# DESIGN OF AN INTERACTIVE VIDEO DISC-BASED LEARNING SYSTEM

by

Michael Dennis O'Keefe

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1985



Signature of Author ______________________________
Department of Electrical Engineering and Computer Science
June 3, 1985

Certified by ______________________________
Dr. Edwin F. Taylor
Thesis Supervisor

Accepted by ______________________________
Professor David Adler
Chairman, Undergraduate Thesis Committee

DESIGN OF AN INTERACTIVE VIDEO DISC-BASED LEARNING SYSTEM

by

Michael Dennis O'Keefe

Submitted to the Department of Electrical Engineering and Computer Science on June 3, 1985 in partial fulfillment of the requirements for the degree of Bachelor of Science.

## Abstract

This thesis describes the development and implementation of a computer controlled interactive video disc program. The software was developed on an IBM XT personal computer equipped with a Visage system video disc controller and graphics board. The intent was to develop a program that could be easily expanded or modified. The particular topic chosen for this program was developmental biology. The program is designed to interactively teach the user specific basic principles, much the way a lecturer would.

Thesis Supervisor: Dr. Edwin F. Taylor

Title: Director, Educational Video Productions, Senior Research Scientist, Physics

# Dedication

I'd like to thank Professors Penman and Hynes and Rhonda Wilson for their help with the biological aspects of this research. The final program could not have been realized with my limited knowledge of biology alone.

Thanks to Dr. Edwin Taylor and his staff at M.I.T.'s Educational Video Productions for their support of this thesis. Without Dr. Taylor's committment, in both equipment and enthusiasm, this project would not have come to fruition.

Lastly, I'd like to express my sincere appreciation to my parents and Professor William Peake. Their faith in my abilities and understanding of my problems got me through early difficulties at M.I.T. Without them, the realization of a degree from M.I.T. would have remained only a dream.

# Table of Contents

# List of Figures

# Chapter 1

## Introduction

Education of people is both a right and duty of any civilized society. As technology changes new educational possibilities arise. The invention of the printing press offered the means for recording and passing on acquired knowledge. The book, today, remains a primary teaching device. Recent developments, such as video recording, have introduced new media for the transmission of information. The computer age has brought with it a wealth of educational software aimed at both adults and children. Few school systems remain that do not introduce their students to computers.

With recent developments in both video disc and computer technologies, new avenues of learning and educational development have been opened. Smaller, faster, and more powerful personal computers have given rise to a wealth of *computer based instruction*, or CBI software. Using this software one can study real estate management, prepare for the SAT's, or learn to fly a plane. Most of these programs present the user with textual information or text combined with graphical simulations. What all these programs lack is a sense of realism. Users often become bored with the inundation of facts presented to them. Without realistic visual mnemonics the user is relegated to memorization of vast amounts of facts.

The laser (optical) video disc offers the possibility of major improvements. A video disc can contain 54,000 individual video frames or images. These frames can be 54,000 different still images or 30 minutes of moving video, or, more commonly, a combination of both. In addition, two audio channels are available on the disc. These features differ little from those of a standard videotape. However, unlike a videotape, the video disc offers rapid random access. The disc player can locate and display any one of the 54,000 frames in little over a second. This is a worst case time. Frames that are located close to one another can be accessed almost instantly.

When combined with a computer the video disc becomes an extremely powerful educational tool. The computer, under software control, can locate and display any location on the video disc. Using applications software one can interact with the computer to call up video images as well as overlay text and graphics on top of the images. With the addition of the video and audio information, the learning experience becomes exciting and more lasting. As AT&T would say, "it's the next best thing to being there".

Currently there are a number of educational video discs on the market. The subject matter on these discs ranges from art in the National Gallery in Washington to the life sciences. Most of these discs, however, lack any easy way for the viewer to access the information on the disc in an organized fashion. One must manually enter the frame location into the disc player each time a new frame is desired. With tens of thousands of images on a single disc, this system of access represents a major investment of time and is extraordinarily inconvenient.

This thesis will involve the development and implementation of original software for a computer driven interactive video disc system. The software will demonstrate the educational capabilities of a personal computer connected to a video disc player. The software will provide a basis on which future additions and modifications can easily be made. The project will be developed on an *IBM XT personal computer* equipped with a *Visage* video disc controller and graphics board.

# Chapter 2

# Technical Description of System

This chapter will describe in some detail the hardware and software utilized in the development of this program.

## 2.1 The Computer

The computer used to develop and run the software was an *IBM XT* personal computer. An *IBM PC* could have been used in place of the *XT*, but the presence of the 10 megabyte hard disk on the *XT* provided a significant convenience in terms of speed. In addition, the *Visage* system will run on any *IBM compatible* computer such as the *COMPAQ Personal Computer.*

The computer was equipped with a full 640 kilobytes of memory to aid in developing lengthy programs. The actual *Visage* system and the program that was developed can be run on much less memory.

A color monitor is required. However, in program development, a two monitor setup was employed. A *Zenith* high resolution color monitor with RGB video inputs and an *IBM monchrome display* were connected to the computer. The two monitor system allowed all images, video, graphics, and text, to be displayed on the color monitor, or, alternately, the text could be separated from the other images and displayed on the monochrome monitor.

A *Microsoft* mouse was connected to one of the serial communications ports of the *IBM*. Serial mouse interface software was provided by *Microsoft* with the mouse. Although *Visage* supports a number of other X/Y input devices, the mouse represented the cheapest alternative, while still satisfying the needs of the program.

All programs, graphics, and data files were stored on the *IBM XT's* 10 megabyte hard disk. This allowed for easy and rapid access.

## 2.2 The Video Disc Player

The video disc player that was used for this program was a *Sony LDP-1000A* industrial player. This player supports all the standard commands such as forward, step forward, slow reverse, search, etc. The *Sony* player was chosen because of availability, not because of any specific advantages over other players. The *Visage* system supports a wide variety of consumer and industrial disc players.

## 2.3 The Visage System

The *Visage* system is a hardware/software system that allows for development of sophisticated, interactive video disc applications. Using the system, one can creatively combine the special capabilities of a personal computer, industrial or consumer video disc player, and advanced graphics software.

The *Visage* system is an interface product. Using an *IBM Personal Computer* or a compatible computer for control and coordination, it links a video disc player and the color monitor on which the contents of the video disc are displayed. The *Visage* software can be used to develop interactive video programs that combine video display, sophisticated graphics overlays, and user input.

### 2.3.1 Visage Hardware

The *Visage* hardware is contained totally on a circuit board that plugs into two of the expansion slots of the computer. The circuit board contains the hardware that controls the video disc player, color graphics, and the overlaying of the graphics onto the video disc image. The *Visage* card interfaces with the video disc player by means of two cables. One cable carries the video signal from the disc. The other communicates control data to the disc player and receives information such as frame numbers from the player.

Figure 2-1 shows a layout of all of the hardware components of the system. This figure does not indicate the only possible setup, but merely the option that was chosen for this project.

### 2.3.2 Software

To develop an interactive video program, essentially two distinct software pieces are required from *Visage.*

The first, called *V:Exec*, is a set of software facilities that allows a high level language application program to communicate with the *Visage* hardware. This interface allows the application program to execute commands that control the video disc player, graphics, and X/Y input device.

The second piece of software is called *V:Paint.* This package allows creation of graphics that can be incorporated into an interactive video application. Essentially, this program is a graphics "paint" program that allows creation of images that can be stored in files on disk for later use in applications programs.

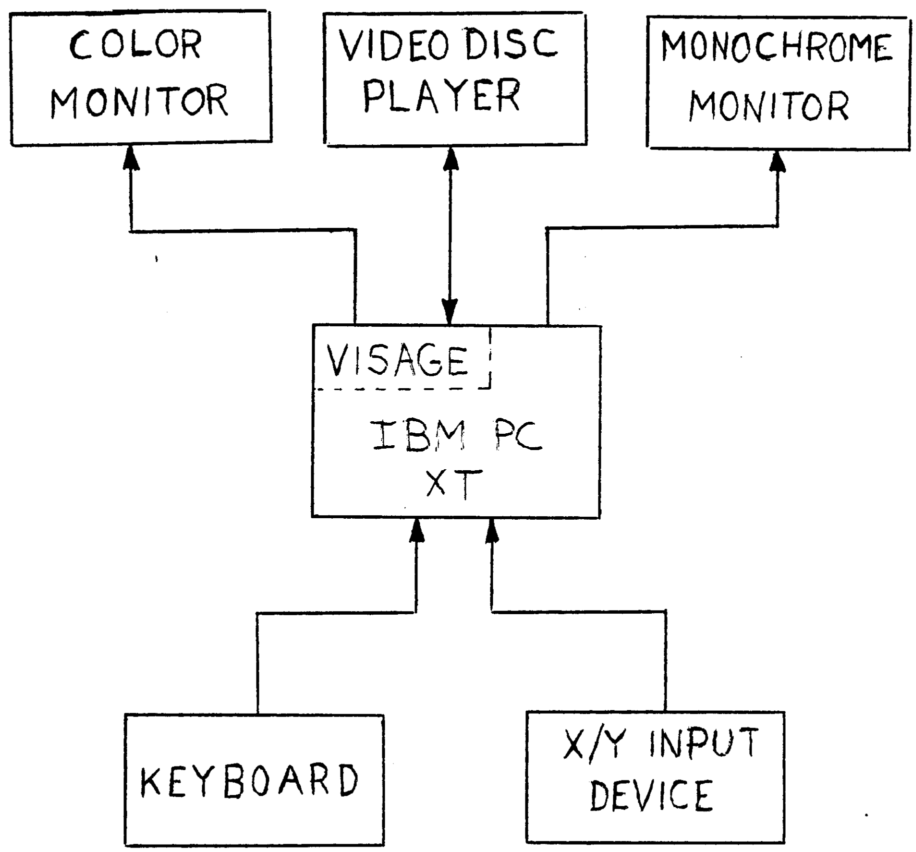



**Figure 2-1:** Hardware System Schematic Diagram

### 2.3.3 Graphics

The *Visage* system gives the user the ability to combine video originating from three different sources: the external video from the video disc player, computer graphics generated by a TI9128 graphics chip contained on the *Visage* board, and computer graphics generated by a *Color Graphics Adapter* which is software and hardware compatible with the standard *IBM Color Graphics Adapter* card (referred to as CGA graphics). The video monitor can be thought of as a *viewing window* as shown in Figure 2-2. Graphics that are displayed on the monitor exist as a series of superimposed "planes". Each successive plane can "cover over" parts of the plane beneath it. The *Visage* software permits each component to be switched on and off independently, defines which plane will take precedence over, or overlay other planes, and allows specified areas or colors within each plane to be switched to transparent, allowing planes to combined in an infinite variety of ways.

