February 16, 2025
Terence Xu 301381999

IAT460 A2 Report

Technical Documentaion:

The System architecture is built in Google Colab based on the L-system discussed in the lab. There are 3 main parts to the project, the initial setup of the L-system rules (create_l_system()), the rules which the L-system will run (draw_l_system()), and the calling of the functions of the L-system (input parameters).

The program uses the ColabTurtle library to create visual sequences based on a mapped pen on a 2D matrix canvas, I used some other functions/tools that weren't in the original lab example to increase the complexity of the visual result. Some functions include
t.pencolor to change the color of the pen
A few variables like **current_pen_size, color_index, colors** to help store values to and be manipulated.

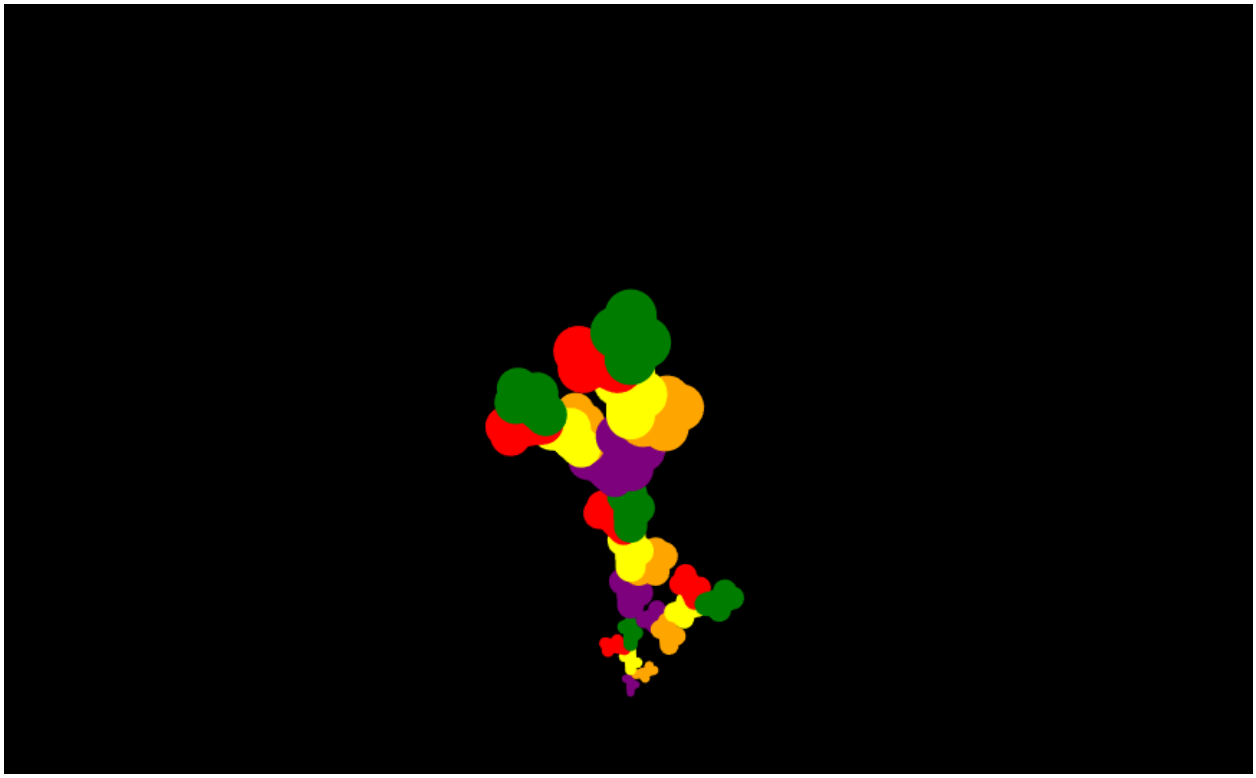
I implemented randomness within the L-system to help replicate a more "natural" look to the fractal pathway generation. This is implemented in the first draw_l_system() to slightly change the angel of the branch whenever the turtle pen is called and moved forward.

Some challenges That I came across was visualizing what I wanted to generate witht he fractals and trying to recreate them in code. I wanted to make the fractals look "natural", It took quite a bit of tweaking and tinkering with the source code to create what I envisioned. In the end the random library was a bit help in changing the angles randomly to create a more random visual to the fratals.

