

Termpot: criação, edição e execução de funções no navegador em tempo de execução .

Guilherme Lunhane¹ , Flávio Luiz Schiavoni²

¹Instituto de Artes e Design – Universidade Federal de Juiz de Fora
Juiz de Fora, MG

gcravista@gmail.com, ttonmeister@gmail.com

²Departamento de Computação – Universidade Federal de São João Del Rei
São João Del Rei, MG

fls@ufsj.edu.br

Abstract. *This panel describes a sound synthesis software, developed according to an interpretation of Max Mathews's GROOVE. From the perspective of a technology called Web Audio, and another software called Wavepot, we named our software Termpot. The last software allows to create, and encapsulate algebraic definitions in a single data vector. The creation/edition function's process occurs during system's runtime. In this sense, we realized a preliminary experience, from a Computer Music's historic example. **Keywords:** GROOVE; Web Audio API; DSP.*

Resumo. *Este painel descreve um programa de síntese sonora, desenvolvido segundo uma interpretação do GROOVE de Max Mathews. Sob a ótica de uma tecnologia chamada Web Audio API, e de um outro software chamado Wavepot, nomeamos nosso programa Termpot. O último programa possibilita criar e encapsular definições algébricas em um único vetor de dados. O processo de criação e edição das funções ocorrem durante o tempo de execução do sistema. Neste sentido realizamos uma experiência preliminar, a partir de um exemplo histórico da Computer Music*

Palavras-chave: GROOVE; Web Audio API; DSP.

1. Introdução

O processamento de sinais digitais de áudio em navegadores de rede, como por exemplo o Mozilla Firefox, Google Chrome ou Apple Safari, é sumarizado por [?, Wyse and Subramanian 2014]. Utilizando a *Web Audio API* [W3C 2012] nós de áudio podem ser concatenados em um grafo de DSP, como demonstrado na Figura 1. [Srikumar 2013] apresenta três instâncias de nós diferentes (*OscillatorNode*, *GainNode*, *DestinationNode*), em sequência, para construir um simples sintetizador.

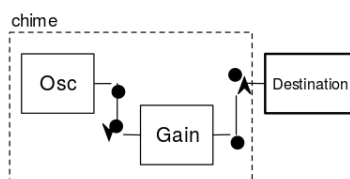


Figura 1: Estrutura de síntese da API webaudio. Fonte: [Srikumar 2013].

1.1. GROOVE

Este artigo envolve a utilização da tecnologia *Web Audio* sob uma perspectiva histórica. Neste sentido, o trabalho de [Mathews and Moore 1970], GROOVE¹, foi fundamental para a estruturação do *Termpot*.

Em um console de computador, um humano digita comandos, renderizados em som. O som, por sua vez, será percebido pela pessoa que controla a máquina, que fornecerá uma nova entrada de dados (através de novos comandos, ou dispositivos, como controles manuais). O sistema pessoa-máquina entra em um estado de *feedback*, e o processo de síntese sonora considera questões performáticas. Um exemplo de música feita com o GROOVE, é *The Expanding Universe* (1980) de Laurie Spiegel [Reynolds 2012]².

2. Objetivos

Descrever um programa *web* desenvolvido com base no *ScriptProcessorNode*, e estruturado segundo uma interpretação GROOVE.

3. Termpot

O Termpot (ver Figura 2) é um ambiente de luteria composicional [Iazzetta 2009, ?], que utiliza o *livecoding* como paradigma de programação. É também uma adaptação do ambiente *Wavepot* (apresentado na nota-de-rodapé 2 da Seção 1).

Comandos utilizam a sintaxe da linguagem *coffeescript* [Burnham 2011]³, que possibilita otimizações diversas, entre elas, o tempo de produção do próprio código. Neste sentido, o ambiente possibilita a improvisação de códigos com a extensão média de uma linha (ver Código 1).

