Final Project Postmortem

Tapesh Sankaran

University of California - Santa Cruz

CMPM 121 - Game Development Patterns

Professor Zac Emerzian

12 June, 2025

General Remarks

This tarot-themed card game project was a rewarding experience, offering both creative flexibility and technical challenge. Compared to past projects, especially one that was based on Greek mythology, this project demanded a thematic overhaul, requiring me to change the cards, source or create new icons, adjust the abilities to better fit the tarot archetypes, and redesign the background to reflect the new aesthetic. It was a satisfying shift from my other class project, which focuses heavily on procedural generation. That project involved a great deal of hyperparameter tuning and manual editing, whereas this one allowed for more focused iteration on visual effects, card mechanics, and game feel.

The development process was mostly smooth, although a few aspects did slow things down. One of the biggest triumphs was implementing the card system using a component-based design, allowing each card to have modular abilities and triggers. Combined with an event queue system, I could control animations and sound effects in a reliable and consistent order. These design patterns brought clarity and scalability to what could've easily become a tangled mess of conditions and triggers.

Challenges

That being said, I encountered several frustrating issues during development. A particularly persistent problem was with sound effects: audio clips would sometimes cut each other off or fail to play entirely due to overlap or mismanaged playback logic. This was especially noticeable during rapid actions like card placement or simultaneous effects. Another significant pain point was testing individual cards. Without a proper debug mode or in-game editor, I had to manually load and trigger each card's ability, which was a slow and error-prone process that made balancing and bug fixing harder than it needed to be.

Animations also proved to be a recurring challenge. While the system supported card flipping, particle effects, and screen shakes, getting these to feel natural required many small tweaks. Some transitions still feel abrupt or visually stiff, suggesting the need for more polish or animation smoothing. The overall game state management also remains a bit crude, occasionally causing unexpected behavior when transitioning between phases or reloading cards.

Concluding Statements

Despite these obstacles, I would absolutely work on a game like this again. In fact, I'd be interested in expanding and refining this exact project, especially with more attention to user experience and debugging tools. Building this game helped me solidify my understanding of programming patterns like the component model, prototype instancing, and event-driven design, and made it clear how valuable these patterns are in organizing gameplay logic and making systems reusable.

One memorable bug that emerged during testing was when a card that copied another card's effect recursively triggered itself when discarded. This resulted in a screen flooded with ghost copies of the same card, eventually overwhelming the system and crashing the game, a surprisingly funny moment that reminded me of the importance of limiting effect chains and protecting against recursive triggers.

All in all, this project was a deeply engaging and creatively satisfying experience. It sharpened both my coding skills and my design intuition, and I'm excited to either take it further or apply what I've learned to future game development projects.