Project 1: Java 2D Graphics

Nabeel Hussain

University of Maryland, University College

CMSC 405 – 7980

March 26th, 2017

/\*\*

\* This class will create 3 simple, binary 25x25 images or your choice and

\* use Java 2D graphic methods to rotate, scale and translate each of the

\* images.

\* The drawing code goes in the paintComponent() method. When the program

\* is run, the drawing is shown in a window on the screen.

\*

\* Name: Nabeel Hussain

\* Class: CMSC 405

\* Professor: Catlin Tudose

\* Project 1

\* Date: 3/26/2017

\*

\* **@author** Nabeel Hussain

\*/

Project 1 involves creating three simple 25x25 binary images that are each stored in a 2D array. The images are then used to perform some basic transformations, which include translating, rotating, and scaling them.

For each image use the existing Java 2D graphics transformation methods to translate, rotate and scale each object. You should perform the following transformations on each image:

a. Translate -5 in x direction, Translate +7 in the y direction.

b. Rotate 45 degrees counter clockwise.

c. Rotate 90 degrees clockwise

d. Scale 2 times for the x component, scale 0.5 times for the y component

e. Each of these transformations should be displayed in sequence with the images always starting from the previous transformation as opposed to the original image.

f. Use Java 2D graphics to display each transformation for each image. (Hint: you can do this with loop and slight pause for each display)

**Test Plan:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Input:** | **Expected Output** | **Did Test Pass?** |
| 1 | Should first display the untransformed original images by calling the displayOriginalImage() method for each image. | Top-left point on image:  Displays image “F” at location (0,0)  Displays image “E” at location (100,0)  Displays image “H” at location (200,0) | Y |
| 2 | Should perform the first transformation on the images by calling the firstImageTransformation() method for each image. | All images moved -5 in x direction and +7 in y direction:  Top-left point on image:  Displays image “F” at location (-5,7)  Displays image “E” at location (95,7)  Displays image “H” at location (195,7) | Y |
| 3 | Should perform the second transformation on the images by calling the secondImageTransformation() method for each image. | All images rotated 45 degrees counter clockwise from previous transformation:  Top-left point on image:  Displays image “F” at location (-5,7)  Displays image “E” at location (95,7)  Displays image “H” at location (195,7) | Y |
| 4 | Should perform the third transformation on the images by calling the thirdImageTransformation() method for each image. | All images rotated 90 degrees clockwise from previous transformation:  Top-left point on image:  Displays image “F” at location (-5,7)  Displays image “E” at location (95,7)  Displays image “H” at location (195,7) | Y |
| 5 | Should perform the fourth transformation on the images by calling the fourthImageTransformation() method for each image.. | All images scaled 2 times for x component and .5 times for y component from the previous transformation. | Y |

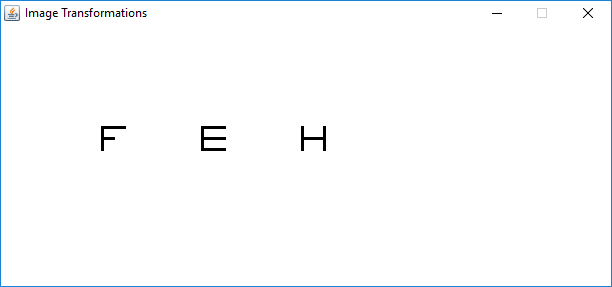
**Screen shot of successful compilation for Test Cases**

Program is run as a continuous animation witha1500 millisecond pause before the next transformation occurs.

\* The origin (0,0) is originally at the top left of the default coordinate system. However, I moved it down 100 and to the right 100 so that the transformations can be easily visible. Thus, the origin (0,0) is now located where the top left pixel of the letter “F” is shown. \*

**Test Case 1:** Displays the untransformed original images

1. Displays the three untransformed original images side by side on the window. The top left pixel of the first image “F” is at location (0,0). The top left pixel of the second image “E” is located at (100, 0), and the top left pixel of the third image “H” is at (200,0). There is a 1500 millisecond pause between each transformation.



