Joshua Villacorta

CSCI4810

Programming Assignment #4

In this report, I’m going to be using the example shape given to us in the instructions.

The screen resolution I used is 800 x 800. I would’ve done 1000 x 1000 but it’s too big for my laptop screen.

I also used the same D and S values from the instructions for my project, as well as having the viewport the same as the screen resolution.

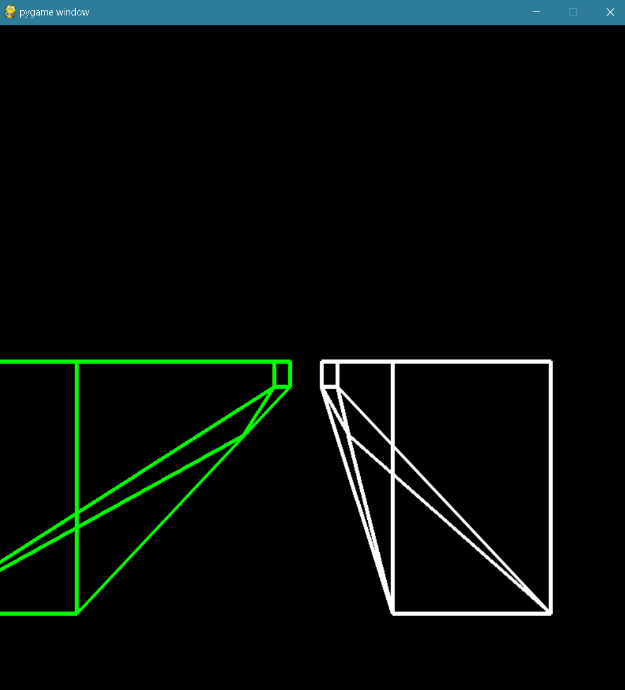
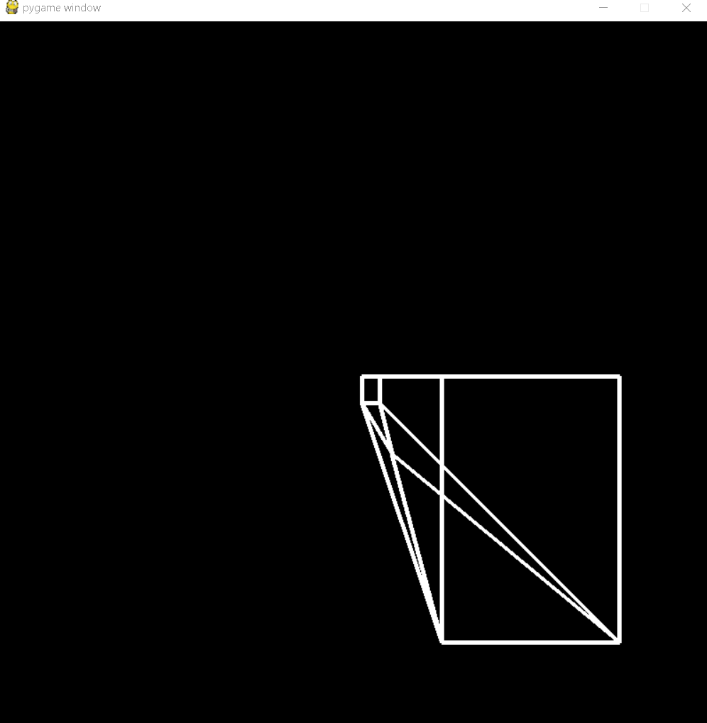
I used the pygame module in Python for displaying the shape. By nature, the pygame window has its origin start from the top left corner instead of the bottom left corner like traditional grid and so the image is going to be flipped in the screenshots I will provide.

In my screenshots, the shape with white lines represents the original shape, while the shape with green lines represents the transformed shape. I programmed it so that once you view the pygame window, it shows the original shape first, then when you click inside the screen, it views the transformed shape. I showed it in the first two screenshots of the transformations just so you can see what’s supposed to happen.

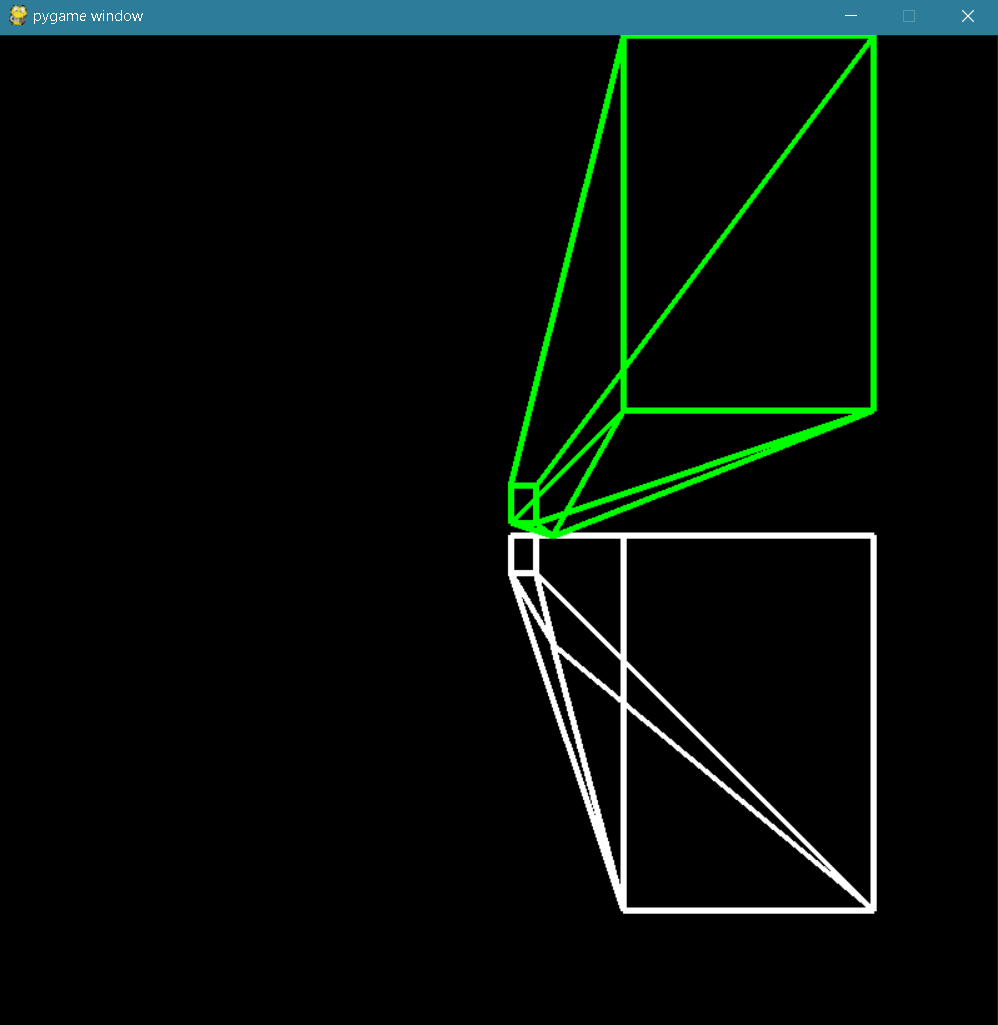
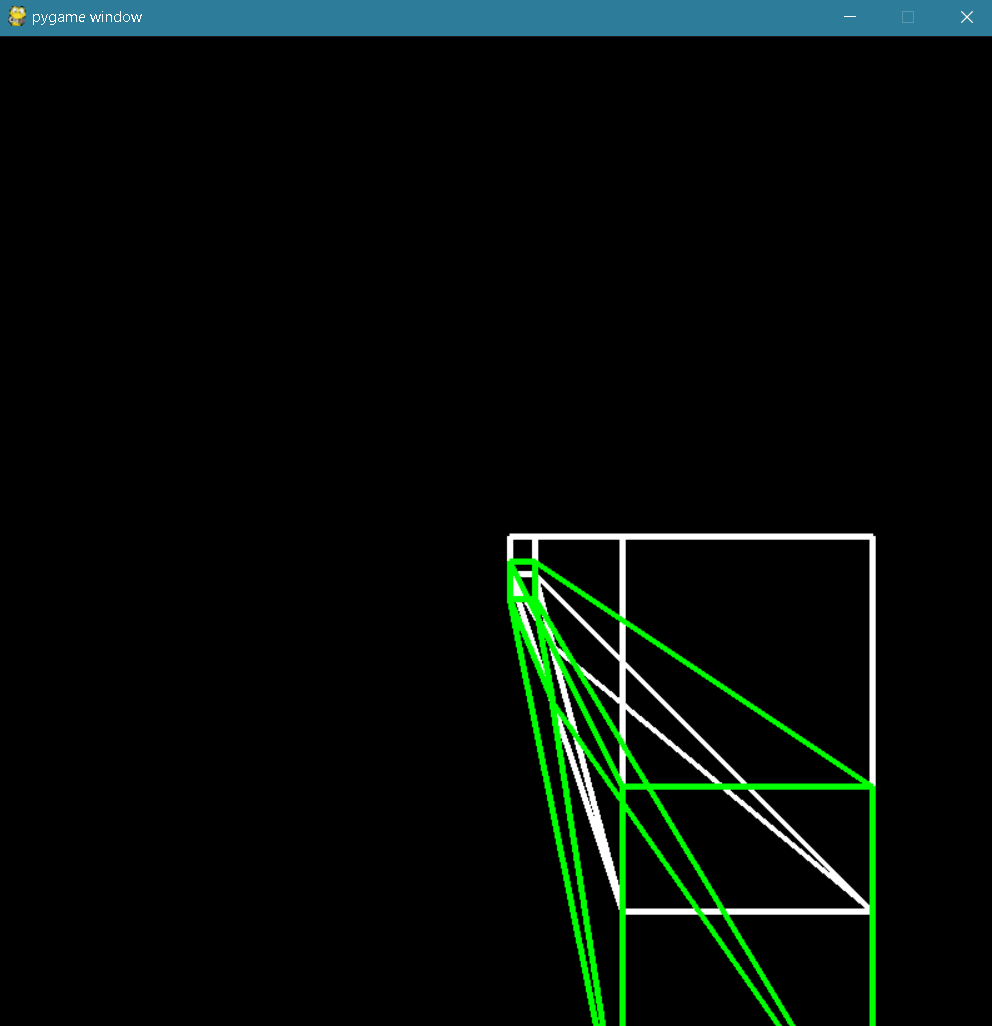


For my program, I had the user pick from the different ways to transform the shape by picking the corresponding number. Depending on the type of transformation, it would ask the user for the translation or scaling factor, or the angle they want to rotate.

**Translation through X-Axis**

Tx value = -200 (Ty and Tz values are kept to zero)

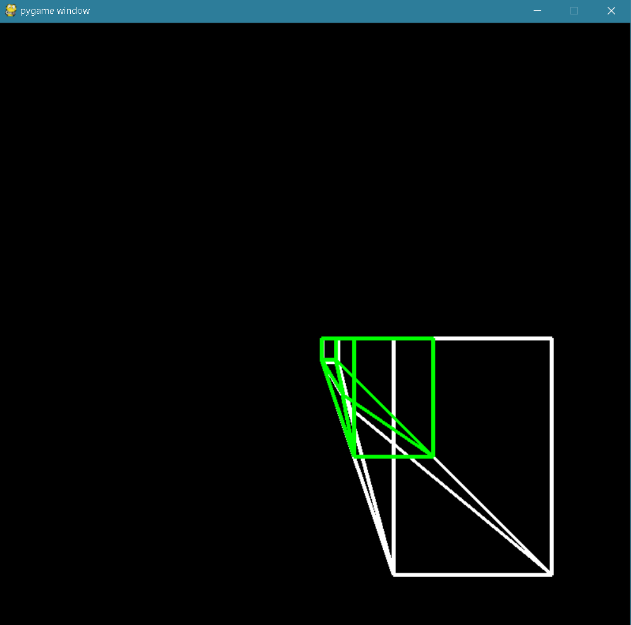
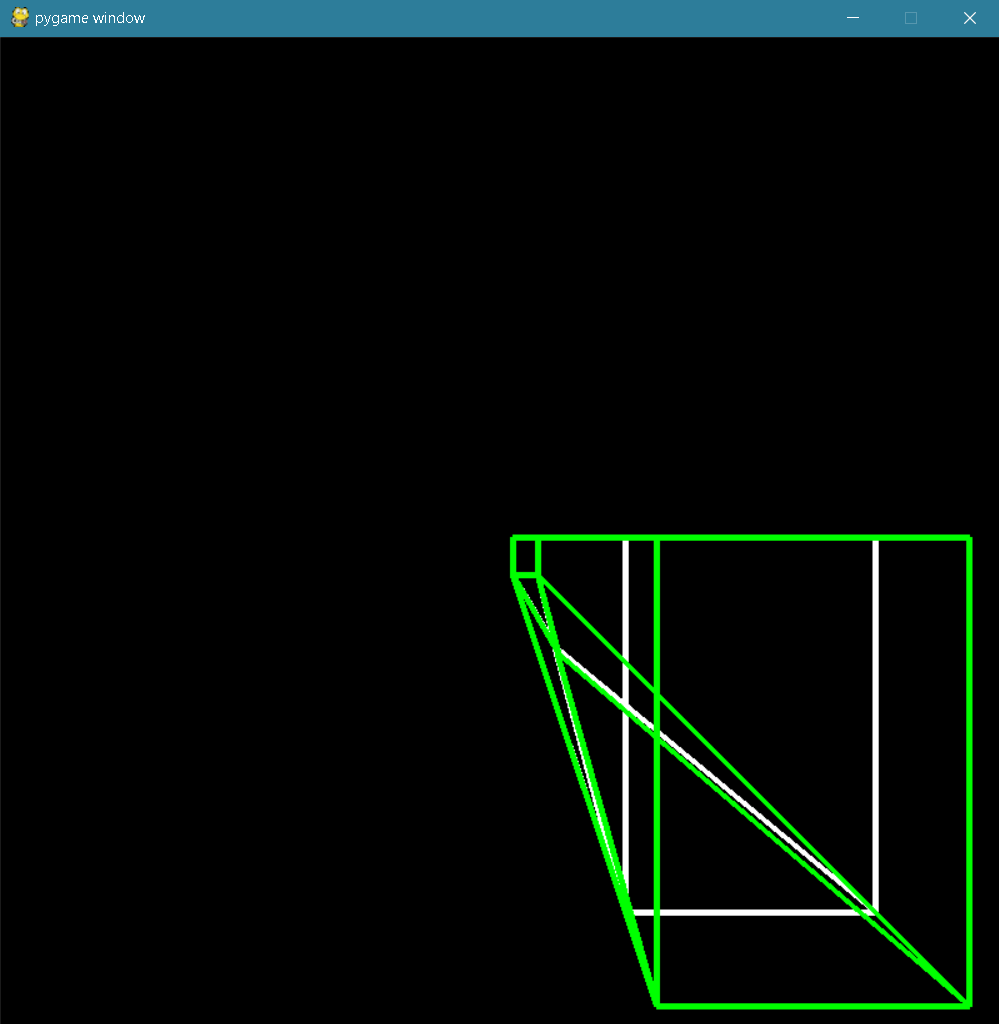
In the screenshots above, I translated the shape by -300 pixels and didn’t change the y or z coordinates. As expected, the shape is shifted to the left while maintaining the same y-axis and z-axis coordinates.

