Ryan Demboski, Beizhe Shen
CS-126L Section 001
11/22/2019

Lab 9 Report

1. Problem Statement:

The task we were given was to accomplish the following:

- Download the stars.txt file for input in the program
- Import the sys module, and use the sys.argv[1] function to call in an input file on the command line (in this case, stars.txt).
- Import the turtle module, which will be what this script is centered around.
- Extract the x coordinate, y coordinate, and magnitude from the stars.txt file and create a script that draws the stars onto the turtle drawing grid space using various methods and functions from the turtle module.
- Use the equation min(round(10/magnitude+2),8) for the magnitude calculation.

2. Planning:

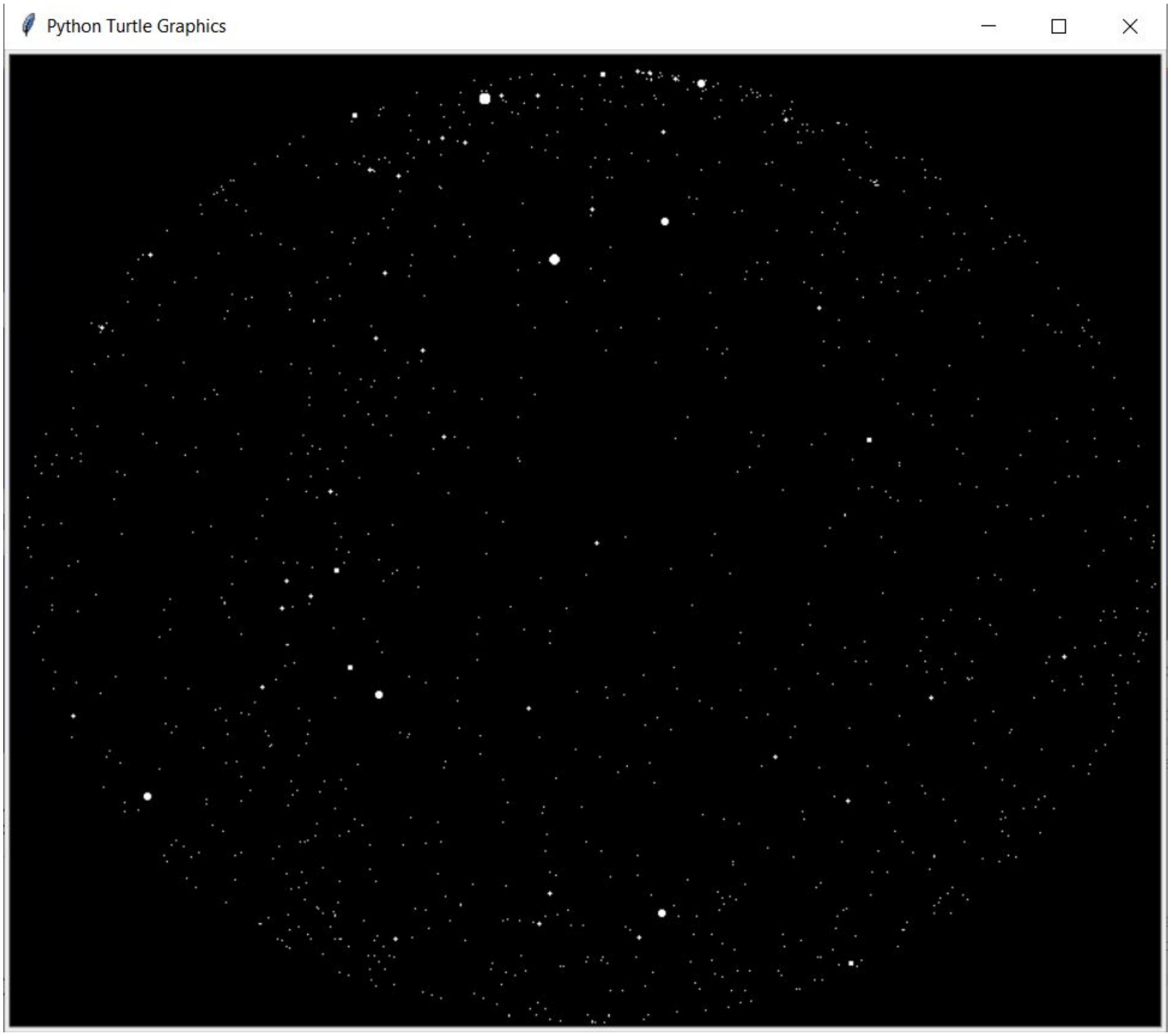
This program was very tedious and frustrating to figure out. The fact that we had no knowledge of the turtle module is what makes this problem so difficult. To plan this program, we had to heavily study the turtle documentation and pick out which specific types of functions we should use for this program. Also, we needed to come up with a strategy on how the input will be taken in from the stars.txt file, and how we will store that data. So, creating a function that can do all that for us seemed pretty logical. Also, we needed another function that calculated the magnitude and created the stars from our read_coordinates function. Lastly, we needed a way to call our functions at the end of the program to make it work as a standalone script with no manual input from the user outside of the input file argument in the command line.

3. Implementation and Testing:

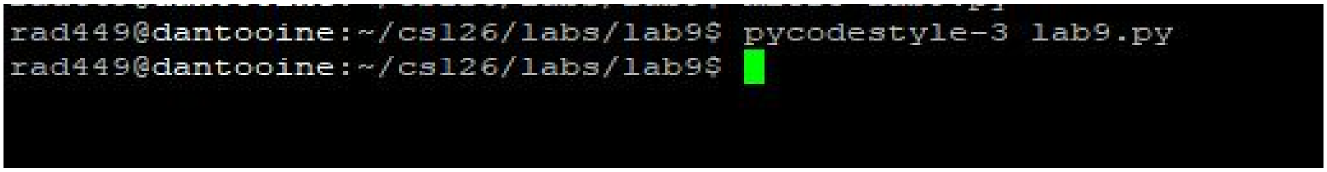
The first idea from our plan to be created was the read_coords() function. We had many attempts on finding the right way to get the input from the stars.txt file, but in the end decided to use two dictionaries. One for the x and y coordinates, and another for the magnitudes. This method made it easy to for loop through the data and assign each value to the correct place. The next piece to be implemented was our magnitude function, which took a very, very long time to figure out. Many hours were spent studying the documentation and figuring out which turtle functions to use in our program. It took a lot of effort, but the end result is exactly what he wanted.

After enough testing, we got our program to output what is shown below. (See Page 3).

Our Lab 9 Script Running:



Proof of PEP8 Compliance:



```
rad449@dantooine:~/cs126/labs/lab9$ pycodestyle-3 lab9.py
rad449@dantooine:~/cs126/labs/lab9$ 
```

4. Reflection:

Overall, we believe the script was created in a clever and effective way. Our use of the two functions, being the read_coords() and magnitude() functions, was a good solution to solving this problem, because those were really the two biggest things that needed to be accomplished in the mission statement. Having a function that reads the command line input file and stores the specific data we need in dictionaries makes it very simple for us to access the data in the magnitude function call at the bottom of our script. The use of for loops made our code shorter and much more effective in the two functions. We ran into many scaling problems along the way, and had issues with the stars being squares as well, but with enough research we were able to come up with solutions for these problems, by setting the world coordinates to [-250, 250] and using turtle.dot(star_size, ‘white’). It was a challenging but fulfilling lab in the end.