

# Middleware OpenDA: 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

IME-USP

30 de novembro de 2015

# Sumário

- Introdução
- Desenvolvimento e implementação
- Resultados
- Referências

# Introdução

# Conceitos

- Trilhas Sonoras
- Jogos Digitais
- Áudio Dinâmico[1]
- Middleware

# Conceitos

- Trilhas Sonoras
- Jogos Digitais
- Áudio Dinâmico[1]
- Middleware

# Conceitos

- Trilhas Sonoras
- Jogos Digitais
- Áudio Dinâmico[1]
- Middleware

# Conceitos

- Trilhas Sonoras
- Jogos Digitais
- Áudio Dinâmico[1]
- Middleware

# Motivação

- Demora de feedback entre projetista de som e programador
- Necessária uma ferramenta capaz de:
  - Reproduzir segmentos de áudio de maneira linear ou não linear
  - Aplicar efeitos e sintetizar áudio em tempo real

# Motivação

- Demora de feedback entre projetista de som e programador
- Necessária uma ferramenta capaz de:
  - Reproduzir segmentos de áudio de maneira linear ou não linear
  - Aplicar efeitos e sintetizar áudio em tempo real

# Motivação

- Demora de feedback entre projetista de som e programador
- Necessária uma ferramenta capaz de:
  - Reproduzir segmentos de áudio de maneira linear ou não linear
  - Aplicar efeitos e sintetizar áudio em tempo real

# Motivação

- Demora de feedback entre projetista de som e programador
- Necessária uma ferramenta capaz de:
  - Reproduzir segmentos de áudio de maneira linear ou não linear
  - Aplicar efeitos e sintetizar áudio em tempo real

# Open Dynamic Audio como solução

- Capaz de criar, manipular e reproduzir áudio dinâmico
- Dinamicidade guiada por parâmetros
- Oferecer suporte tanto ao projetista de som quanto ao programador

# Open Dynamic Audio como solução

- Capaz de criar, manipular e reproduzir áudio dinâmico
- Dinamicidade guiada por parâmetros
- Oferecer suporte tanto ao projetista de som quanto ao programador

# Open Dynamic Audio como solução

- Capaz de criar, manipular e reproduzir áudio dinâmico
- Dinamicidade guiada por parâmetros
- Oferecer suporte tanto ao projetista de som quanto ao programador

# Desenvolvimento e implementação

# Divisão do Open Dynamic Audio

- Separada em dois módulos
  - Interface com Pure Data[2] - Oferecida aos compositores e projetistas de som
  - Motor de Áudio - Oferecido aos programadores
- Evento - comunicação entre os módulos

# Divisão do Open Dynamic Audio

- Separada em dois módulos
  - Interface com Pure Data[2] - Oferecida aos compositores e projetistas de som
  - Motor de Áudio - Oferecido aos programadores
- Evento - comunicação entre os módulos

# Divisão do Open Dynamic Audio

- Separada em dois módulos
  - Interface com Pure Data[2] - Oferecida aos compositores e projetistas de som
  - Motor de Áudio - Oferecido aos programadores
- Evento - comunicação entre os módulos

# Divisão do Open Dynamic Audio

- Separada em dois módulos
  - Interface com Pure Data[2] - Oferecida aos compositores e projetistas de som
  - Motor de Áudio - Oferecido aos programadores
- Evento - comunicação entre os módulos

