

Intro

Things really changed for project three. There was way more documentation of brand new kinds, way more freedom (which maybe wasn't the best), and most concerning, assignments from other classes. If anything, project three certainly made us manage our time better.

Good Stuff

We're really proud about the game we've made so far, so much so that we all expressed feelings of continuing it during the summer. Some of this pride comes from the level of communication we had, our drive to get stuff done quickly and on time, and our ability to follow the "vertical slice" mentality of Scrum and test our game frequently. Throughout the project we had great communication. We didn't have to spend time figuring out how we would work together since it was pretty clear from our first meeting and we all felt pretty comfortable working with each other right off the bat. Along with this we kept in touch pretty easily and consistently, using both discord and a group text in case discord didn't work and vice versa. We also got most of our stuff done quickly and on time. We always had some sort of progress to show after each sprint and while it wasn't exactly clear where the game was headed due to it's heavy reliance on human communication, we were still getting some good feedback from QA. Each week we had something new to test and could pretty easily get groups together to play since we're surrounded by our target market. All of this was a pretty decent ego boost and made us feel good about our progress. However, there were a few things that we slipped up on.

Stuff that hit the fan

While we had a lot of progress in terms of testing, we didn't have a playable digital prototype for quite some time. This was mainly due to the amount of work we had for other classes, the fact that our game is pretty unclear due to the communicative nature of game play, and the fact that we struggled to figure out exactly how some mechanics would be used during gameplay. We had the ability to do some things using the digital since sprint 3, but we still needed to have a narrator or ref to make sure the game was going well and tell everyone who one. We also felt that we could have started doing our work sooner in the week. Due to the nature of our other classes and the fact that our due dates were always on Tuesdays, we were able to easily get away with pushing our work off (and in some sprints it was almost necessary because of work loads from other classes) until the weekend and wrap it all up on Mondays. The clear issue with this was that we frequently didn't have the time to make changes if something went tragically wrong. We also felt that we could have done some better or more structured planning. We frequently would have to review at meetings to make sure what we needed to do and what was done with a lot of confusion throughout the conversation.

What we learned

The two biggest takeaways we got from making Dinner Party were that projects longer than 3 or 4 weeks lend themselves to Scrum much better and making group party games is wildly different from standard game development. With the original projects we did we weren't getting as invested since we knew that we couldn't make a finished product due to time constraints. Unintentionally we subconsciously sandbagged the work we could have done.

However, with this 6 week project, we felt more confident that we could get a product at least to an alpha stage. Along the lines of group party games, we learned a lot about those. First and foremost, it's got a lot of grey areas. There were a few weeks where we just had to make a change for the end of the sprint and wait until our next test session to see if the change worked. On the note of testing, we found that large group party games are pretty much impossible to bring to official QA. Due to the communication required and the sheer number of players, we weren't able to go into QA because of the noise and the fear that the other teams would beat us up after we called "6-10 testers!"

Do it this way next time

If we were to make this game again, hopefully it would be during the summer or some time when we didn't have other classes. But if that's absolutely avoidable, we would like to have kept better track of what we were doing. From the beginning, it would be best to use another planning software to keep checklists and track of features. While Pineapple/Redmine is super useful, it's also kind of awkward to use and not exactly intuitive to navigate. Along with that we definitely want to do some better planning. It was difficult to have a physical backlog since we didn't have a dedicated space to meet where we could leave such a device, but that would have helped us a lot.

Overall, this project went pretty well. We didn't expect to deliver an extraordinary game and took the shortcomings head on to learn from them.