Changelog

Version: v3.0.1 (2026-01-15)

Documentation

- **Docs publishing**: Converted docs to Markdown + PDF and standardized printable margins.
- **Diagrams**: Updated Mermaid syntax for v11 compatibility and exported a landscape, multi-diagram PDF.

Version: v3.0.0 (2026-01-13)

Features

- **Analytics**: K-Means Overall Review (lightweight clustering + visualization).

Documentation

- **Version bump**: Updated documentation to v3.0.0 and performed a language audit (English + Māori only).
- **Documentation expansion**: Expanded docs scope and restored full diagram coverage.

Maintenance

- **Root cleanup**: Removed legacy release notes and unused asset folder.

Version: v2.3.2 (2026-01-09)

Fixes

- **Documentation**: Fixed Mermaid.js syntax errors in `SYSTEM_DIAGRAMS.md` by quoting node labels containing special characters (parentheses, newlines), ensuring diagrams render correctly on GitHub.

Version: v2.3.1 (2026-01-09)

Documentation

- **Enhanced Diagrams**: Major update to `SYSTEM_DIAGRAMS.md` adding comprehensive visualizations:
  - **Game State Machine**: Detailed state transitions (Init → Preview → Playing → GameOver).
  - **Lifecycle Sequence**: Step-by-step API call flows between Game Core, AI Engine, and Storage.
  - **Data ERD**: Entity relationships for Profile, Session, Telemetry, and Config.
  - **Detailed Architecture**: Updated high-level architecture with specific data flows.

Version: v2.3.0 (2026-01-09)

Features

- **Simplified AI Arms**: Reduced Contextual Bandit complexity from 4 to 3 arms (Easy, Standard, Challenge).
  - **Arm 0 (Easy)**: Max hints, slow timer, 5×4 grid.
  - **Arm 1 (Standard)**: Baseline settings, adaptive grid (5×4 → 4×6).
  - **Arm 2 (Challenge)**: Minimal hints, fast timer, always 4×6 grid.
- **Documentation**:
  - Created `TECHNICAL_REFERENCE.md` (consolidating Architecture, Analytics, Scoring).
  - Created `SYSTEM_DIAGRAMS.md` with visual architecture flows.
  - Updated `SYSTEM_OVERVIEW.md` with new Arm definitions.

Version: v2.2.3 (2026-01-08)

Finalization

- **Flow Index Only**: Removed remaining Score/Leaderboard paths and references.

- **Analytics Consistency**: Updated demo/history rendering to avoid Score fields.
- **Testing Output**: Updated simulations to report Flow Index only.
- **Docs Consistency**: Aligned docs to match Flow Index-only behavior.
- **Performance Optimization**:
  - Removed deprecated penalty calculations (error/streak penalties) from `computeFlowIndex` to rely solely on Fuzzy Logic rules.
  - Simplified AI engine logging and internal variable usage.

Verification

- `node tests/run-suite.js` passes.

Version: v2.2.1 (2026-01-08)

Documentation Updates

- **Design System Corrections**: Aligned `DESIGN_SYSTEM.md` with CSS (title/timer/HUD sizing, menu layout).
- **Verification**: Verified consistency between docs and code for scoring parameters and GameHistory interface.
- **Clarity**: Documented the difference between elapsed time and Flow Index in game-over UI.

Version: v2.2.0 (2026-01-05)

Major Updates

- **Flow Index Standardization**: Unified Flow Index as the primary post-game metric across Level 1–3.
- **Flow Index Engine Updates**: Refined fuzzy logic memberships/rules and applied a multiplicative penalty for show-cards usage (hint/cheat behavior).
- **Next Game Feature**: Added "Next Game" button to game-over screen for seamless progression to AI-adjusted difficulty
- **Language Consistency**: Standardized user-facing text and documentation to formal English (Māori terms preserved as cultural features)
- **Design Overhaul**: Enforced fixed 800px width layout across all pages for consistent experience

Improvements

- **Flow Index Calculation**:
  - Added richer input signals (cadence, click accuracy, color/shape accuracy) to stabilize Flow Index behavior
  - Improved telemetry logging for Flow Index diagnostics and analytics rendering
- **Code Quality**:
  - Removed all Chinese comments and documentation
  - Improved code comments and documentation clarity
  - Enhanced logging for Flow Index calculation debugging
- **Documentation**:
  - Added comprehensive `SCORING_BREAKDOWN.md` with detailed scoring formula explanations
  - Updated all documentation to English-only
  - Preserved Māori cultural terms (Matariki, Pīwakawaka, etc.)
- **Accessibility**: Added `aria-label` to buttons and interactive elements
- **Layout Fixes**: Standardized button spacing (30px gaps) and font sizes (18px/24px/96px)
- **Frontend Optimization**: Removed inline styles and absolute positioning in favor of CSS Grid/Flexbox

---

Version: v2.1.1 (2025-12-19)

Major Updates

- **Documentation Upload**: Added all documentation files to repository (removed docs from .gitignore)
- **Version Update**: Updated to v2.1.1

Version: v2.1.0 (2025-12-16)

Major Updates

- **Documentation Consolidation**: Merged and reorganized all documentation files for better maintainability
- **Flow Index Bug Fix**: Fixed critical bug where Flow Index 0.2-0.4 was incorrectly labeled as "Too Easy" (now correctly shows "Too Hard")
- **Code Improvements**: Fixed Flow Index interpretation logic in analytics summary

