Práctica 5: Interacción

Objetivos

- Saber desarrollar aplicaciones gráficas interactivas sencillas, gestionando los eventos de entrada de teclado y ratón usando glut.
- Saber implementar operaciones de selección de objetos de la escena.
- Saber controlar interactivamente la cámara usando el ratón.

Funcionalidad a desarrollar

1. Añadir vistas con proyección en planta, alzado y perfil.
2. Mover la cámara usando el ratón.
3. Seleccionar objetos de la escena con el ratón.
4. Añadir un menú de acciones que permita cambiar el material de los objetos seleccionados.

Desarrollo

Añadir vistas con proyección en planta, alzado y perfil

Se incluirá en la escena un mecanismo para cambiar la proyección ofreciendo las vistas clásicas de frente, alzado y perfil de la escena. Al menos una de ellas deberá tener proyección ortogonal y al menos otra proyección en perspectiva. Se activarán con las teclas F1, F2 y F3.

Mover la cámara usando el ratón y el teclado en modo primera persona

Con las teclas A,S,D y W se moverá la cámara activa por la escena (W: avanzar, S: retroceder, A: desplazamiento a la izquierda, D: desplazamiento a la derecha, R: reiniciar posición), conservando la dirección en la que se está mirando. Para ello será necesario modificar únicamente el punto VRP o eye (esto equivale a modificar la traslación al sistema

de coordenadas de la cámara). Para realizar este desplazamiento es necesario calcular la dirección en la que está mirando la cámara.

Por otro lado, girar la cámara a derecha o izquierda, arriba o abajo se realizará siguiendo los movimientos del ratón con el botón central pulsado. Esto implica modificar el VPN o el lookAt o el giro de la transformación de visualización.

Para controlar la cámara con el ratón es necesario hacer que los cambios de posición del ratón afecten a la posición de la cámara, y en glut eso se hace indicando las funciones que queremos que procesen los eventos de ratón. En el programa principal del código de partida aparece:

```
glutMouseFunc( clickRaton );
glutMotionFunc( ratonMovido );
```

Estos callback están declarados en *mouse.c*.

La función clickRaton será llamada cuando se actúe sobre algún botón del ratón. La función ratonMovido cuando se mueva el ratón manteniendo pulsado algún botón.

El cambio de orientación de la cámara se gestionará en cada llamada a ratonMovido, que solo recibe la posición del cursor, por tanto debemos almacenar el estado de los botones del ratón cada vez que se llama a clickRaton, para que solo se mueva la cámara cuando este pulsado el botón central. La información de los botones se recibe de glut cuando se llama al callback:

```
void clickRaton( int boton, int estado, int x, int y );
```

por tanto bastará con analizar los valores de boton y estado, y almacenar información que nos permita saber si el botón central está pulsado y la posición en la que se encontraba el cursor cuando se pulsó

```
if ( boton == GLUT_MIDDLE_BUTTON )
{
if ( estado == GLUT_DOWN ){
// Se entra en el estado "moviendo camara"
  }
else{
// Se sale del estado "moviendo camara"
  }
}
```

En la función ratonMovido comprobaremos si el botón central está pulsado, en cuyo caso actualizaremos la posición de la cámara a partir del desplazamiento del cursor

```
void ratonMovido( int x, int y )
{
......
if (MOVIENDO_CAMARA) {
//Girar la camara usando segun el vector (x-xant,y-yant);
  xant=x;
  yant=y;
```

```
  }
...
glutPostRedisplay();
```

Para girar la cámara se debe recalcular el valor de sus parámetros (VPN o lookAt) en función del incremento de x e y recibido. Si hasta ahora en la práctica la transformación de visualización se hacía, por ejemplo, en

```
void transformacionVisualizacion ()
{
  glTranslatef (0, 0, -D);

  glRotatef (view_rotx, 1.0, 0.0, 0.0);
  glRotatef (view_roty, 0.0, 1.0, 0.0);

  // glTranslatef(-x_camara,-y_camara,-z_camara);
}
```

ahora habrá que hacer cambiar *view_rotx* y *view_roty*.

Seleccionar objetos de la escena usando el ratón

Se incluirán en la escena los objetos generados en las prácticas 2, 3 y 4, y cualquier otro objeto que se desee. Para realizar la selección usaremos un código de color para cada objeto seleccionable. Se trata de crear una función de dibujado distinta para cuando queremos seleccionar, y cuando el usuario hace clic, se pinta la escena "para seleccionar" en el buffer trasero y se lee el color del pixel donde el usuario ha hecho click. Si no se hace un intercambio de buffers, el usuario no verá esa imagen con falso color.

