WingBeat

Browser-Based Rhythm Game with Real-Time Audio Analysis

Project Overview

Created a rhythm-action game that generates obstacles in real-time based on audio frequency analysis. The game uses Web Audio API with FFT processing to detect beats and create synchronized gameplay elements, delivering 60 FPS performance through optimized rendering techniques.

60	7	<50ms
FPS Target	Frequency Bands	Beat Detection

Core Features

Real-Time Audio Analysis

- Web Audio API with AnalyserNode
- FFT frequency decomposition
- 7 logarithmic frequency bands
- Energy-based beat detection algorithm
- Sub-50ms detection latency

Procedural Generation

- Dynamic obstacle generation from audio
- Parametric curves using Bezier mathematics
- Difficulty scaling based on song intensity
- No pre-authored content required

Performance Optimization

- Object pooling to prevent GC pauses
- Visibility culling for off-screen objects
- Efficient sprite batching
- Frame-rate independent physics

Technical Implementation

Audio Processing Pipeline

- Audio source → AnalyserNode → FFT data
- Frequency band mapping (sub-bass to brilliance)
- Normalization and smoothing
- Beat onset detection with adaptive threshold

Rendering System

- Phaser 3 game framework
- WebGL rendering with Canvas fallback
- Neon aesthetic with HSL color calculations
- Multi-layer glow effects with additive blending

Game Architecture

- Singleton pattern for system managers
- Event-driven architecture for loose coupling
- State machine for game flow
- Configuration-driven design for easy tuning

Tech Stack

> **Framework:** Phaser 3
> **Language:** TypeScript

> **Audio:** Web Audio API, FFT
> **Build:** Vite, Webpack
> **Backend:** Supabase (leaderboards)

Skills Demonstrated

Phaser 3 | TypeScript | Web Audio API | FFT Analysis | Canvas Rendering | Game Development | Performance Optimization | Procedural Generation | State Machines

Deliverables

- Browser-based rhythm game
- Real-time audio frequency analysis
- Beat detection algorithm
- Procedural obstacle generation
- 60 FPS optimized rendering
- Responsive cross-device design
- Leaderboard system
- Production build pipeline

Why This Matters

This project demonstrates real-time data processing and visualization capabilities. The same patterns (continuous data stream → processing → real-time display) apply to dashboards, monitoring systems, IoT

applications, and any system requiring live data visualization with high performance.