King Saud University
College of Computer and Information Sciences
Computer Applications Department

An Educational Multimedia Application for the PC System

Reference Manual

Submitted By:

Najwa A. Al-Twaijry
Abeer S. Al-Humaimeedy
Reem S. Al-Moqbel
Manal M. Al-Zeer

Supervisor
Lec. Afshan Jafri

Project Submitted in Partial Fulfillment for the Degree of B.Sc. in Computer Applications

Safar 1421H. (May 2000)
Second Semester
Riyadh, Saudi Arabia

King Saud University
College of Computer and Information Sciences
Computer Applications Department

An Educational Multimedia Application for the PC System

Reference Manual

Submitted By:

Najwa A. Al-Twaijry
Abeer S. Al-Humaimeedy
Reem S. Al-Moqbel
Manal M. Al-Zeer

Supervisor
Lec. Afshan Jafri

Project Submitted in Partial Fulfillment for the Degree of B.Sc. in Computer Applications

Safar 1421H. (May 2000)
Second Semester
Riyadh, Saudi Arabia

To our families.
Thank you for your support.

ALL GRATITUDE IS DUE TO ALLAH

It is our pleasure to express our appreciation to our supervisor,
Lec. Afshan Jafri
for her invaluable guidance and help, and most of all for her
encouragement and patience during the preparation of this project.

We also wish to thank Mrs. Connie for her time and assistance.

We would also like to thank
Hesham Al-Twaijry and Sulaiman Al-Humaimeedy
for helping us and providing us with the
software we used.

An additional thanks is due to
Lamya Al-Fawzan and Hani Al-Twaijry
for recording the voices we used in our project.

Abstract

One of the most important uses for computers is their utilization in classrooms. Computers can be efficiently used as a medium of instruction to teach students about the basic components of a computer. This reduces the personal requirements for the Computer Application Department. Additionally, an interactive lesson allows the students to study at their own pace. This report provides a framework for the development of such a tutorial for computer components. It presents the steps that need to be taken to develop such a lesson. Finally, a tutorial to show the validity of these steps is developed.

الملخص

من أهم المجالات التي يستخدم فيها الحاسوب مجال التعليم. و في الوقت الحاضر، يتم استخدام الحاسوب في الفصول الدراسية بطريقة فعالة، كوسيلة لتعليم التلاميذ مختلف العلوم، و منها علوم الحاسب، حيث يستخدم لتعليم الطلاب العناصر الأساسية في الحاسب الآلي، وهذا يقلل من حاجة قسم تطبيقات الحاسب الآلي إلى أساتذة مختصين. كما أنه يتيح لكل طالب أن يتفاعل مع الدرس حسب قدراته الخاصة.

هذا التقرير يزودنا بأسلوب لتطوير درس عن أجزاء الحاسوب، حيث أنه يقدم الخطوات الأساسية لتنميته. أخيراً، تم تطوير درس لبيان صحة هذه الخطوات.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Computers are now playing a vital role in society. Therefore, many young people now choose computers as their career. The department of computer applications is the department in King Saud University whose responsibility is to teach female students computers. This department is one of the most popular departments in the university; it faces an ever-increasing number of students. This fact has placed a burden on the department in that it has to present a larger number of sections for its lab classes. This increase in the number of lab classes requires a large number of teaching assistants to supervise them. This means that the university has to recruit more and more assistants. Currently the department is facing a shortage in the number of qualified assistants.

Assigning assistants to handle the 341 CAP lab, which teaches students about the components of a computer, is not very efficient for a variety of reasons. The first reason is that the teaching assistant has to perform the same time-consuming procedure for every lab several times. This causes the teaching assistant to waste her time doing the same procedure. This repetition has the danger that with boredom the assistant may inadvertently, due to the lack of attention, provide the students with some wrong information. The second major problem with the current method for teaching the students is that the allocated time for the lab may not be enough to give the teaching assistant a chance to explain and demonstrate to all the students the lab procedure. Finally, with the current method, students are left by themselves while the teaching assistants are explaining the procedure to the other students. This can cause the students to lose interest in the demonstrations, so that when it is explained they will not be concentrating and they therefore will not comprehend what is being explained.

This project intends to alleviate the problem of not having enough teaching assistants for the labs by developing a tutorial that is able to teach a subject.

To successfully instruct the students about computers we will need to present the information in a clear method that will enable us to show the students the various components of a computer. Cognitive studies have shown that students understand and remember a fact when the fact is presented concurrently using several media. Therefore, we are going to present the information for the tutorial as a multimedia lesson. This requires the use of some specialized software that is capable of integrating text, video and audio.

1.1 Problem Statement

An interactive multimedia system is to be developed to assist CAP 341 (PC Environment and Peripherals) students. The multimedia system will explore the computer components of the Pentium based computer.

The software should allow students to interact with the following motherboard components: the CPU, RAM, BIOS, Expansion Slots, Serial and Parallel Ports, the Clock Chip, and the IDE Controller. The student should be able to visually see the components, be instructed on the installation and learn about the functionality of each component.

The software should be based on proper multimedia and Computer Based Instruction (CBI) principles. It should utilize images, video, sound, animation and text to deliver information.

Deliverables: a technical report, user manual, and a CD containing the executable program. The program should run on Windows based Intel machines.

1.2 Related Work

A previous project was submitted in Safar 1418H, June 1997. The project title was "Computer Based Tutorial Designed to Assist Learning CSC100", and was supervised by Lec. Salwa Al-Jasser. It was an educational program, used to assist learning the topics and basic concepts of the introductory course CSC100, in the Computer Applications Department. The

educational methodology used was the tutorial. It explains computer concepts in a simplified and interesting manner, using three multimedia elements: sound, graphics, and animation. This software was implemented using an authoring package built for interactive learning (Authorware version 5.5).

1.3 Project Importance

Computer Based Instruction (CBI) is one of the fastest growing fields in computers. CBI allows a department to make the best use of its resources. They are extremely useful in teaching classes that are mostly based on rote. Such a class is CAP341. Therefore, it is extremely useful to change the method of teaching this class to CBI. This project develops a tutorial to teach the students the material covered in the class.

1.4 Director System Requirements

The minimum requirements for Windows developers are:

- A 66MHz 486 PC, with Pentium or Pentium Pro preferred.
- A minimum of 16MB RAM.
- A 15", 8-bit color monitor supporting a minimum of 256 colors, and 640×480 resolution.
- Approximately 30-40MB of free hard disk space.
- SoundBlaster-compatible sound card and speakers.

- The latest driver for QuickTime for Windows or video for Windows.
- A double-speed or faster CD-ROM drive to install Director.
- Windows 95/Windows NT 4.0 or later.

1.5 Organization

In this report we will develop a multimedia lesson that will explain the components of a computer and how they work. The report consists of the following chapters:

Chapter 1:

This chapter introduces the problem and it presents the need for the development of a tutorial for CAP 341. It gives the problem statement and discusses related work. It will then conclude with the system requirements for the software (Director 6.5) that is used to develop the system.

Chapter 2:

This Chapter discusses the Computer Based Instruction topic.

Chapter 3:

Covers the multimedia issue in detail by discussing the various media and their formats.

Chapter 4:

This chapter explains the motherboard information our project presents.

Chapter 5:

Chapter 5 presents the high-level flowchart that describes the tutorial. In addition this chapter presents the storyboard that was used in developing the tutorial. It concludes with the user interface and the graphical user interface (GUI) design issues and principles.

Chapter 6:

Describes the steps that we followed in solving the problem. This chapter also discusses the difficulties that we faced while developing the tutorial, and the solutions that we came up with to these problems. It also describes the implementation of the tutorial.

Chapter 7:

This chapter discusses future work that can be done to expand this lesson and gives a brief summary of the work done.

Appendix A:

Explains the software that was used in the development of the tutorial (Macromedia's Director 6.5).

Appendix B:

Lingo listing.

Appendix C:

Photograph listing.

Chapter 2

Computer Based Instruction

Computers have been used for instructional purposes since the 1960s. To take advantage of the computer's capabilites and not to waste them, an instructional CBI package should be developed only when the computer is likely to be beneficial. When the cost of instruction is very high, safety is a concern, and when individual student practice is needed are all situations when computers could be used to provide instruction.

2.1 The Process of Instruction

There are two instructional models; the expository model and the discovery-learning model. The following four phases are present at the expository model:

1. Presenting information.
2. Guiding the student.
3. Practicing by the student.
4. Assessing student learning.

The discovery model omits the first phase, or changes the order of the first two phases. There is evidence that higher-ability students benefit from discovery learning, while normal to low ability students do not benefit as much.

2.1.1 Presenting Information

Presenting new information takes a number of forms. Verbal information, the teacher may present rules or example. To teach the student non-verbal information, the teacher would model the required skills. This phase is instructor-centered; the teacher performs this phase.

2.1.2 Guiding the Student

This is an interactive phase, because it involves both the teacher and the student. In this phase, the student must perform under the teacher's guidance. The student may be required to answer some questions, practice a skill, or apply rules and principles. Then the teacher corrects the student's errors, and gives suggestions and hints. Guidance in the classroom may take the form of the teacher asking the questions and the student answering them. When the

student is learning from a book, questions at the end of the section are included as guidance. However, if the student does not perform correctly, then true guidance does not occur. It may occur at a later time, when the teacher reviews the student's work. Guidance is an important phase of instruction, because no student can learn all that is taught on a single exposure.

2.1.3 Practice

Practicing is student centered. During this phase, the student must practice what he has learned in order to retain information, and to be able to perform quickly with few or no errors.

2.1.4 Assessing Student Learning

Tests are usually used to assess student learning. Tests are usually used as a means of assigning grades, rather than to determine what instruction is needed for the student. Tests can provide information on the quality of teaching and future instructional needs.

2.2 Instructional Methodologies

Currently there are five methods of using the computer to provide instruction:

1. Tutorials.
2. Drills.
3. Simulations.
4. Games.
5. Tests.

Tutorials usually cover the first two phases of instruction. Drills and games engage in the third phase, and tests represent the last phase. Simulations may be used for the first and the second phases, the second and third phases, the first three phases, or for the fourth phase. It should be noted that computers are just one element in the teaching environment, along with teachers and other media. The computer may serve one or more phases, but when the computer is responsible for total instruction, then all four phases must be present.

2.3 Cognitive Psychology

When evaluating or designing a CBI package, the following areas of the cognitive theory are the most important:

2.3.1 Perception and Attention

The process of learning depends on the learner perceiving stimuli. However, human perception is constantly strained by competing stimuli. That is why presentations are designed for easy and accurate perception. Multimedia presentations provide elements, such as the use of color, sound, and animation that facilitate correct perception. The student attention must be attracted and maintained throughout the lesson. There are many factors that affect student attention, such as the level of involvement, lesson difficulty, variety and lesson pacing.

2.3.2 Memory

To enhance memory, two methods are used; organization and repetition. The method of organization is easier and more powerful. It also aids recall. However, when the use of organization is impossible, the use of repetition is

used instead. Repetition is used when there is a large amount of information, or when the information is impossible to organize.

2.3.3 Active Learning

Interaction in a lesson maintains student attention, as well as creating new knowledge and skills. One of the main advantages of CBI is its ability to require and act upon student interaction.

2.3.4 Motivation

Students must be motivated to learn. Simulations and games enhance motivation. Four factors are relevant to motivation: maintaining attention, material relevance, student confidence, and student satisfaction.

2.3.5 Control

Controlling the lesson sequence and content may be determined by the student, the lesson, or a combination of the two. Lesson success depends on which parts the student controls, and which parts are controlled by the lesson. Flexible student control is an advantage, but it has other effects, which must be taken into consideration. The student may not be able to determine the best path to take. When that is the case, and when the students are children, it is better to give the student as little control over the lesson sequence as possible.

2.3.6 Transfer of Learning

Transfer of learning means applying what the student has learned in the real world. Transfer of learning is affected by the type, amount and variety of interaction, by the realism of the lesson, and by the methodology used. Simulations provide the highest transfer.

2.3.7 Individual Differences

Students do not all learn at the same rate, for the same reasons. CBI provides individual instruction, which can be tailored to suit each student. Good software will adapt to the learner, giving extra help in the student's weak areas, and providing motivators that the student responds to.

Chapter 3

Multimedia

In the old days, multimedia referred to a combination of two media, for example, a slide show with background music. Nowadays, the term multimedia refers to information in different formats - text, images, sound, video, and animation – that is integrated into a single entity. These multimedia shows are very taxing on the system resources. However, the increased processing power of personal computers and the development of digital storage devices such as the CD-ROM, as well as the reduction of costs of each to the consumer have all contributed to the growing popularity of multimedia.

3.1 Interactive Multimedia

An interactive multimedia production allows users to interact with the information being presented and to control the flow of information; enabling users to navigate through the information at their own pace, based on their own special interests. Interactive multimedia utilizes hypertext. Hypertext is highlighted text that allows the user to quickly access more information by simply pointing at and clicking the highlighted text. This extra information may take the form of a definition of a term, or it may be a link to sounds, pictures, animation, and charts. The tremendous advantage of hypertext is that it negates the need to look at the bottom of the page, the end of a chapter, in the glossary or in the back of a book.

3.2 General Multimedia Requirements

The process of creating a multimedia application is complex. It is broken into several steps, starting with the collection of required material; text, drawings, and imported graphics are the most basic form of these. Other types of multimedia information are brought in next, such as sound, video and animation. Hardware and a proper software interface are usually sufficient for running a simple multimedia application comprised of text, graphics and simple animation. Running other types of multimedia requires specialized hardware. For example, to play sound or a video with a soundtrack, a sound card must be connected to the PC's speaker or to external speakers. To play a video or a complex animation, a proper VGA card that can run a full motion movie must be installed on the PC, because a slow VGA card maintains memory by skipping some frames. This reduces the quality of the multimedia

application. In general, multimedia machines must have a CD-ROM drive to run multimedia programs. These programs are usually stored on a CD rather than a floppy disk, because of the CD's large storage capacity. This storage capacity enables complex programs that take up a large space to be stored and then distributed among different computers.

3.3 Project Management and Multimedia

The complexity involved in the design and implementation of a multimedia project calls for breaking the task down into small parts, with different teams working together. Each team will undertake a single task. Teams for picture collecting, picture formatting, sound recording, editing, and video recording are only a few of these teams. The large number of teams makes project management an essential part of any multimedia project, for it can assist in three areas: planning, scheduling, and control. These three together determine the most important part of any job, which is the cost. A well-planned, scheduled project is coordinated so that no team sits idle, waiting for the results of another team's work.

3.4 Benefits of Multimedia

Multimedia combines the impact of TV with the power of personal computing to create applications that are:

- **Efficient**: Multimedia replaces abstract textual information with visual, audible information, such as a video.
- **Direct**: Multimedia delivers information using the best, most direct medium. For example, multimedia utilizes the sound capabilities of a computer to demonstrate pronunciation in a language-teaching application.
- **Personal**: The combination of elements allows the design of personalized applications, to be used in their own appropriate setting.
- **Interactive**: Users are able to interact with the information being presented, to control the flow of information and the way it will be presented.

3.5 Multimedia Characteristics

Multimedia has the ability to revolutionize teaching methods, knowledge communication, entertainment, and many other fields. Multimedia is:

- **Powerful:** The multimedia presentation system is more effective than traditional presentation systems. Users have longer attention spans and higher retention rates, which makes it a very effective teaching and communication medium.
- **Cost effective:** The falling price of computer equipment coupled with the low cost of mass CD-ROM replication makes multimedia very cost-effective. In addition, multimedia reutilizes existing content such as videos and text, and easily updates digital material.
- **Flexible:** It is relatively easy to update the content of a multimedia application, and presentations can be made to any size group through the

use of personal computers, overhead projectors or video projection systems.

- **Creative**: Multimedia presentations allow for the development of highly effective communication tools, using powerful graphic and video capabilities, and superior design tools.

3.6 Multimedia Terms:

3.6.1 SCSI:

The Small Computer Systems Interface (SCSI) standard provides an easy common connection for CD-ROM players, scanners, and external hard drives as well as other devices. SCSI enables up to 7 devices to be connected to a single SCSI port. This arrangement is known as the Daisy-chain Arrangement.

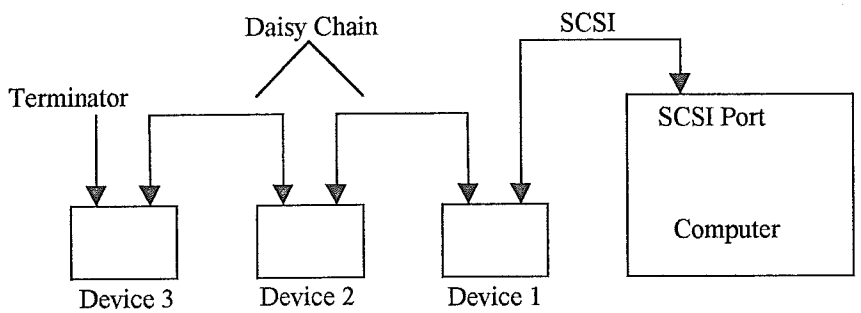


Figure 3-1 *The Daisy-Chain Arrangement*

The multimedia production computer needs one or two SCSI ports; usually one SCSI port is sufficient.

3.6.2 Scanners:

Scanners convert images into digital signals for storage and processing by computers. They come in many forms: Hand-held, Sheet-feed, Flatbed, and Slide scanners. Flatbed and Hand-held scanners are flexible and are used the most in multimedia production. Sheet-feed scanners skew the page slightly, and are suitable for text scanning but not image scanning. For this reason, they are seldom used in multimedia production.

3.6.3 Audio:

Sounds added to a multimedia production include voice, music and special effects. The most widely spread sound formats are:

- *MIDI:* The Musical Instrument Digital Interface (MIDI) is a simple computer code recognized by synthesizers. It is a computer-generated sound that is converted by a synthesizer into sounds, and the sound quality depends on the synthesizer used to play the sound.
- *Waveform Audio*: Is a digitized audio file that can be manipulated using software. It does not describe commands passed to a MIDI synthesizer as MIDI does. Waveform audio files have a .WAV extension. Sound quality depends on the sampling rate used; more sample points achieve higher quality sound, but require a larger storage space, as can be seen from the following figures.

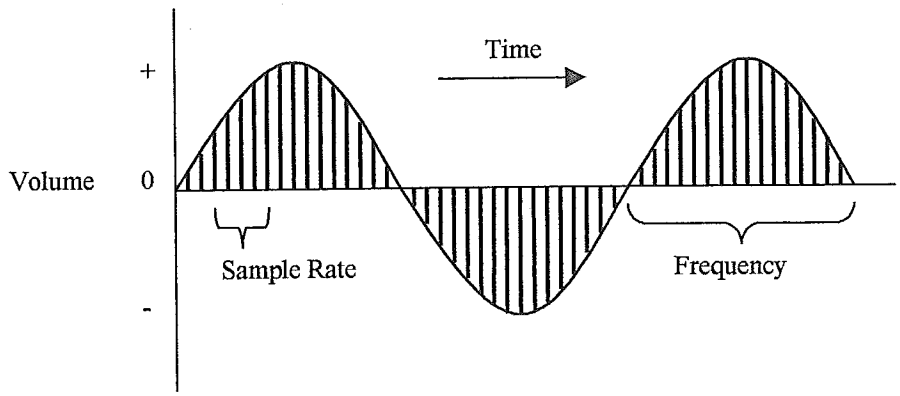


Figure 3-2 *High Sampling Rate*

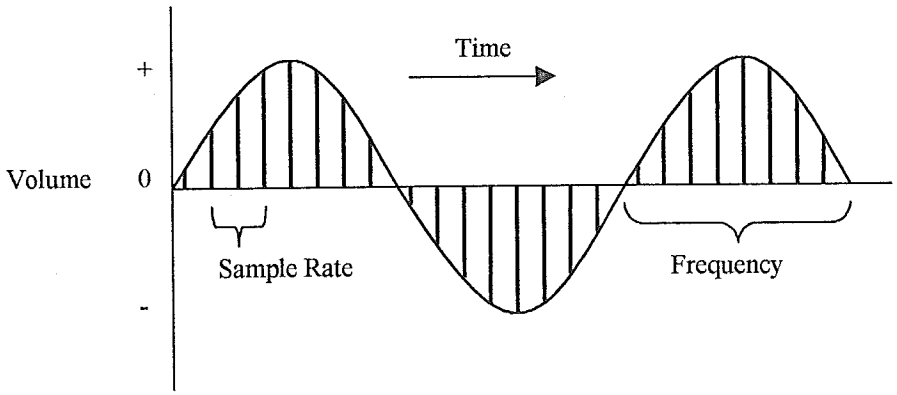


Figure 3-3 *Low Sampling Rate*

3.6.4 Video:

Digital video has been a part of multimedia development for many years. There are two different varieties of video quality. One is Desktop Digital Video, which refers to video that can be displayed on a monitor. The other is High-end Digital Video, which is used in movie theaters and on television. Desktop digital video is a stream of sights and sounds that provide visual and auditory images to the viewer. It is comprised of a series of still photos, or frames. The number of frames displayed per second is known as the digital video's film rate. Standard film rates vary from 24 fps to 30 fps for television, but require vast amounts of storage and resources. That is why most digital videos used in multimedia have to reduce the number of frames per second, the on-screen size of the video, or the quality of the audio track.

3.6.5 Animation:

Animation is mainly a matter of illusion. When a number of slightly different images are displayed in quick succession, the human brain perceives these images and fills in the gaps, creating an illusion of smooth movement. To create an animation sequence, two or more similar images are needed, plus a means of displaying them together at some minimum rate of speed. An important part of creating a convincing animation involves the user's perception and expectation of reality. This means that the animation should behave in a way that is familiar to the audience. For example, a ball that falls to the ground remains perfectly round until it hits the ground. At that point, the energy of the collision distorts the ball slightly, making it a bit flatter. When the ball rebounds into the air again, it regains its original shape. When creating such an animation sequence, the developer would need a picture of the round ball, and at least one picture of the ball as it hits the ground. More pictures would create a smoother animation.

3.6.6 Windows:

Microsoft Windows has a set of Dynamic Link Libraries (DLLs), which are tools that allow applications to communicate with any hardware supported by Windows. This eliminates the need to spend vast amounts of time writing custom drivers to interface the multimedia software to hundreds of different hardware components.

3.6.7 Compression:

Compression reduces the size of files created when storing analog sound or complex color images in digital computer format. There are a number of different compression techniques. One requires the users to have the same technology built into their players. Others are only used internally in the production system. Compression reduces the amount of information stored, resulting in lower quality sound and images. Two of the most widely spread compression techniques are:

- **JPEG:** Joint Photographic Experts Group (JPEG) is used for compressing the image and reducing file size. The digitized image is broken into 16×16 pixel blocks. Every other pixel is then removed from the block, converting it into an 8×8 block. Further processing reduces the file size, and data are then stored. Reversing this process reproduces the image. That is why JPEG is highly computation intensive, and is usually implemented using a dedicated microprocessor.
- **MPEG:** Motion Picture Experts Group (MPEG) standard offers greater compression than JPEG. MPEG is capable of handling multiple frames. It saves storage because it stores a single frame then just records the changes to the subsequent frames.

3.7 The Important PC Aspects In Multimedia

The following criteria are important when purchasing a machine to be used in multimedia production:

3.7.1 Memory:

Multimedia production programs are complex and require vast amounts of memory. The images and sounds they process require a lot of space in memory so that the software can manipulate them.

3.7.2 Storage:

Storage is where data are kept when the computer is turned off. Permanent storage consists of floppy, hard, and optical disks. Multimedia requires large storage space, for image and sound files, as well as video files, consume a large amount of memory.

3.7.3 Computing power:

Playing multimedia productions on old PC's with slow computing power greatly degrades the quality of the presentation. A good machine is needed for the production platform as well. Even so, memory and storage are more important than computing power.

3.8 Multimedia Uses:

Multimedia is a communications tool that can be used to provide:

3.8.1 Entertainment:

Multimedia games include videos, animation, and sound; both voices and sound effects, and they are interactive. Multimedia is also used for children's stories, music, and video and museum tours.

