X power

STUDY OF X-POWER

QUAID-I-AZAM UNIVERSITY

BY

SHEHNILA VERYAMANI
PGD (FDE) 2002-2003

Supervised By:

Mr.Nazim-ud-din
Director Computer Center

QUAID-I-AZAM UNIVERSITY
ISLAMABAD

بسم الله الرحمان الرحيم

The reward for good deed is

to have it done

FINAL APPROVAL

Quaid-I-Azam University, Islamabad

Computer Center

This is to certify that we have read the project report submitted by Shehnila Veryamani and it is our judgment that this report is of sufficient standard to warrant its acceptance by Quaid-e-Azam University Islamabad. For the award of Post Graduate diploma in Computer Science.

Committee

1) External Examiner:

Signatures: __________

2) Supervisor:

Mr. Nazim-ud-din
Director Computer Center
Quaid-I-Azam University
Islamabad.

Signatures: __________

3) Director:

Mr. Nazim-ud-din
Director Computer Center
Quaid-I-Azam University
Islamabad.

Signatures: __________

DECLARATION

I declare that this project, neither as a whole nor as a part has been copied from any other source. It is further declared that I have completed my final project of Post Graduate Diploma in Computer Sciences successfully as a result my own struggle and research. No portion of this work, presented in this report has been submitted in support of any application for any other degree or qualification of this University or any other University or institute of learning. If any part of the project and write up is proved to be copied out or there is any duplication of code then I shall be responsible for the consequences.

Name of Student: ***Miss Shehnila Veryamani***
PGD (FDE)

PROJECT BRIEFS

Title:

The Study of X-Power

Studied by:

Miss Shehnila Veryamani

Supervised by:

Mr. Nazim-ud-din
Director Computer Centre
Quaid-I-Azam University
Islamabad

Organization:

Quaid-I-Azam University
Islamabad

Commencement date:

July, 2003

Completion date:

October, 2003

Software used:

X-POWER

OPERATING SYSTEM:

Windows 98

SYSTEM USED:

Pentium -3

ACKNOWLEDGEMENTS

I am very grateful to **ALMIGHTY ALLAH** WHO has been always KIND to me & HAS blessed, whenever I asked HIM for favors. THE BENEFICIENT, THE MOST MERCIFUL, WHO made it possible for me to complete this project.

Special thanks to my most loving parents whose encouragements, appreciation and kind support has enabled me to stand where I am today. Their endless prayers are a source of destination for me.

I am highly obliged to my supervisor and the director of Computer Center Mr. Nazim-ud-din for his valuable suggestions, guidance and consistent encouragement throughout the project. His kind helps and support made it possible for me to complete this project, which would otherwise, has been a difficult task.

I am also very thankful to Sir Munnawar Tiwana for providing me full technical assistance & support.

Finally, I am very thankful to my fellows & friends and the staff of computer center for their cooperation.

SHEHNILA VERYAMANI

DEDICATED

To

My beloved parents
Brothers, Saima Babur
&
Respected Sir Nazim-ud-din.

PREFACE

Computer graphics is a fascinating area of computer science, as with the help of figures & pictures one can convey his massage much easier, which would otherwise, requires hour of oral explanation. It is widely used as a tool for visualizing information in a broad variety of fields.

Graphics programming used to be a luxury but today's computer users expect and demand more visual and intuitive programs.

The study of X-power was a very interesting experience for me as X-power provides the facilities to work in many` fields that are different yet associated with each other, for example we can make simple presentations, work with text, audio-visuals, bitmaps, animate objects, play midi etc.

G language that is used for programming in X-power is like the c-language but infact is much simpler and easy to learn. Moreover its Help is a blessing for new users.

It took a month to go through the commands, functions and syntax & the remaining time was spent in designing the application.

TABLE OF CONTENTS

CHAPTER # 1

Introduction

A. What Is It? What Is It Used For?

Xpower is a Windows Development Environment for creating high performance applications for Microsoft Windows in the G programming language. Additionally, Xpower allows these same applications to run remotely over the Internet, or any TCP/IP network. You develop the program as you would in Visual Basic or Delphi, simply write the presentation in
The Xpower IDE and let Xpower take care of Windows. Display pictures and text, animations, add sound, apply special effects, create interactive controls, and generate standalone presentations to distribute with a royalty-free runtime. Or just put the presentation on the Internet and let people download a small Xpower runtime to watch your work remotely.

Xpower is built around the G language for the professional who needs complete flexibility and control, and the Xpower Development Environment, with full-feature editing, and built-in debugging with code-trace, variable watch windows, and context-sensitive Help.

System Requirements

Works with Windows 3.1, 3.11 and Win95.

Xpower will work with any 386 or better computer, however an MPC II 486-33 or better computer is recommended. As with most Windows applications, the more memory, the better.

To see the Internet features designed in Xpower you must have a dial-up or direct Internet connection. .

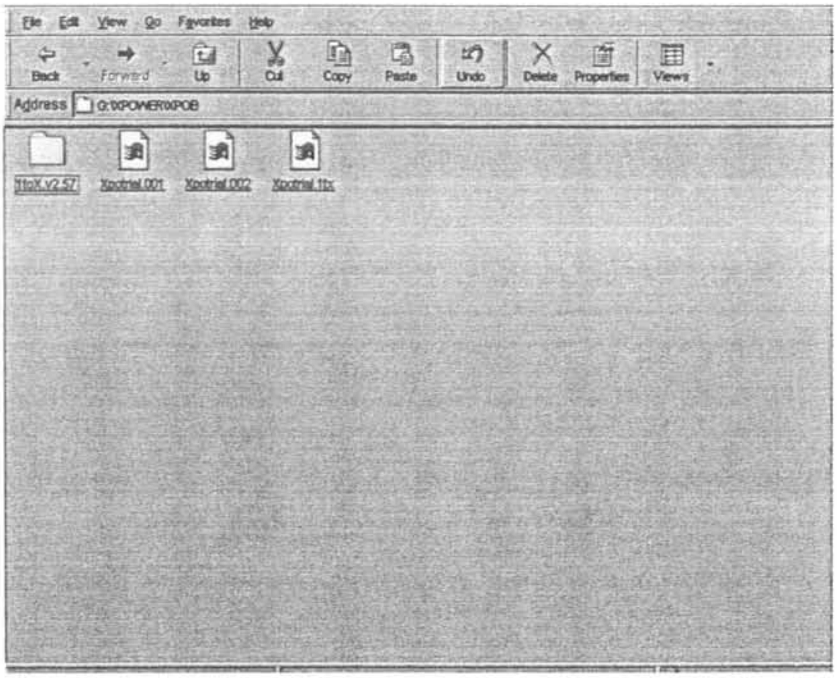
To view an Xpower presentation via the Internet the user must have the Xpower runtime XPOWER.EXE installed along with its associated XPOW16.DLL. To show an Xpower presentation, the source must first be placed on an FTP server.

Installation of X-Power from floppy disk.

1. Create a folder in "C" drive having any name other than X-power.

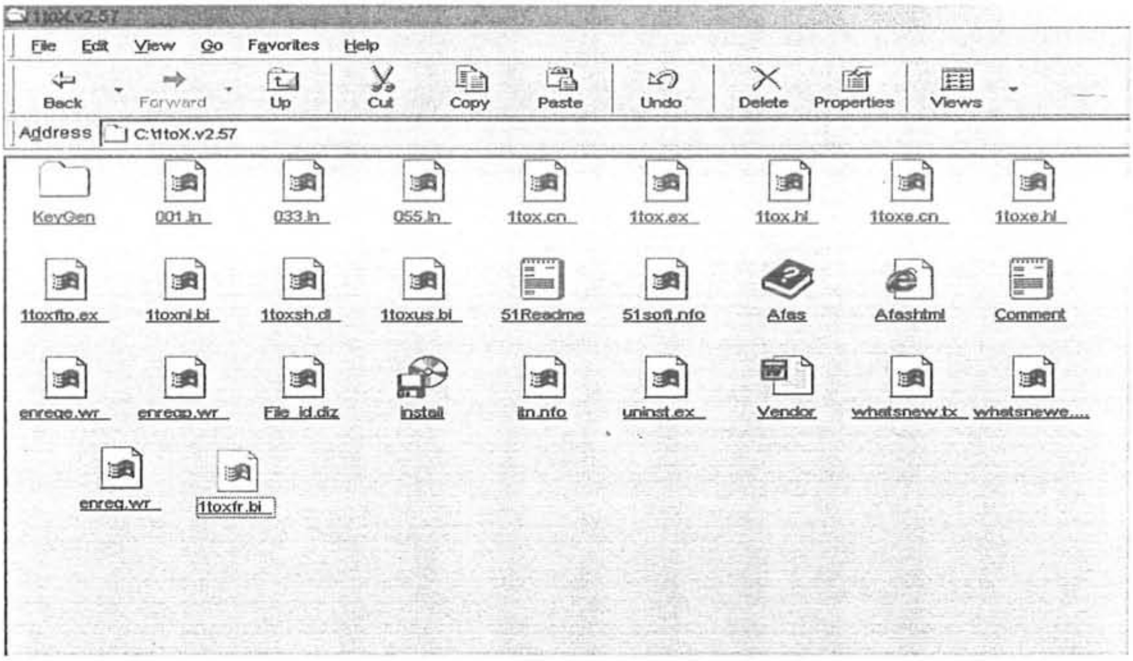
2. Copy all the diskettes into the folder.

