Computer Graphics
Aum Patel

Contents

Chapter 1

Question 1 - Geometry and vertex attributes

Y-axis up, X axis left to right and Z axis towards you is a Right handed coordinate system. I have proven this by using my right-hand with the X-axis being on the thumb, the Y-axis being on the first finger and the Z-axis being on my middle finger and rotating it to meet the 3 requirements. You cannot rotate the left hand to do this.

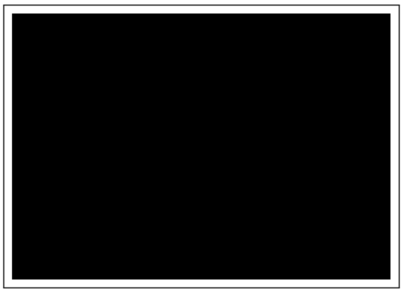

Figure 1.1: Image of hand with axis labeled.

I decided to draw out the model I was going to make in Blender first and then place the vertices on top of it.

The colors of the axis are consistent in all images: Red : X , Green : Y , Blue : Z

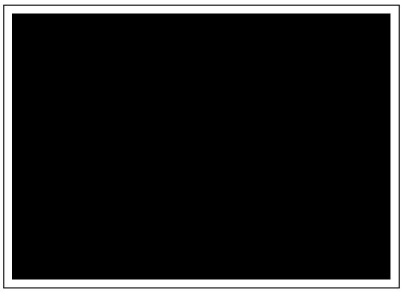

Figure 1.2: House in blender with drawn axis.

I chose the bottom vertex on the rear left of the house as the origin as I would only have to work with positive numbers for the rest of the vertices, and they would also be round numbers (except for the roof) as I am going to make the house with a square base of 1x1.

Below is a wireframe view of the same house with the coordinates labeled.

Coordinates for the vertices:

(a) (0, 0, 0)

[redacted]

Figure 1.3: House in blender with wireframe and coordinates.

(b) (0, 0, 1)

(c) (1, 0, 0)

(d) (1, 0, 1)

(e) (0, 1, 0)

(f) (0, 1, 1)

(g) (1, 1, 0)

(h) (1, 1, 1)

(i) (0.5, 2, 0)

(j) (0.5, 2, 1)

16 triangles will be used to make up the house, and there will be normals that are shared between them.

I calculated the vector normals by using the following equation:

If we have 3 vectors that make up a a triangle in an anti-clockwise manner: $V0, V1, V2$, to calculate the normal facing outwards we do:

$A = V1 - V0 \mid B = V2 - V0$

$Normal = A \times B$

Doing this with the first triangle on the table (triangle on the left face of the house) give you:

$A = (0, 0, 1) - (0, 0, 0)$
$A = (0, 0, 1)$

$B = (0, 1, 1) - (0, 0, 0)$
$B = (0, 1, 1)$

$Normal = (0, 0, 1) \times (0, 1, 1)$
$Normal = (-1, 0, 0)$

I confirmed this calculation to be correct by looking at the shape itself in 3D space, and $(-1, 0, 0)$ is the normal that would be correct.

I had no need to normalise the vectors as they output from the cross product was a sensible number.

Same calculation is done for the rest of the sides, triangles facing the same direction will have the same surface normals.

Triangles	**Normals**
(a, b, f) (a, f, e)	(-1, 0, 0)
(b, h, f) (b, d, h) (f, h, j)	(0, 0, 1)
(d, g, h) (d, c, g)	(1, 0, 0)
(c, e, g) (c, a, e) (g, e, i)	(0, 0, -1)
(a, c, b) (b, c, d)	(0, -1, 0)
(e, j, i) (e, f, j)	(-1, 0.5, 0)
(h, i, j) (h, g, i)	(1, 0.5, 0)

To write the .obj file, I have to change the alphabetical indices that I have been using into numerical. I have rounded the numbers to 6 decimal places. I have the same number of vertex normals as vertices as I decided to make the shape smooth shaded.

NOTE THE VERTEX NORMALS HAVE NOT BEEN NORMALISED.

```
   v 0 0 0
   v 0 0 1
   v 1 0 0
   v 1 0 1
   v 0 1 0
   v 0 1 1
   v 1 1 0
   v 1 1 1
   v 0.5 2 0
   v 0.5 2 1
  vn -0.333333 -0.333333 -0.333333
  vn -0.333333 -0.333333 0.333333
  vn 0.333333 -0.333333 -0.333333
  vn 0.333333 -0.333333 0.333333
  vn -0.666667 0.166667 -0.333333
  vn -0.666667 0.166667 0.333333
  vn 0.0 0.166667 -0.333333
  vn 0.0 0.166667 0.333333
  vn 0.0 0.333333 -0.333333
  vn 0.0 0.333333 0.333333
usemtl matWall
   f 1//1 2//2 6//6
   f 1//1 6//6 5//5
```

```
    f 2//2 8//8 6//6
    f 2//2 4//4 8//8
    f 6//6 8//8 10//10
    f 4//4 7//7 8//8
    f 4//4 3//3 7//7
    f 3//3 5//5 7//7
    f 3//3 1//1 5//5
    f 7//7 5//5 9//9
    f 1//1 3//3 2//2
    f 2//2 3//3 4//4
usemtl matRoof
    f 5//5 10//10 9//9
    f 5//5 6//6 10//10
    f 8//8 9//9 10//10
    f 8//8 7//7 9//9
```

Chapter 2

Question 2 - Vertex Buffer Object(VBO) design and transformations

`You now have to design a VBO to contain the vertex attributes for your house. Choose a suitable la`

`Your house is to be translated so it is centred at a new position on the ground plane, rotated by`

My single VBO will look like this:

`x y z x y z x y z r g b r g b r g b`

which is the standard format, making the attributes tightly packed.

I will add the vertices for the triangles into an array of floats of size 9, three coordinates per triangle in the counter-clockwise order they should be drawn to render outside, for example the first triangle will look like this: 0, 0, 0 0, 0, 1 0, 1, 1

As the RGB values are in the VBO, I will initialise them as an array of size 9 as well. 1, 1, 1, 1, 1, 1, 1, 1, 1