

Laboratori Gràfics

- 7.5 sessions "gràfics" i interacció directa organitzades en 4 blocs + Examen laboratori (19 Nov 2013)
- 4.5 sessions "interfícies en Qt" + Examen (17 Dec 2013)
- Projecte Android (23 de Dec 2013)

- Cal haver lliurat l'aplicació resultat de 3 dels 4 blocs de laboratori per a poder presentar-se a l'examen de laboratori de "gràfics".

- Bloc 1 (1 sessió): 10 minuts inicials de la primera classe del Bloc 2
- Bloc 2 (2 sessions): 10 minuts inicials de la primera classe del Bloc 3
- Bloc 3 (3 sessions): 10 minuts inicials de la primera classe del Bloc 4
- Bloc 4 (2 sessions): el dia abans de l'examen laboratori de gràfics

IDI Q1 2013-2014

1

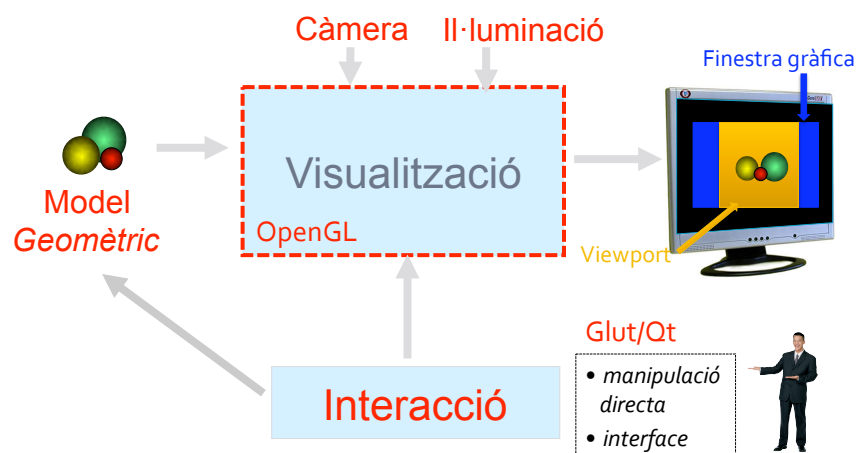
© Professors d'IDI – Curs 2013-2014

Sessió 1 de laboratori

IDI Q1 2013-2014

2

Elements Sistema Gràfic



IDI Q1 2013-2014

3

Aplicació visualització interactiva

- **Inicialització**
 - Crear una (o més) finestres gràfiques.
 - Crear els elements de la interfície gràfica de usuari.
 - Establir quin codi cal executar-se com a resposta a cada acció del usuari (pulsació de una tecla, clic en una opció de menú...).
- **Execució**
 - Respondre a l'entrada de l'usuari.
 - Visualitzar l'escena (segons càmera) en el Viewport.

