

**A MULTIMEDIA SYSTEM FOR LEARNING
NATURAL FORESTRY KNOWLEDGE**

By

SHUIYU ZHANG

Bachelor of Science
Shanxi Agriculture University
Shanxi, China
1983

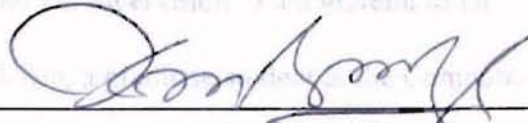
Master of Science
Taiyuan Technology University
Shanxi, China
1989

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 1997

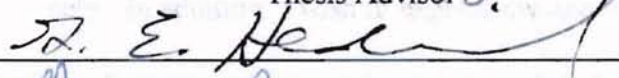
A MULTIMEDIA SYSTEM FOR LEARNING

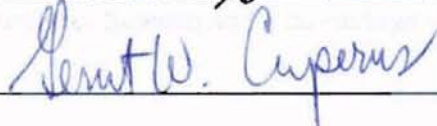
NATURAL FORESTRY KNOWLEDGE

Thesis Approved:



Thesis Adviser







Dean of the Graduate College

ACKNOWLEDGMENTS

I am deeply indebted to my adviser, Dr. K.M. George, for his sound guidance, constant and enthusiastic suggestions and for his infinite patience that have made this study an extremely value experience for me.

I extend a sincere appreciation to Dr. Hedrick and Dr. G.W. Cuperus for their helpful suggestions and patient instructions. I am deeply thankful and give my sincere appreciation to Dr. G.W. Cuperus for his financial support and help while I worked as a research assistant under his supervision. I am grateful to Dr. S. Anderson, Forestry department, and Xia Chen, a graduate student in the Computer Science Department, for their suggestions and help. In addition, I wish to express my appreciation to the faculty and staff in the Computer Science and Entomology departments for their helps and friendship.

Finally, special thanks and appreciation go to my husband, Qin Zhao, and my daughter, Danfeng Zhao, for their love, understanding and encouragement through my study.

A special word is addressed to my father, mother, and my sister and brother for their encouragement.

TABLE OF CONTENTS

Chapter	Page
I. INTRODUCTION	1
1.1 General Statement of the Problem	1
1.2 Project Objective.....	2
II. RELATED WORK.....	3
2.1 Natural Forestry Knowledge Software Version 1.0.....	3
2.1.1 Brief Description of NFKS Version 1.0.....	3
2.1.2 NFKS Version Design via Knowledge Pro	3
2.2 Multimedia.....	8
2.2.1 Multimedia Overview.....	8
2.2.2 Elements of Multimedia	8
2.2.2.1 Text	9
2.2.2.2 Sound.....	10
2.2.2.3 Static Image Pictures	13
2.2.2.4 Animation	14
2.2.2.5 Video	15
2.2.3 Multimedia Development with Windows.....	17
2.2.3.1 Multimedia Capabilities from Windows 3.0 through Window 95	17
2.2.3.2 The Windows Multimedia Architecture.....	19
2.3 Digital Video Compression Techniques	21
2.4 Windows Multimedia System	23
III NFKS 2.0 SOFTWARE DESIGN.....	24
3.1 Organization of NFKS 2.0.....	24
3.2 Software Design via Visual Basic 4.0	29
3.2.1 Visual Basic Environment.....	29
3.2.2 Software Architecture of NFKS 2.0 via Visual Basic 4.0.....	30
3.3 Software Design via Knowledge Pro for Windows	36
3.3.1 Knowledge Pro for Windows Environment	36
3.3.2 Software Architecture of NFKS 2.0 via Knowledge Pro for Windows	37
3.4 Playing Digital Video	39

Chapter	Page
3.4.1 The Multimedia Control Interface (MCI)	39
3.4.2 The MCI OLE Custom Control.....	40
3.5 Digital Video Capture	44
3.5.1 Storage Space.....	45
3.5.2 Digital Video Capture.....	48
 IV CONCLUSION AND FUTURE WORK.....	 52
APPENDIX.....	57
BIBLIOGRAPHY	63

LIST OF FIGURES

Figure	Page
1. Control Flow Chart for NFKS Version 1.0	4
2. Menu Organization.....	5
3. The Screen for the Menu Window.....	6
4. Top Level Structure of NFKS 1.0	7
5. The Windows Multimedia System Architecture	20
6. How a Multimedia API is Sent to a Device Driver	22
7. Flow Chart for NFKS 2.0.....	26
8. One Screen of Quiz Tree Fact	27
9. The Screen Design of NFKS 2.0.....	28
10. Relationship Between the Different Environments	29
11. The Level Structure of NFKS 2.0.....	31
12. The Structure of Start Main Display Child Window.....	35
13. The Structure of Recycling Products	38
14. Multimedia Control Support by MCI Custom Control.....	40
15. One Screen of Playing the Open Welcome Door Video	41
16. One Screen of Playing the Thank You End Video.....	42
17. Decompression Driver Architecture.....	46

Figure	Page
18. Compression Driver Architecture	47
19. The Process of Conversion of Analog Video into Digital Video Data	49
20. The Interface of Video Capture Tool.....	50

LIST OF TABLES

Table	Page
I Code Segment of Global Variables Definition.....	32
II Code Segment for Error Handling.....	32
III Algorithm for Executing an Item.....	35
IV Algorithm for Executing a Quiz.....	36
V A Topic Code Segment of Knowledge Pro.....	38
VI Code Segment of Visual Basic of Playing the Welcome Video.....	42
VII Scores Data Collected From One Class.....	54
VIII Analysis of Data in TABLE VII.....	55
IX Code Segment of One Child From_Load Function.....	57
X Code Segment for Video Capture Generic Interface.....	60

CHAPTER I

INTRODUCTION

1.1 General Statement of the Problem

This thesis describes a PC based multimedia educational program in the forestry knowledge. The purpose of the thesis is to develop a multimedia application software to teach elementary school students forestry knowledge based on the Natural Forestry Knowledge Software Version 1.0 (NFKS 1.0).

Natural Forestry Knowledge Software version 1.0 is an educational software package designed to introduce forestry knowledge to elementary school students. Xia Chen [3] developed it using Knowledge Pro in 1995. In NFKS version 1.0, she used text, still image graphics (bitmap graphics), and hypertext to help students understand forestry knowledge. NFKS version 1.0 consists of five parts namely "Tree Facts", "Does It Comes From a Tree", "Out U.S. Forests", "Forest Products Recycling", and Urban Forestry". Each part has ten questions.

Multimedia application is the incorporation of text, sound, static graphics image, animation, and video by using computing technology. With recent advances in computing, storage, and communication technology, video has becomes an important element of multimedia computing and communication environments. Especially for NFKS

educational application, video not only provides the users with the more accurate information, but also it can make the information more interesting to the users.

1.2 Project Objective

In order to make the users more interested in the learning process; to provide better understanding of the Natural Forestry Knowledge; and to bring new technology to the users, this thesis upgrades NFKS version 1.0 into NFKS version 2.0 by utilizing state-of-art multimedia technology. Besides text, graphics, hypertext, NFKS 2.0 also implements video to make use of the software more appealing.

This thesis will discuss the work related to NFKS 2.0 in chapter II, NFKS 2.0 software design in chapter III, and the conclusion and future work in chapter IV.

CHAPTER II

RELATED WORK

2.1 Natural Forestry Knowledge Software Version 1.0

2.1.1 Brief Description of NFKS Version 1.0

NFKS version 1.0 is an educational software package. The design follows a computer game based model. A question and answer format is used to introduce knowledge. It is organized into five quizzes. Each quiz consists of ten questions. Each question includes hypertext, and each hypertext enables the user to see the definition of the term. Most hypertexts employ the combination of text and graphics. NFKS version 1.0 software is organized according to the control flow chart shown in Figure 1 [3]. Figure 2 shows the menu organization used for interaction with students [3]. Figure 3 shows the screen of Menu Window.

2.1.2 NFKS Version 1.0 Design via Knowledge Pro

NFKS 1.0 is developed using the Knowledge Pro Language[11]. The Knowledge Pro Language is a language for Windows programming. It combines Object-Oriented Programming (OOP) and list processing capabilities. In the Knowledge Pro Language a

