|  |  |
| --- | --- |
| Full Name | John Elehwany |
| Student ID | 1934007 |

Reflection

This project was a really fun opportunity to learn new ways of using new tools in Python. Not only was it innovative in new techniques but it was also fun to try and make your own story out of this. I think the outcome of this project has been good, but maybe a bit shorter than I wanted it to be. And there were definitely issues that arose along the way while creating this.

When I do these projects, I am always referencing back to the algorithm, and then doing all of my work for the project based off of that. This was no exception here; but once again, my initial design was tweaked a lot when trying to write the code and testing it. The biggest change from initial to final was the values of the scores, the confidence multipliers, and the points that you could earn from each response. They were really unbalanced in the first testing of this project; either you could answer really well and do poorly, or you could do really poorly and still pass the presentation. The initial score and final score were tweaked a lot before reaching a good balance of the final product. It wasn’t really a programming issue, but rather something that was at the game’s core. It still was a problem nonetheless.

The biggest key feature I learned more about using here are the “while” statements. I used these in pretty much every prompt in the game. The while statements I implemented are used to check if the inputs the user provided are valid for not. They’re used in the multiple choice prompts, and they check the user inputted “A”, “B”, “C”, or “D”; and they’re also used in the written response to see if the user provided a number that 1) was a number and 2) a number that was in the required range. These are so incredibly helpful and I’m glad to have used them here.

Overall, I think the final project turned out well. I like the fake office story that I made for this game, and I’m glad that I’m learning new ways to code when making these personal projects.