Final Project Report

Kadin McWilliams

Mason Anderson

12/12/2023

Fall 2023

CS 150

**Abstract**

Our goal was to create an entertaining experience for our users through an interactive novel. To achieve our goal, we used mainly functions and the shutil and textwrap modules. Through the philosophy of our architecture and correct use and organization of our functions we achieved our goal. We created a fun and interactive user experience that would entertain anyone who played it. We learned more about how strings work and how to manipulate them. We also learned about game design and how important organization is when coding.

**Introduction**

For our project we decided that we wanted to create a choose your own adventure game. Where the player would go through an interactive novel and have their experience change based on the decisions they made throughout the game. The purpose of this project is to provide entertainment for the user. Many people who enjoy reading wish that they could interact with the story they are reading. Our game allows users to be able to interact with the story they are reading and give them that experience they are looking for.

**Methodology**

To create our project, we used various tools to complete it. Our main tools that we used are Anaconda and Jupyter Notebook. We used Anaconda to launch the Jupyter Notebook where we could do all of our coding for the project. Another resource we used was a pen and some paper. We needed to be able to design our story in a way that was easy to follow so that when we were coding, we would be able to only focus on the coding aspect and not the story design at the same time. In the developmental process the easiest part was testing for potential inputs since there were so few from the user. The user could only pick from one or two options, so testing was easy just to make sure that the options worked and went to where they were intended to go. However there had to be a lot of testing to make sure that the text was lining up correctly and everything was being spaced in a format that looked readable.

When looking at the coding aspect we initially tried to just code things in a way that would require more work but soon realized that it would be inefficient. We then spent a good amount of time setting up automation for things that we did not want to do so that we could save a large amount of time on the coding. To refine our code, we tested all possible outcomes for each option you could pick in the story and made sure to thoroughly check that not only the formatting was lining up but the dialogue as well.

**Design and Implementation**

The architecture of the code was the largest and most important part of this project. While anyone could hard code the whole project, we wanted to make our project in a way that reflected our skills that we had learned and saved a large amount of time. The philosophy of our design was to make our code as readable as possible so we could make edits easier. If we had a large amount of print statements, then finding who was saying what would be monotonous.

In order to accomplish this task, we set up several functions so that we could easily identify the character. The other part of the function was so that we could type how we would normally type without having to worry about formatting or thinking about how to make the text match up with the code. The last major aspect we wanted to account for was making the formatting automatic so that we didn’t have to manually find where to put a new line in our string. We found that having to stop and think about where to put the new space in our code while writing was hindering us from writing.

Our main narration functions were narrator(), player(), dylan(), and generaldia(). narrator(), player(), and dylan() are all characters that have recurring lines so making functions specifically for them made sense for the design of our code. The generaldia() function allows us to format any character in our story so that we can simply input the name of the character and what they are saying. We decided to add this function instead of making a new function for each character because creating a new function would be more time consuming.

We also implemented two new modules in our code which are shutil and textwrap modules. These two modules allowed us to fix our problem with some of the words being split into two because they reached the end of the max character for the text line. Using these modules in combination with our character functions allowed us to easily create all the dialogue in our game.

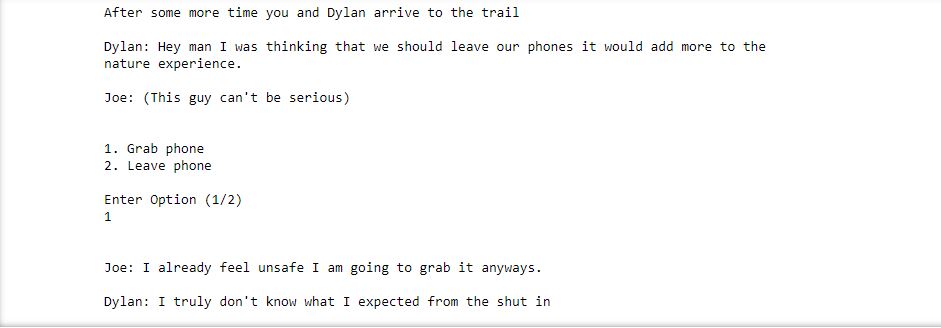
**Challenges and solutions**

Our main challenges that we faced were organization and formatting our text so that it was easy to read. The organization was a challenge as jumping from different parts of the story in a non-chronological order made us lose track of where we were in the story. This meant that when we were working on sections, we would not know which function the story would lead to so we might accidentally write a part in the wrong part of the story.

To fix the problem of not knowing where we are in the story was that we used comments. Comments proved to be quite useful as they enabled us to easily see where in the story the code was and was going so that we didn’t have to spend extra time reading and understanding the code. To fix the other problem of the text cutting of mid-sentence we used the shutil and textwrap modules. The textwrap module automatically wrapped the text for us so that we did not have to manually put in the new place lines. The shutil module allowed us to get the terminal size so that our textwrap function could automatically wrap it to the optimal length.

**Results**

We were able to produce a story that was not only entertaining but was easy to read as well as worked the way we intended it to. Our story functioned exactly the way we wanted to, and I felt like we set up the user experience to be enjoyable, simple, and intuitive. The options are clear on what the action being taken is and the instructions are clear and concise. I think that anyone can play this game, even people who know nothing about computers, and have an enjoyable time playing it. Our code and design philosophy worked exactly how we wanted it to, and we were able to achieve the goal we set out to do.



**Discussion**

Our project, even though the coding may have seemed simple, really helped with our knowledge of how strings work. Having to problem solve and learn how to format the strings helped with our understanding of them as a whole. For example, I didn’t know that you could concatenate an f string with a normal string and the function still come out properly. We could have definitely been more organized with how we presented our code. When creating our functions, we didn’t do them in any logical order and just added them as we worked on each section. Going through and organizing the functions in a logical order would have made working on the code easier as we reached a larger scale of what we were producing. We also learned how to implement functions better into our code instead of hard coding each statement.

**Conclusion**

In summary our project to create an entertaining interactive novel was a success. We managed to achieve our goal through careful planning of how we wanted our story to work. To keep our code organized and consistent we made use of functions so that we could easily add, change, and understand the code we were writing. Comments helped with the organization of the code too. Our experience with this project helped with not only our understanding of a fundamental part of python, but also showed through experience that the creative process of creating something using coding is enjoyable.

**References**

Dunes. (2022). How to make Python start a new line rather than cutting words off. https://stackoverflow.com/questions/73677627/how-to-make-python-start-a-new-line- rather-than-cutting-words-off