Sairam Soundararajan

CMSC405: 6381 Computer Graphics (2228)

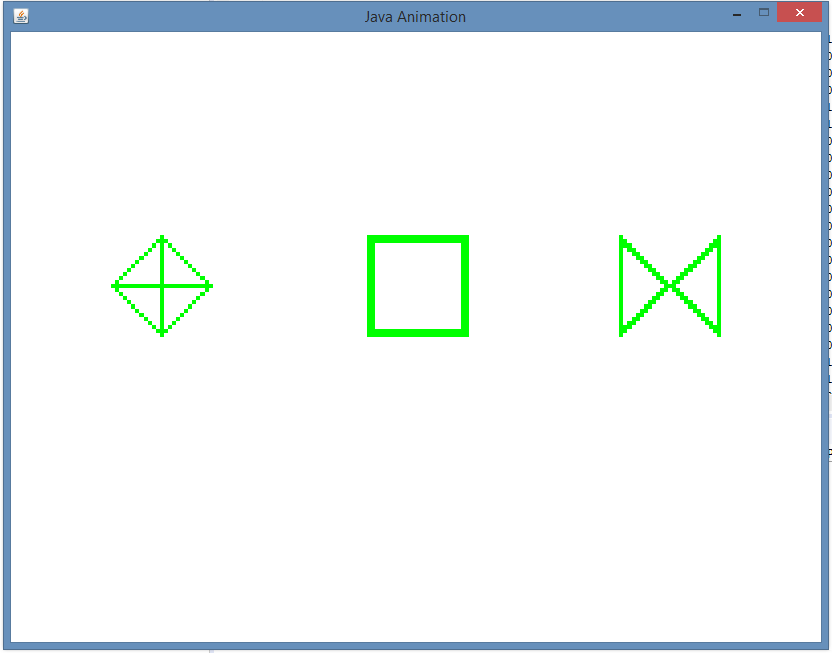
University of Maryland Global Campus

Professor: Lauren King

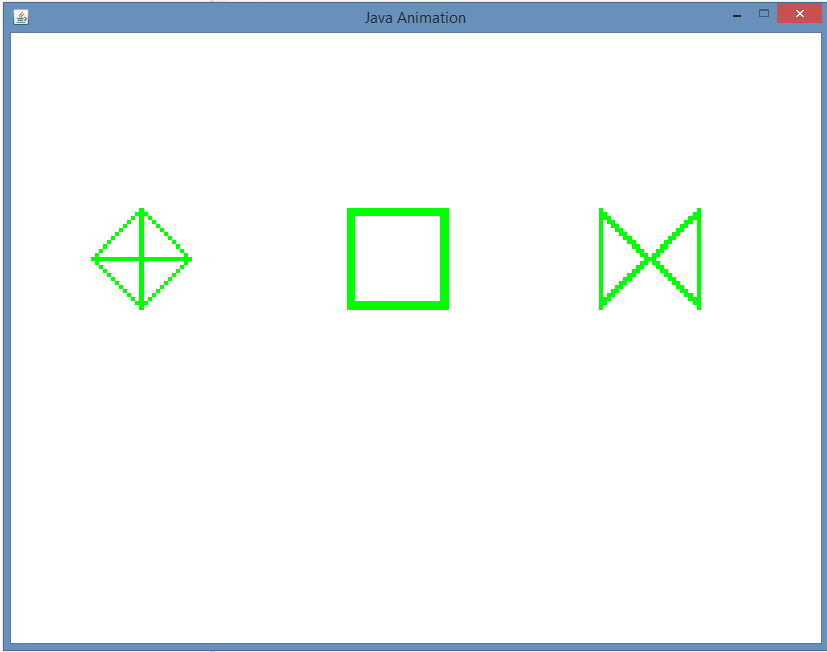
Date: August 29th, 2022

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case | Input | Expected Output | Actual Output | Pass? |
| 1 | Translation | Must translate each image -5 in x direction and +7 in the y direction | Images translated -5 in x direction and +7 in the y direction | YES |
| 2 | Rotation 1 | Must rotate each image 45 degrees counterclockwise. | Images rotated 45 degrees counterclockwise | Yes |
| 3 | Rotation 2 | Must rotate each image 90 degrees clockwise | Images rotated 90 degrees clockwise | Yes |
| 4 | Scaling | Must scale each image 2 times for the x component and 0.5 times for the y component | Images are scaled 2 times for the x component and .5 times for the y component | Yes |

