Programação musical para a web com o Mosaicode

MODALIDADE: COMUNICAÇÃO

SUBÁREA: SONOLOGIA

Flávio Luiz Schiavoni
Universidade Federal de São João Del Rei - Departamento de Computação - fls@ufsj.edu.br

Luan Luiz Gonçalves
Universidade Federal de São João Del Rei - Departamento de Computação - luanlg.cco@gmail.com

Resumo: Este artigo apresenta um ambiente de programação visual para a criação de aplicações musicais. Estas aplicações musicais são baseadas na API webaudio e funcionam em navegadores da Internet. A ferramenta em questão utiliza a programação por blocos e conexões e gera código de aplicações baseado em seus diagramas.

Palavras-chave: Geração de código. Ambiente de programação visual. Webaudio.

Musical Programming to the web with NONONO

Abstract: This paper presents a Visual Programming Environment to Musical Application Creation. These Musical Applications are based on Webaudio API and work on web browsers. The developed tool uses block and connection programming and generates code based on diagrams.

Keywords: Code generation. Visual Programming Environment. Webaudio.

1. Introdução

O desenvolvimento de aplicações computacionais costuma ser feito, em sua grande maioria, por meio de linguagens de programação. Contudo, é possível encapsular este desenvolvimento em outros paradigmas de programação que não implica na manipulação direta da linguagem de programação. Tal encapsulamento pode simplificar o desenvolvimento de protótipos e auxiliar na criação de aplicações e no ensino de programação (FRASER, 2012). Entre as formas utilizadas de construção de programas que não utilizam diretamente a linguagem de programação está a programação por meio de componentes gráficos / visuais.

Ferramentas de programação visual costumam ser específicas para determinado domínio de aplicação como o Pure Data para programação musical (PUCKETTE et al., 1996), programação de Circuitos Lógicos Programáveis com CLP (COSTA, 2006) ou para ensino de computação como a Scratch (RESNICK et al., 2009) e o Google blockly (FRASER, 2012).

Entre as ferramentas que utilizam este paradigma de programação encontra-se a ferramenta Harpia [1], voltada para o processamento de imagens e visão computacional. A ferramenta Harpia possuiu um grande número de usuários e já fez parte do repositório oficial de distribuições Linux como Debian e Ubuntu mas infelizmente teve seu desenvolvimento descontinuado ao fim do projeto.

Diferente de outros ambientes de programação musical visuais como o Pure Data, Max/MSP (DIDKOVSKY; HAJDU, 2008) ou Alsa Modular Synthesis[2], a ferramenta harpia trabalha com geração de código-fonte a partir da programação visual. O código-fonte gerado nesta ferramenta é escrito na linguagem C e utiliza o framework OpenCV (BRADSKI; KAEHLER, 2008) para a implementação das funcionalidades do domínio de aplicação de Visão Computacional.

Apesar de estar descontinuada oficialmente, esta ferramenta possui um grande potencial para o ensino e programação de Processamento Digitais de Imagens além de possuir características claramente atrativas para outros segmentos como Ensino de computação, Visão computacional, Computação Musical e Arte Digital.

A refatoração de código da ferramenta Harpia para a reativação deste projeto deu origem ao projeto *Mosaicode*, apresentado neste artigo. Este projeto teve inicio com a refatoração do código da Ferramenta Harpia e continuou com uma reescrita total desta ferramenta. Isto permitiu isolar a geração de código da ferramenta inicial e adicionar à mesma outros domínios de aplicação e outras linguagens de programação.

Neste artigo, apresentaremos a utilização da ferramenta *Mosaicode* para a programação visual de aplicações musicais em Javascript utilizando para isto a API webaudio.

2. O projeto *Mosaicode*

O projeto *Mosaicode* trata-se de um ambiente de programação visual que utiliza o paradigma de encapsulamento de código em blocos onde cada bloco realizada uma tarefa específica. A visão geral desta ferramenta é ilustrada na Figura 1.

Blocos diferentes podem ser combinados por meio de conexões para a criação de aplicações. Para a geração de código, um Gerador é utilizado de forma que a alteração e manipulação do gerador permita criar diferentes trechos de código.

A seguir, apresentaremos os elementos deste ambiente de programação.

