Sprint Review 4

Features implemented:

- A chat filter function was implemented

Issues fixed:

- Revised issues with the sequence diagrams
- Revised the product backlog
- Revised the sprint backlog
- Resolved security issues in dependencies

Implementation review *(What went well in the implementation, what problems occurred, how problems were solved)*

Went well:

We established our goals for the deliverable which included taking a lighter touch to focus more on the documentation rather than implementing features. We wanted to ensure our documentation was perfect prior to submission, so we started earlier than usual. We also agreed upon meeting on Mondays for a shorter meeting to establish what we are working on this week.

Problems:

We had a minor communication issue regarding setting up a meeting time with the professor involving scheduling on our end.

Solution:

We will attempt to meet with the professor again in the future when we can align our schedules for the opportunity to do so.

Changes made:

- Minor visual changes
- Code for tile verification has been upgraded to include tile adjacency

Plans for next sprint:

The primary focus for the upcoming sprint will be to continue the task we didn't finish this sprint due to taking a lighter coding approach to focus more on the deliverable itself. Most of the changes involve updating the UI or adding primary game functionality.

- In *Python and ReactJS*, create a decrementing move counter based on the moves taken and the moves made. *ReactJS* will reflect the move made and *Python* will decrement the counter.
- In *Python and ReactJS*, create an endpoint /move that will move the character to the appropriate spot based on where the user clicked.
- In *ReactJS*, make a generic error message for when /move returns False.
- In *ReactJS*, reflect a player's valid move on the game board if /move returns True.
- In *Python*, create an endpoint that accepts an object containing a chat message, userid, and username. It will run a validation to ensure the message doesn't violate the community chat guidelines of Squared.
- In *ReactJS*, based on the queue from *Python*, assign colors to the players.
- In *Python*, take players from the queue and stick them into the game data structure.
- In *ReactJS*, based on the output from validate move, move the character accordingly.
- In *HTML*, add a page that shows the player that they are waiting for more players to join before starting.
- In *ReactJS*, add the player characters to the board.
- In *Python*, add a method for hashing passwords with SHA-256
- In *Python*, add routes for registration and login.
- In *HTML*, adjust the UI in game.jsx to make the score, turn, and dice appear on the screen to the side of the board.
- In *HTML*, add a score component to the board.
- In *Python and ReactJS*, create chat box functionality.
- In *HTML*, general improvements to flesh it out to make it appealing to the user.
- In *Python*, start working on various item functions.

Scrum review: *(What went well in Scrum, what could be improved, and what changes will be made)*

I think this Scrum was extremely effective, we communicated well and we established our goals for the sprint. We acknowledged limitations and concluded that this sprint wasn't going to be an extremely code heavy sprint with a primary focus on minor game additions and UI visual upgrades. We concluded that it may not be the best solution to have the only meeting be the mid-week meeting, so we agreed upon having a brief meeting on Monday to establish our goals for the week and then using the Wednesday meeting as checkup and deliverable discussion.