# Divisão do Open Dynamic Audio

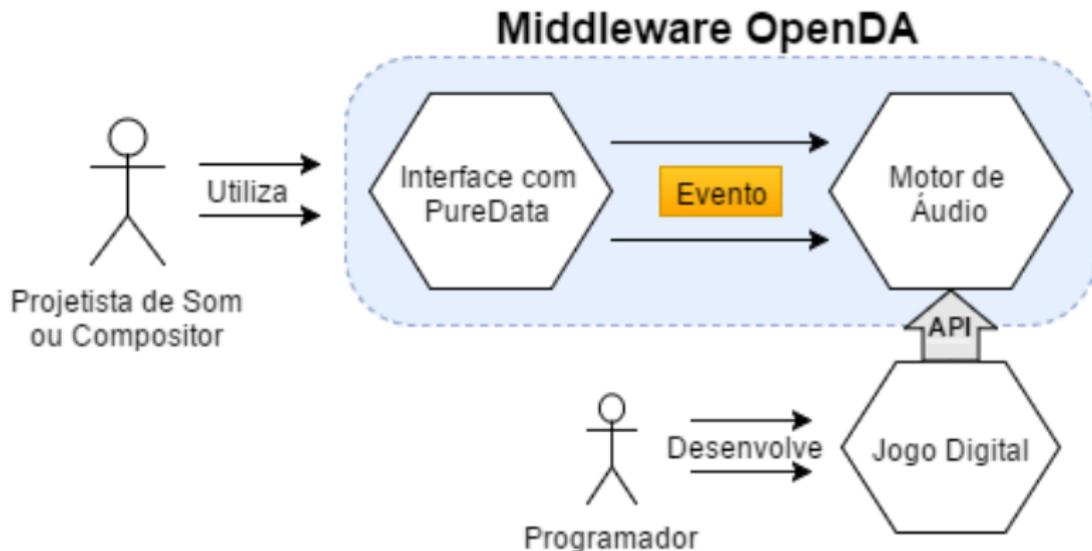


Figura: Utilização do OpenDA

# Interface com Pure Data

- Implementação de submódulos para o Pure Data
- Configuração do áudio é feita baseada nos parâmetros
- Projetista pode testar o resultado enquanto produz ele

# Interface com Pure Data

- Implementação de submódulos para o Pure Data
- Configuração do áudio é feita baseada nos parâmetros
- Projetista pode testar o resultado enquanto produz ele

# Interface com Pure Data

- Implementação de submódulos para o Pure Data
- Configuração do áudio é feita baseada nos parâmetros
- Projetista pode testar o resultado enquanto produz ele

# Motor de Áudio

- Processamento da especificação feita pelo projetista
- Execução do áudio através do jogo
- Leitura de patches e envio de parâmetros através da classe Event
- Oferecer API para os programadores

# Motor de Áudio

- Processamento da especificação feita pelo projetista
- Execução do áudio através do jogo
- Leitura de patches e envio de parâmetros através da classe Event
- Oferecer API para os programadores

# Motor de Áudio

- Processamento da especificação feita pelo projetista
- Execução do áudio através do jogo
- Leitura de patches e envio de parâmetros através da classe Event
- Oferecer API para os programadores

# Motor de Áudio

- Processamento da especificação feita pelo projetista
- Execução do áudio através do jogo
- Leitura de patches e envio de parâmetros através da classe Event
- Oferecer API para os programadores

# Ferramentas utilizadas

- Principal linguagem: C++
- Processamento digital de sinal: PureData e libpd[3]
- Interface com hardware: OpenAL e OpenAL Soft[4]

# Ferramentas utilizadas

- Principal linguagem: C++
- Processamento digital de sinal: PureData e libpd[3]
- Interface com hardware: OpenAL e OpenAL Soft[4]

# Ferramentas utilizadas

- Principal linguagem: C++
- Processamento digital de sinal: PureData e libpd[3]
- Interface com hardware: OpenAL e OpenAL Soft[4]

# Resultados

# OpenDA em ação

- Usada na produção de um protótipo
- Usada para alterar o áudio de um projeto maior

# OpenDA em ação

- Usada na produção de um protótipo
- Usada para alterar o áudio de um projeto maior