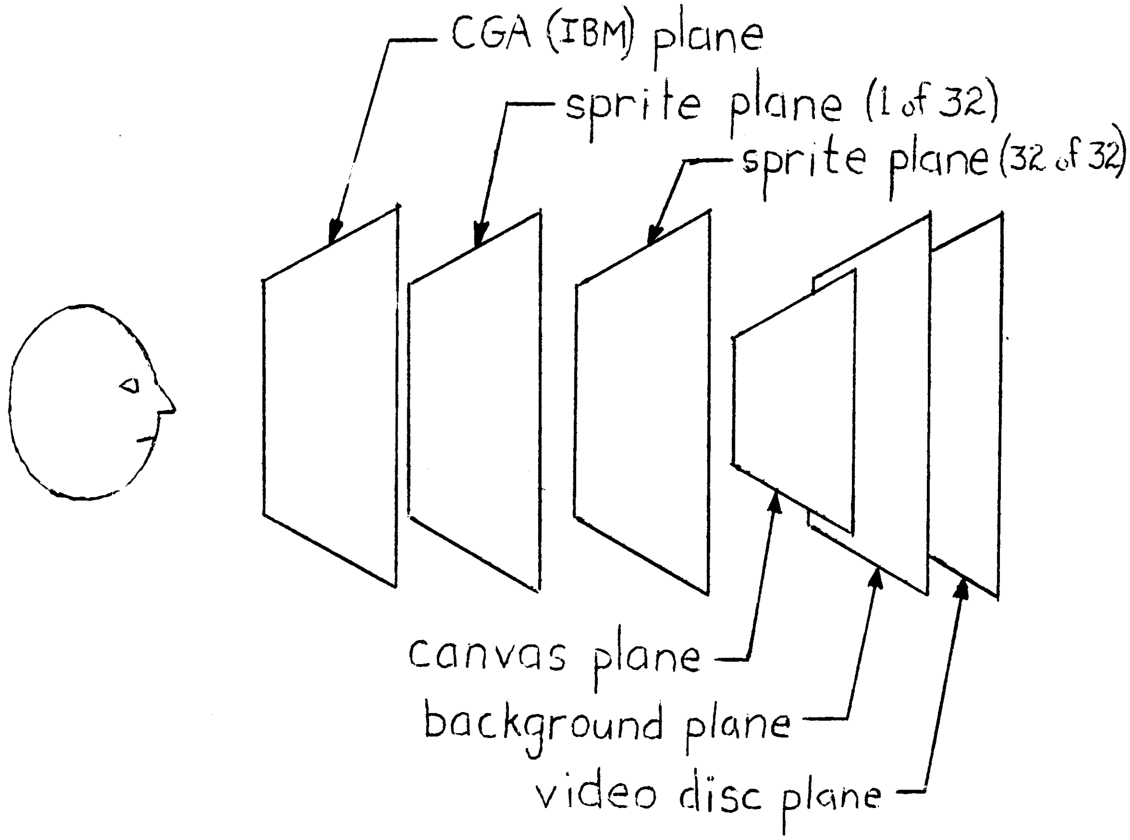



**Figure 2-2:** Graphics Planes of the *Visage* System

**2.3.3.1 TI Graphics**

As mentioned above, the *viewing window* is made up of a number of planes. One of the planes, the TI graphics, is itself made up of distinct layers as shown in Figure 2-2. Graphics that are part of this TI plane can consist of a background, a canvas, and sprites.

The *background* plane is a solid plane the size of the entire video monitor. When the background is set to one of the sixteen available colors, the entire screen is filled with that color. Included in those sixteen colors is a *transparent* color. When set to transparent, the background will become invisible and any graphics and the video disc image will show through.

The *canvas* lies on top of the background, and extends across the screen, but does not fully extend from top to bottom. Any fixed (i.e. motionless) graphics can be created in this layer. Any combination of the sixteen colors can be used to create images. However, due to hardware constraints, each horizontal group of eight pixels can contain no more than two colors.

A *sprite* is a set of movable graphics images that can be displayed on top of a canvas. Sprites offer a way to incorporate animation and movement into the graphics. Under program control, the *Visage* software offers simple commands that allow sprites to be moved around the screen.

**2.3.3.2 CGA Graphics**

The CGA graphics plane consists of any graphics or text created by the *IBM* graphics card. These graphics are created in the application program itself. Like other graphics planes, parts or all of the CGA plane can be made transparent to display images underneath.

# Chapter 3

## Advantages of Visage system

There are a number of companies marketing interactive video disc systems. The capabilities and features of each are quite varied. This chapter will discuss some of the advantages of the *Visage* system that led to its selection for this project.

### 3.1 Compatibility With *IBM PC*

One of the strongest advantages of the *Visage* system is its ability to run on a computer that has practically taken over the business and industrial market of personal computers. The *IBM* personal computer is widely viewed as the "standard". In addition, the *Visage* system runs on most of the *IBM* compatible personal computers. This covers a significant portion of the business and industrial community.

In order to use a program developed on the *Visage* system, an owner of an *IBM PC* need only acquire the *Visage* package. The *IBM* must have at least 256 kilobytes of memory, although 512 to a full 640 kilobytes is recommended.

The issue of penetration of the *IBM* personal computer is not to be taken lightly. The goal of every program designer is to have his work utilized by as many people as possible. To invest large amounts of time, effort, and money in a system that will either become obsolete or under-utilized seems fruitless. The lack of a "standard" all too often leads to downfall of the less accepted models. The consumer video disc player is a perfect example. The lack of a standard for video discs has forced a number of manufacturers to abandon their efforts in this field.

Other interactive video systems, such as Digital Equipment Corporation's *IVIS*, require a VAX mainframe computer to do software development and a *DEC Professional/350* with an *IVIS* backpack to run the completed program.

### 3.2 Cost

The cost of the complete *Visage* software/hardware package, including *Microsoft* mouse, is approximately $3,000. This, of course, does not include the price of the *IBM PC*. Adding the cost of the *IBM XT* brings the total to about $6,000.

By comparison, the DEC *IVIS* system costs approximately $12,000. Purchasing that system alone only allows you to run completed *IVIS* programs. Any program development requires access to a VAX mainframe as well.

The cost, like *IBM* compatibility, is an important issue. Since most business, industrial, and educational institutions already own an *IBM* personal computer, the additional cost of the *Visage* system is minimal. In most cases, introducing interactive video disc training programs would be much more cost effective than scheduling training sessions with instructors.

### 3.3 Input/Output Flexibility

The *Visage* system is extremely flexible in its ability to handle I/O devices (i.e. X/Y input device, video disc player, video monitors). The *Visage* software handles I/O in the proverbial "black box abstraction" fashion. That is, program developers can write their code without any knowledge of what specific I/O devices will eventually be used.

The *Visage* system supports the mouse, digitizing tablet, touch sensitive screen, and keyboard arrows for X/Y input devices. It supports ten different industrial and consumer video disc players. The software can be used with single or dual video monitor setups.

The specific information as to the exact system configuration is contained in a text file that can easily be modified. This flexibility is extremely useful, since it means different versions of the application program are not required for different system configurations. The actual setup is done by the user, not the programmer.

### 3.4 Program Development

The *Visage* system is extremely flexible in terms of program development. The software is essentially a set of machine language subroutines that can be called from a high level program. Subroutines are independent of the language used to do the program development. A developer can write his code in any of the six supported high level languages: BASIC, Compiled BASIC, PASCAL, C, dBASE II, and MACRO Assembler. Program parameters and variables can easily be passed to the *Visage* subroutines. The subroutines are grouped into video disc commands, graphics commands, text commands, and X/Y input device commands.

Since interactive video disc technology is still in its infancy, the growth and availablity of programs on the market will be directly related to the ease of producing such software. The success or failure of the technology will depend on the both the quantity and quality of program material produced.

# Chapter 4

# Development of the Application Progam

This chapter will discuss some of the factors considered and the research involved prior to the development of the final application program.

## 4.1 Pre-programming Research

A significant amount of research was conducted before any actual computer programming was begun.

### 4.1.1 Video Disc Selection

An integral part of the development of an interactive video disc program is the actual production and mastering of the video disc itself. This aspect can often occupy a significant portion of the project's resources. The money and time spent on the video production can be staggering. Careful attention must be paid at this stage, as a mistake will be irreversible once the video disc is mastered.

Although most future interactive video disc projects will have to go through this phase, this thesis will not do so. The video production aspect is beyond the scope of this project. The intent of this thesis was to demonstrate how a computer and a video disc player can be linked to provide a valuable training and educational tool. As a result, an existing video disc was chosen to be used as an example. This allowed the project to focus on the computer aspects of interactive video disc programs.

The disc chosen for the project was the *BIO SCI* video disc produced by

*Videodiscovery, Inc.* of Seattle, Washington. The disc contains close to 20,000 frames of video of biological sciences material. The disc was chosen because, of the number available, it seemed best suited for the development of an educational program. The images contained on the *Bio-Sci* disc can easily be related to topics discussed in an number of biology and biology related college courses.

### 4.1.2 Selection of Program Material

Once the disc was chosen, Professor Sheldon Penman of the M.I.T. Biology Department assisted in determining what portions of the disc were relevant to specific biology courses taught at M.I.T. A significant portion of the video disc contains still frames of a wide selection of plants and animals categorized by genus and species. This type of material is better suited to database type programs. Other parts of the disc, however, demonstrated biological concepts and processes through the use of diagrams, photographs, and movies.

The large amount and varied scope of the information contained on the disc made it impossible to create a program that utilized the entire disc. Professor Richard Hynes, also of the M.I.T. Biology Department, teaches a course in *developmental biology*. Because of his willingness to assist with some of the biological concepts, a section of the disc devoted to developmental biology was chosen for this project.

### 4.1.3 Background Research

The video disc itself contains no information about the images, save a title. The images, series of images, and movies have no narration or audio associated with them. In order to create an interactive program involving this part of the video disc, some background information about the topic had to be researched. Professor Hynes offered to give a lecture to his biology class using the video disc as a visual aid, much the way a lecturer would use slides or filmstrips. Professor Hynes devoted about half an hour to discussing the part of the disc devoted to developmental biology. He discussed in detail the images, often going through them more than once. In order to capture this information, an audio recording of the lecture was made. This tape was later transcribed to paper. The specific video disc locations were correlated and recorded on the paper.

Additional information about the subject was provided by two standard textbooks used extensively in a number of biology courses. *Molecular Biology of the Cell* by Bruce Alberts and *Developmental Biology* by Leon Browder were both referenced. Rhonda Wilson, an undergraduate in the biology department and a student in Professor Hynes' class, assisted in the compilation of relevant information. She offered a unique perspective, as she had just learned the material that term. She had knowledge of the topic, yet was not as well versed as a professor might be. She could point out specific areas that needed to be emphasized more. A professor's ubiquitous knowledge of the subject might prevent him from catching these points.

**4.2 Program Structure Considerations**

Armed with a video disc and the associated data, the next step was to develop a program structure that could convey this information to the user of the video disc system.

In order to demonstrate the capabilities of the interactive video disc system, it was decided to create a program that could guide the user through the material in much the same way a lecturer would. It was hoped that the program could be made flexible and interactive enough to remove some of the rigid structure often associated with lectures. The program was, by no means, intended to replace lecturers or professors. Their vast knowledge could not possibly be transferred to the system. Instead, the program was developed to mimic a small subset of their abilities.

### 4.2.1 Flexibilty

In developing this software, as with most pieces of software, an attempt was made to make the program flexible and easily modifiable. It was designed as a *shell* that was not specific to the video disc or subject matter chosen. Almost any disc and material can be incorporated into an interactive program using this piece of software. Of course, there is a trade-off for this flexibility. In displaying the images and graphics, a set of predefined rules must be followed. These rules allow a significant variety of display styles but are not infinite. Undoubtedly there are variations that are not possible with this specific piece of software. The actual subject material, information about graphics, video, and text, is stored in data files external to the software. These files would be changed for different video disc programs.

