REPORT

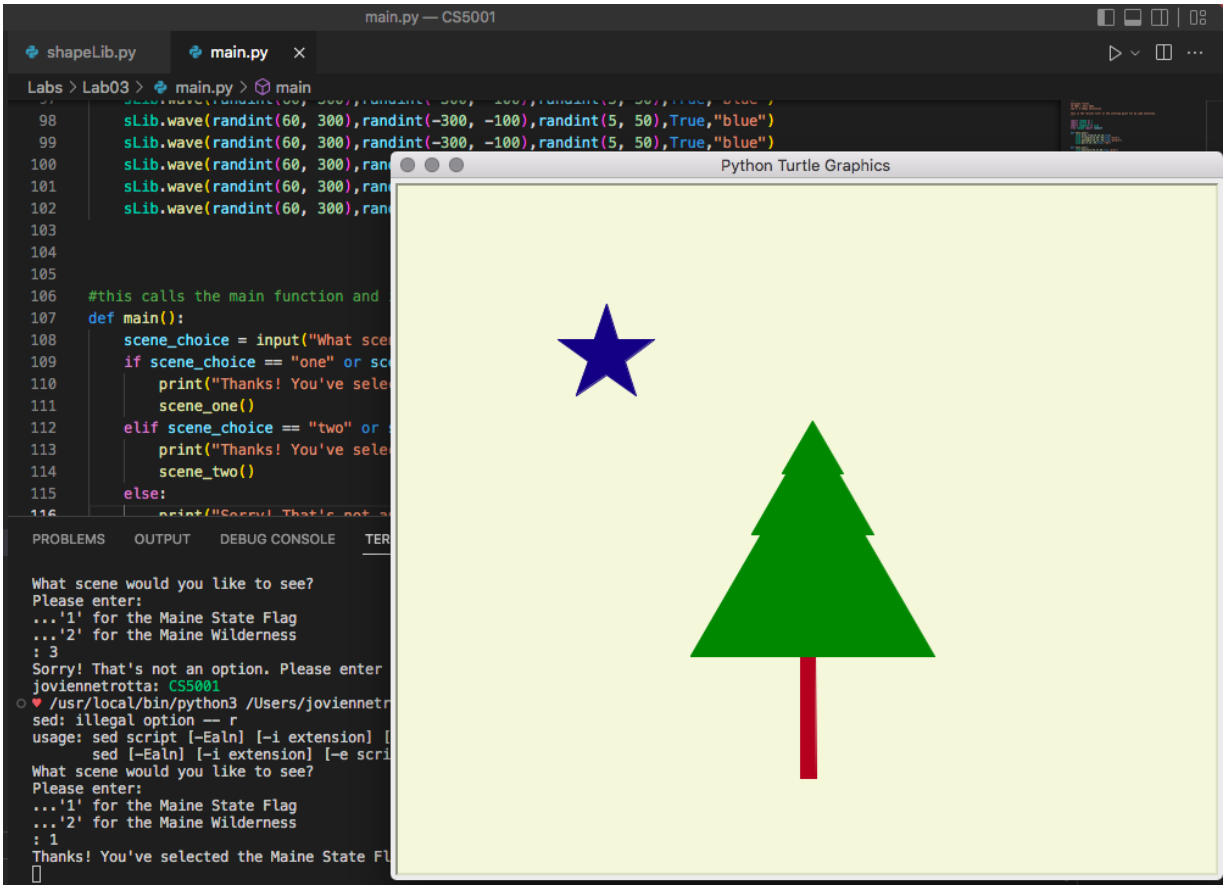
1. Problem Description

The purpose of this lab was to give us practice writing functions. Functions allow us to execute specific lines of code over and over again, which can help condense and organize our code. Parameters allow us to customize each function so that even though we're running the same function, we can still add little variations that make our scenes more interesting. The application of our functions in the case of this lab were to draw shapes, combine them into images, and then use those images to build a scene.

