SPRINT DOCUMENTATION — Week 1 (Planning)

1) Summary data	
Sprint start date	29 Sep 2025
Sprint end date	05 Oct 2025

2) User stories/task cards

- Sprint goal: Establish a validated project scope, requirements baseline, and risk register ahead of technical work.
- [Completed] refined a single aim and four measurable objectives so the supervisor and I have a shared scope for later sprints.
- [Completed] drafted the Week 3 proposal sections covering requirements, timeline, and risks so reviewers can sign off on the plan early.
- [Completed] documented toolchain, API provider, and security approaches so the Week 2 setup tasks can start without blockers.

3) Requirements analysis

- *Functional (shall):*
  - *Provide a foundational Godot 4.5 2D RPG framework with player controller, dynamic UI, and symbolic inventory to host AI-generated content.*
  - *Implement a request/response pipeline that maps game state → structured prompt → LLM reply → in-game events/NPC dialogue, starting with Gemini and allowing OpenRouter selection.*
  - *Maintain a multi-layer context manager (static lore, short-term scene transcript, long-term summaries) to minimise narrative drift.*
- *Functional (should):*
  - *Deliver seeded lore about the Void Entropy vs Positive Energy conflict to anchor AI outputs and guide narrative tone.*
  - *Reuse tagged symbolic assets whose meaning is re-labelled by the AI so text creativity stays playable within a finite art budget.*
- *Non-functional:*
  - *Keep API keys out of version control by collecting them via the settings screen, and avoid transmitting personal data; communicate data use and rely on HTTPS requests.*
  - *Optimise prompts and caching for low latency and controlled token spend while keeping the system controllable for long-form stories.*
  - *Focus on depth over breadth so the prototype supports the 80% weighting on critical analysis and reporting.*
- *Domain requirements:*
  - *Track the Reality vs Positive Energy stat pair and the Prayer System scenario to evaluate thematic coherence in controlled story beats.*
  - *Persist emergent facts in a notes register so future prompts respect established lore and player choices.*

4) Design

- *Runtime dataflow: capture scene state, produce structured prompts, render LLM replies immediately, and roll updates into the notes register with periodic summaries to stay within token limits.*
- *Multi-layer context: world bible for immutable rules, rolling transcript for the current scene, and long-term summaries for arc-level callbacks.*
- *Symbolic asset binding: generic props gain AI-authored names, behaviours, and lore on the fly to bridge text output and finite art.*
- *Systemic values: the Reality vs Positive Energy stat loop feeds back into prompts so tonal shifts and decay patterns remain thematically consistent.*
- *Player configuration: a settings page lets players provide Gemini/OpenRouter API keys and switch providers at runtime.*

5) Test plan and evidence of testing

- *Planned evaluation metrics include narrative consistency annotations, player immersion feedback, latency and cost tracking, plus ablations comparing note-keeping and provider choices.*
- *No automated or system tests have been executed this sprint because implementation work is scheduled to begin in Week 2.*

6) Summary of the sprint

- *Achieved: Completed background research, proposal drafting, and supervisor consultation, identified as the Week 1 milestone; captured the phased plan, risks, and evaluation strategy in the proposal.*
- *Stakeholder feedback: Supervisor reinforced the "depth over breadth" focus and agreed the project should centre on one core aim with four objectives, emphasising report quality.*
- *Prototype status: None yet (planning sprint). The technical setup (Godot project, Git repository, baseline controller) is scheduled for Week 2.*
- *Risks monitored: API quota/access limits, narrative drift, asset mismatch, data loss, and scope creep are logged with mitigations; no changes triggered this week.*
- *Next sprint focus: Stand up the repository and Godot scaffold, implement baseline movement/UI features, and execute the first Gemini API connectivity tests.*