### 4.2.2 User Interaction

An important part of a video disc program is its ability to provide significant interaction with the user. In designing this piece of software, an attempt was made to create a program that avoided much of the stigma associated with "computer aided instruction". In this program, it was decided to remove the computer keyboard from the user. All interaction with the program is accomplished via the X/Y positional input device. In this case, this means the mouse and its buttons. The user selects menu items and icons on the screen entirely with the mouse.

#### 4.2.3 "Book" Like Format

In order to make the program as *user friendly* as possible, it was decided to organize the program into a format similar to a book. The familiarity most people have with books made it an ideal structure to mimic. The flexibilty associated with studying from a book is missing from most *computer aided instruction*. In this program, different topics were organized into separate *chapters*. Instead of a main menu, the user is presented with a *Table of Contents* from which he can choose a chapter to view.

A book can be skimmed through rapidly. If the current chapter is no longer of interest, one can easily move to another. Similarly, a user of this interactive video program can easily leave one chapter, enter another, or quit. Additionally, the user has a number of options open to him at all times. One can recall the table of contents, pause the program, call up a *help* display, or exit the program completely. This freedom keeps users from getting stuck in the program, and, thus, avoids becoming boring and tedious. Within each chapter of the program it is possible to obtain more detailed information about a topic or to continue on without that information.

# Chapter 5

# Program Analysis

This chapter will analyze and describe in some detail the main routines used in the application program. The complete listings of the main and support programs are included in Appendix A. The data files are included in Appendix B.

When producing code for an interactive video application, one is in a sense writing a script for the video disc. As the disc plays, certain operations occur, and certain graphics appear when specified frames are reached on the disc or when specified external conditions, such as user input, occur.

The program was written in *IBM Advanced BASIC.* As mentioned earlier, *Visage* supports a number of languages. Of these languages, BASIC has no particular advantage. It was chosen simply because of the author's familiarity with it as well as its interpretive structure. Both these assets were vital in aiding in debugging. Although some knowledge of BASIC and *IBM's* graphics commands is necessary to understand the routines in detail, a general overview can be understood without this knowledge.

## 5.1 Visage Prefix

The **prefix** section (see program lines 100 - 820 in Appendix A) is supplied by *Visage* and must be included at the beginning of any application program written in BASIC. The **prefix** section links BASIC to the *Visage* software. The beginning of the **prefix** program searches in memory for the *Visage* machine language program and returns the entry point to the **prefix** program. The BASIC interpreter uses this address in subsequent calls to the *Visage* software.

All commands and command arguments passed to the *Visage* program must be in variable form; literal names and values are not permitted. The rest of the **prefix** program assigns *Visage* commands to variables.

### 5.2 Initialization

The **initialization** section (see program lines 1000 - 1680 in Appendix A) sets the dimension of all arrays. Most of the arrays used in this program serve as graphics storage. Through the use of BASIC's GET and PUT commands, many of the program's graphical elements can be manipulated. Such things as menu elements and messages are stored in arrays.

The *IBM* (or CGA) graphics are initialized here. All *IBM* graphics in this program utilize the medium resolution mode. This mode allows a text width of 40 characters as well as color graphics. Most of the graphics that are stored in arrays are created here through the use of the GET statement.

A number of calls to the *Visage* software initialize its state. This initialization includes assigning the video disc player, turning the video on, opening the X/Y input device, and setting *Visage* graphics planes. All *Visage* graphics are loaded into memory from files on the hard disk. In this program this requires about 10 seconds. As a result, a message to "Please Standby" is printed on the screen during the load.

The *initialization* section calls the subroutine that displays the open titles, credits, and instructions.

At the end of the **initialization** section, the program jumps to the routine that displays the table of contents and loops until the user selects a chapter or one of the menu selections.

### 5.3 Opening Titles, Credits, and Instructions

This routine (see program lines 5000 - 5500 in Appendix A) is called only once at the beginning of the program and its purpose is to display the course title, acknowledge some of the people involved in the development of the software, and give a brief description of the program including instructions.

The routine is very flexible and could actually display any text here. The particular text to be displayed is stored in a data file called **titles.mok**. The text is displayed on an area of the screen that allows 10 lines of 18 characters each. The program centers the text in that area.

The text is combined with a *Visage* graphic screen. The graphics make it appear as if the text is being projected onto a movie sceen by a slide projector. The text is actually overlaid onto different video frames from the video disc.

The user "flips" through the text slides by pushing the button on the mouse. He is kept aware of his progress by the display of the number of each slide as well as the total number of slides. When all the slides have been displayed, the routine returns to the **initialization** section.

**5.4 Table of Contents**

The function of the **table of contents** routine (see program lines 6000 - 6470 in Appendix A) is to display the table of contents and wait for the user to select a chapter. A chapter is selected by moving the cursor over the text of the chapter title and pushing the mouse button. The routine can also be exited if a selection from the main menu is made. The **table of contents** section returns the number of the chapter selected.

In order to make the text on the screen look like and read as easily as a page from a book, this routine, as well as others, uses a *Visage* graphic plane that resembles a piece of paper that covers the right half of the screen. This leaves the left half available for the display of messages and prompts. Since the text of the table of contents and the chapters must fit onto this "paper" graphic, each text "page" is limited to a window of 18 lines by 22 characters. The table of contents, consisting of in this case six chapters, is displayed on the "paper" graphics. The main menu is also displayed at the bottom of the screen.

The routine then calls the **update cursor** subroutine. This subroutine returns the location of the X/Y device (i.e. the mouse), the value of the mouse button (i.e. pushed or not pushed), and whether any of the main menu selections were selected. It also moves the cursor to the location of the X/Y device. If a main menu selection was made, then the **table of contents** routine deals with that selection. If not, the position of the mouse is checked against the location of the text of each chapter title on the table of contents. If the mouse location matches any of the titles, that title is highlighted by enclosing it in a box. This gives a visual feedback to the user telling him that he can now select that chapter by pressing the mouse button. If the button is pushed while the cursor is located on one of the chapter titles, the **table of contents** exits, returning the number of the chapter selected. If the button was not pushed, the routine continues looping until a chapter or a main menu item is selected.

## 5.5 Update Cursor

The **update cursor** subroutine (see program lines 3000 - 3220 in Appendix A) serves a number of functions. First it finds the current mouse location. It then checks this against the location of the main menu items on the screen. If the location matches any of these items, the item's color is inverted to signal the user that he can select that item now. The routine then moves the cursor to the current X/Y location of the mouse. The routine limits the position of the cursor to remain on visible portion of the screen. Finally, the **update cursor** routine checks if the mouse button was pushed. This routine, itself, does not use this information, but merely returns it. It is the responsibility of the calling section to deal with dispatching any selections.

Notice that there is no loop in this subroutine. When the subroutine is called it makes one pass. Therefore, in order to simulate fluid cursor movement in real time, the subroutine must be called often. If not, the cursor movement will become jerky and may frustrate the user.

## 5.6 Chapter Display

In a sense this is the meat of the program and most of the program time will be spent in this section. The **chapter display** routine (see program lines 10000 - 11460 in Appendix A) is responsible for presenting the actual video disc program. Figure 5-1 shows the flowchart for the **chapter display** subroutine.

This subroutine is called each time the user selects a chapter from the table of contents. The routine is not specific to any one chapter. It relies upon data stored in files on disc. The text, graphics and video disc data for each chapter are stored in separate files. The files must be named **CHPTRn.MOK**, where **n** is the number of the chapter being displayed.

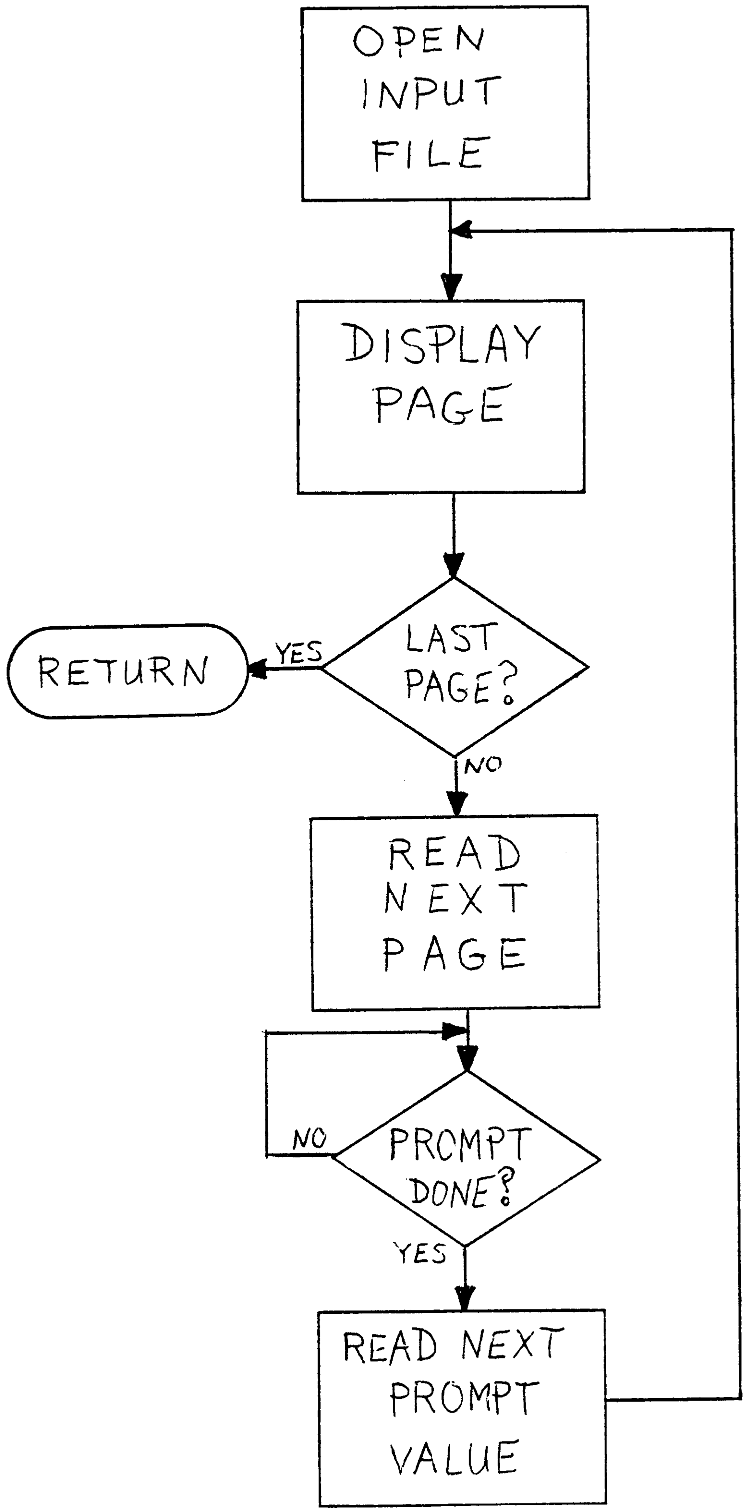
### 5.6.1 Chapter File Specifications

Each chapter, or file, contains a certain number of *pages*. These pages tell the **chapter display** subroutine what text, graphics, and video to display, as well as information about sequencing the *pages*.

