

CS 351 Project 7

CS 351 - Assignment 7: Z-buffer Rendering

Elena Kirillova and Titobiloluwa Awe

Abstract

In this project, we implemented z-buffering in our rendering engine in order to do hidden surface removal. Z-buffering is managing depth in a 3-dimensional image.

Task Description

Our first task was to implement our line draw function with z-buffering. Our next task was to implement z-buffering inside of our polygon scanline fill function.

Solution Description

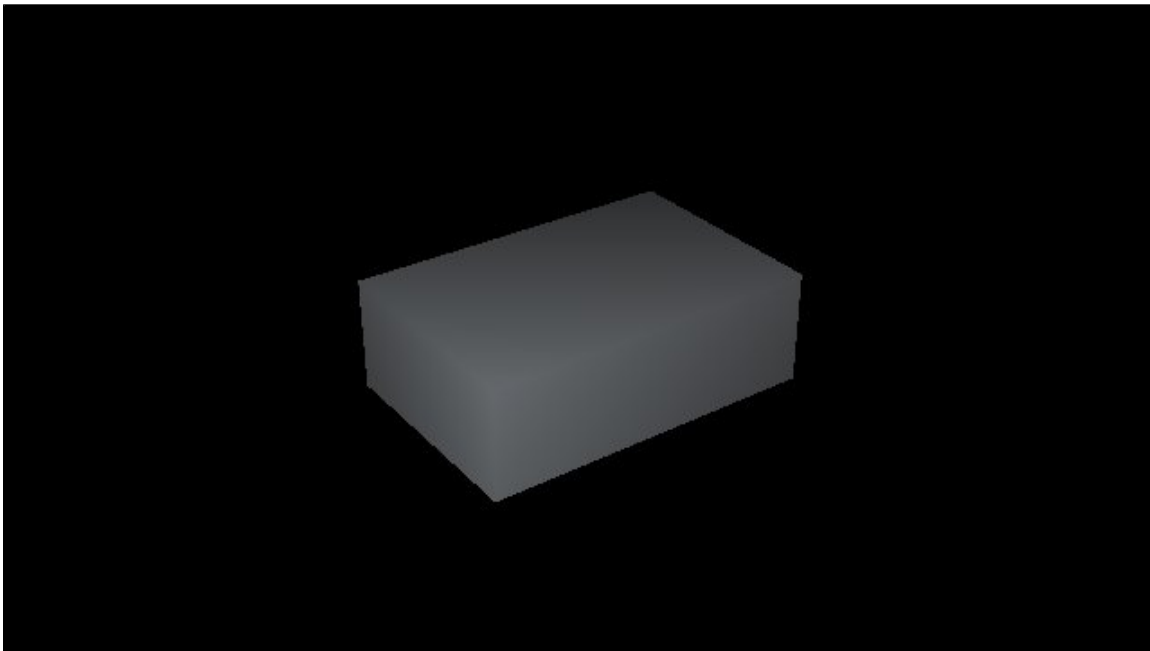
In order to implement our line draw with z-buffering, we first had to check to see if the z-buffer flag was set. If it was set, then calculated our initial z and delta z and if our z was greater than the image's z value, then we drew at that point and set the image's z value. We used $1/z$ instead of z because this translates linearly along our image.

For the scanline fill function, we had to do a similar calculation inside our innermost loop and set the color appropriately.

