

Introducción práctica a OpenGL

Introducción a la Computación Gráfica

Andrea Rueda

Pontificia Universidad Javeriana
Departamento de Ingeniería de Sistemas

Introducción a OpenGL

Pipeline fija de OpenGL (C++)

- Requiere la definición de:
 - Procedimiento de inicialización:
iniciación de la ventana de visualización, parámetros.
 - Procedimiento de visualización:
generación del contenido de la ventana de visualización.
 - Procedimiento de actualización de ventana:
actualización del contenido cuando la ventana cambia de tamaño.
 - Procedimiento principal.

Pipeline fija de OpenGL (C++)

- Descargue el código base:

uvirtual.javeriana.edu.co

- Contenidos
- Presentaciones
- Semana 2
- `introOpenGL.cpp`

Pipeline fija de OpenGL (C++)

- Librerías:

- Windows:

```
#include <GL/glut.h>
```

- Linux:

```
#include <GL/freeglut.h>  
#include <GL/gl.h>
```

- Mac:

```
#include <GLUT/GLUT.h>  
#include <OpenGL/OpenGL.h>
```

Pipeline fija de OpenGL (C++)

- Inicialización:

```
void myInit (void) {  
  
    glClearColor(1.0,1.0,1.0,1.0);  
  
    glMatrixMode(GL_PROJECTION);  
  
    glLoadIdentity();  
  
    gluOrtho2D(0, 640, 0, 480);  
  
}
```

Pipeline fija de OpenGL (C++)

- Inicialización:

```
void myInit (void) {  
  
    glClearColor(1.0,1.0,1.0,1.0);  
  
    glMatrixMode(GL_PROJECTION);  
  
    glLoadIdentity();  
  
    glOrtho(0, 640, 0, 480, 1.0, -1.0);  
  
}
```

Pipeline fija de OpenGL (C++)

- Visualización:

```
void myDisplay (void) {  
    glClear(GL_COLOR_BUFFER_BIT);  
    glColor3f(1.0, 0.0, 0.0);  
}
```


Pipeline fija de OpenGL (C++)

- Visualización:

```
glBegin(GL_LINES);  
  
    glVertex2i(600,15);  
  
    glVertex2i(10,345);  
  
glEnd();  
  
glFlush();  
  
}
```

Pipeline fija de OpenGL (C++)

- Actualización:

```
void myResize (int w, int h) {  
    glViewport(0, 0, w, h);  
  
    glMatrixMode(GL_PROJECTION);  
  
    glLoadIdentity();  
  
    gluOrtho2D(0, w, 0, h);  
  
}
```

Pipeline fija de OpenGL (C++)

- Procedimiento principal:

```
int main (int argc, char** argv) {  
  
    glutInit(&argc, argv);  
  
    glutInitDisplayMode(GLUT_SINGLE |  
        GLUT_RGB);  
  
    glutInitWindowSize(640, 480);  
  
    glutInitWindowPosition(100, 100);  
}
```

Pipeline fija de OpenGL (C++)

- Procedimiento principal:

```
glutCreateWindow( "Graphics Window" );  
  
glutDisplayFunc( myDisplay );  
  
glutReshapeFunc( myResize );  
  
myInit( );  
  
glutMainLoop( );  
return 0;  
}
```

Ahora... a probar!

- Compilar en Linux:

```
g++ -o pruebaApp introOpenGL.cpp  
-lGL -lGLU -lglut
```

- Ejecutar:

```
./pruebaApp
```

Ahora... a probar!

