

Doris CS151 project 2

- **Summary**

This project focuses on the creation of aggregate images. The key is to understand parameters and how relative location functions work. Calculation of relative location is crucial for this project, as this is what makes each scene proportionate. I made simple aggregate shapes, such as flowers (made from triangles) and cactus, for which a "for" loop is used since a cactus is symmetrical. More complicated aggregate shapes include Christmas tree (made from triangles of different scales, rectangle, and a star), painting (made from a rectangle picture frame and random number of flowers as objects), chandelier (made from rectangles of decreasing size downwards and crosses as decorations).

- **Description**

"for" loops are used frequently in this project. For example, a parallelogram is drawn by repeating the loop, since the degree sum of the two interior

```
for i in range (2):  
    t.forward(a)  
    t.left(angle)  
    t.forward(b)  
    t.left(180-angle)
```

r angles on the same side is 180.

All of my basic shape functions work properly as shown in image1.

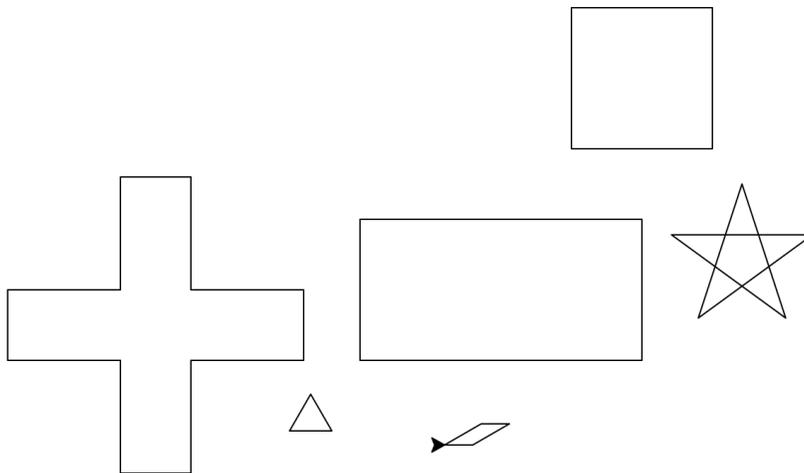


Image1

As for aggregate shapes, random package is imported. For example, flowers are made from random number of triangles (i.e. petals), I limited the number of triangles within the range of 3-12 so as not to make flowers too dense with petals, and here, a "for" loop is also introduced to repeat the process of rotating the triangle according to whatever number of triangles that the turtle intends to draw. When I made the paintings, random package is imported again so that I can draw the random number of flowers on each painting. The location at which the flowers are drawn is also

within a certain range (i.e. the side length of the picture frame).Therefore, the more complex aggregate shape "painting" involves two levels of randomness-the number of petals in a single flower and the number of flowers in a single painting.Also worth mentioning is the shape "table". I managed to create a 2D image by using parallelogram and rectangles to create the perspective. All of my aggregate shape functions work properly as shown in image2.They were tested at different locations and of different sizes.

