

## Exercici lliurable 2 de laboratori d'INDI 2024–2025 Q2

### Instruccions

1. Aquests exercicis són individuals, així que només pots lliurar **codi que hakis generat tu**. No pots fer servir codi que altres estudiants hagin compartit amb tu ni que tu hakis compartit amb d'altres estudiants. Altrament es considerarà còpia.
2. Partiràs del codi que tens a Exercici2.tar (el podeu trobar en el Campus digital). Has de descomprimir aquest arxiu en un directori. Es creara un subdirectori Exercici2 on tindràs tots els fitxers amb els que has de treballar.
3. La solució que lliuris ha de compilar i executar correctament al laboratori. Si no compila o dóna error d'execució, l'avaluació de l'exercici serà un 0, sense excepció.
4. Els exercicis que es demanen només requereixen canvis a la classe *MyGLWidget*. **No has de modificar cap altre fitxer dels que se't proporcionen, ni tampoc canviar el seu nom**. A més, els mètodes *carregaShaders*, *creaBuffersTerra*, *creaBuffersModels* i *creaBuffersCub* de la classe *BL2GLWidget* no els pots modificar.
5. Per fer el lliurament has de generar un arxiu TGZ que inclogui tot el codi del teu exercici i que es digui `Ex2_Cognom_Nom.tgz`, on substituiràs `<Cognom>` i `<Nom>` pel teu nom i cognom. Per exemple, l'estudiant Josep Carreras (des d'una terminal en la que s'ha col·locat dins del directori exercici2) farà:

```
make distclean  
tar zcvf Ex2_Carreras_Josep.tgz *
```

és important el 'make distclean' per a esborrar els arxius binaris del directori, que el nom i cognom sigui correcte (el teu), i que hi hagi el sufix .tgz.

6. Un cop fet això, al teu directori tindràs l'arxiu `Ex2_Cognom_Nom.tgz` que és el que has de lliurar al campus virtual **abans del diumenge dia 11 de maig a les 23:55**.

## Enunciat

El codi que proporcionem pinta una escena amb un terra de 12x8 centrat a l'origen, un model de Rick Sánchez d'alçada 2, un model d'una videocàmera i un model d'un cub amb els centres de les seves tapes inferiors a  $(0,0,-3)$ ,  $(0,0,0)$  i  $(0,0,3)$  respectivament. Es donen ja implementats els mètodes que construeixen els VAOs i VBOs dels tres models (*creaBuffersTerra()*, *creaBuffersCub()* i *creaBuffersModels()*). També es dona el mètode *calculaCapsaModel()* que, donat un model i l'alçada que es vol que tingui a l'escena, calcula el factor d'escala i el centre de la base de la seva caixa contenidora. La càmera que es dona a l'esquelet es una càmera completament arbitrària que permet veure l'escena inicialment, es pot veure una imatge de com queda l'escena inicialment a la figura 1.

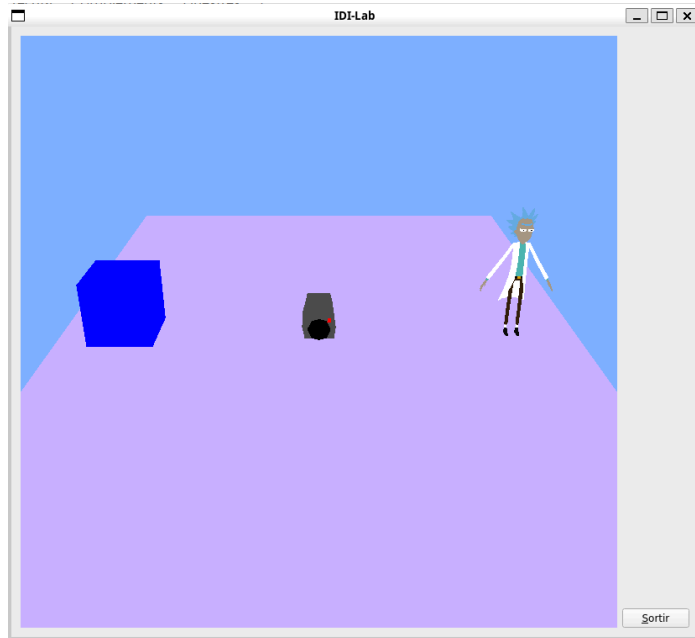


figura 1

**IMPORTANT: Analitzeu el codi donat abans d'implementar funcionalitats.**

Es demana que, donat l'esquelet inicial que us hem proporcionat, resolguis els següents exercicis:

1. Modifica l'escena per a què hi hagi inicialment 2 portes sobre l'eix z representades per dos paral·lelepípedes de mida  $0.5 \times 2 \times 3$  i amb el centre de la tapa inferior situats respectivament a  $(0,0,2.5)$  i  $(0,0,-2.5)$ . El centre de la tapa inferior de la caixa englobant del Rick ha de trobar-se inicialment a la posició  $(-5,0,0)$ . El Rick ha d'estar orientat cap a +X i tenir alçada 1.5. El centre de la tapa inferior de la videocamera s'ha de trobar a  $(0,2.25,-1)$  i ha de tenir alçada 0.5.