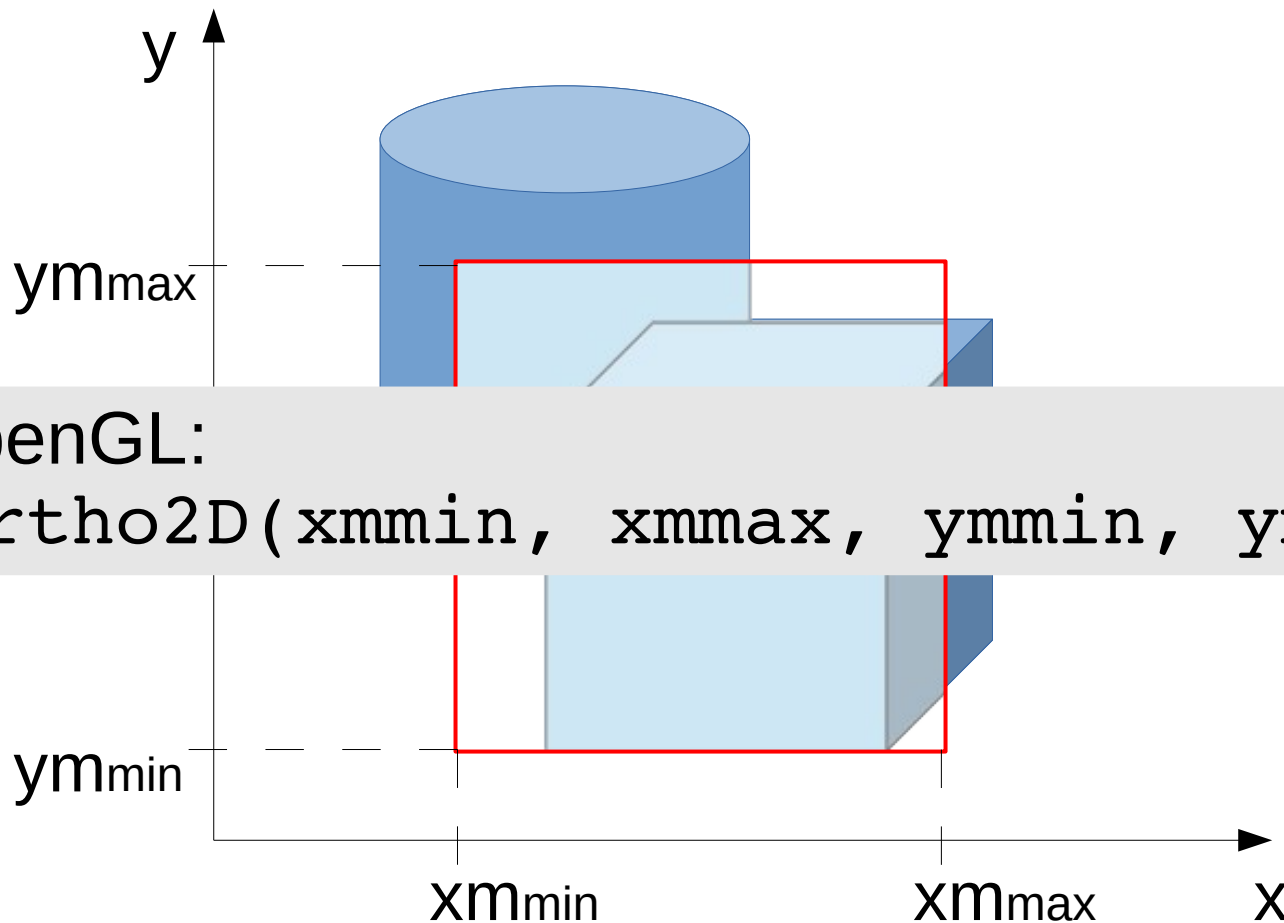
- Modifique el programa para:
 - Añadir una línea de color azul que cruce la línea ya pintada.
 - Dibujar un triángulo usando 3 líneas, cada una de un color diferente.