Documentation Improvements

- Consolidated Flow Index documentation into `SCORING_BREAKDOWN.md`
- Merged `ANALYTICS.md` , `DATA_COLLECTION.md` , and `VIEW_PROFILE_GUIDE.md` into `ANALYTICS_AND_DATA.md`
- Removed redundant documentation files
- Updated main `README.md` to link to the current `docs/` structure
- Updated main `README.md` with new documentation structure

Bug Fixes

- **Flow Index Interpretation**: Fixed `getFlowInterpretation()` function to correctly label Flow Index 0.2-0.4 as "Too Hard" instead of "Too Easy"
- Updated Flow Index interpretation logic to match actual performance metrics (low Flow Index = poor performance = game too hard)

Technical Improvements

- Improved documentation organization and structure
- Enhanced Flow Index guide with debugging information
- Consolidated analytics and data collection documentation

Version: v2.0.0 (2025-12-16)

Major Updates

- **Analytics System**: Comprehensive game analytics with Flow Index, performance metrics, error analysis, and color confusion tracking
- **Game History**: IndexedDB-based game history storage for viewing past game sessions
- **Unified Analytics Page**: Standalone `analytics.html` page supporting both history view and demo mode
- **Documentation**: Complete English documentation organized in `docs/` folder
- **Flow Index Fix**: Fixed Flow Index calculation to use current game session data instead of cached values

Analytics Features

- Real-time Flow Index calculation based on game performance
- Performance overview (time, clicks, matches, accuracy)
- Error analysis (consecutive errors, error rate)
- Color confusion analysis (occurrence and accuracy per color)
- Behavioral patterns (flip intervals, cadence stability, hint usage)
- Adaptive difficulty suggestions
- Game configuration display

Game History

- Automatic saving of game sessions to IndexedDB
- View all historical games sorted by timestamp
- Filter by level
- View detailed analytics for any past game
- Quick summary cards showing key metrics

UI/UX Improvements

- Integrated analytics display in game-over screen
- Centered layout matching standalone analytics page
- Removed duplicate scrollbars

- Improved font sizing (24px titles, 16px content)
- Hidden glass effect on game-over screen
- Fixed game-over page overflow issues

Technical Improvements

- Fixed Flow Index retrieval to prioritize `aiResult.flowIndex` over telemetry events
- Enhanced data extraction to filter events by current game session
- Added comprehensive debug logging for Flow Index calculation
- Improved error handling and data validation
- Fixed variable redeclaration issues

Documentation

- Created `docs/` folder with organized documentation
- Merged duplicate documentation files
- Translated all Chinese content to English
- Expanded troubleshooting guidance across analytics and scoring documentation

---

Version: v1.4.2 (2025-12-10)

Overview

- Enable AI engine on Level 3 page to provide adaptive difficulty suggestions.
- Fix potential lock when cards are flipped without `data-match` on Level 3.
- Standardize Level 3 start telemetry with variant payload (countdown, hide delay, show scale, total pairs).
- Clean up debug output and remove unnecessary `console.log` s.
- Minor consistency updates: replace loose equality with strict equality in timers.

Detailed Changes

- Enable AI engine scripts
  - `lvl-3.html:65-68` : load `js/ai-engine.js` and `js/ai-helper.js` so `AIEngine` is available.
- Level 3 flip lock fix
  - `js/lvl3.js:200-208` : when any card lacks `data-match` , hide both cards, reset `flippedCards` and `lockBoard` instead of returning early.
- Level 3 start telemetry enhancement
  - `js/lvl3.js:167` : add `variant` with `pairsType` , `hideDelay` , `showScale` , `timerMode` , `initialTime` , `totalPairs` .
- Level 3 debug cleanup
  - Remove temporary debug logs in the match flow.
- Level 3 text/image normalization
  - `js/lvl3.js:331` : normalize by removing all spaces via regex for consistent matching.
- Level 1 improvements
  - `js/lvl1.js:77` : change `gameStart == 1` to `gameStart === 1` .
  - Remove a redundant debug print.
- Level 2 improvements
  - `js/lvl2.js:140` : change `gameStart == 1` to `gameStart === 1` .

Behavior Changes

- Level 3: first flip telemetry includes complete variant info for better AI assessment and next-step suggestions.
- Level 3: UI no longer locks on malformed card data; flip state resets safely.
- Level 3: AI engine scripts are loaded; game end can produce cross-level config suggestions written to `localStorage` .

Performance & Reliability

- Fewer lock risk paths on the flip interaction.
- Less debug noise in production, improving performance and privacy.

Risks & Rollback

- Risk: strict equality may affect legacy logic if it depended on implicit type coercion.
- Rollback: revert the equality checks and removed logs if necessary.

Verification

- Local preview `http://localhost:8010/lvl-3.html` and complete one round:
  - Check `start` telemetry includes `variant` fields.
  - Confirm `localStorage` writes `ai_level3_config`.
  - Simulate malformed card data (edit `dataset.match` in devtools) and verify no lock.
- Level 1/2: timer, flipping and scoring remain consistent.

Follow-ups

- Extract common logic across Level 1/2/3 (timer, show/hide all, scoring) into shared modules.
- Add static checks (ESLint/Prettier) and enforce via CI.
- Consider preloading or lazy-loading images to improve first render.