Lima planning document

GUI Planning Review:

Timing issues, Missed milestones, or Unrealistic deadlines:

-Everything seems doable for iteration 2 however designing the first working version of GUI could take more time than expected, since there a lot more things planned for I2, so again, it's possible but it would take a lot of time or people.

-All requirements for I3 are mentioned and their timing seems reasonable, but it could be a good idea to add some buffer time (hurry things up) in case that integration issues are more complex than expected.

Risks that could impact completion:

- GUI depends highly on Networking and Game-Logic teams, to mitigate this problem be sure to have constant communication so that issues can be identified early.

- The scope of GUI is large, there are many things that must be completed by this team, to mitigate the problem ask teams that have nothing going on for some help and spread the most important tasks among your own members.

- GUI Development Plan Enhancements could prove to be a challenge such as, UI components that can be reused across different game interfaces, and ensuring the UI adapts well to different screen sizes, including mobile devices. Which are really good ideas, but I don't think it should be a focus of the project.

Suggestions for improving the timeline or reordering tasks to reduce risks:

- One suggestion we have is to start tests early that way issues can be found faster.

- Another would be to focus on the main components first, such as making sure that games work, or that they are easy to understand/play… and then if there is time, then the enchantments can be done. Which is what your plan seems to be, but we can't tell for sure.

- And one more suggestion we have is for each team member to focus on one thing, for example one member focuses on Game Dashboard, another on Moving pieces… that way no one will be overwhelmed with responsibilities.

Networking Planning Review:

-Overview: The networking_planning.md file contains mostly a list of different tasks and each of their deadlines.

Timing issues, Missed milestones, or Unrealistic deadlines

-The team needs to complete the task of Implementing Profile Search/Leaderboard/Game Joining by March 26 and needs to complete the Implementing Messaging task by March 28, which is why they are only given two days to complete the latter task if they finish the former on March 26. Implementing this task is a tough as completing multiple different components (profile search, leaderboard and game joining), each with their own complications needs more than two days to complete all of them.

-The plan does not mention any of the tasks for iteration 1 of the project that requires designing and planning, both of which have a deadline of March 7. The tasks shown are only appropriate for iterations 2 and 3 of the project.

-The deadline for Final Integrations task has been misspelled to be March 11 instead of April 11. Integrations are part of iteration 3, and all need to be due earliest on April 11.

-I have noticed that the deadline for Final Integrations is earlier than the other tasks. This is a problem as the last integrations for the network system should happen later in development to account for any lasting or remaining issues. The other tasks need to be properly finished and implemented first, so that these integrations don't have to be fixed repeatedly after each task is done.

Risks that impact completion:

-The task Implement profile search/leaderboard/game joining is convoluted as it requires the completion of several different implementations in a short period. Implementing profile searching, the leaderboard or game joining may take each at least a few days to and would be challenging to finish in the 5 days it has since the Implement game records/tracking task due on March 21.

-The plan contains too little information and does not go into detail of the full development plan. There are no tasks involved in testing the network for iteration 3, neither is there any design for any networks. Each of the tasks are only given a name and aren't further explained their purpose or method of implementation.

Suggestions for improving the timeline or reordering tasks to reduce risks:

-The deadline for Final Integrations needs to be changed to April 11th, as this task is part of Project Iteration 3 which is due on this date. Give more time for the team to finish the task of Implement Messaging as when time is tight, they are given a chance to properly commit to this task.

-Another suggestion is to explain each task more such as the purpose, motivation, design and method of implementation. This would make it clear to the whole team how to create a functioning and well-developed network system.

Game-Logic Planning Review:

Timing issues, Missed milestones, or Unrealistic deadlines:

Overall Timing and Deadlines:

-The game logic team's planning document provides a solid outline of their tasks, aligning with the needs of other teams (GUI, Networking, etc.). However, the deadlines provided for key milestones, such as "Get close to game completion" and "Start coding game logic," are too vague. These broad goals need more specific micro deadlines to create a clearer sense of responsibility and to ensure the work progresses efficiently.

