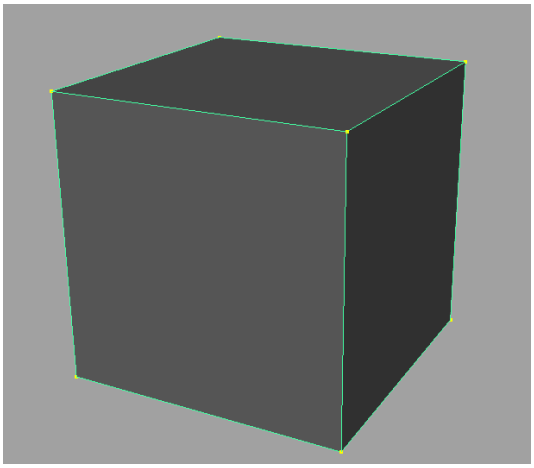
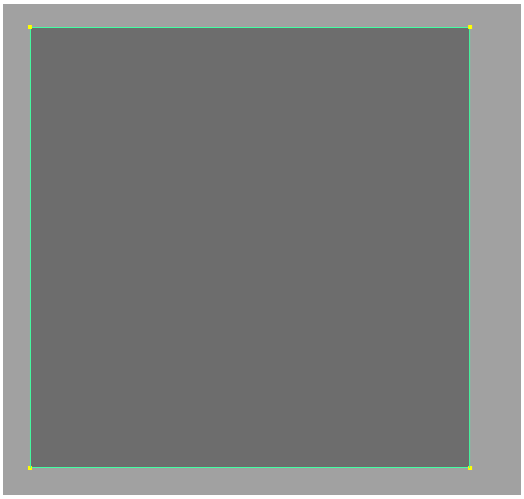
Ayuda (Iluminación)

Esto proyecto es un poco más complejo para iluminar, en el proyecto anterior si ustedes querían texturizar un cubo, tenían que crear el cubo con seis planos colocados adecuadamente.



Cada plano era un cuadrilátero de cuatro vértices:



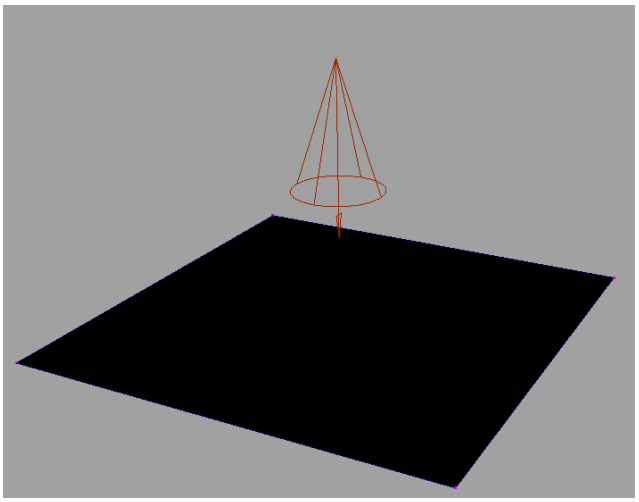
Para este proyecto van a tener que hacer modificaciones en este plano. La razón se debe a la iluminación.

Opengl para iluminar una superficie lo que hace es iluminar cada vértice del objeto, luego mediante una fórmula determina el color del vértice e interpola dicho color entre los otros vértices para pintar el cuadrilátero.

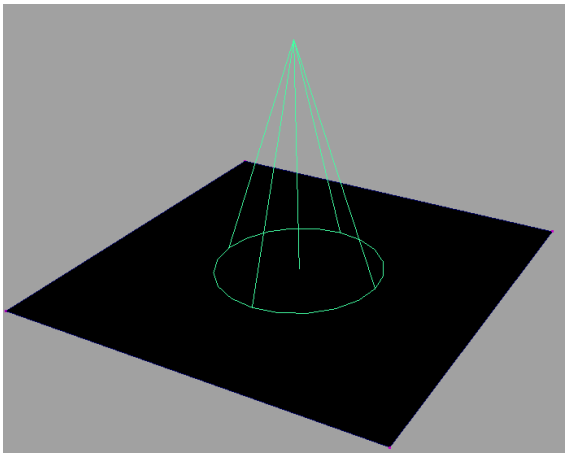
En el proyecto II todos vieron que se iluminaba su escena de manera adecuada por que usaban luces puntuales o luces direccionales, esto garantizaba que cada vértice recibiera un rayo de luz.

Pero para este proyecto tienen que usar una luz con forma de cono (SpotLight), que ilumina solo el área que forma dicho cono, y con esta iluminación el plano de arriba no funciona.

