**Project 2: JOGL OpenGL Project**

Samuel B. Scalf

School of Cybersecurity and Information Technology, University of Maryland Global Campus

CMSC 405: Computer Graphics

Dr. Cynthia V. Marcello

8 February 2022

Test Cases

**Table 1**

*Test Cases for JOGL transformations*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Function | Expected Output | Actual Output | Pass/Fail |
| W/↑ | Rotate on negative X-axis | Scene should rotate on negative X-axis | Scene rotated on negative X-axis | Pass |
| A/← | Rotate on negative Y-axis | Scene should rotate on negative Y-axis | Scene rotated on negative Y-axis | Pass |
| S/↓ | Rotate on positive X-axis | Scene should rotate on positive X-axis | Scene rotated on positive X-axis | Pass |
| D/→ | Rotate on positive Y-axis | Scene should rotate on positive Y-axis | Scene rotated on positive Y-axis | Pass |
| Q/Page Up | Rotate on positive Z-axis | Scene should rotate on positive Z-axis | Scene rotated on positive Z-axis | Pass |
| E/Page Down | Rotate on negative Z-axis | Scene should rotate on negative Z-axis | Scene rotated on negative Z-axis | Pass |
| I | Zoom In/Scale up | Scene should get larger (i.e. zoom in) | Scene got larger (i.e. zoomed in) | Pass |
| O | Zoom Out/Scale down | Scene should get smaller (i.e. zoom out) | Scene got smaller (i.e. zoomed out) | Pass |

*Note*. The function in each row of this table is repeated, at least seemingly, continuously while the associated key is pressed.

Source Code

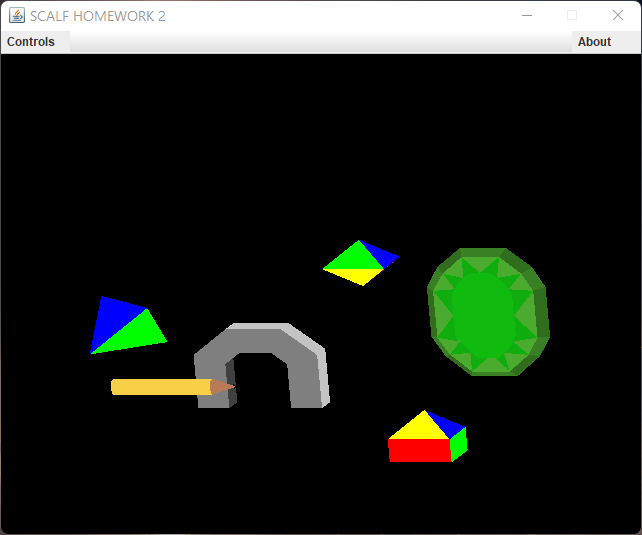
The source code for this project contains three Java classes: CMSC405P2, CMSC405P2Shape, and CMSC405P2Transformation. CMSC405P2 contains the main method of the program, extends GLJPanel, and implements GLEventListener and KeyListener: it drives the program and user interaction. CMSC405P2Shape is a class with only a private constructor that instantiates the three package-private final variables: vertices, faces, and faceColors. All shapes used in the program are defined in this class as public static final variables. This has a similar effect as an enum: Class.VARIABLE is like Enum.NAME. Finally, CMSC405P2Transformation is a bean that holds variables for the x-, y-, z- coordinates and scale for the images when drawn. I didn’t want the methods to have too many parameters, so I wrapped up all parameters I wanted in a “transformation” class. This made it clearer what was happening in the code.

Screen Captures

Unlike the previous project, I did not need to create any breakpoints to pause due to no active animations being present. Instead, I took a screenshot of the default position of the elements as well as one for each of the controls’ effects. For controls that have multiple keys assigned, I only captured one screenshot because it is not possible to show the different effects. The only exception is the one screenshot taken after pressing “Home” to reset the positions. In between all screenshots, however, I did use the “Home” button to reset the positions to better display the effect of the buttons. In addition, I pressed each control button four times to further this impression.

