

XNA Tutorials

Utah State University
Association for Computing Machinery
XNA Special Interest Group
RB Whitaker
16 October 2007

A Little Practice

Tutorial 8

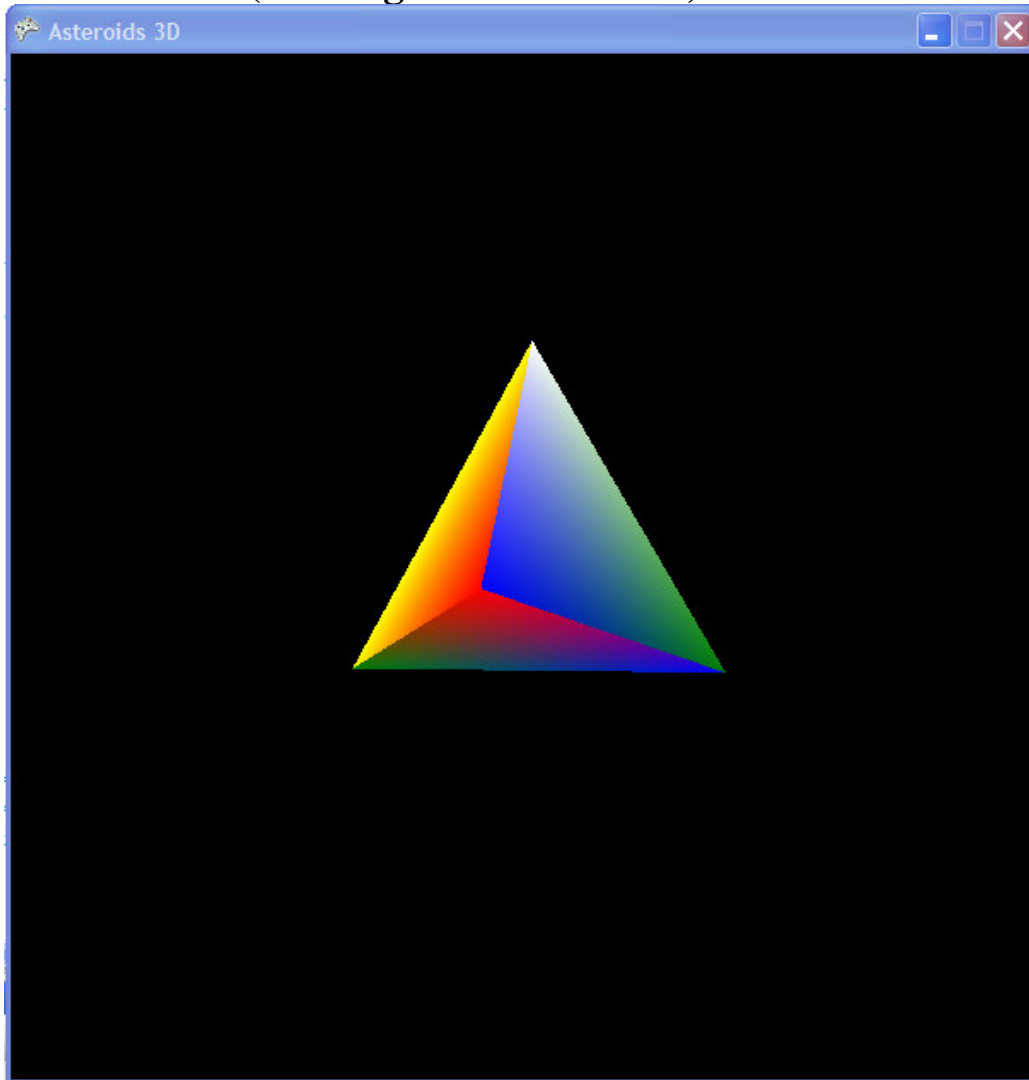
Overview

We now have a better grasp of the world we are working in, and how to manipulate it. In the upcoming tutorials, we will move on to more fancy things, including index buffers, texturing, lighting, and then loading full-fledged models from a file.

For now, though, it would be a good idea to do a little reviewing and practicing of the things we have learned, and also try our hand at a few objects with more than one triangle.

Each of the sections below describes the x -, y -, and z -coordinates of three simple objects. Make the needed changes to your code to store these coordinates in your `vertexArray` variable and draw the models. Don't forget to change the triangle count parameter of the `DrawUserPrimitives()` function to the right number of triangles. Also, feel free to color your polygons in any way that you want. I have provided my coloring code at the end of this tutorial if you want to use that. Also, make sure you set the size of your `vertexArray` object to be $3 * \text{the number of triangles in your object}$, so there is enough vertices for everything.

