

Assignment 2: Rasterization Report

1 Mandatory tasks

1.1 Triangle Soup Editor

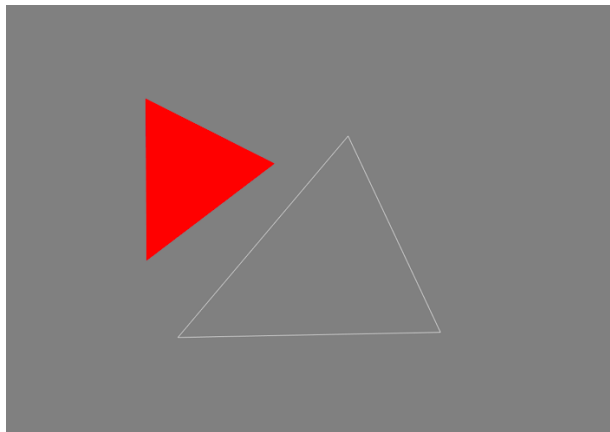
1) Main changes

- a. Added new cases for key 'i', 'o', 'p'.
- b. Set initial V to a 2*6 matrix coordinates in order to draw 3 lines.
- c. Added the `cursor_pos_callback` function to track the cursor as it moves on the screen
- d. Modify the `mouse_button_callback` function to determine the vertice coordinates to be added to V
- e. Added helper functions for each task.

2) Results

a. Insertion screenshot

On the 1st and 2nd click, lines are drawn. This screenshot is taken before the 3rd click. After the last click, the line segments will be filled the triangle becomes red.



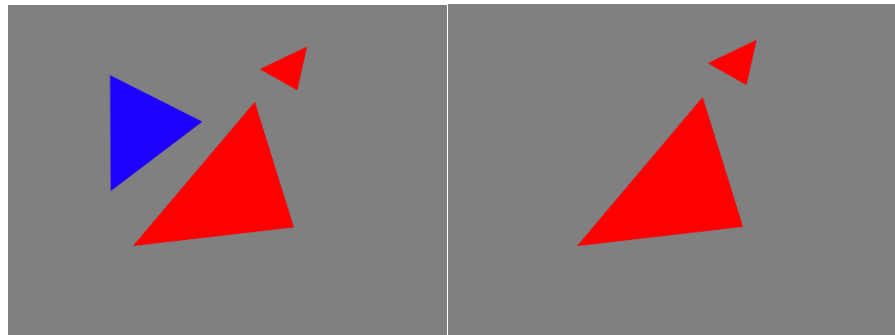
b. Translation screenshot

When translation mode is activated. As a triangle is clicked, it will turn blue.



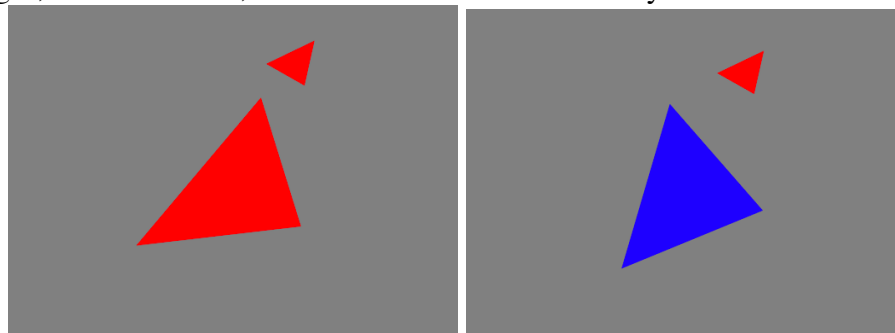
c. Deletion

Realized by traverse to vertex indices of the triangle and find the selected one, then update the V without the 3 vertices and resize it by reducing 3 columns.

