Our process for integration in Project 4 was pretty straightforward and somewhat messy. Since every class we needed for Project 4 already existed from Project 3, it would not have really been possible for us to do something like Bottom-Up Integration where we would code and test the components with lower hierarchical status before even beginning the code on the components with higher hierarchical status. Since Top-Down Integration similarly requires the implementation of certain levels of the hierarchy before implementing others, this was also not a possible strategy for us. So what we did for Project 4 is most accurately labeled as **All-at-Once Integration**. Specifically what we did was we basically coded each of our components concurrently with one another and tested them on our own before implementing them and pushing them onto the master branch of the GitHub repository. We would test them somewhat loosely and haphazardly, but they did all seem to work well enough to integrate together. After everyone pushed every bit of code that they changed, we tested again. This time, we did the same informal testing as we did on our own components, but, with the addition of the test suite, the process was more formalized and rigorous. During these tests, the weaknesses of this integration strategy were obvious because we found a plethora of major errors that we did not find when testing our own components. Luckily, because the c++ error messages are very specific, and because of our rigorous test suite the errors were not too hard to locate and fix, which we did pretty quickly.