

UNO Flip Remix - Development Plan

Team 24

Zain-Alabedeen Garada

Kevin Ishak

Mingyang Xu

Jianhao Wei

Zheng Bang Liang

September 24, 2024

Development Plan

Our team will hold weekly team meetings to track progress and align on project tasks. Each team member is expected to attend these meetings. Additionally, we will organize informal work sessions throughout the week to collaborate on specific tasks and document our progress. The link to our GitHub: [Github Link](#)

Team Meeting Plan

Scrum Meetings: Weekly Scrum meetings will be conducted with all team members attending. Each meeting will last between 15 and 30 minutes and will occur on Microsoft Teams. Members will discuss progress and any blockers encountered. At least one additional work session per week will be held where coding and design tasks will be conducted collaboratively.

Team Communication Plan

Primary communication will occur on Microsoft Teams, supplemented by cell phone coordination when necessary. The team will also utilize GitHub for version control and code reviews.

Team Member Roles

- **Kevin Ishak:** Full-Stack Developer, Will do research and implement the different types of cards that are possible to add in the game to make the game challenging whilst keeping the core essence of the card game intact.

- **Zain-Alabedeen Garada:** Developer/Project Manager, Will do research and implement the server-side development and real-time game synchronization.
- **Mingyang Xu:** Full Stack Developer, Will do research and implement 3D visualizations and user interface of the game.
- **Jianhao Wei:** Full Stack Developer, Oversees back-end implementation.
- **Zheng Bang Liang:** Full Stack Developer, Oversees front-end design and implement 3D visualizations.

Workflow Plan

The project will follow a Git-flow branching model to streamline development and ensure effective collaboration among team members. In this model, each new feature or bug fix will be developed in its own dedicated branch. This approach allows for focused work on individual components without disrupting the main codebase.

Once a feature is completed, team members will submit a pull request to merge their changes into the master branch. Each pull request will undergo a review process where other team members can provide feedback, suggest improvements, and identify potential issues before integration. This collaborative review process enhances code quality and promotes knowledge sharing within the team.

In addition to code reviews, we will implement unit testing to ensure the reliability and functionality of our code. Each module or feature will include a set of unit tests that validate its behavior and performance against specified requirements. These tests will be written alongside the code and run regularly to catch any regressions early in the development process. By emphasizing unit testing, we aim to maintain high standards of code quality and reduce the likelihood of bugs in the final product.

Proof of Concept (POC) Demonstration Plan

The success of the UNO Flip Remix project hinges on effectively demonstrating multiplayer synchronization and basic AI functionality within a simplified framework. The inputs will consist of player actions and game state updates, which will be managed through basic networking protocols. The processing will involve real-time updates to the game state and AI decision-making, while the outputs will include visual feedback on player moves and AI responses.

To achieve this, the Proof of Concept aims to create a working prototype that showcases these interactions clearly. The main risk to the success of this POC is the potential challenges of network latency affecting the multiplayer experience and the limitations of the AI's decision-making capabilities within a constrained timeframe.

Technology Stack

- **JavaScript and Node.js:** Core programming languages for both front-end and back-end development.

- **TensorFlow:** For implementing reinforcement learning algorithms for AI behavior.
- **Socket.IO:** To handle real-time communication for multiplayer games.
- **GitHub Actions:** For CI/CD integration to streamline testing and deployment.

Risks and Mitigation

AI Complexity: The complexity of AI learning may exceed expectations. To mitigate this, a simpler rule-based AI will be developed first.

Network Latency: Real-time synchronization may suffer from latency. Rollback networking techniques will be explored to manage this risk.

Coding Standards

- The code will follow JavaScript ES6 standards for front-end and back-end development.
- Pylint will be used to ensure Python code (for AI) adheres to PEP8.
- Testing will be implemented using Mocha (for JavaScript) and Pytest (for Python-based AI modules).

Project Scheduling

A Gantt chart has been created. Here is the link: [Gantt Chart](#)