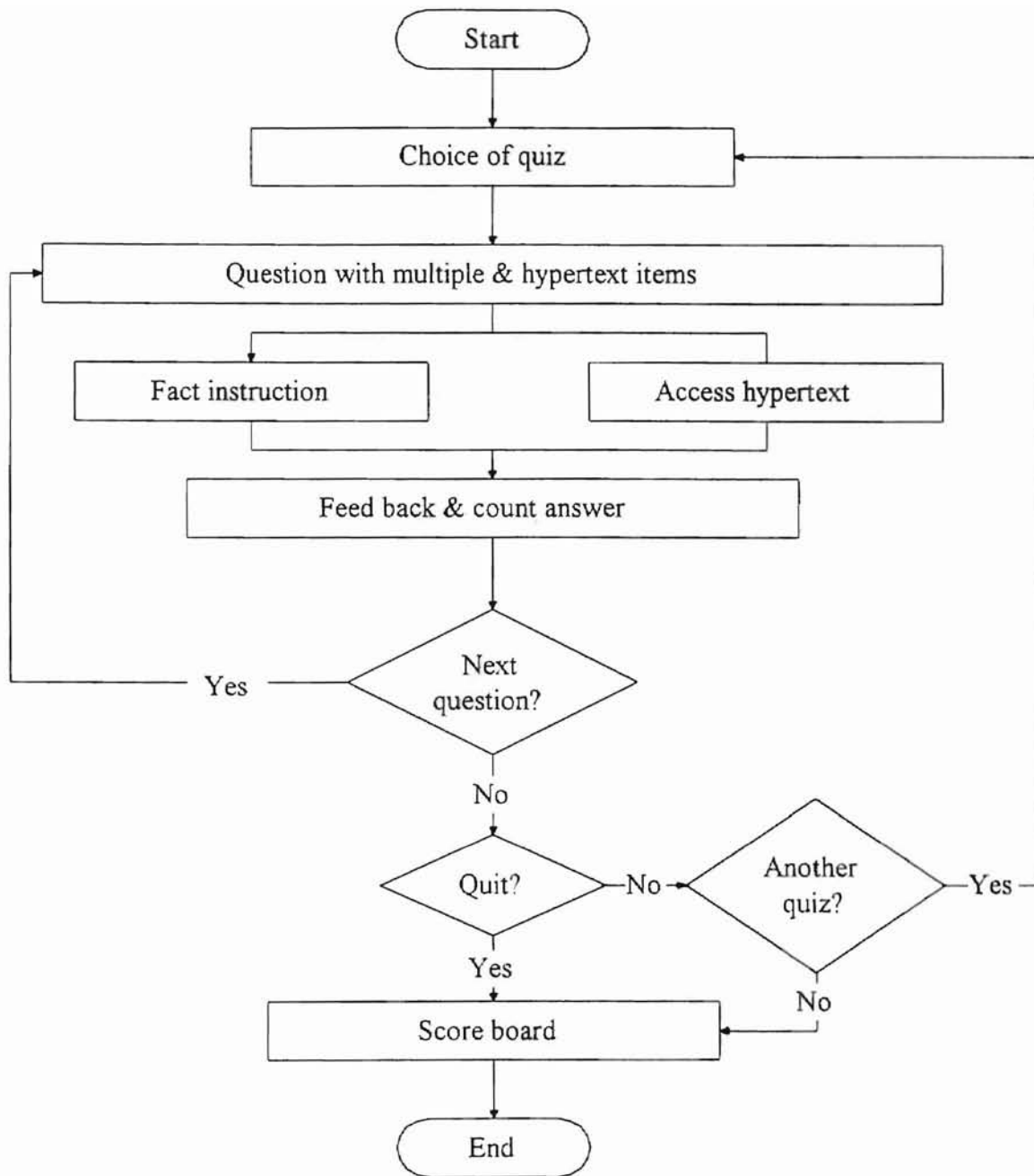


Figure 1 Control Flow Chart for NFKS Version 1.0

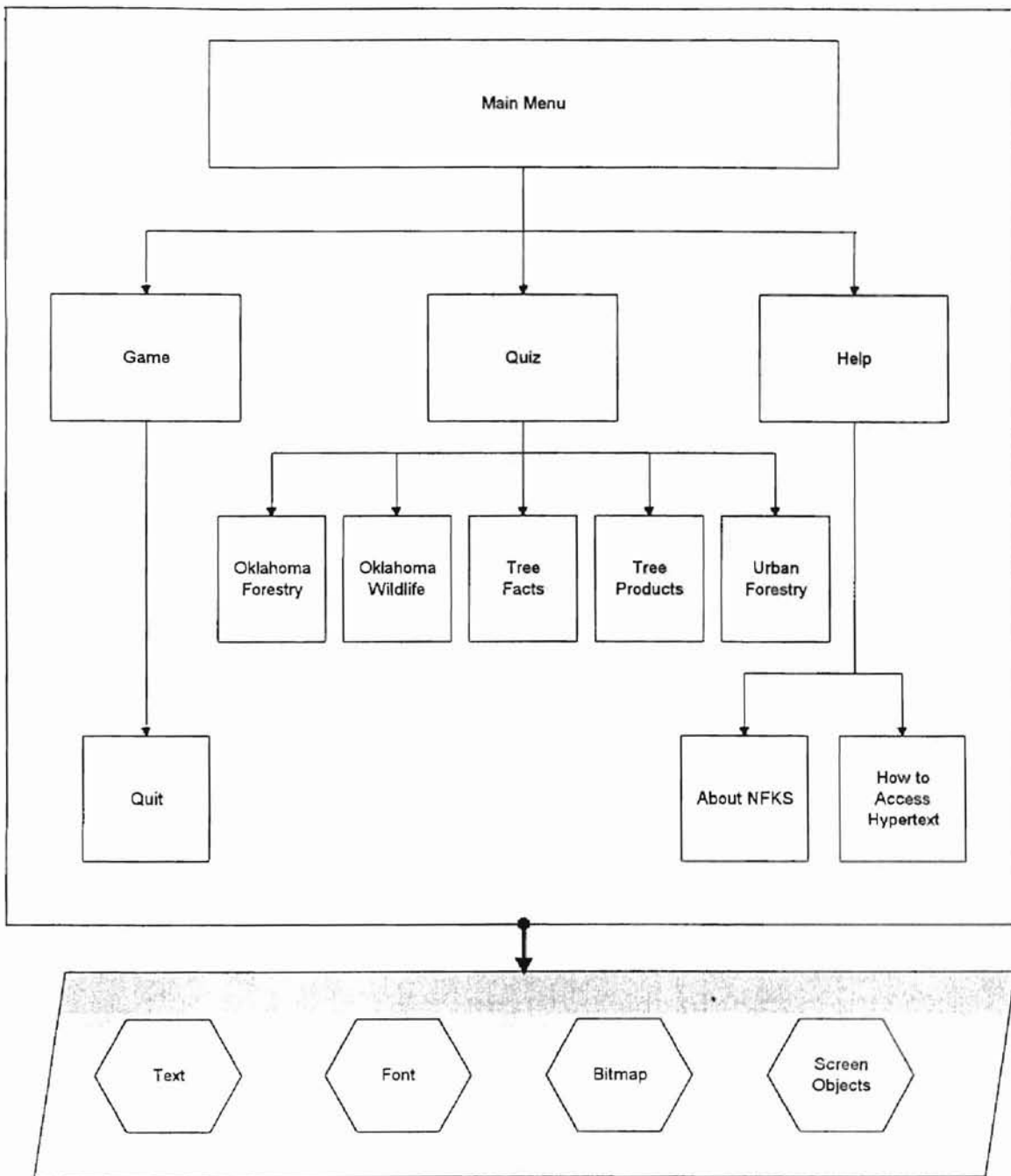


Figure 2 Menu Organization

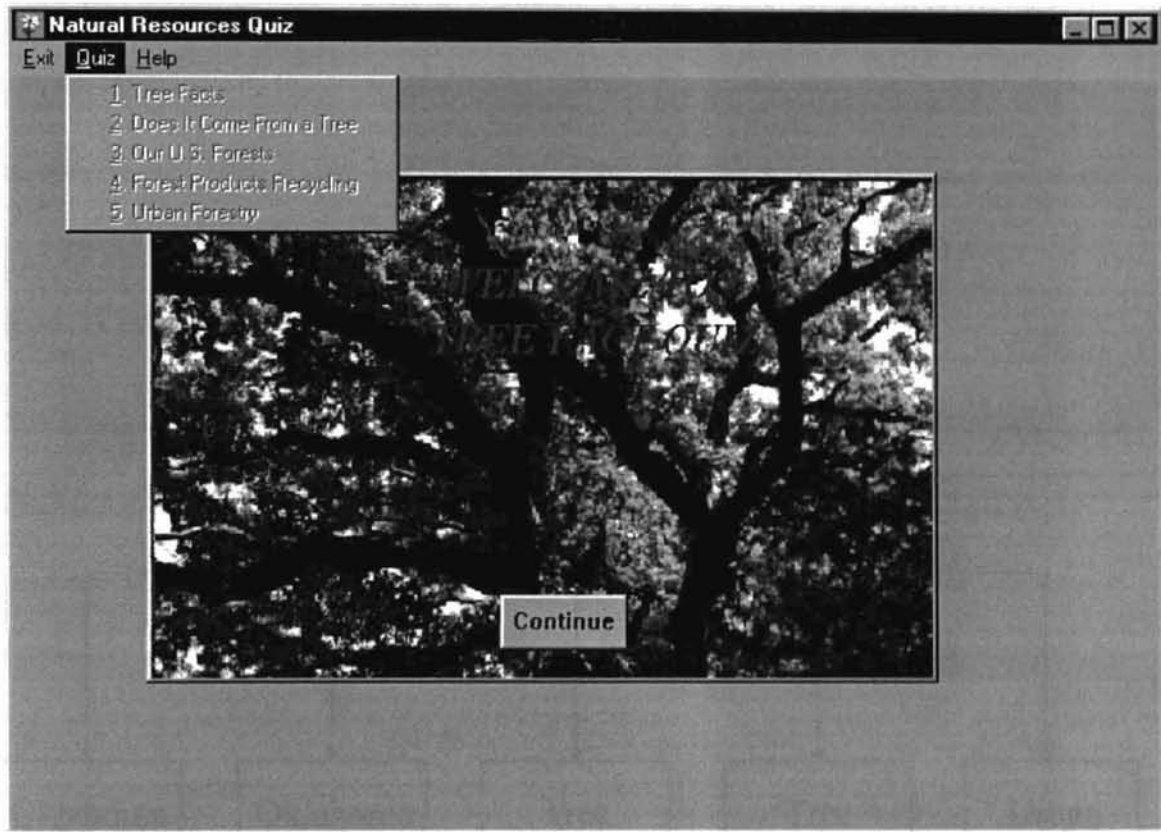


Figure 3 The Screen for the Menu Window

topic is considered to be the basic unit. A topic can be viewed as an object in Object-Oriented Programming (OOP) . Figure 4 shows the Top Level Structure of topics in NFKS 1.0. The five children of main, the root node, are five objects and they correspond to the user interface menu items.

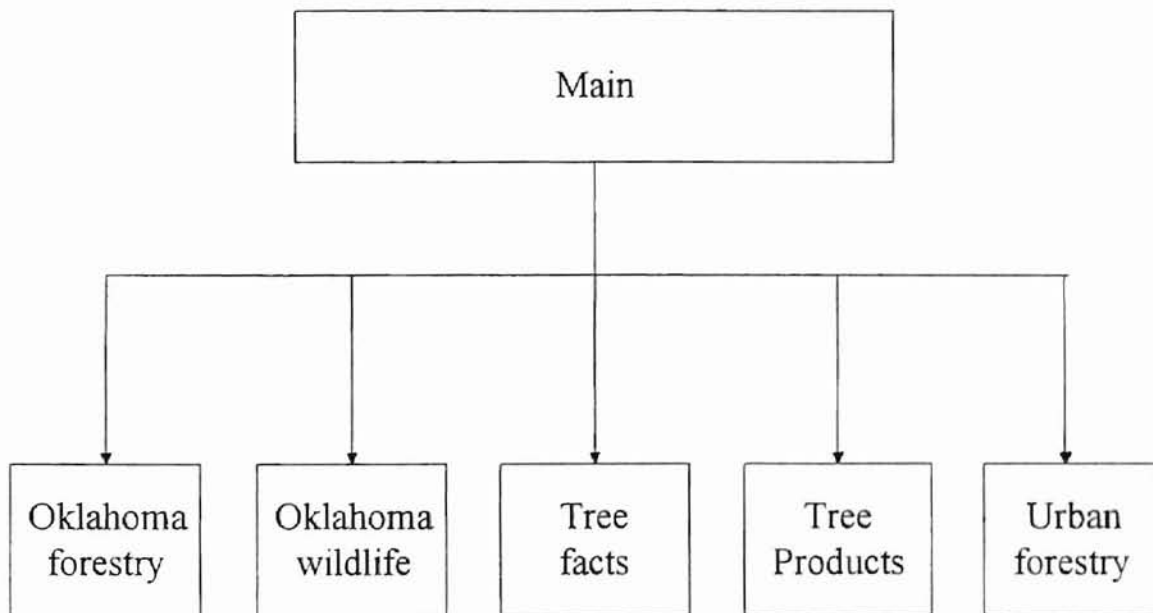


Figure 4 Top Level Structure of NFKS 1.0

2.2 Multimedia

2.2.1 Multimedia Overview

Multimedia has two component parts *multi* and *media*. *Multi* means more than one, and *media* means form of communication. We can say “multimedia is the incorporation of text, sound, static graphics image, animation and full-motion video used in conjunction with computing technology [8]”. Multimedia may use any or all of these aspects of communication.

The Multimedia concept has been around for several years. With the development of computer systems, multimedia is playing an increasingly important role in today’s computer world[19]. Today multimedia is changing the way people are using computers. Multimedia enhances people’s ability to deal with information by allowing more than one media type[10]. For example, instead of displaying alphanumeric characters, several media such as text, sound, static graphics images, animation and video are combined all together to create effective presentations[18].

2.2.2 Elements of Multimedia

The definition of multimedia in the above section emphasized that the main elements of multimedia are: text, sound, static graphics images, animation, video. Each of these elements is described in more detail below.

2.2.2.1 Text

Text is the basic element and most commonly used in multimedia systems. It provides an effective way to communicate ideas and instructions to users. There are four kinds of text.

Printed Text

Printed text in a multimedia application is in the form a machine can read. In order to use printed text in a multimedia application, generally, we type the text into a word processor or text editor. But typing is boring and needs much more time than scanning. If the text previously existed such as in a book or magazine, then scanning the text is a more effective way to enter text.

Scanned Text

All scanners even the low-cost scanners can read printed texts and convert them into machine readable form. People can use scanners to scan the text from books or magazines and directly use the scanned text into their documents.

Electronic Text

Most texts can be read by a computer and transmitted over networks, so we call them electronic texts. Electronic texts are the most commonly used form today because almost everyone who writes books or publishes manuscripts today use word processing and electric publishing equipment [6].

Hypertext

Hypertext is a word first used by Ted Nelson [20]. “It refers to text that has been linked.” [6] When the user view hypertext and click a word linked, the computer executes the object which has been linked. For example, in the “Natural Forestry Knowledge Software (NFKS)” , the user can click on the word in the hypertext window to see more information of it. If a user is already familiar with the term, the user can bypass the hypertext and continue the rest of the game without interruption.

Using the hypertext technology, one can also create hypertext documents in the form of on-line help systems. With the index, user can quickly locate a particular topic, read the information, and then browse related topics.

2.2.2.2 Sound

The integration of audio sound into a multimedia application can provide the user with more accurate information by verbally explaining concepts. Some types of information cannot be described effectively through other methods of communication without using sound, for example, the sound of an animal or the sound of music. Audio sound can also help users understand the information better. Furthermore, it can make the information more interesting to the users.

There are several types of sound objects that may be used in multimedia applications including waveform audio, compact disc (CD) audio, Musical Instrument Digital Interface(MIDI), Netscape Audio Player(NAPlayer), and Real Audio(RA).

Waveform Audio

Waveform is audio that has been converted to a digital format by using waveform audio digitizers [2]. Typically, waveform audio is captured through a soundboard which is connected to a PC[7]. When capturing audio, we connect the sound source to the audio input port on the soundboard . As soon as a recording command is sent to the soundboard, the incoming audio is digitized and stored in a file on the computer's hard disk. A waveform audio digitizer can record any sound one can hear. Every sound has a waveform that describes its frequency, amplitude, and harmonic content.

CD Audio

The CD Audio, often called *red book audio*, is the digital sound format used by standard audio CDs. CD Audio can be played from CD-ROM drive while using Windows. By connecting the audio outputs from the CD drive to the input from the audio card, the users can hear CD audio from the same speakers as waveform audio. Also, CD audio output can be connected directly to a set of speakers.

CD Audio files can be accessed randomly down to an accuracy of 75th of a second, which is fast enough to record any sound audible to humans.[6] Although CD audio sounds better than WAV files, it requires considerably more disk space. So, CD Audio is only used when multimedia application requires top quality sound.

MIDI Audio

MIDI is short form for Musical Instrument Digital Interface. MIDI is a method of saving instrumental keynotes in a computer file (with extension *.mid*). When these files are played back, the instruments represented by the notes are simulated. MIDI files are played back through internal or external sequences. A sequence interprets the control events, which are stored in the MIDI file. MIDI supplies a very efficient way to record the performance information which is required to play music, For example, we can turn notes on and off, make them loud or soft, change their timbre or tone quality, and modify them or add other effects by using MIDI. “They can be randomly accessed down to the accuracy of 128th of a second [6]”. They require less space than waveform audio and CD audio files.

NAPlayer

The NAPlayer is an MS Windows 3.1 or Windows NT 3.5 application which plays audio files of type AU and AIFF. AU files are sound files originally produced for consumption by Sun and NeXT computers. AIFF files are sound files originally produced for consumption by Mac and SGI computers.

NAPLAYER is invoked in one of two ways:

1. Started by the user from an icon.
2. Started by another program with an audio filename specified during startup.

RA client-server architecture (C/S) for execution. Vector graphics are used when

the image is not to be modified often.

Before RealAudio the Web was silent. The RealAudio client-server architecture was designed for reliable, pragmatic delivery over the existing infrastructure. Progressive Networks' goal was to provide the best possible experience of streaming audio over the existing infrastructure.

2.2.2.3 Static Image Pictures

Static image pictures are static representations on-screen. They can be drawings, scanned photographs or line art, charts, maps and graphs.

The images used in multimedia applications are either bitmaps or vector graphics.

Bitmaps

Bitmap images consist of sets of bits in the computer's memory that define the color and intensity of each pixel in an image [16]. Even though there are a lot of different image file formats in Windows applications, the standard bitmaps are the recommended formats for using images in multimedia Windows applications[13].

