Technological Literacy, Creativity and Electronic Art: "Why Do I Need This Class, Anyway?"

Blake Learmonth

We can use technology to liberate the energy, the intelligence of the poor, of people who've been left behind--if they have access to it, if they know how to use it, and if the educational opportunities are out there.

Bill Clinton

Academic Setting

Garfield Middle School is an urban school with rural roots, located in Albuquerque's North Valley near to the city's center but in an area that mixes business and industry with alfalfa fields and irrigation canals. The school's population is predominantly Hispanic (80+%) with the majority having well-established roots in the neighborhood. Some families can trace their heritage back several hundred years (Albuquerque was established in the early 1500s.), and many for several generations. Approximately 15% of these Hispanics are more recent arrivals, primarily from Mexico, and qualify as LEP—Limited English Proficient. There is considerable social tension between the "native" Hispanics and the newer immigrants. The composition of the remaining students is 13% Anglo, 4% Native American, just over 1% are Black, and .3% classify themselves as "Other." Approximately two-thirds of our students qualify for the federally funded free lunch program and Title I services. Students score below city and national averages on a variety of standardized tests.

In terms of technological infrastructure, the school is well equipped. Networked computers exist in all classrooms, with both Internet and local connections. There are three computer labs and a few mini-labs-(classes with several computers). Few students have access to computers at home, but most have been exposed to computers since early elementary school. There is a school-wide computerized reading program called *Accelerated Reader*, which is used by all students, including those who speak Spanish as their primary language. As is typical in Albuquerque schools, most of the computers are Macintosh, primarily reasonably modern iMacs.

My classes are seventh and eighth grade Introduction To Computers, a basic computer literacy class, and Computer Graphics, which emphasizes learning design principles while learning to use basic computer drawing and painting programs. Classes are "full inclusion" or "mainstreamed" with all levels of regular and special education students with no additional support in the classroom. The classes are part of the elective department, although most students claim to have been assigned to classes as opposed to having chosen them.

State standards for technology, as a subject, do not yet exist, although there are references to technology in the standards for other subjects. For example, history benchmarks 1-C/4 and 1-C/9 require secondary students to explain "how scientific and technological innovations have brought about change." The following History of Computers section will address that question directly. The "Guiding Principles" section of the language arts standards requires that students attain "literacy in all forms of media."

Unit Goals

Introduction to Computers Class/ Goals:

- Talk about the future of technology in terms of jobs and power.

- Introduce the history of computers/information technology
- Briefly explain the inner workings of computers.
- Have students make predictions about the future.
- Integrate learning about word processing applications and journal writing exercises.

Computer Graphics/ Goals:

- Introduce the computer as a tool for producing images.
- Compare pencil and paper design work with computer aided design, pros and cons (again using journal writing as an integration tool).
- Talk about plagiarism and cheating; what is and what isn't okay.
- Experiment with perspective and 3-d rendering tools.
- Discuss / demonstrate trustworthiness or lack of trustworthiness of digital images.

Context and Background

Rapidly advancing technology will accentuate the differences between the "haves" and "have-nots," leaving the have-nots further and further behind (NTIA). Being left out and left behind is something with which middle school students can identify and which they would classify as "a bad thing." They seem to have an ability to comprehend the powerlessness of being left behind. As a teacher of technology, part of my responsibility is to promote an understanding of how important and empowering technological competence is.

The Need for Computer Education

And Computer Geeks Shall Inherit the Earth
-Greg Smith

In the fairly near future, if not today, society will be divided into three groups based largely on technical competence. The first group (or last, depending on one's point of view) will be those without computer or technical skills; they will be functionally illiterate in technological terms. They will be sweeping the floors in fast food restaurants and doing other menial and unrewarding tasks (This is of course a generalization). Next will be people who just know "what button to push," and can use computers and other technologically advanced devices to accomplish a task. They will have decent jobs and, potentially, good lives. Their productivity will, in many cases, be tied to their level of technological literacy. The third group will be people who really understand and can use computers and other technology creatively. These people will be designing and running the world in which they live.

It is the nature of technology to improve productivity; a person can produce more with tools than without them. In a world with static demand, an increase in productivity (more goods produced by fewer people) would necessarily lead to unemployment. In the past, and today new technologies, in fact, eliminate jobs. For example, a St. Louis bank was able to increase its daily transactions by 35,000 while using ten percent fewer employees, by using computers (Treadwell), Ford Motor company made about as many cars in 1978 as it did in 1988 but with half as many workers. In the US, agricultural production has steadily increased while the number of farmers has decreased dramatically. Similar stories have played out in various industries since the late 1700s (Volti145). Why then do we not have massive unemployment? Fortunately, although "technology's effects on employment are substantial, … they are also contradictory." (Volti 145). Although people are displaced by technology, technology also creates jobs, both directly in manufacturing and maintaining the new technology, and in spin-off industries and services. The example used by Volti is, again, Ford. Mass-producing automobiles uses far fewer worker hours than hand building them; it also makes them much less expensive. At a much lower price, far more people can afford to buy cars. The boom in automobile ownership indirectly created many jobs, including highway builders, mechanics, motel workers and employees of car dealerships, just to name a few.

Today we often think of technology as meaning computers. The widespread use of computers has certainly eliminated some jobs, but it has created a huge information industry. This has, in turn, created a boom in employment, even creating shortages of competent workers:

> "Sun Microsystems and many other giants camped along Highway 101 holding out banners begging for technical workers these days. Their desperate need reflects the fact that our increasingly techno-driven culture doesn't value a technical education highly enough. We could help both the engines of our prosperity and the poor in our society by making a commitment to higher teachers' salaries, better-funded schools and more-expansive financial aid programs (Merritt 2).

Just as hammers (or pneumatic nail guns) and saws are the tools of carpentry, knowledge of how to use and manipulate information is the tool of the information industry. For the most part that means knowing how to access information and manipulate it with various computer programs, which is the answer to the question in the title of this paper, "Why do I need this class any way?"

Robert Walker also answers the question:

> Whether working on Wall Street or at Wal-Mart, knowing computer technology is a necessity. Regardless of what future occupations today's young children seek--whether in medicine, education, industry, auto mechanics, banking, or food service--those who learn their basic skills at an early age and are comfortable in the use of technology will be at a definite advantage.

So if we agree that students need computer skills, what should they learn? There is considerable debate on this subject. Sherry Turkle, an MIT professor and researcher on the effects of computers and computer simulations on learning, offers insight into the two opposing points of view.