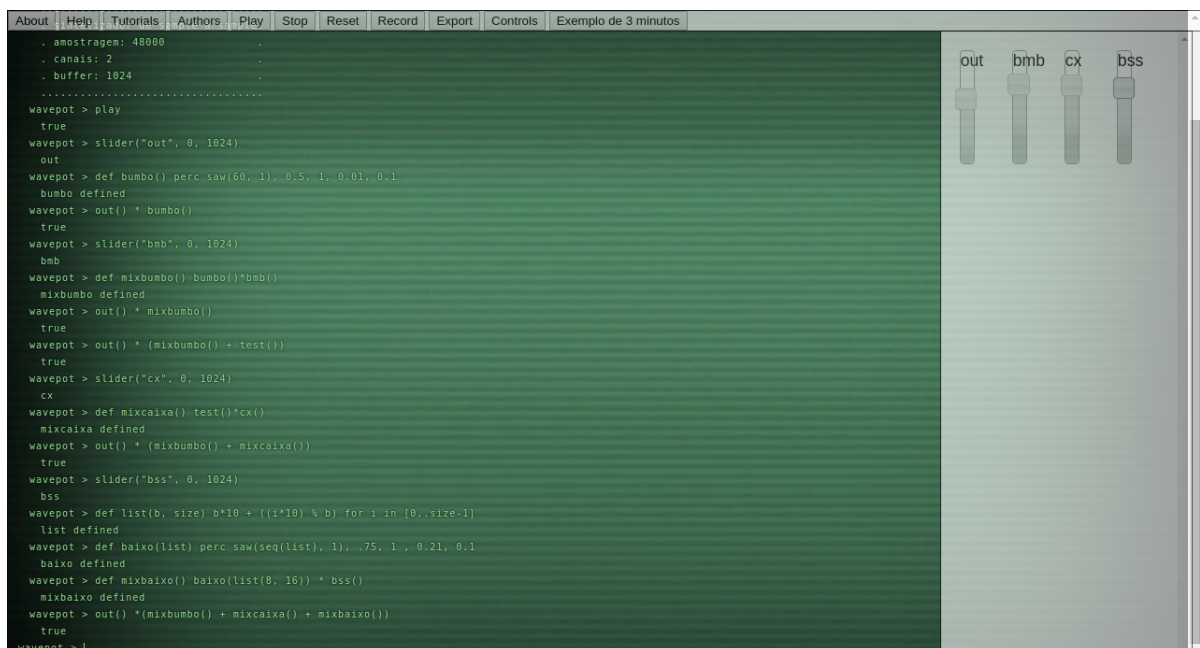


Figura 2: Aplicativo *Termpot*. Fonte: autores.

É interessante notar aqui um princípio de compilação *Just In Time* [Aycock 2003]. Antes que uma função de áudio específica seja executada, o sistema é executado, à

¹Generated Real-time Operations On Voltage-controlled Equipment [Serra 2008, Reynolds 2012].

²Disponível em <https://www.youtube.com/watch?v=dYUZmsfm4Ww>.

³<http://coffeescript.org/>, acessado em 5 de novembro de 2015.

```

1 .....
2 . Virtual machine started at .
3 . Thu Sep 03 2015 13:32:15 GMT-0300 (BRT).
4 . type help for instructions .
5 .....
6 $$$$ | (1)
7 $$$$ wavepot 1024 (2)
8 .....
9 . sintetizador de sample a sample. (3)
10 . amostragem: 44100 .
11 . canais: 2 .
12 . buffer: 1024 .
13 .....
14 wavepot > play (4)
15 wavepot > 0.71 * sin 440, sin(330, sin(220, 0.5) (5)
16 true

```

Código 1: Console do *termpot* aguardando dados de entrada do improvisador (1). O improvisador inicia o ambiente wavepot com um buffer de 1024 pontos flutuantes (2). Informações diversas do sistema de áudio (3). O improvisador inicia o processamento de áudio (4). O improvisador define o processamento de áudio (5).

princípio, com silêncio. Ao se definir uma função de saída, o sistema automaticamente executa a função para os alto-falantes. O processo de síntese é semelhante ao *Wavepot*.

