Project 1

How does your garden grow?

Due Dec 10, 2021 at 5pm

(5% bonus for submission before 5pm Dec. 9)

Project 2 allows you to use your skills in an even more creative way. This program may not be as long as some recent ones, but may take longer to develop due to the difficulty of the concepts involved.

Once again, in order to encourage you to start planning (and implementing) during the first week of the assignment, you have a **mandatory design review due before 5pm Friday December 3**. This design review consists of demonstrating your progress to date to a member of the course staff. More details about expectations and course staff office hours are available below.

Learning Goals

1. Design and implement a larger scale program than previously.
2. Make good choices about which functions to design and their inputs and outputs.
3. Do more complicated things with strings, lists, and loops.
4. Use recursion to solve a computing problem.
5. Use a complex Python module (graphics.py).
6. Create graphical output.
7. Identify and fix errors.

The Assignment

Create a recursive program that produces growing 2D plant models based on L-system growth techniques. L-sytems model plants using a grammar to simulate plant growth. Successive application of the production rules create a string which can be translated into geometry for display. In order to render your string, interpret the characters of the string as following:

- **F** Move forward a step of length d. Draw a line from the previous location to the new location.
- **X** Do nothing. For now, this can be ignored in the creation of geometry.
- **+** Turn left by angle A. The positive orientation of angles is counter-clockwise.
- **-** Turn right by angle A.

The angle A is given as part of the grammar. The twig length **d** should be initialized to a suitable value and decrease to 70% of what it is currently in each successive generation. Your program should be able to handle at least six generations of a plant type. Draw your plant to a window interactively using the graphics.py library.

Your program should be able to make a plant created from a number of generations, angle, and grammar file provided by the user. For example:

```
I will draw you a beautiful plant!
How many generations? 3
Angle between stems? 25.7
```

File with grammar: **plant1.txt**

should generate a plant three generations old using the grammar from plant1.txt and a rotation angle of 25.7 degrees.

The L-system rules will be contained in a text file. The first line will be the start symbol, with each subsequent line containing a rule for a particular symbol. For example:

```
X
X=F-[[X]+X]+F[+FX]-X
F=FF
```

For each generation, each symbol in the string should be replaced according to the relevant rule using recursive calls. More information about L-systems can be found here (https://en.wikipedia.org/wiki/L-system#Example_8:_Fractal_plant).

An example of a plant created by this system:

Drawing module

Once again you will use the module graphics.py (created by John Zelle) to create your display. The most recent version of the library can obtained at http://mcsp.wartburg.edu/zelle/python. Go to that site and download the file "graphics.py". The site also contains documentation in html and pdf format. A simple program to draw a single line might look like this:

```
from graphics import *

def main():
     win = GraphWin("My Drawing", 600, 600)
     aLine =Line(Point(150,150), Point (325, 440))
     aLine.setOutline("purple")
```

```
    aLine.draw(win)
    win.getMouse() # pause for click in window
    win.close()
main()
```

Some comments on coding standards

For this assignment, you'll need to follow the class coding standards, a set of rules designed to make your code clear and readable.

- You should be commenting your code, and using constants in your code (not magic numbers or strings).
- Any numbers other than 0 or 1 are magic numbers!
- Adhere to the coding standards by including function header comments for each of the functions (other than main).
- Including an explicit collaboration statement describing any people you may have talked with about this project or any sources you might have consulted.

Extra credit

Each of these additional capabilities are worth a possible extra five percent:

- Add randomness to twig length and rotation angle. Make your results as realistic as possible.
- Modify the colors of twigs so that older branches have darker colors than younger branches.
- Create a "flower" object using at least two different colors and change the grammar to accommodate it.
- Define your own interesting grammar and implement it.

Preliminary design review

The preliminary design review is worth 10% of this project. For the preliminary design review, you should write a description of the project design structure as you did in Lab 8. Specifically, you should sketch the contents of the program, breaking it down into **data** and **functions** and then order the functions into a program structure. Your sketch should contain the following elements:

1. Data: what data is required by the project
   Variable names w/ description of its purpose
   Data type of each variable (string, int, list, etc.)

2. Algorithms:
   High-level processes the program needs to have.

3. Full Program pseudocode:

   - Arrange the various functions in order that they need to happen.
   - Include loops and some notion of when they terminate.
   - Place the necessary functions in the loops.

- The pseudocode should contain a mix of high and low level items. Some aspects you might not solve until you start coding, so it's fine to give a high-level description.

Explain your project design structure to a member of the COS 125 staff during office hours before 5pm on Friday December 3.

How to turn in your homework

Turn in each program in its own file. When turning in your own assignment make sure to add your last name to the file name (for example: Rheingans_p2.py). You should not turn in the training file. If you do the extra credit, submit two files: YourName_p2.py with the base assignment and YourName_p2-xc.py with the extra-credit version.