CS126L Gamebook Lab 2

Porter DeFazio and Daniel Williamson

24 September 2015

Lab Section 3

1. **Understanding the Problem:**

Use python to code a gamebook after sketching on paper what your finished product should look like.

Requirements:

* Two decision points in every path.
* At least one path with three or more decisions
* At least one decision with three consequences
* One numeric comparison, not simply strings

Important Features:

* Use if-else statements
* Using nested statements

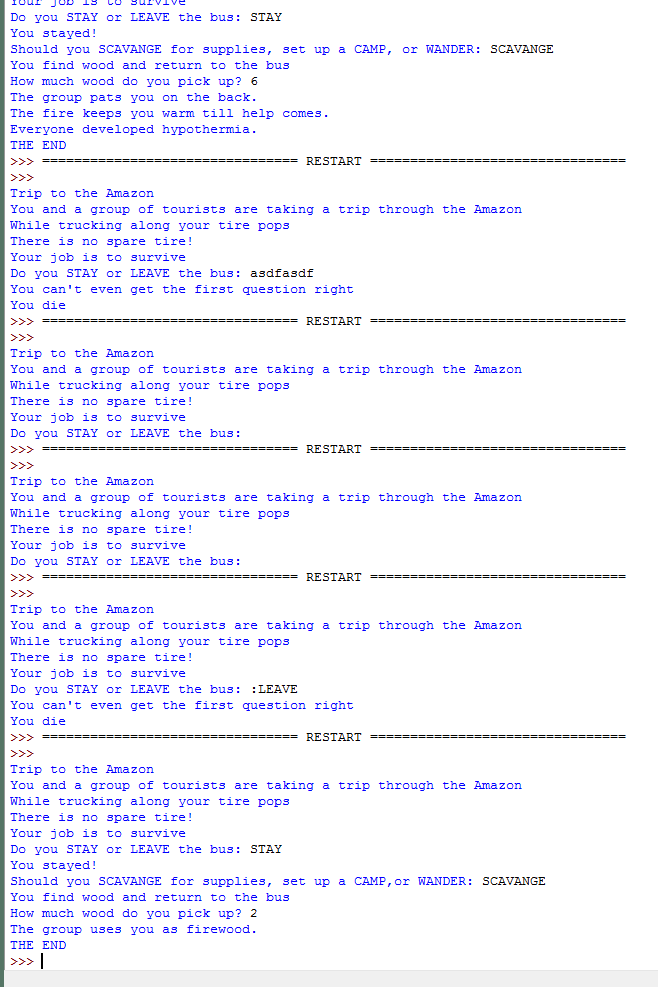
1. **Planning the Solution**

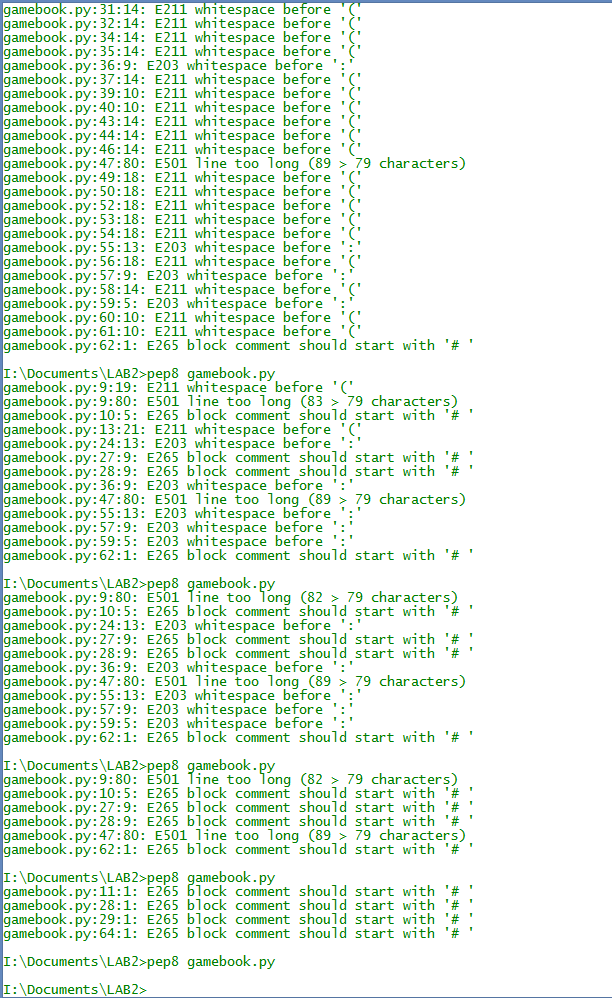
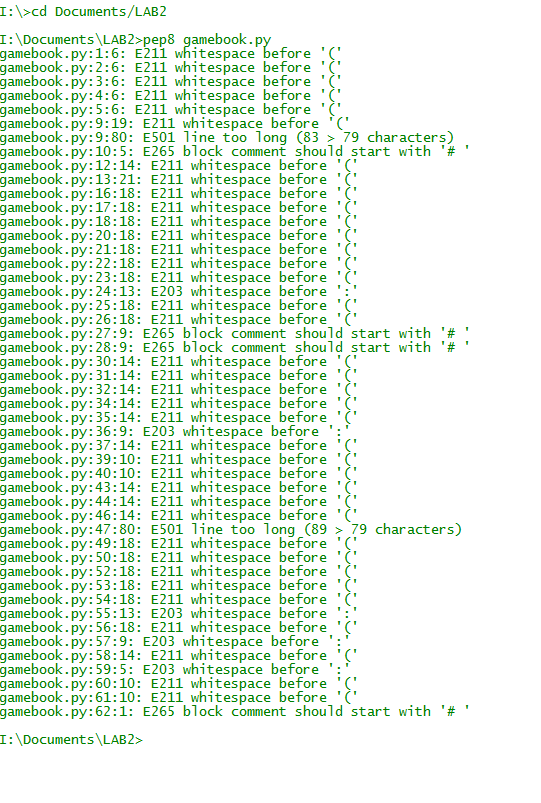
In order to determine exactly how we would implement the code we reviewed the sample code. This enabled us to recognize exactly what we needed to implement in the code as well as refresh our memory on using if-else statements. We also roughly sketched what we wanted to happen in the story - including the choices/paths.

