

MECE 4606

Assignment 1

Fuhao Lu

UNI: f12570

2021/2/7 2am

96 Grace hours + 22 gained = 118 left
photos:



Table of contents

- 1. Description of the software**
- 2. Pictures of two SVG files and photos of boxes**
- 3. Four examples with four different parameters**
- 4. Screenshots of the box posted on discussion session**
- 5. Edited video link**
- 6. Reference and bibliography**
- 7. Appendix of codes**

1. Description of the software:

Language: Python 3.7

Set the svg format:

```
SVG_HEADER = "<?xml version=\"1.0\" encoding=\"UTF-8\"
standalone=\"no\"?>"
SVG_PARAMS = "<svg width=\"3280\" height=\"2900\"
viewBox=\"0 0 3280 2900\"
xmlns=\"http://www.w3.org/2000/svg\">"
SVG_FOOTER = "</svg>"
```

First step:

Set functions:

1. get_rectangle: input x & y position; width and height as dimensions; stroke and color set as default.
 2. get_boxbase: input x & y position; width and height as dimensions.
 3. get_boxwall1&2: input x & y position; width and height as dimensions.
- 1-3 using the same coding method, which is :

```
"<rect x=\"{}\" y=\"{}\" width=\"{}\" height=\"{}\"
fill=\"none\" stroke-width=\"1\" stroke=\"black\" />"
```

4. text: input x & y position; output with 'Digital Manufacturing'.
 5. Text: input x & y position; output with 'Columbia Engineering'.
- 4-5 all set the text in the middle of the line, but 'Columbia Engineering' reversed 180 degrees because they are on the opposite. Svg code method:

Text transform; translate; text-anchor.

```
"<text transform=\"translate({}, {}) rotate(180)\" text-
anchor=\"middle\" font-size =\"30\" \" \
\"> Columbia Engineering </text>\" />"
```

6. get_polygon: input x & y positions of every point.

Method:

```
"<polygon points =\"({}, {} ), {} {}, {} {}, {} {}\" />"
```

7. dashed_line: input x & y positions.

Method:

```
"<line stroke-dasharray=\"5, 10\" \" \
\"fill=\"none\" stroke=\"red\" stroke-width=\"8\"
x1=\"{}\" y1=\"{}\" x2=\"{}\" y2=\"{}\" />"
```

8. columbia_logo: input x & y positions.

Method: use the image "href" link to a png picture.

```
" <image
href=\"https://www.logolynx.com/images/logolynx/fc/fc5018df8
```

```
2ef3dae3f35144a2296f513.png\" \" \  
"x=\"{ }\" y=\"{ }\" height=\"150\" width=\"150\"/>"
```

Second Step:

Use these functions to draw the expanded view of box.

1. Set the "start" and "starty" as the original point of the box.
2. Ask the user to get the input of box's height, width, depth, and the board's thickness, also convert them into inches instead of pixels.
3. Testify the numbers user just entered, if it is ≤ 0 , let user to enter a valid number again.

```
Please enter width for your box in inch:  
0  
Please enter a valid width for your box:  
5  
Please enter height for your box in inch:  
-3  
Please enter a valid height for your box:  
3
```

4. Use the function of "with open" to create a document as .svg file.

```
with open('box_view.svg', 'w') as svg_file:
```

5. Write the program by using svg_file. write to begin the software:
Begin with the svg_header and svg_params.

```
svg_file.write(SVG_HEADER)
```

```
svg_file.write(SVG_PARAMS)
```

6. Draw the extended view by using the functions, e.g.

```
svg_file.write(get_boxbase(startx, starty, box_width,  
box_depth))
```

