

MIDS Fall 2021 Data Science W200

10/27/2021

Project 1 Reflection

Megan Martin

Instructions for testing:

- User's name: I did not place any limits on what a user could input here.
- Plot length/width: The program expects a positive integer input. If someone provides a string, it will return an error message and prompt user again.
- Plot dimension: In my final iteration of the code, I added a validation of the user's plot dimensions. In the instructions of the code, I state a user should enter at least a 4x4 foot plot. I added a new function to validate the dimension and if the dimensions are smaller than 3x3 foot, it prompts the user to enter again with a helpful message.
- Plants: If you elect to enter your own plant selections, it does check against the list. It should account for capitalization and space differences. If you enter a plant not on the list, it will prompt the user to enter again with a helpful message.
- Zip code: All zip codes should be 5 digits. This is checked in the program.
- All yes/no prompts can handle capitalization or space added and will prompt user to re-enter if yes or no is not entered.

Challenges and thoughts for future work:

Overall, I was able to complete the majority of functionality I originally described in my project 1 proposal. However, there are some refinements I would make in the future. First, my simulated diagram is not as effective of a simulation as I originally intended. I needed to consider scaling of the diagram (some plants are spaced every 3 inches and some are placed multiple feet apart). This posed a challenge to make a scaled version of the garden diagram that allows for large garden size input and is still viewable in the command line. Thus, the diagram is a general idea and doesn't necessarily represent actual plant placing. It turned out to be more of a visual representation of the number of each type of plant in each row.

One aspect of the project that I think is particularly helpful to the user is the saved garden plan that can be utilized for reference. This is a program that I will use to help plan my vegetable garden next year, so it is nice to have created something that is functional. In the future, I would figure out how to write the garden plan information to a word document. I would improve the diagram as stated above and use different symbols for each plant rather than just the first letter of the plant. As the instructor pointed out, this isn't a year-round garden plan. I did optimize the program for my specific use in my area, which is in a cold weather zone. To make this program more useful for warm-weather zones, I could include different information about additional planting cycles if a user enters a warm-weather zip code.

There are a few challenges I encountered. First, the identification of the absolute path to read the .txt zip code data and the JSON file causes issues when run in VS Code. However, the program works correctly from the command line. I looked into this and the problem is intermittent- I did not spend more time trying to investigate why absolute path identification was not working well in the VS code interface. The other big issue with this code is if a zip code is not found in the .txt zip code lookup file, it will result in an error. I tried to account for this issue in clear instructions to the user in the README.md file and in the future, I can implement a try except block in the code to provide a helpful message and prompt the user to re-enter again.