lambda linha=i, coluna=j: self.Movimentar(linha, coluna))
                self.buttons[i][j].grid(row=i, column=j)

    def Movimentar(self, linha, coluna):
        if(self.jogadorAtual == "X" and self.tabuleiro[linha][coluna] == " "):
            self.tabuleiro[linha][coluna] = self.jogadorAtual
            self.buttons[linha][coluna].config(text=self.jogadorAtual)
            if(self.VerificarGanhador(linha, coluna)):
                messagebox.showinfo("Resultado", "Você venceu!")
                self.ResetarTabuleiro()
            elif(self.ChecarClick()):
                messagebox.showinfo("Resultado", "Empate!")
                self.ResetarTabuleiro()
            else:
                self.jogadorAtual = "O"
                self.ComputadorJoga()

    def VerificarGanhador(self, linha, coluna):
        jogador = self.jogadorAtual
        #Verificar linha
        if(self.tabuleiro[linha][0] == self.tabuleiro[linha][1] ==
           self.tabuleiro[linha][2] == jogador):
            return True
        #Verificar coluna
        if(self.tabuleiro[0][coluna] == self.tabuleiro[1][coluna] ==
           self.tabuleiro[2][coluna] == jogador):
            return True
        #Verificar diagonais
        if(linha == coluna and
           self.tabuleiro[0][0]==self.tabuleiro[1][1]==self.tabuleiro[2][2]==jogador):
            return True
        if(linha + coluna == 2 and
           self.tabuleiro[0][2]==self.tabuleiro[1][1]==self.tabuleiro[2][0]== jogador):
            return True
        return False
```

```

def ChecarClick(self):
    for linha in self.tabuleiro:
        for cell in linha:
            if(cell == ""):
                return False
    return True

def ResetarTabuleiro(self):
    for i in range(3):
        for j in range(3):
            self.tabuleiro[i][j] = ""
            self.buttons[i][j].config(text="")
    self.jogadorAtual = "X"

def ComputadorJoga(self):
    empty_cells=[(i,j) for i in range(3) for j in range(3) if self.tabuleiro[i][j]==""]
    if(empty_cells):
        linha, coluna = random.choice(empty_cells)
        self.tabuleiro[linha][coluna] = "O"
        self.buttons[linha][coluna].config(text="O")
        if(self.VerificarGanhador(linha, coluna)):
            messagebox.showinfo("Resultado", "O Computador venceu!")
            self.ResetarTabuleiro()
        elif(self.ChecarClick()):
            messagebox.showinfo("Resultado", "Empate!")
            self.ResetarTabuleiro()
        else:
            self.jogadorAtual = "X"
#-----
def main():
    janela = tk.Tk()
    Jogo = ClsJogoVelha(janela)
    janela.mainloop()
#=====
#Programa principal
if(__name__ == "__main__"):
    main()
#Fim do programa "JogoDaVelha" -----

```