



Product Requirements Document — Neon Snake v2.7 (MVP Mobile)



Visão Geral

Nome do produto: *Neon Snake — Synthwave Edition*

Tipo: Game interativo HTML5 (PWA-ready)

Plataforma-alvo: Navegadores mobile e desktop modernos

Stack: HTML + CSS + JavaScript puro (Web Audio API + Canvas 2D)

Status: MVP funcional completo, em fase de estabilização mobile

Visão:

Um jogo minimalista e audiovisual, inspirado em *Snake* e no universo *TRON*, onde **som e luz reagem ao ritmo do jogador**. Cada sessão é uma micro performance generativa.



Objetivos do MVP

- Garantir **jogabilidade fluida e responsiva** em dispositivos móveis.
 - Integrar **áudio procedural sincronizado com o gameplay**.
 - Fornecer **feedback visual dinâmico e sinestésico** para cada ação.
 - Permitir **replay contínuo sem telas de pausa** (looping automático).
 - Tornar o jogo **instalável e jogável offline** (PWA).
-



Core Gameplay Loop

- Início:** Jogador toca em START → música procedural inicia.
- Movimento:** Snake se desloca em grid 30×30, coletando maçãs.
- Eventos:**
 - Comer maçã → aumento de score + som + partículas + combo.
 - Power-ups → efeitos temporários e sinestésicos.

4. Progressão:

- Cada 50 pontos → mudança de mundo (tema + BPM + trilha).
- Novos obstáculos gerados dinamicamente.

5. Game Over: Explosão luminosa → reinício automático.

Mecânicas Implementadas

Sistema	Descrição
Mundos	5 mundos jogáveis: <i>Cyberpunk</i> , <i>Bloodmoon</i> , <i>Matrix</i> , <i>Ultraviolet</i> , <i>Rainbow</i> — cada um com trilha procedural própria e variações de cor, BPM e timbre.
Power-ups	5 tipos: ⚡ Golden, 🔥 Speed, 👻 Ghost, 💰 Multiplier, 📉 Shrink — todos com temporizadores e efeitos sonoros dedicados.
Combo System	Acúmulo progressivo de multiplicadores por tempo curto entre maçãs.
World Transitions	Animações e flashes cromáticos com troca de música, cor e filtros.
Áudio Procedural	Motor completo com 5 instrumentos: Kick, Bass, Lead, Pad, Arpeggio. Sequências sincronizadas por BPM com filtro e forma de onda customizados por mundo.
Visuals	Canvas com glow, blur, efeitos de grade pulsante e orbes de energia para frutas e power-ups.
UI e HUD	Badges reativas, timer de power-ups, score/best persistente via <code>localStorage</code> .
Controles	Suporte a teclado e D-Pad tátil (HTML interativo).

Direção Visual

Estilo: *Synthwave minimalista* — preto, magenta e ciano com gradientes pulsantes.

Referências: *TRON Legacy*, *Blade Runner 2049*, *Rez Infinite*.

Regras visuais principais:

- Grid pulsante em resposta à batida do som.
- Luzes projetadas no canvas a partir de halos dinâmicos.

- Cada mundo redefine bordas, sombras e cor principal do HUD.
-

Design Sonoro

Motor: Web Audio API, sem dependências.

Composição procedural:

- Cada “beat” sincroniza:
 - Kick (base rítmica 808-style).
 - Bass (sequência grave looping).
 - Lead (melodia progressiva).
 - Pad (ambiente contínuo).
 - Arpeggio (a cada 2 batidas).
 - **BPM variável por mundo:** 110–130.
 - **Efeitos:** Delay analógico com feedback e variação de cutoff por tema.
 - **Eventos:**
 - Comer maçã → nota aguda curta.
 - Pegar power-up → arpeggio ascendente.
 - Game over → sweep descendente.
-

Mobile & PWA

UI responsiva:

- Layout 3 colunas em desktop, compacto em telas $\leq 1024\text{px}$.
- Controles táteis com animação e feedback visual (`:active`).
- Altura do canvas limitada para evitar rolagem.

PWA (próxima entrega):

- `manifest.json` (nome, ícone, splash).
- `serviceWorker.js` (cache estático).
- Instalação via “Adicionar à tela inicial”.

Arquitetura Técnica

Módulo	Função
<code>index.html</code>	Estrutura base, layout, estilos e script inline.
<code>setupAudio()</code>	Inicializa motor sonoro e inicia loop rítmico procedural.
<code>startGame()</code>	Reseta variáveis e inicia o loop lógico.
<code>step()</code>	Atualiza lógica de movimento, colisão, score e power-ups.
<code>animate()</code>	Renderiza frame a frame o canvas e efeitos visuais.
<code>activatePowerUp(type)</code>	Executa o comportamento e feedback audiovisual de cada tipo.

Performance:

- Loop visual com `requestAnimationFrame`.
- Loop lógico com `setInterval` (150ms ou 75ms em speed mode).
- Cache de cores via `colorCache` para otimização de render.
- Até **100 partículas** e **10 trilhas** gerenciadas simultaneamente.

User Stories (MVP técnico)






ID	História do Usuário	Critério de Aceitação
G 1	Como jogador, quero mover a cobra com D-pad ou teclado.	Cobra responde a direções sem atraso perceptível.
G 2	Como jogador, quero ouvir música que se adapta ao mundo atual.	Ao trocar de mundo, o BPM e timbre mudam automaticamente.

G 3	Como jogador, quero receber feedback visual e sonoro ao comer uma maçã.	Cada fruta toca uma nota e gera partículas coloridas.
G 4	Como jogador, quero ativar power-ups visuais e sonoros.	Power-ups surgem periodicamente, têm duração e cor distintas.
G 5	Como jogador, quero que o jogo recomece automaticamente após perder.	Reinício ocorre em até 1s após o <i>Game Over</i> .
G 6	Como jogador, quero poder adicionar o jogo à tela inicial.	PWA instalável, executando offline.

KPIs (para versão Beta)

Métrica	Alvo
FPS médio em mobile	≥ 50
Latência sonora	< 50ms
Tempo de carregamento	< 2s
Tempo médio de sessão	> 90s
Taxa de crash/reload	0%

Roadmap (v2.7 → Beta)

Versão	Feature	Status
2.7	MVP funcional com áudio procedural e mundos reativos	
2.8	PWA + manifest + cache offline	
2.9	Variações dinâmicas de BPM conforme pontuação	
3.0	Modo “Eclipse Cinemático” (visual alternativo escuro)	
3.1	Edição pública no Itch.io + otimização mobile final	

Declaração Criativa

*O Neon Snake é uma serpente feita de som.
Cada batida é um movimento, cada luz uma nota.
Não se joga apenas com os dedos — mas com o ritmo.*
