

Gamebook I

psp-03-02

1 Overview

A gamebook is a work of fiction that allows the reader to participate in the story by making choices that effect the outcome of the story. One of the most popular series in this genre is "Choose Your Own Adventure" by Bantam Books. In this project you will create your own gamebook adventure.

2 Learning Outcomes

By the end of this project students should be able to:

- read and write programs with if-elif-else constructs;
- read and write programs with nested control structures;
- read and write programs that use type conversion;
- read and write programs that use comparison operators;
- work effectively with a partner using pair-programming;
- write an effective report that describes the students' problem solving process.

3 Pre-Lab Instructions

Do this part before you come to lab:

While the gamebook should be written in the lab with your partner, there are a few things you should do to prepare before coming to the lab:

- Read Problem Space Chapter 3: Primitive Expressions
- Write a short answer to the following question: "What do tabs indicate in Python?"
- Come an idea for a gamebook. Try and be unique and come up with ideas that will be interesting and fun.

- Write down a short paragraph for your idea prior to coming to lab. Be prepared to show this to the lab aide at the beginning of lab.
- If you have never played a gamebook before, try one before class:
<http://chooseyourstory.com/story/the-firebird>

Part I. Lab Instructions

Do this part in lab:

Step 1. Discuss your gamebook ideas with your partner. What are the important features? What makes it fun?

Step 2. Plan the story for your gamebook on paper.

There are three requirements for this lab:

1. There must be at least two decision points in every path through your story.
2. There should be at least one possible path through your story in which you encounter three or more decision points.
3. At least one decision point must have at least three possible consequences.
4. At least one decision points must involve a numeric comparison. Your story choices don't have to involve only strings!

Write an outline of your gamebook in tree form on a piece of paper. If you are at all confused on if your design meets the requirements, ask a lab aid.

Step 3. Write the Python code to implement your gamebook. You will need to use print statements, if statements, and simple comparisons. Note that your if statements will be nested! That is, one if statement will be inside another. Here's a simple gamebook fragment to give you some ideas:

```

1 print("It is cold in this Egyptian tomb and you")
2 print("aren't sure how you got here, but you")
3 print("can't think about that because a Mummie")
4 print("is chasing you!")
5 print("")
6 choice = input("Do you DODGE, RUN, or remain STILL?")
7
8 if choice == "DODGE":
9     print("The Mummie careens to your right but")
10    print("as he does you drop your matches.")
11
12    matches = input("How many do you pick up?")
13    matches = int(matches)
14
15    if matches > 2:

```

```
16         print("Why did you try to collect so many?")
17         print("The mummie has grabbed you!")
18     else:
19         print("You take the matches and run!")
20 elif choice == "RUN":
21     print("Hopefully you get the idea ..")
22 else:
23     print("You keep adding scenarios to the story ..")
```

Step 4. Test your story and when you are happy with it demonstrate it to an instructor.

When you have completed the lab run the pep8 program against your code until all formatting errors have been corrected and your code is PEP 8 compliant. See the Getting Started lab if you need instructions on running the program.

4 Lab Report

Each pair of students will write a single lab report together and each student will turn in that same lab report on BBLearn. Submissions from each student on a pair should be identical.

Your lab report should begin with a preamble that contains:

- The lab assignment number and name
- Your name(s)
- The date
- The lab section

It should then be followed by four numbered sections:

1. Problem Statement

In this section you should describe the problem in **your** own words. The problem statement should answer questions like:

- What are the important features of the problem?
- What are the problem requirements?

This section should also include a reasonably complete list of requirements in the assignment. Following your description of the problem, include a bulleted list of specific features to implement. If there are any specific functions, classes or numeric requirements given to you, they should be represented in this bulleted list.

2. Planning

In the second section you should describe what planning you did in order to solve the problem. You should include planning artifacts like sketches, diagrams, or pseudocode you may have used. You should also describe your planning process. List the specific data structures or techniques you plan on using, and why.

3. Implementation and Testing

In the third section you should describe how you implemented your plan. As directed by the lab instructor you should (as appropriate) include:

- a copy of your source code (Submitted in BBLearn as a .py file)
- a screen shot of your running application / solution
- results from testing

4. Reflection

In the last section you should reflect on the project. Consider different things you could have done to make your solution better. This might include code organization improvements, design improvements, etc.

You should also ask yourself what were the key insights or features of your solution? Were there alternative approaches or techniques you could have employed? How would these alternatives have impacted a different solution?

5. Partner Rating

Every assignment you are required to rate your partner with a score -1, 0 or 1. This should be submitted in the comment section of the BBLearn submission, and not in the report document. You do not have to tell your partner the rating you assign them. A rating of 1 indicates that your partner was particularly helpful or contributed exceptional effort. A rating of 0 indicates that your partner met the class expectations of them. Rating your partner at -1 means that they refused contribute to the project, failed to put in a resonable effort or actively blocked you from participating. If a student recieves three ratings of -1 they must attend a mandatory meeting with the instructor to dicuss the situation, and recieving additional -1 ratings beyond that, the student risks losing a letter grade, or even failing the course.

Colophon

This project was developed by Dr. James Dean Palmer of Northern Arizona University. Except as otherwise noted, the content of this document is licensed under the [Creative Commons Attribution-ShareAlike 4.0 International License](#).