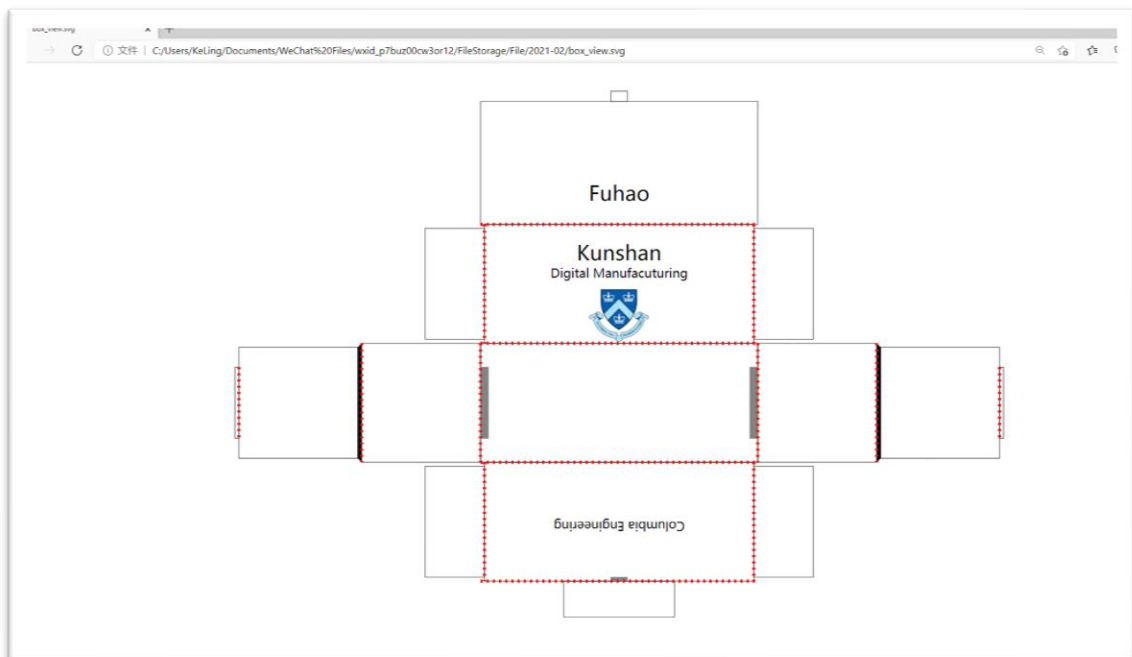
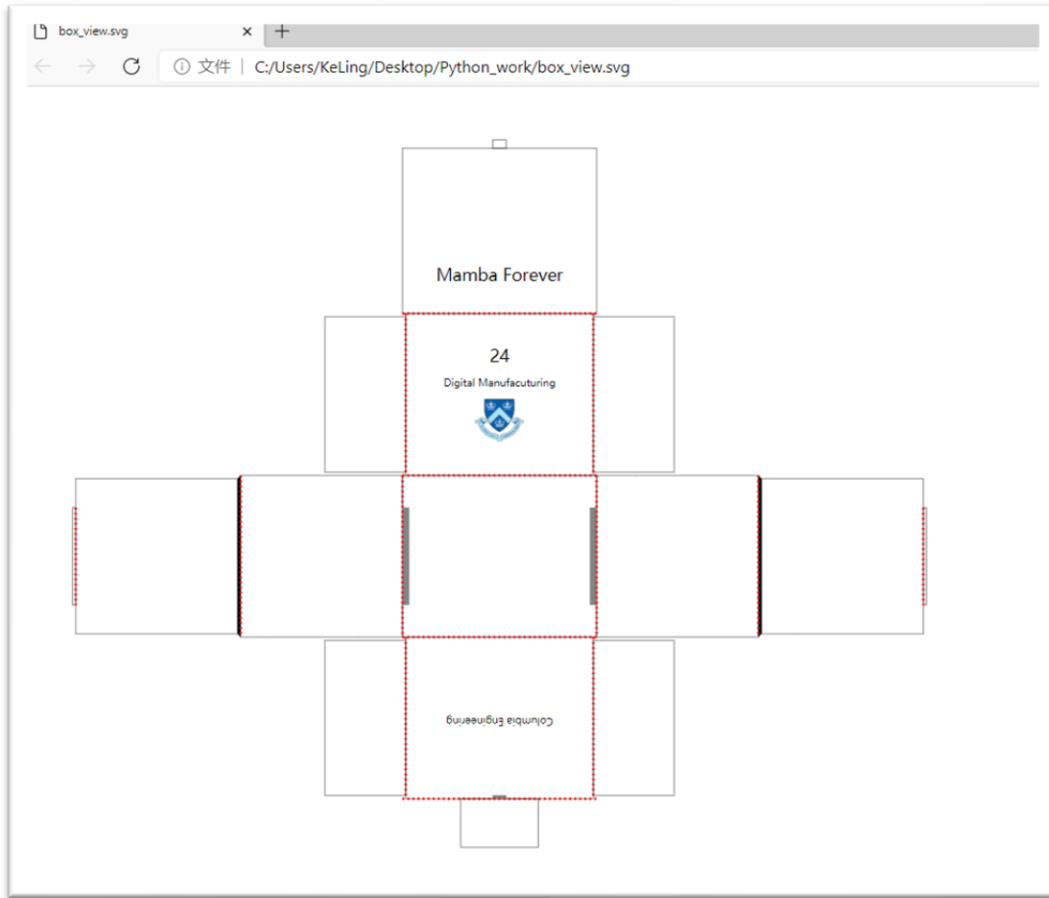
```
svg_file.write(get_boxwall1(startx + box_width, starty,  
box_height, box_depth))
```