Each *page* can contain a predefined *Visage* graphics canvas, including a blank transparent canvas. The "paper" canvas mentioned in the **table of contents** section is an example of a canvas. Additional canvases, containing such items as arrows, were created for some of the chapters.

The **background** can be specified as a solid dark blue or a frame number representing a video image on the disc. Choosing a dark blue background covers over any video image, whereas choosing a frame number clears the background and displays that video frame. If a movie is to be played, the beginning frame is specified here.

**Figure 5-1:** Flowchart for **Chapter Display** Subroutine



**Text** generated by the *IBM* graphics can be overlaid on both the **background** and *Visage* graphics. To display **text**, the number of lines, the text itself, and the X/Y location on the screen must be specified. The most common use of this is to display explanations over the "paper" graphics canvas. This mimics a printed piece of paper.

If a **movie** is to be displayed, the ending frame number must be specified. When a non-zero value is given, the video disc will be played from the video frame number given in the **background** section to the frame number in the **movie** section.

The way pages are sequenced is determined by the **prompt** section. The most frequently used **prompt** option is one that requires user input to advance the page. When all the graphics, text, and video have been displayed, or in the case of a movie, when it has finished playing, the user receives a message to "push the mouse button to continue". When the button is pushed the next *page* is displayed. A second **prompt** option is a timed sequence. In this case the number of seconds for which the *page* is to be displayed is specified. One use of this option is to allow a number of pages to be sequenced together to simulate animation. The third option allows a defined number of pages to be skipped. When this option is specified, the words "YES" and "NO" appear on the screen. Like any other selectable items in this program, the user chooses yes or no with the mouse. If "NO" is selected, the *page* specified in the **prompt** option will be the next *page* displayed. If "YES" is selected, the program continues on sequentially. The main use of the option is to allow the user to view more detailed information about a topic within a chapter. Text asking the user if he wants more information can be displayed. If he answers yes, the program will continue and display *pages* containing this information. If he answers no, these pages will be skipped.

Figure 5-2 lists the **chapter file** options as well as legal values for each option.

| **Option** | **Value** | **Meaning** |
|---|---|---|
| Graphics: | 0 | No graphics (i.e. transparent) |
| | 1 | "Paper" canvas |
| | 2,3,...,n | Predefined graphics canvases |
| Background: | 0 | Dark blue, no video |
| | pos. int., **n** | Video frame number **n** |
| Number of Text Lines: | 0 | No text, skip to **movie** option |
| | $0<n<26$ | Number of text lines to display |
| Text: | *text* | Actual text to be displayed |
| Text location | **x,y** | Location of upper left corner of text on screen. Limits: $0<x<41$, $0<y<26$ |
| Movie: | 0 | No movie |
| | pos. int., **n** | Play disc to frame number **n** starting from frame number specified in **background** field |
| Prompt: | 0 | Wait for user to push button |
| | pos. int., **p** | Wait **p** seconds, then continue |
| | neg. int., **n** | Skip to *page* number **n** if user answers "NO". Continue with next *page* if "YES" |

**Figure 5-2:** Chapter File Options

## 5.7 Help

The program has a **help** display (see program lines 9000 - 9440 in Appendix A) that is available from the main menu. This display describes the main menu options available and how to select them. If the user calls the **help** display when the table of contents is being displayed, the **help** subroutine will give information on how to select a chapter. Alternately, if he is viewing a chapter and **help** is selected, details on how to answer questions that come up during the chapter will be provided.

This **help** is entirely directed at aiding the user in the *operation* of the program. No attempt has been made to offer help on the particular subject matter being presented in the program.

### 5.8 Pause

The **pause** option (see program lines 7000 - 7170 in Appendix A) is available from the main menu at all times. When called, it freezes any movement on the video disc, displays a message indicating that the program is paused, and waits for the user to push the mouse button. When the button is pushed, the program continues.

This **pause** option is particularly useful for freezing a movie or a sequence of images for closer inspection.

### 5.9 Quit

As the name implies, the **quit** section exits the program. This section basically does some needed housecleaning before returning to BASIC's interpretive level. The *IBM* screen is returned to text mode, the *Visage* graphics are cleared, and the video disc player is cleared.

The option to quit is available to the user at any time during the program.

# Chapter 6

## Conclusion and the Future

This research demonstrates that fairly sophisticated interactive video disc programs can be developed on equipment that is both inexpensive and readily available. The combination of the *IBM XT* personal computer and the *Visage* system prove to be quite capable of producing professional results. Until recently, video disc programs had to be developed and run on large expensive mainframe computers with interface hardware that often had to be specially designed and built. The work in this thesis clearly demonstrates the advances in technology that have been made in recent years. Experienced computer programmers, of which the number is growing rapidly, can write interactive video software with little or no experience in that area. The software developed in this thesis is just a small example of what can be accomplished with larger, more in-depth projects.

Unfortunately, extensive testing of the software developed in this thesis was not possible. Ideally, a number of students and professors should have been allowed to "play" with the the final interactive video program. Their criticisms and suggestions could have been used to "fine tune" the program. As it stands, however, only a handful of people have actually used the program. The response has been very positive.

The software itself has many possible expansions. An obvious growth option is the addition of more chapters. Although this project restricted itself to a small segment in the field of biology, it would be quite possible to expand the database to cover the entire field of biology. The concept of an *interactive video textbook* is not unfathomable. The program could be modified to include an index in which the user could cross reference words to "page numbers" in the program. If a user is interested in "enzymes" he could use the index to call up the section(s) devoted to this topic. A glossary could also be included. If, while viewing a chapter, the user is unfamiliar with a term, he could instantly call up a glossary and read the definition.

The actual amount of user interaction incorporated into this program is only a small indication of the powers of a video disc system. Artificial intelligence can be built in, whereby the actual pace and detail of the program can be based on the user's response to questions. Quizzes and checkpoints can be added to evaluate a user's performance. Ultimately, an entire biology course could be created.

As mentioned earlier, the field of interactive video is still in its infancy. Although the video disc player itself is fighting to stay alive in the consumer market, it is unlikely it will be abandoned altogether. There are no current technologies that can mimic the capabilities of a video disc. A number of companies, both small and large, are rushing to introduce video systems for personal computers. Digital Equipment Corporation (DEC) already has a system, *IVIS*, which is currently being field tested in a number of locations. IBM claims to be coming out with a system, similar to Visage's, that will run on their personal computers. Clearly, there is excitement and enthusiasm surrounding this field. The future is unpredictable. Although the laser video disc may soon be a thing of the past as a home entertainment device, it is entirely likely that it may be showing up in the workplace, the school, and eventually in the home as an interactive video disc learning system.

# Appendix A
# Program Listings

This appendix contains the listings of the main and support programs used in the video disc program. All listings are in *IBM* Advanced BASIC.

