

MC 302EF - Atividade de Laboratório no. 8

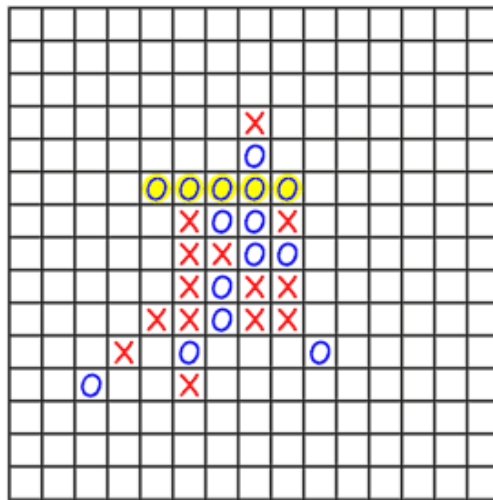
[criado em 04/05/15]

Objetivos

- Desenvolvimento de uma interface gráfica (GUI) em Java
- Uso do padrão MVC
- Tratamento de eventos

Descrição do Problema

Esta atividade consiste no 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.

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.
- **public boolean carregaPartida(String nome):** carrega uma partida previamente salva em arquivo. Retorna um valor booleano indicando sucesso ou falha na operação.
- **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", para realização dos testes pelo Suzy.

A Apresentação

A tela principal do jogo deverá mostrar o tabuleiro 15x15, e botões para disparar as operações pelos jogadores. Uma janela do tipo 'pop-up' deverá ser apresentada quando

- o jogador que fez a última jogada venceu a partida
- a última jogada é inválida
- o jogo termina sem vencedor, por falta de opção.

O Controle

Será o responsável pelo tratamento dos eventos gerados pela interface com o usuário, e quando necessário, ativar os métodos do modelo para atualizar os dados. Será responsável também pela ativação da apresentação (deverá ter, portanto, o método **main()**). A classe principal deste módulo deverá se chamar 'Controller'.

Avaliação

A avaliação desta atividade será feita em 2 etapas:

- Na primeira etapa a classe que implementa a interface **ModelInterface** será testada automaticamente através do Suzy, usando um programa de testes (fornecido como anexo desta atividade).
- Na segunda fase, os programas que tiverem passado pela primeira fase serão testados manualmente a partir da interface de usuário.

Itens Opcionais

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

- Uso de um diálogo (p. ex. 'JFileChooser') para escolher a pasta e o arquivo ao salvar ou restaurar uma partida.

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: 16/05/15