Course-End Project
Building a Python Adventure Game with GitHub Copilot

Overview

In this project, you will build a text-based adventure game using Python and GitHub Copilot. The game will allow users to explore different locations, make choices, and complete a simple quest, focusing on fundamental Python concepts such as variables, lists, loops, conditionals, and functions. You will use GitHub Copilot to assist in writing functions and improving code efficiency. By completing this project, you will reinforce core Python skills in a fun and interactive way.

Tool: VS Code with GitHub Copilot extension

Dataset: None

Instructions

- Read the situation, tasks and actions, and result sections carefully to understand the assignment thoroughly
- Follow the tasks and actions provided below to develop the game
- Create a brief report (PDF) summarizing how GitHub Copilot assisted in writing and optimizing the code, the key challenges faced, and any enhancements or modifications made to the original game structure
- Complete and submit your assignment via the Learning Management System (LMS)

Situation

You are a Python programmer looking to practice writing functions and working with conditionals. You decide to create a text-based adventure game where users explore different locations, encounter challenges, and complete a quest. You will use GitHub Copilot to help generate and refine your code to speed up development. The final product should be an interactive command-line interface (CLI) where players can make choices and navigate the game world.

Tasks and actions

In this adventure game, the player takes on the role of an explorer searching for a legendary treasure hidden in an ancient land. The player must navigate through various challenges, make strategic decisions, and overcome obstacles to complete their quest successfully. The game consists of multiple decision-based scenarios where each choice leads to a different outcome. Some choices will move the player closer to the treasure, while others might result in failure or setbacks. The player will navigate different locations—such as a dense forest and a mysterious cave—while making strategic choices that determine their success.

1. The game starts by introducing the player's quest and asking for their name.
2. The player is given an initial choice of exploring different paths (a dark forest or a mysterious cave).
3. Each choice will trigger a new event, leading to more decisions.
4. The player must think critically and choose wisely to advance toward the treasure.
5. The game ends in one of the following three ways:
    - Winning: Finding the treasure
    - Losing: Making a poor decision that ends the adventure
    - Restarting: Choosing to replay the game after an unsuccessful attempt

The objective is to find the treasure by making the right choices and overcoming obstacles, successfully navigating the adventure. If the player makes poor decisions, they may lose their way or fail the quest.

Task 1: Set up the project
Actions:

- Open VS Code and create a new folder for your project
- Inside the folder, create a new Python file named adventure_game.py
- Add an inline comment to describe the purpose of the script
- Run a simple print statement to confirm that the setup is working

Task 2: Create a function to start the game
Actions:

- Define the function *start_game()* to display the game introduction
- Ask the player for their name and store it in a variable
- Provide the player with an initial choice (explore a forest or enter a cave)
- Use GitHub Copilot to generate the function body

Task 3: Create the forest path
Actions:

- Define the function *forest_path()* that describes the forest scenario
- Provide the player with choices (follow a river or climb a tree)
- Use an if-else structure to handle player choices

Task 4: Create the cave path
Actions:

- Define the function *cave_path()* that describes the cave scenario
- Provide the player with choices (light a torch or proceed in the dark)
- Use conditionals to determine the outcome

Task 5: Run the adventure game
Actions:

- Call *start_game(*) to begin the adventure
- Ensure the program runs in a loop until the player completes their journey
- Provide an option to restart the game after completion

Result

The result will be an adventure_game.py script that runs a fully functional text-based adventure game. The final project (script and report) should be submitted on the LMS.