## A.1 Main Application Program Listing

This section contains a listing of the Main Application program. See Chapter 5 for a discussion of the program itself.

```
1 ' *********************************************************
2 ' *                          An                           *
3 ' *      INTERACTIVE VIDEO DISC APPLICATION PROGRAM      *
4 ' *                          by                           *
5 ' *                Michael D. O'Keefe  '85                *
6 ' *  Developed as partial fulfillment of the requirements *
7 ' *                  for the degree of                    *
8 ' *                 Bachelor of Science                   *
9 ' *                       at the                          *
10 '*         Massachusetts Institute of Technology         *
11 '*                      June 1985                        *
12 '*********************************************************
13 '
14 '
100 '=========> Visage Prefix begins here <=========
110 '===> (This part copyright (c) 1984 Visage) <===
120 DEF SEG=&H3F94
130 BLOAD "blink1"              ' Find "visage"
140 BL1%=0: SEG%=0: BL2%=0      ' program in
150 CALL BL1%(SEG%,BL2%)        '  memory.
160 IF SEG%=0 THEN END
170 DEF SEG=SEG%: VSG = BL2%
180 AUD1ON$ = "aud1on" : AUD1OFF$="aud1off"
190 AUD2ON$   = "aud2on"
200 AUD2OFF$ ="aud2off": INDEXON$="indexon"
210 INDEXOFF$ = "indexoff"
220 CLRVP$ = "clear"    : SEARCHX$="searchx"
230 SEARCH$   = "search"
240 FWD$    = "fwd"     : STEPFWD$="stepfwd"
250 SLOWFWD$  = "slowfwd"
260 FASTFWD$="fastfwd" : SCANFWD$="scanfwd"
270 STILL$    = "still"
280 STEPREV$="steprev" : SLOWREV$="slowrev"
290 FASTREV$  = "fastrev"
300 SCANREV$ ="scanrev"
310 PASTE$ = "@$paste"   : VP$ = "@$vp"
320 OPENVP$="@$openvp":CLOSEVP$ ="@$closevp"
330 COMPRESS$ = "@$compress"
340 INIT$="@$init":BKCOLOR$="@$bkcolor"
350 EXVON$ = "@$exvon"
360 EXVOFF$ = "@$exvoff": CLCVS$ = "@$clscrn"
370 LDCVS$ = "@$ldcvs"
380 CLSP$ = "@$clsp" : MOVSP$ = "@$movsp"
390 DELSP$ = "@$delsp"
400 READIMG$ ="@$readimg": DELIMG$ = "@$delimg"
410 DISPIMG$ = "@$dispcvs"
420 CLSCRN$ ="@$clscrn" : PLAYTO$ = "@$playto"
430 ARRIVED$ = "@$arrived"
```

```
440 TEXT$ = "@$text" :BLANKOFF$ ="@$blankoff"
450 BLANKON$ = "@$blankon"
460 EXPCVS$ ="@$expcvs" : CMPCVS$ = "@$cmpcvs"
470 COPY$ = "@$copy"
480 EXPAND$ = "@$expand": EXIT$ = "@$exit"
490 GETFRAME$ = "@$getframe"
500 PLAYTO$ = "@$playto": ARRIVED$ = "@$arrived"
510 READFONT$ = "@$readfont"
520 DISPCVS$ = "@$dispcvs": TION$ = "@$tion"
530 TIOFF$ = "@$tioff"
540 CGAON$ = "@$cgaon": CGAOFF$ = "@$cgaoff"
550 TICGA$ = "@$ticga"
560 CGATI$ = "@$cgati": CGATRANS$ = "@$cgatrans"
570 CGAOPAQ$ = "@$cgaopaq"
580 CGACOLOR$ = "@$cgacolor": FSCROLL$ = "@$fscroll"
590 FINIT$ = "@$finit":  FCOLOR$ = "@$fcolor"
600 VPRINT$ = "@$vprint":
610 VPRINTN$ = "@$vprintn"
620 FHOME$ = "@$fhome": FUP$ = "@$fup"
630 FDOWN$ = "@$fdown": FLEFT$ = "@$fleft"
640 FRIGHT$ = "@$fright"
650 FPOS$ = "@$fpos": FERSLINE$ = "@$fersline"
660 FCLRSCRN$ = "@$fclrscrn"
670 OPENXY$ = "@$openxy"
680 GETXY$ = "@$getxy": SETXY$ = "@$setxy"
690 CLOSEXY$ = "@$closexy"
700 INITXY$ = "@$initxy": GETBUT$ = "@$getbut"
710 FERSSCRN$ = "@$fersscrn"
720 MONON$ = "@$monon": MONOFF$ = "@$monoff"
730 TRANS%    = 0: BLACK% = 1
740 MGREEN% = 2: LGREEN% = 3
750 DBLUE% = 4: LBLUE% = 5: DRED% = 6
760 CYAN%    = 7: MRED% = 8: LRED% = 9
770 DYELLOW% = 10:LYELLOW% = 11:DGREEN% = 12
780 MAGENTA% = 13:GRAY% = 14
790 GREY% = 14:WHITE% = 15:INVIS = -99
800 GFRAME$ = "16003":SFRAME$="00000"
810 KEYON$="@$keyon":KEYOFF$="@$keyoff"
820 GETRFRAME$="@$getrframe"
1000 '*********************************************************
1010 '*            Application program starts here            *
1020 '*   This part copyright (c) 1985, Michael D. O'Keefe    *
1030 '*********************************************************
1040 DIM BAR(323),INV10(85),INV9(78)
1050 DIM INV7(61),INV6(52),CHXY%(6)
1060 DIM PROMPT(105),YES(17),NO(17)
1070 DIM INVY(17),HELP(4051),TEXT$(20)
1080 CALL VSG(INIT$)          ' initialize visage software
1090 CLS: KEY OFF
1100 SCREEN 1,0:COLOR 1,0
1110 IBMGRN%=2:IBMBLK%=0                  '*
1120 LEFT%=1:PUSH%=0                      '*
1130 PRV=0:FWFLG%=0:CONST=5               '*
1140 FOR X=1 TO 6:READ CHXY%(X):NEXT X    '* initialize
1150 DATA 274,234,266,258,274,202:        '* variables
1160 MOUSE$="mouse":XYLUN%=2              '*
1170 MENU$="  Table of Contents  Pause  Help  Quit"
1180 VDP$="ldp1000":LUN%=1                '*
1190 CALL VSG(OPENVP$,LUN%,VDP$)      :': open video disc
1200 CALL VSG(OPENXY$,XYLUN%,MOUSE$) :': open x/y device
1210 CALL VSG(VP$,CLRVP$)             :': initialize disc
1220 CALL VSG(VP$,INDEXOFF$)          :': turn index off
1230 CALL VSG(BKCOLOR$,BLACK%)
1240 CALL VSG(EXVON$)         :': turn video on
1250 CALL VSG(CGATI$)         :': CGA graphics priority
1260 CALL VSG(CGATRANS$)      :': make CGA transparent
1270 LINE (8,191)-(311,199),2,BF          '*
1280 GET (8,191)-(311,199),BAR            '*
1290 LINE (0,191)-(100,199),1,BF          '*
1300 GET (0,191)-(23,198),INVY            '*
1310 GET (0,191)-(79,199),INV10           '*
1320 GET (0,191)-(71,199),INV9            '*
1330 GET (0,191)-(55,199),INV7            '*
1340 GET (0,191)-(47,199),INV6            '*  INITIALIZE
1350 LOCATE 2,2,0:PRINT "LEFT button"     '*     MENU
1360 LOCATE 3,2,0:PRINT "to continue"     '*     AND
1370 PUT (0,5),INV10,XOR                  '*   GRAPHICS
1380 PUT (80,5),INV10,XOR                 '*  VARIABLES
1390 PUT (0,14),INV10,XOR                 '*
1400 PUT (80,14),INV10,XOR                '*
```

```
1410 PUT (0,23),INV10,XOR                      '*
1420 PUT (80,23),INV10,XOR                     '*
1430 GET (7,7)-(96,24),PROMPT                  '*
1440 LOCATE 6,2,0:PRINT "YES NO"               '*
1450 PUT (0,39),BAR,OR                         '*
1460 GET (6,39)-(33,47),YES                    '*
1470 GET (38,39)-(65,47),NO                    '*
1480 CFILE$="pix.img":CURSOR$="*"           '*
1490 CLS:LOCATE 12,15,0                     '*
1500 PRINT "Please STANDBY"                 '*
1510 LOCATE 13,14,0                         '*    Load
1520 PRINT "Loading graphics"               '*  graphics
1530 LINE (100,84)-(236,107),2,B            '*    files
1540 CALL VSG(CGAON$)                       '*
1550 CALL VSG(READIMG$,CFILE$,CURSOR$)      '*
1560 EMPTY$="empty.cvs"                     '*
1570 GOSUB 5000       ' Display opening titles and instr.
1580 CURSOR$="cursor.spr"
1590 PAPER$="paper.cvs"
1600 GOSUB 4000       ' Display MENU
1610 X%=135:Y%=23     ' Initial cursor position
1620 CALL VSG(SETXY$,X%,Y%)  'Set mouse to x,y
1630 GOTO 6000        ' Display Table of Contents
1640 GOSUB 4000       ' Display MENU
1650 IF CURCHP%>0 THEN GOSUB 10000  ' Goto CHAPTER
1660 IF CURCHP%=0 THEN 1610 'No selection, loop again.
1670 CLOSE #3:DEF SEG:POKE &H4E,3:DEF SEG=SEG%
1680 GOTO 1600        ' Loop again.
2000 ' =====> QUIT <======
2010 CALL VSG(BLANKON$)            '*
2020 CALL VSG(EXVOFF$)             '*
2030 WIDTH 80 :CLS                 '* Restore
2040 SCREEN 0,0,0                  '*  screen
2050 COLOR 7,0:KEY ON              '*  before
2060 CALL VSG(CGAON$)              '* exiting
2070 CALL VSG(CGATI$)              '*
2080 CALL VSG(CLSCRN$,TRANS%)      '*
2090 CALL VSG(CGAOPAQ$)            '*
2100 CALL VSG(VP$,STILL$)          '*
2110 END
3000 ' ========> Sub to update cursor <=========
3001 ' ===>check menu, and invert selection <===
3010 CALL VSG(GETXY$,X%,Y%)   ' Get mouse location
3020 IF X%>285 THEN X%=285:CALL VSG(SETXY$,X%,Y%) 'Limit
3030 IF Y%>185 THEN Y%=185:CALL VSG(SETXY$,X%,Y%) 'cursor
3040 IF Y%>7 THEN CH=0 : GOTO 3090      '
3050 IF X%<138 THEN CH=1 : GOTO 3090    '  Check
3060 IF X%<190 THEN CH=2 : GOTO 3090    '   MENU
3070 IF X%<232 THEN CH=3 : GOTO 3090    ' selection
3080 IF X%<275 THEN CH=4 : GOTO 3090    '
3090 IF PRV=CH THEN 3200
3100 ON PRV GOTO 3140,3160,3170,3180
3110 PRV=CH
3120 ON CH GOTO 3140,3160,3170,3180
3130 GOTO 3190
3140 PUT (8,191),INV10,XOR                 '
3150 PUT (88,191),INV9,XOR: GOTO 3190     '  Invert
3160 PUT (160,191),INV7,XOR: GOTO 3190    ' selection
3170 PUT (216,191),INV6,XOR: GOTO 3190    '
3180 PUT (264,191),INV6,XOR: GOTO 3190    '
3190 IF PRV<>CH THEN 3110
3200 CALL VSG(MOVSP$,CURSOR$,X%,Y%) 'Move coursor to x,y
3210 CALL VSG(GETBUT$,LEFT%,PUSH%)  'Button pushed?
3220 RETURN
4000 ' =====> Draw menu sub. <=====
4010 CLS
4020 LOCATE 25,1 : PRINT MENU$;
4030 PUT (8,191),BAR,OR
4040 PRV=0
4050 RETURN
5000 ' =====> Sub to display opening titles <=====
5001 ' =======> credits, and instructions <=======
5010 OPEN "i",#3,"titles.mok"
5020 INPUT #3,MAXSLD
5030 CALL VSG(CGACOLOR$,IBMGRN%)
5040 CLS:CALL VSG(CGAON$)
5050 INTRO$="intro.cvs"
5060 CALL VSG(DISPCVS$,INTRO$)
5070 CALL VSG(BKCOLOR$,BLACK%)
5080 LOCATE 12,15
```

```
5090 PRINT "Hit any KEY";
5100 LOCATE 13,17
5110 PRINT "to Begin";
5120 LINE (103,84)-(207,107),2,B
5130 A$="":A$=INKEY$:IF A$="" THEN 5130
5140 IF A$="x" THEN 5420
5150 FOR Y=101 TO 200
5160 X=CINT(Y*1.6)
5170 LINE (319-X,199-Y)-(X-1,Y-1),1,BF
5180 NEXT Y
5190 CALL VSG(CGAOFF$)
5200 CALL VSG(TICGA$)
5210 SCREEN 0,1,0:WIDTH 40
5220 COLOR 14,2,2:CLS
5230 CALL VSG(CGAON$)
5240 FOR X=1 TO MAXSLD
5250 CALL VSG(KEYOFF$)
5260 CLS:INPUT #3,SLIDE1,SLIDE2
5270 FRAME$=RIGHT$(STR$(100000!+SLIDE1),5)