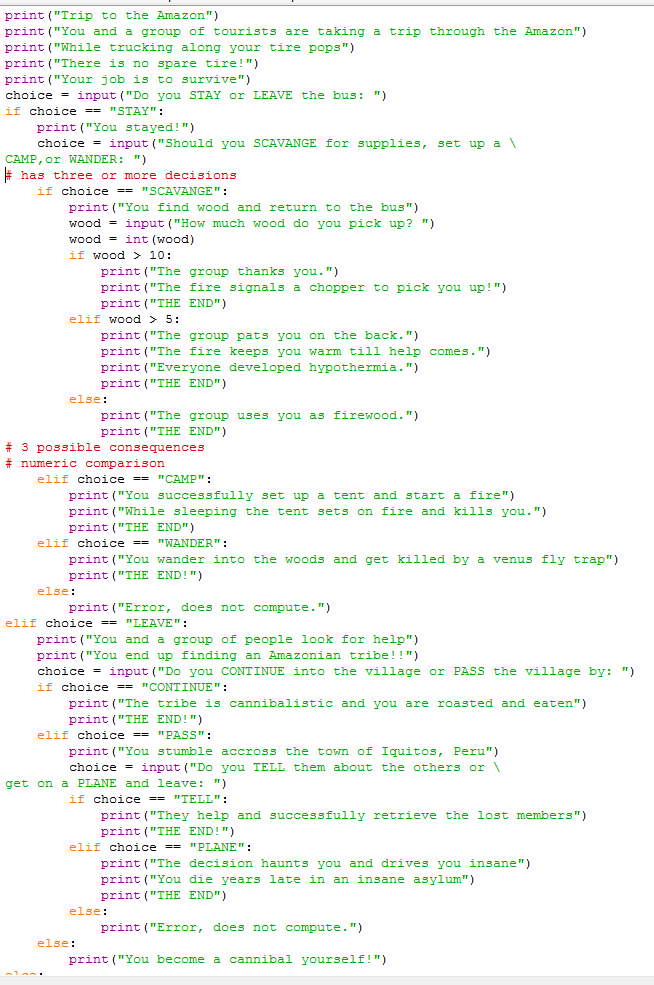
We planned for the code to meet the base requirements but also to take them a little further. We also planned for including a “The End” so there would be an actual official end to the story. Alongside this, we planned roughly where we would include comments.

1. **Implementing and Testing**

We implemented the plan by coding the program and by following the plan that we had made. We stopped at different points in order to test our code as well as to ensure that it would execute without error - and so we could locate the problem faster if one occurred. Upon completion, we executed the code a few times in order to test the various paths that one could take. Next, we ran the code through PEP8 and we had a large return of errors. The main issue was the fact that we put a space between print and the parentheses. The minor issues were that two line were too long and that the comments were set up wrong. These issues were numerous, but they only took a few minutes to repair.







1. **Reflecting on the Solution**

Overall, the lab went extremely smooth with only some slight bugs with the code. In hindsight, it would have been a lot easier if we would have ensured that the code was PEP8 compliant as we went along so we could catch the little mistakes we made such as the spaces after ‘print’. We also could have written the code out on paper so that it was easier to apply rather than constantly having to check as we coded the program. We finished the code with some time to spare and it executed beautifully, however, it could have been slightly easier if we would have known about the whitespace after the print being noncompliant.