3. Open 1tox.v2.57.



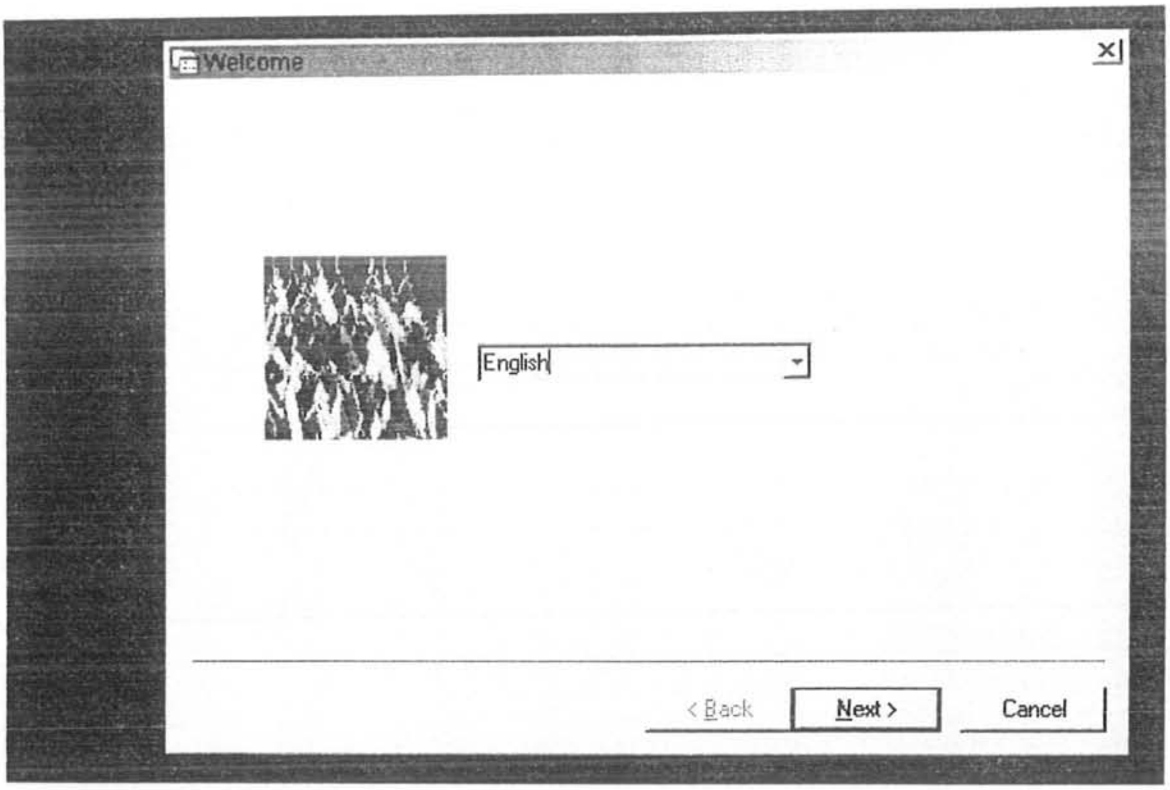
Address G:\XPOWER\XPOB

4. You will see…



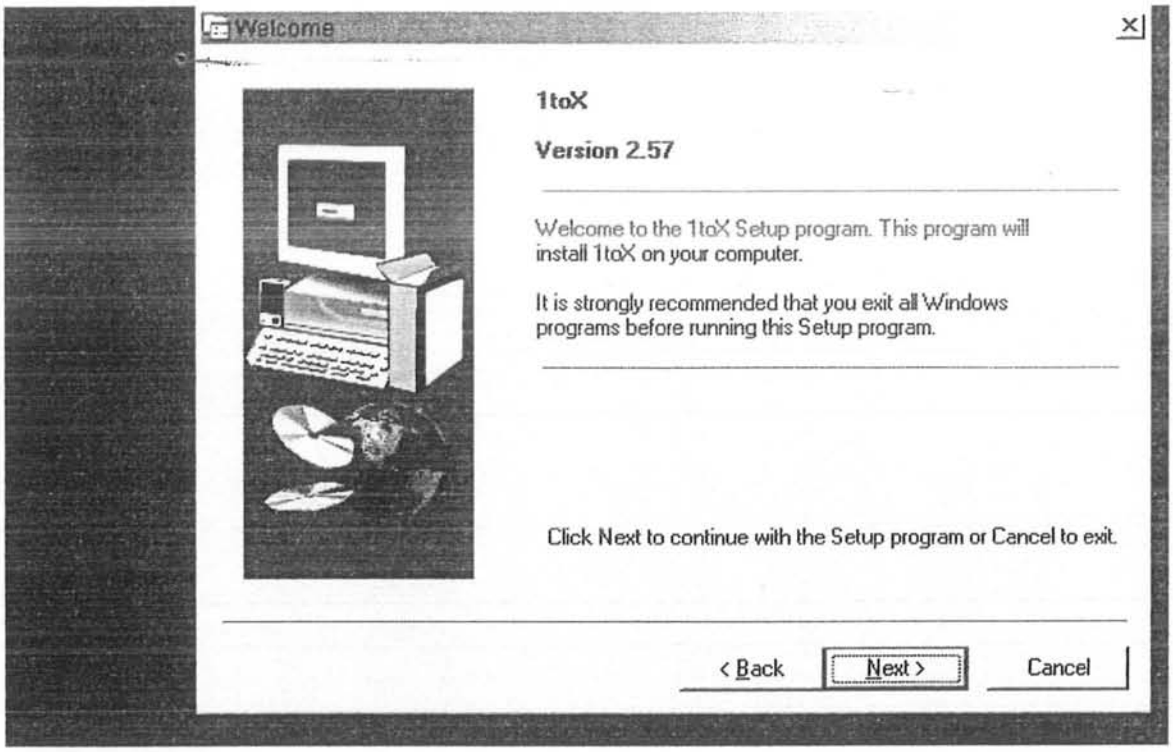
4. Click “install”.

5. A dialogue box will appear …



English

Select English.
Click "next".



1toX

Version 2.57

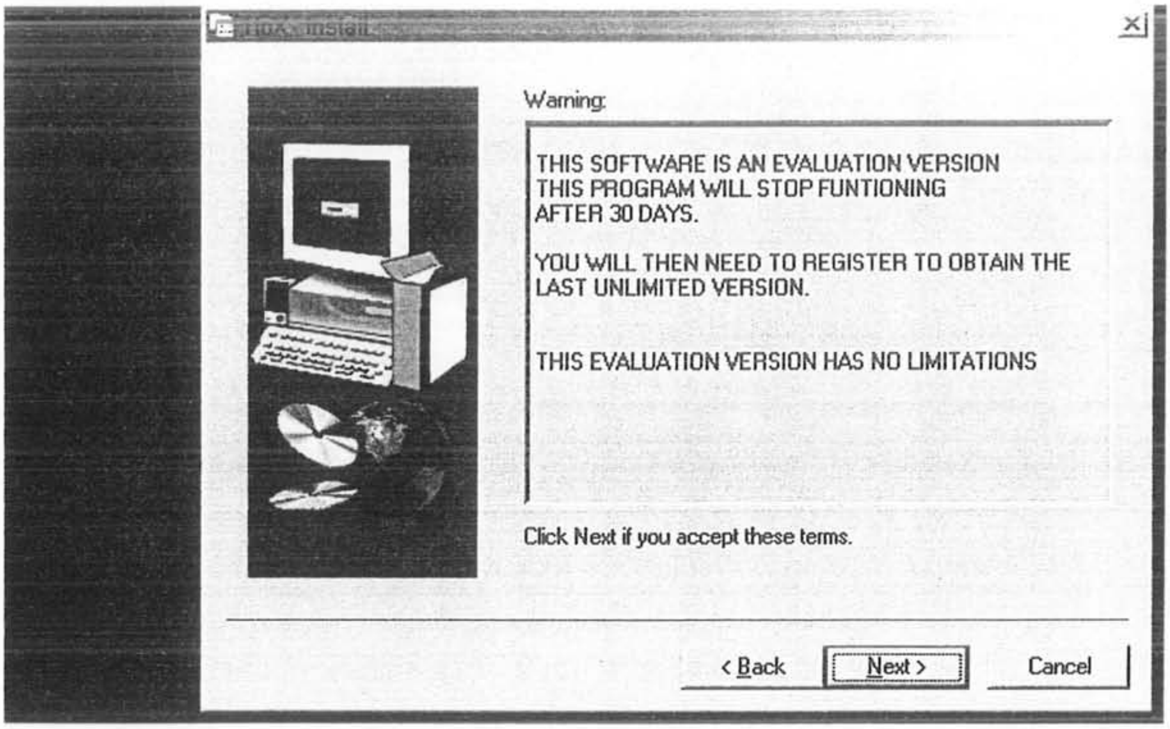
Welcome to the 1toX Setup program. This program will install 1toX on your computer.

It is strongly recommended that you exit all Windows programs before running this Setup program.

Click Next to continue with the Setup program or Cancel to exit.

Click "next".

6.1to X install dialogue box will appear…



Warning:

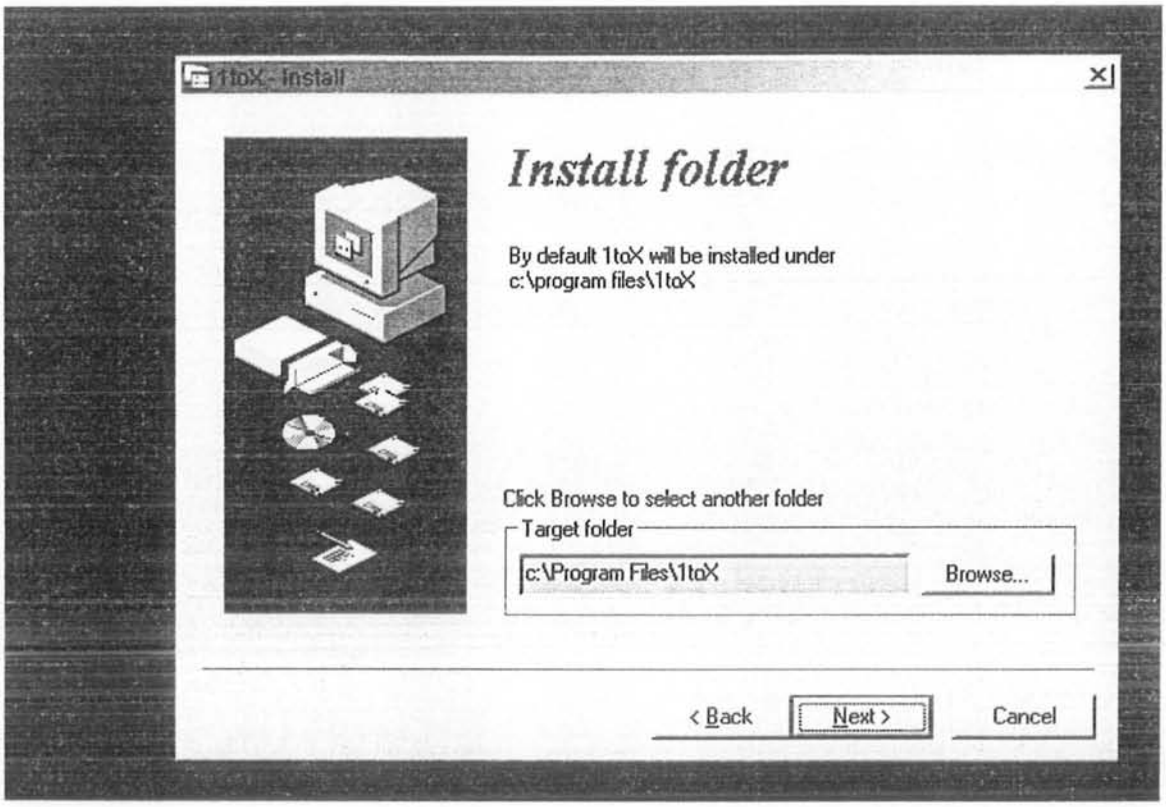
THIS SOFTWARE IS AN EVALUATION VERSION
THIS PROGRAM WILL STOP FUNTIONING
AFTER 30 DAYS.

YOU WILL THEN NEED TO REGISTER TO OBTAIN THE LAST UNLIMITED VERSION.

THIS EVALUATION VERSION HAS NO LIMITATIONS

Click Next if you accept these terms.

7. Install Folder



Install folder

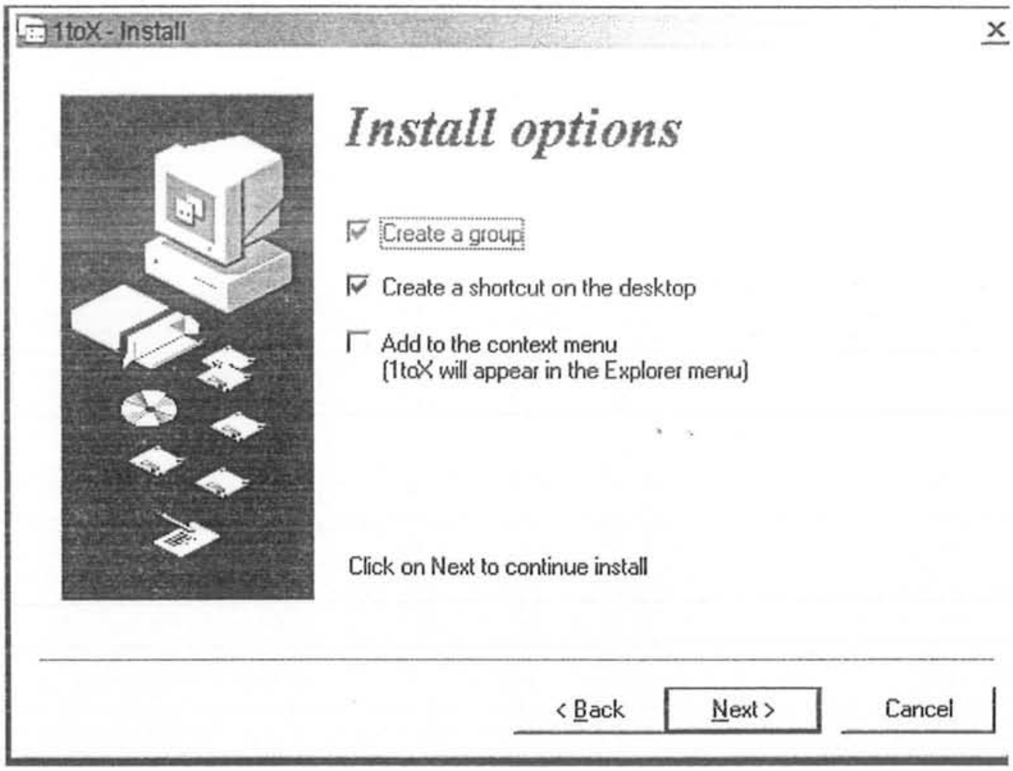
By default 1toX will be installed under
c:\program files\1toX

Click Browse to select another folder

Target folder: c:\Program Files\1toX

Click "next".