5280 CALL VSG(VP$,SEARCH$,FRAME$)
5290 YTOP%=INT((10-SLIDE2)/2)+4
5300 FOR YLINE%=YTOP% TO YTOP%+SLIDE2-1
5310 LINE INPUT #3,TXT$
5320 A%=22+INT((18-LEN(TXT$))/2)
5330 LOCATE YLINE%,A%,0:PRINT TXT$;
5340 NEXT YLINE%
5350 LOCATE 14,24,0
5360 PRINT "Slide"X"of"MAXSLD
5370 CALL VSG(KEYON$)
5380 CALL VSG(GETBUT$,LEFT%,PUSH%)
5390 A$=INKEY$:IF A$="x" THEN 5420
5400 IF PUSH%=0 THEN 5380
5410 NEXT X
5420 CLOSE #3
5430 CALL VSG(KEYOFF$)
5440 CALL VSG(CGACOLOR$,IBMBLK%)
5450 CALL VSG(CLSCRN$,DBLUE%)
5460 CALL VSG(EXVOFF$)
5470 SCREEN 1,0
5480 COLOR 0,0:CLS
5490 CALL VSG(CGATI$)
5500 RETURN
6000 ' =====> Display Table of Contents  <=====
6010 CALL VSG(EXVOFF$)              ' Turn video off
6020 CALL VSG(DISPCVS$,PAPER$)      ' Put paper up
6030 CALL VSG(BKCOLOR$,DBLUE%)      ' Blue back
6040 CALL VSG(BLANKOFF$)            ' Graphics on
6050 CALL VSG(VP$,STILL$)           ' Stop player
6060 PCHP%=0:DEF SEG:POKE &H4E,2 'Red text
6070 CURCHP%=0
6080 LOCATE 5,21,0                  '
6090 PRINT "Table of Contents"      '
6100 LOCATE 8,19                    '
6110 PRINT "1. Fertilization"       '
6120 LOCATE 10,19                   ' Display
6130 PRINT "2. Cleavage"            ' chapter
6140 LOCATE 12,19                   ' titles.
6150 PRINT "3. Gastrulation"        '
6160 LOCATE 14,19                   '
6170 PRINT "4. Neurulation"         '
6180 LOCATE 16,19                   '
6190 PRINT "5. Morphogenesis"       '
6200 LOCATE 18,19                   '
6210 PRINT "Summary"                '
6220 LINE (156,29)-(300,43),2,B '
6230 POKE &H4E,3:DEF SEG=SEG%
6240 GOSUB 3000
6250 IF CH=0 THEN 6290
6260 IF PUSH%=0 THEN 6240
6270 IF CH=1 THEN 6010 ELSE GOSUB 8000
6280 GOSUB 8000
6290 IF X%<127 OR X%>245 THEN CHP%=0:GOTO 6370
6300 IF Y%<44 OR Y%>142 THEN CHP%=0:GOTO 6370
6310 IF Y%<62 THEN CHP%=6:GOTO 6370   '*
6320 IF Y%<78 THEN CHP%=5:GOTO 6370   '* Check if
6330 IF Y%<94 THEN CHP%=4:GOTO 6370   '* cursor
6340 IF Y%<110 THEN CHP%=3:GOTO 6370 '* is on
6350 IF Y%<126 THEN CHP%=2:GOTO 6370 '* a chapter
6360 IF Y%<142 THEN CHP%=1            '*
6370 IF PCHP%=CHP% THEN 6450
```

```
6380 T%=PCHP%*16+38:S%=T%+10
6390 IF PCHP%=0 THEN 6410
6400 LINE (142,T%)-(CHXY%(PCHP%),S%),0,B 'Erase old box
6410 T%=CHP%*16+38:S%=T%+10
6420 IF CHP%=0 THEN 6440
6430 LINE (142,T%)-(CHXY%(CHP%),S%),1,B  'Make new box
6440 PCHP%=CHP%
6450 IF PUSH%=0 THEN 6240    ' No button pushed
6460 CURCHP%=CHP%  :': Chapter
6470 GOTO 1640     :': selected.
7000 ' =====> Sub. to pause <=====
7010 IF FWFLG%=1 THEN CALL VSG(VP$,STILL$)  'Stop disc
7020 DEF SEG:T%=PEEK(&H4E):POKE &H4E,3      'Yellow text
7030 LOCATE 3,2,0:PRINT "PROGRAM PAUSED":':
7040 PRINT " Hit LEFT mouse"             :':
7050 PRINT "   button to"                :':
7060 PRINT "    continue"                :': Display
7070 POKE &H4E,T%:DEF SEG=SEG%           :':  PAUSE
7080 FOR I%=13 TO 40 STEP 9              :': message
7090 PUT (0,I%),INV10,XOR                :':
7100 PUT (80,I%),INV6,XOR                :':
7110 NEXT I%                             :':
7120 CALL VSG(GETBUT$,LEFT%,PUSH%) 'Button pushed?
7130 IF PUSH%=0 THEN 7120          'If not, try again.
7140 LINE (0,13)-(128,50),0,BF     'Clear PAUSE message
7150 IF FWFLG%=1 THEN CALL VSG(VP$,FWD$) 'Start disc
7160 PUSH%=0
7170 RETURN
8000 ' =====> Sub. to dispatch menu selection <=====
8010 PUSH%=0
8020 ON CH-1 GOTO 8030,8040,8050
8030 GOSUB 7000:RETURN  'Pause subroutine
8040 GOSUB 9000:RETURN  'Help subroutine
8050 GOTO 2000          'Quit
9000 ' =====> Help sub. <=====
9010 GET (0,0)-(319,199),HELP 'Save screen in HELP.
9020 DEF SEG: POKE &H4E,(1 XOR PEEK(&H4E))
9030 CLS:COLOR 1,0
9040 LOCATE 1,15
9050 PRINT "HELP Display"
9060 IF CURCHP%>0 THEN 9390 'Display chapter help.
9070 PRINT
9080 PRINT "To select any Chapter from the Table of"
9090 PRINT "Contents, use the mouse to move the"
9100 PRINT "green cursor over the TITLE. The title"
9110 PRINT "will be highlighted with a box around"
9120 PRINT "it. If you desire to view this chapter,"
9130 PRINT "push the LEFT mouse button."
9140 PRINT
9150 PRINT "Additionally, you may choose any of the"
9160 PRINT "MENU selections located at the bottom"
9170 PRINT "of the screen in the same manner. The"
9180 PRINT "bottom menu will always be available:"
9190 '   **** Table of Contents help text ****
9200 PRINT
9210 PRINT " TABLE OF CONTENTS: Exits chaper and"
9220 PRINT "       recalls Table of Contents."
9230 PRINT
9240 PRINT " PAUSE: Will pause the program.  This is";
9250 PRINT "       particularly useful during movies."
9260 PRINT " HELP: Displays this screen."
9270 PRINT
9280 PRINT " QUIT: Exits the program completely."
9290 POKE &H4E,(1 XOR PEEK(&H4E)):DEF SEG=SEG%
9300 LOCATE 25,1,0
9310 PRINT "    HIT LEFT MOUSE BUTTON TO RETURN";
9320 CALL VSG(GETBUT$,LEFT%,PUSH%) 'Get button
9330 IF PUSH%=0 THEN 9320  'No push, loop.
9340 PUSH%=0
9350 COLOR 0,0:CLS        :': Restore
9360 PUT (0,0),HELP,PSET :': display
9370 RETURN
9380 '   **** Chapter help text ****
9390 PRINT
9400 PRINT "To answer questions during a chapter,"
9410 PRINT "use the mouse to move the cursor over"
9420 PRINT "πYES' or πNO'. Then push the LEFT mouse"
9430 PRINT "button."
9440 GOTO 9140
10000 ' =====> Display Chapter Subroutine <=====
```

```
10010 FWFLG%=0:YESFLG%=0
10020 TI%=1:BKFR=0
10030 TXVI%=1:TXX%=19
10040 TXY%=4:TEXT$(1)=""
10050 MOVIE=0:PRWT%=1
10060 COLR%=DBLUE%:PRVV=-1
10070 ON CURCHP% GOTO 10080,10130,10180,10230,10280,10330
10080 LOCATE 1,24
10090 PRINT "1. Fertilization"
10100 PUT (176,0),INV9,XOR
10110 PUT (248,0),INV9,XOR
10120 GOTO 10360
10130 LOCATE 1,29
10140 PRINT "2. Cleavage"
10150 PUT (216,0),INV7,XOR
10160 PUT (272,0),INV6,XOR
10170 GOTO 10360
10180 LOCATE 1,25
10190 PRINT "3. Gastrulation"
10200 PUT (184,0),INV10,XOR
10210 PUT (264,0),INV7,XOR
10220 GOTO 10360
10230 LOCATE 1,26
10240 PRINT "4. Neurulation"
10250 PUT (192,0),INV10,XOR
10260 PUT (272,0),INV6,XOR
10270 GOTO 10360
10280 LOCATE 1,24
10290 PRINT "5. Morphogenesis"
10300 PUT (176,0),INV9,XOR
10310 PUT (248,0),INV9,XOR
10320 GOTO 10360
10330 LOCATE 1,33
10340 PRINT "Summary"
10350 PUT (248,0),INV9,XOR
10360 DEF SEG:POKE &H4E,2:DEF SEG=SEG%
10370 CALL VSG(KEYON$)
10380 CALL VSG(DISPCVS$,PAPER$)
10390 CALL VSG(BKCOLOR$,DBLUE%)
10400 CALL VSG(EXVON$)
10410 FILE$="chptr"+RIGHT$(STR$(CURCHP%),1)+".mok"
10420 OPEN "i",#3,FILE$
10430 INPUT #3,NPAGE%:PAGE%=-1
10440 FOR C%=1 TO NPAGE%+1
10450 PUSH%=0
10460 IF YESFLG%=1 AND JUMP%>C% THEN 10680
10470 LINE (0,9)-(319,189),0,BF  ' clear lines 2 - 24
10480 IF BKFR=0 OR BKFR=PRVV THEN 10570
10490 IF BKFR<>PRVV+1 THEN 10520
10500 CALL VSG(VP$,STEPFWD$)
10510 PRVV=PRVV+1:GOTO 10570
10520 IF BKFR<>PRVV-1 THEN 10550
10530 CALL VSG(VP$,STEPREV$)
10540 PRVV=PRVV-1:GOTO 10570
10550 CALL VSG(VP$,SEARCH$,BKFR$)
10560 PRVV=BKFR
10570 IF TI%=0 THEN CALL VSG(DISPCVS$,EMPTY$):GOTO 10610
10580 IF TI%=1 THEN CALL VSG(DISPCVS$,PAPER$):GOTO 10610
10590 CNVS$=RIGHT$(STR$(TI%+100),2) + ".cvs"
10600 CALL VSG(DISPCVS$,CNVS$)
10610 CALL VSG(BKCOLOR$,COLR%)
10620 IF TXVI%=0 THEN 10730
10630 FOR J%=1 TO TXVI%
10640 LOCATE TXY%,TXX%,0
10650 PRINT TEXT$(J%)
10660 TXY%=TXY%+1
10670 NEXT J%
10680 IF TI%<>1 THEN 10730
10690 PAGE%=PAGE%+1
10700 IF YESFLG%=1 AND JUMP%>C% THEN 10890
10710 LOCATE 22,27,0
10720 PRINT "-"PAGE%"-";
10730 IF YESFLG%=1 AND JUMP%>C% THEN 10890
10740 IF MOVIE=0 THEN 10890
10750 CALL VSG(GETRFRAME$,FM)
10760 IF FM<>BKFR THEN 10750
10770 FWFLG%=1
10780 MOVIE$=RIGHT$(STR$(MOVIE+100000!),5)
10790 CALL VSG(PLAYTO$,MOVIE$)
10800 PRVV=MOVIE
```

```
10810 CALL VSG(VP$,FWD$)
10820 CALL VSG(ARRIVED$,FLG%)
10830 IF FLG%=1 THEN FWFLG%=0:GOTO 10890
10840 GOSUB 3000
10850 IF PUSH%=0 OR CH=0 THEN 10820
10860 IF CH=1 THEN RETURN
10870 GOSUB 8000
10880 GOTO 10820
10890 IF C%>NPAGE% THEN 11010
10900 INPUT #3,TI%,BKFR,TXVI%
10910 BKFR$=RIGHT$(STR$(BKFR+100000!),5)
10920 COLR%=DBLUE%
10930 IF BKFR>0 THEN COLR%=TRANS%
10940 IF BKFR=-1 THEN COLR%=TRANS%
10950 IF TXVI%=0 THEN 11000
10960 FOR J%=1 TO TXVI%
10970 LINE INPUT #3,TEXT$(J%)
10980 NEXT J%
10990 INPUT #3,TXX%,TXY%
11000 INPUT #3,MOVIE
11010 IF YESFLG%=1 AND JUMP%>C% THEN 11440
11020 IF PRWT%<>0 THEN 11100
11030 PUT (0,170),PROMPT,PSET
11040 GOSUB 3000
11050 IF PUSH%=0 THEN 11040
11060 IF CH=0 THEN 11440
11070 IF CH=1 THEN RETURN
11080 GOSUB 8000
11090 GOTO 11040
11100 IF PRWT%<0 THEN 11190
11110 COUNT=0
11120 GOSUB 3000
11130 IF PUSH%=0 OR CH=0 THEN 11160
11140 IF CH=1 THEN RETURN
11150 GOSUB 8000
11160 COUNT=COUNT+1
11170 IF COUNT<(CONST*PRWT%) THEN 11120
11180 GOTO 11440
11190 PUT (8,150),YES,PSET
11200 PUT (40,150),NO,PSET
11210 PCHP%=0
11220 X%=80:Y%=45
11230 CALL VSG(SETXY$,X%,Y%)
11240 GOSUB 3000
11250 IF CH=0 THEN 11290
11260 IF PUSH%=0 THEN 11240
11270 IF CH=1 THEN RETURN
11280 GOSUB 8000
11290 IF Y%>50 OR Y%<33 OR X%>64 THEN ANS%=0:GOTO 11320
11300 ANS%=40
11310 IF X%<32 THEN ANS%=8
11320 IF PCHP%=ANS% THEN 11360
11330 IF PCHP%>0 THEN PUT (PCHP%+1,150),INVY,XOR
11340 IF ANS%>0 THEN PUT (ANS%+1,150),INVY,XOR
11350 PCHP%=ANS%
11360 IF PUSH%=0 OR ANS%=0 THEN 11240
11370 YESFLG%=0
11380 IF ANS%>8 THEN YESFLG%=1
11390 JUMP%=ABS(PRWT%)+1
11400 IF YESFLG%=0 THEN 11440
11410 LINE (0,9)-(319,189),0,BF
11420 CALL VSG(DISPCVS$,EMPTY$):CALL VSG(BKCOLOR$,DBLUE%)
11430 CALL VSG(BKCOLOR$,DBLUE%)
11440 IF C%<=NPAGE% THEN INPUT #3,PRWT%
11450 NEXT C%
11460 RETURN
```

