

Encapsulando patches Pure Data com a libpd

Pedro Bruel

Grupo de Computação Musical IME/USP

pedro.bruel@gmail.com

22 de outubro de 2013

Estrutura

- 1 Pure Data
- 2 Exemplos de Aplicações
- 3 Encapsulando o Pd
- 4 Exemplos Práticos
- 5 Referências

A linguagem

Linguagem de programação visual, voltada a aplicações interativas musicais e multimídia.

A linguagem

Linguagem de programação visual, voltada a aplicações interativas musicais e multimídia.

Projeto de código aberto, sob licença BSD.

Paradigmas

A entrada de um programa em Pure Data é tratada como um fluxo de informação, que é direcionado e processado em blocos, produzindo uma saída em tempo real.

Paradigmas

A entrada de um programa em Pure Data é tratada como um fluxo de informação, que é direcionado e processado em blocos, produzindo uma saída em tempo real.

A linguagem fornece abstrações de alto nível que encapsulam diversas funcionalidades, como operações matemáticas, de entrada/saída, e outras operações sobre sinais.

Paradigmas

Um programa é composto pela conexão dessas funcionalidades, ou objetos, e é chamado de patch.

Paradigmas

Um programa é composto pela conexão dessas funcionalidades, ou objetos, e é chamado de patch.

A combinação dos paradigmas de programação visual e controle de fluxo de informação também é encontrada nos softwares proprietários MAX e KYMA.

Implementação

- Frequência padrão de amostragem de 44100Hz;
- Tamanho padrão do bloco é 64 amostras;
- Números representados em float 32 bits;
- Tipos: Mensagens, Objetos, Átomos, Comentários e Símbolos;

Implementação

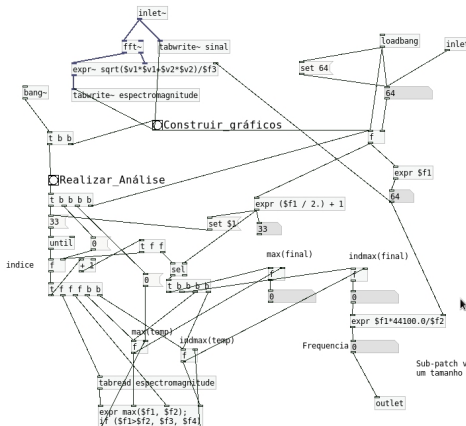
- Frequência padrão de amostragem de 44100Hz;
- Tamanho padrão do bloco é 64 amostras;
- Números representados em float 32 bits;
- Tipos: Mensagens, Objetos, Átomos, Comentários e Símbolos;
- Alta modularização e extensibilidade;
- Execução independente.

Patches



Figura: Um patch simples em Pure Data.

Patches



Sub-patch visto em aula, modificado para receber um sinal e um tamanho de bloco, e devolver a frequência estimada.

Figura: Outro patch, não tão simples.

Para saber mais

- Manual **FLOSS**: flossmanuals.net/pure-data/
- Código Fonte:

```
$ git clone git://git.code.sf.net/p/pure-data/pure-data
```



[Puckette, Miller and others](#)

Pure Data: another integrated computer music environment

Proceedings of the Second Intercollege Computer Music Concerts 37–41



[Puckette, Miller \(1988\)](#)

The Patcher

The-International

Exemplos

Apresentação de alguns exemplos que ilustram usos do Pure Data.

Reactable

Site: reactable.com

- Um instrumento musical eletrônico com interface tangível, em formato de mesa.

Reactable

Site: reactable.com

- Um instrumento musical eletrônico com interface tangível, em formato de mesa.
- Desenvolvido pela Universidade de Pompeu Fabra, em Barcelona.[Jorda et al., 2005]

Reactable

Site: reactable.com

- Um instrumento musical eletrônico com interface tangível, em formato de mesa.
- Desenvolvido pela Universidade de Pompeu Fabra, em Barcelona.[Jorda et al., 2005]
- Engine de som baseada em Pure Data e SuperCollider.

Reactable

Site: `reactable.com`

- Um instrumento musical eletrônico com interface tangível, em formato de mesa.
- Desenvolvido pela Universidade de Pompeu Fabra, em Barcelona.[Jorda et al., 2005]
- Engine de som baseada em Pure Data e SuperCollider.

O framework de visão computacional utilizado pelo projeto é software livre, e pode ser encontrado em:

`reactivision.sourceforge.net/#files`

Reactable



Figura: Reactable

▷ Reactable Basic Demo #1

youtube.com/watch?v=0h-RhyopUmc

Site: rjdj.me

- Uma startup de 2008, que produz aplicativos musicais para celular.

