COMPUTER ASSISTED INSTRUCTION

VIA GRAPHICS TERMINALS

by

CATHERINE TROTTER KOSTETSKY, B.S.

August 1974 TR-42

This report constituted in part the author's thesis for the M.A. degree at The University of Texas at Austin, August 1974.

Technical Report 42

Department of Computer Sciences

The University of Texas at Austin

Austin, Texas 78712

ACKNOWLEDGEMENTS

I wish to express my appreciation to my supervising professor, Dr. Terrence W. Pratt for his guidance and assistance. I am also grateful to Dr. A. G. Pearson for his many helpful criticisms and suggestions.

C. T. K.

July, 1974

ABSTRACT

This paper examines how CAI/graphics systems can be developed for educational institutions which have specific graphics capabilities in mind. The diversity of graphics capabilities and the effect they have on the system design is emphasized. Some existing CAI/graphics systems are described in order to give insight into the problem. Next the variety of ways in which both hardware and software components of a CAI/graphics system can be realized is discussed. The hardware components are graphics terminals, central control computer, and communication links; the software components are operating systems, graphics subsystems, CAI/graphics applications, and CAI/graphics author languages. Finally some guidelines for development of a CAI/graphics system are given.

TABLE OF CONTENTS

Chapter		Page
I.	INTRODUCTION	1
	A. Purpose	1
	B. Graphics Capabilities in CAI	2
II.	DESCRIPTION OF EXISTING CAI/GRAPHICS SYSTEMS	6
	A. Brentwood Tutorial Mathematics Program	7
	B. University of Alberta Drill and Practice Mathematics Program	9
	C. University of Texas CAI Program in the Arabic Writing System	11
	D. The University of Illinois PLATO IV Biology Lesson	13
	E. The University of North Carolina Numerical Analysis CAI Program	15
	F. University of California at Irvine Physics Lesson	17
	G. The University of Illinois Geometry Program	19
	H. CATTS: Computer-Aided Training in Trouble-Shooting	21
	I. The University of Connecticut Music Program	23
	J. Massachusetts Institute of Technology Elementary Mathematics Program	25
	K. Point Park College Surgery Lesson	29
III.	THE COMPONENTS OF CAI/GRAPHICS SYSTEMS	32
	A. Hardware	32
	B. Software	52

Chapter Page

IV. THE CAI/GRAPHICS SYSTEM 63

A. Hardware 66
B. Software 70

V. CONCLUSIONS 75

BIBLIOGRAPHY 77

VITA

LIST OF FIGURES

Figure	Page
1	27
2	32
3	47
4	48
5	49
6	50
7	51
8	59

CHAPTER I

INTRODUCTION

A. PURPOSE

The primary objectives of this paper are the following: (1) an analysis of some existing CAI/graphics systems, (2) a discussion of the varieties of and requirements for each component of a CAI/graphics systems, i.e. hardware, software, and courseware, and (3) the formulation of how to select the right components for a desired CAI/graphics system.

One of the main problems encountered in a discussion of CAI/graphics is the lack of agreement among educators and computer scientists as to what class of computer applications can be called "CAI/graphics." In this paper, a CAI/graphics system refers to a single or multi-terminal instructional system, where the terminals used have the capability of presenting "smooth curved pictures" on the screen rather than simply alpha-numeric characters, but do not provide a hard-copy of the interaction. An instructional system refers to the situation where a student sits at a terminal, interacts with the system, and learns, as defined by educators. So CAI as used in this paper encompasses the traditional drill and practice programs, tutorial programs, simulations, and learner controlled systems.

This paper does not attempt to justify CAI as a valid method of instruction. The assumption is made that CAI is a reasonable alternative to the traditional approach to teaching. Bunderson (1970) presents an argument for CAI/graphics in education.

This work will examine how CAI/graphics can be developed for educational institutions which have specific graphics capabilities in mind. These capabilities can vary considerably. For example, they can range from pre-defined static pictures being presented to the student, to the student himself creating animated pictures. These differences in desired graphics capabilities have a direct effect on the type of CAI/graphics system an educational institution uses. In the next section, a description is given of different graphics capabilities and how they could be used.

B. GRAPHICS CAPABILITIES IN CAI

A graphics capability that some educators might desire for a CAI/graphics system is simply to be able to present pictures to a student on an individualized basis and ask the student questions about them. Depending on the student response, the next appropriate picture is chosen for that student, so that the sequence of pictures and quantity of pictures could vary from student to student. However, at the end of the interaction with the CAI/graphics system, the students all would have achieved at least a minimum level of mastery of the information presented. The CAI/graphics system in this case controls the student's progress through the material based on previously defined rules.

Simulation of some real world situation using a CAI/graphics system can require a strikingly different graphics capability from that mentioned above. In this case, the student might be presented with a picture of a real world situation which can be dynamically changed based on the student's interactions with the system. In essence, the student "sees" the effects of his requested actions on the situation. Oftentimes, the results of his actions are shown to have an adverse effect on the situation. Since the student is still learning about the real-world situation, for the student to see the results of his actions immediately is an invaluable use of graphics.

In teaching quantitative courses using CAI/graphics, the desired graphics capabilities can range from showing a graph of a given equation to letting the student input geometric figures.

In some newer CAI/graphics systems where the students control the system, rather than the system controlling the students, a novel graphics capability is needed. In this case, the student programs the computer using a simple on-line graphics programming language to produce pictures and even motion pictures on a graphics terminal. Student controlled systems, where the student writes graphics programs, are certainly a break from the traditional use of graphics in education. Rather than using the computer as a delivery mechanism to teach the same old "knowledge," the computer and a graphics terminal are used to let the student "learn by doing." The objective is to develop critical thinking skills rather than merely to transfer information. (Papert, 1971a)

The variety of graphics capabilities is endless. Some educational objectives could possibly be accomplished more efficiently and with less cost by using a medium other than a CAI/graphics system. However, this paper is limited in scope to CAI/graphics systems.

Chapter II of this paper describes and analyzes existing CAI/graphics systems representing the different uses of graphics in CAI. Chapter III outlines the hardware and software parts of a CAI/graphics system, such as graphics terminals, communication links, author languages, etc., and discusses the various possibilities that exist for each part of the system. Chapter IV gives some guidelines on the selection of the different parts of the system to suit a particular CAI/graphics requirement.

CHAPTER II

DESCRIPTION OF EXISTING CAI/GRAPHICS SYSTEMS

The types of CAI/graphics systems that have been implemented are as diverse as the individuals who conceived them. The technology of CAI/graphics systems is still in the research and experimental stage as evidenced by the types of systems that have been developed. Many of the systems that have been implemented were used only long enough to be crudely evaluated, and then were abandoned, either because of lack of financial support or lack of conclusive evidence that the system was effective. There is relatively little student experience with graphics. Reliable guidelines for equipment selection and software development for a CAI/graphics system are non-existent. Courseware is generally custom-made for each individual system.

In this section of the paper, some CAI/graphics systems are described. The educational objectives, hardware, software, and courseware of these systems are discussed. The methods by which the systems were evaluated, if at all, are also mentioned. Conclusions regarding what can be learned from these implementations are drawn.

The types of systems to be described range from static or passive graphics applications to dynamic interactive graphics programs. The systems are presented in order from the simplest to the most complex.

A. BRENTWOOD TUTORIAL MATHEMATICS PROGRAM

This program was one of many implemented by the Institute for Mathematical Studies in the Social Sciences at Stanford University under the direction of Patrick Suppes. (Suppes, 1969) The goal of this program was to examine the feasibility of tutorial CAI in elementary (first and second grade) mathematics. Approximately 400 lessons covering counting, addition, subtraction, sets, and geometry were written.

The Stanford-Brentwood system uses an IBM 1500 computer, which was designed and built by IBM working with Stanford personnel. The 1500 was one of the first tailor-made CAI computers. The hardware included the central computer, IBM 1500, and several student response terminals which consisted of a Philco-Ford CRT unit, keyboard, light pen, film projector, and headphones with microphone. (Wilson, UD) The CRT unit was capable of presenting only alphanumeric characters and simple line drawings. If the student answered a question correctly, either by light pen pointing or by keyboard response, a picture of a smiling face would appear on the CRT unit; correspondingly, if the student answered incorrectly, a picture of a sad face would appear on the CRT unit. The system was used to teach the concept of "sets" to elementary school children by showing pictures of the members of a set and asking the student how many members were in the set.

The author language that was used in this system was Coursewriter II, developed jointly by IBM and Stanford. The features of this language are described in a later section of this paper.

During the initial evaluation of this program, observation of student behavior suggested changes to be made. Later, this program was evaluated by extensive statistical analysis, based on pretests, post-tests, and groupings of experimental and control classes. The results of the analysis were mixed, but slightly more favorable than unfavorable. Some of the unfavorable results were attributed to a response time sometimes in excess of ten seconds.

A random-accessed computer-controlled slide projector is a cheaper alternative for the type of graphics presented in this system. The slides could contain all the pre-defined pictures which were drawn on the CRT. The only justification this system has for a graphics terminal is the light pen input capability. To use a light pen would be easier for elementary children than to type a response via a keyboard.

B. UNIVERSITY OF ALBERTA DRILL AND PRACTICE MATHEMATICS PROGRAM

This program is similar to the Stanford-Brentwood program, except for some interesting variations in evaluation of the system. (Hicks, 1972) The objective here was to drill each student in addition and subtraction of the numbers 0-9 until 90% accuracy was reached with a response time less than 15 seconds. The problems were generated randomly. The hardware, IBM 1500 with Philco-Ford CRT units, and author language, Coursewriter, were the same as at Stanford. The graphics output types were as follows: a picture of Donald Duck to draw the attention of the student, large numbers in the statement of the problem, stars for correct answers, big X's for incorrect answers, and percent right indicated graphically.

The student inputs his answer via the keyboard. If the student answered incorrectly, the light pen was used. For example, if the student were asked the sum of 3 and 2, and if he answered incorrectly, he would have to use the light pen to point to 3 items and then 2 items.

The student would point to an item, and the cardinal number of the item would then be output, so that the student could actually count up to the correct answer.

