Sprint 2 End Document

Date: 12 Oct 2025

Sprint Changelog

- Godot project now launches into a fully themed start menu, story scene, pause menu, settings hub, AI settings overlay, prayer interface, journal, and achievement viewer wired to shared style helpers.
- Added GameState autoload to track Reality/Positive Energy/Entropy, log events, manage autosaves/save slots, and broadcast stat changes to UI and AI prompts.
- Implemented AIManager with Gemini and OpenRouter HTTP integrations, timeout/retry handling, memory trimming, token and latency metrics, and a MockAIGenerator fallback.
- Built an AI Settings UI that allows provider selection, API key entry, custom tone, memory budgets, and live connection tests with real-time metrics charting.
- Scripted prayer disaster generator, achievement tracking, notifications, audio pools, markdown parsing tests, and asset card rendering to stress-test domain requirements.

Sprint Review

Are we at a point where we expected to be at the beginning of this sprint?

The sprint targeted a playable UI shell and an AI skeleton, both of which are now in place with mock data.

Did the implementations of features go smoothly, or was there trouble?

Most systems came together smoothly; the main challenge was taming long-form scripts into reusable managers so the story scene stayed readable.

Did feature implementations match our design documents?

Yes. The Story Scene follows the proposed multi-layer context flow, and the AI Settings screen reflects the configuration hooks described in the Week 3 proposal.

What went well this sprint?

- Shared managers (GameState, AIManager, AudioManager) proved reusable across scenes.
- Mock AI allowed end-to-end flow testing without blocking on live credentials.
- Supervisor feedback cycle on the proposal was closed quickly with a 09 Oct revision.

What should we continue doing to make sure things keep going well?

- Continue structuring UI scripts around helper managers and signal-driven updates.
- Keep the mock AI path up to date so we can test safely when keys are unavailable.

What didn’t go well this sprint?

- Large scripts (story_scene.gd, ai_manager.gd) need refactoring before they become harder to maintain.

What could we do better next sprint?

- Allocate explicit time for gameplay scaffolding before adding more UI overlays.
- Start breaking AI logic into smaller strategy classes so different prompt shapes are easier to test.

Organisation & Planning

What new plans or organisation changes have occurred during this sprint?
Have any roles been shuffled around?
Have planning documents like Class diagrams changed? If so, how do they look now?

- Stored both the annotated 07 Oct proposal and the revised 09 Oct submission for supervisor review, confirming the Week 3 milestone plan.
- Added sprint documentation folders so start/end notes can be tracked alongside gameplay updates.

Other Notes

- *Next sprint (Week 3) will concentrate on literature review and interim report drafting, enabling live Gemini/OpenRouter calls with real keys (with clear error surfaces), and capturing latency/token metrics for interim report evidence.*