

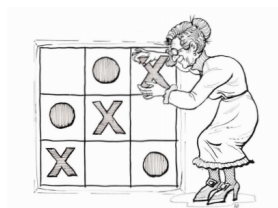


Universidade Federal  
do Espírito Santo

Departamento de Computação e Eletrônica - CEUNES  
PROGRAMAÇÃO I / PROGRAMAÇÃO FUNCIONAL  
Prof. Oberlan Romão

## Exercício Programa 2

### Jogo da Velha



O **Jogo da Velha** é um jogo muito popular e com regras extremamente simples, que não traz grandes dificuldades para seus jogadores e é facilmente aprendido. O seu nome Jogo da Velha originou-se na Inglaterra quando mulheres ao fim de tarde se reuniam para tomar o chá, bordar e brincar. O jogo da velha era jogado pelas senhoras de mais idade já que as mesmas não enxergavam bem e não podiam então realizar seus bordados. A opção de diversão então era o Jogo da Velha. Entretanto, acredita-se que sua origem seja ainda mais antiga, pois foram encontrados tabuleiros esculpidos em rochas em templos do antigo Egito, que teriam mais de 3.500 anos<sup>1</sup>. O jogo é disputado entre dois jogadores, onde um jogador é X e o outro jogador é O e as regras são as seguintes:

- O tabuleiro é uma matriz de três linhas por três colunas;
- Um dos jogadores escolhe se será X ou O. Consequentemente, o outro jogador será o símbolo não escolhido pelo primeiro jogador;
- Os jogadores jogam alternadamente, uma marcação por vez, numa lacuna que esteja vazia (ainda não marcada);
- O objetivo é conseguir três O ou três X em linha, quer horizontal, vertical ou diagonal, e ao mesmo tempo, quando possível, impedir o adversário de ganhar a jogada;
- Quando um jogador conquista o objetivo, costuma-se riscar os três símbolos;
- Se os dois jogadores jogarem sempre da melhor forma, o jogo terminará sempre em empate.

As figuras abaixo ilustram duas partidas. A primeira mostra a uma jogada onde o jogador X ganha o jogo e a segunda, um empate.

---

<sup>1</sup>Fonte: <https://www.bigmae.com/regras-jogo-da-velha/>

Partida em que o jogador X ganha:

	X	O	X	O	X	O	X	O	X	O	X	O	X

Partida que termina com empate (chamado de velha no Brasil, costuma-se dizer que o jogo "deu velha"):

X	X	X	O	X	O	X	O	X	O	X	O	X	O

## Tarefa

Como você já deve imaginar, sua tarefa nesse EP é implementar o Jogo da Velha para ser jogado contra o computador (usando alguma estratégia definida por você), ou seja, um jogador será uma pessoa e o outro o computador. Para facilitar e padronizar a implementação usaremos cada posição do tabuleiro como se fosse um teclado numérico, conforme mostra a Figura 1. Veja alguns exemplos de execução do programa na seção de Exemplos mais abaixo.

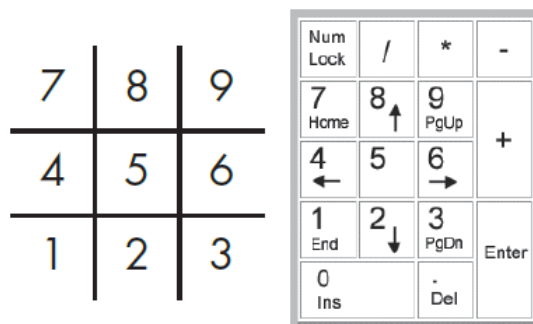


Figura 1: O tabuleiro é numerado como um teclado numérico.

O programa começa solicitando ao jogador se ele deseja ser X ou O (o programa também deve aceitar x ou o, mas sempre “trabalhar” com a opção maiúscula). Quem joga primeiro (jogador ou computador) é escolhido aleatoriamente. Em seguida, o jogador e o computador se revezam nas jogadas.

O jogador e/ou computador deve escolher uma posição válida (entre 1 e 9) e que ainda não foi marcada. Caso o jogador escolha uma posição inválida, o programa deve exibir uma mensagem apropriada e solicitar novamente a posição (é obvio que o computador não deve escolher uma posição inválida). Veja o Exemplo 4 mais abaixo. Após o jogador ou computador escolher a posição, deve-se verificar se um deles ganhou ou se deu empate. Caso o jogo ainda não tenha terminado, muda-se a vez da jogada e esse processo se repete até que o jogador ou computador vença a partida ou dê empate.

