

Project 11, syman

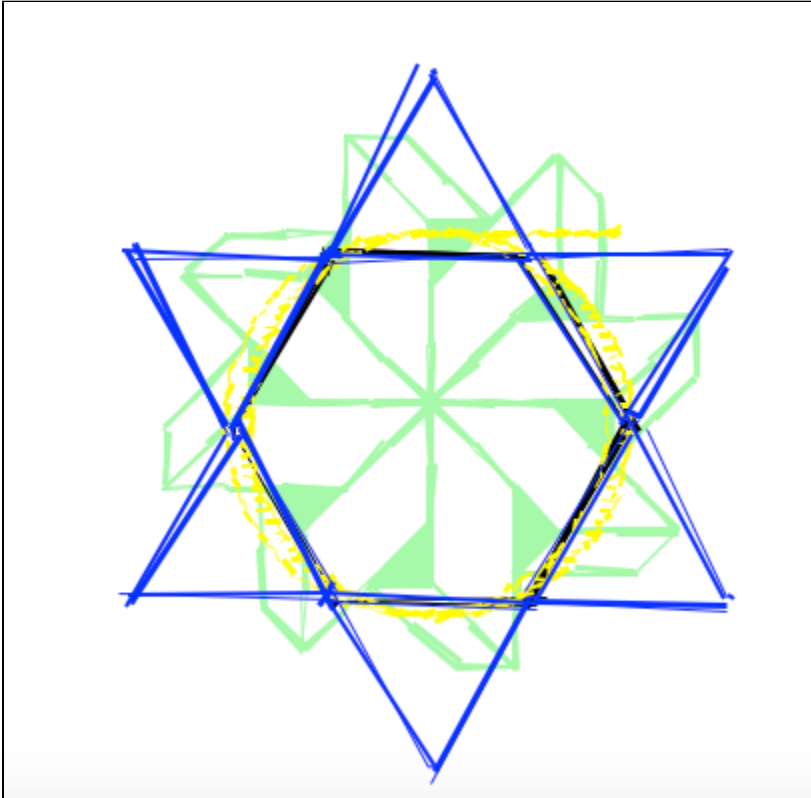
Summary:

In this week's ?project, I learned how to draw images in 3d using a turtle. This allowed me to practice new ways of coding such as using my classes and using recursive functions.

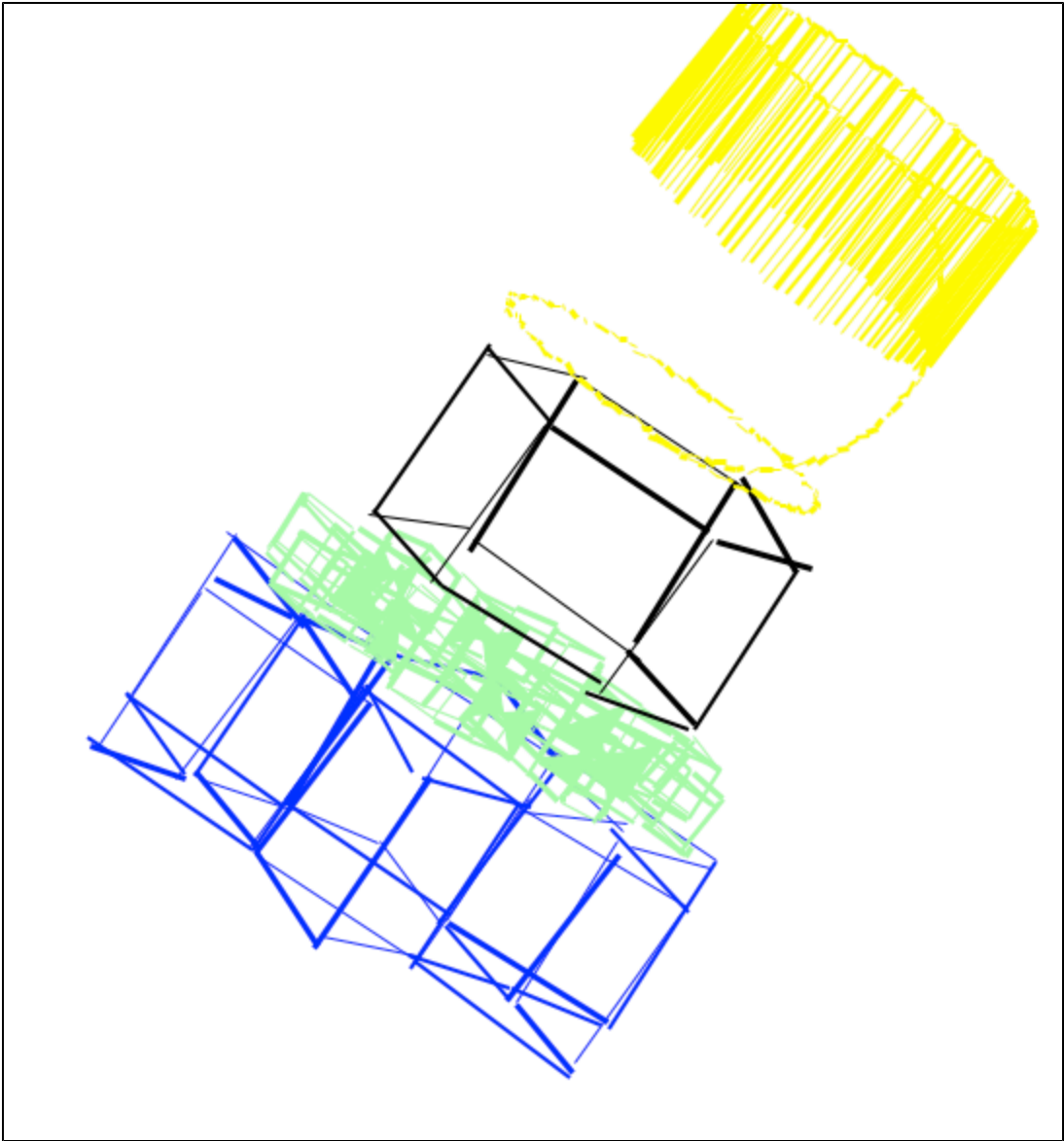
Task 1:

In this task, I created 4 images in 3d and put them creatively in a 3d scene. I arranged the 3d scene to appear like an abstract camera. The camera is composed of a star as the body and several objects as a lens.

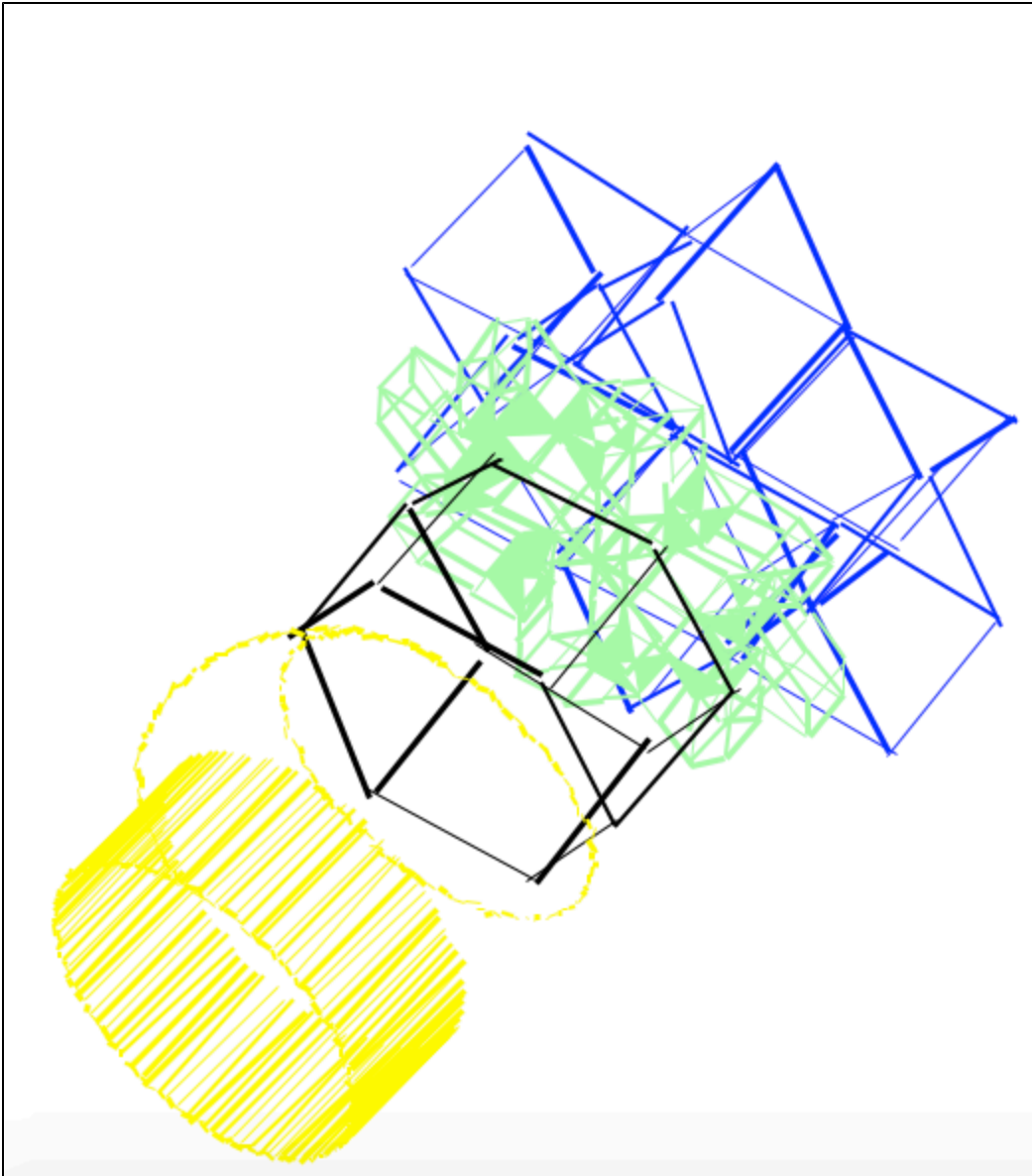
This is the front view:



This is a rotated view:



This is another rotated view:



Task 4 and extension:

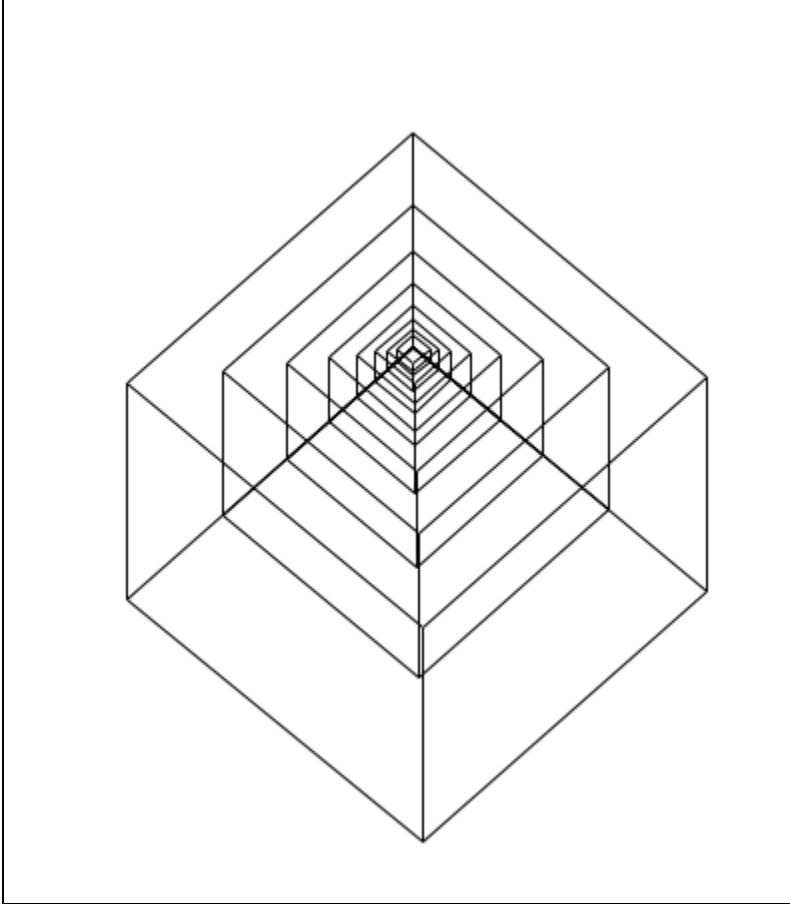
In this task, I aimed at creating a complex image using simple string. In order to keep the code simple and modular, I didnt implement as many aspects as I initially intended. This use tiny movements of a few degrees by the turtle with a fixed angle.



```
class Lamp3D(Shape):  
    '''draws circular patterns that mimick a lamp'''  
    def __init__(self, distance=100, color='yellow' ):  
  
        Shape.__init__(self, 6,3.6, color= color, istring = 100*'F-'+  
            '(3.6)'+50*'F(3.6)'+100*'F-[(90)&(15)F]')
```

Task recursive:

Below is an image that represent nested blocks, this used a recursive formula below.



Below is the recursive function for the nested boxes above:

```
def drawboxes(x):
    '''draws nested boxes with initial scale at 1, until scale reaches a
    value greater than 16'''

    box=s.Box3D()
    box.draw(0,0,scale=x)

    if x>16:
        return True

    else:
        drawboxes(x*1.5)

it.TurtleInterpreter().hold()

drawboxes(1)
```

Extension:

- I used complex strings to make my shapes as shown below:

The first string was to design my star

The second string was used to create a lamp

```
istring = 8*'+FF(F--F)+++FF+F+++[FF]-FFF+FF'+ '(90)^F(90)&'+
8*'+F[(90)&F]F[(90)&F]{F[(90)&F]}--F[(90)&F]}+++F[(90)&F]F[(90)&F]+F[(90)&F]+++[F[(90)&F]F[(90)&F]]-F[(90)&F]F[(90)&F]F[(90)&F]+F[(90)&
```

```
class Lamp3D(Shape):
    '''draws circular patterns that mimick a lamp'''
    def __init__(self, distance=100, color='yellow' ):

        Shape.__init__(self, 6,3.6, color= color, istring = 100*'F-'+
        '(3.6)^^+50*'F(3.6)&' +100*'F-[(90)&(15)F]')
```

- I made a complex scene that depicts a camera.

The project helped me implement various methods that I developed during my semester with the cs department.

Credit:

I would like to thank Phoebe for her help as a tutor.

I would like to thank Professor Li for her support and allowing to submit this work late.