

# 1632 D1 Introduction

---

*By: Josh Rodstein & Will Taylor*

All-in-all the deliverable did not present too many challenges. There were however a few points that became more clear as we worked, which would lead us to approach things a bit differently the next time around. Our development and workflow was made quite easy and enjoyable by the use of Git/hub to manage our documents and collaborations process. The requirements laid out by professor Laboon, in a sense, guided our decisions for the test plan. If we were ever in doubt about the cases we had come up with, we need only look to the next requirement to see that perhaps we should move one of those cases into the next set.

We decided early to use the Requirements to craft our test cases. For instance, the difference between R1, and R2. R1 required us to test invalid input, as specified by the requirement. While R2 required us to test expected valid input. It simply became a matter of us deciding what the best invalid and valid input to test. We decided on edge cases in these cases, as from our lectures in this class and in 1530, we have learned that most of the unexpected/defective functionality occurs on the edged and ad the bounds of possible input. For instance, is invalid input is anything other than a non-negative integer, we would test with -1, and -1000 to provide a high and low boundary. Similarly with valid input, 1 and 1000.

It was clear that the goal of this in our minds was to suss out our test plans as much as possible from our requirements Admittedly, we came across a couple defects just by playing the game, however, it took a systematic breakdown of game functionality into tests for us to understand how to quantify those defects into a readable and operable test plan in order to reproduce them, and to subsequently fix them.