This program was evaluated by sitting next to the student and observing him. After evaluation, light pen input after incorrect responses was eliminated. Another improvement made in the program, surmised by studying the student response record, was to have the program "learn" where the students made their mistakes and drill on those problems rather than randomly generating problems.

The "attention-getting" effectiveness of the CRT was evaluated by recording the eye movements of the students as material was displayed. The tentative conclusion reached was that "although a fancy TV screen display may be pleasing to the author, there is no guarantee that it is helpful to the child in the learning situation."

The same level of effectiveness, in this case, could probably be achieved by a random-accessed computer-controlled slide projector. The more elaborate graphics seemed to be ineffective. Graphically displaying to the first grade student the current percentage of correct answers was useless.

C. UNIVERSITY OF TEXAS CAI PROGRAM IN THE ARABIC WRITING SYSTEM

This program is slightly more sophisticated than either of the two previously described programs. (Abboud, 1971) Its main objectives were to teach adult students to write Arabic correctly in the shortest possible time, and to maintain interest and motivation in the study of the language. The ability to discriminate sounds from a computer-controlled tape recorder and the ability to read simple material are secondary objectives of this program.

The hardware used was the IBM 1500 system, including a light pen, which was used only to point to correct answers, and an IBM 1510 Instructional Display, which is a CRT for display of alphanumeric information. In addition to the standard alphanumeric characters, the user can define and load three new dictionaries of character sets or images, where each dictionary can contain 128 separate images. These images are defined in a dot matrix 12 dots high and 8 dots wide. These dictionaries of images are one form of "graphics" on this machine. "Graphics" sets can also be defined in the system, but only three sets can be used in any one course written in Coursewriter, the author language used here. Each set can contain 64 pictures which are encoded in a 36 x 16 dot matrix. Obviously, defining the pictures of the Arabic characters was a time-consuming task.

A novel technique of graphics instruction was used to teach Arabic writing. The student is asked to write a word, in Arabic, on the CRT screen in a "blanked out" rectangle using a felt tip pen. After the student indicates that he has completed the word, the correct form will appear behind or through his written form for purposes of comparison.

This method of instruction was evaluated by comparing the test results of the students taught by this method and those taught by two alternative methods. The length of time spent in instruction using the CAI/grahics system was on the average half as long as with the two alternative methods. The test results showed that the CAI/graphics system was significantly more effective in teaching the students to write Arabic than either of the two alternative methods.

This program is outwardly very elegant, with the exception of the felt tip pen "input" which the student must erase manually. Hardware is available that would let the student draw the Arabic word using a light pen and the image of what he drew could be "tracked" and output on the screen. Erasing would be under computer control. More sophisticated and more expensive CRT units are available, such that the task of creating the Arabic character set would have been a trivial task for the author.

D. THE UNIVERSITY OF ILLINOIS PLATO IV BIOLOGY LESSON

This program is different from those previously described because it is a simulation CAI application rather than the more common drill and practice or tutorial CAI program. (Hyatt, 1972) The objective of this simulation is to teach the laws of biological inheritance. The students perform the traditional fruit fly matings experiment in the simulated laboratory environment as displayed on the graphics terminal screen. The graphics output is pictures of the genetically different types of fruit flies, i.e. wingless, red-eyed, etc. The student selects two fly types he would like to mate by pointing to them with the light pen. The next graphics output is a picture of their offspring. The students collect data and maintain logbooks based on their observations of the types of offspring, just as they would do in a normal laboratory. The students can request matings for successive generations since recessive traits do not appear in first generation offspring. This simulated biology laboratory takes only minutes, whereas the traditional laboratory to perform the same experiments can take months.

The hardware used in this program was the CDC 6500 with a plasma graphics terminal. The features of this terminal are described in the hardware section of Chapter III.

The courseware was written by a biologist in the author language TUTOR, which is also described in Chapter III.

The software which is used in this program to transmit graphics information to the terminals, is designed efficiently. At the beginning of the lesson, the pictures of all possible fly parts are transmitted to the terminal and are maintained in local storage. When a new "fly" is to be drawn on the screen, the central computer simply transmits a control word that is encoded information (like a chromosome), which tells the terminal which fly parts to put together for a new "fly" and where to put them. This feature saved transmission time during the lesson.

This CAI lesson was not evaluated in the traditional manner using a control group and an experimental group. It was evaluated only by subjective observations of the author, who concluded that the simulated laboratory saved the students much busy work and time, and saved the biology department the cost of maintaining the actual laboratory equipment. However, the biology department probably incurred at least an equal cost in the computer time used by the students.

The lesson was only one of a CAI course in biology. The use of a graphics terminal in this lesson is justified because many different types of flies can be generated using random numbers and Mendel's Laws of Inheritance. A random-accessed computer-controlled slide projector could not do the job in this case.

E. THE UNIVERSITY OF NORTH CAROLINA NUMERICAL ANALYSIS CAI PROGRAM

This program was used in conjunction with traditional teaching to help develop insights and skills necessary in numerical analysis. (Oliver, 1969) It can be used either as a lecture aid or by individual students. The objective of this program was to provide meaningful illustrations in numerical analysis which would give students an intuitive understanding of the relationship between symbolic and graphic representation of equations.

