

© Professors d'IDI – Curs 2013-2014

Bloc_2: Transformacions Geomètriques i Models

Sessions i Objectius

- Sessió 1 –seccions 1 a 3- : Transformacions Geomètriques
 - Objectes glut
 - Entendre el funcionament de les transformacions geomètriques per: posicionar i animar objectes.
 - Utilització en OpenGL.
 - Exercici.
- Sessió 2 –seccions 4 i 5-:
 - Carregar models geomètrics (OBJ) i visualitzar en OpenGL.
 - Aplicació resum de conceptes: crear una escena concreta, poder girar l'escena, poder moure un dels objectes.

IDI- Q2 2013-2014

Primer exemple vist

```

...
void refresh(void) {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glBegin(GL_TRIANGLES);
        glColor(1,0,0);
        glVertex3f(-0.5,0.0,0.0);
        glVertex3f(0.5,0.0,0.0);
        glVertex3f(0.0,0.5,0.0);
    glEnd();
    glutSwapBuffers(); }
int main(int argc, const char *argv[]) {
    glutInit(&argc, ((char **) argv);
    glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGB);
    glutInitWindowSize(600,600);
    glutCreateWindow("IDI: Practiques OpenGL");
    glClearColor(0.5,0.5,0.5) => initGL
    glutDisplayFunc(refresh);
    ...
    glutMainLoop();
    return 0; }

```

1. Inicialitzacions glut

2. Registre *callbacks*

3. Bucle
processament
events

IDI- Q2 2013-2014

Repàs callbacks

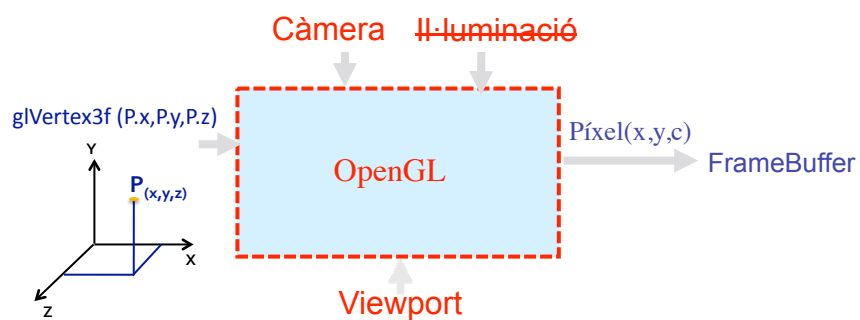
- void glutDisplayFunc (void (*funcName)void);
- void glutReshapeFunc (void (*func)(int width, int height));
- void glutKeyboardFunc (void (*func) (unsigned char key,
int x, int y));
- void glutMouseFunc (void (*func)(int button, int state,
int x, int y));
- void glutMotionFunc (void (*func) (int x,int y)
- ... altres que anirem veient

glutPostRedisplay(); Per marcar finestra per repintar

IDI- Q2 2013-2014

Repàs: OpenGL

■ Màquina d'estats

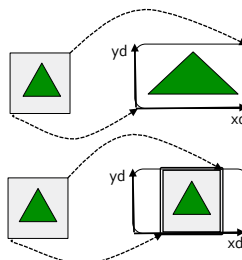
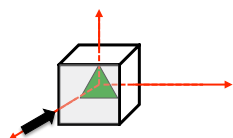


IDI- Q2 2013-2014

Funcions per modificar l'estat

- OpenGL té valors per defecte de les variables d'estat.
- De moment utilitzarem volum de visió per defecte:

Volum per defecte:
cub de $(-1, -1, -1)$ fins $(1, 1, 1)$



- Definició del *viewport* (`glViewport()`): mateixa relació d'aspecte que el *window* per a no tenir deformacions; per defecte tota la pantalla => registrar *callback* `glutReshapeFunc(resize)` per redefinir *viewport* (`glViewport(...)`) quan es modifica grandària finestra.

