

# Prompt para GitHub Copilot: Simulador de Partículas e Áudio 3D

## CONTEXTO GERAL

Criar um simulador profissional de partículas 3D que reage a áudio WAV, com física gravitacional e comportamento fluídico. Desenvolvimento em fases incrementais para manter coerência e testabilidade.

### ESPECIFICAÇÕES TÉCNICAS:

- Plataforma: Aplicação Web (HTML5 + Three.js + WebGL)
- Otimização: Intel UHD Graphics 128MB, 8GB RAM, Core i5
- Objetivo: Performance estável com 200-5000 partículas (escalável até 20.000)

## FASE 1: ESTRUTURA BASE E CARREGAMENTO DE ÁUDIO

### Objetivo: Interface funcional + carregamento de arquivo WAV

Copilot, crie:

#### 1. Estrutura HTML básica:

- Canvas 3D fullscreen responsivo
- Interface de controles lateral (collapsible)
- Drag & drop para arquivos WAV
- Botões: Play/Pause, Reset
- Slider para volume

#### 2. Setup Three.js:

- Cena 3D com câmera orbital
- Iluminação ambiente suave
- Controles de câmera (OrbitControls)
- Renderer otimizado para Intel UHD

#### 3. Sistema de áudio:

- Web Audio API setup
- AnalyserNode para análise de frequência
- Carregamento e decodificação de WAV
- Visualização básica do waveform

### REQUISITOS DE PERFORMANCE:

- 60fps estável sem partículas

- Análise FFT em 1024 samples
- Interface responsiva

**TESTE DA FASE 1:** Carregar música WAV, tocar com waveform visível, câmera funcionando.

---

## FASE 2: SISTEMA BÁSICO DE PARTÍCULAS

**Objetivo:** Partículas coloridas reagindo ao áudio

**Copilot, implemente:**

**1. Sistema de partículas:**

- Geometry instanciada para performance
- 500 partículas esféricas iniciais
- Cores do arco-íris (HSL mapping)
- Posições aleatórias em esfera 3D

**2. Reação ao áudio:**

- Dividir espectro em 8 bandas de frequência
- Mapear bandas para grupos de partículas
- Escala/movimento baseado em amplitude
- Cores reagindo à frequência dominante

**3. Controles básicos:**

- Slider: Número de partículas (200-2000)
- Slider: Sensibilidade ao áudio
- Toggle: Mostrar/ocultar partículas

**OTIMIZAÇÃO OBRIGATÓRIA:**

- Buffer geometry reutilizável
- Update apenas partículas visíveis
- Throttle de análise de áudio (30fps max)

**TESTE DA FASE 2:** Partículas dançando com a música, cores mudando, performance estável.

---

## FASE 3: ANÁLISE AVANÇADA DE FREQUÊNCIA E MAPEAMENTO

**Objetivo:** Mapeamento sofisticado áudio-visual

**Copilot, desenvolva:**

**1. Análise de frequência avançada:**

- 16 bandas especializadas (graves, médios, agudos)
- Detecção de picos e transientes
- Smoothing temporal para evitar jittering
- Mapeamento logarítmico de frequências

## 2. Mapeamento visual:

- Graves (20-250Hz): Partículas grandes, movimento lento, cores quentes
- Médios (250-4000Hz): Movimento moderado, cores neutras
- Agudos (4000-20000Hz): Partículas rápidas, cores frias
- Transientes: Bursts de partículas

## 3. Sistema de rastros:

- Trail renderer para cada partícula
- Comprimento baseado em velocidade
- Alpha decay configurável
- Slider para intensidade dos rastros

## CONTROLES ADICIONAIS:

- Mapeamento de frequência personalizado (sliders por banda)
- Intensity curve (attack, decay, sustain, release)

**TESTE DA FASE 3:** Diferentes instrumentos gerando padrões visuais únicos.

---

# FASE 4: FÍSICA GRAVITACIONAL E COMPORTAMENTO FLUÍDICO

## Objetivo: Simulação física realística

### Copilot, implemente:

#### 1. Sistema gravitacional:

- Centros de gravidade móveis (3-5 simultâneos)
- Força gravitacional realística ( $F = Gm_1m_2/r^2$ )
- Órbita estável sem colapso
- Gravidade influenciada por amplitude do áudio