IDI Q1 2013-2014

4

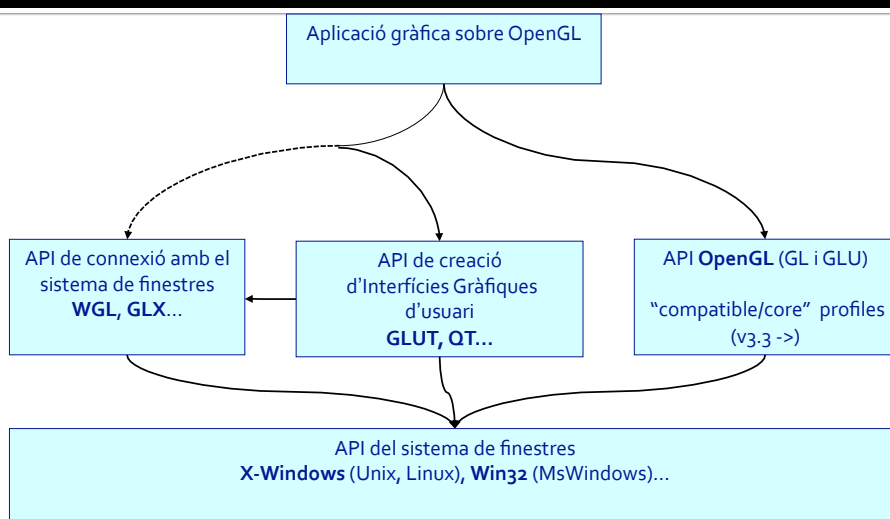
Introducció

- Què és OpenGL?
 - OpenGL és una **API per a la programació d'aplicacions gràfiques 2D i 3D** que actua com a interfície amb el *hardware* gràfic.
 - L'API OpenGL és multiplataforma gràcies a la seva **independència del sistema de finestres**.
 - La interfície consisteix en una biblioteca de funcions (C/C++) que permeten la visualització d'una escena 3D.
 - **No gestiona "inputs" d'usuari.**

IDI Q1 2013-2014

5

Arquitectura bàsica d'una aplicació OpenGL



IDI Q1 2013-2014

6

Objectius sessió

- Mínima introducció a OpenGL i glut.
- Entendre les parts d'una aplicació glut i practicar registre de *callbacks* i tractament d'events.
- Desenvolupar una primera aplicació.

IDI Q1 2013-2014

7

© Professors d'IDI – Curs 2013-2014

Introducció a OpenGL

IDI Q1 2013-2014

8

Característiques d'OpenGL

- Generals:
 - Entorn d'execució d'OpenGL:
 - Client
 - Servidor
 - OpenGL és **independent del hardware**.
 - **No proporciona** una estructura de representació per als objectes 3D. Simplement els pinta.
 - Es comporta com una **màquina d'estats**

IDI Q1 2013-2014

9

Característiques d'OpenGL

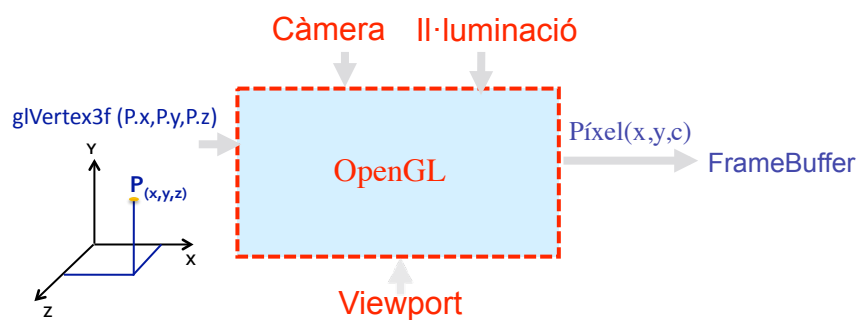
- Les funcions d'OpenGL són de tres tipus:
 - Permeten modificar una variable d'estat:
 - Color, material, fonts de llum, **matrius**-camera-
 - Permeten consultar el valor d'una variable d'estat:
 - glGetBooleanv(),..
 - Permeten modificar de forma directa la informació visible.
 - Per exemple, el dibuixat de primitives gràfiques

