

Complemento práctica 5

Con respecto al guión de la práctica 5 se hacen las siguientes cambios/matizaciones para poder realizarla en el periodo establecido.

Los objetivos que se van a evaluar son los siguientes:

- Movimiento de la cámara mediante el ratón
- Selección usando OpenGL (según hemos visto en teoría)
- Crear una cámara que gire alrededor del objeto seleccionado
- Poder cambiar de tipo de proyección (de perspectiva a paralela y viceversa)
- El movimiento de la rueda del ratón servirá para acercarse o alejarse en ambos tipos de proyecciones.

El resto de ítems que vienen en la práctica se dejan como opcionales, aunque se deberá entender cómo se hacen.

¡Aclaración!

Con respecto al código que se muestra en los apuntes:

```
void pick(unsigned int x, unsigned int y, unsigned int Width, unsigned int Height)
{
    GLuint Selection_buffer[BUFFER_SIZE];
    GLint Hits, Viewport[4];

    glGetIntegerv (GL_VIEWPORT, Viewport);
    glSelectBuffer (BUFFER_SIZE, Selection_buffer);

    glRenderMode (GL_SELECT);
    glInitNames();
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity ();
    gluPickMatrix ( x, Viewport[3] - y, Width, Height, Viewport);
    glFrustum(Min_x,Max_y,Min_y,Max_y,Front_plane,Back_plane);
    draw ();
    Hits = glRenderMode (GL_RENDER);
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity();
    glFrustum(Min_x,Max_y,Min_y,Max_y,Front_plane,Back_plane);
    ...
    Obtener informacion
}
```

hay que tener en cuenta los siguientes detalles:

- 1) Para que funcione correctamente hay que meter un primer elemento en la pila (este es un problema del SELECT).
- 2) Después de poner la matriz que se encarga de ajustar el espacio de selección con el `gluPickMatrix`, hay que poner la matriz de proyección que se usa sin pick.
- 3) Y también la transformación de modelado y vista para que aplique las modificaciones al objeto

Los puntos 2 y 3 son los que van incluidos en el `Draw()`, pero dado puede haber dudas, se aclara con el presente documento.

El código final debe quedar de la siguiente manera:

```

void pick(unsigned int x, unsigned int y, unsigned int Width, unsigned int Height)
{
    GLuint Selection_buffer[BUFFER_SIZE];
    GLint Hits, Viewport[4];

    glGetIntegerv (GL_VIEWPORT, Viewport);
    glSelectBuffer (BUFFER_SIZE, Selection_buffer);

    glRenderMode (GL_SELECT);
    glInitNames();
    glPushName(0);
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity ();
    gluPickMatrix ( x, Viewport[3] - y, Width, Height, Viewport);
    glFrustum(Min_x,Max_y,Min_y,Max_y,Front_plane,Back_plane);
    Draw(){
        Limpiar_ventana();
        Cambiar_proyeccion(); // PROJECTION
        Cambiar_observador(); // MODELVIEW
        Dibujar_escena();
    }
    Hits = glRenderMode (GL_RENDER);
    if (Hits>0){
        // Obtener información de la selección y marcar el objeto seleccionado
    }
    // volver a dibujar la escena pero teniendo en cuenta si hay algún objeto seleccionado
    Draw(){
        Limpiar_ventana();
        Cambiar_proyeccion(); // PROJECTION
        Cambiar_observador(); // MODELVIEW
        Dibujar_escena();
    }
}

```

Tener en cuenta que la inversión de la Y se hace en el propio pick. También que el ancho y alto del pick no deben ser muy grandes para que la selección no sea muy genérica. Valores entre 1 y 3 son suficientes.