

Instituto Politécnico Viana do Castelo
Escola Superior de Tecnologia e Gestão
Engenharia de Computação Gráfica e Multimédia
2024/2025

António Luís Cepa Coelho Rebelo - N°28837

Relatório do Projeto da U.C. de Sistemas Multimédia

IPVC AUDIO-SURF

Acabado a 13/10/2024

Outubro 2024

Índice

1. Introdução	3
1.1. Como Jogar e Regras.....	4
2. Definição da Estrutura	5
2.1. Diagrama do Ficheiro	5
2.2. Métodos	6
3. Desenvolvimento e Implementação.....	8
4. Conclusão	10
5. Referências Bibliográficas	11

1. Introdução

O presente relatório descreve o desenvolvimento e implementação do jogo "IPVC Audio-Surf", criado no contexto do curso de Sistemas Multimédia. Inspirado no famoso jogo "**AudioSurf**", este projeto combina elementos musicais com mecânicas de obstáculos, proporcionando uma experiência interativa única. O objetivo principal é permitir que o jogador navegue entre faixas enquanto evita obstáculos ativados pela própria voz, em sincronia com a música de fundo. A seguir, será discutida a estrutura do jogo, as suas regras, componentes e a lógica utilizada para implementar as funcionalidades essenciais.

O desenvolvimento foi realizado usando **p5.js**, uma biblioteca JavaScript voltada para arte e visualizações interativas. A escolha desta ferramenta deu-se pela simplicidade na manipulação de sons, gráficos e interações com o usuário, que são elementos-chave no desenvolvimento do jogo. Além disso, p5.js facilita a integração com bibliotecas como **p5.sound**, que foi crucial para a análise da música e o reconhecimento do volume captado pelo microfone, elementos essenciais para a jogabilidade.

A estrutura do jogo foi dividida em várias telas, incluindo a tela inicial, de seleção de níveis, tela de jogo, e as telas de fim de jogo (vitória ou derrota). Cada uma das telas apresenta a sua própria lógica e interação com o jogador. O jogo também inclui um sistema de pontuação e uma tabela de "top scores", onde o jogador pode comparar e ver as suas pontuações.

1.1. Como Jogar e Regras

O jogo IPVC Audio-Surf é um **jogo de reflexos e de ritmo**, no qual o jogador precisa mudar de faixa para evitar obstáculos, que são criados dinamicamente com base no volume captado pelo microfone. Além disso, a música de fundo define o ritmo e a intensidade do jogo, o que significa que os obstáculos aparecem com mais frequência quanto mais o jogador canta ou emite sons.

Instruções do Jogo

- O jogador inicia o jogo na faixa central (de três disponíveis).
- As teclas seta esquerda e seta direita são usadas para mover o jogador entre as três faixas disponíveis.
- Ao longo do jogo, obstáculos aparecerem em cada uma das três faixas. O objetivo do jogador é evitar os obstáculos mudando de faixa no momento certo.
- O jogador deve cantar ou emitir sons, que serão captados pelo microfone. O jogo usa o volume do microfone para determinar quando os obstáculos aparecem.
- O jogador acumula pontos com o passar do tempo e ao evitar obstáculos.
- O jogo termina quando o jogador colide com um obstáculo (derrota) ou quando a música acaba (vitória).

Letras e os seus significados no Jogo

Não existem letras específicas no jogo. A mecânica é baseada no volume captado pelo microfone, o que torna a experiência dinâmica e interativa, permitindo que os jogadores participem ativamente através da voz.

2. Definição da Estrutura

A estrutura do jogo foi desenhada com uma combinação de variáveis globais e funções para controlar o fluxo da jogabilidade. A seguir, serão detalhados os principais componentes da estrutura de dados e os métodos usados para implementar as funcionalidades.

2.1. Diagrama do Ficheiro

Embora o jogo não utilize classes formalmente (uma vez que p5.js é uma biblioteca orientada a funções), é possível representar a estrutura em um formato de **pseudo-diagrama de classes** para facilitar a compreensão da relação entre os elementos do jogo.

- **Jogo**
 - **Atributos:**
 - **lanes[]:** Vetor que armazena as posições X das faixas.
 - **playerPos:** Posição do jogador.
 - **obstacles[]:** Vetor que armazena as posições dos obstáculos.
 - **song, backgroundSound, winSound, loseSound:** Sons utilizados no jogo.
 - **fft:** Análise da transformada rápida de Fourier (para análise de som).
 - **mic:** Entrada de áudio do microfone.
 - **score:** Pontuação do jogador.
 - **gameTime:** Tempo total de jogo.
 - **Métodos:**
 - **setup():** Inicializa o ambiente de jogo.
 - **draw():** Controla o ciclo de vida do jogo, desde a tela inicial até a tela de vitória/derrota.
 - **resetGame():** Reinicializa as variáveis do jogo.

- **Obstáculos**
 - **Atributos:**
 - **x:** Posição X do obstáculo (faixa).
 - **y:** Posição Y do obstáculo (altura).
 - **Métodos:**
 - **checkCollision():** Verifica a colisão entre o jogador e os obstáculos.

2.2. Métodos

Cada método desempenha uma função crítica na execução do jogo. Aqui estão os principais métodos usados no jogo, com as suas respetivas descrições:

- **setup():** É a função responsável por configurar o jogo no início. Aqui são inicializadas as variáveis, como a posição inicial do jogador, a configuração das faixas, o carregamento de sons e imagens, e a ativação do microfone. Além disso, a música de fundo começa a tocar em loop nesta função.
- **draw():** A função principal do jogo, executada continuamente. Ela chama as outras funções de acordo com o estado atual do jogo (tela inicial, seleção de nível, jogabilidade, etc.). Além disso, é responsável por redesenhar o cenário, incluindo os obstáculos e o jogador, a cada frame.
- **startGameScreen():** Exibe a tela inicial do jogo, com as instruções de controlo e introdução ao conceito do jogo. A navegação para a tela de seleção de nível ocorre quando o jogador pressiona a tecla ENTER.
- **showLevelSelectScreen():** Mostra a tela de seleção de nível, onde o jogador pode escolher o nível que deseja jogar. A seleção é feita com as teclas direcionais e confirmada com a tecla ENTER.

- **playGame():** A função central da jogabilidade. Nela, o jogador é desenhado, os obstáculos são gerados e movimentados com base no volume captado pelo microfone, e a pontuação é calculada. Também inclui a verificação de colisões entre o jogador e os obstáculos.
- **handlePlayerMovement():** Gere o movimento do jogador pelas faixas usando as teclas de seta. A função também usa interpolação linear (lerp) para suavizar a transição do jogador entre as faixas.
- **checkCollision():** Verifica se o jogador colidiu com algum obstáculo. A colisão é determinada com base nas posições do jogador e dos obstáculos, usando um cálculo simples de interseção entre as áreas ocupadas por cada objeto.
- **pausePopup():** Exibe um popup de pausa quando o jogador pressiona a tecla ESC. Oferece opções para retomar o jogo ou retornar ao menu principal.
- **resetGame():** Reinicializa todas as variáveis de estado do jogo, como pontuação, tempo de jogo, obstáculos, etc., para iniciar uma nova sessão.
- **updateTopScores():** Atualiza a lista de pontuações mais altas quando o jogador finaliza o jogo. As pontuações são ordenadas em ordem decrescente e armazenadas na lista topScores.

3. Desenvolvimento e Implementação

A implementação do "IPVC Audio-Surf" foi dividida em várias etapas, cada uma responsável por um aspecto essencial do jogo. Desde a criação das interfaces gráficas até a lógica de detecção de colisão e controle do jogador, cada elemento foi cuidadosamente projetado para proporcionar uma experiência de jogo fluida e dinâmica. O jogo foi desenvolvido em p5.js, utilizando suas bibliotecas de som e gráficos, permitindo uma fácil manipulação de áudio e animações.