IDI Q1 2013-2014

10

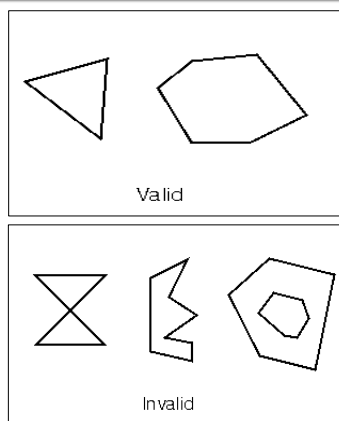
Repàs: OpenGL

■ Màquina d'estats

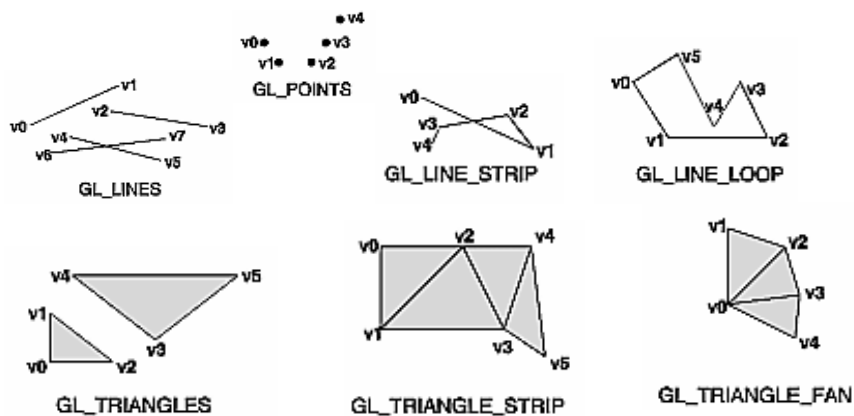


Primitives gràfiques

- OpenGL soporta quatre primitives gràfiques:
 - Punts
 - Línies poligonals
 - Polígons simples convexos
 - Bitmaps



Primitives gràfiques



IDI Q1 2013-2014

13

Primitives gràfiques

Tipus de primitiva

GL_POINTS

GL_LINES

GL_LINE_STRIP

GL_LINE_LOOP

GL_TRIANGLES

GL_TRIANGLE_STRIP

GL_TRIANGLE_FAN Triangle fan

GL_QUADS

GL_QUAD_STRIP

GL_POLYGON

Significat

Punts independents

Segment independent (2 vert.)

Poligonal

Polígon tancat.

Triangle independent (3 vert.)

Triangle strip

Quadrilàters indep. (4 vert.)

Quad strip

Polígon simple i convex

IDI Q1 2013-2014

14

Sintaxi de les comandes OpenGL

- Una mateixa funció adopta diferents prototipus (*poor man's mangling*) que varien únicament en:
 - Nombre d'arguments
 - Tipus dels arguments (*int, float, double...*)
 - Adreçament (referència o valor)

IDI Q1 2013-2014

15

Sintaxi de les comandes OpenGL. Exemple

```
void glVertex3d (GLdouble x , GLdouble y , GLdouble z );
void glVertex3f (GLfloat x , GLfloat y , GLfloat z );
void glVertex3i (GLint x , GLint y , GLint z );
void glVertex3s (GLshort x , GLshort y , GLshort z );
void glVertex3dv (const GLdouble *v );
void glVertex3fv (const GLfloat *v );
void glVertex3iv (const GLint *v );
void glVertex3sv (const GLshort *v );
```

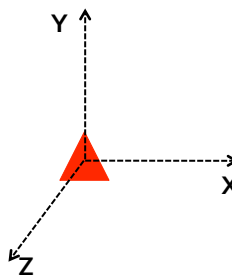
IDI Q1 2013-2014

16

Primitives gràfiques: exemple

- Les primitives geomètriques es dibuixen usant blocs **glBegin-glEnd**:

```
glBegin(GL_TRIANGLES);
glColor(1,0,0);
glVertex3f(-0.5,-0.5,0.0);
glVertex3f(0.5,-0.5,0.0);
glVertex3f(0.0,0.5,0.0);
glEnd();
```



En quin lloc de la finestra gràfica ho pinta? => funció del viewport
En quina orientació? => funció de la càmera definida

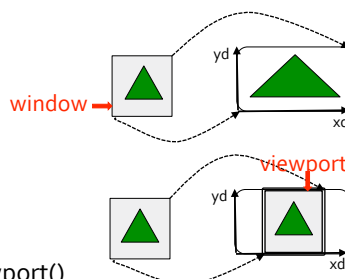
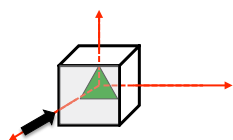
IDI Q1 2013-2014