#### 2. Comportamento fluídico:

- Flocking algorithm (separação, alinhamento, coesão)
- Viscosidade ajustável
- Fluxos direcionais baseados em frequência
- Turbulência procedural

### 3. Modos de simulação:

- "Campo Neutro": Partículas em brownian motion até perturbação
- "Sistema Solar": Órbitas estáveis ao redor de centros
- "Fluido": Comportamento líquido com surface tension
- "Tempestade": Caos controlado com vórtices

### OTIMIZAÇÃO CRÍTICA:

- Spatial partitioning para cálculos de vizinhança
- Simplified physics para partículas distantes
- Adaptive timestep

**TESTE DA FASE 4:** Transições suaves entre modos, física convincente.

---

## FASE 5: INTERATIVIDADE E CONTROLES AVANÇADOS

### Objetivo: Controle total do usuário

**Copilot, crie:**

#### 1. Interação com mouse/touch:

- Raycasting para seleção de partículas
- Drag para criar campos gravitacionais temporários
- Click para adicionar/remover centros de gravidade
- Scroll para zoom da câmera

#### 2. Interface de controle completa:

- Painel de física (gravidade, viscosidade, massa)
- Painel de visual (cores, rastros, brilho)
- Presets salvos em localStorage
- Randomização controlada

#### 3. Modos de visualização:

- Osciloscópio 2D integrado
- Vista debug (vetores de força)
- Modo performance (FPS counter, stats)
- Fullscreen mode

### CONTROLES ESSENCIAIS:

- Timeline scrubber para navegar na música

- Speed control (0.1x - 4x)
- Gravity wells editor
- Color palette customization

**TESTE DA FASE 5:** Experiência interativa completa, controles intuitivos.

---

## FASE 6: POLIMENTO E RECURSOS PROFISSIONAIS

**Objetivo: Produto final profissional**

**Copilot, finalize:**

**1. Presets visuais:**

- "Aurora Boreal": Fluxos verticais, cores frias
- "Galáxia Espiral": Rotação orbital, centro massivo
- "Explosão Estelar": Expansão radial nos picos
- "Oceano Sonoro": Ondas fluídicas horizontais
- "DNA Sônico": Hélice dupla seguindo harmônicos

**2. Exportação e compartilhamento:**

- Screenshot em alta resolução
- Gravação de vídeo WebM (MediaRecorder API)
- Export de configurações (JSON)
- URL sharing com parâmetros

**3. Otimização final:**

- Dynamic LOD system
- Frustum culling
- Memory management
- Progressive loading

**4. Acessibilidade:**

- Keyboard shortcuts
- Reduced motion mode
- High contrast option
- Screen reader compatibility

**RECURSOS BÔNUS:**

- Análise de BPM automática
- Sync com tempo musical

- Multi-track support
  - VR mode (WebXR)
- 

## **DIRETRIZES DE DESENVOLVIMENTO PARA TODAS AS FASES:**

### **PERFORMANCE (CRÍTICO para Intel UHD):**

- Sempre manter 30fps mínimo
- Usar Object pooling para partículas
- Batch rendering sempre que possível
- Throttle updates em 60fps max
- Profiling contínuo com stats.js

### **CÓDIGO LIMPO:**

- Arquitetura modular (classes separadas)
- Comentários detalhados em português
- Error handling robusto
- Progressive enhancement

### **TESTABILIDADE:**

- Cada fase deve ser funcional independentemente
- Debug mode em todas as fases
- Fallbacks para hardware limitado
- Logs detalhados para troubleshooting

### **COMPATIBILIDADE:**

- Chrome/Firefox/Safari/Edge
  - Mobile responsive (touch controls)
  - Graceful degradation
- 

## **INSTRUÇÕES ESPECÍFICAS PARA O COPILOT:**

1. **NUNCA** implemente mais de uma fase por vez
2. **SEMPRE** teste a fase atual antes de prosseguir
3. **PRIORIZE** performance sobre recursos visuais
4. **COMENTE** cada função crítica
5. **INCLUA** console.logs para debugging

6. **AVISE** sobre possíveis gargalos de performance

**COMEÇAR PELA FASE 1 AGORA!**