The graphic outputs were plots of various curves. The students could dynamically change variables in a symbolic equation and have it plotted. The students could see how the change affects the curve by plotting both the original and modified equations simultaneously.

The hardware used for this system is the IBM 360/40 with an IBM 2250 Model I display unit as the graphics terminal. The terminal was directly connected to the IBM 360 by a channel. The IBM 2250 is a dot matrix 1024 x 1024. Images are generated by programming appropriate points on the grid to be lighted. The IBM 2250 has a local memory which is used to refresh the display. Vectors are drawn by specifying (X,Y) coordinates of end points. A light pen, alphanumeric keyboard, and function control keys are used for user input. The function control keys are used for many things. For example, one control key labeled INST displays the general instructions when it is depressed.

The author language of this program was 360 assembly language and some PL/1. The graphics program ran in a 44K partition of the 360 memory. The graphics program was multi-programmed with other jobs running in the 360. The only constraint was that the graphics program had to have highest priority so that none of the light pen interrupts would be missed.

This program was evaluated by having a control group which received the traditional instruction and an experimental group which received the same amount of traditional instruction, but in addition was given access to the terminal. Pretest differences were taken into account during the final statistical analysis of the post-test. For example, the results of the post-tests of the students in both groups that had similar scores on the pretest were compared. The results showed that the students that used graphics developed a better intuitive understanding of numerical analysis than the control group.

The evaluation technique used in this program is slightly questionable. The experimental group received more hours of instruction than the control group, so that variable alone would seem to justify their better performance on the post-test.

F. UNIVERSITY OF CALIFORNIA AT IRVINE PHYSICS LESSON

This CAI/graphics lesson was used in addition to regular lectures. (Bork, 1971a, 1971b, 1972) Its purpose was simply to demonstrate one concept, the laws of motion, in a college physics course. The program allowed the students to launch a lunar lander from the earth to the moon, where the initial velocity and angle of the launch were the variables chosen by the student. The graphics output was a picture of the earth, the moon, and the path of the currently active lunar lander. The students could see that his lunar lander missed the moon, crashed into the moon's surface, or landed successfully. Depending on the results of one lunar landing attempt, the student could change the velocity and/or angle of launch and see how the path of the lunar lander was affected.

The hardware used for this program is a SIGMA 7 computer with either a Tektronix 4002 or an Adage ARDS 100 graphics terminal. Since program communications with the two different types of graphics terminals are different, the student has to identify which type of terminal he is using when he signs on. The operating system provided with the SIGMA 7 was changed by the author of the lesson in order to support graphics. For example, the operating system automatically inserted carriage returns into the output stream after enough characters to fill one line going to a terminal were output. The internal code of the SIGMA 7 is EBCDIC, and the code recognized by the terminals is ASCII. The graphics output is sent to the terminals over a 1200 baud line, which the lesson author indicated was adequate to support graphics output which does not have to be refreshed from the central computer. Graphics input can not be handled in this program.

A new author language for graphics was developed for this program and for future use. The language is called METASYMBOL, which is a macro-assembly type language with embedded FORTRAN routines. An attempt at evaluating this program was made, but the conclusion reached by the author was that the physics talents needed in the non-graphics version of the program were different from those needed in the graphics version. In the non-graphics version, students took the printed output for an attempted lunar landing and calculated the requirements for a more successful launch. In the graphics version of the program, students just relied on the curves to get a more intuitive feeling for the laws of motion. The author "suspects" that the students learn more using the graphics version.

The terminals used in this lesson are limited in their capabilities. For example, selective erasure or animation is not possible. However, these terminals are the cheapest graphics terminals. It seems that insufficient funds was one of the constraints of this project. Therefore, since graphics input is not even used here, a small hard-copy terminal plotter with a teletype is cheaper than the graphics terminals, and would provide the same plots in just slightly longer time.

G. THE UNIVERSITY OF ILLINOIS GEOMETRY PROGRAM

This program was designed to give high school students experience with facts of geometry before introducing them to formal proofs. (Dennis, 1969) A student could draw geometric figures on the graphics terminal by moving the cursor about. The student is asked to draw a specific type of geometric figure. Then after the student has drawn the figure by moving the cursor, a sophisticated pattern recognition algorithm is executed to judge the response of the student. The geometric figure was drawn as follows: a matrix of dots, about 1/8 inch apart, was output on the terminal screen. Then the student could move the cursor, a bright spot on the screen, in any one of eight different directions by pushing one of the direction function keys. When the student has positioned the cursor to the desired dot in the matrix, the "mark point function" key would be pushed. After the student has located the next point and "marked" it, a line segment is automatically drawn between the two most recently marked points. If the student wants a closed figure, a "close function" key is depressed which draws a line segment between the first marked point and the last marked point.

The hardware used for this program was a CDC 1604 computer, a CRT, and a function keyset. The characters and lines were drawn by plotting points in the desired configuration. The program was written in the CAI author language TUTOR.

The graphics input in this program could have been accomplished in an easier fashion by using an additional piece of hardware, the joystick. The student draws a line by pushing the joystick in the desired direction of the line. Another disadvantage of the graphics hardware here was that circles or curves could not be input, a serious deficiency in a geometry lesson.