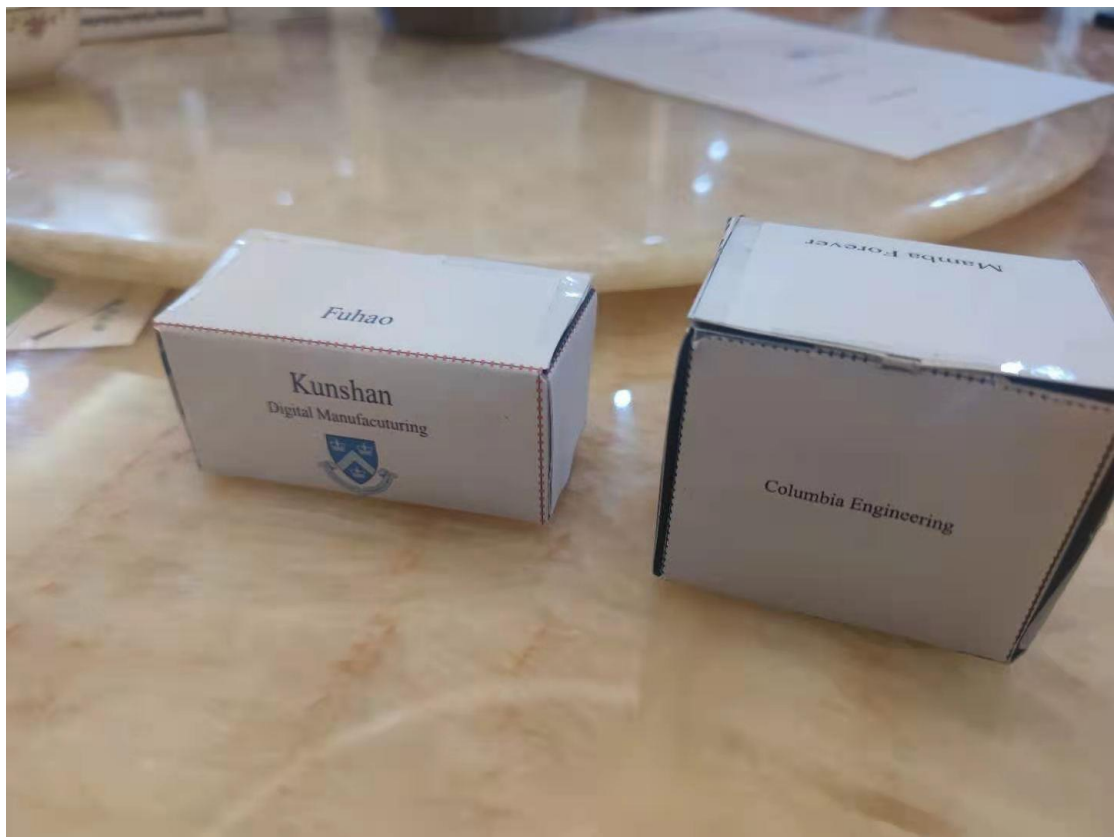
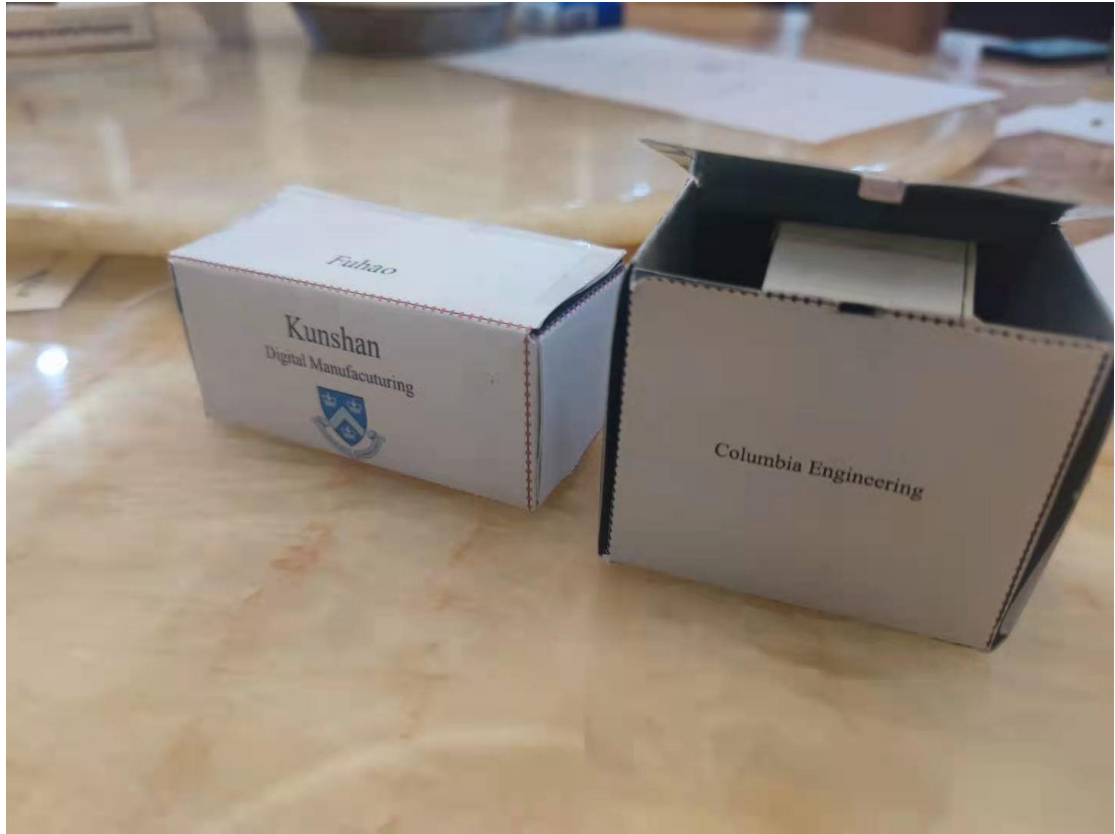
```
svg_file.write(get_boxwall1(startx - box_height, starty,  
box_height, box_depth))
```

