Project 03: Adventure Game

Regarding Milestones:

A milestone or milepost is a marker placed along a highway to tell you how far you have come, or to indicate your progress toward your destination. In software development projects, a Project Milestone marks a specific point along a project timeline.

To help you make progress toward finishing this project, you will complete part of the program during the middle of the week and submit a "Milestone Submission." Then, by the end of the week, you will complete the program and submit the finished version.

You should read over the complete project description first. Then, at the bottom of this page, you will see which features are required for the milestone and which are required for the overall project.

Overview

In a text-based adventure game, the user is presented a scenario with different options. Depending on the option they choose, they have different consequences, which in turn present different choices for the next action.

Consider the following example:

> *You are walking through a dark forest and find two items: a MATCH and a FLASHLIGHT. Which one do you want to pick up?*

The user can then type "match" or "flashlight".

If the user types "match" they may see a response such as:

> *You pick up the match and strike it, and for an instant, the forest around you is illuminated. You see a large grizzly bear, and then the match burns out. Do you want to RUN, or HIDE behind a tree?*

Whereas, if the user typed "flashlight" in response to the original question, they may see a response such as:

> *You pick up the flashlight and turn it on. You see the pathway lit up in front of you, but you thought you also heard something off to the side. Do you want to FOLLOW the path or LOOK in the trees for the thing that made the noise?*

Your choice at each step of the game determines what you see next and the options you have available at that point.

Instructions

Create your own text-based adventure game with at least three levels of choices. It's up to you to determine the scenarios, the choices, and the consequences.

Keep in mind the following guidelines and requirements:

1. You need to have at least three levels of scenarios with possible choices. (This means that every time the user plays the game they should answer at least three questions because every branch of the game should go three levels deep.)
2. Scenarios that follow an earlier answer should be handled with nested `if` statements. In other words, the body/block of the first if statement will contain a print statement and then another if statement inside it.
3. At least one of your scenarios must have more than two possible choices.
4. In each prompt, write the choices in ALL CAPS, so that the user knows which words are possible responses to choose.
5. Responses should be words (strings) rather than numbers.
6. When checking the users responses, you should match on the keyword, regardless of the uppercase/lowercase used in the response (e.g., "match", "MATCH", "Match" should all work).
7. Making different choices should take you to different scenarios. (You shouldn't have the same result for different choices.)
8. Choices should only work for the correct question.
   In other words, if one scenario resulted in choices of Run/Hide, but another resulted in choices Follow/Look, you should not be able to respond with "Follow" to the question that asked for Run/Hide.
   A good way to accomplish this is to have a series of nested `if` statements. (That is, the print and then the next `if` statement will be within the body/block of the first if statement.)

9. For each question, you should provide an "else" clause to handle the case that the user tries to type something other than the possible choices. It is up to you how to handle this case.

Showing Creativity and Exceeding Requirements

Obviously, you'll show some creativity by customizing the prompts and choices. To receive points for the "Shows creativity and exceeds requirements" portion of the rubric for this assignment, you need to add something additional to the framework of the game. For example, you might add even more levels or you might have more choices at each level. You might add hidden choices or trick questions. Have fun with this and see what you can do!

If you've already learned other programming concepts (for example, loops, lists, etc.) you are welcome to use those concepts here as a way to make show creativity and exceed requirements.

Considering Sharing this Program

For this assignment specifically, an additional way that you can receive points for showing creativity and exceeding requirements by showing your program to at least two other people to have them play it. After they play it, briefly show them your code and explain how it works. Then, describe your experience in your comment at the top of the program.

> **Important:** In order to receive credit for showing creativity, you must include a comment at the top of the program that describes in 1-2 sentences what you have added.

Milestone Requirements

By the middle of the week, to help make sure you are on track to finish the assignment, you need to complete the following:

1. The program should work correctly for the first question and possible choices, and displays a follow-up response to each choice (including an else condition).
    - For the milestone, you do not need to implement any additional scenarios/questions, you only need the first one.
2. Create a design for your complete game.

- Prepare for the rest of your game by deciding on all the possible prompts, choices, and responses that the user might see. You should design the complete game, including else conditions. Then, to finish up the assignment for the next lesson, you'll just need to code up all of these options.
- Feel free to write this design out on paper or in a document (Word, Google Docs, Powerpoint, etc.), whatever is most convenient for you. You should write each possible scenario along with its choices. Then, for each choice, write the resulting scenario with its choices, etc.
- You are not required to submit this design to Canvas, but you should complete it as part of your Milestone to make sure you are prepared to finish the program.

Final requirements

At the end of the week, you need to finish your implementation. If you change your mind slightly from your original design, that's okay.

Submission

There is an assignment submission in Canvas for both the Milestone and the Final Submission. Make sure to:

- Return to Canvas to submit:
  1. The milestone submission (midweek)
  2. The finished project (end of week)

Up Next

- Ponder: Week 03 Reflection

Other Links:

- Return to: Week Overview | Course Home