DOCUMENT RESUME

ED 276 400 IR 012 389

AUTHOR	Dickson, W. Patrick
TITLE	Experimental Software Project: Final Report. Program Report 86-10.
INSTITUTION	Wisconsin Univ., Madison. School of Education.
SPONS AGENCY	National Inst. of Education (ED), Washington, DC.
PUB DATE	Jul 86
GRANT	NIE-G-84-0008
NOTE	52p.
PUB TYPE	Computer Programs (101) -- Reports - Descriptive (141)
EDRS PRICE	MF01/PC03 Plus Postage.
DESCRIPTORS	Computer Assisted Instruction; *Computer Software; Elementary Education; Elementary School Students; Feedback; Input Output Devices; Learning Processes; *Microcomputers; *Systems Development; *Videodisks; *Word Processing; *Writing (Composition)
IDENTIFIERS	Voice Recognition

ABSTRACT

Focusing on the application of new technologies to the creation of rich environments for writing, this project entailed the assembling of a low-cost microcomputer system that included a microcomputer, speech synthesis, voice recognition, and videodisk player. A variety of alternative input devices were explored, and software that permitted the smooth integration of these different technologies was developed. Summaries are provided of the development and pilot testing of the Talking Text Editor, exploration of videodisks and development of the Video Browser Program, development of the Name Frame Program as a bridge between the Video Browser and the Talking Text Editor, pilot testing of the videodisk integrated with word processors, and pilot testing of the voice recognition system. References are provided, and program listings for the Integrated Writing Environment Software (Talking Text Editor, Video Browser Program, and Name Frame Program) are appended. (KM)

Program Report 86-10

EXPERIMENTAL SOFTWARE PROJECT:
FINAL REPORT

W. Patrick Dickson

Wisconsin Center for Education Research
School of Education
University of Wisconsin
Madison, Wisconsin

July 1986

The research reported in this paper was funded by the Wisconsin Center for Education Research which is supported in part by a grant from the National Institute of Education (Grant No. NIE-G-84-0008). The opinions expressed in this paper do not necessarily reflect the position, policy, or endorsement of the National Institute of Education.

Wisconsin Center for Education Research

MISSION STATEMENT

The mission of the Wisconsin Center for Education Research is to improve the quality of American education for all students. Our goal is that future generations achieve the knowledge, tolerance, and complex thinking skills necessary to ensure a productive and enlightened democratic society. We are willing to explore solutions to major educational problems, recognizing that radical change may be necessary to solve these problems.

Our approach is interdisciplinary because the problems of education go far beyond pedagogy. We therefore draw on the knowledge of scholars in psychology, sociology, history, economics, philosophy, and law as well as experts in teacher education, curriculum, and administration to arrive at a deeper understanding of schooling.

Work of the Center clusters in four broad areas:

- Learning and Development focuses on individuals, in particular on their variability in basic learning and development processes.
- Classroom Processes seeks to adapt psychological constructs to the improvement of classroom learning and instruction.
- School Processes focuses on schoolwide issues and variables, seeking to identify administrative and organizational practices that are particularly effective.
- Social Policy is directed toward delineating the conditions affecting the success of social policy, the ends it can most readily achieve, and the cons.raints it faces.

The Wisconsin Center for Education Research is a noninstructional unit of the University of Wisconsin-Madison School of Education. The Center is supported primarily with funds from the Office of Educational Research and Improvement/Department of Education, the National Science Foundation, and other governmental and nongovernmental sources in the U.S.

Contents

Introduction

Background. The Experimental Software Project in the Wisconsin Center for Education Research was initiated with the purpose of exploring a variety of new technologies and approaches to software development. The original proposal envisioned a collaborative effort in which a team of investigators would participate in developing a coherent approach to the use of microcomputers in educational research. The departure of Professor Alex Wilkinson resulted in a shift in direction in the project away from the collaborative venture originally planned.

The project came to focus on the application of new technologies to the creation of rich environments for writing. A low-cost microcomputer system was assembled (Dickson, Neal, & Gillingham, 1984) that included a microcomputer, speech synthesis, voice recognition, and a video disk player. In addition, a variety of alternative input devices were explored, including light pens, touch tablets, mice, and joysticks.

In order to take advantage of this rich variety of hardware and peripherals, considerable effort was devoted to designing software that permitted the smooth integration of these different technologies. In retrospect it is discouraging to reflect on how many weeks and months were spent in obtaining the various cables, interface cards, dip switch settings, and software utilities, as well as in writing and debugging our own software.

In the end, we succeeded in developing the "Integrated Writing Environment" software, which is included in this final report. The "Talking Text Editor," which is one part of the larger system, was used in a dissertation (Borgh, 1985) study of the effects of spoken feedback on children's writing. A number of pilot studies were conducted using the "Video Browser" program in conjunction with the "Talking Text Editor." A small number of elementary school students were also observed using the voice recognition system to interact with the computer.

The outcome of these observations led me to a fundamentally different perspective on what might be the most profoundly important uses of microcomputers in education. This perspective was described in my paper on "thought provoking software" (Dickson, 1985).

During the past months, the software developed in the project has been cleaned up, debugged, and made more user-friendly. Programs that resided on a number of disks have now been brought together on a single disk (titled "Integrated Writing Environment") containing the most important programs developed in the project. The programs have also been adapted so that they can be explored by users who do not have the full multi-media system developed in the Wisconsin Center for Education Research.

In this final report, I will summarize some of the observations we have made and the software products developed.

Talking Text Editor

Over the two years of the Experimental Software Project, we examined a large number of educational software programs. I was involved in reviewing software for Family Computing magazine and for "The New Tech Times," a television program produced for public television. Although parts of some of the software that I reviewed was intriguing, much of it was disappointing. In addition, observations of computer use in schools from preschool through university levels produced a growing sense of the wide disparity between the claims being made for the potential impact of microcomputers on education and the reality I observed.

As did a number of other educators, I became convinced that the major application of microcomputers in schools that held significant promise for substantial impact in the next ten years was in the general area of writing. We, therefore, gave priority to thinking about how the new technologies we were studying might be used to augment the application of microcomputers to writing.

Speech synthesis is a technology that has advanced rapidly in the past few years, so that it is now possible to equip a microcomputer with an acceptable speech synthesizer for about $200. At the time we began working on the application of speech synthesis to writing, no commercial word processors were available that included the option of having the computer read aloud the sentences written.

We developed a simple but adequate text editor, "Talking Text Editor," (Appendix A) that could be easily used by children as young as grade one. The text editor allows a child to enter lines in a story and as soon as the child enters a terminal punctuation mark (period, question mark, or exclamation point), the computer automatically reads the sentence aloud and asks the child whether he or she wishes to hear it again, change it, or leave it the way it is. The child has the option of hearing the whole story at any time.

Pilot testing this software gave hints that children were more likely to detect mistakes in their writing when they heard it read back aloud.

Borgh decided to follow up on these observations in her dissertation, which was supported by the Project. The results of her dissertation (Borgh, 1985), which were presented at the 1986 meeting of the American Educational Research Association (Borgh, 1986), confirmed our hypotheses. Children in grades two and five wrote four stories using the Talking Text Editor. For two of the stories the spoken feedback was eliminated. The results showed that children engaged in between three and seven times more editing under the spoken feedback condition as compared with the no spoken feedback condition. A manuscript from this dissertation has been submitted for review. Copies of the dissertation and manuscript are included with this report.

Video Browser Program

The Experimental Software Project also explored potential applications of videodisks in education. We looked at several videodisks, including ones designed to teach specific topics, such as high school economics and college chemistry, as well as more general videodisks, such as the "First National Kids Videodisc" and a videodisk containing 50,000 archival pictures from the Smithsonian Institution.

These explorations led to feelings of intense dissonance over the gulf between the extraordinary potential of the videodisk technology and the difficulty in assimilating this technology into education. Some of the most carefully developed instructional materials using videodisks, such as one devoted to the single topic of high school economics, seem unlikely to be widely used and seemed to us to be only slightly better than other media for presenting the same content.