Last Step:

print out the extension view and testify if that works. Come back to improve the software if it meets any problem.

2. Pictures of two SVG files and photos of boxes

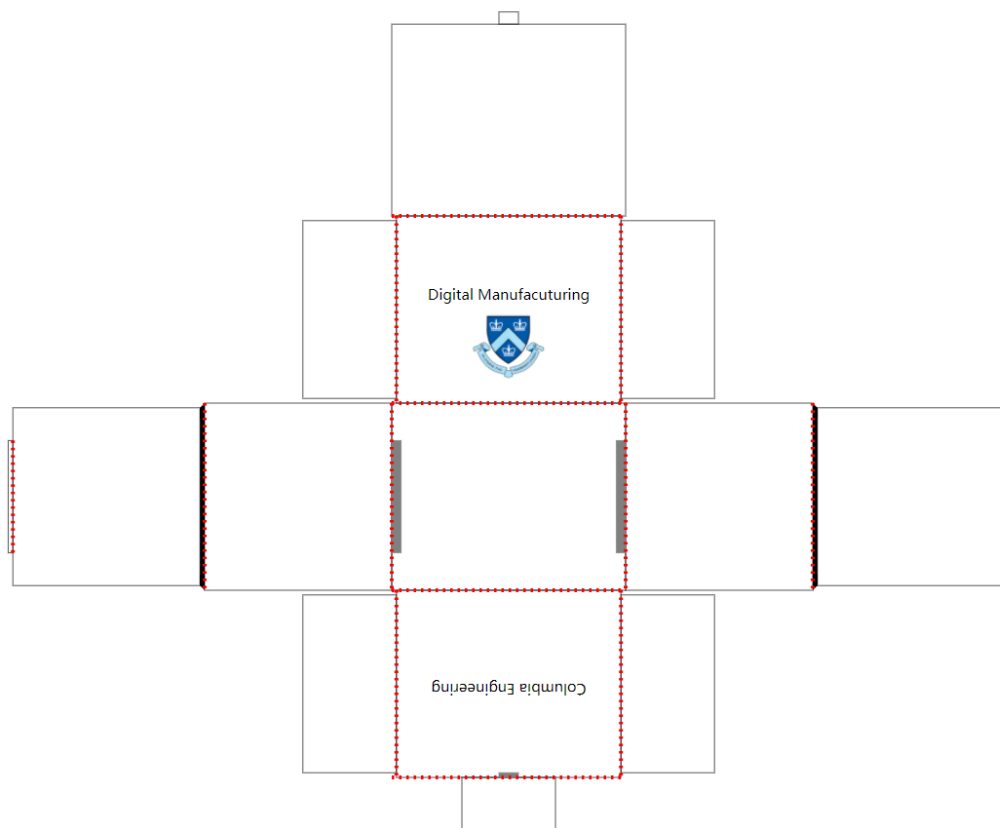




3. Four Examples:

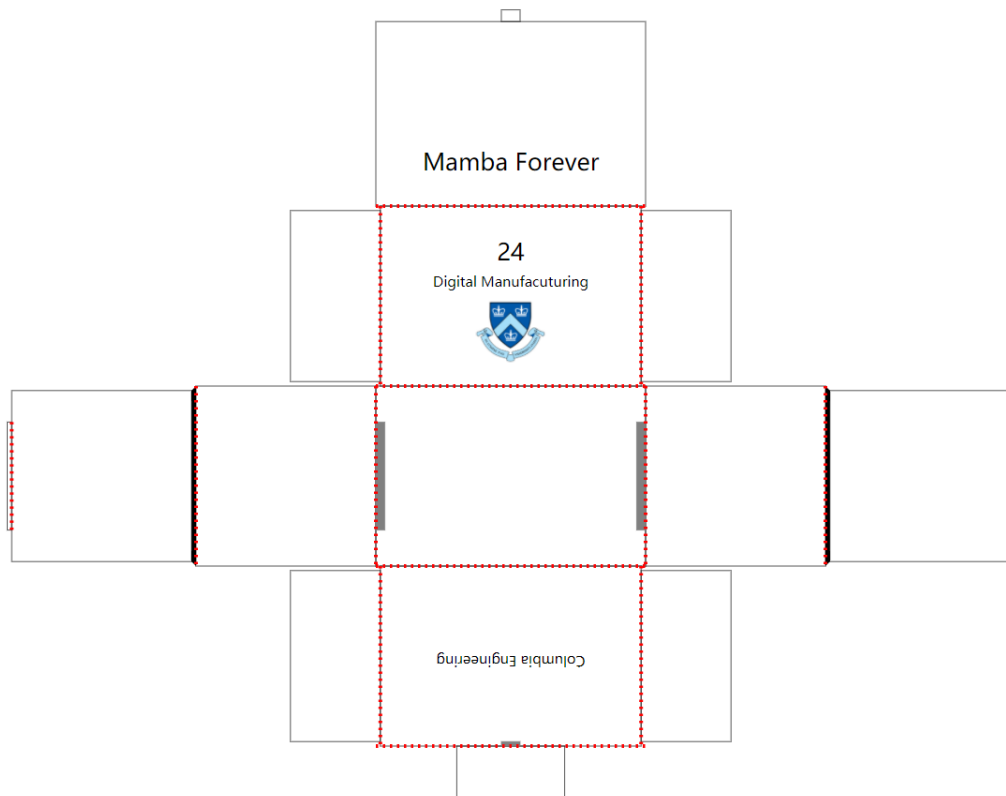
```
Please enter width for your box in inch:  
0  
Please enter a valid width for your box:  
5  
Please enter height for your box in inch:  
4  
Please enter depth for your box in inch:  
0  
Please enter a valid depth for your box:  
4  
Please enter board thickness for your box in inch:  
0.1  
