



Zumbi Shooter - ESTG Edition

Exercício Prático- Sistemas multimédia

João Ribeiro

ENGENHARIA DA COMPUTAÇÃO GRÁFICA E MULTIMÉDIA 2024/2025

1. OBJETIVO DO JOGO

O objetivo deste projeto é desenvolver um exercício multimédia interativo utilizando a biblioteca P5.js. O jogo “Zumbi Shooter - ESTG Edition” oferece uma experiência de tiro onde os jogadores devem usar um microfone para disparar projéteis para salvar os alunos dos zumbis. A interação baseada em som acrescenta um desafio único e divertido ao jogo.

2. DESCRIÇÃO SUMÁRIA

O jogo é estruturado em diferentes níveis, cada um apresentando desafios distintos e zumbis para derrotar. O jogador controla um herói que deve disparar em zumbis para salvar os alunos, enquanto gera vidas e pontos. O tiro é ativado pela detecção de som, tornando a jogabilidade mais interativa. As telas do jogo incluem um menu inicial, um tutorial, uma seleção de nível, uma tela de jogo e uma tela final.

3. FUNCIONALIDADES IMPLEMENTADAS

3.1. Funcionalidades Básicas

- **Menu Inicial:** Permite ao jogador selecionar entre diferentes níveis ou aceder a um tutorial.
- **Jogo Interativo:** O jogador controla um herói que dispara projéteis contra zumbis.
- **Microfone:** Utiliza a captura de áudio do microfone para ativar os disparos.
- **Vidas e Pontos:** O jogador dispõe de um número limitado de vidas e deve acumular pontos ao derrotar zumbis.

3.2. Funcionalidades Adicionais

Transição de Nível: Após completar um nível, o jogador pode avançar para o próximo.

Tela Final: Exibe a pontuação final e permite que o jogador reinicie o jogo.

Tutorial Interativo: Fornece instruções sobre como jogar e interagir com o jogo.

Informações do Aluno: Exibe o nome, número do aluno e sigla do curso na tela.

4. DETALHES SOBRE ASPETOS RELEVANTES DA IMPLEMENTAÇÃO

4.1. Estrutura do código

O código encontra-se estruturado em diversas funções, cada uma responsável por um componente específico do jogo: interface do utilizador, mecânicas de jogo, interação com dispositivos de áudio e apresentação de dados do jogador.

```
let volume = mic.getLevel();

if (volume > 0.1 && millis() - tempoUltimoTiro > 1000 / maxProjeteis) {
  heroi.atirar(projeteis, nivel);
  tiroSom.play();
  tempoUltimoTiro = millis();
}

for (let i = projeteis.length - 1; i >= 0; i--) {
  if (projeteis[i]) {
    projeteis[i].mover();
    projeteis[i].desenhar();

    if (projeteis[i].foraDaTela()) {
      projeteis.splice(i, 1);
    }
  }
}
```

Figura 1 - Captura de Áudio e Disparo

O código verifica interações entre zumbis e um personagem. Ao ser atingido, o personagem perde pontos de vida, um efeito sonoro é reproduzido e o zumbi é eliminado da partida. Caso o personagem perca toda a sua vida ou atinja uma pontuação específica, o jogo passa para uma nova fase.

```
// Verifica se o zumbi atinge o herói
if (zumbis[i] && heroi && zumbis[i].atinge(heroi)) {
    vidas--;
    vidaSom.play();
    zumbis.splice(i, 1);

    if (vidas <= 0 && pontos < 10) {
        tela = 3;
    }
}
```

Figura 2 - Verificar se zumbi atinge herói

```
if (proj.x > xMinZumbi && proj.x < xMaxZumbi && proj.y > yMinZumbi && proj.y < yMaxZumbi) {
    zumbis.splice(i, 1);
    projeteis.splice(j, 1);
    pontos++;
    break;
}
```

Figura 3 - Verifica se Zumbi é atingido

Este código verifica se um projétil atingiu um zumbi, remove tanto o zumbi quanto o projétil das suas respectivas listas, aumenta a pontuação do jogador e encerra a verificação de outros possíveis acertos.

```

for (let j = projeteis.length - 1; j >= 0; j--) {
  if (projeteis[j] && projeteis[j].x !== undefined && projeteis[j].y !== undefined) {
    let proj = projeteis[j];

    if (proj.x > xMinZumbi && proj.x < xMaxZumbi && proj.y > yMinZumbi && proj.y < yMaxZumbi) {
      zumbis.splice(i, 1);
      projeteis.splice(j, 1);
      pontos++;
      break;
    }
  }
}