Our experience with the videodisk from the Smithsonian gave me a perspective on how videodisks might be used in schools that is quite different from that common among instructional media designers. The videodisk from the Smithsonian Institution came with no documentation or catalog. The videodisk contained an absolutely fascinating collection of colorful pictures of almost every imaginable subject matter distributed almost at random across the disk. Photographs of astronauts were juxtaposed with native Americans, panda bears, insects, and old musical instruments. We decided to create a simple catalog for our use but after a few days realized that we were not making much progress.

I did a simple calculation and found that if we looked at each image on the videodisk for only one minute (the minimum to write down even a two or three word description), it would require someone to work 40 hours a week for half a year to look at each image only once! So we abandoned the notion of cataloging the disk.

We also determined that it would require a hard disk, expensive data base software, and a substantial number of person-years to create such a catalog. I concluded that the sheer potential of videodisks may turn out to be a massive impediment to their successful use in schools.

This led us to develop the "Video Browser" program, that is included in the "Integrated Writing Environment" software. This program is simple enough for a first grader to use. It permits anyone to "browse" through a videodisk by pressing the arrow keys on a microcomputer. When the user sees an image of interest, it can be added to his or her file by simply pressing "K" for "Keep". This little program, which requires only 7K of memory, essentially allows a user to treat any videodisk as a collection of pictures from which to sample. An analogy may be drawn to having a very large box full of color slides. A teacher might use a program like the "Video Browser" to select a set of "slides" to show to the class in connection with a social studies unit on native Americans, for example. Given a videodisk with a wide variety of images, the potential uses in a classroom would be quite broad.

The "Video Browser" can help make the videodisk more accessible to teachers and students without requiring the larger data base programs required for more comprehensive uses of videodisks. This program can be seen as small-scale technology that allows the teacher or student to use the videodisk as a tool in ways they wish to use it. This approach contrasts sharply with that taken by designers of instructional packages, where the videodisk is designed to teach a particular content in relatively constrained fashion.

Articulating the Video Browser with the Talking Text Editor

The Name Frames Program

The Experimental Softwaro Project next sought to articulate the videodisk with the word processor in order to allow writers to insert into their writings images taken from a videodisk.

We found that children and adults seem better able to make use of the images they select if they are able to name the images. The "Name Frames" program, that is included in the "Integrated Writing Environment," was developed as a bridge between the "Video Browser" and the "Talking Text Editor." In this program, which the user can enter automatically on leaving the "Video Browser," the user can review the pictures selected and name them.

In the same spirit discussed above, there is a fundamentally different relationship between user and videodisk when the user is able to name the images with whatever name is most useful, as opposed to having to remember a frame number or remember the name given in a data base of someone else's devising. Indeed, the same image might be used in multiple ways and named accordingly.

Integrating Videodisk with Word Processors

The next step in our attempt at creating a rich, integrated writing environment was to articulate the videodisk with the word processor.

The version of the "Name Frames" program in the "Integrated Writing Environment" software allows the user to go directly from naming the images into the "Talking Text Editor" program. Within the augmented text editing program, the user can insert pictures from the videodisk into the story he or she is writing by simply entering the name given the picture into the story, enclosed in square brackets. When the program detects the square brackets, the name is matched with the names in the users file, and the image from the videodisk is displayed as the text is scrolled by pressing the return key.

We conducted pilot tests with a small number of elementary school children in our laboratory. The students seemed to be enthusiastic about the effectively unlimited number of pictures they could look at and choose from. The variety of pictures seemed to lead to whimsical, creative stories.

Although I do not have hard data on a large sample to support it, my impression is that the augmentation of the word processor with images of the kind available with videodisks would help create a highly stimulating writing environment.

It is worth mentioning that this same system would allow a teacher, or student, to create a "slide show" out of images selected from the videodisk. By using the text editor and entering the names of the images in the desired order, a teacher could, for example, have a highly flexible tool to extend the kind of teaching commonly done with filmstrips or boxed sets of s'ides. The attractive features of using a videodisk in this fashion include the extraordinary number of images available and the possibility of reordering and intermingling the images for different purposes. The images also do not become disordered with use, as can occur if students are allowed to sort through sets of slides, for example.

Voice Recognition

The multi-media system developed in the Experimental Software Project included voice recognition. The system we used could recognize up to eighty words for an individual user. Each user must first train the computer to recognize his or her voice. The users voice patterns are stored on disk for later use.

We conducted a small pilot study of several elementary school students using voice recognition with several different pieces of software. The most interesting application of voice recognition that we observed was with the programming language, Logo. The students we observed trained the computer to recognize key Logo commands, such as "FORWARD," "PENUP", and "COLOR". We then gave them assignments to draw certain shapes with the Logo turtle, as is commonly done in computer programming classes. We had the students contrast programming under voice control with typing in commands on the keyboard.

The students reported, and our observations confirmed, that programming the turtle under voice control was much more engaging. The students commented that when they used the keyboard, the turtle usually had already moved before they looked up from the keyboard, whereas with voice recognition the students' eyes were focussed directly on the turtle. In view of the widespread interest in Logo as an instructional tool, I believe that further research on programming Logo under voice control should be conducted. Our small scale study suggests this might be a fruitful area for investigation.

Concluding Comments

The topics raised in this report offer many interesting points of departure for further research. The Experimental Software Project, true to its name, developed several innovative software programs. Most importantly, the immersion in a wide variety of technologies led me to a fundamentally different perspective on educational uses of computers than I h-d at the outset of this project.

Undoubtably, by trying to explore so many different kinds of hardware and software, the project was unable to pursue any of them in sufficient depth. Each new program was infested with bugs. Each new interface board spoke a different language.

But the "Integrated Writing Environment" we have created can serve as a prototype for the work station of the near future in our schools. Speech synthesis can be added to microcomputers for about $200. Compact videodisks are becoming available at costs hardly exceeding that of a disk drive. Voice recognition can be added to microcomputers for about $1000.

What is needed is a vision of how to articulate these different systems and integrate them into classrooms and the curriculum. The explorations conducted in the Experimental Software Project suggest that the potential benefits of such an integrated approach may be great but they may also be difficult to achieve.

References

Borgh, K. The effects on children's writing of adding speech synthesis to a word processor. Doctoral dissertation, University of Wisconsin-Madison, 1985. (University Microfilms No. 86-01, 820)

Borgh, K. (1986, April). The effects of computer-generated spoken feedback on children's writing with a word processor. Paper presented at the annual meeting of the American Educational Research Association, San Francisco.

Dickson, W. P. (1985). Thought-provoking software: Juxtaposing symbol systems. Educational Researcher, 14(5), 30-38.

Dickson, W. P., Neal, V. A., & Gillingham, M. (1984). A low-cost multimedia microcomputer system for educational research and development. Educational Technology, 24(8), 20-22.

