Development of Computer Graphics

- 1951
  - Whirlwind, Jay Forrester (MIT)
  - CRT displays
  - SAGE air defense system
  - command & control CRT, light pens
- mid 1950s
  - Computer Art, James Whitney Sr.
  - Visual Feedback loops
- late 1950s
  - Sketchpad, Ivan Sutherland
  - data structures, light pen for drawing and choices

Development of Computer Graphics

- Early display devices (mid-60s):
  - Vector, stroke, line drawing, calligraphic displays
  - Architecture of Vector Display

Development of Computer Graphics

- Architecture of a vector display - random scan
  - vector generator converts digital coordinates to beam deflections

Development of Computer Graphics

- 1962
  - Sketchpad, Ivan Sutherland
  - data structures, light pen for drawing and choices

Development of Computer Graphics

- 1964
  - CAD and CAM
  - General Motors DAC, Itek Digitek for Lens Design
- 1964-1970s
  - Photorealism at University of Utah
  - Sutherland, Evans, Catmull, Blinn
- 1968
  - Evans & Sutherland
  - commercial company - flight simulators
  - 3D vector pipeline, matrix multiplier, clipping
- 1969
  - First SIGGRAPH

Development of Computer Graphics

- 1970
  - Pierre Bezier - Bezier curves

Development of Computer Graphics

- 1971
  - Gouraud Shading
Development of Computer Graphics

1974-1977
- Catmull - Z-buffer
- Bui-Toung Phong creates Phong Shading (Utah)
- Martin Newell’s teapot (Utah)
- Computer graphics at NYIT - computer animation
- Raster Graphics (Xerox PARC, Shoup)

Architecture of a raster display

Architecture of a raster display - raster scan
- beams (3 beams) intensity set to reflect pixel intensity
- scan speed: originally 30Hz now 60Hz

Raster Scan
- need to store whole image
- 1024 x 1024 x n - n bits per pixel
- mono 1 bit, color 8 (256 color), 24 (16 million)
- 32 to 96 bits used (double buffering, z-buffering)
- 1280x1024x24 needs only 3.75 MB video RAM

Random Scan versus Raster Scan
- note ragged lines

Random Scan versus Raster Scan
- Raster advantages:
  - low cost, superior fill ability, refresh rate independent of complexity, 70Hz sufficient to avoid flicker
- Raster disadvantages:
  - discrete nature of pixel representation, need for scan conversion in software or RIP chips
  - real-time dynamics more demanding
  - approximation of lines by sequence of pixels
  - aliasing - jagged or staircasing
  - manifestation of sampling error in signal processing
  - need for anti-aliasing
Development of Computer Graphics

1976
- Image and texture mapping (Blinn)

1977
- 3D Core Graphics System, first "standard" for device independent graphics package
- Allowed portable graphics programming
- ACM SIGGRAPH committee including Foley, Van Dam, Feiner
- Baseline specification - many implementations

1978
- Image and texture mapping (Blinn)
- 3D Core Graphics System, first "standard" for device independent graphics package
- Allowed portable graphics programming
- ACM SIGGRAPH committee including Foley, Van Dam, Feiner
- Baseline specification - many implementations

1979
- IRIS - Integrated Raster Imaging System, SGI
- High-end workstation
- Hardware acceleration of graphics pipeline

1982
- TRON - "non-realism" and relatively low technical quality special effects
- Star Trek - Genesis Effect; Lucasfilm's computer graphics division (later split into Industrial Light and Magic, and Pixar)
- Used key technical effects (such as particle systems and caustics)

1982
- Clarke, Geometry Engine
- Hardware support for transforms (matrix-vector multiplies), clipping (variant of Sutherland-Hodgman algorithm)
- IRIS - Integrated Raster Imaging System, SGI
- High-end workstation
- Hardware acceleration of graphics pipeline

1982
- Ray Tracing, Turner Whitted
- Good at rendering reflections, refractions and shadows

1983
- VRAM, Video random access memory, Texas Instruments
- Can read out all pixels in one memory cycle

1983
- Fractals
- Allowed generation of the key components of natural-looking landscapes

1985
- Radiosity, Don Greenberg (Cornell)
- GKS, Graphical Kernel System
- First ANSI standard
- Elaborated cleaned up version of CORE but only 2D

1986
- Renderman - an extensible 'procedural language' for controlling the animation/rendering process

1988
- PHIGS, PHIGS+
- Programmer's Hierarchical Interactive Graphics System
- More complex than CORE
PHIGS v GKS

- GKS allowed grouping of primitives into “segments”
- no nesting of segments
- PHIGS allowed nested hierarchical grouping of 3D primitives into “structures”
- all primitives subject to geometric transformations
- editable database of structures
- auto-update of screen when database altered

PHIGS+
- extension for pseudo-realistic rendering on raster devices
- PHIGS, PHIGS+ large packages
- run best with hardware support of transformations, clipping and rendering

Development of Computer Graphics

- 1993
  - OpenGL - Open Graphics Library
  - derived from SGI’s GL library
- 1993
  - Open Inventor, OO layer on OpenGL
- 1995
  - QuickDraw 3D, Apple
- 1995
  - Direct3D, Microsoft, game playing API

Input Devices

- early light pens to modern mice
- data tablet
- touch sensitive screens
- 3D input devices (spaceballs etc.)
- button and dial boxes

Describing Scene to be viewed

- Application Program - creates application
- Application Model - independent of display system
  - program must extract geometry and convert to primitives of graphics system
  - primitives: points, lines, rectangles, ellipses, text, polygons, polyhedra, spheres, curves, surfaces
  - application must convert geometry to primitives supported
  - attributes (line style, color, line width, fill style)

Graphics Systems

- Typically libraries: output subroutines
  - user programs in logical display device terms
  - graphics library converts to device dependent instructions
  - abstraction of display device
    - locator - mouse, tablet, joystick etc.
  - sample - return from locator
  - event - generated by user input

Interaction Handling - event driven loop

```java
while (!quit) {
    enable selection of commands/objects
    wait for user selection
    switch (selection) {
        process selection, updating model and screen as necessary
    }
}
```

User interaction

- change in screen appearance - does not involve update of model: application updates state and calls graphics package
- change in model: must recalculate