



**Universidad Nacional
Autónoma de México.**



Facultad de Ingeniería.

Laboratorio de computación gráfica.

Proyecto final.

Manual de usuario.

José de Jesús Hernández Castro.

Semestre 2020-2

Funciones del teclado.

Dentro del entorno virtual construido, es posible realizar algunas acciones con ayuda del teclado, a continuación se muestra una lista de las teclas específicas y su funcionalidad.

*Ninguna de las acciones es sensible a mayúsculas/minúsculas.

- **+/-**: Aumento en la velocidad del tiempo transcurrido entre el día y la noche.

*La velocidad mínima es de **x1** y la velocidad máxima es de **x10**.

- **L**: Enciende/apaga la luz de la lámpara sobre el escritorio.

*No se puede encender la lámpara durante el día.

*Para poder encender/apagar la lámpara durante la noche, se debe estar posicionado cerca de la misma.

- **O**: Abre/cierra la puerta del cuarto principal.

*Para poder abrir/cerrar la puerta, se debe estar posicionado cerca de la misma.

- **R**: Rota en un movimiento de 0° a 360° el globo terráqueo ubicado en una de las repisas del librero.

- **Cursores de navegación** ↑↓←→: Movimiento de vista de cámara arriba, abajo, izquierda y derecha respectivamente.

- **W/S/A/D**: Movimiento de cámara adelante, atrás, izquierda y derecha respectivamente.

- **Repag/Avpag**: Movimiento de cámara arriba y abajo respectivamente.

Animaciones.

A continuación se detallan las transformaciones utilizadas en las animaciones dentro del entorno virtual.

- **Animación de la apertura/cierre de puerta:** Para lograr la animación de la puerta, se aplicó una traslación equivalente a la mitad de las unidades de ancho de la puerta, colocando así el centro de la geometría en el pivote de rotación deseado en uno de los costados del marco, después se realizó una rotación gradual entre 0° y 90° con incrementos de $\pm 2^\circ$ dependiente de la acción abrir/cerrar, para finalizar nuevamente con una traslación equivalente a la mitad de las unidades de ancho de la puerta, ubicándola así completamente dentro del marco.
- **Animación del encendido/apagado de lámpara:** En primer lugar se dibuja la geometría de la lámpara siendo el último elemento el foco, una vez hecho esto, se crea una luz del tipo reflector (spotlight) con su componente difusa en color amarillo, ángulo de apertura de 45° (90° en total), exponente de la luz equivalente a 5 (difuminado en los bordes), dirección $\{0.25, -1.0, 0.5\}$ (apuntando al libro en la mesa) y posición igual a la del foco. Para lograr el efecto de fuente de luz, los pasos son:
 1. Desactivar modelo de iluminación.
 2. Dibujar la geometría del foco.
 3. Ubicar la fuente de luz en las coordenadas de la geometría.
 4. Activar modelo de iluminación.
- **Animación del globo terráqueo:** Es una animación sencilla que se logra aplicando una transformación de rotación gradual de 0° a 360° sobre el eje vertical. Es importante que la geometría se haya modelado jerárquicamente para que la rotación siga la orientación del soporte de medio arco.
- **Animación de la nube voladora:** Se aplicó una función del tipo $y = A\sin(z) + B$, con $z \in [90, -90]$ en donde:
 - A es la amplitud del recorrido arriba-abajo de la nube mientras se mueve sobre el eje Z.
 - B es la altura desde el suelo (origen del sistema de referencia).

Logrando así un efecto de movimiento suave como si la nube flotara con el aire.

*La animación de la nube voladora sólo aparecerá durante el día.

- **Animación de la luna:** Se aplicó una función del tipo $y = Az^2 + Bz + C$ (parábola) con $z \in [90, -90]$ para el movimiento de traslación, en donde:
 - A es un valor negativo para lograr la orientación “hacia abajo” de la parábola y menor a 1. Mientras más pequeño, mayor es la apertura de la trayectoria.
 - B es igual a 0 para no desplazar la trayectoria con referencia al sistema de coordenadas del mundo.
 - C el valor del máximo en y (altura máxima).

Durante el recorrido, se utilizó una transformación de rotación de 0° a 360° sobre el eje vertical de la geometría de la luna.

*La animación de la luna sólo aparecerá durante la noche.

