UNO Flip Remix - Development Plan

Team 24
Zain-Alabedeen Garada
Kevin Ishak
Mingyang Xu
Jianhao Wei
~~Zheng Bang Liang~~

April 2, 2025

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~~

GitHub repository: https://github.com/simon-0215/UNO-Flip-3D

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, issue tracking, and code reviews.

Team Member Roles

- **Kevin Ishak**: Full-Stack Developer. Responsible for implementing card behavior logic and assisting with AI player development.

- **Zain-Alabedeen Garada**: Developer/Project Manager. Oversees project scope, manages GitHub, and leads server synchronization and multiplayer logic.
- **Mingyang Xu**: Full-Stack Developer. Leads development of the game's UI/UX and visual feedback.
- **Jianhao Wei**: Full-Stack Developer. Assists with backend logic and Mirror-based networking integration.
- ~~**Zheng Bang Liang**: Full-Stack Developer. Focuses on frontend gameplay features and user interaction components.~~

Workflow Plan

The project will follow a Git-flow branching model. Each new feature or bug fix will be developed in its own dedicated branch. When complete, a pull request will be submitted for team review before being merged into the main branch. Regular code reviews will promote team collaboration, ensure quality, and support knowledge sharing.

To maintain code reliability, unit tests will be written for each module or feature using the NUnit framework. Tests will validate behavior and catch regressions early. GitHub Actions will be configured to automate test runs on new pull requests.

Proof of Concept (POC) Demonstration Plan

The POC will demonstrate basic multiplayer gameplay and rule-based AI in Unity. Mirror will be used for real-time communication between clients. The AI will simulate strategic choices in a simplified version of Uno Flip. Players should be able to take turns, draw, play, and end the game successfully.

The main risks are multiplayer desyncs and ensuring the AI behaves reasonably with hidden card information. The POC will be considered successful if two clients can connect and play through a short, valid game session.

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.~~
- **Unity with C#**: Core platform for game development, including gameplay logic and UI.
- **TCP**: Networking library used for multiplayer client-server communication.
- **NUnit**: Unit testing framework used with Unity for validating code behavior.
- **GitHub Actions**: Used for automated testing and CI workflows.

Risks and Mitigation

AI Complexity: AI must operate with limited knowledge and make reasonable decisions. Mitigation: use rule-based AI with deterministic strategies.
Network Latency: Mirror-based multiplayer may encounter de-synchronization. Mitigation: implement state reconciliation and test with multiple clients in local and online conditions.
UI Responsiveness: Ensuring smooth UX under varying network conditions may be challenging. Mitigation: separate UI updates from backend processing and perform regular usability tests.

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).~~
- **C# conventions**: Follow Microsoft's C# style guidelines and naming conventions for Unity scripts.
- **Unit Testing**: NUnit will be used to validate functionality during development. All features will be accompanied by corresponding test cases.

Project Scheduling

A Gantt chart has been created. Here is the link: Gantt Chart