Vector

Vector graphics are images that are stored as a set of drawing instructions instead of pixel information about the images [14]. MS Windows metafile format is vector graphics files. When the file is read, each instruction is interpreted and sent to the

Windows Graphics Device Interface(GDI) for execution. Vector graphics are used when the sections or dimensions of an image must be modified often.

2.2.2.4 Animation

Animation is a series of images that are displayed in rapid succession, giving viewers the impression that it is moving. By adding animation to an application, one can make complex ideas appear simple and make them easier to understand. There are two kinds of animation, *frame animation* and *cast animation*.

Frame Animation

“Frame animation is the process of designing a separate frame for each screen view then flipping through the frames in rapid succession [16]”. Frame animation is very similar to some digital video technologies. Each frame can be edited as a unique unit because it represents an actual picture in a specified time period[1]. Animated cartoons are examples of frame animation.

Cast Animation

“Cast animation is the process of designing all the moving objects in a separate process, then assigning character traits to each object” [16]. The character traits are the position pattern, size, and color of the objects. Cast animation is strictly time-based. Each

frame is a time-slice that defines the usage and position of each element in the animation.

Cast animation shows all cast members relative to each other.

Animation files depend on the player used to run the animated sequences in Windows. The animation tool that creates the animated sequence determines the player to use at runtime.

2.2.2.5 Video

A few years ago, it was not possible for video to be displayed on a low-priced microcomputer. The required hardware was expensive, and the amounts of storage for a video clip file was big. Now, due to falling hardware prices, larger storage capacities, and more advanced data compression techniques, video playback capabilities are becoming standard on the new PCs.

There are four types of video that can be used as the objects of links in multimedia application: live video feeds, videotape, videodisc, and digital video.

Live Video Feeds

Live video feeds can supply interesting real-time objects of multimedia links. [6] Any television channel or live camera recorder feed can be the objects of a link. For example, if you are a supermarket supervisor needing to inspect what is happening on your store floor, a mouse click can immediately display the store floor on your multimedia computer.

Videotape

Videotape is the most common video medium. Videotape can work as the object of multimedia links. However, most videotapes are linear and not computer controllable. In order to use videotape in a multimedia presentation, you must press the Play, Stop, Fast Forward, and Rewind button yourself. However, the new generation of Sony Hi8 videotape players is computer controllable. Sony VISCA protocol can translate commands from a computer onto codes that control a wide range of Sony video products.[6]

Videodisc

There are two kinds of formats for videodiscs: CAV and CLV. CAV discs are able to store up to 54,000 still frames or 30 minutes of motion video with a stereo sound track [6]. The CAV format lets you display still images and play motion video. CLV discs can store up to an hour of video on each disc side [6]. However, you can not display still frames from CLV discs without using an expensive player.

Because of its fast random access and minimal occupation of the multimedia computer's resources, video disc has become one of the most popular means of providing video to multimedia applications[12].

Digital Video

Digital video is the most popular video storage medium. Most common digital video files are Waveform files. NFKS 2.0 uses waveform files as digital video files.

In this thesis, the use of the term “video” refers to digital video.

Digital video differs from animation. Digital video describes images of real events stored in a digital form, animation is image generated by computer. Video image files usually are larger and need much more storage space than animation images. This prevents digital video files from being loaded and played back from memory. So digital video must be accessed from a hard disk or CD-ROM.

The problem with video is that it occupies too much storage. To make video manageable, video files must be compressed using special software.

2.2.3 Multimedia Development with Windows

2.2.3.1 Multimedia Capabilities from Windows 3.0 through Windows 95

Windows 3.0 was the first PC operating system with graphical interface[5]. Even though Windows 3.0 Software Development Kits (SDK) had no built-in multimedia capabilities, Windows 3.0's graphical interface and multitasking capabilities allowed users to develop advanced graphics, presentation, and drawing capabilities[9]. In late 1991, Microsoft released Windows multimedia extensions and the Windows Multimedia Development Kit. They were a superset of the Windows 3.0 programming environment.

Windows 3.1 is a 16-bit operating system with cooperative multitasking capabilities. Windows NT and Windows 95 each consist of a 32-bit operating system with preemptive multitasking capabilities. Windows 95 offers the following capabilities:

- Plug and play support for multimedia devices

Plug and play is a standard process that automatically adds new capabilities to the PC or changing Personal Computer Memory Card International Association adapters in notebook computers. With plug and play, the computer detects, manages and configures the hardware automatically.

- A new 32-bit CD file system (CDFS)

With a new 32-bit CDFS significantly enhances the performance of CD-ROM.

- Improved MIDI support

This allows transmission of complex MIDI messages.

- New audio compression encode /decode

Windows 95 has improved the audio codec (compression encode/decode algorithm) and allows 4:1 compression ratio for music-quality audio without noticeable loss.

- Built-in digital video support

Video for Windows is a standard component for Windows 95.

- Display Control Interface

This allows applications to completely control VLB (Video Local Bus) and PCT (Peripheral Control Interface) interface boards.

- Media Control Interface (MCI) device drives

MCI device drivers can control many of the popular multimedia devices on the market today. CD audio device, MIDI sequences, motion video device, VCRs, wave audio devices, etc.

2.2.3.2 The Windows Multimedia Architecture

The architects of the Windows multimedia system achieved the specified design goals by adding a new .DLL(Dynamic Link Library) to Windows - MMSYSTEM.DLL for Windows 3.1 and WINWW.DLL for Windows 95[17]. Figure 5 shows the Windows Multimedia System Architecture.

MMSYSTEM.DLL or WINWW.DLL is a translation layer between the application code and device driver code (see Figure 6). MMSYSTEM.DLL or WINWW.DLL contains all the multimedia application program interfaces (APIs) that the Windows multimedia system provides and the low-level multimedia support functions.

These functions provide support for the following devices:

- Waveform audio,
- MIDI audio,
- Low-level movie player support,
- Auxiliary devices,
- Multimedia timer services, and
- Joystick support.

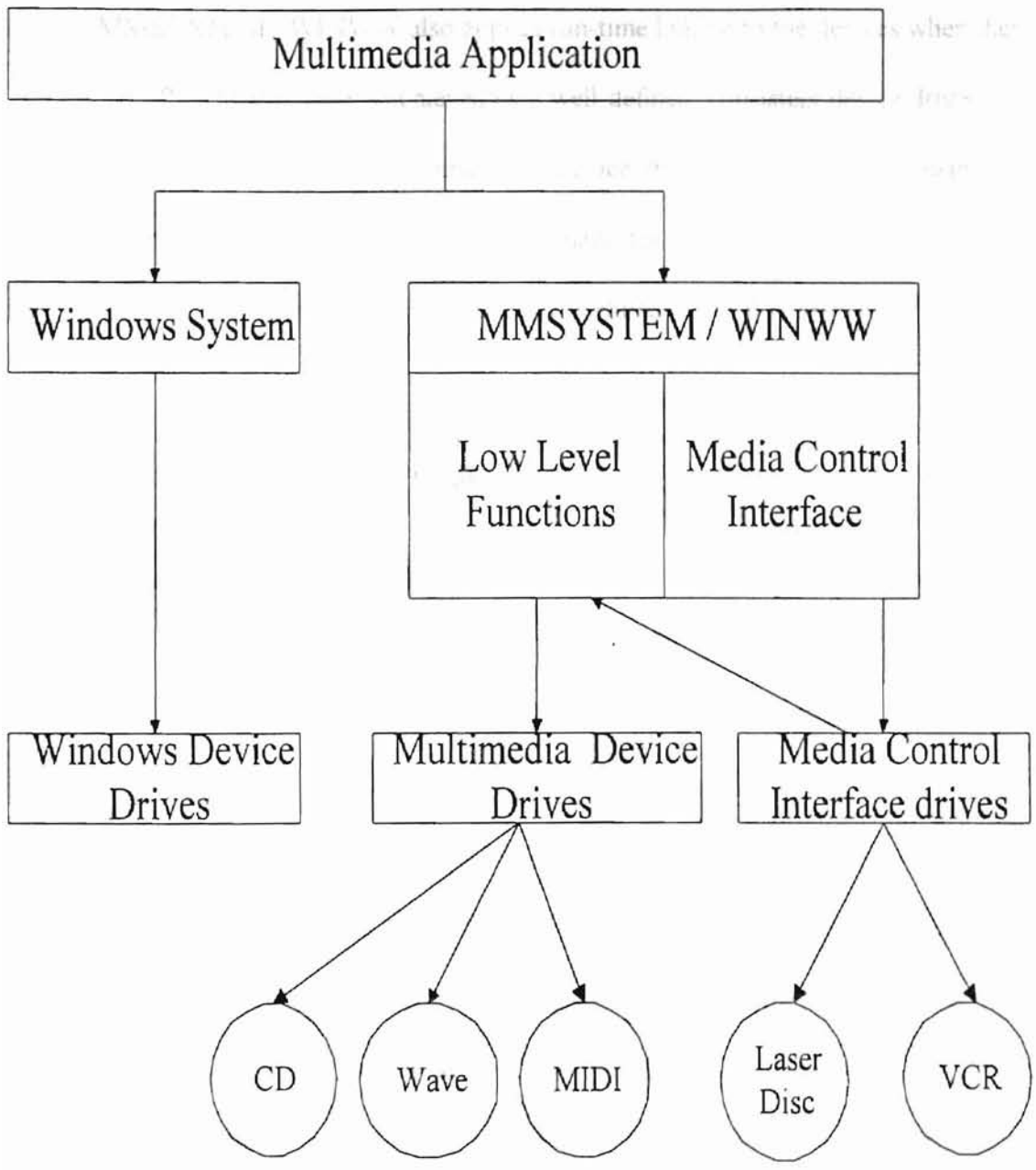


Figure 5 The Windows Multimedia System Architecture

MMSYSTEM / WINWW also applies run-time linking to the devices when they are needed. Run-time linking is achieved by a well-defined, consistent device driver interface. Microsoft has defined an installable device driver interface. By following the defined interface, the software developers can guarantee that their applications are device-independent. The installable drivers' interface is defined in the Windows Multimedia Development Kit (MDK) [9].

2.3 Digital Video Compression Techniques

There are two types of compression techniques: interframe compression and intraframe compression.

“The key frame is a reference point used for marking the changes or deltas that occur in subsequent frames until the next key frame is captured [4].” Intraframe compression is applied on each individual frame independently. Most codecs (compression / decompression) use both techniques. This means the key frame is compressed individually and only the changes between the frames are compressed.

Both interframe and intraframe compression can be either *lossless* or *lossy* compression. Lossless compression allows all compressed data to be recovered after decompression. This kind of compression has only a small compression ratio, typically 2:1[16]. This is not appropriate for the applications described in this thesis. Lossy compression provides much higher compression ratio. Lossy compression is based on complex algorithms that discard unimportant data during compression. This enables the user to set compression ratios at arbitrary levels. The higher the compression ratio, the smaller the storage requirement and the poorer quality of the video.

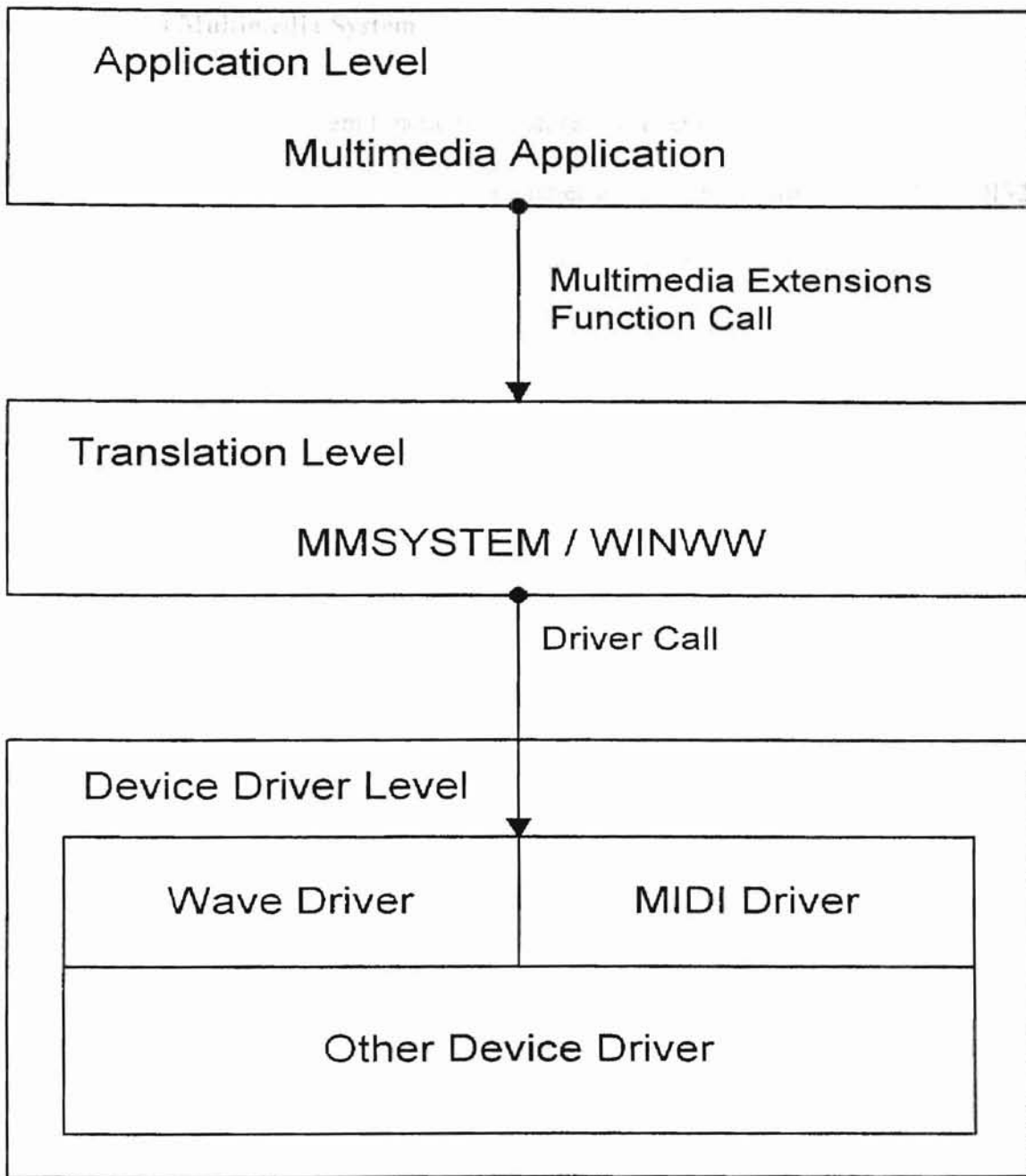


Figure 6 How a Multimedia API is Sent to a Device Driver (referenced from [15])

2.4 Windows Multimedia System

The Multimedia System functions are stored in several dynamic link libraries, or DLLs. In Windows 95 operating system, multimedia has three libraries: Kernel32, GDI32 (Graphic Device Interface), and User32. These libraries contain about 1,000 functions.

The multimedia system provides an interface for the following services:

- WAV audio playback and recording
- Animation playback
- Video Playback
- Joystick services
- Operating of external media devices [15].

CHAPTER III

NFKS 2.0 SOFTWARE DESIGN

3.1 Organization of NFKS 2.0

NFKS 2.0 has two main parts: “*Study by items*” and “*Study by quizzes*”. *Study by items* contains forty seven items. Most items have some hypertext. Each hypertext combines the text and images together and gives users more detailed information than pure text. Some items display vivid video and audio, this makes the users to understand better and be more involved in the display. *Study by quizzes* has five quizzes. Each quiz has ten questions. *Study by quizzes* is designed based on NFKS 1.0. The flow-chart given in figure 7 shows the organization of NFKS 2.0.

NFKS 2.0 employs several different media: hypertext, video, audio, images, and text. When starting NFKS 2.0, the full screen *welcome* video will be played. The door of NFKS 2.0 is being opened accompanied with the music and “welcome to the study Natural Forestry Knowledge” sound. When the Exit of NFKS 2.0 is chosen, the full screen *thank you* video is played. The door of NFKS 2.0 is being closed accompanied with the music and “Thank you very much, Have a nice day!” sound. When the “Study by quizzes” is chosen, the user can chose one of the five quizzes. Each question in the quiz has five answer choices. After the user answers a question, the description window pops up to show the correct answer with a short description (see Figure 8). With each

question, the top left corner of the window has a hypertext window. It shows links to several hypertexts (see Figure 8). The user can click the hypertext link to learn more information about the items showed in the hypertext window. When the user chooses the “Study by items”, a combo box (see Figure 9) shows the complete list. The user can select any one of the items. When items that include video clips are chosen, the 160X120 video window pops up, the bottom of this windows shows several button controls, so the video can replay, stop, rewind, and fastforward at any time.

NFKS 2.0 is written in Visual Basic 4.0 and Knowledge Pro for Windows. The user interface uses Visual Basic 4.0. Figure 9 shows the screen design of NFKS 2.0. Two combo boxes (see Figure 9) are used to list the 47 items and 5 quizzes. When the items or quiz is launched (the command button “Display” under “Study by items” or “Study by quizzes” is clicked), the appropriate Knowledge Pro for Windows program is linked to it.

In order to play the video files, OLE Custom Control links the Media Control Interface (MCI) and gives the users multiple choices to play the video and audio interleaved files (.avi files).

Figure 10 shows the relationship between Visual Basic, Knowledge Pro, and OLE Custom Control.

In NFKS 2.0, some video and audio interleaved files are captured directly from the VCR by interface software. We will discuss the details of this software in Section 3.5.