> One currently popular position about computer literacy is underrepresented. This is the view that computer literacy should no longer be about the computer at all but rather about the application programs you can run on it. The arguments for this position are strong. One is grounded in practical, economic concerns. Entering today's workforce requires fluency with software. Word processors, spreadsheets, databases, Internet search engines, computer-aided design programs—these are the tools of contemporary trades. Learning to use these tools demands a new kind of craftsmanship, one that confers a competitive edge. Additionally, like all craftsmanship, there is a thin line between craft and artistry. These tools, artfully used, enable users to discover new solutions to old problems and to explore problems that were never previously envisaged.

A recent article by Sherry Tutkle was presented by Margaret Goldsborough in the *New York Times*. It presents the view that just knowing how to run software may fall short of computer literacy:

> When schools first began using computers in higher education, individuals were not considered literate unless they understood something of the innards of the hardware and software they used. Today, Turkle says, students are considered computer literate if they can use software programs but need not understand how those tools work or what their underlying assumptions are**.** That shift troubles her. We live in a world where economic and political decisions are increasingly based on simulation, she insists, and empowered citizens need to understand the **nature** of the simulation (Goldsborough).

To understand "the nature of the simulation" students should have something beyond a "black box" vision of computers including knowledge of the history that has brought us to the current state of computing and some knowledge of what goes on inside the machine.

Computer History

"I think there is a world market for maybe five computers."

IBM Chairman Thomas Watson, 1943

Computers have been around for a long time although not in their current form. The term "computer" originally referred to a person who did computations. (Sometimes these were rooms full of people working together on some large, complicated calculations.) The abacus, which in many ways deals with arithmetic in a way similar to modern computers, has been around for several thousand years. In 1642 Blaise Pascal built the first calculator, the "Pascaline," which could do addition and subtraction. Gottfried Leibniz (who also invented calculus at the about the same time as Newton, though independently) improved on the design by adding multiplication and division using repetitive additions and subtractions, an algorithm still used by computers today. The next breakthroughs came in the industrial revolution when punch card technology was first used in Jacquard's power looms. In the 1820s and 1830s Charles Babbage, who ironically is known as the "father of the computer," conceived and attempted to build a large mechanical computer, the "Difference Engine" it never really functioned, although its design could have worked had Babbage not run out of time and money. There is an interesting and entertaining "alternate" historical novel, *The Difference Engine* written by science fiction writers Bruce Sterling and William Gibson, based on the idea that Babbage's machine was built and worked, starting the computer revolution a hundred years too early. A critically important idea incorporated in to the Difference Engine was the use of punch cards, which included not only the numbers to be input into the machine, but also the pattern for processing them (the first "software"), the architecture of the machine was similar in concept to modern computers in that it included a "mill" (the mechanical equivalent of today's central processing unit and mechanical storage or memory) and used an automated typesetter as an output device. Although nothing as ambitious as the Difference Engine was actually produced, several different mechanical adding machines were built, and mechanical cash registers became common in business.

Ada Lovelace, daughter of the poet Lord Byron, wrote sequences of instructions for the Difference Engine, which included concepts such as using repeatable subroutines, now used in modern computers. She earned the title of the "first computer programmer." She also envisioned machines emulating human intelligence.

In the 1890s Herman Hollerith developed an electrically powered punch card reading computer, with an automatic card feeder, which was used to tabulate the 1900 census. Hollerith’s company later became International Business Machines, IBM.

During the same period discoveries in mathematics laid the groundwork for modern computers. George Boole constructed an algebra of logic. His concept was that logical statements could be depicted symbolically. He developed the concept of binary logic, calculating with ones and zeros. Boolean logic is used by modern search engines, the most important operations centering around the words "and," "or," and "not."

In 1925 MIT's Vannevar Bush created the differential analyzer, a large-scale analog calculator that could do many kinds of scientific computations (Comp Tech On-line). In 1937 Alonzo Church and Alan Turing independently came up with the idea that any logical or mathematical problem which could be solved by a human being could be reduced to an algorithm, an algorithm ultimately processable by a machine using switches or gates corresponding to Boole’s binary algebra operators - the words "and," "or," and "not." Turing proposed "Turing Machines," "black boxes" that could process information according to any step-by-step logical procedure. He also proposed that these machines could be combined into a "Universal Turing Machine," which could read instructions and carry out the functions of all the other Turing machines. It could, in other words, be programmed.

In the 1940s the worlds first operational computer, known as Robinson, was created by Ultra, the ten thousand person, British computer war effort. It used electromagnetic relays and successfully decoded messages created by the Nazi Enigma code machine. Robinson evolved into Colossus, which used electronic tubes and did calculations one hundred to a thousand times faster than Robinson (Kurzweil 268). At about the same time, Konrad Zuse developed the first fully programmable electronic computer, which was programmed by Arnold Fast , a blind mathematician. The Germans, confident that they had won WW II, turned down Zuse’s offer to sell them the machine (Gonick 71). In America, WWII spawned computer projects sponsored by both the army and navy. Harvard professor Howard Aiken’s Mark 1 used thousands of noisily clicking relays, occupied 1200 cubic feet and used enough electricity to power an electric locomotive (Moy). It was followed by the quieter and much faster (1000 times faster) ENIAC, which used tubes like Colossus. At about this time John von Neumann, a Princeton math professor, came up with the idea of storing computer programs in electronic memory, so computers would not have to be mechanically reprogrammed for every new problem. Also in the late 1940s, the transistor was invented. It was (and still is) a tiny, silent, cheap, long lasting, and very fast replacement for the relays and tubes of early computers.

Through the 1950s, despite steady advances, computers were sill huge, expensive devices only used by government agencies, major universities and a very few large companies. CBS acquired one, UNIVAC, which successfully predicted Dwight Eisenhower’s 1952 landslide presidential election. The network failed to believe the computer’s prediction, however, and only reported it when other sources seemed to support the computer’s surprising conclusions. Even with advances like computer chips (1959) and computer languages [COBOL was created in 1959 by Grace Hopper, who was also one of the original programmers of the Harvard Mark I, and the discoverer of the first computer "bug" (a moth stuck in the relays of Mark 1)] there were only about 6000 computers in the United States.

The 1960s started computers shrinking and people seriously thinking about the concept of artificial intelligence. Computers and robots became characters in popular fiction from the *Jetsons* to *2001: A Space Odyssey*, as well as becoming more common in science and industry. IBM grew, Intel was founded, and Gordon Moore noticed a trend in computer power and complexity that would come to be known as Moore's Law. Moore's law states that information processing capability doubles about every two years along with the speed of computer central processing unit chips (Leebaert 294).

