World of Tetris: Fragments of the Sleepless Realm

Game Design Document — Page 18

Technical Architecture Overview

18. Architectural Philosophy

The technical architecture of World of Tetris is designed for clarity, modularity, and educational value. Systems are loosely coupled, data-driven where possible, and organized around a deterministic core loop.

18.1 High-Level Stack

1. Frontend: JavaScript (ES Modules), HTML5 Canvas
2. Backend: PHP (session-based), MySQL database
3. Assets: JSON (data), PNG/Sprite Sheets (visuals), Audio files

18.2 Core Game Loop

The game loop is driven by requestAnimationFrame, maintaining a fixed update order: input handling, simulation update, AI evaluation, rendering, and feedback dispatch.

18.3 State Management

1. Finite State Machine (FSM) for menu, gameplay, pause, and game over
2. Deterministic board state updates
3. Isolated systems for Pressure, Difficulty, Pets, and AI

18.4 Data-Driven Systems

Character abilities, pet behaviors, difficulty scaling, and localization are defined via data files. This allows rapid iteration and safe extension without modifying core logic.

18.5 Persistence & Accounts

Player accounts store progress, settings, language, difficulty, unlocked content, and statistics in MySQL. Server-side validation ensures integrity while allowing stateless frontend operation.

Educational Notes

This architecture is intentionally readable and modular, making it suitable for teaching real-world game development patterns, debugging, and system expansion.