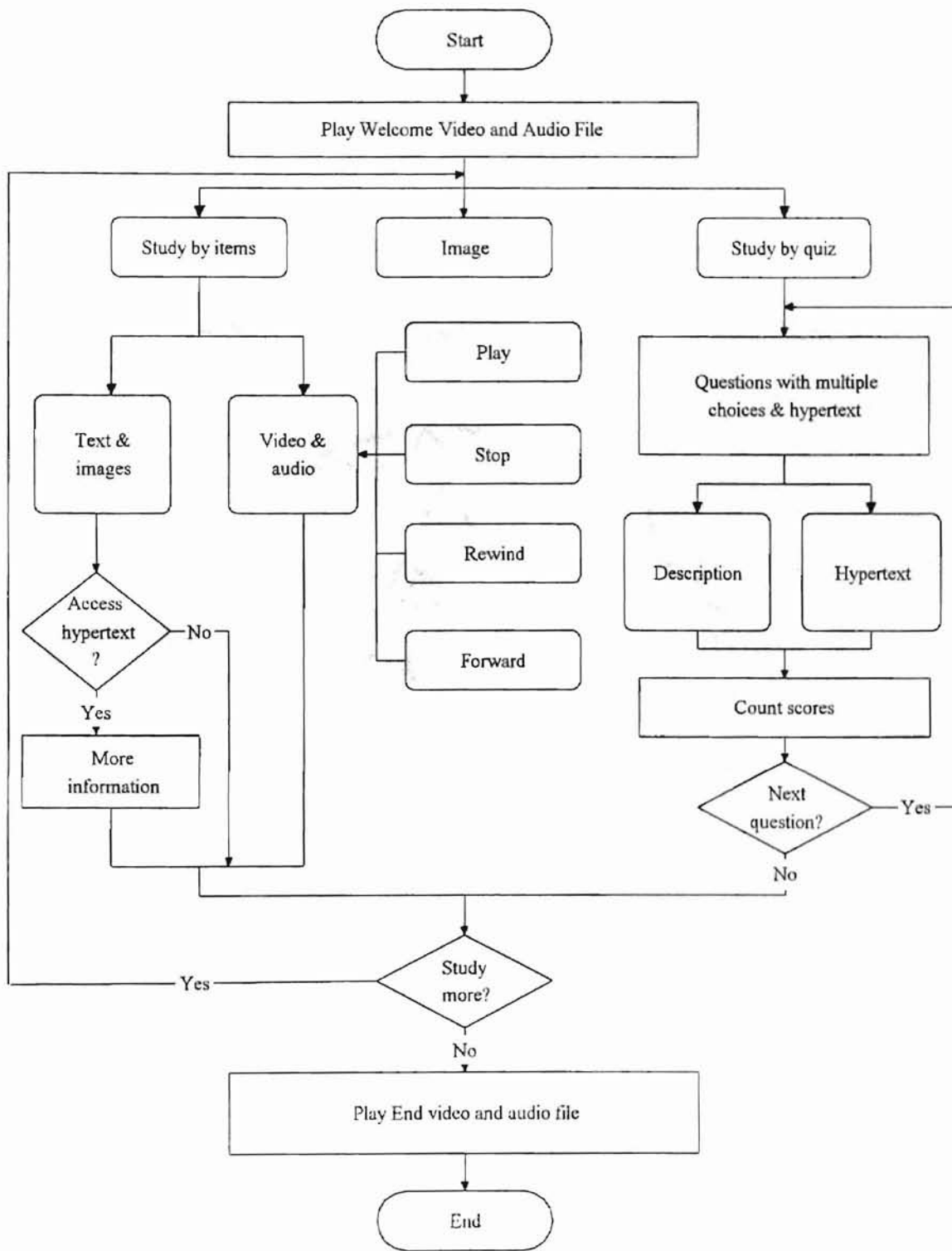


Figure 7 Flow Chart for NFKS 2.0

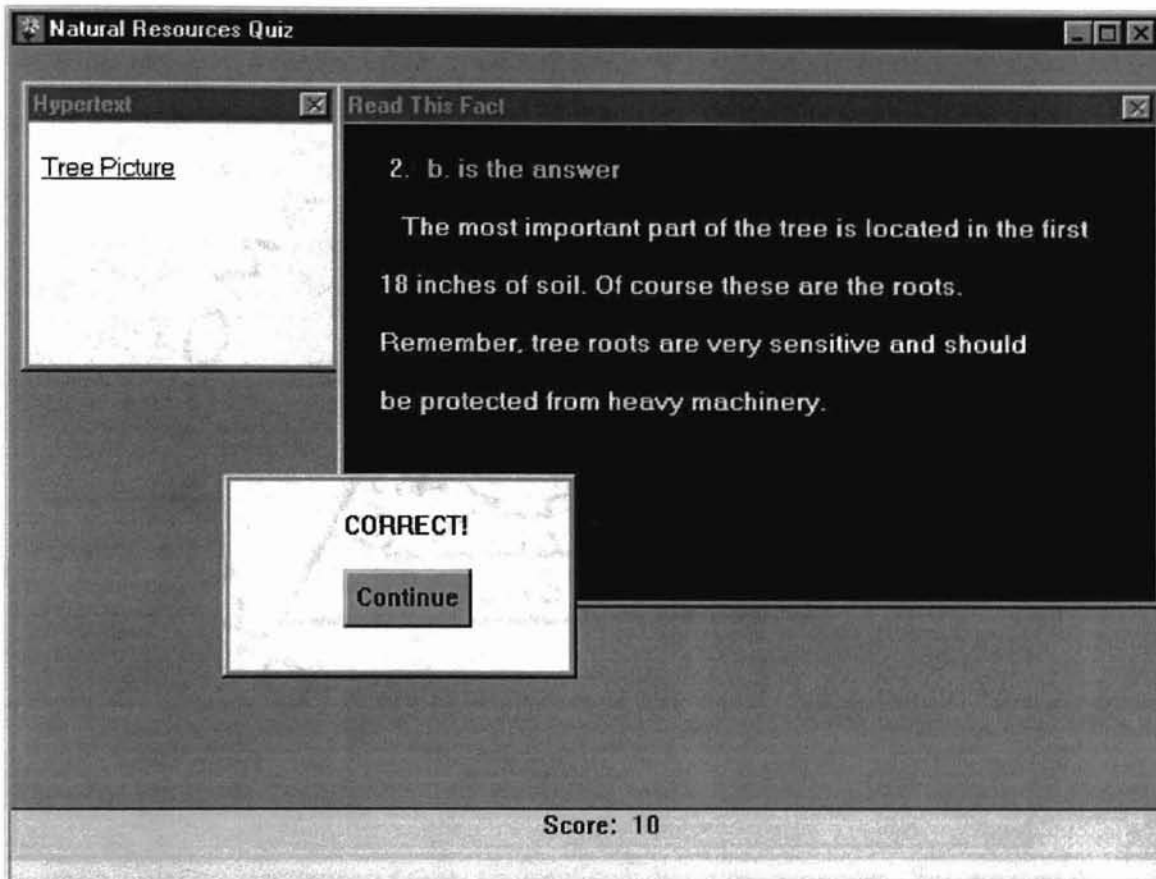


Figure 8 One Screen of Quiz Tree Fact

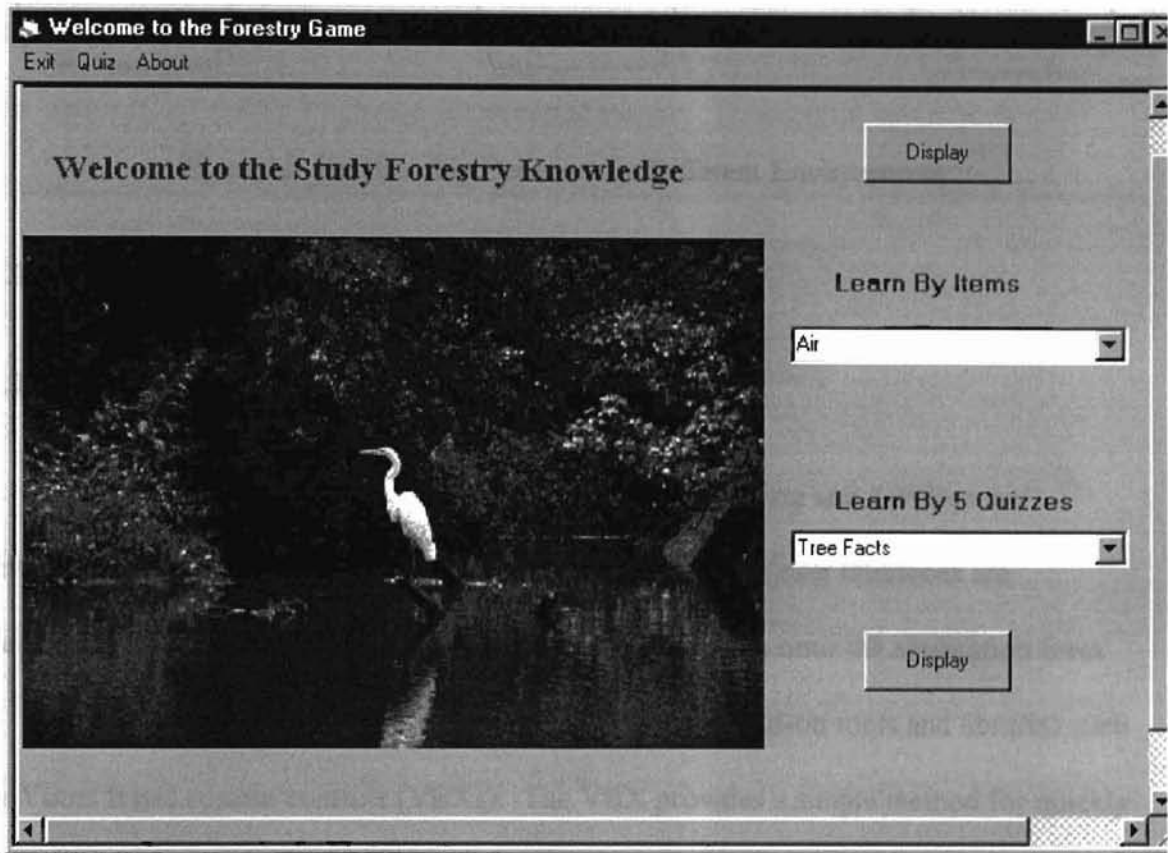


Figure 9 The Screen Design of NFKS 2.0

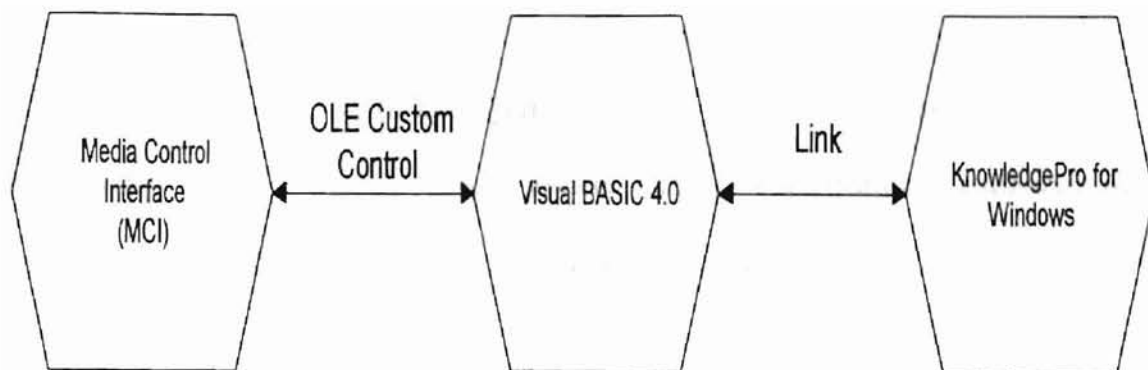


Figure 10 Relationship Between the Different Environments

3.2 Software Design via Visual Basic 4.0

3.2.1 Visual Basic Environment

Visual Basic is Microsoft's powerful visual programming and development environment for developing multimedia applications[12]. The user interfaces are developed by dragging objects from the Visual Basic Toolbox onto the application form.

Visual Basic development can be extended through add-on tools and libraries such as Visual Basic custom controls (VBXs). The VBX provides a simple method for quickly and easily adding programming power to the Visual Basic environment. For multimedia programmers, this is very convenient. Today, there are VBXs for image display, image manipulation, overlay video control, etc.. In this thesis, we use CAPWNDX.VBX to capture video.

Visual Basic uses an event-driven model. Visual Basic combines the event-driven nature of Windows with its development environment. When user clicks on a command

button in a form, some program code would be called to perform an operation specific to that event.

NFKS 2.0 requires a total integration of many capabilities such as audio, digital video, video capture, images, peripheral devices, CD-ROM and many other software and hardware technologies. This requires that these features are available to the development environment. Visual Basic 4.0 is such a development environment that is equipped with the power to run on the Windows 95 operating system. Therefore it can provide user-friendly access to all the capabilities of the operating system such as multimedia, and other common controls.

3.2.2 Software Architecture of NFKS 2.0 via Visual Basic 4.0

Form is a basic concept in Visual Basic. A form is actually an object and is the equivalent of a window frame. A form can have a system menu, minimize button, and/or maximize button. All the controls on a form can be modified by the properties. Properties are a list of control options as well as how they are first set. All of a control's properties are set initially during application coding. They also can be modified programmatically during execution. Also, a control has predefined methods.

A method is used to perform some kind of predefined action on the control. For example, every form has a show method to cause the form to be visible to the users and to be active. Along with predefined methods, each control has a series of predefined events. An event is an action to which the control is capable of responding, for example, the user clicking on a control. The form supplies some event procedures. Generally, we can think of a form as a window within which to place controls. NFKS 2.0 implements Multiple

Document Interface (MDI) technology. It contains several child forms (windows) within a container form (the parent window). The parent window has three events on the menu bar structured as in Figure 11. Each event on the menu bar supports one child form (window). NFKS 2.0 defines the global variables and functions in the Module files. Table I shows the code segment of the global defined variables and functions. Table II shows the code segment for error handling.

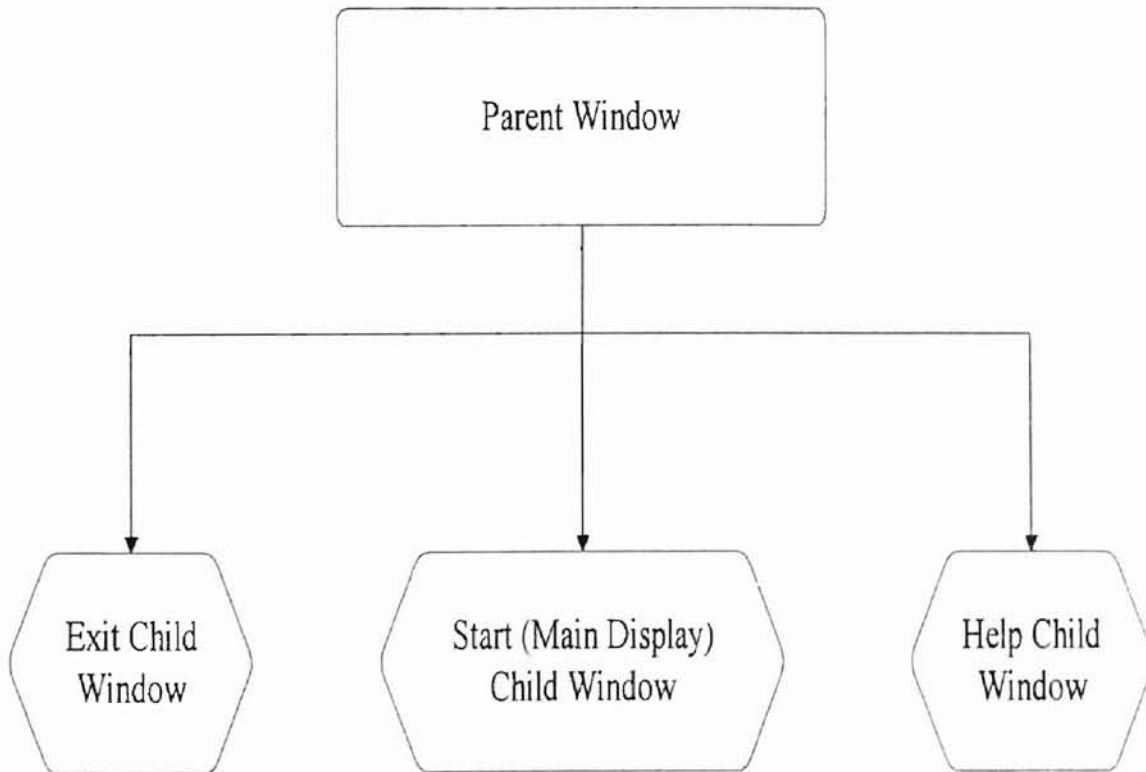


Figure 11 The Level Structure of NFKS 2.0

TABLE I
CODE SEGMENT OF GLOBAL VARIABLES DEFINITION

```
' Public Constants
Public Const conMCIAppTitle = "MCI Control Application"
' These constants are defined in mmsystem.h.
Public Const conMCIErrInvalidDeviceID = 30257
Public Const conMCIErrDeviceOpen = 30263
Public Const conMCIErrCannotLoadDriver = 30266
Public Const conMCIErrUnsupportedFunction = 30274
Public Const conMCIErrInvalidFile = 30304
#If Win32 Then
    Declare Function GetFocus Lib "User32" () As Long
#Else
    Declare Function GetFocus Lib "User" () As Integer
#End If
' Public variables
Public DialogCaption As String
```

TABLE II
CODE SEGMENT FOR ERROR HANBDLING

```
' Force all run-time errors to be handled here.
Sub DisplayErrorMessageBox()
```

TABLE II (Continued)

```
Dim Msg As String
Select Case Err
    Case conMCIErrCannotLoadDriver
        Msg = "Error load media device driver."
    Case conMCIErrDeviceOpen
        Msg = "The device is not open or is not known. "
    Case conMCIErrInvalidDeviceID
        Msg = "Invalid device id."
    Case conMCIErrInvalidFile
        Msg = "Invalid filename."
    Case conMCIErrUnsupportedFunction
        Msg = "Action not available for this device."
    Case Else
        Msg = "Unknown error (" + Str$(Err) + ")."
    End Select
    MsgBox Msg, 48, conMCIAppTitle
End Sub

' This procedure allows any Windows event to be processed.
' This may be necessary to solve any synchronization
' problems with Windows events.
' This procedure can also be used to force a delay in
' processing.
Sub WaitForEventsToFinish(NbrTimes As Integer)
    Dim i As Integer
    For i = 1 To NbrTimes
```

TABLE II (Continued)

dummy% = DoEvents()
Next i
End Sub

The second level of the structure applies to the three child Windows. Compared with the *start* child window, the *exit* child window and *help* child window are easier to structure. We will focus on the description the *start* child window. The *start* on the menu bar supply the main display child window structures as in Figure 12 which is the main part of NFKS 2.0. Table IX in appendix shows the code segment of this child Form-Load function. Table III shows an example code for playing an item. Table IV shows an example code for playing a quiz.

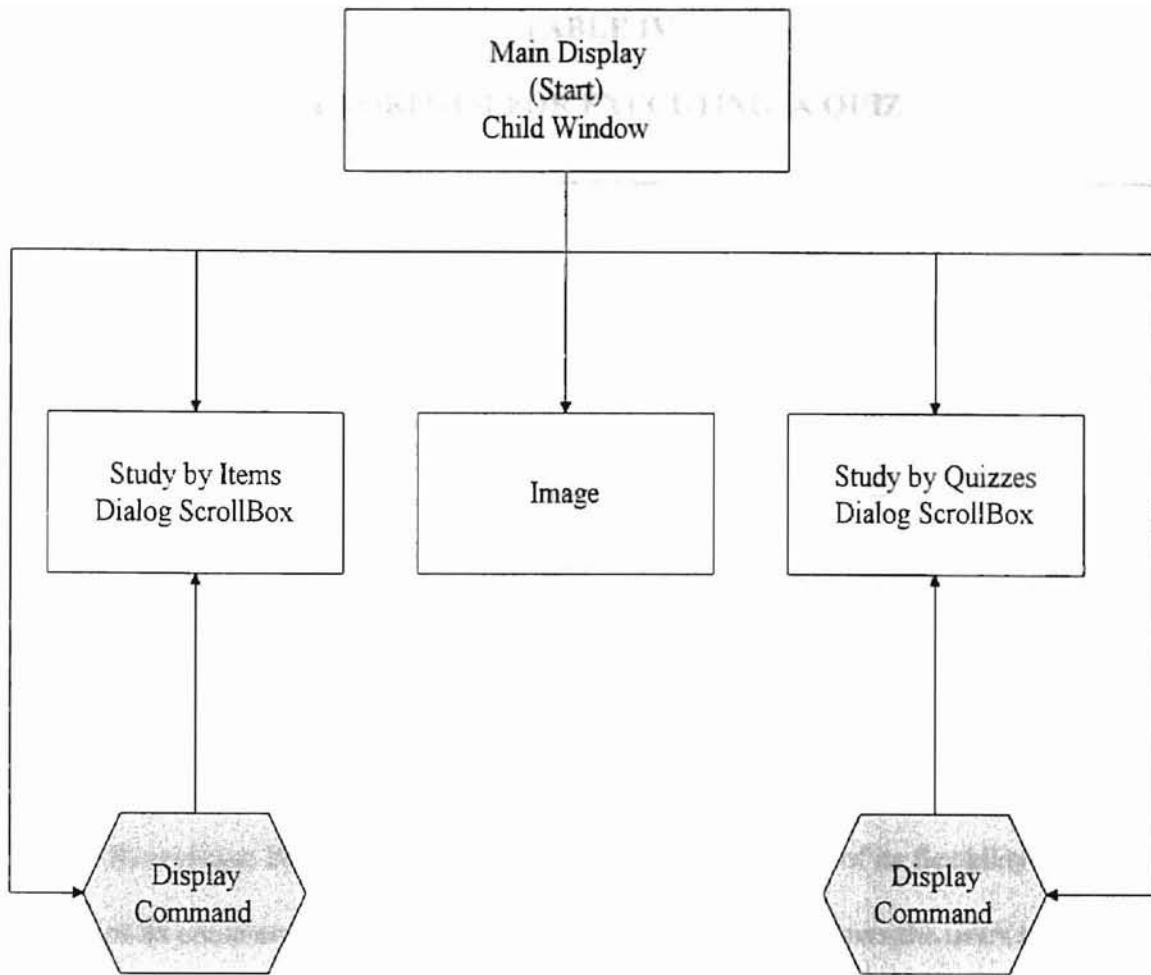


Figure 12 The Structure of Start Main Display Child Window.