In the 1970s things got personal. Floppy disks allowed programs and data to be transported and transferred easily. Xerox developed the first personal computer. It pioneered the use of bitmapped graphics, windows, icons, and mouse pointing devices (Kurzweil 273). Sound familiar? Strangely, Xerox didn't market the "Alta," and it was seven years until the borrowed technology showed up in Steven Wozniac and Steven Jobs' Apple Computers. The pocket calculator was introduced and became cheap and ubiquitous. *Byte* and *Creative Computing* Magazines were published, the first widely distributed computer magazines. Annual sales of personal computers topped 5000 in 1975. The ARPANET, a precursor of the Internet, was a success from its very beginning. Although originally designed to allow scientists to share data and access remote computers, email quickly became the most popular application. The ARPANET became a high-speed digital post office as people used it to collaborate on research projects and discuss topics of various interests (PBS).

The next decade launched productivity with electronic spreadsheets and databases, desktop publishing, laser printers and fax machines. IBM decided that the personal computer market had potential and launched its first PC to compete with Apple products. By the early eighties six million personal computers were sold; by the end of the decade computer memory cost one hundred millionth of what it did in 1950. Toward the end of the decade many of our students were born, and notebook computers became common. The term "Internet" was used for the first time in 1982. Though Internet usage was growing, the Internet was still a mostly academic/scientific tool. More people caught on as news groups and email lists became more common. Computers continued to be more powerful, faster and more affordable. The number of Internet "hosts" had grown dramatically from four in 1969 to 23 in 1971, 213 in 1981, 1000 in 1984, and 10,000 in 1987.

The nineties was the decade of the web. Although processor speed and memory size kept leaping ahead, just as Moore's Law had predicted, development of the Internet and World Wide Web was the big news. In 1990, the language of the web, HTML (Hypertext Markup Language), was developed in Switzerland at CERN, a high-energy physics research facility. HTML allows computers with different operation systems on different types of networks to talk to each other. Within four years the World Wide Web came into existence, and AOL had more than a million subscribers. By the end of the decade the World Wide Web/ Internet was simply an accepted part of every day life; children and local businesses commonly had their own web sites, and people had a hard time remembering when it wasn't around. The next step seems to be wireless connection to the web and for smaller networks.

On the hardware front hand-held computers became common. A computer built for the purpose of defeating the world chess champion was successful. A $1,000 computer can do over 10^8 calculations per second (Kurzweil 25)

The past 100 years has seen incredible leaps in humankind's ability to process, store and distribute data. Whether we can deal with the explosion of information is another question. The future? Read *The Age of Spiritual Machines*, or have your students make predictions.

How Computers Work

Any sufficiently advanced technology is indistinguishable from magic.

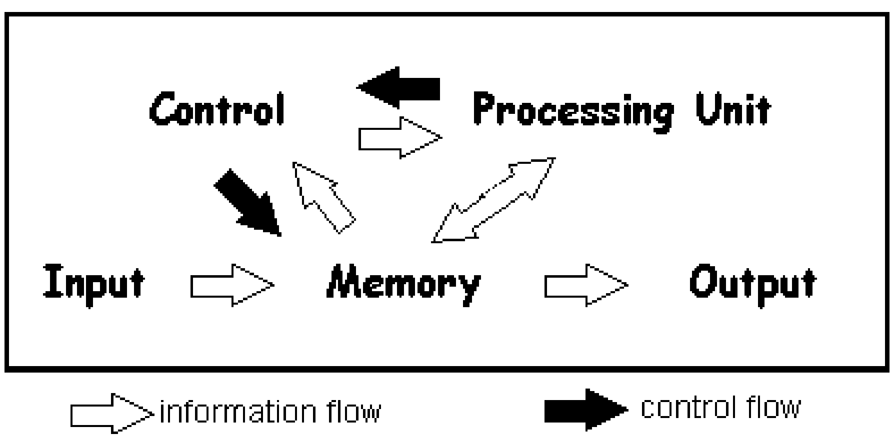
Arthur C. Clarke: *Clarke's Third Law*

Larry Gonick’s *The Cartoon Guide to the Computer* provides in-depth information about how computers work, probably more than you will ever need to know unless you are an electrical engineer or computer programmer (and certainly more than I can provide with this unit). Presented in cartoon style, much of it is accessible to middle and high school students. Originally written in 1983, the book is laughably out or date when it describes the speed of a "modern" computer, and the Internet is not even mentioned; but the basic processes in computers have not changed and the vast majority of the information is still accurate.

Students need to have a basic understanding of what goes on inside a computer to understand why things happen while working with computer applications. Without understanding how hardware and software function, it will be difficult for students to grow to be adults who will have a part in designing their own futures, futures which will almost certainly involve technology.

A basic knowledge of Boolean algebra and the base 2, or binary system, is the basis for understanding all computer processing of information and is important to good practices when using search engines on the Internet. Students should be able to recognize major components of computers, and have a good idea of what they do, and why that function is important. Some knowledge of the components will help them make reasonable consumer choices if they purchase a computer.

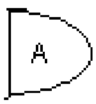
Described as a "black box," a computer takes in bits of information in the form of ones and zeros from some sort of input device (a keyboard for example), processes it through a switch or multiple switches in its processing unit and produces an output, which is translated into some useable form such as an image on a screen or a printed document. We can go beyond the "black box" and look at what really goes on inside. In the simplest form there are five components to a computer (three of which I have already mentioned) input, memory, control, the processor, and output. (Adapted from Gonick 84)

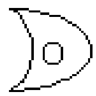


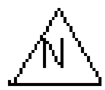
Input is what goes into the computer, both the raw data and the program that tells the computer what to do with the data. **Memory** stores the input and the results that come out of the processing unit. The **control** component translates the information in the program into sequences of machine operations timed with great precision by the computer’s clock. The **processor** is a switch or, more likely, billions, or even trillions of switches in the form of transistors in a silicon sandwich. The processor does the additions, multiplications, counts, and comparisons on the information it receives from the memory. The results of the processor’s calculations (which are really just a sequence of electrical pulses from its switches) are stored so they can be used to generate **output** by some device like a speaker or printer.

The processor doesn’t really know how to add or subtract or solve logical problems; all it knows how to do is switch electricity off and on in its innumerable microscopic switches. However, if electrical pulses are input in the proper pattern and run through the proper series of switches, what comes out of the processor can be interpreted as the answer to some problem, or the result of some operation—output. Each of these theoretical components has a physical counter-part in computers. Input devices include keyboards, mice, scanners, draw pads, and digital cameras. Memory can be in the form of chips with switches that stay in position after receiving an input until reset by another signal and "hard" memory stored on magnetic disks. The CPU, a collection of trillions of specialized transistor switches, does the processing. Control is a number of chips that basically direct traffic inside the computer. Output devices include monitors, printers, and speakers.

