

INF1636 – P.O.O.

Relatório do Trabalho 1



“Jrobot v1.0”

Grupo:

Diogo Hahlbohm Carneiro
Mauricio Azevedo Lage Ferreira
Savio Sena Machado Barbosa

INTRODUÇÃO:

Neste trabalho, o objetivo era construir um editor gráfico de mapas para gerar cenários que serão utilizados para o jogo. Na implementação usamos as bibliotecas extras jogl e luajava. As classes e interfaces criadas foram resumidas abaixo:

IMPLEMENTAÇÃO:

O Projeto implementa o pacote chamado Jrobot que utiliza os três outros pacotes criados no projeto: jrobot.gui , jrobot.gui.opgl e jrobot.opgl.

1) jrobot.gui

Esse pacote trata a parte de objetos GUI / SWING do software. Ele implementa a interface BarProperty e as classes: AboutBox, FileDialog, MainFrame, MenuBar, NewBox, NewProp, PropTab e ToolBar.

- **BarProperty:**

Essa interface define os métodos que são usados pelo menuBar e pelo ToolBar do aplicativo como addNewtoBar e addOpentoBar.

- **AboutBox:**

Essa classe estende JDialog e cria a janela que exibe as informações sobre a janela About.

- **FileDialog:**

Essa classe estende JFrame e cria as janelas de iteração com o usuário para salvar e carregar os arquivos de mapa implementando os métodos necessários para isso.

- **MainFrame:**

Essa classe estende JFrame e é a principal classe do pacote. Ela instancia todos os objetos do frame como o ToolBar, MenuBar, Canvas e PropBar.

- **MenuBar:**

Essa classe estende JMenuBar implementando BarProperty criando novos métodos para o menuBar do aplicativo colocando funções de File, view e about para o usuário.

- **NewBox:**

Essa classe estende JDialog e criando métodos para o usuário criar um novo mapa capturando Altura, comprimento, pixels da célula e boolean de alongar mapa utilizando as devidas exceções.

- **NewProp:**

Essa classe estende JDialog e criando métodos para o usuário criar uma nova propriedade para o mapa capturando o nome e o seu valor default utilizando as devidas exceções.

- **PropTab:**

Essa classe estende JPanel implementa os métodos do JTabbedPane que contém o panel de propriedades padrão (profundidade, erro e pressão) e o panel de propriedades customizadas. Essa classe reúne os métodos que controla o fluxo de informações das propriedades de cada célula exibindo-as ou alterando-as.

- **ToolBar:**

Essa classe estende JToolBar implementando BarProperty e criando novos métodos para o toolBar instanciando todos os botões usados.

2) jrobot.gui.ogl

Esse pacote trata a parte de objetos GUI / SWING que o canvas Jogl utiliza. O pacote implementa a interface `oglFunctions` e as classes `oglPanel` e `LegendPanel`.

- **oglFunctions:**

Essa interface define os métodos que o canvas do Jogl fornece para o seu usuário, tais métodos que relacionam o `MainFrame` com o canvas de desenho como: `saveMap`, `loadMap`, `closeMap`, `ZoomIn`, `ZoomOut` e `FitToPage` por exemplo.

- **oglPanel:**

A classe `oglPanel` é uma extensão de `JPanel` que implementa a interface `oglFunctions` criando os métodos citados acima.

- **LegendPanel:**

A classe `LegendPanel` é também mais uma extensão de `Jpanel` cria o painel que exibe a legenda do mapa que está sendo visualizado. Cria métodos para trocar a imagem da legenda e do `Label` da legenda.

3) jrobot.ogl

Esse pacote trata das funções de desenho no canvas do jogl. O pacote implementa a interface `oglObjects` e as classes `oglDraw`, `oglRenderer`, `SelectionMatrix`

- **oglObjects:**

Essa interface define as primitivas básicas de desenho criadas para OpenGL utilizadas no editor de mapas.

- **oglDraw:**

Essa classe implementa a interface `oglObjects` criando as primitivas básicas de desenho para `openGL`.

- **oglRenderer:**

É a principal classe do pacote. Essa classe gerencia toda a parte gráfica do `openGL` se preocupando com funções básicas como: `init`, `reshape` e `redraw`. A classe através da interface `oglObjects` desenha na tela as células e pinta as células para simbolizar a seleção. Ela cuida também da parte de interação com o usuário no que diz respeito ao seu canvas controlando eventos de mouse em sua área de desenho, cuidando portanto da parte de seleção e múltiplas seleções de células através de `CTRL` e `SHIFT`.

`oglRenderer` implementa: `GLEventListener`, `MouseListener` e `MouseMotionListener`.

- **SelectionMatrix:**

Essa classe foi criada para manipular operações com a matriz criada para trabalhar em cima da seleção de células no canvas.

4) **jrobot e outros**

Aqui estão listadas as demais classes que fecham o pacote: `GameMap`, `GUILoad`, `Jrobot`, `Utils` e `loadMap.lua`

- **GameMap e loadMap.lua:**

São responsáveis pelo binding de arquivos `.lua` para o tratamento dos arquivos de mapa.

- **GUILoad:**

Classe que instancia o `MainFrame`.

- **Jrobot:**

Classe que contém a main e instancia o aplicativo.

- **Utils:**

Essa classe trata os paths do aplicativo.

MAIS INFORMAÇÕES:

- Mais detalhes sobre a implementação pode ser encontrada no documento Javadoc criado:

<http://www.malf.com.br/jrobot/API>

- Mais informações sobre a instalação do Jogl e LuaJava, e detalhes sobre a compilação do Jrobot:

<http://www.malf.com.br/jrobot/readme.txt>

- Mais informações sobre comandos de teclado:

<http://www.malf.com.br/jrobot/keyboard.txt>