Dungeon Dudes

Test Plan

WO1 Joosung, Min

This Test Plan is documented to described the scope and approach for all the testing activities of the project.

Scope

The following files are to be tested:

1. ranger.py
2. ranger_item_generator.py

Test Setup

```
def setUp:
    self.character = Ranger("Ranger")
```

Test Cases

ranger.py

Contains attributes and methods to create a Ranger Class Object

Customer object requires a 'name' of the character from the user

Methods

```
def test_base_stat:
    'Checks to see if the base stat is correct'
    Ranger level 1 base state:
        - Hit point = 90
        - Strength = 7
        - Agility = 14
        - Intelligence = 5
        - Special (focus) = 2
    All assert statments should pass
```

```
def test_active_skill:
    '''Test to check if the Ranger learn skills correctly
       when leveling up'''
       - skills_list = list of all available skills
            at different levels
       - skills_dict = dictionary of all available skills
            at different levels
       Check if only learned skills exist in the skills_list
       and if the Ranger learned a new skill, check if the
       skill exists in the skills_dict
    All assert statments should pass

def test_attack:
    '''Test to check if the Ranger and companions attack corerectl'''
        combat = self.character.modify_damage(damage)
            - modify_damage returns a list of tuples to create
            individual CombatActions used during monster encounter
            - The lenght of the combat action should match the number
            of actions
        If No companion is summoned:
            One combat action is done
        If 1 companion is summoned:
            There should two combat actions, one for the ranger and
            one for the companion
        If 2 companions are summoned:
            There should be three combat actions
        If steel trap is activated:
            There should be 4 combat actions, since steele trap
            does damage every turn
    All assert statments should pass

def test_equipment_attack:
    '''Check if the weapon has the correct weapon modifier'''
    weapon = self.character._weapon
    if weapon has a "Wrath" or "Defense" in the name:
        The moidify should exist and the amount should be
        less or equal to the level + 10
    All assert statments should pass

def test_equipment_armor:
    '''Check if the armor has correct Elemental Resist'''
    armor = self.charater._armor
    Ranger armor should have 3 different types of elemental resist
        and all three should add up to (level * 3) + 30
    All assert statments should pass
```

```
def test_equipment_accessory:
    '''Check if the accessory has the correct element affinity''
    accessory = self.character._accessory
    Ranger accessory should have one elemental affinity
        between 'Ice', 'Fire', or 'Lightning'
    All assert statments should pass
```

How To Build

To run the automatic unit test:

- Move the test source file to the top level directory and run: python3 -m unittest discover test

All of the Test Cases should pass without error or failures.