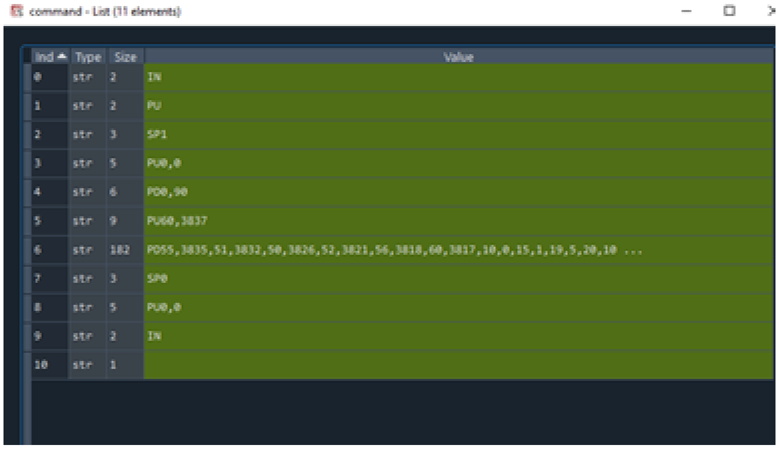
Memorandum

To: Charlie Refvem
From: Barrett Osborne, Ruodolf Rumbaoa
Date: 5/25/2021

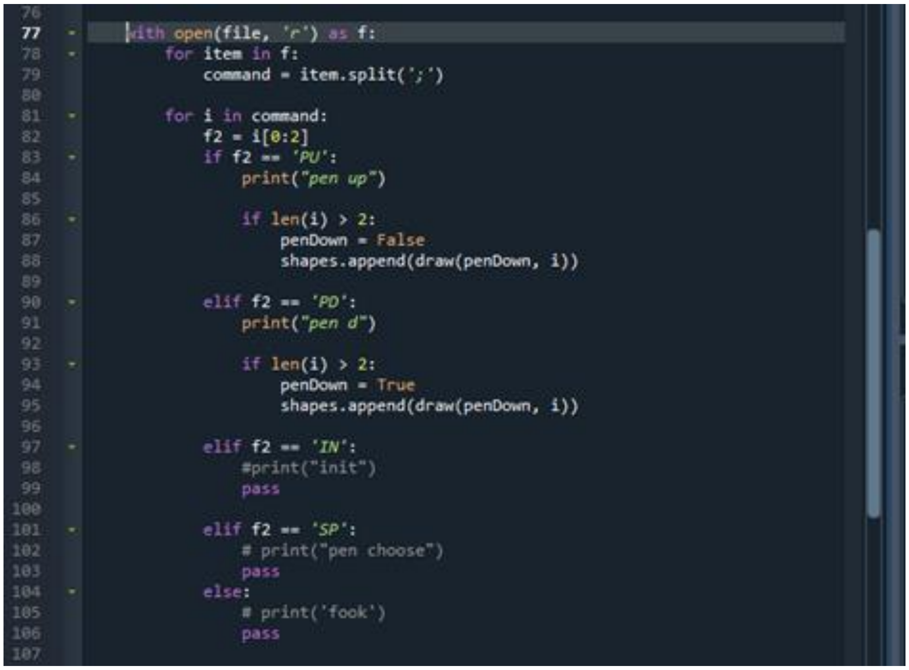
For Lab 0x03, our task was to generate an HPGL file using Inkscape, save the file as a text file, and parse the contents of the text file. Our implementation began by separating the large string within the text file by semi-colons using split (';'). This resulted in a list of lists containing the strings of instructions and coordinates.



Ind ▲	Type	Size	Value
0	str	2	IN
1	str	2	PU
2	str	3	SP1
3	str	5	PU0,0
4	str	6	PD0,90
5	str	9	PU60,3837
6	str	182	PD55,3835,51,3832,50,3826,52,3821,56,3818,60,3817,10,0,15,1,19,5,20,10 ...
7	str	3	SP0
8	str	5	PU0,0
9	str	2	IN
10	str	1	

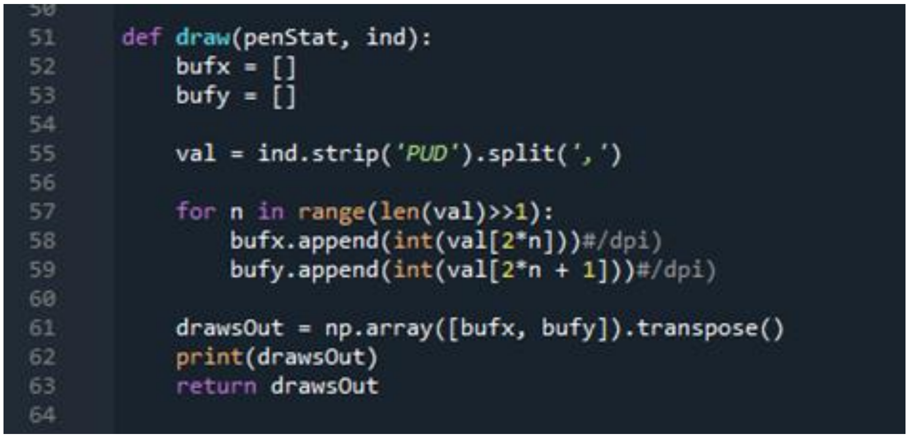
Figure 1. Data split by semi colons

To interpret the instructions (pen up or pen down) we iterated through, looking at the first two characters of each element. By doing this we could interpret the instructions generated by Inkscape. We then stripped these instruction characters to generate an array of all the coordinates to be processed through our Newton Raphson code from HW2.



```
with open(file, 'r') as f:
    for item in f:
        command = item.split(';')

    for i in command:
        f2 = i[0:2]
        if f2 == 'PU':
            print("pen up")

            if len(i) > 2:
                penDown = False
                shapes.append(draw(penDown, i))

        elif f2 == 'PD':
            print("pen d")

            if len(i) > 2:
                penDown = True
                shapes.append(draw(penDown, i))

        elif f2 == 'IN':
            #print("init")
            pass

        elif f2 == 'SP':
            # print("pen choose")
            pass
        else:
            # print('fook')
            pass

