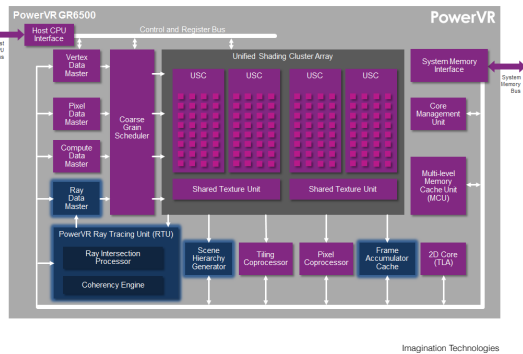
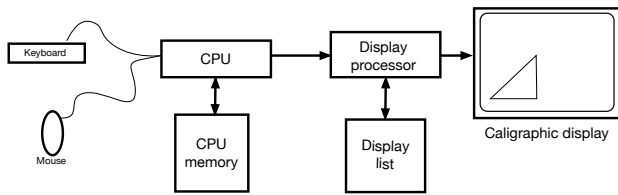


Graphics Hardware Overview



Early Hardware

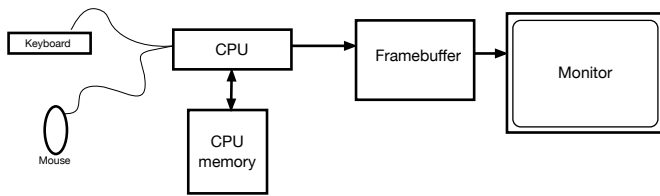


- retained mode
- vector display
- commands: moveto, lineto
- complex scene -> flicker

```
...
moveto 100 100
lineto 500 100
lineto 500 500
lineto 100 100
...
```



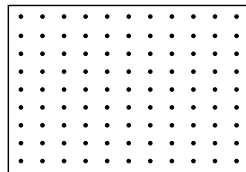
Early Hardware



- immediate mode
- CPU does all the work

Frame Buffer (Color buffer)

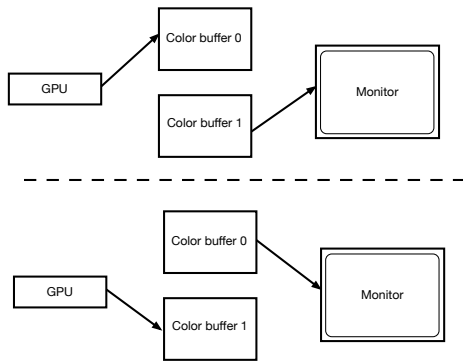
- 2-D array of pixels
- intensity/color sample (R, G, B)
- raster
- resolution
- bits per pixel: 1, 8, 24, 36
- integer/floating point
- refresh rate
- double/multiple buffers



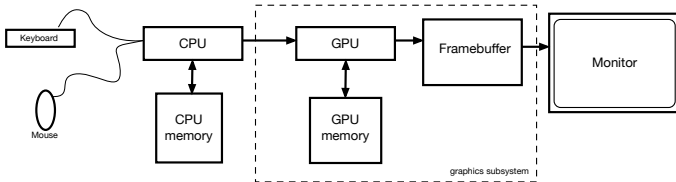
Pixel

- Red, Green, Blue triple (R, G, B)
 - three-dimensional space
- 0.0-1.0 or 0-255 or 0-4,095 or 0-65,535
- Red, Green, Blue, Alpha (R, G, B, A)
- Alpha
 - 0.0 = completely transparent
 - 1.0 = completely opaque

Double Buffering

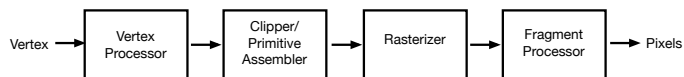


Hardware



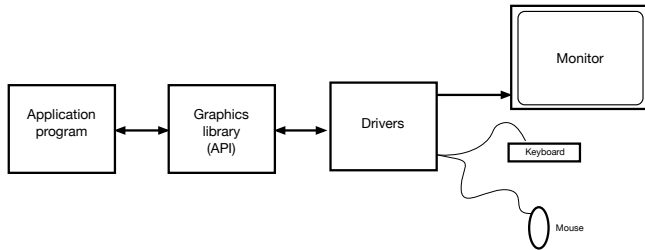
- immediate mode
- retained mode
- parallelism

Graphics Pipeline



- transformation, rasterization
- originally fixed-function VLSI
- pipeline, parallelism
- GPUs evolve -> more powerful, programmable
- vertex shaders, fragment shaders

