Planning analysis

Echo team's workflow planning was decent and demonstrates that the team was very communicative and collaborative during their planning process. This team made an excellent planning document that was highly detailed and well made. What stands out as particularly good about their planning is that all teams are making a concerted effort to keep possible conflicts and dependencies in mind for each task they scheduled. Of all the sub-teams, the Game Logic branch has the most realistic timeline with work efficiently spread out across all three Iterations. The Statistics team had a broad plan that resulted in their team not causing too many issues, however they also lack several key milestones for the project. This team has very few conflicts, with only one of them being notably serious. No team specifies Iteration 2 deliverables, implementing improvements, or creating a demo video at the end of Iteration 3.

Scheduling:

Echo team has some major scheduling issues that should be addressed. These issues are composed primarily of unrealistic estimations of workload with a few conflicts between sub-team tasks, missed milestones, and potential bottlenecks.

Unrealistic Deadlines:

- Every sub-team is scheduled to implement major portions of their code by the halfway point of Iteration 2 and make major progress in their integration by the end of Iteration 2.

This scheduling is unrealistic in a few ways: It is somewhat unrealistic to expect to implement enough of their systems to integrate them all into each other by the end of Iteration 2 is an underestimation of workload that is meant to be completed during Iteration 3. It is yet more unrealistic to expect every sub-team to be able to stick to this incredibly tight schedule without interruptions, which is asking a lot from busy students and creates a severe vulnerability to stoppages. These deadlines are most unrealistic in that they are scheduled at the same time as the Iteration 2 deliverables are supposed to be worked on, which were not scheduled for at all.

Scheduling Conflicts:

1. The entire group has opted to do much of their coding for Iteration 3 during Iteration 2 without fulfilling their deliverables.

2. The GUI team specifically starts with integration during Iteration 2 where the other sub-teams, especially the Game Logic team, are only just starting on their logic.

These conflicts arise from a lack of cohesive planning based on the assignment document. The lack of planning for the iteration 2 deliverables makes creating the letter difficult for submission and could potentially cause bottlenecks to accommodate the unplanned work. Pushing to have so much coding done so early on will make implementing the suggestions from other sub-teams much more difficult. The GUI team working on integration early on during Iteration 2 while the other sub-teams are only just starting to develop their systems will block the GUI team from accomplishing their goals on time.

3. The Networking team waits until Iteration 2 before creating a written design document which means that they are spending time on something in Iteration 2 that could have been completed in Iteration 1.
4. The Networking and Statistics teams did not specify final integration work for Iteration 3 which could mean that other teams are expected to take on that work when it is not in their schedules.
5. The front-loading of collaborative coding means that the Game Logic branch may have difficulties with other teams due to not being ready for integration until Iteration 3.

These conflicts are less serious and are more easily addressed, stemming from pervasive issues of mismatched work priorities between teams when planning their schedules. These conflicts span the entirety of the project and should be addressed quickly.

Scheduling Bottlenecks:

1. The entire group is focused on coding instead of the required Iteration 2 Deliverables which could potentially cause a severe pileup of responsibilities on all their sub-teams as they juggle too much work at once. This could result in deadlines being missed or requiring time to be put into rearranging schedules to alleviate workload.
2. Because the entire group's focus is on very quickly coding their backend and moving into integration really early on, this creates an incredible vulnerability to the smallest of delays in any of the sub-teams causing group-wide work stagnation.\
3. The preparatory work for the deliverables of Iteration 3, such as the demo videos, are completely unplanned for, which may cause issues or even a failure to meet the deadline when completing these vital portions of the project.

These are important to fix before they become a problem. Unfortunately, Iteration 2 has already happened so Echo team may be struggling with these bottlenecks already. Rearranging scheduled workloads to better match the expected pace of the project will largely alleviate these issues.

Milestones:

Echo team did quite well to prioritize the right goals to ensure efficient project completion and flow, however there were some missed deadlines, some incorrectly prioritized goals, and some goals that we recommend for them to prioritize in Iteration 3.

Missed Deadlines:

1. The Integration team did not consolidate the Iteration 2 deliverables into two files

   This issue was potentially caused by these tasks being entirely unscheduled.

Good Priorities:

1. GUI Design work in Iteration 1
2. Implementation work in Iteration 3

   These aspects are both important for ensuring the project is of high quality and that work flows well.

Incorrect Priorities:

1. Coding and integrating too early in Iteration 2
2. Load testing and error handling are pushed to the end of Iteration 3

   Early coding and integration cause too many potential issues and error handling is well defined but takes place far too late into the project to account for issues and may result in incomplete work by the end of Iteration 3.

Recommended Priorities:

1. Spread out implementation, integration, and testing throughout the project
2. More communication between sub-teams about integration and dependency

   These aspects will help keep the project flow efficient and eliminate surprises and hangups later in the project when time is precious.

Potential Risks:

Echo team has some very notable risks that should be addressed. These risks are likely to happen and are composed of scheduling conflicts, inter-sub-team dependencies, technical issues, and resource consumption risks.

Timeline Risks:

- Several sub-teams have timelines that may be too aggressive, and this aggressive approach may lead to multiple bugs and create bottlenecks or blockers for progress.
    - The GUI Team's plan to implement core GUI components with stubs in just one week could be challenging.
    - The Game Logic team plans to define core game mechanics and begin coding for four different games in a single week, which could lead to rushed or incomplete implementations filled with bugs.

Dependency Risks:

- Inter-team dependencies, such as needing to integrate simultaneously, may cause significant workload stress and delays.

Technical Risks:

- A new networking library may cause integration issues and should be closely examined to ensure stability.
- The AI sub-team depends on accurate game state representation and so the Game Logic team must pay close attention to testing and communicate well.

Resource Management and External Risks:

- Limited multiplayer expertise may slow development so more people may be needed to make up for this.
- API/service dependencies pose potential disruptions so fallback mechanisms should be established to make sure no issues arise.
- Coordination with external artists may cause delays so it is vital to have clear timelines and communication.

Suggested Improvements:

All sub-teams, If they haven't already, need to adjust everyone's schedules so that the group isn't overloaded with work and potentially piling up at important junctures. The expectations are both unrealistic and generate conflict within the workflow of the group's project. Missed milestones need to be rectified during the improvement documentation process. Iteration 3 deliverables need to be planned for to ensure completion without running into issues that could cause a failure to submit. To mitigate risks, readjust timelines and then maintain them and communicate about them frequently, reevaluate necessity of unique libraries and external collaborators, and implement standardized coding practices to ensure easier integration and workflow for all team members. By identifying and addressing risks and conflicts proactively, Echo team can minimize disruptions and ensure a high-quality product!