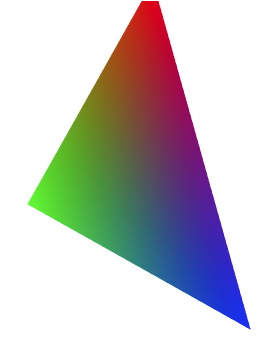
Week 1

Task 1 involved rotating a triangle around its centroid. This was achieved by setting a float uniform value (uRotation) to represent its’ current rotation and a vec2 uniform value (uPivot) to represent the local position of the centroid in the triangle. Vertices are first translated to the pivot, then rotated around the origin (using a matrix built in the shader), then translated back to their position (uPivot \* -1).

The program also needed to keep rendering frames rather than just rendering once at the start. To achieve this, I used window.requestAnimationFrame, which calls a function to update an animation (render in this case) at an appropriate time and passes it the time the application has been running for. This prevents rendering when the page can’t be seen, and essentially lets the browser control when frames should be rendered. Consistent movement is achieved by subtracting the previous time from the current time to create a deltaTime, and adjusting the rotation based on that delta time (fRotation += deltaTime \* speed;).



The achievement task was completed by adding an additional vec2 uniform (uPosition), setting it before 2 separate draw calls in the render function, and offsetting vertices with it after the rotation.

