

# Sistemas Musicais Interativos

Mariana Zapparoli Martins

Computação Musical – IME USP

# Roteiro

- Sistemas Musicais Interativos
  - Introdução
  - Máquina com conhecimento musical
  - Classificação
  - Score Followers
  
- Fundamentos
  - Padrão MIDI

# Roteiro

- Etapas de processamento

- *Sensing*

- *Processing*

- *Response*

- Bibliografia

# Introdução

- Sistemas Interativos  $\Leftrightarrow$  entrada
- Três campos de impacto
  - Teoria Musical
  - Cognição Musical
  - Inteligência Artificial
- Max
- Cypher

# Máquina Musicianship

- Qual o propósito musical?
- Interpretação da entrada
  - O que a máquina pode ouvir?
  - Abstração fornecida pelo padrão MIDI
  - Análise do sinal de áudio (informações adicionais)
- Representações baixo-nível
  - interpretação da entrada é indispensável

# Máquina Musicianship

- Interpretação de uma entrada musical
  - Coleções de conceitos
  - Métodos para construção de respostas
- Possibilidade de eliminação do músico
  - Situação desejada?

# Máquina Musicianship

- Composição por Refinamento
  - Reação imediata
  - Refinamento contínuo de idéias iniciais
- Sistemas Interativos vistos como Teorias Musicais Aplicadas
  - Teoria musical descreve o processo de compor ou ouvir música

# Classificação

- Reconhecimento de similaridades entre os Sistemas e identificação das relações entre eles
- Classificação construída sob a combinação de três dimensões:
  - Tipos de interpretação da entrada
  - Métodos de resposta
  - Modelos de abordagem

# Classificação

- Tipos de Interpretação da Entrada
  - Programas Score-Driven
    - Coleção de eventos pré-determinados armazenados
    - Casamento com a entrada que chega
    - Categorias tradicionais de batida, tempo
  - Programas Performance-Driven
    - Não antecipam a realização de qualquer partitura particular armazenada

# Classificação

- Não possuem uma representação da música armazenada
- Utilizam normalmente parâmetros mais gerais envolvendo medidas de percepção: densidade, regularidade

## ■ Métodos de Resposta

### ■ Métodos Transformativos

- Aplicação de transformações na entrada que chega
- Fonte é a entrada musical completa [armazenada]

# Classificação

- Algoritmos Generativos
  - Material fonte elementar ou fragmentado
  - Aplicação de conjuntos de regras no material armazenado
  - Produzem uma saída musical completa
- Técnicas Sequenciadas
  - Usam fragmentos pré-gravados em resposta a entrada em tempo-real
  - Alguns aspectos desses fragmentos podem ser variados

# Classificação

## ■ Modelos

### ■ Instrumento

- Instrumento musical extendido
- Saída elaborada de acordo com gestos do tocador
- Um músico => saída será um solo

### ■ Tocador (*Player*)

- Construção do tocador
- Presença musical com personalidade e comportamento
- Um músico => saída será um dueto

# Score Followers

- Grupo de programas capaz de acompanhar um solista
  - Casa a entrada com a partitura armazenada
  - Gera o acompanhamento (também armazenado)
  - Programas Score-Driven
  - Técnica de respostas sequenciada
  - Modelo Tocador

# Fundamentos

- Dois pilares básicos dos sistemas interativos
  - Manipulação MIDI (*Musical Instrument Digital Interface*)
    - Comunicação entre computadores, sintetizadores e controladores
  - Escalonador (*Scheduling*)
    - Execução de tarefas em pontos de tempo específicos
    - Música é uma arte temporal
    - Necessidade de facilidades na representação do tempo

# Fundamentos

## ■ Padrão MIDI

- Especificação de hardware e protocolo de comunicação
  - Troca de informações entre computadores, sintetizadores e controladores
- Abstração – representação mais alto nível
  - Baseada no conceito de notas e a velocidade
  - Notas como *switches*
    - Pressionada => mensagem *Note On* (indica nota e velocidade)
    - Solta => mensagem *Note Off* (*Note On* com valor zero)

# Fundamentos

- Limitações do padrão MIDI
  - Banda de Transmissão em 31.250 bits por segundo
  - Cada byte (8 bits) é circundado por um bit de *start* e um bit de *stop*, totalizando efetivamente 10 bits
  - Uma mensagem padrão *Note On* requer 3 bytes (30 bits) de informação, levando aproximadamente 1 milissegundo para ser transmitido
  - Acorde de 10 notas em 10 canais => 100 milissegundos de atraso entre a primeira e a última nota serão percebidos

# Fundamentos

- Controladores MIDI como transdutores de gestos
  - Teclados são muito bem representados
  - Outros instrumentos não são tão bem representados (cordas, sopros, voz)
  - Estímulo para pesquisa, que trouxe crescimento na área

# Fundamentos

- Estágios da cadeia de processamento de sistemas interativos de computação musical:
  - *Sensing*
    - Coleta de dados
  - *Processing*
    - Interpretação dos dados
  - *Response*
    - Produção do som

# Fundamentos

## ■ *Sensing*

- Coleta de dados utilizando controladores MIDI
- Recursos com vários modos de operação que podem ser programados
- Passagem dos dados para o próximo estágio, o *Processing*

# Fundamentos

## ■ *Processing*

### ■ Recebe e empacota comandos MIDI

- *Driver* => armazena um fluxo contínuo de informação serial vinda da porta física
- Introdução de *time stamps*, indicações do tempo da chegada de algum pacote MIDI
- Função de escalonamento de tempo-real

# Fundamentos

- Escalonador em tempo-real
  - Processo no qual uma ação será executada em algum ponto no tempo futuro
  - Pode-se atrasar a execução de um procedimento
  - Os argumentos da rotina a ser executada são salvos junto com o nome da própria rotina

# Fundamentos

- Exemplo:

- TaskPtr

Scheduler\_CreateTask(time, tol, imp, per, fun, args)

long time

short imp, tol, per

void (\*fun)()

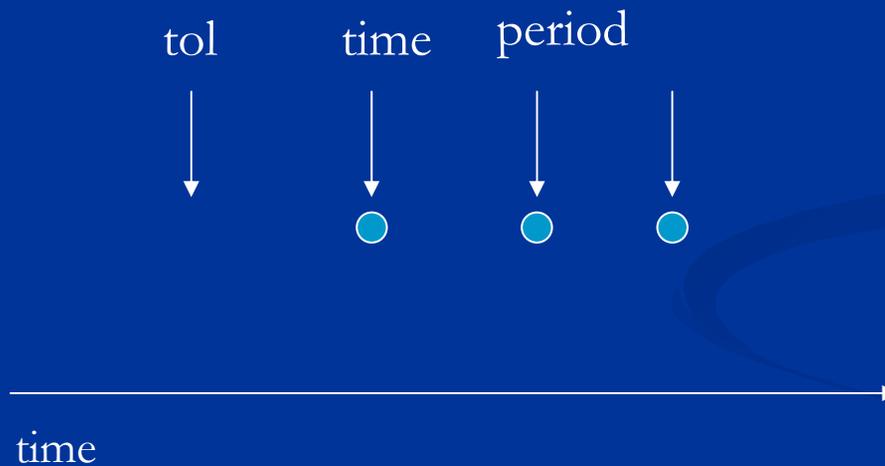
arglist arg

# Fundamentos

- Argumentos
  - *Time*: tempo absoluto (*clock*) em centisegundos
  - *Tol*: tolerância para o início da função
  - *Per*: período. Se este argumento for diferente de zero, a tarefa irá se reescalonar *per* centisegundos depois de cada execução
    - Tarefas periodicamente escalonadas podem ser encerradas por um comando explícito *kill*

# Fundamentos

## ■ Ilustração:



# Fundamentos

- Escalonador mantém três filas de prioridades
  - Tarefas nas filas de alta-prioridade executadas primeiro
  - Argumento *imp* (importância) determina qual prioridade a tarefa indicada irá receber
  - *Fun*: ponteiro para a função a ser chamada
  - *Args*: argumentos da função. Estes são avaliados no tempo escalonado, ao invés de quando a função foi realmente invocada

# Fundamentos

- Assume-se que nenhuma rotina escalonada irá requerer um longo tempo de processamento
- Procedimento escalonado executa até o fim
- Número estritamente limitado de formas de interrupção
- Requisição de mais tempo de CPU: reescalamento para garantir a execução de outras tarefas na fila

# Fundamentos

## ■ *Response*

- Comunicação através do padrão MIDI
- Saída dependente dos recursos utilizados
- Recursos MIDI tendem a cair em dois grupos
  - Síntese
  - Amostragem
- Módulo de síntese usam um algoritmo pra produzir som
  - Exemplo: modulação de frequência

# Fundamentos

- Módulos de amostragem
  - Possuem formas de ondas armazenadas
  - Normalmente gravadas de instrumentos tradicionais
  - Notas específicas são tocadas em resposta a mensagem MIDI
  - Comandos são mandados aos recursos, incluindo mensagens *Note On/Off* e valores de controle

# Bibliografia

- Rowe R. *Interactive Music Systems*. MIT Press, 1994.
- Orio N., Lemouton S., Schwarz D. *Score Following: State of the Art and New Developments*. Proceedings da Internacional Computer Music Conference (ICMC), 2003. Disponível em [http://suac.net/NIME/NIME03/NIME03\\_Orio.pdf](http://suac.net/NIME/NIME03/NIME03_Orio.pdf)

# Bibliografia

- Thom B., Nelson M. *An In-Depth Analysis of Real-Time MIDI Performance*. Proceedings da Internacional Computer Music Conference (ICMC), 2004, páginas 430 - 437. Disponível em [http://www.cc.gatech.edu/~mnelson/papers/Thom\\_and\\_Nelson\\_-\\_MIDI\\_Performance\\_-\\_ICMC04.pdf](http://www.cc.gatech.edu/~mnelson/papers/Thom_and_Nelson_-_MIDI_Performance_-_ICMC04.pdf)