Graphics API_{Interactive}

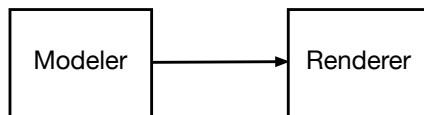


Graphics APIs

- IRIS GL: SGI hardware only 1980s
- OpenGL: cross-platform 1992
evolved, powerful, immediate + retained mode, powerful GPUs, shaders
- OpenGL ES: embedded systems 2003
constrained devices, simpler, shaders
primitives: points, lines (+++), triangles (++)
- WebGL: OpenGL 2.0 ES for browsers 2011
javascript, GLSL
- Vulkan: high-performance successor to OpenGL 2015
simpler API, more direct control of GPU, designed for modern architectures
competitors: Direct 3D, Metal

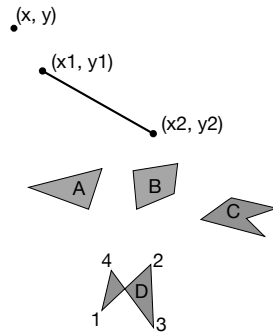
Graphics API_{non interactive}

- eg. RenderMan
- Server farms



Graphics Primitives

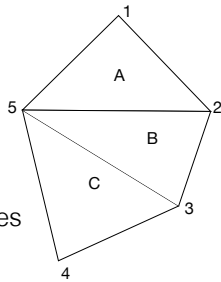
- points
- lines
- polygons
 - convex
 - concave
 - planar
 - simple
- curves/surfaces



Polygonal Meshes

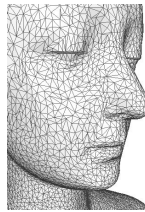
Polygonal Meshes

- List of vertices
- List of polygons
 - each has list of vertices



V: 1, 2, 3, 4, 5

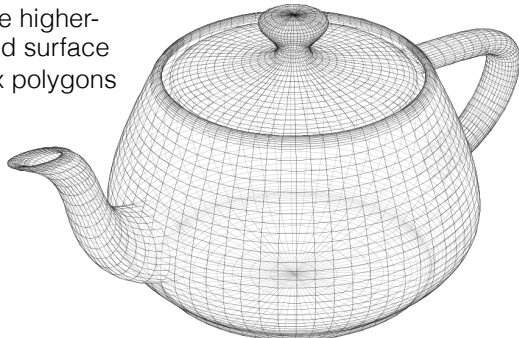
P: A(1, 2, 5), B(2, 3, 5), C(3, 4, 5)



© Ken Power 2011

Tessellation

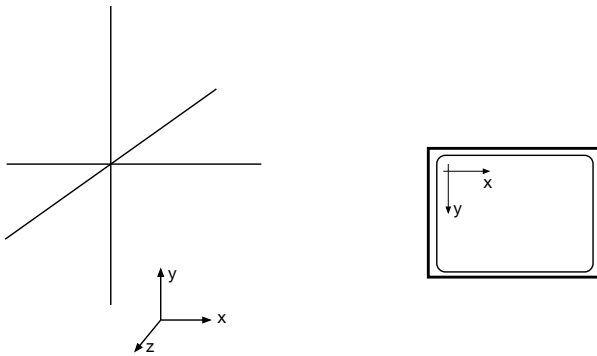
approximate higher-order curved surface with convex polygons (tri, quad)



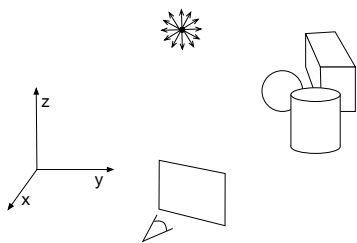
Coordinate Systems

- Model/Object Coordinate System
 - where you define object
- World Coordinate System
 - where objects are placed relative to each other
 - R^2 or R^3
- Screen Coordinate System
 - device specific coordinates

Coordinate Systems



What do I See?

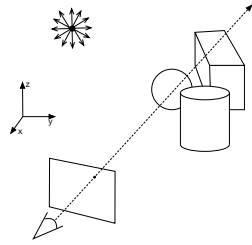


How?

- Ray tracing (screen to world)
- graphics pipeline (world to screen)

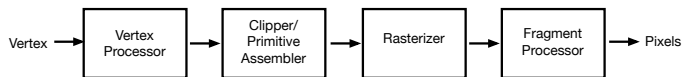
Ray Tracing

- shoot ray from eye through screen into world
- intersect objects with ray
- find closest intersection
- do shading/lighting calculation
- very floating-point intensive





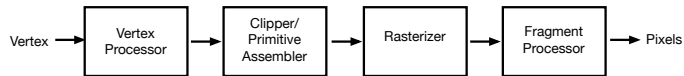
Graphics Pipeline



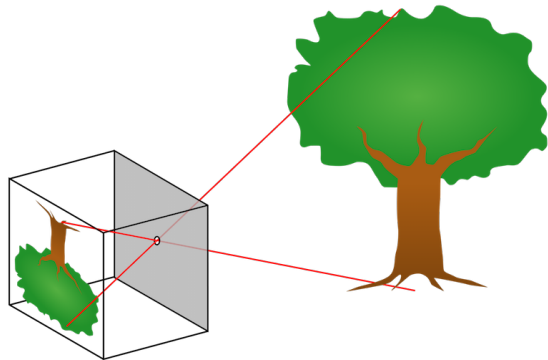
- world to screen
- hardware accelerated
- interactive

