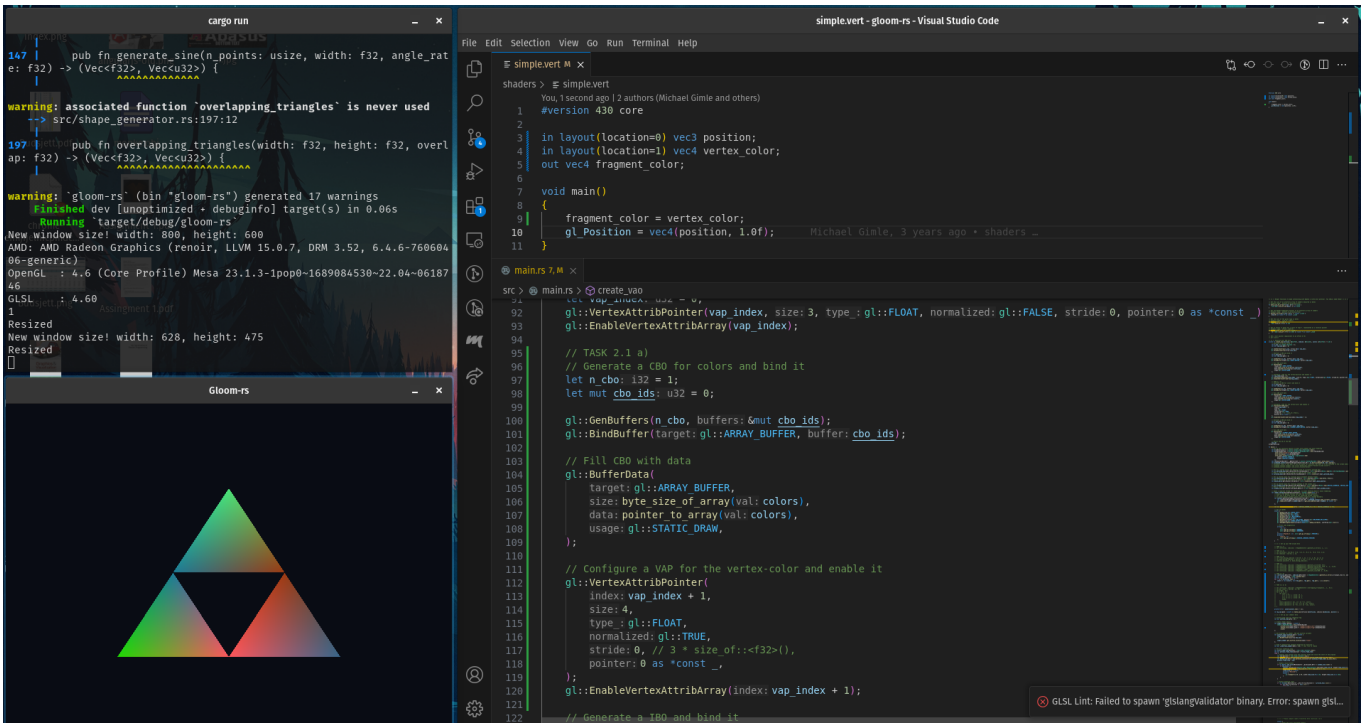
Assignment 2

Peter Johan Flått-Bjørnstad (Individual, Group 32)

