Computer Graphics - Assignment 2

Christopher Braathen & Michael Brusegard

August 2024

1 Per-Vertex Colors

b) What OpenGL does in between the vertices for each fragment is called interpolation. In the context of graphics rendering, interpolation is the process of estimating values between two points. In this case, OpenGL will interpolate the colors between the vertices of the triangles, which will result in a gradient effect, where the color smoothly transitions from one vertex color to the next across the face of the triangle. You can see this gradient effect on the triangles below.

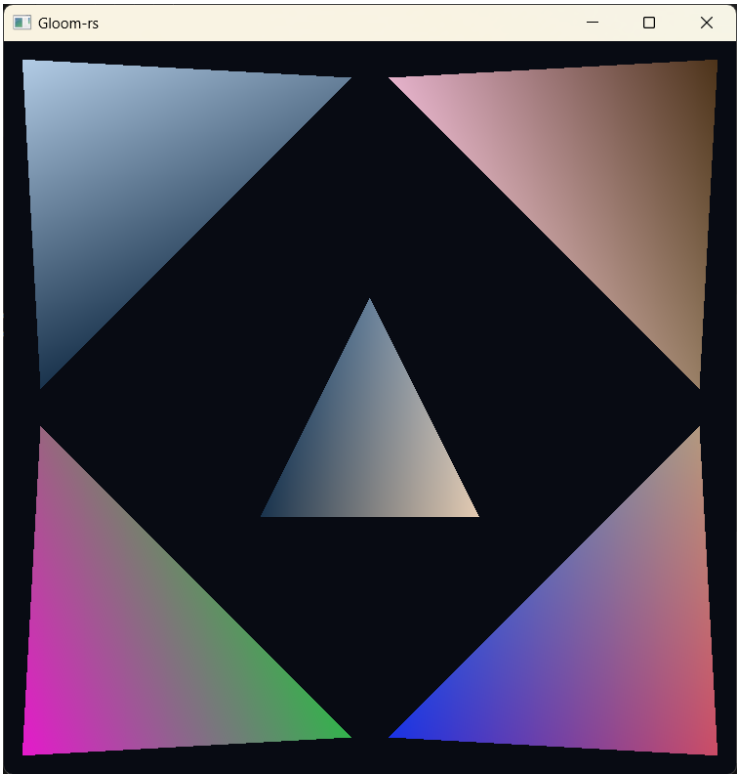


Figure 1: Each vertex of each triangle has a different color.

2 Alpha Blending and Depth

a) Triangles with a blended color when multiple triangles overlap from the camera's perspective.

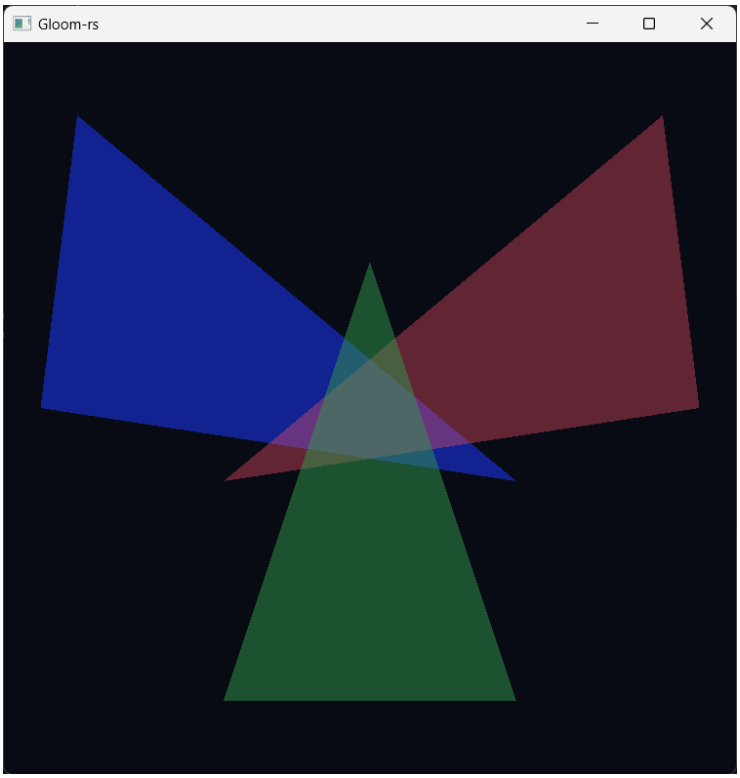


Figure 2: Transparent overlapping triangles.