3.1 Configuração Inicial e Carregamento de Recursos

No início do código, foram configuradas as variáveis principais, como as faixas de movimento (lanes), a posição inicial do jogador (playerPos), e o vetor de obstáculos. Também foi implementado o carregamento dos arquivos de som e imagens do jogo através da função preload(), utilizando a função loadSound() para carregar as músicas e efeitos sonoros, e loadImage() para carregar os gráficos, como a imagem de fundo, jogador, obstáculos e níveis.

Além disso, foi configurado o sistema de entrada de áudio do jogador através de mic = new p5.AudioIn(), permitindo a interação com o microfone para capturar o som da voz e acionar obstáculos no jogo.

3.2 Estrutura das Telas do Jogo

O jogo possui várias telas, cada uma com uma funcionalidade específica. Estas telas são controladas por variáveis booleanas como startScreen, levelSelectScreen, endScreen, winScreen, e a lógica de troca de telas é gerida pela função draw(). Dependendo do estado do jogo, a função chama uma função específica para renderizar a tela correspondente.

3.2.1 Tela Inicial

A tela inicial, criada pela função startGameScreen(), é onde o jogador recebe as primeiras instruções. Nela, o título do jogo é exibido com informações sobre como jogar. O jogador deve pressionar a tecla ENTER para iniciar o jogo, o que leva à tela de seleção de nível.

3.2.2 Tela de Seleção de Nível

A tela de seleção de nível, criada pela função `showLevelSelectScreen()`, apresenta os diferentes níveis disponíveis para o jogador. Cada nível é representado por uma imagem específica dos diferentes logos das Escolas do IPVC, e o jogador pode navegar entre os níveis usando as setas direcionais. A confirmação da escolha é feita ao pressionar a tecla ENTER, o que inicia o jogo no nível selecionado e carrega a próxima música.

3.2.3 Tela de Jogo

Na tela de jogo principal, a função `playGame()` é responsável por gerir a maioria das mecânicas, como a movimentação do jogador e a criação de obstáculos. O jogador pode mover-se entre três faixas utilizando as teclas direcionais esquerda e direita, e deve evitar os obstáculos que aparecem. Tais obstáculos são acionados quando o nível de som do microfone atinge um certo limite, criando uma mecânica dinâmica que exige que o jogador interaja ativamente com a música para aumentar o nível de dificuldade.

3.2.4 Tela de Pausa

A função `pausePopup()` é responsável por exibir o menu de pausa quando o jogador pressiona a tecla ESC. Este menu oferece a opção de continuar o jogo ou sair e voltar à tela inicial. Durante o estado de pausa, a música também é pausada, e o jogo permanece suspenso até que o jogador decida retomar ou encerrar.

3.2.5 Tela de Fim de Jogo e Tela de Vitória

Quando o jogador colide com um obstáculo, a tela de fim de jogo (`endScreen`) é exibida, mostrando a pontuação final e os três melhores resultados até o momento. Se o jogador completar o jogo com sucesso, a tela de vitória (`winGameScreen`) é apresentada, junto com a pontuação final e os melhores resultados. Ambas as telas têm um sistema de top scores que atualiza automaticamente as pontuações mais altas.

3.3 Mecânica de Jogo

A movimentação do jogador entre as faixas é controlada pela função `handlePlayerMovement()`, que responde às teclas de setas esquerda e direita para alterar a posição do jogador. Para garantir

uma transição suave entre as faixas, a função `lerp()` é utilizada para interpolar a posição do jogador, criando uma animação suave.

A criação de obstáculos é feita com base no nível de som captado pelo microfone. Quando o som detetado é maior que um valor mínimo predefinido (`minMicLevel`), novos obstáculos são criados em uma das três faixas, com intervalos regulares determinados pela variável `obstacleInterval`.

