Outline

- History of computer graphics.
- Conceptual models for computer graphics.
- Briefly introduction to 3D graphics.
- Geometric modelling.

History of Computer Graphics

- 1950: Whirlwind Computer - MIT used a vector Cathode Ray Tube (CRT) display for output.
- Mid 1960's: Computer Aided Design (CAD) and Computer Aided Manufacturing (CAM) systems where being used.
- The 1970's: development of television technology - cheap raster displays.
- At the same time colour systems became more common.
- Early 1980's: the advent of the personal computer, with built in raster display capabilities.
- Lead to widespread adoption of bitmap and interactive graphics.

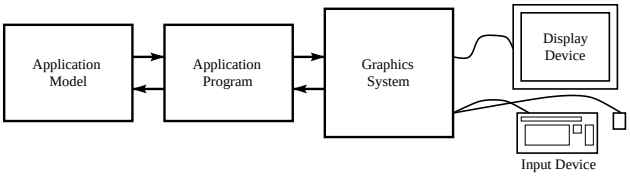
History of Computer Graphics

- As the hardware has developed, software has also changed.
- Development of Graphical User Interfaces (GUIs) allowed novice users to access a large variety of applications.
- Computer screen became the electronic 'desktop'.
- The first graphics specification to receive an official standard (in 1985) was the Graphics Kernel System (GKS).
- Provided a high level 2D graphics standard.
- Complemented (in 1988) by GKS-3D

History of Computer Graphics

- 1988: the Programmer's Hierarchical Interactive Graphics System (PHIGS - pronounced figs)
- Allows a nested hierarchical grouping of 3D sub-primitives called structures.
- 1992: an extension PHIGS PLUS included pseudo-realistic rendering.
- Now several 'standards': OpenGL (Silicon Graphics), X Windows System, PostScript (Adobe) and Direct 3D (Microsoft).
- Many of the functions in these graphics specifications are supported by hardware.

Conceptual models for Computer Graphics



- The software part has three components:
  - the application program;
  - the application model;
  - the graphics system.

Conceptual models for Computer Graphics

- The application program handles the exchange of data between the application model and the graphics system.
- The application model represents the data or objects to be visualised on the display device.
- The graphics system produces the output to drive the display device and parses inputs.
- The design of interactive graphics application programs centres around the definition of the data items and objects in the application model.

Application models

- The form that the application model takes will depend on the aim of the application program.
- E.g. a spreadsheet will store the application model in arrays.
- The application program will then have at least two graphical aspects: the display of the primary data and graph based visualisation of the data – which will typically have its own application model.
- The application models we shall consider, store graphics primitives, such as points, lines, curve, polygons (2D or 3D) and polyhedra and surfaces (3D).
- Might also include attributes and connectivity relations.

Displaying the application model

- The application program converts the data in the application model to commands used in the graphics system to produce a view of the application model.
- Typically done interactively.
- Either creates and stores a geometrical representation of the application model, or does it on the fly as it is needed.
- First the application program queries the application database to extract those parts of the application model required for the desired view.

Displaying the application model

- This data is then converted into a geometrical description (if necessary) and sent to the graphics system.
- The primary job of the graphics system is to manage input and output between the user and the application program.
- In this course the graphics system will be OpenGL, while we will write the application model and application program.
- Our application models will be quite simple.

What's in the application model

- In this course it will be geometric objects.
- Composed of points, lines and polygons.
- Next we consider what geometric modelling means.

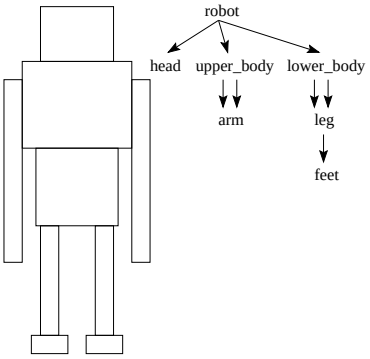
Geometric modelling

- Modelling is a very familiar concept to computer scientists.
- We use models to represent objects, processes and abstract ideas in a way which makes understanding more simple.
- More directly, computer graphics might be concerned with different types of models such as:
  - organisation models - hierarchies, flowcharts; directed graph representations.
  - quantitative models - graphs, maps.
  - geometric models - engineering and architectural structures, chemicals, people, real world objects.