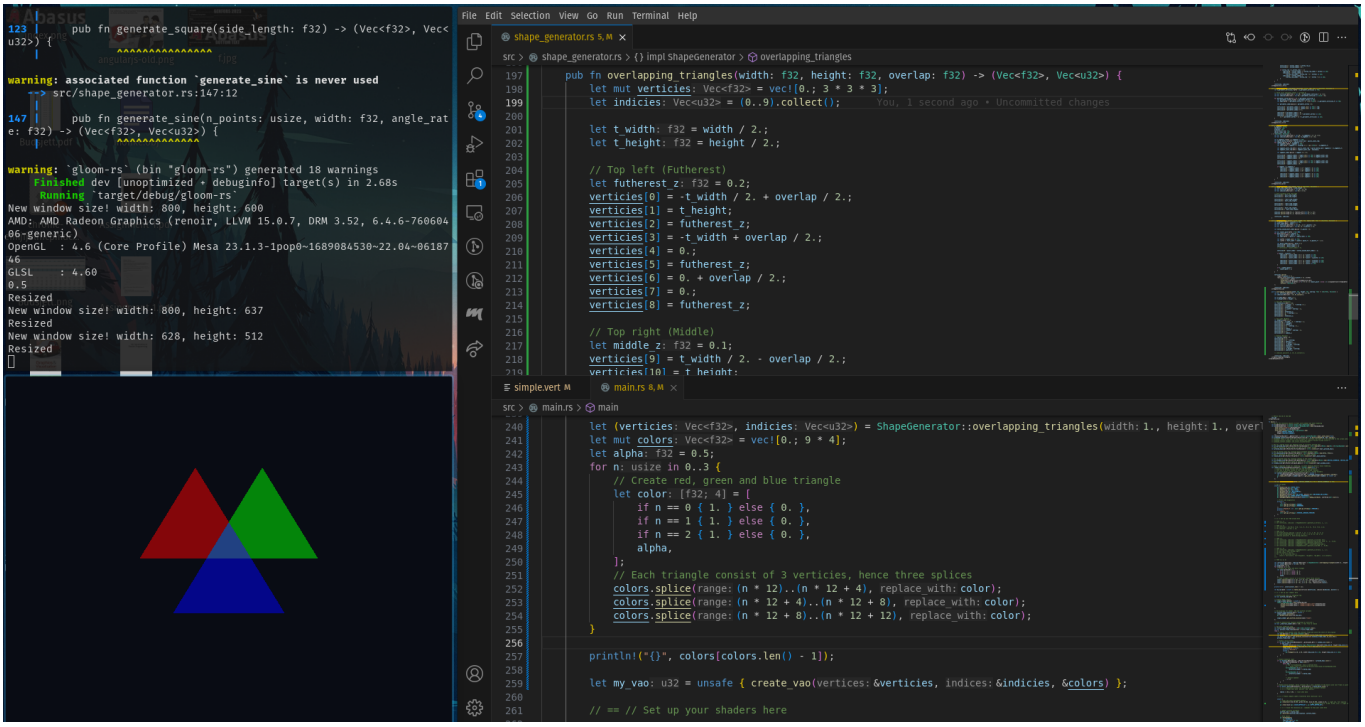
Task 1

b)



Task 2

a)

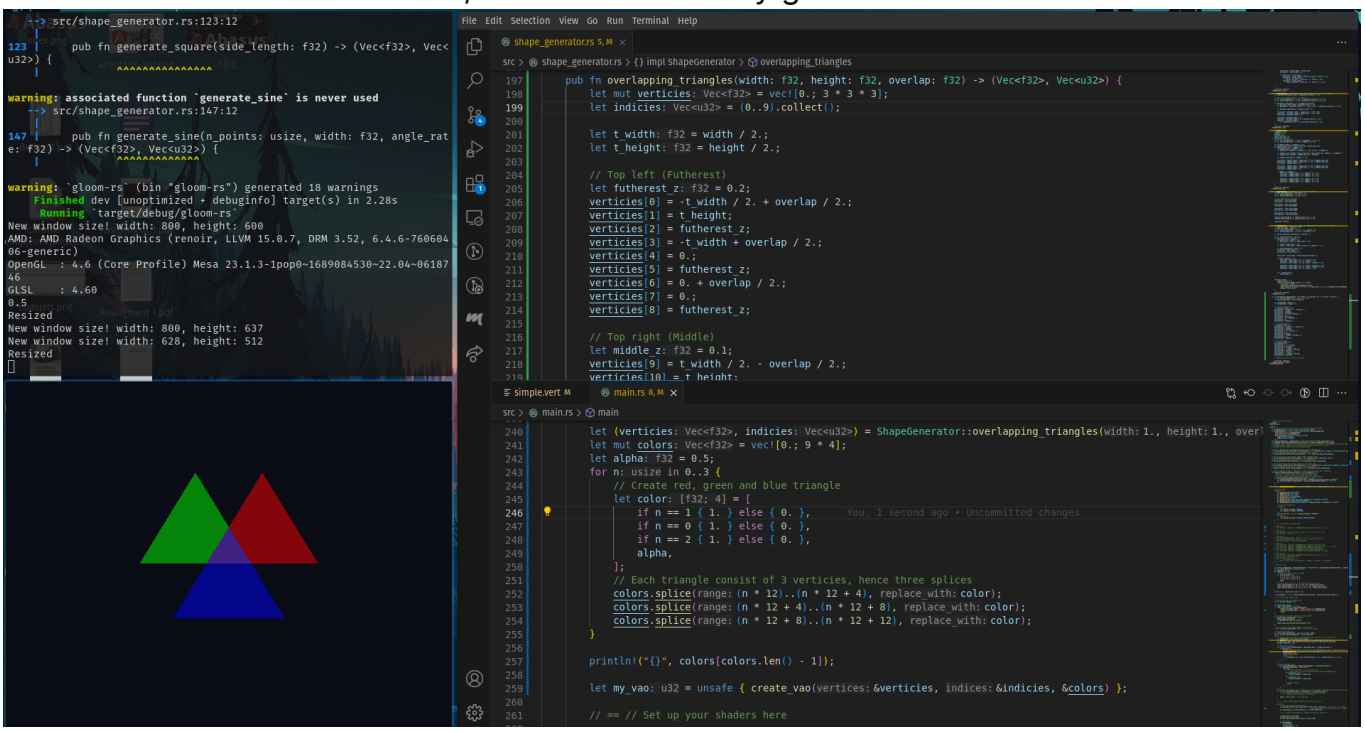


b)

i)