3.8.2 Business uses:

- *Presentations:* Business presentations that include sound and animation capture the attention of the audience more than traditional presentations. A presentation that is designed to be interactive can be sent to the offices of the customers to view at their leisure.
- *Point-of-Sale:* Multimedia presentations can be used in retail stores to provide information to potential customers, to supplement the efforts of the sales staff, and to catch the attention of a passer-by. Point-of-Sale multimedia has many advantages, for it can run on its own, with no staff member intervention. Some customers may feel more comfortable obtaining information in this way, especially if they do not intend to make a purchase right away. In addition, Point-of-Sale multimedia can demonstrate some features of products that cannot be demonstrated in a store setting, such as features of products that are too expensive to carry in stock.
- *Video Conferencing:* The final multimedia application for business users is video conferencing. Real-Time video conferencing is not really a multimedia production, but it does use the same hardware. When video conferencing uses audio clips, text, charts, images, and video in addition to the voices and images of conferees, it becomes multimedia.

3.8.3 Education and training:

Multimedia is a very powerful tool when used in the educational environment, for it can provide visual and audible training that is well-suited to the individual needs of students. It can also be used to teach the disabled and young children. Yet another beneficial application is using multimedia in training and simulations, where workers or students can learn how to operate expensive equipment or train in simulated hazardous environments.

3.9 Producing a Multimedia Presentation

The first step in producing a multimedia presentation is determining the goal and objectives of the project, by deciding what to teach, and the level of skill and knowledge the students must achieve at the end. The next step is to outline the audience and their level. The third step is outlining the scope and content of the presentation. This includes the user interface design, the presentation flowchart and storyboard. After that, the components of the multimedia project are produced. Text, sound, video, images and graphics are all examples of multimedia components. The final step is creating the multimedia application, then testing and re-testing it on colleagues and students.

3.10 The Importance of Storyboarding in Multimedia

Storyboarding is probably the most critical step in multimedia production, yet novice multimedia producers often overlook it. It is the process of drawing a preliminary sketch showing the various points in the multimedia show. Producing a storyboard is a relatively simple task, using pencil and paper and drawing concepts with squares and circles. In this way, the multimedia producer does not have to be an artist. Storyboarding gives the producer an idea about how many particular pieces will be needed for a multimedia presentation. It also makes it easy to anticipate potential problems early in the production process. Outlining the presentation in such a way helps the producer think out all sequences and avoid changing the way the presentation is headed halfway through production, wasting time and effort.

Chapter 4

The Motherboard

A personal computer's motherboard contains the components that make up the computer. The motherboard is a printed circuit board in which the traces (wires) are used to interconnect the components. In addition, the motherboard has the ability to increase the number of components that it has by the use of some expansion slots. Numerous components make up the motherboard. In our project, we have explained the components included in this chapter.

4.1 CPU

Computation takes place in the microprocessor. When the computer retrieves a file from the hard disk, the information is sent to the main memory and then to the microprocessor. The microprocessor, or the processor, is the brain that handles the actual work. In many PCs the processor gets help for certain jobs; for instance a math coprocessor does the heavy arithmetic, or the video board may have a special chip to handle the graphics. There are several types of processors in use today. Older computers have processors made by the Intel Corporation called the 8088, the 80286, the 80386, and the 80486. These chips seem very slow compared to the current generation of processors, which have high clock rates. The processing that occurs in the processor can only happen on a tick of the clock and therefore the higher the clock rate of a processor the higher is the performance of the machine.

4.2 Memories

Clock speed is not the only factor in determining the performance of a personal computer. The size and speed of the memory is just as important in deciding the performance of the PC. Computers have several kinds of memories, and memory is designed as a hierarchy where the fastest memory is kept beside the processor and the slowest memory is kept the farthest away from the processor. The management of this memory hierarchy is the responsibility of the operating system with the help of some hardware in the processor. The PC memory can be split into two distinct parts; the first is on disks or drives, the second on a variety of memory chips. When a user is working at his PC, data is temporarily stored in the memory chips. After

finishing the work, the PC copies the data onto the disks. The memory that stores the data while the user is working on it is called RAM or Random Access Memory. This memory is volatile, that is, it requires power to be available to the memory chips so that the data is retained. In contrast the disk memory is non-volatile and it retains the data after the power is switched off in the PC. The main memory for the processor is divided into several levels with the closest level to the processor being called the cache memory. The cache is a small, extremely fast memory. Another type of memory that is found in the PC is called the ROM BOIS, which stands for Read-Only-Memory, Basic Input/Output System. This memory consists of a group of chips that contain the permanent code that the PC uses when it is first turned on. The ROM BIOS also contains the code that the operating system uses to communicate with the keyboard, monitor and other system hardware. The last type is called RAM, which is composed of dynamic RAM (DRAM) chips.

4.3 Buses

The third component in determining the performance of a PC is the buses that are inside the PC. A bus is composed of a collection of wires that are used to connect the processor to the system peripherals, and on which the information is transferred. The wider the data bus the more data the bus is able to transfer in a single bus cycle. The original bus in the IBM PC used an 8-bit bus to transmit the data. This bus was called the ISA or Industry Standard Architecture bus. This bus was later expanded to 16-bits. To remain compatible this bus also accepts 8-bit cards. EISA, which stands for Enhanced Industry Standard Architecture bus, doubles the data transfer size to 32-bits. This bus is backwards compatible with the ISA bus cards. However the EISA bus did not

increase the clock speed of the bus. The bus clock speed remained the same as in the ISA bus. Therefore, the PCI bus was developed. This bus solved many of the problems that were present in the EISA bus, namely, the low transfer rate, and the need to manually configure the cards that are to be connected to the system. The PCI bus was the first bus that had cards that were plug and play. These cards could be configured under to control of software without the need for the user to manually set dipswitches. This was a major improvement over the previous cards since it allowed non-technicians to install cards in their systems.

4.4 Clock Chip

The clock chip contains a precisely cut crystal that vibrates at a specific rate when electricity is sent through it. This crystal is also found in the quartz watches. The vibrations of the crystal are used to generate the clock that is used to time all the operations in the personal computer.

4.5 Parallel Port

The parallel port is the basic interface for printers. In its simplest form, the parallel port is a unidirectional. It transfers data in one direction from the computer to the printer. More modern parallel ports support bi-directional data transfers. That is the data can flow from and to the printer. These newer parallel ports may also support higher data transfer rates. These new parallel

ports are known as enhanced parallel ports or EPP for short. These new EPP ports are essential for the operation of the latest printers.

4.6 Serial Port

Serial ports transfer data a bit at a time, in contrast to parallel ports where each byte is sent at once. The transfer rate for serial ports is lower than that for parallel ports. The serial port is used to connect relatively slow devices like the modem or devices that have low data transfer rates like a mouse. A relatively recent addition to the serial ports is the universal serial bus or USB port, which is a high-speed serial connection. Devices may be connected to the system while the PC is powered on and the machine will recognize this new device and the user will be able to use this new device immediately.

4.7 Sound Card

A sound card is connected to the personal computer using one of the expansion slots in the machine. There are several kinds of connectors on the sound card. These different connectors are used to connect the sound card to the different audio devices. For example, there is a connector that is used to send signals to the external speakers. A different plug is used to receive signals for the microphone. Yet another connector is used to receive the inputs from an external audio source such as a CD player. Some sound cards have a connection that can be used to connect a joystick. Replacing a sound card

requires replacing the software that is used to control the card, which is called the audio driver.

4.8 Video card

The video card is connected to the system bus. It affects the quality of the appearance of the data on the screen. A video card determines how quickly data can be sent to the screen. The video card can be changed very easily, and requires that the software that is used to drive the old card be replaced with new software that is compatible with the new card. Video cards vary in the number of colors that they can display. Video boards also differ in the number or dots, or pixels that a card can display on the screen. Another aspect of the video board is whether they act as simple frame buffers that simply display the data that is available, therefore requiring that the CPU do all the video work, or they contain some circuitry in the form of a specialized video processor that is capable of performing most of the processing required to present the data on the screen thereby reducing the processing burden of the CPU.

4.9 Secondary storage

There are several types of secondary storage. The most common kind of removable storage is floppy drives. Floppy drives come in several different formats and sizes although the most popular nowadays is the 3.5-inch floppy with a capacity of 1.44 megabytes.

Hard disks are fixed secondary storage. The hard disks are part of the memory hierarchy of a PC. They are the farthest part of the memory system from the CPU. Their system consists of several components; the actual hard disk and the hard disk interface. The interface is a disk controller that is used to communicate between the hard disk and the system. There are several different types of controllers, such as SCSI and IDE. These controllers differ in the speed at which the data is transferred to and from the disk.

Another feature that is differentiates between the hard disks is the how the data is encoded on the disk. The different encoding schemes have different capacities for data storage.

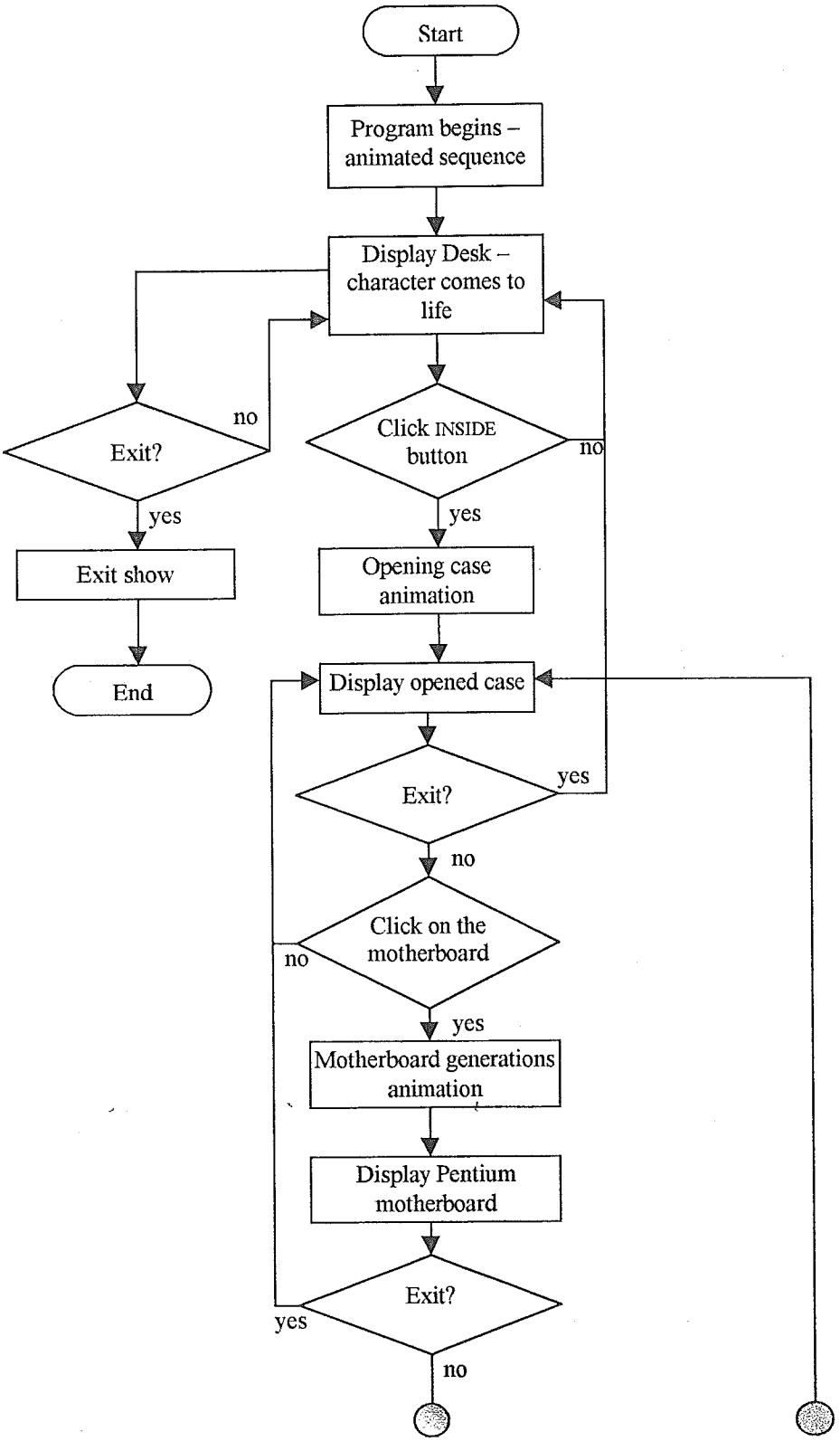
In addition to the two previous kinds of secondary storage several other less popular kinds exist. These other kinds or secondary storage include CD-ROMs which are optical disks that are used to read data. Another type of secondary storage is IOMEGA zip drives which can be thought of as high capacity floppy disks.

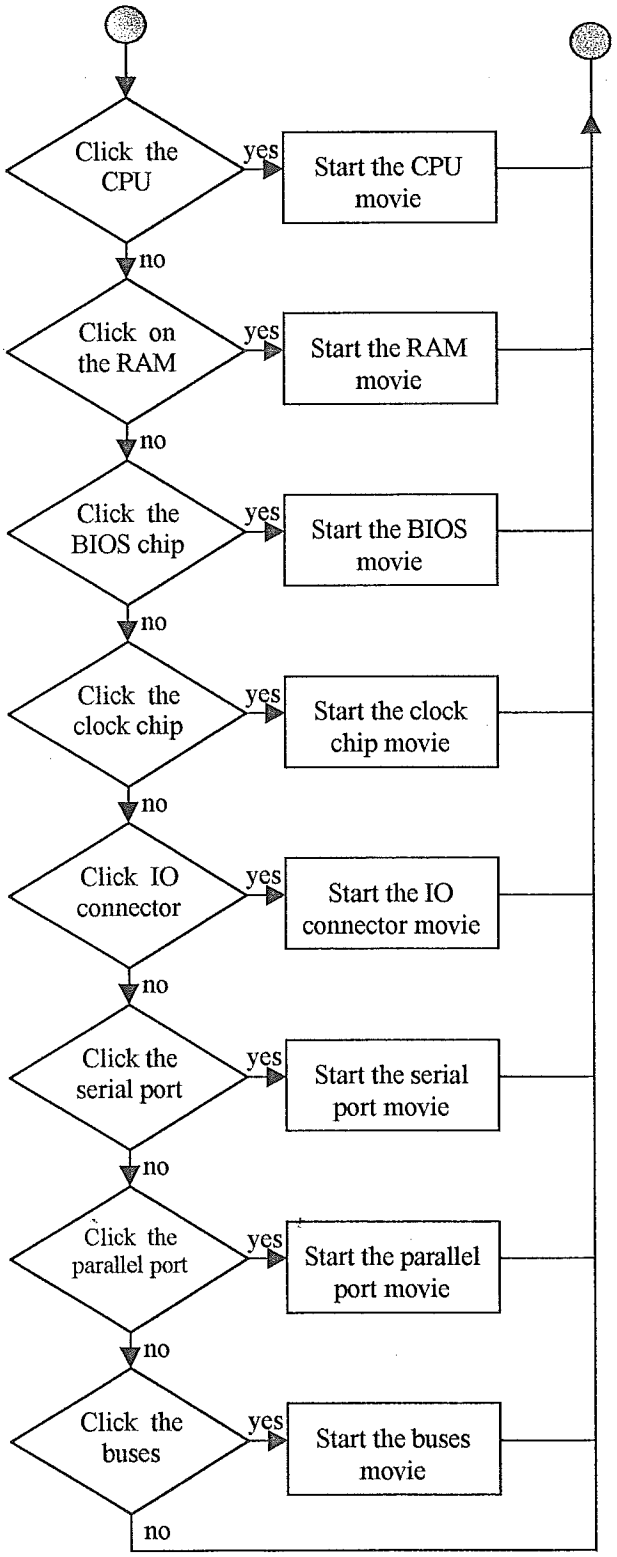
Chapter 5

Graphical Interface

This chapter covers the graphical interface for our program. The first section presents the high-level flowchart of the program. The second section is the storyboard of our multimedia presentation. As mentioned previously, the storyboard is the most important step in multimedia design. The storyboard presents the main movie and one component movie (Please refer to chapter 6 – Methodology and Implementation – for more information on our project layout). The chapter ends with a small section on User Interface and Design fundamentals.

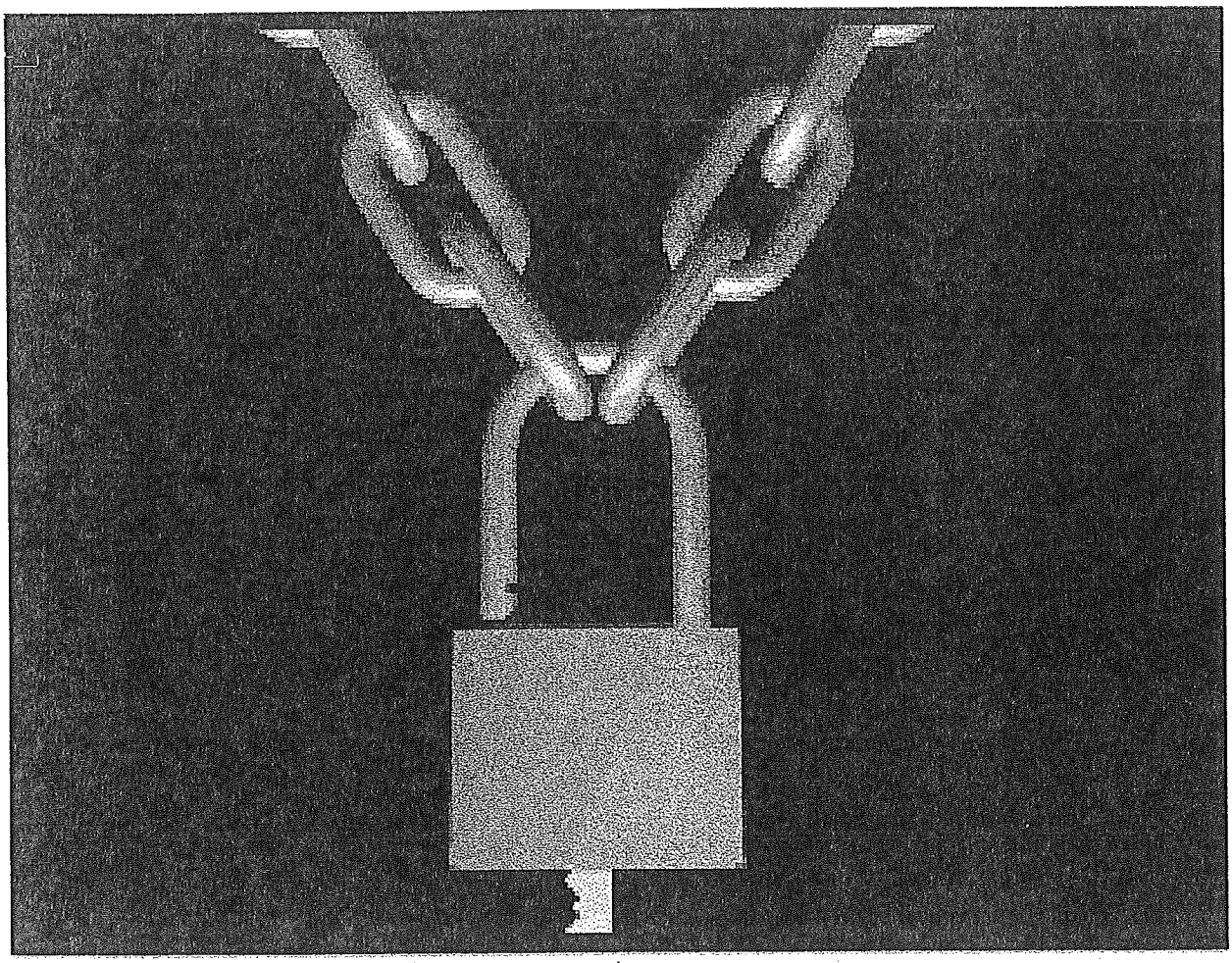
5.1 Program Flowchart




Click the CPU
yes
Start the CPU movie
no
Click on the RAM
yes
Start the RAM movie
no
Click the BIOS chip
yes
Start the BIOS movie
no
Click the clock chip
yes
Start the clock chip movie
no
Click IO connector
yes
Start the IO connector movie
no
Click the serial port
yes
Start the serial port movie
no
Click the parallel port
yes
Start the parallel port movie
no
Click the buses
yes
Start the buses movie
no

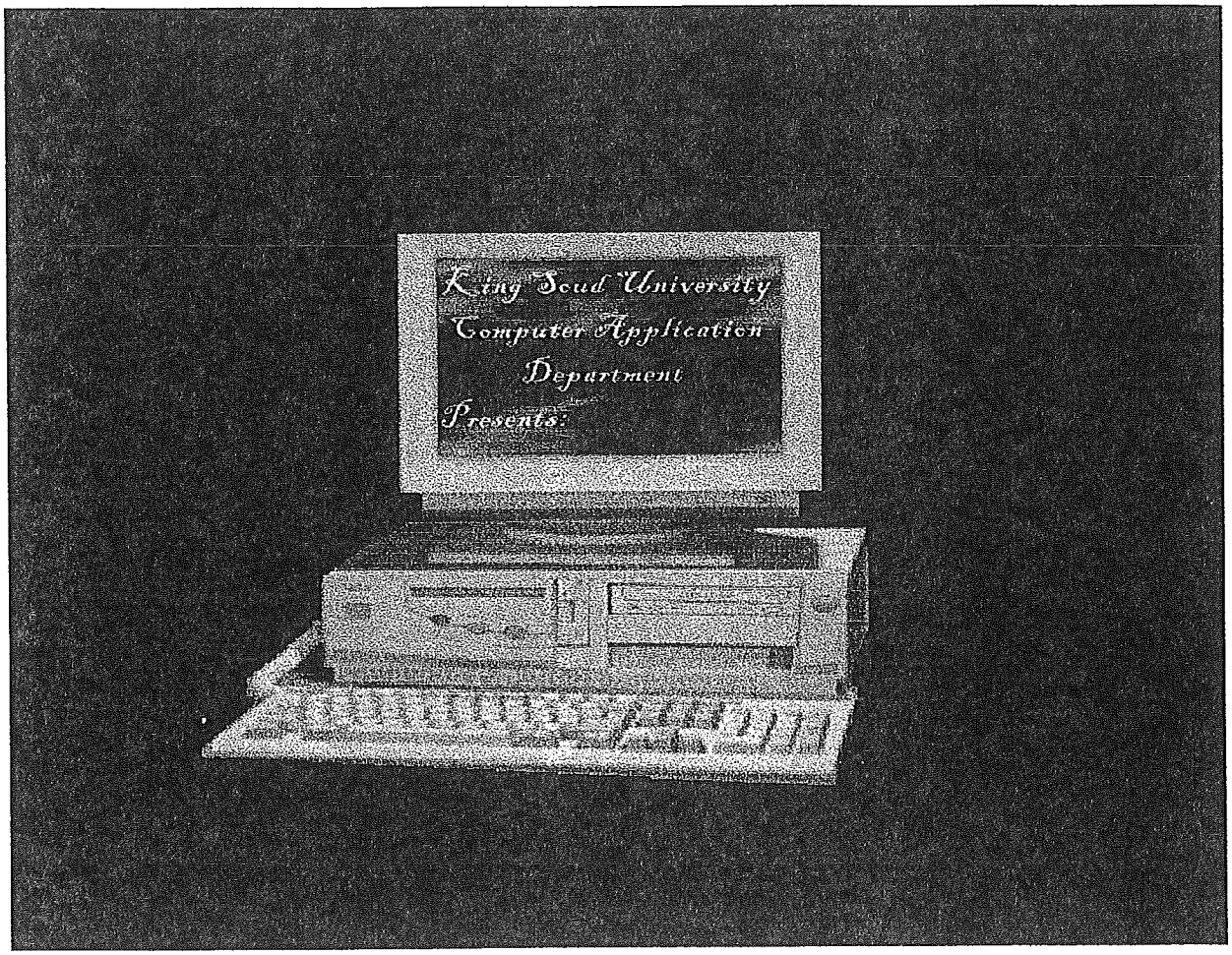
5.2 Storyboard

Storyboard Number	*p1*	Lesson Name	*Computer Environment*
		Author	*Abeer, Manal, Najwa, Reem*
		Date	*Friday, May 19, 2000*



