

PROJECT 'dungeon_dudes'

WriteUp

MOD O - OOP 1B

170D WOBC

Class 23-001

WO1 Josh Kaplan

October 25, 2023

1 *Project Summary*

The project *dungeon_dudes* is an object-oriented program in Python which are to create a Character or Monster for a Dungeon-Crawler style RPG in accordance with an already existing codebase. The Character or Monster must match the associated Character Stats and information found in the Manual for Dungeon Dudes. This is in order to teach us the appropriate way to integrate work into an existing codebase and master the art of git collaboration.

2 Challenges and Successes

As mentioned in class before, I believe the biggest challenge with this project (and to falling-in on an existing codebase) is fully knowing and understanding what functionality is available from the BaseClasses (essentially what is available from the API).

I had many Successes once I figured out how much of the Fighter Class I could reuse and port over for the Cleric. The difficulty lay in the implementation of the passive skills, as they aren't necessarily always implementable as a class Method like the "active" skills. Some passive skills were essentially just "bolted-on" to the active skills as necessary.

Unittests were kindof a mixed-bag, as its hard to fully unittest subsets of a codebase without also testing the entirety of the codebase that the subset calls upon.

Git was both a Challenge and a Success, as there were a time where I forogt to `git pull` before merging with master, causing massive merge-conflicts. However, solving the merge-conflicts and solidifying the right way to go about group collaboration was a great learning process and a very valuable skill.

I enjoyed this as a project, from an "applicability" point-of-view, and from a "fun" point of view. Subject Matter was different from the usual projects and felt more in line with the applicability of `nfl`.

3 Lessons Learned

My lessons learned are to take some time to better research what code is given in a large OOP codebase (i.e. understanding the API). Lessons learned also includes unittesting small subsets of the codebase and mastering git, despite screwing up the master branch.