# Protótipo demo para OpenDA

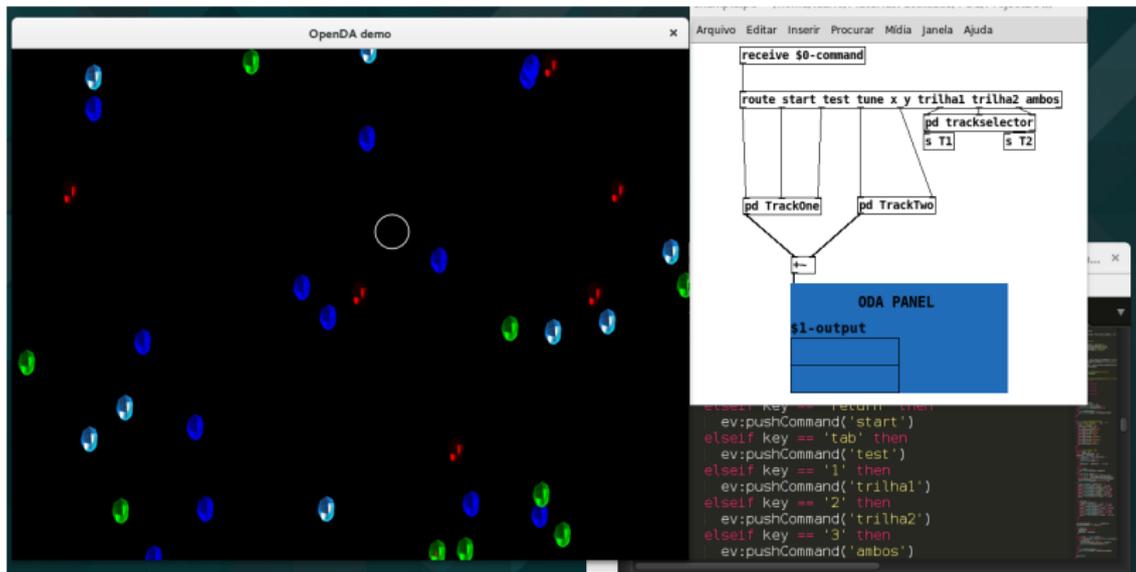


Figura: Protótipo para demonstração do OpenDA

# Alteração do áudio do jogo Mari0[5]

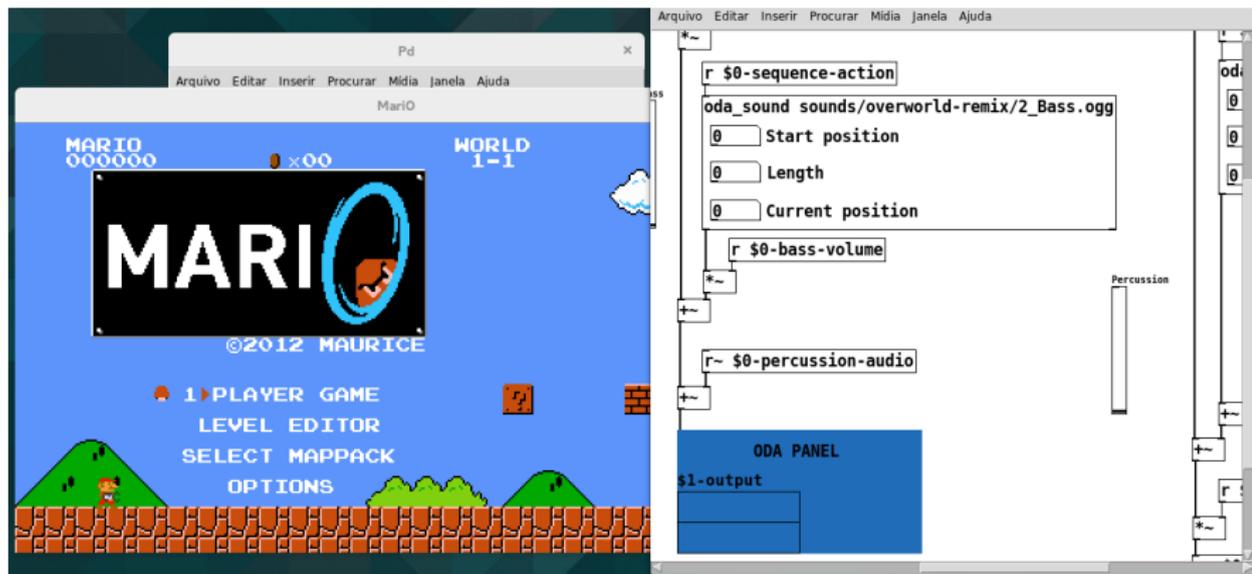


Figura: Jogo Mari0 utilizando OpenDA

# Referências

# Referências

- 1 K. Collins. An Introduction to the Participatory and NonLinear Aspects of Video Games Audio. 2007.
- 2 Pure Data website. <https://puredata.info/>. Acesso em: 2015-11-16.
- 3 libpd website. <http://libpd.cc/>. Acesso em: 2015-11-16.
- 4 OpenAL Soft website.  
<http://kcat.strangesoft.net/openal.html>. Acesso em: 2015-11-16.
- 5 Mari0. <http://stabyourself.net/mari0/> . Acesso em: 2015-11-16.

# Middleware OpenDA: 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

IME-USP

30 de novembro de 2015