Results

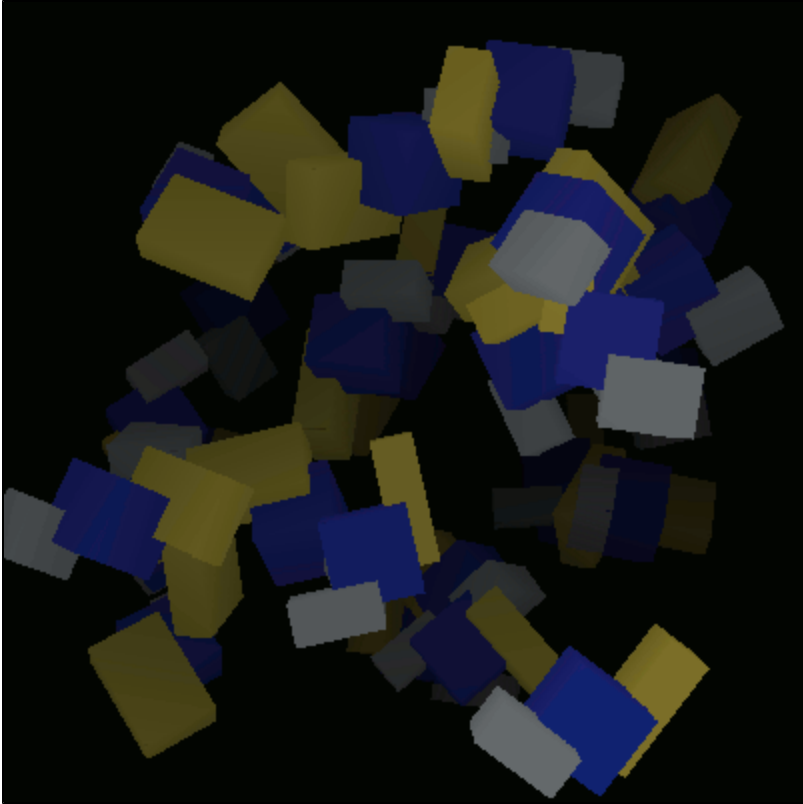
Single Cube Shaded

Required Image 1: A cube with the depth inverted



Z-buffer Test

Required Image 2: A grouping of cubes rotating showing depth



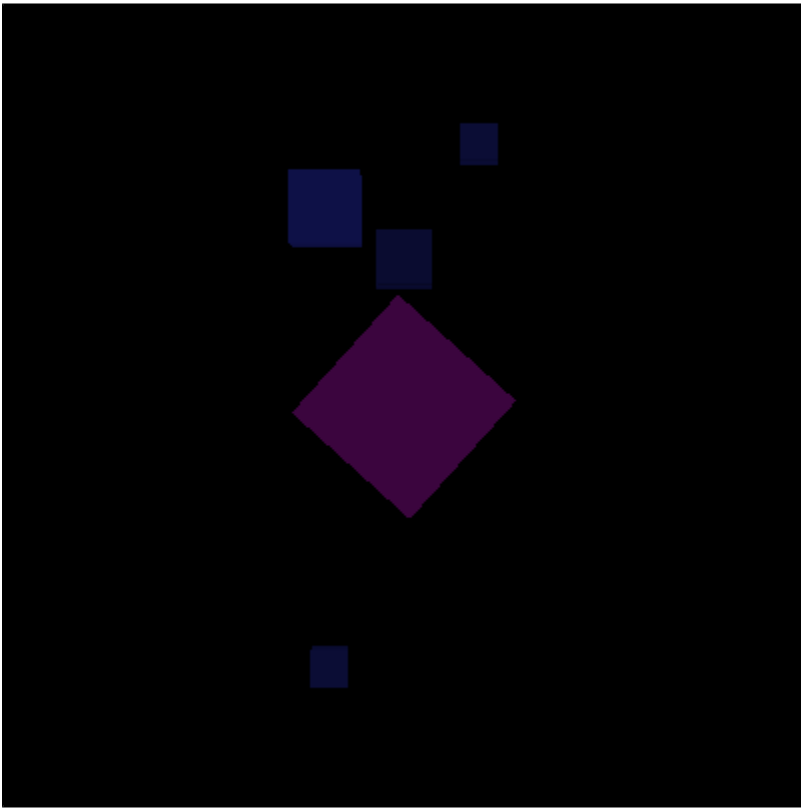
3D images demonstrating our Z-buffer

Blue Crosses getting closer to the screen.

In this last picture, you can see that as we get closer, the crosses get brighter and this shows a change in depth.



The image below is a formation of one big cube rotating around y-axis, while small cubes orbit it rotating around various axis.



Summary

In summary, this was a very interesting project because we got to implement z-buffering which is a very useful drawing tool and we will definitely be using this in our rendering system again. Overall, a very successful project.