Sistemas de referencia

- Sistema de referencia de la pantalla:
 - Gestión de la ventana de visualización:
 - Creación (Título):
`glutCreateWindow("Graphics Window");`
 - Posición en la pantalla:
`glutInitWindowPosition(100, 100);`
 - Tamaño:
`glutInitWindowSize(640, 480);`

Sistemas de referencia

- Área de los objetos (ventana del mundo real):

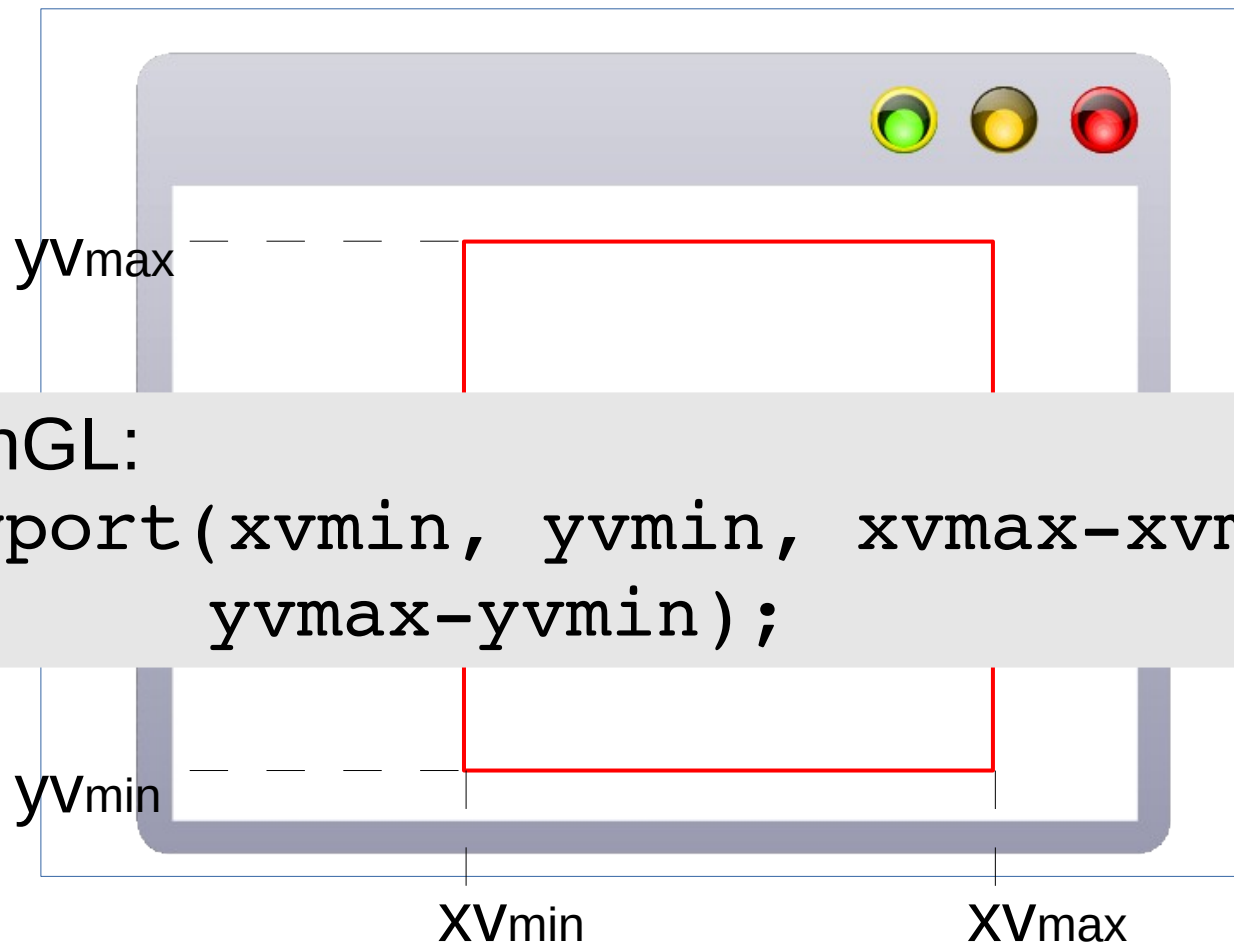


en OpenGL:

```
gluOrtho2D(xmin, xmax, ymin, ymax);
```


Sistemas de referencia

- Área de visualización (en pantalla):



en OpenGL:

```
glViewport(xvmin, yvmin, xvmax-xvmin,  
yvmax-yvmin);
```

Sistemas de referencia

- En el procedimiento de inicialización, el área de visualización se fija al mismo tamaño de la ventana.
- Al cambiar el tamaño de la ventana (procedimiento de actualización), ¿qué pasa con el tamaño del área de visualización?
- Ejercicio:
en el archivo `introOpenGL.cpp`, analizar el comportamiento al usar o no `glViewport`

