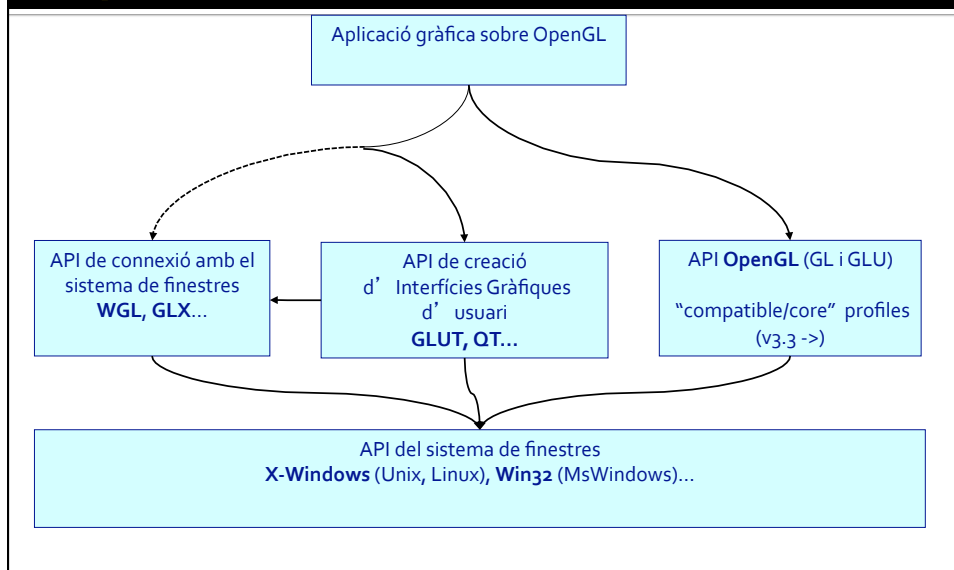


© Professors d'IDI – Curs 2012-2013

Introducció a glut

Arquitectura bàsica d'una aplicació OpenGL



Introducció

- GLUT: OpenGL Utility Toolkit
 - Biblioteca de funcions implementada per Mark J. Kilgard
 - Construcció d'aplicacions d'OpenGL independents del sistema de finestres
 - Implementada inicialment per X windows, i portada a Microsoft Windows per Nate Robins
 - Útil per a construir petites aplicacions en OpenGL i aprendre
 - Continguts extrets del tutorial de Lighthouse3d a www.lighthouse3d.com

Introducció

- Les distribucions de glut vénen amb molts exemples
- Recordeu utilitzar el Manual de glut.
- Hi ha versions de codi lliure:
 - [freeGLUT](http://freeglut.sourceforge.net) (<http://freeglut.sourceforge.net>)
 - [OpenGLUT](http://openglut.sourceforge.net) (<http://openglut.sourceforge.net>)

Estructura d'una aplicació glut

- Tres parts:
 - Inicialitzacions
 - Registre callbacks
 - Bucle d'atenció als esdeveniments

```
int main(int argc, char **argv)
{
    // init GLUT and create window
    // register callbacks
    // enter GLUT event processing cycle
}
```

Inicialitzacions

- Totes les funcions de *glut* tenen prefix *glut*
- Les que fan quelcom relacionat amb inicialitzacions comencen per *glutInit*
- Inicialització bàsica:
void glutInit(int *argc, char **argv);
 - Inicialitza glut
 - Paràmetres:
 - *argc* – Punter a la variable *argc* del programa principal
 - *argv* – Punter a la variable *argv* del programa principal

Inicialitzacions: configuració de finestres

void glutInitWindowPosition(int x, int y);

- Suggereix la posició de la finestra
- Paràmetres:
 - x – Distància en píxels del costat esquerre. Un valor per defecte de -1 indica que el gestor de finestres la posi on consideri. El gestor de finestra pot no fer cas d'aquest valor ☹.
 - y – Ídem anterior en vertical

Inicialitzacions: configuracions de finestres

void glutInitWindowSize(int sizex, int sizey);

- Determina la mida de la finestra
- Paràmetres:
 - *sizex* – Mida en horitzontal.
El gestor de finestres pot no fer-ne cas
 - *sizey* – Ídem anterior en vertical
- Exemple: *glutInitWindowSize (600,600);*

Inicialitzacions: definició del context d'OpenGL

void glutInitDisplayMode(int mode);

- Determina els buffers que conformen el context OpenGL
 - GLUT_RGB o GLUT_RGBA determina un buffer RGBA
 - GLUT_SINGLE, GLUT_DOUBLE: Un o dos buffers de pintat
 - GLUT_DEPTH: buffer de profunditat
 - GLUT_STENCIL: utilitza el buffer de stencil
- Exemple:

```
glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE |  
GLUT_DEPTH);
```