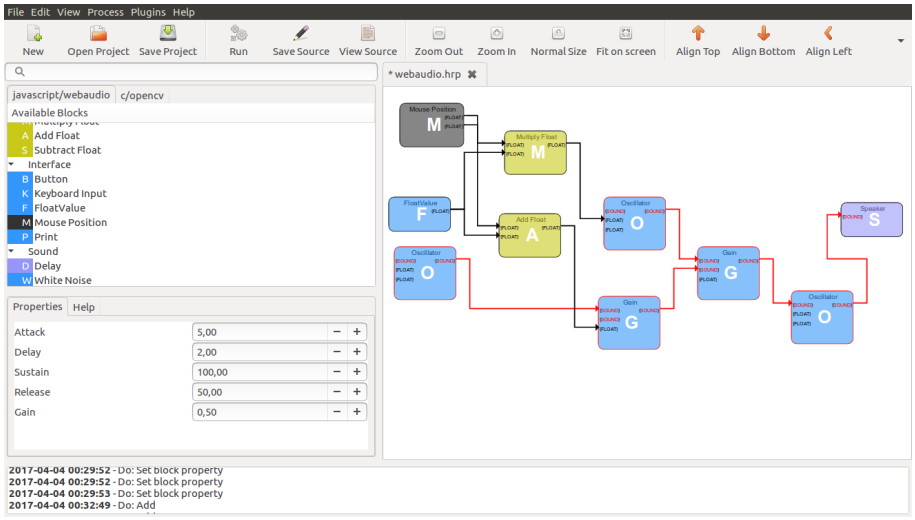

Figura 1: Ambiente de programação Visual da ferramenta *Mosaicode*

2.1. Blocos

Os Blocos da ferramenta *Mosaicode* são as unidades de código iniciais de programação. Cada bloco pode possuir um conjunto de propriedades estáticas, configuradas na própria ferramenta *Mosaicode* ou propriedades dinâmicas, configuradas por outros objetos. A configuração de propriedades estáticas é apresentada na Figura 2.

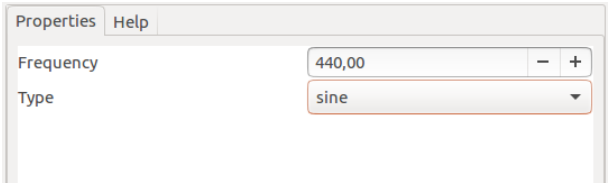

Figura 2: Configuração de propriedades do Oscilador

As propriedades dinâmicas dos blocos são representadas na forma de portas de conexões. As portas de entrada são utilizadas para configurar dinamicamente um Bloco e as portas de saída permitem utilizar a saída de processamento de um Bloco para a configuração de outro Bloco.

Entradas e saídas são combinadas por meio de Conexões.

2.2. Conexões

Conexões são as ligações entre os blocos. Uma conexão possui um tipo e apenas conexões compatíveis podem ser feitas. Isto garante que um bloco só irá receber valores que o mesmo sabe como utilizar. A conexão entre blocos deve ajustar a troca de valores de entrada e saída entre dois trechos de código no momento da geração do código pelo Gerador. A combinação de blocos e conexoes geram diagramas de programação, conforme ilustrado na Figura 3.

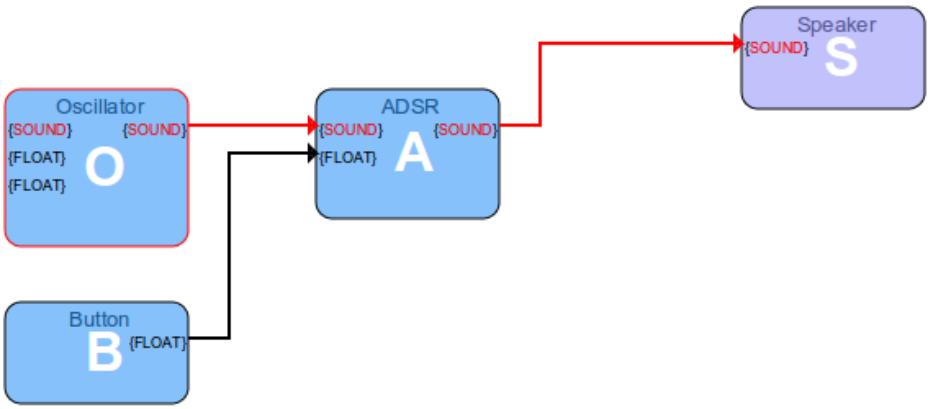

Figura 3: Exemplo de Diagrama e Conexões

2.3. Geradores

Blocos e Conexões costumam pertencer a um domínio de aplicação específico e são desenvolvidos para uma linguagem específica. Da mesma forma, um Gerador de código específico pode ser criado para um determinado domínio e linguagem de programação. Ao Gerador cabe processar individualmente cada bloco e conexão para a criação do código-fonte final do Diagrama.

3. Geração de aplicações musicais Web

A geração de aplicações musicais para a web pela ferramenta *Mosaicode* iniciou-se com a criação de uma biblioteca para este propósito. Isto implicou no desenvolvimento de um conjunto de Blocos, Conexões e um Gerador para esta linguagem.

O conjunto inicial de Blocos contava com objetos que encapsulavam funções matemáticas, Objetos para interface, Objetos sonoros, entre outros. Uma parte do conjunto de blocos para este fim é apresentado na Figura 4.

Os objetos de Som desta biblioteca são baseados nos objetos já existentes na API webaudio. Esta API possui diversos objetos sonoros já definidos e também diversos efeitos de áudio e permitem transformar o navegador Web em um ambiente de programação musical (ROBERTS; WAKEFIELD; WRIGHT, 2013).

As conexões em Javascript / webaudio, diferentemente das conexões em C, não podem utilizar ponteiros pois os mesmos não existem nesta linguagem de programação. Por esta razão, as conexões de saída de um objeto são funções armazenadas em um vetor e que podem ser chamadas pelo objeto origem. Um trecho de código gerado é apresentado na Figura 5.