17

Funcions per modificar l'estat

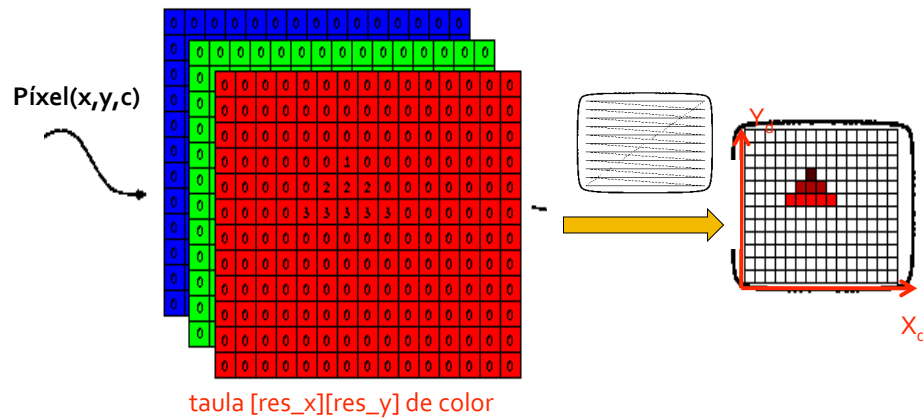
- OpenGL té valors per defecte de les variables d'estat.
- De moment utilitzarem volum de visió per defecte (càmera de defecte):

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



- Definició del *viewport* => `glViewport()`
Ha de tenir la mateixa relació d'aspecte que el window per a no tenir deformacions. Per defecte és tota la pantalla (pista pel futur... en glut cal definir el viewport en el *callback* de `glutReshapeFunc()`)

Frame buffer



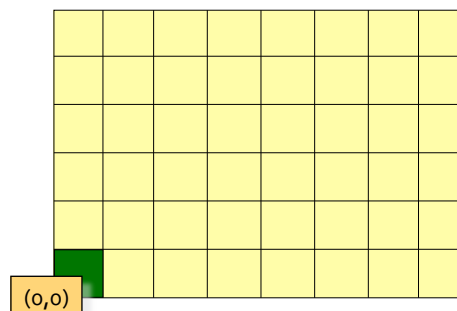
Pinta en un *Buffer*:

Definit per la seva resolució (n x m) –finestra gràfica- i
la seva profunditat (nombre de bits/píxel)

19

Característiques d'OpenGL

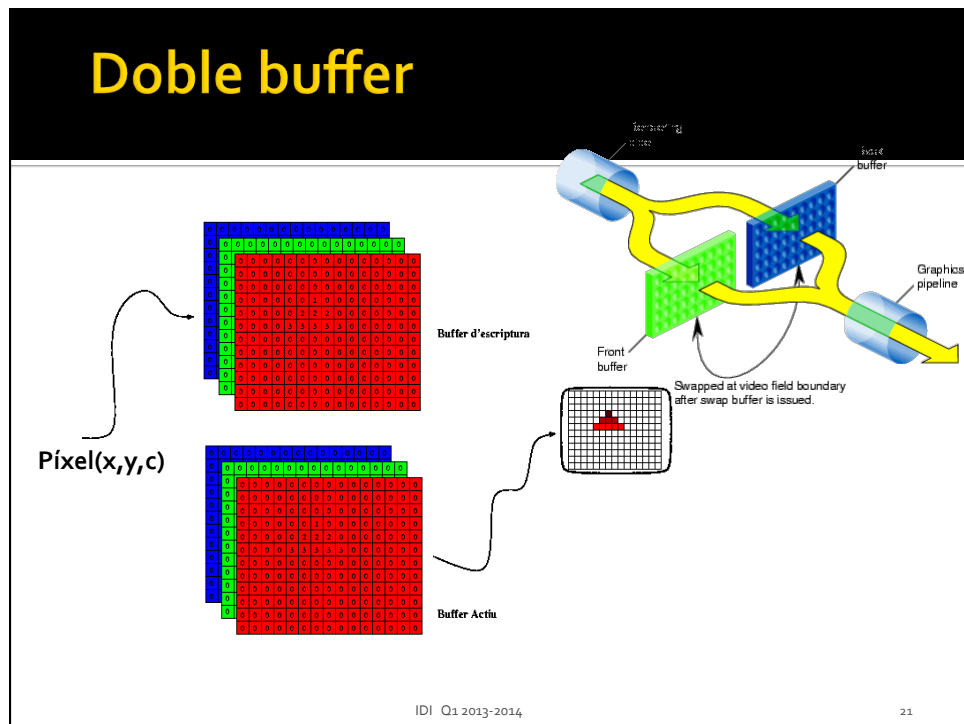
- OpenGL considera l'origen del SC dispositiu a la cantonada inferior esquerra.