George Boole came up with the idea that all logically solvable problems could be solved by machines using algorithms reduced to a system of switches representing the concepts of "and," "or," or "not." Those switches, or logic gates can be represented by symbols:

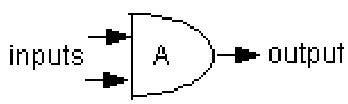
and = 

, or = 

, and not = 

.

As pictured here, information in the form of one or zero, on or off, goes in the left side of the symbol and a signal, also a one or zero, comes out the right side.



For an **"and"** switch, if **all** the inputs are "one" then the output is "one." For an "**or"** switch if **any** of the inputs is "one" then the output is "one." A **"not"** switch outputs the opposite of whatever comes in. Combinations of these switches can give answers to most, if not all, logical and mathematical problems.

Binary calculation matches such a system perfectly since it has only two characters, "one" and "zero," which correspond to a switch being on or off. Much of what computers do is process numbers by sorting and analyzing pulses of electricity that represent a 1 and the absence of a pulse, which represents 0.

Binary Basics

Probably the most important trick to working in the base 2 or binary system is to see 10 as "one, zero" not "ten." There is really only one number in the binary system, the number one. One = one, one + one is "one, zero" (10). 1+1+1 = 11, 1+1+1+1= 100. Each zero adds a power of 2.

BinaryPower of 2	Decimal
1= 2^0	= 1
10= 2^1	= 2
11= 2^1+1	= 3

$100 = 2^2 = 4$

$101 = 2^2+1 = 5$

$110 = 2^2+2^1 = 6$

$111 = 2^2+2^1+1 = 7$

$1000 = 2^3 = 8$

$1001 = 2^3+1 = 9$

$1010 = 2^3+2^1 = 10$

$1011 = 2^3+ 2^1+1 = 11$

$1100 = 2^3+2^2 = 12$

$1101 = 2^3+2^2+1 = 13$

$1110 = 2^3+2^2+2^1 = 14$

$1111 = 2^3+2^2+2^1+1 = 15$

$10000 = 2^4 = 16$

$100000 = 2^5 = 32$

$1000000 = 2^6 = 64$

$10000000 = 2^7 = 128$

$100000000 = 2^8 = 256$

As you can see, the binary system creates some very long numbers, which makes them hard for us to deal with; but computers have no problem keeping those huge numbers lined up.

Addition works as in the decimal system: when a column adds up to 10 (two) you just carry a 1 to the left the same as you would in adding 9+1 in the decimal system.

```
1 carried
14
+7
21
```

For example 14+7 in the decimal system looks like**:**

1 carried
14
+7
21

In base 2 it looks like this**:**

11 carried
1110
+111
10101

Multiplication can be done with repeated additions!

1101
-1100
0001

becomes

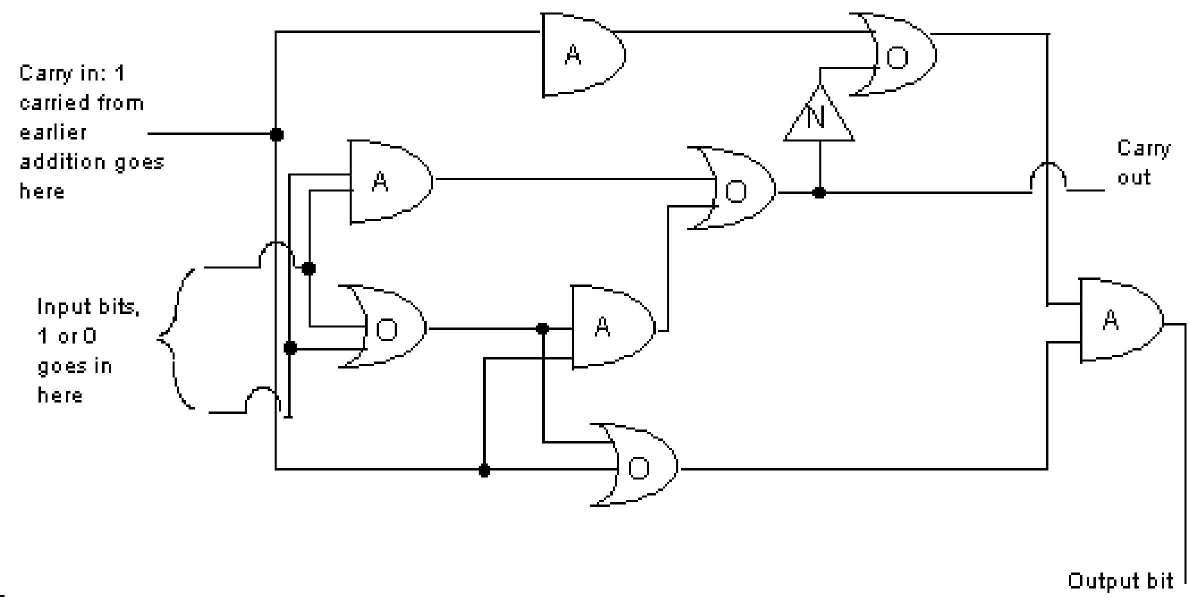
1101
+0011
10001 = 1

Subtraction can be done by adding too. By

inverting the number to be subtracted, adding the two numbers, and ignoring the final carry. Like multiplication, division can be done by repeated subtractions. So, a computer that only knows how to add can do all arithmetic.

The following diagram is an example of a collection of switches that can simulate addition of 2 binary numbers. It can be described as a two bit adder since what it does is simulate the adding of two bits of binary information (two **BI**nary digi**T**s - BITs). Each bit can represent either one or zero (Gonick 115-1121).

Two-Bit-Adder



(Moy)

The adder can only add one number at a time, so larger numbers require either a chain of adders or a chain of calculations. Other combinations of switches can simulate other mathematical operations or logical truth tables. Sequences of binary numbers can also code for letters of the alphabet, symbols, colors, and shapes, as well as our commonly used digital, base ten numbers. Programs and programming languages can use the controlled stream of ones and zeros to simulate virtually anything, even art.

Using the Computer as a Tool to Generate Art: "Isn't This Cheating?"

