COP3503 Final Report
Spring 2015
Group 22

During the course of our group project, we learned many things about developing large scale projects in a software development setting. The most important concept that our group learned about was the importance of communication and version control. In order to keep the group's work coherent with each other, we held regular weekly meetings to discuss the direction that we wanted to take with our project. These meetings usually lasted between one and two hours each. We also used a website called GitHub, which is a popular repository for the git revision and version control system. We used this in order to keep our program organized, to allow for members of the group to code the project without us having to meet each other whenever we wanted to code the project, and to keep track of specific versions of the project in case we wanted to revise our project to a specific version. This is relevant to the software industry because git and similar systems such as subversion are commonly used in large projects in the software industry as they provide an easy, functional, mature, and necessary for organization and collaboration when programming.

That being said, our group also divided up certain aspects of the project in order to further organize our project to keep it easier to code as a whole. While we also used git to synchronize development online, we also spent much time physically meeting with each other in order to code our project as well. The experiences that we all had while working on this project gave us valuable insight into some of the main experiences that are in the software development industry, especially in terms of the importance of version control and being organized as a group when we were designing our programs and our code. It also gave us a better understanding of deadlines, code complexity, and what is feasible to meet those deadlines. Through this project, we also learned that teamwork is extremely important in designing most modern programs, even if they seem relatively easy as compared to professionally-made software.

In the future, we would definitely use a similar organization technique in terms of organizing labor. We however would research the specific gui calls and libraries we will need before attempting to layout the program as we have here. We had to make the changes in how our function runs over time from the original plan, using a main gui loop that updates everything. The mouse states were rather interesting in that we could actually use them to trigger the functions that process the data in the game. Our original plan was to have a turn rotator loop that would do similar things based on input rather than one that reads states, but the GUI in SDL forced us to change. It would be best to lay out all the functions before a unified engine loop in this game, with access to everything defined elsewhere.