Ejm:

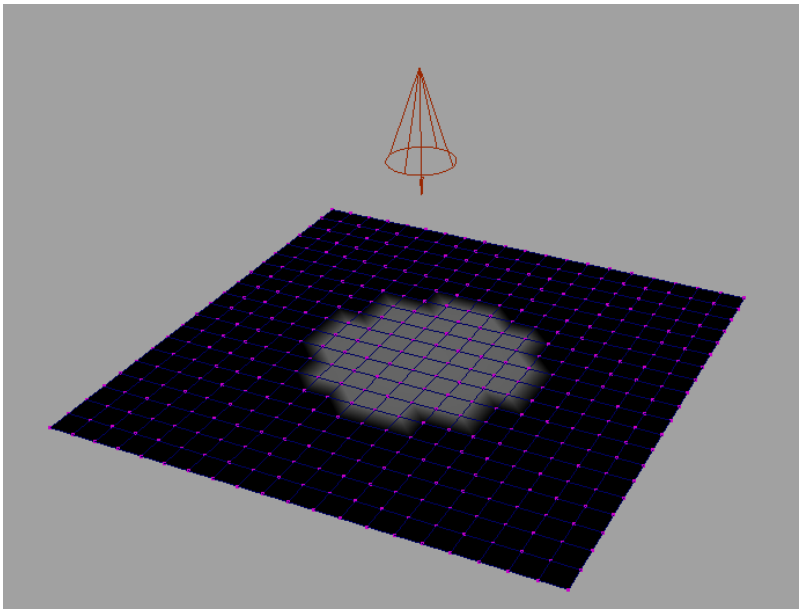


Noten que el plano no está iluminado ya que está completamente negro, esto se debe a que el cono no logra iluminar ningún vértice:

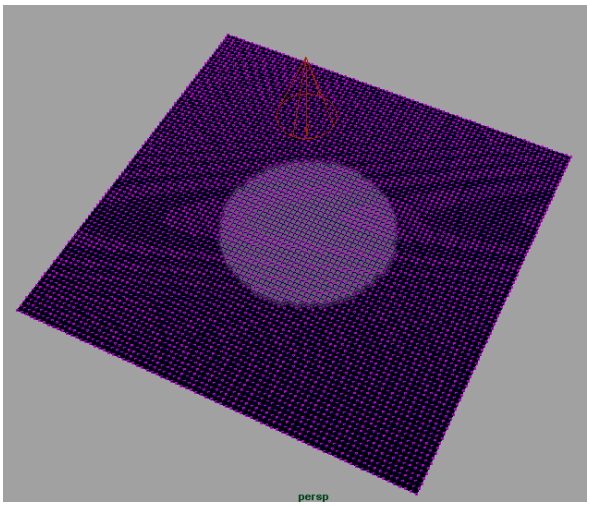


En la imagen, noten como el área del cono no encierra ningún vértice del plano.

La solución para ello, no es necesariamente hacer el cono mas grande, sino dividir (añadir más vértices) el plano.



Noten en la imagen, como el plano empieza a iluminarse debido a que ahora si hay vértices (puntos morados) que les llega la iluminación del spot light, mientras más vértices el plano tenga, mas circular será el área iluminada.



¿Cómo hacemos eso en opengl?.

Basicamente tienen que crear una función que construya el plano según un parámetro de división.

Ejm

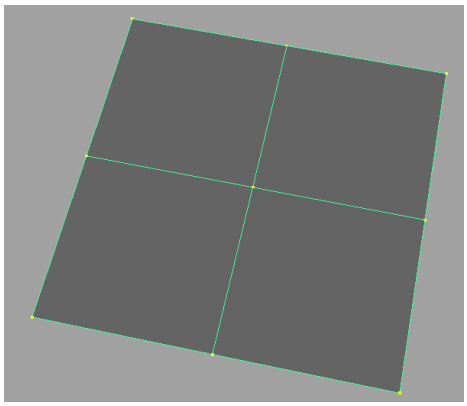
Función crearPlano(división = 1)

```
glBegin(GL_QUADS);
            glVertex3f(1,0,1);
            glVertex3f(-1,0,1);
            glVertex3f(-1,0,-1);
            glVertex3f(1,0,-1);
glEnd();
```

Función crearPlano(división = 2)

```
glBegin(GL_QUADS);
            glVertex3f(1,0,1);
            glVertex3f(0,0,1);
            glVertex3f(0,0,0);
            glVertex3f(1,0,0);
glEnd();
glBegin(GL_QUADS);
            glVertex3f(0,0,1);
            glVertex3f(-1,0,1);
            glVertex3f(-1,0,0);
            glVertex3f(0,0,0);
glEnd();
glBegin(GL_QUADS);
            glVertex3f(0,0,0);
            glVertex3f(-1,0,0);
            glVertex3f(-1,0,-1);
            glVertex3f(0,0,-1);
glEnd();
glBegin(GL_QUADS);
            glVertex3f(1,0,0);
            glVertex3f(0,0,0);
            glVertex3f(0,0,-1);
            glVertex3f(1,0,-1);
glEnd();
```

En división = 1, tienen el plano sencillo generado por 4 vértices en división = 2 tendrían un plano igual a este:



Es decir, aunque yo muestro un proceso manual, ustedes deben crear una función que construya el plano dividiendolo según la variable de entrada de la función (división).

¿Esto afecta las texturas?

Si. Pero lo solución está en también dividir el espacio UV del plano.

TIP

Trabajen con un plano unitario de coordenadas (0,0,0), (1,0,0), (1,1,0) y (0,1,0), y luego para crear un paralelepípedo seria llamar la función que les crea el plano 6 veces, rotándola y trasladándola convenientemente.