8. You will see
   C:\Program files\1to X. does not exist create?
   Click "yes".
9. Install option.



1toX - Install

Install options

☑ Create a group

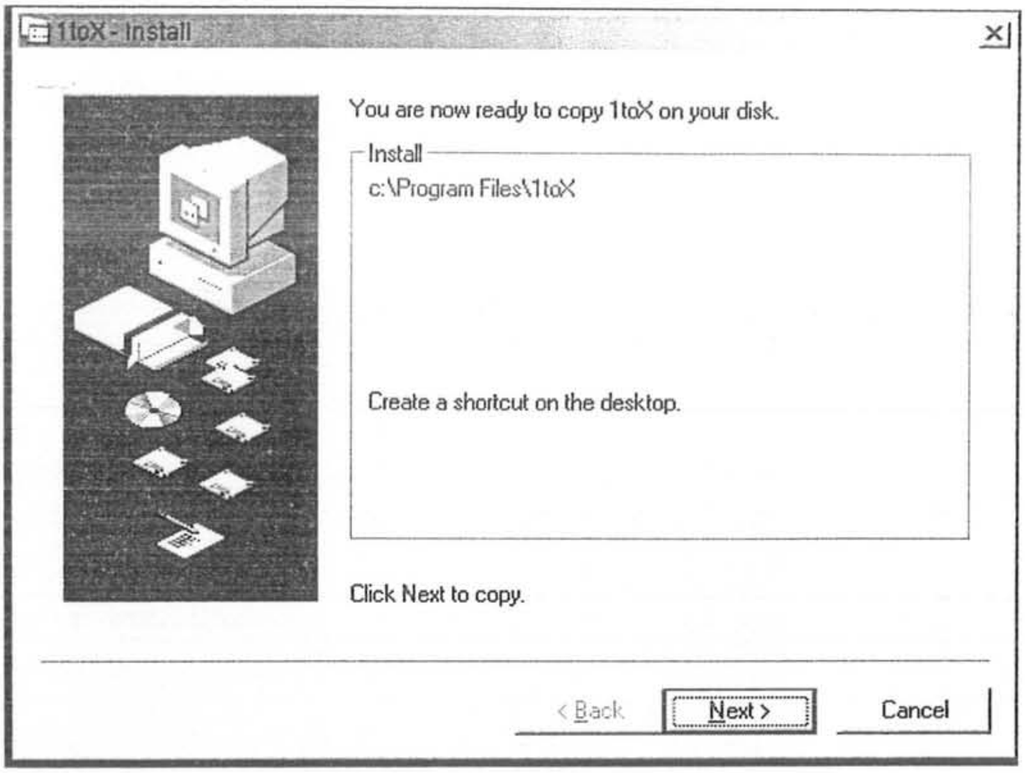
☑ Create a shortcut on the desktop

□ Add to the context menu
(1toX will appear in the Explorer menu)

Click on Next to continue install

Click “next”.

10. 1toX install


1toX - Install

You are now ready to copy 1toX on your disk.

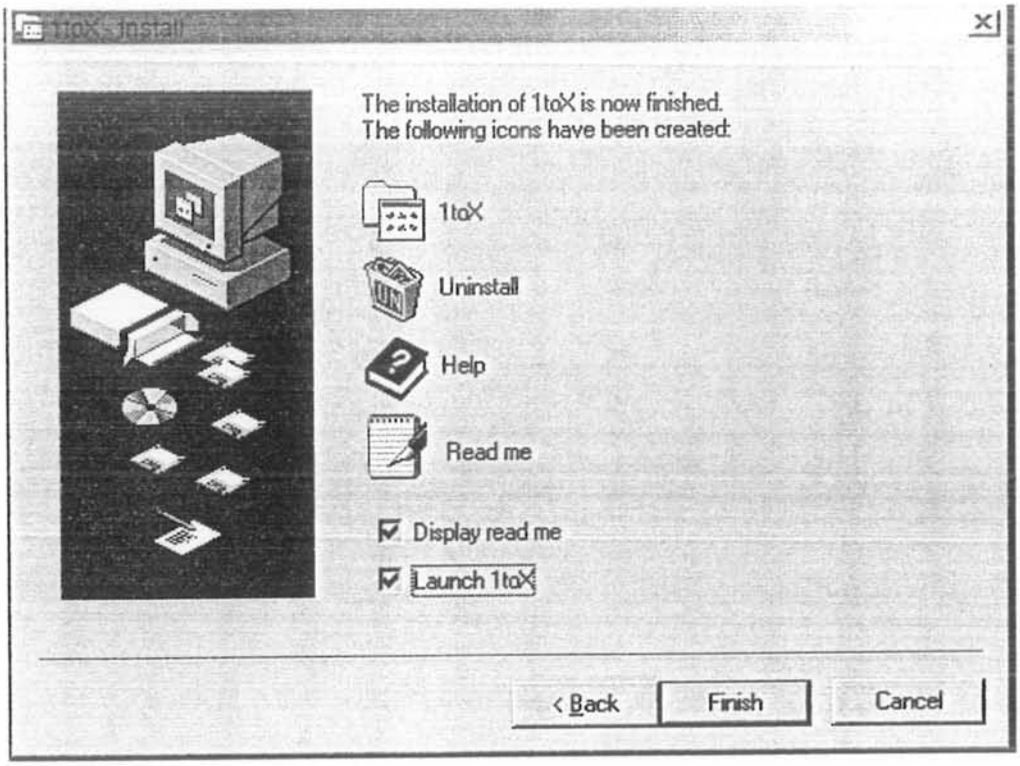
Install

c:\Program Files\1toX

Create a shortcut on the desktop.

Click Next to copy.

Click “next”.


1toX - Install

The installation of 1toX is now finished.
The following icons have been created:

1toX

Uninstall

Help

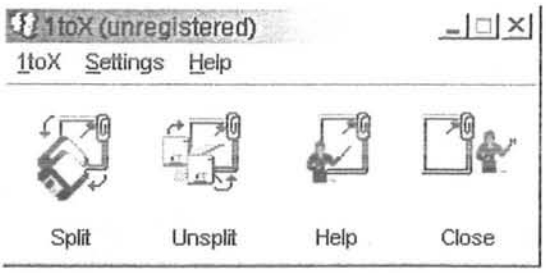
Read me

☑ Display read me

☑ Launch 1toX

10. Click “Finish”.

11. Unregister


1toX (unregistered)

a) Split

b) Unsplit

c) Help

d) Close

Click “unsplit”

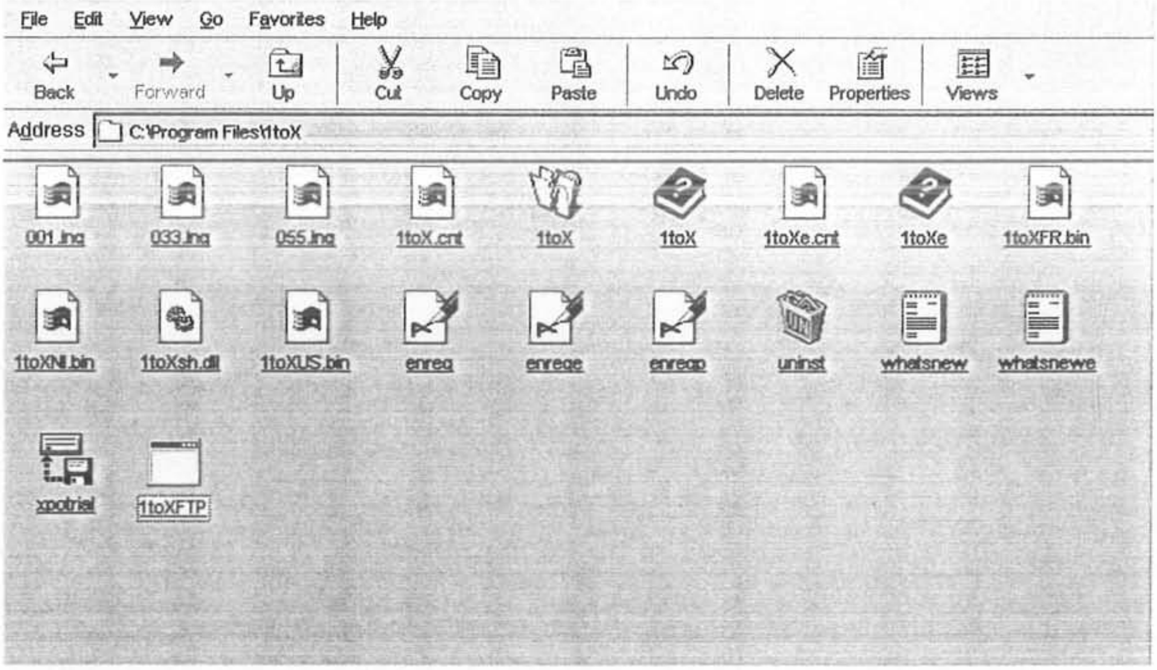
12. Open new folder (that you have created).

13. Open Xpotrial.

14. Unsplit

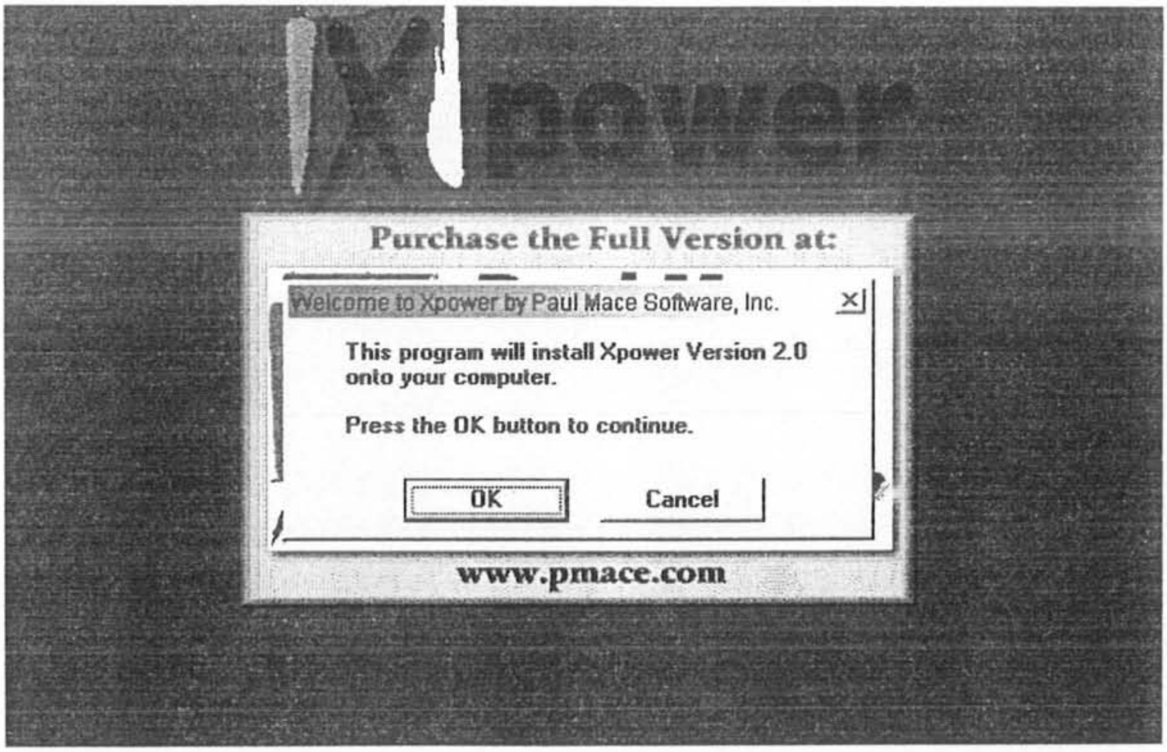
Click “ok”.

15. In program files double click “1 to X”.


Address C:\Program Files\1toX

16. Double click “ xpotrial”

17. Welcome to X-power



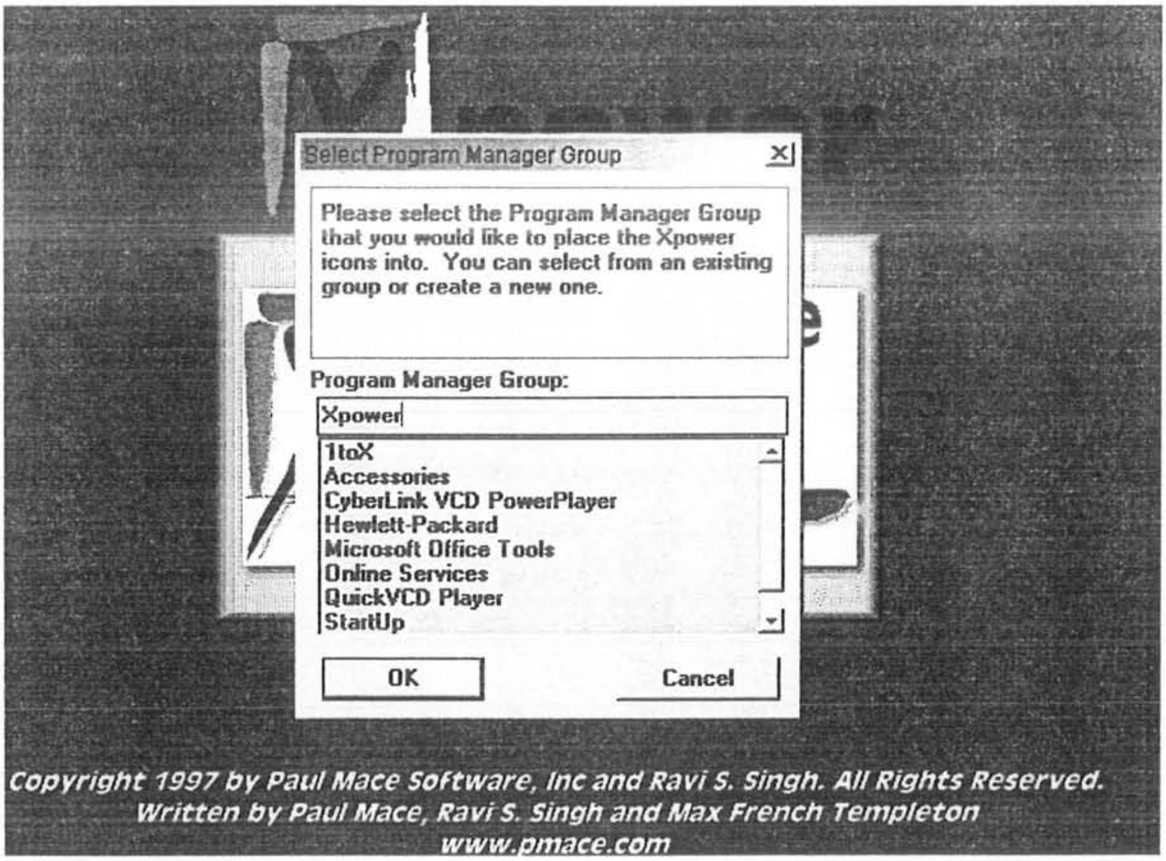
Welcome to Xpower by Paul Mace Software, Inc.