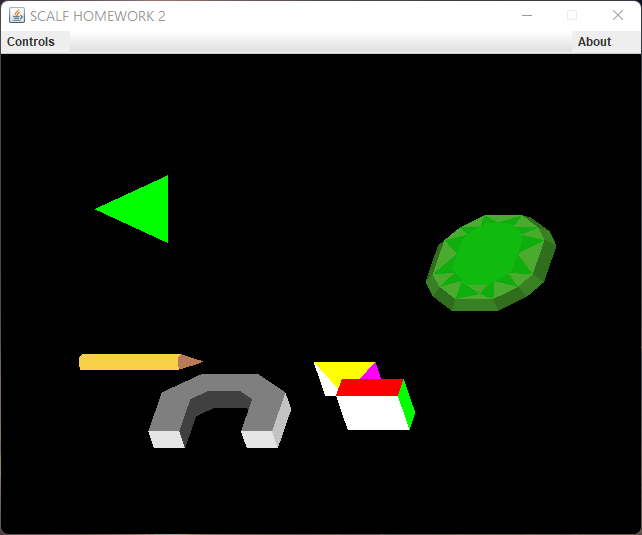
**Figure 1**

*Screen Capture of the Default Configuration*



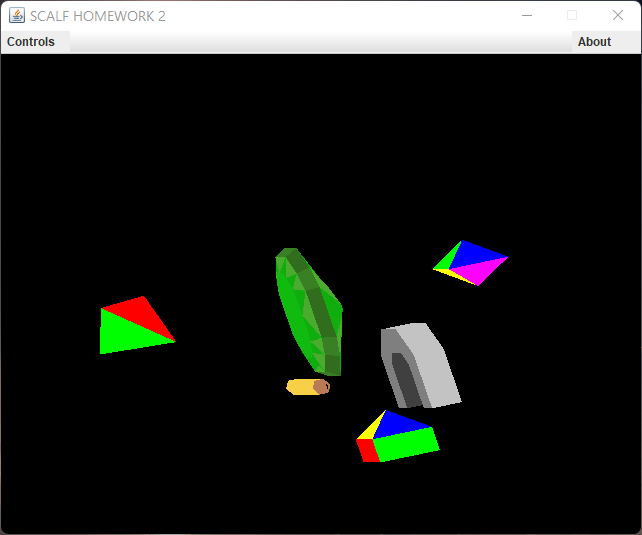
**Figure 2**

*Screen Capture of W/↑: Negative Rotation on X-Axis*

**

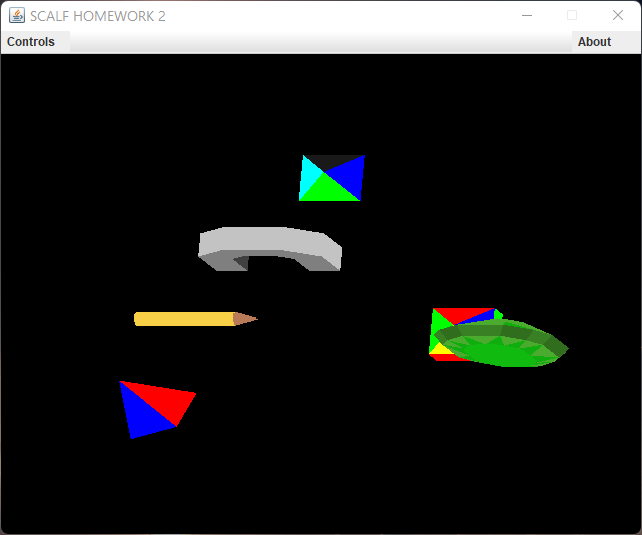
**Figure 3**

*Screen Capture of A/←: Negative Rotation on Y-Axis*



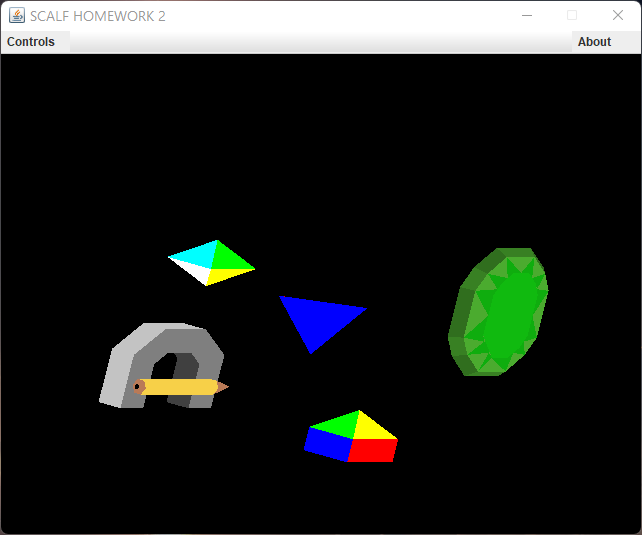
**Figure 4**

*Screen Capture of S/↓: Positive Rotation on X-Axis*



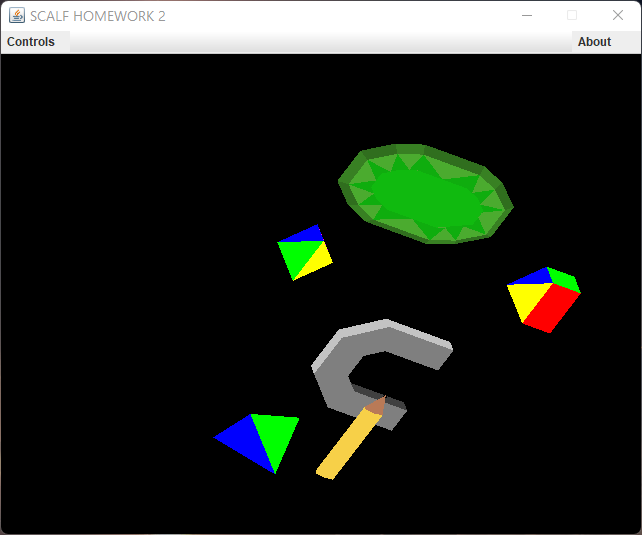
**Figure 5**

*Screen Capture of D/→: Positive Rotation on Y-Axis*