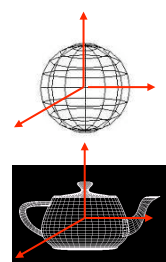
IDI- Q2 2013-2014

Secció 1: Objectes glut

```
#include <GL/gl.h>
#include <GL/freeglut.h>
void refresh (void)
{ glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
  glBegin(GL_TRIANGLES);
    glVertex3f(-0.5,-0.5,0.0);
    glVertex3f(0.5,0.0,0.0);
    glVertex3f(0.0,0.5,0.0);
  glEnd();
  glutSwapBuffers();
}

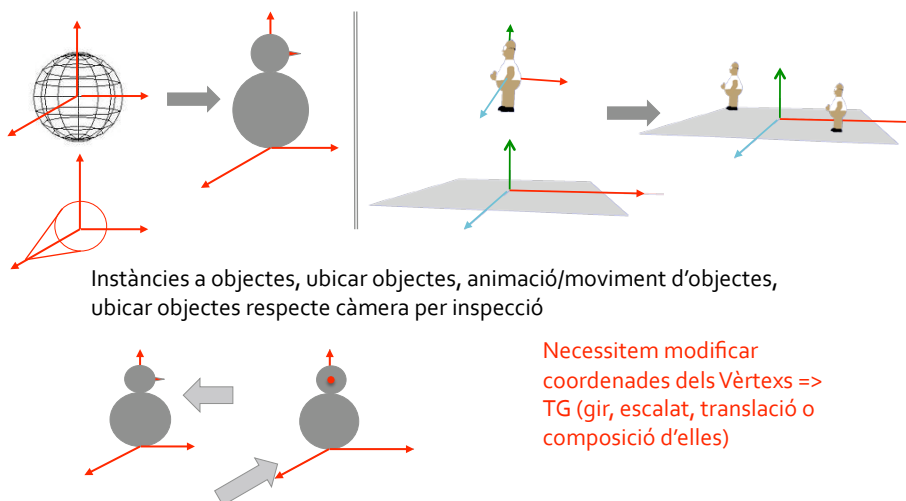
int main(int argc, const char *argv[])
{ glutInit(&argc, ((char **) argv);
  glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGB);
  glutInitWindowSize(600,600);
  glutCreateWindow("IDI: Practiques OpenGL");
  glutDisplayFunc (refresh);
  glutMainLoop();
  return 0;
}
```

glutWireSphere(GLdouble radius, GLint slices, GLint stacks)
 glutWireCone(GLdouble base, GLdouble height, GLint slices, GLint stacks)
 glutWireTeapot(GLdouble size)
 ...



IDI- Q2 2013-2014

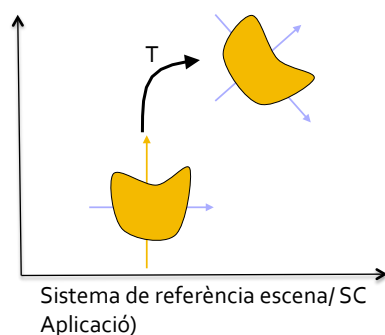
Secció 2: TG. Necessitat



IDI- Q2 2013-2014

Secció 2: TG. Repàs (mireu apunts racó)

- Transformacions geomètriques bàsiques: escalat, rotacions, translació

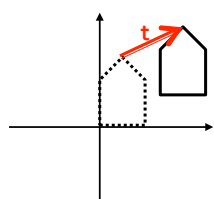


Transformació geomètrica →  → Matriu 4x4 TG

IDI- Q2 2013-2014

Exemples 2D de Transformacions

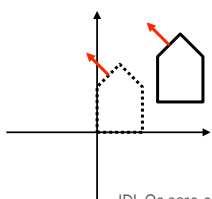
- Translació de punts



$$\begin{aligned}x' &= x + t_x \\ y' &= y + t_y\end{aligned}$$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x + t_x \\ y + t_y \\ 1 \end{pmatrix}$$

- Translacions no afecten als **vectors**

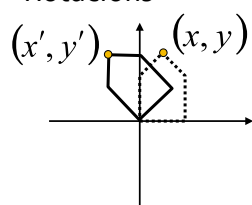


$$\begin{pmatrix} x' \\ y' \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 0 \end{pmatrix} = \begin{pmatrix} x \\ y \\ 0 \end{pmatrix}$$