**Translation through Y-Axis**

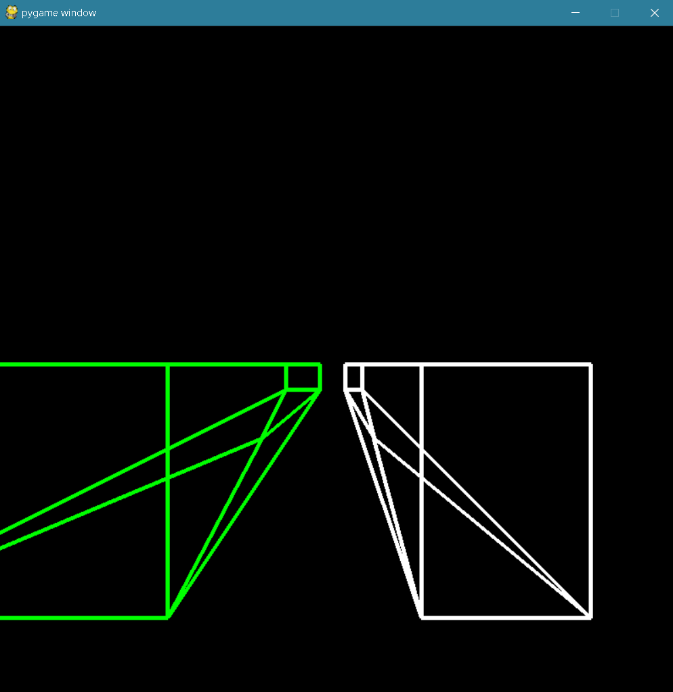
Top image: this shows the result of translating through y-axis by -200. Because the pygame window is flipped, instead of going down like a normal graph, the image goes up.

