Mad Libs

psp-02-02

1 Overview

A Mad Lib is a game where you use a template to tell a story. The game works by having someone fill words into a story template with minimal context or information about the story. The Mad Lib might ask the player for an adjective or a proper name, with no hint of how the word will become part of the story. The player must decide on all the words before the story is read. The fun and humor of the game is discovering how those words are used in the story.

2 Learning Outcomes

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

- write, save, and evaluate simple programs;

- read and write programs with string literals;

- read and write programs with simple function calls (e.g., input, print);

- break up simple problems into multiple steps;

- 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 Mad Lib game 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.

- If you have never played a Mad Lib before, try one before class: http://www.eduplace.com/tales/

- Come up with your own Mad Lib and bring it to lab. Be prepared to show this to the lab aide at the beginning of lab.

4 Lab Instructions

Do this part in lab:

Step 1. Discuss your Mad Lib from your pre-lab with your partner. What are the important features? What makes it fun?

Step 2. Use a sheet of paper to design a new Mad Lib with your partner. This should look like a Mad Lib you might play in a book. This template is your plan for this project.

Step 3. Create a file called "madlib.py" and convert your Mad Lib into Python code. There's really only three things you need to know:

1. how to get user input and store it in a variable,

2. how to concatenate (join) strings, and

3. how to print strings.

Here's an example that illustrates these processes:

```python
first_adjective = input("Type an adjective: ")
first_noun = input("Type a plural noun: ")
first_name = input("Type a friend's name: ")

print("My First Mad Lib Story")
print("=========================")
print("There once was a " + first_adjective + " boy " +
      "who liked " + first_noun + ".")
print("His best friend was named " + first_name)
```

Your Mad Lib should be a lot longer. You should have at least 12 questions and 10 lines to your story. But feel free to add more and most importantly be creative!

Step 5. Run the application to make sure your Mad Lib works. If it doesn't, see if you can figure out how to fix it. Your partner and the lab staff are here to help.

Step 6. Take a screenshot of your window. In Microsoft Windows you can do this by selecting the window and press ALT and Print Screen.

When you have completed the lab run pep8 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, or the pep8 documentation found here.

5   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 funtions, classes or numeric requirements given to you, they should be represented in this bulleted list.

2. Planning

In the second section you should describe your planning process. Identify any input and output required. Develop an algorithm (or listing of steps) to solve the probelm. Identify variables you'll need and coding elements you'll use. List specifcs about data structures, functions, and/or classes you plan to use and why.

3. Implementation and Testing

In the third section you should describe how you implemented your plan and discuss how it went. You should also include:

- a well-commented copy of your source code with the appropriate header (i.e., lab assignment number and name, author names, date, and lab section) submitted in BBLearn as a .py file

- a screen shot of your running application / solution

- a screen shot showing pep8 compliance

- test results (as needed)

4. Reflection and Refactoring

In the last section you should reflect on the project. Make sure you solution meets all the requirements identified in Step 1, and then summarize the key aspects of your approach and the coding techniques you implemented. Also, consider things you could have done to make your solution better. This might include code organization improvements, design improvements, etc. Ask yourself if there were alternative approaches or techniques you could have employed? How would these alternatives have impacted a different solution? Discuss how you have (or could) refactor your code based on these observations.

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 indcates 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. Be sure to include a comment along with any -1 or +1 ratings. 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.