Para no duplicar el método de dibujo, y evitar inconsistencia a la hora de seleccionar, crearemos una función de dibujo de la escena, que contendrá todo el código del callBack *Dibuja*, salvo la llamada a *glutPostRedisplay*. Cambiaremos el código de *Dibuja* para llamar a esta función

```
void Dibuja()
{
  dibujoEscena();
  glutSwapBuffers();
}
```

Crearemos una función para codificar los identificadores de los objetos como colores (ten en cuenta que tendrás que almacenar los identificadores de los objetos). Por ejemplo, si tienes objetos articulados (no mas de 255) y quieres seleccionar sus componentes:

```
void ColorSeleccion( int i, int componente)
{
  unsigned char r = (i & 0xFF);
  unsigned char g = (componente & 0xFF);
```

```
   glColor3ub(r,g,0);
}
```

Es necesario que no se modifique el color asignado al pixel, por eso damos los colores como *unsignedbyte* con *glColor3ub*, es decir, como enteros de 0 a 255.

Además tendremos que desactivar el *GL_DITHER*, *GL_LIGHTING* y *GL_TEXTURE*.

En *dibujoEscena* asigna a cada objeto su material y su color de selección. Puedes controlar si se usa uno u otro para dibujar activando o desactivando el cálculo de iluminación (asumiendo que no lo cambias durante el dibujo). Otra opción es usar un parámetro en *dibujaEscena* para controlar el uso de uno u otro.

El proceso de selección se inicia cuando se pulsa el botón izquierdo del ratón, ejecutará las siguientes acciones:

- Desctivar el cálulo de iluminación.
- Llamar a la función *dibujoEscena*.
- Leer el color del pixel (x,y).
- Decodificar el identificador
- Lanzar un evento de redibujado de la escena.

Para leer el color del pixel podemos usar la función *glReadPixels*:

```
void glReadPixels(GLint x, GLint y, GLsizei width, GLsizei height,
                  GLenum format, GLenum type, GLvoid *pixels);
```

donde

x, y es la esquina inferior izquierda del cuadrado a leer (en nuestro caso el x,y, del pick).

$width, height$ son el ancho y alto del área a leer (1,1 en nuestro caso).

$format$ es el tipo de dato a leer (coincide con el format del buffer, *GL_RGB* o *GL_RGBA*).

$type$ es el tipo de dato almacenado en cada pixel, según hayamos definido el glColor (p.ej. *GL_UNSIGNED_BYTE* de 0 a 255, o *GL_FLOAT* de 0.0 a 1.0).

pixels es el array donde guardaremos los pixels que leamos. Es el resultado de la función.

Por ejemplo:

```
unsigned char data[4];
...
glReadPixels(x,viewport[3]-y,1,1,GL_RGBA,GL_UNSIGNED_BYTE,data);
```

El proceso de decodificación dependerá de como se hayan codificado los identificadores, teniendo en cuenta que el último parámetro recibe el color del pixel. Por ejemplo, en el caso descrito anteriormente:

```
*i=data[0];
*componente=data[1];
```

En resumen, el proceso de selección puede realizarse en un función del tipo:

```
int pick(int x, int y,int * i)
{

GLint viewport[4];
unsigned char data[4];

  glGetIntegerv (GL_VIEWPORT, viewport);
  glDisable(GL_DITHER);
  glDisable(GL_LIGHTING);
  dibujoEscena();
  glEnable(GL_LIGHTING);
  glEnable(GL_DITHER);
  glFlush();
  glFinish();

  glReadPixels(x, viewport[3]-y,1,1,GL_RGBA,GL_UNSIGNED_BYTE, data);

  *i=data[0];

  glutPostRedisplay();
  return *i;
}
```

Añadir menú de acciones que permitan cambiar el objeto seleccionado

Se incluirá un menú de glut que permitirá al menos asignar el color, de entre varios predefinidos, al objeto seleccionado.

Opcionalmente se podrá usar el menú para para transformar un objeto o para actuar sobre la articulación asociada a un componente de un objeto articulado. En este caso se puede además hacer que la transformación o el parámetro de la articulación se modifique con el ratón.

Evaluación

- Añadir vistas de planta, alzado y perfil (2 puntos)
- Mover la cámara y rotarla con el ratón (3 puntos)
- Selección de objetos (3 puntos)
- Cambio de material al seleccionar (2 puntos)
- Transformación de objetos y cambio de parámetros de articulaciones (4 puntos)

Temporización

Esta práctica se debe realizará en tres sesiones de prácticas:

Grupo C1 Jueves 01/12/22, 15/12/22 y 22/12/22

Grupo C2 Miércoles 07/12/22, 14/12/22 y 21/12/22