APPENDIX A

```
TALKING TEXT EDITOR

--------------------
100   REM -------------------------
--------------------
110   REM   TALKING TEXT EDITOR
--------------------
120   REM -------------------------
--------------------
1000  HIMEM: 16382
--------------------
1005  P2R = 6 * 4096
    : DIM AL(2),AH(2)
--------------------
1010  NUMQUES = - 1
    : PTR = 16640
    : D$ = CHR$ (4)
--------------------
1015  GOSUB 8000
--------------------
1017  C$ = STRY$
    : GOSUB 62000
    : C$ = "BY"
    : GOSUB 62000
    : C$ = NAME$
    : GOSUB 62000
    : PRINT
--------------------
1019  REM
(cJ)
(cJ)"main body
(cJ)
--------------------
1020  NUMQUES = NUMQUES + 1
--------------------
1030  HI = INT (PTR / 256)
    : LO = 256 * (PTR / 256 - HI)
    : LC = (16384 + (NUMQUES * 2))
    : POKE LC,LO
    : POKE (LC + 1),HI
--------------------
1040  PRINT RIGHT$ (" " + STR$ (NUMQUES + 1),2);": ";
--------------------
1050  DL$ = "."
    : GET A$
    : POKE P2R, ASC (A$)
    : P2R = P2R + 1
    : IF A$ = DL$ THEN 9000
--------------------
1060  OV =       (37)
```

```
--------------------
1070  OLD = PTR
    : CH = 1
    : GOSUB 2020
    : POKE PTR, ASC (DL$)
    : PTR = PTR + 1
--------------------
1080  OLD = PTR
    : GOSUB 3000
--------------------
1085  PRINT
    : PRINT
    : PRINT
--------------------
1090  VTAB 22
    : PRINT "C)HANGE W)HOLE STORY N)EXT";
    : IF (VOT) AND (SOUND) THEN PRINT " H)EAR";
--------------------
1092  IF VID THEN PRINT
  * : PRINT "P)ICTURES";
--------------------
1095  GET A$
    : POKE P2R, ASC (A$) + 128
    : P2R = P2R + 1
    : IF A$ = "c" OR A$ = "C" THEN GOTO 10000
--------------------
1100  IF (A$ = "H" OR A$ = "h") AND (VOT) THEN PRINT
  * : GOSUB 3000
  * : GOTO 1090
--------------------
1102  IF (VID) AND (A$ = "P" OR A$ = "p") THEN GOSUB 8900
  * : GOSUB 7005
  * : GOTO 1085
--------------------
1105  IF A$ = "W" OR A$ = "w" THEN GOSUB 5000
  * : GOTO 1090
--------------------
1110  IF A$ < > "N" AND A$ < > "n" THEN 1095
--------------------
1140  GOSUB 7000
--------------------
1500  PTR = OLD
    : GOTO 1020
--------------------
1990  GET A$
    : IF NOT ((A$ = "y") OR (A$ = "Y") OR (A$ = "n") OR (A$ = "N")) THEN 1990
--------------------
1992  POKE P2R, ASC (A$) + 128
    : P2R = P2R + 1
--------------------
```

```
1995  PRINT A$
    : RETURN
---------------------
2000  REM
(cJ)
(cJ)"input routine
(cJ)
---------------------
2010  GET A$
    : POKE P2R, ASC (A$)
    : P2R = P2R + 1
    : CH = CH + 1
---------------------
2020  IF (A$ = ".") OR (A$ = "!") OR (A$ = "?") THEN DL$ = A$
  * : PRINT DL$
  * : RETURN
---------------------
2030  IF (A$ = CHR$ (8)) AND (PTR < > OLD) THEN PTR = PTR - 1
  * : CH = CH - 1
  * : PRINT A$;" ";A$;
  * : GOTO 2010
---------------------
2035  IF A$ = "[" OR A$ = "]" THEN 2050
---------------------
2040  IF NOT ((A$ > = " " AND A$ < = "Z") OR (A$ > = "a" AND A$ < = "z")) THEN
         PRINT CHR$ (7);
  * : GOTO 2010
---------------------
2050  PRINT A$;
    : POKE PTR, ASC (A$)
    : PTR = PTR + 1
---------------------
2060  GOTO 2010
---------------------
2200  REM
(cJ)
(cJ)"STUFF
(cJ)
---------------------
2210  VTAB X
    : HTAB 1
    : CALL - 958
    : RETURN
---------------------
2250  POKE - 16368,0
---------------------
2255  IF PEEK ( - 16384) < > 141 THEN 2255
---------------------
2260  POKE - 16368,0
    : RETURN
```

```
--------------------
3000  REM
(cJ)
(cJ)"speak a sentence.
(cJ)
--------------------
3005  GOSUB 3990
--------------------
3007  PIC = 0
    : IF CHR$ ( PEEK (PTR)) = "[" THEN PIC = 1
--------------------
3010  GOSUB 4000
    : IF LEFT$ (W$,1) = "[" THEN GOSUB 8800
--------------------
3015  IF RIGHT$ (W$,1) = "!" THEN W$ = LEFT$ (W$, LEN (W$) - 1)
--------------------
3020  IF (SOUND) AND (VOT) THEN PRINT D$;"PR#2"
  * : PRINT W$
  * : PRINT D$;"PR#";SL
--------------------
3030  IF NOT (W) THEN 3010
--------------------
3040  RETURN
--------------------
3990  LC = 16384 + (NUMQUES * 2)
    : PTR = 256 * PEEK (LC + 1) + PEEK (LC)
    : RETURN
--------------------
4000  REM
(cJ)
(cJ)"get a word.
(cJ)
--------------------
4010  W$ = ""
    : W = 0
--------------------
4020  X$ = CHR$ ( PEEK (PTR))
    : IF (X$ = ".") OR (X$ = "!") OR (X$ = "?") THEN W = 1
  * : W$ = W$ + X$
  * : RETURN
--------------------
4025  IF PIC THEN 4050
--------------------
4030  IF (X$ = " ") AND W$ < > "" THEN W$ = W$ + " "
  * : PTR = PTR + 1
  * : RETURN
--------------------
4050  W$ = W$ + X$
--------------------
4060  PTR = PTR + 1
```

```
     : GOTO 4020
--------------------
5000  REM
(cJ)
(cJ)"speak & list story.
(cJ)
--------------------
5005  HOME
    : C$ = STRY$
    : GOSUB 62000
    : C$ = "BY"
    : GOSUB 62000
    : C$ = NAME$
    : GOSUB 62000
    : PRINT
--------------------
5006  REM IF(SOUND) AND (VOT) THEN  PRINT D$"PR#2":C$ = STRY$: GOSUB 62000:C
      $ = "BY": GOSUB 62000:C$ = NAME$: GOSUB 62000: PRINT D$"PR#";SL: PRI
      NT
--------------------
5010  TEMP = NUMQUES
    : FOR NUMQUES = 0 TO TEMP
--------------------
5015    GOSUB 3990
    :   PRINT RIGHT$ (" " + STR$ (NUMQUES + 1),2);": ";
--------------------
5020    GOSUB 5500
    :   PRINT
    :   GOSUB 3000
--------------------
5030    IF INT ((NUMQUES + 1) / 22) < > (NUMQUES + 1) / 22 THEN 5050
--------------------
5040    GOSUB 2250
--------------------
5050  NEXT
    : NUMQUES = TEMP
--------------------
5055  PRINT
    : PRINT
    : PRINT
--------------------
5056  RETURN
--------------------
5060  VTAB PEEK (37) - 2
    : RETURN
--------------------
5070  POP
    : GOTO 10000
--------------------
5500  REM
(cJ)
```

```
(cJ)"list a line
(cJ)
--------------------
5510
    : C = 5
    : PIC = 0
--------------------
5520  GOSUB 4000
    : IF LEN (W$) + C > COL THEN PRINT
  * : PRINT "     ";
  * : C = 5
--------------------
5525  IF LEFT$ (W$,1) = " " AND C = 5 AND LEN (W$) < > 1 THEN W$ = RIGHT$ (W$,
        LEN (W$) - 1)
  * : GOTO 5525
--------------------
5527  IF W$ = " " AND C = 5 THEN 5520
--------------------
5530  PRINT W$;
    : C = C + LEN (W$)
--------------------
5540  IF W THEN RETURN
--------------------
5550  GOTO 5520
--------------------
6000  REM
(cJ)
(cJ)"Edit last line.
(cJ)
--------------------
6005  GOSUB 6010
    : OLD = PTR
    : PRINT RIGHT$ (" " + STR$ (NUMQUES + 1),2);": ";
    : GOSUB 2000
    : POKE PTR, ASC (DL$)
    : PTR = PTR + 1
    : OLD = PTR
    : GOSUB 3000
    : GOSUB 7005
    : GOTO 1090
--------------------
6010  VTAB 22
    : HTAB 1
    : CALL - 958
    : GOSUB 3990
    : T1EMP = PTR
--------------------
6015  PRINT RIGHT$ (" " + STR$ (NUMQUES + 1),2);": ";
--------------------
```

```
6020  GOSUB 5500
--------------------
6030  PRINT
    : PRINT
    : PRINT "REPLACE LINE NUMBER "; RIGHT$ (" " + STR$ (NUMQUES + 1),2)
--------------------
6040  PTR = T1EMP
    : RETURN
--------------------
7000  REM
(cJ)
(cJ)"List the story.
(cJ)
--------------------
7003  IF NOT (CH > 35) THEN X = OV + 2 * (OV < 20) + 1 * (OV > 19)
  * : GOSUB 2200
  * : RETURN
--------------------
7005  HOME
    : C$ = STRY$
    : GOSUB 62000
    : C$ = "BY"
    : GOSUB 62000
    : C$ = NAME$
    : GOSUB 62000
    : PRINT
--------------------
7010  TEMP = NUMQUES
    : FOR NUMQUES = 0 TO TEMP
--------------------
7020    GOSUB 3990
    :   PRINT RIGHT$ (" " + STR$ (NUMQUES + 1),2);": ";
--------------------
7030    GOSUB 5500
--------------------
7035    IF INT ((NUMQUES + 1) / 22) < > (NUMQUES + 1) / 22 THEN 7040
--------------------
7036    GOSUB 2250
--------------------
7040    PRINT
    : NEXT
    : NUMQUES = TEMP
--------------------
7050  PRINT
    : PRINT
    : PRINT
    : VTAB PEEK (37) - 2
--------------------
7060  RETURN
--------------------
```

```
8000  REM
(cJ)
(cJ)"query.
(cJ)
---------------------
8001  POKE 768,160
    : POKE 769,0
    : POKE 770,32
    : POKE 771,44
    : POKE 772,254
    : POKE 773,96
---------------------
8003  HOME
    : C$ = "TALKING TEXT EDITOR"
    : GOSUB 62010
    : PRINT
    : C$ = "DESIGNED BY PATRICK DICKSON"
    : GOSUB 62010
    : PRINT
    : C$ = "PROGRAMMED BY"
    : GOSUB 62010
    : PRINT
    : C$ = "DAN JATNIEKS AND CURT ELLMANN"
    : GOSUB 62010
    : PRINT
    : C$ = "VERSION: JULY 15, 1986"
    : GOSUB 62010
    : GOTO 8006
---------------------
8005  INPUT "DO YOU HAVE 80 COLUMNS? -->";IN$
    : IF IN$ = "Y" OR IN$ = "y" THEN COL = 80
  * : SL = 3
  * : GOTO 8008
---------------------
8006  COL = 40
    : SL = 0
---------------------
8008  PRINT
    : PRINT
    : INPUT "DO YOU HAVE A VIDEO DISK? (Y/N):";IN$
    : IF IN$ = "N" OR IN$ = "n" THEN 8011
---------------------
8009  PRINT D$"BRUN VIDEO.CODE"
    : POKE 9,3
    : VID = 1
---------------------
8010  & "PLAY"
---------------------
8011  INPUT "DO YOU HAVE A VOTRAX? ";IN$
    : IF IN$ = "Y" OR IN$ = "y" THEN VOT = 1
  * : SOUND = 1
 ---------------------
 8012  LO = 0
 ---------------------
 8015  PRINT
     : INPUT "LOAD EXISTING STORY? ";IN$
     : IF IN$ = "Y" OR IN$ = "y" THEN LO = 1
 ---------------------
 8019  PRINT D$"PR#";SL
     : HOME
     : PRINT
     : PRINT
     : INPUT "WHAT IS YOUR NAME? -->";NAME$
 ---------------------
 8020  INPUT "WHAT IS THE STORY NAME? -->";STRY$
 ---------------------
 8021  INPUT "WHAT IS THE DATE? -->";DTE$
 ---------------------
 8025  N = 3
     : R$ = "@R" + STR$ (N)
     : IF SOUND THEN PRINT D$"PR#2"
   * : PRINT R$
   * : PRINT D$"PR#";SL
 ---------------------
 8030  IF NOT (VID) THEN HOME
   * : RETURN
 ---------------------
 8032  PRINT
     : INPUT "DO YOU WANT TO INCLUDE PICTURES ? ";IN$
     : IF IN$ < > "Y" AND IN$ < > "y" THEN 8034
 ---------------------
 8033  INPUT "ENTER PICTURE FILE NAME :";F$
     : F$ = F$ + ".FRAMES.NAMED"
     : GOSUB 8500
 ---------------------
 8034  IF LO THEN F$ = STRY$ + "." + NAME$ + "." + DTE$
   * : GOTO 8600
 ---------------------
 8035  HOME
     : RETURN
 ---------------------
 8037  N = 3
     : GOTO 8055
 ---------------------
 8040  INPUT "CHANGE THE SPEECH RATE? (Y/N) -->";IN$
     : IF IN$ = "N" OR IN$ = "n" THEN HOME
   * : PRINT
   * : RETURN
 ---------------------
 8050  INPUT "ENTER NEW SPEECH RATE (1-9) -->";N
```

```
     : IF N < 1 OR N > 9 THEN 8050
 --------------------
 8055  PRINT
     : PRINT "THE SPEECH RATE IS ";N
 --------------------
 8060  R$ = "@R" + STR$ (N)
     : PRINT D$"PR#2"
     : PRINT R$;"THIS IS THE SPEECH RATE."
     : PRINT D$"PR#";SL
     : GOTO 8040
 --------------------
 8500  REM
 (cJ)
 (cJ)"LOAD IN PICTURE FILE
 (cJ)
 --------------------
 8505  PRINT D$"VERIFY";F$
 --------------------
 8510  PRINT D$"OPEN";F$
     : PRINT D$"READ";F$
 --------------------
 8520  INPUT L
     : DIM FR(L)
     : DIM NAM$(L)
 --------------------
 8530  FOR Z = 1 TO L
     :   INPUT FR(Z)
     :   INPUT NAM$(Z)
     : NEXT Z
 --------------------
 8540  PRINT D$"CLOSE";F$
     : RETURN
 --------------------
 8600  REM
 (cJ)
 (cJ)"GET EXISTING FILE
 (cJ)
 --------------------
 8610  PRINT D$"BLOAD";F$;",A16383"
     : NUMQUES = PEEK (16383)
 --------------------
 8620  LN = NUMQUES
     : X = 0
     : FOR NUMQUES = 0 TO LN
     :   GOSUB 3990
