Sprint 4 Reflection

Brogrammers

MWF 10:20

There was a lot of adjustment and learning that had to be done by the team for this sprint with the virtual setup. However, at the beginning of the sprint, we had a group call to delegate roles and set a clear timeline. Our initial goal was to have all functionality for the sprint done by the end of the second week so that we could spend the third week refactoring and performing code reviews. This way, we could avoid silly mistakes or areas of questionable code like in past sprints. However, once again, our group faced issues with communication and commitment from some members. This resulted in certain requirements and areas lacking, even with other members picking up extra responsibilities to try and satisfy all the criteria.

In addition to picking up some extra work, some members also decided to have a conversation with the TA, Steven, to see where they could improve with documentation and code quality. These recommendations were taken seriously and the group attempted to use as much of them as possible in their implementations and documentation. On Trello, additional information about implementation and steps were added to each card. Additionally, Trello cards were split up more specifically, which was a big improvement over the past sprints. To track accountability of each member, the group implemented sprint points like Steven suggested. While it was hard to go back and assign specific point values for all sprints, the group decided to use color labels on Trello in order to show the general importance, time, and difficulty of each feature implemented. Easiest and quickest tasks were given a green label, followed by yellow, then orange, and finally red for extremely long and complex tasks.

Below is a breakdown of each members contributions over each sprint.

Note: In earlier sprints, Trello cards were not divided that well. For example, cards were named “Player,” “Enemies,” and “Items.” There were probably more tasks involved with some of these cards than others. Link required a whole state machine and many states while Enemies also involved movement code and advanced sprites while things like Items may have only involved creating a class and a stationary sprite for each item. While this difference is marked by a green vs a red label on Trello, the following counts may not provide as detailed a picture as we would like.

Sprint 2:

	Green	Yellow	Orange	Red
Austin	2	1	0	1
Ben	4	2	0	0
Dan	1	0	0	1
Huang	1	0	0	0
Suraj	2	0	0	1
Xueyang	1	0	0	0

Sprint 3:

	Green	Yellow	Orange	Red
Austin	1	1	1	1
Ben	0	0	1	0
Dan	0	1	2	0
Huang	1	0	0	0
Suraj	0	1	4	1

Sprint 4:

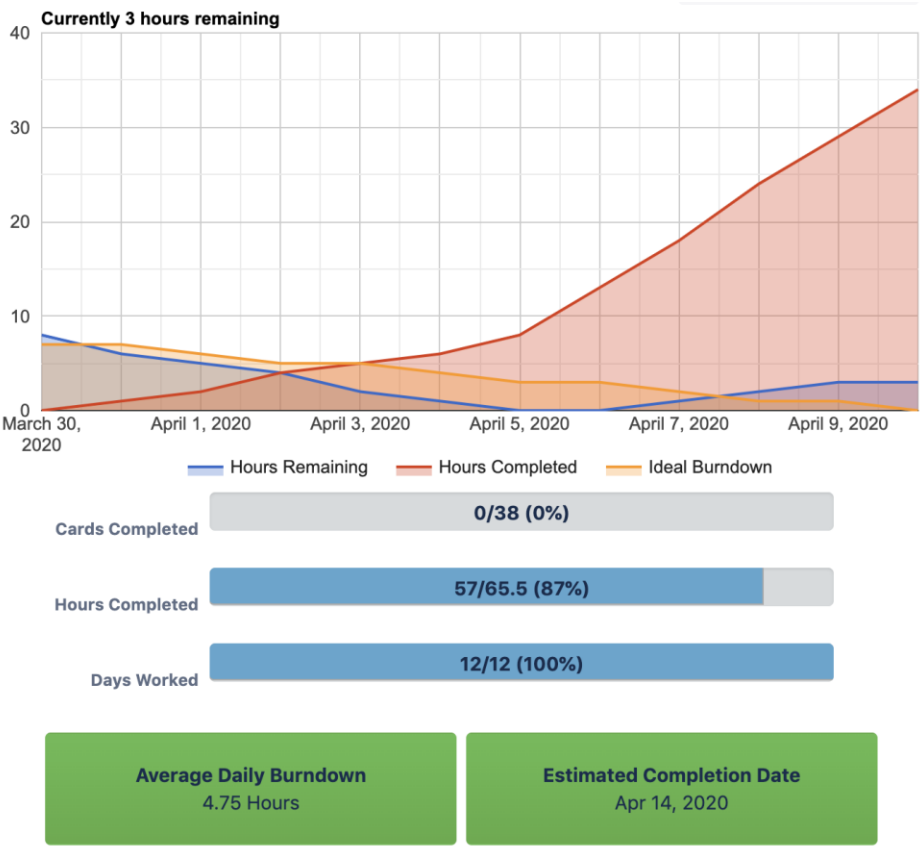
	Green	Yellow	Orange	Red
Austin	1	1	0	1
Ben	2	4	0	0
Dan	3	2	2	1
Huang	0	0	0	0
Suraj	4	1	1	2
Xueyang	0	0	0	0

With these metrics in mind (although not completely accurate), there is clear reason for certain sections of the code not being fully complete up to this point. For this sprint, the responsibilities were as follows:

- Austin: Level Scrolling and Link Item Fixes
- Ben: Mouse Clicking for Room Switching
- Dan: HUD
- Huang: Game States (Game over, intro state, pause)
- Suraj: Level Save State, Sound
- Xueyang: Utility Classes/Configuration to Avoid Magic Numbers/Strings

Regarding the above responsibilities, all of Austin, Ben, Dan, and Suraj's work was done for the most part. To handle unfinished game states, Dan took on the additional responsibility of working on the Pause State since picking items in the pause state was acting as a blocker to the HUD screen. However, this work wasn't able to be completely finished. Suraj and Ben worked on adding functionality for multiple items and Ben also added an additional "Bullet Hell" room. Austin also helped fix multiple bugs with level save and collisions. Finally, for the unfinished utility classes and configuration, Ben started working on a framework to implement that. However, this couldn't be finished for all magic numbers and strings across the project. Suraj also added csv files for all 17 dungeon rooms from the original LOZ first dungeon and added special features like item drops and door changes based on the current room state.

The burndown chart for this sprint can be seen below.



Again, the “Cards Completed” metric didn’t update on the burndown chart for some reason. About 30 cards were fully completed with a few being in progress.

To improve our performance, it would help if all members were able to communicate throughout the sprint and complete their tasks on time. If not, it would be helpful for members to at least alert others ahead of time that they needed additional help with their tasks. However, this trend of certain members contributing unfinished code late or not submitting code at all without any comment made it hard for all the tasks to be met, especially in heavy sprints like Sprint 3 and Sprint 4. For the amount of commitment in the group, the number of features that were able to be implemented successfully was definitely more than expected.

Hopefully in Sprint 5, communication can improve and all members of the group can provide equal amounts of effort towards completing missing parts and adding new special features. Or if not, hopefully members who aren’t able to complete their tasks can alert the group well beforehand so that tasks can be reallocated effectively.