-Start Coding on March 21st: The deadline for beginning coding on March 21st is considered too late. Since the game logic team is crucial for developing the core mechanics, which other teams depend on (e.g., GUI), starting later risks delaying the entire project. The coding phase should ideally start earlier, perhaps on March 7th, to allow sufficient time for completion before other teams need the game logic.

Missed Milestone: The plan lacks specificity about when certain tasks, like coding specific games, will be started and completed. The document would benefit from breaking down the work into smaller tasks with clear, individual deadlines.

Risks that could impact completion:

-Lack of Specific Deadlines for Tasks: Broad statements like "Get close to game completion" do not outline concrete steps, potentially causing confusion and lack of accountability. These vague goals also do not promote a sense of responsibility within the team, which can lead to delays or incomplete tasks.

-Delays in Game Logic Could Affect Other Teams: The game logic team's late start could cause delays for the GUI and Networking teams, as their work depends on the game logic being implemented first. If the game logic team underestimates the time required for their tasks, it may negatively impact the integration process for the other teams.

Suggestions for improving the timeline or reordering tasks to reduce risks:

-More Specific Deadlines: The game logic plan should break down broad milestones into more specific tasks with individual deadlines. For example, instead of "Get close to game completion" by March 21st, they should list detailed steps for the games they plan to implement and assign realistic dates to each task.

-Start Coding Sooner: To avoid delays, coding should begin earlier, ideally by the March 7th deadline. This allows more time for game mechanics to be developed and gives other teams a chance to integrate their components more efficiently.

-Begin Testing in Phase 2: Testing should be integrated throughout Phase 2, not just at the end of development. Implementing unit tests during the coding phase helps catch issues early, ensuring

smoother integration with the other teams' work. This will also help the team identify and fix bugs early on, improving the overall project timeline.

Statistics Planning Review:

Timing issues, Missed milestones, or Unrealistic deadlines:

Overall Timing and Deadlines:

-The development plan assigns reasonable deadlines to tasks, with no apparent missed milestones. Each task is allocated enough time for completion.

-Phase 1 (March 7 deadline): This phase involves creating documents and a class diagram, which seems manageable and has a clear purpose.

-Phase 2: There are concerns regarding insufficient time allocated to implementing significant components for statistics, which could affect other major parts like matchmaking. The integration framework is expected to be developed at the same time, which might backfire, as insufficient time for core components could lead to more bugs when integrating the code in Phase 3. Additionally, Phase 3 assumes smooth integration and debugging, but without prior testing in Phase 2, debugging will be much more difficult.

-Missed Milestone: There is a lack of detailed planning for certain tasks, like "Start Coding Statistical Components." This vague description needs a more structured approach, such as identifying which components should be developed first to ensure smooth implementation.

Risks that could impact completion:

-Insufficient Planning for Critical Systems: The leaderboard system lacks a clear plan for data storage and dynamic updates, risking data issues that could affect matchmaking and leaderboards.

-Inter-team Dependencies: The statistics system relies on in-game data from the game logic team. If the game logic team faces delays or bugs in completing their components, it could delay the statistics team or introduce bugs in the system.

Suggestions for improving the timeline or reordering tasks to reduce risks:

-More Testing in Earlier Phases: Instead of waiting until Phase 3 to test, it's suggested to incorporate testing throughout Phase 2 and Phase 3. Testing in Phase 2 would focus on smaller, less complex

code, making debugging easier and more efficient. It may also free up time for Phase 2, ensuring a better core statistics system.

-Detailing Task Descriptions: Tasks such as “Start Coding Statistical Components” should include more details on the order in which components need to be developed. For instance, the implementation of statistics tracking (e.g., wins/losses, rank) should be prioritized before working on other components like matchmaking to ensure smooth integration.

-Coordinate with Game Logic Team: Since statistics integration depends on accurate in-game data from the game logic team, it’s crucial to establish clear communication and set deadlines with them. This will help prevent delays and bugs caused by misalignment between teams.