This program will install Xpower Version 2.0 onto your computer.

Press the OK button to continue.

Click “ok”.

18. Select Program Manager Group.



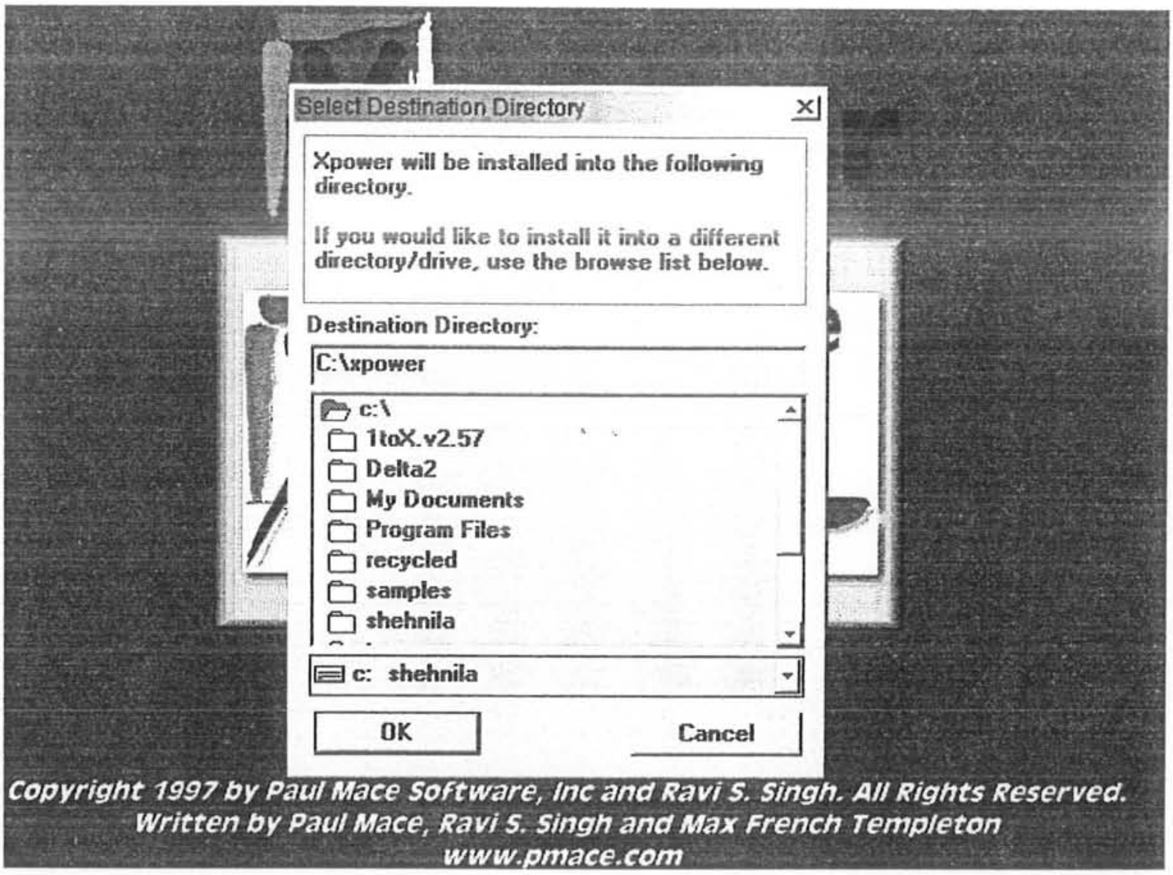
Please select the Program Manager Group that you would like to place the Xpower icons into. You can select from an existing group or create a new one.

Program Manager Group: Xpower

Copyright 1997 by Paul Mace Software, Inc and Ravi S. Singh. All Rights Reserved.
Written by Paul Mace, Ravi S. Singh and Max French Templeton
www.pmace.com

Click "ok".

19. Select directory.



Xpower will be installed into the following directory.

If you would like to install it into a different directory/drive, use the browse list below.

Destination Directory:

C:\xpower

c:\
1toX.v2.57
Delta2
My Documents
Program Files
recycled
samples
shehnila

c: shehnila

Copyright 1997 by Paul Mace Software, Inc and Ravi S. Singh. All Rights Reserved.
Written by Paul Mace, Ravi S. Singh and Max French Templeton
www.pmace.com

Click "ok".

20. Installation complete.
    Click "ok".
21. Register.
    Click "ok".

The installation of X-power is complete. You can work with it, make your own presentations and be content.

What You Get

Xpower Integrated Development Environment (IDE) includes a full-featured interactive Windows editor, a debugger which runs your applications scaled in a window, with single step, breakpoints and variable watch window, access to context-sensitive Help, the Xpower Librarian, built-in programmer's tools and sample applications.

CHAPTER # 2

G Programming Language

G is a special-purpose graphics programming language. It has commands, functions, and symbols for performing mathematical and logical operations. G is designed specifically for creating powerful, clever or complicated special visual effects and illusions, integrating them with digital imagery, animation, text and sound on a computer, and making the result fully interactive via mouse, keyboard or other device with an audience of one or more.

A G Program consists of one or more code files with a .G extension and the graphics resource files they will make use of (if any). The main program file is specified at runtime (in XPOWER it defaults to Xpower.G) , and any additional program files must be loaded with the LOAD filename command. These LOADed files become a part of the code to be interpreted and any functions or labels defined inside them may be called and executed as if they were part of a single large file.

People familiar with C or Basic, or Grasp, will find numerous similarities in G. No one ever imagined twenty years ago that you would do graphics in C, which was designed to write operating systems and compilers, or BASIC, which was a teaching tool. And no one ever imagined ten years ago that GRASP, a tool for doing tradeshow demos, would ever be expected to do things like a real language. As a consequence, none of those languages is particularly well suited to non-programmers writing for graphical operating systems like Microsoft Windows--let alone, the Internet. While G represents some of the best of what is tried and proven in those older languages, its design purpose and strengths are unique. As a new language, its shortcomings are mostly to be discovered. There is plenty of room in the design for growth and improvement, which will be driven by those who choose to write in G. Simplicity, economy of expression and readability have been balanced against the creative desire to produce transcendental effects with a single command.

CHARACTERISTICS

- **Simplicity** requires that the rules of grammar be few: Don't run words together, use commas, and spaces, (and parentheses) to meaningfully separate them.

- **Economy** of expression means use of simple, declarative statements: Play Beethoven. Increase volume. If done, repeat.

- **Readability** combines the first two requirements with a third. The words the language employs should be drawn from everyday written language and, when put together in a program, should make some literal sense. Words should not be as short or mangled as to be indecipherable.

The G command set follows the logic of an English imperative command (square brackets mean the word is optional):

English statements. Note - this code is for example only and will not run if pasted into the editor.

```
Dothis      [tothis],      [there],      [likethis, this...

eat
eat         pizza
drink
drink coke          from can.,   quickly
drink                            quickly
drink               from can
```

G statements are formed similarly, according to the same rules.

```
    dothis [tothis],      [there],      [likethis, this...

    Fade  "orangut.bmp", 3              ;fade-in with fade 3 (clock fade)
    Show "man1.bmp",          x, y      ; display bmp at coordinates x,y
```

```
Text   "ecce homo!" 0, 0          ; Latin for "i am man" at 0,0
Waitkey        ;wait for any key to be pressed
Exit        ; return to windows
```

Commands

Xpower has 60 commands that let you create Windows multimedia applications. You can display a picture, draw, paint, animate, play video, or midi sounds, WAV files, manipulate images and apply special effects, text at any angle in any font or style in any color, control devices, detect keyboard or mouse input, control program flow, debug, trace code execution, and watch variables as they change. See the list of Commands that follow.

Functions

1. Introduction

Functions are requests for information. Xpower includes over 70 functions to let you obtain picture information, manipulate text strings, read and write data files, create offscreen images and Windows control. See the Function Reference in section XII.

Unlike commands, which always do what you expect, functions involve answering questions (how wide is this image?) or performing operations that can produce many possible outcomes (write file to disk...fails because disk is full, or fails because someone else is using the file, or succeeds).

Functions in G are distinguished lexically from Commands by having their arguments contained between parentheses. For instance, bmpwidth (filename) returns the width of a bitmap image whose name is filename. Normally you would assign the answer to a variable for later use like this:

```
        width=bmpwidth ("man1.bmp")
```

Functions are not case-sensitive. Bmpwidth( ), BMPWIDTH( ) and BmpWidth( ) are all interpreted as the same function request.

Functions are not reserved words. You could have a variable bmpwidth, but it is not a commendable practice.

2. User Defined Functions

G supports user-defined functions, which operate syntactically just like library functions, except, of course, that the function must be written in G and included as part of the source code:

Example. Note - this code is for example only and will not run if pasted into the editor.

```
        clearscr                    ; clear the screen

show_file("man1.bmp")  ;display, passing filename

;when show_file returns
        exit                ;quit to windows

; show_file function assigns data passed to it a local variable 'filename'.

show_file(filename):       ; colon signifies function declaration 'filename'

; is only ;visible inside show_file

show filename,100,100   ;display the file you were ;passed

Waitkey, 2000              ;wait 2 seconds

return                  ;return to whomever called
```

Returning a value from a user-defined function is procedurally identical to library functions.
Example:

```
        width=MyFileWidth("man1.bmp")      ; ask for the
                        ; width, assign to variable width
        text width, 0, 0                ;display the results
        waitkey
        exit
```

MyFileWidth(filename): ;function definition

```
        w=bmpwidth (filename)  ; use library function
        return w                ;send result back to whomever; asked
```

User-Defined Functions are not case-sensitive. my_bmpwidth( ), MY_ BMPWIDTH( ) and MyBmpWidth( ) are all interpreted as the same function request.

User-Defined Functions may be placed in multiple files, provided you use the .G extension. Before you can call these functions, the files must be loaded with the LOAD filename command, after which the code they contain will be treated as if it were part of a single, large file.

User-defined functions should not be used as arguments as you would in C.

Bad Example:

```
                                        ; Don't try it!
        text MyFileWidth("man1.bmp"),0,0    ; It wont Work!
```

Good Example:

```
Width=MyFileWidth("man1.bmp")

Text width, 0, 0     ;Works readable too!
```

Comments

Comments are optional words used to describe your code. They are ignored by the interpreter but are helpful for reminding you what you have done. Anything following a semicolon on a line is considered a comment and will be disregarded. The following is an example of the use of a comment:

```
x=0
; now increment variable for bean counter
x=x+1
```

The following is also common:

```
x=0
x=x+1                ;now increment variable for bean counter
```

The following will add 1 to x, but do nothing to y, as everything beyond the first comment is disregarded:

```
x=0
x=x+1        ; now increment y ; y=y+1
```

Syntax and Rules

There must be a space after a command. Arguments must be separated by commas. Otherwise, spaces and tabs may be used freely for readability and style and are ignored.
Ex:

```
        Fade Clown.bmp, 10

Waitkey, 1000       ; note space between waitkey +leading comma
```

not:

```
fade Clown.bmp 5
waitkey,1000        ;this will result in an error message
```

Ex:

```
x=x+5
waitkey ,500
```

is evaluated the same as:

```
x = x + 5
Waitkey    , 500
```

With Commands or Functions, optional parameters may be omitted, but you must hold the missing arguments place with a comma.
Ex: Inside commas are mandatory.

```
waitkey ,5000
```

not

```
waitkey 5000
```

Trailing optional parameters are ignored and can be omitted.

