

## 05 Prove: Milestone - Adventure Game

**Regarding Milestones and Assignments:** Just as with the Prove assignments for Lessons 03 and 04, the Prove assignments for Lessons 05 and 06 will be working toward completing the same assignment. At the end of Lesson 05, you will complete part of the overall program, and then finish it in Lesson 06.

Read over the full requirements first. Then, at the bottom of this page, you'll see which features are required for the milestone at the end of Lesson 05.

### 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:

01. | You need to have at least three levels of scenarios with possible choices.
02. | At least one of your scenarios must have more than two possible choices.
03. | In each prompt, write the choices in ALL CAPS, so that the user knows which words are possible responses to choose.
04. | 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).
05. | Making different choices should take you to different scenarios. (You shouldn't have the same result for different choices.)
06. | 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.)

07. 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.

## Make it your own

---

Obviously, you'll make this assignment your own by customizing the prompts and choices. To achieve the grade category of "Make it your own" for this one, 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 this your own.

## Extra credit

---

For this assignment, you can earn **+5 extra credit points** 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.

## Milestone Requirements

---

At the end of Lesson 05, to help make sure you are on track to finish the assignment, you need to complete the following:

01. The program is working 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.
02. Create a design for your complete game.

This means you should have a document (Word Document, Google Doc, etc.) where you type out each possible scenario along with its choices. Then for each choice, you type out the resulting scenario with its choices, etc.

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.

## Final requirements

---

At the end of Lesson 06, you need to finish your implementation. If you change your mind slightly from your original design, that's okay.

## Submission

---

For the milestone submission, complete the associated I-Learn quiz that asks you to declare which of the milestone steps you successfully finished.

In the next lesson, you'll finish the remaining requirements.