Creating art has always been tied to technique or craft. To produce art in any medium tools are required: brushes for painting, chisels for carving stone, pencils or pens for drawing, a camera and darkroom for photography. (Francastle) Two things separate the master artist from the masses: mastery of technique and creative vision.

It is a fairly common belief that suffering or struggle is required to attain artistic prominence. For those subscribing to this belief only long, often frustrating, practice produces the skills in manipulating the artist's medium required for mastery of the art. Furthermore, for them, the process is nearly as important as the product. Technical advances that create new media (or make manipulating old media easier) are cause for concern in the established art community

Although I do not usually consider myself an artist, I have spent large parts of my life closely associated with people who do, and I have taken a number university classes in art and architecture. I was truly surprised by the conservative attitudes expressed by artists toward innovation in **their** medium. Potters have told me (in the extreme), that "real" potters dig their own clay and use kick wheels rather than those powered by electricity to turn their pots. Some potters were suspicious of using natural gas or electrically heated kilns to fire their work. In the world of painting similar attitudes exist. "Real" painters grind and mix their own pigments, use oil based paints, and stretch and prime their own canvasses; people who use factory- made acrylics on store-bought canvases are perceived as hobbyists, little removed from the paint-by-numbers crowd. Photography (and later holography) has struggled to be considered an art form. Its influence on other art forms such as painting has been seen as both beneficial and destructive. Photographs replaced sketches as resources for paintings and allowed artists to see details and capture motion in ways the unaided eye could not. Painters, however, have feared that depending on photos would limit creativity to a camera-like vision rather than human vision (Van Deren 299, 300). For anyone who has either worked in a darkroom or closely observed a skilled photographer at work, there can be little doubt that what is going on is clearly "art," if not magic, and that it is as far removed from the work of people who take vacation snapshots or photos of their friends and have them printed in an hour by Wal-mart, as the work of a house painter is from that of Van Gogh.

Artists who use computers to assist them in creating art are running into similar problems is being accepted in the artistic mainstream. Like photography, because of electronic and mechanical reproducibility, electronic art challenges the accepted paradigm of artistic genius: the concept of a work of art as a unique image produced by hand by a trained artist. Many artists and critics argue that technology corrupted art by turning it into a purely mechanical process (Kodumal).

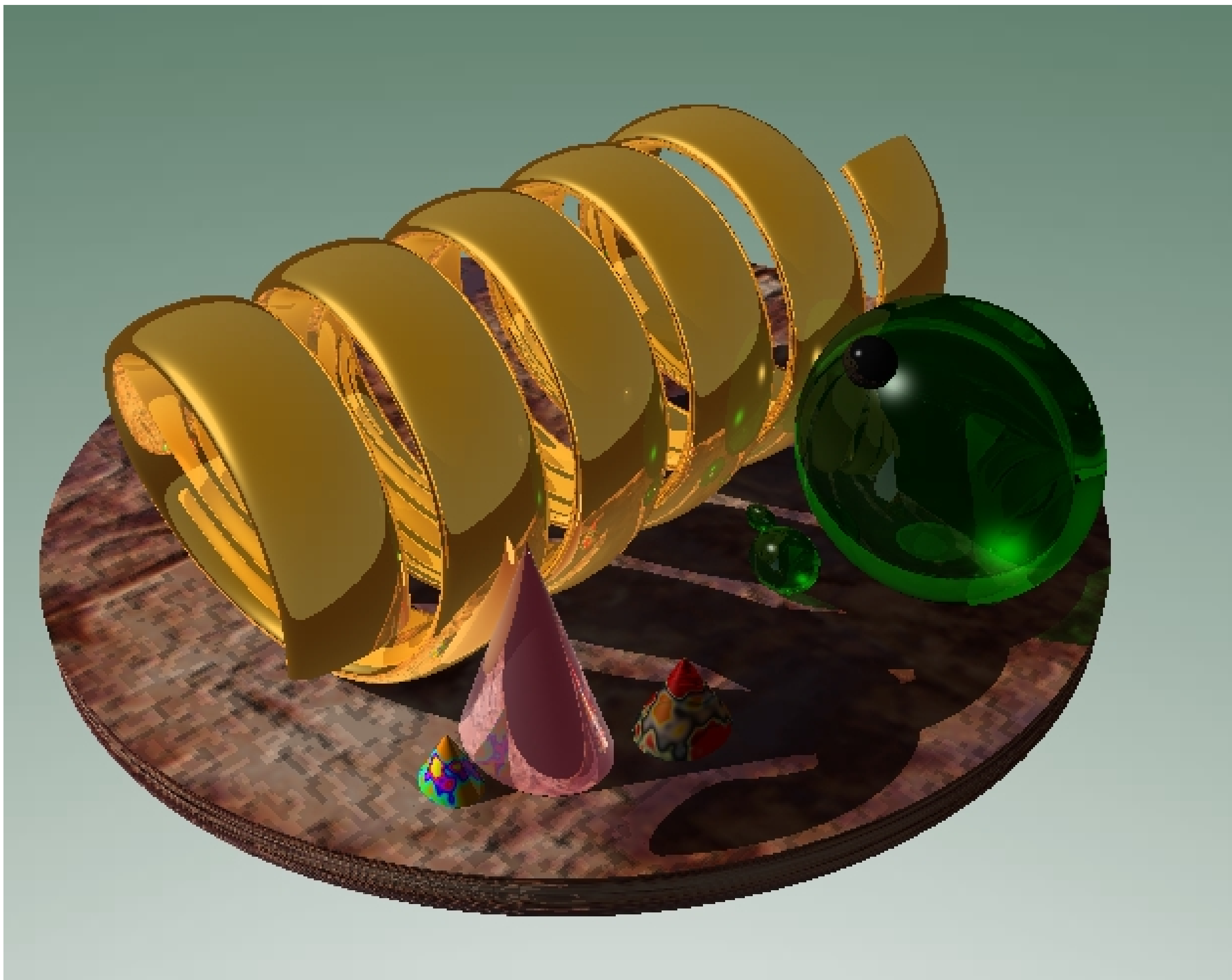
In seeking a broader market than just fine artists, the makers of graphics software have done nothing to dispel the feeling that the computer is a tool for non-artists. Advertisements and technical manuals for graphics software frequently point out that the software allows non-artists to produce high quality images. One of the chapter headings in Ken Milburn and John Croteau’s *Flash 4 Web Animation: F/X & Design* is, "You Don’t Need Drawing Talent." They go on to state, "You don’t have to be a practiced artist to produce worthwhile results." The introduction to Rockport Publishers’ *Cyber Design: Computer Manipulated Illustration*, discusses the situation in detail:

> Effects that would take a traditional illustrator **hours to create** can now **be done in minutes**. These capabilities are both a blessing and a curse. … For example, an assignment requiring gold lettering can call for Kai’s Power Tools *Gradient Designer* for *Photoshop*. Though this task can be accomplished by hand, with an airbrush (Interesting that an airbrush is considered "by hand.") **having this tool is a big time-saver.** With a computer that shiny gold look can be created in less than a minute by **dragging** and **clicking** a few buttons. Which leads me to the potential curse. Realistically, with all the **ready-made effects** available one doesn’t need a great deal of **artistic talent** to create an image on a computer…with everyone using the same filters and presets there is potential for redundancy and mediocrity in our artwork (Rockport 5, parenthesis mine).