3.1. Criação de funções em tempo de execução

Uma das potencialidades do *software* é a capacidade de definir novas funções em tempo de execução, do ponto-de-vista dos desenvolvedores, de maneira bastante rápida. Estas funções podem ser encapsuladas em outras novas funções. Existem também funções de base, como *noise* (Ruído branco), *sin* (senóide), *saw* (dente-de-serra), *tri* (triangular), *pulse* (pulso), *env* (envelope) e *sequenciamento* (seq)..

```

1 wavepot > def tresAM(f, f1, f2, a) a * sin f, doissin(f1, f2)
2 true
3 wavepot > def doissin(f1, f2) sin f1, sin(f2, 0.5)
4 true
5 wavepot > inspect tresAM
6 var tresAM;
7
8 tresAM = function(f, f1, f2, a) {
9   return a * sin(f, doissin(f1, f2))
10 };
11 wavepot > inspect doissin
12 var doissin;
13
14 doissin = function(f1, f2) {
15   return sin(f1, sin(f2, 0.5))
16 };
17 wavepot > tresAM 440, 330, 220, 0.71
18 true

```

Além da prototipação de funções, habilitamos a criação dinâmica de GUIs de controle (*sliders*, ver Código 2)

```

1 wavepot > slider "left", 0, 1024
2 true
3 wavepot > slider "right", 0, 1024
4 true
5 wavepot > stereo sin(440,0.5)*left(), sin(330, 0.5)*right()
6 true

```

Código 2: Exemplo de código do Wavepot

Uma característica singular do *Wavepot* original, é a possibilidade de separação da programação-partitura em dois arquivos, muito semelhante ao método *Instrumento-Orquestra* descrito por Max Mathews e utilizado no CSound [Mathews 1963, Di Nunzio 2010]. Isso é possível adicionando um marcador `@module` aos comentários iniciais de um código. Desta forma, serão reconhecidos dois arquivos durante a performance de improvisação: *index.js* e *test.js*. O primeiro permite elaborar os instrumentos, enquanto o segundo realiza o DSP (teste).

Já no *Termpot*, buscamos utilizar outro método de codificação focado em uma abordagem mais performática.

Arriscamos a comentar uma inspiração no GROOVE de [Mathews and Moore 1970, Nunzio 2010], quando este propõe a criação de novas funções em tempo de execução. Ao mesmo tempo em que utilizamos a biblioteca *Ptty.js* dando ao *Termpot* as características de um emulador de terminal no que tange a utilização de comandos em tempo de execução, como em um terminal, integrado com controles manuais. Por esta razão, acreditamos que esta ferramenta é inspirada no conceito de compor, memorizar, editar e controlar funções do tempo, algoritmicamente e manualmente [Mathews and Moore 1970].

Referências

- Aycock, J. (2003). A brief history of just-in-time. *ACM Computer surveys*, pages 97–113.
- Burnham, T. (2011). Coffeescript: Accelerated javascript development.
- Di Nunzio, A. (2010). Genesi, sviluppo e diffusione del software "MUSIC n" nella storia della composizione informatica.
- Iazzetta, F. (2009). *Música e mediação tecnológica*. Ed. Perspectiva-Fapesp.
- Mathews, M. V. (1963). The digital computer as a musical instrument. *Science*, 142(3592):553–557.
- Mathews, M. V. and Moore, F. (1970). GROOVE a program to compose, store, and edit functions of time. *Bell Telephones Laboratories*, page 7.
- Nunzio, A. d. (2010). Groove.
- Reynolds, S. (2012). Resident visitor: Laurie Spiegel's machine music.
- Serra, X. (2008). Sound and music computing research: Historical references.
- Srikumar, S. (2013). Taming the script processor node.
- W3C (2012). Web audio API.
- Wyse, L. and Subramanian, S. (2014). The viability of the web browser as a computer music platform. *Computer Music Journal*, 37(4):10–23.