UNIVERSIDADE DE UBERABA

GAME DESIGN DOCUMENT PARA TECH DEMO EM TECNOLOGIAS DE ÁUDIO

Equipe:

- Lailaque Luna Lamounier Guimarães Borges

Orientador:

- Humberto Patrick Lacerda

UBERABA

2020-2022

Sumário

Lista de Figuras

INTRODUÇÃO

Este documento apresentará a importância do áudio para jogos digitais utilizando dados de estudos e pesquisas, focando principalmente no uso e aplicação do áudio binaural e ambiente utilizando uma *tech demo* produzida na linguagem de programação C sem a utilização de uma *game-engine* utilizando principalmente as funções da API FMOD Core e Studio.

O áudio binaural é uma técnica de gravação e reprodução de áudio que teve suas origens em 1881 com o desenvolvimento do *Théâtrophone* o qual era um sistema de microfones de carbono arranjados de uma maneira a imitar o formato de uma cabeça humana para fins de aumentar a imersão do telespectador ao ouvir a gravação de um concerto musical. Desde de então a tecnologia tem melhorado e expandido seu uso, sendo possível hoje a transformar uma gravação estéreo e uma binaural pelo uso de softwares.

Na área de jogos o áudio binaural tem sido aplicado esporadicamente tendo ganhado tração recentemente diante a jogos de alto orçamento produzido por grandes empresas, porém mesmo para os jogos de estúdio pequenos, a aplicação não demanda extremo esforço e é proposto por este projeto a demonstrar a superioridade de um áudio trabalhado binaural a métodos convencionais.

Há também outras técnicas que no cinema e na música são prevalentes mas em jogos são pouco utilizadas em principal a reverberação e propagação das ondas sonoras que se compõe em um sistema onde o material do ambiente e a composição de objetos afeta o volume e a reverberação do som. Na série de jogos *Half Life* este sistema é realizado onde se é possível ouvir a diferença entre um corredor estreito e um grande galpão por exemplo.

Com o desenvolvimento desta *tech demo* será possível ouvir e sentir todas estas técnicas(áudio binaural, reverberação e propagação) através do simples movimento por um mundo aleatoriamente gerado através de uma *seed* configurável sendo o áudio se alterando dinamicamente através do ambiente ao redor do jogador, hora do dia, e local.

TECNOLOGIAS E APIs

CONCEITOS BÁSICOS

A área de áudio engloba várias artes, como trilha, efeitos, interface, …; e sua importância é enorme para a qualidade de um jogo. Pelo fato da mesma área ser tão extensa este projeto focará apenas na área de efeitos sonoros e trilha ambiente, onde as propostas técnicas serão aplicadas.

Para propriamente simular perfeitamente o áudio binaural primeiro é preciso entender o funcionamento da orelha humana:

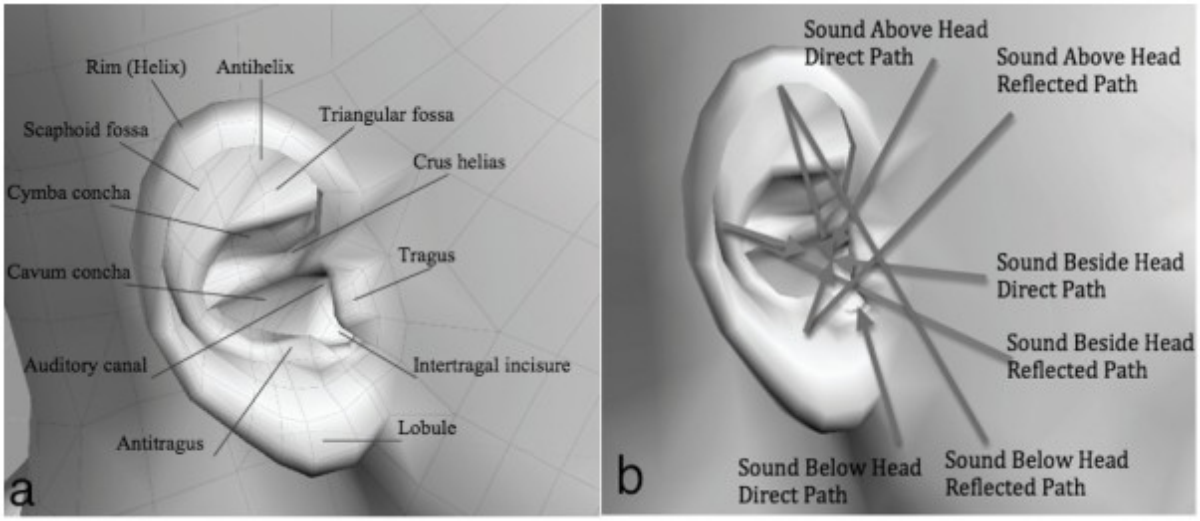

Figura 1: Estrutura da orelha humana

Seguindo o diagrama podemos perceber como a reflexão das ondas sonoras e determinadas partes da orelha, isto é possível a ser simulado pela API FMOD.

A reverberação costuma funcionar de maneira mais genérica com o produtor de áudio assumindo os materiais de objetos fictícios dentro do ambiente virtual e como os mesmos se relacionariam com a absorção sonora.

Ao fim temos a propagação, a mais complexa das implementações e a mais demandadora no hardware, é preferível o usuário possuir uma placa de som dedicada para a experiência completa porém é possível implementar propagação básica sem o uso de tecnologias de *tracing* similares ao *raytracing*. A mais simples forma é a simulação da velocidade do som. Todos estes métodos são automaticamente selecionados e executados pela FMOD, seguindo as capacidades de hardware do mesmo.

LINGUAGEM C

A linguagem C é uma linguagem de programação de baixo nível desenvolvida na década de 1970 com o fim de simplificar a complexa linguagem *assembly* usada previamente. Desde então a mesma foi expandida com vários protocolos e continua recebendo os mesmos. Este projeto será feito na versão mais recente da linguagem: **C17**.

O motivo pelo qual foi optada a utilização da mesma ao invés da linguagem C++(Mais comumente usada no desenvolvimento de jogos) foi o fato da mesma ser mais portátil e rápida, demonstrando assim praticamente apenas o processamento da API FMOD, a qual também é escrita na linguagem C.

OPENGL

É um renderizador 3D de código aberto com documentação ampla, o mesmo será o motor gráfico deste projeto.

GLFW E GLAD2

São bibliotecas de código aberto que facilitam a implementação multiplataforma de várias funções para janelas e input.

CGLM(COM MÓDULO IVEC)

É uma biblioteca para matemática gráfica computacional de código aberto essencial para o código de renderização.

STB IMAGE

É uma API de código aberto para processar arquivos de imagem(como texturas).

GTK-GLIB

Parte da biblioteca GNOME padrão do Linux, utilizada neste projeto por seus *arrays* variáveis.

FREETYPE

Biblioteca de código aberto para o processamento de fontes(para renderização de texto).

TOMLC

Biblioteca oficial da implementação de arquivos .*toml*(similares a *.json*), os quais são arquivos de armazenamento de dados usados neste projeto para configurações e modularidade.

DIRENT

Implementação para o Windows da biblioteca padrão do Linux para acessar pastas.

FMOD (CORE, STUDIO E FSBANK)

A API mais importante deste projeto, traz uma poderosíssima *engine* de som que é simples de aplicar e totalmente ligada ao código de baixo nível. A mesma é utilizada em vários jogos modernos.

OPENMP

Um padrão internacional para *multithreading* paralelizado, utilizado neste projeto para as funções mais computacionalmente complexas.

PTHREADS-W32

Uma implementação de *threads* individuas portada do Linux para o Windows para fins de portabilidade.

LUAJIT2

Um interpretador da linguagem de *scripting* Lua escrito em *assembly* direcionado a performance. Um *script* em Lua e utilizado neste projeto para a geração da fase, em lua para que o mesmo possa ser modificado sem a necessidade da compilação do código fonte.

