Calvin Stoehr
Collin Dornbos
Preston Kelley
Nhan Nguyen
Jeff Everett
Julio Lemos

Flip Flops (Team 2) Project Reflection

The tools we used in making our project were Slack and Trello for communication,and Github and Unity for development. We also used the Agile methodology. Overall, these tools and methods served the group well, and there were few major issues with the tools themselves.

For communication, we primarily used Slack. Slack turned out to be a fantastic platform for quickly discussing project needs, sharing ideas, and ultimately became our medium for our group meetings since our schedules were not always consistent and meeting in person was difficult. As a whole, we would all use Slack in the future as a primary instant messenger with many additional features to make project collaboration easy.

Trello was our planning board. The application itself was very user-friendly, with a very interactive GUI it made planning our sprints easy, because we could see what needed to be worked on next and by who at any given moment. Unfortunately, Trello was not a perfect fit for this project or this class, since the Trello board was overwhelmed with lists containing a lot redundant information. For example, in the “Functional Requirements” list shared a lot of the same items as the “Current Sprint List.” While both are required, we ended up having 9 lists, each with many redundant entries. A better solution would be to have multiple pages to better organize lists so that nobody has to read “UI: Scoring Display” three times.

Rather than building a game from the engine up, we decided to use Unity. For making games, Unity turned out to be a good tool for beginners to game making and there were a lot of tutorials available from the creators of Unity as well as the community that uses the engine. It was fairly simple to make the game and it helped reduce development time. Unfortunately, Unity does not play particularly nice with Git, and it requires some tweaking in the git ignore file so that Unity’s object metadata doesn’t get in the way of committing changes.Other than that, Unity was great to work with.

As mentioned earlier, Git was used as our project repository. For standard files and non-application specific files (e.g. Unity files) it worked like a charm and it was fairly simple to share resources and simultaneously work on milestones and the project. It is a fairly simple tool that we barely scratched the surface of, but for our purposes it performed well.

To drive the project forward, we used Agile to model our workflow. The weekly meetings were beneficial to the progress of the game that we made, especially since it helped us space out the work, rather than making an entire game in the week before the presentation. The key to making Agile work, we found, was to be true to the weekly deadlines. There was a week or two where some of us didn’t finish the work we had designated, and it was reflected in the lists in the Trello board, and the overall progress of the game. With Agile, it was easy to get back on track, and still make the next deadline without putting everything else off track. If we had been using a waterfall methodology, it would have been much harder to bounce back and make meaningful project.