Do you want to add text on the top of your box? (Y or N)  
n  
Do you want to add text on the front of your box? (Y or N)  
n  
Created generated.svg
```

1.



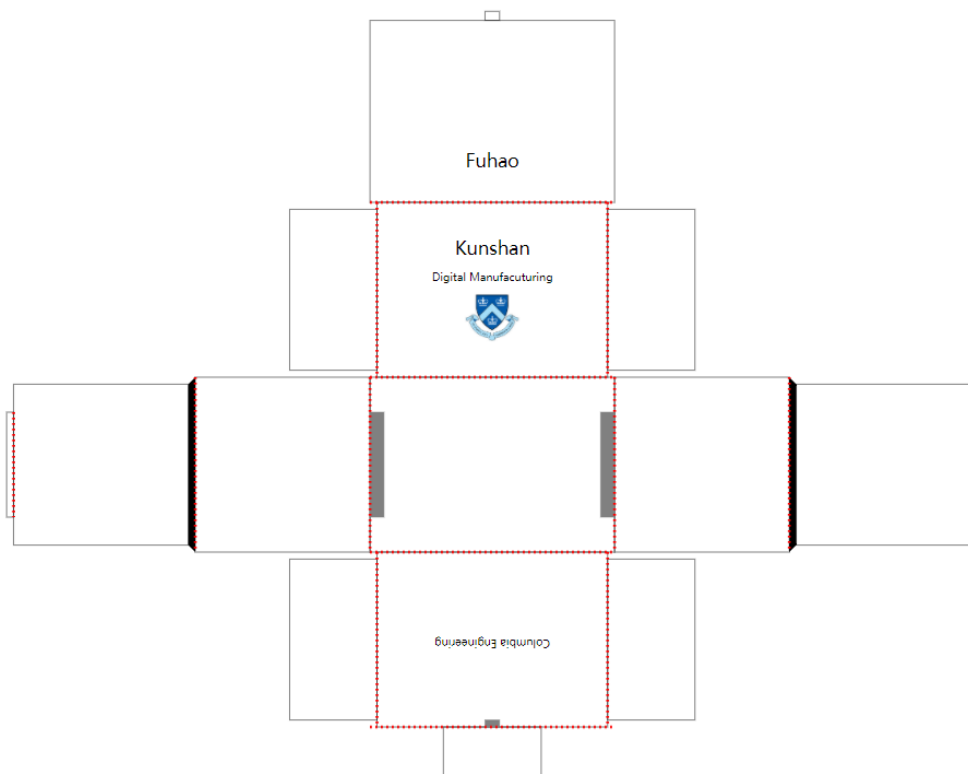
```
Please enter width for your box in inch:
6
Please enter height for your box in inch:
4
Please enter depth for your box in inch:
4
Please enter board thickness for your box in inch:
0.1
Do you want to add text on the top of your box? (Y or N)
y
Type here:
Mamba Forever
Do you want to add text on the front of your box? (Y or N)
y
Type here:
24
Created generated.svg
```

2.



