

# ICS - Trabalho II

## Síntese Aditiva

Juarez A.S.F. 11/0032829

Universidade de Brasília

2 de Junho de 2014

- 1 Requisitos
- 2 Classes Desenvolvidas
  - Dispositivo RAN
  - Instrumento 1
  - Instrumento 2
  - Instrumento 3
- 3 Interface Gráfica em JAVA
  - Componentes Utilizados
  - Layout
- 4 Problemas Encontrados

# Requisitos

- Desenvolver uma interface gráfica em Java que implemente três instrumentos aditivos(ver figura a seguir) e seja capaz de tocar melodias e notas com ele. A interface deve possuir operações de:
  - Música
  - Nota

# Dispositivo RAN

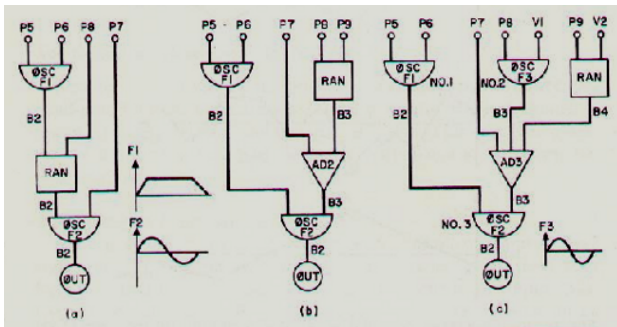


Figura: Instrumentos a serem implementados

- 1 Requisitos
- 2 Classes Desenvolvidas
  - Dispositivo RAN
  - Instrumento 1
  - Instrumento 2
  - Instrumento 3
- 3 Interface Gráfica em JAVA
  - Componentes Utilizados
  - Layout
- 4 Problemas Encontrados

# RAN

- Vemos pelas especificações que **RAN precisa aceitar** como entrada tanto uma **constante** como um **dispositivo**.
- Dentre as classes disponíveis na API SomA, a única que faz isso é **Oscilador**
- Essa é a escolha natural para ter como classe base
- Gostaríamos de poder setar a tabela SIN da classe oscilador para aquela tabela gerada com números aleatório. Como isso não é possível, **foi necessário uma gambiarra**.
- **A frequência do sin é setada para 0 e sua fase para 90**. Desta forma temos um sinal constante em  $+1$
- a entrada da amplitude é um dispositivo multiplicador
- as entradas desse multiplicador são duas envoltórias: Uma é gerada aleatoriamente em  $-1$  e  $+1$  e a outra é uma envoltória de amplitude gerada pelo usuário

# Dispositivo RAN

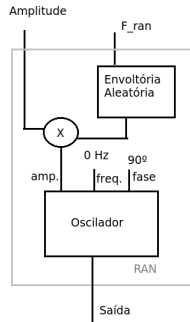


Figura: Dispositivo RAN: esquemático da construção

# RAN

Classe desenvolvida para gerar a envoltória aleatória.

- extends **Oscilador**
- entradas controláveis:
  - `f_ran`: define o número de amostras aleatórias geradas no intervalo
  - `a`: define a amplitude da envoltória gerada. Isto é, os números aleatórios vão de  $-A$  até  $+A$ . Pode ser uma constante ou um objeto `Envoltoria`.
- fórmula utilizada para gerar números entre  $-A$  e  $+A$  :

```
float random = 2f*A*((float)Math.random()) - A;
```

- possui método de visualização gerado com o pacote **jmathtool**
- Toda vez que um dos dois parâmetros acima é atualizado, uma nova envoltória é gerada. Pode-se também gerar uma nova envoltória a qualquer momento.



# Dispositivo RAN

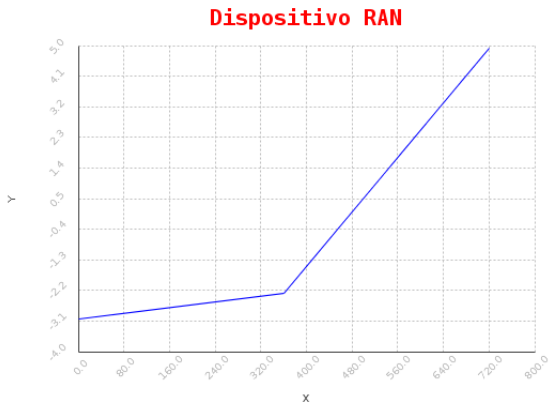


Figura: Dispositivo RAN,  $f_{\text{ran}} = 2$ ,  $A = 10$

## Dispositivo RAN

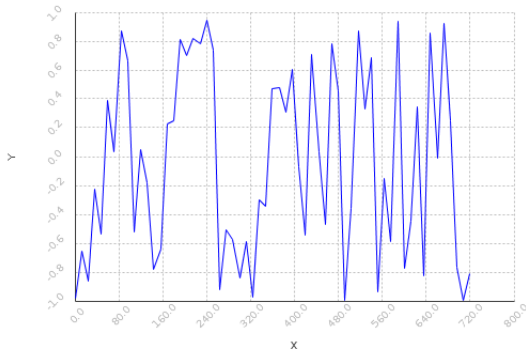


Figura: Dispositivo RAN,  $f_{\text{ran}} = 60$ ,  $A = 1$

## Dispositivo RAN - Detalhes

- é necessário alterar a frequência do ruído quando a duração da nota for alterada. Pois a frequência calcula pontos no intervalo total da duração, mas o que importa é a frequência por segundo!
- precisamos então de uma variável para `f_ran_atual` e outra para `f_ran_base`

# Instrumento 1

- Extends **UnidadeH**
- Como a unidade H é o menor instrumento possível, nossos instrumentos são todos descendentes dela.
- Criamos uma unidadeH com **super()** e pegamos o seu oscilador.
- setamos então a entrada de ganho o oscilador para um dispositivo **RAN**
- redefinimos então os métodos **setGanho** e **setDuracao** para atuarem no objeto **RAN**

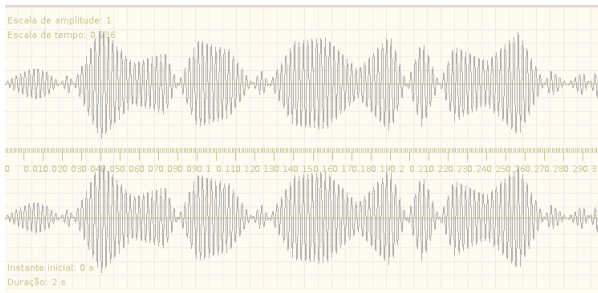


Figura: Saída do Instrumento 1

# Instrumento 2

# Instrumento 3

- 1 Requisitos
- 2 Classes Desenvolvidas
  - Dispositivo RAN
  - Instrumento 1
  - Instrumento 2
  - Instrumento 3
- 3 Interface Gráfica em JAVA
  - Componentes Utilizados
  - Layout
- 4 Problemas Encontrados



# Componentes

Alguns componentes utilizados na interface:

- **Imagelcon, 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

- 1 Requisitos
- 2 Classes Desenvolvidas
  - Dispositivo RAN
  - Instrumento 1
  - Instrumento 2
  - Instrumento 3
- 3 Interface Gráfica em JAVA
  - Componentes Utilizados
  - Layout
- 4 Problemas Encontrados