Surprisingly, the use of computer tools even comes into question in the field of commercial art. The June 2001 issue of *How*, a pro-technology design magazine, included results of a survey of its readers which asked the following question: "Does starting a design on the computer, rather than with pencil and paper, make designers conceptually lazy?" Sixty four percent of their responding readers said "yes." Here are two contrasting samples of the comments that readers included with their votes: "Sketching ideas out leads to more creativity and originality because you’re starting from what’s available in your mind, not your menu." And on the "no" side: "Not thinking is what makes designers conceptually lazy. The computer is nothing more than a tool that allows for a more fluid process." (*How* 10).

While teaching Computer Graphics to middle school students, I have frequently been asked whether using graphic programs’ *copy* and *paste* features (or their built in shape creating tools) was cheating. By middle school, many students seem to have picked up the idea that anything other than free-hand image creation is some how not really art. If "art," in this case, is producing images that are pleasing or interesting to look at, my middle school students have produced art with very little practice using very basic computer graphics tools.

There is no question that students with little or no skill at drawing or painting on paper can quickly learn to create clean-lined compositions with recognizable shapes. They complete design exercises very quickly, using almost all of the time to explore the design concept and practically none mastering technique. Is this a good thing? Have they missed something by not mastering pencil and paper or painting techniques? Or are they learning design concepts more quickly and being encouraged by producing appealing, finished-looking products using better tools? Should students be involved in discussing these questions?



The following image was created by the author using Strata 3-D, a free, downloadable program available at www.strata.com . I include the image, not to show what a great artist I am, but just the opposite. This was created in about 35 minutes, using a program with which I have only a few hours of experience. Middle school students in my classes have produced similar quality images. It would take several hours more work to come up with a really pleasing composition, but the textures, shadows, and reflections, all of which were generated by the computer are, stunning.

The future will only serve to throw new challenges into the "what is art?" question, making it more complicated. There are already machines that can take a "drawing" created with a 3-D drafting or image program and create a real, solid model of the object, complete with textures and colors. So, at least on a small scale (so far), sculpture is not beyond the capability of a computer-aided artist. Certainly, laser cut wax models for small castings can be made. A recent *Wired* magazine article described the concept of the "personal fabricator," a machine evolved from inkjet technology capable of printing silicon circuit boards, electromagnetic ink displays, and even three-dimensional objects (Lewis). This technology, which may not be far off, would allow artists to produce artwork on the Internet, which customers could download and recreate in their homes. The whole concept of what constitutes "original" art would have to be rethought.

If art produced with hi-tech tools is suspect, in terms of being "real" art (I think as good a case could be made for technology having freed artists to be more creative and more productive), what about art created by machines with little or no human input? In *The Age of Spiritual Machines*, Kurzweil presents several examples of visual and literary art created by computers, which although programmed by humans, were then turned loose to produce what they would (166,167). In his prediction scenarios for the year 2029, Kurzweil describes the following situation: "Cybernetic artists in all of the arts…no longer need to associate themselves with humans or organizations that include humans. Many of the leading artists are machines." (Kurzweil 223).

Involving students in discussing the issues described above seems important to their having a true understanding of what they are doing when they are using powerful tools in a graphic arts class. Beyond just knowing what button to push or how to draw a blue triangle with a gray shadow, students need to understand what the tools are doing for them. For example, students study the history of the use of perspective, shadows and reflections to add depth to two-dimensional art in conjunction with learning to use the tools of Strata 3-D or other 3-D rendering programs.

Implementation

Unit Objectives

During the course of this unit, which is designed to be a major part of a semester-long technology class for seventh and eighth grade students, students will development an awareness of computer hardware and several kinds of computer applications. In the second part of the class the emphasis will turn to graphics. Students will produce a variety of projects in which they will both produce a product using a particular application and explore a new technique or information source.

My class lab is an iMac lab, but all of the applications that the students will be using in this unit are cross platform, that is they can be run on either Macintosh or PC computers. Ideally, I would trade labs for half the semester with the business teacher next door so that students could be exposed to PC style computers as well as iMacs. Additionally, using cross platform programs allows students to take work home (most students who have computers have PCs).

Lesson One

The first few days of class will be spent acquainting students with basic computer hardware and basic procedures for turning machines on and off properly, opening programs, and (perhaps most importantly) saving files on our network server. I have acquired both a dead PC and a dead iMac, which I have disassembled to so that I can show the innards to students. I usually also destroy a floppy disk to show students what is inside. These props are used to give students a chance to demystify "the black box" as well as associating a physical object with the computer component words that they have heard from their friends or sales people. If I can get several old dead computers, I encourage the kids to take them apart. This encourages "discovery" learning as well as taking the edge off of their curiosity (this seems to discourage them from taking apart the classroom computers). We will discuss vocabulary associated with hardware including *input devices*, *storage devices*, *processors* and *output devices* while the corresponding hardware pieces are available. Input devices include keyboards, mice, scanners, microphones and cameras (students may reasonably argue that CD-ROMs and floppies are input devices, though they meet all the criteria for storage devices). Storage devices include all kinds of disk drives (floppy and hard), CD-ROMs, and DVD-ROMs. If your server is available to show students, it probably has a tape back up for storage. Additionally, RAM chips are storage and may be easy to find in your taken apart computers. The most obvious processor is the CPU on the "motherboard." Output devices include printers, monitors, and speakers.

Divide the students in three groups: one for input devices, one for storage, and one for output devices (leave processors for later). Have students in each group list as many things that go in their group as they can. They will report to the class what their devices are, what they do, and why they think they belong in their group. Make groups as diverse as possible; particularly don't let boys and girls segregate.

Duration:
One or two class periods, depending on complexity of props.