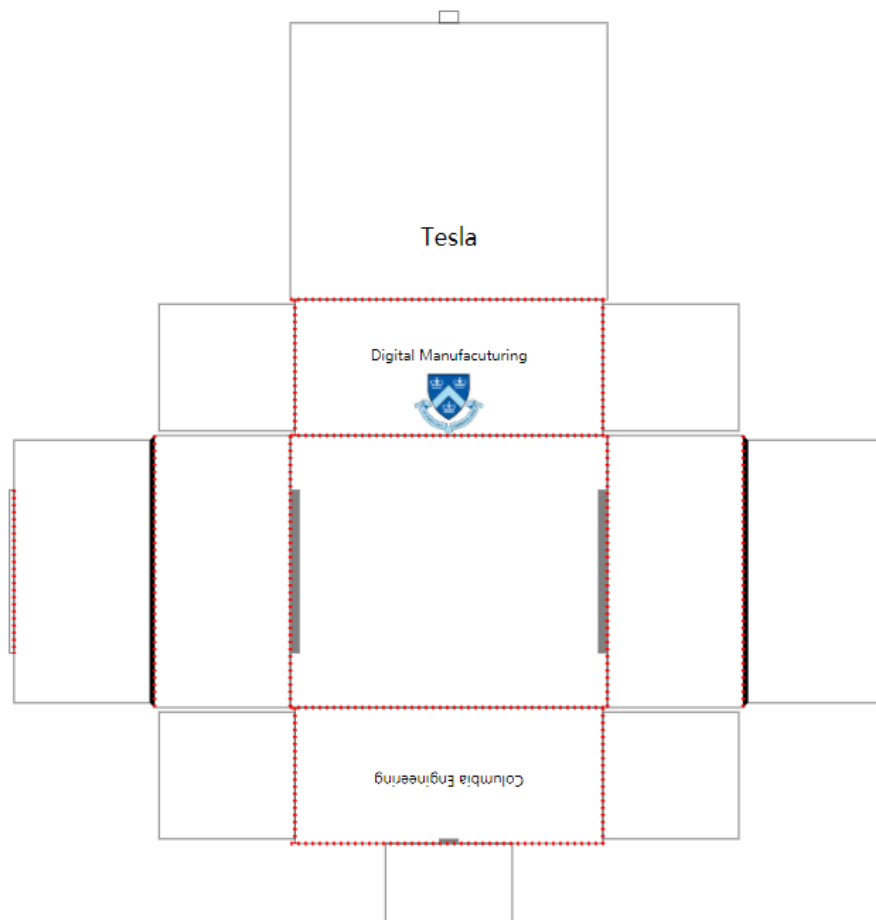

```
Please enter width for your box in inch:  
7  
Please enter height for your box in inch:  
5  
Please enter depth for your box in inch:  
5  
Please enter board thickness for your box in inch:  
0.2  
Do you want to add text on the top of your box? (Y or N)  
y  
Type here:  
Fuhao  
Do you want to add text on the front of your box? (Y or N)  
y  
Type here:  
Kunshan  
Created generated.svg
```

3.



```
Please enter width for your box in inch:  
7  
Please enter height for your box in inch:  
3  
Please enter depth for your box in inch:  
6  
Please enter board thickness for your box in inch:  
0.1  
Do you want to add text on the top of your box? (Y or N)  
Y  
Type here:  
Tesla  
Do you want to add text on the front of your box? (Y or N)  
n  
Created generated.svg
```

4.



4. Screenshots of the box posted on discussion session

image of my current box

[All Sections](#)

Last post at Feb 5 at 10:15am

 image of my current box
Fuhao Lu
[All Sections](#)

Hi,

This is my current box, the final version of the box might change because I am keep improving some details. Thanks.



5. Edited video link

<https://drive.google.com/file/d/1J6AUEDVkpB44RnUNjTN2rnQezX0egOto/view?usp=sharing>