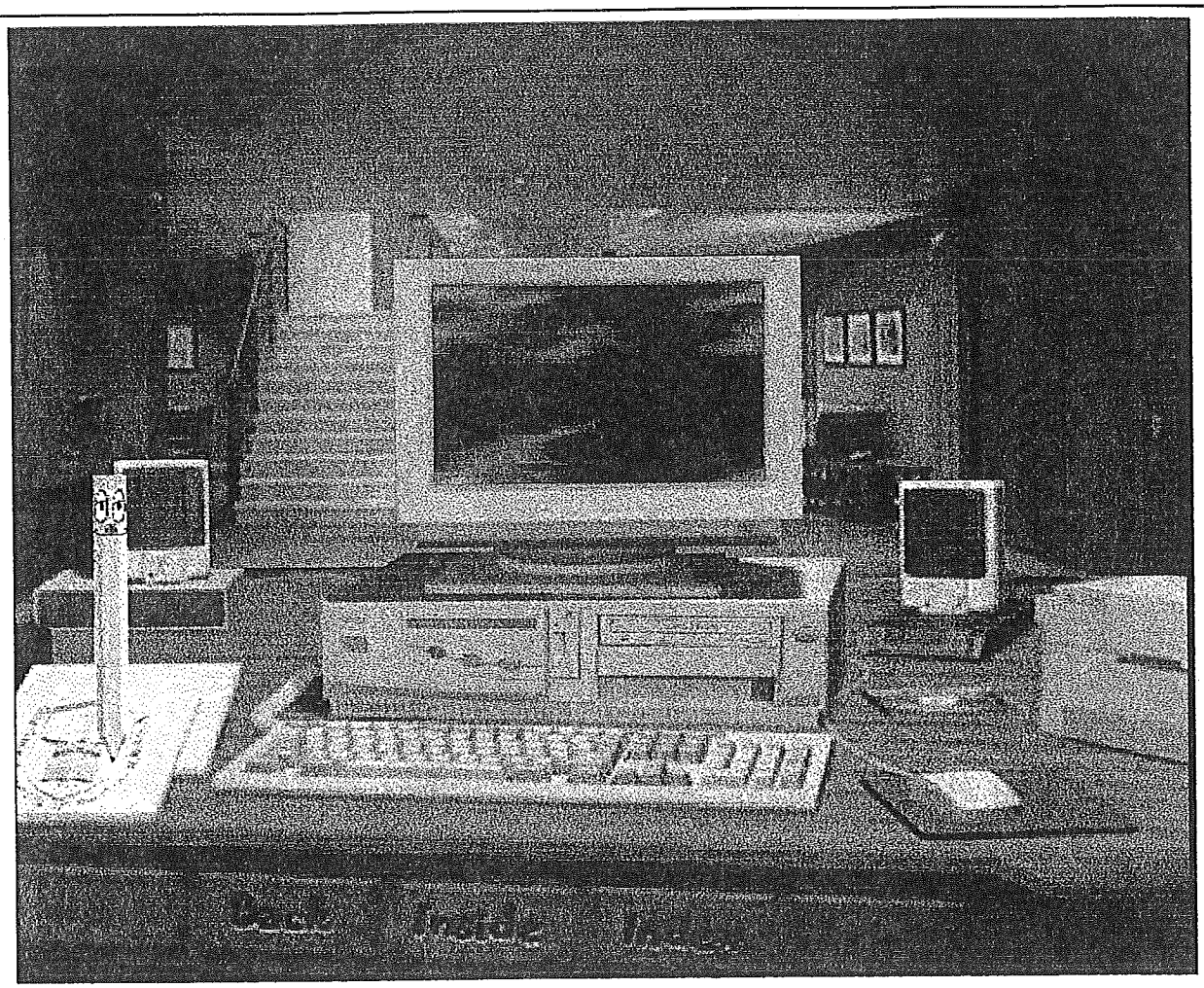
Comments

Storyboard Number	*p2*	Lesson Name	*Computer Environment*
		Author	*Abeer, Manal, Najwa, Reem*
		Date	*Friday, May 19, 2000*



Comments

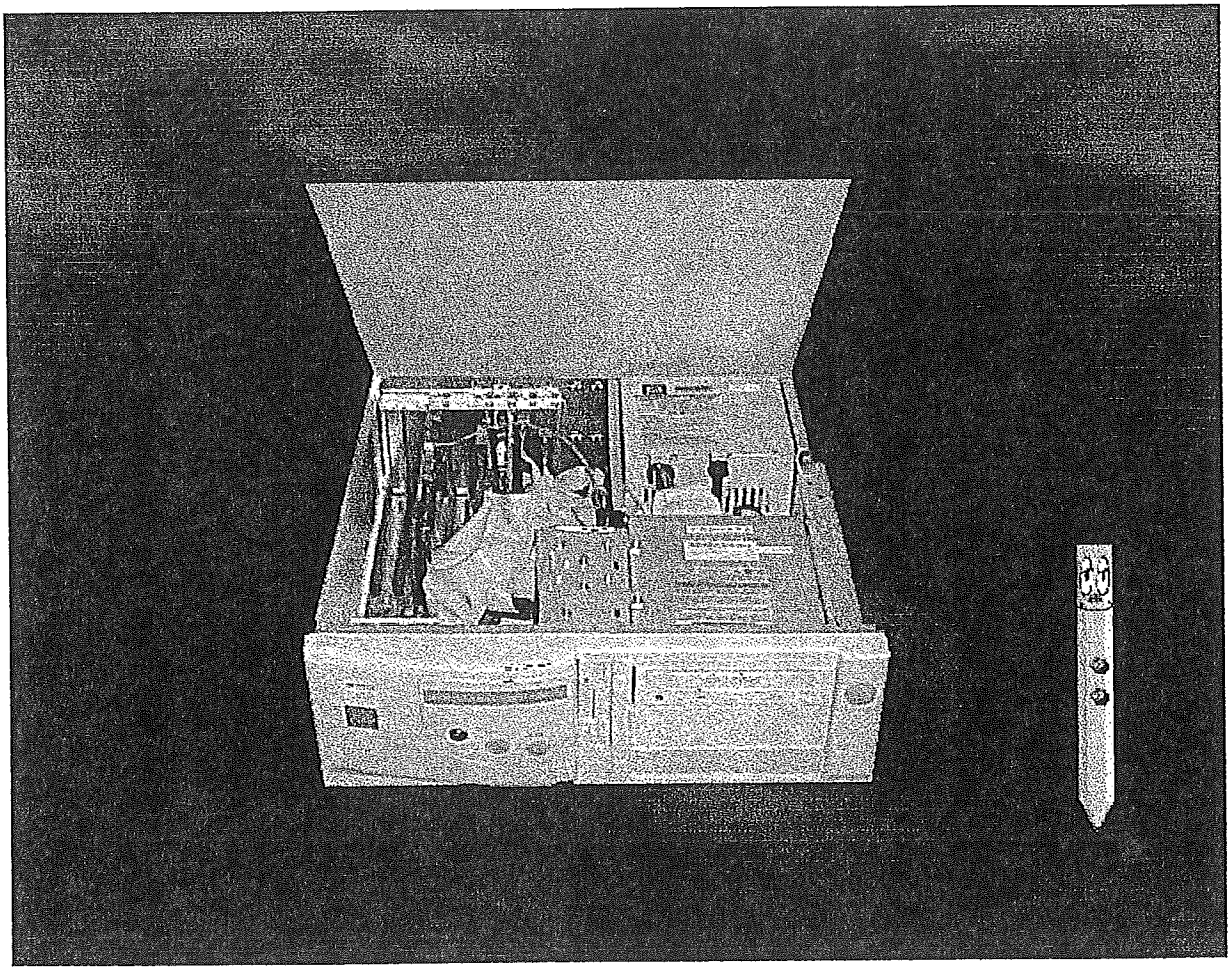
Storyboard Number	*p3*	Lesson Name	*Computer Environment*
		Author	*Abeer, Manal, Najwa, Reem*
		Date	*Friday, May 19, 2000*



Comments

Click Inside -> p4
Click the door -> p17
Click KSU Sign -> open About KSU file
Click Index -> p20

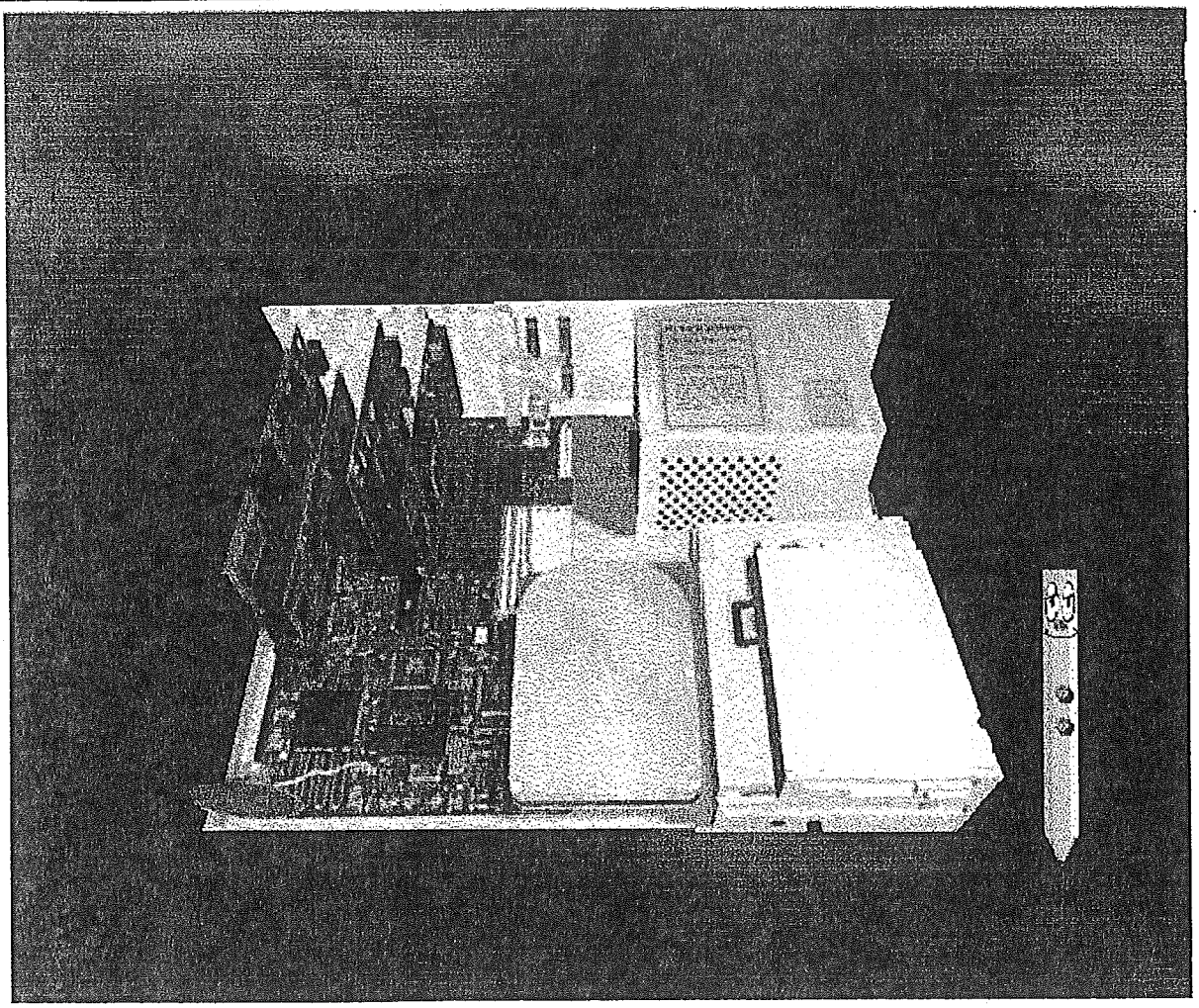
Storyboard Number	*p4*	Lesson Name	*Computer Environment*
		Author	*Abeer, Manal, Najwa, Reem*
		Date	*Friday, May 19, 2000*



Comments

Click the upper button -> p3
Click the lower button -> open help file

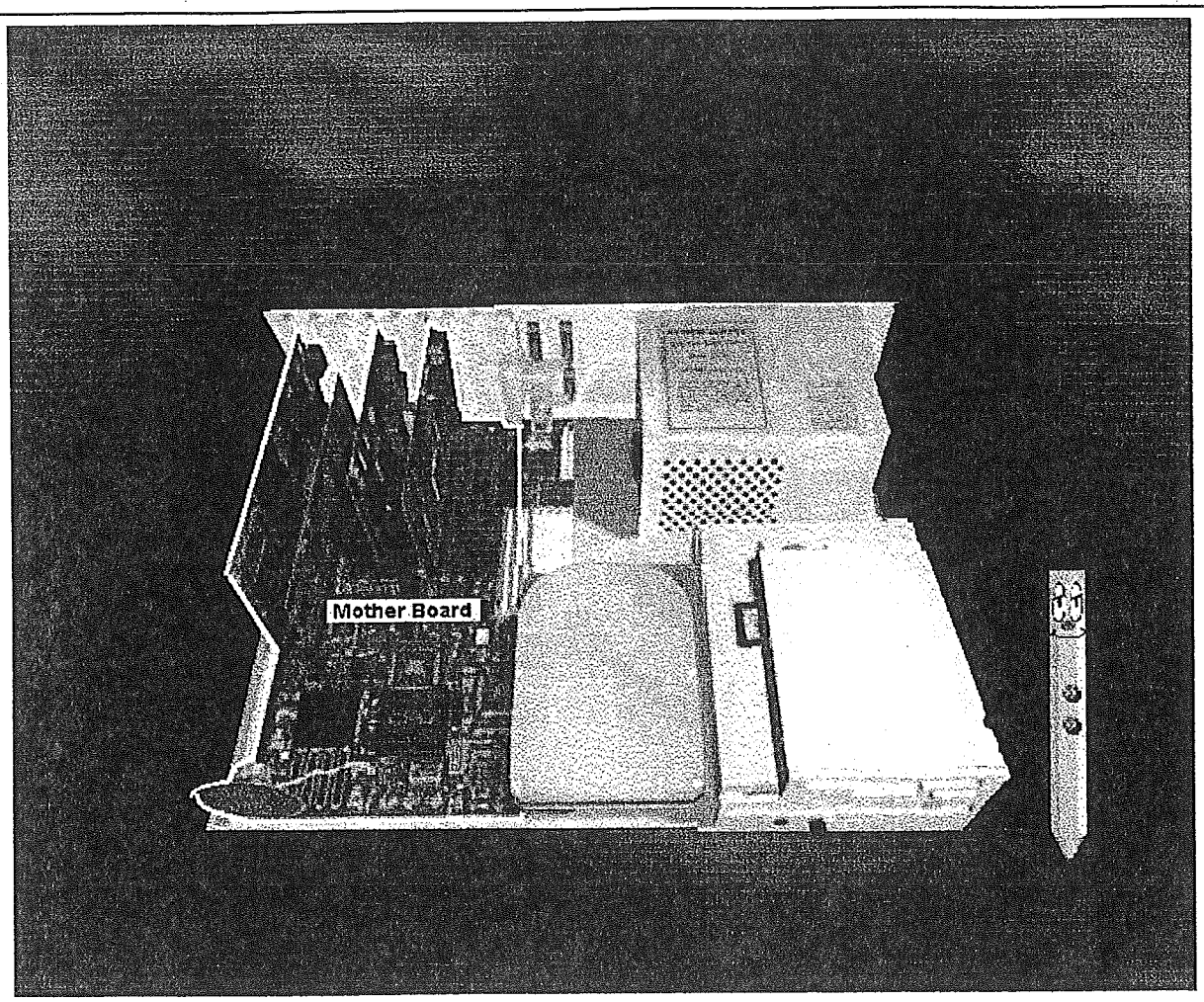
Storyboard Number	*p5*	Lesson Name	*Computer Environment*
		Author	*Abeer, Manal, Najwa, Reem*
		Date	*Friday, May 19, 2000*



Comments

Click the first button -> p3
Click the second button -> open help file
Click any computer component -> go to this component presentation

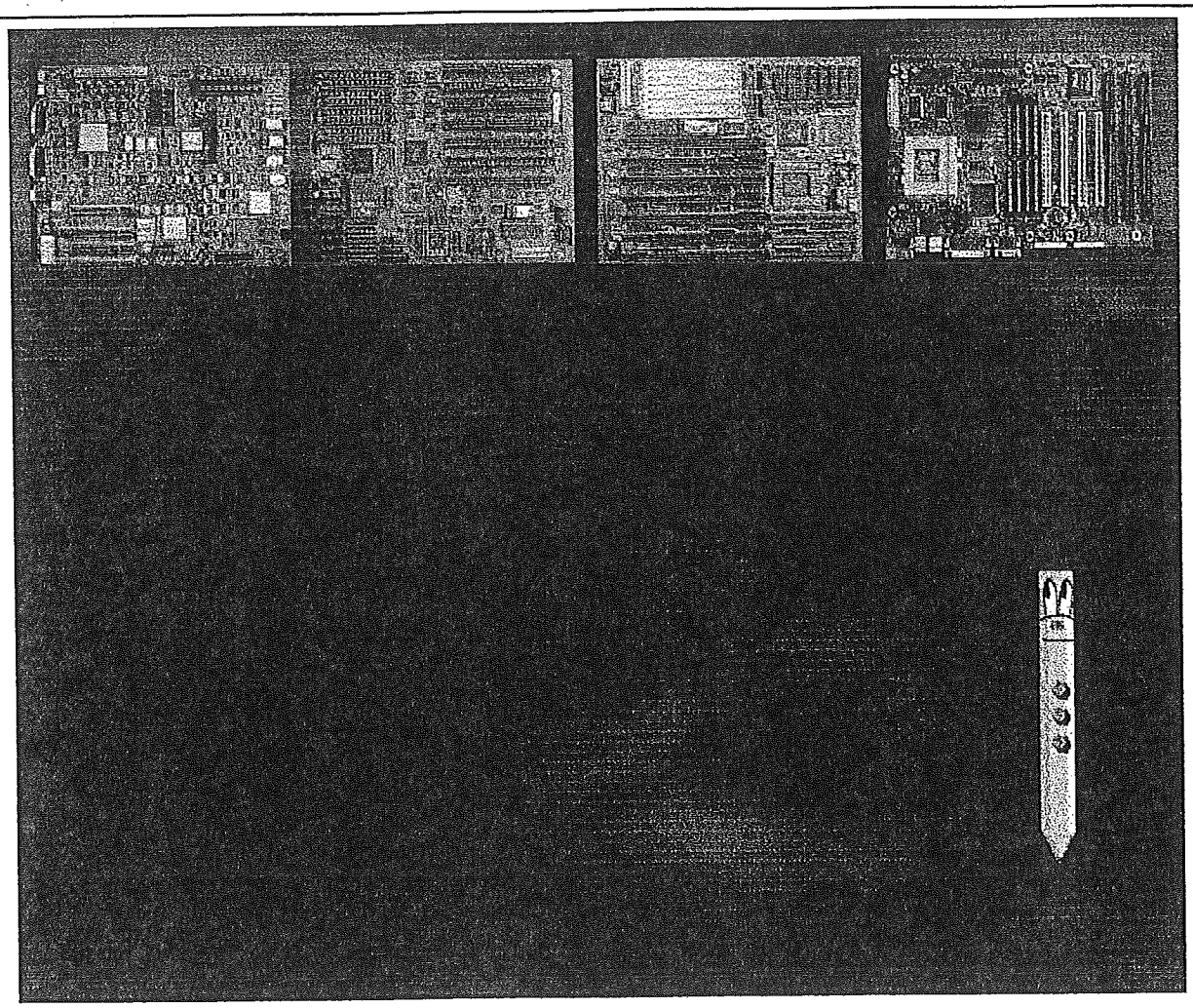
Storyboard Number	*p6*	Lesson Name	*Computer Environment*
		Author	*Abeer, Manal, Najwa, Reem*
		Date	*Friday, May 19, 2000*



Comments

Example:
Click on the motherboard -> p7
Click the first button -> p3
Click the second button -> open help file

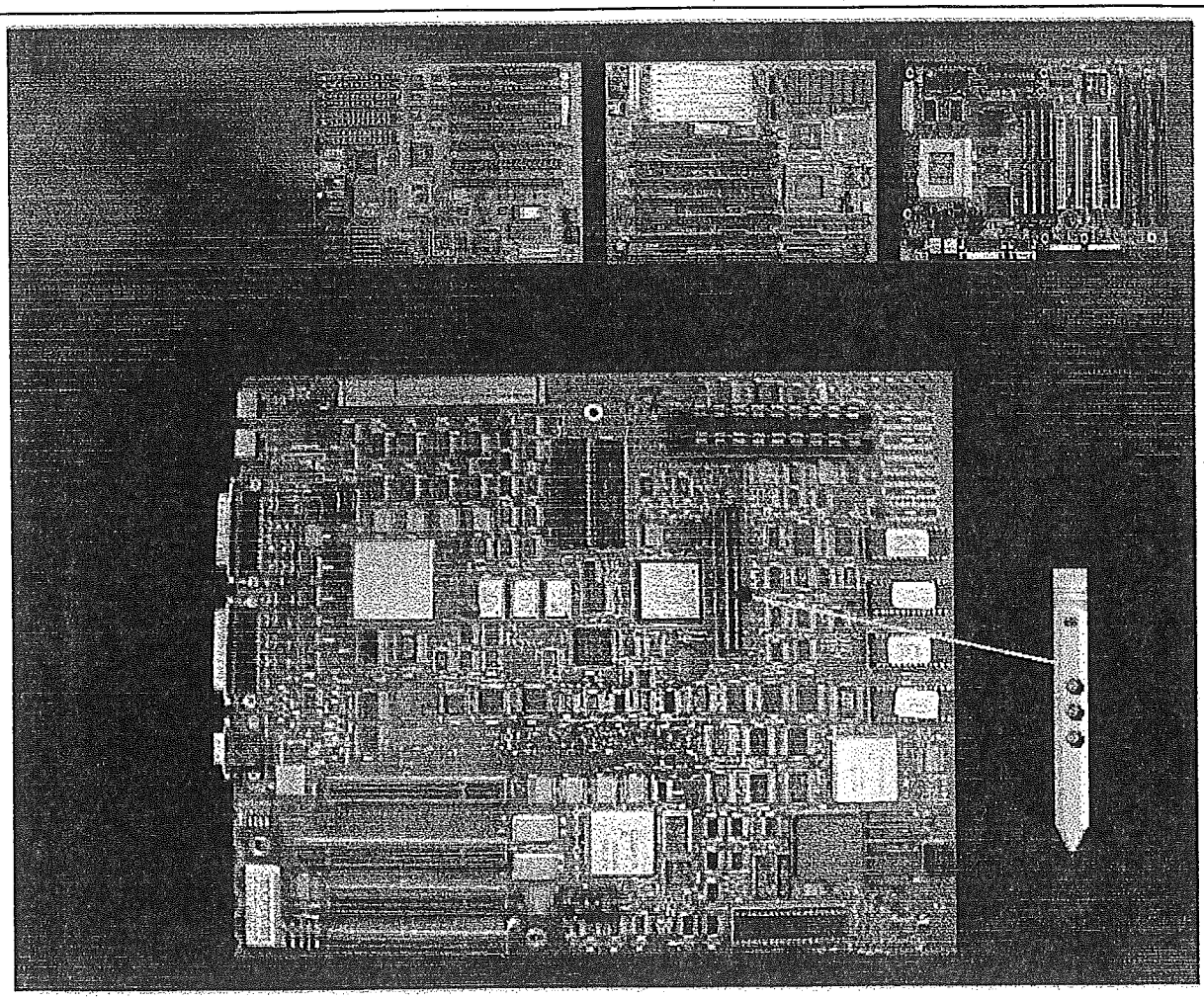
Storyboard Number	*p7*	Lesson Name	*Computer Environment*
		Author	*Abeer, Manal, Najwa, Reem*
		Date	*Friday, May 19, 2000*



Comments

Click the first button -> skip old motherboard's show
Click the second button -> p5
Click the third button -> open a help file

