DOCUMENTACIÓN PRÁCTICAS 3, 4, 5 INFORMÁTICA GRÁFICA

Alumna: Rocío Barragán Moreno

PRÁCTICA 3

Para realizar esta práctica, decidí hacer un modelo jerárquico de una grúa.
Usé tanto las funciones que se nos proporcionaban en el fichero estructura.h (con la respectiva implementación de las funciones en estructura.c) como objetos creados por mí en la práctica 1.

Las tareas que he realizado han sido las siguientes:

1. **Grafo: claro, bien etiquetado,con transformaciones geométricas adecuadas, parámetros de transformaciones y croquis de submodelos**

Antes de implementar la grúa, hice un grafo a mano explicando las transformaciones geométricas, además de los croquis que iban apareciendo conforme se transformaba la figura. El grafo consiste en lo siguiente:

- Aplicamos un glRotatef que girará la grua en conjunto. Todo este Rotate afectará a toda la grúa en su conjunto.
- Crea la estructura de la grúa.
- Aplico otro glRotatef de manera que afectará solo al resto de piezas creadas a continuación.
- Hago un traslate para colocar el brazo de la grúa.
- Ahora, bajamos de nivel con un glPushMatrix(), de manera que lo implementado a partir de este momento afecte sólo a esa zona de la grúa, y funcione todo como una sola pieza.
    - Creo el brazo de la grúa.
    - Translate para colocar el cubo de la grúa junto al brazo-
    - Creo el cubo.
    - Translate para crear el otro brazo.
    - Creo el otro brazo.
    - Translate para crear la caja debajo del brazo.
    - Creo la caja.

- Subimos de nivel, con glPopMatrix().Queda definido completamente el brazo de la grúa.
- Translate para colocar el gancho
- Volvemos a bajar de nivel, con glPushMatrix(), de forma que los movimientos creados a partir de ahora sólo afectarán al gancho.
    - Hago un translate que hace que aumente y decremente el gancho, y por tanto la cuerda. Es un movimiento que es considerado translate, no rotate.
    - Creo el gancho.
    - Creo la cuerda, con la estructura del cilindro.

- Subimos de nivel con glPopMatrix()

2. **Modelo creado con articulaciones correctas**

Para resolver este apartado, cree una función en el fichero modelo.c llamada dibujaP3 (línea 1084) que implementa lo anterior descrito en el grafo. He usado las estructuras proporcionadas por los ficheros estructura.h y estructura.c, aplicando los respectivos Rotates y Translates, y glPush/PopMatrix() para poder crear una estructura adecuada, y que los movimientos se aplicasen a las zonas requeridas. Los tamaños fueron asignados a medida, para que las piezas salieran proporcionales.

3. **Modificación interactiva de parámetros**

Para resolver este apartado, creé una serie de variables globales que se irán modificando en unas funciones al pulsar determinadas teclas, para así realizar los distintos movimientos de la grúa. Estas variables se pueden encontrar desde la línea 879 hasta la 885.

Las funciones definidas son las siguientes: (desde la línea 887-925)

- Para mover la grúa completa (incluyendo la estructura, los brazos, y el gancho), declaré la variable giro_total. Esta variable será usada en las funciones giroTotalIzquierda() y giroTotalDerecha() según si la grúa gire a la izquierda(incrementa variable) o a la derecha(decrementa variable)
- Para mover tan sólo el brazo con el gancho, declaré la variable giro_brazo. Será usada en las funciones giroBrazoIzquierda() y giroBrazoDerecha() según si quiero que el brazo gire a la izquierda(incrementando la variable) o a la derecha (decrementando la variable).
- Para mover el gancho hacia arriba y hacia abajo usé la variable altura, que iba creando la cuerda según subiese o bajase, dentro de un límite establecido (no puede subir más de la altura de la grúa ni bajar más de la altura del pie de la grúa). Todo esto conseguido con las funciones bajaGancho() y subeGancho().

4. **Animación**

Para la animación, cree una función que activase y desactivase la animación utilizando una variable booleana. Si está activada, en la función idle que se nos proporciona en la práctica (línea 1258) utilicé 3 variables (contadorGrua, contadorBrazo, contadorGancho), que se iban incrementando de manera infinita cada vez que se pulsasen las teclas de sus respectivas funciones (giroTotalIzquierda(), giroBrazoIzquierda(), bajaSubeGancho()).