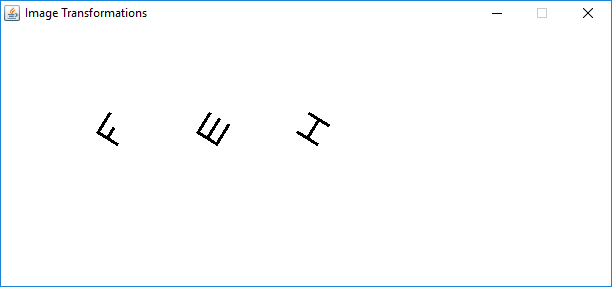
**Test Case 2:** Displays the first transformation

2. The translation of -5 in the x direction and 7 in the y direction. (1500 millisecond pause)



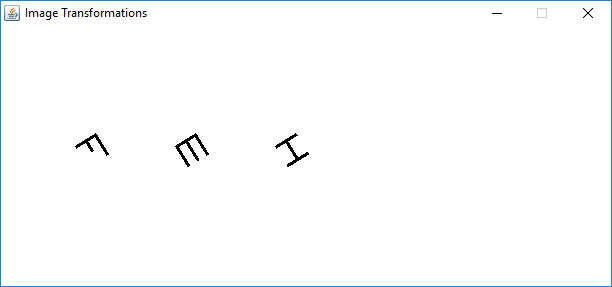
**Test Case 3:** Displays the second transformation

3.  The rotation of 45 degrees counter clockwise from the previous transformation. (1500 millisecond pause)



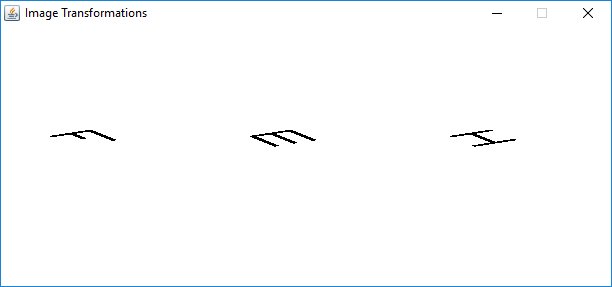
**Test Case 4:** Displays the third transformation

4.  The rotation of 90 degrees clockwise from the previous transformation. (1500 millisecond pause)



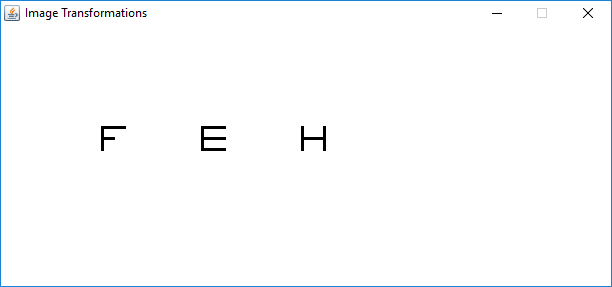
**Test Case 5:** Displays the fourth transformation

5. The scaling of 2 times for the x component and 0.5 times for the y component, from the previous transformation. (1500 millisecond pause)



End of first loop, at which point the counter resets so we can begin the sequences again.

6. Shows the three untransformed original images again. (1500 millisecond pause)

“

Etc….

**Lessons Learned:**

While working on Project 1, there were a lot of new concepts and techniques that I learned about computer graphics, such as how images are created, and how they can be used to perform transformations and animations. I had some basic knowledge about using and manipulating images in Java from previous courses, however most of the concepts I learned to complete this project were new to me. As a result, I had some initial difficulty getting started with the project, but once we were given a bit more detail I was able to complete the project and get everything to work.

For this project, I choose to create my three 25x25 binary 2D array images as the letters “F,” “E,” and “H.” I created an animation to display the transformations using the swing Timer class and a loop, to show the different transformations in sequence with a slight pause after each one occurs. All three images are displayed side by side and the transformations are performed stimulatingly on all three of them at the same time. I used a counter variable to keep track of which transformation is being performed, and once the final one has taken place the counter resets so that the animation can repeat.

Overall, this assignment was a good introduction to computer graphics, as it helped me understand and reinforce all the material I learned from the first two weekly reading assignments. I learned about bitmaps and image pixels, and how to convert them into BufferedImages. I learned about the Java Graphics2D class and API, and how to manipulate images by using the translate, rotate, and scaling functions. I also learned about animations, and how to move images using the swing Timer class.