i) When swapping the colors of the overlapping transparent triangles, the final blended color in the overlapping region changes because blending is order-dependent. Triangles are drawn from back to front, and each triangle's color is combined with the previously drawn ones based on their transparency (alpha).

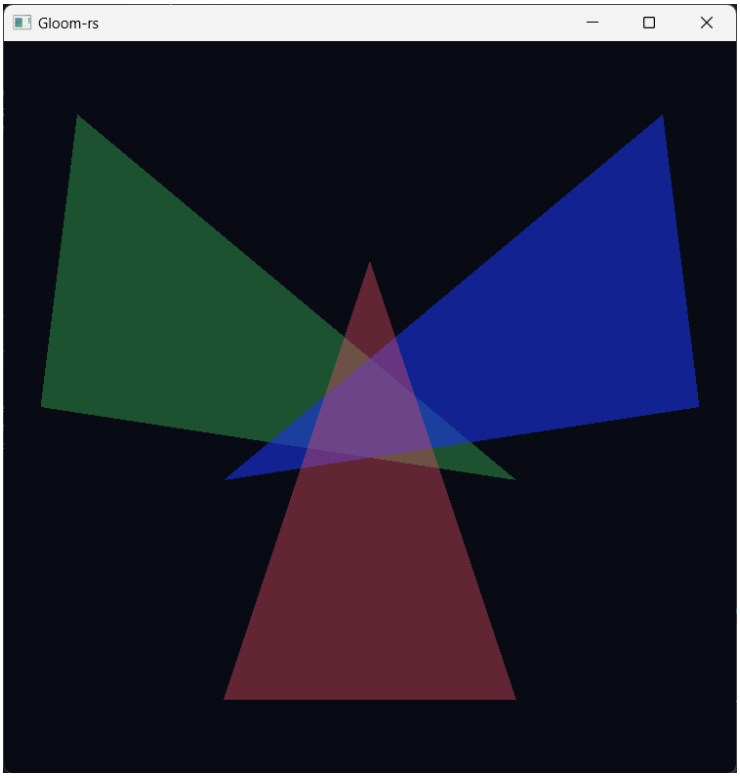
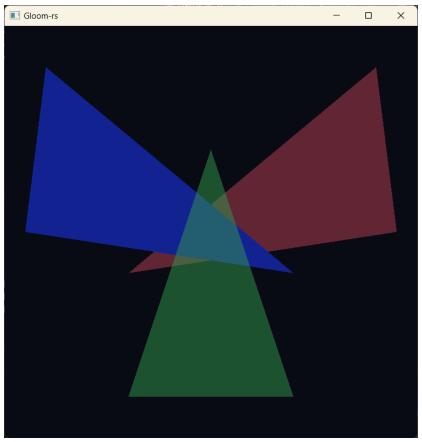
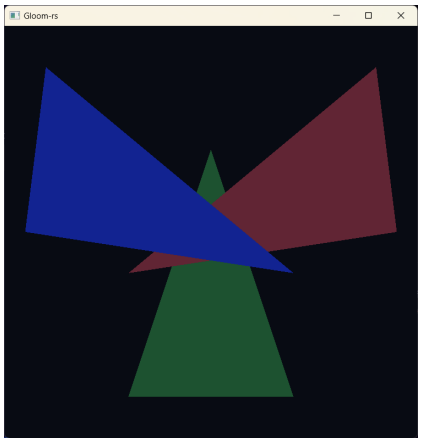


Figure 3: Transparent overlapping triangles with swapped colors.

ii) When we change the z-coordinates of the triangles, the depth order changes, affecting the blending. We observed that if the triangle is buffered first and is closest to the camera it will not blend, and if the closest triangle is buffered second it will blend with the ones behind.