     :   IF PTR > X THEN X = PTR
   * :   TEMP = NUMQUES
 --------------------
 8630  NEXT
     : NUMQUES = TEMP
```

```
      : GOSUB 3990
      : OLD = PTR
      : NUMQUES = NUMQUES - 1
--------------------
8640  POP
      : GOSUB 7005
      : OV = PEEK (37) - 1
      : GOTO 1090
--------------------
8800  REM
(cJ)
(cJ)"FIND & DISPLAY PICTURE
(cJ)
--------------------
8810  W$ = MID$ (W$,2, LEN (W$) - 3)
      : FOR Z = 1 TO L
      :   IF W$ = NAM$(Z) THEN 8820
--------------------
8815  NEXT Z
      : RETURN
--------------------
8820  C$ = "SEARCH FRAME " + STR$ (FR(Z)) + " SEARCH"
--------------------
8825  & C$
--------------------
8827  GOSUB 2250
--------------------
8828  & "FLIP"
--------------------
8830  GOSUB 2250
--------------------
8840  & "FLIP"
--------------------
8850  POP
      : RETURN
--------------------
8900  REM
(cJ)
(cJ)"LIST PICTURES
(cJ)
--------------------
8910  HOME
      : PRINT
      : PRINT
      : C$ = "PICTURES AVAILABLE"
      : GOSUB 62000
      : C$ = "-=================-"
      : GOSUB 62000
      : PRINT
      : PRINT
```

```
--------------------
8920  FOR Z = 1 TO L STEP 2
    :   PRINT RIGHT$ (" " + STR$ (Z),2);". ";NAM$(Z);
    :   HTAB 20
    :   IF Z < > L THEN PRINT RIGHT$ (" " + STR$ (Z + 1),2);". ";NAM$(Z + 1)
--------------------
8925  NEXT
    : PRINT
    : PRINT
    : PRINT
    : PRINT "ENTER NUMBER TO VIEW.  "0" TO EXIT : ";
--------------------
8930  INPUT "";IN
    : IF IN = 0 THEN RETURN
--------------------
8935  IF IN > 0 AND IN < = L THEN 8945
--------------------
8940  GOTO 8930
--------------------
8945  Z = IN
--------------------
8946  C$ = "SEARCH FRAME " + STR$ (FR(Z)) + " SEARCH"
--------------------
8948  & C$
--------------------
8950  & "FLIP"
--------------------
8955  GOSUB 2250
    : HOME
--------------------
8960  & "FLIP"
--------------------
8970  GOTO 8910
--------------------
9000  REM
(cJ)
(cJ)"the end.
(cJ)
--------------------
9020  POKE PTR, ASC (DL$)
    : PTR = PTR + 1
--------------------
9025  PRINT
    : PRINT
    : INPUT "WHAT IS THE STORY NAME? -->";STRY$
--------------------
9030  S$ = LEFT$ (STRY$ + "." + NAME$ + "." + DTE$,30)
--------------------
9035  PRINT
    : PRINT
```

```
     : PRINT "GOODBYE ";NAME$;", HAVE A NICE DAY!"
     : GOTO 9045
--------------------
9040  PRINT "YOUR FILE IS ";PTR - 16384;" BYTES LONG"
--------------------
9045  POKE 16383,NUMQUES
--------------------
9050  PRINT CHR$ (04);"BSAVE";S$;",A16383,L"; STR$ (PTR - 16393)
--------------------
9060  REM
(cJ)
(cJ)"Save history file
(cJ)
--------------------
9070  S$ = S$ + ".HST"
--------------------
9080  PRINT CHR$ (04);"BSAVE";S$;",A24576,L"; STR$ (P2R - 24576)
--------------------
9090  PRINT
    : INPUT "DO YOU WANT TO WRITE ANOTHER STORY? ";IN$
    : IF IN$ = "Y" OR IN$ = "y" THEN RUN
--------------------
9095  END
--------------------
10000 REM
(cJ)
(cJ)"change any line
(cJ)
--------------------
10005 X = 22
    : GOSUB 2200
--------------------
10007 INPUT "CHANGE WHICH LINE? -->";IN$
    : IF LEN (IN$) = 0 THEN 10000
--------------------
10010 FOR X = 1 TO LEN (IN$)
    :   POKE P2R, ASC ( MID$ (IN$,X,1))
    :   P2R = P2R + 1
    : NEXT X
    : N = VAL (IN$)
    : IF N > NUMQUES + 1 OR N < 1 THEN 10000
--------------------
10015 N = N - 1
--------------------
10016 LN = NUMQUES
    : N1 = N
--------------------
10017 X = 0
    : FOR NUMQUES = 0 TO LN
    :   GOSUB 3990
```

```
     :   IF PTR > X THEN X = PTR
   * :   TEMP = NUMQUES
--------------------
10018 NEXT
--------------------
10020 NUMQUES = N1
     : GOSUB 3990
     : N = PTR
--------------------
10025 X = N
     : O = N
     : LIM = N
     : GOSUB 10500
     : AL(0) = XL
     : AH(0) = XH
     : REM    " TO addr.
--------------------
10040 GOSUB 10600
--------------------
10050 X = N + 1
     : O = X - O
     : GOSUB 10500
     : AL(1) = XL
     : AH(1) = XH
     : REM    " FROM addr.
--------------------
10060 NUMQUES = TEMP
     : GOSUB 3990
     : N = PTR
     : GOSUB 10600
--------------------
10065 IF X - 1 = N THEN 6000
--------------------
10070 X = N
     : GOSUB 10500
     : AL(2) = XL
     : AH(2) = XH
     : REM   " THRU addr.
--------------------
10073 NUMQUES = N1
     : GOSUB 6010
--------------------
10075 X = X + 1
     : GOSUB 10500
     : NUMQUES = N1
     : GOSUB 3990
     : POKE LC,XL
     : POKE LC + 1,XH
--------------------
10080 POKE 66,AL(0)
```

```
     : POKE 67,AH(0)
     : POKE 60,AL(1)
     : POKE 61,AH(1)
     : POKE 62,AL(2)
     : POKE 63,AH(2)
     : CALL 768
--------------------
10090 FOR NUMQUES = 0 TO LN
     :   GOSUB 3990
     :   X = 256 * PEEK (LC + 1) + PEEK (LC)
     :   IF X > LIM THEN X = X - 0
   * :   GOSUB 10500
   * :   POKE LC,XL
   * :   POKE LC + 1,XH
--------------------
10095 NEXT
--------------------
10100 NUMQUES = N1
     : GOSUB 3990
     : OLD = PTR
--------------------
10110 PRINT RIGHT$ ("  " + STR$ (NUMQUES + 1),2);": ";
     : GOSUB 2000
     : POKE PTR, ASC (DL$)
     : PTR = PTR + 1
     : OLD = PTR
     : GOSUB 3000
--------------------
10160 NUMQUES = LN
     : GOSUB 7005
     : GOTO 1090
--------------------
10500 XH = INT (X / 256)
     : XL = 256 * (X / 256 - XH)
     : RETURN
--------------------
10600 X$ = CHR$ ( PEEK (N))
     : IF X$ = "!" OR X$ = "?" OR X$ = "." THEN 10620
--------------------
10610 N = N + 1
     : GOTO 10600
--------------------
10620 RETURN
--------------------
62000 REM
(cJ)
(cJ)"center a line
(cJ)
--------------------
62010 HTAB 20 - LEN (C$) / 2 - 1
```

```
: PRINT C$
: RETURN
```

