Equestrian Jump Game - Game Design Document

Game Overview

A casual pixel art equestrian jumping game where players control a horse and rider through progressively challenging show jumping courses. The goal is to complete as many jumps as possible within a time limit, advancing through increasingly difficult levels.

Core Gameplay Loop

1. Player selects an avatar
2. Player attempts to clear jumps using timing-based mechanic
3. Successfully cleared jumps earn points and progress
4. Complete enough jumps within time limit to advance to next level
5. Difficulty increases with each level
6. Game ends when player fails to complete a level
7. Final score based on levels completed and total points

Game Mechanics

Avatar Selection

5 Character Options:

- Girl with brown hair and horse ears (default)
- Girl with blonde hair and horse ears
- Brown-skinned girl with black hair and horse ears
- Golden-skinned girl (Asian) with black hair and horse ears
- White boy with brown hair and horse ears

Implementation:

- Character selection screen on game start
- Selected avatar persists through gameplay session

- Avatar is displayed on horse during jumping sequences

Jumping Mechanic (Core Gameplay)

Visual Interface:

- Vertical timing meter displayed on right side of screen
- Meter contains three zones from top to bottom:
    - **Early Zone** (Red/Gray)
    - **Perfect Zone** (Green) - target area
    - **Late Zone** (Red/Gray)
- Moving indicator travels vertically up and down the meter
- Speed of indicator movement increases with each level

Player Input:

- Single tap/click to attempt jump when indicator is in position
- No button holding or multi-button combinations

Outcome States:

1. **Too Late (After Green Zone)**
    - Horse stops short of jump
    - Horse backs up 2-3 steps
    - Time continues running
    - Horse automatically attempts same jump again
    - Player feedback “Oops! Jump again”
    - Points awarded (see scoring section)
2. **Perfect (Green Zone)**
    - Horse successfully clears jump
    - Points awarded (see scoring section)
    - Horse continues to next jump
    - Visual/audio success feedback
    - Player feedback “Perfect!”
3. **Too Early (Before Green Zone)**
    - Horse knocks rail off jump
    - Horse slows down for 2-3 steps
    - Minor point penalty or no points for that jump
    - Time continues running
    - Horse automatically continues to next jump

    - Player feedback “Keep Going!”
    - Points awarded (see scoring section)

Level Structure

Level Objective: Complete X jumps within Y seconds (values scale per level)

Level 1 Starting Values (Suggested):

- Time Limit: 60 seconds
- Required Jumps: 10 jumps
- Meter Speed: Baseline (e.g., 2 seconds per full cycle)

Progression Per Level:

- Time limit may decrease by 5 seconds OR stay constant
- Required jumps increase by 2-3 per level
- Meter speed increases by 10-15% per level
- Jump visual variety can increase (aesthetic only)

Level Completion:

- Player must complete required number of jumps before time runs out
- Successfully completing a level unlocks the next level
- Player proceeds immediately to next level after completion

Level Failure:

- Time runs out before required jumps are completed
- Game ends
- Player sees final score screen

Scoring System

Points Awarded:

- Perfect Jump (green zone): 100 points
- Jump with Fault (late timing): 50 points

Score Multipliers (Optional Enhancement):

- Consecutive perfect jumps: +10% per consecutive perfect (max 50% bonus)
- Reset on failed attempt

High Score Tracking:

- Primary metric: Highest level reached
- Secondary metric: Total points accumulated
- Both displayed on game over screen

Display During Gameplay:

- Time remaining: Countdown
- Jumps Cleared: Number of successful jumps in current level
- Rails Down: Number of knocked rails in current level
- Current Score: Total points accumulated

User Authentication & Save System

Login System:

- Email-based authentication (no password required for MVP)
- Email serves as unique user identifier
- Magic link or simple code verification recommended

Saved Data Per User:

- Highest level reached
- High score (points)
- Selected avatar preference
- Total games played (optional)
- Total jumps cleared (optional)

Data Persistence:

- Saved to backend database
- Retrieved on login
- Updated at end of each game session