Green and blue triangle blend, but blue blocks reds.



None of the triangles blend.

Figure 4: Transparent overlapping triangles with swapped depth coordinates.

3 The Affine Transformation Matrix

b) When modifying each of the values in the identity matrix we make the following observations for the transformation type and axis:

- a: Scaling along the x-axis.
- b: Shearing along the x-axis (affecting y).
- c: Translation along the x-axis.
- d: Shearing along the y-axis (affecting x).
- e: Scaling along the y-axis.
- f: Translation along the y-axis.

Note that in GLSL the matrices are column-major, which we interpreted as when defining a row it is actually being defined as a column, thus we have to transpose the matrix such that it adheres to the correct properties.

c) None of the transformations observed were rotations because the changes only affected the scale, shear, or position of the shapes along the x and y axes, without altering their orientation relative to those axes. The shapes maintained their original angles throughout the transformations, which is a clear indication that no rotational effects were present. Additionally, the transformation matrix utilized did not include non-zero values in the off-diagonal elements that correspond to rotation, further confirming that the transformations were strictly limited to scaling, shearing, or translating rather than involving any rotation.

4 Combinations of Transformations

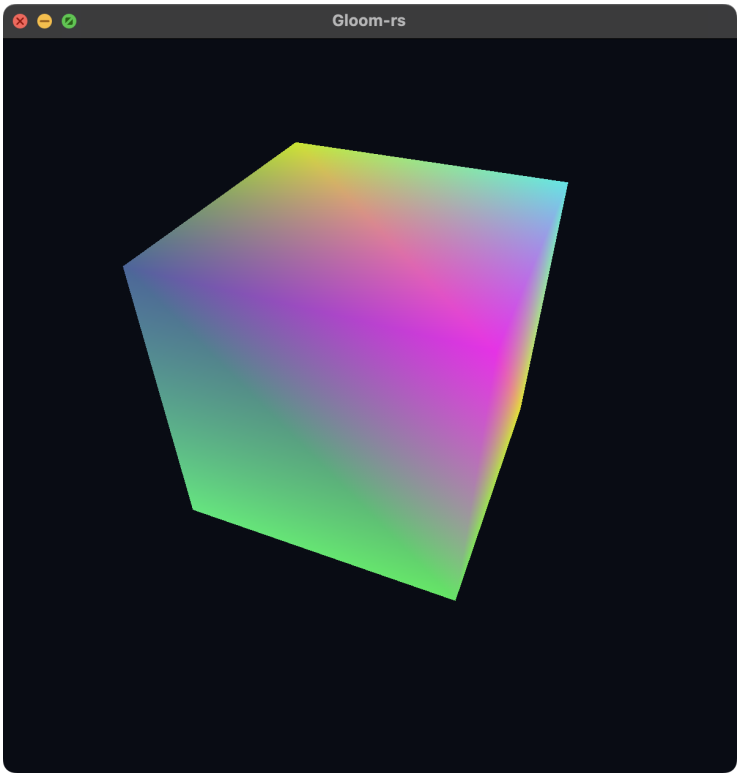


Figure 5: Cube that can be rotated around with the camera.

c) *b)* *vi)*

Key	Movement
W	Moves the camera forward along the Z-axis
A	Increases the yaw, rotating the camera to the right
S	Moves the camera backward along the Z-axis
D	Decreases the yaw, rotating the camera to the left
Space	Moves the camera downward along the Y-axis
LShift	Moves the camera upward along the Y-axis
Q	Increases the pitch, rotating the camera upwards
E	Decreases the pitch, rotating the camera downwards