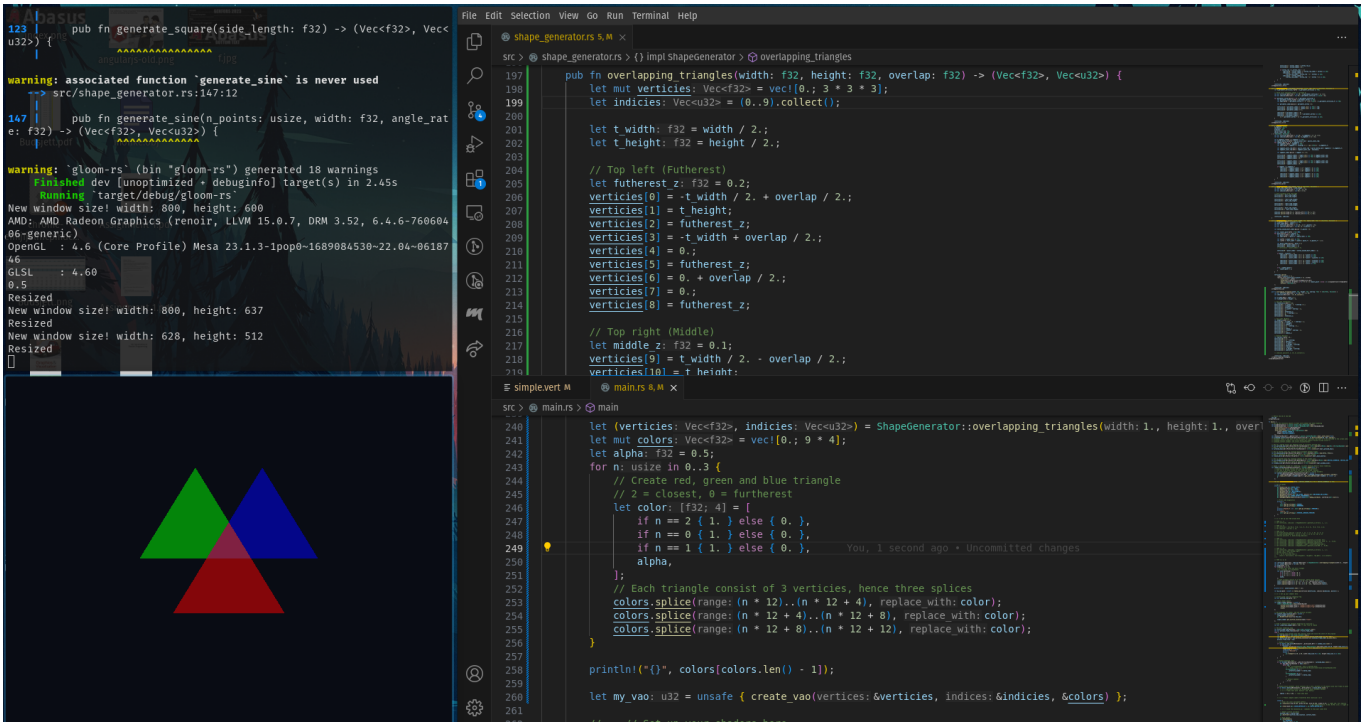
Picture in Task 2 a has blue closest, green in the middle, and red far behind.

Picture below has blue closest, then red and finally green.



Notice how it has a more red-ish hue than the picture above. We can make it even more

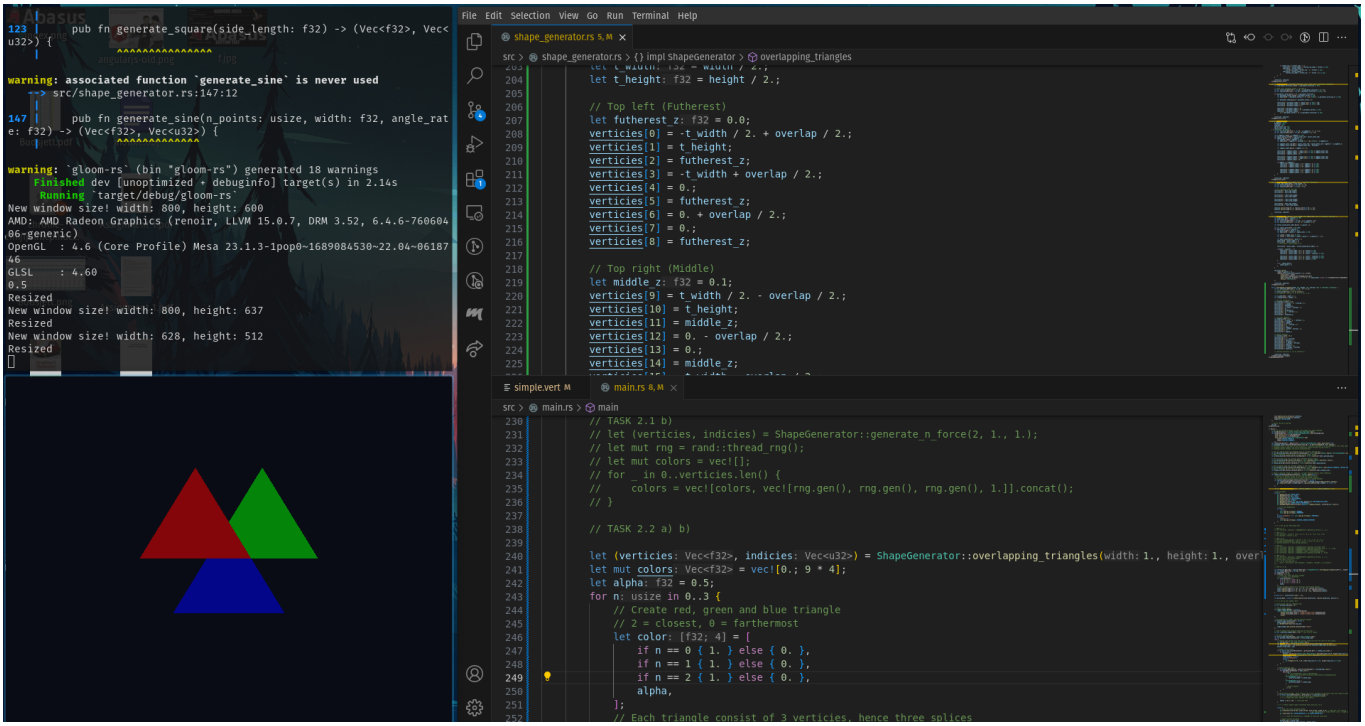
red by moving it to the front:



Note that the computer has to decide some order to apply the colors to each pixel. The way this is solved is by applying the color farthermost from the camera, then the middle, and finally closest. Since alpha is 50%, the closest color will contribute 50% to the final color, while the middle will contribute 50% to the colors behind it and so on, making the closest color most prominent.

ii)

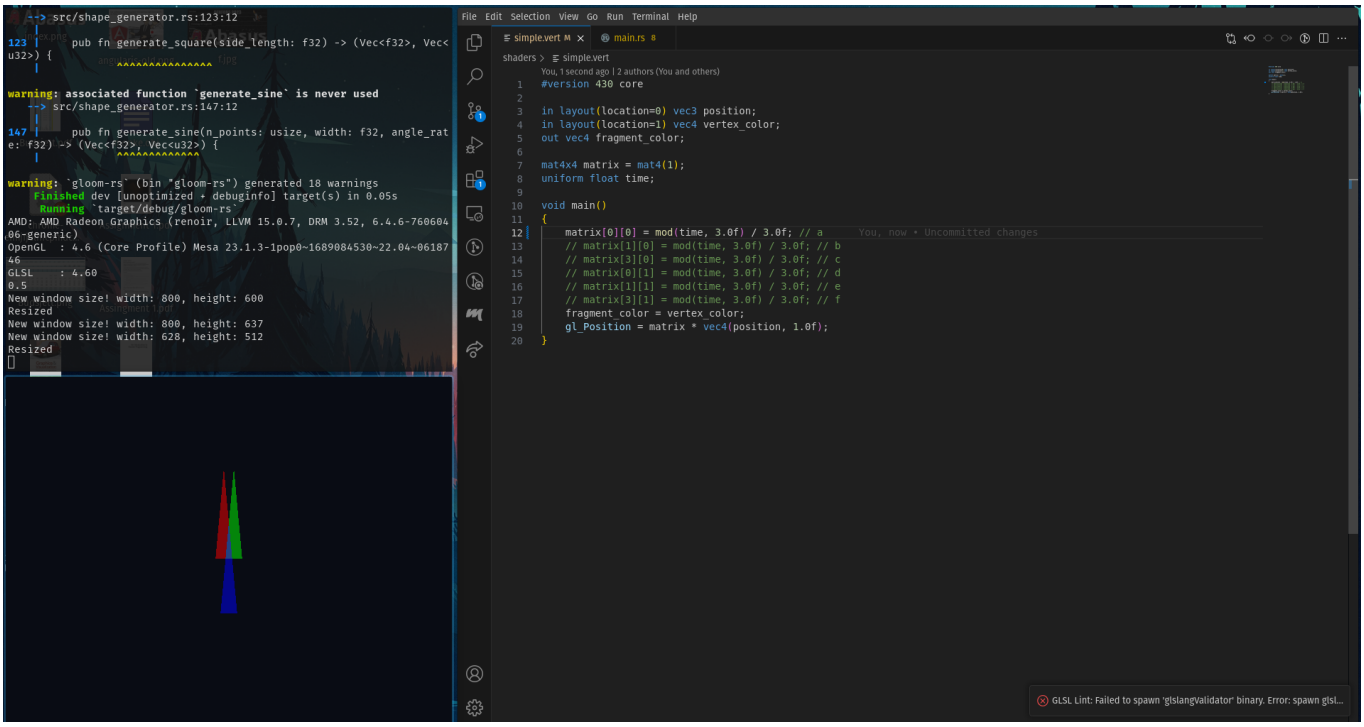
Picture below is back to same situation as Task 2 a, but red and blue have swapped places in the z-buffer:



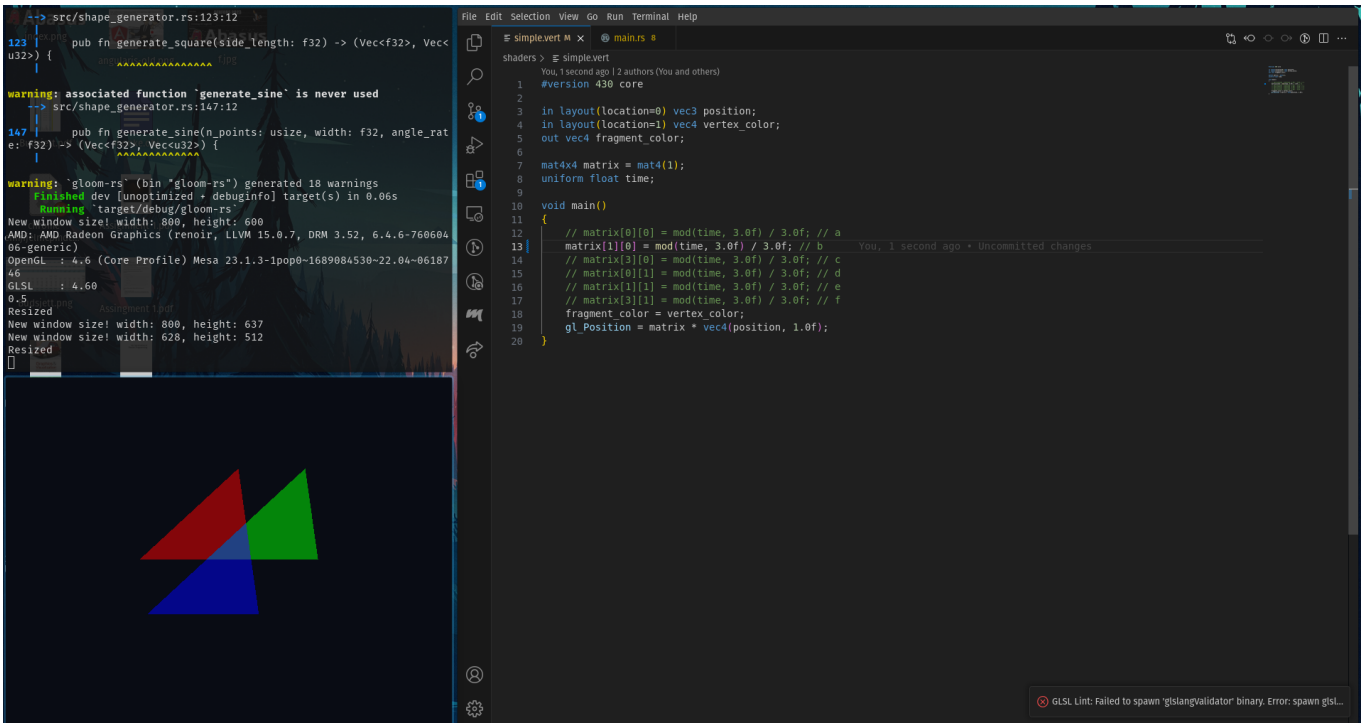
Where did the other colors go?
Here it's worth noting that the order of indicies has not been updated. This means the top-left triangle is still the first entry, which implicitly makes it the first triangle drawn. The computer will then use depth culling to save computing the trangles below, creating no transparency-effect :(

