

Project Documentation

The code is meant to mimic an adventure game and allow users to play the role of a hero when exploring and fighting off the enemies. Heroes are rewarded with 'munnies' when they have successfully defeated an enemy, which they can buy items with at the town shop. Heroes continue to level up the more battles they win. The code runs properly, works as expected and is able to complete everything as described.

In order to run the program, open the terminal and once you are in the directory containing `adventure_game.py`, run the script using the command `'python adventure_game.py'` or `'python3 adventure_game.py'`.

How to use the program / how to interpret the output of the program:

- Character Creation

At the start, a character is automatically created with default attributes (health, strength, etc.).

- Game Actions

Explore: Engage in battles with random enemies.

Enter Shop: Buy items to assist in your adventures.

Quit: Exit the game.

- Navigating the Game

The game prompts the user for inputs to choose an action, type the corresponding number (e.g., 1 for explore) and press enter.

- Battles

The user chooses to attack or not. If they choose to attack, the enemy's health decreases. If not, the user's character loses health. The battle continues until the enemy or the user's character is defeated. If the character's health reaches zero, the game ends, but if they have a health potion, they can choose to use it to restore their health.

- Shopping

In towns, users can buy items using "munnies". Players can choose items to purchase, and the cost is deducted from their munies.

- Understanding the Output

Battle Output: Displays the health of the player and the enemy, the result of the battle, any munies gained, and if the user has been leveled up.

Shop Output: Shows the available items, their costs, and the player's current munies.

Annotated Bibliography

1. W3Schools - Python Tutorial

- Link: <https://www.w3schools.com/python/default.asp>
- Summary: This site provides useful tutorials on Python programming. It covers a wide range of topics from basic syntax to advanced concepts such as classes and file handling.
- How it was useful: This site answered a lot of questions that came up about working with classes, different functions, objects and more.

2. AskPython - Text-based Adventure Game

- Link: <https://www.askpython.com/python/text-based-adventure-game>
- Summary: This article provides a guide on creating a simple text-based adventure game in Python. It explains the basic game structure, including setting up a game loop, creating game scenarios, and handling user input.
- How it was useful: It focuses directly on creating a text-based adventure game which is our topic as well, so it was helpful to get some guidance and see examples.

3. Python.org - Python 3 Tutorial on Classes

- Link: <https://docs.python.org/3/tutorial/classes.html>
- Summary: This site gave an in-depth look at classes in Python. It covers class definition, instance objects, method objects, class and instance variables, inheritance, and more.
- How it was useful: The use of classes were important in our code and when faced with some challenges, this site came in handy. There were many examples and clear explanations included.

4. Plain English - Learn Classes by Making a Game in Python

- Link: <https://plainenglish.io/blog/learn-classes-by-making-a-game-in-python-5aedee38fc18>
- Summary: This article gives a tutorial on learning Python classes through the development of a game. It walks you through the process of creating a game using classes, and explains key concepts along the way.
- How it was useful: It had nice examples that made it easier to develop the code, it served as a good reference.