A colisão entre o jogador e os obstáculos é gerida pela função `checkCollision()`, que verifica se as áreas ocupadas pelo jogador e pelo obstáculo se sobrepõem. Caso isso ocorra, o jogo é finalizado e a tela de fim de jogo é exibida.

3.4 Sistema de Pontuação e Temporização

O sistema de pontuação é baseado no tempo de jogo e na interação do jogador com o microfone. O jogador acumula pontos ao longo do tempo enquanto canta ou faz sons, e esses pontos são somados à pontuação total. Além disso, o jogo possui um temporizador (`gameTimer`) que limita a duração da partida. Se o tempo acabar e o jogador ainda estiver vivo, ele vence o jogo e a tela de vitória é exibida.

3.5 Sons e Efeitos

A trilha sonora é um dos elementos centrais do jogo. A música principal é reproduzida durante a partida, enquanto sons adicionais, como os de vitória e derrota, são acionados em momentos chave. Os efeitos sonoros aumentam a imersão do jogador e tornam a experiência de jogo mais rica. Além disso, o jogo faz uso de um fundo sonoro constante para criar uma ambientação relaxante nas telas de início e seleção de nível.

4. Conclusão

O desenvolvimento do jogo **IPVC Audio-Surf** proporcionou uma valiosa experiência prática na criação de um jogo interativo baseado em som e reflexos, usando a biblioteca **p5.js**. Desde o início, o principal objetivo foi criar uma experiência envolvente que integrasse música e participação ativa do jogador através de sons captados pelo microfone. Tal tipo de interação proporciona uma jogabilidade inovadora, onde a música influencia diretamente a mecânica dos obstáculos, e a voz do jogador serve como um elemento essencial para o funcionamento do jogo.

Durante o processo de implementação, vários desafios surgiram, principalmente relacionados à captura e análise de som em tempo real. Utilizar a biblioteca **p5.sound** e as suas funcionalidades para manipulação de áudio foi um passo essencial para que o jogo se adaptasse dinamicamente ao volume captado pelo microfone. Além disso, o equilíbrio entre a dificuldade do jogo e a responsividade das teclas de movimento foi crucial para garantir que a experiência fosse tanto desafiadora quanto justa para o jogador.

Outro ponto importante foi a organização da estrutura de dados, especialmente na manipulação dos obstáculos e das diferentes telas do jogo (inicial, de seleção de nível, e de fim de jogo). Utilizar métodos específicos para controlar a transição entre as telas e para lidar com a lógica de colisão ajudou a manter o código modular e fácil de depurar. Isso foi fundamental durante o desenvolvimento iterativo, permitindo que melhorias fossem implementadas de forma incremental e organizada.

Além disso, a implementação de um sistema de **"top scores"** adiciona um fator de competição saudável, incentivando o jogador a tentar melhorar a sua pontuação e continuar a jogar para bater os seus recordes pessoais.

Por fim, o projeto atingiu os seus objetivos principais ao proporcionar uma experiência divertida, interativa e desafiadora. A integração da música e da voz como parte ativa da jogabilidade acrescentou uma camada extra de imersão e originalidade ao jogo. Com mais tempo e recursos, seria possível explorar ainda mais essa mecânica, adicionando níveis adicionais, novos tipos de obstáculos e uma maior variedade de faixas musicais.

5. Referências Bibliográficas

- McCarthy, L. et al. **"Getting Started with p5.js: Making Interactive Graphics in JavaScript and Processing."** Maker Media, 2015.
- Karlsson, A. **"AudioSurf Game Mechanics and Analysis"**. Indie Game Developers Conference Proceedings, 2009.
- p5.js. **"p5.js Reference Manual."** Acesso em <https://p5js.org/reference/>.
- p5.sound. **"p5.sound: A Sound Library for p5.js"**. Acesso em <https://p5js.org/reference/#/libraries/p5.sound>.
- Game Programming Patterns, Robert Nystrom. **"Game Loops and State Management."** GameDev Press, 2014.