Primitivas 2D

Primitivas 2D OpenGL

Elemento base de especificación:

```
glVertex* ( );
```

- Sufijos (*):
 - Dimensionalidad: 2, 3, 4
 - Tipo de dato: (i) entero, (s) entero corto, (f) real, (d) real de doble precisión.
 - Especificación vectorial (v).

Primitivas 2D OpenGL

- Primitiva punto:

```
glBegin (GL_POINTS);  
glVertex* (<coordenadas>);  
glEnd ();
```

Primitivas 2D OpenGL

- Primitiva punto:

```
glBegin (GL_POINTS);  
glVertex* (<coordenadas>);  
glEnd ();
```

- 2D entero:

```
glVertex2i (50, 100);
```

Primitivas 2D OpenGL

- Primitiva punto:

```
glBegin (GL_POINTS);  
glVertex* (<coordenadas>);  
glEnd ();
```

- 2D entero vector:

```
int point[] = {50, 100};  
...  
glVertex2iv (point1);
```

Primitivas 2D OpenGL

- Primitiva punto:

```
glBegin (GL_POINTS);  
glVertex* (<coordenadas>);  
glEnd ();
```

- 3D real:

```
glVertex3f (-78.5, 14.6, 188.33);
```


Primitivas 2D OpenGL

- Primitiva punto:

```
glBegin (GL_POINTS);  
glVertex2i (50, 100);  
glVertex2i (75, 150);  
glVertex2i (100, 200);  
glEnd ();
```

Primitivas 2D OpenGL

- Ejercicio:

Modificar el programa `introOpenGL.cpp` para dibujar puntos en varias posiciones.

¿son visibles los puntos?

Primitivas 2D OpenGL

- Propiedades de color:

```
glColor* ( );
```

- Sufijos (*):
 - Dimensionalidad: 3 (RGB), 4 (RGBA)
 - Tipo de dato: (i) entero, (s) entero corto, (f) real, (d) real de doble precisión.
 - Especificación vectorial (v).

Primitivas 2D OpenGL

- Propiedades de color:

```
glColor* (<R>, <G>, <B>, <A>);
```

- Valores:

0.0 (intensidad cero) - 1.0 (intensidad completa).



Primitivas 2D OpenGL

Elemento base de especificación:

```
glVertex* ();
```

- Propiedades de color:

```
glColor3f (1.0, 0.0, 0.0);  
glBegin (GL_POINTS);  
glVertex2i (150, 200);  
glEnd ();
```