TABLE III

ALGORITHM FOR EXECUTING AN ITEM

If Combol.Text = "Air" Then

MyAppID = Shell("C:\kpwing\wasteman\kpwirun.exe C:\thesis\air.ckb", 1)

AppActivate MyAppID ' Activate Knowledge Pro

End If

TABLE IV Code file

ALGORITHM FOR EXECUTING A QUIZ

```
If Combo2.Text = "Tree Facts" Then
MyAppID = Shell("C:\kpwing\wasteman\kpwinrun.exe C:\thesis\quiz1.ckb", 1)
AppActivate MyAppID ' Activate Knowledge Pro.
End If
```

3.3 Software Design via Knowledge Pro for Windows

3.3.1 Knowledge Pro for Windows Environment

Knowledge Pro is a development environment. Because of its flexibility and the power of its combined OOP and list processing capabilities, it allows the users to build complex Windows application rapidly. Knowledge Pro is a high level language because:

It uses an untyped language,

It is interpreted,

It has a simple syntax,

It has large building blocks,

It does not require the programmer to do any memory management, and

It provides intelligent defaults whenever possible[11].

Although the Knowledge Pro language is a high level language, it provides low level access to the Windows Application Program Interface (API) and messages.

Knowledge Pro provides low level functionality because:

It permits a programmer work directly with source code file,

It does not restrict an application to a pre-defined format or paradigm,

It lets application programs directly send and receive all Windows messages,

It provides direct access to API (Application Program Interface) function calls, and

It handles all DDE (Dynamic Data Exchange) messages[11].

3.3.2 Software Architecture of NFKS 2.0 via Knowledge Pro for Windows

Topic is the heart of the Knowledge Pro language[9]. It is an extremely flexible structure which acts like a variable, function, procedure, or object depending on its usage. In NFKS 2.0, each quiz in the “Study by quiz” and some items in the “Study by items” is written by the Knowledge Pro language. In these parts of the program we use topic as the basic unit.

The structure of the five quizzes are identical. Here we give as an example the “Recycling Products” topic (see Figure 13). Each quiz first loads and displays the welcome image. The TABLE V shows the topic code segment for loading and displaying an image.

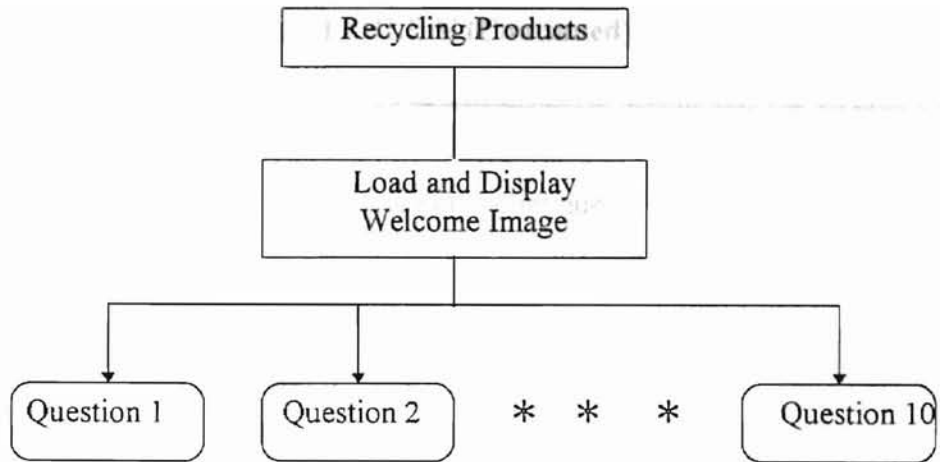


Figure 13 The Structure of Recycling Products

TABLE V
A TOPIC CODE SEGEMENT OF KNOWLEDGE PRO

topic welcome.

disable_menu_item (?m1,[&quiz]).

welcome_window is window(, 10, 5, 71, 22, ,[DialogWindow, Visible]
,?main_window).

welcome_pict is load_bitmap('c:\fquiz\q3bmp\wlcom.bmp').

bitmap(?welcome_pict).

welcome_font is create_font([30, 9, 700, t, f, f, 0, 0, 16, Roman]).

use_font (?welcome_font).

text('#x26#y4#fred WELCOME TO').

text('#x20#y6#fred NATURAL RESOURCES') .

TABLE V (Continued)

```
text('#x32#y8#fred QUIZ').
welcome_continue is button(Continue, continue, 33.5, 20).
set_focus(?welcome_window).
wait().
hide_window(?welcome_window).
do(?instruction).
close_window(?welcome_window).
delete_font(?welcome_font).
delete_bitmap(?welcome_pictr).
enable_menu_item (?m1,[&quiz]).
end. (*welcome*)
```

3.4 Playing Digital Video

3.4.1 The Multimedia Control Interface (MCI)

The MCI provides Windows programs with high-level, device-independent commands for controlling multimedia devices. This means that the command strings can be passed to the Media Control Interface without having to call a lot of functions to access the device. These command strings consist of commands that act on the multimedia device. All MCI commands can be passed to the Media Control Interface with the mciSend String Multimedia Application Program Interface (API) function.

Video is an important way to present information. Using the MCI's digital video interface, you can display video sequences. These video sequences include not only the

video images, but also digitized sound. This can be achieved by storing both types of data in one type of file format called the Audio-Video Interleaved (.AVI) file.

3.4.2 The MCI OLE Custom Control

Visual Basic 4.0 has two MCI OLE custom controllers *mci16.ocx* and *mci32.ocx*. *Mci16.ocx* is for 16-bit operating system and *mci32.ocx* for 32-bit operating system. The MCI OLE custom control provides a visual interface that consists of VCR-type controls such as *play*, *stop*, *fastforward* and *rewind*, etc. (see Figure 14). Actually these buttons in Figure 14 automatically control the MCI. At the beginning and the end of NFKS 2.0, the full screen videos are automatically played. Figure 15 shows one moment of Playing the opening the welcome door video. Figure 16 shows one moment of playing the thank you end video. Table VI shows the Visual Basic program code segment for playing video.



Figure 14 Multimedia Control support by MCI Custom Control

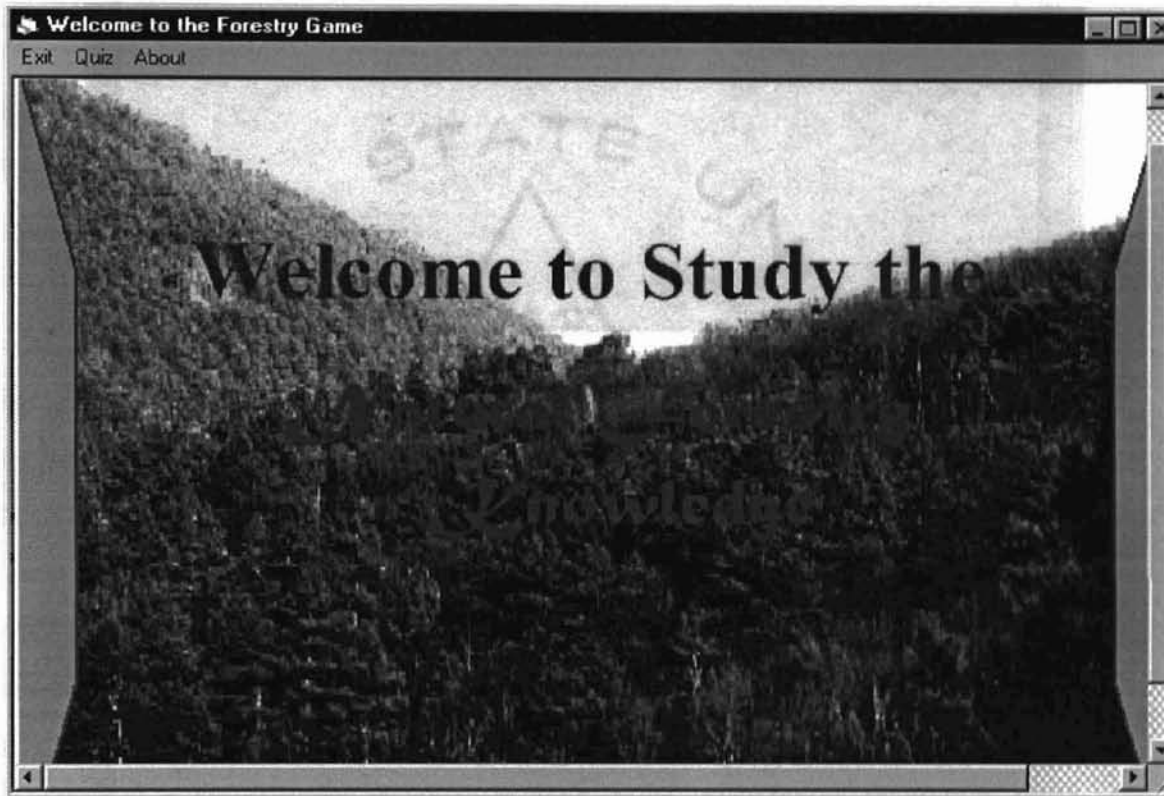


Figure 15 One screen of playing the open welcome door video

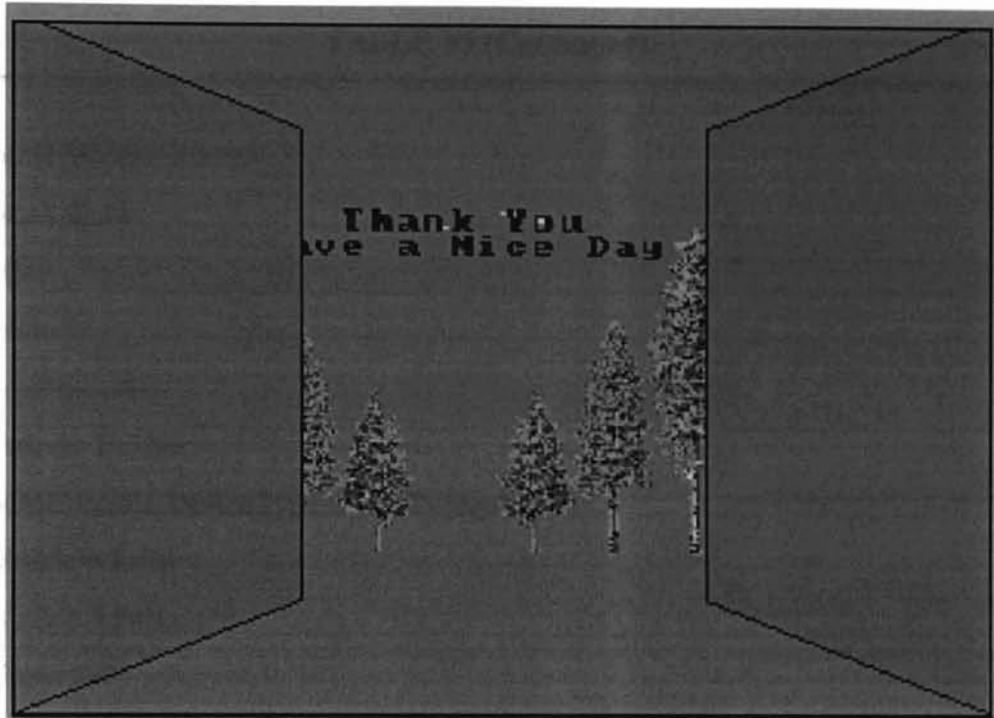


Figure 16 One screen of playing the thank you end video

TABLE VI

CODE SEGEMENT OF VISUAL BASIC OF PLAYING THE WELCOME VIDEO

```
Private Sub MDIForm_Load()  
MDIForm1.Top = 0  
MDIForm1.Left = 0  
MDIForm1.Height = 7000
```

TABLE VI (Continued)

```
MDIForm1.Width = 10000
MDIForm1.Show
Wave.Hide
child2.Hide
'-----
Dim msec As Double
Wave.MMControl1.DeviceType = "AVIVideo"
Wave.Visible = False
child2.Visible = False
Wave.Top = -320
Wave.Left = 0
Wave.Height = 6900
Wave.Width = 10000
child2.Top = -300
child2.Left = 0
child2.Height = 6900
child2.Width = 10000
Wave.MMControl1.UpdateInterval = 0
Wave.MMControl1.hWndDisplay = 0
If Not Wave.MMControl1.Mode = vbMCI ModeNotOpen Then
Wave.MMControl1.Command = "Close"
End If
' Open the device with the new filename.
Wave.MMControl1.filename = "C:\vb4\working\try1.avi"
On Error GoTo MCI_ERROR
Wave.MMControl1.Command = "Open"
Wave.Visible = True
Wave.MMControl1.hWndDisplay = Wave.hWnd
```

TABLE VI (Continued)

```
Wave.MMControl1.Command = "Play"  
On Error GoTo 0  
Caption = "Welcome to the Forestry Game"  
' Set the timing Wave.Labels on the form.  
Wave.MMControl1.TimeFormat = vbMCIFormatMilliseconds  
Wave.Label1.Caption = "0.0"  
msec = (CDBl(Wave.MMControl1.Length) / 1000)  
Wave.Label2.Caption = Format$(msec, "0.00")  
' Set the scrollbar values.  
Wave.Hscroll1.value = 0  
CurrentValue = 0#  
Exit Sub  
MCI_ERROR:  
    DisplayErrorMessageBox  
    Resume MCI_EXIT  
MCI_EXIT:  
    Unload Wave  
End Sub
```

3.5 Digital Video Capture

Before digital video can be played back, they must be captured. This digital video capture tool will capture digital videos and save the video files as .AVI file on the hard disk. Typical .AVI (Audio-Video Interleaved) files contain multiple streams of data, but most files contain only one audio and one video stream.

Before describing how to capture the digital videos, we discuss storage space.