```

Figura 4 - Colisão zumbi

O trecho de código verifica se algum projétil colidiu com um zumbi. Caso uma colisão seja detetada, tanto o zumbi quanto o projétil são removidos dos respectivos arrays, e a pontuação do jogador é incrementada em 1. O ciclo é executado de forma reversa para evitar problemas de indexação ao eliminar elementos.

```

for (let j = alunos.length - 1; j >= 0; j--) {
  if (alunos[j] && alunos[j].x !== undefined && alunos[j].y !== undefined) {
    let d = dist(zumbis[i].x, zumbis[i].y, alunos[j].x, alunos[j].y);
    if (d < zumbis[i].raio + alunos[j].raio) {
      alunos.splice(j, 1);
      alunoMorreSom.play();
      break;
    }
  }
}

```

Figura 5 - Aluno colide com zumbi

Este trecho de código verifica se um aluno colidiu com um zumbi. Para fazer isso, ele itera o array de alunos ao contrário, certificando-se de que cada aluno existe e possui coordenadas definidas. Calcula a distância entre o zumbi e o aluno, e se a distância for menor que a soma dos seus raios, considera que ocorreu uma colisão. Neste caso, o aluno é retirado da matriz e o som da morte é tocado. O loop para depois da remoção, para evitar verificações desnecessárias.

4.2. Desenho das Informações do Jogo

Todas as informações relativas a vidas, pontos, níveis e informações do criador do jogo são exibidas em locais estratégicos, permitindo que o jogador possa monitorizar o seu desempenho durante o jogo.

```
// Desenho das informações do jogo
fill(255);
textSize(20);
text(`Nível: ${nivel}`, width - 20, 30); // Nível no canto superior direito
textSize(20);
text(`Vidas: ${vidas}`, width - 20, 60); // Vidas no canto superior direito

// Adicionando pontos no canto superior esquerdo
textSize(20);
text(`Pontos: ${pontos}`, 100, 30); // Coloca os pontos no canto superior esquerdo

// Desenhar informações do aluno
fill(255);
textSize(16); // Tamanho do texto para as informações do aluno
textAlign(RIGHT, BOTTOM); // Alinha o texto à direita e na parte inferior
text('João Ribeiro', width - 10, height - 70); // Exibe o nome do aluno
text('24534', width - 10, height - 90); // Exibe o número do aluno
text('Curso: ECGM', width - 10, height - 110); // Exibe a sigla do curso
```

5. ASPETOS QUE PODERIAM SER MELHORADOS

5.1. Detecção de som aprimorada

A sensibilidade da detecção de som pode ser ajustada para fornecer uma experiência de jogo mais fluida, considerando variações em ambientes e níveis de ruído.

5.2. Imagens de Fundo para os níveis

Um aspeto que gostaria de ter implementado, mas que não consegui, foi a adição de imagens distintas de fundo para cada nível do jogo. Esta melhoria poderia ter proporcionado uma experiência visual mais rica e envolvente.

5.3. Adição de animações e efeitos visuais

Implementar animações mais elaboradas para os zumbis e o herói, bem como efeitos visuais para tiros e colisões, pode melhorar significativamente a experiência do jogador.

5.4. Acessibilidade

A introdução de opções de acessibilidade, como modos de jogo alternativos que não dependem de som, pode tornar o jogo mais inclusivo para jogadores com deficiência auditiva.

CONCLUSÃO

Em suma, o desenvolvimento deste projeto revelou-se uma experiência enriquecedora e desafiante. A implementação de um jogo com múltiplos níveis, elementos visuais, efeitos sonoros e interfaces interativas permitiu-me explorar de forma prática os conceitos de programação orientada a objetos, interação com utilizadores e gerenciamento de recursos multimédia. Através do uso de bibliotecas gráficas e sons, bem como recursos adicionais, como controle de voz integrado, complexidade e jogabilidade podem ser adicionados para fornecer uma experiência de jogo mais envolvente.

Além disso, o desenvolvimento modular do código, criando funcionalidades específicas para cada funcionalidade, não só melhora a organização e manutenção do projeto, mas também proporciona uma potencial expansão para melhorias futuras. O relatório documenta o processo de construção e os desafios encontrados, e as soluções implementadas podem servir de base para novas iterações do projeto ou mesmo para o desenvolvimento de jogos com estruturas e mecânicas semelhantes.

Por fim, este projeto não só me ajudou a aprofundar os conhecimentos técnicos adquiridos ao longo da disciplina, mas também enfatiza a importância de sistemas interativos bem desenhados, onde a experiência do utilizador é dada prioridade máxima.