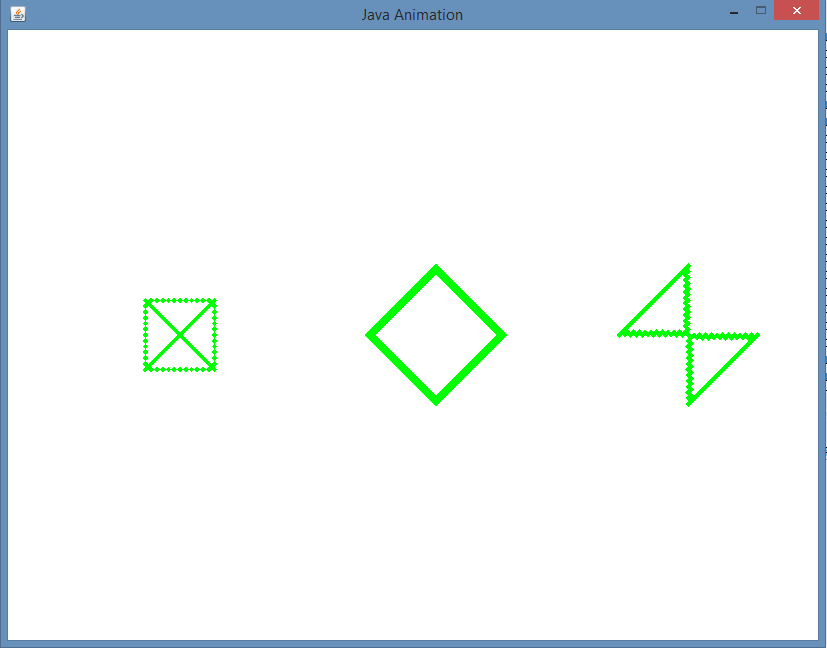
**Snapshots of Test Runs:**



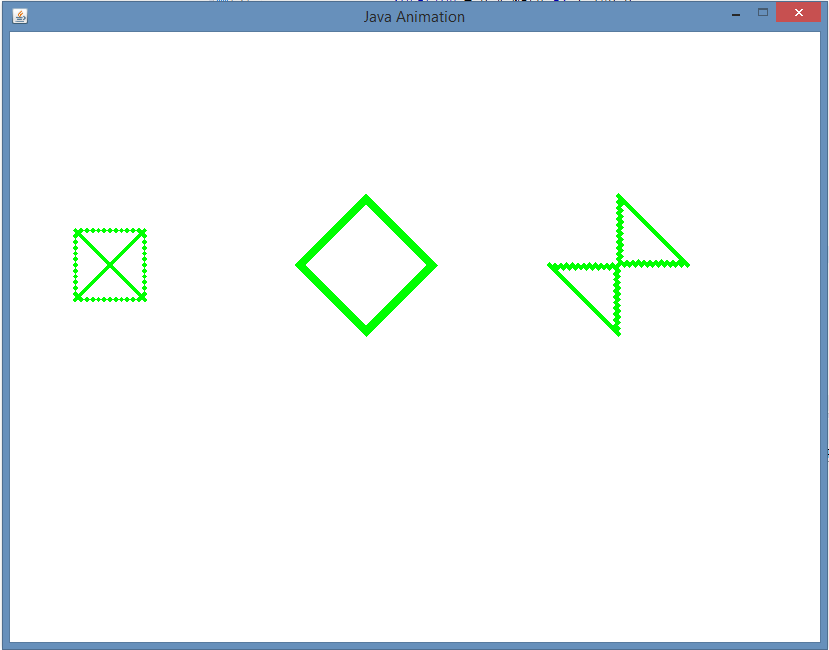
Images at starting places



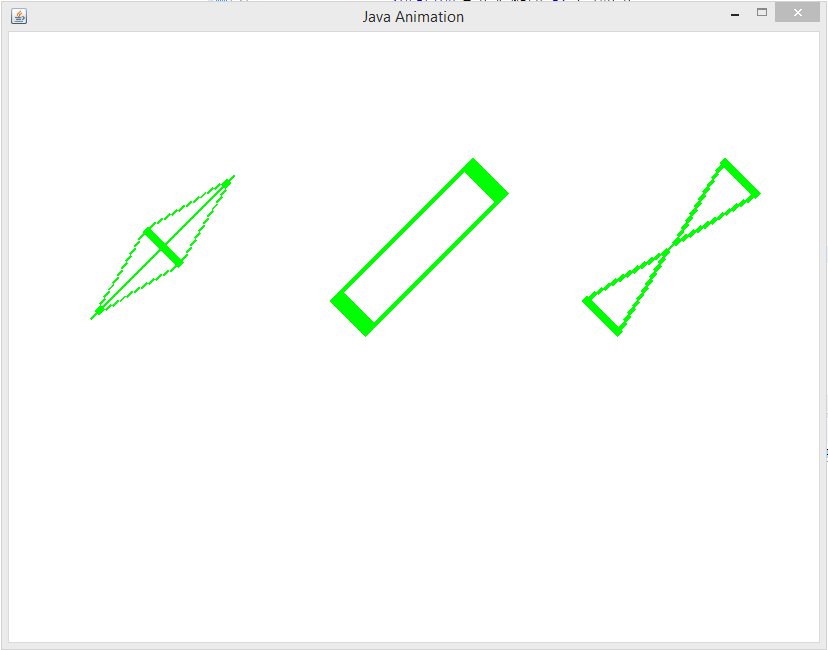
Images translated -5 on x and +7 on y



Images rotated 45 degrees counterclockwise



Images rotated 90 degrees clockwise



Images scaled 2 times on x component and .5 times on y component

Lesson Learned:

For this project, I used Eclipse to work on with Java 2D Graphics. I was able to modify the main class by declaring buffered images that would get used in the image template as 2D arrays and using the switch statement to make each case the specified translation, rotations, and scaling per project requirements. For each image, I specified 3 different horizontal translations when setting the transformations. Additionally, I also modified the image template by initializing the constants values to 25 for 25 x 25. LetterT does not get used. Instead, I initialized 3 images as a 2D array that stores 0’s and 1’s. The rows and columns can be thought of as a square matrix since it is 25 by 25. For my three shapes, I decided to make a square, kite, and an hourglass (which appears sideways). With just few modifications, I was able to get the program to translate the shapes in the specified vertical and horizontal directions, rotations clockwise and counterclockwise, and scaling for x and y component. Despite not being able to make a circle or complex shape, my favorite part was learning how to make three shapes of my favorite choice. I referred to the BImage and a few 2D graphic book example programs that were provided in the course content. I was surprised it did not take long since it only required a few modifications. I am looking forward to working more with Java 2D Graphics and expanding my knowledge of it. So far, I enjoyed working with Java 2D Graphics.

References

UMGC. (n.d.). *BImage Template*. CMSC405. Retrieved from https://learn.umgc.edu/d2l/le/content/719405/viewContent/26836710/View.

UMGC. (n.d.). *Java 2D*. CMSC405. Retrieved from https://learn.umgc.edu/d2l/le/content/719405/viewContent/26836709/View.

UMGC. (n.d.). *Project 1 Instructions*. CMSC405. Retrieved from https://learn.umgc.edu/d2l/common/viewFile.d2lfile/Database/NjE2MTU0NjE/Project%201.pdf?ou=719405.

UMGC. (n.d.). *Project 1 Templates*. CMSC405. Retrieved from https://learn.umgc.edu/d2l/le/content/719405/viewContent/26836719/View.