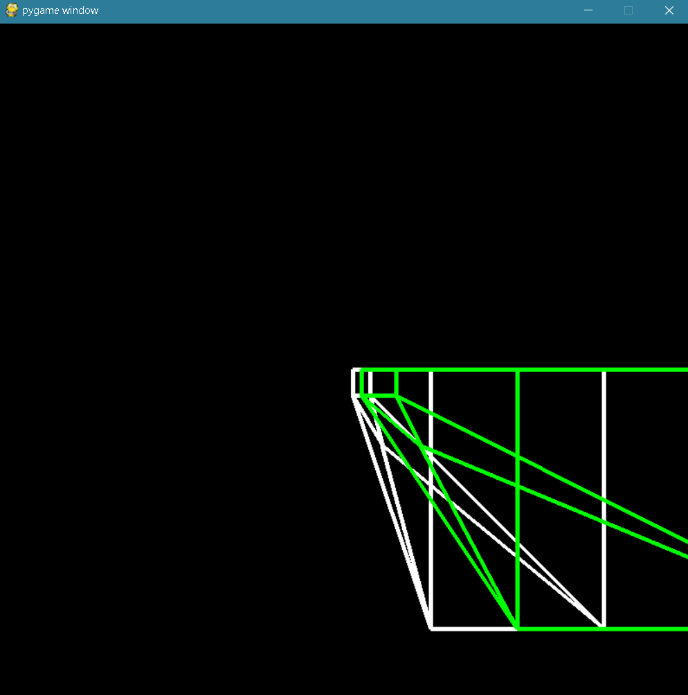
Bottom image: this shows the result of translating through y-axis by 100. Now the image looks like it’s going downwards.

**Translation through Z-Axis**

  
The image on the left shows the result of translating with a Tz value of 10. As expected, the image seems to be smaller and farther away from sight.

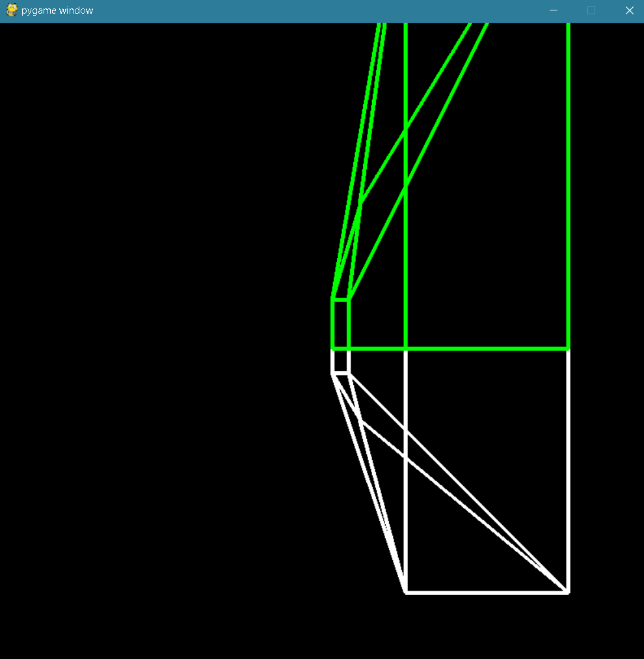
The image on the right shows the result of translating with a Tz vale of -2. As expected, the image looks larger and closer to the screen and the viewer.

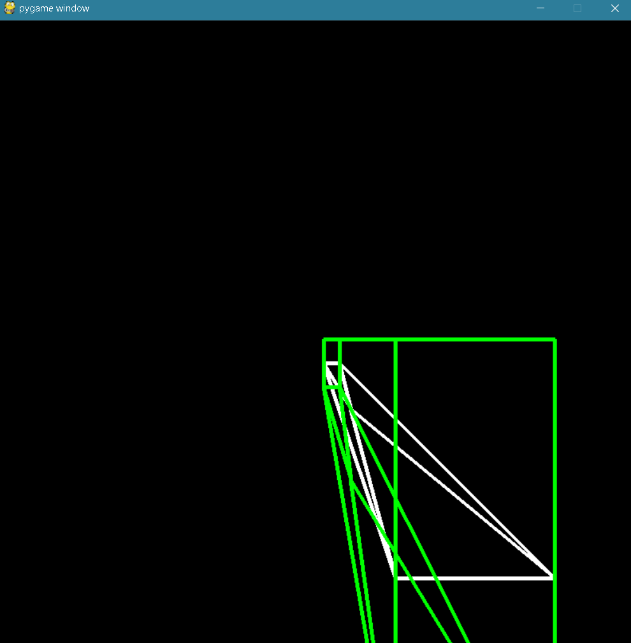
**Scale (x-axis)**



Left image: scaling by a factor of 2. The image gets wider and moves to the right, which is expected.

