

```
// colors used
glm::vec4 color1 = glm::vec4(1.0f, 0.0f, 0.0f, 1.0f); // Red
glm::vec4 color2 = glm::vec4(0.564f, 0.933f, 0.564f, 1.0f); // Light Green
glm::vec4 color3 = glm::vec4(0.45f, 0.75f, 0.45f, 1.0f); // Dark Green
glm::vec4 color4 = glm::vec4(1.0f, 0.6f, 0.7f, 1.0f); // Pink
glm::vec4 color5 = glm::vec4(1.0f, 1.0f, 0.6f, 1.0f); // Yellow
glm::vec4 color6 = glm::vec4(1.0f, 0.65f, 0.0f, 1.0f); // Orange
glm::vec4 color7 = glm::vec4(0.0f, 0.0f, 0.55f, 1.0f); // Deep Blue
glm::vec4 color8 = glm::vec4(0.68f, 0.85f, 0.9f, 1.0f); // Light Blue
```

Figure 1: Code for colors definition to use

```
// vase drawing and coloring
glBindVertexArray(VA02);
glDrawArrays(GL_LINE_STRIP, 0, 61);
glUniform4fv(colorLocation, 1, glm::value_ptr(color8)); // vase top deep blue
glDrawArrays(GL_TRIANGLE_FAN, 61, 25);
glUniform4fv(colorLocation, 1, glm::value_ptr(color7)); // vase body light blue
glDrawArrays(GL_TRIANGLE_FAN, 0, 61);
```

Figure 2: Vase points drawing

```
// flowers drawing and coloring
glBindVertexArray(VA01);
glDrawArrays(GL_LINE_STRIP, 0, 11);
glDrawArrays(GL_LINE_STRIP, 11, 45);
glDrawArrays(GL_LINE_STRIP, 56, 23);
glDrawArrays(GL_LINE_STRIP, 79, 26);
glDrawArrays(GL_LINE_STRIP, 105, 70);
glDrawArrays(GL_LINE_STRIP, 175, 30);
glDrawArrays(GL_LINE_STRIP, 205, 26);
glDrawArrays(GL_LINE_STRIP, 231, 34);
glDrawArrays(GL_LINE_STRIP, 265, 17);
glDrawArrays(GL_LINE_STRIP, 282, 42);
glDrawArrays(GL_LINE_STRIP, 324, 24);
glDrawArrays(GL_LINE_STRIP, 348, 30);
```

