

Halfling chess report
10/30/24
Josh Dobbs
CIS163

My test_game file includes a test case for each and every method, sometimes checking both invalid and valid inputs. I realized that making a unittester file is one of the first things you should do before writing it all. If I were to read the full description of how the game works and what each method should output/do, I could make a unittest for that method before writing the actual method first, this is because if I know exactly what should be happening through this method I can generate the different results the test should have. Then write the method and run the test to see if the method works with all cases I provided. I did not do this on this project because I did not think of it this way. I always want to know why all cases don't pass in PrairieLearn and I usually want to see which test did not pass, if I were to make my own I would be able to. Obviously sometimes PrairieLearn cases have a test that I don't.

The project went ok, it was definitely difficult to understand at some points. Not having output examples makes it difficult sometimes to understand what exactly prairieLearn wants. Or sometimes an attribute or argument isn't really explained for why it should be there nor what it should be checked for. For example, range and move are supposedly self explanatory for the project, but this can be difficult for unittesting for PrairieLearn. I was stuck on is_valid_move and attack for a while because I included the check for range and move in the character class instead of the abstract classes of Hero and Villain or Necromancer or Paladin. And for some reason range is mentioned to be implemented in the character_at function on the project description so that had me tripped up for a while, not really sure why that is there.

The process I took for this process was to do my best understanding the game, then work through the steps one by one. Turned it in my first time and only had about 20% of cases pass, went to fix the creatures stuff first since most of it was simply fixed like instance variables being incorrect, then got to my first challenge which was discovering why calc_dice in warrior was not working. This a lot of people had struggles with, eventually it came down to something difficult to notice (comparing objects). Next was fixing the basics of Hero and Villain, and the instance variables of character. This all got me up to about 70% then the abstract methods needed fixing and so did a lot of stuff in dungeon. Some of my stuff still doesn't pass and I'm not quite sure why. But I landed at a decent 90%.