

The process of our code integration was not a very strict structure. For most of the project, all of our members were working on separate branches and merging whenever each individual saw fit. Oftentimes, we would find ourselves working on features small enough that they could just be done on our main branch. Alternatively, we would have people working on larger features as well that would have them working on a branch for a long time without merging. One specific hardship that we had with integrating was due to a refactor. We had hit indicators working well with raycasting and attacking, and then the raycasting needed a refactor. The hit indicators were not working with the refactor and merging the code to get it to work was a massive pain. Coming down towards the end of the project's time frame, most of us started working on our repository's main branch and pushing at will. With that being said, I think there was still just one main integration strategy that we used. We started out using a "shake n' bake"-like method when we were all working on separate branches. While we were using separate branches, we would still end up merging to the main branch sporadically based on when each member's respective features were complete. As the deadline approached, our strategy still resembled the "shake n' bake" method as we were all on the same branch trying to get each feature completed, tested, and then pushed as soon as possible.