La imatge inicial de l'aplicació ha de ser la que mostra la imatge de la figura 2.

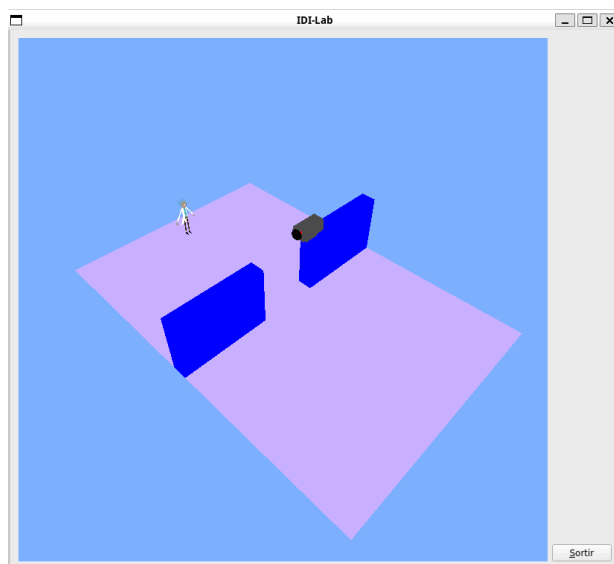


figura 2

2. L'escena s'ha de poder visualitzar amb una càmera en tercera persona que permeti inicialment veure l'escena centrada, sencera, sense deformar i ocupant el màxim del viewport (essent el viewport tota la finestra gràfica). La càmera ha de tenir una òptica perspectiva. En cas de redimensionament de la finestra (resize) l'escena no s'ha de deformar ni retallar.

Aquesta camera també ha de permetre la inspecció mitjançant rotacions dels angles d'Euler (angles  $\psi$  i  $\theta$ ), és a dir, l'usuari ha de poder modificar aquests angles utilitzant el ratolí com s'ha fet al laboratori. La càmera inicial ha de tenir angles  $\psi = M\_PI/4.0$  i  $\theta = M\_PI/4.0$ . El moviment d'inspecció que es demana ha de fer que en moure l'usuari el ratolí cap a la dreta del viewport, l'escena es mogui també cap a la dreta (això implica que la càmera es mogui cap a l'esquerra) i el mateix anant cap amunt. Fixa't que amb aquest moviment d'inspecció sembla que l'usuari arrossegui l'escena en moure el ratolí.

3. Implementa el moviment del Rick de manera que:

- Mitjançant la tecla *Left* (*Qt::Key\_Up*) el Rick es desplaçarà 1 unitat cap a X positiu.
- Mitjançant la tecla *Right* (*Qt::Key\_Down*) el Rick es desplaçarà 1 unitat cap a X negatiu.

El Rick sempre ha de mirar cap a la direcció de desplaçament així que caldrà aplicar-li un gir de 180 graus sempre que calgui. A més, no sobrepassarà mai el límit marcat pel terra.

4. Implementa una segona càmera que s'activi i es desactivi mitjançant la tecla C (*Qt::Key\_C*). Aquesta nova càmera estarà situada a la posició (0, 2.25, -1), tindrà òptica perspectiva amb un angle d'obertura  $M\_PI/2$ , *zFar* i *zNear* suficients per a què no es retalli l'escena i sempre estarà mirant cap a la posició del Rick. Tampoc s'ha de deformar la imatge quan es faci un redimensionament de la finestra gràfica.

El resultat d'activar aquesta càmera es pot veure a la figura 3. Quan estigui activada aquesta càmera les interaccions amb el ratolí no han de modificar els angles d'Euler de l'altra càmera. A més has d'afegir un element d'interfície que permeti canviar entre les dues càmeres. Quan s'activi aquesta càmera, deshabilita el render de la videocàmera per tenir una imatge més adequada.

