



## **Graphics engines and their APIs**

**Code packages with a well-defined interface**

### **2D or 3D**

**2D: Usually pixel-graded coordinates**

**3D: Normalized coordinates, world-graded  
coordinated, scalable**



## **Features in a 2D API**

**Examples from two commercial 2D API's:**

**GDI (Graphics Device Interface)  
(Microsoft, MS Windows)**

**QuickDraw  
(Apple, MacOS)**



## Graphics environments

Information about pens, colors, clipping etc.

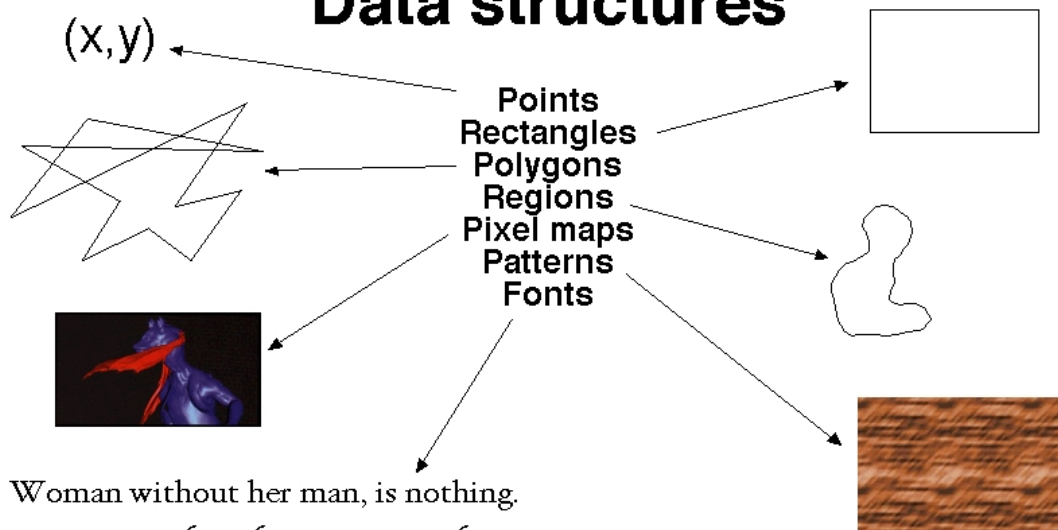
**GDI:  
Drawing Context (DC)**

**QuickDraw:  
GrafPort**

**OpenGL:  
The OpenGL current context**



## Data structures



Woman without her man, is nothing.  
*Woman! Without her, man is nothing!*



## Drawing operations

**Draw points, lines, rectangles, ovals, rounded-corner rectangles, polygons, regions.**

**Drawing variants applicable on one shape:  
frame, paint, fill, invert, erase**

**Draw text.**

**Bit blitting.**



## Metafiles

**A metafile is a sequence of recorded drawing commands. Not necessarily a disk file.**

**WMF, EMF, PICT, EPS.**

**WMF = Windows Metafile**

**EMF = Enhanced Metafile**

**PICT = Macintosh Picture**

**EPS = Encapsulated Postscript**

**In OpenGL: Display lists**



Information Coding / Computer Graphics, ISY, LITH

## **2D versus 3D**

**2D: Points, lines, text**

**3D: Polygons, surfaces, camera**

**2D has more different primitives**

**3D builds everything from polygons**



Information Coding / Computer Graphics, ISY, LITH

## **Features in a 3D API**

**Geometrical:**

**Polygons (triangles)**

**(Points and lines exist but are rarely used.)**

**Rendering:**

**Light sources, Textures, Shading**

**Camera specification**



Information Coding / Computer Graphics, ISY, LiTH

# OpenGL

**Portability: Excellent!**

**Performance: Good**

**Source: Yes, at least through Mesa**

**Language: Any**

**Fairy low-level, add scene graphs etc on top**



Information Coding / Computer Graphics, ISY, LiTH

# Direct3D

**Portability: Poor**

**Performance: Good**

**Source: No**

**Language: Any**

**Includes many utilities**



# Java3D

Portability: Good!

Performance: So-so

Source: No

Language: Java only

High-level, scene graph mandatory



*Do you need extreme low level graphics programming these days?*

*Mobile phones*

*Embedded purpose systems*

*Low-power devices*

*Mobile systems*

*Do you*

*Special,*

*Other future low*

*OpenGL ES helps on it*



Information Coding / Computer Graphics, ISY, LiTH

## The new low level: Shader programming

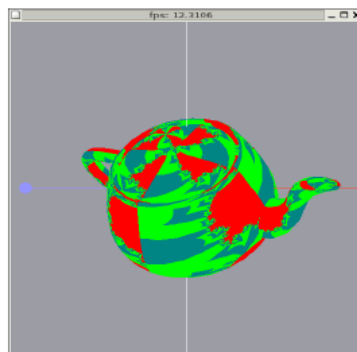
Modern GPUs support “shader programs”, short programs executed on the GPU. They allow more freedom in shading and texturing.

You will write your own in lab 3!



Information Coding / Computer Graphics, ISY, LiTH

## Shader programming example: Texture generated in real time, Mandelbrot fractal



Can't be done in real time on this computer,  
but I will bring one that can