Ex:

```
clearscr 255, 0,0
```

instead of:

```
clearscr 255,0,0,,,
```

Nonsense arguments are ignored.
Ex:

```
debug baboon        ;The argument "baboon" is ignored
```

Commands are not case-sensitive: EXIT and exit and Exit are all interpreted as the same command.
Commands names are reserved--you cannot use a reserved word as a variable name.
In G you may only have one statement per line (except when declaring large data arrays). This encourages simple, readable scripts. There is no performance

advantage to writing complex expressions, and there is nothing you can do in a complex fashion that can't be done simply.

Strings must be in quotes.

Ex:

```
Text `"This is a test", 100,100
Waitkey
```

Variables are not in quotes.

Ex 1:

```
x=10
text x,100,100
waitkey
```

Ex 2:

```
        button affirm,100,100,80,30,"OK"
        button affirm, on
        waitkey
```

Strings and Concatenation

A string consists of one or more printable characters enclosed in quotes. A string can be alpha or numeric.

Ex 1: The text enclosed in quotes is a string.

```
text "This is a test"
```

Ex 2: Strings are assigned to a variables w,x,y, and z.

```
w="1"
        x=" This"
        y=" is"
        z=" a test."
```

Concatenation is the act of combining strings.

Ex 3: Strings are assigned and concatenated.

```
w="1"
        x=" This"
        y=" is"
        z=" a test."
        result=w+x+y+z
        text result,100,100
        waitkey
```

Strings have no explicit mathematical or logical value. Consequently, the string "5" cannot be added to the string "8". (Converting strings to numbers is done with the abs() function).

Ex 4: Adding two strings results in concatenation.

```
x="100"
        y="200"
        result=x+y
        text result,100,100 ;texts out 300
        waitkey
```

Variables

1. Introduction

A variable is a name that represents a location in memory where some value is stored. You can retrieve the value by using the name, or reassign some new value, which can be a number in the range of -2,147,483,647 to 2,147,483,647 or some ASCII text (denoted by placing the characters within quotation marks).

Variable names may be up to 64 characters in length, but must not begin with a number and must not contain symbols for arithmetical or logical operations.

Variables in the G language are created the first time something is assigned.

Ex 1:

```
        message=This is a test
        text message,100,100
        waitkey
```

Variable names are not case-sensitive. X=1 and x=2 are both interpreted as assigning a value to the same variable.

Variables are created when they are first assigned a value.

Ex 2:

```
x=10  ;create variable 'x' assign it a value of 10
y=100;create the variable 'y' and assign it a value of 10
Words="Hello world!"    ; create the variable 'words' and

; assign it the legendary "Hello
                                  ; world!"
```

We can now use these stored values with the text command to display our text on the screen.

```
        Text words, x, y
        Waitkey
```

We can also modify one or more of these values and re-use them.

```
; Print hello message on eleven different lines

        Count 0,10,1        ; count from 0 to 10 by 1s
        Text words, x, y    ; write hello to screen at x, y
        y=y+10              ; add 10 to y (move down 10 pixels on
                            ; screen) and store the new value

Loop                ; keep looping back to the previous
                            ; count command until done
```

You can do mathematical and logical operations on variables as you would on any literal numbers.

```
If x==1
        text "done",0,0
        exit
endif
```

Common Error:

Assuming we have a program composed of just the following lines:

```
x=10
```

```
y=y+10       ; add 10 to y (move down 10 pixels on
        ; screen) and store the new value
```

```
Text words, x, y    ; write hello to the screen at x, y
```

Line 2 will not work because, unlike the previous example, y has no stored value to begin with.

2. Scope

A variable defined at the top level of a program is visible to all levels of the program. This is known as a global variable.

Variables defined in subroutines are only visible inside that subroutine. These are known as local variables.

Local variables and global variables cannot have the same name. This departure from programming languages such as C, Pascal or BASIC is not a defect, but a consequence of not forcing you to declare variables before using them.

If you create a local variable with the same name as a global variable, the global has precedence. The IDE will usually issue a warning if you do this.

Good programming practice suggests that you give all global variables some distinctive prefix to let you readily identify it as global. For example:

```
gIndex=1     ; a global index value
display ()
exit
display ():   ; user defined function

Index=1              ; local index
text gIndex,100,120
text Index, 100,120

return
```

System Variables

```
CPU_TYPE
XPOW_TIME
KEYPRESS
KEYCODE
KEYCHAR
MOUSE_X
MOUSE_Y
MOUSE_STATE
MSG_REPLY
WIN_COLORS
XFER_RATE
MIDI_PRESENT
AVI_PRESENT
VIDEO_X
VIDEO_Y
```

Internet Security   Read-only variables denoting security privileges.

CPU_TYPE

This holds the Intel processor pseudo-designation for the machine Xpower is running on. It will be one of the following strings: 386,486,586,686

XPOW_TIME

This number represents the elapsed time, in thousandths of a second, since your Xpower application was launched. It allows you to determine the exact time between events. For current date and time, see Date_Time ().

This example resets the counter output every 5 seconds

```
Start=Xpow_time

top:

elapsed_time=Xpow_time-start
text elapsed_time+   ,,,0
if elapsed_time>5000
        start=Xpow_time
endif

goto top
```

KEYPRESS

KEYPRESS tells whether a keyboard or mouse key has been pressed since the last time you evaluated KEYPRESS.

If no key was pressed, KEYPRESS== 0. If a keyboard key or mouse button was pressed KEYPRESS==1. The act of evaluating KEYPRESS re-sets its value to 0.

```
do
```

```
text keypress,100,150

repeat if,keypress==0

text KEY WAS PRESSED,100,150
        waitkey
```

KEYCODE

KEYCODE stores a symbolic constant corresponding to the last key pressed. G keeps this constantly updated. This is commonly tested following a WAITKEY. If the left mouse button was pressed, KEYCODE==LBUTTON;
Right mouse button, KEYCODE==RBUTTON.

If no key was pressed, KEYCODE== -1. (default) For example, you would get this if you timed out from a Waitkey command. The following short program will print out the KEYCODE of any input key.

```
top:
        waitkey
        text "Windows Virtual KEYCODE is "+KEYCODE,100,150
        waitkey ,300
        clearscr 0,0,0
        goto top
```

A list of common KEY Codes:

Key	KEYCODE (Associated Symbolic Constant)
Enter	ENTER
Backspace	BACKSPACE
Tab	TAB
Space	SPACEBAR
Up Arrow	UP
Dn Arrow	DOWN
Lft Arrow	LEFT
Rgt Arrow	RIGHT

See the Appendix Section A for a Table of Keycode and Keychar Values.

KEYCHAR

KEYCHAR stores the printable character from the character key on the keyboard that was pressed. This is true only of printable characters. For instance 'backspace', 'enter', and function keys are not printable characters and they produce a KEYCODE, but no KEYCHAR.

The following program demonstrates the printing out of a single keyboard input - (KEYCHAR)

```
top:
        waitkey
        text "Character is "+KEYCHAR,100,100
        waitkey ,300
        clearscr 0,0,0
        goto top
```

See the Appendix Section A for a Table of Keycode and Keychar Values.

MOUSE_X and MOUSE_Y

MOUSE_X holds x coordinate of current mouse cursor position--constantly updated.
MOUSE_Y holds y coordinate of current mouse cursor position--constantly updated.

```
top:
        text mouse_x+" "+mouse_y,100,100
        clearscr 0,0,0
        goto top
```

MOUSE_STATE

MOUSE_STATE holds the state of the mouse buttons.

Value	Meaning
0	no mouse button depressed.
1	left button depressed.
2	right button depressed.
3	middle button depressed on 3 button mouse.

MSG_REPLY

The MESSAGE command can have multiple buttons. This global variable lets you detect which button was pressed.

WIN_COLORS

Windows can run on machines with everything from 16 to over 4 billion displayable colors. The global WIN_COLORS variable contains the actual number of colors available on the machine. Note: FLCs will not run correctly in 16 colors you'll simply see a series of flickering bands. Also, palette considerations are important in 256 colors, but virtually meaningless at greater color depths. Check WIN_COLORS and let your application adapt.

XFER_RATE

When running over the Internet, or any TCP/IP network, resources are downloaded to the presentation site. Depending upon net traffic, the data transfer rate will vary. On a high -speed connection, it might be practical to play large AVI files, while on a modem connection it would not. Check the global XFER_RATE variable, and then let your application adapt.

XFER_RATE contains the number of bytes per second being transferred, rounded to the nearest 1000

Value	Meaning
0	0 to 500 bps
1	500-1500 bps (typical 14.4 modem)
2	1500-2500 bps (typical 28.8 modem).
3	2500-3500 bps
4	3500-4500 bps (typical 56K line)

Etc.

MIDI_PRESENT

Equals 1 if MIDI is supported, otherwise 0;

AVI_PRESENT

Equals 1 if AVI is supported, otherwise 0;

VIDEO_X, VIDEO_Y

Contain the screen resolution of the machine Xpower is running on.

The following command will cause Xpower to launch a full-screen window no matter what the target machine resolution is set to.

VIDEO video_x,video_y

Internet Security Read-only variables denoting security privileges.

Arrays

An array is a name assigned to a series of values stored in memory.
Arrays are created with unique names, just like any other variable, followed by square brackets, and the values the array will refer to, listed in order.

alphabet[ ] =a, b, c, d, e, f, g, h, i

Refer to any item in the alphabet[] array by using the name and its place (starting at zero) in the original declaration:

place	0	1	2	3	4	5	6	7	8
alphabet[ ] =	a,	b,	c,	d,	e,	f,	g,	h,	i

alphabet [0] is the letter a
alphabet [4] is the letter e

Reassign the value of an array element as you would any variable.

alphabet[3] =?

The list now shows

place	0	1	2	3	4	5	6	7	8

```
        alphabet[ ] =   a,     b,     c?    e,    f,    g,    h,    i
```

Once an array has been declared, it can be used like any other variable. Heres a more obvious example of how an array can simplify an iterative process, such as file loading:

```
;first, make an array of files to load

file[]="bird.bmp","bird.wav","bird.flc","tree.bmp","wind.wav",
"fire.wav"

count 0,5,1,cnt       ;count from 0 to 5 by 1s and use cnt
        load file [cnt]        ; cnt variable is the current count
loop                  ; loop back to previous count command until
                              ; done
```

Don't do it this way.

```
load "bird.bmp"
load "bird.wav"
load "bird.flc"
load "tree.bmp"
load "wind.wav"
load "fire.wav"
```

The powerful advantage to the first approach is that once the array is declared, you have an easy way of locating and modifying the information (in this case, filenames) for use elsewhere in your application.

The true power of indexed arrays can be demonstrated with another example. Filenames above were simply lumped together, because they are all files.

Example: Note - this code is for example only and will not run if pasted into the editor.

```
Picture [] ="lion.bmp","tiger.bmp","bear.bmp","ohmy.bmp"
sound[] = "roar.wav", "purr.wav","growl.wav","help.wav"
animation[] ="bird.flc","trees.flc"
caption[] ="King","Bengal Tiger","Kodiak Bear","Tourist"
```

```
        top:

        waitkey         ; wait for a mouse or keyboard event

if (KEYCODE==LBUTTON)
                idx=idx-1
        endif
        if idx<0
                idx=0
        endif

        if (KEYCODE==RBUTTON)
                idx =idx+1
        endif
        if idx>3
                idx=3
        endif

        show picture[idx]  ; display bitmap
        text caption[idx],0,0       ; show text
        playwav sound[idx]          ;play sound
        load animation[idx]
        playflc animation[idx],15,15,0,-1
        waitkey ,8000
        goto top
```

The example structures the data by creating parallel arrays of related items. Index 0 now refers to the picture of a lion, the sound of a lion, the appropriate lion caption and the associated animation file. Simply choosing an index number keys access to related information.

A re-declaration of an array changes its entire contents.

```
        bozo [] =a,b,c,d,e
        x=array_end (bozo)          ; obtain index of last element in array
        ; x equals final index 4 at this point
        ;bozo array contains a,b,c,d,e

        bozo [] =x,x,x
        x=array_end (bozo)          ;obtain index of last element in array
```

```
;x now equals final index 2
;bozo array contains x,x,x
```

Array elements can be reassigned one at a time.

```
bozo[1]=y
bozo[2]=z
; bozo array contains x, y, z
```

Arrays are the single most powerful programming feature in G.