LUACOMPAT

Uma biblioteca que traz funções das versões mais novas da Lua para a Lua 5.1 usada pelo interpretador LuaJIT.

PERLINNOISE

Uma biblioteca para funções de geração de ruído Perlin, utilizados na geração do terreno procedural.

LIBNOVA

Uma biblioteca de funções matemáticas astronômicas, usadas para o posicionamento de objetos no céu da fase.

GAMEPLAY

RESUMO

A *tech-demo* terá uma *gameplay* simples onde o jogador terá a liberdade de navegar no mundo e construir coisas através do uso de *voxels*, enquanto um ciclo de dia noite será calculado. O objetivo e imersar o jogador neste simples mundo que possui visuais simples mas áudio avançado.

GÊNERO, SEMELHANÇAS E DIFERENÇAS

O projeto pode ser classificado como um jogo *Sandbox* onde não existe um objetivo definido o limite é apenas a criatividade do jogador.

Figura 2: Um mundo em Minecraft

Figura 3: Um mundo em Hytale

Similarmente a jogos como *Minecraft* e *Hytale*, a diferença está no escopo menor e no foco em ambientes e som.

ATRATIVOS

Além do áudio avançado, este projeto também tem o atrativo visual do céu onde elementos são configurados um por um além de uma quantidade configurável de objetos aleatórios.

INTERFACE

A *tech-demo* será jogada por mouse e teclado, segue abaixo o plano do HUD.

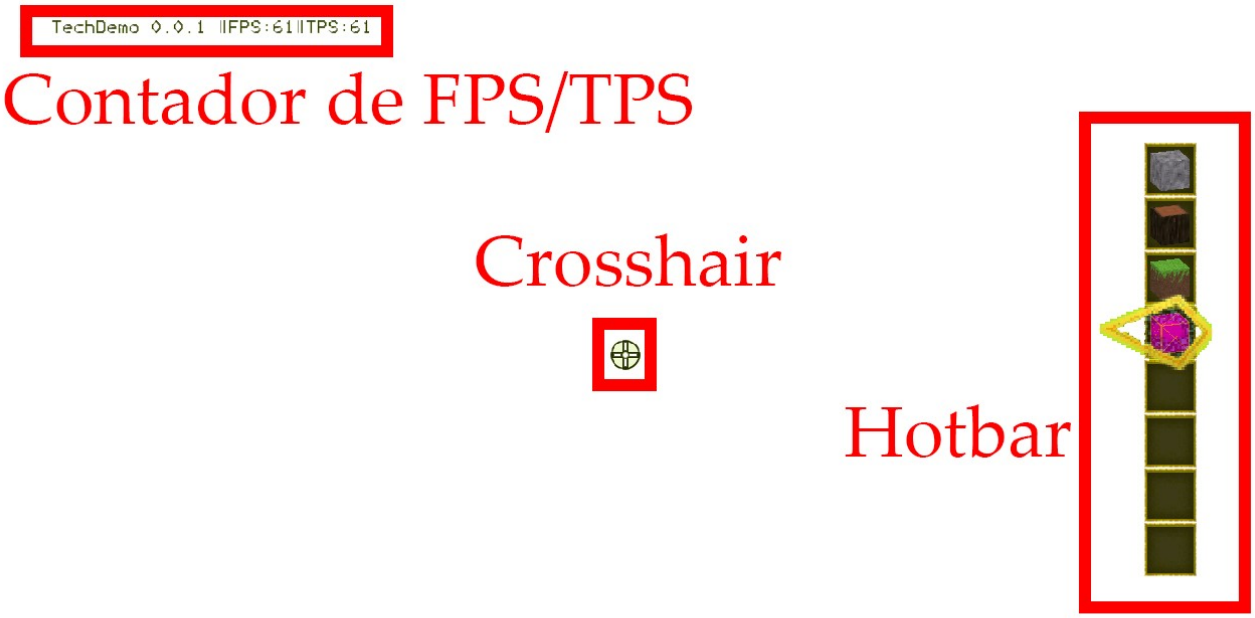

Figura 4: Plano do HUD

Contador de FPS/TPS: Conta o atual FPS do jogo e o atual TPS, que significa *Ticks Per Second* o que é a quantidade de vezes que a engine atualiza, em tempo fixo onde: $1 Tick = \frac{1}{60} Segundo$.

