Trabalho de INF1636

09/06/2022

Prof. Ivan Mathias Filho

Instruções para a 4ª iteração

Na 4ª e última iteração devem ser implementadas as funcionalidades relativas ao término de uma partida. Uma partida deverá ser encerrada no momento em que for acionado um botão (ou um item de menu) disponibilizado para tal. Nesse momento será apurado o capital acumulado por cada jogador e definida a posição de cada um deles, em função do capital acumulado. O encerramento de uma partida irá, também, ocorrer caso reste apenas um único jogador, devido à falência dos demais. Neste caso, o jogador restante será o vencedor, não sendo necessário definir as posições dos demais.

Salvamento e Recuperação

O salvamento e a recuperação de uma partida devem seguir as seguintes regras:

1. O salvamento só precisa ser feito após um jogador ter concluído a sua jogada. Isto é, para facilitar o salvamento e a recuperação do estado de uma partida essa opção só precisa ser disponibilizada imediatamente antes de um jogador lançar os dados. Após o lançamento ter ocorrido o botão de salvamento pode ser desabilitado, sendo que a reabilitação só precisará ocorrer imediatamente antes do jogador seguinte lançar os dados;
2. Os dados relativos ao estado de uma partida TÊM de ser gravados em um arquivo texto (.txt) no formato ACII puro (arquivo texto usual);
3. O nome e a localização do arquivo devem ser de livre escolha do usuário. Apenas o filetype pode ser pré-definido;
4. Para que o usuário possa escolher livremente o local do sistema de arquivos em que será gravado/lido o estado de uma partida, o componente Java Swing **JFileChooser** TERÁ DE SER USADO;
5. O diálogo (janela) inicial terá de modificado de modo a permitir que seja possível escolher entre iniciar uma nova partida ou carregar uma partida não concluída. Após essa escolha ter sido feita é que a janela com o tabuleiro será exibida.

As transparências relativas à manipulação de streams de E/S (arquivos) já se encontram disponíveis no EAD (Capítulo 16).

Vale ressaltar que devido à facilitação que o salvamento e a recuperação proporcionam à execução dos testes de aceitação que serão feitos por mim, a **não implementação** dessas funcionalidades **acarretará na perda de 2,0 pontos**, embora o peso delas na nota final do trabalho, caso elas sejam entregues corretamente, vá ser menor.

Escolha do Valor do Dado

Para que as regras do jogo possam ser testadas rapidamente é fundamental que se possa escolher os valores dos dados, contornado, assim, a definição aleatória desses valores.

Fica a sugestão de inserir, logo abaixo da figura do dado, uma combo box contendo os valores inteiros de 1 a 6. Quando se quiser simular o lançamento (aleatório) do dado, o botão será pressionado, mas quando for desejável escolher o valor de um dado, para que se possa testar a ocorrência de determinado evento, essa combo box será acionada. O uso de 6 radio buttons também é uma boa opção.

A não inclusão dessa funcionalidade na versão final do jogo acarretará na perda de **2,0 pontos** na nota final do trabalho.

Arquitetura do Software

A arquitetura do software, bem como as funcionalidades a serem implementas, já foram definidas. Nenhum requisito novo será tratado. Sendo assim, nesta última iteração é importante

- Terminar a integração da View com o Model tendo em vista permitir que todas as regras do jogo possam ser acionadas a partir da interface gráfica;
- Empregar os padrões Singleton, Façade e **OBSERVER** na integração descrita acima;
- Consolidar o papel do Controller, que será o responsável por organizar a sequência de eventos que ocorrem em uma partida de Banco Imobiliário.

Em relação ao padrão **Observer**, cabe destacar que ele **TEM** de ser empregado na atualização da interface gráfica com o objetivo de refletir mudanças de estado do jogo. Por mudança de estado deve-se entender a movimentação dos piões, a distribuição de cartas, o lançamento dos dados ou a contabilização do montante de capital gasto (ou recebido) por um jogador.