Tetrahedron (4 triangles – 12 vertices)



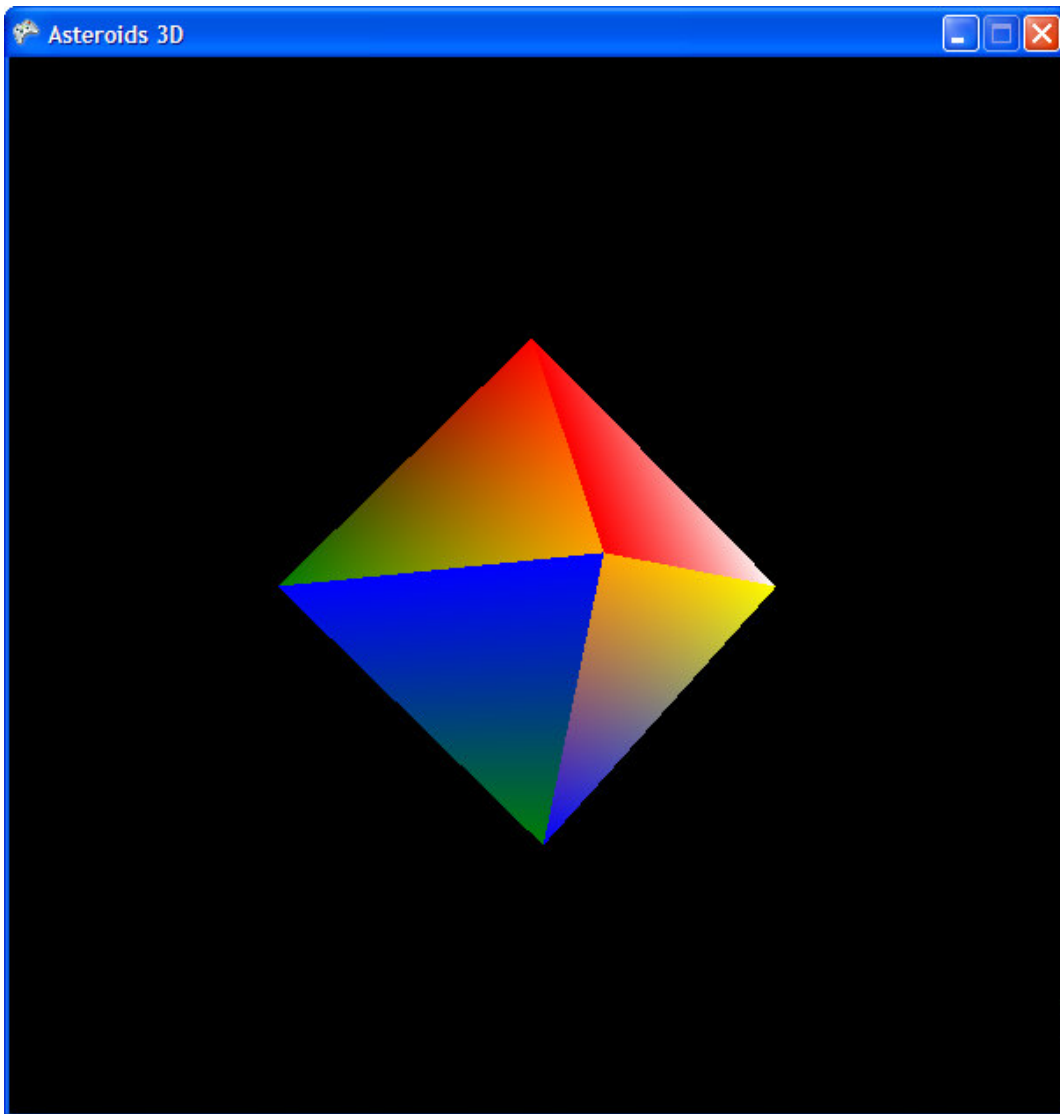
```
0.000f, 1.000f, 0.000f  
-0.816f, -0.333f, -0.471f  
0.000f, -0.333f, 0.943f
```

```
0.000f, 1.000f, 0.000f  
0.816f, -0.333f, -0.471f  
-0.816f, -0.333f, -0.471f
```

```
0.000f, -0.333f, 0.943f  
0.816f, -0.333f, -0.471f  
0.000f, 1.000f, 0.000f
```

```
-0.816f, -0.333f, -0.471f  
0.816f, -0.333f, -0.471f  
0.000f, -0.333f, 0.943f
```