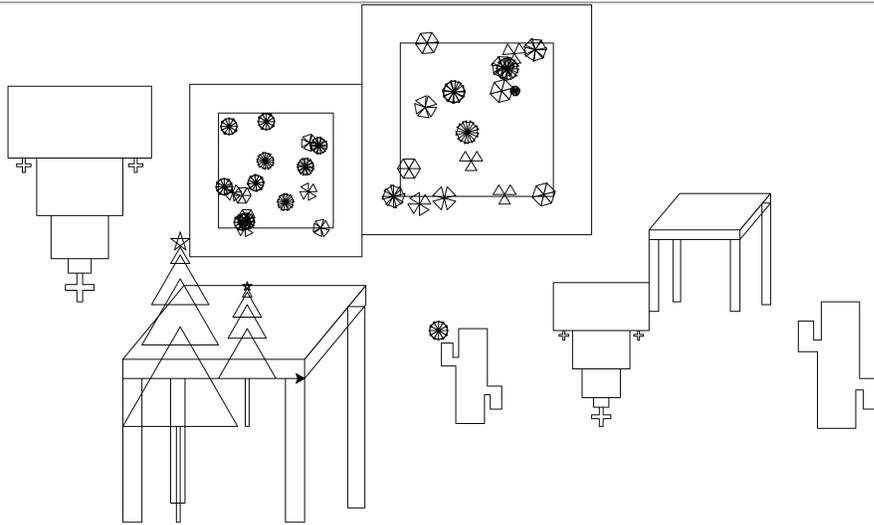


image2

Creating scenes are more complicated, since I have to gauge the size of every objects and add color. The one problem I encountered is that the coloring would follow the path by which the turtle moves to the next (x,y) position. To solve this problem, for example, in my "chandelier" shape, I added "beginfill" and "endfill" in the basic shape "block" so that the turtle knows when to fill colors in the aggregate shapes.

The museum scene, as shown by image 3, has a chandelier, two paintings, two tables, and one Christmas tree. I did not use random package for the number of paintings, since doing so would make the scene messy and packed.

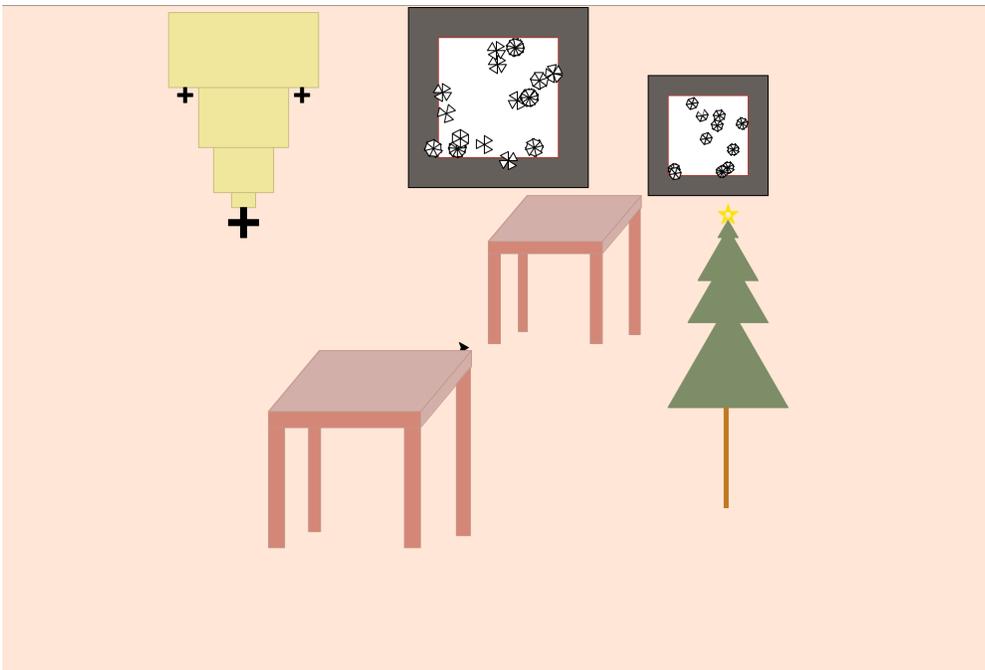


image3

The desert scene, as shown by image 3, has 7 cactuses and 100 stars which are dispersed randomly. I set the range of the position of stars within $(-300*size, 300*size)$. A "for" loop was also introduced.

```
for i in range (100):  
    sl.star(random.randint(-300*size,300*size),random.randint(-300*size,300*size),12)
```