## A.2 Chapter File Program

This is a listing of the program used to create each of the **chapter files.**

```
10 INPUT "Filename";N$
15 OPEN "o",#3,N$
20 INPUT "No. of pages";N%:PRINT #3,N%
30 FOR X%=1 TO N%
35 PRINT:PRINT "Page"X%
40 INPUT "ti%=";T%
50 INPUT "bkfr=";BK
55 INPUT "No. text lines";TX%
60 PRINT #3,T%,BK,TX%
70 IF TX%=0 THEN 87
80 FOR J%=1 TO TX%
82 PRINT "line"J%;:LINE INPUT T$
84 PRINT #3,T$:NEXT J%
85 INPUT "Text X loc.";TXX%:INPUT "Text Y loc.";TXY%
86 PRINT #3,TXX%,TXY%
87 INPUT "Movie";MOV:PRINT #3,MOV
90 INPUT "prwt%=";P%
100 PRINT #3,P%
105 NEXT X%
110 CLOSE #3
120 END
```

## A.3 Title File Program

This is a listing of the program that creates the file called **titles.mok**. This is the file that contains the opening titles, credits, and instructions.

```
10 OPEN "o",#1,"titles.mok"
20 INPUT "Number of slides";MAXSLD
30 PRINT #1,MAXSLD
40 FOR X=1 TO MAXSLD
50 PRINT "Frame number for slide #"X;:INPUT SLIDE1
60 PRINT "Number of text lines for slide #"X;:INPUT SLIDE2
70 PRINT #1,SLIDE1,SLIDE2
80 FOR Y=1 TO SLIDE2
90 PRINT "Line"Y;:LINE INPUT TEXT$
100 PRINT #1,TEXT$
110 NEXT Y:PRINT:NEXT X
120 CLOSE #1
140 END
```

# Appendix B

# Data File Listings

This appendix lists the data files used by the main application program. This includes the **chapter files** and the **titles file**. A description of the values contained in the **chapter files** can be found in Figure 5-2. The files are listed in multi-column format to conserve space. They are actually stored in single column format in the files.

## B.1 TITLES.MOK Listing

This is a listing of the data contained in the **titles.mok** file. It includes opening titles, credits, and instructions.

```
 14
 16764        3
Push the LEFT
MOUSE button to
change each slide
 14823        5
WELCOME
to an
Interactive
Videodisc Based
Learning System
 14888        5
This software
was developed
by
Michael O'Keefe
copyright (c) 1985
 14912        6
with assistance
from
Dr. Edwin Taylor
Prof. S. Penman
and
Prof. R. Hynes
 17056        5
This program is
intended to
demonstrate
basic principles
in the area of...
 13940        2
DEVELOPMENTAL
BIOLOGY
 14953        6
This will be
done through the
presentation of
VISUAL as well
as TEXTUAL
information
 16870        4
The program is
structured in a
way similar to
a book
 15838        6
Different topics
are divided into
chapters.  A
table of contents
is available to
list all chapters.
 16330        8
Like a book, you
can view the
chapters in any
order. You can
review a chapter,
pause, or quit
at any time
during the program
 16254        6
All interaction
with the program
is accomplished
through the use
of the MOUSE
and its buttons
 17000        8
A MENU will be
displayed at the
bottom of the
screen at all
times. You can
select any menu
item at any time
during the program
 16283        8
To select an item
or answer a
question, use the
mouse to move the
cursor over the
appropriate box
```

```
and push the LEFT
button on the mouse
 16259         8
This system is
intended to
provide a flexible
alternative to the
often rigid
structure of a
lecture.
So HAVE FUN!
```

## B.2 Chapter 1, Fertilization, Data File (CHPTR1.MOK)

```
 14
 1             64            13
This chapter describes
fertilization of the
egg and changes which
take place just after
fertilization but
prior to cleavage in
amphibians such as the
frog Rana pipiens.

The eggs are laid by
the female frog and
fertilized by the male
as they are laid.
 18            4
 0
 0
 0             65            0
 0
 0
 0             66            0
 0
 0
 0             67            0
 0
 0
 0             68            0
 0
 0
 1             69            14
The eggs swell up to
produce jelly coats
once they are laid.
The egg sits, pigmen-
ted animal pole up,
surrounded by the fer-
tilization membrane
and jelly coats.
Between the plasma
and fertilization
membranes of the egg
lies the perivitelline
space, in which the
egg is free to rotate.
 18            4
 0
 0
 0             70            0
 0
 0
 0             71            0
 0
 0
 0             72            0
 0
 0
 1             0             18
The sperm enters the
egg in its animal
hemisphere.  The fer-
tilized egg undergoes
cytoplasmic reorgani-
zation as a result of
sperm entry.  The
cortex of the egg
shifts, carrying with
it pigment granules
that previously marked
the animal hemisphere.
As a result of this
rearrangement the
egg changes from
radial to bilateral
symmetry, and the
grey crescent appears.
 18            4
 0
 0
 0             73            0
 0
 0
 3             74            0
 0
 0
 3             75            0
 0
 0
 1             0             16
The grey crescent
plays an important
role in later develop-
ment of the zygote.
Its appearance is an
external indication of
the shift to bilateral
symmetry.  Internal
changes have occurred
as well.  The one-
celled egg is now
ready to begin
cleavage.  Select
Chapter 2, Cleavage,
from the Table of
Contents to continue.
 18            4
 0
 0
```

## B.3 Chapter 2, Cleavage, Data File (CHPTR2.MOK)

```
 27
 1             0             16
In order for a one-
celled zygote to
become a multicellular
organism, a number of
mitotic divisions must
occur in rapid suc-
cession.  This series
of cell divisions is
called cleavage.  Dur-
ing cleavage the size
and shape of the em-
bryo stays the same,
while the cells, or
blastomeres, become
smaller at each di-
vision.
 18            4
 0
 0
 1             0             15
These early mitotic
divisions in amphi-
bians are synchronous
and extremely rapid,
with a cell cycle time
of about thirty
minutes.  Would you
like more detailed
information on these
rapid cell divisions?
Select 'YES' or 'NO'
with the cursor and
push the left mouse
button to enter your
choice.
 18            4
 0
 -4
 1             0             18
The rapid divisions
are possible because
supplies of RNA, pro-
tein, membrane mole-
cules, and other
materials accumulate
in the egg while it
matures in the mother,
and therefore do not
need to be made during
cleavage.  DNA, how-
ever, is made during
cleavage.  The needed
DNA is replicated very
rapidly by means of an
exceptionally large
number of replication
origins.
 18            4
 0
 0
 1             0             15
At the completion of
each cell division,
blastomeres are sep-
arated from one
another by the for-
mation of cleavage
furrows.

The 'movie' you are
about to see shows the
process of cleavage in
a frog embryo.  You
can select 'PAUSE'
with the cursor to
stop the action.
 18            4
 0
 0
 0             76            0
 2101
 0
 1             2101          7
Would you like more
detailed information
about the pattern of
cleavage in the frog
embryo?  Select 'YES'
or 'NO' with the
cursor.
 18            4
 0
 -15
 1             0             18
In frog embryos, the
first cleavage furrow
originates at the pig-
mented animal pole of
the egg and spreads to
the opposite pole, bi-
secting the grey
crescent.  The animal
hemisphere contains
the egg's nucleus and
most of its cytoplasm,
whereas the vegetal
hemisphere contains
mostly thick yolk.
Cleavage goes more
slowly in the yolky
vegetal portion than
near the animal pole.
 18            4
 0
 0
 0             76            0
 316
 0
 1             316           4
The second furrow
begins at the animal
pole as well, at right
angle to the first.
 18            4
 0
 0
 0             316           0
 755
 0
 1             755           16
The third cleavage
furrow, which divides
four cells into eight,
lies in the equatorial
plane of the egg,
slightly closer to the
animal than the veg-
etal pole.  As a re-
sult of this asymmetry
the cells in the an-
imal hemisphere are
smaller than those in
the vegetal hemi-
sphere, and they re-
main so throughout
cleavage.
 18            4
 0
```

```
 0
 0              755            0
 1275
 0
 1              1275           8
Cleavage continues
until about 10,000
cells have been
formed.  Note how
synchronous the
divisions are--each
cell divides at the
same time.
 18             4
 0
 0
 0              1280           0
 2101
 0
 1              0              5
Would you like to see
the complete process
of cleavage again?
Select 'YES' or 'NO'
with the cursor.
 18             4
 0
-17
 0              76             0
 2101
 0
 1              0              17
Small, fluid-filled
spaces appear between
blastomeres at early
stages of cleavage.
As cleavage proceeds,
these spaces come to-
gether to form a large
central cavity, the
blastocoel, surrounded
by a layer of cells,
the blastoderm.  As a
result of the size
difference between
cells in the two hemi-
spheres, the blasto-
coel lies near the
animal pole.
 18             4
 0
 0
 2              2103           0
 0
 0
 2              2104           0
 0
 0
 2              2105           0
 0
 0
 2              2106           0
 0
 0
 2              2107           0
 0
 0
 2              2108           0
 0
 0
 2              2109           0
 0
 0
 2              2110           0
 0
 0
 2              2111           0
 0
 0
 1              0              18
The embryo at this
time is called a
blastula.  The frog
blastula reaches the
8-cell stage in 3
hours and the 10,000-
cell stage in 6 hours.
The final feeding tad-
pole, grown in 110
hours, will have about
1 million cells.
When the blastula has
completed about 12 or
13 divisions, gastru-
lation begins.  To
continue, select Chap-
ter 3 from the Table
of Contents.
 18             4
 0
 0
```

