ICS - Trabalho I
MIDI Player

Juarez A.S.F. 11/0032829

Universidade de Brasília

28 de Abril de 2014

Requisitos

- Desenvolver uma interface gráfica em Java que implemente um tocador MIDI com as funcionalidades básicas:
  - Controle Play/Pause
  - Stop
  - Posicionamento do instante inicial
  - Controle de volume
  - Exibição de parâmetros partitura da música que está sendo (ou está para ser) tocada:
    - fórmula de compasso
    - metro
    - andamento
    - armadura de tonalidade
    - indicação em tempo real[hh mm ss]

MidiPlayer

- Encapsula as funcionalidades do tocador
- Utiliza mecanismo de composição e não de herança.
- atributos:
  - File arqMidi
  - Sequencer sequenciador;
  - Sequence sequencia;
  - long tickPosition;
- métodos de controle: play, **pause**, stop, goTo.
- métodos informativos: getDuracao, getResolucao ....

LogWindow

- Estende a classe JScrollPane
- é simplesmente uma tela de texto para reportar mensagens durante o programa. Mensagens de erro são impressas em vermelho, de warning em amarelo....
- atributos:
  - JTextPane logArea
- métodos:
  - report, reportGood, reportBad, reportWarning..

GUI

- Estende a classe JFrame
- contém diversos componentes dos pacotes java.awt e javax.swing
- possui botões de play/pause e stop
- menu 'file' para carregar arquivo midi
- sliders de posicionamento para tempo e volume
- tabela para informar dados da partitura

Componentes

Alguns componentes utilizados na interface:

- **ImageIcon, Icon e JButton** para botões
- **JSlider** para entrada/saída de informações de tempo
- **JLabel e JTable** para informações textuais
- **JMenu, JMenuItem e JMenuBar** para menu
- **JFileChooser** para navegar e carregar arquivo
- **JTimer** para controlar tempo com que o JSlider é atualizado

Composição de Layouts

- Para posicionarmos espacialmente os componentes como desejamos, utilizamos uma composição de layouts.
- O layout mais externo é um BorderLayout

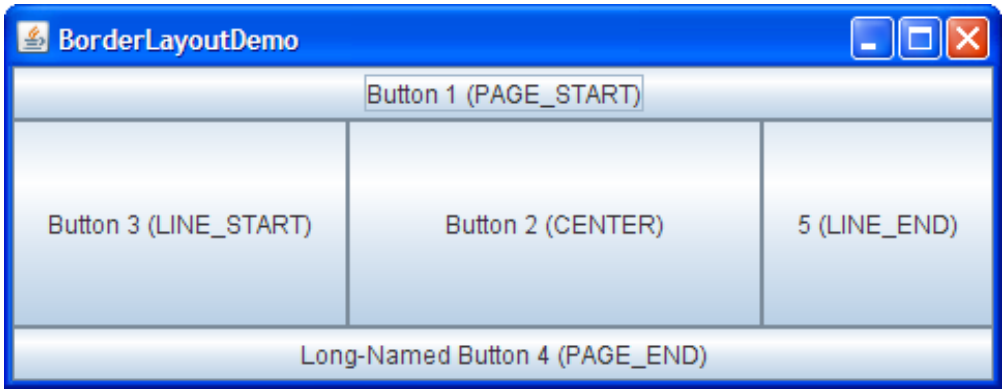

Figura: BorderLayout

- Cada parte desse layout externo possui um layout próprio, como FlowLayout ou GridLayout

Layouts Utilizados

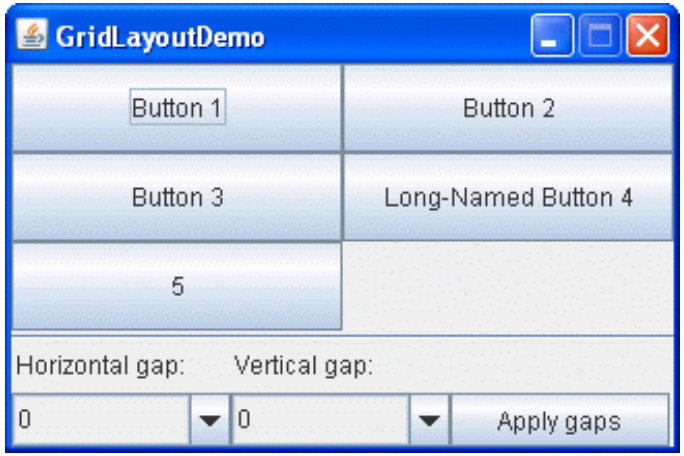

Figura: GridLayout

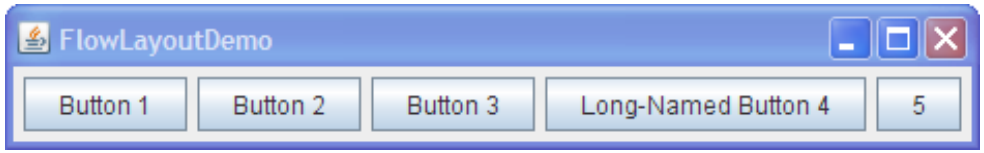

Figura: FlowLayout

Slider de Tempo

- O JSlider usado para mostrar o tempo da música possui duas funcionalidades:
  1. mostrar o tempo
  2. capturar o instante em que o usuário deseja posicionar a música
- O seja, apresentas funcionalidade de entradas e saídas de dados.
- A alteração do seu posicionamento pode ser feita:
  1. **internamente** pela controle da passagem do tempo
  2. **externamente** pelo mouse do usuário ao posicionar o slider ou ao clicar no botão stop.
- Enquanto o usuário estiver posicionando o Slider, a música continua tocando, mas deve-se desabilitar a passagem do tempo.

Slider de Tempo

- Criamos uma variável controleExterno que é setada verdadeira quando o usuário está segurando o cursor do slider(isMoving).
- Enquanto essa variável está setada, ignoramos as mudanças de relógio
- Quando o usuário solta o cursos, posicionamos a música no instante desejado e setamos essa variável de controle para falso novamente.

Controle de Pause

- A classe de sequenciador MIDI implementada em Java apresenta as funcionalidades:
  1. start: inicia a sequência de onde estiver
  2. stop: para a sequência
  3. goto: posiciona a sequência no número de tiques desejado
- Para implementar a função de stop devemos parar com stop e setar a posição(em tiques) para zero.

Load SoundBank

- É preciso manter uma instância do SF atual

```
//no construtor
    banco = MidiSystem.getSynthesizer().getDefaultSoundbank();
```

- É preciso descarregar todo o SF antigo

```
     Synthesizer sintetizador = MidiSystem.getSynthesizer();
sintetizador.unloadAllInstruments(banco);
```

- Carregar novo banco de som

```
     Soundbank novo_banco = MidiSystem.getSoundbank(new File(sf_adress));
     sintetizador.loadAllInstruments(novo_banco);
banco = novo_banco;
```