VIDEO BROWSER PROGRAM Appendix B

```
--------------------
10    PRINT CHR$ (4)"BRUN VIDEO.CODE"
    : POKE 9,3
--------------------
20    HIMEM: 29179
--------------------
30    & "PLAY"
--------------------
40    D$ = CHR$ (4)
    : ADDR = 29184
    : GOSUB 10000
--------------------
50    FRAME = 1
    : C$ = "FRAME CHAPTER"
--------------------
55    & C$
--------------------
60    IF FIRST = 0 THEN FIRST = 56001
  * : LAST = 0
--------------------
70    & "FLIP"
    : GOTO 1030
--------------------
1000  REM
(cJ)
(cJ)"MAIN PROGRAM BODY
(cJ)
--------------------
1005  & "FLIP"
--------------------
1010  C$ = "SEARCH FRAME " + STR$ (FRAME) + " SEARCH"
--------------------
1015  & C$
    : GOTO 1030
--------------------
1017  C$ = "STEPREV"
    : FRAME = FRAME - 1
    : IF REVIEW THEN D = - 1
  * : GOSUB 9500
--------------------
1020  IF FRAME < 1 THEN FRAME = 1
  * : C$ = "SEARCH FRAME 1 SEARCH"
--------------------
1021  GOTO 1025
--------------------
1022  C$ = "STEPFWD"
    : FRAME = FRAME + 1
    : IF REVIEW THEN D = 1
  * : GOSUB 9500
--------------------
1023  IF FRAME > 56000 THEN FRAME = 56000
  * : C$ = "SEARCH FRAME 56000 SEARCH"
--------------------
1025  & C$
--------------------
1030  REM
--------------------
1040  GET A$
    : PRINT
    : IF ASC (A$) = 8 THEN 1017
--------------------
1045  IF ASC (A$) = 21 THEN 1022
--------------------
1047  HOME
--------------------
1050  & "FLIP"
--------------------
2000  HOME
    : GOSUB 9000
    : IF TEST THEN I$ = " IN "
  * : GOTO 2010
--------------------
2005  I$ = " NOT IN "
--------------------
2010  PRINT "CURRENT FRAME IS ";FRAME
    : PRINT "IT IS ";
    : INVERSE
    : PRINT I$;
    : NORMAL
    : PRINT "YOUR FILE."
--------------------
2020  PRINT
    : PRINT
    : PRINT "K)EEP THIS FRAME"
    : PRINT "D)ELETE THIS FRAME"
    : PRINT "G)O TO NEW FRAME NUMBER"
    : PRINT "R)EVIEW FRAMES IN CURRENT FILE"
    : PRINT "Q)UIT"
--------------------
2030  PRINT
    : PRINT
    : PRINT "ANY OTHER KEY RESUMES BROWSING."
    : PRINT "     -> BROWSE FORWARD"
    : PRINT "     <- BROWSE BACKWARD"
--------------------
2040  PRINT
    : PRINT
    : PRINT "ENTER SELECTION : ";
```

```
---------------------
2050  GET A$
    : PRINT A$;
---------------------
2060  IF A$ = "Q" THEN 2200
---------------------
2070  IF A$ = "R" THEN REVIEW = 1
  * : GOTO 2097
---------------------
2080  IF A$ = "G" THEN 2100
---------------------
2090  IF A$ = "D" THEN GOSUB 8000
  * : GOTO 2000
---------------------
2092  IF A$ = "K" THEN GOSUB 7000
  * : GOTO 2000
---------------------
2095  IF REVIEW THEN REVIEW = 0
  * : FRAME = F1
  * : GOTO 1000
---------------------
2097  & "FLIP"
    : F1 = FRAME
    : GOTO 1030
---------------------
2100  REM
(cJ)
(cJ)"ENTER FRAME #
(cJ)
---------------------
2105  PRINT
    : PRINT
    : PRINT
---------------------
2110  PRINT
    : PRINT "CURRENT FRAME IS ";FRAME
    : INPUT "ENTER NEW FRAME # : ";T1
---------------------
2120  IF T1 < 1 OR T1 > 56001 THEN 2110
---------------------
2130  FRAME = INT (T1)
    : GOTO 1000
---------------------
2200  REM
(cJ)
(cJ)"SAVE FILE
(cJ)
---------------------
2210  HOME
    : VTAB 11
```

```
     : INPUT "FILENAME TO SAVE :";F$
--------------------
2215  FLASH
     : PRINT "SAVING ";F$
     : NORMAL
     : PRINT "  MAY TAKE SEVERAL MINUTES..."
--------------------
2216  PRINT
     : PRINT "    DEPENDING ON FRAME  NUMBERS CHOSEN."
--------------------
2217  XH = INT (FIRST / 256)
     : POKE 29180,XH
     : POKE 29181,256 * (FIRST / 256 - XH)
--------------------
2218  XH = INT (LAST / 256)
     : POKE 29182,XH
     : POKE 29183,256 * (LAST / 256 - XH)
--------------------
2220  PRINT D$"BSAVE ";F$;".PICS,A29180,L7004"
--------------------
2225  F$ = F$ + ".FRAMES"
--------------------
2230  PRINT D$"OPEN";F$
     : PRINT D$"DELETE";F$
     : PRINT D$"OPEN";F$
     : PRINT D$"WRITE";F$
--------------------
2235  PRINT COUNT
--------------------
2240  FOR Z1 = FIRST TO LAST
     :   FRAME = Z1
     :   GOSUB 9000
     :   IF TEST THEN PRINT Z1
--------------------
2245  NEXT Z1
--------------------
2250  PRINT D$"CLOSE";F$
--------------------
2255  HOME
     : VTAB 11
--------------------
2260  INPUT "DO YOU WANT TO NAME THESE FRAMES NOW    TYPE Y FOR YES  N FOR N."
         ;IN$
--------------------
2270  IF IN$ = "Y" OR IN$ = "y" THEN PRINT D$"RUN NAME FRAMES"
--------------------
2290  HOME
     : END
--------------------
2300  REM
(cJ)
```

```
(cJ)"LOAD A FILE
(cJ)
--------------------
2305  PRINT
    : PRINT
    : PRINT "OMIT '.PICS' SUFFIX IN FILENAME."
    : PRINT
--------------------
2310  INPUT "NAME OF FILE TO LOAD :";F$
--------------------
2320  PRINT D$"BLOAD";F$;".PICS,A29180"
--------------------
2325  FIRST = 256 * PEEK (29180) + PEEK (29181)
--------------------
2326  LAST = 256 * PEEK (29182) + PEEK (29183)
--------------------
2330  PRINT D$"OPEN";F$;".FRAMES"
    : PRINT D$"READ";F$;".FRAMES"
--------------------
2340  INPUT COUNT
    : PRINT D$"CLOSE";F$;".FRAMES"
--------------------
2350  RETURN
--------------------
6000  REM
(cJ)
(cJ)"GET A RETURN
(cJ)
--------------------
6010  GET A$
    : IF ASC (A$) < > 13 THEN 6010
--------------------
6020  PRINT
    : RETURN
--------------------
7000  REM
(cJ)
(cJ)"ADD A PICTURE
(cJ)
--------------------
7002  COUNT = COUNT + 1
--------------------
7005  GOSUB 9000
    : IF TEST THEN RETURN
--------------------
7010  POKE BYTE, PEEK (BYTE) + 2 ^ BIT
--------------------
7020  IF FRAME > LAST THEN LAST = FRAME
--------------------
```

```
7025  IF FRAME < FIRST THEN FIRST = FRAME
---------------------
7030  RETURN
---------------------
8000  REM
(cJ)
(cJ)"DELETE A PICTURE
(cJ)
---------------------
8005  GOSUB 9000
    : IF NOT (TEST) THEN RETURN
---------------------
8007  COUNT = COUNT - 1
---------------------
8010  T1 = PEEK (BYTE) - 2 ^ BIT
---------------------
8020  IF T1 < 0 THEN STOP
---------------------
8030  POKE BYTE,T1
---------------------
8040  IF FRAME = LAST THEN D = - 1
  * : GOTO 8065
---------------------
8050  IF FRAME = FIRST THEN D = 1
  * : GOTO 8065
---------------------
8060  RETURN
---------------------
8065  F1 = FRAME
    : IF LAST = FIRST THEN LAST = 0
  * : FIRST = 56001
  * : RETURN
---------------------
8070  FRAME = FRAME + D
    : GOSUB 9000
    : IF NOT (TEST) THEN 8070
---------------------
8075  IF D = - 1 THEN LAST = FRAME
  * : FRAME = F1
  * : RETURN
---------------------
8080  IF D = 1 THEN FIRST = FRAME
  * : FRAME = F1
  * : RETURN
---------------------
8085  STOP
---------------------
9000  REM
(cJ)
(cJ)"TEST PICTURE
(cJ)
```

```
--------------------
9010  BYTE = ADDR + INT (FRAME / 8)
--------------------
9020  BIT = INT ((FRAME / 8 - INT (FRAME / 8)) * 8 + .05) * SGN (FRAME / 8)
--------------------
9030  T1 = PEEK (BYTE)
    : IF T1 = 0 THEN 9080
--------------------
9050  FOR Z = 0 TO BIT
--------------------
9060    T1 = INT (T1)
    :   T1 = T1 / 2
    : NEXT
--------------------
9070  IF INT (T1) < > T1 THEN TEST = 1
  * : RETURN
--------------------
9080  TEST = 0
    : RETURN
--------------------
9500  REM
(cJ)
(cJ)"REVIEW FRAMES
(cJ)
--------------------
9510  GOSUB 9000
    : IF TEST THEN RETURN
--------------------
9520  FRAME = FRAME + D
    : IF (FRAME < FIRST) THEN FRAME = FIRST
  * : GOTO 9540
--------------------
9525  IF (FRAME > LAST) THEN FRAME = LAST
  * : GOTO 9540
--------------------
9530  GOSUB 9000
    : IF NOT (TEST) THEN 9520
--------------------
9540  C$ = "SEARCH FRAME " + STR$ (FRAME) + " SEARCH"
    : RETURN
--------------------
10000 REM  "INITIALIZE"
--------------------
10001 HOME
    : C$ = "VIDEO BROWSER PROGRAM"
    : GOSUB 20000
--------------------
10002 PRINT
    : C$ = "DEVELOPED AS PART OF"
```

```
     : GOSUB 20000
--------------------
10003 PRINT
     : C$ = "THE EXPERIMENTAL SOFTWARE PROJECT"
     : GOSUB 20000
--------------------
10005 VTAB 18
     : C$ = "INITIALIZING ARRAY"
     : GOSUB 20000
--------------------
10006 PRINT
     : C$ = "THIS WILL TAKE ABOUT ONE MINUTE"
     : GOSUB 20000
--------------------
10010 A = ADDR
     : B = A + 7000
--------------------
10020 FOR Z = A TO B STEP 2
     :   POKE Z,0
     :   POKE Z + 1,0
     : NEXT
--------------------
10030 & "SEARCH FRAME 1 SEARCH"
--------------------
10100 HOME
     : C$ = "VIDEO BROWSER PROGRAM"
     : GOSUB 20000
--------------------
10102 PRINT
     : C$ = "ALLOWS YOU TO SELECT PICTURES"
     : GOSUB 20000
--------------------
10104 PRINT
     : C$ = "FROM A VIDEO DISK FOR USE WITH"
     : GOSUB 20000
--------------------
10105 PRINT
     : C$ = "THE TALKING TEXT EDITOR PROGRAM."
     : GOSUB 20000
--------------------
10110 C$ = "DESIGNED BY W. PATRICK DICKSON"
     : PRINT
     : PRINT
     : GOSUB 20000
--------------------
10120 C$ = "PROGRAMMED BY DAN JATNIEKS"
     : PRINT
     : PRINT
     : GOSUB 20000
--------------------
```

```
10130 PRINT
    : PRINT
--------------------
10135 C$ = "VERSION: JULY 15, 1986"
    : GOSUB 20000
--------------------
10140 C$ = "(C) 1984, 1986"
    : GOSUB 20000
    : PRINT
--------------------
10141 C$ = "BY THE BOARD OF REGENTS OF THE"
    : GOSUB 20000
    : C$ = "UNIVERSITY OF WISCONSIN-MADISON"
    : GOSUB 20000
--------------------
10150 GOSUB 20100
--------------------
10200 HOME
    : PRINT
    : PRINT "     THIS PROGRAM IS DESIGNED TO ALLOW"
    : PRINT "A USER TO BROWSE THROUGH A VIDEODISK ANDSELECT THE PICTURES THAT
         ARE OF"
    : PRINT "INTEREST.  THE PROGRAM WRITES A FILE"
--------------------
10210 PRINT "THAT CONTAINS THE FRAME NUMBERS OF THE"
    : PRINT "SELECTED IMAGES."
--------------------
10250 PRINT
    : PRINT "     THE PROGRAM BEGINS WITH FRAME 1."
    : PRINT "USE THE RIGHT ARROW TO BROWSE THROUGH"
    : PRINT "THE PICTURES.  USE THE LEFT ARROW TO"
    : PRINT "BACK UP.  PRESS ANY OTHER KEY KEEP A"
--------------------
10260 PRINT "PICTURE IN YOUR FILE."
    : PRINT
    : PRINT
--------------------
10270 PRINT "     ->   BROWSE FORWARD"
    : PRINT "     <-   BROWSE BACKWARD"
    : PRINT "     ANY OTHER KEY TO GO TO MENU."
--------------------
10280 GOSUB 20100
--------------------
10290 HOME
    : VTAB 11
    : PRINT "WOULD YOU LIKE TO LOAD AN EXISTING"
    : PRINT "FILE ?  (Y/N)  -->";
    : GET A$
    : PRINT A$
    : IF A$ = "Y" OR A$ = "y" THEN 2300
```

```
--------------------
10300 RETURN
--------------------
20000 HTAB (20 - LEN (C$) / 2)
    : PRINT C$
    : RETURN
--------------------
20100 VTAB 23
    : C$ = "PRESS RETURN TO CONTINUE"
    : GOSUB 20000
    : POKE - 16368,0
--------------------
20110 IF PEEK ( - 16384) < > 141 THEN 20110
--------------------
20120 POKE - 16368,0
    : RETURN
```

