

MC 302EF-Atividade de Laboratório no. 8

[criado em 20/05/16]

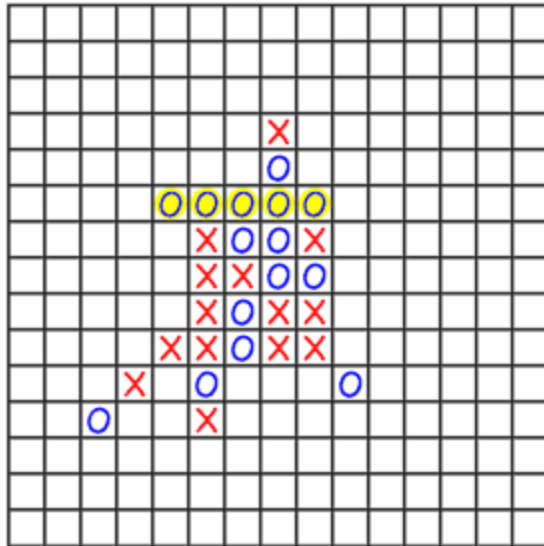
Objetivos

As atividade de laboratório 8 e 9 têm como objetivo o desenvolvimento de uma aplicação em Java usando o padrão MVC (*Model, View, Control*).

A atividade 8 consiste na implementação da classe que terá o papel do modelo no padrão MVC. A atividade 9 será responsável pela implementação das classes responsáveis pela apresentação e controle.

Descrição do Problema

O resultado final das atividades 8 e 9 será o desenvolvimento do jogo Gomoku, também conhecido como '5 em linha'. O jogo é baseado num tabuleiro de 15 linhas por 15 colunas, onde cada jogador coloca uma peça (branca ou preta) e vence o primeiro jogador que fizer uma linha com 5 peças da sua cor. O jogador com as peças pretas é quem inicia o jogo. Pode também ser jogado num papel quadriculado, conforme mostra a figura abaixo.



O padrão MVC

O padrão MVC é um estilo de arquitetura de software no qual a aplicação é composta por 3 módulos (ver material de apoio na página do curso):

- modelo ('M'): responsável pelo gerenciamento dos dados da aplicação.
- apresentação ('V' de 'view') : responsável pela interface (visual) com o usuário.
- controle('C') : responsável pelo tratamento dos eventos gerados pela apresentação e disparo das ações no modelo.

Atividade 8: O Modelo

A classe que terá o papel de modelo no padrão MVC deverá implementar a interface `ModelInterface`, que define os seguintes métodos:

- `public void iniciaPartida()` : dá início a uma nova partida.

- `public boolean salvaPartida(String nome)`: salva a partida atual num arquivo cujo nome é passado como parâmetro. Retorna um valor booleano indicando sucesso ou falha. Sugestão: armazenar no arquivo a sequência de jogadas que levou à configuração da partida a ser salva.
- `public boolean carregaPartida(String nome)`: carrega uma partida previamente salva em arquivo. Retorna um valor booleano indicando sucesso ou falha na operação. Caso siga a sugestão acima, restaurar a configuração da partida consiste na 'execução' das jogadas salvas.
- `public boolean joga(int jogador, int x, int y)`: registra jogada de um jogador (1 ou 2), na linha x e coluna y do tabuleiro. Retorna um valor booleano indicando 'sucesso' (true) ou 'jogada inválida' (false).
- `public int tab(int x, int y)`: retorna um inteiro indicando qual a peça na posição (x,y) do tabuleiro (0: nenhuma, 1: branca, 2: preta). Os valores de x e y indicam a linha e coluna respectivamente, a partir de 0.
- `public int verifica()`: verifica se o jogo tem um vencedor. Retorna 0 indicando 'nenhum', 1 indicando 'jogador 1', ou 2 indicando 'jogador 2'. (por convenção, o jogador 1 usa as peças pretas e o jogador 2, as brancas).

Além desses métodos, a interface define também constantes auxiliares como por exemplo o tamanho do tabuleiro. A classe principal deste módulo deverá se chamar "Model", pois esse é o nome pelo qual será conhecida pelo módulo de testes.

A classe de testes

A classe de testes, `xxxx.java`, disponível junto com este enunciado será usada na avaliação da atividade 8 através do `runCodes`. Ela usa a classe `Model` para a 'execução' de diversas partidas, exercitando todos os métodos previstos na interface.

Itens Opcionais

- Possibilidade de desfazer a última jogada, tantas vezes quanto possível.
-

Desafio

- Versão em que o usuário joga contra o computador (o usuário decide quem começa a partida)

Referências na web

- <http://en.wikipedia.org/wiki/Gomoku>
- <http://gomoku.yjyao.com/>
- <https://docs.oracle.com/javase/tutorial/uiswing/components/filechooser.html>

Data de entrega: 27/05/16