
Resultat de l'objecte seleccionat

Escriu un **effect plugin** que resalti d'alguna manera l'objecte seleccionat, si n'hi ha cap (per exemple, pintant la seva capsula englobant en filferros). Caldrà que ho feu re-implementant el mètode `postFrame()`.

L'índex de l'objecte seleccionat es pot obtenir amb el mètode `selectedObject()` de la classe `Scene` (vigileu que un valor -1 indica que no hi ha cap objecte seleccionat):

```
int seleccionat = scene()->selectedObject();
```

Per defecte, l'objecte seleccionat és el primer (índex 0), si s'ha carregat algun objecte.

Selecció d'objectes per teclat

Escriu un **action plugin** per tal que quan l'usuari premi una tecla 0..9, es seleccioni l'objecte corresponent de l'escena (si existeix). Per exemple, amb '0' es seleccionarà el primer objecte del vector d'objectes de l'escena.

Per tal d'aconseguir això, haureu d'implementar el mètode `keyPressEvent()`. Aquest mètode haurà d'establir l'objecte seleccionat amb el mètode `setSelectedObject` de la classe `Scene`, si aquest existeix, i després cridarà a `updateGL` per tal que es repinti l'escena.

Si l'usuari prem una tecla 0..9 que no correspon a cap objecte (per exemple, '5' quan l'escena només té 3 objectes), cal posar l'objecte seleccionat a -1.

Per tal de provar el funcionament correcte d'aquest plugin, cal combinar-lo amb el plugin de l'exercici anterior que indiqui d'alguna manera l'objecte seleccionat.

Nota: podeu afegir més objectes a la vostra escena amb la tecla 'L'.

Selecció d'objectes amb el mouse

Escriu un **action plugin** per tal que quan l'usuari faci clic amb el mouse (per exemple, LMB + Ctrl), es seleccioni l'objecte visible més proper a l'observador que estigui sota el cursor (si n'hi ha cap, és clar).

El mètode **onPluginLoad** carregarà un VS i FS minimalistes. El VS simplement escriurà `gl_Position` com és habitual. FS simplement escriurà com a color el valor d'una variable uniform que rebrà de l'aplicació,

```
uniform vec4 color;
```

El mètode `mousePressEvent()` implantarà la selecció pròpiament dita. Aquest mètode haurà de:

- (a) Comprovar que efectivament s'ha fet click amb el botó adient i els modificadors (Shift, Control...) adients. Per exemple:

```
if (! (e->button() & Qt::RightButton)) return;  
if ( e->modifiers() & (Qt::ShiftModifier)) return;  
if (! (e->modifiers() & Qt::ControlModifier)) return;
```

- (b) Esborrar els buffers amb un color de fons únic (ex. blanc).
- (c) Activar (bind) el shader program amb el VS+FS d'abans.
- (d) Enviar els uniforms que facin servir els vostres shaders.
- (e) Pintar l'escena assegurant-se que cada objecte es pinta amb un color únic que permeti identificar l'objecte (i diferent del color de fons); això ho fareu definint el uniform `vec4` color que farà servir el FS:

```
// per cada objecte  
for (unsigned int i=0; i<scene()->objects().size(); ++i)  
{  
    GLubyte color[4];  
    encodeID(i,color); // trieu la conversió que vulgueu  
    program->setUniformValue("color", QVector4D(color[0]/255.0,  
color[1]/255., color[2]/255., 1.0));  
    drawPlugin()->drawObject(i);  
}
```

- (f) Llegir el color del buffer de color sota la posició del cursor,

```
int x = e->x();  
int y = glWidget()->height()-e->y();  
GLubyte read[4];  
glReadPixels(x, y, 1, 1, GL_RGBA, GL_UNSIGNED_BYTE, read);
```

- (g) Obtenir l'identificador de l'objecte corresponent `i`, si no és color de fons, establir l'objecte seleccionat amb el mètode `setSelectedObject` de la classe `Scene`,
- (h) Cridar a `updateGL` per tal que es repinti l'escena.

Si l'usuari fa Ctrl-clic on no hi ha cap objecte, cal posar l'objecte seleccionat a -1.

Per tal de provar el funcionament correcte d'aquest plugin, cal combinar-lo amb el plugin de l'exercici anterior que indiqui d'alguna manera l'objecte seleccionat.

Podeu suposar que l'escena tindrà com a màxim 255 objectes. Nota: podeu afegir més objectes a la vostra escena amb la tecla 'L'.