NAME FRAMES PROGRAM Appendix C

```
--------------------
2     REM  --------------------
--------------------
4     REM    NAME FRAMES PROGRAM
--------------------
6     REM  --------------------
--------------------
10    D$ = CHR$ (4)
--------------------
12    PRINT D$"BRUN VIDEO.CODE"
    : POKE 9,3
--------------------
20    HOME
    : C$ = "NAME FRAMES PROGRAM"
    : GOSUB 20000
--------------------
22    PRINT
    : C$ = "DEVELOPED AS PART OF"
    : GOSUB 20000
--------------------
24    PRINT
    : C$ = "THE EXPERIMENTAL SOFTWARE PROJECT"
    : GOSUB 20000
--------------------
26    PRINT
    : PRINT
    : C$ = "DESIGNED BY PATRICK DICKSON"
    : GOSUB 20000
--------------------
28    PRINT
    : C$ = "PROGRAMMED BY DAN JATNIEKS"
    : GOSUB 20000
--------------------
30    PRINT
    : C$ = "VERSION: JULY 15, 1986"
    : GOSUB 20000
--------------------
31    PRINT
    : PRINT
    : PRINT
--------------------
32    PRINT
    : PRINT "PLEASE WAIT ABOUT ONE MINUTE."
--------------------
40    & "PLAY WAIT 150"
--------------------
42    & "STEPFWD"
--------------------
```

```
50    HOME
    : C$ = "NAME FRAMES PROGRAM"
    : GOSUB 20000
--------------------
51    PRINT
    : PRINT
--------------------
52    PRINT "THIS PROGRAM ALLOWS YOU TO NAME"
--------------------
54    PRINT
    : PRINT "PICTURES PREVIOUSLY SELECTED WITH"
--------------------
56    PRINT
    : PRINT "THE VIDEO BROWSER PROGRAM."
--------------------
58    PRINT
    : PRINT "SUCH PICTURE SETS APPEAR IN THE DISK"
--------------------
60    PRINT
    : PRINT "CATALOG AS TWO FILES WITH SUFFIXES"
--------------------
62    PRINT
    : PRINT "OF '.PICS' AND '.FRAMES'."
--------------------
64    GOSUB 20100
    : HOME
--------------------
66    PRINT
    : PRINT "ENTER THE PICTURE FILE NAME BELOW"
--------------------
68    PRINT
    : PRINT "WITHOUT A SUFFIX AND PRESS RETURN."
--------------------
70    PRINT
    : PRINT "WHEN THE PROGRAM PAUSES,"
--------------------
72    PRINT
    : PRINT "  ALWAYS PRESS RETURN TO GO ON"
--------------------
74    PRINT
    : PRINT
    : PRINT
    : PRINT
--------------------
80    INPUT "FILE NAME: ";F$
--------------------
90    F$ = F$ + ".FRAMES"
--------------------
100   PRINT D$"OPEN";F$
    : PRINT D$"READ";F$
```

```
--------------------
110   INPUT L
    : DIM FR(L),NAM$(L)
--------------------
120   FOR Z = 1 TO L
    :   INPUT FR(Z)
    :   NAM$(Z) = ""
    : NEXT
--------------------
130   PRINT D$"CLOSE";F$
--------------------
140   GOSUB 600
--------------------
200   HOME
    : Z = 1
--------------------
210   GOSUB 500
    : Z = Z + 1
    : IF Z < = L THEN 210
--------------------
300   HOME
    : PRINT
--------------------
310   FOR Z = 1 TO L
    :   T$ = " " + STR$ (Z)
    :   PRINT RIGHT$ (T$,2);".  ";NAM$(Z)
    : NEXT Z
--------------------
320   PRINT
    : INPUT "WANT TO CHANGE ANY NAMES? (Y/N) :";I$
--------------------
330   IF I$ = "Y" OR I$ = "y" THEN 340
--------------------
335   IF I$ = "N" OR I$ = "n" THEN 400
--------------------
337   GOTO 320
--------------------
340   INPUT "ENTER FRAME TO CHANGE :";Z
    : GOSUB 500
    : GOTO 300
--------------------
400   PRINT
    : PRINT "ENTER A FILE NAME FOR THESE PICTURES"
    : PRINT
    : INPUT "FILE NAME: ";F$
--------------------
405   F$ = F$ + ".FRAMES.NAMED"
    : PRINT D$"OPEN";F$
    : PRINT D$"WRITE";F$
--------------------
```

```
410   PRINT L
--------------------
420   FOR Z = 1 TO L
     :   PRINT FR(Z)
     :   PRINT NAM$(Z)
     : NEXT Z
--------------------
430   PRINT D$"CLOSE";F$
     : PRINT
--------------------
440   HOME
     : PRINT "YOU HAVE NOW NAMED THE PICTURES IN"
--------------------
442   PRINT
     : PRINT "YOUR FILE AND SAVED THE NAMED FILE."
--------------------
444   PRINT
     : PRINT "YOU MAY STOP AT THIS POINT AND RETURN"
--------------------
445   PRINT
     : PRINT "LATER BY RUNNING THE 'TALKING TEXT"
--------------------
446   PRINT
     : PRINT "EDITOR' PROGRAM AND NAMING YOUR"
--------------------
447   PRINT
     : PRINT "PICTURE FILE."
--------------------
448   PRINT
     : PRINT "OR YOU MAY GC DIRECTLY INTO THE"
--------------------
449   PRINT
     : PRINT "TALKING TEXT EDITOR NOW AND WRITE"
--------------------
450   PRINT
     : PRINT "A STORY USING THE PICTURES."
--------------------
452   GOSUB 20100
     : HOME
--------------------
456   PRINT
     : PRINT "IF YOU WANT TO USE THE TEXT EDITOR NOW,"
--------------------
458   PRINT
     : PRINT "TYPE 'Y' BELOW."
--------------------
460   PRINT
     : PRINT
     : INPUT "USE TEXT EDITOR NOW? (Y/N) :";I$
--------------------
```

```
462   IF I$ < > "Y" AND I$ < > "y" THEN END
--------------------
470   PRINT D$"RUN TALKING TEXT EDITOR"
--------------------
500   HOME
    : VTAB 11
    : PRINT "GETTING FRAME ";Z
--------------------
505   PRINT "PRESS 'RETURN' WHEN DONE VIEWING."
--------------------
510   C$ = "SEARCH FRAME " + STR$ (FR(Z)) + " SEARCH"
--------------------
525   & C$
--------------------
530   & "FLIP"
--------------------
540   GET I$
    : IF ASC (I$) < > 13 THEN 540
--------------------
550   & "FLIP"
--------------------
551   PRINT
    : PRINT
--------------------
552   INPUT "WANT TO KEEP THIS PICTURE? (Y/N) :";I$
    : IF I$ = "Y" OR I$ = "y" THEN 560
--------------------
553   IF I$ < > "N" AND I$ < > "n" THEN 552
--------------------
555   FOR Z1 = Z + 1 TO L
    :   FR(Z1 - 1) = FR(Z1)
    :   NAM$(Z1 - 1) = NAM$(Z1)
    : NEXT
    : L = L - 1
    : Z = Z - 1
    : RETURN
--------------------
560   PRINT
    : PRINT "ENTER NAME FOR THIS FRAME"
    : PRINT
    : INPUT "NAME: ";NAM$(Z)
--------------------
565   NAM$(Z) = LEFT$ (NAM$(Z),18)
--------------------
570   RETURN
--------------------
600   & "FLIP"
--------------------
605   FOR Z = 1 TO L
--------------------
```

```
610     C$ = "SEARCH FRAME " + STR$ (FR(Z)) + " SEARCH"
---------------------
620     & C$
---------------------
640     GET I$
    :   IF ASC (I$) < > 13 THEN 640
---------------------
660   NEXT Z
---------------------
665   & "FLIP"
---------------------
670   RETURN
---------------------
20000 HTAB (20 - LEN (C$) / 2)
    : PRINT C$
    : RETURN
---------------------
20100 VTAB 23
    : C$ = "PRESS RETURN TO CONTINUE"
    : GOSUB 20000
    : POKE - 16368,0
---------------------
20110 IF PEEK ( - 16384) < > 141 THEN 20110
---------------------
20120 POKE - 16368,0
    : RETURN
```

