**UML and Classes:**

* On a scale from 0-10, how would you rate the usage of your UML diagram?
* How was it helpful/not helpful?
* Did you notice gaps between what you planned in your UML Class diagram and what you developed?
* How do you think you can approach the design process to eliminate the gaps?

I think we used the UML well to map out our ideas so I’d put it around an 8? It was useful to lay down our thoughts and visualize the relationships each class had with each other before we started coding things. I don’t think there were any major gaps within our diagrams? It was really nice to have done multiple iterations of the UML so that we could keep making changes as we see fit once we actually start the coding process. Larger gaps would've occurred due to not making changes in different UML iterations but I think it worked well for us at least.

**AGILE:**

* On a scale from 0-10, how would you rate the usage of your AGILE methodology?
* How was it helpful/not helpful?
* How do much do you think you understand how to implement the methodology/the ideology driving it?
* How might a better understanding of the methodology have helped in your development process?

I think the agile methodology of doing short sprints and consistent interviews was beneficial and helped us stay on track (8/10?). It really forced us to prioritize things, section off work, and help us keep members up to date with the project progress overall. I think kanban is a really helpful tool when there's a lot of individual tasks that need to be kept organized but, in our case, the tasks we put like “decide system” or “code demo” for the day of the sprint were unnecessary to organize (since those are just basic steps in the project process). Maybe if we used it to organize individual classes into tasks it could’ve been more helpful. This way, it’s easy to see quickly exactly which classes are done and what the progress is on the rest of them. I don’t think the kanban board was particularly helpful to us in this project but I think it has the potential to be really useful.

**Goal Setting:**

* How closely did you follow your initial goals set out for yourself?
* Did you find your goals realistic/achievable?
* How does this experience influence your goal setting for development in the future?

I think most of the initial goals had been met at the end. While it was tough, there was a very significant amount of work done and a lot of that was likely due to ambitious goals that had been set. I don’t think we exactly met our goals we had set for sprints but the goals were sufficiently reasonable such that we were close each time (just off by a couple method implementations or minor classes). For the future, I’d like to set the same type of realistic but slightly ambitious goals (such that even if we dont make it, it’s still okay) but also have more short term goals at each step of the way. These can simply be added to the kanban board or visualized as a daily to do list and can be more flexible than the stricter set overall goals. This would just help with organization and staying on track.

**Team Development:**

* How are developing by yourself and developing in a team different?
* What were main problems you encountered when developing your program as a team? How did you resolve it? What do you think you could do differently next time to prevent/anticipate it?

I enjoyed working with a team, especially such a nice one. While there was still individual work, the group members were very responsible and helpful in everything. A slight problem we encountered seemed to be coordinating our schedules outside of class time. Sometimes not everyone read the messages in the chat so maybe not everyone was always on the same page. Some suggestions to mitigate this are setting up meeting times beforehand so we can each plan our schedules around those meetings.