Huy Le, Brenner Campos December 14, 2021

COMP596 – 2D Game Design

Partnered Project – Communication

**COMMUNICATION METHODS FOR PARTNERED**

**2D GAME DEVELOPMENT PROJECT**

Throughout the construction of our “Battle Ships” game project in the 2D Game Design course, our two-man team (Huy, Brenner) had to develop a useful communication system in order to effectively work on our game. Because of our different daily schedules, real-time remote “code together” solutions like were usually not the best option for the team, but JetBrains’s *CodeWithMe* was used on a few occasions. More often than not, however, we ideally wanted to be able to work on parts of the project separately and then be able to easily merge our partner’s updates with ours. For this, we used GitHub. As for communicating about the project itself, we used several methods such as texting, video calls and in-person meetings.

GitHub is known in the software engineering world to be an industry-standard when it comes to reliable development platforms. It is a widely used solution to team-based projects, and so it made sense for us to use it for “pushing” and “pulling” our individual updates - terms that refer to storing and retrieving versions of the code. Our main process of working with GitHub involved each of us making a separate branch derived from a master branch for our own work, then manually merging with our partners when the time came to avoid merging conflicts.

Our modes of communicating *about* the project were mostly SMS texting, meeting before/after class to sit and talk in person, and through about ten or so virtual meetings on Microsoft Teams. Texting was the quickest and easiest way to get answers to less complicated questions. Inquiries like asking if our partner we knew how to fix a certain error or reference a particular value were easily communicated through SMS. We would also frequently update each other on our progresses through text messages. For harder questions, such as how to split up the work, possible ways to implement certain concepts, and general brainstorming, we met in person. This was done before or after our 2D Game Design class, usually at a table in the common space of the science building. MS Teams provided an ideal communication platform for us to go through and debug text together, as well as to being able to chat about more general project approaches and/or ideas.

Overall, the actual communication throughout our project worked effectively. One of the bigger struggles we faced was our conflicting schedules, meaning that we usually were not to spend a lot of time together trying to work through the project’s problems, but rather would often have to do our work on our own, occasionally huddling up to discuss and alter plans if necessary. Although our communication methods certainly worked for the project, parts of it would probably have been easier had we physically and/or virtually worked together more often. Regardless, doing work with a partner or team is always a favorable practice as software developers because we frequently tend to work in teams within our field. Learning the benefits, challenges, and intricacies of properly communicating and sharing our work with others is usually a worthwhile experience, and we felt like that certainly held true for this project.