

Middleware Open Dynamic Audio

Uma solução para manipular áudio dinâmico na produção de jogos digitais

Lucas Dário

Orientador: Prof. Dr. Marcelo Gomes de Queiroz

Co-Orientador: Wilson Kazuo Mizutani

Universidade de São Paulo, Instituto de Matemática e Estatística

lucas.dario@usp.br

<https://linux.ime.usp.br/~ldario/mac0499/>



IME-USP

Introdução

No processo de produção de um jogo digital o projetista de som é responsável por produzir a descrição de como todos os áudios do jogo irão se comportar, enquanto que o programador é responsável pela implementação desse áudio dentro do jogo, seguindo a descrição feita pelo projetista.

Alguns problemas existentes nesse processo são, por exemplo: a demora de *feedback* entre os projetistas de som e os programadores, uma vez que o áudio submetido pelo projetista deve ser inserido no jogo pelos programadores para que possa ser testado e a necessidade de uma ferramenta que permita a reprodução de diversos segmentos de áudio de maneira linear e não-linear e aplicação de efeitos e sintetização de áudio em tempo de execução.

Esse projeto visa a implementação de uma solução em forma de *middleware* que seja capaz de lidar com áudio dinâmico[1] através de parâmetros de jogos digitais, oferecendo suporte tanto ao projetista de som quanto ao programador durante o desenvolvimento de um jogo.

O *middleware* Open Dynamic Audio

O Open Dynamic Audio, ou OpenDA, é entendido como um *middleware*, pois disponibiliza diversas funcionalidades que facilitam no processo de produção de um jogo digital, uma vez que oferece recursos capazes de descrever e manipular áudio dinâmico e chamadas de funções para inserção desse áudio em aplicações como jogos digitais.

O sistema é dividido em dois módulos:

- Interface com Pure Data - oferecida aos projetistas de som
- Motor de Áudio - oferecido aos programadores

A interface é composta por submódulos criados para o Pure Data. Utilizando desses submódulos, o projetista é capaz de moldar, dentro de um patch do Pure Data, o comportamento do áudio do jogo de acordo com parâmetros a serem recebidos e, então, enviar esse áudio para o motor de áudio. Para que o motor possa processar esses patches e enviar os parâmetros, foi implementada uma estrutura chamada de Evento, que executará toda a interação com esses patches de Pure Data e que é instanciável e manipulável pelo programador durante o desenvolvimento do jogo através de uma API.

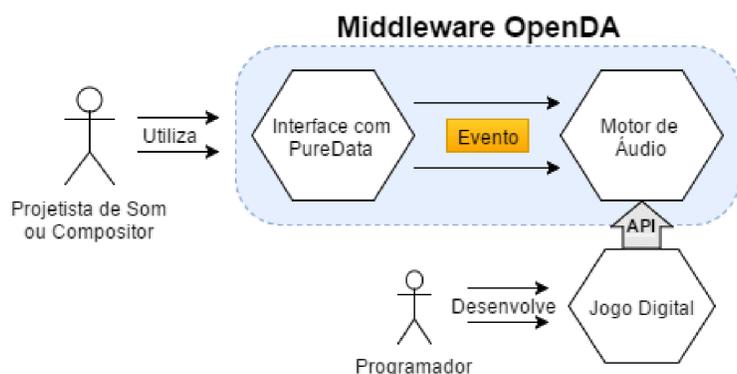


Figura 1: Utilização do OpenDA

Implementação

O código do motor de áudio foi implementado na linguagem C++ e o processamento digital de sinais faz uso da linguagem Pure Data através

da biblioteca libpd[2]. Para a reprodução do áudio, utilizou-se a biblioteca OpenAL Soft[3]. Todas essas tecnologias são ferramentas de código aberto, assim como o *middleware* OpenDA[4]. A interface com o Pure Data é composta por um conjunto de submódulos para essa linguagem, que são reconhecidos pelo motor de áudio.

Resultados

Utilizando-se o Open Dynamic Audio, foi possível a implementação de um protótipo que explorasse as funcionalidades do *middleware*.

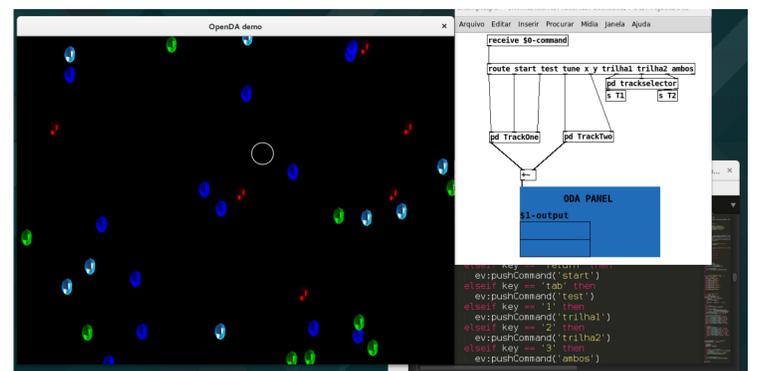


Figura 2: Protótipo de demonstração utilizando a OpenDA

Além disso, a ferramenta mostrou-se eficiente para utilização em projetos de porte maior. Foi possível, por exemplo, a alteração da trilha sonora do jogo Mario[5] para que ela fosse reproduzida utilizando a ferramenta OpenDA.

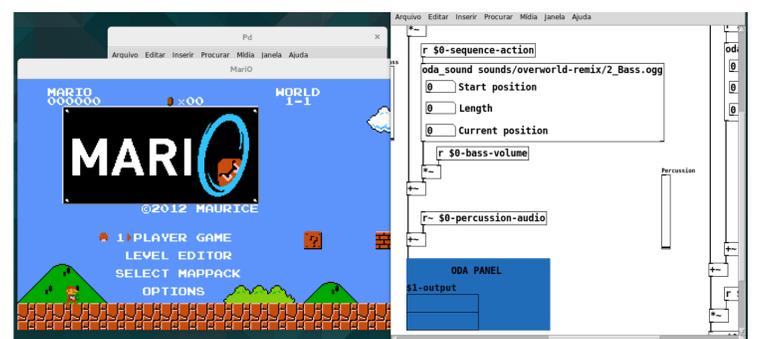


Figura 3: Jogo Mario0 utilizando OpenDA. Em destaque, parte do patch que manipula a música de fundo do jogo

Referências

- [1] K. Collins. An Introduction to the Participatory and NonLinear Aspects of Video Games Audio. 2007. Definição - "audio which reacts to changes in the gameplay environment or in response to a user".
- [2] libpd website. <http://libpd.cc/>. Acesso em: 2015-11-16.
- [3] OpenAL Soft website. <http://kcat.strangesoft.net/openal.html>. Acesso em: 2015-11-16.
- [4] OpenDA Github repository. <https://github.com/open-dynamic-audio/>. Acesso em: 2015-11-16.
- [5] Mario0. <http://stabyourself.net/mario0/>. Acesso em: 2015-11-16.