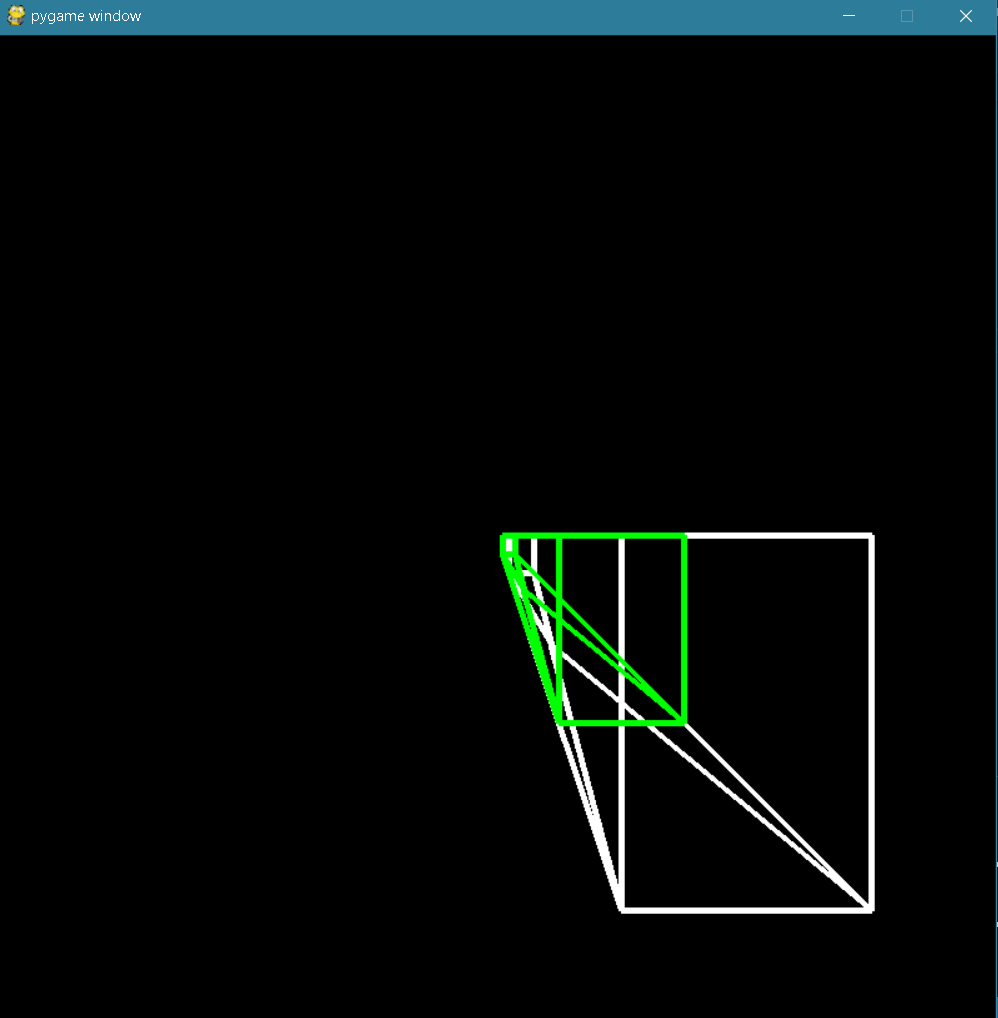
Right image: scaling by a factor of -2. The image also gets wider but it now moves to the left.