O tabuleiro **deve** ser representado por uma variável do tipo `list` de tamanho 10, onde cada posição, entre 1 e 9 (a posição 0 deve ser ignorada), da lista representa a respectiva posição no tabuleiro, conforme a Figura 1. Dessa forma, se o jogador escolheu ser X e, após algumas jogadas, jogou nas posições {1, 5, 9} e o computador escolheu as posições {7, 3}, a lista será preenchida da seguinte forma:

`[" ", "X", " ", " ", "O", " ", "X", " ", " ", "O", " ", "X"]`

Note que as posições que não foram jogadas (posições livres) estão marcadas com `" "` (esse deve ser o padrão adotado para representar uma posição livre). Além disso, observe que o jogador venceu!

Para padronizar as implementações de algumas funções (que possivelmente serão usadas em um campeonato), baixem o arquivo `base.py` no AVA, renomeie o arquivo para a sua matricula e complete o

código. Comece preenchendo a sua matrícula e nome completo nas funções `getMatricula` e `getNome`, respectivamente.

No fim do arquivo, você verá o código abaixo (que não deve ser modificado). Não se preocupe em entendê-lo. Esse trecho de código “diz” ao interpretador do Python que se o arquivo `base.py` for o arquivo principal, deve ser executado a função `main`, caso o arquivo seja usado como um módulo em outro arquivo (`import base`), a função `main` não é executada.

```
1 if __name__ == "__main__":
2     main()
```

A função `jogadaComputador` é responsável por definir em qual posição o computador jogará. Ela deve receber dois parâmetros obrigatórios (outros parâmetros **opcionais** podem ser adicionados): uma lista (de tamanho 10) e um string representando a letra do computador (X ou O). Além disso, a função deve, obrigatoriamente, retornar **apenas** a posição do tabuleiro (valor inteiro entre 1 e 9) que o computador irá jogar. Seu objetivo, nessa função, é propor alguma estratégia para o computador tentar vencer o jogador humano. Por exemplo, uma estratégia extremamente ingenua seria escolher aleatoriamente uma das posições válidas (ainda não marcadas). Isso pode ser feito com a função `choice` do módulo `random`. Pense ou pesquise por outras estratégias mais inteligentes e implemente uma delas. Você **deve** explicar a estratégia adotada no `"""docstring"""` da função `jogadaComputador`.

Como você deve imaginar, você pode e deve implementar outras funções para que o programa funcione por completo. Por exemplo, você pode fazer uma função para:

- Solicitar a letra que o jogador quer ser;
- Imprimir o tabuleiro;
- Verificar se o jogador ou computador venceram;
- Intercalar as jogadas entre o jogador e o computador;
- Entre outras...

## O que entregar

Neste EP, você deve entregar, pelo **AVA**, apenas um arquivo contendo o código fonte do jogo. O nome do arquivo deve ser no seguinte padrão: `PrimeiroNomeAluno_Matricula.py` (Por exemplo: `Oberlan_2022123456.py`).

**Data de entrega:** até às 6h do dia 12/07/2023.

### Observações:

1. Não é permitido usar **variáveis globais**, **estruturas de repetição (loop)**, como `while`, `for`, **funções impuras**, *nested functions* (declaração de funções dentro de funções), `class` (paradigma orientado a objetos), e operações que não sejam do Paradigma Funcional (por ex., `lista.count()`, `lista.sort()`, `lista.reverse()`, `lista.append()`, `lista.index()`, `str.lower()`, `str.upper()`, `str.replace()`, etc.). A utilização dessas estruturas/funções implicará em nota 0. Se não tem certeza se pode ou não usar determinada estrutura/função, me pergunte.
2. Quando necessário, utilize funções recursivas;
3. Funções que modificam uma lista passada como argumento, devem ser chamadas passando uma cópia da lista. Isso evita que a função se torne impura (leia a observação 1). Leia mais [aqui](#);

4. Use apenas instruções/comandos visto em sala de aula (teórica ou prática);
5. Documente o seu programa: comente e use `"""docstring"""` nas funções. O `"""docstring"""` deve conter o que a função faz e retorna e o que representa cada parâmetro (se houver);
6. Em caso de plágio (ou auto-plágio), será atribuído 0 a todos os envolvidos.