```

Figure 2. Instruction interpretation code snippet



```python
def draw(penStat, ind):
    bufx = []
    bufy = []

    val = ind.strip('PUD').split(',')

    for n in range(len(val)>>1):
        bufx.append(int(val[2*n]))#/dpi)
        bufy.append(int(val[2*n + 1]))#/dpi)

    drawsOut = np.array([bufx, bufy]).transpose()
    print(drawsOut)
    return drawsOut
```

Figure 3. Instruction character stripping and coordinate array creation code snippet

The Newton Raphson code generated an array of coordinates that we plotted and put together as a GIF.

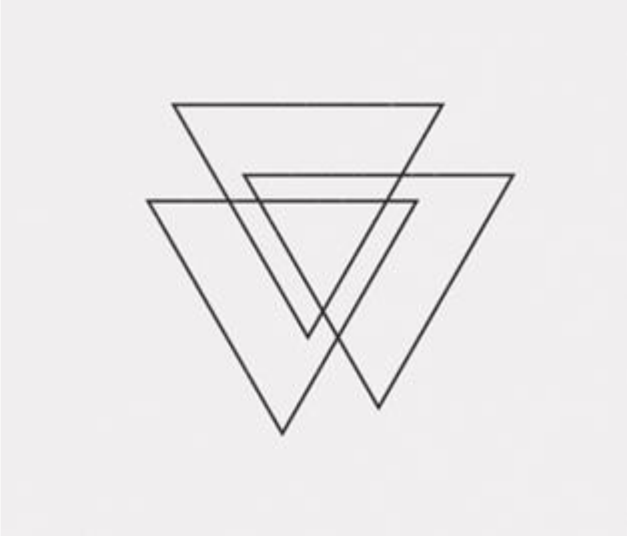


Figure 4. Original png

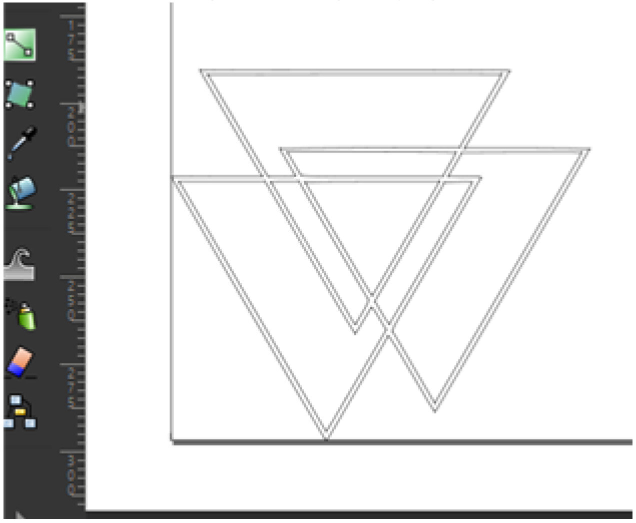


Figure 5. Inkscape generated plot

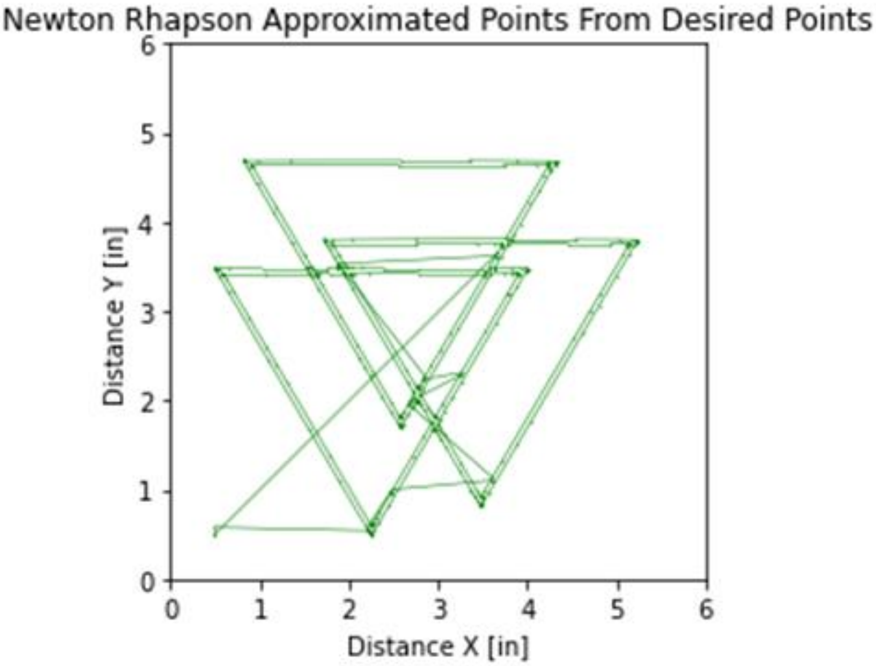


Figure 6. Plot generated from Newton Raphson code. NOTE: this plot and GIF shows the position of the end effector and doesn't show the pen lifting or going down