Site: rjdj.me

- Uma startup de 2008, que produz aplicativos musicais para celular.
- Em parceria com compositores, produz experiências musicais, que chama de Reactive Music.

Site: rjdj.me

- Uma startup de 2008, que produz aplicativos musicais para celular.
- Em parceria com compositores, produz experiências musicais, que chama de Reactive Music.
- Utiliza o Pure Data como biblioteca para processamento digital de sinais.

NRCI

Networked Resources for Collaborative Improvisation

Site: `ccrma.stanford.edu/~cburns/NRCI`

- Uma biblioteca de ferramentas desenvolvidas para o Pd.

NRCI

Networked Resources for Collaborative Improvisation

Site: `ccrma.stanford.edu/~cburns/NRCI`

- Uma biblioteca de ferramentas desenvolvidas para o Pd.
- Voltada para a produção e performance de música em grupo e live coding.

NRCI

Networked Resources for Collaborative Improvisation

Site: `ccrma.stanford.edu/~cburns/NRCI`

- Uma biblioteca de ferramentas desenvolvidas para o Pd.
- Voltada para a produção e performance de música em grupo e live coding.
- Desenvolvido pelo Center for Computer Research in Music and Acoustics, de Stanford. [Burns et al., 2008]

Prototipagem

Utilização do Pd para rápida prototipagem de engines de som, por produtoras de jogos eletrônicos.[Jolly, 2011]

Prototipagem

Utilização do Pd para rápida prototipagem de engines de som, por produtoras de jogos eletrônicos.[Jolly, 2011]

Desacopla o trabalho de designers de som e desenvolvedores.

Encapsulando o Pure Data

Os exemplos apresentados utilizam o Pure Data em sua implementação, porém, são aplicações que não dependem da interface gráfica ou controle de execução do Pd.

Encapsulando o Pure Data

Os exemplos apresentados utilizam o Pure Data em sua implementação, porém, são aplicações que não dependem da interface gráfica ou controle de execução do Pd.

Um dos modos de se fazer isso é através do uso da libpd, uma pequena interface para encapsulamento do Pure Data.
[Brinkmann et al., 2011]

Motivação

O Pure Data fornece um ambiente de desenvolvimento capaz de execução independente, e ferramentas potentes voltadas a aplicações sonoras e musicais.

Motivação

O Pure Data fornece um ambiente de desenvolvimento capaz de execução independente, e ferramentas potentes voltadas a aplicações sonoras e musicais.

Porém, essas características estão amarradas a interfaces de usuário e APIs de áudio que são direcionadas a certos formatos de aplicação.

Conceito

A libpd permite utilizar patches e funcionalidades do Pure Data no contexto de outras aplicações.

Conceito

A libpd permite utilizar patches e funcionalidades do Pure Data no contexto de outras aplicações.

Com este objetivo, a biblioteca encapsula e simplifica a interface do Pure Data com o desenvolvedor.

Conceito

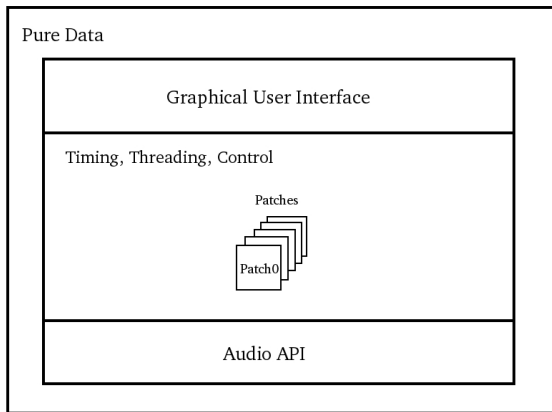


Figura: Ambiente Pure Data.

Conceito

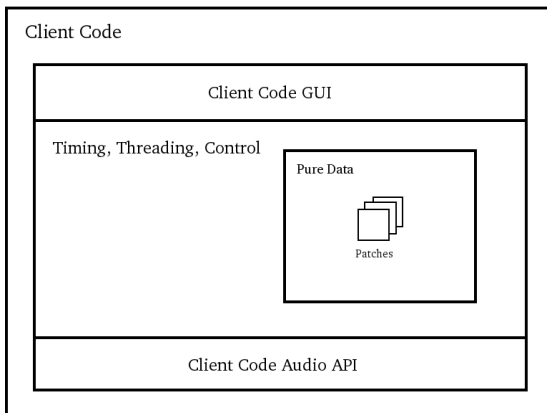


Figura: Pure Data encapsulado em código cliente.