Materials Required:

- old computers that can be disassembled,
- vocabulary cards with the words "input," "output," "processing" and "storage" (or write these on the board),
- photos, posters, product boxes or illustrations of computer components.

Evaluation:
Hold up pieces of hardware and have students identify them; call on different students to describe their function.

Extension:
Have students check this website. http://www.kids-online.net/kidsframe.html, which is a good, self guided, visual tour through a PC style computer. It offers surfers choices of different levels of complexity from very basic to fairly complicated. It both presents the hardware visually and gives detailed descriptions of component functions. The site seems to be growing and may have other interesting features by the time this is published.

Following lessons should give very basic operation instructions for whatever kind of computers your students will be using. Students should be comfortable with opening programs and files, how menus work and how to save files.

Lesson Two: Word Processing and Journal Writing

As soon as students have acquired the skill to open a program and deal with menus, this lesson should be introduced. Since this is a computer class, daily journals will be kept as files and created using word processing software. Journals will be used both to practice word processing skills and to explore issues related to technology. I recommend using *Apple Works 5* to introduce word processing. It is an inexpensive program commonly found in schools, easy to use, not a memory hog, and is totally cross platform. It will run on PCs and Apple products. The files created can be opened in either platform, which is good for homework assignments. For older students it is probably important to use *Word*, since it is such a standard program in businesses and universities; but *Word* is a lot more complicated than *Apple Works* (*AW*) and needs a lot of memory to prevent crashes, particularly on Apple Machines. *Star Office* is a good program available free from Sun

Microsystems. It runs on Windows and Unix/Linux machines. It is not as user friendly as *AW* or even *Word*, but really is free-- no catch. *Star Office* also has the advantage of being able to read and write files compatible with *Word* and other *Office* products. Like *AW*, *Star Office* includes database, spreadsheet, paint and draw programs. (Did I mention that *Star Office* is free?)

Word Processing Concepts:

Students should be able to format text, changing font, size and color; format paragraphs, setting indents and line spacing; insert pictures and other objects, and control how text wraps around them; set up pages with margins and tabs; and use spell checker functions. Daily journals give students something to write about and can contain formatting instructions.

Sample Journal Topics:

1. Write a paragraph describing what computers are good at doing. Give the paragraph a title in a font different from the body and sign your name at the end using a font that looks like hand writing.
2. How would you feel if you discovered that your teacher was really a machine? What would a machine teacher do better than a live one? Would you treat a machine teacher differently? Why or why not? (There are several good *Star Trek* episodes including "The Measure of a Man" which could be used to stimulate thinking on this topic. There is also a very short Isaac Asimov short story "Oh, the Fun We Had," that describes a day in two future kids' lives when their mechanical teacher breaks and they discover books. This could serve as an introduction.
3. Put a horribly spelled sentence up either on the board or on a website wherein students can access it. Include some misused homonyms like "there," "their," and "they're," which the spell checker won't catch, to demonstrate the strengths and weaknesses of spell checkers. If your word processor program includes a thesaurus discuss how to use that both to improve variety and to check the meaning of sound-alike words.
4. Have students create a template journal page which they can use for future journals that includes a title, their heading information and a date. Different word processor programs handle dates differently, so students can experiment with the date-auto formatting function. Using the *save as* function they can save their new journals without saving the changes to the templates.
5. Talk about plagiarism. Have students answer the question, "Is it okay to cut and paste several paragraphs of text from a website directly into your science project?"
6. Using the Internet, find information about an animal that interests you. Write a paragraph about the animal and insert a photo or drawing of the animal into the text. This prompt would have to follow a lesson on using search engines and the various ways to capture images from the Net.

Duration:
The first few minutes of every class period, however, some topics may take more than one day. Some skills may require direct instruction.

Materials:

- word processing software (Apple Works, Word, or Star Office, for example),
- Internet access,
- floppy disks or a place on the school server or Internet to save journals,
- science fiction texts or video clips for discussion starters.

Lesson Three: Search Engines

At some point students get the idea that if they want information on frogs all they need to do is type www.frog.com into the location bar on their web browser. When that turns out to be a publishing company in Florida that has nothing to do with frogs, you may get the comment "there is nothing on the Internet about frogs" from students. Learning how to use search engines is one of most important computer skills your students can learn. The fact that search engines use the same Boolean Logic that is the basis for computers being able to process information at all is a good segue into teaching about processor function.

1. *Introduce search engines*
   Tell students that search engines are websites that help them find other websites. Assuming you have a lab with an Internet connection, give students the addresses of several search engines. Explain the difference between regular search engines like Alta Vista or Excite and meta search engines like Google, the Yahooligans search function or Dogpile, which search dozens of other search engines. Use an Internet scavenger hunt for a practice exercise. If students learn nothing else about the Internet or search engines they have still come out way ahead.
2. *The Power of the Quotation Mark*:
   Doing a search by typing in a person's first and last name, or the name of a state like New Mexico, often produces discouraging results. The New Mexico entry, for example, will bring up every website that has any mention of the word "new" or the word "Mexico." Students who search for Tony Hawk (a professional skateboarder) may find the information they want, but will also find information on hawks. Putting "Tony Hawk" in quotes will produce a list of websites that contain the two words together. Have students try some two-word searches with and without quotation marks, and compare the results. (Search engines are getting smarter about this, particularly with commonly searched for names, so searching for Bill Clinton (without quotes) would produce better results than you might expect.)
3. Most search engines offer "advanced" or "expert" versions of their searches. The words "advanced" or "expert" frequently scare students away, which is unfortunate because the "advanced" or "expert" modes are often easier to use than the regular version. Google's "advanced" search allows students to do a true Boolean search without typing in the Boolean notation (which often doesn't work on search engines. Search engines have personalities. Their help pages are great for finding out about their personalities. For example, many search engines use the symbol "-" for the Boolean NOT and simply assume that AND is between each word). "Advanced" searches also allow students to specify the language in which they want responses to be. The following websites have good information on Boolean searching: http://adam.ac.uk/info/boolean.html#bool c .

For most people, finding a few favorite search engines and becoming familiar with their quirks produces the best results. So, give the students things to search for and let them practice. It is a good idea to try a search yourself before assigning it. The classic booboo is searching for "White House," which produces a fairly notorious pornography site that has made an effort to have people "hit" it accidentally.

Materials:

- Internet access
- search engine names and addresses
- "scavenger hunt" list of things that you want students to search for.

Assessment
Grade scavenger hunts, did students find sites that you expected?

Lesson Four: Can You Trust Photographs or Digital Images?