5. **Modificación interactiva de velocidades de cambio de parámetros**

En este apartado, asocié las variables de la animación a las respectivas funciones(línea 948-972)

- gruaRapida(), incrementa contadorGrua, por lo que la grúa entera se moverá más rápido.
- gruaLenta(), decrementa contadorGrua.

- brazoRapido(), incrementa contadorBrazo, por lo que sólo el brazo se moverá más rápido.
- brazoLento(), decrementa contadorBrazo.
- ganchoRapido(), incrementa contadorGancho, por lo que subirá y bajará más rápido.
- ganchoLento(), decrementará contadorGancho.

PRÁCTICA 4

Las tareas que he realizado en esta práctica son las siguientes:

1. **Representación y visualización de materiales**

Para ello, en el archivo modelo.c, línea 811, declaré una función llamada Materiales, cuyos parámetros son 4 arrays que cada uno contiene los distintos elementos que dan lugar a los parámetros ambiente, especular, difuso y brillo. A raíz de esta función se aplicará el material requerido sobre el objeto de revolución indicado después.

En mi caso, en la línea 1123, creo una función llamada dibujaP4, y en ella se puede ver que declaro los distintos arrays de los 3 materiales que utilizaré (bronce, turquesa y rubí).
En las líneas 1168, 1174 y 1180 uso la función creada antes para cada uno de estos materiales, y ya debajo de las respectivas líneas dibujo un beethoven leído de un archivo ply y represento los distintos materiales en cada uno de ellos.

2. **Aplicar texturas a mallas de revolución**

Para aplicar texturas a la malla de revolución, en la clase de MallaTriangulo (línea 407), creo un booleano de textura, para ver si hay o no textura (línea 429) ya que en los objetos de revolución usaremos el draw de mallaTriangulo (línea 541) y deberemos comprobar si hay textura o no. Si hay textura, significa que el booleano de textura (hayTextura) es true, por lo que la función cargaTexturaRev (línea 777) de la clase MallaRevolucion (línea 643), ha puesto esa variable a true y ha cargado la correspondiente imagen que se le pasa por parámetro.
Volviendo al draw de mallaTriangulo (línea 541), vemos que se activa la correspondiente textura. En cada cara sacamos sus normales, y de cada vértice de la cara se sacarán sus respectivas coordenadas uv, guardadas en el vector de struct uv (declarado en la línea 430).

Este vector uv sirve tanto para guardar la textura de revolución de la lata, como la de la tapa y la de la base. Para ello, he creado 2 booleanos en la clase MallaRevolucion (tapa, base), línea 649, en la que al cargar la textura correspondiente, sólo calculará la textura de aquel booleano inicializado a true. Esto se puede apreciar desde la línea 684 - 736.
Así, el vector uv irá almacenando valores dependiendo de la textura que quiera conseguirse.

Los objetos de esta práctica se encuentran declarados en las líneas 805-807.
Las distintas imágenes cargadas de la lata y tapas se encuentran en initModel(línea 988) y luego las dibujé en la función dibujaP4(línea 1155-1158).

3. **Aplicar texturas al cubo**

En este apartado, utilicé la clase Cubo creada en la práctica 1 (línea 81).
Le añadí una función llamada cargaTextura() que carga la imagen del dado (dado.jpg).
Además, en el draw de la clase cubo, en cada cara antes de dibujar un vértice, hice un glTexCoord con las respectivas coordenadas de ese vértice en cuánto a la imágen cargada.
Finalmente, declaré un objeto dado en la línea 804, lo cargué en initModel(línea 988) y lo dibujé en la función de dibujaP4 en la línea 1162.

4. **Añadir fuentes de luz**

Para este apartado, lo que hice fue crearme dos funciones, Iluminacion1() (línea 845) e Iluminacion2() (línea 860), en las que se modifica una variable booleana que activa y desactiva la luz correspondiente creada.

En la líneas 1139-1142, en dibujaP4, declaré 2 colores de luz(morado y verde) y dos posiciones (pos1 y pos2). Y con glLightfv activé los colores y la posición de cada luz.
La activación y desactivación de estas luces las realicé en entradaTeclado.c.

5. **Composición de la escena**

En este apartado, creé una función llamada dibujaP4, que dibuja un dado con su textura; una lata con su textura tanto en el cuerpo de la lata como base y tapa; 3 objetos ply de la figura de Beethoven, a los que se le aplica un material distinto a cada uno.
Además pulsando las teclas correspondientes se pueden ver las luces creadas junto con su color y posición.