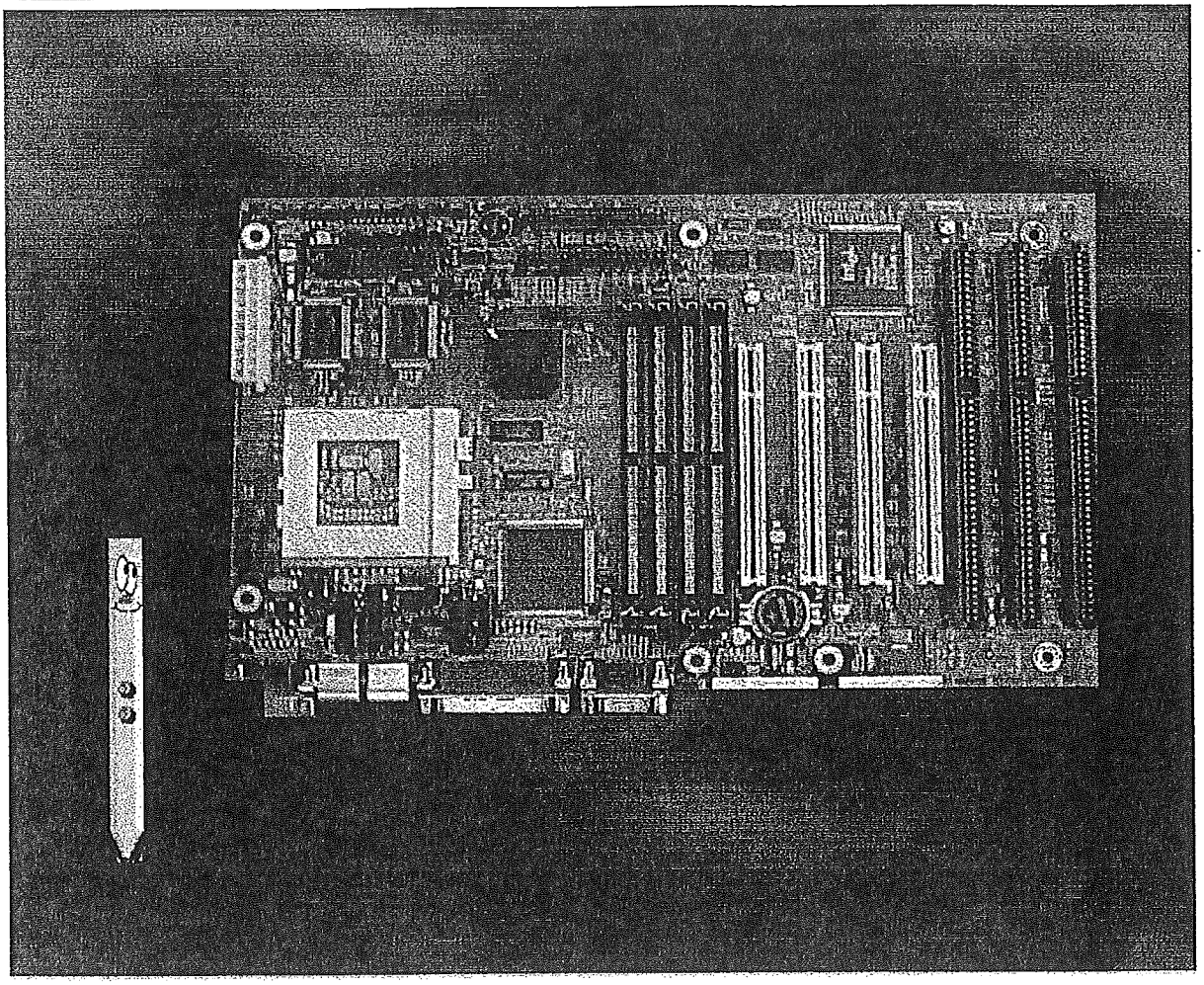
Storyboard Number	*p8*	Lesson Name	*Computer Environment*
		Author	*Abeer, Manal, Najwa, Reem*
		Date	*Friday, May 19, 2000*



Comments

Example:
Explain the 286 motherboard
Click the first button -> skip old motherboard's show
Click the second button -> p5
Click the third button -> open a help file

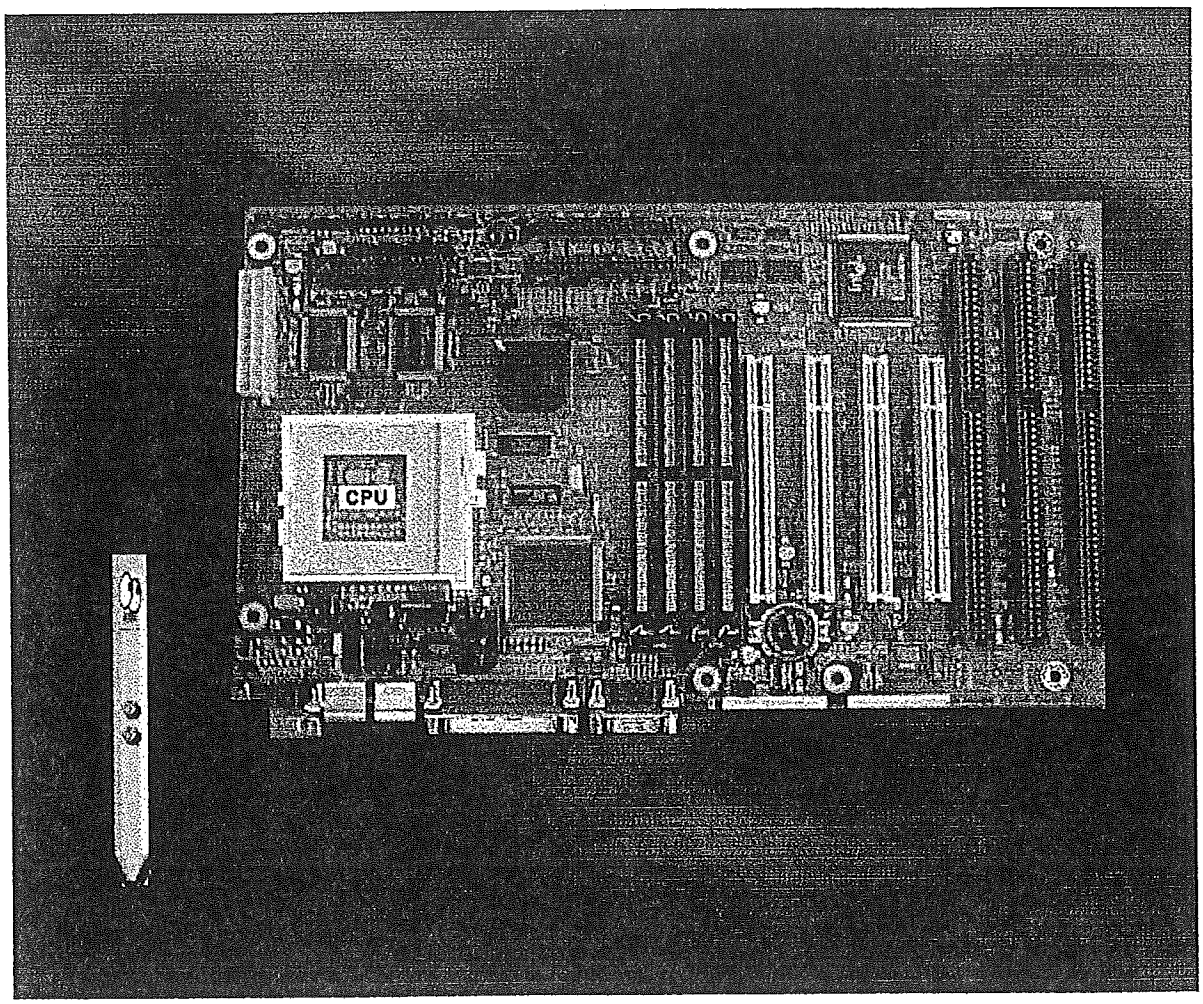
Storyboard Number	*p9*	Lesson Name	*Computer Environment*
		Author	*Abeer, Manal, Najwa, Reem*
		Date	*Friday, May 19, 2000*



Comments

Click any Pentium motherboard component -> jump to this component movie
Click the first button -> p5
Click the second button -> open a help file

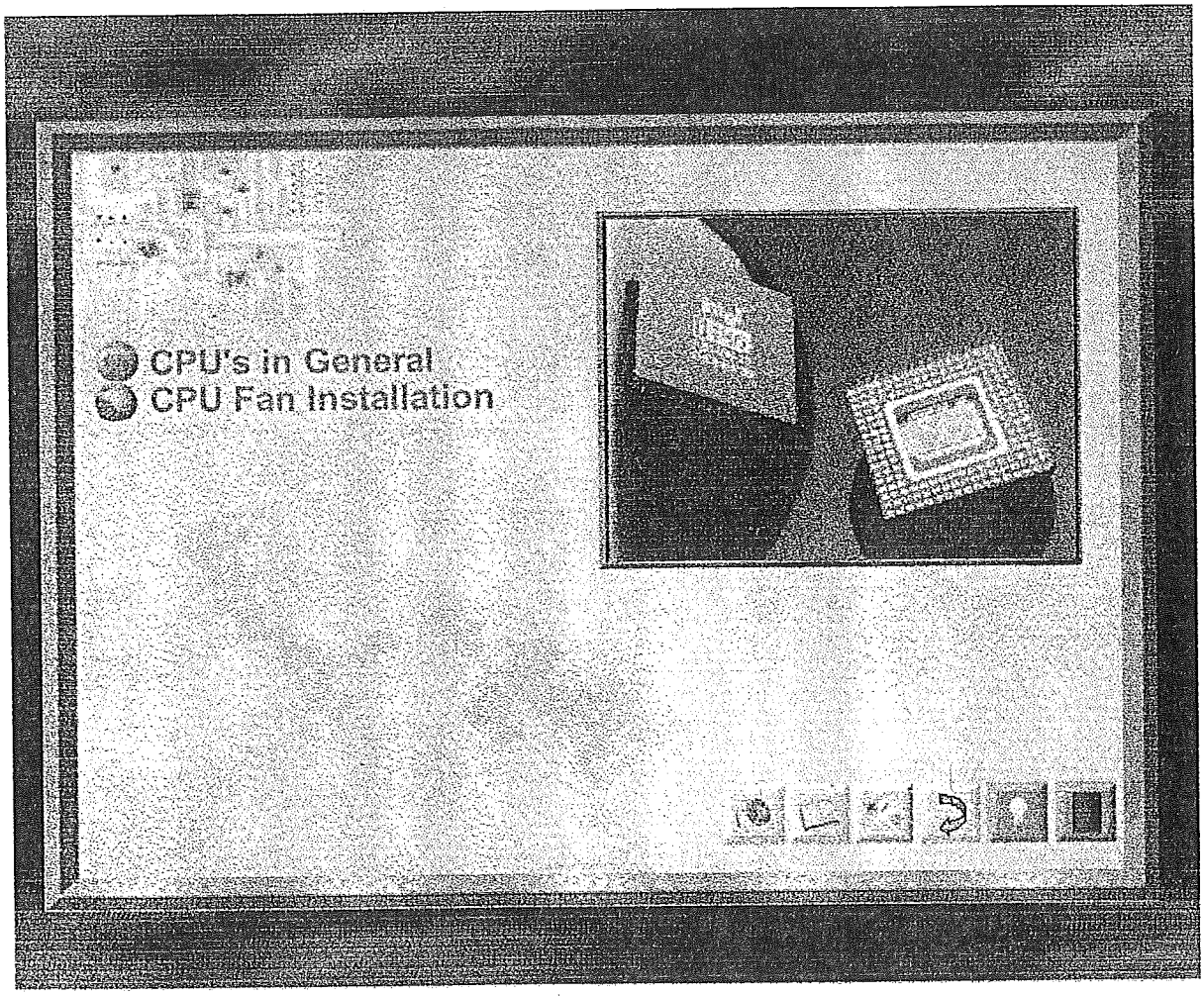
Storyboard Number	Lesson Name	*Computer Environment*
p10	Author	*Abeer, Manal, Najwa, Reem*
	Date	*Friday, May 19, 2000*



Comments

Click the CPU -> p11
Click the first button -> p5
Click the second button -> open a help file

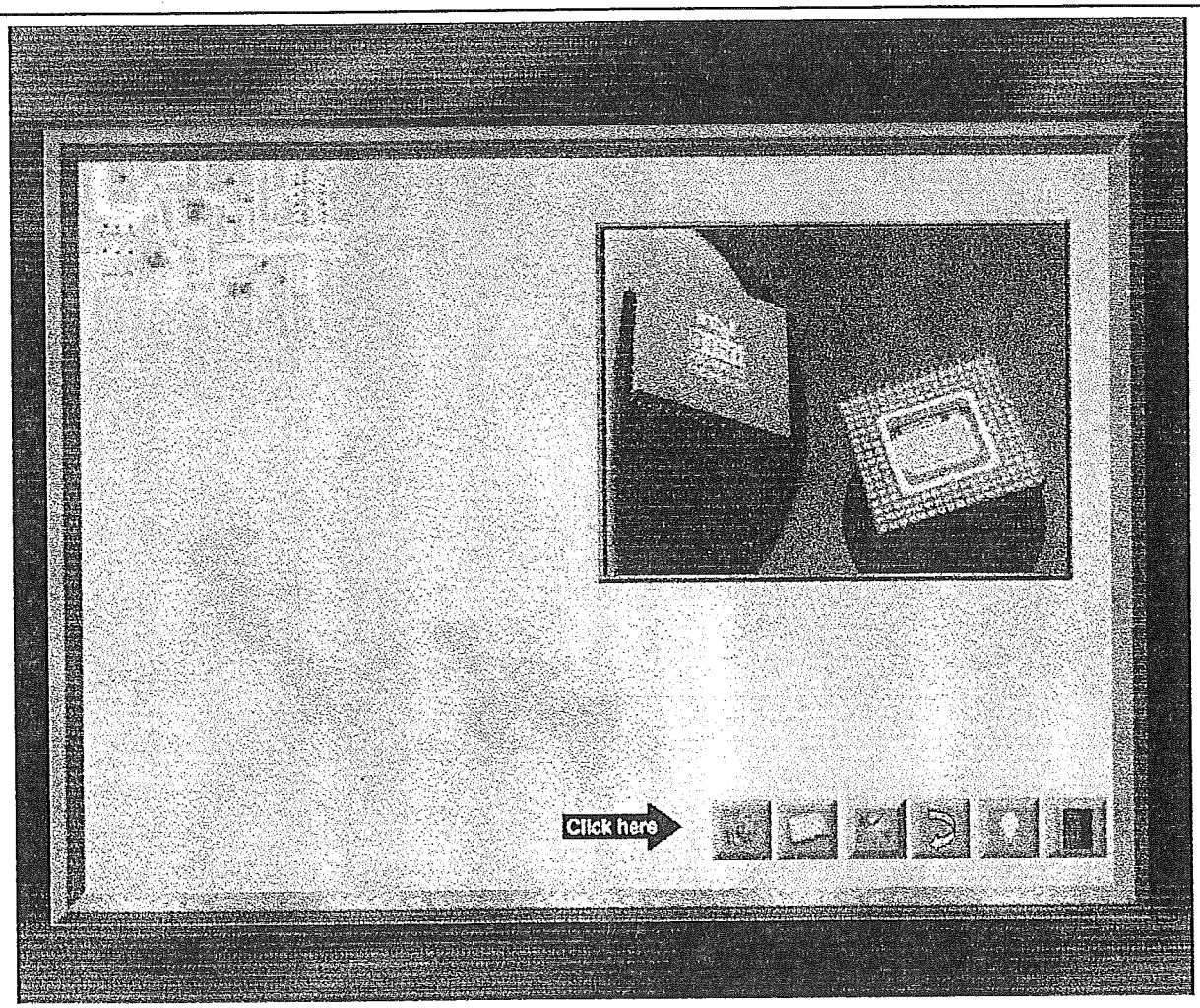
Storyboard Number	Lesson Name	*Computer Environment*
p11	Author	*Abeer, Manal, Najwa, Reem*
	Date	*Friday, May 19, 2000*



Comments

Click the first ball -> p12
Click the second ball -> p15
Click the exit button -> p9
Click the help button -> open a help file

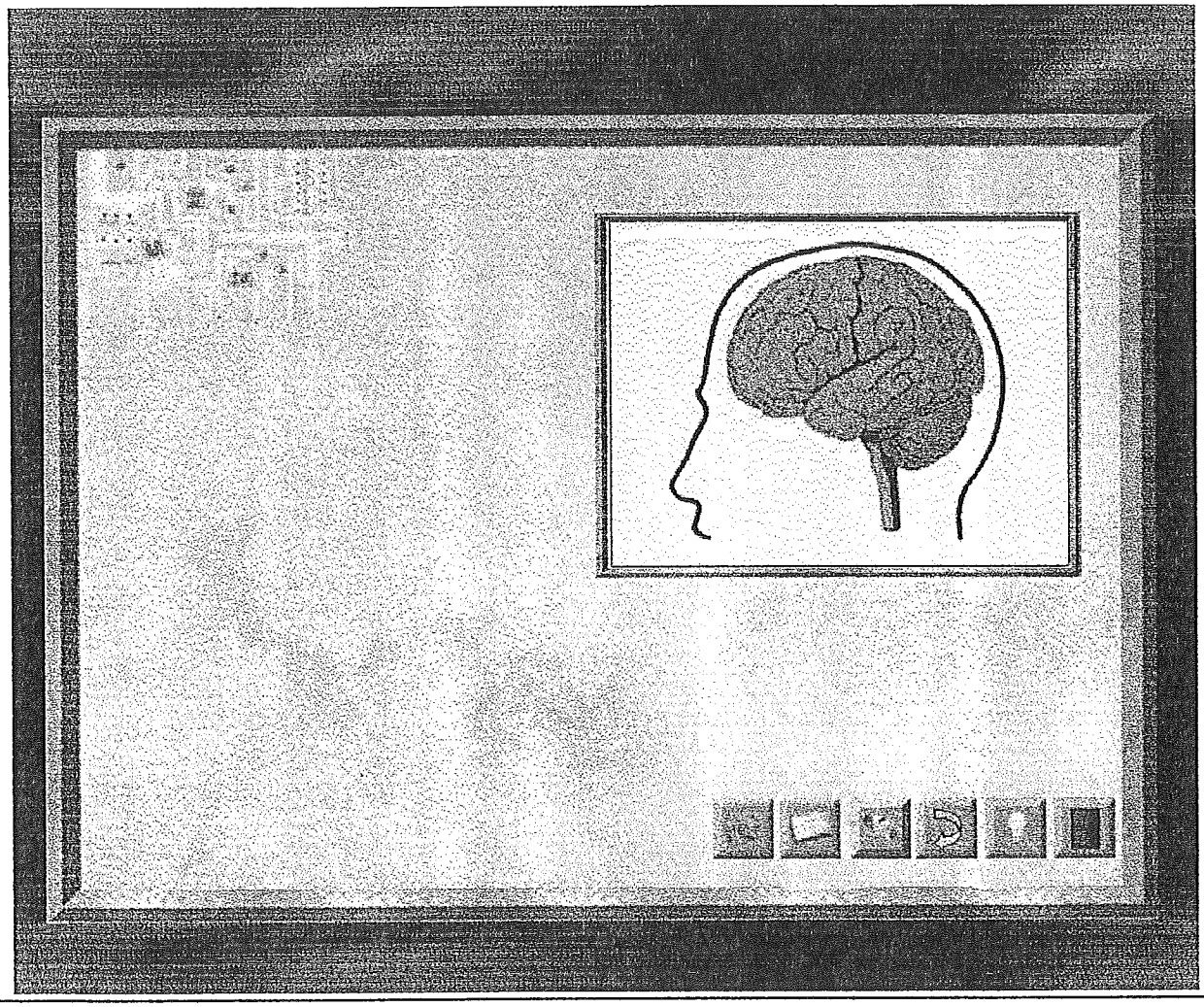
Storyboard Number	Lesson Name	*Computer Environment*
p12	Author	*Abeer, Manal, Najwa, Reem*
	Date	*Friday, May 19, 2000*



Comments

Click the play/Rewind button -> p13
Click the view text button -> p14
Click the pause button -> pause the presentation
Click the menu button -> p11
Click the exit button -> p9
Click the help button -> open a help file

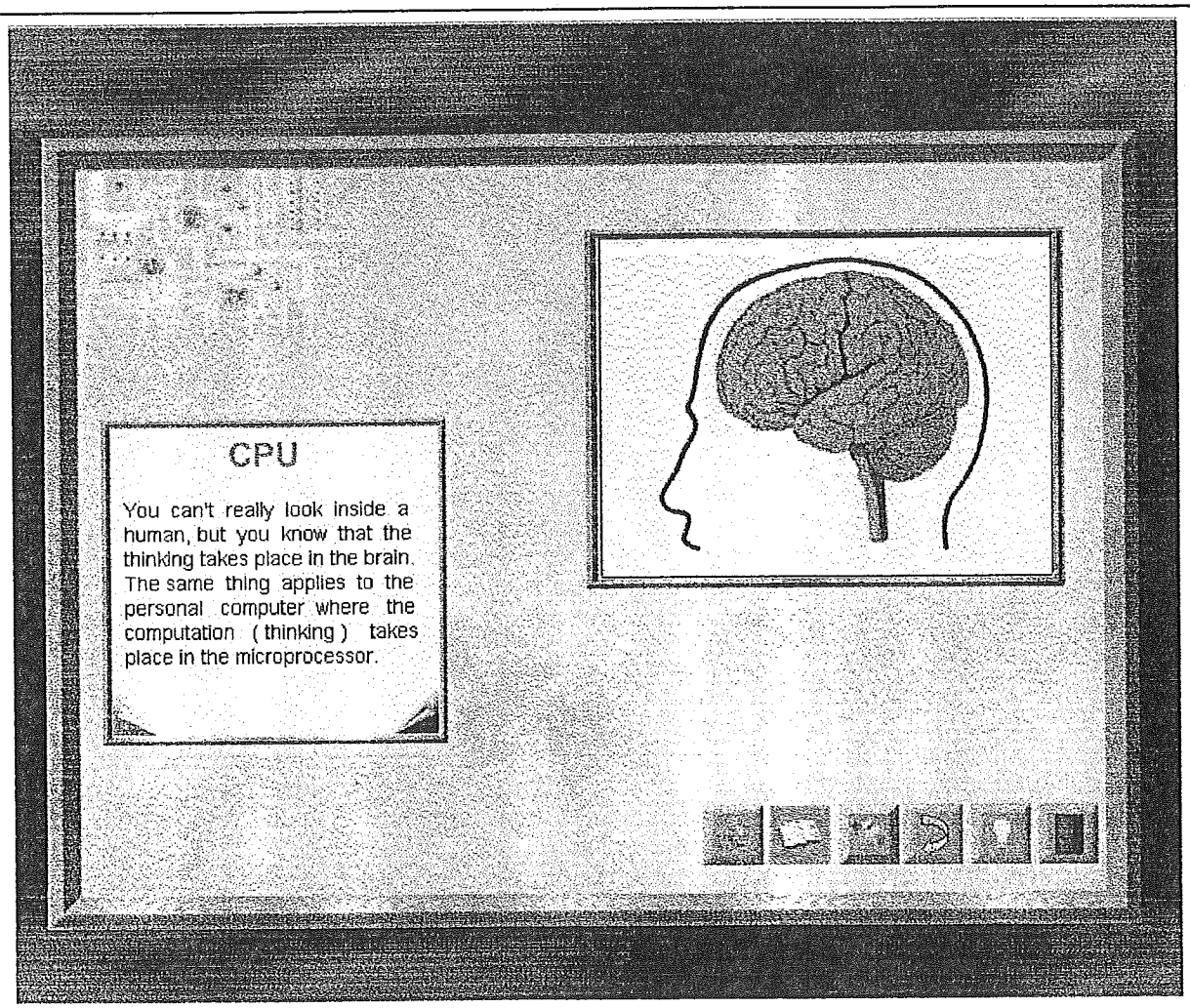
Storyboard Number	Lesson Name	*Computer Environment*
p13	Author	*Abeer, Manal, Najwa, Reem*
	Date	*Friday, May 19, 2000*



Comments

Click the play/Rewind button -> p13 (rewind)
Click the view text button -> p14
Click the pause button -> pause the presentation
Click the menu button -> p11
Click the exit button -> p9
Click the help button -> open a help file