Octahedron (8 triangles – 24 vertices)



```
1, 0, 0  
0, 0, 1  
0, -1, 0
```

```
1, 0, 0  
0, 0, -1  
0, 1, 0
```

```
-1, 0, 0  
0, 0, 1  
0, 1, 0
```

```
-1, 0, 0  
0, 0, -1  
0, -1, 0
```

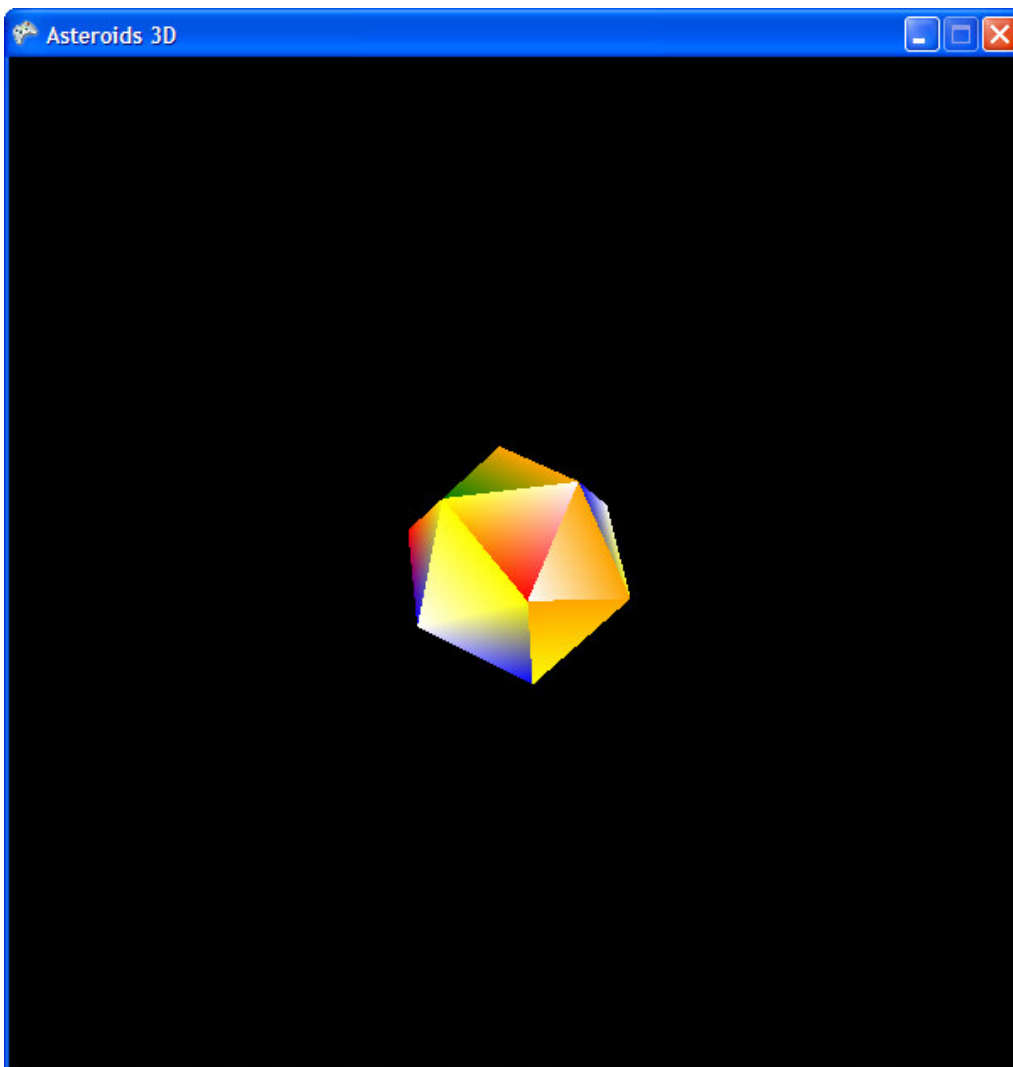
```
0, 1, 0
0, 0, 1
1, 0, 0
```

```
0, 1, 0
0, 0, -1
-1, 0, 0
```

```
0, -1, 0
0, 0, 1
-1, 0, 0
```

```
0, -1, 0
0, 0, -1
1, 0, 0
```

