

Exercício Programa 3 – Design de Software

Jogo da Velha (Tic-tac-toe)

Fábio Ayres/Raul Ikeda

Entregável: O prazo é **25/Abr/2016** até às **7:30 AM** via Github.

Introdução

O Exercício Programa 3 consiste em programar uma interface gráfica (GUI) que gerencia um jogo da velha¹. Considere que ambos os jogadores utilizarão o mesmo computador e o mesmo programa para realizar as jogadas.

O EP deve ser realizado em dupla. Não será permitido fazer sozinho. Em turmas com número ímpar de alunos será admitido um único trio. Será exigida a utilização do GitHub como gerenciador de arquivos e entrega do conteúdo.

Git e GitHub

O **git** é um *sistema de controle de versão*. Em uma análise mais superficial, é como um Dropbox para código, que sabe fazer algumas coisas extras:

- Guarda toda o histórico de mudanças nos arquivos
- Facilita a tarefa de integrar duas versões distintas do mesmo arquivo, ou seja, se dois desenvolvedores trabalharem no mesmo arquivo ao mesmo tempo em partes separadas, git irá ajuda-los a unificar as modificações.
- Facilita a gerência de linhas independentes de trabalho (*branches*)

Existe muita documentação sobre o git, tanto online como em livros. Algumas sugestões:

- <https://try.github.io/>
- <http://rogerdudler.github.io/git-guide/>

NOTA IMPORTANTE: para criar um repositório, você pode

- Criar um repositório local com `git init`
- Ou clonar um repositório remoto com `git clone`. Este repositório remoto já foi criado por alguém (potencialmente você mesmo).

Neste EP iremos criar um repositório remoto no site do GitHub usando a interface web deles, e então iremos **clonar o repositório remoto**.

O **GitHub** é um serviço de hospedagem de repositórios git. Neste EP você deve:

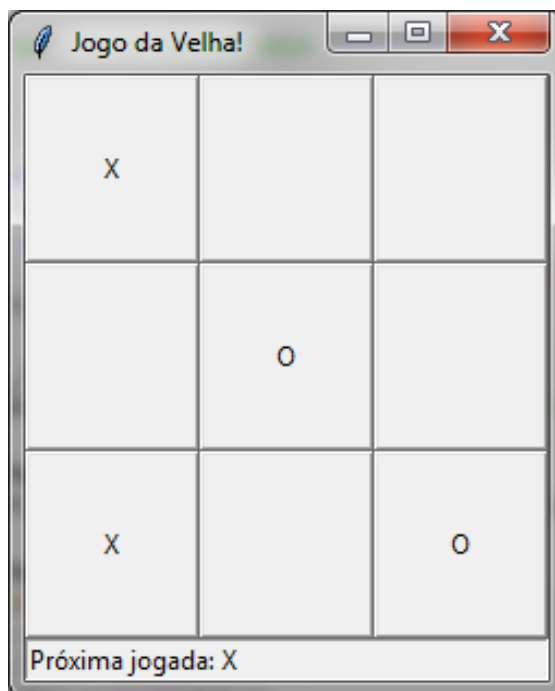
¹Dúvidas: https://pt.wikipedia.org/wiki/Jogo_da_velha

- Criar uma conta no GitHub (<https://github.com>).
- Decidir em qual conta (dentre os membros da dupla ou trio) o EP será armazenado.
- Criar um repositório inicial público com apenas o arquivo README.md – trata-se de uma das opções de criação de repositórios.
- Permitir acesso aos colegas do grupo a esse repositório.
- Enviar por email para o professor:
 - a URL do repositório, e
 - o nome dos membros da dupla.

Especificação de requisitos do sistema

A dupla deve codificar apenas duas classes (em arquivos separados):

- Classe Tabuleiro: responsável por criar e administrar a interface gráfica em tkinter que interage com os jogadores. Deverão ser disponibilizados 9 botões dispostos em uma grade de tamanho 3x3, que quando pressionados, representam uma jogada. A tela deve ser limpa quando o jogo reiniciar, ou seja, quando houver um ganhador ou um empate. Ela também deve indicar qual jogador realizará a próxima jogada. Veja o exemplo abaixo:



- Classe Jogo: responsável pelo gerenciamento do jogo. Deverá receber a jogada quando um botão for pressionado e também verificar se o jogo terminou. Será responsável também por alternar os jogadores.
A classe Tabuleiro, descrita anteriormente, deverá obrigatoriamente conter como atributo um objeto da classe Jogo.
A classe Jogo não terá acesso direto aos componentes da tela.

¹Dúvidas: https://pt.wikipedia.org/wiki/Jogo_da_velha

A classe deverá conter pelo menos um atributo que indicará para a tela qual é o próximo jogador. Também deverá conter no mínimo 3 métodos:

1. `recebe_jogada(linha, coluna)`: Método para receber e registrar uma jogada. Deve também alternar os jogadores. Não há retorno de valores.
2. `verifica_ganhador()`: Método para verificar se o jogo acabou. A função deve retornar 0 em caso de empate, 1 caso o X seja o vencedor, 2 caso o O seja o vencedor ou -1 caso contrário.
3. `limpa_jogadas()`: Método para limpar as jogadas, ou seja, reinicia o jogo (não limpará a tela – lembre-se que a classe `Jogo` não tem acesso à classe `Tabuleiro`), mantendo a vez do jogador.

Entregáveis

A dupla deve gerenciar as tarefas e dividir o trabalho em uma classe para cada membro. Deve-se trabalhar em paralelo e utilizar o Github para a realização de *commits* intermediários nos códigos. Os logs serão analisados para efeito de análise de esforço empregado por cada membro.

Portanto, o resultado final do seu trabalho será:

- A URL do repositório git, e
- Neste repositório, devemos ter dois arquivos `.py`: um da classe `Tabuleiro`, outro da classe `Jogo`.

O seu resultado final será o estado do repositório às 7:30AM do dia 25/Abr/2016.

SE VOCÊ NÃO FEZ O “PUSH” DO SEU CÓDIGO LOCAL PARA O REPOSITÓRIO REMOTO ANTES DESSE HORÁRIO, VOCÊ TERÁ FALHADO NA ENTREGA DO PROJETO.

FAÇA PUSH FREQUENTEMENTE!

¹Dúvidas: https://pt.wikipedia.org/wiki/Jogo_da_velha