3.5.1 Storage Space

The space required for storing a video digital file is very high. For example, an uncompressed bitmap file with a resolution of 460×480 pixels and with 8 pixel depth (need 8 bits for storing 1 each pixel), with a playback rate of 15 frames per second will require 36,864,000 bytes for 1 second of uncompressed video file with no audio. This is incredible to implement video without compressing. Fortunately, Video for Windows (VFW) supplies the Installable Compression Manager (ICM) to handle the compression and decompression of digital video.

Before video capture begins, the several megabytes capture file is allocated. After the video segment is completely captured, the contents of the file are compressed and stored into a new file in final playback format.

Video compression is completed through Compression and Decompression drives (Codecs). A PC can have many video codecs available. The user can choose which codec to apply when compressing the video file. The Video for Windows playback driver will automatically choose the correct decompressor to decompress when playing back. In this thesis, we use lossy compression techniques to yield a higher compression ratio.

Audio compression is similar to the video compression, but this is done by Audio Compression Manager. Audio capture is handled by a sound card instead of the video capture board. After capturing the audio, combine the audio file with the video file and save them in the AVI file. Figure 17 shows the decompression driver architecture. Figure 18 shows the compression Driver Architecture.

The different compression methods will affect the storage space greatly. In this thesis, we choose the lossy compression method to get higher compression ratio. Besides compression methods, the following parameters also affect the storage space required.

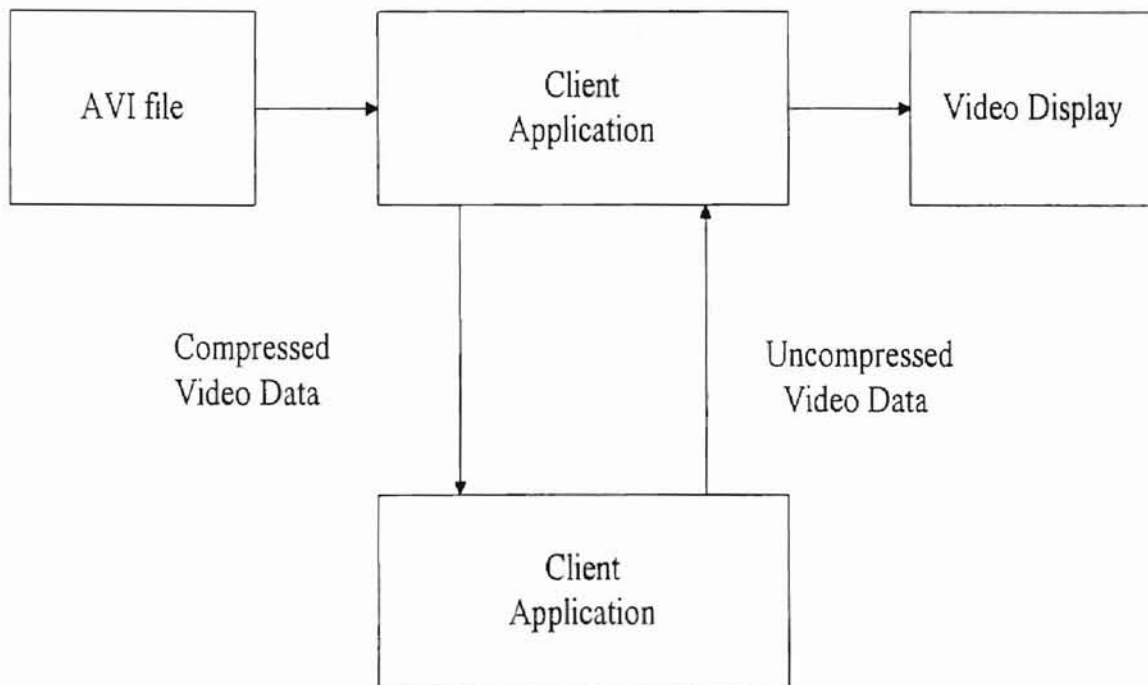


Figure 17 Decompression Driver Architecture

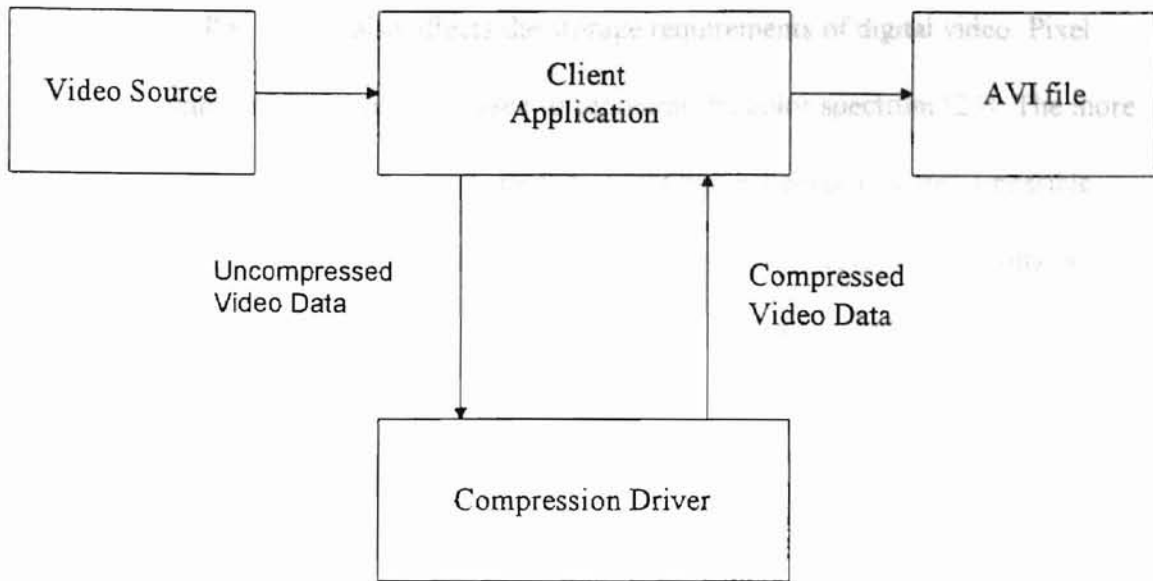


Figure 18 Compression Driver Architecture

- Playback and Capture Speed

The playback and capture speed of video can affect the storage requirements for digital video segments. The faster the playback and capture rates, the higher the storage requirements. In this thesis, we choose 15 frames per second, which provides video without incurring video and audio breakup during playback.

- Playback and Capture Size

The size of video frames affects the storage requirements greatly. By reducing the uncompressed video frame size from 640X480 to 160X120, we can reduce the space requirement from 2,457,600 bytes to 153,600 bytes. In this thesis, we choose 160X120 to display the video except at the beginning and at the end of the NFKS 2.0.

- Pixel Depth

Pixel depth also affects the storage requirements of digital video. Pixel depth is the number of bits per pixel used to represent the color spectrum [23]. The more bits per pixel, the more storage space needed. Eight bits per pixel indicates a possible range of 256 colors. Sixteen bits per pixel is needed for 16,000 colors and 24 bits per pixel for more than a million colors. In this thesis, we have chosen the 8 bits pixel depth.

3.5.2 Digital Video Capture

Digital video capture is the process to convert analog video into digital data. With the help of the CAPWNDX.VBX and Video for Windows Developer's Kit, we create a Visual Basic software to capture the video. CAPWNDX.VBX is available in the Microsoft Video for Windows SDK (Software Development Kit) library. It is used to help to capture the digital video files. Figure 19 shows the process of conversion of analog video into video data.

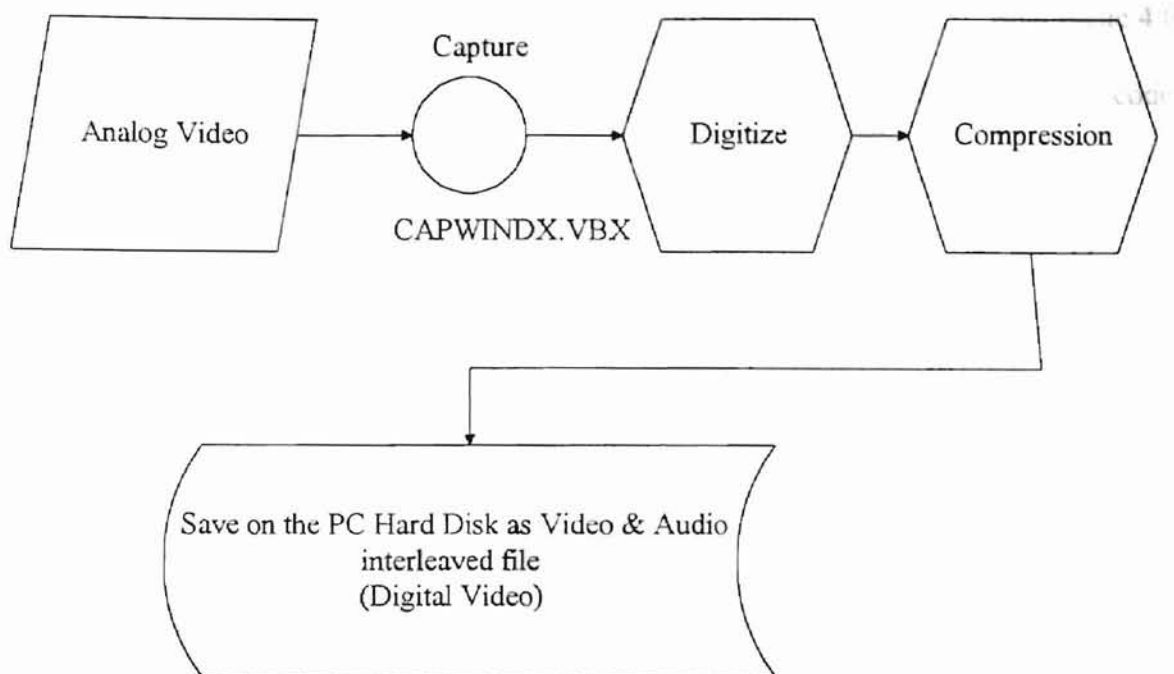


Figure 19 The Process of conversion of analog video into digital video data.

To create a video capture application with CAPWINDX.VBX, the following prerequisites are necessary.

- A Video for Windows compatible capture board and its drivers must be installed and set up correctly.
- The Video source VCR needs to be connected to the video input using the correct cables.
- The audio source needs to be connected to the audio inputs of the soundboard.
- The Video for Windows Capture VBX (CAPMNDX.VBX) must be installed.

In this thesis, we create the video capture application tool in Visual Basic 4.0. Figure 20 shows the interface of this application tool. Table X in appendix lists the code segment of this application tool.

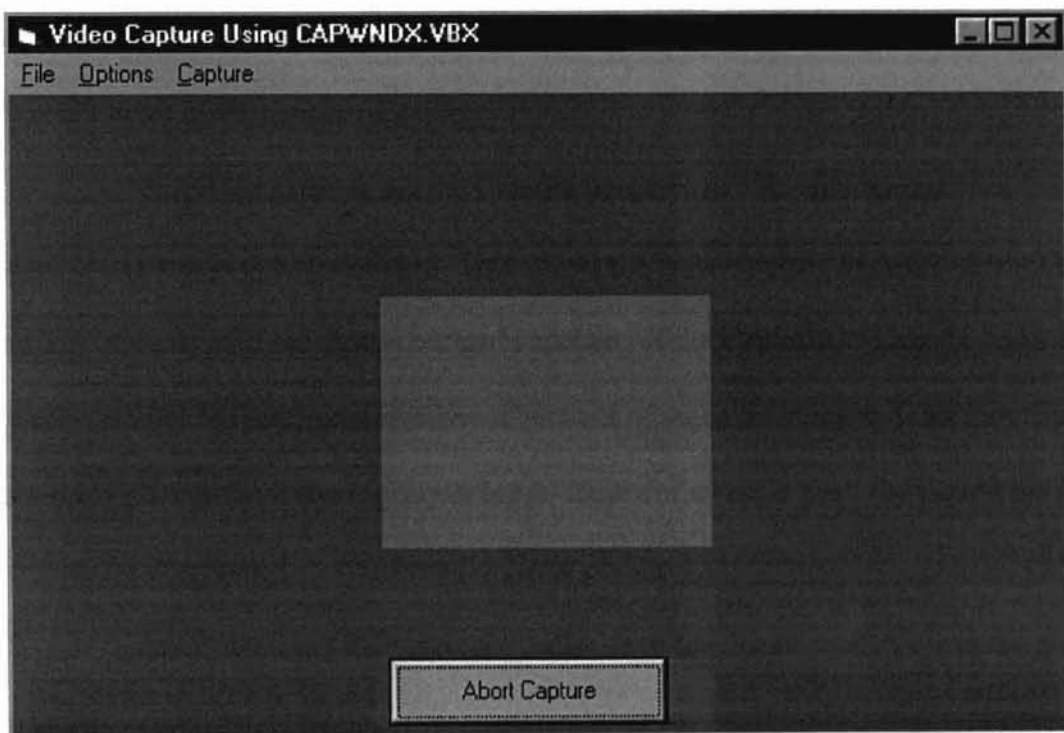


Figure 20 The Interface of Video Capture Tool

In order to capture video, we must establish the connection with the video capture driver. This is done by setting the *Connect* properties of the CAPWND.VBX control to *true*. The driver must be disconnected when the application exits, otherwise no other application will be able to access it. In a similar way, this is done by setting the *Connect* properties of the CAPWNDX.VBX control to *false*.

For audio capture, the audio parameters are set for 11 Khz AudioRate, 8 AudioBits, and 1 AudioChannel. This leads the lowest possible storage requirements and also the lowest quality audio recording.

