Software and Computer Graphics
CMP 80c, Winter 2001

Jane Wilhelms, wilhelms@cse.ucsc.edu

1 Basic Computer Software

In previous notes we talked about computer *hardware*, the actual physical parts of the machine. What makes computers special is really their *software*, programmable instructions that make it possible for a computer to accomplish an enormous variety of tasks. Computer software (*computer programs*) provides detailed instructions on what a computer should do.

These programs are written by *computer programmers* in special *computer languages*. There are many standard languages. Some of the most well-known languages are *C* (commonly used on Unix workstations and the most popular among graphics programmers), *Pascal* (used mainly for instruction), *Fortran* (an old language mainly used for scientific programming, but being superseded by C), *Cobol* (an old language common for business programs), *Lisp* (developed for artificial intelligence purposes), *Basic* (elementary language used by non-professional programmers), and *Java* (developed for use on the internet but now a popular general purpose language).

These languages are somewhat like a cross between English and mathematics. They are close enough to English to be fairly easy to learn, and even non-programmers can probably get a general idea of what the program does by looking at the program (sometimes called the *code*). The vocabulary is very limited and words and other symbols must be used in exactly the correct way.

For example, a simple *C* program that adds two numbers and prints the result is

```
main()
{
int a,b,sum;

a = 10;
b = 20;
sum = a + b;
printf("sum is %d\n",sum);
}
```

When it is *run*, it prints out

```
sum is 30
```

Computer programs written in a standard language are the same (or very nearly) all over the world, so that programs can be shared and re-used. In fact, computer programs should be written in *modules* (also called *procedures* or *subroutines*) that include a part of the program that does a particular thing. This is rather like a book consisting of chapters. Subroutines are flexible and can be re-used between different programs.

The same programs generally run on very different kinds of computers, such as Unix workstations, personal computers, supercomputers, etc. Each different type of computer has different hardware. Before the program written in a particular computer language can be used by the computer, it must be converted to a language closely tied to the kind of computer. This *machine language* isn't at all like English. The process of taking a program from a standard programming language and converting it to machine language is called *compiling*. It is done automatically by a special program called a *compiler*.

The result of compiling is an *executable*, that is, a program that is runnable by the machine. For example, *AdobePhotoshop* is an executable program that was created by compiling some very complex software written in a programming language. Basically, all the commands you have been using, like *jot*, *cp*, *mail*, etc., are just compiled and executable versions of programs written by people in standard programming languages.

2 Operating Systems

An operating system is a set of programs that come with the machine and provide many basic utilities for using and communicating with it. *UNIX* or simple variations of UNIX are common operating systems on workstations such as *SUN SPARCstations* and *SGI Graphics workstations*. Commands such as *ls, cp, mv, rm* that allow you to list all the files, copy them, move them or remove them are part of an operating system. *Windows* and *Linux* are common operating systems on personal computers (*PCs*). Operating systems also take care of scheduling problems when many people are trying to access the same resource. I.e., if more than one person are logged into a machine and trying to read their mail at once, the machine has to process multiple commands from different users. You will not have to worry about who is getting to the computer first because operating systems do this job for you. Operating systems also ensure that you can use only the amount of disk space allocated to you.

3 Editors and Other programs

An editor is a program that allows you to write a letter, a paper or any text. Examples of editors are *vi*, *jot*, *emacs*, *notepad*, and *word*.

Two other popular programs are *SpreadSheets* and *Databases*. Spreadsheets allow you to create, load, view and manipulate data and text that are in a tabular format. Databases allow you to create, store, access and retrieve information about particular types of data. For example, a library database allows you to retrieve information about library holdings. We will *not* be discussing spreadsheets and databases in this course.

Instead, we will be focusing on *graphics software* that allows you to paint, draw, model, render and animate images. There are thousands of graphics software packages available in the market. We will focus on *Lightwave* because it is one of the 3D modeling, rendering and animation packages.

4 The Development of Computer Graphics

Technology for computer graphics was developed in the 1950s but largely for use by the military or manufacturing, e.g., flight simulators and computer-aided design and manufacturing (CAD-CAM). Graphics on more general-purpose computers was rudimentary, and usually specialized displays were necessary because terminals were only capable of showing ASCII text.

This gradually changed in the 1970s and 1980s. Initially, most graphics were *line drawings* and used special *calligraphic* CRTs where the electron gun traced out random lines anywhere on the screen. Gradually, color

raster CRTs took over, where the image is created by coloring a fixed grid of *pixels* that is always scanned in the same order, whatever image is displayed.

The first interactive graphics system was *Sketchpad* in the early 1960s by Ivan Sutherland at MIT. This did simple line drawings of objects using a light pen. In the 1960s, more graphics algorithms were developed, but machines for graphics were very expensive. Little software was generally available for graphics.

In the 1970s and, especially, the 1980s, both hardware and software was developed that made computer graphics more widely available. In the 1980s, Silicon Graphics workstations using a special hardware *geometry engine* made advanced graphics capabilities available at a reasonably cost. In the 1990s, with PCs, the internet, and video games, advanced graphics has become common on all computers. Specialized graphics boards, fast processors and memory, and large memories and disks are important for best performance, though.

Commercial software for graphics has also become commonplace. Modeling and animation software such as *Maya* from Alias-Wavefront, *Softimage*, *Lightwave*, and *3D Studio Max* makes sophisticated 3D modeling widely available.

The text and the text web site (see class web page) has an interesting timeline of developments in computer graphics.