Figure 3: Flowers and leaves points drawing

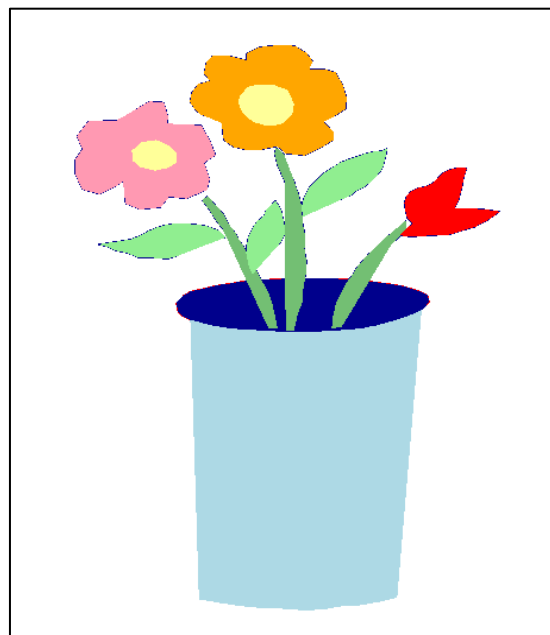
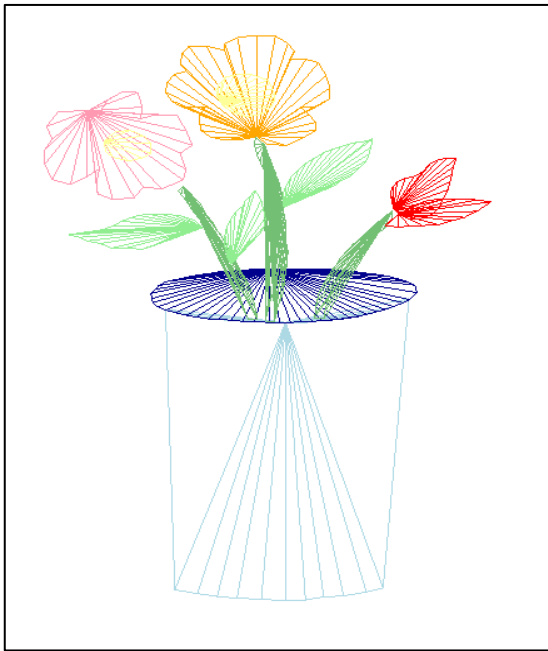
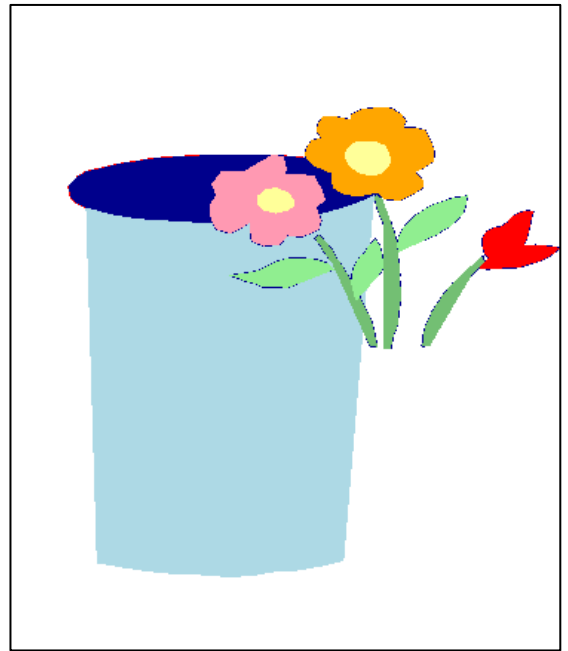


Figure 4: Flowers and Vase



(a)



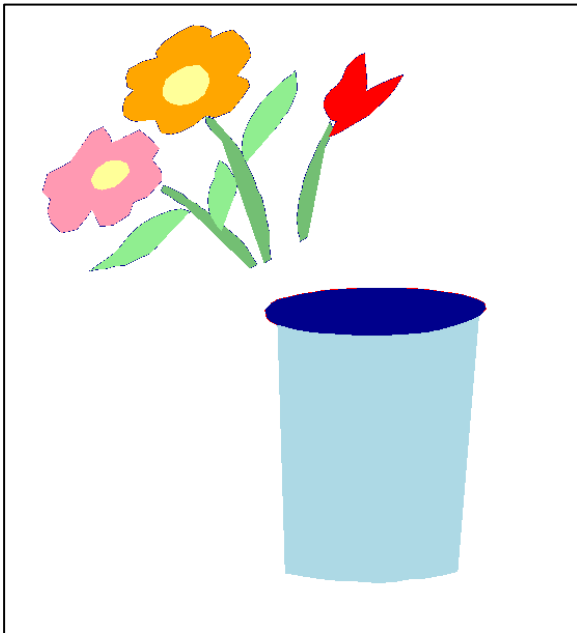
(b)