Using the title of this lesson as a journal topic would be a good way to introduce the lesson. The ability to create real looking photos of unreal situations has been around for a long time, but technology has made fake photos look very real. Adobe Photoshop has probably been used to create more photo forgeries than any other tool. Many schools have the program. This assignment could be accomplished with most paint-style photo programs.

Students will bring in photographs of themselves as well as magazine photos of famous people or famous places. Using Photoshop's lasso tool, students will cut themselves out of a scanned version of their photo and paste themselves into a scanned photo of a famous person or place. Students will be encouraged to be very detail oriented, using their agility to zoom in on photo details and "paint" individual pixels if necessary to produce clean, realistic edges. Blur and smudge tools can also be used to soften edges. As an alternative assignment students could retouch themselves into space aliens for *National Enquirer* style magazine covers or switch heads with their friends. Keep instruction on how to use the tools very brief; hands-on messing with the tools is the way students get good at using them. Later versions of Photoshop and some other programs allow the use of multiple "undos," so students can back out of techniques that didn't work without losing the image. Introduce the Ctrl-z shortcut (apple button-z on Macintosh and Apple computers) as a way to immediately recover from mistakes.

Use daily Journals to discuss the implications of what they are accomplishing for news photography.

Durration:
Several days. *Photoshop* is a very powerful program. A full semester could easily be spent exploring it thoroughly.

Materials:

- computers with at least 64mb of RAM (less RAM makes running *Photoshop* really frustrating, because rendering the image after the changes takes so long).
- a photo-retouching program
- a scanner
- a printer that can print photo quality images.

Assessment:
Peer comments about which pictures are the most believable and what characteristics make them believable. Allow students to revise their work after peer comments.

Other Lesson Ideas

1. Create still life compositions with *Strata 3D* or another 3-D graphics program for the purpose of creating a school gallery show with the art classes doing the same assignment with conventional media.
2. Have students create an on-line timeline of computer history from the 1800s to 2020 from research using Boolean expressions in a web search.
3. Have students create a web based tutorial for elementary students on the inner workings of computers based on library and Internet research.
4. Create a Rube Goldberg, cartoon-style explanation for how computers work.

Experiment with drawing the cartoon with pencil and paper then trying to duplicate the work using a drawing program. Possibly include the results with #3 above.

Bibliography

Asimov, Isaac. “The Fun They Had." *The Complete Stories*, New York: Doubleday Foundation, 1990.

An interesting short story based on the premise that a pair of future kids computer teacher breaks down and they discover books

Clinton, William J. *Weekly Compilation of Presidential Documents*, 20 March 2000, Vol. 36 Issue 11: 6,54.

The Computer History Center Homepage, The Computer Museum. 2000. http://www.computerhistory.org August 2001.

This website has a great illustrated history of computers, plus links to other good sites.

Computer Tech Online. "Computer History Timeline" http://computertech.server101.com/index.html?main.htm&2 (click "Timeline")August 2001.

Francastel, Pierre. *Art and Technology*. New York: Zone Books, 2000.

Gibson, William and Bruce Sterling. *The Difference Engine*. Bantum Books, New York: 1999.

Goldsborough, Margaret. "M.I.T. Professor Reconsiders Children's Online Lives" *New York Times*, February 14, 2001.

Gonick,Larry. *The Cartoon Guide to the Computer*. New York: HarperCollins,1991

A wonderful resource in cartoon form. Contains both very basic and technically challenging information. A lot of fun to read. Excellent computer history.

Kids on line. "computer parts tutorial". http://www.kids-online.net/kidsframe.html

This is a really well done visual tour of the computer. It has tutorials about computer hardware at several levels.

Kodumal, Lorraine and Rachel Eggebeen. "Historical Overview." *INTERFACE 1998*, 15 July, 2001, Duke University, http://www.duke.edu/~lmk2/INTERFACE/overview.html

Kurzweil, Ray. *The Age of Spiritual Machines: When Computers Exceed Human Intelligence*. New York: Penguin Group, 1999.

Lewis, Judith, "Design is Evil," *Wired* Jan. 2001: 194.

Leebaert, Derek, and Timothy Dickinson. " A World to Understand: Technology and the Awakening of Human Possibility," *Technology 2001*. Derek Leeabaert ed. Cambridge, MA: MIT ,1999:293-321.

Macaulay, David. *The New Way Things Work*. New York**:** Houghton Mifflin Company, 1998.

Like all of Macaulay's books this is a well-done, cartoon style guide. It has a lot of good information about how computers work and on various computer components

Merritt, Rick Boyd. "Digital Divide," *Electronic Engineering Times*, 1 May 2000. Issue 1111: 84.

Milburn, Ken and John Croteau. *Flash 4 Web Animation: F/X & Design.* Scottsdale, AZ: The Coriolis Group, 2000.

Moy, Tim. Seminar Notes. Multidisciplinary Perspectives on Society, Science and Technology. Albuquerque Teacher's Institute, UNM, , Summer 2001.

New Mexico State Department of Education, *Standards and Benchmarks*, http://sde.state.nm.us, 2001.

National Telecommunications and Information Administration (NTIA) U.S. Department of Commerce. "Falling Through the Net: Defining the Digital Divide," August, 2001, http://www.ntia.doc.gov/ntiahome/fttn99/contents.html 1999.

PBS. Life on the Internet: Net Timeline. 1998 PBS August 2001. http://www.pbs.org/internet/timeline/timeline-txt.html

Rockport Publishers. *Cyber Design: Computer Manipulated Illustratuon*. Rockport MA: Rockport Publishers, 1996.

Smith, Greg, "Do You Have What It Takes To Succeed?" *Career World*, Feb/Mar2001, Vol. 29 Issue 5: 13.

Sommerner, Christa and Laurent Mignonneau, eds. *Art @ Science*, New York: Springer Wien, 1998.

Treadwell,David and Tom Redburn. "Workplace: Site of the Latest Revolution," *Los Angeles Times*, 24 April 1983, I:12.

Turkle,Sherry. "Seeing Through Computers: Education in a Culture of Simulation,"*The American Prospect* .Volume 8, Issue 31,1997 (Also: http://www.prospect.org/print/V8/31/turkle-s.html).

Volti,Ridi. *Society and Technicological Change*, New York: Worth Publishers, 2001.

Van Deren, Coke. *The Painter and the Photograph*. Albuquerque NM: University of New Mexico Press, 1972.

Walker, Robert J. "Computers Should be Made Available to Children of Low-income Families." *USA Today* . v. 126 (Sept. 1997) 64-6

"Web Poll." *How Magazine* June 2001: 10.