Labels

Labels are single words followed by a colon that serve as execution branch points.

Example:

```
;'top' and 'display' are labels
 top:
        waitkey
        if KEYCODE==13
                goto display
        goto top
 display:
        etc.
```

Assignment, Math and Logic

1. Assignment

To make an assignment just use a single equal sign (=). A double equal sign is only used in condition testing and not used for an assignment (see below).

```
=;              Equals used for assignment
```

Example:

```
a=1          ;assign 'a' a value of 1
b=a          ;assign 'b' the value of 'a'
```

2. Math

Version 1.0 of Xpower does not allow floating point arithmetic only integer arithmetic. This means that you cannot do arithmetic with decimal numbers (floating point). Doing so will yield an incorrect result.

Ex:

```
    x=4+5
    text x,100,100          ;yields 9
    waitkey
```

Don't do this:

Ex:

```
    x=4.2+5.6
    text x,100,100      ;yields an incorrect result
    waitkey
```

In integer arithmetic Xpower, always drops the decimal remainder. Asking for the square root of 35 will yield round the answer down to the nearest whole number, 5. Asking for the square root of 36 will not.

Arithmetic Operators

+	add
-	subtract
*	multiply
/	divide
^	power
%	modulus
~	bitwise xor
<	less than
>	greater than

```
<=      less than or equal to
>=      greater than or equal to
!= or <>        not equal;
```

3. Logic and Comparison

&& is the symbol for logical AND. It returns TRUE, i.e. 1, if both expressions evaluate to be nonzero, otherwise returns FALSE (0).

|| is the symbol for logical OR. It returns TRUE (1) i.e. 1 if either of the expressions evaluate to be nonzero, otherwise returns FALSE (0).

Ex 1. Logical OR-ing.

```
    x=5
    y=6
    if (x==5)||(y==10) ;since y=5 it's true
        clearscr 255,0,0
    endif
    waitkey
```

Ex 2. Logical AND-ing

```
    x=5
    y=6
    if (x==5)&&(y==10)      ;won't be true
        clearscr 255,0,0
    Endif
    waitkey
```

When comparing expressions, remember to always group them with parentheses.

Coordinates

The coordinate system in Xpower is consistent with Windows. It is a modification of the Cartesian system. 0,0 will be in the upper left coordinate of your screen. Values increase to the right and down, to a maximum of 32763.

Negative values extend to the left and up from the origin to a maximum of (-32762).
If you were putting up clippings in Grasp for DOS, your x coordinate will be the same but your y coordinate will now be different. To calculate the new y position for your clipping in a 640 x 480 presentation:

Xpower y coordinate = 480-(old Grasp y coordinate+clipping height).

Compression

The Xpower Librarian automatically compresses resources and performs other optimizations to reduce the size of the distributed application when making the EL1.

Memory

You have access to all the memory that Windows has. This includes virtual memory. For performance reasons, if you load objects you should always free them from memory when you are done with them. See the FREE command. When Windows runs out of memory, your Xpower application will run out.

Color Palettes

G works in Windows 24 bit virtual color space. What is actually displayed will depend on the quality of the source material and the capabilities of the machine on which your G application is running.

The video hardware and device driver installed by the end user determines the actual palette depth available for display.

Images with fewer, simpler colors translate perfectly to greater color depths. Complex images with many colors will be simplified by Windows with dithering and color averaging and image quality will deteriorate. If you intend to distribute your presentation for use on machines whose hardware capabilities may be lesser than your own, understand that Xpower will run the presentation just fine, but the quality of appearance will depend on the capability of the end users machine.

If you are using 16 or 24-bit images (hicolor), you do not need to worry about color palettes. G offers two types of palette control for 256 color images.

1. Universal

Xpower offers an orthonormal PALETTE command that will allow pictures created with different palettes and color resolutions to be displayed simultaneously on screen without adversely affecting each other. All images are mapped to a universal palette. With this feature, you don't have to worry about matching the palettes of your images when displaying them simultaneously. The benefit is simplicity. G will take care mapping all images to the universal palette. The drawback is that complex images may not look as good depending on the color selections in the image.

2. Custom

For those requiring precise color controls, Xpower supports Windows virtual color palette. Although the number is a 32-bit number, Windows currently uses only 24 bits for color. A Custom palette gives you control over color numbers 10-234. Colors 0-9 and 245-255 are used by Windows for the system colors. Do not make custom palettes that alter these colors, as conflicts with other Windows applications will occur! The benefit of a custom palette is that your displayed images will closely match their original colors.

The drawback is that this requires an understanding of how pallete colors are handled in Windows 256 color mode. The Xpower Editor tools allow bulk conversion of most formats to BMPs of equivalent color range, up to 24 bits. Palette resolution is decided at LOAD or SHOW time, at your discretion, with final quality limited only by the resources used to make the application and the target machines Windows and hardware capabilities. Optionally, you can use the Xpower Editor tools to do a bulk conversion of images to a single color palette.

Note:

When Xpower is launched, it uses whatever palette Windows is set to. That palette could be a Windows system palette or a palette from your Word Processor, Paint Program etc. In this condition, when you ask for a particular rgb color, Windows will look it up in the current palette and give you the closest match. It may or may not be the same. This is adequate for the majority of users. For those who prefer to be exact, they can set Xpower's palette by loading or putting up an image. Once the palette is set by this technique, the RGB values of any particular pixel will exactly map what you put up.

Debugging

If you are getting an error try to determine which line the error occurs on. Does the error message describe what you did wrong on that line? Check the syntax on that line first. Is the command or function entered correctly? Have you shut down any other programs that may be causing the problems?

ERRORS!

While working with X-Power you can come across with errors like:

1. Generation protection error.

There is a logic error in the code or an under - process program is interrupted.

2. Floating point invalid.

The index of defined array is less than it has elements.

3. Unrecovered parsing error

Typing mistake in the command.

4. Array not defined or index out of bounds.

An array should be declared before it is used in the command.

- 5. Un-expected symbol in simple expression or Un-expected symbol in statement.

Un-expected symbol has been typed in the command.

6. Too few arguments.

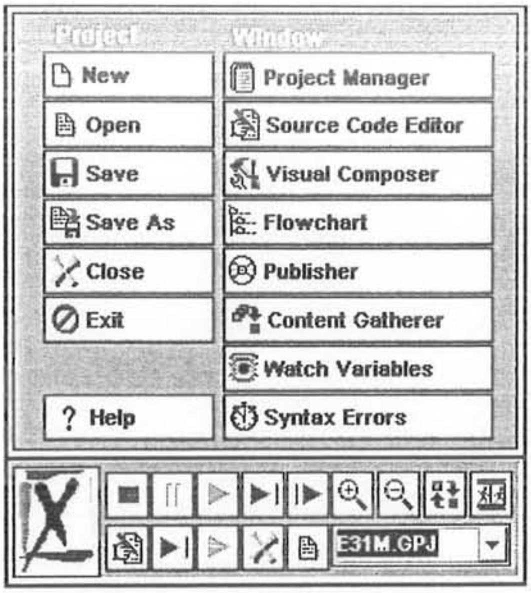
Syntax of the command is not followed.

7. Comma missing between arguments.

The space between the arguments is greater than required, for example instead of writing bmp [0...0], you have entered bmp [0. .0].

CHAPTER# 3

Xpower IDE



Xpower IDE

Overview

Xpowers Integrated Development Environment is designed to help you create, organize, debug and publish G language applications as quickly and efficiently as your talents allow.

The G language is designed to be grammatically simple and readable without sacrificing the power to write any Windows application you can imagine. The G language requires simplicity of expression. The author regards this as a virtue, though it may irritate people accomplished in other programming languages. Clarity of expression is rewarded in programming as it is in real life. Avoid cleverness if you can; it is usually a sign you have not fully understood what you are saying. Simple, declarative statements work best:

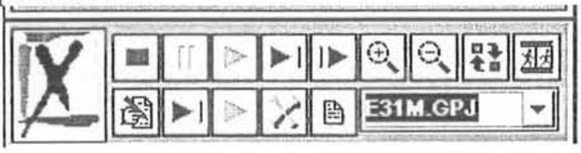
```
Text Welcome to my world
        Waitkey
```

This is a complete Windows application written in G. If you are looking for a place to begin, begin with this.

The Xpower IDE is project-oriented, a project being all the source code and resources required by your finished application. This might be a single source-code file or it might be many megabytes of images and sounds as well as a number of source files.

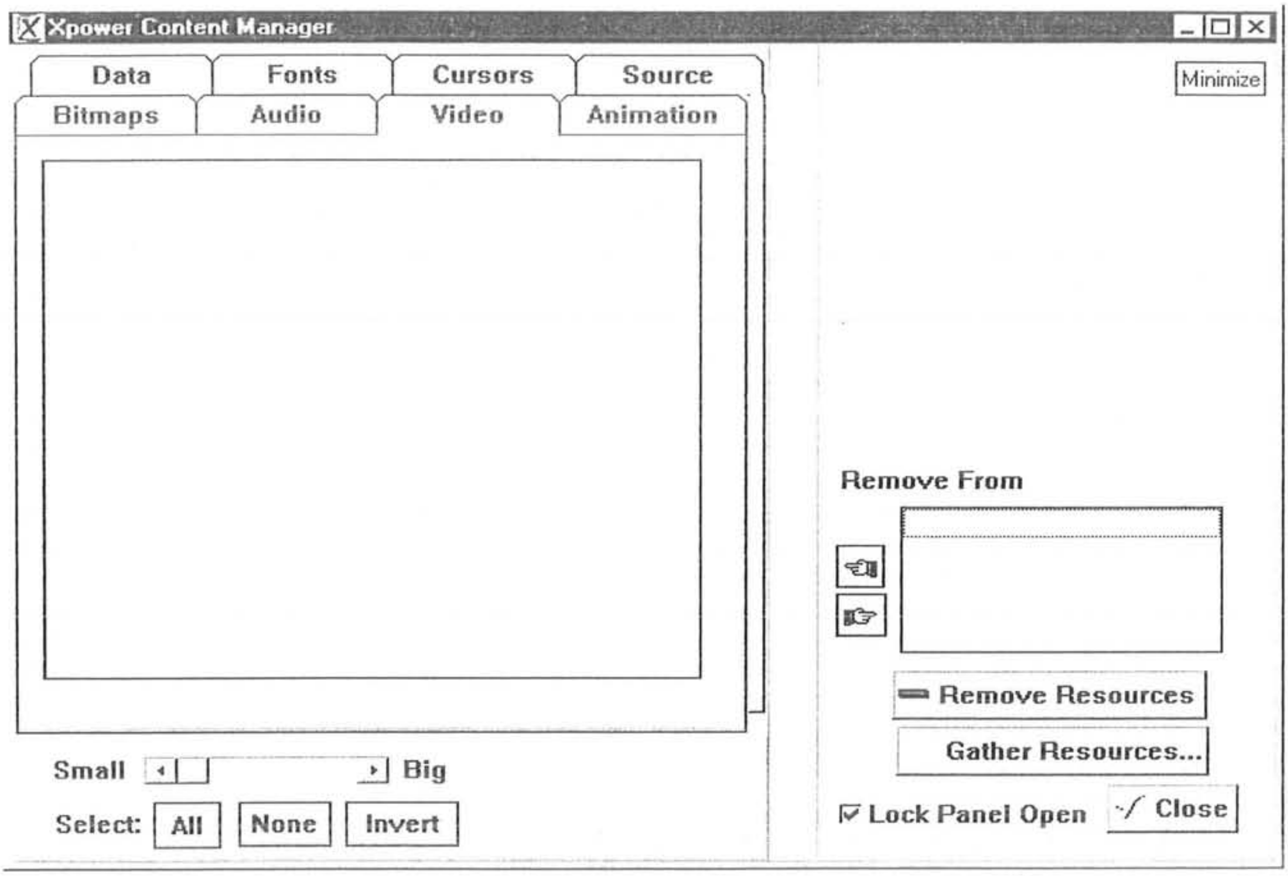
The IDE is designed to let you see what you are doing in all aspects of the development process.

The Toolbar



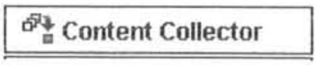
The Toolbar is the central control of the IDE. It contains the debugger controls, the icons for the tools you've used most recently or will logically need next. Touch the Xpower logo to see the main menu with project controls and controls for the other major components. This is the fastest way to maneuver the IDE.