Implementação

- A interface **m_pd.h**, do Pure Data, contém:
 - Definição da API completa do Pd.
 - Funções relacionadas à interface gráfica, controle temporal, threading.
 - Definição de diferentes tipos e estruturas.

Implementação

- A interface **m_pd.h**, do Pure Data, contém:
 - Definição da API completa do Pd.
 - Funções relacionadas à interface gráfica, controle temporal, threading.
 - Definição de diferentes tipos e estruturas.
- A interface **z_libpd.h**, da libpd, contém:
 - A interface **m_pd.h**.
 - Funções que encapsulam a inicialização do Pd, abertura de patches, métodos **process**.
 - Definições dos ponteiros para funções que serão chamadas pelo Pd (callback assíncrono).

Implementação

Uma aplicação que utiliza a libpd deve se preocupar com a inicialização do Pd e de suas funções para callback, e com a chamada dos métodos de processamento nos momentos em que precisar de amostras de áudio.

Implementação

Uma aplicação que utiliza a libpd deve se preocupar com a inicialização do Pd e de suas funções para callback, e com a chamada dos métodos de processamento nos momentos em que precisar de amostras de áudio.

O código cliente pode tratar um patch como uma "caixa-preta" que recebe e devolve amostras e dados, desde que o patch respeite convenções de símbolos *send* e *receive*.

Implementação

Ao encapsular a interface **m_pd.h**, a libpd permite a utilização patches e funções das bibliotecas do Pure Data no contexto de aplicações em diferentes linguagens e plataformas.

Implementação

Ao encapsular a interface **m_pd.h**, a libpd permite a utilização patches e funções das bibliotecas do Pure Data no contexto de aplicações em diferentes linguagens e plataformas.

Nesse processo, são removidas algumas das características que dão independência à execução do Pd, e torna-se mais fácil utilizar patches como engine de som, biblioteca DSP, e interface com o usuário.

Para saber mais

- Livro: Making Musical Apps - Peter Brinkmann
- Site: libpd.cc
- GitHub: github.com/libpd
- Código Fonte:

```
$ git clone https://github.com/libpd/libpd.git
```

Exemplos Práticos

Implementações simples da libpd em C, Java e Java no Android.

- Código (C e Java):

```
$ git clone https://github.com/phrb/libpd_tutorials.git
```

- Código (Android):

```
$ git clone https://github.com/deusanyjunior/Sensors2PD.git
```

Exemplos Práticos

- Estrutura do código da libpd;
- Como compilar a libpd.

Exemplo Prático em C

Uma implementação básica da libpd em C deve conter:

- Definição de taxa de amostragem e tamanho de bloco;
- Declaração dos buffers de entrada/saída de áudio;
- Definição das funções para callback do Pd;
- Mensagem para iniciar o ciclo DSP do Pd;
- Chamada aos métodos de inicialização:
*libpd_set_*hook; libpd_init; libpd_init_audio; libpd_openfile;
libpd_closefile;*
- Chamadas ao método *process* adequado.

Exemplo Prático em C

- Compilando o exemplo.

Exemplo Prático em Java

Uma implementação básica da libpd em Java deve conter:

- Definição de taxa de amostragem e tamanho de bloco;
- Declaração dos buffers de entrada/saída de áudio;
- Implementação da classe PdDispatcher, com as funções de callback;
- Mensagem para iniciar o ciclo DSP do Pd;
- Chamada aos métodos de inicialização da classe PdBase:
openAudio; computeAudio; setReceiver; openPatch; release;
pollPdMessageQueue;
- Chamadas ao método *process* adequado.

Exemplo Prático em Java

- Configurando o Build Path (Eclipse).

Exemplo Prático em Android

- Download e instalação: Sensors2PD.apk;
- Carregar patches Pd no Android.



Jolly, Kent (2011)

Usage of Pd in Spore and Darkspore

Proceedings of the Pure Data Convention 36–39



Brinkmann et al. (2011)

Embedding pure data with libpd

Proceedings of the Pure Data Convention



Jorda et al. (2005)

The reactable

*Proceedings of the international computer music conference (ICMC 2005),
Barcelona, Spain 579–582*



Burns et al. (2008)

NRCI: Software Tools for Laptop Ensemble

Proceedings of the International Computer Music Conference

Obrigado!