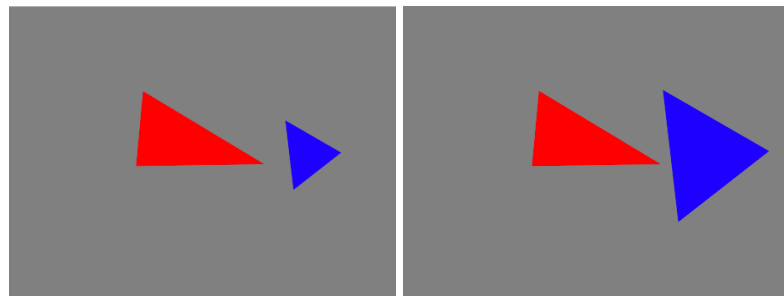
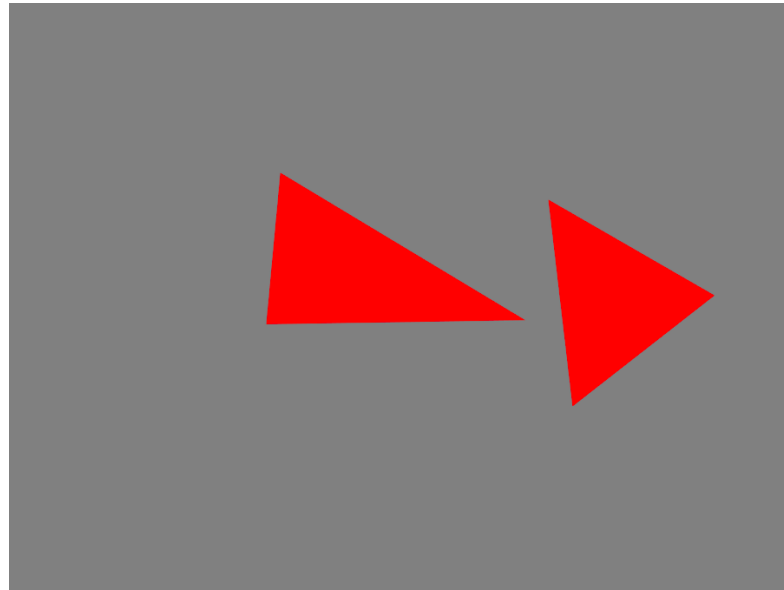


1.2 Rotation and Scale

1. Rotation around the barycenter is achieved by transform the selected triangle to the origin, do the rotation, and transform it back to its barycenter coordinates.

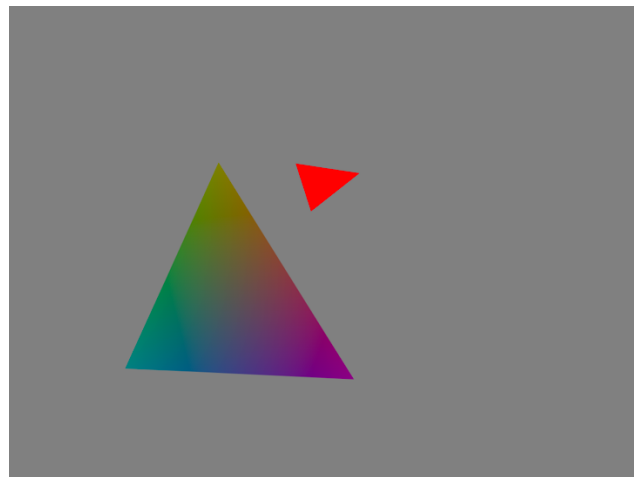


2. Similar to rotation, the result of scale is shown below. These are achieved by vertex shader.



1.3 Colors

Predefine a color choice matrix and calculate the distance between the clicked point coordinates and each vertex. Find the minimum one and set the color of the matrix to the input color choice.



1.4 View Control

1. An example of shifting left.

Change the p_screen matrix by multiply a factor and reflect the change by adding/subtracting it from the original view matrix.



2. The screenshot of zoom out and zoom in.
Directly time a factor on the view matrix.



1.5 Keyframing

By pressing key B, the translation in keyframing start. During the translation, we record the starting time t_start and animation ending time t_now . After the move, press G to play the linear interpolation between the starting and end scene

1.8 Shader Translation

In the previous steps, all transformations are stored in 4×4 matrices. As they are determined, the model matrix is calculated by multiplying them together. Using `glUniform` to map the object coordinates to the results on canvas as the vertex shader and fragment shader are ready.