## B.4 Chapter 3, Gastrulation, Data File (CHPTR3.MOK)

```
 28
 1              0              18
The blastocoele of the
frog embryo, at the
10,000 cell stage, has
a floor of large,
yolk laden cells sev-
eral cell layers deep
and a thin roof of
small, yolk poor
cells.  The develop-
mental fates of the
surface cells of this
embryo have been
mapped by the vital
dye marking method.
Would you like more
detail on vital stain-
ing? Select YES or NO,
push left button.
 18             4
 0
-3
 1              0              17
Vital dyes stain sur-
face cells without
drasticly affecting
their viability, and
so they can be used to
follow the displace-
ment of the cells
during development.
The resulting maps of
the embryonic surface,
indicating the pro-
spective fates of the
different regions, are
called fate maps.  The
amphibian embryo was
mapped correctly by
Vogt in 1929.
 18             4
 0
 0
 1              0              5
The fate maps you are
about to see document
the prospective fates
of the surface cells
of the frog embryo.
 18             4
 0
 0
 0              2113           0
 0
 0
 1              2113           11
The prospective ecto-
derm lies in the dark-
ly pigmented animal
hemisphere.  The pro-
spective notochord and
mesoderm lie in the
equatorial zone, and
the prospective endo-
derm lies in the
pigment-free vegetal
hemisphere.
 18             4
 0
 0
 0              2114           0
 0
 0
 0              2115           0
 0
 0
 0              2116           0
 0
 0
 1              0              11
The ectoderm and the
mesoderm will eventu-
ally end up inside the
organism.  The process
of folding which
brings these cells to
the interior, leaving
the ectoderm covering
the exterior portions
of the embryo, is
called gastrulation.
 18             4
 0
 0
 1              0              12
The first morpho-
logical indication of
gastrulation is the
appearance of a slit-
like blastopore at the
lower edge of the grey
crescent.  The grey
crescent lies on the
future dorsal side of
the embryo at the mar-
gin between the two
hemispheres.
 18             4
 0
 0
 0              2117           0
 0
 0
 4              2120           0
 0
 0
 1              0              15
The blastopore is
formed by the sinking
below the surface of
endodermal cells at
the base of the grey
crescent.

Would you like more
information on the
formation of the dor-
sal lip of the blasto-
pore?  Select 'YES'
or 'NO'.


 18             4
 0
-16
 1              0              11
The initial indenta-
tion of the cells to
form the blastoporal
groove is due to an
indentation of the
superficial cell
sheet, which in turn
depends on a change in
shape of endoder-
mal cells at this
site.
 18             4
 0
```

0
1 0 18
By contraction of microfilaments at the base of the cells, they develop long, narrow necks and bulbous bases and are called 'bottle cells'. because the cells remain tightly attached to their neighbors, an indentation in the cell sheet results from this change in cell shape. The dorsal lip of the blastopore is often called the primary organizer of the cell.
18 4
0
0
1 0 18
As gastrulation continues, surface cells converge toward the blastopore and turn inward, causing the two ends of the blastopore groove to extend around the embryo and eventually meet. The endodermal cells rimmed by blastopore form the 'yolk plug'. As gastrulation concludes, the embryo's mass shifts, the endoderm withdraws inside, and the egg rotates in place.
18 4
0
0
5 2349 0
0
0
0 2350 0
2540
0
1 0 12
The internal changes of gastrulation result in the shrinkage of the blastocoele and formation of the archenteron, which will later become the lumen of the gut. The following movie shows in diagrams the changes that occur during gastrulation.
18 4
0
0
1 0 8
Note how the surface cells move inward over both caps of the blastopore to bring the mesoderm and endoderm inside. The blastocoele shrinks as the archenteron is formed.
18 4
0
0
6 2541 0
0
0
0 2542 0
2758
0
7 2759 0
0
0
0 2760 0
2851
0
8 2852 0
0
0
1 2852 3
Would you like to see this movie again, with no interruptions?
18 4
0
-28
0 2541 0
2852
0
1 0 7
With the disappearance of the yolk plug, gastrulation concludes. To continue, select Chapter 4, Neurulation, from the Table of Contents.
18 4
0
0

## B.5 Chapter 4, Neurulation, Data File CHPTR4.MOK)

```
 20
 1              0               17
As gastrulation con-
cludes and the yolk
plug disappears, neuru-
lation begins.  The
dorsal ectoderm flat-
tens and thickens to
form the neural plate.
The edges of this
plate rise above the
surface to form the
neural folds, which
flank a central de-
pression called the
neural groove.  This
groove extends along
the entire middorsal
line of the embryo.
 18             4
 0
 0
 1              0               8
The neural folds even-
tually meet above the
deepening neural
groove, where they
fuse to form the
neural tube, the basis
of the central nervous
system.
 18             4
 0
 0
 0              2874            0
 3572
 0
 1              3572            2
Would you like to see
the movie again?
 18             4
 0
-6
 0              2874            0
 3572
 0
 1              0               4
The diagrams that
follow depict the var-
ious stages of neuru-
lation.
 18             4
 0
 0
 9              3576            0
 0
 0
 10             3577            0
 0
 0
 11             3578            0
 0
 0
 0              3579            0
 0
 1
 0              3580            0
 0
 1
 0              3581            0
 0
 1
 0              3582            0
 0
 1
 0              3583            0
 0
 1
 0              3584            0
 0
 1
 0              3585            0
 0
 1
 1              3585            12
The flattening of the
neural plate and the
rising of the neural
folds can be accounted
for by changes in the
shapes of individual
cells.

Would you like more
detailled information
on the changes in
cell shapes?
 18             4
 0
-20
 1              0               10
During neurulation,
cells of the neural
plate become taller
and thinner.  Cells of
the neural folds be-
come wedge-shaped by
means of constriction
at the cell apex, and
cause the folding of
the cell sheet.
 18             4
 0
 0
 1              0               14
Drug experiments indi-
cate that both micro-
filaments and micro-
tubules are involved
in these shape
changes.  During the
apical constriction,
bundles of microfila-
ments are seen to en-
circle the cell apex
like a draw string.
Microtubules play a
part in cell elonga-
tion.
 18             4
 0
 0
 1              0               14
The neural tube is the
rudiment of the cen-
tral nervous system.
At its anterior end it
expands and elaborates
to form the brain,
while in the trunk it
becomes the spinal
chord.

To continue select
Chapter 5, Morphogen-
esis, from the Table
of Contents.
 18             4
 0
 0
```

## B.6 Chapter 5, Morphogenesis, Data File (CHPTR5.MOK)

```
 19
 1            0            7
During neurulation,
the frog embryo
elongates, and by the
end of neurulation
the organism looks
more like a tadpole
than an egg.
 18           4
 0
 0
 0            3588         0
 0
 0
 1            3588         6
A diagramatic cross-
sectional view shows
that most of the
basic elements of the
tadpole's body have
been formed.
 18           4
 0
 0
 0            3590         0
 0
 0
 0            3591         0
 0
 0
 0            3592         0
 0
 0
 12           3593         0
 0
 0
 13           3589         0
 0
 0
 1            3589         4
Evaginations of the
gut will become the
liver, pancreas, and
other organs.
 18           4
 0
 0
 0            3595         0
 0
 0
 1            3595         5
A cross-section through
what will become the
brain reveals an
already complicated
folding.
 18           4
 0
 0
 0            3598         0
 0
 0
 0            3599         0
 0
 0
 0            3600         0
 0
 0
 1            3600         5
More folding will com-
plicate parts of the
organism, but at this
point the basic tadpole
shape has been made.
 18           4
 0
 0
 0            3588         0
 0
 0
 0            3602         0
 0
 0
 0            3607         0
 0
 0
 1            3607         5
For a summary of the
material that has been
presented, select
Summary from the
Table of Contents.
 18           4
 0
 0
```

# B.7 Chapter 6, Summary, Data File (CHPTR6.MOK)

```
20
1              0              5
Once fertilized, the
frog zygote undergoes
a number of rapid
divisions to form a
many-celled blastula.
18             4
0
0
0              76             0
2101
0
1              0              5
This blastula next
invaginates near the
grey crescent to form
a multi-layered gat-
trula.
18             4
0
0
0              2541           0
2852
0
1              0              5
Neurulation then
occurs, with the for-
mation of the neural
tube from the dorsal
ectoderm.
18             4
0
0
0              2860           0
3572
0
1              0              6
During neuraltion the
organism has elonga-
ted, and the basic
shape and basic inter-
nal organs of the
tadpole are present.
18             4
0
0
0              3542           0
0
0
1              0              12
What has happened
here?  A complicated
organism has been
formed from a hollow
ball of cells.  The
main mechanism of this
complication is the
process of folding,
which in turn is made
possible by changes in
the shapes of
individual cells.
18             4
0
0
1              0              15
The brain, for
example, a most com-
plicated final pro-
duct, begins as one
end of an uncompli-
cated neural tube.  In
similar ways, many of
the complicated move-
ments of morphogenesis
can be broken down and
seen as elaborations
on much simpler, basic
processes, like the
folding of a sphere to
form a tube.
18             4
0
0
0              16040          0
0
1
0              16049          0
0
1
0              16051          0
0
1
0              16055          0
0
1
0              16059          0
0
1
0              16067          0
0
1
0              16074          0
0
1
0              16075          0
0
1
0              16077          0
0
1
0              16076          0
0
0
```

## Bibliography

Alberts, Bruce, et al. *Molecular Biology of the Cell.* New York, NY: Garland Publishing, Inc., 1983.

Bejar, Isaac I. Videodiscs In Education, Integrating the Computer and Communication Technologies. *Byte*, June 1982, pp. 78-104.

Browder, Leon W. *Developmental Biology.* Philadelphia, PA: Saunders College Publishing, 1984.

Hon, David. Interactive Training In Cardiopulmonary Resuscitation. *Byte*, June 1982, pp. 108-138.

Horder, Alan. *Video Discs - Their Application to Information Storage and Retrieval* (NRCd Publication No. 12). Bayfordbury, Hertford, England: National Reprographic Centre for documentation, June 1979.

Graphics and Interactive Techniques. *Movie-Maps: An Application of the Optical Video Disc to Computer Graphics.* Seattle: 1980.

Mohl, Robert. *Cognitive Space in the Interactive Movie Map: An Investigation of*

*Spatial Learning in Virtual Environments.* Doctoral dissertation, Massachusetts Institute of Technology, 1982.

Okum, Henry. Picassofile or Using a Computer to Look at Picasso. MIT Architecture Machine Group: Internal Paper.

Purcell, Patrick and Henry Okum. Information Technology & Visual Images: Some Trends & Developments . Paper presented at the General Conference Munich meeting of the IFLA. 1983.

Sigel, Efrem. *Video Discs: The Technology, the Applications and the Future.* White Plains, New York: Knowledge Industry Publications, Inc., 1980.

Visage, Inc. *V:Exec Users's Guide* (Version 2.0 ed.). Author, 1984.

Visage, Inc. *V:Paint I User's Guide* (Version 2.0 ed.). Author, 1984.

Yellick, Steven E. *The Authoring of Optical Videodiscs with Digital Data.* Master's thesis, Massachusetts Institute of Technology, 1979.