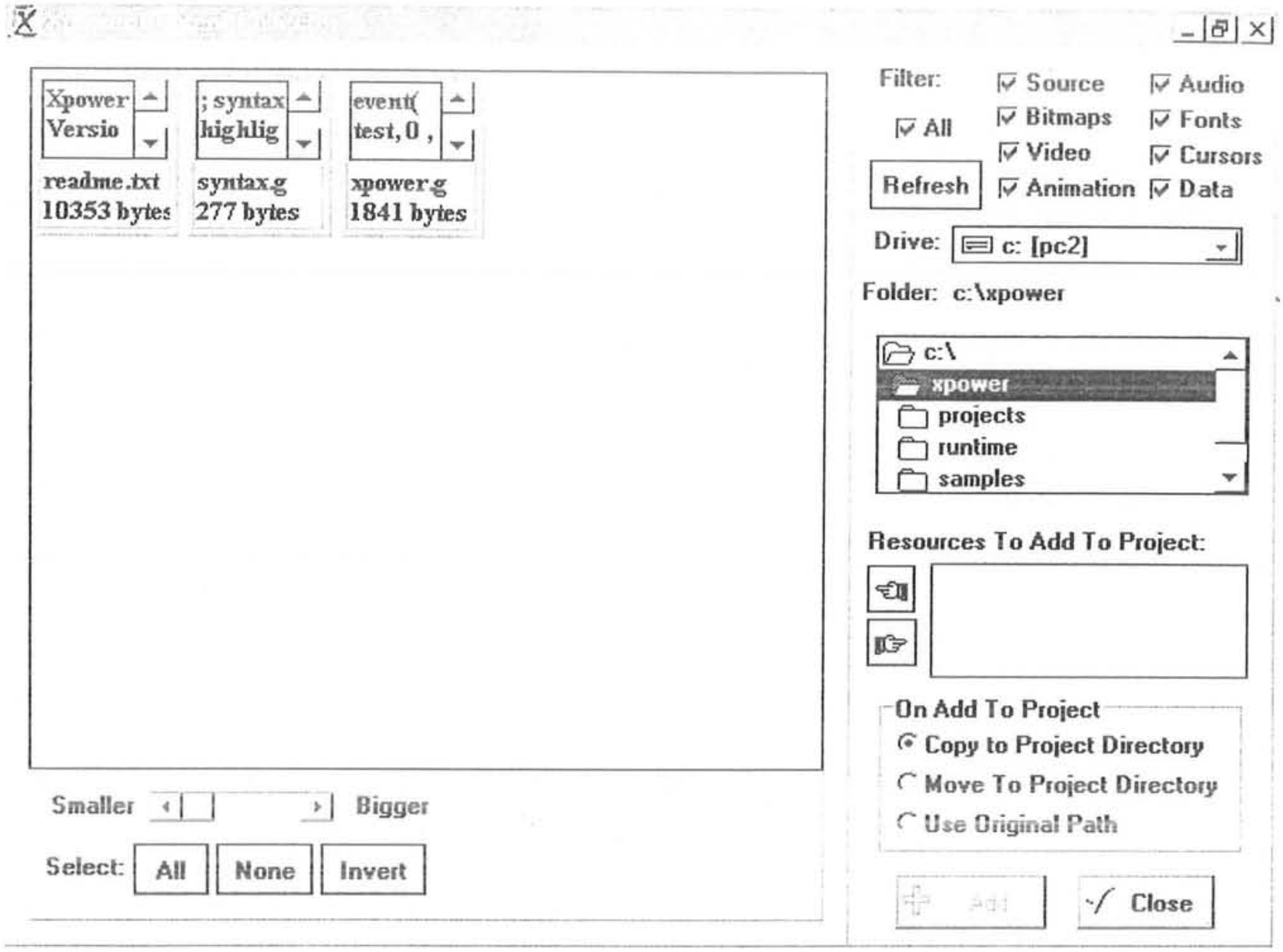
Project Manager



The Project Manager

Window contains folders for the source code files, bitmaps, sounds, video, animation, cursors and text files that go into a project. You can actually see and preview the things that go into your project. You preview and gather content into the Project Manager with the Content Gatherer, which you can get to from the toolbar menu or by touching the right edge of the Project Manager.





Filter: ☑ All ☑ Source ☑ Bitmaps ☑ Video ☑ Animation ☑ Audio ☑ Fonts ☑ Cursors ☑ Data

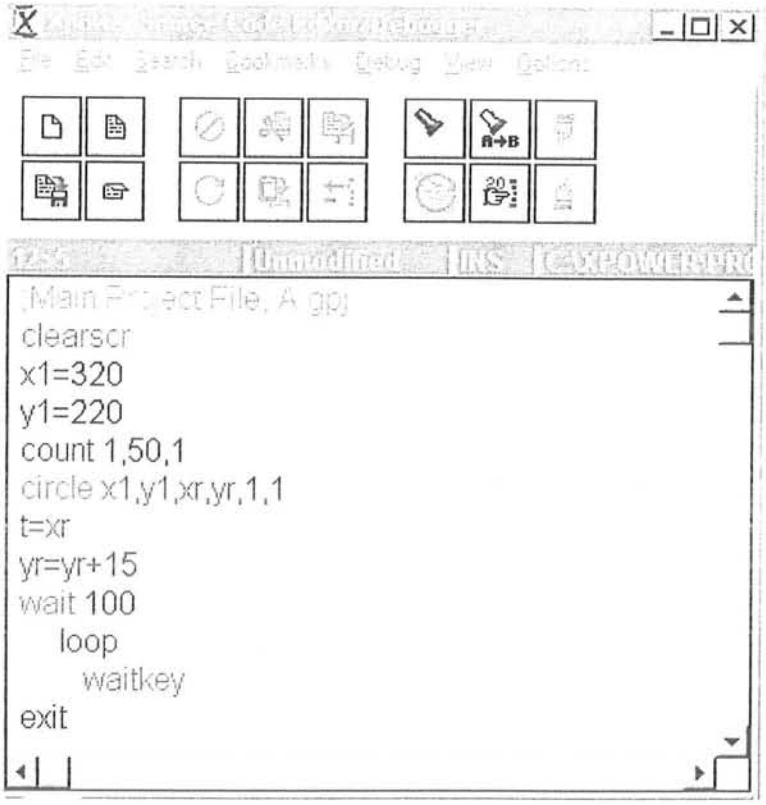
Drive: c: [pc2]

Folder: c:\xpower

On Add To Project
● Copy to Project Directory
○ Move To Project Directory
○ Use Original Path

The Content Gatherer lets you browse your system in search of the stuff of which Xpower applications are made: images, FLCs and AVIs, sound files, text files, G source files, cursors. You can preview them. When you find something you want, click on it to select it, and then click the right finger button when you're ready to add it to the list of things to be put into the project manager. You can leave the files in their current directory but we don't advise it unless you are sophisticated in your ability to manage resources. Its wiser to select the Copy to Project Directory option before clicking the Add button.

Source Editor/Debugger



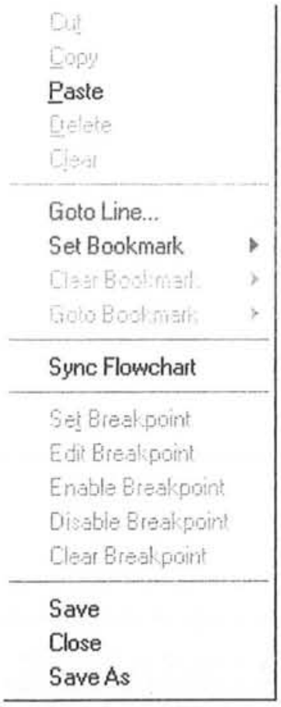
```
;Main Project File, A.gp)
clearscr
x1=320
y1=220
count 1,50,1
circle x1,y1,xr,yr,1,1
t=xr
yr=yr+15
wait 100
   loop
      waitkey
exit
```

The Source Code Editor

It is a fully equipped text editor with built-in syntax hi-lighting and G language grammar checking. By coloring parts of G language speech, the editor offers powerful visual clues to such common mistakes as missing quotation marks, accidental use of reserved words as variables, and misspelled commands. Getting help in the editor is as simple as double-clicking on something you've typed. And every time you save or run the file it gets grammar checked. You'll be warned of mistakes before you run and fail. The editor comes to you set to a default tabbed file design, but the Options menu allows reconfiguration of many aspects of editor appearance and behavior. This makes the Xpower editor in every respect as powerful as any other professional development tool on the market.

Editor options and features are available through the menu, through the icons at the top of the edit window, or by right clicking the mouse on the editor window to see the following quick selections:

SOURCE CODE (Tools)



File Handling:

New

Opens a new file. If you specify a G file and there is no current project, you will be prompted to create a new project.

Open

Opens an existing file.

Save

Saves the currently selected file.

Close

Closes the currently selected file under a new name.

Undo/Redo:

Undo

Undoes the last edit change.

Redo

Reverses the last Undo.

Cut/Copy/Paste/Delete:

Cut

Removes the selected text and places it on the Windows clipboard.

Copy

Copies the selected text to the Windows clipboard.

Paste

Inserts the contents of the Windows clipboard at the current cursor position.

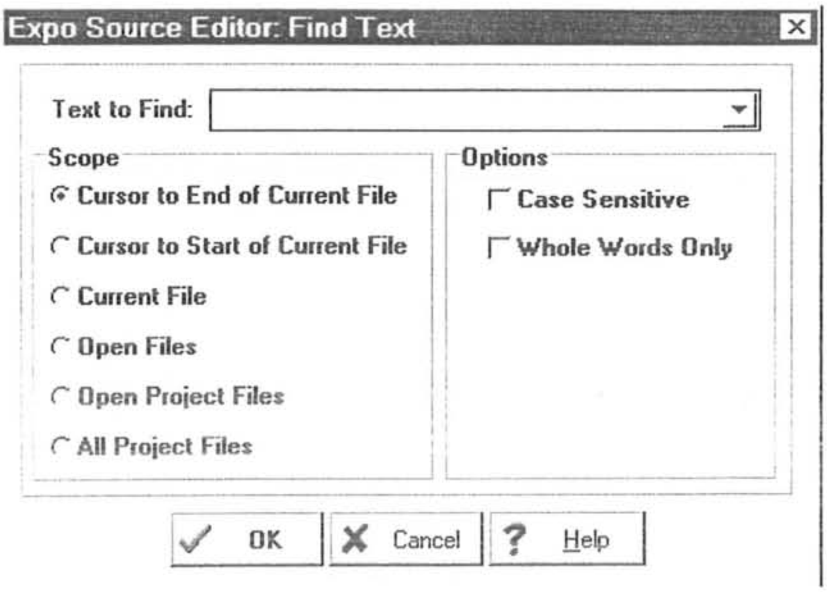
Delete

Removes the selected text.

'Replace/Find Again/Goto Line Number:

Find (or Ctrl-F)

Locates a text string:



Expo Source Editor: Find Text

Text to Find:

Scope

● Cursor to End of Current File

○ Cursor to Start of Current File

○ Current File

○ Open Files

○ Open Project Files

○ All Project Files

Options

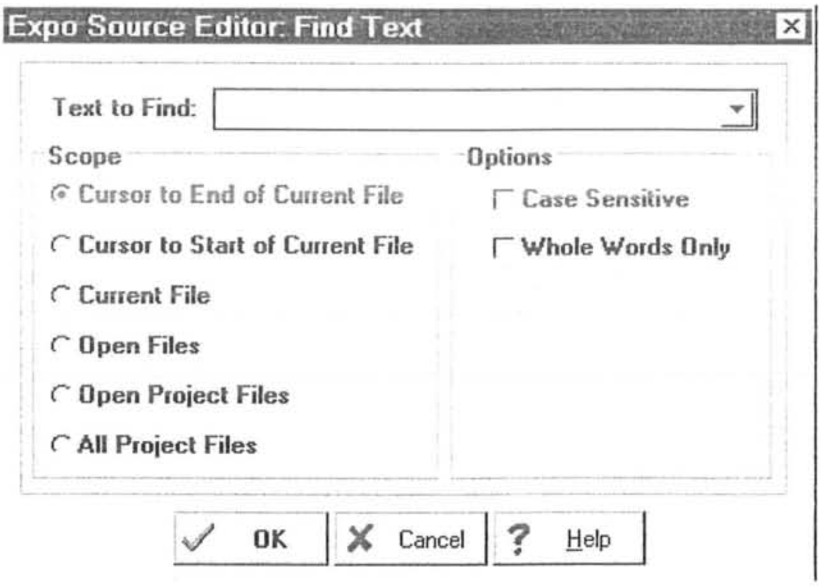
□ Case Sensitive

□ Whole Words Only

e the default; note also, you have the ability to search one or more files in any ction.

Replace (or Ctrl-R)

Replaces one text string with another:



Expo Source Editor: Find Text

Text to Find:

Scope

● Cursor to End of Current File

○ Cursor to Start of Current File

○ Current File

○ Open Files

○ Open Project Files

○ All Project Files

Options

□ Case Sensitive

□ Whole Words Only

Note the default; note also, you have the ability to search and replace text in one or more files in any direction.