Task 3

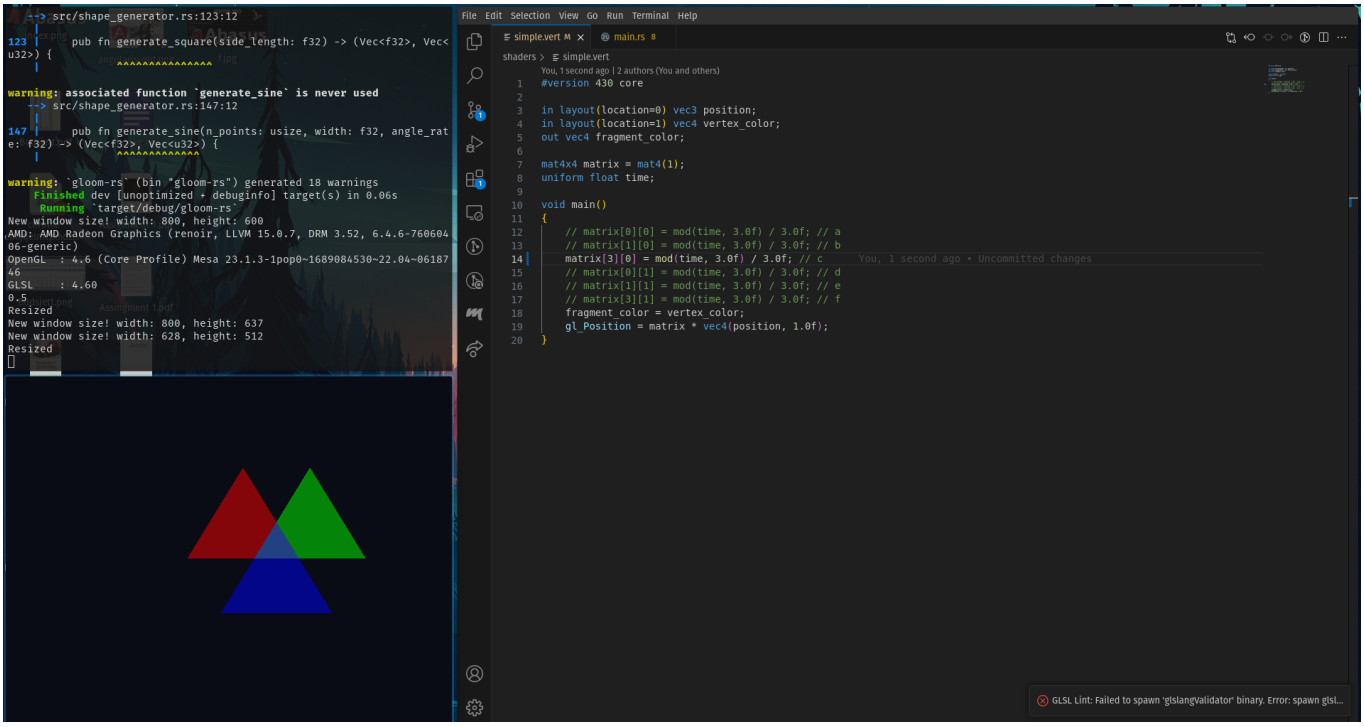
b)
Image below shows how i went forward displaying each modification.