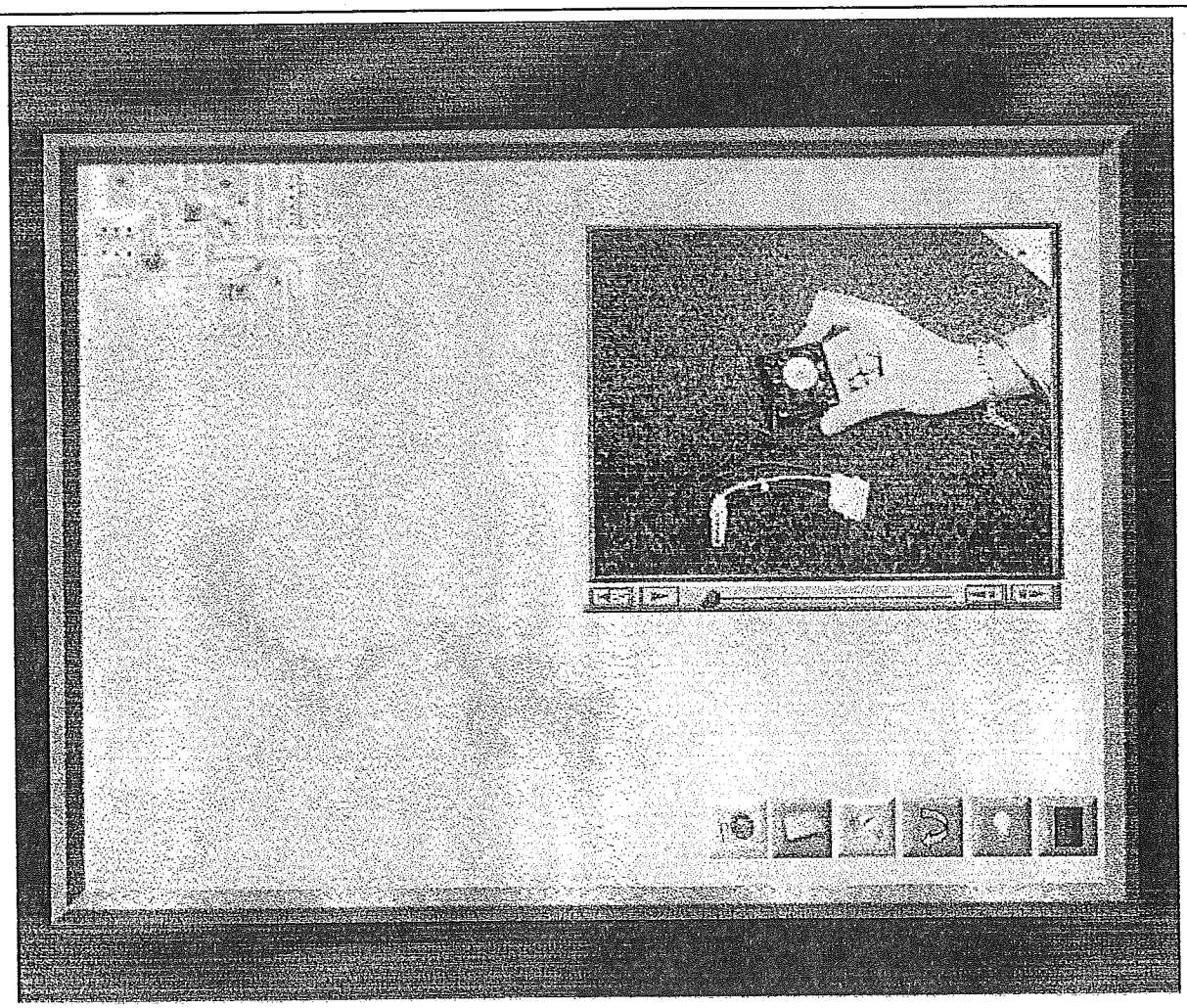
Storyboard Number	Lesson Name	*Computer Environment*
p14	Author	*Abeer, Manal, Najwa, Reem*
	Date	*Friday, May 19, 2000*



Comments

Click the play/Rewind button -> p13 (rewind)
Click the view text button -> p13
Click the pause button -> pause the presentation
Click the menu button -> p11
Click the exit button -> p9
Click the help button -> open a help file
Click the lower left part of the text page -> to the previous page
Click the lower right part of the text page -> to the next page

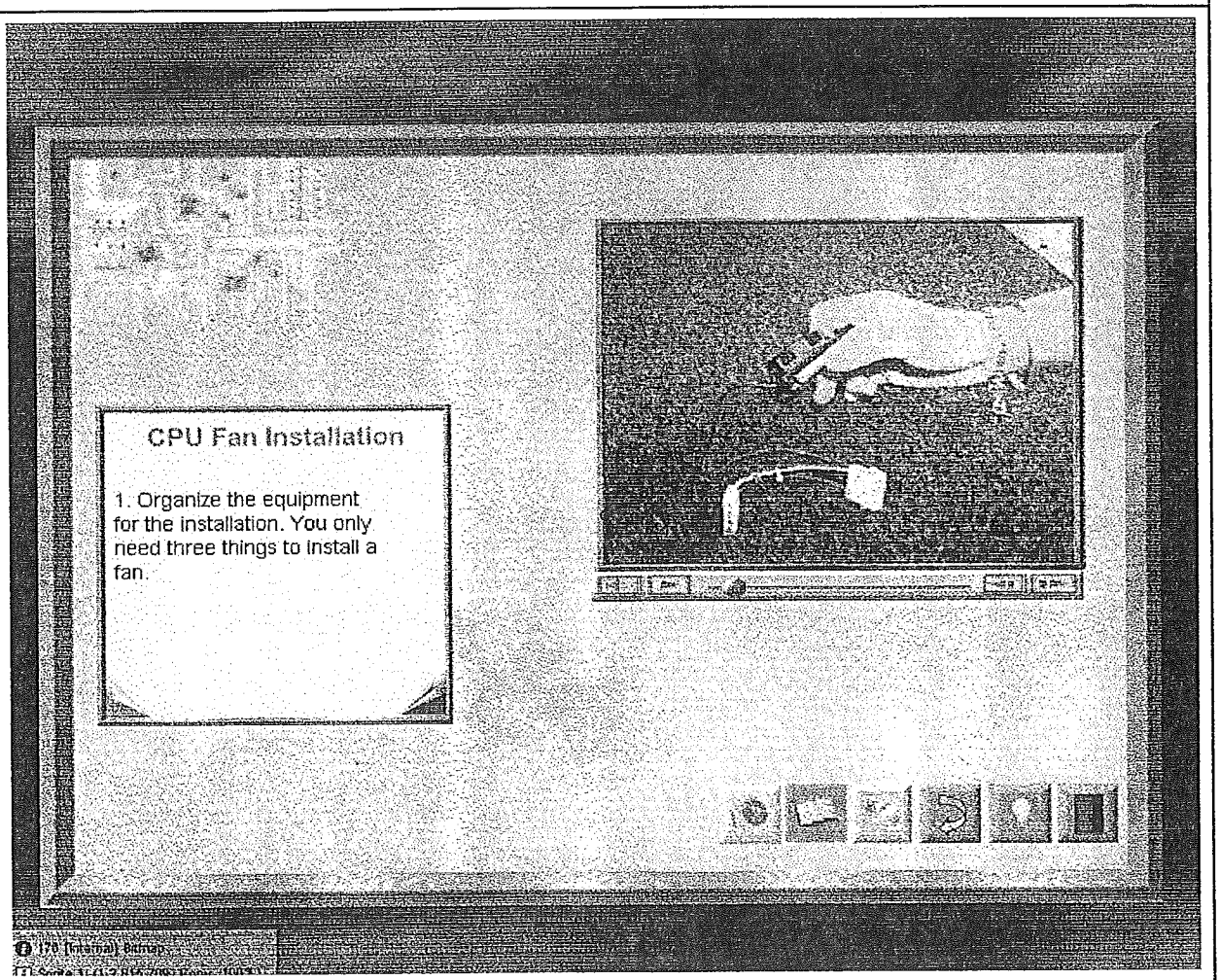
Storyboard Number	Lesson Name	*Computer Environment*
p15	Author	*Abeer, Manal, Najwa, Reem*
	Date	*Friday, May 19, 2000*



Comments

Video control slider and buttons -> control the video
Click the view text button -> p16
Click the menu button -> p11
Click the exit button -> p9
Click the help button -> open a help file

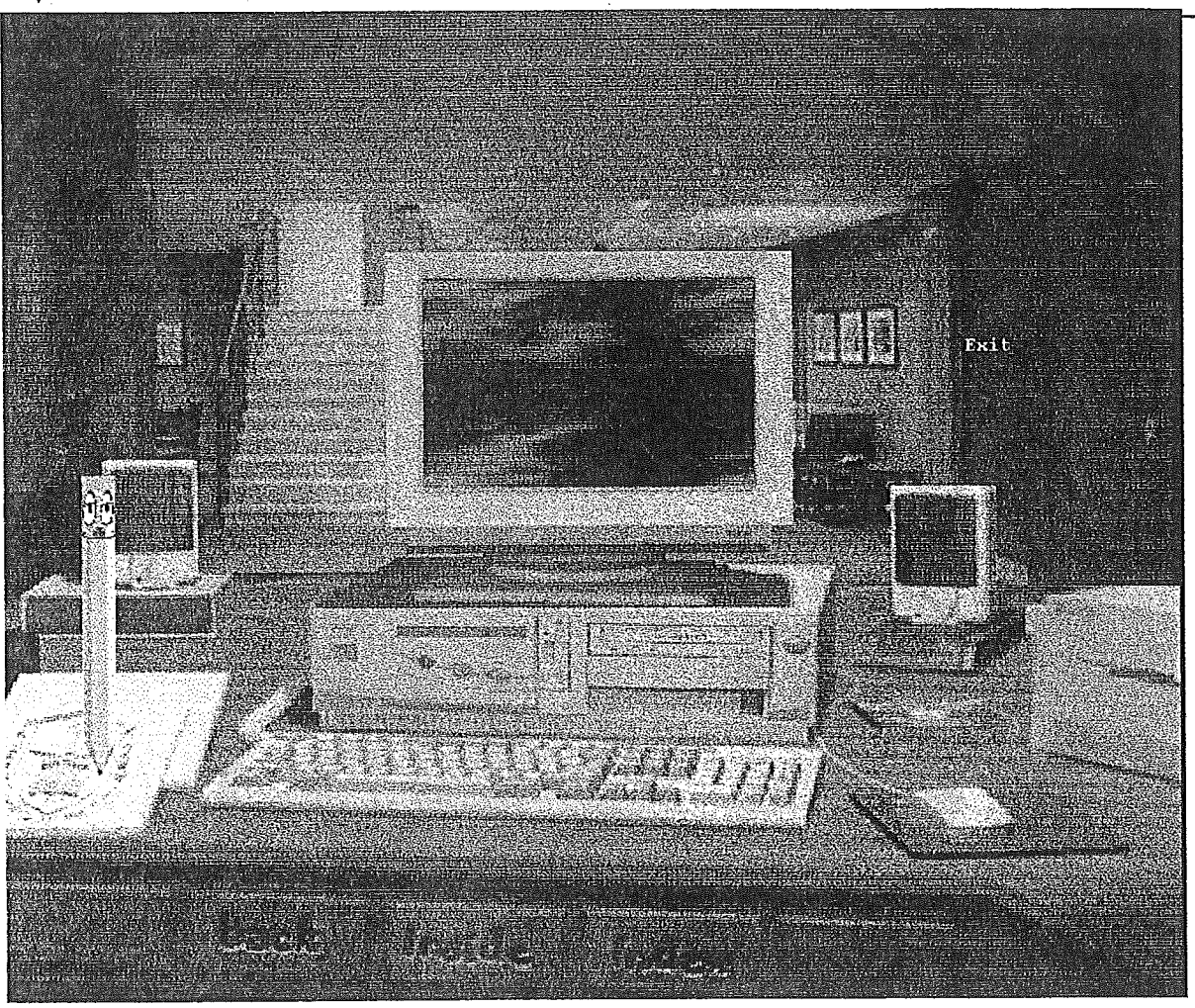
Storyboard Number	Lesson Name	*Computer Environment*
p16	Author	*Abeer, Manal, Najwa, Reem*
	Date	*Friday, May 19, 2000*



Comments

Video control slider and buttons -> control the video
Click the view text button -> p16
Click the menu button -> p11
Click the exit button -> p9
Click the help button -> open a help file
Click the lower left part of the text page -> to the previous page
Click the lower right part of the text page -> to the next page

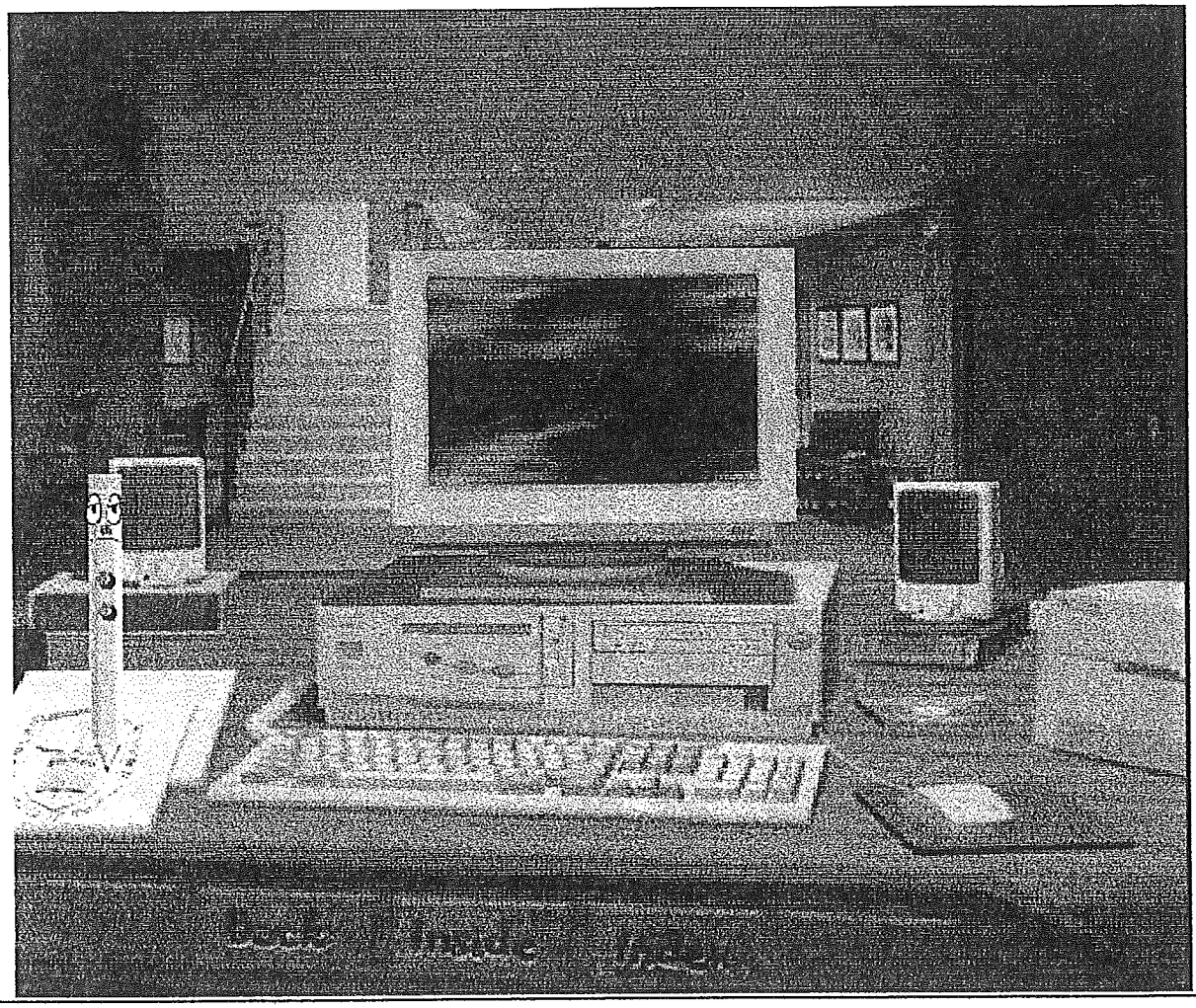
Storyboard Number	Lesson Name	*Computer Environment*
p17	Author	*Abeer, Manal, Najwa, Reem*
	Date	*Friday, May 19, 2000*



Comments

Click the door -> p18

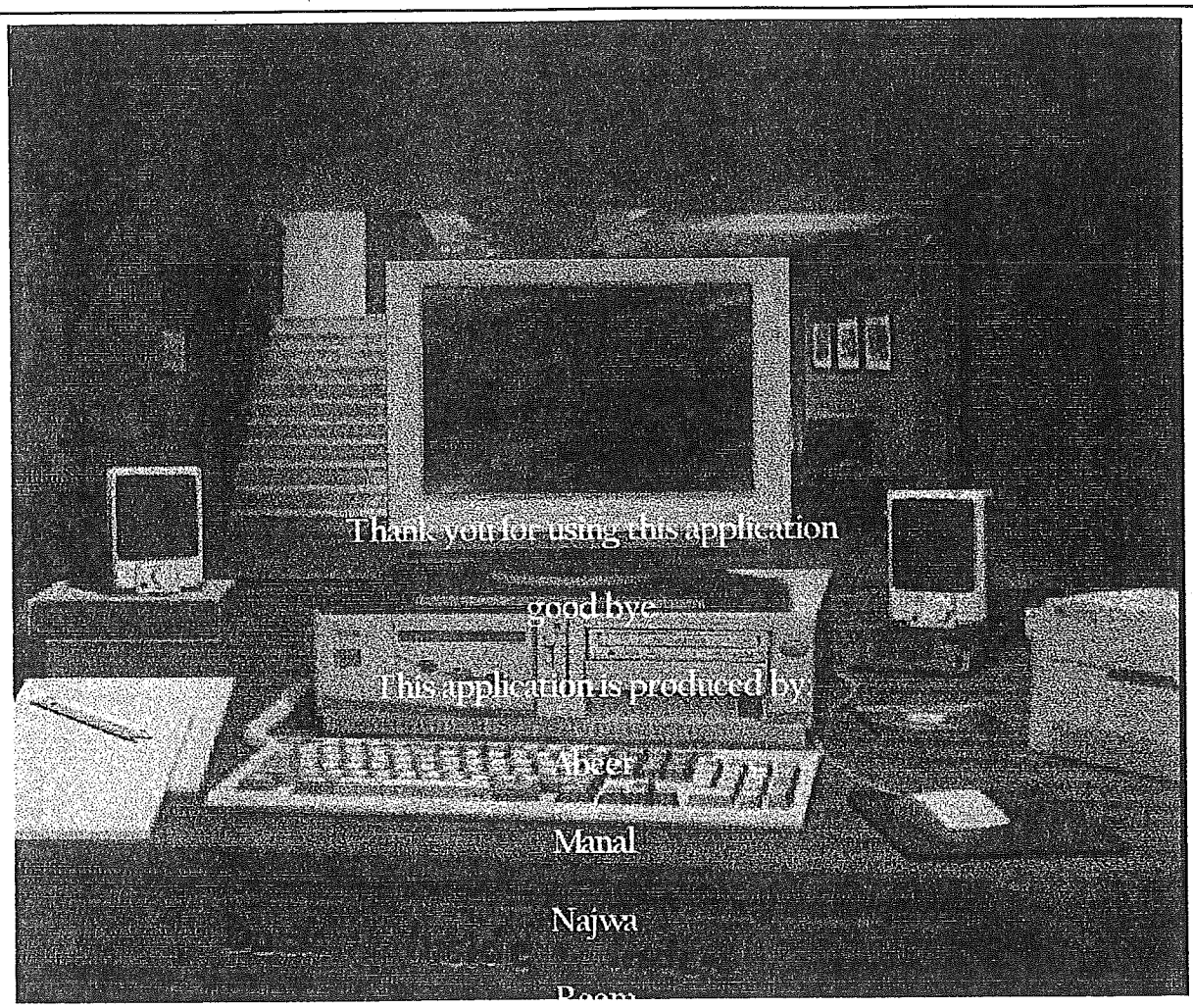
Storyboard Number	Lesson Name	*Computer Environment*
p18	Author	*Abeer, Manal, Najwa, Reem*
	Date	*Friday, May 19, 2000*



Comments

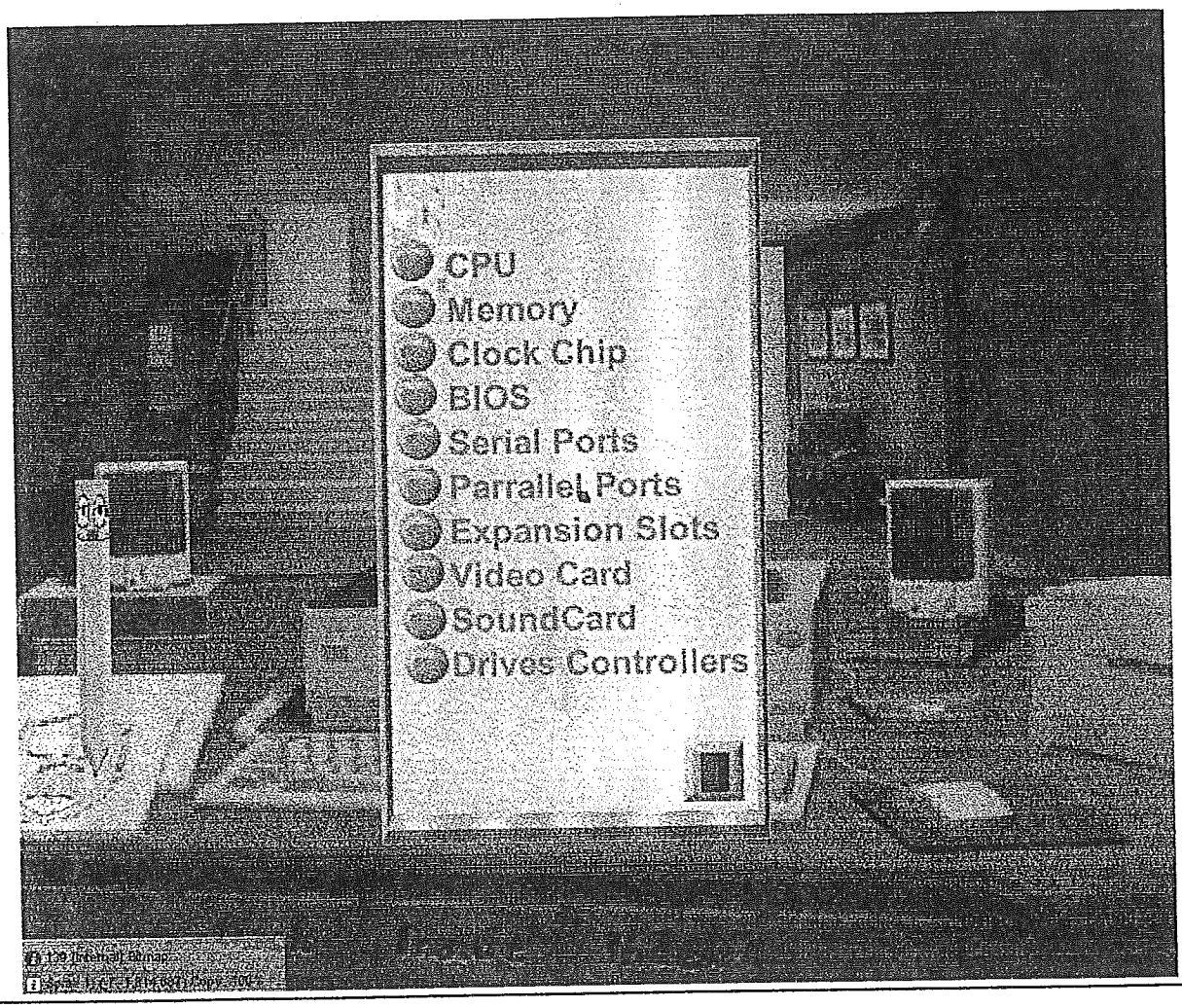
Click the first button -> p19
Click the second button -> p3

Storyboard Number	Lesson Name	*Computer Environment*
p19	Author	*Abeer, Manal, Najwa, Reem*
	Date	*Friday, May 19, 2000*



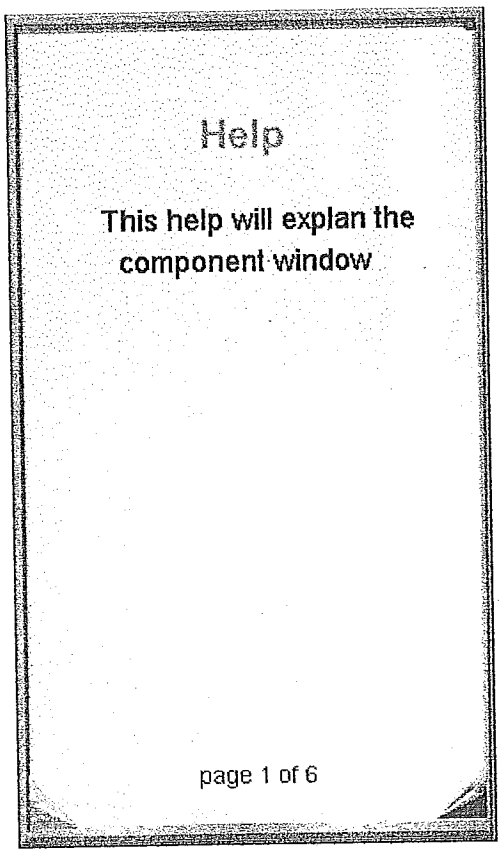
Comments

Storyboard Number	Lesson Name	*Computer Environment*
p20	Author	*Abeer, Manal, Najwa, Reem*
	Date	*Friday, May 19, 2000*



Comments
Click CPU ball -> jump to CPU movie
Click memory ball -> jump to memory movie
Click Clock Chip ball -> jump to Clock Chip movie
Click BIOS ball -> jump to BIOS movie
Click Serial Ports ball -> jump to Serial Ports movie
Click Parallel ports ball -> jump to Parallel ports movie
Click Expansion Slots ball -> jump to Expansion Slots movie
Click Video Card ball -> jump to Video Card movie
Click Sound Card ball -> jump to Sound Card movie
Click Drives' Controller's ball -> jump to Drives' Controller's movie
Click the exit button -> p3

Storyboard Number	Lesson Name	*Computer Environment*
p21	Author	*Abeer, Manal, Najwa, Reem*
	Date	*Friday, May 19, 2000*



Comments

Click the lower left part of the text page -> to the previous page
Click the lower right part of the text page -> to the next page

5.3 User Interface

A technology solution for any problem is really made up of two systems; the technology system, comprised of the computer and the software, and the human system, made up of the people who use it. The user interface is the link between the technology and the users who will interact with it.

In our tutorial, we take visual, cognitive, and physical considerations into account in the design of the graphical interface for the tutorial.

5.3.1 Cognitive Considerations

a) Limiting memory loads:

In any multimedia presentation, the user is allowed control over the flow of information. To select a specific path to follow, the user would either point and click an icon, a hypertext term describing the path the user wishes to follow, select from a menu or list, or use the mouse in a similar fashion. The other type of control would require the user to memorize some key-code; for instance, CTRL+N could move the user forward to the next page. This places a load on the memory, especially if there are many different codes. In our project, we limit the load on memory as much as possible by not forcing students to remember information. For example, if a student wishes to study the CPU part of the lesson, she only needs to point at the CPU on the motherboard and click that part instead of typing data in a data entry box. A descriptive label appears by each item the student is able to click.

b) Breaking-down decision-making:

In our presentation, decision steps are broken into manageable chunks. The student's first choice is selecting which part of the inside of the computer to study, by clicking on that part to select it. For instance, the student would click

on the motherboard. Then the student decides which element of the motherboard to study, for example, to study the CPU, the student would need to click it. Upon clicking a part, a new window opens, containing a title of that part; in this case, "CPU" would be the title. The user then has the option of selecting one of two topics: *General Information* concerning the CPU, or *CPU Installation*. For each topic there are a group of labeled buttons, which allow the student control over the presentation, such as a button that shows/hides textual information, or a button to play a video or animation.

c) Providing context:

We use context to help students understand and remember what is being displayed to them. We provide context by using titles on screens and windows, as in using "CPU" for a title of the CPU part of the lesson. All buttons are appropriately labeled to make them easily understandable, if a student points at the "Text" button, a "Text" label appears above the button. This label remains as long as the student holds the mouse pointer over the button. We also use menus to show what is available, for example, *General Information*, and *CPU Installation*.

d) Consistency:

We provide consistency to help students find information quickly and make decisions. For instance, all buttons that perform the same task look and act in the same way throughout the lesson. The way information is displayed in every window is consistent as well, the upper-left part of the window contains the title, and the video or animation is displayed on the right. All buttons remain at the bottom-right of the window at all times. The text window is always displayed on the left, if the "Text" button is depressed.

e) Forgiving:

Because people like to explore and try things out, our interface allows students to explore without doing damage. After choosing a topic, options are available that allow the student to cancel, go back, and undo previous actions. If a student selects the CPU part of the lesson, and then changes her mind and decides to study the BIOS part, the "Stop" button stops the lesson, and the "Exit" button takes the student back to the motherboard to select again.

5.3.2 Visual Considerations

There are two main visual considerations that need to be taken into account; one is the judicious use of color and highlighting, and the other is the use of visual coding.

a) Using color and highlighting judiciously:

Colors and boxes are very powerful ways to visually grab attention. Extra care must be taken not to use too many contrasting colors, as that would cause confusion and would no longer attract attention. In our project, we applied proper coloring conventions, and exercised caution when using colors. We used a minimum amount of colors, and kept it consistent throughout the tutorial. We also made use of highlighting, to show different parts of the motherboard.

b) Using visual coding:

Visual coding means using specific colors to signify different states. A well-known visual coding example is hypertext, where hypertext is recognized by its color, usually blue. Another example is often seen in menus, when unavailable options are grayed out. We used visual coding in our project by graying out buttons that are temporarily unavailable. After the student selects a part of the

motherboard, she must select whether to study *General Information* or watch a video. At this time, buttons that play or stop the animation are grayed out, and are made available after the student selects to watch a video.

5.3.3 Physical Considerations

a) Avoiding key combinations:

As mentioned previously, key combinations are difficult to remember, and hard to execute. The mouse on the other hand is intuitive and very easy to use. Making a selection using the mouse is as simple as point-and-click. For this reason, our work depends on the mouse rather than the keyboard, so there are no key combinations to remember, such as CTRL+N to go forward.

b) Avoiding a 50/50 split:

Some multimedia productions force users to use a keyboard 50% of the time, and the mouse 50% of the time. This can be both frustrating and demanding, as it would be necessary for the user to remember when to use the mouse, and when to use the keyboard. This also causes the user to continually take his eyes off the screen to look at the keyboard, and to move his hand back and forth from mouse to keyboard. As mentioned above, our project is designed to avoid these problems, by using a mouse all the time.

5.3.4 The Purpose of the Interface

Because students are using our project as a learning tool, they will need to have their attention caught and maintained. To achieve this, our project was designed with the following points in mind:

- **Maintaining student's interest:** This was accomplished by using graphics, color, and animation to make the information interesting, and to keep students from being distracted by the environment.
- **Using instructional design principles:** Information was broken down into meaningful bits.
- **Minimizing the teacher's need to verbally explain information:** This is achieved by using an option of a text box that contains the verbal explanation, which may be accessed at any time during the presentation.
- **Reducing the demands on the student** by minimizing the number of decisions the student needs to make. For instance, viewing the opening of the case is not optional.
- **Planning for flexibility:** The student may choose any part of the motherboard to start with by clicking on it. The component the mouse rolls over is highlighted as a visual cue to the student.

Chapter 6

Methodology & Implementation

Our project is a multimedia application that is intended to help teaching assistants teach the 341 CAP lab. It uses with the Pentium motherboard as a backdrop, and to explain the components that are in a PC. In addition, it lightly touches on the history of previous generation motherboards. To develop such an application, we needed a program capable of producing a multimedia package. Therefore a search of the available programs for developing multimedia productions was required. After conducting the search we narrowed our choice to using either Macromedia's Director® or Macromedia's Authorware®. These two programs are both capable of developing multimedia applications, however Director has more capabilities and provides better multimedia editing tools. Furthermore, Authorware is incapable of creating an animated presentation using simple 2D images. We therefore decided on the use of Director. We then conducted a search for the latest version of Director. We discovered that the latest release of Director is version 8. However, this version is not readily available in Riyadh. Since we are time constraint,

ordering this version of the program is not feasible to the large delivery time required. Therefore we used the latest version of Director that was obtainable (version 6.5).

6.1 Obtaining Required Materials

Acquiring the material for our application proved to be the most difficult and time-consuming step.

6.1.1 Pictures

These include pictures of the motherboard, the motherboard components, background, buttons, images to be used in animated sequences, and so on. Pictures used in multimedia applications are either custom made by the developer, or gathered from various sources. A good number of the pictures used in our project were collected from external sources, and we photographed the rest. When collecting these pictures, it was imperative that they seem as if they were all taken from a single source. We needed pictures that had approximately the same size and quality. A picture that was too small would require resizing, which would degrade the picture's quality. We started our search in the Internet, and visited many sites, such as:

http://www.microsoft.com/
http://www.sybex.com/
http://www.intel.com/
http://www.imimage.com/
http://www.symantec.com/

http://www.animationfactory.com/
http://encarta.msn.com/category/categorymedia.asp
http://home.hyperlink.net.au/~chart/whatpci.htm

Most of these sites concentrated on technical information, and have few pictures. The pictures that were present were mostly of poor quality, very small, or they did not describe the topic very well. There were many motherboard pictures, but none of the motherboard components. Nevertheless, we saved the suitable pictures that we found for future use.

We then searched for suitable pictures in books, magazines, and CD's. Almost all of the pictures we found in books and magazines were blurred, or the photograph was taken at an angle that obscured the details that we wanted to explain. Many of the pictures we found on CD's were suitable, but they were not quite enough. We therefore decided to take our own pictures, using an Olympus 35 mm auto focus camera. But after scanning these pictures – using a Microtek SlimScan C3 scanner – we found that the resulting clarity was not good. Placing the motherboard directly on the scanner, and then scanning did not achieve the desired result. We then obtained a digital camera, but the pictures we took using it were of poor quality. We then used a Sony MVC-FD71 digital still camera, which gave us pictures that were of sufficient quality. We were able to photograph the different generations of motherboards, with all components showing clearly. However, when we began implementing the project, we discovered that not only do we need high-quality pictures of motherboard components, but also these pictures should be taken at a specific angle that would allow us to use them in an animation sequence. Unfortunately, since we did not have a tripod, it was very difficult to take pictures of all the components with the same perspective.

After collecting the images, we started the process of editing all the pictures, altering their coloring and shading slightly, so that they had a similar appearance. For a listing of the programs we used in image editing please refer to section 6.3 - Implementation.

Upon completing the editing process, we imported the pictures into Director. Director supports a Background Transparent option that may be used with images placed on the Stage. It makes the backcolor of the image transparent. We soon realized that further editing of the pictures was required, because although we set the pictures' ink to Background Transparent, a white "halo" surrounded the picture. We had to remove these extra white pixels in Director's own Paint Window. This lengthy process, although not difficult, had to be repeated many times – as many as 4 to 5 times - before we finally obtained desirable results.

This process is very time consuming so some suggested solutions to the halo problem are:

- Using the rubylith feature, if the program supports it. The rubylith feature uses a red mask to control areas that are painted, which allows the producer to view the areas that are "near-misses" in the image. This option is the best, but unfortunately we were unable to use it, as the programs we used to edit the pictures did not have it.
- Reducing the image size by one pixel. This makes the image appear jagged.
- Using special Xtras to assign a transparency level to the graphic, to create a smooth edge. Third-party developers make this Xtra, and we were unable to acquire it.

We discovered a severe limitation in Director. Director is only capable of 2D animation. We therefore need another program to develop 3D programs. Even so, any 3D animation would have to be imported as a video into Director. For a listing of these programs, please refer to the section 6.3 - Implementation.

Following are some 3D animation programs that we obtained and attempted to use to create 3D effects, but were unable to, with the reasons stated by each:

- *3D Studio Max:* A very complex program to create stunning 3D effects and animation, but we were unable to obtain the manual. 3D objects have to be created within the program.
- *Bryce 2:* Creates 3D landscapes only.
- *Simply 3D:* Does not support an Import feature, and does not import 2D images.
- *Canoma 3D Modeling:* Cannot import 2D images.
- *3D Photo:* Runs under DOS. We were unable to acquire a licensed copy of this program, and the trial edition did not perform any useful functions.
- *Macromedia's Flash 3:* Creates animated 3D web banners.
- *Corel Motion 3D 6:* Creates 3D Logos.

6.1.2 Sounds

Sounds in multimedia presentations include voice, music and special effects. Our project features speech accompanying the animated explanation of each motherboard component. We recorded the sound files using Microsoft Windows Sound Recorder, in WAV file format. To reduce file size, we recorded sounds in 11.025KHz format, although this degraded sound quality to a certain extent. Our project features an option of a text box, which displays the

speech on the screen in a window. This option is useful if the user does not have a sound card installed on his PC. The user can use the "Text" button to hide/show the text window.

Component	File Size	Length (seconds)
286.avi	288KB	7.427sec
386.avi	274KB	7.00sec
486.avi	263KB	7.00sec
Hello.avi	90.5KB	2.372sec
Leave.avi	78.4KB	2.00sec
Cpu.avi	1.51MB	1min 3.609 sec
Clockchip.avi	1.17MB	47.838sec
BIOS.avi	874KB	32.326sec
Ide.avi	876KB	32436sec
Memory.avi	1.55MB	1 min 5.789sec
Pentium.avi	166KB	5.866sec
Serial.avi	1.51 MB	1 min 3.609sec
Sound.avi	1.51MB	I min 3.609sec
Video.avi	2.18MB	1 min 35.413sec
Bus.avi	2.87MB	2 min 8.394sec
Classic.avi	2.74MB	56.100sec
Loop.avi	520KB	15.903sec
Explore.avi	106KB	3.121sec
Loop.wav	520KB	15.903sec
Door.wav	11.1KB	0.491sec

Table 6-1 *Some Sound Files*

6.1.3 Video

Our project contains videos to demonstrate the installation process of some of the motherboard components. These components are the CPU Fan, the IDE Adaptor Card, an additional Interface Card, and Extending RAM Memory. We obtained these videos from various CDs:

Component	File Size	Length (seconds)	Screen Size
CPUfan.avi	10.5MB	42.800sec	320 x 240
Additonalcard.avi	17.8MB	1 min 15.400 sec	320 x 240
IDEcard.avi	23.4MB	1 min 37.800 sec	320 x 240
ExtendRAM.avi	17.2MB	1 min 10.0 sec	320 x 240

Table 6-2 *Video Files*

6.1.4 Text

Textual information in our project is identical to the speech recordings. We were already familiar with the motherboard components, and used the following references in writing the text:

- Brooks, Charles, *"A+ Certification Concepts and Practice"*, Second Edition, Que E & T, 1999.

- Brooks, Charles, *"A+ Certification Concepts and Practice – Lab Guide"*, Second Edition, Que E & T, 1999.

6.2 Picture Samples

6.2.1 The White Halo Problem

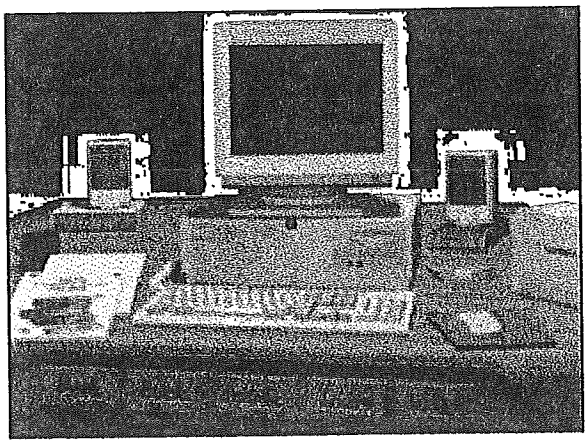


Figure 6-1 *The white halo surrounding the picture after applying the Background Transparent ink effect*

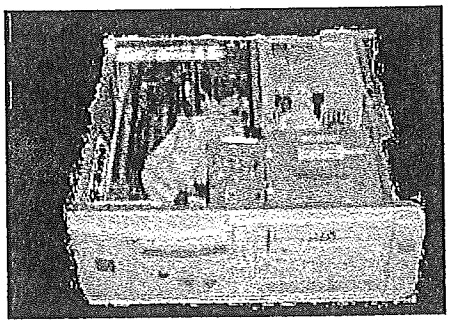


Figure 6-2 *Another sample of the white halo surrounding the picture after applying the Background Transparent ink effect*

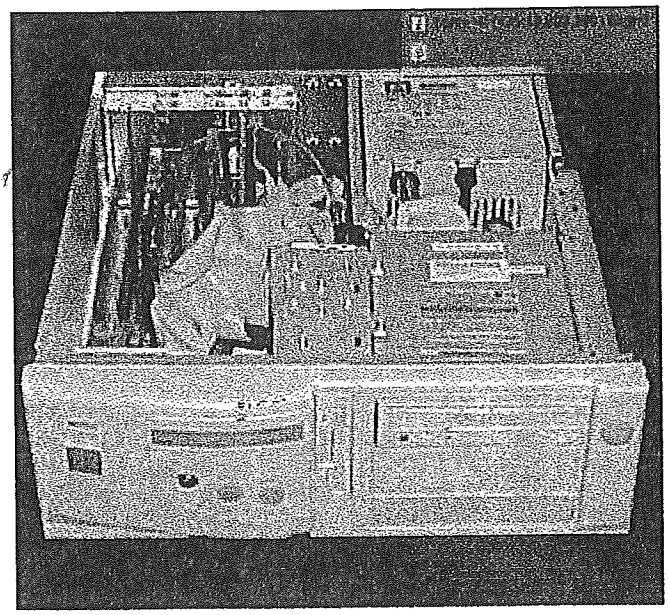


Figure 6-3 *The white halo is removed, and the picture is edited*

6.2.2 Cropping Pictures

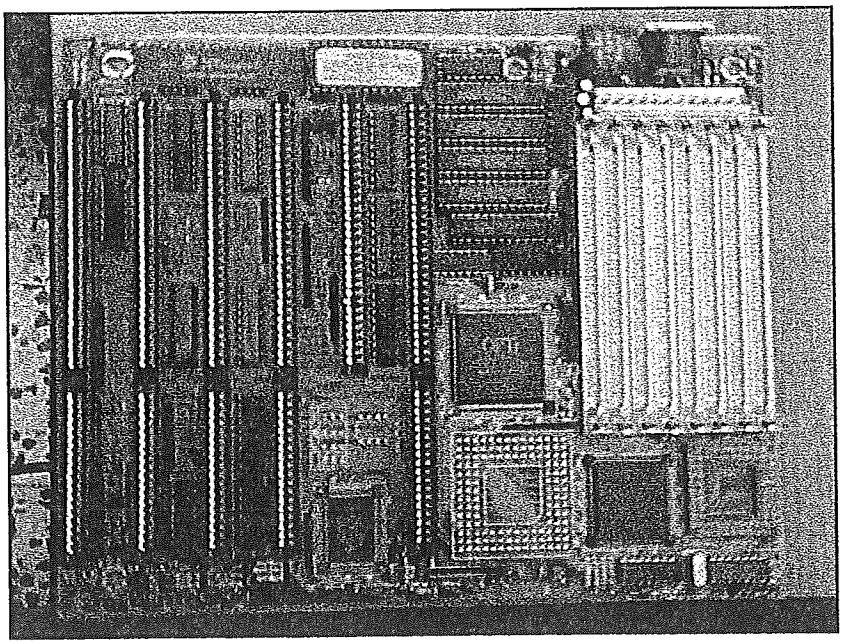


Figure 6-4 *A picture of a motherboard before Cropping*

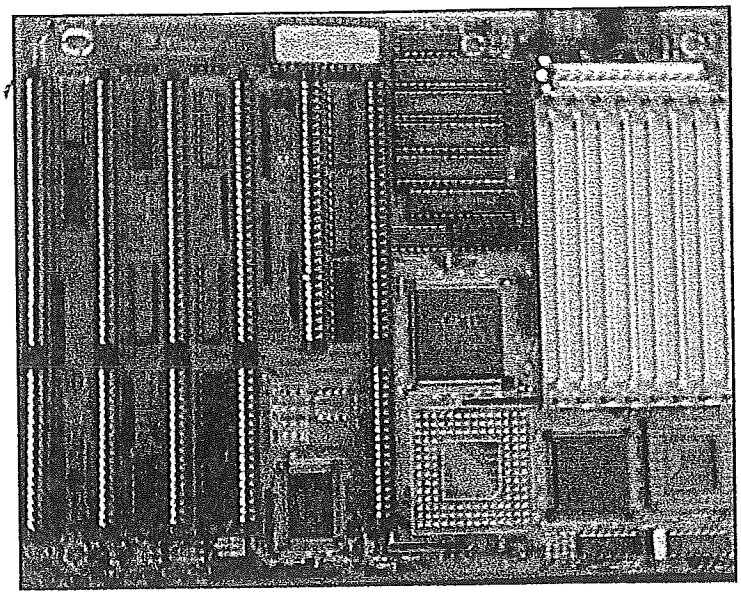


Figure 6-5 *A picture of a motherboard after Cropping*

6.2.3 Color and Shading Effects

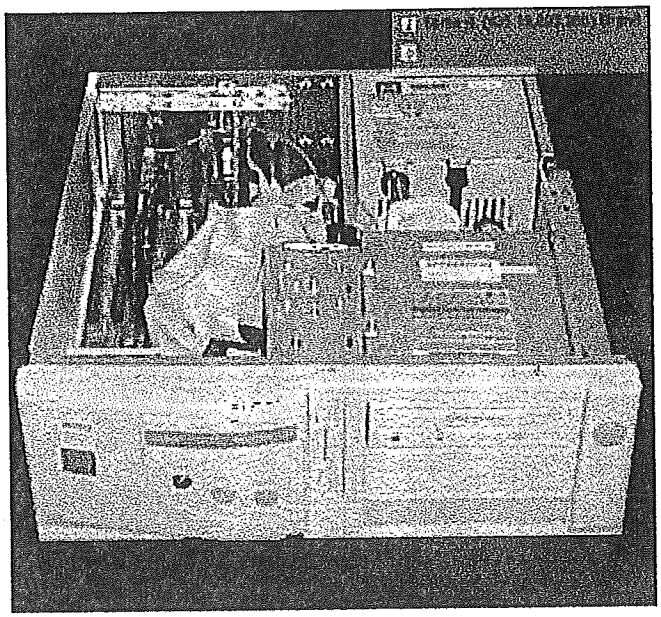


Figure 6-6 *A picture before applying the Color Balance effect*

6.2.4 Blurred Pictures

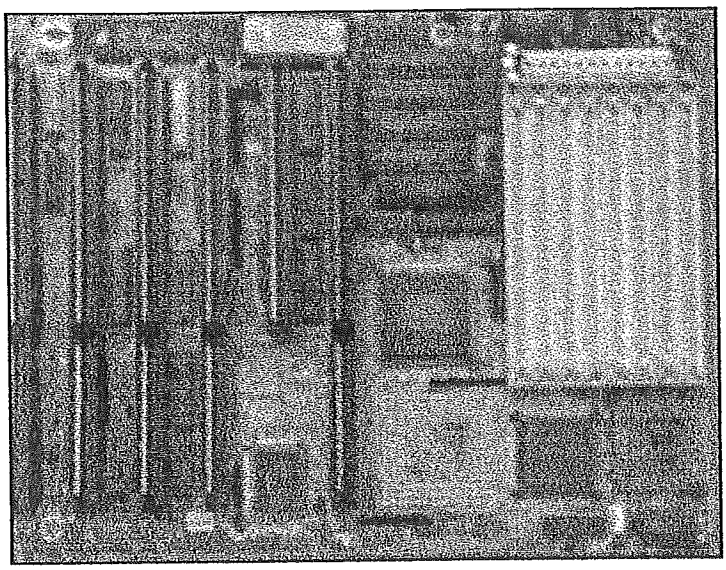


Figure 6-7 *A blurred picture of a motherboard*

6.2.5 Size

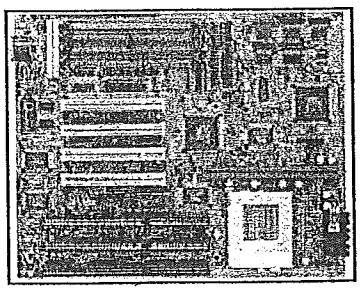


Figure 6-8 *A picture of a motherboard that is too small to be used*

6.3 Implementation

Our project explains the main components of the motherboard. It begins by showing a movie of the previous generations of motherboards, starting with a 286. After the movie completes a picture of a Pentium® motherboard is centered on the screen. This picture is the main menu that the student uses to select the topic that they want to learn about: the CPU, RAM, Expansion slots, BIOS, Drives Controllers, Serial and Parallel ports, and the clock chip.

The tutorial was implemented by dividing it into a number of movies:

1. The main movie includes the introduction and the history of the motherboards, and the main Pentium motherboard, where the student can select a component to learn about.
2. Component movies - one movie is used to explain about a single component. The student clicks the component from the picture of the motherboard, and the corresponding movie is "called" and run.

