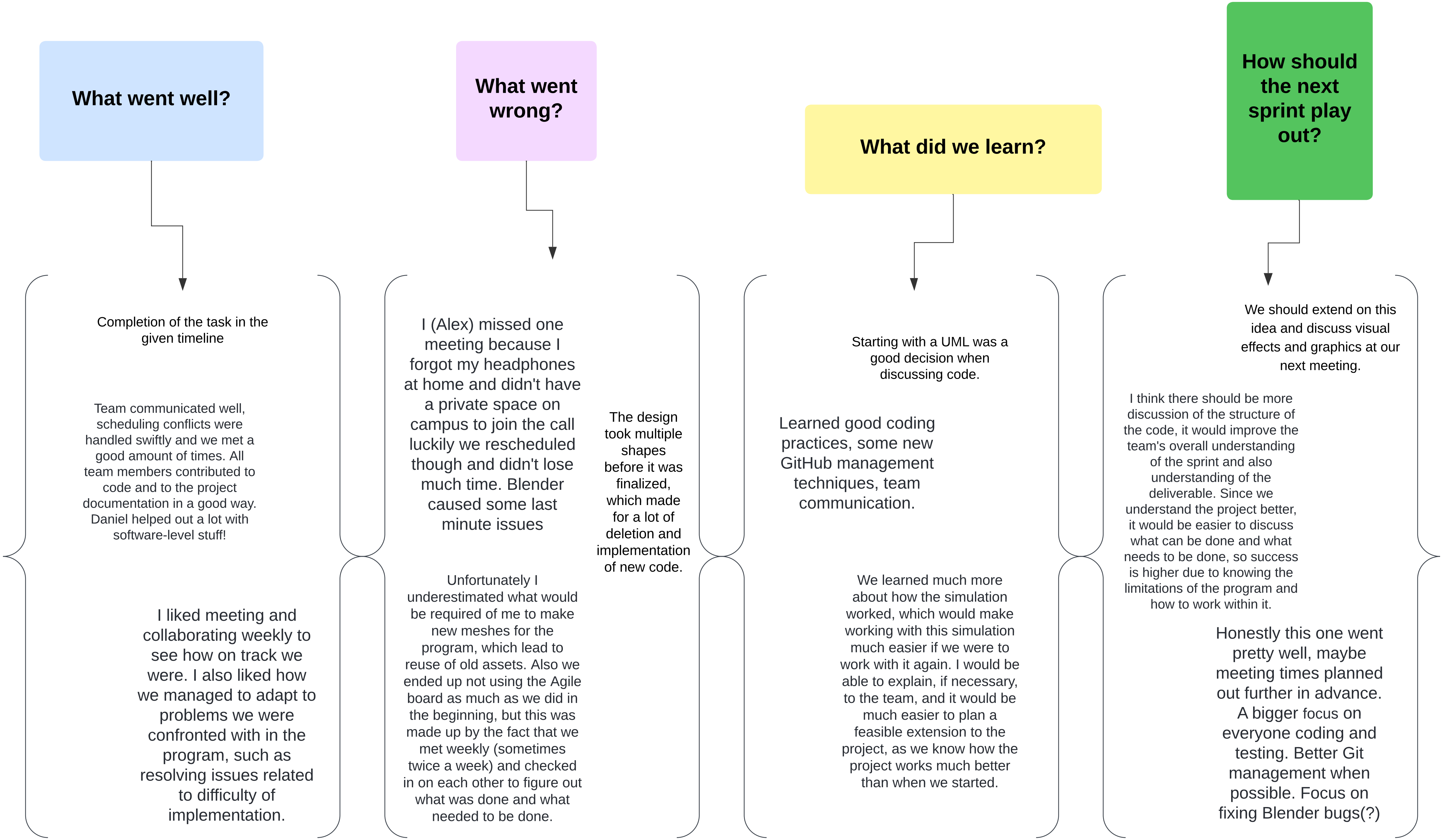
Sprint Retrospective

Khalid Barakzai, Daniel Kong, Alex Zhu | May 1, 2022



What went well?

Completion of the task in the given timeline

Team communicated well, scheduling conflicts were handled swiftly and we met a good amount of times. All team members contributed to code and to the project documentation in a good way. Daniel helped out a lot with software-level stuff!

I liked meeting and collaborating weekly to see how on track we were. I also liked how we managed to adapt to problems we were confronted with in the program, such as resolving issues related to difficulty of implementation.

What went wrong?

I (Alex) missed one meeting because I forgot my headphones at home and didn't have a private space on campus to join the call luckily we rescheduled though and didn't lose much time. Blender caused some last minute issues

The design took multiple shapes before it was finalized, which made for a lot of deletion and implementation of new code.

Unfortunately I underestimated what would be required of me to make new meshes for the program, which lead to reuse of old assets. Also we ended up not using the Agile board as much as we did in the beginning, but this was made up by the fact that we met weekly (sometimes twice a week) and checked in on each other to figure out what was done and what needed to be done.

What did we learn?

Starting with a UML was a good decision when discussing code.

Learned good coding practices, some new GitHub management techniques, team communication.

We learned much more about how the simulation worked, which would make working with this simulation much easier if we were to work with it again. I would be able to explain, if necessary, to the team, and it would be much easier to plan a feasible extension to the project, as we know how the project works much better than when we started.

How should the next sprint play out?

We should extend on this idea and discuss visual effects and graphics at our next meeting.

I think there should be more discussion of the structure of the code, it would improve the team's overall understanding of the sprint and also understanding of the deliverable. Since we understand the project better, it would be easier to discuss what can be done and what needs to be done, so success is higher due to knowing the limitations of the program and how to work within it.

Honestly this one went pretty well, maybe meeting times planned out further in advance. A bigger focus on everyone coding and testing. Better Git management when possible. Focus on fixing Blender bugs(?)