Project 4 - Scanline

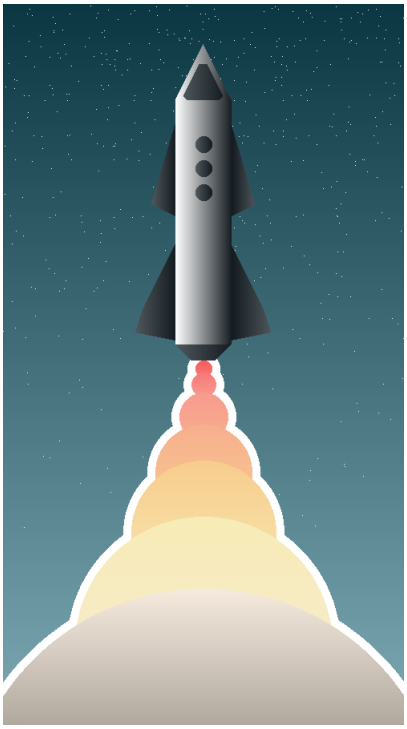
Created by Gregory Attra, last modified less than a minute ago

Greg Attra | Prof. Maxwell CS 5310

Overview

Project 4 was implementing the `Polygon` API which is responsible for allocating, initializing, updating and drawing polygons on an image. The `Polygon` is a list of `Point` structs, which are each connected with a `Line` , and the final point connects back to the first point, forming a shape. There were two algorithms which were to be implemented for filling a polygon: scanline fill and barycentric fill.

Required Images



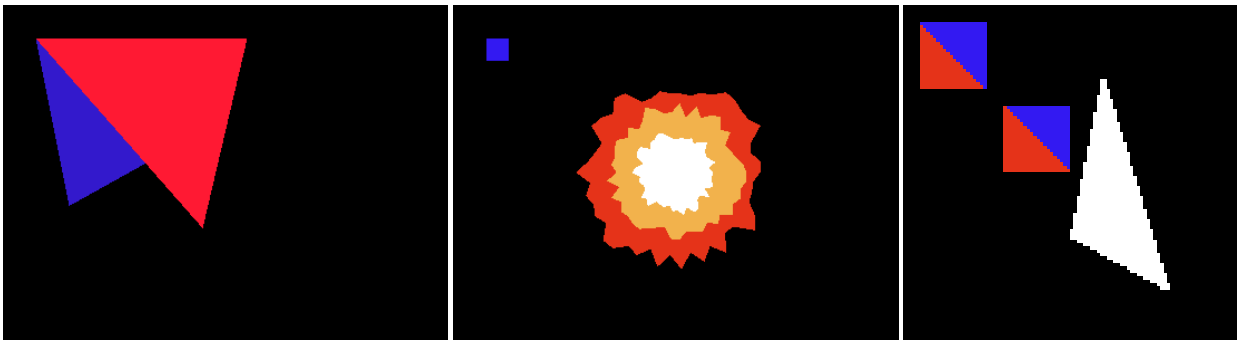
For my custom image, I used all the features currently supported by the graphics engine to draw a minimalist rendering of a spaceship taking off. The background is a gradient fill from skyblue to dark blue representing the lower and upper atmosphere respectively. To generate the stars, I created a simple algorithm which loops over the rows of the image and at each row randomly generates a float between 0 and 1. If that float is greater than a threshold, we draw some stars again using `rand()` and checking that the result is greater than the threshold. After each row, we increment the threshold, which results in more stars toward the top and fewer stars toward the bottom.

```
double thresh = 0.3;
for (int s   = 0; s < src->rows; s++)
{
    float random = (float) rand() / (float) RAND_MAX;
    while (random >= thresh)
    {
        float xrand = (float) rand() / (float) RAND_MAX;
        int x = (int) (xrand * (float) src->cols);
        image_setColor(src, s, x, white);

        random = (float) rand() / (float) RAND_MAX;
    }
    thresh += 0.001;
}
```

The exhaust fumes are circles with a subtle gradient fill (most noticeable on the bottom-most circle).

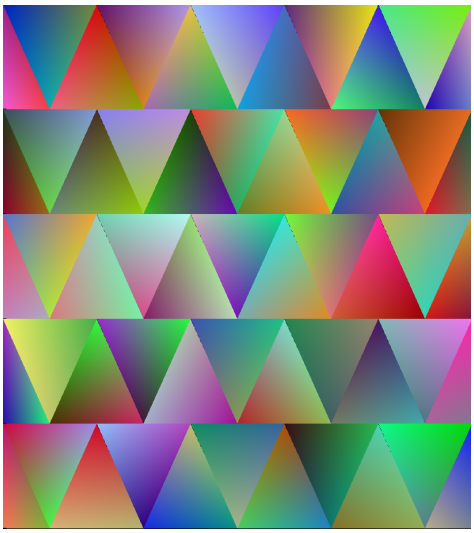
The spaceship itself is comprised almost entirely of polygons, each filled with a gradient to give it some shape.



Above are the required images from running `test4a.c` , `test4b.c` , and `test4c.c` .

Custom Images

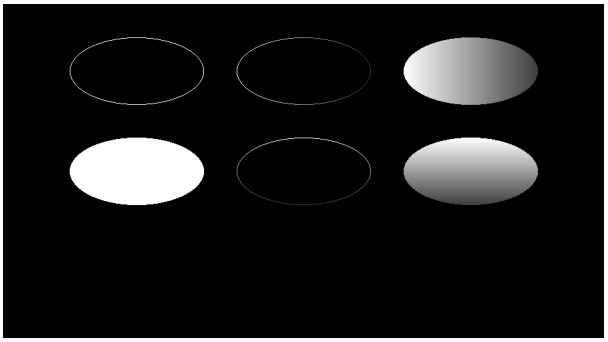
Bary Blend



For one of my portfolio images I created a tapestry of blended triangles using barycentric fill and the barycentric coordinates to interpolate between the colors.

Extensions

Gradient Fill



As mentioned above, the polygons and circles can be filled with a gradient between two colors (both in the horizontal and vertical directions). I also implemented gradient fill for `Elipse` objects, again both vertically and horizontally. Both the outline and fill algorithms support gradients.

Acknowledgements

- Prof. Maxwell's Skeleton Code for Scanline Fill (scanline.c, list.h, list.c)
- Prof. Maxwell's Lecture Notes

No labels