IDI- Q2 2013-2014

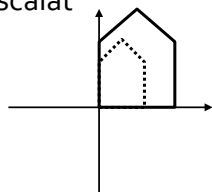
Exemples 2D (en coord, homogènies)

■ Rotacions



$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

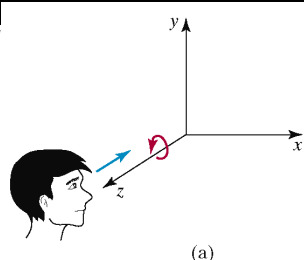
■ Escalat



$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} s_x x \\ s_y y \\ 1 \end{pmatrix}$$

IDI- Q2 2013-2014

Matriu de rotació sobre l'eix Z



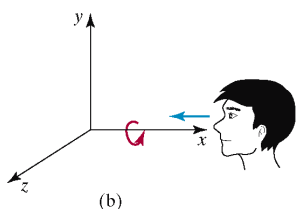
$$\begin{aligned} x' &= x \cos \alpha - y \sin \alpha \\ y' &= y \sin \alpha + x \cos \alpha \\ z' &= z \end{aligned}$$

`glRotatef(angle, 0, 0, 1);`

$$R_Z(\alpha) \cdot P = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 & 0 \\ \sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \cos \alpha - y \sin \alpha \\ x \sin \alpha + y \cos \alpha \\ z \\ 1 \end{bmatrix}$$

IDI- Q2 2013-2014

Matriu de rotació sobre l'eix X



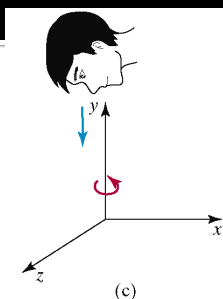
$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

`glRotatef(angle,1,0,0);`

$$R_x(\alpha) \cdot P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \cos \alpha - z \sin \alpha \\ y \sin \alpha + z \cos \alpha \\ 1 \end{bmatrix}$$

IDI- Q2 2013-2014

Matriu de rotació sobre l'eix Y



$$R_y(\alpha) = \begin{bmatrix} \cos \alpha & 0 & \sin \alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \alpha & 0 & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

`glRotatef(angle,0,1,0);`

$$R_y(\alpha) \cdot P = \begin{bmatrix} \cos \alpha & 0 & \sin \alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \alpha & 0 & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \cos \alpha + z \sin \alpha \\ y \\ -x \sin \alpha + z \cos \alpha \\ 1 \end{bmatrix}$$

IDI- Q2 2013-2014

Matriu d'escalat 3D

$$S(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{glScalef}(sx, sy, sz);$$

$$S(s_x, s_y, s_z) \cdot P = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} xs_x \\ ys_y \\ zs_z \\ 1 \end{bmatrix}$$

IDI- Q2 2013-2014

Matriu de translació

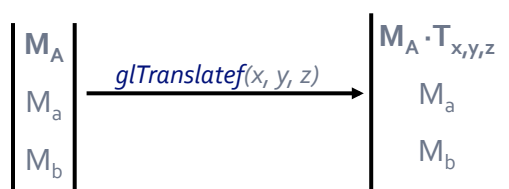
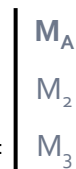
$$T(t_x, t_y, t_z) = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{glTranslatef}(tx, ty, tz);$$

$$T(t_x, t_y, t_z) \cdot P = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \\ z + t_z \\ 1 \end{bmatrix}$$

IDI- Q2 2013-2014

Transformations geomètriques i OpenGL

- OpenGL manté *dues piles de matrius* (la MODELVIEW i la PROJECTION). Només una pot estar activa per a ser modificada. Per defecte: la pila MODELVIEW
- La matriu *activa* M_A és sempre la del top de la *pila activa*
- Té instruccions per a **crear matrius de TG que afecten a la M_A** :
 - `void glTranslatef(TYPE x, TYPE y, TYPE z);`
 - `void glRotatef(TYPE angle, TYPE x, TYPE y, TYPE z);`
 - `void glScalef(TYPE x, TYPE y, TYPE z);`

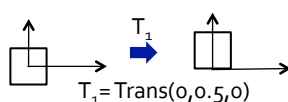


$M_A = M_A * TG$
Noteu l'ordre de la multiplicació!!!