H. CATTS: COMPUTER-AIDED TRAINING IN TROUBLE-SHOOTING

The objective of this CAI/graphics program is to teach the students how to trouble-shoot some electronic and mechanical systems. (Landa, 1972) This system did not teach any new concepts, but taught the students how to assimilate their bits and pieces of knowledge into an efficient and effective trouble-shooting process.

The most noteworthy feature of this system is the way in which the author encoded the graphics information to be output during the program. The author was relieved of the tedium of specifying in detail each point on each frame for the entire program. The author was allowed to simply draw on an "electronic tablet" the set of pictures which would be used somewhere in the program. Then the control program for the "electronic tablet" built a file in the central computer which contained a description of the picture currently being input. This technique let the author create interactive graphics programs without having to learn the intricacies of graphics subroutines.

The author language which was used to create this program is called POGO, Programmer Oriented Graphics Operation. One of the strong features of POGO is the way in which pictures are input as described above. This feature facilitates on-line interactive graphics programming. However, POGO has no CAI features. (Boehm, 1969)

The hardware used for this system, in addition to the electronic tablet for graphics input, is an IBM 360/40, and IBM 2250 graphics display console with light pen, keyboard and function keys.

The student interfaces with the system using the light pen on the CRT. He points with the pen to the suspected trouble spots in the system which he is troubleshooting.

The system was not evaluated with the exception of noting that had the students been taught using real systems, the actual equipment would have been expensive and each student would not have had a system of his own to debug.

This system is really an elegant CAI/graphics set-up. The only disadvantage is the cost of the system; the graphics terminal alone rents for $2550 per month.

I. THE UNIVERSITY OF CONNECTICUT MUSIC PROGRAM

The objective of this program was dependent on the abilities the student already had acquired. (Heller, 1971) If the student had not yet mastered a musical instrument on which to play his own composition, the student could still compose music, in a nontraditional manner, using the graphics terminal as his input device, and could hear his composition immediately played back. If the student was already familiar with basic music theory, this program could illustrate advanced concepts by graphical means.

The hardware for the system was an 8K PDP-9, an 18-bit machine, and a type 339 CRT with light pen. The student could draw a music graph on the CRT with the light pen. The high and low spots on the graph of the music would represent the high and low notes respectively. Longer plateaus in the graph would represent notes of longer value than short plateaus. The graph could be continuous or broken, and the music would be affected in a similar manner. The pitch of the tone depends on the relative vertical position in the graph. The graph of the music could be played back immediately by converting the graphics information into signals which can drive a device called a CASS (Computer Assisted Synthesizer System) which would produce music. The speed and range of the notes could be varied by the student. Music lines could be saved and concatenated with other lines to produce a complete composition.

The program was very modular; the student could enter different modes of operation, such as composer, music editor, etc. The program was probably written in assembly language because only 8K was available, although the author does not say.

This unique technology was evaluated by teaching an experimental group of fifth and eighth grade students using CASS. Two control groups were maintained; one group was taught by conventional vocal instruction, and the other was taught instrumental music. The objective of a series of well-documented lessons was to teach music notation and performance skills. The performance medium was voice, music instrument, or CASS. The groups were given a series of pretests which provided additional information for evaluation of the final scores. After the results of the post-tests were analyzed, the author concluded that CASS and instrumental groups both achieved the same abilities in the fifth grade group, whereas vocal instruction was superior in the fifth grade groups. In the eighth grade groups, CASS and instrumental instruction again were equal in accomplishment, but the vocal instruction was inferior, which the author attributed to lack of motivation on the part of the eighth grade vocal group. So CASS instruction was equal to instrumental instruction for the same length of time.

This CAI/graphics system certainly has some powerful features, but in this case, the objective for musically untrained students seemed inappropriate. Why should a fifth grade student learn to perform music on a CASS? In a college level composition class, this system would be of more value.

J. MASSACHUSETTS INSTITUTE OF TECHNOLOGY ELEMENTARY MATHEMATICS PROGRAM

This lesson is one of many which can be learned at MIT's CAI laboratory. (Papert, 1971b, 1971c) The objective here is that by giving students "tools" such as the CRT and drawing commands, he will discover the laws of geometry by drawing a lot of different geometric figures. The educational philosophy employed is that children learn by doing. So rather than having a "canned" CAI program to teach specific things, they have a basic system on which the students write and execute programs. The students learn by debugging their programs and by seeing the results of the program changes they make.

"Seeing the results" in this particular lesson involves seeing the graphics output on the CRT. The student draws pictures by moving the cursor, using commands such as FORWARD x(units), BACK x(units), LEFT x(degrees), RIGHT x(degrees), PENUP, and PENDOWN. The meanings of these commands are obvious. The language is called LOGO. The student can define routines with input parameters and call them recursively. For example, the student can define a routine with DISTANCE as the input parameter as follows:

```
TO DRAW:  DISTANCE
1 FORWARD:  DISTANCE
2 BACK:  DISTANCE
END
```

This routine can then be executed by saying DRAW 100. If the CRT is in the PENDOWN state, a line 100 units long will be drawn on the screen when FORWARD: DISTANCE is executed. Then the line will be retraced when BACK: DISTANCE is executed.