6. Reference and bibliography

https://www.w3schools.com/graphics/svg_examples.asp

7. Code Appendix

```
SVG_HEADER = "<?xml version='1.0' encoding='UTF-8' standalone='no'?>"
SVG_PARAMS = "<svg width='4280' height='3900' viewBox='0 0 4280 3900'
xmlns='http://www.w3.org/2000/svg'>"
SVG_FOOTER = "</svg>"
```

```
def get_rectangle(x, y, width, height, stroke_width=1, line_color="black"):
    svg_rect = "<rect x='{}' y='{}' width='{}' height='{}' fill='none' stroke-
width='{}' stroke='{}' />"
    return svg_rect.format(x, y, width, height, stroke_width, line_color)
```

```
def get_boxbase(x, y, width, depth):
    svg_box = "<rect x='{}' y='{}' width='{}' height='{}' fill='none' stroke-
width='1' stroke='black' />"
    return svg_box.format(x, y, width, depth)
```

```
def get_boxwall1(x, y, height, depth):
    svg_box = "<rect x='{}' y='{}' width='{}' height='{}' fill='none' stroke-
width='1' stroke='black' />"
    return svg_box.format(x, y, height, depth)
```

```
def get_boxwall2(x, y, width, height):
    svg_box = "<rect x='{}' y='{}' width='{}' height='{}' fill='none' stroke-
width='1' stroke='black' />"
    return svg_box.format(x, y, width, height)
```

```
def cut_rect(x, y, width, height):
    cut_box = "<rect x='{}' y='{}' width='{}' height='{}' fill='grey' stroke-
width='1' stroke='grey' />"
    return cut_box.format(x, y, width, height)
```

```
def text1(x, y):
    text = "<text text-anchor='middle' x='{}' y='{}' font-size ={}> Digital
Manufacturing </text>"/>"
    return text.format(x, y)
```

```

def text2(x, y):
    text = "<text transform=\"translate({},{}) rotate(180)\" text-anchor=\"middle\" font-size
=30\" \" \" \
        "> Columbia Engineering </text>\"/>"
    return text.format(x, y)

```

```

def user_text1(x, y, a):
    text = "<text text-anchor=\"middle\" x={}\" y={}\" font-size =50\"> {} </text>\"/>"
    return text.format(x, y, a)

```

```

def get_polygon(x1, y1, x2, y2, x3, y3, x4, y4):
    polygon = "<polygon points ={{},{},{{},{},{{},{},{{},{}}\" />"
    return polygon.format(x1, y1, x2, y2, x3, y3, x4, y4)

```

```

def dashed_line(x1, y1, x2, y2):
    dashed = "<line stroke-dasharray=5, 10\" \" \" \
        "fill=\"none\" stroke=\"red\" stroke-width=8\" x1={}\" y1={}\" x2={}\"
y2={}\" />"
    return dashed.format(x1, y1, x2, y2)

```

```

def columbia_logo(x, y):
    logo = " = "
    href = "https://www.logolynx.com/images/logolynx/fc/fc5018df82ef3dae3f35144a2296f513.png"
    "x={}\" y={}\" height=150\" width=150\"/>"
    return logo.format(x, y)

```

```

def main():
    startx = 12 * 96
    starty = 12 * 96
    box_width = 96 * float(input("Please enter width for your box in inch: \n"))
    while box_width <= 0:
        box_width = 96 * float(input("Please enter a valid width for your box: \n"))
    box_height = 96 * float(input("Please enter height for your box in inch: \n"))
    while box_height <= 0:
        box_height = 96 * float(input("Please enter a valid height for your box: \n"))
    box_depth = 96 * float(input("Please enter depth for your box in inch: \n"))
    while box_depth <= 0:

```

```

    box_depth = 96 * float(input("Please enter a valid depth for your box: \n"))
    box_thickness = 96 * float(input("Please enter board thickness for your box in inch: \n"))
    while box_thickness <= 0:
        box_thickness = 96 * float(input("Please enter a valid thickness for your box: \n"))
    user_yn1 = input("Do you want to add text on the top of your box? (Y or N) \n")
    if user_yn1.lower() == 'y':
        text_top = input("Type here: \n")
    user_yn2 = input("Do you want to add text on the front of your box? (Y or N) \n")

with open('box_view.svg', 'w') as svg_file:
    svg_file.write(SVG_HEADER)
    svg_file.write(SVG_PARAMS)
    svg_file.write(get_boxbase(startx, starty, box_width, box_depth))
    svg_file.write(get_boxwall1(startx + box_width, starty, box_height, box_depth))
    svg_file.write(get_boxwall1(startx - box_height, starty, box_height, box_depth))
    svg_file.write(
        get_boxwall2(startx + box_thickness, starty - box_height, box_width - 2 *
box_thickness, box_height))
    svg_file.write(
        get_boxwall2(startx + box_thickness, starty + box_depth, box_width - 2 *
box_thickness, box_height))

    svg_file.write(get_boxwall1(startx + box_thickness - (0.5 * box_depth), starty -
box_height + box_thickness,
                                0.5 * box_depth, box_height - 2 * box_thickness))
    svg_file.write(get_boxwall1(startx + box_width - box_thickness, starty - box_height
+ box_thickness,
                                0.5 * box_depth, box_height - 2 * box_thickness))
    svg_file.write(get_boxwall1(startx + box_width - box_thickness, starty + box_depth
+ box_thickness,
                                0.5 * box_depth, box_height - 2 * box_thickness))
    svg_file.write(get_boxwall1(startx + box_thickness - (0.5 * box_depth), starty +
box_depth + box_thickness,
                                0.5 * box_depth, box_height - 2 * box_thickness))
    # top
    svg_file.write(get_boxwall1(startx, starty - box_height - box_depth-box_thickness,
box_width, box_depth+box_thickness))
    svg_file.write(get_boxwall1(startx+0.3*box_width,    starty+box_depth+box_height,
0.4*box_width, 0.3*box_depth))
    # fix
    svg_file.write(get_boxwall1(startx+0.5*box_width-20,          starty-box_height-
box_depth-box_thickness-25, 40, 25))
    svg_file.write(cut_rect(startx+0.5*box_width-20,    starty+box_depth+box_height-
box_thickness, 40, box_thickness))

