Programação Orientada por Objetos 2022/2023

Ficha de Laboratório #13

Objetivos

- Consolidação dos objetivos dos laboratórios anteriores.
- Propriedades avançadas do JavaFX.
- Janelas e Modals

Programa

- Utilização de Propriedades Avançadas.
- Utilização de janelas.
- Gestão de múltiplos ecrãs.

Regras de implementação

- Utilização do IDE BlueJ com o código fornecido.
- Implementar o código necessário e testar no fim de cada nível.
- Use as convenções de codificação adotadas para a linguagem Java (ver **Notas**).

Implementação

Nível 1:

Aceite o Assignment GitHub Classrooms deste laboratório através do link https://classroom.github.com/a/A3cQcQl_

Abra o código template fornecido no repositório do laboratório e analise as classes fornecidas e entenda o funcionamento das mesmas.

Na classe MenuPane:

- Crie um botão chamado de btnStart que tem o texto "Jogar Agora" na função *init()*.
- Altere o tamanho, cor de fundo e cor de letra ao seu gosto.
- Coloque o mesmo no centro do atributo BorderPane da classe MenuPane.

Adicione ainda uma nova instância da classe **InicialMenuBar** e coloque-a no topo do atributo BorderPane da classe MenuPane.

Complete o botão com a ação que é dada a seguir e entenda o seu funcionamento.

```
btnStart.setOnAction(e -> {
    primaryStage.setScene(new GamePane(primaryStage));
});
```

Nível 2:

- Crie uma classe TileButton que extende de uma classe Button do JavaFX, esta classe será a responsável por descobrir as bombas e marcar as bandeirolas.
- Esta classe terá um atributo **Tile** que receberá através do seu construtor.

Adicione o seguinte método *click()*.

```
public void click() {
    if(MinesweeperLogic.getInstance().isFlaggingMode()){
        markFlag();
        return;
    }

    if(tile.isFlagged()) return;
    if(tile.isIsUnveiled()) return ;


    tile.setUnveiled(true);

    setText( tile.getValue() + "");

}

public void changeButtonSize(ImageView image, int xPixels, int yPixels) {
    image.setFitHeight(yPixels);
    image.setFitWidth(xPixels);
}
```

- Crie o método getTile().
- Crie uma função similar á função *click()*, que não retornará valores, e se chamará *markFlag()*. Esta função terá de verificar se o atributo tile está com o atributo flagged com o valor **true** através da função *isFlagged()*.
- Se estiver flagged, altere o texto do botão para "", caso contrário altere para "Flag".
- Finalmente invoque a função *toggleFlagged()* do atributo tile a seguir á alteração do texto do Botão na função *markFlag()*.

Nivel 3

Para completar a **GamePane** que será a Scene que terá o jogo Minesweeper adicione no topo da sua BorderPane uma nova instância da classe **GameMenuBar**. Esta classe terá várias opções e funcionalidades.

Na classe **GameMenuBar** na função *init()*:

- Crie um Menu da classe **Menu**, com o nome "Ficheiro".
- O menu "Ficheiro" terá três **MenuItem** com o nome de "Ajuda", "Menu Inicial" e "Sair"

O MenuItem "Sair" fecha a aplicação, o "MenuInicial" irá navegar para a "MenuPane" e o "Ajuda" irá criar um novo Alert com o texto contido na função getHelpText().

Implemente cada um dos MenuItems, e adicione-os ao **Menu** menuFile. Como pode ver, o Menu "Definições" já está implementado.

Verifique o funcionamento da aplicação. Nesta fase a aplicação estará funcional.

Nível 4

Como pode ver, a aplicação marca as bombas como -1, as posições vazias como 0, as bandeirolas como "flag" e os números como eles mesmos. Mas não apresenta um visual apelativo. Por essa razão:

- Crie um atributo na classe **TileButton** que será do tipo ImageView e terá a foto "Undiscovered.png" como é demonstrado no código seguinte:

```
private ImageView image = new ImageView(new Image("Images/Undiscovered.png"));
```

- Adicione as imagens contidas na pasta ***Images*** substituindo os métodos *setText()* que utilizou no nível 2 deste laboratório nos métodos *click()* e *markFlag()* da classe **TileButton**.

Adicione a lógica necessária para que dependendo do valor da tile, esta irá utilizar a Imagem correta.

Não se esqueça de adicionar a **ImageView** á **TileButton**. Utilize a função dada *changeButtonSize()* com os valores de xPixels e yPixels a 50 para manter os botões com um tamanho fixo.

Nível 5

Para concluir as funcionalidades do jogo, crie uma classe **GameOverStage** que extenderá da classe Stage.

- Crie dois atributos nessa classe, sendo o primeiro uma **Scene** e outro uma **BorderPane**.
- Inicialize cada atributo no construtor da classe. Nota: o atributo **Scene** terá um tamanho de 150 píxeis por 100 píxeis.
- O construtor deverá receber *Stage primaryStage*.
- Crie um método init que receberá como parâmetro a primary Scene e irá criar um nó Text com a string "Game Over", e um Button com a string "Try Again".
- Adicione o texto e o botão ao atributo da classe BorderPane.
- Altere o comportamento setOnAction do botão para utilizar o *Singleton* MinesweeperLogic e executar a função *setGame()*. Ainda dentro do *setOnAction()*, execute a função *close()*.
- Finalmente, altere as propriedades da janela/Stage de modo a não ser possível fechar a janela. Utilize as seguintes funções:

```
initModality(Modality.APPLICATION_MODAL);
initStyle(StageStyle.UNDECORATED);
```

Para concluir, adicione uma instância da classe GameOverStage ao método init da classe **GamePane** onde existe um print na consola de "Bomb!"

Notas:

Para os identificadores siga as convenções adotadas normalmente, em particular:

1. A notação **camelCase** para o nome das variáveis locais e identificadores de atributos e métodos.
2. A notação **PascalCase** para os nomes das classes.
3. Não utilize o símbolo '_', nem abreviaturas nos identificadores.