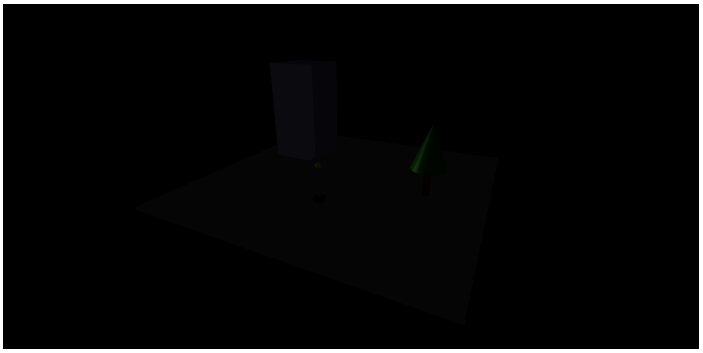
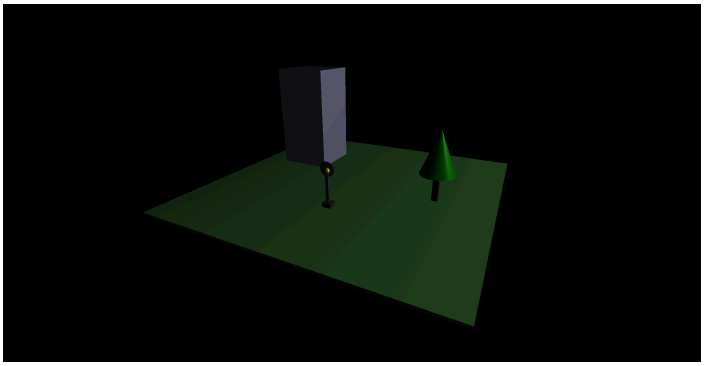
INFORME AA2 - Informática gráfica

Roger Alamañac, Albert March, Jimena De Aldecoa

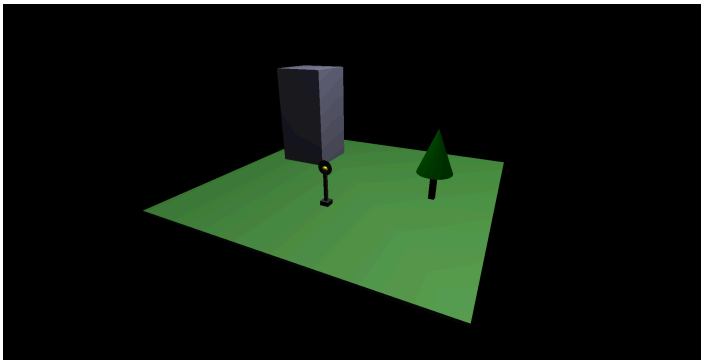
NOCHE:



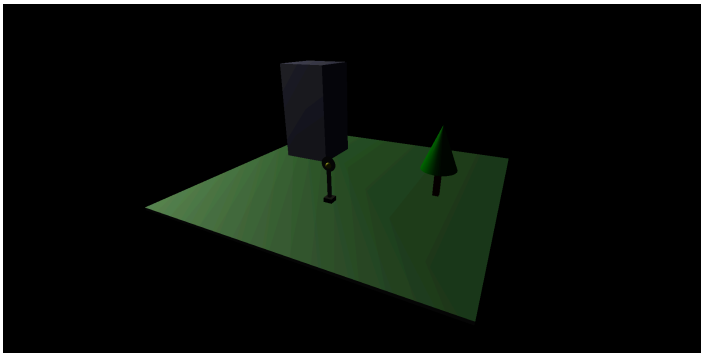
MAÑANA:



MEDIODIA:



TARDE:



Transformaciones 3D en el modelo

Las transformaciones se aplican manualmente a cada objeto mediante las funciones glTranslatef, glScalef y glRotatef, dentro de bloques glPushMatrix() y glPopMatrix() que aíslan las transformaciones locales de cada objeto.

1. **Suelo**
   glScalef(10.0f, 0.1f, 10.0f); glutSolidCube(1);
   Se parte de un cubo de tamaño 1 y se aplana en el eje Y para simular una plataforma, y se extiende en X y Z para ocupar más terreno.

2. **Edificio**
   glTranslatef(-3.0f, 2.0f, -2.0f); glScalef(2.0f, 4.0f, 2.0f); glutSolidCube(1);
   Se traslada a la izquierda, arriba y hacia el fondo.
   Se escala para simular una forma rectangular alargada como un bloque de edificio.

3. **Árbol**
   Tronco:
   glTranslatef(3.0f, 0.5f, 0.0f); glScalef(0.2f, 1.0f, 0.2f); glutSolidCube(1);
   Se posiciona a la derecha y se estira en vertical, simulando un tronco delgado.
   Copa:
   glTranslatef(3.0f, 1.0f, 0.0f); glRotatef(-90, 1, 0, 0); glutSolidCone(0.6, 1.5, 20, 20);
   Se coloca sobre el tronco, rotada -90° en el eje X para que la copa cónica apunte hacia arriba.