Crosshair: Indica onde o jogador está mirando, utilizado para determinar qual objeto a ser destruído ou onde se deve ser colocado.

Hotbar: Mostra uma lista rolável das opções de *voxels* que o jogador pode colocar.

CONTROLES

Lista de ações e seus inputs por mouse e teclado.

- Movimento da Câmera: movimentado o mouse para cima/baixo/esquerda/direita movimenta a câmera pelo mesmo eixo.

- Trocar *Voxel*: rolando a roda do mouse trocará o *voxel* selecionado indicado na Hotbar pela seta no centro da mesma.
- Trocar tipo do *Voxel*: rolando a roda mouse ao segurar a tecla alt trocará o tipo de *voxel* entre blocos e fluídos.
- Movimentação do jogador: é feita através das teclas WASD.
- Aumento da velocidade do jogador: segurando a tecla shift a velocidade do jogador é duplicada.
- Diminuição da velocidade do jogador: segurando a tecla control a velocidade do jogador é reduzida pela metade.
- Pular: é realizado pela tecla space.
- Colocar *voxel*: apertando o botão direito do mouse colocará um *voxel* ondo o jogador estiver observando.
- Remover *voxel*: apertando o botão esquerdo do mouse retira o bloco sendo observado.
- Sair: apertando e segurando a tecla esc fexa a janela da *tech-demo*.

DETALHES TÉCNICOS

REQUISITOS DE HARDWARE

- Processador com suporte a OpenMP 1.2 ou superior.
- Windows 7 ou superior ou Linux ambos 64 bits apenas.
- Placa de vídeo com suporte a OpenGL 4.3 ou superior.
- 2 Gb de memória RAM.
- Mouse
- Teclado
- Um monitor

SOFTWARES UTILIZADOS

- Visual Studio Code: Semi-IDE para toda a programação
- LLVM-Clang: Compilador
- FMOD Studio: Usado para fazer o design de áudio.
- Adobe Photoshop: Usado para fazer as texturas.
- Doxygen: Usado para documentação.
- CMake: Gerador de arquivos de build e versões multiplataforma, além de ser um gerenciador de dependências.

- Orbit Profiler: Depuração da performance.
- Git: Gerenciamento do repositório do código.
- FL Studio: Usado para a criação das músicas.
- Audacity: Usado para edição de sons.
- Musescore: Usado para a criação das partituras das músicas

REFERENCIAS BIBLIOGRÁFICAS

GORMANLEY, STEPHEN. **Audio immersion in games — a case study using an online game with background music and sound effects**: selected references. Comput Game J 2, 103–124 (2013).

GALLACHER, N. **Game audio — an investigation into the effect of audio on player immersion**. Comput Game J 2, 52–79 (2013).

RODGERS, KATJA. **Vanishing Importance: Studying Immersive Effects of Game Audio Perception on Player Experiences in Virtual Reality**, Waterloo: Univerity of Waterloo, Reino Unido, 2018.

GRIMSHAW, MARK. **Game Sound Technology and Player Interaction: Concepts and Developments**, Bolton: Univerity of Bolton, Reino Unido, 2011.

SCHMIDT, BRIAN. **GameSoundCon Game Audio Industry Survey 2017**, GameSoundCon(2018), https://www.gamesoundcon.com/post/2017/10/02/gamesoundcon-game-audio-industry-survey-2017, Acesso 01/12/2020.

UNITED STATES BUREAU OF STATISTICS, **Multimedia Artists and Animators**, https://www.bls.gov/ooh/arts-and-design/multimedia-artists-and-animators.htm#tab-1, Acesso 01/12/2020.