Project Documentation

Project: FunFlip Game

Authors:

Ashish Ghaskata  
Krishna Raj Bhandri

Suraj Bhatta

Mohammad Adnan Khan

**Document name:** AB-Funflip Game-ProjectDoc

**Creation date:** 20.06.2025

Table of Contents

[Project Documentation 3](#_Toc201689903)

[Project Approach 3](#_Toc201689904)

[Lessons Learned 4](#_Toc201689905)

# Project Approach

The **FunFlip Game Project** was developed using a structured, iterative approach, integrating best practices from software engineering disciplines (as outlined in lectures and applied in practice):

1. **Requirements Engineering**
   * Defined clear vision, SMART goals, personas, and user stories to align the product with educational objectives.
   * Documented functional (e.g., card flipping logic, category and level selection) and non-functional requirements (e.g., ≤0.5s response time, offline operation).
2. **System Architecture & Design**
   * Applied a strict 5-layer architecture (UI, SceneLoader, Game Logic, Data, Services) for modularity, low coupling, and high cohesion.
   * Prioritized usability, accessibility, and performance in design decisions, supported by diagrams (e.g., domain model, activity diagrams).
3. **Implementation**
   * Developed using the Godot Engine with reusable components (e.g., Card.tscn, AudioControl.gd) and data-driven design (e.g., categories.json).
   * Applied agile-inspired iterations focusing on completing functional increments (e.g., category selection, card matching).
4. **Quality Assurance & Testing**
   * Created detailed test specifications and protocols covering functional and non-functional requirements.
   * Performed system, integration, and acceptance tests on target platforms (Android/iOS) under realistic conditions.
   * Executed both positive and negative test cases; no critical defects remained at acceptance.
5. **Documentation & Handover**
   * Produced comprehensive documentation, including user manuals, architecture descriptions, and acceptance reports.
   * Validated system usability and accessibility for the target audience (children aged 4–6).
6. **Process Alignment**
   * Reflected a tailored V-Model/W-Model process with early QA integration and parallel preparation of test activities.
   * Aligned with software engineering standards from lecture guidelines (e.g., requirements traceability, design principles, validation focus).

# Lessons Learned

Through this project, the team gained significant insights into software engineering practices:

1. **Importance of Early and Clear Requirements**
   * Investing time in detailed requirements (functional and non-functional) reduced ambiguity during design and coding phases.
2. **Value of Modular Architecture**
   * A strict layered approach and component reuse (e.g., Card.tscn) simplified future updates and ensured maintainability.
3. **Iterative Testing Prevents Late Surprises**
   * Systematic, phased testing (unit → integration → system → acceptance) allowed early detection and resolution of issues.
4. **User-Centered Design is Crucial**
   * Designing for the end-user (young children) required thoughtful consideration of usability and accessibility, beyond basic functional correctness.
5. **Documentation is Not Just a Deliverable**
   * Keeping documentation up to date alongside development (architecture, requirements, testing) saved time and improved handover quality.
6. **Team Communication and Coordination**
   * Clear communication and role separation (design, coding, testing) were vital for staying aligned and efficient.
7. **Tool and Process Familiarity**
   * Mastering tools (e.g., Godot, draw.io) and following structured processes (e.g., V-Model principles) significantly contributed to project success.