The classic example of a geometry lesson is represented in the following routine which has two input parameters, STEP and ANGLE:

```
TO POLY:  STEP:  ANGLE
1 FORWARD:  STEP
2 LEFT:  ANGLE
3 POLY:  STEP:  ANGLE
END
```

When this routine is called POLY 150 120, an equilateral triangle is drawn with sides 150 units long. When this routine is called POLY 75 60, an equilateral hexagon is drawn with sides 75 units long. The output of these two different calls and the student's desire for creating "interesting" pictures will certainly cause the student to think about why each figure is drawn.

An interesting variation of this routine which produces fascinating results is as follows:

```
TO POLY:  STEP:  ANGLE
1 FORWARD:  STEP
2 LEFT:  ANGLE
3 POLY:  STEP+5:  ANGLE
END
```

Then when POLY 5 90 is executed, the following figure results:

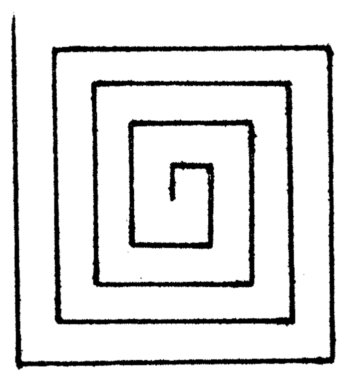

Figure 1

This system is currently in the development stage. The proposed hardware which will be used for this system is a PDP-11 with 5 display terminals and other output devices. (Papert, 1971a) Currently a prototype system is implemented on an overloaded research computer to which this CAI project has limited access.

This system was not evaluated because there is no comparable traditional teaching. The students that used the system showed a very positive attitude toward it. The system is actually a research project studying how children learn. The hypothesis is that children learn by doing, so this system represents an attempt at finding better things for children to do than the traditional educational activities.

The graphics terminal is just one of the new toys for children associated with this system. Other programmable toys are available. Evaluation of this project is difficult. Specific lesson objectives are non-existant so even a post-test to see if students met the objectives is irrelevant. This system is perhaps best used only as a source of motivation for some better organized form of instruction.

K. POINT PARK COLLEGE

SURGERY LESSON

The objective of this program is to teach medical students surgical techniques by giving them access to an elaborate surgical simulation environment. (Canter, 1971) The student surgeon can conduct a simulated operation and graphically view the progress of his activity.

There were two types of graphics terminals used in this project: 1) an Adage ACT/10 two-dimensional terminal and 2) a full-color 3-D CRT. The output for the 3-D terminal was assembled from thousands of digitized photo images stored on a peripheral device of the main computer. The main computer was time-shared, multi-user system, where each user had access to the two types of CRTs. A light pen and function buttons were used for the input devices.

The graphics interaction would begin by the student selecting an operation he wanted to perform. The picture of the prepared patient is displayed on the 3-D color CRT. On the two-dimensional CRT are animated-simulated graphs showing the patient's heart rate, pH levels, respiration, etc. The student can define the current use of the light pen by pushing a function key, such as scalpel, clamp, sponge, etc. Then when the student touches the light pen to the CRT it performs the function that was most recently assigned to it. When the student defines the light pen as a scalpel and draws it across the prepared patient on the 3-D CRT, bleeding is shown on the CRT. The student goes through all the steps for an entire operation. His procedures are compared against a pre-established norm, and if he varies from the norm in excess of a fixed maximum, then the program goes into "tutor" mode and reverses the operation back to the point where the student made his first mistake and begins questioning his actions and giving the student hints. Random life-critical events are generated during the course of an operation.

The development of this program is still going on, but it was mentioned here to show the wide range of graphics capabilities a CAI/graphics system can have.

The evaluation of this simulation, as in any simulation, is difficult to do because the identical learning situation is not available in the real world. The advantages of simulation in this case are obvious; however, the simulated operations are not a complete replacement for practical exposure to surgical procedures.

As shown there is quite a variety of CAI/graphics systems. The justifications for each of the graphics systems include at least one of the following:

1. Students seem to be motivated more when graphics are used.

2. Lots of information cannot be easily or clearly represented in words, i.e. maps, mathematical graphs, cardiograms, etc.

3. Pictures more fully utilize the student's eyes, which are one of his input devices for learning.

4. "A picture is worth a thousand words."

In the next chapter, the hardware and software components are studied in detail. This includes author languages, courseware, graphics terminals, etc. The various ways in which each component can be realized are discussed.

CHAPTER III

THE COMPONENTS OF CAI/GRAPHICS SYSTEMS

A computer assisted instructional graphics system is composed of two major components: hardware and software. In this section of the paper, each component of a CAI/graphics system is discussed in detail.

A. HARDWARE

The hardware generally consists of a central control computer and a number of graphics terminals tied to the central computer through some sort of communications link. See Figure 2.