```

```

# cut area
svg_file.write(cut_rect(startx, starty + (0.2 * box_depth), 2 * box_thickness, 0.6 *
box_depth))
svg_file.write(cut_rect(startx + box_width - 2 * box_thickness, starty + (0.2 *
box_depth), 2 * box_thickness,
0.6 * box_depth))
svg_file.write(text1(startx + 0.5 * box_width, starty - 0.55 * box_height))
svg_file.write(text2(startx + 0.5 * box_width, starty + box_depth + 0.5 * box_height))

# polygon
svg_file.write(get_polygon(startx + box_width + box_height, starty, startx +
box_width + box_height,
starty + box_depth, startx + box_width + box_height
+ box_thickness,
starty + box_depth - box_thickness, startx +
box_width + box_height + box_thickness,
starty + box_thickness))
svg_file.write(get_polygon(startx - box_height, starty, startx - box_height, starty +
box_depth,
startx - box_height - box_thickness, starty +
box_depth - box_thickness,
startx - box_height - box_thickness, starty +
box_thickness))

svg_file.write(get_boxwall1(startx + box_width + box_height + box_thickness, starty
+ box_thickness, box_height,
box_depth - 2 * box_thickness))

svg_file.write(
get_boxwall1(startx - box_height - box_thickness - box_height, starty +
box_thickness, box_height,
box_depth - 2 * box_thickness))

svg_file.write(
get_boxwall1(startx + box_width + box_height + box_thickness + box_height,
starty + (0.2 * box_depth),
box_thickness, 0.6 * box_depth))
svg_file.write(get_boxwall1(startx - 2 * box_thickness - 2 * box_height, starty + (0.2
* box_depth),
box_thickness, 0.6 * box_depth))

svg_file.write(dashed_line(startx, starty, startx + box_width, starty))
svg_file.write(dashed_line(startx, starty + box_depth, startx + box_width, starty +
box_depth))
svg_file.write(dashed_line(startx, starty, startx, starty + box_depth))
svg_file.write(dashed_line(startx + box_width, starty, startx + box_width, starty +
box_depth))

```

```

        svg_file.write(dashed_line(startx, starty - box_height, startx + box_width, starty -
box_height))
        svg_file.write(dashed_line(startx + box_thicknness, starty - box_height, startx +
box_thicknness, starty))
        svg_file.write(dashed_line(startx + box_width - box_thicknness, starty - box_height,
startx + box_width - box_thicknness, starty))
        svg_file.write(dashed_line(startx + box_thicknness, starty + box_depth, startx +
box_thicknness,
                                starty + box_depth + box_height))

        svg_file.write(
            dashed_line(startx + box_width - box_thicknness, starty + box_depth, startx +
box_width - box_thicknness,
                                starty + box_depth + box_height))

        svg_file.write(
            dashed_line(startx, starty + box_depth + box_height, startx + box_width, starty
+ box_depth + box_height))
        svg_file.write(
            dashed_line(startx + box_width + box_height, starty, startx + box_width +
box_height, starty + box_depth))
        svg_file.write(dashed_line(startx - box_height, starty, startx - box_height, starty +
box_depth))
        svg_file.write(dashed_line(startx + box_width + 2 * box_height + box_thicknness,
starty + 0.2 * box_depth,
                                startx + box_width + 2 * box_height + box_thicknness,
starty + 0.8 * box_depth))
        svg_file.write(dashed_line(startx - 2 * box_height - box_thicknness, starty + 0.2 *
box_depth,
                                startx - 2 * box_height - box_thicknness, starty + 0.8
* box_depth))
        # logo
        svg_file.write(columbia_logo(startx + 0.5 * box_width - 75, starty - 0.5 * box_height))
        if user_yn1.lower() == 'y':
            svg_file.write(user_text1(startx + 0.5 * box_width, starty - box_height - 0.2 *
box_depth, text_top))

        if user_yn2.lower() == 'y':
            text_front = input("Type here: \n")
            svg_file.write(user_text1(startx + 0.5 * box_width, starty - 0.7 * box_height,
text_front))

        svg_file.write(SVG_FOOTER)

    print("Created generated.svg")

```



```
if __name__ == "__main__":  
    main()
```