
Resaltat de l'objecte seleccionat

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

Per pintar la caixa englobant d'un objecte, podeu obtenir-la amb el mètode **boundingBox()**, i pintar-la amb el mètode **render()** de la classe **Box**.

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

Selecció d'objectes per teclat

Escriu un **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 de 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 **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).

Per tal d'aconseguir això, haureu d'implementar el mètode **mouseReleaseEvent()**. Aquest mètode haurà de:

- (a) 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); podeu fer servir **glColor3ub()** amb valors entre 0 i 255. Abans de pintar l'escena, heu d'esborrar els buffers amb **glClear**, usant un color apropiat (**glClearColor**) que no es pugui confondre amb cap identificador d'objecte. Podeu mirar l'exemple que teniu al mètode **RenderDefault::paintGL()** del plugin **render-default**. Donat que cada objecte s'ha de pintar d'un color diferent, haureu d'usar la vostra pròpia funció per dibuixar l'escena (quelcom similar al que fa el **DrawImmediate::drawScene** del plugin **draw-immediate**, però enviant colors únics per cada objecte).
- (b) Llegir el color del buffer de color sota la posició del cursor. Recordeu que Qt reporta les coordenades del mouse (**QMouseEvent::y()**) respecte la cantonada **superior** esquerra, mentre que OpenGL espera coordenades de finestra respecte la cantonada **inferior** esquerra. La lectura del buffer de color la podeu fer amb **glReadPixels**.
- (c) 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**,
- (d) 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.

La implementació de la selecció haurà d'estar basada en la lectura del buffer de color (no feu servir el mode **GL_SELECT** d'OpenGL).

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