Icosahedron (20 triangles – 60 vertices)



-0.26286500f, 0.00000000f, 0.26286500f,	0.00000000f, -0.42532500f, 0.00000000f,	0.42532500f 0.26286500f 0.42532500f
-0.26286500f, -0.42532500f, 0.00000000f,	0.00000000f, -0.26286500f, -0.42532500f,	0.42532500f 0.00000000f 0.26286500f
0.26286500f, 0.00000000f, -0.26286500f,	0.00000000f, 0.42532500f, 0.00000000f,	0.42532500f 0.26286500f 0.42532500f
0.26286500f, 0.42532500f, 0.00000000f,	0.00000000f, 0.26286500f, 0.42532500f,	0.42532500f 0.00000000f 0.26286500f
0.26286500f, 0.42532500f, 0.42532500f,	0.00000000f, -0.26286500f, 0.26286500f,	0.42532500f 0.00000000f 0.00000000f
-0.26286500f, 0.00000000f, 0.26286500f,	0.00000000f, 0.42532500f, 0.00000000f,	-0.42532500f -0.26286500f -0.42532500f
-0.26286500f, -0.42532500f, 0.00000000f,	0.00000000f, 0.26286500f, 0.42532500f,	-0.42532500f 0.00000000f -0.26286500f
-0.26286500f, -0.42532500f, -0.42532500f,	0.00000000f, -0.26286500f, 0.26286500f,	-0.42532500f 0.00000000f 0.00000000f
0.26286500f, 0.00000000f, -0.26286500f,	0.00000000f, -0.42532500f, 0.00000000f,	-0.42532500f -0.26286500f -0.42532500f
0.26286500f, 0.42532500f, 0.00000000f,	0.00000000f, -0.26286500f, -0.42532500f,	-0.42532500f 0.00000000f -0.26286500f
0.00000000f, 0.42532500f, 0.00000000f,	0.42532500f, 0.26286500f, 0.42532500f,	0.26286500f 0.00000000f -0.26286500f
0.00000000f, -0.42532500f, -0.26286500f,	0.42532500f, 0.26286500f, 0.00000000f,	0.26286500f 0.00000000f 0.42532500f
0.00000000f, 0.42532500f, 0.26286500f,	0.42532500f, 0.26286500f, 0.00000000f,	-0.26286500f 0.00000000f -0.42532500f
0.00000000f, -0.42532500f, 0.00000000f,	0.42532500f, 0.26286500f, 0.42532500f,	-0.26286500f 0.00000000f 0.26286500f
0.00000000f, 0.00000000f,	-0.42532500f, 0.26286500f,	0.26286500f

0.42532500f,	-0.26286500f,	0.00000000f
0.26286500f,	0.00000000f,	0.42532500f
0.00000000f,	-0.42532500f,	0.26286500f
-0.42532500f,	-0.26286500f,	0.00000000f
0.00000000f,	-0.42532500f,	-0.26286500f
0.00000000f,	-0.42532500f,	-0.26286500f
0.42532500f,	-0.26286500f,	0.00000000f
0.00000000f,	-0.42532500f,	0.26286500f
0.00000000f,	-0.42532500f,	-0.26286500f
-0.42532500f,	-0.26286500f,	0.00000000f
-0.26286500f,	0.00000000f,	-0.42532500f
0.42532500f,	0.26286500f,	0.00000000f
0.42532500f,	-0.26286500f,	0.00000000f
0.26286500f,	0.00000000f,	-0.42532500f
-0.42532500f,	0.26286500f,	0.00000000f
-0.42532500f,	-0.26286500f,	0.00000000f
-0.26286500f,	0.00000000f,	0.42532500f

Coloring Code

Each vertex needs a color. You can color your objects in any way that you want. I'm including my code for coloring here, in case you want to use it. This code needs to go before the actual drawing is done, so I've put it at the bottom of the `SetupVertexArray()` method.

```

Random random = new Random();
for (int i = 0; i < vertexArray.Length; i++)
{
    switch(random.Next(6))
    {
        case 0:
            vertexArray[i].Color = Color.Red;
            break;
        case 1:
            vertexArray[i].Color = Color.Blue;
            break;
        case 2:
            vertexArray[i].Color = Color.Yellow;
            break;
        case 3:
            vertexArray[i].Color = Color.Green;
            break;
        case 4:
            vertexArray[i].Color = Color.White;
            break;
        case 5:
            vertexArray[i].Color = Color.Orange;
            break;
    }
}

```