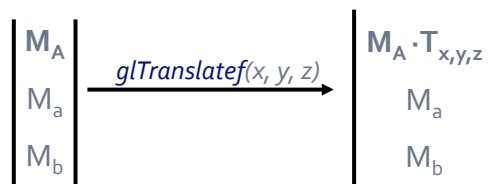
IDI- Q2 2013-2014

Transformations Geométriques i OpenGL: exemple 1

- Quan s'envia a pintar un vèrtex V , li aplica primer la matriu del top de la pila MODELVIEW (el "mou") i després "el pinta".
 - $V_transformat = M_A * V$



```
glMatrixMode(GL_MODELVIEW);
glTranslatef(0,0.5,0);
glutWireCube(1);
```



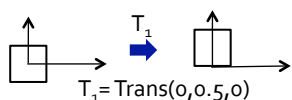
Segur? Quin valor M_A ?

IDI- Q2 2013-2014

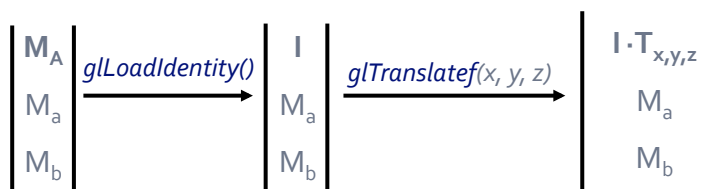
Transformations Geomètriques i OpenGL

- Quan s'envia a pintar un vèrtex V , li aplica primer la matriu del top de la pila MODELVIEW (el "mou") i després "el pinta".

$$V_transformat = M_A * V$$



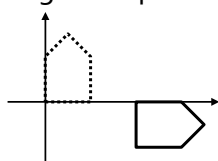
```
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
glTranslatef(0, 0.5, 0);
glutWireCube(1);
```



IDI- Q2 2013-2014

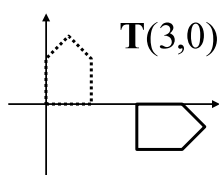
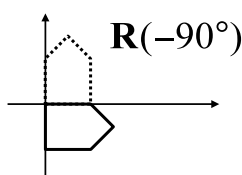
Composició de Transformacions

- Imaginem que volem



No es pot fer amb cap de les matrius anteriors

- Cal composar/efectuar dues transformacions (moviments)



$$M = T(3,0) \cdot R(-90^\circ)$$

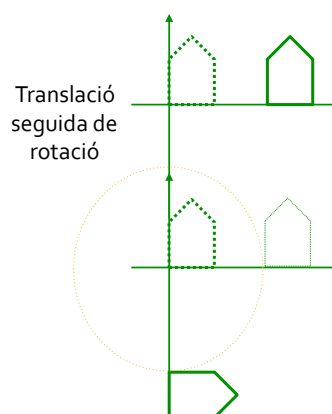
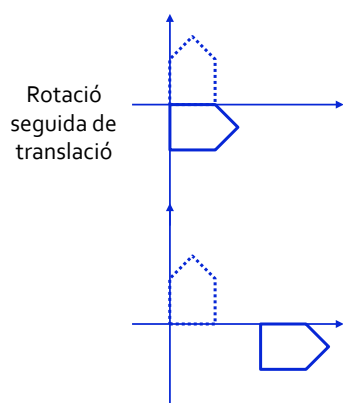
$$P' = T(3,0) \cdot (R(-90^\circ) P) = (T(3,0) \cdot R(-90^\circ)) P = M \cdot P$$

IDI- Q2 2013-2014

Composició de Transformacions

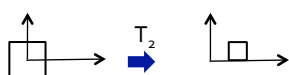
$$\underset{2}{T(3,0)} \cdot \underset{1}{R(-90^\circ)} \neq \underset{2}{R(-90^\circ)} \cdot \underset{1}{T(3,0)}$$