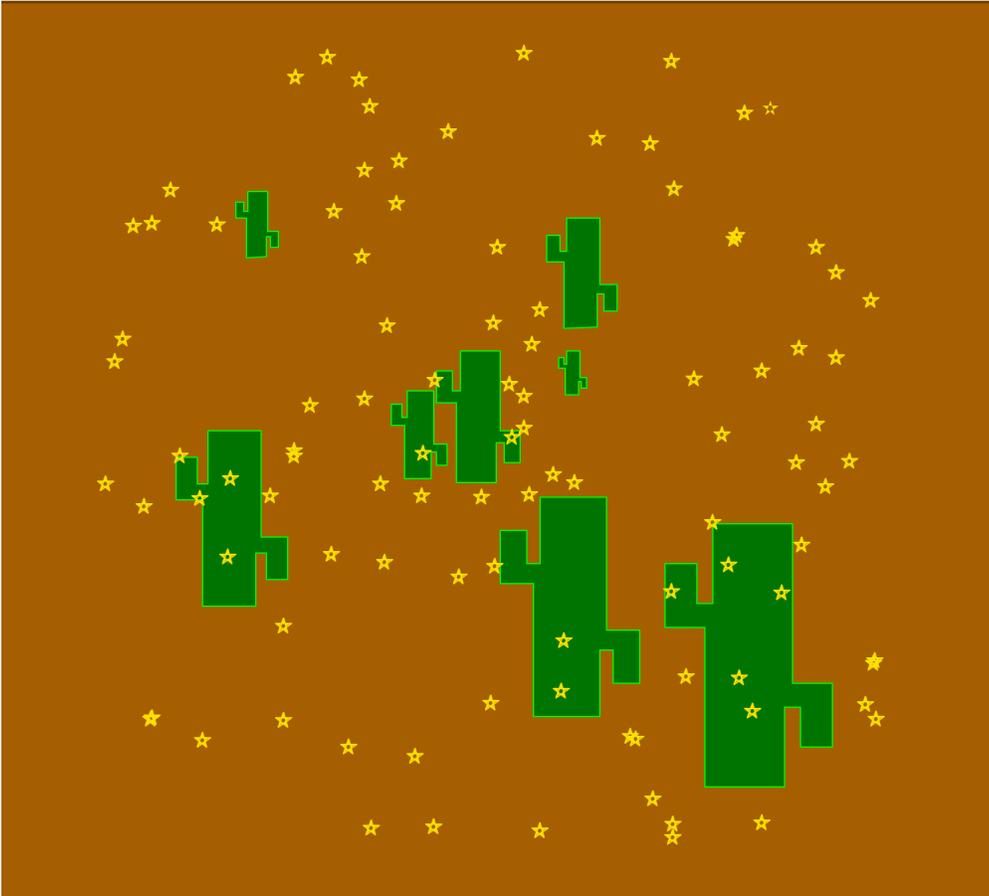


image4

• Follow-up questions

1. Does breaking a scene into pieces make it easier to create?
Yes, this the level of complexity increases from pieces to a scene, you do not have to worry about the entirety of the whole scene once you make sure each subordinated function works effectively.
2. What is a for loop and why are they useful?
A "for" loop includes the number of times you want this function to repeat, and it makes your actions efficient, since you don't have to repeat the same command.
3. What is the difference between using "from turtle import *" and "import turtle"? Explain from both a practical use point of view and how they differentially affect the global symbol table.
Using "from turtle import" means that you don't have to type "turtle." every time you want to use the symbol table for turtle module. In contrast, when you write "import turtle", you have to write "turtle." plus any symbol table you wish to utilize, for example (turtle.forward).
4. What is your favorite museum?
5. Isabella Stewart Gardner Museum 😊

• What I learned

The idea is to build aggregate shapes from basic shapes and to build a complete scene involving aggregate shapes so that the level of complexity increases. One things that happens a lot is that the computer does not take the step that I intended to take. The problem still lies in the barrier between

human minds and computational language. To be precise, one should be detail-oriented, and this project is an example of how small objects can affect the whole scene.

- **Acknowledgement**

I would like to thank lab TAs Yixuan Qiu and Bingya Liu for debugging my code and also my friends who have taken CS151 before--Roujia Zhong, Di Luo, Luhang Sun--for helping figuring out problems associated with locations and size of the images.