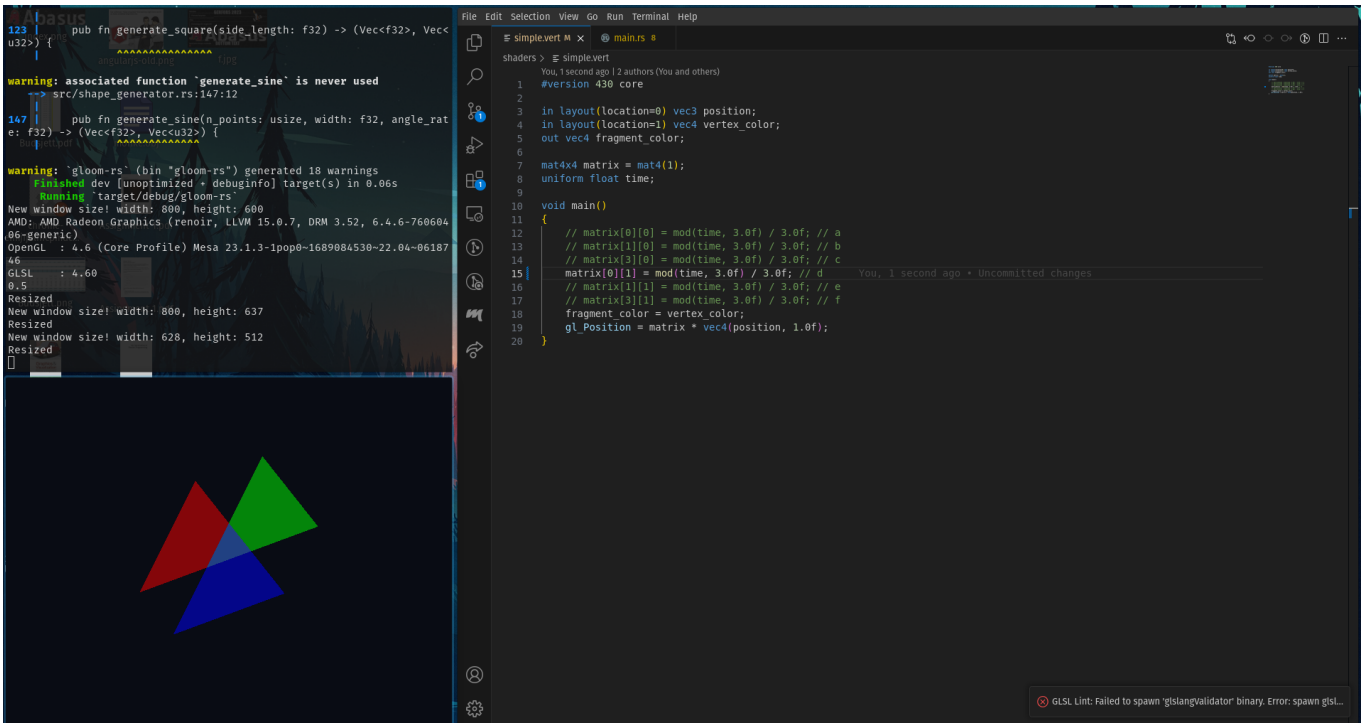
- a: Scaling along x-axis



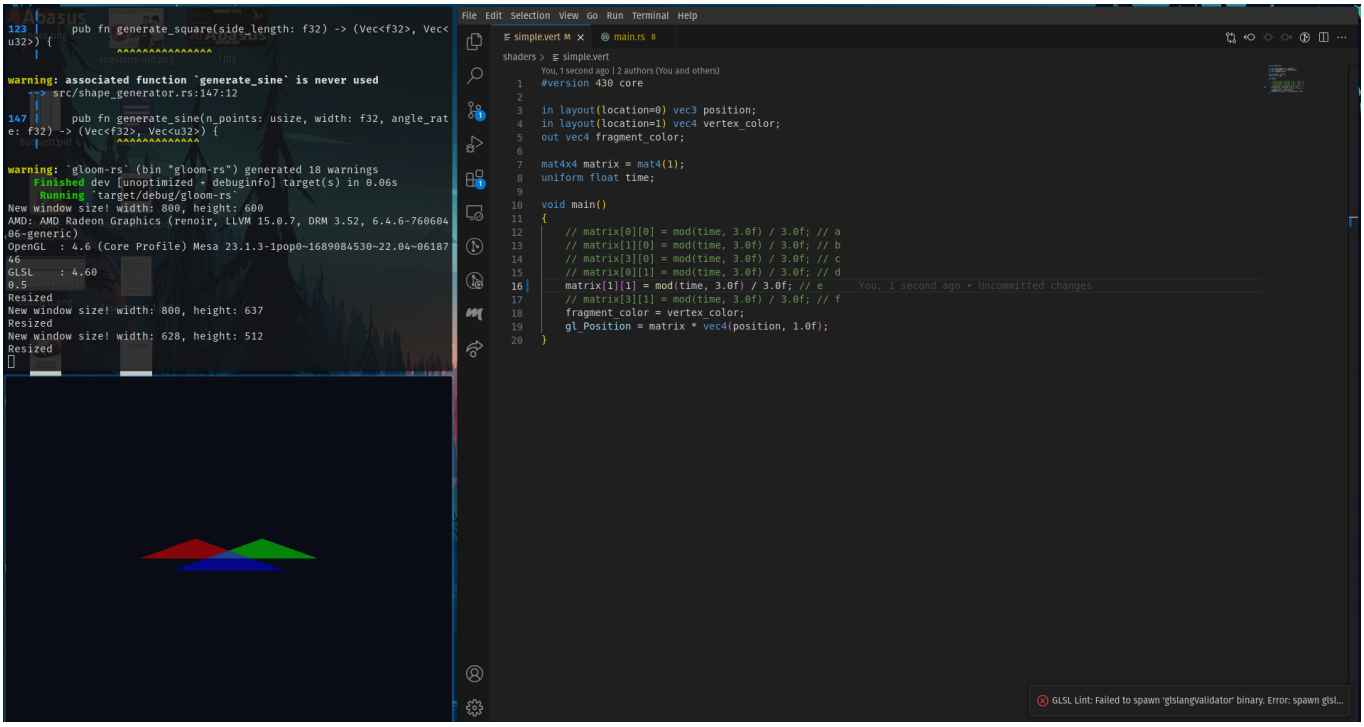
- b: Shearing x with respect to y-axis



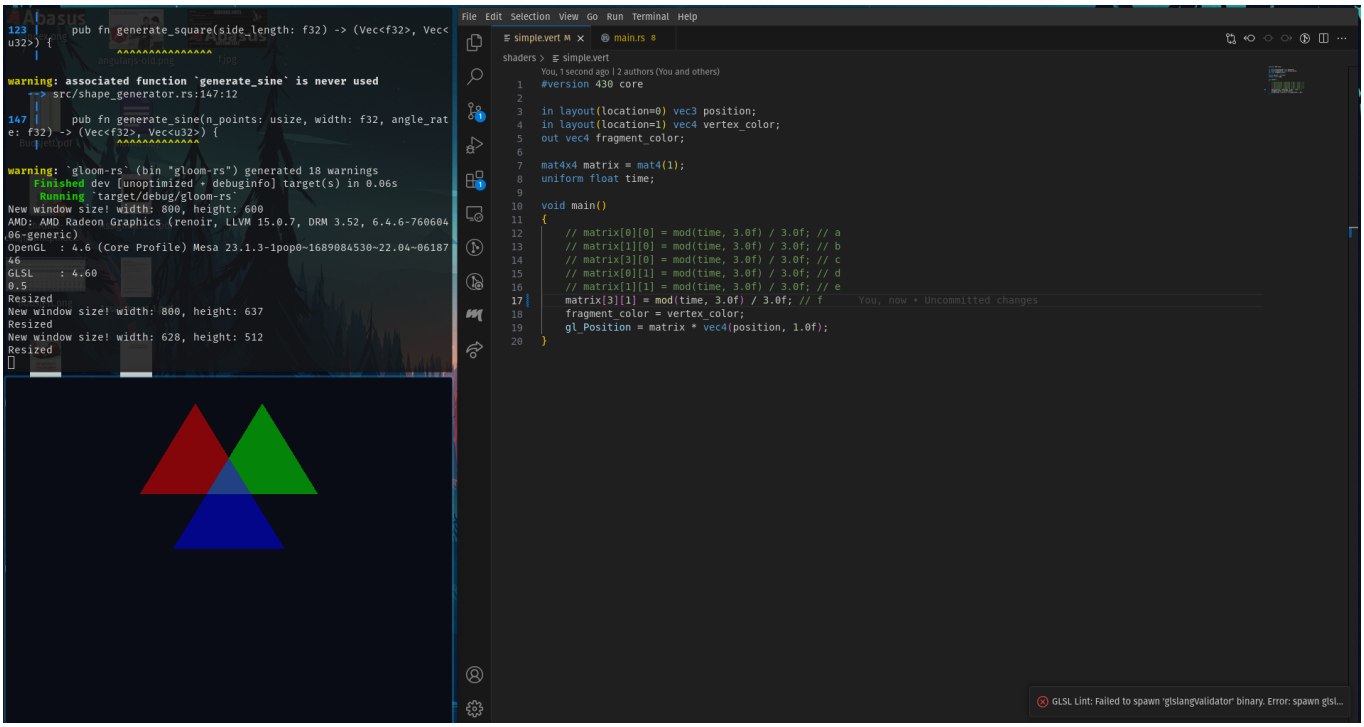
- c: Translation along x-axis



- d: Shearing y with respect to x-axis



- e: Scaling along y-axis



- f: Translation along y-axis

c)
A rotation requires modification of both x and y, but since we always changed a single value of the identity matrix, it would only modify a single coordinate axis, making a rotation impossible.

Task 4

c)
(Fancy) Camera and keyboard movement implemented :)

