Team Analysis

for

CentipedeArmy Team

Prepared by Emmanuel (Tobi) Afolayan, Drew Grubb, Jed Hutto, Jesse Miara, Ryan Weeks

CS 3398.264

March 7, 2018

During this project, the team worked in a collaborative manner to complete most of the required tasks. When the project was assigned, we decided to meet up every Friday in Alkek library from 2pm – 6pm. Because of this, we were very productive as a team and communication was not a problem because of the face-to-face interaction we had. We were able to share ideas on the spot and determine which is best for the project. Our means of communication was through GroupMe and Slack.

One of the challenges we experienced when writing the SRS was taking into consideration several nonspecific requirements and the extra credit requirements. These requirements created a difficult dynamic for functions within the SRS, because the requirements were developer driven as opposed to customer driven. This led to somewhat arbitrary labeling of design priorities because understanding how long a priority would take to implement was unclear.

The challenge we experienced when documenting the SDD was how to organize the class diagrams and how to create relationships between all the class diagrams. Doing some extra research and consulting the professor helped us get through that hurdle. We felt that our GUI experience, as well as Java experience among some members, was limited enough to create an extra learning curve. Due to nature of scheduling conflicts and the class environment, a different difficulty was getting all 5 members together at the same time. This forced us to work independently and we noticed that the progress level was not as successful as when we all met together. This can be contrasted with a real Software Engineering environment, where the entire job is dedicated to communication and design, so scheduling conflicts and missing members will not occur nearly as often.

The coding phase of this project is where most of the challenges were created by time constraints. Due to the nature of waterfall, any core changes we had to make to make the project work needed to be edited in the SRS and SDD first. However, time constraints forced those changes to be limited to just the code portion, so many classes have extra functionalities not listed in the SDD. Time constraints limited Unit Testing to the algorithms for piece move sets. Unit testing is also difficult for testing GUI implementation since an individual unit test cannot look directly at the display canvas.

We did not have any major disagreements throughout this project. Everyone seemed to have a clear understanding of the project requirements and the class functionalities. The collaborative meetings we had were key to the success of this project due to the ability to bounce ideas off one another. In an ideal recreation of this project, we would like to have more time for concrete design and an environment where personal schedules do not conflict with the work of the group. Overall, each member of the team gained a great learning experience in waterfall design and working within a team to complete a nontrivial project.