6.3.1 Director 6.5 Features

To have the tutorial proceed as expected we need to use the following Director features:

a) Importing:

Pictures and other media elements are imported into Director using the Import command. The pictures we used in our presentation were imported into the Cast first, and then dragged onto the Stage or Score. Using the Score, we could place the Sprite in the frame where we want it to first appear on the Stage, and adjust the Sprite's life span, lengthening or shortening it according to our

needs. Then the picture is aligned either on the Stage by dragging it, or using the Score by entering the exact coordinates.

b) Using the Background Transparent ink:

Each Sprite has a rectangular bounding box surrounding it. If the Sprite is irregular, then the area between it and the bounding box is white. When such a Sprite is placed on the Stage, the default ink used is the Copy ink, so the white background surrounding the Sprite is visible. To change the Sprite's ink to Background Transparent, we used the ink control on the Score. This made the white background invisible. We used this with all sprites placed on the Stage.

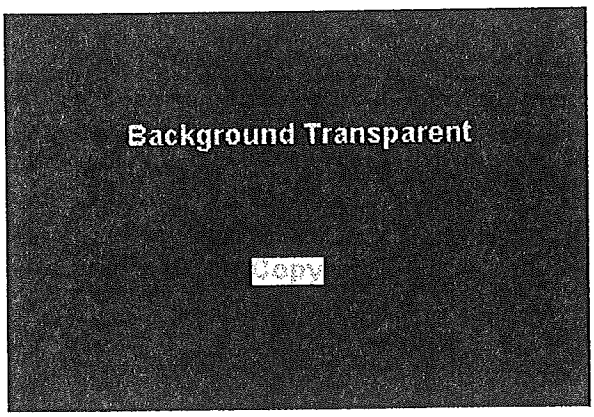


Figure 6-9: *The Background Transparent and Copy inks*

c) Keyframes:

Keyframes describe the movement of Sprites in a Director movie. To move a Sprite across the Stage, we specified keyframes in the Score, along the Sprite's life span. Movement of sprites is achieved by specifying a different position for the Sprite in subsequent keyframes. The animation at the history of motherboards sequence was done using keyframes.

d) Autodistort:

This effect causes Director to produce resized images of a Cast Member. Selecting the image in the Paint window and resizing it, then selecting the Autodistort option and specifying the number of intermediate Cast Members will create the specified number of Cast Members. These images are placed in the Cast sequentially with the members ranging in size from the original to the resized image. This option, along with the Filmloops feature, was used to produce the opening case sequence.

e) Filmloops:

The Filmloop feature is used to create a Cast Member that is actually a number of pointers to other Cast Members that were created earlier. Placing the Filmloop on the stage creates an animation showing the Cast Members in sequence. A filmloop, along with the autodistort feature, were used together to create the opening case effect. Creating 15 Cast Members showing the cover getting smaller using autodistort, and then creating a filmloop to obtain an effect of an opening case achieved this.

f) Real-Time Recording:

The yellow ball that moves across the stage was made using the Real-Time Recording option. We moved the ball across the Stage using the mouse and recorded this movement. It is important to do this at a steady pace.

g) Text:

The text that appears in the movie was written either in Director's Text Editor, or in external editors, such as Microsoft Word, where it was later transferred to Director by Copying it into the clipboard, and then Pasting it in Director.

h) Pausing the movie:

A pause in a Director movie forces the playback head to loop on the same frame. This feature is necessary to make the movie wait while the user interacts with it. This is achieved by using a script, called "Hold on Frame", which causes the movie to loop on the selected frame. We used this script at the motherboard screen, to pause and wait until the student clicks a component.

i) Interactive Buttons:

The buttons in our project are all simple graphics, with custom scripts and predefined behaviors attached to them. An interactive button Sprite has some visual characteristics that indicate that the Sprite is actually a button. Among these characteristics:

- When the mouse pointer is over the button, the mouse pointer image changes.
- When the mouse pointer is over the button, and the mouse button is clicked, the appearance of the button changes.
- When the mouse button is released, the button reverts back to its original state, and some action occurs.
- If the mouse button is clicked over a button, then moved off the button and released, no action associated with the button occurs.
- If the button is an image, a tool-tip appears when the mouse pointer is moved over the button, regardless of the button being clicked.

Following are the main behaviors associated with most of the buttons in our project.

1. **Changing the mouse pointer:**

   The mouse pointer changes over a button sprite, from a pointer (arrow) to a hand. This behavior was applied to all buttons.

2. **Toggle button:**

   A toggle button is a button that switches between two states, depressed and not depressed. Unlike a regular button, this button remains in the same state until it is clicked again. Regular buttons appear to be depressed when the left mouse button is held down, but return to their initial state when the mouse button is released. We used a toggle button with the text button.

3. **Changing the button image:**

   The 3D images that we used for the buttons change, and the button assumes a depressed image. This was accomplished by changing the Cast Member upon which the button Sprite is based, so that when the mouse button is clicked, the button appears to be depressed. This was applied to all buttons.

4. **GOTO frame:**

   This is used to move the playback head to a different frame. We specify a frame in the movie that the playback head will jump to when we apply the script to the Sprite. We used this with the Play button, to start playing the animation describing a component part.

5. **GOTO Movie:**

   Behaves in the same way GOTO frame does, but it jumps to a specified frame in another movie. We have to specify the frame to jump to, as well as the file name of the new movie. We used this with the exit button, to jump between the main movie and the component movies, and with the component parts of the motherboard.

6. **Open external file:**

   This feature is used with the help button to open an external application to view the help.

7. **Tool Tips:**

   Sprites have a property called `the visible`. This property signifies the visibility state of the Sprite. Setting this property to False makes a Sprite invisible on the Stage. This is the property we used to make tool tips appear/disappear according to the mouse's location. It was also applied to the text box button, except it depends on the button's state, rather than the mouse's location.

8. **Quit:**

   This was used to permanently close and exit the movie.

j) Sound:

After creating the sound files, we imported them into Director. The Score window has two sound channels, in which sound Cast Members can be placed. Director has other sound channels that can be accessed through Lingo (For more information on Lingo, please refer to section 6.3.2). Director can not play multiple sound files simultaneously by mixing the sound files in different channels to produce a single sound. We placed the sound Cast Members in their desired location on the Score, in one of the sound channels. We then set the volume of the movie using the Control Panel or by lingo behavior. In the sound Cast Member properties, we selected the "Loop" option. We also added a wait state in the Tempo Channel (For more information on the Tempo Channel, please refer to Appendix A – Director), and specified that the movie wait until the sound file is done. We discovered that the button behaviors were not sufficient to deal with the sound Sprites, so we added extra Lingo statements to the behaviors. We added commands to control when the sound starts or stops, these commands enabled us to control when the sound file starts playing in the movie. The "Exit" and "Play" button behaviors were successfully altered, but we encountered a problem with the "Pause" button. The sound file would play correctly, but when the movie is paused, the animation and sound would pause, but upon restarting, the animation would

continue from the point it had reached, while the sound file would restart from the beginning. We came up with the following solutions:

- The first solution to this problem was deleting the Pause button. However, as we had put a lot of time and effort into creating this button, we set aside this option as a last resort.
- The second solution was inserting cue points into the sound Cast Member, but as Macromedia's SoundEdit 16 – the program that inserts cue points in the sound Cast Member – was not available, we were forced to look for other options.
- The third option was using Lingo to pause the Cast Member at some point. Lingo does not support such instructions for WAV Cast Members. It does, however, support instructions that enable a Shockwave Audio file (SWA) to be paused. This instruction uses the "Percent of Member" property to pause the SWA member. We converted the WAV files into SWA files. But after placing the SWA Cast Member on the Stage, and creating a projector, the sound did not play.
- The last option was converting the WAV file into a digital video, which would enable sound to be restarted from the pause point. The digital video could be placed on the Stage, and made invisible through `the visible` property. We converted the sound files into digital video files using Asymatrix 3D F/X. The size of the digital video file was larger than the size of the WAV file by a few kilobytes. No new instructions were needed; the available behaviors were sufficient. We discovered that making the digital video Sprite invisible caused it not to play, so we resized it to a very small dot that is not visible to the eye. This solved the problem.

Note:

Only the sound Cast Members that were placed in the same frames as a Pause button had to be treated in this manner. Other sound Cast Members were treated as regular WAV files, and required a few extra behaviors to control the sound Start/Stop.

k) Video:

Director imports video files using one of two methods, the first is linking the video to the movie and the second is placing the video file into the Cast. We imported the videos, after obtaining them from various CD's, into Director. When a file is linked to a movie, a pointer to the location is placed in the Cast. So it is essential that the video remain in the same location. We gathered all the videos into one directory and then linked them to the movie using the Import command. We then placed the videos in the required positions on the Score, extended the Sprite's life span to the video length, and cropped the video to fit the window.

6.3.2 Lingo

In the older versions of Director, when it was known as VideoWorks, the developer could not control the interaction between the computer and the user. With the next iteration of Director, Lingo - Director's scripting language - was introduced. Movies created using the Score are frame driven, that is, the visible scene changes from one frame to the next. Lingo is event driven; scripts are written to handle different events.

6.3.2.1 Scripting Concepts

Scripts are composed of a number of routines, called *handlers* in the Lingo scripting language. Handlers begin with the keyword on and end with the

keyword `end`. The keyword `end` tells Director that any code following it is part of another handler. The Director projector doesn't stop here; it terminates at the `quit` or `halt` keyword. The body of a handler contains the code that will execute when the handler is called. A typical handler looks like this:

```
on something
      doThis
      doThat
end
```

Handlers are executed in response to a *system message*. The user's computer generates a system message when an *event* occurs. An event is an incident in the movie that initiates the execution of a script. The start of a movie, entering or exiting a frame, user interaction through the mouse or keyboard, and the conclusion of a movie are all examples of events. The most frequently generated messages are `mouseDown` and `mouseUp`. The order of actions follows this sequence:

- Events occur as a movie plays.
- Events generate messages.
- Scripts that are composed of one or more handlers respond to specific messages.

6.3.2.2 Director 6.5 Script types

Director has four types of scripts: Cast Member scripts, Movie Scripts, Parent Scripts, and Score Scripts.

1. Cast Member Scripts:

A Cast Member script is attached to the Cast Member. To create a Cast Member script, we selected the Cast Member in the Cast window, and then clicked the script button in the Cast window to enter the script. We attached Cast Member scripts to Cast Members that constantly perform the same function, such as the exit button in the component movies. This button always

exits the component movie and returns to the main movie at the same frame. Following is an extract from the script required to jump to a different movie:

```
on mouseUp me
  if whichEvent = #mouseup     then init me
end

on prepareFrame me
  if whichEvent = #prepareframe then init me
end

on exitFrame me
  if whichEvent = #exitframe  then init me
end

on init me
  set the movieName of me = get_filename( movieName )
  case ( playMode ) of:
    #"Go to":
      go to movie movieName
    #"Play and Return":
      play movie movieName
  end case
end
```

2. Score Scripts:

There are two kinds of score scripts; Sprite scripts and frame scripts.

a. Sprite Scripts:

A Sprite script is attached to a Sprite. Sprite scripts are stored as Cast Members in the Cast. To attach them to the Sprite, we dragged the script Cast Member onto the Sprite. Most of the buttons have Sprite scripts attached to them, such as a script to move the playback head to the beginning of the movie in component movies, the script attached to the text button that makes it behave as a toggle button, and the script to switch Cast Members so that buttons appear to be depressed when clicked. Following is an extract of the script we attached to the text button that makes it behave as a toggle button:

```
on mouseDown me
  set the tracking of me = TRUE
  toggle me
end

on mouseEnter me
  if the tracking of me then
    toggle me
  end if
end

on mouseLeave me
  if the tracking of me then
    toggle me
  end if
end

on mouseup me
  if the tracking of me then
    set the tracking of me = false
  end if
end

on mouseupOutside me
  if the tracking of me then
    set the tracking of me = false
  end if
end


on Toggle me
  setValue( me, NOT the Setting of me )
end
on SetValue me, v_me
  if v_me = 0 then  -- setting "OFF"
    set the member of sprite the spriteNum of me =
the normalMember of me
    set the Setting of me = 0
  else              -- setting "ON"
    set the member of sprite the spriteNum of me =
the toggleMember of me
    set the Setting of me = 1
  end if
  updateStage
end
```

b. Frame scripts:

A frame script is a script attached to a frame of the movie. Frame scripts are stored as Cast Members in the Cast, just as Sprite scripts are stored. To attach a frame script to the movie, we dragged the script Cast Member onto the Script Channel in the Score. An example of a frame script we used in the project is a frame script to pause the movie at the motherboard picture.

```
on exitFrame
  go to the frame
end
```

The above script forces the movie's playback head to loop on the current frame by using `the frame`, a keyword that denotes the current frame the movie is in.

3. Movie Scripts:

Movie scripts are global scripts that can be invoked from anywhere in the movie.

4. Parent scripts:

Parent scripts are special scripts used to create child objects. Parent scripts are used with object-oriented programming. We did not use parent scripts in our movie.

When an event is generated and multiple scripts exist for this event, the Sprite script has the highest priority and is executed first. Cast Member scripts are executed next, followed by frame scripts, and movie scripts have the lowest priority.

6.3.2.3 Sample Events

Some of the most commonly used events are:

Message	Event that causes the message to occur	Script type
EndSprite	The playback head leaves the last frame of the Sprite's span.	Score
enterFrame	The playback head enters the frame.	Frame, Movie
exitFrame	The playback head has not moved to the next frame yet, but the frame has been drawn	Frame, movie
mouseDown	The primary mouse button (left) has been pressed.	All
MouseUp	The primary mouse button (left) has been released.	All

Table 6-3 *Messages, events, and script types*

6.3.3 Supplemental Programs

When we began the implementation part of our project, we used many programs to create and edit the materials that we used.

6.3.3.1 Picture Editing Programs

1. Ulead PhotoImpact 4.2

We used this program to create the frames surrounding the component movie, and the animation and text window. PhotoImpact has a color-balancing option that we used to visually shift the prevalent tints in an image, so that all the images appeared to come from one source. We also used it to make the image lighter or darker, and to make some parts stand out more than others. Many other features were useful. All buttons were edited in PhotoImpact.

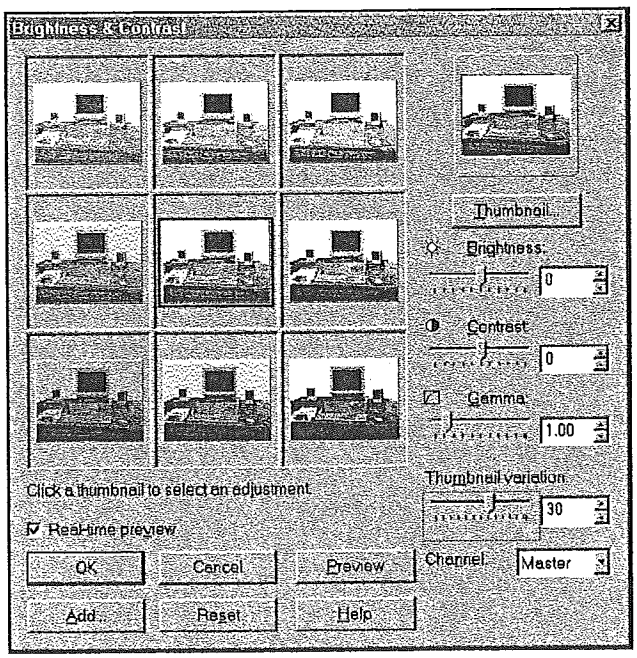


Figure 6-10 *Changing a picture's Brightness and Contrast using PhotoImpact's Brightness and Contrast Dialog Box*

2. Adobe Photoshop 5.0 ME

We used Photoshop to place a shadow around the picture edges. This causes the student to focus on the center of the image, and not to be distracted by other unimportant parts of the picture.

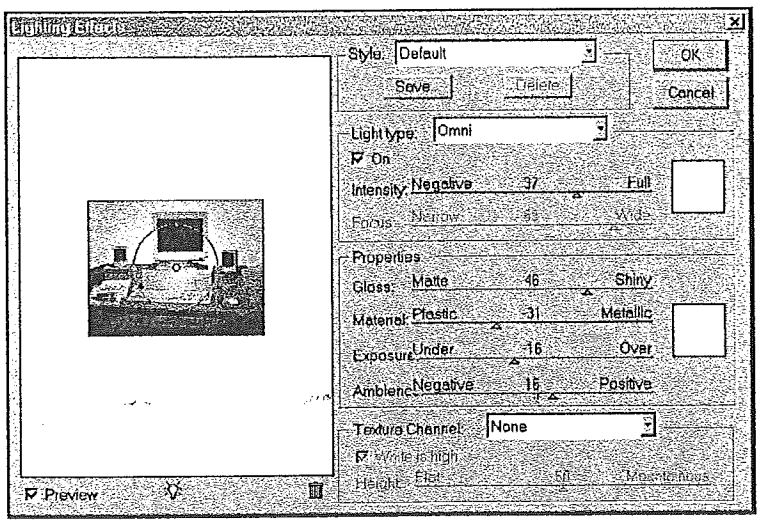


Figure 6-11 *Creating Shadow effects using Photoshop*

3. Microsoft PhotoDraw 2000

PhotoDraw has an edge-finder feature, which was indispensable to us in editing and creating pictures, because they came from many different sources, and we had to remove some undesirable parts, while keeping the part we needed.

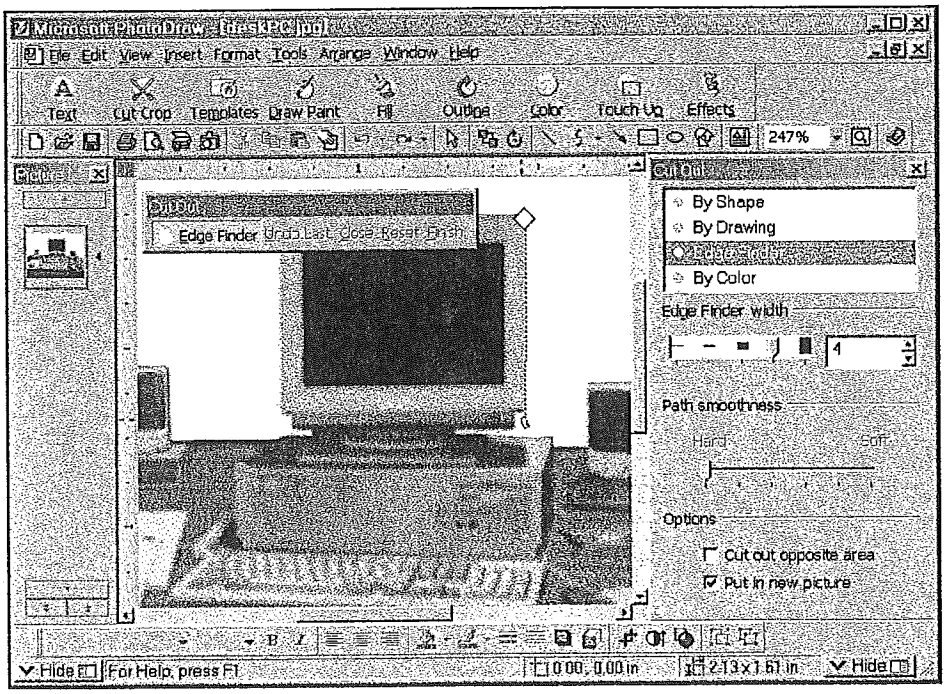


Figure 6-12 *Using the Edge-Finder feature to select the parts we needed*

4. Other Programs

We used Microsoft Paint, to remove extra pixels. We also used Corel Gallery to create backgrounds and some other effects, and Corel Draw to resize and further edit the images.

6.3.3.2 3D Programs

We used many 3D programs in our efforts to create 3D animations, but since Director does not support 3D computer graphics, any animation we created had to be saved as a video file, in AVI format, and imported into Director as a regular video file. 3D Studio Max can create interactive 3D multimedia presentations, but we could not use it due to its complexity.

Some of the programs we used to create some 3D animations were:

- Macromedia Extreme 3D 2: We used this to create a 3D Video, but did not use it in our project..
- 3D Crystal Impact: Rotates images in 3D directions.
- Asymatrix 3D F/X: We used this for some 3D modeling in the introduction, and creating the digital video sound files.

6.3.3.3 Sound

We used Microsoft Sound Recorder to record and edit all sound files. We then used Asymatrix 3D F/X to convert these files into digital video files.

6.3.3.4 Text

The text that appears in the text window was created in Microsoft Word 2000. Formatting was applied in Director's Text Editor, where the Font, Font Size, and Font Style settings were set.

6.3.4 Implementation Issues

During the implementation process, we came across some problems that might cause our project to fail to run. These were all caused by the danger that some missing file, or lost link, would be encountered.

1) Internal Casts Vs. External Casts

As mentioned previously, Internal Casts are stored within a Director movie, while External Casts are saved in a separate file with a .CST extension, and are linked to the movie. Fearing that a Cast may be displaced or lost, we used

Internal Casts in our project. That way, the Cast would be automatically packaged with the projector.

2) Linking Videos

When packaging a movie, all linked files must be specified by their path and file name. For this reason, we had to save the videos on the application CD, and then link these files to the movie, so that the file path would specify the correct file location.

3) Linking Movies

Our project is composed of one main movie, and a number of component movies. These movies are all linked to the main movie using Lingo scripts, placed at the Pentium motherboard picture, which serves the purpose of a menu. When packaging the presentation, we created a projector using the main movie, but the component movies did not need to be packaged. We also had to place these movies on the application CD and then link them to our movie, before creating the projector, for the same reason stated above.

Chapter 7

This chapter presents the project summary, and outlines future work options.

7.1 Summary

We have successfully completed a tutorial that teaches students about the major motherboard components. We designed the tutorial using an intuitive graphical interface that is easy to use. The interface was designed using the principles described in the previous chapters. The contents of the tutorial presented the material as specified by book number 5 in our bibliography. The book describes the optimal method to teach using computers, in an easy to understand manner.

7.2 Future Work

After completing the project a few shortcomings of the project became apparent. Therefore, as a continuation of our project, we would like the following work to be done:

The desk picture has other components that may be added to our project. Separate component movies may be created for the external speakers, the keyboard, and the monitor.

The desk drawers present three options; Inside, Back, and Index. Currently, the Index and Inside buttons work. The Inside button it takes the user to the inside of the PC. The Index option gives an index of all possible component movies, so the user could access any part at a glance. We would like the Back option to be completed. The back would show the back of the PC, and allow the user to directly access the information concerning the Ports, the power supply plug, and any other items available at the back of the PC

The Inside of the PC shows the motherboard, the power supply, the CD, and the hard and floppy disks. We completed the motherboard section. The other sections can be completed in another project.

Director provides, through Lingo, the potential to personalize a presentation. We would like our project to be linked to a database that would store student information. The student would enter her name and ID, and her information would be accessed and – after using the program - updated accordingly.

And finally, we would like the tutorial to provide the students with some questions to gauge the understanding of the student. The project should not allow a student to leave a subject for another one until she has fully understood the subject.

Appendix A

Director

Director 6.5 is the program we used to develop our multimedia application. This appendix provides information on Director.

Macromedia Director® is an animation program. It enables the application developer to combine all the elements of multimedia in one format. A Director file is called a movie, and has a .DIR extension. Upon completing the multimedia product, the movie is packaged into an executable file, called a projector. The projector can then be distributed.

A.1 Director Basics

When Director is first started, a new movie file is created by default. A black rectangle is centered in the window, and a few other windows are activated. This black rectangle is called the Stage, and the final animation is displayed in it. The visible windows displayed by default are the Cast window, the Score window, and the Control Panel.

A.1.1 The Stage:

The Stage is "the stage" in the theatrical sense where a play is acted out. This stage is a blank screen that is visible when Director is opened. It has no close box, no resize or zoom box, and no scroll bar. As long as you are in Director, the Stage window stays in view. Elements placed in a Director movie can be viewed on the Stage frame by frame.

A.1.2 The Cast Window:

The Cast window is an array of 32,000 cells. These cells contain the different elements that are found in a multimedia production, such as graphics, text, sound and video. When multimedia elements are placed in the Cast window, they are called Cast Members. Cast Members can either be created directly in Director, such as painting a bitmap Cast Member in the Paint window, or added to the Cast window using the Import command.