- Uses the recommended keys for movement

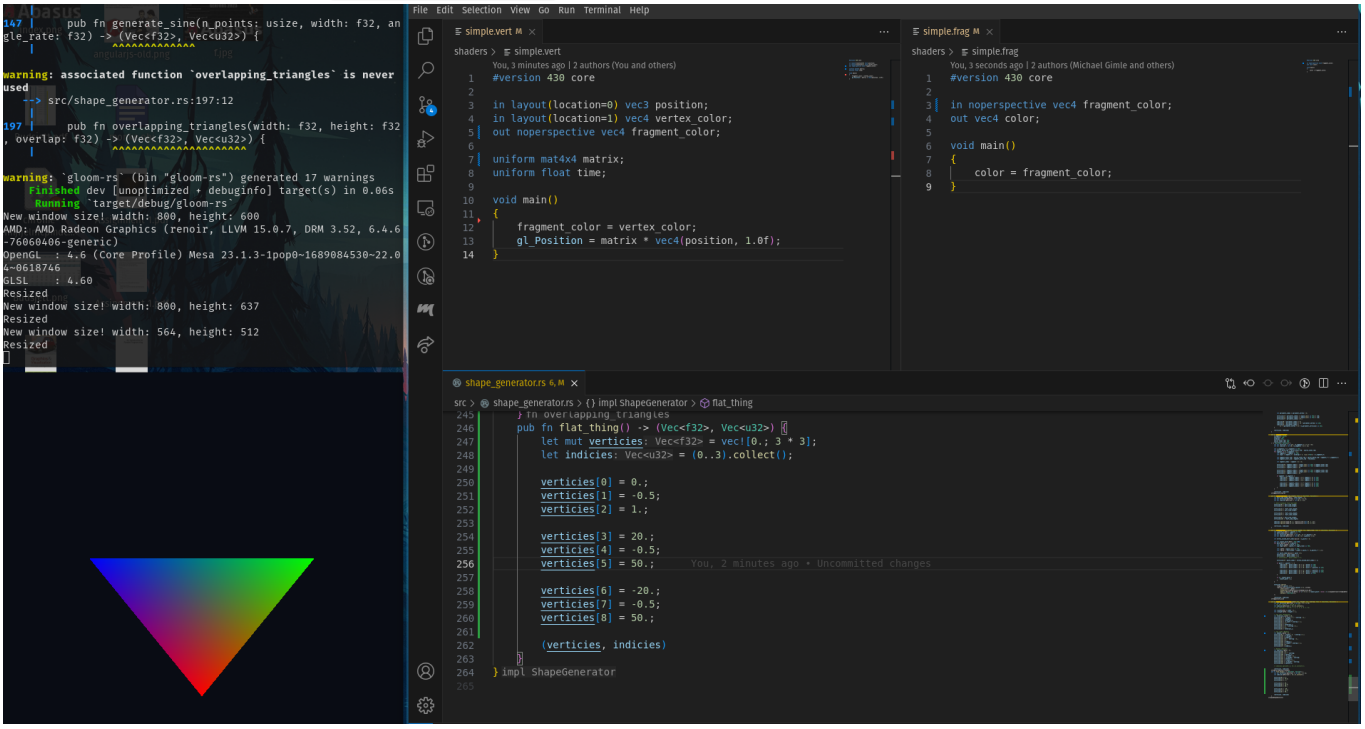
Task 5

a) Implemented ;)

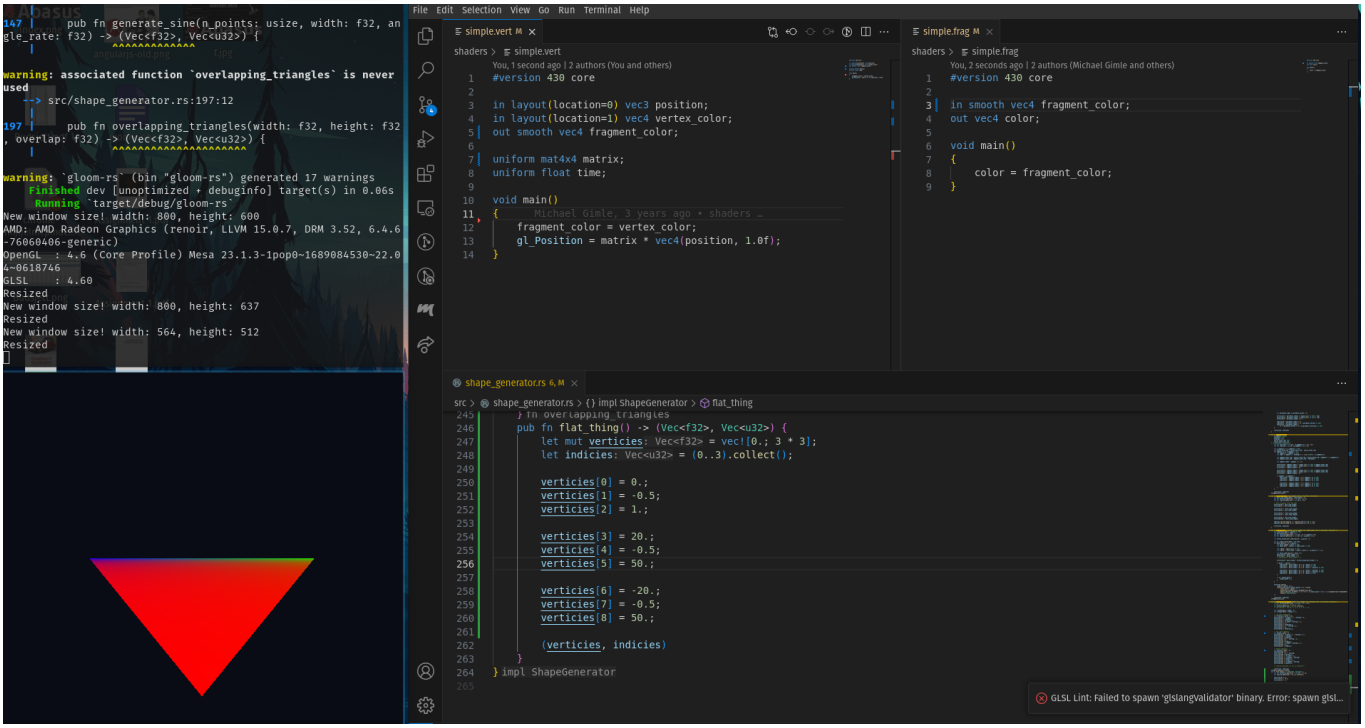
b)

Created a triangle where the blue and green edge is faaaaaar away from the camera:

Without perspective: `noperspective`



With perspective: `smooth`



Notice the dominating red color. when `noperspective` is used, the color between the edges are interpolated without considering how close the vertex is to the camera. This creates the illusion that the triangle is facing the camera. The picture below uses `smooth` which takes this into consideration, creating the illusion that the large triangle is facing upwards, and therefore a more correct color interpolation.