Isto permitiu que uma mesma saída fosse conectada a diversos objetos e que o

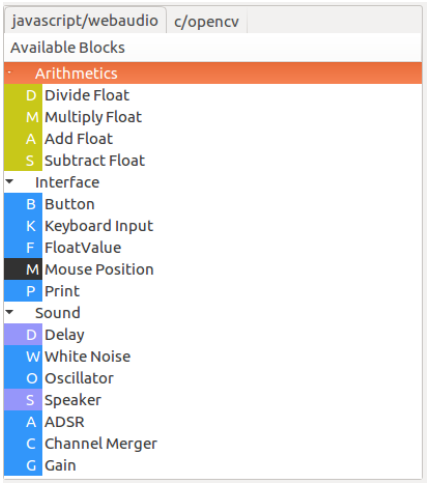

Figura 4: Alguns objetos para a geração de aplicações Web

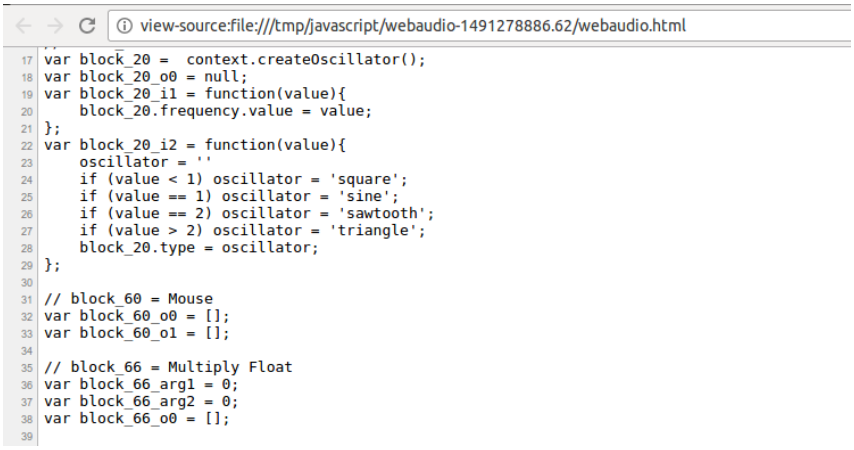

Figura 5: Exemplo de código Gerado

aproveitamento de código da mesma fosse ainda maior.

Por fim, o Gerador de código criado consegue mesclar elementos de GUI, como sliders ou botões, com código Javascript "não visível" de forma a permitir a criação de aplicações complexas integradas a páginas HTML.

4. Conclusão

Este artigo apresentou a refatoração de código da ferramenta Harpia que culminou na criação do projeto *Mosaicode*. Este projeto, além de refatorar a ferrament original, alterou seu código para permitir que esta ferramenta gerasse código para outros domínios de aplicação.

Neste artigo apresentamos a geração de código para a API webaudio em Javascript utilizando a ferramenta *Mosaicode*. Esta geração de código foi possível graças ao desenvolvimento de uma biblioteca de blocos, conexões e um Gerador de código para este

domínio de aplicação e linguagem de programação.

Esta biblioteca permite a prototipação rápida de aplicações musicais para a web permitindo a não programadores o aprendizado de programação e o desenvolvimento de aplicações musicais.

Apesar de termos um conjunto inicial de objetos para este domínio de aplicação, é intenção deste projeto desenvolver mais blocos para a programação web para permitir a criação de aplicações ainda mais robustas e complexas.

Também está na lista de trabalhos futuros o desenvolvimento de exemplos de algoritmos clássicos de síntese, como AM e FM e efeitos clássicos como modulação em anel e outros a ser distribuído juntamente com esta ferramenta.

Referências:

BRADSKI, Gary; KAEHLER, Adrian. *Learning OpenCV: Computer vision with the OpenCV library*. [S.l.]: "O'Reilly Media, Inc.", 2008.

COSTA, César da. Projetando controladores digitais com fpga. *Editora Novatec, primeira ediçao, Maio de*, v. 9, 2006.

DIDKOVSKY, Nick; HAJDU, Georg. Maxscore: Music notation in max/msp. In: *ICMC*. [S.l.: s.n.], 2008.

FRASER, Neil. Google blockly-a visual programming editor. *URL: http://code. google. com/p/blockly, accessed Aug*, 2012.

PUCKETTE, Miller et al. Pure data: another integrated computer music environment. *Proceedings of the Second Intercollege Computer Music Concerts*, p. 37–41, 1996.

RESNICK, Mitchel et al. Scratch: programming for all. *Communications of the ACM*, ACM, v. 52, n. 11, p. 60–67, 2009.

ROBERTS, Charles; WAKEFIELD, Graham; WRIGHT, Matthew. The web browser as synthesizer and interface. In: CITESEER. *NIME*. [S.l.], 2013. p. 313–318.

Notas

[1]Disponível em <http://s2i.das.ufsc.br/harpia/>
[2]Disponível em <http://alsamodular.sourceforge.net/>.