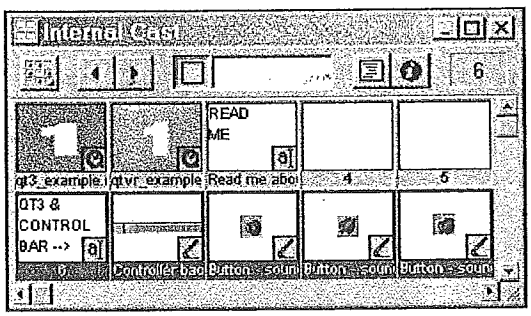


Figure A-1 *The Internal Cast Window*

There are two kinds of Cast windows:

- *The Internal Cast Window:* The Internal Cast is the default cast for a movie. It is always linked to the movie, and is saved in a Director movie internally.
- *The External Cast Window:* The External Cast contains Cast Members stored in a separate file. It can be shared between movies, or used as a library of commonly used Cast elements. To use an External Cast in a Director movie, the Cast file must be linked to the movie. External Cast files have a .CST extension.

A Director movie can have multiple internal and external casts.

A.1.3 Sprites:

When a Cast Member is placed on the Stage or in the Score window, it becomes a Sprite. A Sprite is not actually the graphic that appears in the Cast window, but a copy of it. Multiple copies of the same Cast Member may be placed on the Stage, creating different Sprites. Changes made to a Cast Member effect its associated Sprite, but changes made to Sprite do not effect the Cast Member. Director 6.5 has 120 sprite channels, so the application developer is able to have 120 sprites on the Stage at a time. The overall positions of the sprites can be viewed through the Score. The lifetime of a Sprite is the time the Sprite resides in the Stage, which is the same as the number of frames the Sprite spans in the Score.

A.1.4 The Score Window:

It is a frame-by-frame record of what occurs on the Stage. Visually the Score window is a grid of vertical frames and horizontal channels plus a number of

controls. A frame is the shortest time span available in a movie, and can be thought of as an individual view of the movie each time it changes. When one frame varies a little from the next, it produces an animation effect. Channels are the layers in which Sprites are placed in an animation. An animation showing a person in front of a house may have these layers: the background, the house, and the person. Sprites are drawn on the stage in the order they appear in the Score, starting with channel 1. So channel 1, 2, and 3 will contain the background, the house, and the person, respectively.

Director supports 6 special purpose channels, in addition to the 120 regular channels. These channels control the sound, palette, tempo, and store frame scripts. Controls provide information about the Sprites that live in the Score, such as the originating Cast member and location of each Sprite that appears on the Stage.

The Score window enables the application developer to alter the appearance of Sprites using controls such as ink effects, and the performance of Sprites using behaviors.

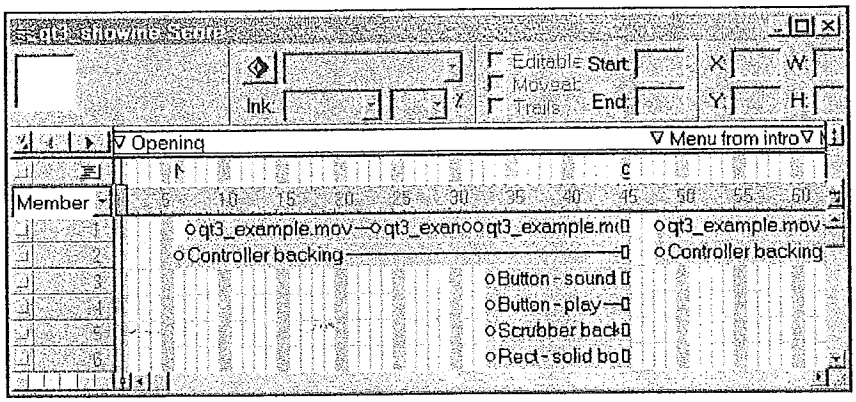


Figure A-2 *The Score Window*

A.1.5 Behaviors:

A Director behavior is a script or page of code that includes subroutines for events that occur in a movie. Scripts can be attached to a Sprite, a frame of the

movie, a Cast Member, or may be a global script, called a movie script. A Sprite behavior is associated with that particular Sprite and has no effect on the Cast Member the Sprite is based on. A behavior attached to a Cast Member effects all the Sprites based on that Cast Member.

A.1.6 The Control Panel:

The Control Panel enables the application developer to view and control the playback of a Director movie. The Control Panel has controls that play, stop, and rewind a movie. It also controls the frame rate and volume of the movie. Other options that the Control Panel provides are: Step Forward by one frame, Step Back one frame, a frame counter that displays the current frame, and whether to play the whole movie or a selected number of frames.

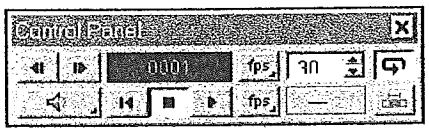


Figure A-3 *The Control Panel*

A.1.7 The Paint Window:

The Paint window is Director's graphics editor, and is similar to Microsoft Windows' Paint program. The Paint window enables the application developer to edit bitmap images pixel by pixel. It employs standard painting tools, such as a pencil tool and a fill tool, and two unique tools; the Registration tool, and the Onion Skin tool. The Registration tool sets the registration point of an image, which is used to align the Sprite on the Stage, and the Onion Skin tool, which displays preceding and subsequent Cast Members, in order to assist the application developer in producing images that vary slightly from the original, to create a smooth animation sequence.

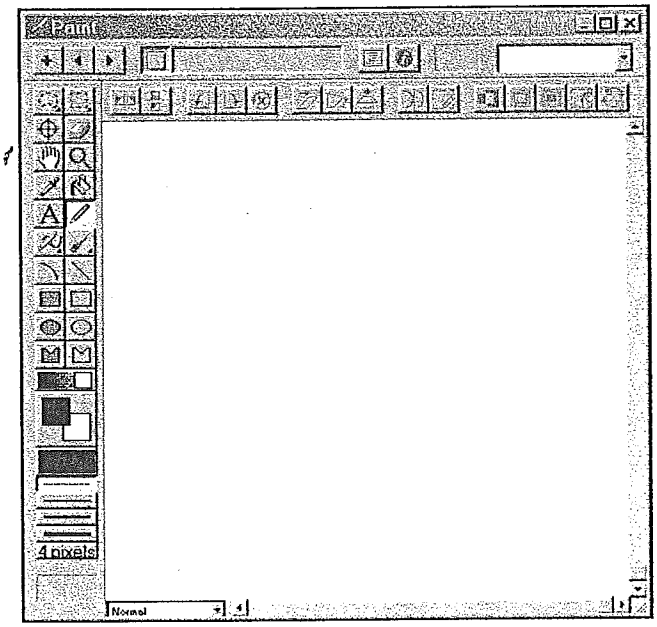


Figure A-4 *The Paint Window*

A.1.8 Xtras:

Xtras are support programs, which add to the capabilities of Director. Xtras produced by Macromedia come packaged with Director. To use Xtras made by third party developers, they must be placed in the Xtras folder. The Xtras folder is in the folder that holds the Director program. Xtras must be distributed with the multimedia application, or integrated directly into the application itself.

A.2 Director 6.5 Multimedia Issues

A.2.1 Import formats

Director for Windows can import the following file types: BMP, GIF, JPEG, LRG (xRes), Photoshop, MacPaint, PNG, TIFF, and PICT file formats. It also

supports Photo CD, PCX, WMF, PostScript and the FLC and FLI multiple image formats. Director imports these files either in a Standard Import Format, which imports the complete image into Director's Internal Cast, so the image can be edited in the Paint window, or as a Link to External File, where a temporary image is imported into the Internal Cast, but when the movie is later opened and run, the graphic is loaded from the external file. Although Director supports the above formats, it saves the imported pictures internally as bitmaps (BMP), which is very space consuming.

A.2.2 Sound

Sound is available in 2 formats in Director; Waveform Audio files (.WAV), and Audio Interchange files (.AIFF). The format a producer chooses for an application depends on the type and quality of the sound the application developer wishes to play, the amount of storage space available to sound files, and whether audio CDs will be distributed with the application.

- *Waveform audio:* A wave audio (.WAV) file contains sound recorded with a microphone and stored in digital format. WAV files used in a Director application include voiceovers, sound effects, and high-quality music. To create a WAV file, the recording device samples the incoming sound waves and creates a digital representation of the sound. When a WAV file is played, the sound card plays it through the attached speakers. Because WAV files represent an actual recording, they can contain any type of sound. However, WAV files can become very large; a one-minute recording can be 1.5MB to 10MB in size. The file size depends on the quality of the recorded sound, which is influenced by three factors:
    a. **Sampling frequency:** The higher the sampling frequency, the wider the range and quality of the sound. Sound is sampled at three standard frequencies: 44.1 KHz for full-range music and

high quality speech; 22.05 KHz for medium quality speech; and 11.025 KHz for sound effects.

b. **Bits used:** The number of bits used to represent the sound determines how closely the recorded sound approximates the original sound. An 8-bit recording reasonably represents speech or sound effects, but a 16-bit recording is required for music. To play a 16-bit recording, a 16-bit sound card is needed, because most 8-bit sound cards can only play 8-bit recordings.

c. **Stereo or monophonic recording:** The higher the quality, the more space is required to store the file. For example, a 44.1 KHz recording requires twice as much storage space as a 22.05 KHz file, and stereo recordings require twice as much space as monophonic recordings.

- *Audio Interchange:* AIFF files do not support compression, but they have the advantage of being multi-platform. Both PCs and Macintosh machines can play AIFF sound files. This format is not a native format for PC's, but Director and a variety of other software programs support AIFF files on the PC without the use of external software.

A new feature of Director 6 is that it incorporates streaming audio; it can stream audio off a CD or the Internet.

A.2.3 Video

In Director, video is typically used with the following formats: Audio/Video Interleaved (AVI), Apple® QuickTime™ for Windows, and MPEG (Motion Picture-Experts Group) formats. The format the application developer chooses depends on the quality of the video and the level of convenience needed when distributing the application. Digital video files contain both a digital video image and an accompanying soundtrack. Director can play digital video files without special hardware, using appropriate drivers. For this reason, the

computer's CPU plays the video. To reduce the impact of playing video on other CPU operations, digital video files often have:

- An image size one-fourth smaller than the screen size. Expanding the video image lowers the resolution.
- About one-half the number of frames as videodisc-quality video (15 per second versus 30 per second), which makes the video look less smooth.
- Only 256 colors, which can make the video appear grainy.

An accelerator board is used to obtain the highest quality playback. Digital video files can be edited with a video editor, provided that they reside on the hard disk. Digital video files stored on CD cannot be edited. Digital video files are often very large; a minute of video can occupy as much as 20MB of disk space. To use long video flies, the application should be distributed on CD-ROM.

MPEG format:

Director doesn't directly support MPEG (Motion Picture Experts Group) format for digital video files. MPEG uses interframe compression to achieve compression rates up to 200:1. Interframe compression analyses each frame of a video for redundancy, to create reference frames. By comparing previous and subsequent frames in a video, only the difference between the frames is stored. That way, redundant information is removed, and the result is a smaller file. Files compressed using MPEG can be lossy; as the compression ratio goes up, the quality of the video is reduced. However, MPEG is less lossy than compression schemes employed by QuickTime or AVI formats. To play MPEG videos in Director movies calls for the assistance of any of the following: an Xtra, Lingo MIC messages, a Windows driver, or a Macintosh system extension designed for this purpose. The drawback with MPEG is that each type of software implements the compression differently, and this can

make it difficult to control the quality of playback on all types and platforms of computers.

A.2.4 Color depth

Color depth determines the number of bits that are used for each pixel in the image. The way computers handle color is by storing the relative intensity of the three primary colors red, blue and green for each pixel. In 24-bit color mode, each of the three primary colors has 8 bits to represent it, so it can assume 256 possible colors. The same concept applies to 16-bit colors, except the primary colors have 5 bits to represent them.

A.2.5 Palettes

When Director is installed and running on a Windows system, it uses the current system palette. Director either remaps any imported image to its own palette, or imports an 8-bit color palette with the imported image. The latter option is not available if the movie's color depth is not 8-bit. When Director imports the palette, it stores it as a Cast Member separate from the image.

A.2.6 Exporting a Director Movie

A Director movie can be exported in Video for Windows format using Director Windows, or in QuickTime format using Director for the Macintosh. This causes any interactivity in the movie to be lost, and if exporting in the Windows AVI format, it also causes the loss of sound. Director enables the application developer to compress the file, but a high compression rate lowers the quality of the video image.

Appendix B

Lingo Listing

(Partial Listing)

The listed scripts in this appendix are some of the scripts we used in our project. The listed scripts are built-in scripts that come packaged with Director 6.5, and our custom scripts. There are too many scripts for a complete listing.

Script 1: prepares sprites to start movie (programmed by us)

```
on enterFrame
  clearFrame
  repeat with count = 23 to 28
    puppetSprite count, True
    set the visible of sprite count to false
    end repeat
end
```

Description:
Clear the Director parameters before starting or restarting the movie and make the text cast members which represent the tool tips invisible on the stage.

Script 2: Cycle Me Continuous (built - in behavior adjusted by us)

```
-- Cycle Me Continuous -- updated 3-6-97 lisa
-- n state cycling, bounce list
-- relies on cast members being contiguous

property Enabled
property MemberMin, MemberMax, ImageCastLib
property CycleStyle  -- cycling, bouncing
property tracking
property goingDown     -- TRUE   if going through the list in
reverse order
--                        FALSE if traversing the list normally
property CurrentState
property NumStates
property name

on mouseEnter me
  set tracking = TRUE
  bumpstate_intern me
end

on prepareframe me
  if tracking then
    bumpstate_intern me
  end if
end

on mouseLeave me
  set tracking = FALSE
end

on SetState me, statenum
  if the enabled of me then
    if statenum > the numstates of me then
      set statenum = the numstates of me
    else if statenum < 1 then
      set statenum = 1
    end if
```

```
    set the member of sprite the spritenum of me = ¬
            member ( the membermin of me + statenum - 1 ) of
castLib imageCastLib
  end if
end
--
on setGoingDown me, value
  set the goingDown of me to value
end

on SetCycleStyle me, value
  set the CycleStyle of me to value
end
---
on bumpstate_intern me
  if the enabled of me then
    if ( the CycleStyle of me = #Repeat ) then
      if NOT the goingDown of me then -- going Up
        set curr = the currentstate of me  + 1
        if curr > the numstates of me then
          set curr = 1
        end if
      else -- going Down
        set curr = the currentstate of me  - 1
        if curr < 1 then set curr = the numstates of me
      end if
    else   -- Reverse cycle
      if NOT the goingDown of me then -- going Up
        set curr = the currentstate of me  + 1
        if curr > the numstates of me then
          set curr = curr - 2
          set the goingDown of me = TRUE
        end if
      else   -- going Down
        set curr = the currentstate of me  - 1
        if curr < 1 then
          set curr = 2
          set the goingDown of me = FALSE
        end if
      end if
    end if
    set the currentstate of me = curr
    setstate me, curr
  end if
end

on BeginSprite me
  set tracking = FALSE
  set membermin = the membernum of member membermin
  set membermax = the membernum of member membermax
  puppetsprite the spritenum of me, TRUE
  set the enabled of me = TRUE
  set the numstates of me = (the membermax of me) - (the
membermin of me) + 1
```

```
  set the currentState of me = ( the memberNum of sprite the
spriteNum of me ) - ¬
                                the MemberMin of me + 1
  set the imageCastLib of me = the number of castLib ¬
                                    ( the castLibNum of
sprite ( the spriteNum of me ))
  setstate me, the currentstate of me
end

on endSprite me --added, fmk
  puppetSprite the spriteNum of me, FALSE
end

on getPropertyDescriptionList
  set description = [:]
  if the currentspritenum = 0 then
    set cycledefault = 0
  else
    set memref = the member of sprite the currentspritenum
    set castlibnum = the castlibnum of memref
    set cycledefault = member (the membernum of member memref)
of castlib castlibnum
  end if
  addprop description, #Name, [ #comment:   "Item Name:", ¬
                                #format:   #symbol, ¬
                                #default:   #MultiState_1 ] ¬
  addprop description, #MemberMin, [ #comment:      "First
Image:", ¬
                                #format:   #member, ¬
                                #default:    cycledefault ] ¬
  addprop description, #MemberMax, [ #comment:   "Last Image:",
¬
                                #format:   #member, ¬
                                #default:    cycledefault ] ¬
  addprop description, #CycleStyle, [ #comment:      "Cycle
Style:", ¬
                                #format:   #symbol, ¬
                                #range: [ #Repeat, #Reverse ],
¬
                                #default:   #Repeat ] ¬
  addprop description, #GoingDown, [ #comment:      "Reverse
Order:", ¬
                                #format:   #boolean, ¬
                                #default:    FALSE ] ¬
  return description
end

on getBehaviorDescription
  return ¬
"Prepares a sprite to use a Series of adjacent Bitmap
castmembers to represent an set of mutually exclusive states
that the user can cycle through.  See below for usage.  " &
RETURN & ¬
"PARAMETERS:" & RETURN & ¬
```

```
"• Item Name - ( optional ) name of this control."  & RETURN &
¬
"• First Image - number of first castmember in the sequence."
& RETURN & ¬
"• Last Image  - number of last castmember in the sequence."  &
RETURN & ¬
"• Cycle Mode - specifies how images in cycle are repeated on
subsequent iterations. The Repeat mode repeats the same
sequence. The Reverse mode cycles back through the sequence in
reverse order." & RETURN & ¬
"• Reverse Order - proceed from last to first image by
default."  & RETURN & ¬
"MESSAGES:" & RETURN & ¬
"• CycleState - advance to the next image in the sequence, or
the previous image, if the GoingDown property is true. & RETURN
& ¬
"• SetState state_number - advance directly to position
state_number in the sequence."
end

on getAssocMembers --added, fmk
  set myPropList = [ ]
  --  repeat with x = MemberMin to MemberMax
  repeat with x = (MemberMin + 1) to MemberMax
    add myPropList, member x of castLib imageCastLib
  end repeat
  return myPropList
end getAssocMembers

on mouseUp me
  go to frame 113
end
```

Description:
Behavior that makes a sprite cycle within a range of cast members on mouse enter or stop cycling on mouse leave and jump to specific frame on mouse up.

Script 3: Play Rewind button (programmed by us)

```
on mouseUp me
  go to frame 23
  set the member of sprite the currentSpriteNum to "Rewind Up"

end

on mouseDown me
  set the member of sprite the currentSpriteNum to "Rewind
Down"

end

on mouseEnter me
  set the visible of sprite 28 to true
```

```
end

on mouseLeave me
  set the visible of sprite 28 to false
end
```

Description:
Behavior that sets the appearance and function of play button.

Script 4: pause button (programmed by us)

```
property flag


on mouseUp me
  if flag then
    set the flag of me = false
    continue
  else
    set the flag of me = true
    pause
  end if
  UpdateStage
end

on mouseEnter me
  set the visible of sprite 26 to true
end

on mouseLeave me
  set the visible of sprite 26 to false
end
```

Description:
Behavior that sets the appearance and function of the play button.

Script 5: Exit button (programmed by us)

```
on mouseUp me
  set the member of sprite the currentSpriteNum to "Exit Down"
  go to frame 228 of movie "main project"
end

on mouseEnter me
  set the visible of sprite 23 to true
end

on mouseLeave me
  set the visible of sprite 23 to false
end
```

Description:
Behavior that sets the appearance and function of Exit button.

Script 6: show text button (programmed by us)

```
on mouseUp me
  if the visible of sprite 9 = True then
    set the visible of sprite 9 to False
  else
    set the visible of sprite 9 to True
  end if
  if the visible of sprite 10 = True then
    set the visible of sprite 10 to False
  else
    set the visible of sprite 10 to True
  end if
  if the visible of sprite 11 = True then
    set the visible of sprite 11 to False
  else
    set the visible of sprite 11 to True
  end if
  if the visible of sprite 12 = True then
    set the visible of sprite 12 to False
  else
    set the visible of sprite 12 to True
  end if
  if the visible of sprite 17 = True then
    set the visible of sprite 17 to False
  else
    set the visible of sprite 17 to True
  end if
  updateStage
end

on mouseEnter me
  set the visible of sprite 27 to true
end

on mouseLeave me
  set the visible of sprite 27 to false
end
```

Description:
Behavior that sets the appearance and the function of show text button, which makes text filed on the stage visible or invisible.

Script 7: back to main menu button (programmed by us)

```
on mouseUp me
  UpdateStage
  go to frame 1
```

```
end

on mouseEnter me
  set the visible of sprite 25 to true
end

on mouseLeave me
  set the visible of sprite 25 to false
end
```

Description:
Behavior that sets the appearance and the function of the back to main menu button which return to main menu when it is clicked.

Script 8: help button (programmed by us)

```
on mouseEnter me
  set the visible of sprite 24 to true
end

on mouseLeave me
  set the visible of sprite 24 to false
end

on mouseup me

 end
```

Description:
Behavior that sets the appearance and the function of the help button.

Script 9: loop in current frame (built - in)

```
-- loop on current frame
-- Nav
-- This can be a frame script or sprite script because it has
-- no properties,
-- drag it into the frame script cell to have the movie loop
-- on the current frame

on ExitFrame
  go the frame
end

on getBehaviorDescription
  return "Goto the current frame"
end
```

Description:
After it draws the current frame and before it exits, it returns to it and does not go to the next frame.

Script 10: Hold on Current Frame (built - in)

```
-- Frame    Hold on Current
-- Nav
-- behavior library version 1.1
-- This can be a frame script or sprite script,
-- drag it into the frame script cell to have the movie loop on
-- the current frame
-- or drag it to a sprite to have the same effect.

on exitFrame
  go the frame
end

on getBehaviorDescription
  return "Loops the Playback Head on the current frame to pause
the movie. All interactive elements in the frame continue to
function.  Drag to a sprite or frame in the script channel.  No
parameters."
end
```

Description:
Makes Director loop on the current frame and doesn't go to the next frame.

Appendix C

Picture Listing

(Partial Listing)

We have applied some effects on most of the pictures listed here, so the pictures may differ from those found on the CD. The edited pictures can be found on D:\project.

1. Buttons

Text	Corel Gallery
Pause	Corel Gallery
Play/Rewind	Corel Gallery
Exit	Corel Gallery
Back	Microsoft Office Clipart
Menu Option	3D

2. Background

Intro	Cut from Main Office
Main Office	Corel Gallery, PhotoImpact
Model Window	Corel Gallery
Background	Corel Gallery
Border	PhotoImpact

3. Character

Body	Master Clip
Opened eyes	Master Clip
Closed eyes	Master Clip
Mouth	Master Clip

4. Motherboard:

286	Digital Camera

386	Digital Camera
486	Digital Camera
Pentium	Digital Camera

5. Opening Case:

Cover	Digital Camera
Inside	Digital Camera

6. CPU:

Inside	Digital Camera
Ball	Crystal 3D IMPACT! Pro

7. Buses:

Expansion Slots	http://web.syr.edu/~asalmukb
Inside	Digital Camera

8. Parallel and Serial Ports:

Back of the Computer	CD
Port	Digital Camera

9. Text Window:

Border	PhotoImpact
Background	Director

10. Memory:

SIMM	CD
Banking	http://web.syr.edu/~asalmukb
Ball	Director

11. Tool Tip:

Tool Tip Box	Director

Bibliography

- Weinschenk S., Pamela J., Sarah C., *"GUI Design Essentials"*, Wiley Computer Publishing, 1997.

- McCormick, John A., *"Create Your Own Multimedia System"*, Windcrest / McGraw-Hill, 1995.

- Jonathan Bacon, Deborah Miller, and Kurt Cagle, *"Director 6 Bible"*, IDG Books, 1998.

- Brooks, Charles, *"A+ Certification Concepts and Practice"*, Second Edition, Que E & T, 1999.

- Brooks, Charles, *"A+ Certification Concepts and Practice – Lab Guide"*, Second Edition, Que E & T, 1999.

- Stephen M. Alessi and Stanley R. Trollip, *"Computer Based-Instructoin Methods and Development"*, Second Edition, Prentice-Hall, 1991.

- Saxton, Clint, *"A Guide to Managing and Maintaining Your PC – Lab Manual"*, Second Edition, Course Technology, 1999.

- Meyers, Michael, *"A+ Certification Exam Guide"*, McGraw-Hill, 1998.

- Minasi, Mark, *"The Complete PC Upgrading & Maintainance Guide"*, Eighth Edition, Sybex, 1996.

- Andrews, Jean, *"A+ Exam Prep"*, Coriolis, 1997.
- Steinhauer, Luren, *"Director 6 Studio Skills"*, Hayden Books, 1997.
- http://www.course.com/