IDI Q1 2013-2014

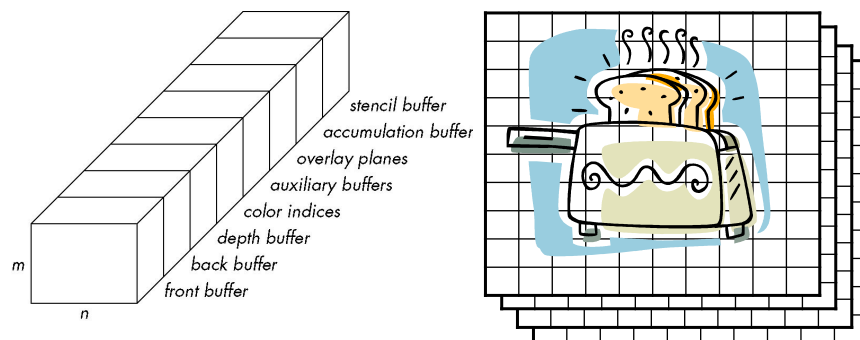
20

Doble buffer



Característiques d'OpenGL

Multi-layered framebuffer



Altres funcions bàsiques de dibuix

- Netejar el buffer
 - Definir color neteja:
`glClearColor(red, green, blue, alpha);`
 - Fer efectiva la neteja:
`glClear(GL_COLOR_BUFFER_BIT);`
 - Podem netejar més d'un buffer:
`glClear(GL_COLOR_BUFFER_BIT, GL_DEPTH_BUFFER_BIT);`

IDI Q1 2013-2014

23

Primer exemple de pintat

```
void refresh (void)
{ glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
  glBegin(GL_TRIANGLES);
    glVertex3f(-0.5,0.0,0.0);
    glVertex3f(0.5,0.0,0.0);
    glVertex3f(0.0,0.5,0.0);
  glEnd();
  glutSwapBuffers();
}
```

IDI Q1 2013-2014

© Professors d'IDI – Curs 2013-2014

Introducció a Glut

IDI Q1 2013-2014

25

Introducció glut

- 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

IDI Q1 2013-2014

26

Introducció glut

- 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>)

IDI Q1 2013-2014

27

Primer exemple

```
#include <GL/gl.h>
#include <GL/freeglut.h>
void refresh(void)
{ glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
  glBegin(GL_TRIANGLES);
    glVertex3f(-0.5,0.0,0.0);
    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_RGBA);
  glutInitWindowSize(600,600);
  glutCreateWindow("IDI: Practiques OpenGL");
  glutDisplayFunc(refresh);
  ...
  glutMainLoop();
  return 0;
}
```

1. inicialitzacions

2. Registre *callbacks*3. Bucle
processament
events

IDI Q1 2013-2014

28

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

IDI Q1 2013-2014

29

Inicialitzacions: configuració de finestres

```
void glutInitWindowPosition(int x, int y);
```

- Suggereix la posició de la finestra gràfica en pantalla
- 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

IDI Q1 2013-2014

30

Inicialitzacions: configuracions de finestres

void glutInitWindowSize(int sizex, int sizey);

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

IDI Q1 2013-2014

31

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);
```

IDI Q1 2013-2014

32

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");

IDI Q1 2013-2014

33

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
}
```