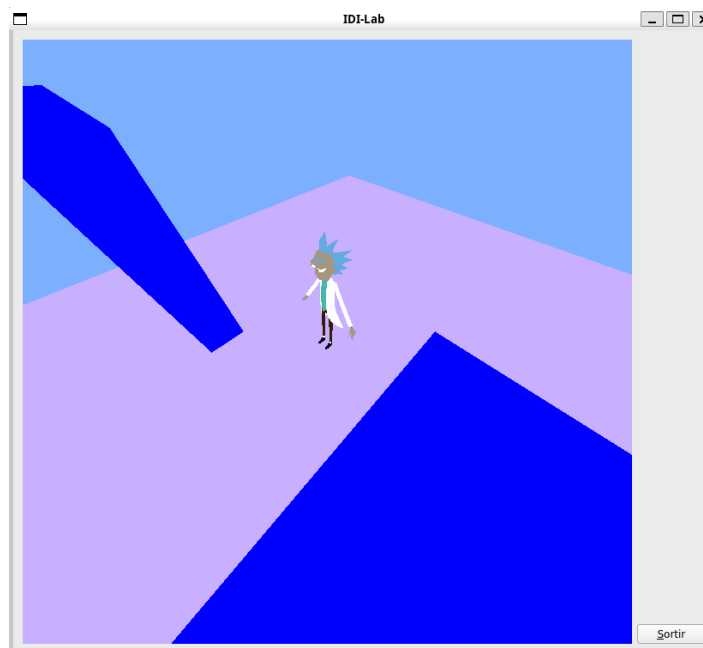


figura 3

5. Implementa el reset de l'escena. Mitjançant la tecla R (*Qt::Key\_R*) l'escena ha de tornar al seu estat inicial (el que es pot veure a la figura 2). A més has d'afegir un element d'interfície que faci la mateixa acció que aquesta tecla.

6. Afegir una tercera porta que es trobi enmig de les dues anteriors i que sigui de color vermell. Aquesta porta es genera a partir del cub inicial que teniu amb l'esquelet i, per aconseguir que sigui vermella, caldrà que envieu una variable uniform al vertex shader. EL centre de la tapa inferior de la porta es trobarà inicialment a la posició (0,0,0) i tindrà mides 0.25x2x2, com es mostra a la figura 4.

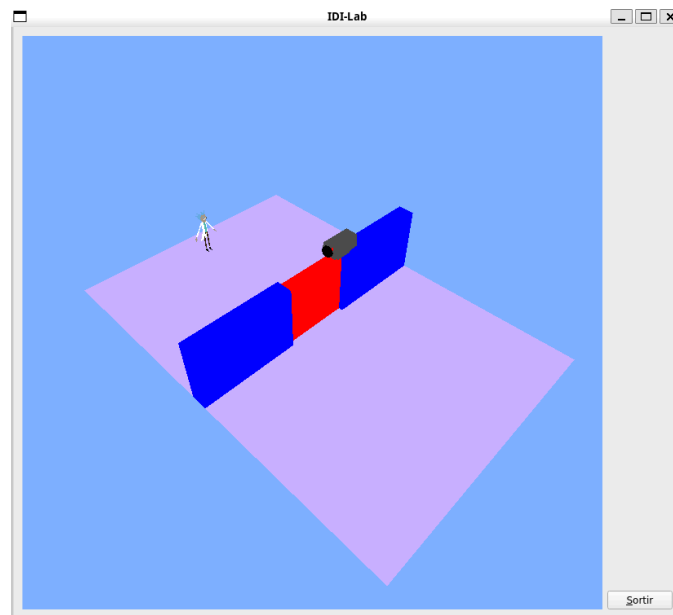


figura 4

Quan el Rick s'apropa a la porta (es troba a la posició 2 o -2 sobre l'eix X), llavors, la porta s'obrirà i es col·locarà a la posició (0,0,2) (de fet, es teletransporta) permetent el pas del Rick. Quan el Rick superi aquesta posició, la porta es tancarà tornant a la seva posició inicial.

7. (Exercici extra). Implementeu el moviment automàtic del Rick fent servir l'event *timeout* del QTimer que teniu a l'esquelet. Aquest event s'activarà i desactivarà amb la tecla A i produirà que el Rick es desplaci sobre l'eix X del món en increments de 0.25 cap a X positiu i X negatiu alternativament (el Rick ha de canviar de direcció quan arriba al límit del terra i ha de mirar sempre en la direcció de moviment). El funcionament de la videocàmera i de la porta vermella no s'han de veure alterats pel moviment automàtic del Rick. Quan el moviment sigui automàtic, les tecles Key\_Up i Key\_Down no tindran efecte. És important, a més, que, quan es desactivi el moviment automàtic, les tecles tornin a tenir efecte i es mantingui la direcció del Rick. Afegiu un element d'interfície que permeti activar i desactivar el moviment automàtic.

8. (Exercici extra 2). Implementeu el moviment de la porta vermella fent que llisqui des de la posició inicial a la posició final, tant per obrir-se com per tancar-se en comptes de què es teletransporti.

**IMPORTANT: Les accions que es realitzin amb el teclat han d'estar coordinades amb l'element d'interfície corresponent.**

A l'Atenea teniu un executable de la solució demanada.