Mat Mania Wrestling Game - Project State Documentation

Project Overview

Building a browser-based wrestling game inspired by the 1985 arcade game "Mat Mania" with modern features. The game uses HTML5 Canvas, pixel art sprites, and aims to capture classic arcade feel while adding progression systems and modern gameplay hooks.

Current Implementation Status

✅ Completed Features

1. Core Game Engine

- **HTML5 Canvas** setup with 800x600 resolution
- **60 FPS game loop** with requestAnimationFrame
- **Pixel-perfect rendering** (imageSmoothingEnabled: false)
- **Image loading system** with file upload for sprites

2. Sprite System

- **Sprite dimensions:** 32x32 pixels base, scaled 1.5x
- **Animation system:**
  - 4 directional movement (up, down, left, right)
  - 3 frames per animation
  - Row-based sprite sheet organization (9 columns x 4 rows expected)
  - Animation speed: 6 frames for walking, 10 for idle

3. Movement System

- **Player controls:** Arrow keys or WASD
- **Diagonal movement** with proper speed normalization (0.707 multiplier)
- **Movement speed:** Player = 3 pixels/frame, AI = 2 pixels/frame
- **8-directional freedom** within ring boundaries

4. Boundary System

- **Trapezoid-shaped ring boundaries** to match perspective
- **Dynamic boundary editor** with draggable corners
- **Visual boundary indicators** during gameplay (green dashed lines)
- **Collision detection at wrestler's feet** (80% down the sprite)

- **Perspective-aware boundaries** (narrower at top, wider at bottom)

5. AI System

- **Follow behavior** with 120-pixel minimum distance
- **Smooth pursuit** with normalized direction vectors
- **Proper animation states** based on movement direction
- **Boundary respect** for AI movement

6. Visual Debugging

- **Boundary visualization** (green trapezoid)
- **Collision point indicators** (red dot at feet)
- **Sprite bounding boxes** (yellow for player, cyan for opponent)
- **Coordinate display** for boundary tuning

🔧 Technical Details

Ring Boundaries Object Structure

```javascript
RING_BOUNDS = {
    topLeft: 280,      // X coordinate of top-left corner
    topRight: 520,     // X coordinate of top-right corner
    top: 290,          // Y coordinate of top edge
    bottomLeft: 200,   // X coordinate of bottom-left corner
    bottomRight: 600,  // X coordinate of bottom-right corner
    bottom: 480        // Y coordinate of bottom edge
}
```

Collision Detection Method

- Checks collision at sprite's feet position (not top-left corner)
- Uses interpolation for perspective boundaries based on Y position
- Formula: `leftBound = topLeft + (bottomLeft - topLeft) * yProgress`

Critical Code Patterns

1. **Sprite drawing from sheet:**
    - Frame width = spriteSheet.width / 9
    - Frame height = spriteSheet.height / 4
    - Source position calculated by frame index and animation row

2. **Boundary clamping with perspective:**
   - First clamp Y position
   - Calculate horizontal bounds based on Y
   - Then clamp X position

📁 Required Assets

1. **Ring background image** - Full arena with mat
2. **Player sprite sheet** - 9x4 grid of animation frames
3. **Opponent sprite sheet** - Same format as player

🎮 Game Design Goals (From Original Discussion)

Core Philosophy

- "Easy to learn, difficult to master"
- Quick matches (90-180 seconds)
- Mobile-friendly controls
- Dramatic rope physics and finishing moves

Planned Features Not Yet Implemented

1. **Combat System**
   - Two-button system (Strike + Grapple)
   - Timing-based counters
   - Momentum system
   - Signature moves
2. **Match Features**
   - Health/stamina bars
   - Rope bounce physics (2.5x force multiplier planned)
   - Pin/submission system
   - Multiple match types
3. **Progression Systems**
   - Character creation/customization
   - Move unlocks
   - Championship progression
   - Stats (Power, Speed, Stamina, Charisma)
4. **Monetization (F2P Model)**

- Cosmetic items
- Battle pass system
- Character slots
- No pay-to-win mechanics

🐛 Known Issues/Adjustments Needed

1. Sprite sheet frame counts may need adjustment based on actual artwork
2. Foot offset (80%) may need tuning based on sprite design
3. Boundary editor saves could be persisted to localStorage
4. Performance optimization needed for mobile devices

💡 Next Development Priorities

1. Implement basic strike/grapple mechanics
2. Add health system and match win conditions
3. Implement rope bounce physics
4. Add sound effects and hit feedback
5. Create multiple AI difficulty levels

🔑 Key Insights from Research

Based on analysis of wrestling games across all eras:

- **Timing-based combat** > button mashing
- **Momentum systems** feel more authentic than health bars
- **Creation tools** extend game lifespan indefinitely
- **Technical stability** trumps feature quantity
- **Simple controls** can create deep gameplay

📝 How to Continue Development

To restore this project in a new conversation:

1. Share this document with the AI
2. Provide the current HTML file
3. Upload the three image assets (ring, player, opponent sprites)
4. Specify which feature to work on next

Current file can be run by:

1. Save as `.html` file

2. Open in modern browser
3. Load the three required images
4. Adjust boundaries in editor
5. Click "Start Game"

🎯 Success Metrics (Original Goals)

- Create an addictive "just one more match" experience
- Capture Mat Mania nostalgia while modernizing gameplay
- Build for browser first, then mobile, then Steam/consoles
- Focus on fun, accessible gameplay over simulation
- Enable community content creation

Code Architecture Notes

Class Structure

- **Wrestler Class**: Handles all character logic (player and AI)
  - Properties: position, sprite, animation state, velocity
  - Methods: update(), handleInput(), followPlayer(), draw()

Game States

1. **Image Loading** → 2. **Boundary Editor** → 3. **Game Loop**

Performance Considerations

- Sprite batching not yet implemented
- No particle effects system yet
- Canvas size fixed (not responsive)
- No touch controls for mobile yet

This document represents the complete state of the Mat Mania prototype as of the current conversation. Use it to continue development without losing context.