Beginner

OpenGL ES & GLKit

Hands-On Challenges

Beginner OpenGL ES & GLKit Hands-On Challenges

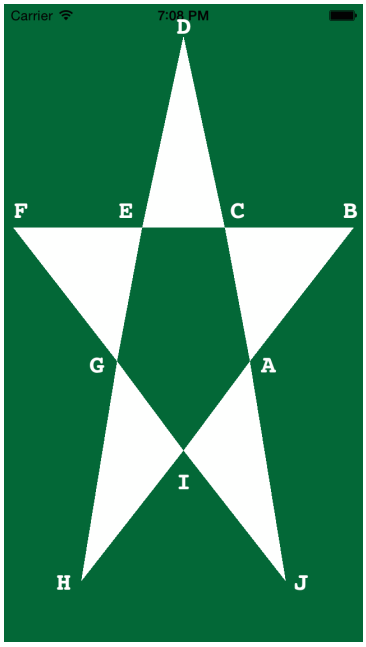
Challenge #2: Star White, Star Bright

In this short challenge, you will get practice working with OpenGL vertices by creating a star with triangles.

Part 1: Half Star

In the resources for this challenge, you will find a starter project. This is a project that is equivalent to where things left off in the lecture, with a triangle drawn to the screen. Look through the project and make sure you have a good understanding of how it works.

Your goal is to create the following star outline:



You need to create five different triangles, with the vertices shown above (A-J). To do this, open **RWTViewController.m** and inside `setupVertexBuffer`, replace the declaration of `vertices` with the following:

```
const static RWTVertex vertices[] = {
  {{+0.37, -0.12, 0}}, // A
  {{+0.95, +0.30, 0}}, // B
  {{+0.23, +0.30, 0}}, // C
```

```
    {{+0.23, +0.30, 0}}, // C
    {{+0.00, +0.90, 0}}, // D
    {{-0.23, +0.30, 0}}, // E

    {{-0.23, +0.30, 0}}, // E
    {{-0.95, +0.30, 0}}, // F
    {{-0.37, -0.12, 0}}, // G

    {{-0.37, -0.12, 0}}, // G
    {{-0.57, -0.81, 0}}, // H
    {{+0.00, -0.40, 0}}, // I

    {{+0.00, -0.40, 0}}, // I
    {{+0.57, -0.81, 0}}, // J
    {{+0.37, -0.12, 0}}, // A
};
```

Try to match up each of the vertices with the diagram above to make sure you understand how each triangle works.

Finally, scroll down to `glkView:drawInRect:` and replace the `glDrawArrays` line with the following:

```
glDrawArrays(GL_TRIANGLES, 0, 15);
```

This tells OpenGL that there are 15 vertices in your array (5 triangles x 3 vertices each).

Build and run, and you now have the outline of a star!

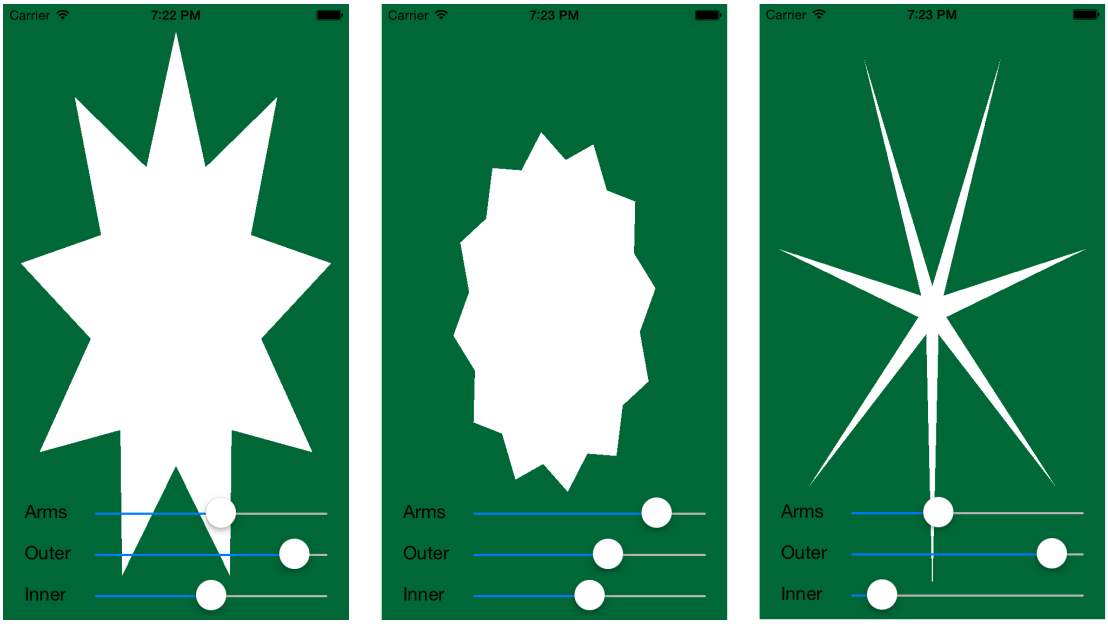
Part 2: Full Star

In the next part of the challenge, your goal is to fill in the "hole" in the middle of the star:

To do this, you will need to create 5 triangles that each have the center (0, 0) as one of the coordinates. See if you can figure this out on your own!

Uber Haxx0r Challenge: Uber Star

If you're feeling particularly awesome today, why not see if you can make a dynamic star generator?



Your app should have three sliders that let you tweak the number of "arms" of the star, the outer star radius, and the inner star radius.

Here are a few tips on how to accomplish this:

- Start by figuring out how you could get X points along a circle. Hint: you'll need cos and sin.
- It may be helpful to make a loop that alternates through the circles by a delta angle, alternating between the inner and outer circles to get the points in the sequence. Once you have three vertices, that's a triangle, but you'll need to repeat the final vertex to begin the next triangle.
- Like you did in the earlier examples, it's easiest to get the "outer" part working first, then do the inner part.
- The easiest way to update the vertex data is just to re-send the data with `glBufferData` when it needs to be refreshed.