6. **Añadir la tapa y la base a la lata**

Este apartado ya fue explicado en el apartado 2. Apliqué la textura correspondiente a la tapa y base de la lata.

PRÁCTICA 5

Lamentablemente, debido a la carga de asignaturas que tengo este cuatrimestre, no he podido dedicarle suficiente tiempo a esta práctica. He intentado hacer lo máximo posible pero no me ha dado tiempo. Aún así, he hecho una serie de tareas que describiré a continuación.

1. **Añadir vistas de planta, alzado y perfil**

En este apartado, añadí las vistas de planta, alzado y perfil. Para ello, en el fichero entradaTeclado.c, redeclaré las variables rotxCamara y rotyCamara(inicializadas a 0). Se encuentran en la línea 85,86.

En la función especial (línea 252), añadí 3 casos nuevos al switch:

- GLUT_KEY_F1 → pone la vista en planta, modificando rotxCamara y rotyCamara (90, 0).
- GLUT_KEY_F2 → pone la vista en alzado, modificando rotxCamara y rotyCamara (0, -90).
- GLUT_KEY_F3 → pone la vista en planta, modificando rotxCamara y rotyCamara (0, 0).

En los 3 casos, también se definen 2 variables especificas también creadas anteriormente, cx y cz.
cx modifica la cámara para que esta se vaya desplazando a la izquierda tecleando la letra A.
cz modifica la cámara para que esta se vaya alejando.
En las vistas estas 2 variables permanecen asignadas a valores que no cambian para que la cámara se quede fija.

2. **Mover la cámara con las teclas s,w,a, d**

Para realizar este apartado, en entradaTeclado.c, en la función letra (línea 98), añadí 4 casos más:

- w/W: decrementa la variable cz en 0.5, de manera que la cámara se aleja poco a poco
- a/A: incrementa la variable cx en 0.5, de manera que la cámara se mueve a la izquierda poco a poco
- s/S: incrementa la variable cz en 0.5, de manera que la cámara se acerca poco a poco
- d/D: decrementa la variable cx en 0.5, de manera que la cámara se mueve a la derecha poco a poco

Todos estos parámetros declarados, se ajustan en la función setCamara() del archivo visual.c, ahí se les asignan las distintas posiciones de la cámara para que vayan incrementando o decrementando en el eje correspondiente(x o z).

DIBUJAR CADA PRÁCTICA PRESIONANDO UNA NÚMERO

En esta práctica, también he añadido que si pulsas un número correspondiente a una práctica, se dibuja todo lo relativo a esa práctica. Para ello en el archivo modelo.c declaré una variable llamada practica (línea 1005), y respectivas funciones que iban modificando esa variable.

Así si la variable es 3, se dibuja la práctica 3, ya que en el método Dibuja (línea 1203), entra en el switch en el caso 3, y esta llama a la función dibujaP3. Y así con todas las prácticas.

He creado un dibuja por cada práctica, que es llamado en cada caso del switch.

TECLAS USADAS:

PRÁCTICA 1

- p/P → representa las figuras según sus vértices.
- r/R → representa las figuras según sus aristas
- f/F → representa las figuras rellenas
- i/I → activa/desactiva Iluminación

PRÁCTICA 3

- C → gira la grúa a la izquierda
- c → gira la grúa a la derecha
- B→ gira el brazo a la izquierda
- b → gira el brazo a la derecha
- M → baja el gancho
- m → sube el gancho
- x/X → anima la figura
- G → la grúa gira más rápido en animación
- g → la grúa gira más lento en animación
- K → el brazo gira más rápido en animación
- k → el brazo gira más lento en animación
- L → el gancho sube y baja más rápido en animación
- l → el gancho sube y baja más lento en animación

PRÁCTICA 4

- 9 → activa/desactiva primera luz
- 0 → activa/desactiva segunda luz

PRÁCTICA 5

- w/W → aleja la cámara poco a poco
- A/a →Mueve la cámara a la izquierda poco a poco
- s/S → acerca la cámara poco a poco
- d/D → mueve la cámara a la derecha poco a poco
- F1 → planta
- F2 → alzado
- F3 → perfil
-

ADICIONALES A TODAS LAS PRÁCTICAS

- 1 → dibuja práctica 1
- 2 → dibuja práctica 2
- 3 → dibuja práctica 3
- 4 → dibuja práctica 4
- 5 → dibuja práctica 5