## Critérios de avaliação

---

A nota do EP se dará pela seguinte fórmula:

$$(1 - P) \times F \times C \times r \times N_{EP},$$

onde,

- $P = \begin{cases} 1, & \text{se houve plágio ou auto-plágio;} \\ 0, & \text{caso contrário.} \end{cases}$
- $F = \begin{cases} 1, & \text{se respeitou a observação 1;} \\ 0, & \text{caso contrário.} \end{cases}$
- $C = \begin{cases} 1.0, & \text{se comentou o código e fez o docstring das funções de forma correta;} \\ 0.9, & \text{caso contrário.} \end{cases}$
- $r = \begin{cases} 0.8, & \text{se o código possui replicação;} \\ 1.0, & \text{caso contrário.} \end{cases}$
- $N_{EP}$ : Nota geral do EP, sendo  $0.0 \leq N_{EP} \leq 10.0$ .

A estratégia usada (implementada na função `jogadaComputador`) valerá 3.0 pontos. Se a função usar uma estratégia que **apenas** sorteia uma posição livre ela receberá, no máximo, 1.0 ponto.

“Programming is not about typing, it’s about thinking.” - Rich Hickey

## Exemplos de jogada do Jogo da Velha

Veja alguns exemplos de execução do programa. Os valores em negrito e sublinhados foram inseridos pelo jogador.

### Exemplo 1 - Computador ganha

Bem-vindo ao clássico Jogo da Velha!

Você quer ser 'X' ou 'O'? X

O computador começa.

```
| |
---+---+---
```

```
| |
---+---+---
```

```
O | |
```

Qual posição deseja marcar (1-9): 9

```
| | X
---+---+---
```

```
| |
---+---+---
```

```
O | | O
```

Qual posição deseja marcar (1-9): 2

```
O | | X
---+---+---
```

```
| |
---+---+---
```

```
O | X | O
```

Qual posição deseja marcar (1-9): 4

```
O | | X
---+---+---
```

```
X | O |
---+---+---
```

```
O | X | O
```

O computador venceu!

## Exemplo 2 - Jogador ganha

Bem-vindo ao clássico Jogo da Velha!

Você quer ser 'X' ou 'O'? O

Você começa.

```

  |  |
  +--+

```

```

  |  |
  +--+

```

```

  |  |

```

Qual posição deseja marcar (1-9): 1

```

  |  |
  +--+

```

```

  |  |
  +--+

```

```

O |  | X

```

Qual posição deseja marcar (1-9): 7

```

O |  |
  +--+

```

```

X |  |
  +--+

```

```

O |  | X

```

Qual posição deseja marcar (1-9): 9

```

O |  | O
  +--+

```

```

X | X |
  +--+

```

```

O |  | X

```

Qual posição deseja marcar (1-9): 8

```

O | O | O
  +--+

```

```

X | X |
  +--+

```

```

O |  | X

```

Parabéns! Você venceu o jogo!

### Exemplo 3 - Empate

Bem-vindo ao clássico Jogo da Velha!

Você quer ser 'X' ou 'O'? O

O computador começa.

```
|  |
---+---+---
```

```
|  |
---+---+---
```

```
|  | X
```

Qual posição deseja marcar (1-9): 5

```
|  | X
---+---+---
```

```
| O |
---+---+---
```

```
|  | X
```

Qual posição deseja marcar (1-9): 6

```
|  | X
---+---+---
```

```
X | O | O
---+---+---
```

```
|  | X
```

Qual posição deseja marcar (1-9): 7

```
O |  | X
---+---+---
```

```
X | O | O
---+---+---
```

```
X |  | X
```

Qual posição deseja marcar (1-9): 2

```
O | X | X
---+---+---
```

```
X | O | O
---+---+---
```

```
X | O | X
```

Empate!

## Exemplo 4

Bem-vindo ao clássico Jogo da Velha!

Você quer ser 'X' ou 'O'? A

Simbolo inválido!

Você quer ser 'X' ou 'O'? x

Você começa.

```

  |  |
  ---+---+---
  |  |
  ---+---+---
  |  |

```

Qual posição deseja marcar (1-9): 10

Valor inválido. Você deve digitar um número inteiro entre 1 e 9.

Qual posição deseja marcar (1-9): -1

Valor inválido. Você deve digitar um número inteiro entre 1 e 9.

Qual posição deseja marcar (1-9): a

Valor inválido. Você deve digitar um número inteiro entre 1 e 9.

Qual posição deseja marcar (1-9): 1.5

Valor inválido. Você deve digitar um número inteiro entre 1 e 9.

Qual posição deseja marcar (1-9): 1

```

  |  |
  ---+---+---
  |  |
  ---+---+---
  X |  | 0

```

Qual posição deseja marcar (1-9): 3

Qual posição deseja marcar (1-9): 1

Qual posição deseja marcar (1-9): 2

```

  |  | 0
  ---+---+---
  |  |
  ---+---+---
  X | X | 0

```

Qual posição deseja marcar (1-9):