Visual & Audio Design

Art Style

- Pixel art aesthetic
- Bright, cheerful color palette
- Equestrian/countryside theme

- Clear visual feedback for all game states

Required Visual Elements

- 5 avatar character sprites in various skin tones, all in the red jacket riding outfit. Same model, just different coloring
- 3 horse sprites with animation frames (idle, running, jumping) - Same model, in colors grey, black, and med brown with black mane/tail, all with various white markings
- Jump obstacles (multiple variations for visual variety)
- Background arena/course environment
- UI elements (timer, meters, score display)
- Timing meter with colored zones

Audio (I will source and provide for integrating into game)

- Background music (calm, upbeat)
- Jump success sound
- Jump fault sound (rail knock)
- Failed attempt sound (horse stopping)
- Level complete fanfare
- Game over sound

Technical Requirements

Platform

- HTML5 Canvas for rendering
- JavaScript for game logic
- Responsive design for desktop and mobile

Controls

- Mouse click (desktop)
- Touch tap (mobile)
- Single input method for jumping

Performance Targets

- 60 FPS gameplay
- Responsive input (<50ms delay)
- Fast load times (<3 seconds)

Browser Compatibility

- Modern browsers (Chrome, Firefox, Safari, Edge)
- Mobile browsers (iOS Safari, Chrome Mobile)

Game Flow

1. Start Screen

- Game title/logo
- "Start Game" button
- "How to Play" button (optional tutorial)

2. Login Screen

- Email input field
- "Continue" button
- Display returning player's high score if applicable

3. Avatar Selection

- Display 5 avatar options → Click to select
- Display 3 horse options → Click to select
- "Start Riding" button

4. Game Screen

- Arena background
- Horse and rider in center/left
- Jumps approaching from right
- Timing meter on right side
- HUD displaying: Time, Clears, Faults, Score

5. Level Transition

- Brief "Level Complete!" message
- Display level number
- Show current score
- Auto-proceed to next level after 2 seconds

6. Game Over Screen

- "Game Over" message
- Final level reached
- Final score
- "Play Again" button
- High score comparison (if improved)

MVP Feature Priority

Must Have (Phase 1)

- Avatar selection (5 characters), horse selection (3 characters)
- Jumping timing mechanic with meter
- Three outcome states (early/perfect/late)
- Level progression with increasing difficulty
- Score tracking
- Email login and save system
- Basic pixel art graphics
- Sound effects and music
- Animations (horse running, jumping, reacting)
- Score multipliers
- Tutorial/How to Play screen
- Leaderboards (compare with other players)

Future Enhancements (Phase 2)

- Multiple horse breeds
- Customization options (outfit colors, accessories)
- Different arena environments
- Power-ups or special abilities
- Multiplayer races or competitions
- Achievement system

Success Metrics

Player Engagement:

- Average session length
- Return player rate
- Average level reached

Difficulty Balance:

- Level 1 completion rate (target: 90%+)
- Level 5 completion rate (target: 50%)
- Level 10 completion rate (target: 10-20%)

Technical Performance:

- Frame rate stability (target: consistent 60 FPS)
- Input responsiveness (target: <50ms)
- Load time (target: <3 seconds)

Development Notes for Engineer

Timing Meter Implementation

- Meter should use easing function for smooth movement
- Green zone should be consistent pixel height per level
- Consider tweening library for smooth animation
- Ensure input detection is frame-perfect

Difficulty Scaling

- Use configurable JSON for level parameters
- Allow easy tweaking of timing windows
- Implement progressive difficulty curve formula

State Management

- Implement clear game states (menu, playing, paused, game over)
- Handle transitions cleanly
- Ensure proper cleanup between game sessions

Mobile Considerations

- Larger tap target for timing meter
- Touch-friendly UI elements
- Test on various screen sizes
- Optimize asset loading for mobile networks

Estimated Development Timeline

- **Week 1:** Core game loop, jumping mechanic, basic visuals
- **Week 2:** Level progression, difficulty scaling, UI/HUD
- **Week 3:** Avatar system, login/save system, polish**Week 4:** Testing, bug fixes, performance optimization

---