**Figure 6**

*Screen Capture of Q/Page Up: Positive Rotation on Z-Axis*



**Figure 7**

*Screen Capture of E/Page Down: Negative Rotation on Z-Axis*



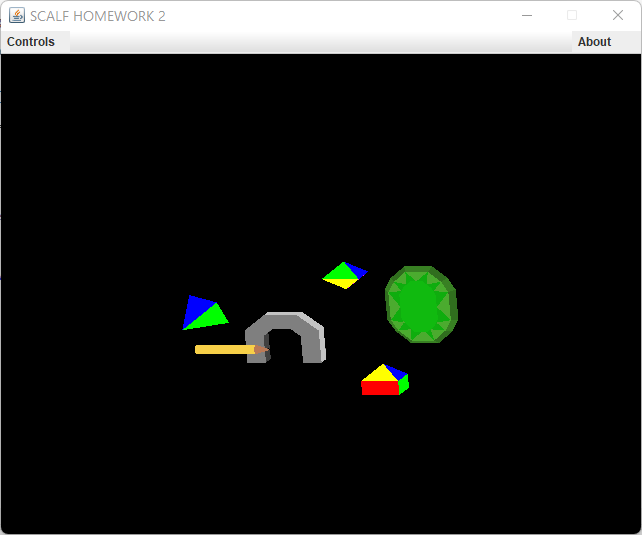
**Figure 8**

*Screen Capture of I: Zoom In or Scale Up*



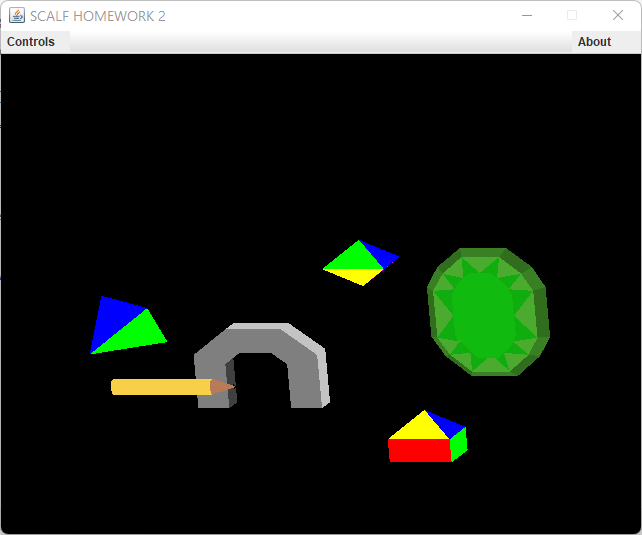
**Figure 9**

*Screen Capture of O: Zoom Out or Scale Down*



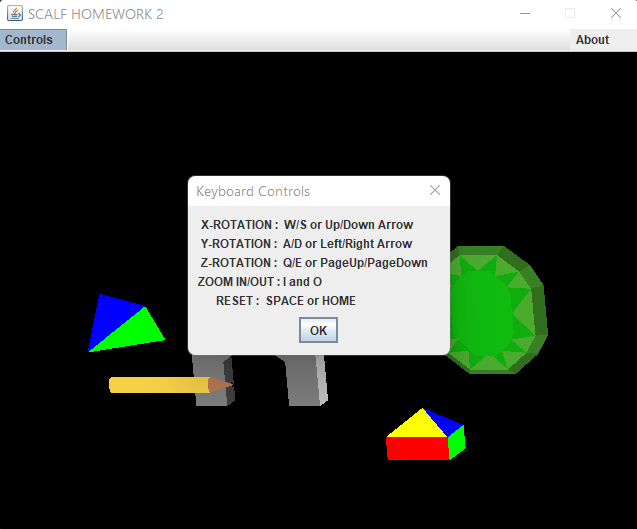
**Figure 10**

*Screen Capture of Home: Reset to Default*



**Figure 11**

*Screen Capture of Controls Window*



**Figure 12**

*Screen Capture of About Window*