4. **Farola**
   Base:
   glTranslatef(0.0f, 0.1f, 2.0f); glScalef(0.3f, 0.2f, 0.3f); glutSolidCube(1.0f);
   Poste:
   glTranslatef(0.0f, 0.7f, 2.0f); glScalef(0.1f, 1.2f, 0.1f); glutSolidCube(1.0f);
   Pantalla:
   glTranslatef(0.0f, 1.35f, 2.0f); glRotatef(180, 1, 0, 0); glutSolidCone(0.25, 0.3, 20, 20);
   Bombilla:
   glTranslatef(0.0f, 1.3f, 2.0f); glutSolidSphere(0.1, 20, 20);
   La farola se construye por partes: base, mástil, pantalla y bombilla, todas posicionadas verticalmente una sobre otra. Las transformaciones aseguran que se monten en la posición correcta.

Viewing: cámara y proyección

Cámara (View Transformation)

En la función display() se establece la cámara con:
La cámara está colocada en (5, 5, 10), una vista desde arriba a la derecha.
Apunta hacia el centro de la escena (0, 0, 0).
El eje Y es considerado como “arriba” (up vector).
Esto genera una vista en perspectiva con profundidad clara.

Proyección

En la función reshape() se define una proyección en perspectiva:
Ángulo de visión vertical de 60°.
aspect es la relación de aspecto entre ancho y alto.
El frustum va del plano cercano 1.0 al lejano 100.0.
Esto crea un volumen de visión en forma de pirámide truncada, simulando una percepción realista de profundidad.

Transformación dinámica del sol

La luz simula un sol girando alrededor del escenario:
float x = radius * cosf(sunAngle); float y = radius * sinf(sunAngle); float pos[] = { x, y, 0.0f, 1.0f }; glLightfv(GL_LIGHT0, GL_POSITION, pos);
La posición de la luz se actualiza cada frame en la función idle(), modificando sunAngle.
Esto genera un movimiento circular de la luz alrededor de la escena.
Además, se ajusta la intensidad de la luz en función de su altura (y), simulando día y atardecer:
float intensity = fmax(0.2f, y / radius);

Modelo de Iluminación Utilizado

Iluminación ambiente global

float globalAmbient[] = { 0.2f, 0.2f, 0.2f, 1.0f };
glLightModelfv(GL_LIGHT_MODEL_AMBIENT, globalAmbient);
Esta luz se aplica de manera uniforme en toda la escena, independientemente de la posición de la fuente de luz.
Simula la iluminación ambiental del entorno (como luz difusa del cielo en un día nublado).

Fuente de luz direccional (GL_LIGHT0)

glEnable(GL_LIGHTING); glEnable(GL_LIGHT0);
Se activa el sistema de iluminación de OpenGL y la luz 0.
Esta fuente tiene componente ambiente, difusa y especular:
float ambient[] = { 0.1f, 0.1f, 0.1f, 1.0f }; float diffuse[] = { 1.0f, 1.0f, 0.9f, 1.0f };
float specular[] = { 1.0f, 1.0f, 0.9f, 1.0f };
La luz se define como blanca cálida, con un leve tono amarillo.

Materiales de los objetos

Cada objeto usa una función que define cómo reacciona su superficie a la luz:
glMaterialfv(GL_FRONT, GL_AMBIENT, ambient);
glMaterialfv(GL_FRONT, GL_DIFFUSE, diffuse);
glMaterialfv(GL_FRONT, GL_SPECULAR, specular);
glMaterialf(GL_FRONT, GL_SHININESS, 50.0f);
Todos los materiales tienen reflexión ambiente, difusa y especular. El brillo (shininess) de 50.0 genera reflejos especulares moderadamente nítidos, como superficies semi-brillantes (plástico o metal suave).

Variación de la iluminación con el tiempo

La iluminación varía en tiempo real simulando el movimiento del sol alrededor de la escena.

Movimiento circular del sol

sunAngle += timeStep * (M_PI / 180.0f);
sunAngle se incrementa progresivamente en la función idle(), simulando el paso del tiempo.
Se convierte en coordenadas (x, y) para colocar la luz en un círculo:
float x = radius * cosf(sunAngle); float y = radius * sinf(sunAngle); float pos[] = { x, y, 0.0f, 1.0f }; glLightfv(GL_LIGHT0, GL_POSITION, pos);

Esto hace que la luz se mueva en el plano X-Y como si fuera el sol orbitando la escena.

Variación de intensidad según la altura del sol

float intensity = fmax(0.2f, y / radius);
La intensidad de la luz varía según la altura (Y) del sol.
Si el sol está por encima del horizonte, la intensidad es alta; si está abajo, se limita a un mínimo de 0.2f para evitar oscuridad total.
Esta intensidad afecta a:
float diffuse[] = { intensity, intensity, intensity * 0.9f, 1.0f };
float specular[] = { intensity, intensity, intensity * 0.9f, 1.0f };
La luz se vuelve más suave o más brillante dependiendo del momento del "día".