TbcSystem

Nathan Pinkelton

5/28/2025

Things Needed:

- A file named tbc (Turned based Combat)
- A file That imports tbc and uses its properties/classes
    - We can use the code written by Andy

Turned based combat file:

- Need functions or classes for:
    - Character
        - Name:
        - Hit Points
        - Hit Chance
        - Max Damage
        - Armor
    - Print Stats
    - Fight

Character:

- Needs to be a class I believe
- All the attributes listed will need to have properties so they can easily be edited in the code and in other files.
- Will start the Character class with initializing attributes listed below:
- Attributes include: Name, Hit points, hit Chance, max damage, and armor
    - Name
        - Set up property/property setter for name
        - Make name.setter have the dunder <name> get value
        - This way we can change the name of the characters with ease
        - We can set self.name in the initializer to be “anonymous”
    - Hit points
        - Should be an integer

        - We need hit points to be allowed to go negative or to zero
        - Somewhere in the code (whether it's in here or in the fight function/class (Probably in fight)) we need to make it so that when Hit points <= 0, the fight ends and the winner is proclaimed
        - In HP's setter, run if statements to make sure Programmer/user types in a positive number
            - Def hitPoints(self, value)
            - If it's an integer that is greater than 0, let the input be the value
                - If not, automatically set HP to 10
                - Tell them What they did wrong
    - Hit chance
        - Should be an integer from 0 – 100
        - This will determine if an attack is successful
        - Will most likely want to import random
        - In hit chance's setter, run if statements to make sure Programmer/user types in a positive number
            - If hit chance is an integer that is greater than 0 and less than 100, let the input be the value
                - If not, automatically set hit chance to 50
                - Tell them what they did wrong
    - Max damage
        - Should be a reasonable integer
        - Will be the maximum range for hits that go through.
        - Damage (somewhere) will need to subtract armor blockage from itself
        - In Max damage's setter, run if statements to make sure Programmer/user types in a positive number
            - If max damage is a positive integer, let the input be
            - If not, tell them what went wrong and set Max damage to 1
    - Armor
        - Should be an integer lower than opponents max damage (so that things are actually fun)(we'll worry about that later if we can)
        - Will need to be used to block/negate some damage that comes through in Fight sequence
        - In Armor's setter, run if statements to make sure Programmer/user types in a positive number
            - If armor is greater than or equal to zero, let the input be
            - If not, explain the problem, and set armor to 0

Print stats:

- Method will need to have ‘self’ as a parameter
- Print out character's name
- Then print out the character’s attribute/variable names that you wish to describe (hit points, max damage, etc.) and their values.

Fight: (function I think)

- Parameters: self, Character
- We need the fight sequence to, when the user presses enter, have an effect on the hit points of the characters fighting.
    - KeepGoing While loop with an input to run multiple rounds
        - Maybe just an input? No need for a variable to store what’s typed. We just need to have a pause between rounds.
- When we use this code in a different program/file, it will take the parameters of the created characters.
- How the process should look though:
    - Have <diceRoll> get a random number from 0 – 100. Then, if <diceRoll> is less than or equal to the Hit chance of the character, damage will be dealt
    - If damage is dealt, roll again (random int) between 1 and the Max damage of the character. Subtract that number from the armor the opponent may have.
        - If the Damage is less than or equal to zero, then the attack will be blocked and have no effect on the opponent’s health
    - Subtract the damage amount from the opponent’s hit points
    - Each player will get a chance to attack the other
        - Which means the process will be repeated for the other character
    - After an attack misses, is blocked, or lands, it will inform the user what has happened
    - After that, the characters’ health will be shown.
- If a character’s hit points are less than or equal to 0, then the other player will win.
    - If both characters’ hit points are less than or equal to 0, then it will be a draw.

Main():

- The main function will assign the Character() class to a variable name
- Then that variable (in this case, < hero>) will have its attributes defined (hero.name, hit points, hit chance, etc.)
- Make another variable and assign it the Character() class (this one will be <other>)

- Do the same thing as before. Feel free to vary the two's attributes.
- Print the status of both variables
- Run the fight function

NEXT:

Combat file:

- Make a file named combat.py
- Import the tbc file
- Copy Andy's code
    - Here is the "borrowed" code:
    - ```
      import tbc

      def main():
          hero = tbc.Character()
          hero.name = "Hero"
          hero.hitPoints = 10
          hero.hitChance = 50
          hero.maxDamage = 5
          hero.armor = 2

          monster = tbc.Character("Monster", 20, 30, 5, 0)

          hero.printStats()
          monster.printStats()

          tbc.fight(hero, monster)

      if __name__ == "__main__":
          main()
      ```
- See if it works
      ```