Assignment 1 : Surface Normal Calculation

1. Normal array should have the size of a vertex array.
2. Normal is a vec3
3. An example has been uploaded (under module week 3) that shows how to calculate normals for a cube. Each vertex of a face has the same normal, i.e., normal of the face.

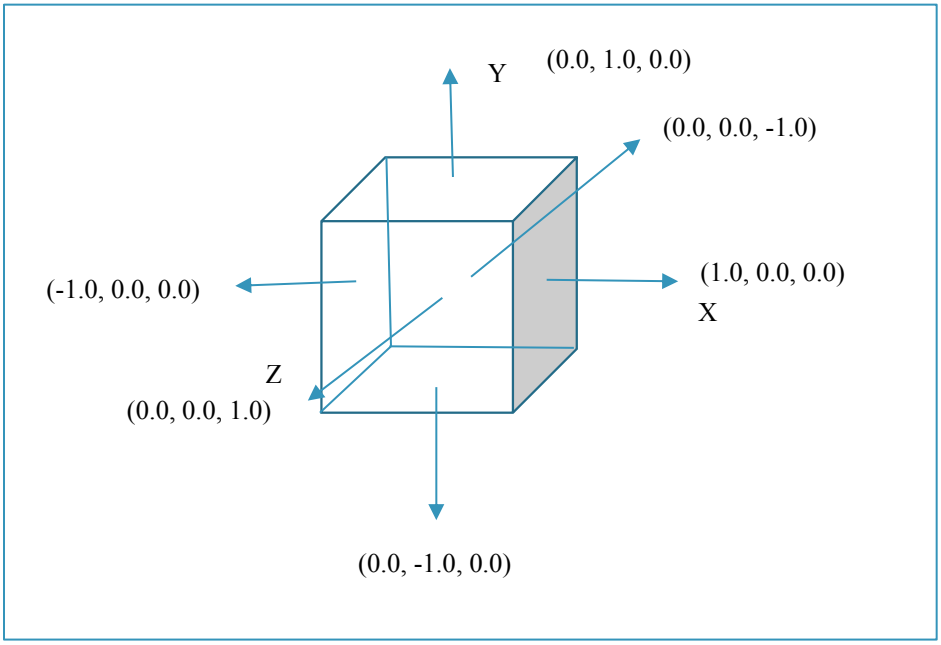


Figure 1: Direction of normals for different faces of a cube.

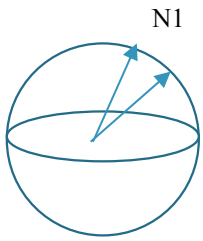
Normal Calculation for a Cone

// Per vertex normal calculation. Suppose you already have the index array for the cone.

```
for (i = 0; i < NumIndices; i += 3) { // each three indices represent a triangle/ face of cone

              vec4 p1 = points[indices[i]];
              vec4 p2 = points[indices[i + 1]];
              vec4 p3 = points[indices[i + 2]];

              vec3 normal = normalize(cross(vec3(p2 - p1), vec3(p3 - p1)));// triangle normal

              // keeps on adding normals for the same index for different triangles
              normals[indices[i]] += normal;
              normals[indices[i + 1]] += normal;
              normals[indices[i + 2]] += normal;
}

       // Normalize the normal:
```

// Calculation of normal for a sphere:

This is the easiest one. For a sphere with its center at origin, any point on the surface of the sphere represents the direction of the normal. But you need to normalize the normal.




```
for (int a = 0; a < STEP; a++) {
                        // lat/lon coordinates

     phi = (1.0 * a / STEP) * 2 * kPI;
     theta = (1.0 * b / STEP) *kPI;

     vertex[i].x = cos(theta)*sin(phi);
     vertex[i].y = sin(theta);
     vertex[i].z = cos(theta)*cos(phi);
     vertex[i].w = 1.0f;
      normal[i] = normalize(vec3(vertex[i]));
     i++;
}
```