Mike Petrus
CS 180

Assignment 3

1. *Transform the triangles to NDC space and then screen space*

First we get our transform our vertices the same way as assignment 2. Create view transformation matrix to get them in camera space, then create the perspective transformation matrix and apply it to get the vertices in the canonical cube. Finally divide our homogenous coordinates by **w** to ensure we have the correct NDC coordinates.

Next we use the viewport transformation matrix: {{(image_width/2.0f), 0, 0, 0}, {0, (image_height/2.0f), 0, 0}, {0, 0, 1, 0}, {(image_width/2.0f), (image_height/2.0f), 0, 1}}

to get the vertices in screen space.

Then we can take our coordinates for each triangle to create the bounding box. For the **xmin** we find the smallest x-value of the 3 coordinates, and then take **min**(0, xmin). For **xmax** we find the largest x-value and take **max**(image_width, xmax). Then repeat this process for **ymin** and **ymax**, but use the image_height. Now we know our max and min values don't fall outside the screen.

2. *Implement inside_triangle*
For inside_triangle we can find the barycentric values of alpha, beta, and gamma and apply them to the test point.

If all 3 values are non-negative then return true and draw the pixel.

3. *Implement Z-Buffer*

For each triangle we repeat the rasterization process. If we find that the point inside_triangle is true, then we interpolate the Z-value and test it against the Z buffer. If it is less than the current value then add it to the frame-buffer (image) and draw the triangles corresponding color. The result is overlapping triangles of correct depth placement.