Figure 5: (a) Poly gone mode picture (b) Scaled flowers

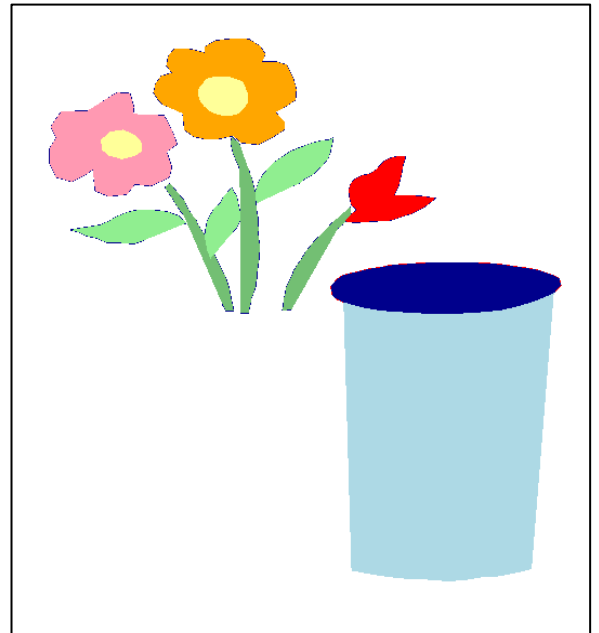
```
void processInput(GLFWwindow* window)
{
    if (glfwGetKey(window, GLFW_KEY_ESCAPE) == GLFW_PRESS)
        glfwSetWindowShouldClose(window, true);

    // For Flower Movement:
    // Rotation:
    if (glfwGetKey(window, GLFW_KEY_F) == GLFW_PRESS)
        rotateAngle += 0.03f;
    // Translation:
    if (glfwGetKey(window, GLFW_KEY_UP) == GLFW_PRESS)
        translate_Y += 0.0005f;
    if (glfwGetKey(window, GLFW_KEY_DOWN) == GLFW_PRESS)
        translate_Y -= 0.0005f;
    if (glfwGetKey(window, GLFW_KEY_LEFT) == GLFW_PRESS)
        translate_X -= 0.0005f;
    if (glfwGetKey(window, GLFW_KEY_RIGHT) == GLFW_PRESS)
        translate_X += 0.0005f;
    // Scaling:
    if (glfwGetKey(window, GLFW_KEY_EQUAL) == GLFW_PRESS)
    {
        scale_X += 0.0005f;
        scale_Y += 0.0005f;
    }
    if (glfwGetKey(window, GLFW_KEY_MINUS) == GLFW_PRESS)
    {
        scale_X -= 0.0005f;
        scale_Y -= 0.0005f;
    }
}
```

Figure 6: Code for Flowers Transformations (Rotation, Translation and Scaling)



(a)

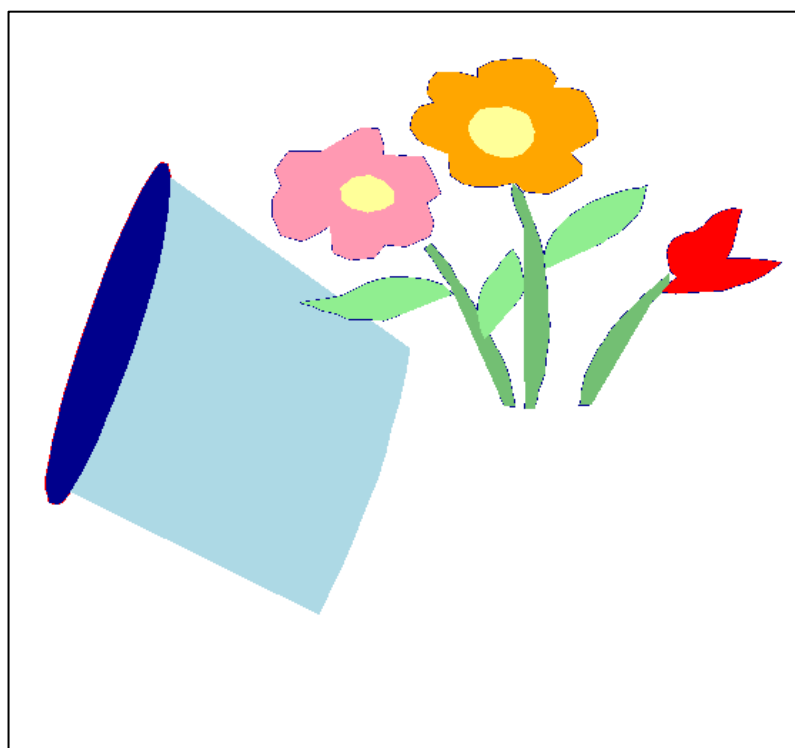


(b)

