Sprint 2 Reflection – Team 4

For sprint two, our team accomplished all the necessary points of functionality and were able to finish on time. However, as was reflected in our burn down chart, our team tended to start working on the sprint rather late. As a result, for the early functionality check, we submitted the night it was due. In addition, we were missing too many components to complete the requirements for the functionality check in.

Some of our team members did start working on the project early on. In particular, the blocks and items classes were finished early on in the process. However, more involved classes such as enemy and Link classes were not finished until the last couple of days. This is reflected in our burndown chart as most tasks on our project board did not have significant progress until the few days before the functionality check in. Similarly, the tasks were not finished until the day of the sprint two deadline.

This tendency to complete tasks later in the sprint negatively impacted our group’s ability to refactor code. As some classes were not completely finished until late in the sprint, emphasis became placed on functionality and not on the quality of code. As a result, some code became unorganized. In addition, some methods could have been made to be more cohesive and complete one function. This tendency to finish late in the process also limited our team member’s ability to coordinate and work together throughout the sprint.

Overall, for sprint three, our team could do a better job of getting started earlier and doing consistent work throughout the sprint. This will allow us to better manage the completion of the project tasks and better coordinate with one another. In addition, finishing early will give us more time to refactor code at the end of the sprint to achieve more maintainable and efficient code.

Another change our team will make for sprint 3 is to communicate more effectively. Our burndown chart shows that each team member and task were completed at very different times, and there was not much overlap. This reflects the fact that our team members each worked on this sprint independently and did not collaborate much. Some of our code could have been implemented faster and more effectively if more team communication and collaboration occurred. Instead, we each tended to work largely independently of each other.

Despite working independently from each other, our team did implement many very beneficial changes from sprint zero. For example, none of our team members implemented a fully functional command interface for sprint zero. However, for sprint two, our team implemented a command interface and the suggested state pattern for Link and enemy classes. In addition, factory methods were used when creating sprites. This greatly improved our code quality, and these features will continue to be used and expanded in future sprints.

Overall, our team was able to effectively complete the functionality requirements of sprint two. In addition, we were able to implement the suggested design patterns for this sprint. However, our team will make necessary changes for sprint three to work more effectively together. In addition, we are going to work to complete tasks throughout the entirety of the sprint, and at the end.