Campaign Mode Spell Arena

High-Level Overview (Slide Breakdown)

Slide 1: What Is Being Added

The campaign mode is a major expansion to the Hand Gesture Wizard game. It transforms the project from a single-scenario demonstration into a structured, multi-stage experience with clear goals and progression.

Slide 2: Why It Is Important for Users

For players, the campaign introduces long-term objectives, increasing challenge, and a clear sense of advancement. Rather than repeating the same encounter, users progress through new stages, enemies, and mechanics.

Slide 3: Learning and Skill Progression

Progressive unlocking of spells and encounters helps manage cognitive load. Players are encouraged to master core hand gestures and mechanics before being introduced to additional complexity, supporting learning and retention.

Slide 4: Technical and Project Importance

From a technical standpoint, the campaign establishes foundational systems such as progress tracking, unlock logic, scalable AI difficulty, and map-based navigation. These systems enable future extensibility and structured content growth.

Slide 5: Assessment and Validation Value

The campaign mode also enables measurable development progress. Clearly defined features, requirements, and test cases support validation, burndown tracking, and alignment with capstone assessment criteria.

This section provides a technical breakdown of the planned campaign mode for the Hand Gesture Wizard game. It decomposes the system into features, formal requirements, and corresponding test strategies to ensure correctness, stability, and measurable progress.

Feature 1: Campaign Progression System

Description

Provides structured progression through the game, transitioning from a single demonstration scenario into a multi-stage campaign with increasing difficulty and content unlocks.

Requirements

- R1.1: The system shall track player campaign progress persistently across sessions.
- R1.2: The system shall define discrete campaign stages (nodes or levels).

- R1.3: Completion of a stage shall unlock the next available stage.
- R1.4: Campaign progress data shall be stored using Unity’s save system or equivalent persistent storage.

Testing

- T1.1: Verify campaign progress is saved and restored after restarting the application.
- T1.2: Complete a stage and confirm the next stage becomes accessible.
- T1.3: Attempt to access locked stages and verify access is denied.

Feature 2: Spell Progression and Unlocking

Description

Introduces a learning curve by progressively unlocking spells as the player advances through the campaign.

Requirements

- R2.1: The system shall initialize new campaigns with a restricted spell set.
- R2.2: Spells shall be unlockable based on campaign milestones.

- R2.3: The player shall only be able to equip spells that have been unlocked.
- R2.4: Unlock conditions shall be data-driven to allow easy tuning.

Testing

- T2.1: Start a new campaign and verify only starter spells are available.
- T2.2: Complete a milestone and verify the correct spell is unlocked.
- T2.3: Attempt to equip a locked spell and confirm the action is blocked.

Feature 3: World Map Navigation

Description

Implements a map-based interface where players select and travel between encounters.

Requirements

- R3.1: The system shall present a navigable campaign map UI.
- R3.2: Each map node shall correspond to a combat encounter or event.

- R3.3: Locked nodes shall be visually distinguishable from unlocked nodes.
- R3.4: Selecting an unlocked node shall transition the player into the associated encounter.

Testing

- T3.1: Verify all map nodes correctly represent their campaign state (locked/unlocked).
- T3.2: Select an unlocked node and confirm correct scene loading.
- T3.3: Attempt to select a locked node and verify no transition occurs.

Feature 4: AI Opponent Scaling

Description

Expands beyond a single static CPU opponent by introducing difficulty scaling across the campaign.

Requirements

- R4.1: The system shall support multiple AI difficulty profiles.
- R4.2: AI difficulty shall increase as campaign progression advances.
- R4.3: AI behavior parameters shall be configurable without code changes.

Testing

- T4.1: Compare AI behavior across early and late campaign encounters.
- T4.2: Validate difficulty parameters update correctly when loading different stages.

Feature 5: Tutorial Integration

Description

Ensures the existing tutorial integrates cleanly into the campaign flow as the entry point.

Requirements

- R5.1: The tutorial shall be the first required campaign stage.
- R5.2: Completion of the tutorial shall unlock the first combat node.
- R5.3: Tutorial completion state shall be recorded in campaign progress data.

Testing

- T5.1: Complete the tutorial and verify campaign progression updates.
- T5.2: Restart the game and confirm the tutorial is not replayed unless manually

Feature 6: Metrics and Validation Support

Description

Provides instrumentation to support progress tracking and burndown metrics for project evaluation.

Requirements

- R6.1: The system shall expose counts of implemented features and requirements.
- R6.2: Unimplemented features shall be clearly identifiable.
- R6.3: Feature and requirement completion data shall be manually verifiable via documentation or logs.

Testing

- T6.1: Cross-check documented requirements against implemented systems.

Current Implementation Summary

- Implemented Features: Tutorial system, core gesture recognition, basic combat loop
- Pending Features: Campaign progression, spell unlocking, map navigation, AI scaling
- Feature Burndown Rate: 0 (campaign features not yet implemented)
- Total Identified Requirements: Defined per feature above and subject to iteration