

UFES - CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA
Prof. Thomas W. Rauber

1º Trabalho, Programação Básica de Computadores (Linguagem Python) INF09325, T02 Earte

Data de entrega: veja Mural na Sala de Aula

Linguagem de Programação para Implementação: Python
Grupo de até dois alunos

Programação Estruturado no Exemplo do Jogo da Velha.

1 Problema

Dois jogadores alternadamente colocam uma marca (cruz e o círculo) sua em um dos nove campos de um tabuleiro quadrático 3×3 , até que o fim do jogo seja atingido. Existem três possibilidades de terminar

1. O primeiro jogador ganha;
2. O segundo jogador ganha;
3. Empate (“deu velha”);

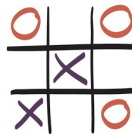


Figura 1: Tabuleiro do jogo da velha após cinco jogadas.

Um jogador ganha, se conseguir três das suas marcas em linha, horizontalmente, verticalmente, ou diagonalmente. Quando um jogador conquista o objetivo, costuma-se riscar as três marcas em linha. O jogo termina em empate, se todos os nove campos forem preenchidos, sem que haja três marcas em linha.

O objetivo do trabalho é simular uma partida do jogo, usando o módulo da tartaruga como interface gráfica. Deve ser usado para desenhar o campo, as marcas e o risco em caso de vitória.

2 Objetivo

O objetivo do trabalho é a utilização dos elementos de programação estruturada, disponível na linguagem Python, especialmente seleção (“IF”), repetição (“WHILE”, ou “FOR”). O aluno deve usar estruturas de dados apropriados para armazenar as informações. Por exemplo, a estrutura mais apropriada para representar o tabuleiro e o conteúdo de cada campo é uma matriz, (lista de lista).

3 Tarefas

Simule o jogo por jogadas aleatórias dos dois jogadores, até que a partida seja encerrada. Podemos identificar as seguintes etapas:

1. Definição de constantes e variáveis. Ajuda: Uma matriz 2×2 com todos os seus elementos vazios pode ser definida como

```
campo = [[None, None], [None, None]]
```

Ajuda: Pode usar uma cor para identificar o jogador ('red', 'blue'). Assim é possível saber a cor atual da tartaruga e atribuir uma cor à tartaruga com a função `fillcolor`.

2. Desenho do campo. Use a tartaruga para desenhar o campo de jogo. Deve definir apenas uma constante que define a medida do lado do quadrado. Exemplo:

```
lado = 300
```

Todas as outras medidas devem ser definidas, relativas a esta constante. Por exemplo a divisão horizontal do campo. Use uma função para encapsular a criação do campo. Ajuda: A função deve receber como parâmetros a tartaruga inicial e a extensão do campo, e retornar a tartaruga

3. Jogo. Enquanto o jogo ainda não terminar. Ajuda: Repetição condicional
 - (a) Gere aleatoriamente uma posição no campo que ainda não esteja ocupada por uma marca. Ajuda: Faça uma função que gere e retorne dois números aleatórios que representam a linha e coluna da posição da próxima jogada. Essa função não tem parâmetros de entrada. Ajuda: Repetição condicional, pois tem que tentar gerar uma posição válida.
 - (b) Após achar uma posição válida, tem que verificar, se o jogador ganhou. Ajuda: Somente oito combinações de três marcas implicam uma vitória.
4. Se houver um ganhador, a tartaruga deve desenhar o risco pelos três campos

Referências

- Jogo da Velha
https://pt.wikipedia.org/wiki/Jogo_da_velha
- turtle – Gráficos Turtle
<https://docs.python.org/pt-br/3/library/turtle.html>
- Interface de funções aleatórias
<https://docs.python.org/pt-br/3/library/random.html>

Elaboração: O resultado deve ser código em Python que simula o jogo. O usuário (neste caso o professor) não deve ter o trabalho de digitar nada, além da linha de comando no Python que executa o programa.

O produto final deve ser um arquivo Python com a seguinte sintaxe: `aluno1+aluno2.py`, a ser submetido como resposta por um dos autores pelo sistema da sala de aula.

A documentação deve ser em forma de descrição de projeto, preferencialmente gerado por \LaTeX , (ou LibreOffice) contendo os seguintes tópicos:

- Capa do Projeto
 - Título
 - Autoria
 - Data
 - Resumo
- Introdução
- Objetivos
- Metodologia
- Resultados e Avaliação
- Referências Bibliográficas

Qualquer dúvida? Não hesite em me contactar por E-mail institucional `thomas.rauber@ufes.br`, ou na aula (preferencialmente no início, ou no final).

Recomendo um desenvolvimento por partes. Não avance, antes de ter testado o seu programa em seu estado atual. É difícil detectar erros, se primeiro se escreve muito código sem testar, e somente no final começa a testar.

Última atualização: 10 de abril de 2021, 18:43

Bom trabalho!