2. Screenshots

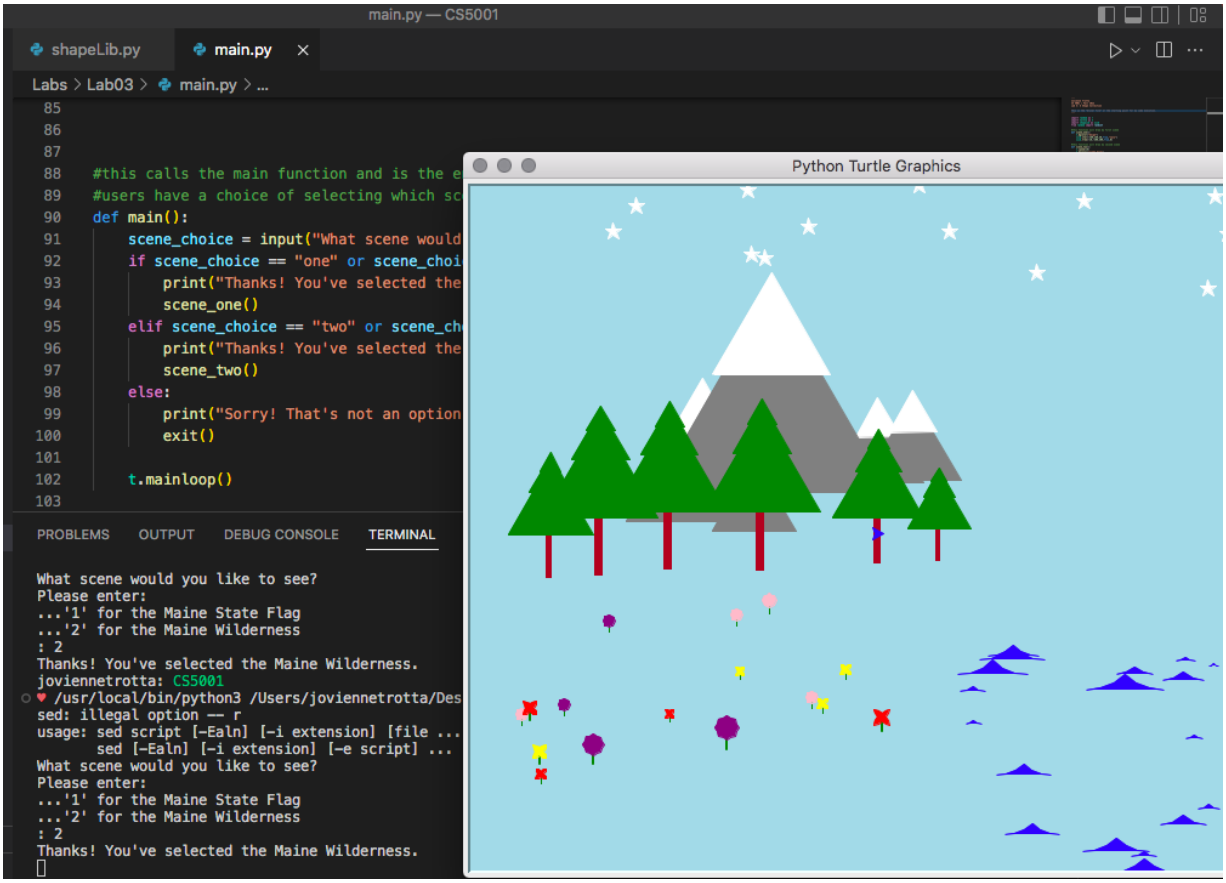
Scene 1

In this screenshot you can see the program executing scene one. I know this isn't the current Maine flag, but I always found it interesting that the older version of the flag seems more popular to display than the modern one. This was a fairly simple scene to code, but in order to create it I needed to first get the tree right. The tree was made of three triangles and one bar image, all of which needed to be centered. The star also required some trial and error to get right, especially since my original version didn't color fill properly. However, during that process I was inspired to create my flowers, which I use in the next scene.



Scene 2

This shows my second scene, in which I wanted to incorporate all of my favorite features from Maine (mountains, forests, starry skies, and the ocean). A lot of basic shapes were used to create the images in this scene. Additionally, I was able to get acquainted with the randit method and the parameters used within it, which allowed me to set a range for where each image could be featured on the scene window.



3. **Extensions**

For my extensions, I created a total of 12 different functions which execute different images of varying complexity. Some of these images I used in my final scenes, but some of them I created just for practice. I also added an extra parameter to my shapes, which allowed me to select a color for them. I experimented with setting a default color for some shapes, while others had colors that were selected when the function ran.

```
#this will draw one mountain
def mountain(x,y,scale,fill,color):
    triangle(x,y,scale,fill,color)
    triangle(x+(scale*0.25),y+(scale*0.5),scale*0.5,fill,"white")
```

4. **Reflection**

During this lab I became much more comfortable defining and calling functions. I also had the opportunity to practice importing functions from other files, which is something I was not able to do during the last lab. I understand now how functions can be used to create much cleaner, easier to read programs and am excited to use them more going forward.

5. **Acknowledgements**

I used geeksforgeeks.com to help me find some of the new turtle commands that I used in this lab, including randit, width, and home. I also used the Module 3 lecture notes and videos to help me get a better understanding of how to write a good function.

6. **Grading Statement**

I would give myself a 27 for this lab. I only did a few extensions, and they were pretty basic for the most part. I believe I achieved the main objectives for this lab, tested my functions, and wrote code that followed best practices. I've included the rubric before to help explain how I evaluated myself.

	Possible	Given
Shape Library		
goto, block, main (from recitation)	1	1
draw_circle	1	1
basic shape function 1	1	1
basic shape function 2	1	1
object function 1	2	2
object function 2	2	2
functions have fill options	1	1
Driver		
scene1 function	3	3
scene 2 function	3	3
tests created for each function	3	3
Misc		
Report (all or nothing)	4	4
Code Quality (correct indentation, comment blocks, variable naming, etc)	4	4
Not included in total possible:		
Shape library not used as a separate file	-15	0
Extensions (Not calculated without report)	4	1
Creative or went above and beyond	4	0
Code does not compile	-30	0
Late penalty	-6	0
Not implemented as requested	-30	0
TOTAL POINTS POSSIBLE out of 30	26	27