

Project Structure Document

Part 4 - Alpha Release

Computer Games Laboratory

Summer Term 2021



Meeple People

Anastasia Pomelova

Eugene Ghanizadeh Khoub

Mert Ülker

Shyam Rangarajan

Contents

- 1. Status Overview 2
- 2. Game Logic and UI 2
 - 2.1. Game Logic 3
- 3. Challenges 4
- 4. Next Steps 4

This report will discuss our alpha release for Flee Fi Fo From, and an update of our status at this stage. We will then cover the challenges we encountered as well as our next steps.

1. Status Overview

As discussed in the interim report, we split the project into four subsections to facilitate independent development, and have now integrated a majority of the components. At a high level, the game logic and UI is complete with all components in place. The game is now fully playable on a single device. The networking branch will also be integrated shortly to facilitate play over the internet.

2. Game Logic and UI

This layer consists of the visual components needed to play the game (perform actions part). In its current state, the UI allows for a player to interact with all of the pieces on the game board, as well as the UI elements including actions, resets and priority tracker. The riot command is now fully implemented, along with undo functionality for all sub commands. The worker loop system is also fully implemented now, allowing for worker depiction per action space as well as the removal of workers via the reset commands. All objective related actions are implemented as placeholder methods, until we start working on the objective mechanic in our higher goal tasks. These facets will be toggled off during our alpha release and playtesting sessions. The honor tracker above the field is depicted in our desired UI style, along with a scoreboard to track victory progress across players. An illustration is shown below in Figure 1.

The UI leverages the use of Button Manager and Field Manager classes to depict the player interactable actions and the game board respectively. These classes have been significantly reworked and will continue to be refactored to create as much decoupling as possible between visuals and logic in the game.

