

Yixin Li's CS151 Project 2

This project focuses on using functions to create a space-themed scene made up of simple and compound shapes. As the Required Image 1, the first shape I made is triangle stacks coming from triangles with different scales. The Required Images 2 and 3 show triangles and rectangles with parameters such as x, y, scale, length, and width. The Required Images 4 and 5 are two compound shapes: rocket and meteor, a series of complex commands made of triangles, rectangles, and stars with various scales and colors. Last but not least, the Required Image 6, shown as the function `space1` in the `main.py` file, is the space scene, which includes nine main planets and the sun surrounded by two spaceships and two rockets, with simple and compound shapes in my shape library.

To accomplish these basic tasks, I created simple shapes and then assigned proper parameters, including scales and colors, to make them compound shapes. Besides this, I made other basic shapes like star, circle, and oval to make my scene more vivid. I also used for loops and random function to create random positions for meteor ten times. Other turtle properties and parameters such as pen size, pen/fill color, and background color are also used. All of these operations count as several extensions I made beyond the basic requirements for the space scene.

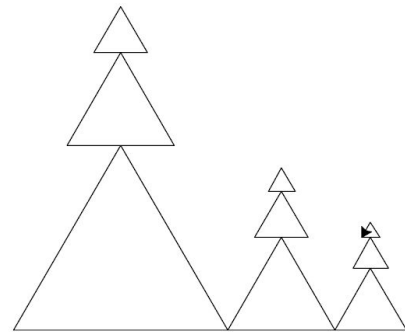
Through the project, I learned how to improve simple functions with different Python turtle parameters and scales to create attractive images. Besides repeated commands, I understood how to use loops to draw shapes with randomness.

Required Image 1:

```

ain.py  lab2.py  shapelib.py  compoundShape.py
Project02 > shapelib.py > ...
5  '''Function drawing three triangles on top of each other.
6  # Smaller triangles are placed on top of larger ones'''
7
8  def triangleStack():
9      # Largest triangle
10     triangle2(0, 0, 2)
11     # Medium triangle in middle
12     triangle2(50, 173.2, 1)
13     # Small triangle on top of stack
14     triangle2(75, 259.81, 0.5)
15
16
17 def triangleStack(x, y, scale):
18     goto(x, y)
19     triangle2(x, y, scale * 2)
20     triangle2(x + scale * 50, y + scale * 173.2, scale * 1)
21     triangle2(x + scale * 75, y + scale * 259.81, scale * 0.5)
22
23 triangleStack(-200, 0, 1) # draw three stacks of triangles
24 triangleStack(0, 0, 1/2)
25 triangleStack(100, 0, 1/3)

```



Required Image 2:

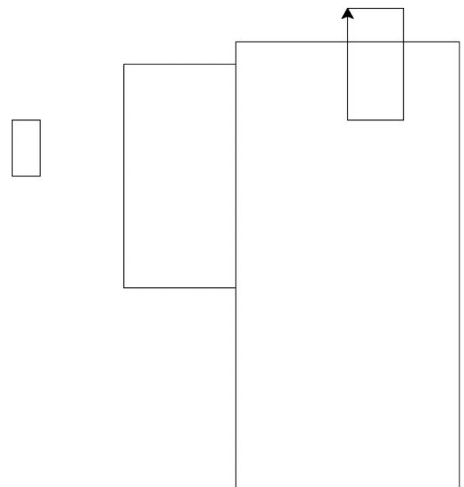
```

def testScene():
    shapelib.rectangle2(t, 0,
    shapelib.rectangle2(t, 10
    shapelib.rectangle2(t, 20
    shapelib.rectangle2(t, 30

    # shapelib.triangle2(t, -
    # shapelib.triangle2(t, -
    testScene()

screen.exitonclick()

```



Required Image 3:

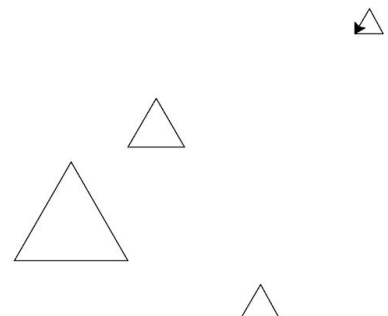
```

def testScene():
    # shapelib.rectangle2(t, 0, 0, 1/4)
    # shapelib.rectangle2(t, 100, 50, 1)
    # shapelib.rectangle2(t, 200, 70, 2)
    # shapelib.rectangle2(t, 300, 100, 1/2)

    shapelib.triangle2(t, 0, 0, 1)
    shapelib.triangle2(t, 100, 100, 1/2)
    shapelib.triangle2(t, 200, -50, 1/3)
    shapelib.triangle2(t, 300, 200, 1/4)

    testScene()

```



Required Image 4:

```

95     star2(t, 0, 0, 2)
96     star2(t, 200, 0, 1)
97     star2(t, 300, 0, 0.5)
98     rectangle2(t, 350, 0, 1, 0.5, 1)
99
100 def meteor(t, x, y, scale, color):
101     goto(x, y)
102     t.pencolor(color)
103     t.fillcolor(color)
104     t.begin_fill()
105     star2(t, x, y, scale * 2)
106     star2(t, x + scale * 200, y + scale * 0, scale * 1)
107     star2(t, x + scale * 300, y + scale * 0, scale * 0.5)
108     rectangle2(t, x + scale * 350, y + scale * 0, scale * 1/3)
109     t.end_fill()
110 meteor(t, -200, 0, 1, 'yellow')
111 meteor(t, -200, 100, 1/2, 'orange')
112 meteor(t, -200, 200, 1/3, 'GreenYellow')
113

```

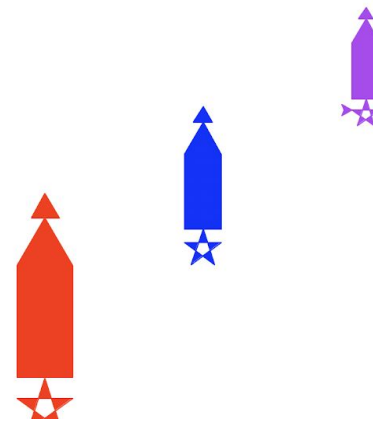


Required Image 5:

```

# rocket()
def rocket(t, x, y, scale, color):
    goto(x, y)
    t.pencolor(color)
    t.fillcolor(color)
    t.begin_fill()
    triangle2(t, x, y, scale * 1)
    triangle2(t, x + scale * 25, y + scale * 86.61, scale * 0.5)
    rectangle2(t, x + scale * 0, y + scale * 0, scale * 1)
    star2(t, x + scale * 0, y + scale * -238.2, scale * 1)
    t.end_fill()
rocket(t, -200, -100, 1/2, 'red')
rocket(t, -50, 0, 1/3, 'blue')
rocket(t, 100, 100, 1/4, 'purple')

```



Required Image 6:

