Today’s Outline

OpenGL introduction
  OpenGL primitives
  Demos / code
Rasterization rules
The OpenGL graphics pipeline
Graphics machines (hardware)
Demonstrations
OpenGL Shape Primitives
Pixel Coordinates

(0,0) (1,0) (0,1) (1,1)
Pixel Coordinates

OpenGL: Pixel centers correspond to non-integer coordinates
Rasterization Rules: Area Primitives

Output fragment if pixel center is *inside* area
Fragments vs. Pixels

Combine fragment color with pixel color and store in framebuffer
Rasterization Rules: Lines

Output fragment if line intersects diamond
OpenGL Drawing Primitives

Geometric Shapes

Bitmaps

Images
OpenGL Architecture
Simplified Pipeline

- Vertices
  - Per-vertex ops
    - Rasterizer
      - Per-fragment ops
        - Frame buffer ops
          - Frame buffer ops
            - Pixels
  - Texturing

- OpenGL commands
Modern PC

3.0 Ghz Intel Core2 Duo
Core 1
Core 2
4MB L2 Cache

1GB main memory (DDR2)

NVIDIA GeForce 8800 GTX (575 MHz)
(16 cores)

512MB video memory (GDDR3)

NVIDIA 8800GTX

System board (Intel D975)

PCIe Bus (max 4 GB/sec)

12.8 GB/sec

84 GB/sec
Game Machines – Xbox 360

- 3.2 Ghz PowerPC CPU
  - Core 1
  - Core 2
  - Core 3
- L2 Cache
- 500 Mhz ATI GPU
  - 48 3D Cores
  - Frame buffer
  - Video out
- 512 MB memory
- IO Chip
- Display (TV)
- controllers/ethernet/audio/DVD/etc.
Game Machines – PS3

3.2 Ghz Cell
- PPC Core
- SPU0
- SPU1
- SPU2
- SPU3
- SPU4
- SPU5
- SPU6
- SPU7
- L2 Cache

3.2 Ghz Cell
- 550 Mhz NVIDIA RSX GPU
- Multiple 3D cores
- Video out

Controllers/ethernet/audio/DVD/etc.

IO Chip

256 MB Memory (XDR)

256 MB video Memory (GDDR3)

Display (TV)
Modern Hardware Pipeline

- OpenGL commands
- Programmable processing cores
  (runs vertex and fragment programs)
- Texturing
- Frame buffer
- Textures

Programmable processing cores

Memory
Summary

Graphics state stores attributes
Graphics commands output primitives
  Shapes (points, lines, triangles, quads, etc.)
  Bitmaps (fonts)
  Images
Different coordinate systems
  User coordinates
  Pixel coordinates
Graphics system produces fragments from primitives
Demos: use simple GL commands and use GLUT for windowing/interaction