**Space Game**

### **Sistema de Scoreboard via API**

**Objetivo:** Exibir os melhores scores dos jogadores no painel inicial.

* **Alterações:**
  + Adição da função updateScoreboard() com fetch() para buscar dados da API: <http://localhost:3000/api/scores/>
  + Ordenação dos resultados pelo campo score (decrescente) e exibição dos 5 primeiros.
  + Exibição em uma <ul id="scores"> no painel HTML.
* **Como foi feito:**  
   Utilizou-se fetch() com then() para transformar os dados JSON em HTML dinâmico.

### **Envio de Score ao Final do Jogo**

**Objetivo:** Registar a pontuação no final da partida.

* **Alterações:**
  + Implementação da função saveScore() que faz um POST para a API com os dados:

js

CopiarEditar

{  
 datascore: new Date().toISOString(),  
 nickname,  
 score,  
 game: 'Space Game'  
}

* + Essa função é chamada no botão de "Game Over".
* **Como foi feito:**  
   Envio dos dados usando fetch() com method: POST e body: JSON.stringify(data).

### **Tela de Game Over e Controle de Fluxo**

**Objetivo:** Exibir um overlay visual de fim de jogo com opções.

* **Alterações:**
  + Criação do #gameOverOverlay no HTML.
  + Exibição ao final do jogo com style.display = 'flex'.
  + Botão "Continuar" que:
    - Salva o score
    - Atualiza o ranking
    - Retorna para o painel inicial
* **Como foi feito:**  
   O controle de visibilidade foi feito via DOM (.style.display) e eventos com onclick.

### **Tela de Pausa**

**Objetivo:** Permitir ao jogador pausar o jogo e escolher entre continuar ou sair.

* **Alterações:**
  + Adição do pauseOverlay com botão "Continuar" e "Sair"
  + Pressionar a tecla "P" ativa/desativa o pause
  + Se pausado: o jogo para (paused = true) e exibe overlay
  + Se continuar: fecha overlay e retoma o loop de jogo
* **Como foi feito:**  
   Uso de keydown para detectar "P", alteração do paused e clearInterval().

### **Sistema de Inimigos com Dificuldade**

**Objetivo:** Tornar o jogo mais desafiador com inimigos variados.

* **Alterações:**
  + Adição de três tipos de inimigos:
    - **Fracos**: 1 de vida, velocidade alta
    - **Médios**: 2 de vida, velocidade moderada
    - **Fortes**: 3 de vida, lentos, muito dano
  + Cada um com sprite personalizado
  + Função spawnEnemies() com lógica de sorteio
* **Como foi feito:**  
   Arrays e objetos com atributos (type, life, damage, fireChance), lógica com Math.random().

### **Melhorias Visuais e Interface**

**Objetivo:** Tornar a UI mais clara e interativa.

* **Alterações:**
  + Exibição da barra de vida (drawHealthBar())
  + Adição de instruções (ex: “Aperte 'P' para pausar”)
  + Painel com scores e nickname
  + Troca de telas fluida entre menu, jogo e game over

## **Sistema de Som do Jogo**

**Objetivo:** Garantir que a música de fundo funcione corretamente desde o menu principal, continue durante o jogo e evite reinícios desnecessários.

#### **1. Início da Música após Interação do Usuário**

* **Problema:** Navegadores modernos bloqueiam autoplay de áudio antes de interação.
* **Solução:**  
   Adicionada uma escuta de evento global:

js

CopiarEditar

document.body.addEventListener("click", () => {  
 startBackgroundMusic();  
}, { once: true });

* **Como foi feito:**  
   O evento click com { once: true } garante que a música toque apenas uma vez após a primeira interação.

#### **2. Prevenção de Reinício Desnecessário da Música**

* **Problema:** A música de fundo reiniciava (voltava ao início) toda vez que o jogo acabava ou o menu era acessado.
* **Solução:**  
   A função startBackgroundMusic() foi modificada para tocar a música **apenas se estiver pausada**, sem reiniciar:

js

CopiarEditar

function startBackgroundMusic() {  
 if (backgroundMusic.paused) {  
 backgroundMusic.play().catch((e) => {  
 console.warn("Autoplay bloqueado. Música será iniciada após interação.");  
 });  
 }  
}

* **Como foi feito:**  
   Usado backgroundMusic.paused como condição antes de executar .play().

#### **3. Reinício Controlado da Música no Início de Jogo**

* **Objetivo:** Ao iniciar uma nova partida, a música deve reiniciar do início para dar sensação de "novo começo".
* **Solução:**  
   Dentro da função startGame(), mantida a reinicialização da música:

js

CopiarEditar

backgroundMusic.currentTime = 0;  
startBackgroundMusic();

* **Como foi feito:**  
   Reinício manual com currentTime = 0 apenas no momento de início do jogo, sem interferir em outras telas.

#### **4. Pausa da Música de Fundo em Game Over**

* **Objetivo:** Substituir a música de fundo pela trilha de game over ao morrer.
* **Solução:**  
   Pausa da música de fundo e execução de gameOverMusic e gameOverVoice:

js

CopiarEditar

backgroundMusic.pause();  
gameOverMusic.currentTime = 0;  
gameOverMusic.play();  
gameOverVoice.currentTime = 0;  
gameOverVoice.play();