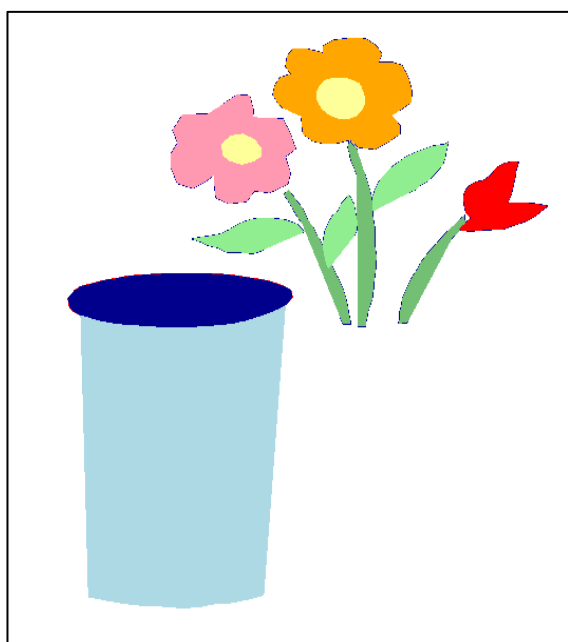
Figure 7: (a) Rotated flowers (b) Translated flowers

```
// For Vase Movement:
// Rotation:
if (glfwGetKey(window, GLFW_KEY_V) == GLFW_PRESS)
    rotateAngle2 += 0.03f;
// Translation:
if (glfwGetKey(window, GLFW_KEY_U) == GLFW_PRESS)
    translate_Y2 += 0.0005f;
if (glfwGetKey(window, GLFW_KEY_D) == GLFW_PRESS)
    translate_Y2 -= 0.0005f;
if (glfwGetKey(window, GLFW_KEY_L) == GLFW_PRESS)
    translate_X2 -= 0.0005f;
if (glfwGetKey(window, GLFW_KEY_R) == GLFW_PRESS)
    translate_X2 += 0.0005f;
// Scaling:
if (glfwGetKey(window, GLFW_KEY_P) == GLFW_PRESS)
{
    scale_X2 += 0.0005f;
    scale_Y2 += 0.0005f;
}
if (glfwGetKey(window, GLFW_KEY_M) == GLFW_PRESS)
{
    scale_X2 -= 0.0005f;
    scale_Y2 -= 0.0005f;
}
```

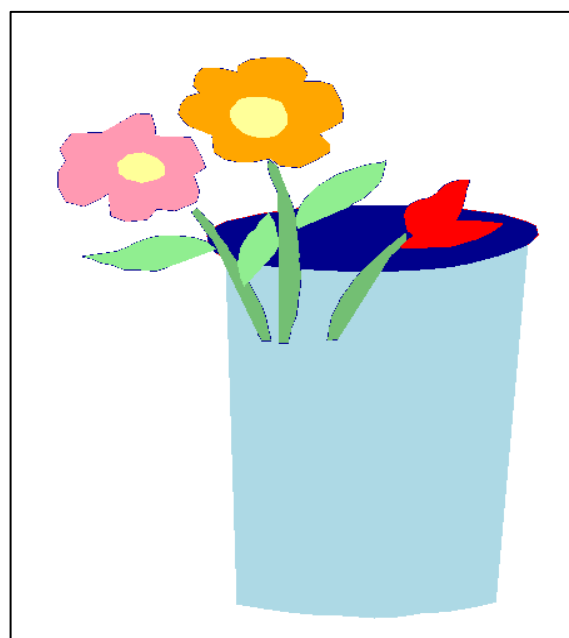
Figure 8: Code for Vase Transformations (Rotation, Translation and Scaling)



(a)



(b)



(c)

Figure 9: (a) Rotated Vase (b) Translated Vase (c) Scaled Vase