

Laboratori: navegació i càmeres

Partint de les explicacions i el codi vist a classe de laboratori, en grups de dos, implementarem una sèrie de funcionalitats per tal de navegar per un món virtual 3D.

1. Codi base

Fixa't en el codi de l'habitació aportat i repassa el funcionament de la funció **gluPerspective** fent una ullada a la seva especificació i jugant una mica amb els seus paràmetres.

```
void gluPerspective(GLdouble fovy, GLdouble aspect, GLdouble zNear, GLdouble zFar);
```

Tot seguit esbrina per què l'escena s'està veient des d'un punt de vista fix. Modifica'l al teu gust per tal de corroborar-ho.

2. Càmeres fixes

Repasa la funció **gluLookAt** i utilitza-la per tal de definir quatre punts de vista diferents accessibles, en temps real, mitjançant les tecles '2', '3', '4' i '5'.

```
void gluLookAt(GLdouble eyex, GLdouble eyey, GLdouble eyez,  
               GLdouble centerx, GLdouble centery, GLdouble centerz,  
               GLdouble upx, GLdouble upy, GLdouble upz);
```

3. Entitat i moviment

Afegeix un element al nostre escenari que podrem moure pel nostre escenari. Per a representar-lo, farem servir una esfera utilitzant la funció **gluSphere** i, la seva posició (x,z), serà possible modificar-la a partir de les tecles 'w', 'a', 's' i 'd'.

```
void gluSphere(GLUquadric *qobj, GLdouble radius, GLint slices, GLint stacks);
```

```
GLUquadricObj *q = gluNewQuadric();  
gluSphere(q, 1, 16, 16);  
gluDeleteQuadric(q);
```

4. Càmera en primera persona

Per tal de posar-nos en la pell d'aquesta entitat que acabem d'afegir, farem possible la navegació amb els seus propis ulls. Amb aquest propòsit, farem que l'entitat sigui visible sempre i quan la navegació no sigui en primera persona. La navegació amb vista subjectiva serà accessible amb la tecla '1' i estarà restringida als eixos X i Z tal qual teníem en el punt anterior.

5. Direcció de visió

Afegirem la possibilitat de rotar la vista en el nostre sistema de navegació. Per tal de fer això, per una banda, mourem el codi que teníem abans per a les tecles 'a' i 'd' a les tecles 'q' i 'e' (moviment horitzontal amb direcció de visió fixa). Per una altra banda, ara les tecles 'a' i 'd' implementaran la rotació respecte l'eix Y.

Seguint la Figura 1, les tecles 'q' i 'r' provocaran el moviment LEFT-RIGHT i 'a' i 'd' modificaran el valor α . Les tecles 'w' i 's' implementaran el moviment FORWARD-BACKWARD.

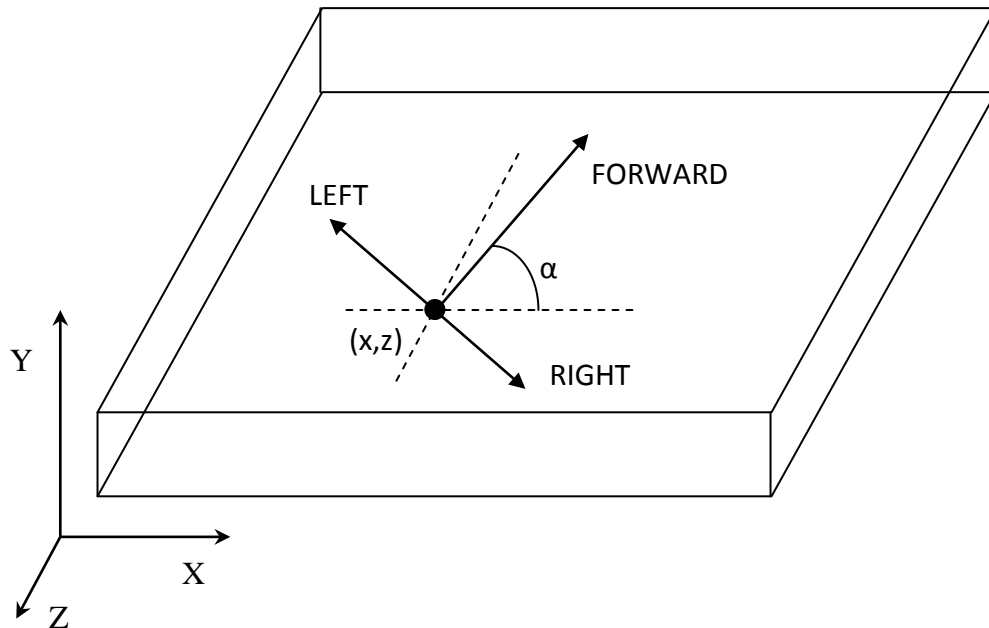


Figura 1. Donat un punt i un angle de gir, així es com quedarien definits els eixos de translació.

6. Teclat i ratolí

Farem un últim afegit al nostre sistema de navegació enriquint-lo amb la possibilitat d'interacció a través del ratolí. El moviment en l'eix X del ratolí farà possible incrementar i decrementar el valor del paràmetre α vist amb anterioritat.

Tingues en compte les següents qüestions:

- Com afecta aquest tipus d'interacció segons l'aplicació sigui a pantalla completa (*fullscreen*) o amb una finestra (*windowed*)?
- Quina estratègia es pot seguir per controlar el moviment del ratolí?
- Llegeix l'especificació de la funció **SetCursorPos** de la API de Windows i estudia la seva aplicació:

```
#include <windows.h>
BOOL WINAPI SetCursorPos(int X,int Y);
```

- Per a què podríem fer servir el moviment en l'eix Y del ratolí?
- Com implementaries el sistema de navegació d'una nau?
- Ets capaç d'implementar un sistema de navegació lliure amb els 6 graus de llibertat?