Research Staff

LEARNING AND DEVELOPMENT AREA

B. Bradford Brown
Assistant Professor
Educational Psychology

Anne M. Donnellan
Associate Professor
Studies in Behavioral Disabilities

William Epstein
Professor
Psychology

Arthur M. Glenberg
Associate Professor
Psychology

William M. Reynolds*
Professor
Educational Psychology

Laurence Steinberg*
Professor
Child and Family Studies

CLASSROOM PROCESSESS AREA

Thomas P. Carpenter
Professor
Curriculum and Instruction

Elizabeth H. Fennema
Professor
Curriculum and Instruction

Penelope L. Peterson
Professor
Educational Psychology

SCHOOL PROCESSES AREA

William H. Clune*#
Professor
Law

Gary D. Gaddy*
Assistant Professor
Journalism and Mass Communication

Adam Gamoran*
Assistant Professor
Sociology

Carl A. Grant
Professor
Curriculum and Instruction

Herbert J. Klausmeier
Founding WCER Director and V.A.C. Henmon Professor
Educational Psychology

Mary H. Metz*
Associate Professor
Educational Policy Studies

Fred M. Newmann*
Secondary Center Director and Professor
Curriculum and Instruction

P. Martin Nystrand*
Associate Professor
English

Janice H. Patterson*#
Assistant Scientist

Allan J. Pitman
Lecturer
School of Education
Deakin University

Stewart C. Purkey*#
Assistant Professor
Lawrence University

Thomas A. Romberg
Professor
Curriculum and Instruction

Richard A. Rossmiller*
Professor
Educational Administration

Robert A. Rutter*
Assistant Scientist

Gary G. Wehlage*
Professor
Curriculum and Instruction

Kenneth M. Zeichner+
Associate Professor
Curriculum and Instruction

SOCIAL POLICY AREA

William H. Clune*#
Professor
Law

W. Lee Hansen°
Professor
Economics

Carl F. Kaestle
Professor
Educational Policy Studies and History

Joseph F. Kauffman°
Professor
Educational Administration

Cora B. Marrett*
Professor
Sociology and Afro-American Studies

Michael R. Olneck
Associate Professor
Educational Policy Studies and Sociology

Thomas A. Romberg
Professor
Curriculum and Instruction

Francis K. Schrag*
Professor
Educational Policy Studies

Marshall S. Smith*#
WCER Director and Professor
Educational Policy Studies and Educational Psychology

Jacob O. Stampen°
Assistant Professor
Educational Administration

Research Support Staff

Jacob Evanson
Statistical Data Analyst

Janice Gratch
Project Specialist

Susan D. Pittelman
Project Coordinator

Deborah M. Stewart
Administrative Program Manager

Dan G. Woolpert
Program Coordinator

*affiliated with the National Center on Effective Secondary Schools, University of Wisconsin-Madison
#affiliated with the Center for Policy Research in Education, Rutgers University
°affiliated with the Center on Postsecondary Management and Governance, University of Maryland
+affiliated with the Center on Teacher Education, Michigan State University