Geometric modelling

- Geometric models describe the geometry of the objects which they represent! This includes:
  - spatial layout and shape of the component parts of the object (geometry),
  - connectivity of the component parts (topology),
  - attributes (which affect the appearance),
  - attributes (which pertain to the object but do not affect appearance).
- We often use hierarchical constructs to help store geometric models.

A hierarchical description of a robot



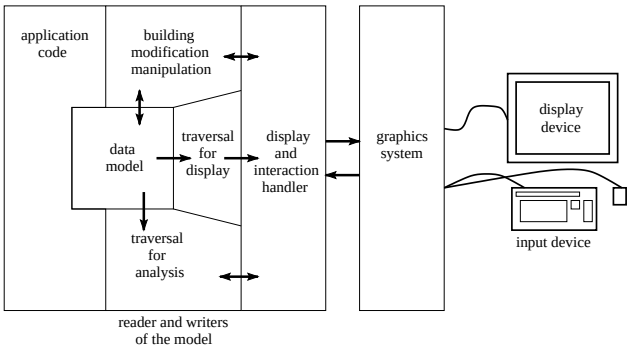
Geometric modelling

- The hierarchy will start with base components and combine these to form successive levels of objects, in a manner similar to programming approaches.
- Can be symbolised using a Directed Acyclic Graph (DAG)
- If each object appeared only once in the hierarchy the resulting data structure would be a tree.
- The DAG may include details of the topology such as specifying where the objects are attached (or equivalently about which axes the objects can rotate or translate).

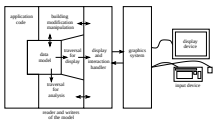
Geometric modelling

- Object hierarchies are useful because:
  - complex models can be constructed in a simple modular fashion,
  - stored efficiently and
  - updated simply (the updating of one level in the hierarchy automatically updates elements below it).
- In OpenGL hierarchies can be built up using display lists (more on that later).

The application model revisited



The application model revisited



- The application program is composed of several subsystems which have variable degrees of access to the application model.
- In many industrial application the 80/20 rule is generally true:
  - 80% of the program deals with modelling objects (the database) and interaction,
  - 20% deals with producing the pictures.

Retained versus Immediate mode packages

- Retained mode graphics packages store the model details and draw these when necessary.
- A record of all the primitives exists in the application model which allows automatic updating of the screen and simple editing of the primitives.
- Immediate mode graphics packages draw directly to the screen buffer (lower level).
- The effects on the screen are stored, not the generating primitives.

OpenGL

- OpenGL is an immediate mode graphics package.
- Many additions to OpenGL have been written to allow the user to treat it as a retained mode graphics package – these are often called scene graph description languages.
- Immediate mode packages give the user greater control over the drawing process and can thus be more heavily optimised.
- Immediate mode packages most often used when speed / control / flexibility is important.
- At the highest level of abstraction a retained mode graphics library might use descriptions such as chair, house and tree.

Introduction to viewing in 3D

- Most of the objects that will be stored in the application model will naturally exist in either:
  - 2D (plans, cross-sections, simple graphs),
  - 3D (real world objects, more complex graphs).
- Since the viewport is currently a 2D representation of whatever is in the application model, 3D coordinate systems call for a little extra work.
- We need to define the 2D projection of the 3D objects.
- First we consider defining and manipulating objects in 3D.

Summary

- Having finished this lecture you should:
  - be able to describe the components of a graphics system;
  - understand their roles in processing graphical data;
  - discuss the different types of models used in computer graphics;
  - contrast the advantages and drawback of retained versus immediate mode graphics packages;
  - explain where OpenGL fits into the equation.