

Csound para Android

Como utilizar csd's em dispositivos móveis

Antonio Deusany de Carvalho Junior - dj@ime.usp.br

IME - USP

Outubro 1, 2013

Introdução

Histórico

Csound

Android e Computação Musical

Suporte para o processamento de áudio

Csound para Android

Solução para não desenvolvedores

Solução para desenvolvedores

Histórico de linguagens voltadas à Computação Musical

- ▶ Em 1957 no Bell Labs, Max Mathews iniciou o modelo MUSIC-N
 - ▶ Max Mathews é considerado o Pai da Computação Musical
- ▶ Várias linguagens foram desenvolvidas a partir deste modelo:
 - ▶ Music I, II, III, IV, IV-B, IV-BF, e Music V
 - ▶ Music 360 (para o 360 no fim dos anos 60)
 - ▶ Music 11 (para o PDP-11 em 1973)
- ▶ As duas últimas linguagens citadas foram desenvolvidas por Barry Vercoe, que partiu delas e escreveu Csound
 - ▶ Barry Vercoe é o Pai da Csound

Histórico da Csound

- ▶ Csound foi desenvolvida no MIT por volta de 1986 por Barry Vercoe, inicialmente com ajuda de alguns estudantes:
 - ▶ Kevin Peterson, Alan Delespinase, Bill Gardner, Dan Ellis e Paris Smaragdis
- ▶ O primeiro paper sobre a linguagem foi:
 - ▶ Vercoe, B. & Ellis, D. "Real-time Csound: Software Synthesis with Sensing and Control", ICMC 1990
- ▶ Um dos trabalhos mais recentes publicados sobre ela:
 - ▶ Fitch, J. & Lazzarini, V. & Yi, S. "Csound6: old code renewed", LAC 2013

Histórico da Csound

- ▶ Continua em desenvolvimento até os dias atuais
- ▶ Desde 2005 existe o Csound Journal onde são publicadas as novidades sobre a linguagem
- ▶ Grandes pesquisadores contribuíram com seu desenvolvimento:
 - ▶ John ffitch, Istvan Varga, Gabriel Maldonado, Robin Whittle, Richard Karpen, Michael Gogins, Matt Ingalls, Steven Yi, Richard Boulanger, e Victor Lazzarini
- ▶ A linguagem serviu como base para o *Structured Audio component* do padrão MPEG-4, sendo 10 vezes mais compacto que o MP3
- ▶ Csound foi selecionada para o projeto OLPC

Csound

- ▶ Linguagem de programação voltada para som, processamento de sinais de áudio, computação musical, *live coding*, etc
- ▶ Site oficial com links para documentação e código fonte:
 - ▶ <http://www.csounds.com/>
- ▶ Escrita em C

Csound

- ▶ O código é estruturado utilizando *markup language tags*
- ▶ Há dois tipos diferentes de entradas/estruturas de código:
 - ▶ *orchestra*
 - ▶ descreve os parâmetros e os instrumentos
 - ▶ é traduzida pelo compilador
 - ▶ *score*
 - ▶ descreve as notas e define ao longo do tempo
 - ▶ pode ser compilada para controlar a síntese
 - ▶ pode ser substituída por eventos em tempo real providos de MIDI, OSC, frontends, etc

```
<CsoundSynthesizer>
  <CsOptions>
    csound -W -d -o tone.wav
  </CsOptions>
  <Csinstruments>
    sr = 96000;
    kr = 9600;
    ksmpr = 10;
    nchnls = 2;
    instr 1
    a1 oscil p4, p5, 1;
    out a1;
    endin
  </Csinstruments>
  <CsScore>
    f1 0 8192 10 1;
    i1 0 1 20000 1000;
    e
  </CsScore>
</CsoundSynthesizer>
```

Csound 6

- ▶ A última versão é Csound 6
 - ▶ Apresenta aproximadamente 1700 *unit generators (opcodes)*
 - ▶ Tanto a *orchestra* quanto o *score* podem ser alterados em tempo real, permitindo *true "live coding"*

Suporte para o processamento de áudio

Suporte para o processamento de áudio

- ▶ OpenSL
- ▶ Csound
- ▶ Pure Data
- ▶ Processing
- ▶ SuperCollider
- ▶ Csound (Csound + Processing)
- ▶ Java*

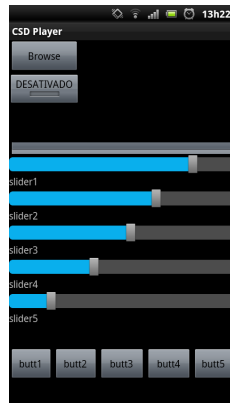
Desenvolvimento

- ▶ Suporte feito por Victor Lazzarini
- ▶ Desenvolvido usando JNI e OpenSL
- ▶ Suporta interação em tempo real
- ▶ Apresenta soluções para desenvolvedores e não desenvolvedores

Solução para não desenvolvedores

Solução para não desenvolvedores

- ▶ Usar o CSD Player
- ▶ Adicionar seu .csd no sdcard
- ▶ Carregar o CSD
- ▶ Play!



Solução para não desenvolvedores

- ▶ Configurações de saída de áudio
 - ▶ Alterar o cabeçalho do arquivo .csd

```
<CsOptions>  
-odac -+rtaudio=null -d  
< /CsOptions>
```

- ▶ Configuração para entrada de dados
 - ▶ Configurar canais para receber valores de variáveis da interface

```
k1 chnget "trackpad.x"  
k2 chnget "slider1"  
k3 chnget "button1"  
k4 chnget "accelerometerX"
```

Solução para não desenvolvedores

► Prática!

► Acesse o link:

► <http://bit.ly/djcsound>

► Faça o Download dos arquivos:

► android-wgflute.csd

► CSD Player

► Instale o CSD Player e carregue o arquivo android-wgflute.csd

Solução para desenvolvedores

- ▶ Utilizando o projeto CsoundAndroid
- ▶ Baseando-se em CsoundAndroidExamples

Utilizando o projeto CsoundAndroid

- ▶ Baixar o projeto CsoundAndroid
- ▶ Importar no projeto do App para Android
- ▶ Adicionar .csd em */res/raw/*
- ▶ Criar um objeto da classe CsoundObj para enviar *scores*
- ▶ Carregar .csd
- ▶ Enviar *score* através do objeto csound

Baseando-se em CsoundAndroidExamples

▶ Exemplos disponíveis:

- ▶ Simple Test 1
- ▶ Simple Test 2
- ▶ Button Test
- ▶ Ping Pong Delay
- ▶ Harmonizer
- ▶ Accelerometer
- ▶ Csound Haiku IV
- ▶ Multitouch XY

Referências

- ▶ Csound - Site oficial
 - ▶ <http://www.csounds.com/>
- ▶ Csound FLOSS Manual
 - ▶ <http://en.flossmanuals.net/csound/>
- ▶ Csound Journal
 - ▶ <http://www.csounds.com/journal/>
- ▶ Csound Conference
 - ▶ <http://csound.org/>
- ▶ Android CSD Player
 - ▶ <http://sourceforge.net/projects/csound/files/csound5/Android/>
 - ▶ http://www.csounds.com/journal/issue17/android_csd_player.html
 - ▶ <http://audioprograming.wordpress.com/2012/03/16/a-general-purpose-ui-for-csound-on-android/>

Solução para desenvolvedores

FINE



Csound para Android

Como utilizar csd's em dispositivos móveis

Antonio Deusany de Carvalho Junior - dj@ime.usp.br

IME - USP

Outubro 1, 2013