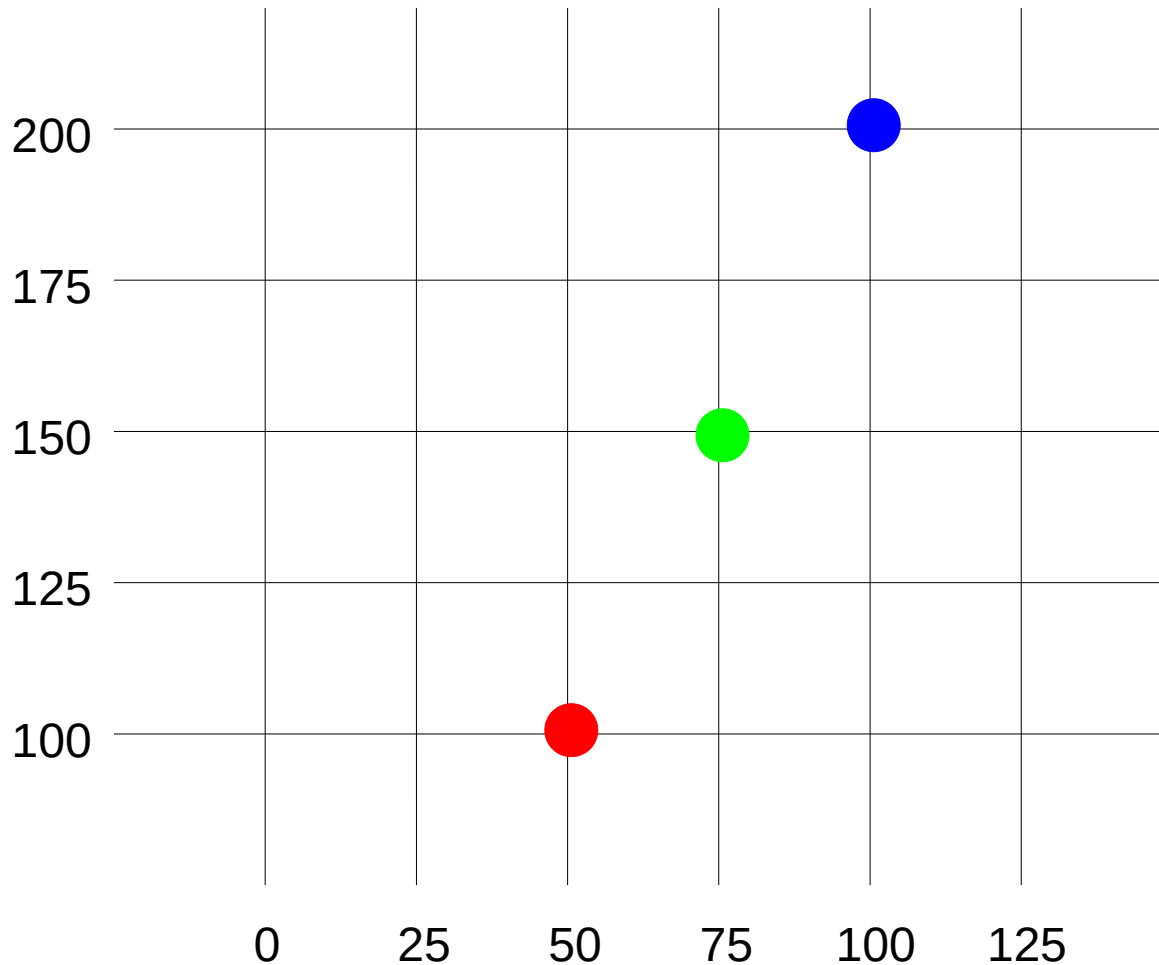
Primitivas 2D OpenGL

- Propiedades de color:

```
glBegin (GL_POINTS);  
glColor3f (1.0, 0.0, 0.0);  
glVertex2i (50, 100);  
glColor3f (0.0, 1.0, 0.0);  
glVertex2i (75, 150);  
glColor3f (0.0, 0.0, 1.0);  
glVertex2i (100, 200);  
glEnd ();
```

Primitivas 2D OpenGL

- Propiedades de color:



Primitivas 2D OpenGL

- Ejercicio:

Modificar el programa `introOpenGL.cpp` para dibujar puntos de diferentes colores.

¿son visibles ahora los puntos?

Primitivas 2D OpenGL

Elemento base de especificación:

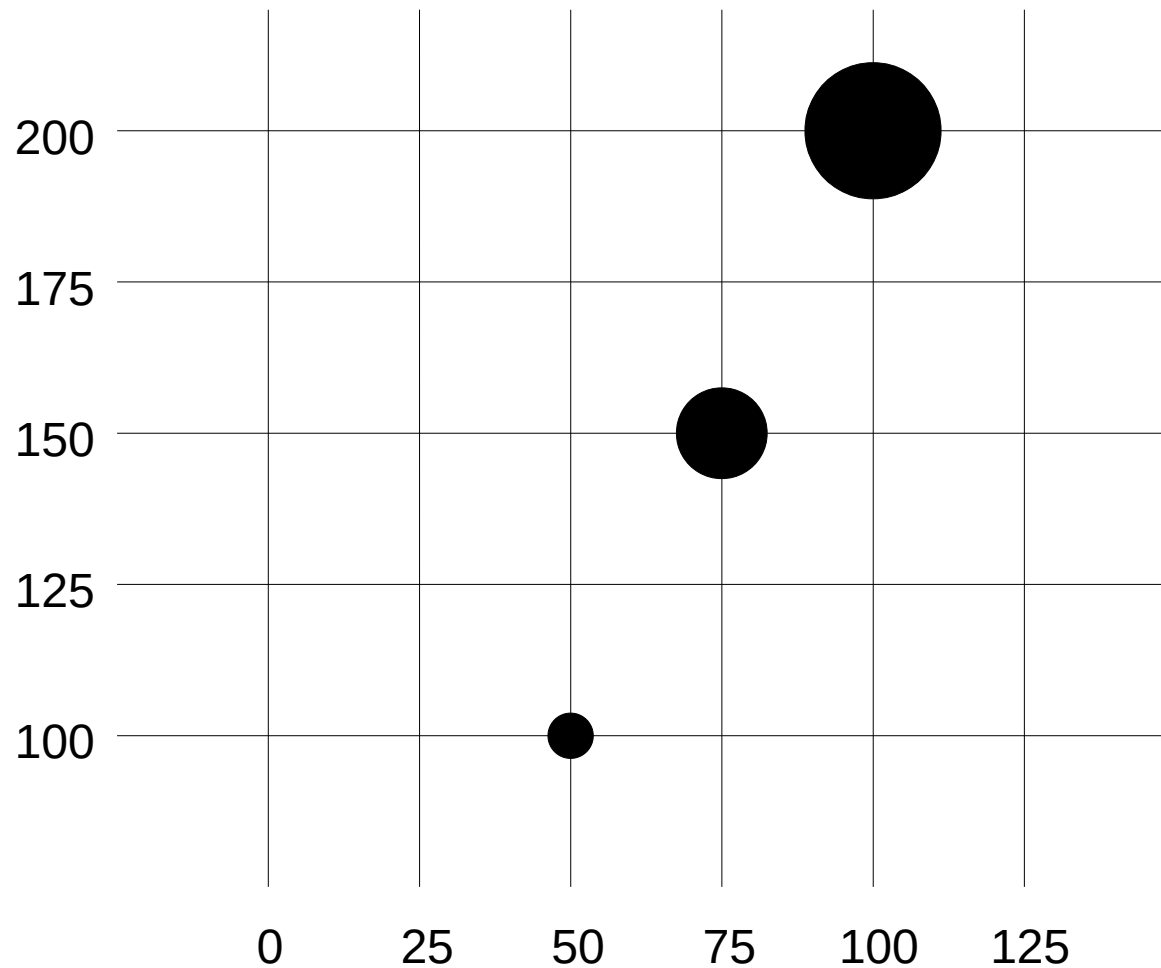
```
glVertex* ();
```

- Propiedades de grosor:

```
glPointSize (1.0);  
glBegin (GL_POINTS);  
glVertex2i (150, 200);  
glEnd ();
```

Primitivas 2D OpenGL

- Propiedades de grosor



Primitivas 2D OpenGL

- Ejercicio:

Modificar el programa `introOpenGL.cpp` para dibujar puntos de diferentes tamaños.

¿son visibles (ahora sí) los puntos?

Referencias

- D. Hearn, M.P. Baker. Gráficos por computadora con OpenGL, 3a edición. Pearson Prentice Hall, 2006.
- J.D. Foley, A. van Dam, S.K. Feiner, J.F. Hughes. Computer graphics: principles and practice, 2nd edition in C. Addison-Wesley, 1996.
- en.wikipedia.org/wiki/OpenGL
- www.opengl.org/about/
- www.arcsynthesis.org/gltut/
- www.cse.ohio-state.edu/~whmin/courses/cse581-2012-winter/2D.pdf
- www.sci.brooklyn.cuny.edu/~weiss/cisc3620/Lectures/02-Chapter02.html