- **Animación del reloj día/noche:** Para lograr el texturizado en 2D sobre la pantalla, se implementó una función dedicada cambiar el modelo de visualización de perspectiva a ortogonal, luego se procede a dibujar la texturas del reloj y la manecilla, esta última con una rotación cíclica de 0° a 360° sobre el eje Z. Finalmente se regresa al modelo en perspectiva.
- **Efecto de cambio de escenario día/noche:** Debido a que las texturas no interactúan con el modelo de iluminación, para lograr el efecto de oscurecimiento durante la noche, estos fueron los pasos realizados:
 1. A cada una de las texturas originales, se aplicó un filtro de iluminación baja en un editor de imágenes.
 2. Se cargaron todas las texturas, tanto para el día como para la noche.
 3. En un arreglo de 40 elementos, se almacenaron los índices de las texturas de día.
 4. Se implementó una función dedicada al intercambio de índices entre las texturas de día y las de noche.
 5. Se implementó una función dedicada a la recuperación de los índices originales, desde el arreglo, de las texturas de día.

El proceso de re-texturizado depende de una variable del tipo bool (bandera).

*Los modelos en formato .3ds, que cuentan con texturizado, no tienen ninguna modificación.

Listado de funciones en el código fuente principal y su utilidad.

- void Texture_Load(): Carga todas las texturas utilizadas en el programa.
- void Models_Load(): Carga todos los modelos con formato .3ds.
- void Luces(): Crea y controla todas las luces del programa.
- void Oscurecer(): Intercambia los índices de las texturas de día y noche.
- void Restaurar_Indice(): Recupera los índices originales de las texturas de día.
- void Dibuja_suelo (): Construye las geometrías que componen el suelo.
- void Dibuja_Silla(): Construye las geometrías que componen la silla.
- void Dibuja_Escritorio(): Construye las geometrías que componen el escritorio.
- void Dibuja_Librero(): Construye las geometrías que componen el librero.
- void Dibuja_Globo(): Construye las geometrías que componen el globo terráqueo.
- void Dibuja_Libro(CTexture front, CTexture reverse, CTexture spine): Construye las geometrías que componen el libro sobre la mesa de centro.
- void Dibuja_Mesa(): Construye las geometrías que componen la mesa de centro.
- void Dibuja_Lampara(): Construye las geometrías que componen la lámpara.
- void OnScreen_GUI(): Cambia el modo de visualización de perspectiva a ortogonal, dibuja las texturas 2D sobre la escena 3D, regresa el modo de visualización a perspectiva.
- void InitGL(GLvoid): Parámetros iniciales de la aplicación.
- void pintaTexto(float x, float y, float z, void *font,char *string): Muestra texto en pantalla.
- void display(): Función de dibujo.
- void animación(): Controla los parámetros, banderas y condiciones de las animaciones.
- void reshape (int width , int height): Función de redimensionado y modelo de visualización de la ventana.
- void keyboard (unsigned char key, int x, int y): Función para el manejo de teclado.
- void arrow_keys (int a_keys, int x, int y): Función para el manejo de teclas especiales.
- int main (int argc, char** argv): Función principal.

Listado de funciones de la cabecera de figuras y su utilidad.

- **void esfera(GLfloat radio, int meridianos, int paralelos, GLuint text):** Esfera texturizada.
- **void cilindro(float radio, float altura, int resolucion, GLuint text):** Cilindro texturizado.
- **void cono(float altura, float radio, int resolucion, GLuint text):** Cono texturizado.
- **void cubemap(GLuint text, GLuint text2, GLuint text3, GLuint text4):** Cubemap especial dedicado al cuarto de gohan.
- **void prisma (float altura, float largo, float profundidad, GLuint text):** Prisma con texturizado homogéneo.
- **void prisma2(GLuint text, GLuint text2):** Prisma con texturizado 5-1.
- **void prisma_r_u(GLuint text, GLuint text2):** Prismar que repite la textura 50 veces en U.
- **void prisma_l_o(GLuint text, GLuint text2):** Prisma que texturiza un prisma con la misma imagen en lados opuestos.
- **void prisma3(GLuint text, GLuint text2):** Prisma con texturizado 5-1.
- **void plano(GLuint text):** Plano texturizado.
- **void skybox(float altura, float largo, float profundidad, GLuint text):** Cubemap del tipo skybox con coordenadas U dividido en 4 y V dividido en 3
- **void torus(GLfloat radioM, GLfloat radiom, int meridianos, int paralelos):** Toroide no texturizado.
- **void arco(GLfloat radioM, GLfloat radiom, int meridianos, int paralelos):** Función para la creación de 1/2 de sección toroide no texturizada.
- **void medio_arco(GLfloat radioM, GLfloat radiom, int meridianos, int paralelos):** Función para la creación de 1/4 de sección toroide no texturizada.
- **void cup(GLfloat radio, int meridianos, int paralelos, GLuint text):** Función para la creación de 1/2 de sección de esfera.
- **void texturaGUI(GLuint text):** Función para la creación de texturas en 2D, pensadas para la GUI sobre la escena 3D.

*Para mayor información referirse al código fuente de la aplicación.

Contacto del programador.

jose_hernandez_castro@outlook.com

+52 55 5053 3845.