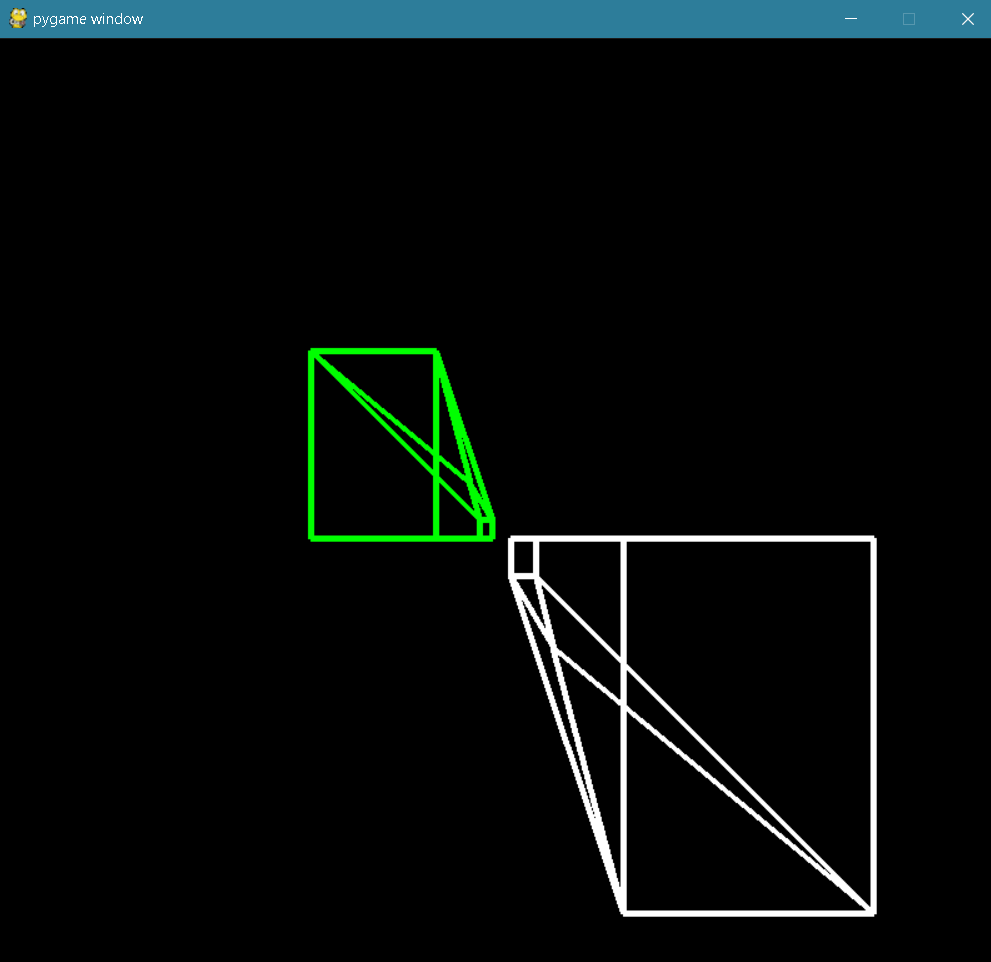
**Scale (Y-Axis)**



Left image: scale by factor of 2. The image looks taller and goes downward, which is expected since the y-axis starts from 0 at the very top and ends with 800 at the bottom.

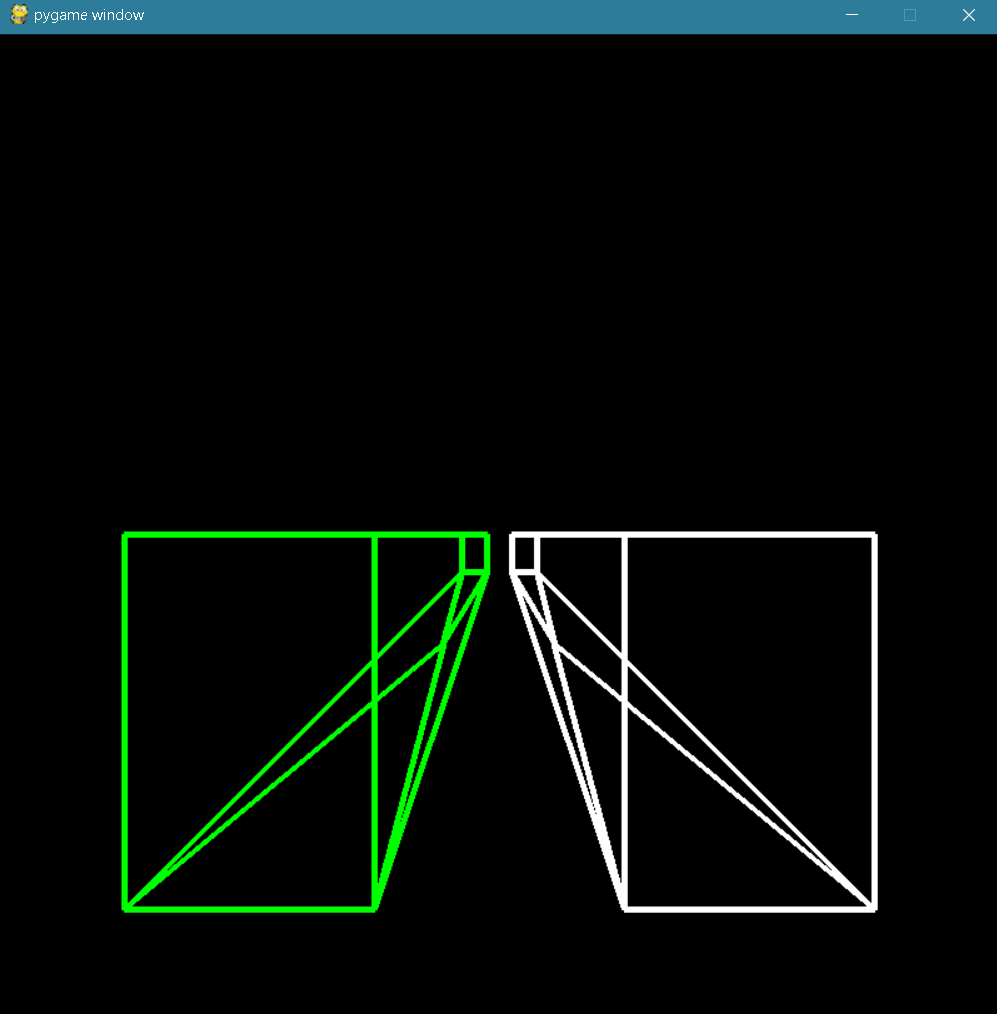
Right image: scale by factor of -2. The image also looks taller but goes upward now.

