

## Texturizado 2

**Objetivo:** El alumno utilizará el mapeo de texturas para cubrir diversas primitivas.

1. Investigue la manera de construir matemáticamente las siguientes primitivas geométricas, es decir, obtener los vértices que formaran a la figura y sus reglas de unión: (no utilizar funciones de GLUT)

a) **Prisma, debe recibir como parámetros el alto (eje Y), ancho (eje X) y profundidad (eje Z).**

```
void Mycube(GLfloat length, GLfloat height, GLfloat depth)
{
    //base
    glBegin(GL-POLYGON);
    glVertex3f(-length/2, -height/2, depth/2);
    glVertex3f(-length/2, -height/2, -depth/2);
    glVertex3f(length/2, -height/2, -depth/2);
    glVertex3f(length/2, -height/2, depth/2);
    glEnd();
    // front face
    glBegin(GL-POLYGON);
    glVertex3f(-length/2, -height/2, depth/2);
    glVertex3f(length/2, -height/2, depth/2);
    glVertex3f(length/2, height/2, depth/2);
    glVertex3f(-length/2, height/2, depth/2);
    glEnd();
    // right side face
    glBegin(GL-POLYGON);
    glVertex3f(length/2, -height/2, depth/2);
    glVertex3f(length/2, -height/2, -depth/2);
    glVertex3f(length/2, height/2, -depth/2);
    glVertex3f(length/2, height/2, depth/2);
    glEnd();
    // back side face
    glBegin(GL-POLYGON);
    glVertex3f(-length/2, -height/2, -depth/2);
    glVertex3f(-length/2, height/2, -depth/2);
    glVertex3f(length/2, height/2, -depth/2);
    glVertex3f(length/2, -height/2, -depth/2);
    glEnd();
    //left side face
    glBegin(GL-POLYGON);
    glVertex3f(-length/2, -height/2, depth/2);
    glVertex3f(-length/2, height/2, depth/2);
```

```

glVertex3f(-length/2, height/2, -depth/2);
glVertex3f(-length/2, -height/2, -depth/2);
glEnd();
// top
glBegin(GL-POLYGON);
glVertex3f(-length/2, height/2, depth/2);
glVertex3f(length/2, height/2, depth/2);
glVertex3f(length/2, height/2, -depth/2);
glVertex3f(-length/2, height/2, -depth/2);
glEnd();
}

```

**a) Cono, debe de recibir como parámetros la altura (eje Y), radio y la resolución de su circunferencia final.**

```

void MyCone(GLfloat radius, GLfloat height, GLint circle_points)
{
    GLint i;
    GLdouble theta, ntheta;
    for (i=0; i < circle_points; i++)
    {
        glBegin(GL-POLYGON);
        theta = (2*Pi*i/circle_points);
        ntheta = (2*Pi*(i + 1)/circle_points);
        glVertex3f(radius*cos(theta), 0, radius*sin(theta));
        glVertex3f(0,height,0);
        glVertex3f(radius*cos(ntheta), 0, radius*sin(ntheta));
        glEnd();
    }
}

```

**Cilindro, debe de recibir como parámetros la altura (eje Y), radio de las circunferencias y su resolución.**

```

Void MyCylinder(float radius, float halfLength, int slices)
for(int i=0; i<slices; i++)
{
    float theta = ((float)i)*2.0*M_PI;
    float nextTheta = ((float)i+1)*2.0*M_PI;
    glBegin(GL_TRIANGLE_STRIP);
    /*vertex at middle of end*/
    glVertex3f(0.0, halfLength, 0.0);
    /*vertices at edges of circle*/
    glVertex3f(radius*cos(theta), halfLength, radius*sin(theta));
    glVertex3f (radius*cos(nextTheta), halfLength, radius*sin(nextTheta));
    /* the same vertices at the bottom of the cylinder*/
    glVertex3f (radius*cos(nextTheta), -halfLength, radius*sin(nextTheta));
}

```

```
glVertex3f(radius*cos(theta), -halfLength, radius*sin(theta));  
glVertex3f(0.0, -halfLength, 0.0);  
glEnd();  
}  
b)
```

**2. Indique la diferencia entre un objeto transparente y un objeto translucido.**

El cuerpo transparente: Es aquel que deja pasar casi toda la luz que incide en él, la intensidad de la luz incidente es muy parecida a la transmitida (la que lo atraviesa). Como ejemplo están el agua, el vidrio, algunas bebidas distintas al agua y algunos tipos de plásticos.

Los cuerpos translúcidos: Son los que dejan pasar una porción de la luz incidente. El cuerpo translucido dejaría pasar aproximadamente la mitad de la luz que incide sobre él. Los objetos translúcidos tienen la propiedad de dejar pasar la luz pero no se puede ver a través de él, las formas y colores se distorsionan, este es el caso del hielo, fibra de vidrio y algunos plásticos.

En la página de la materia pueden descargar el libro "Principles of Computer Graphics" y checar las páginas 95 a 100 de ejemplos de cómo construir las geometrías.

**Conclusiones**

Con el desarrollo de este previo aprendí cómo construir las primitivas básicas geométricas: el prisma, el cono y el cilindro. Además, aprendí las diferencias dentro de la computación gráfica que tienen un objeto transparente y uno translucido.