For video capture, set the Capture property to True and set the CaptureUseLimit property to False. This allows the video capture to continue until Esc key is pressed or until the mouse button is clicked. If the CaptureUseLimit is set to True, we can specify the maximum number of seconds of video to capture. If an error occurs during any part of the video capture process, the Error event is fired, the reason for the error can be determined by checking the Error property.

After capturing the video and audio, save the file as an AVI file in the given file name.

The "Study by Items" and "Self-assessment Quiz" in NFKS 2.0 are

developed by subject matter experts in forest

CHAPTER IV

CONCLUSION AND FUTURE WORK

As mentioned earlier, NFKS 2.0 is an enhanced version of NFKS 1.0. With the increasing popularity of the multimedia applications and the availability of hardware, users are enthusiastic about the merger of audio, video, images, text, and other modes of communication. In order to attract the students' attention to this educational software, we have added video and audio.

NFKS 2.0 also adds "study by items", this gives the students a quick way to get the information which they want to know. Some items provide short movies and not only attract the students' attention, but also provide much more information than other forms of communication such as image, text, etc.

The goal of NFKS 2.0 is to provide a tool for students to learn Natural Forestry Knowledge. This goal has been accomplished. The design of NFKS 2.0 is characterized by the following components:

- The contents of NFKS 2.0

1. The material in the “Study by items” and “Study by quiz” in NFKS 2.0 are based on NFKS 1.0. It is designed by several professors who are experts in these particular areas.

2. The difficulty level is suitable for the elementary school students. NFKS has been tested by several students and teachers and at several state fairs. Students feel that it is interesting and informative.

3. The length of NFKS 2.0 is reasonable for the students. Each quiz has ten questions. It is easy for students to concentrate on one quiz at a time.

4. The important concepts are emphasized effectively. Hypertext are linked to each important concept. Students simply click on the hypertexts to get more detailed information.

5. The basic knowledge of concepts of Natural Forestry are given in “Study by items”. The content of each item is accurate and interesting.

- The format of NFKS 2.0

1. The *Welcome* and Thank you video are designed as full screen video. This gives the users the vision of opening the knowledge door when starting the software and closing the door when exiting this software.

2. The score bar is displayed at the bottom of the main window when user chooses study by quizzes. The user can know how well he/she is doing before the end of the quiz.

3. The windows formats and other formats are designed according to both the contents of NFKS and the interest of the students. It is appropriate to the material and the educational needs of the students.

- The option in NFKS 2.0

NFKS 2.0 gives the users several options on the menu. Students can study by items, or study by quiz, or access help information, or review the previous quiz, or access hypertext, or quit the system at any time.

- The data collection

NFKS 2.0 can automatically collect the answer and the score of each student. The score results are saved for each student in a data file. Table VII shows the data collected from one class. Table VIII shows the analysis of this data. From the data, we can get quantitative analysis that shows how much a student knows and which student need some more training.

Due to a lot of images and video playing, this software package requires a significant amount of hard disk space. Therefore we suggests to use CD as the storage device instead of floppy disks in the future. In order to speed up the speed of loading video information, we suggest to use high performance video RAM in the future.

TABLE VII
SCORES DATA COLLECTED FROM ONE CLASS

QUIZ NAME: TREE FACTS	DATE:
-----------------------	-------

TABLE VII (Continued)

NAME	ANSWERS	SCORES
Austin	c d d c b a a b c b	80
Erica	c b a c a c a c c b	70
Robert	a d a c d c b b c d	40
Jennifer	c b a c b d a d c b	70
Nathan	d a d a d c c b c b	50
Steven	c b c c d a b d a d	30
Erica	c b d c b c a b c b	100
Elizabeth	c b b c b c b b c b	80
Jim	b b d c d c a b c b	90
Luke	c b b d d a a b c d	50
Nathan	d d d c a c a b c d	70
Kelly	c d d c b c a b c b	80
Susan	c b d c d a a d d b	60
Rebecca	c b d c b c b b c b	90
Bill	c d d a b b d c b b	40

TABLE VIII

ANALYSIS OF DATA IN TABLE VII

Number of Question	Number of Students who choose			
	a	b	c	d
No. 1	1	1	11	2
No. 2	1	9	0	5
No. 3	3	2	1	9
No. 4	2	0	11	2

TABLE VIII (Continued)

No. 5	2	7	0	5
No. 7	4	1	8	2
No. 8	0	10	2	3
No. 9	1	1	12	1
No. 10	0	11	0	4

Accuracy Analysis:

Number of Question	Correct Answer	Accurate Percentage
No. 1	c	73
No. 2	b	60
No. 3	d	60
No. 4	c	73
No. 5	b	47
No. 6	c	53
No. 7	a	60
No. 8	b	67
No. 9	c	80
No. 10	b	73

APPENDIX

TABLE IX

CODE SEGMENT OF ONE CHILD FORM_LOAD FUNCTION

```
Private Sub Form_Load()  
    child2.Visible = False  
    Combo1.AddItem "Air"  
    Combo1.AddItem "Carbon"  
    Combo1.AddItem "Embryo"  
    Combo1.AddItem "Emergence"  
    Combo1.AddItem "Evergreen Tree"  
    Combo1.AddItem "Female Perfumes"  
    Combo1.AddItem "Fire"  
    Combo1.AddItem "Food"  
    Combo1.AddItem "Forest Area"  
    Combo1.AddItem "Greenhouse"  
    Combo1.AddItem "Harvesting"  
    Combo1.AddItem "Human Stress"  
    Combo1.AddItem "Information about Rings"  
    Combo1.AddItem "Land Area"  
    Combo1.AddItem "Land Preserved"  
    Combo1.AddItem "Loblolly Pine Needles"
```

TABLE IX (Continued)

Combo1.AddItem "Male Fragrances"
Combo1.AddItem "Making Paper"
Combo1.AddItem "National Forests"
Combo1.AddItem "Noise"
Combo1.AddItem "Old Growth"
Combo1.AddItem "Products out of Trees"
Combo1.AddItem "Prunning"
Combo1.AddItem "Recycling"
Combo1.AddItem "Satellite Picture"
Combo1.AddItem "Seeds of Pine"
Combo1.AddItem "Shading"
Combo1.AddItem "Small Tree of Pine"
Combo1.AddItem "Some Products of Trees"
Combo1.AddItem "Survival of Pine"
Combo1.AddItem "Tallest Tree"
Combo1.AddItem "Tapping the Tree"
Combo1.AddItem "Timber Harvest"
Combo1.AddItem "Tree Champion"
Combo1.AddItem "Tree Cross-Section"
Combo1.AddItem "Tree Growth"
Combo1.AddItem "Tree Making Food"
Combo1.AddItem "Tree Picture"
Combo1.AddItem "Tree Planting"
Combo1.AddItem "Tree Species"

TABLE IX (Continued)

Combo1.AddItem "Tree Trunk Picture" A SPUR IN THE FOREST
Combo1.AddItem "Unusual Products" 117
Combo1.AddItem "Usage of Loblly Pine"
Combo1.AddItem "Video1"
Combo1.AddItem "What is in a Cord"
Combo1.AddItem "Wildlife"
Combo1.AddItem "Wild Land Project"
'-----'
Combo2.AddItem "Tree Facts"
Combo2.AddItem "Does It Come From a Tree"
Combo2.AddItem "Our U.S. Forests"
Combo2.AddItem "Forest Products Recycling"
Combo2.AddItem "Urban Forestry"
End Sub

TABLE X (continued)

CODE SEGEMENT FOR VIDEO CAPTURE GENERIC INTERFACE

Private Sub Form_Load()

 CAPWindow.DriverNum = 0

 CAPWindow.Connect = True

 CenterWindow

End Sub

Private Sub Video_Click()

 CAPWindow.CAPUseLimit = False

 CAPWindow.CAP = True

End Sub

Private Sub Video_Click()

 CAPWindow.VNumColors = 236

 CAPWindow.VNumFrames = 30

 CAPWindow.VCreate = True

End Sub

Private Sub SaveCAPdVideoAs_Click()

 CAPDialog.DialogTitle = "File Save"

 CAPDialog.Action = 1

 CAPWindow.CAPSave = CAPDialog.filename

End Sub

TABLE X (Continued)

```
Private Sub SaveVideoe_Click()  
    VDialog.DialogTitle = "File Save"  
    VDialog.Action = 1  
    CAPWindow.VSave = VDialog.filename  
End Sub
```

```
Private Sub SingleFrame_Click()  
    CAPWindow.SingleFrameOpen = True  
    CAPWindow.SingleFrame = True  
    CAPWindow.SingleFrameClose = True  
End Sub
```

```
Private Sub VideoDisplay_Click()  
    CAPWindow.VideoDisplayDlg = True  
End Sub
```

```
Private Sub VideoFormat_Click()  
    CAPWindow.VideoFormatDlg = True  
    CenterWindow  
End Sub
```

```
Private Sub VideoSource_Click()  
    CAPWindow.VideoSourceDlg = True  
End Sub
```

TABLE X (Continued)

Private Sub View_Click()

 If CAPWindow.Overlay Then

 If ViewAnalog.Checked Then

 CAPWindow.Overlay = False

 ViewAnalog.Checked = False

 Else

 CAPWindow.Overlay = True

 ViewAnalog.Checked = True

 End If

 End If

End Sub

Public Sub CenterWindow()

 CAPWindow.Top = CAPVBX.Height \ 2 - CAPWindow.Height \ 2

 CAPWindow.Left = CAPVBX.Width \ 2 - CAPWindow.Width \ 2

End Sub

BIBLIOGRAPHY

- [1]. Apteker, Romnie T., James A. Fisher, Valentinse S. Kisinov, and Hanoch Neishlos, "Video Acceptability and Frame Rate", *IEEE Multimedia*, Fall, 1995. pp.32-40.
- [2]. Chen, Ming-San and Dilip D. Kandlur, "Stream Conversion to Support Interactive Video Playout", *IEEE Multimedia*, Summer, 1996.
- [3]. Chen, Xia, "A CAI System for Learning Natural Resources Knowledge (NRKS)," Oklahoma State University, MS. Thesis, 1995.
- [4]. Golubchik, Leana, John C. S. Lui, and Richard R. Muntz, "Adaptive piggybacking: a novel technique for data sharing in video-on-demand storage servers", *Multimedia System*, Vol. 4, June, 1996, pp.140-155.
- [5]. Gurewich, Nathan and Ori Gurewich, "Mastering Visual Basic 3", a Division of Macmillan Computer Publishing, 1994.
- [6]. Hofster, Fred T., "Multimedia Literacy" McGRAW-HILL, 1995.
- [7]. Jain, Ramesh, "The Role of Music in Multimedia", *IEEE Multimedia*, Spring, 1994.
- [8]. Jain, Ramesh, "What is multimedia, anyway?", *IEEE Multimedia*, Fall, 1994.
- [9]. Jarol, Scott and Anthony Potts, "Visual Basic 4 Multimedia Adventure Set", the Coriolis Group, Inc, 1995.

- [10]. Jeffocate, Judith, "*Multimedia in Practice Technology and Application*", Prentice Hall, 1995.
- [11]. Knowledge Garden, Inc. "*Knowledge Pro Windows version 2.0 manual*", Stony Brook Technology Center, 1991.
- [12]. Little, Thomas D. C. and Dinesh Venkatesh, "Prospects for Interactive Video-on-Demand", *IEEE Multimedia*, Fall, 1994.
- [13]. Nielsen, Jakob, "Putting the User in User-Interface Testing", *IEEE Software*, May, 1996. pp. 89-90.
- [14]. Peddie, Jon, "*Multimedia & Graphics Controllers*", Wincrest, 1994.
- [15]. Ramanathan, Srinivas and P. Venkat Rargan, "Architectures for Personalized Multimedia", *IEEE Multimedia*, Spring, 1994, pp.37-46.
- [16]. Rodriguez, Arturo A. and Ken Morse, "Evaluating Video Codes", *IEEE Multimedia*, Fall, 1994. PP.25-33.
- [17]. Rosenborg, Victoria, Barbara green, Jeff Hester, Walt Knowles, and Mile Wirsching, "*A Guide to Multimedia*", New Riders Publishing. 1993, PP. 1-61.
- [18]. Schank, Roger C, Korcuska Michael, and Jona Menachem, "Multimedia application for education and training: revolution or red herring?" *Computer Surveys*, vol.27, December, 1995, pp.663-635.
- [19]. Schank, Roger C., "Active Learning through Multimedia", *IEEE Multimedia*, Spring, 1994, pp. 69-78.

- [20]. Wnedy, Hall, "The Role of hypermedia in multimedia information systems",
Computing Surveys, v27, December, 1995, pp. 598-601.

VITA

Shuiyu Zhang

Candidate for the Degree of

Master of Science

Thesis: A MULTIMEDIA SYSTEM FOR LEARNING NATURAL FORESTRY
KNOWLEDGE

Major Field: Computer Science

Biographical:

Personal Data: Born in Yuanping, China, May 15, 1962, the daughter of Rong Zhang and Hui Fan.

Education: Graduated from the Yuanping High School, Yuanping, China, in June 1979; Received Bachelor of Engineering from Shanxi Agriculture University in July 1983; Received Master of Engineering from Taiyuan Technology University in July 1989; Completed requirements for the Master of Science degree at Oklahoma State University in May, 1997.

Professional Experience: Instructor, Shanxi Agriculture University, Shanxi, China, August, 1983, to July 1986; Assistant Professor, Taiyuan Heavy Mechanical Inst., August 1989, to December 1992; Research Assistant, Department of Entomology, Oklahoma State university, January 1995, to December 1996; Teaching Assistant, Computer Science Department, Oklahoma State University, August 1996, to December 1996.