1996-10-28 Bo Nordin (bosse&cb.uu.se)

Lecture notes, computer graphics (F01) (Foley 1.1-1.4, some from Foley 10)

Agenda:
Practical questions (language!)
Introduction & overviw
AVS overview

Practical questions

Teacher:	Bo Nordin 183470
Goals:	theoretical and practical understanding of principles and methods used in computer graphics today. knowledge about hardware and graphics standards. (improved understanding of terms in CG and perhaps of computer technology in general).
Prerequisites:	Mathematics (~20p), Scientific programming MN2 (5p) (15p TDB) Numerical analysis MN2 recommended (5p),
Desirable:	Familiar with UNIX, knowledge of C++ (C), Makefiles, (into lab??) Familiar with dpt. of Scientific Computing (computers always available, doors locked (cards from UDAC -ask at office).
Literature:	1. Foley et al: Introduction to Computer Graphics (mention MN2) 418 kr - LundeQ, 425 kr - Studentbokhandeln 2. Hand-outs, (3. your notes)
Mandatory exercises:	AVS (introduced today, returned before 2/12) Programming in PHIGS (introduced 13/11, returned before 30/1) Programming a GUI (introduced 14/11, returned before 9/12) Programming in Motif (introduced 14/11, returned before 16/12) 1,2, or 3 students per group.
Useful info:	Gula Luntan (TDB computer systems information)
Lectures:	will use the blackboard most, OH sometimes
Examination:	Written exam 17/12 and accepted assignments. Exercises shown "live", all students must be prepared to "defend" the code (explain, etc.) orally (and perhaps hand in print-outs).

This is an overview course, will give you some knowledge about computer graphics basics. Won't discuss virtual reality & animation (at least not in detail). (Discuss MN2)

Goals (Interactive Computer Graphics) (stress all 3 words)

Knowledge of basic technologies (hardware a little) including input
Limitation (hardware imposed and other)
Familiarise you with some standard graphics packages (from a programmer's p-of-view)
Human Computer Interaction issues, GUI
Learn how to program standard graphics packages

Computer knowledge in general (C++, UNIX)
Overview of advanced topics

Implementation (how do we reach the goals)

Lectures for overview
Book and hand-outs (important issues in boldface in the schedule ?)
(Individually solving problems during the course)
No laborations with teacher planned (wishes??)
Programming tasks
Visit at CBA (animation)
(Show the schedule and group lectures)
Information ~grafik & http://www.cb.uu.se/~bosse/graphics
mail (obtain e-mail addresses from all students)

Introduction

(OH with how image analysis and computer graphics are related).
What is computer graphics?

```
Computerized creation of images and graphical information.
```

Usage:

```
User interfaces
Business (2D, 3D graphs)
Games
Cartography (accurate, schematic reprs. of geography, etc.)
Medicine ( MRI Images – often in conjunction with Image Analysis)
Art
CAD (computer aided drafting/design) ( cars, aeoroplanes, ships)
Simulation for scientific visualization (nucl. reaction, flight sim.)
Animation for entertainment (movies)
Multimedia
Virtual reality
```

How do we produce computer graphics?

```
Input data description (image) -> program -> output
```

What's input?

```
A file and algorithms (batch)
Interactive data generation
```

How is it presented?

```
Usually on a screen, but sometimes paper hardcopies are required.
```

Image analysis vs. computer graphics

```
Graphics create an image – the image is the result
In IA the image is input, output varies (but display of images is CG)
```

History of computer graphics

The course concentrates on *fundamental* principles and techniques, applicable yesterday, today, and tomorrow.
Computer graphics is not an old "science". Rapidly evolving area of computer science.
Early 50's: teletypes, line printers (try to find an example of a cell).
Landmark: Ivan Sutherland's thesis 1963: Sketchpad, data structures, algorthims, interaction
stand. comp (primitives), keyboard / light pen
Raster vs. vector (vector displays dominated until ~1980)
Until early 80's a small specialized field. Then along came Mac, PC. Batch -> interactive

(pointing-and-clicking) desktop, window, mouse
ANSI/ISO (software crisis): software portability: the Core, GKS, PHIGS, X

1951 1st graphics display	1
1964 A few graphics applications $100 000	100
1968 1st cheap screen "storage tube" $5000	1000
1978 Raster screens begin their domination	100 000
1984 MAC 1st graphics-based PC	1 000 000
1994 Graphics on all compuers	70 000 000

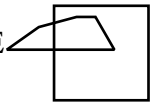
Increase: * 100 per decade

Topics in CG, course overview
Performance, hardware: L2

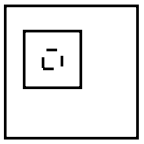
Software, standardization:

Humal-computer interaction: input techniques (hardware & software)
not only a technical problem

2D graphics: transformations, limited in X, Y clipping

DRAW SMALL FIGURE  that illustrates clipping

3D graphics: limited in X, Y, and Z: clipping, view position, perspective, hidden lines and surfaces, illumination, shading, reality
2D screen -> 3D transformed to 2D for display

DRAW SMALL FIGURE  to illustrate prjection, hidden lines
but preferrably a cube or something

Complicated objects: curves, surfaces (Bezier, B-spline)
Realism: modelling, illumination, colour, shading, volume visualization, rendering, visible-surface determination
Usage: many, visits at CBA

Levels of sophistication
Line drawings wire-frame models -> several interpretations (fast - exists in SPHIGS)
a set of lines does not necessarily describe a closed area
Polyhedra (surfaces) solid objects (chapter 10) (shaded)
a set of surfaces does not nesessarily describe a cloes volume
Spatial-occupancey enumeration (voxels)

Interactive Computer Graphics Systems
Overview (from Burger/Gillies, page 8):

Application program	(your program, with an application model)
Functional level	(high-level support: draw diagrams, show image)

Procedural level (GKS/PHIGS functions)
Device level (drivers for hardware, "intelligent" hardware, OS functions)
display, plotter mouse,
printer keyboard
Hardware

Compare this with the schedule!! (lect2-3: hardware, procedural level,
Also comment on that lecture groups *Basic Implementation Techniques, Raster graphics* (fundamentals), curves & surfaces, realism, Visualization & Animation concerns what's **inside** graphics packages. The first two in particular - the rest could be regarded as application issues. The lectures (and exam) concentrates on the guts of CG packages while the exercises (and the lecture groups Introduction,Graphics Standards) is about using packages.

Overview (from Foley, page 15):

application model (database)	application program	graphics system	device level	hardware

Model (database): primitives (line, point, curves, polyhedra) *attributes, connectivity*
Display - set of functions corresponding to the primitives. (cf. retained/immediate)
Interaction - event-driven loop

Graphics primitives
There are graphics primitives for output and input.

AVS overview
The purpose of the exercise is to make you understand what can be done with computer graphics, a little about how it is done (more about that later in the course), and also to make you curious (look things up in the book).

AVS (Application Visualization System) is a program (system) for visualization originally developed by Stardent computer (see the sleeve of the book) but now available on a multitude of platforms. AVS is written in PHIGS (see inside leaf of the book).

(INFO FROM AVS exercise paper)

pixel-based visualization vs. *geometry-based* visualization

pixel-based simple - raw value -> translated to a colourt table lookup (colormap)

AVS colourmaps are independent of hardware colourmaps. 24-bit "true colour"

geometry: spheres, triangles, lines

modules: general, flexible, can be used in a variety of applications
networks:

Explain Geometry Viewer: