

## Trabalho Prático – Jogo da Velha

### Disciplina Fundamentos de Programação – Semestre 2020.1

Prof. Miguel Franklin

Desenvolver um programa em linguagem Python, em modo texto (terminal), representando o jogo da velha, onde um usuário deverá ser capaz de jogar contra outro jogador.

#### Requisitos:

1. Cada usuário deverá, a cada rodada, visualizar a tabela de jogo atualizada na forma seguinte:

```

      A   B   C
1  |   |   | 1
  ---+---+---
2  |   |   | 2
  ---+---+---
3  |   |   | 3
      A   B   C

```

2. O primeiro usuário (usuário 1) deve poder escolher entre jogar com X ou O. Ao segundo jogador (usuário 2), será atribuído o outro símbolo.
3. Cada usuário entrará sua jogada com as coordenadas da casa desejada. (Ex. "A,2"). Esta entrada pode ser flexível (Ex. "a,2" = "A,2" = "a2" = "A2"). Toda entrada deverá ser validada (Ex. O computador não deve receber uma jogada como "D5". Caso uma entrada seja inválida, o computador deverá solicitar nova jogada válida até que o usuário o faça.
4. Após cada jogada, o computador deve identificar quando o usuário que acabou de jogar ganha a rodada, fechando uma linha, coluna ou diagonal. O empate (todas as casas marcadas sem um vencedor) também deve ser identificado pelo computador.
5. O computador deve também detectar e comunicar a um usuário que está na iminência de jogar que ele já perdeu a rodada, mesmo antes do outro usuário jogar. Isso acontece quando um usuário tiver, na jogada seguinte, duas possibilidades de fechar o jogo, e o outro usuário não poderá evitar ambas. Vide exemplo abaixo. Obviamente, essa mensagem só deve ser exibida se o outro usuário não puder, já na próxima jogada, se antecipar e fechar o jogo.

```

      A   B   C
1  X | O | O 1
  ---+---+---
2  O |   |   2
  ---+---+---
3  X |   | X 3
      A   B   C

```

6. O computador deverá, desde quando o programa for executado até o usuário 1 escolher terminar o programa, guardar o placar entre usuário 1 e usuário 2, contabilizando também os empates. No final de cada rodada, o computador deverá exibir o placar e perguntar se o usuário 1 deseja jogar novamente. Se sim, o jogo é recommençado.

## Critérios de Avaliação

A avaliação será realizada em duas fases:

1. Análise do código-fonte;
2. Análise da execução do programa (teste).

O código-fonte será avaliado de acordo com os seguintes critérios qualitativos:

- i. Eficácia do programa em suprir todos os requisitos;
- ii. Eficiência do programa (otimização);
- iii. Organização do código (uso racional de subprogramas, estruturas, etc.);
- iv. Legibilidade do código (uso de indentação e semântica dos identificadores de variáveis);
- v. Documentação (comentários dentro do código fonte).

Obviamente, funcionalidades adicionais às que foram solicitadas neste documento são bem vindas e serão gratificadas na nota (na medida do possível). O código-fonte deve conter, em comentário no início, os nomes e matrículas dos alunos que compõem o grupo. O código-fonte deve ser submetido na data fixada através do SOLAR.

Apenas será admitida programação em linguagem **Python**, sendo atribuída nota **ZERO** caso este requisito não seja respeitado.

Os trabalhos serão corrigidos no **Linux** utilizando a versão 3 do Python. Portanto, certifique-se que o trabalho feito no Windows também roda no Linux.

Trabalhos cujos códigos fonte apresentem problemas que impeçam a sua execução terão nota **ZERO**. Não haverá, por parte da equipe de avaliação do trabalho, tentativa de correção do problema. Portanto, antes de fazer a entrega, certifique-se que o programa executa de forma limpa no LINUX usando o *Python 3*.

Lembramos que todos os programas serão submetidos a análise léxica automática, que pode evidenciar cópia (plágio) de código. Nenhum tipo de cópia ou “inspiração” será tolerado. Caso ocorra, será atribuída nota **ZERO** às equipes com trabalhos semelhantes.

## TRABALHO INDIVIDUAL

**Prazo de Entrega: 18 de outubro de 2020, até 23:59.**

*Upload através do SOLAR*  
Não serão aceitas entregas por e-mail.