- Multiplicació de matrius no és commutativa



IDI- Q2 2013-2014

Transformations Geomètriques i OpenGL: Exemple 2

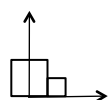


$$T_2 = \text{Trans}(0.75, 0.25, 0) * S(0.5, 0.5, 0.5)$$

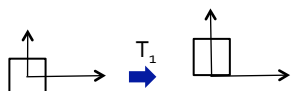
```
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
glTranslatef(0.75, 0.25, 0);
glScalef(0.5, 0.5, 0.5);
glutWireCube(1);
```

IDI- Q2 2013-2014

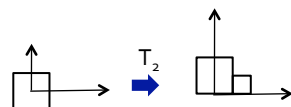
TG i OpenGL: exemple 3



Escena a pintar utilitzant instàncies de `glutWireCube()`



$$T_1 = \text{Trans}(0, 0.5, 0)$$



$$T_2 = \text{Trans}(0.75, 0.25, 0) * S(0.5, 0.5, 0.5)$$

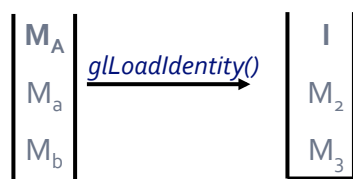
Comporta aplicar TG diferents als diferents objectes.

Com fer-ho amb OpenGL?

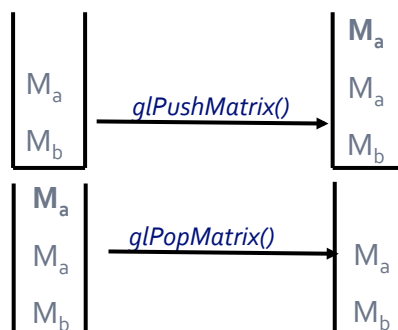
IDI- Q2 2013-2014

TG i OpenGL

- Substituir la matriu al top de la pila: (`glLoadIdentity`, `glLoadMatrix*`)



- Duplicar la matriu del top de la pila: `glPushMatrix()`
- Eliminar la matriu del top de la pila: `glPopMatrix()`

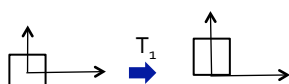


IDI- Q2 2013-2014

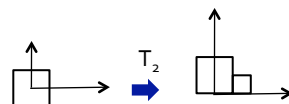
TG i OpenGL: exemple 3



Escena a pintar utilitzant instàncies de `glutWireCube()`



$$T_1 = \text{Trans}(0, 0.5, 0)$$



$$T_2 = \text{Trans}(0.75, 0.25, 0) * S(0.5, 0.5, 0.5)$$

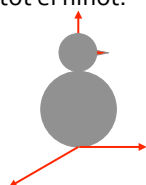
```
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
glPushMatrix();
glTranslatef(0, 0.5, 0);
glutWireCube(1);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75, 0.25, 0);
glScalef(0.5, 0.5, 0.5);
glutWireCube(1);
glPopMatrix();
```

Ara gireu els dos cubs entorn de l'eix Y. Com?

IDI- Q2 2013-2014

Què heu de fer en 1ra sessió Bloc 2?

- Pintar algun objecte *glut* (secció 1)
- Utilitzar OpenGL per aplicar TG a un objecte (secció 2)
 - Entendre els paràmetres de les crides i composició d'operacions
 - Recordeu que aplica la matriu del top de la pila
 - Utilitzeu *callbacks* de teclat i ratolí per modificar TG; els reutilitzareu!!
- Utilitzar OpenGL per a aplicar diferents TG als diferents objectes de l'escena
 - Caldrà Push/Pop Matius
- Crear una escena utilitzant objectes *glut* (secció 3) => TG per a crear el ninot i TG per a moure tot el ninot.



IDI- Q2 2013-2014

© Professors d'IDI – Curs 2013-2014

Bloc_2: Transformacions Geomètriques i Models