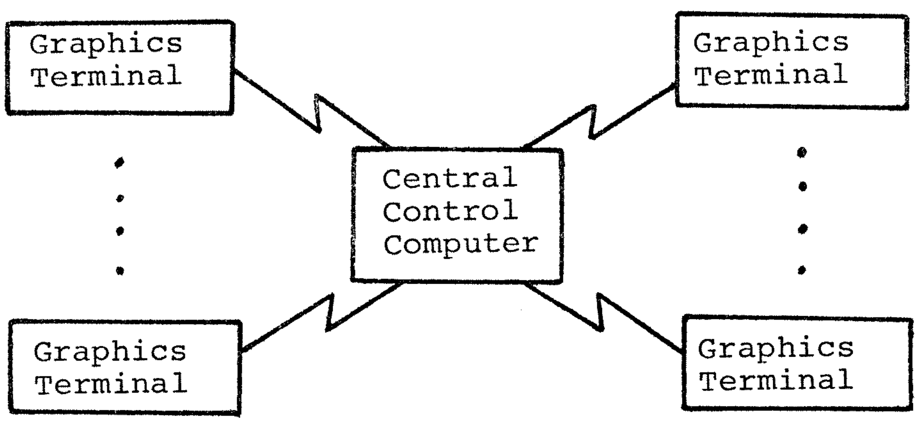

Figure 2

The function of each part of the system can be performed in a variety of ways. Some of the possibilities for realizing the different parts of the system are described below.

Graphics Terminals

The basic elements of the graphics terminal are the display device, display control, the user interface, programming characteristics, and peripheral storage equipment.

DISPLAY DEVICES: The most commonly used display device is the cathode ray tube (CRT). The basic component of the CRT is an electrode gun which shoots a beam of electrons onto the inner surface of the CRT screen, which is coated with one of several compounds composed of rare-earth elements. These compounds are called phosphors. When electrons bombard the phosphors, they glow for a short time. The length of time the phosphors will glow and the color of the glow is dependent on the compound used for the phosphors. The deflection of the beam of electrons is controlled by two sets of electro-static or electro-magnetic deflectors, which affect the (X,Y) positioning of the beam. One disadvantage of the CRT is that the phosphors must be refreshed or re-bombarded at regular intervals in order to maintain the picture on the screen, otherwise the picture will fade. Many CRTs must be periodically refreshed from a central computer which maintains an image of the picture on the screen. This process creates a heavy burden on the central computer because it may be controlling many CRTs. Some CRTs have their own memory in which an image of the picture is maintained and, therefore, do their own refreshing at regular intervals. However, to provide a memory for each CRT can be rather expensive.

A variation of the refreshing CRT is called the direct view storage tube. To the user, a direct view storage tube acts like a refreshing CRT with the deflection system and phosphor coating similar to the refreshing CRT. However, rather than writing directly on the screen as with the refreshing CRTs, the electron beam of the direct view storage tube is directed toward a fine mesh wire grid mounted behind the screen. The wire grid, which is coated with a nonconducting material, acts like a memory from which refreshing can occur. By using an electron gun, the grid is positively charged at the points which comprise the picture. Then a flood of electrons coming from a second electron gun continuously copies what is on the grid onto the phosphor coated screen. Refreshing, as is described above, is not necessary in this case because the image is maintained on the wire grid. According to Newman (1973) the disadvantage of the direct view storage tube is that selective erasing is hard to program. Another disadvantage is that the "flooding electrons" gradually charge the entire wire mesh, and the picture that was there becomes invisible after about an hour.

A modification to the direct view storage tube that makes it highly desirable is the use of the raster scan to copy the image on the screen. Instead of using a continuous flood of electrons to copy the grid image to the screen, two beams of electrons scan alternating rows of the grid, continuously copying each point onto the screen in a top to bottom, left to right pattern (raster scan). This interlaced scan reduces flicker. The raster scan technique can also be employed to read the wire mesh, and then a signal can be generated to drive television receivers. The same pattern can be sent to any number of television receivers.

Another device for image storage and viewing is the silicon target tube. The silicon target tube uses a plate coated with silicon as a target instead of a wire mesh as in the conventional direct view storage tube. Generally, the tube is very small and used only for storage. The image is scanned, and the signal generated from the scanning can be used to drive a television receiver. A "silicon storage CRT" has a silicon storage tube and a television type monitor for the display. The charge on the silicon plate at each point can be varied, thus creating a way of shading a picture. The field of view of the scanning process can also be varied, thus creating a means to scale pictures.

One display device that does not use a CRT in any way at all is the plasma display memory panel. Two sheets of glass are sealed together with a thin sheet of a neon-based gas in between. Parallel electrodes covered with a nonconducting material are attached to the inside of each sheet of glass. On one sheet of glass the electrodes are vertical, and on the other sheet of glass they are horizontal. Then, according to Newman (1973), "by applying voltages between the electrodes, the gas within the panel is made to behave as if it were divided into tiny cells." Each cell can be made to glow independently and is directly addressable using an (X,Y) coordinate. The advantage of this type of device is that it does not need to be refreshed, thus saving the cost of a local memory. The plasma panel is transparent, which permits the use of rear-projected data from color slide microfilm to be projected onto the panel.