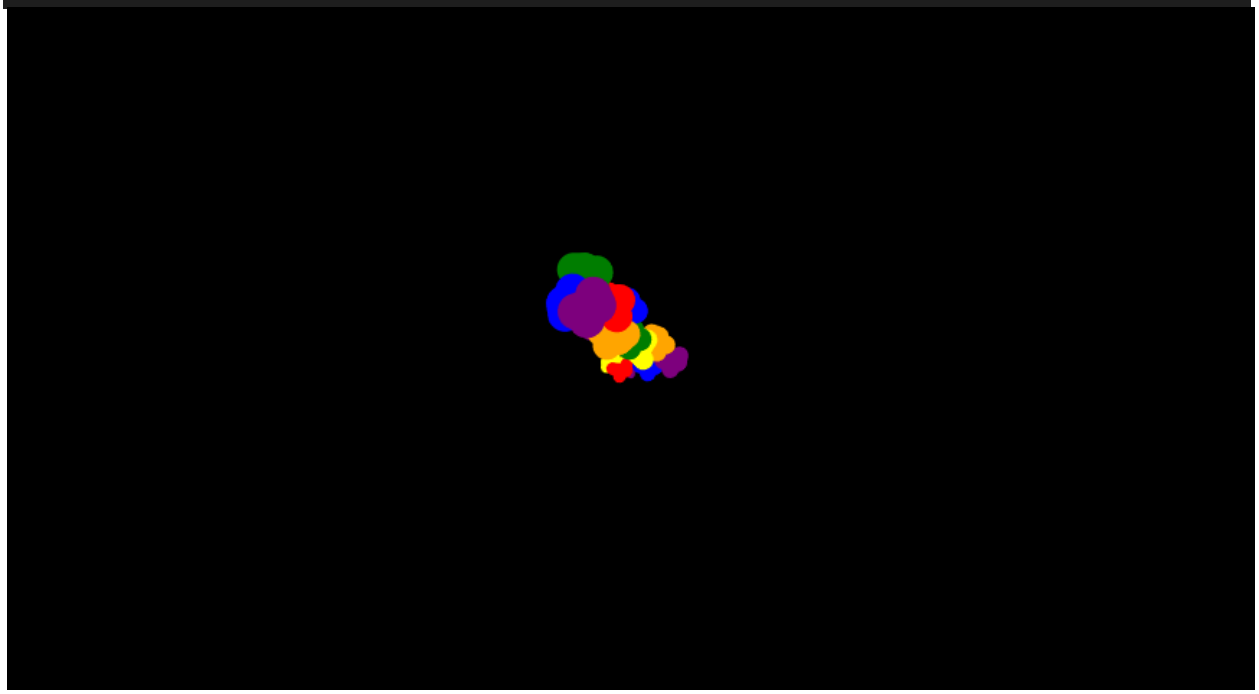
Analysis of Generation:

For the first L-system rule set, I wanted to create "paint blotches", so I used the "f" (lowercase) rule to create spaces between the individual branches. The resulting tree still looks pretty organized and not really all that random, but the colors are noticeably striking next to eachother and separates each branch. The increasing in thickness of lineweights also helps each branch turn more and more into a "blob" shape, similar to paint droplets.

In reflection, I would have tweaked the distances of the lowercase "f" to be wider or more random to help create even more separation between branches to make them look more like nodes of paint splashes.

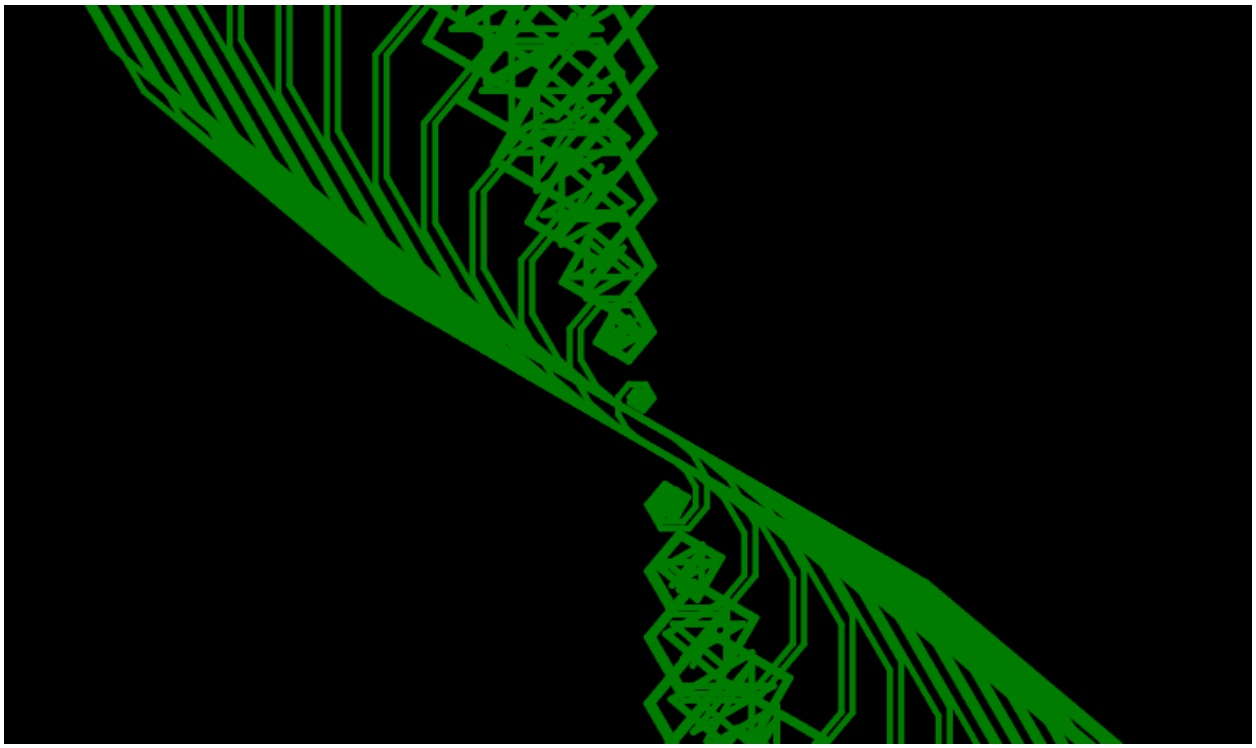


```
tree_rules = {"F": "fF[+F]F[-F]F"} #f creates gaps between branches to
create visual seperation
```



