Group 5 Retrospective

One area of our development process that can be improved upon is having one pull request per issue. Currently, we begin a programming session by branching off of master with the intention of having one developer task per branch. During development we often notice several issues that should be addressed so we include these fixes in the pull request. Furthermore, since the next developer task may have code dependencies in the current branch, we often end up including that feature in the same branch. This occurs as it is more convenient than creating the pull request for the original task and having to wait until it gets merged into master before we can branch off master with the new changes to begin working on the next feature. This leads to our pull requests growing in size and containing multiple developer tasks. These bloated pull requests become much harder to digest for other team members reviewing them, and their size acts as a deterrent to thoroughly go through the code. This caused some pull requests to be not as closely examined as they should have been before being merged into master.

There are multiple ways to improve upon this. One solution is if we notice some code that needs a sizable amount of refactoring, rather than including it all in one pull request is to utilize TODOs in the code, so it can be addressed in a future branch. Similarly, this can also be achieved by adding a developer task/story that addresses the concern. If the next feature has dependencies on the current branch we are working on, we can make the pull request for the current feature and then branch off the current branch instead of master to work on the new feature. This allows us to keep the pull request limited to a single developer task. Additionally, we do not block further productivity by having to wait until the pull request merges into master to get the dependencies needed to begin working on the next feature. If in the case, we finish the next feature before the previous feature's pull request is merged, we will set the new feature's pull request to be merged into the branch it was branched off of so reviewers only see the changes in that new branch. We will ensure that we merge the previous feature's branch before this new pull request and once it is merged we will set this new pull request to be merged into master rather than the branch that it was branched off of, fixing any merge conflicts in the process. Success can be concretely measured if all of our pull requests have no more than a single issue linked to them at the end of the iteration.