DISPLAY CONTROL: Many of the features provided by some graphics terminals manufacturers, such as character generation, line generation, function keys, etc., are hardwired logic, while other manufacturers provide these features via a display processor. The display processor decodes and executes instructions fetched from either ROM (read only memory) or its own core memory, into which the user has loaded his program. Oftentimes, the ROM has standard subroutines for predefined features, such as vector generation, rotations, scaling, etc. The display processors are usually 16 bit computers, sometimes with an elaborate interrupt structure. These display processors usually are coded in assembly language, but some manufacturers provide a FORTRAN compiler.

USER INTERFACE: The user generally interfaces with the graphics program via a keyboard, which provides an alphanumeric character set. The program in turn writes on the graphics terminal screen using the same character set to communicate with the user.

Besides character keys, control keys are often available. The control keys vary from one manufacturer to another. The commonly available control keys include: blank display, backspace one character, horizontal and vertical tab, repeat key, interrupt display processor, carriage return, and general function keys, which can be defined by the user program in the display processor. Control character responses are either hardwired or stored in one of the memories of the display processor.

In some cases, the keyboard is an inadequate means of user communication. For example, the user may wish to point to a symbol on the screen. Most terminals are provided with a small cross or <u>cursor</u> that is visible on the screen. By various mechanical means, the user moves the cursor about the screen. The position of the cursor is directly or indirectly available to the display processor or to the central computer. A CRT can respond to an inquiry command sent to it from the central computer or from the display processor, by returning the (X,Y) coordinates which directly describe the present cursor position. In some instances only the keyboard control characters such as space and carriage return and horizontal and vertical tabs are available for controlling the movement of the cursor. The program in the central computer or display processor must input all the space and tab characters that the user transmits by pushing those character keys, and from them determine the cursor position.

User interface devices which lessen the burden of moving the cursor about have been developed. One such device is the joystick, which is a mechanical device similar to a hand lever that moves back and forth and sideways. The lever outputs (X,Y) coordinates each time it is moved. These (X,Y) coordinates are fed back to the terminal and cause the cursor to move appropriately.

Another device that issues (X,Y) coordinates for the cursor position is the mouse, which is a small box on wheels. The mouse is moved about by hand, and each change in its position is translated into (X,Y) coordinates which are sent to the CRT. The cursor on the screen is then appropriately moved to reflect the movement of the mouse. The cursor movement may be done by the display processor or the central computer.

The tablet is another device that is used to control the cursor movement by the user. Similar to the mouse, the tablet transmits (X,Y) coordinates back to the CRT. The user draws on the tablet with a special stylus. The tablet is so constructed that it can detect where the stylus is touching the tablet, and translates that position into (X,Y) coordinates for cursor positioning.

An acoustic tablet uses a stylus that generates a sound that is detected by microphones. The position of the stylus with respect to the sides of the tablet generates the (X,Y) coordinates, but by lifting the stylus up, a third coordinate, Z, can be determined for three-dimensional control of the cursor.

On the graphics terminal, a direct means of pointing is provided by the light pen. A light-detecting pen is held in the hand and pointed to the display screen. A photocell in the light pen will set a flip-flop that can be read by the display processor or the central computer whenever the pen is pointed to a light source. The light pen is useful for pointing, but if it is used for positioning or drawing, a tracking program must be provided which is sufficiently fast to keep up with the changes in the pen's position on the screen.

PROGRAMMING CONSIDERATIONS: The hardware differences of graphics terminals create the need for various programming techniques to use the terminals effectively. Some basic programming considerations such as character generation, straight line generation, curve generation, and 3-D picture generation are discussed here.

Character Generation: Many less expensive CRTs have no direct means of writing characters. The program of the central computer or display processor must output enough (X,Y) coordinates to generate points in the pattern of a character. Some CRTs have hardwired dot character generators which use a seven by nine matrix of dots for each character.

Another method of character generation is the shaped beam. The character set available is etched into a thin metal sheet. (Chase, 1970) The program issues a code to the CRT for each character to be written. When a particular character code is received, the electron beam is directed to the corresponding etching of the character which is then copied onto the screen. This approach is limited in that once all of the character etchings have been mounted in the tube, they cannot be changed.

Characters are also generated in graphics terminals by the stroke generator. Short strokes are written to represent a character. The pattern of the strokes for each character is part of the control logic of the terminal. When a character code is received, the control logic directs the beam in the shape of the character.

The dot and stroke generators are the most commonly found in graphics terminals. A separate memory diode matrix or integrated circuit is required to store the character shapes for both dot and stroke generation. Some manufacturers provide various character sets by simply changing the memory boards.

Most CRTs use ASCII character codes; many use extended ASCII; and some use EBCDIC code. Some manufacturers offer upper and lower case, bold face, and italic character types. The extended ASCII character set is sometimes required for compilers, like APL, which is often used as an author language, and should be considered when purchasing a graphics terminal.

Straight Line Generation: Straight lines are generated by two different methods: 1) a series of dots are written to represent a line or 2) a line is drawn by the continuous movement of the CRT beam from the start point to the end point. Some CRTs require the program to issue the dots needed to draw a line. Other CRTs only require start and end points of the line, and the CRT hardware generates the needed dots. In the case of the beam movement, only the start and the end points are required. An analog circuit is used to move the beam. Some manufacturers feature vectors that can be solid, dashed, or blinking.

Curved Line Generation: With the vector generation many manufacturers offer circular arc generation, ellipses