**Scale (Z-Axis)**

Top image: scaled by factor of 2. The image looks smaller as expected when only the z-axis is scaled.

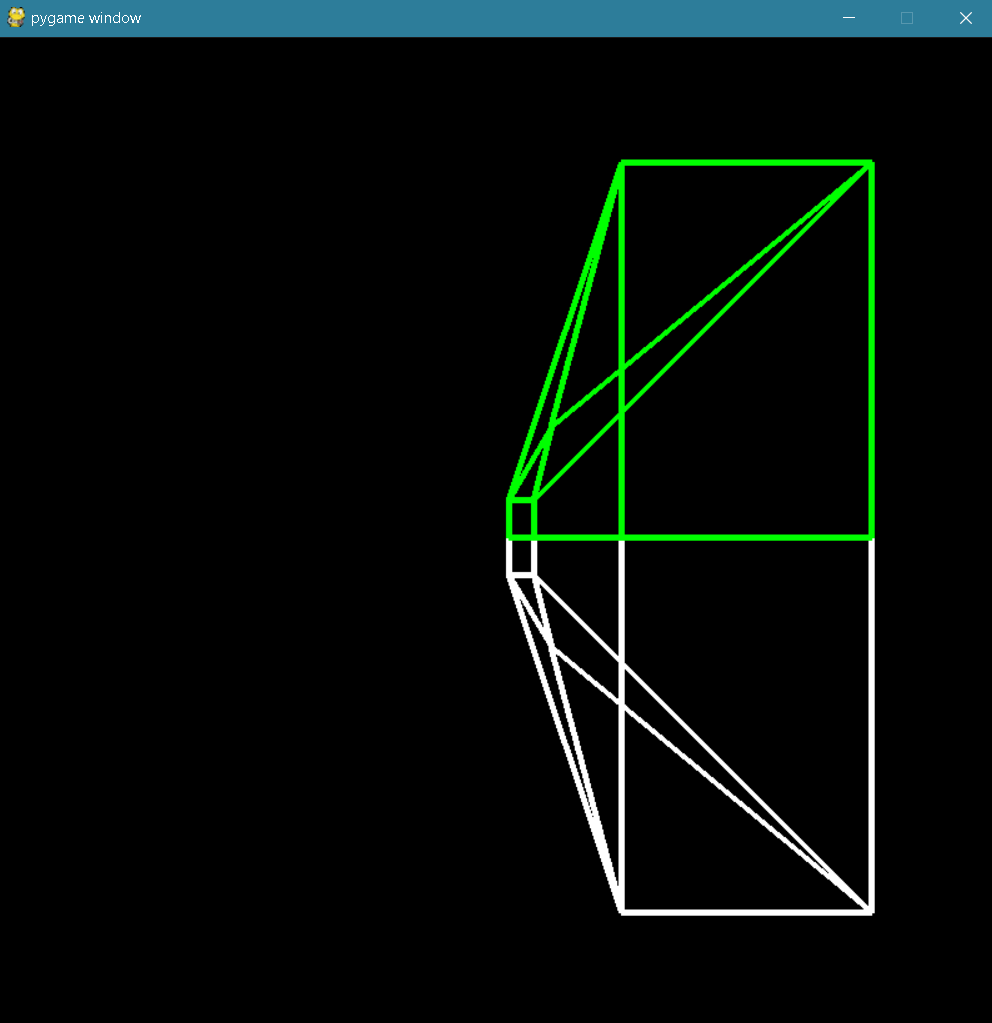
Bottom image: scaled by factor of -2. The image now looks flipped and is going farther away from the user.

**Rotation (X-Axis)**



Here, the shape is rotated 180 degrees on the x-axis. As expected, the transformed shape looks like a mirror flip of the original shape.

**Rotation (Y-Axis)**



Here, I rotated around the y-axis by 180 degrees. The transformed shape now looks like it’s been flipped vertically which is expected.

**Rotation (Z-Axis)**

Result of shape rotated 180 degrees around the z-axis.