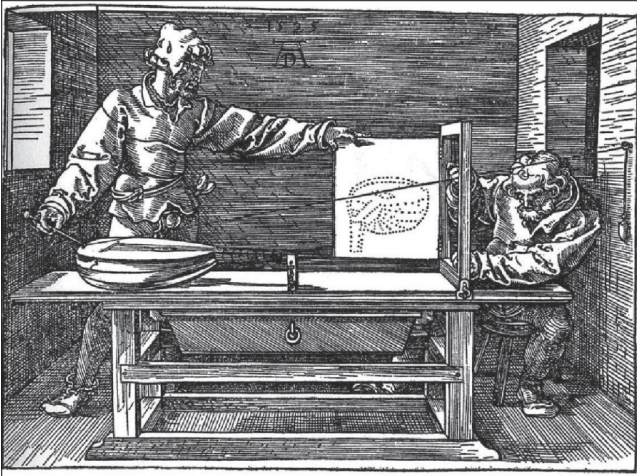
World to Screen

- transform and project graphics primitives onto projection screen
- determine what's inside (clipping)
- determine what's visible
- break down into pixels
- shade appropriately



Pinhole Camera





Wikipedia

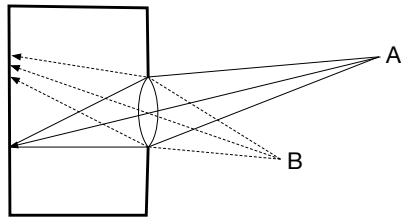
Pinhole Camera



Deviant Art

Camera with Lens

- more light
- focus
- depth-of-field



Graphics API

Major tasks:

- specify objects to be viewed
- specify properties of these objects
- specify how these objects to be viewed

WebGL

- HTML5 + WebGL
- Javascript + GLSL ES 2.0 (OpenGL Shading Language)
- Vertex and Fragment Shaders

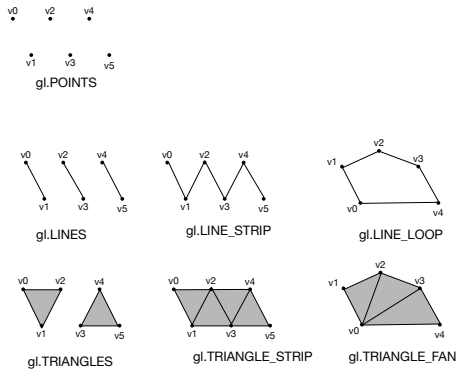
Dividing Work

- HTML5:
 - performs interactive tasks
 - event queue
 - callback functions
- WebGL:
 - 3D graphics
 - graphics pipeline

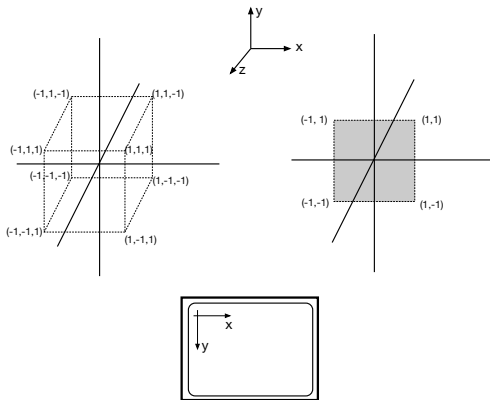
Shaders

- **Vertex Shaders:** programs that describe the traits (position, colors, and so on) of a vertex. The vertex is a point in 2D/3D space, such as the corner or intersection of a 2D/3D shape.
- **Fragment Shaders:** programs that deal with the per-fragment processing such as lighting. The fragment is a WebGL term that you can think of as a kind of pixel and contains color, depth value, texture coordinates, and more.

WebGL Primitives

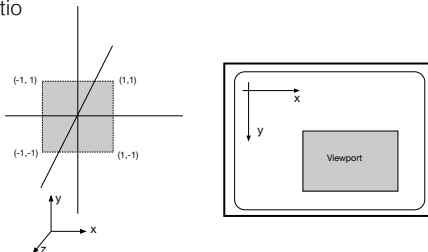


WebGL Defaults



Viewport

- Viewport specifies where on the screen the window will appear
- `gl.viewport(x, y, width, height)`
- Aspect ratio



A1: Koch Curve

- Your job is to write a simple interactive program that draws the recursive Koch curve.
- on prism `/eecs/course/3431/koch` contains skeleton