Inicialitzacions: creació de finestres

void glutCreateWindow(char* nom);

- Crea la finestra amb la configuració determinada i el títol que li passem.
- Exemple:

```
glutCreateWindow("IDI: Practiques OpenGL");
```

Inicialitzacions: exemple

- Integrant les inicialitzacions:

```
int main(int argc, char **argv)
{
    // init GLUT and create Window
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE |
                        GLUT_RGBA);
    glutInitWindowPosition(100,100);
    glutInitWindowSize(600,600);
    glutCreateWindow("IDI: Practiques OpenGL")
    // register callbacks
    // enter GLUT event processing cycle
}
```

Registre de callbacks

- glut* permet la definició de funcions que atendran events del sistema (ratolí, teclat...).
- S'anomenen *callbacks*.
- El procés d'associació de rutines d'atenció als events. s'anomena **registre**.
- Un exemple de funció de callback:

void glutDisplayFunc(void (*funcName)void);

- Registra la funció de pintat.
- S'executa cada vegada que glut detecta que el contingut de la finestra s'ha de repintar.
- El paràmetre conté un punter a la funció de pintat que no requereix cap paràmetre

Registre de callbacks

- Definició del callback:

```
void renderScene(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();
}
```

- Es registraria com: `glutDisplayFunc(renderScene);`

Registre de callbacks: exemples

- `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

Processament d'events

- Una vegada s'ha configurat tot i els callbacks estan registrats, cal entrar en el bucle de processament d'events:

void glutMainLoop(void)

Primer exemple

```
#include <GL/gl.h>
#include <GL/glut.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_RGBA);
  glutInitWindowSize(600,600);
  glutCreateWindow("IDI: Practiques OpenGL");
  glutDisplayFunc(refresh);
  glutMainLoop();
  return 0;
}
```


Més informació

- Links:
 - <http://www.opengl.org/resources/libraries/glut/>
 - <http://freeglut.sourceforge.net>
 - <http://openglut.sourceforge.net>
- Alternatives a glut:
 - Nui: <http://libnui.net>
 - CPW: <http://mathies.com/cpw/about.html>
 - ...

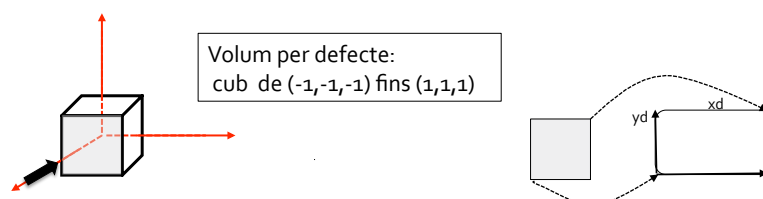
Bloc 1: visualització bàsica

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

- editar codi, makefile, executar, entendre ordre
- implementar initGL() –color fons- ; refresh () -neteja buffers-
- completar refresh per pintar un triangle (*recordeu el volum de visió de defecte*)
- evitar deformacions i centrat (registrar callback...glutReshapeFunc();
modificar viewport per no deformació: glViewport(...))
- anàlisi retallat

Funcions per modificar l'estat

- OpenGL té valors per defecte de les variables d'estat.
- Volum de visió per defecte:



- Definició del *viewport* (`glViewport()`): per defecte tota la finestra gràfica.

Bloc 1: interacció i més color

- Esdeveniments ratolí i callbacks associats
 - `glutMouseFunc(...)`
callback funció amb 4 paràmetres: botó, press/un_press, posició ratolí
 - `glutMotionFunc(...)`
callback funció amb 2 paràmetres: posició ratolí
 - a l'arrossegar el ratolí, el color de fons es fagi més fosc o més clar
- Esdeveniment teclat i callbacks associats:
 - `glutKeyboardFunc(...)`
callback funció amb tres paràmetres: caràcter, posició ratolí
 - escriure help; que ESC (ASCII 27) tanqui aplicació
 - canvi estat aplicació: 'F' modificació color fons; 't' entrar 3 vèrtexs de triangles
- Instrucció `glutPostRedisplay();`
- Més color
 - Color per vèrtex als triangles
 - Altres tipus de primitives

Lliuraments

- Cal haver lliurat l'aplicació resultat de 3 dels 4 blocs de laboratori per a poder presentar-se a l'examen de laboratori.
- Bloc 1 (1 sessió): 10 minuts inicials de la primera classe del Bloc 2
- Bloc 2 (3 sessions): 10 minuts inicials de la primera classe del Bloc 3
- Bloc 3 (2 sessions): 10 minuts inicials de la primera classe del Bloc 4
- Bloc 4: dia abans examen laboratori (darrera setmana lectiva)