Universidad del País Vasco
Euskal Herriko Unibertsitatea

Gráficos por computador

Manual básico

Adrián San Segundo y Unai Salas

Septiembre 30, 2019

1 Introduction

En este manual se tratarán los aspectos técnicos de la aplicación, tal y como se explicaba en el manual de usuario. Se tratarán partes de código que son necesarios para que el usuario experimentado entienda la aplicación.

2 MAIN.C

Cuando ejecutemos la aplicación, desde el main se inicializará la pantalla principal de la aplicación, en la que se mostraran los objetos con los que trabajaremos. Se realizaran inicializaciones de las proporciones de la ventana, proyecciones ortográficas, funciones glut, funciones externas (io.c, transformaciones.c)...
Se inicializarán también las luces, un sol, una bombilla y un foco, pero solo el sol estará activado por defecto.

3 IO.C

Una vez imprimida en la terminal la función help, el usuario puede realizar diferentes acciones dependiendo de las teclas que pulse y el orden en el que lo haga. Todas ellas están reunidas en el switch y explicadas brevemente en el manual de usuario.
Si el usuario está en modo objeto y quiere trasladarlo, rotarlo o escalarlo, hará una llamada a la función *aplicar transformaciones* pasando como parámetro unos valores u otros, dependiendo de la tecla pulsada.
En caso de estar en modo cámara, se debe tener en cuenta si se está maniobrando con el sistema de referencia del mundo (global) o el sistema de referencia del objeto (local).
En caso de ser local se aplican las transformaciones a la cámara, en este caso al tratar con cámaras tendremos disponible la opción de trasladar, rotar y ampliar o disminuir la proyección.
En caso de estar en modo local y realizar una rotación pasando como parámetro una variable x e y que se usaran en la función *modo análisis* para calcular la nueva posición de la cámara. Primero se saca px py y pz restando la m 12 13 y 14 del objeto y de la cámara y se resta al objeto los de la cámara. Para obtener la distancia entre ambos sacamos la suma de las raíces cuadradas. Para la nueva posición multiplicaremos las matrices de m minus, m plus, m rot para luego usar la matriz inversa de la cámara.
Para el modo luz se cuenta con varias funciones. Primero tenemos que tener en cuenta que al inicializar la aplicación se reserva la primera posición de global lights para un sol, la segunda para una bombilla y la tercera para un foco y contaran con unos valores predeterminados de posición, ambiente, difusión e imagen especular. Solo el sol estará encendido al iniciarse.
Para añadir una luz, usaremos la función *añadir luz*, con la que preguntaremos el tipo de elemento que quiere crear, en que posición de teclado del 4 al 8 quiere introducir el nuevo elemento y sus respectivas propiedades. Pueden dejarse

predeterminadas o introducirlas manualmente. Como las posiciones de global light empiezan en 0, restaremos 1 a la posición que el usuario ha pedido.
La aplicación cuenta también con 2 materiales, rubi y obsidiana, las cuales podrá cambiar pulsando la W y haciendo una llamada a la función *cambiar materiales*, que cambiara al que no se este usando.

4 TRANSFORMACIONES.C

Cuando se realiza la llamada a *aplicar transformaciones* desde io.c y se pasa un parámetro, dependiendo del estado en el que se encuentre el usuario en el momento (TRASLACION, ROTACION, ESCALADO) hará una transformación u otra.
Primero se cargara la matriz del objeto con *transformación matrix*, después se aplica la transformación y por último se guarda la matriz en la lista de matrices del objeto con *set transformación matrix* guardando el espacio de puntero y apuntando a la matriz actual.
En caso de estar en modo cámara y modo vuelo cargara la matriz inversa de la cámara seleccionada para operar con ella, y estando en modo análisis trasladaremos solo la m 12, 13 y 14.
En *set transformation matrix* en caso de estar en modo cámara y estar en modo vuelo operaríamos con la matriz inversa de la cámara. En caso de estar en modo análisis solo usaríamos la m 12, 13, 14 para multiplicar la inversa.
En caso de estar en modo luz obtendremos la m obj de la luz seleccionada y operaremos con ella.

5 CAMERA.C

Una vez se ejecute el main y llame a la función *default camera*, tendremos creada la primera camera, la cual se situara en la primera posición de la lista de cámaras y tendrá unos valores predefinidos que habremos cargado antes con *set camera projection*.
La función cambiar cámara cambiara de cámara hasta llegar a la ultima y una vez posicionado en la última, pasar a la primera.
La función *add camera input* se ejecuta desde io.c cuando se quiere meter una cámara nueva. Se le pedirán al usuario la posición de la cámara y el front, y el resto de los valores se introducen automáticamente. La cámara se añadirá a la lista de cámaras.
En la función *centre camera to obj* crearemos una cámara centrada mirando al objeto usando la matriz inversa en 12, 13 y 14 y sustituyendo a la cámara que estaba seleccionada en ese momento.
Por último, tenemos *matriz inversa*. Con la función y teniendo la inversa obtenemos la que devuelve el *look at*.

6 DISPLAY.C

Para obtener los vectores normales del objeto, usamos la función *normal vector*, la cual hará una llamada en cada cara del objeto a la función *calculate surface normal* para obtener así la normal de cada cara del objeto.
También se ha implementado en la función *display* las proyecciones de la cámara y los valores para sus atributos, en la perspectiva aplicando *glFrtustum* y en la ortográfica *glOrtho*.