

## Práctica 1

### Parte 2

- Añade a la `clase Camera` los métodos para desplazar la cámara en cada uno de sus ejes, sin cambiar la dirección de vista.

```
void moveLR(GLdouble cs); // Left / Right
void moveFB(GLdouble cs); // Forward / Backward
void moveUD(GLdouble cs); // Up / Down
```

- Añade a la `clase Camera` el método para rotar la dirección de vista de la cámara sin modificar su posición.

```
void rotatePY(GLdouble incrPitch, GLdouble incrYaw);
```

- Será necesario modificar los métodos:

```
void pitch(GLdouble a);
void yaw(GLdouble a);
void roll(GLdouble a);
```

- Añade a `main` las respuestas a las teclas `WASD`. W y S desplazan la cámara en la dirección de vista. A y D desplazan la cámara a derecha e izquierda.
- Añade a `main` una variable global `dvec2 mCoord` para guardar las coordenadas del ratón, y los callbacks para los eventos del ratón:

```
glutMouseFunc(mouse);
glutMotionFunc(motion);
```

void `mouse`(int button, int state, int x, int y) captura, en `mCoord`, las coordenadas del ratón invirtiendo el eje Y

void `motion`(int x, int y) captura las coordenadas del ratón y obtiene el desplazamiento con respecto a las anteriores coordenadas, rota la dirección de vista de la cámara en función del desplazamiento y renderiza la escena.

- Añade a la `clase Camera` un atributo `bool orto`; y un método `setPrj()` para cambiar de proyección ortogonal a perspectiva.

Define la tecla `'p'` para cambiar entre proyección ortogonal y perspectiva