IDI Q1 2013-2014

34

Registre de callbacks

- *glut* permet definir funcions –**callbacks**– que atendran els events del sistema (ratolí, teclat...).
- El procés d'associació de rutines d'atenció als events. s'anomena **registre**.
- Exemple:
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

IDI Q1 2013-2014

35

Primer exemple

```
#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_RGBA);
  glutInitWindowSize(600,600);
  glutCreateWindow("IDI: Practiques OpenGL");
  glutDisplayFunc (refresh);
  ...
  glutMainLoop();
  return 0;
}
```

1. inicialitzacions

2. Registre *callbacks*3. Bucle
processament
events

IDI Q1 2013-2014

36

Registre de callbacks: exemples

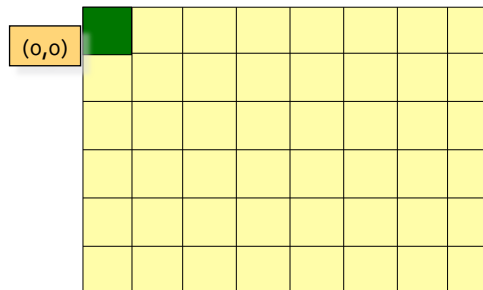
- `void glutDisplayFunc (void (*funcName)void);`
- `void glutReshapeFunc (void (*func)(int width, int height));`
- `void glutKeyboardFunc (void (*func) (unsigned char key,
int x, int y));`
- `void glutMouseFunc (void (*func)(int button, int state,
int x, int y));`
- `void glutMotionFunc (void (*func)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

IDI Q1 2013-2014

37

Consideració

- glut considera l'origen del SC dispositiu a la cantonada superior esquerra de la finestra gràfica.



IDI Q1 2013-2014

38

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)
```

IDI Q1 2013-2014

39

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(-0.5,0.0,0.0);
    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_RGBA);
  glutInitWindowSize(600,600);
  glutCreateWindow("IDI: Practiques OpenGL");
  glutDisplayFunc(refresh);
  glutMainLoop();
  return 0;
}
```

IDI Q1 2013-2014

40

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>
 - ...

IDI Q1 2013-2014

41

© Professors d'IDI – Curs 2013-2014

Bloc 1

IDI Q1 2013-2014

42

Bloc 1: visualització bàsica

1. - Editar codi (crea finestra), makefile, executar, entendre ordre.
 - Implementar:
 - `InitGL()` inicialitzant el color de fons
 - Callback *refresh()* amb neteja buffers
2. - Completar `refresh()` per pintar un triangle
 - Registrar i implementar el callback "resize"
 `glutReshapeFunc()`
 per a evitar deformacions al modificar finestra i generar dibuix
 optimitzant espai finestra → modificar viewport : `glViewport(...)`
 - Analitzar l'efecte de "retallat" si algun vèrtex del triangle no està dins
 del volum visió de defecte.

IDI Q1 2013-2014

43

Bloc 1: interacció bàsica

3. 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 faci més fosc o més clar
4. Esdeveniment teclat i callbacks associats:
 - **`glutKeyboardFunc(...)`**
 callback funció amb tres paràmetres: caràcter, posició ratolí
 → escriure help al premer "h"
 → si ESC (ASCII 27) es tanqui aplicació

IDI Q1 2013-2014

44

Bloc 1: interacció

5. Interacció més elaborada:

- ➔ Al premer 't' permet entrar les coordenades dels 3 vèrtexs dels triangles amb ratolí i que pinti el triangle amb els vèrtexs en els píxels indicats.

Pistes:

- recordeu el mapeix del window $(-1,-1,1,1)$ a viewport.
- el ratolí retorna coordenades dispositiu (píxels) –recordeu origen- que caldrà transformar a coordenades window i enviar a pintar amb OpenGL.

6. Més color... (opcional): Llegiu el guió

7. Resum de l'aplicació que heu de lliurar: Llegiu el guió (pràcticament, el resultat dels punts 1 a 6 anteriors)