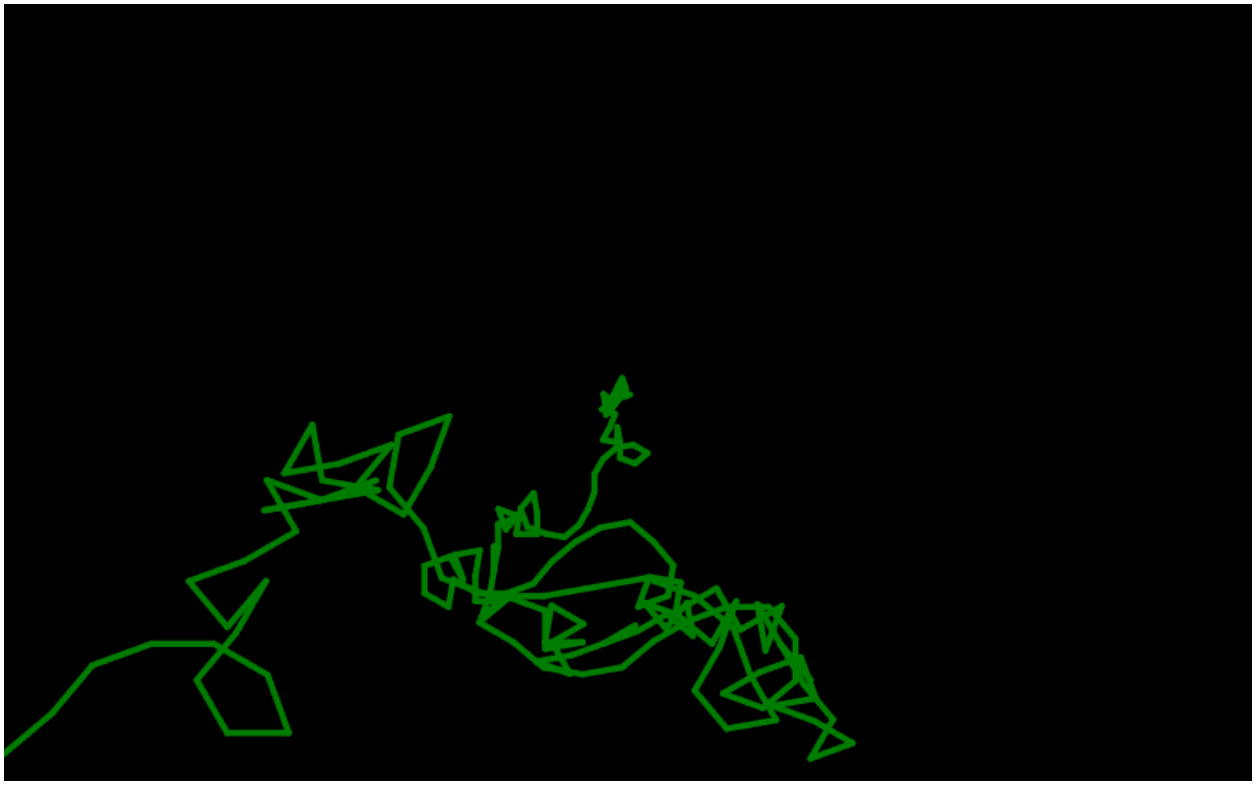
```
tree_rules = {"F": "fF[+F]F-F"} #a more cluster like generation
```

For the second rule set, I wanted to create more “vine” like appearance in the fractals, so I made the distance/length of line between every iteration increase. Increasing the length of the lines also creates longer and slender lines reminiscent of vines. And green was an obvious choice. I also generated these without using the save and load previous states.



```
tree_rules = {"F": "FFFF"} #repeated drawn lines and an incremental
increase in angel everytime “F” is called to create a “fiddle head” like
appearance
```

```
tree_rules = {"F": "-FFFF"} #a more vine like generation by adding a “-” in the rules
```



```
tree_rules = {"F": "-FFF+F"}# more messy/random generation when added ”+” into the rules
```