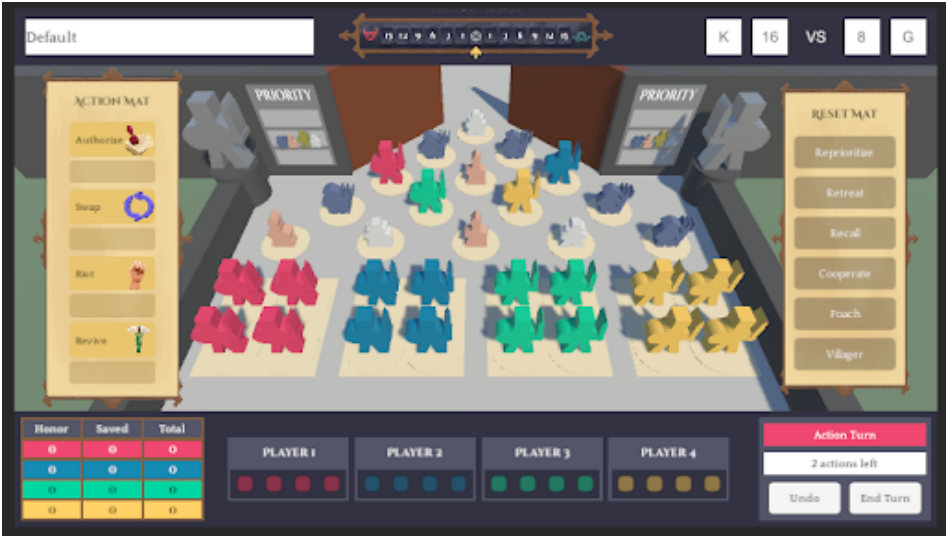


Figure 1: User Interface

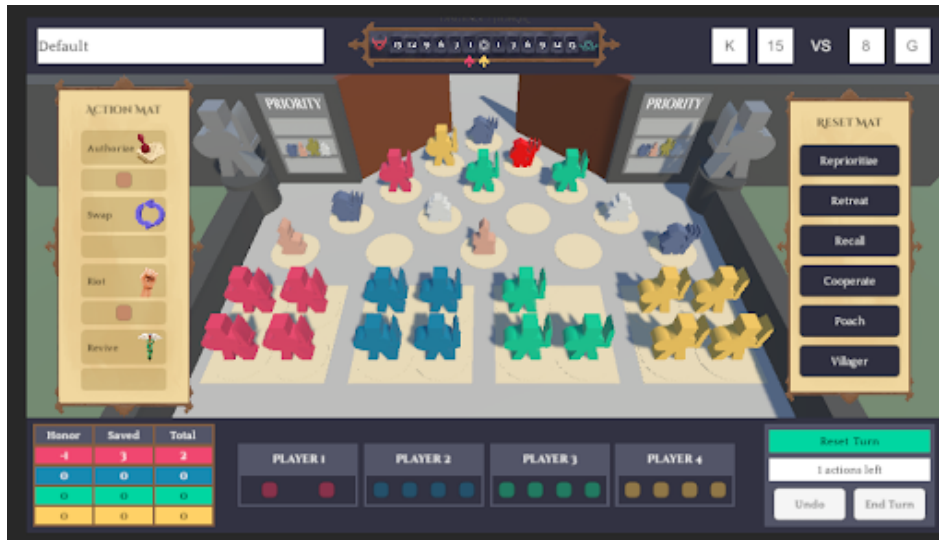


Figure 2: Mid-Game State

2.1. Game Logic

The Observable interface of the Game State has been extended to encompass as many of the game components as possible, allowing the visual layer to listen for changes to particular values and reflect them efficiently on the board (e.g. position of a villager or whether they are injured or not).

The Game Manager now acts as a central bus between various other modules: the visual elements communicate interactions and their feedback to the Game Manager, Action checkers communicate with the Game Manager to retrieve a list of possible interactions at current state, and passes logical commands to the command processor which in turn results in updates in the Game State and will be synced via network.

Rule Changes:

In light of playtesting feedback, a modified FIFO system using an externally tracked priority system across each piece type has been implemented. The UI for this is already available, while the logical connections are in progress.

Networking:

The networking capability of the game has been enhanced from LAN to internet connection, in order to reach a wider user base. This has been performed by means of an AWS Linux instance deployed with a relay server. The board initialization (random) has been synchronized across players, such that each local client works with the same original seed and sees the identical game state. The logic for the turn control (current action and reset player) has also been completed. The synchronization of the actions across each local instance will be implemented in the next phase.

3. Challenges

Our encountered challenges are a continuation of those identified during the interim phase. The rules-heavy nature of the game has been greatly simplified, but might still have potential to reduce the cognitive load on the player further. A by-product of these myriad rules and action combinations is that there is the potential for edge cases that are contrary to what the game logic would expect. Our approach to this is to playtest internally and externally as much as possible, in order to identify and address as many bugs as possible. This process has already commenced. Lastly, in order to keep the stress on the network as low as possible, we are regularly reviewing our code base to identify any refactoring opportunities. Some of the earlier challenges around undo functionality and command stack have already been addressed with our current architecture, but we will continue to test them for bugs, as listed above.

4. Next Steps

In terms of next steps, we will be working on preparing the game for playtesting, as well as further refining the current version.

Game UI and Logic:

For the game UI, we will be focusing on visual improvements to the current interface, including piece models and animations. We also plan to implement a simplified version of the inner castle, which contains all non-objective related castle spots. We want to round up the player experience via helpful tooltips or messages on the UI, which would provide guidance on the actions that a player could take on their turn.

From a game logic standpoint, the integration with the existing priority UI is in the works, such that the various actions and resets earn the corresponding honor effects based on priority. We are also interested in monitoring and refactoring the code base of the game to make sure that the game components are decoupled as much as possible, as well as to remove any existing duplication of code. We will also be verifying that the modified rules work well with the game, while continuing to explore further opportunities to clean up and simplify any rules that are cumbersome or unclear.

Networking:

For networking, we will be working on synchronizing the player actions across the network, so that all clients have the same local copy of the actions. We are also aiming to implement fallback methods to handle potential server connection errors. These steps will facilitate the playtesting of our game across the Internet, as we move into the next phase of the project.