Find Again (or F3)

Repeats the last Find

Goto Line Number (or Ctrl-L)

Goes to the specified line number in the current file.

Bookmark:

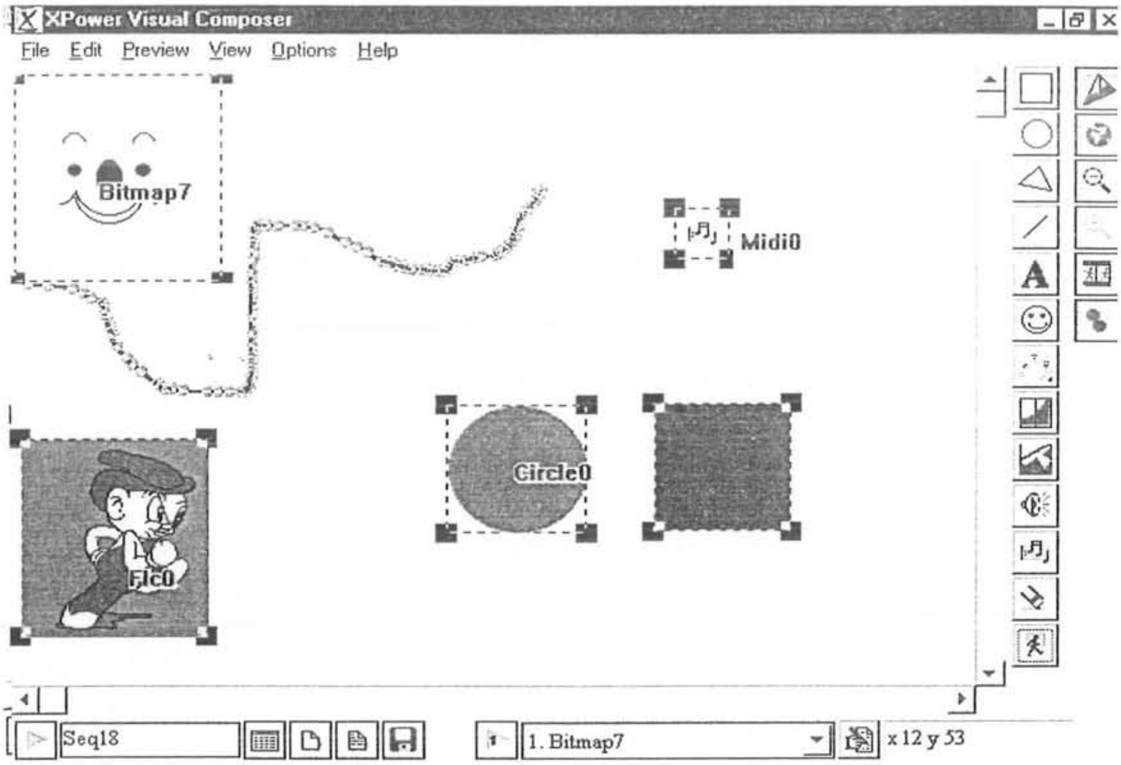
Next Bookmark (or Ctrl-N)

Goes to the next bookmark, if any.

Previous Bookmark (Ctrl-P)

Goes to the previous bookmark, if any.

Visual Composer



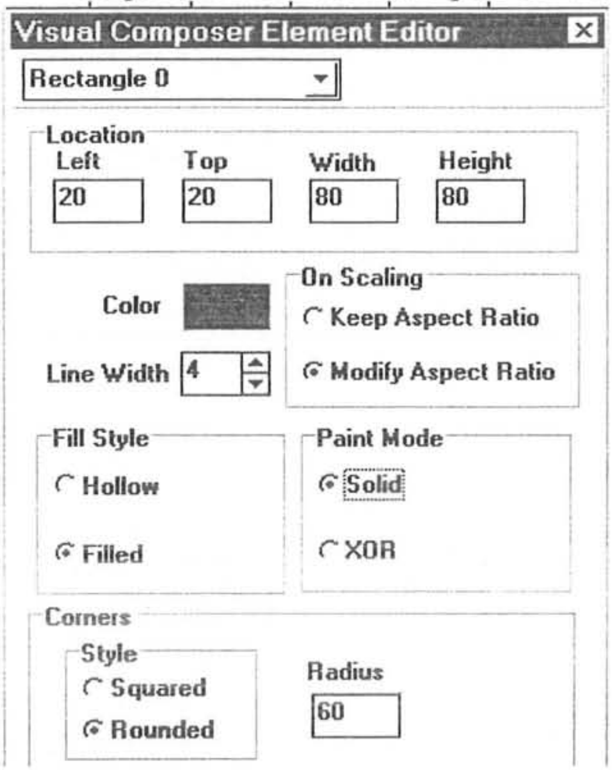
The Visual Composer lets you create visual sequences visually, as opposed to the abstract method of writing code and then running the application to see what you said. This allows you to play with a number of visual elements at once to compose them before you incorporate the resulting code into your application. Typing SHOW tiger.bmp, 100,100 is a perfectly good command, and might do exactly what you intend. But you won't know for sure until you run the application and see. Dragging tiger, bmp to the location you want on the current background and positioning it visually alongside other elements is unambiguous. The Visual Composer writes the command precisely, based on your actions and preferences.

The (default) white drawing area represents the actual viewing area of your presentation as determined by the Video commandthe black borders allow you to position elements partially overlapping the edges, or to start and end fly animations offscreen. If you are working on a sequence within a larger presentation, the viewing area defaults to the image currently on the screen at the time this sequence begins.

VISUAL COMPOSER(Tools)

Edit

Edit current element shows the properties dialog. You can select another element from the pull down box containing the element name here Rectangle 0. You can also select the entire Sequence from the same pull down.



Visual Composer Element Editor

Rectangle 0

Location

Left	Top	Width	Height
20	20	80	80

Color ▨

Line Width 4

On Scaling
○ Keep Aspect Ratio
● Modify Aspect Ratio

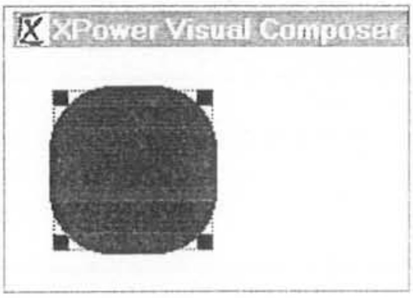
Fill Style
○ Hollow
● Filled

Paint Mode
● Solid
○ XOR

Corners
Style
○ Squared
● Rounded

Radius 60

<u>Rectangle</u> Draws a Rectangle or square.



Left mouse button lets you move and resize rectangle.

Shift-left click lets you lock multiple images so they move together.

from the same pull down.

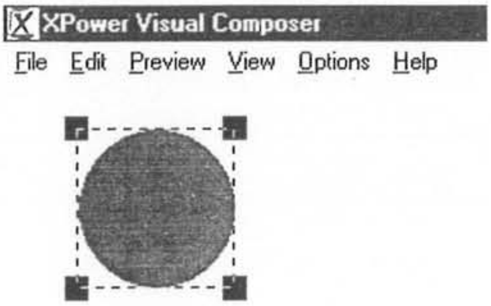
The controls are interactive with the rectangle and their behavior is straightforward. But three bear special mention:

Click on Color to get a color selection palette.

The Scaling Aspect Ratio buttons allow you to keep the proportions of the rectangle fixed while scaling or moving.

The Paint Mode buttons allow you to XOR the rectangle with the screen. The important part of XOR is that the same color, if XORed again, restores the screen to its original state. You can test this by creating two rectangles the same color and dragging them so they overlap. When you turn XOR on, notice where they overlap the underlying screen looks normal. XORing is what a mouse cursor does to keep from destroying the screen image as it moves around.

○ <u>Circle</u> Draws an ellipse or circle.

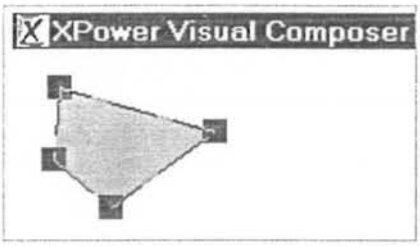


Click on Color in the Visual Composer Editor to get a color selection palette.

The Scaling Aspect Ratio buttons allow you to keep the proportions of the ellipse fixed while scaling or moving.

The Paint Mode buttons allow you to XOR the ellipse with the screen. The important part of XOR is that the same color, if XORed again, restores the screen to its original state. You can test this by creating two circles or ellipses the same color and dragging them so they overlap. When you turn XOR on, notice where they overlap the underlying screen looks normal. XORing is what a mouse cursor does to keep from destroying the screen image as it moves around.

△ Polygon Draws a polygon.



With the mouse cursor, left-click to begin outlining the polygon region. Continue to click on additional points. Double click to finish -- Xpower automatically connects the last point and first point.

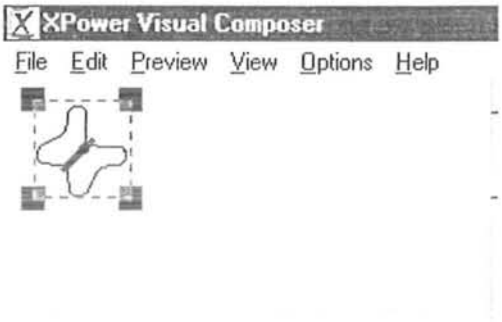
Adjust the polygon by clicking and dragging the square handles.

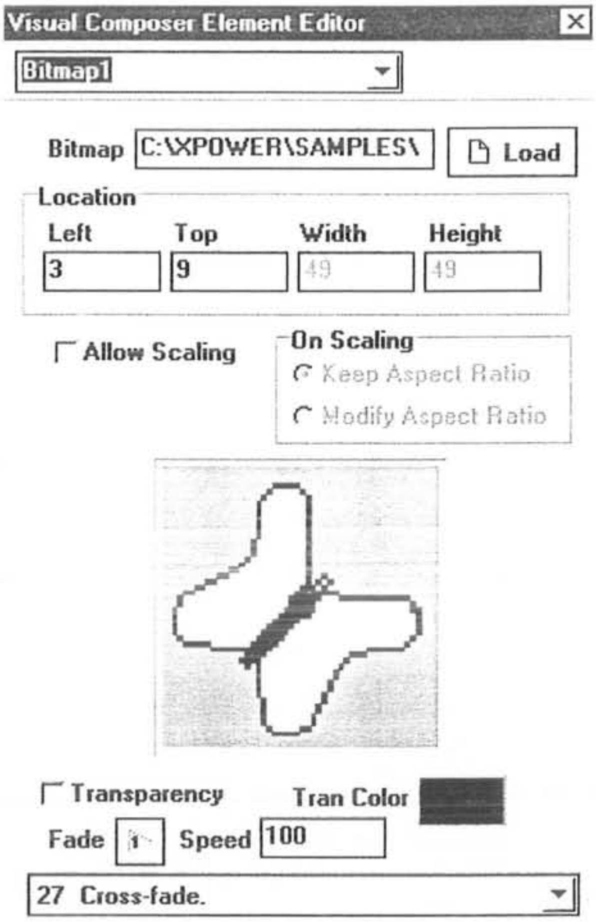
Insert a new vertex handle by double-clicking on a line joining two handles.

Delete a vertex by double clicking on a handle and answering yes to the Delete Vertex? query.

Shift-left click lets you lock multiple images so they move together.

☺ Bitmap SHOW or FADE a bitmap image.





Bitmap1

Bitmap C:\XPOWER\SAMPLES\

Location

Left	Top	Width	Height
3	9	49	49

□ Allow Scaling

On Scaling
● Keep Aspect Ratio
○ Modify Aspect Ratio

□ Transparency Tran Color

Fade 1 Speed 100

27 Cross-fade.

The controls are interactive with the bitmap image and their behavior is straightforward. But several bear special mention:

Load allows you to select a new image to be displayed.

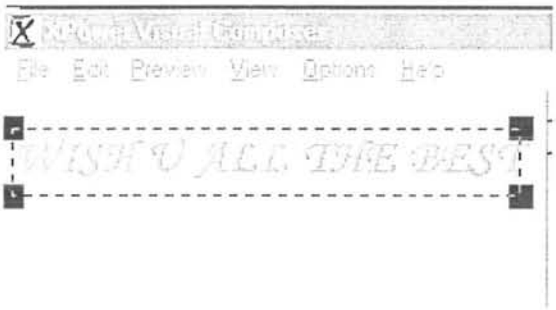
The Allow Scaling and On Scaling Aspect Ratio buttons allow you to keep the proportions of the image fixed while scaling or moving.

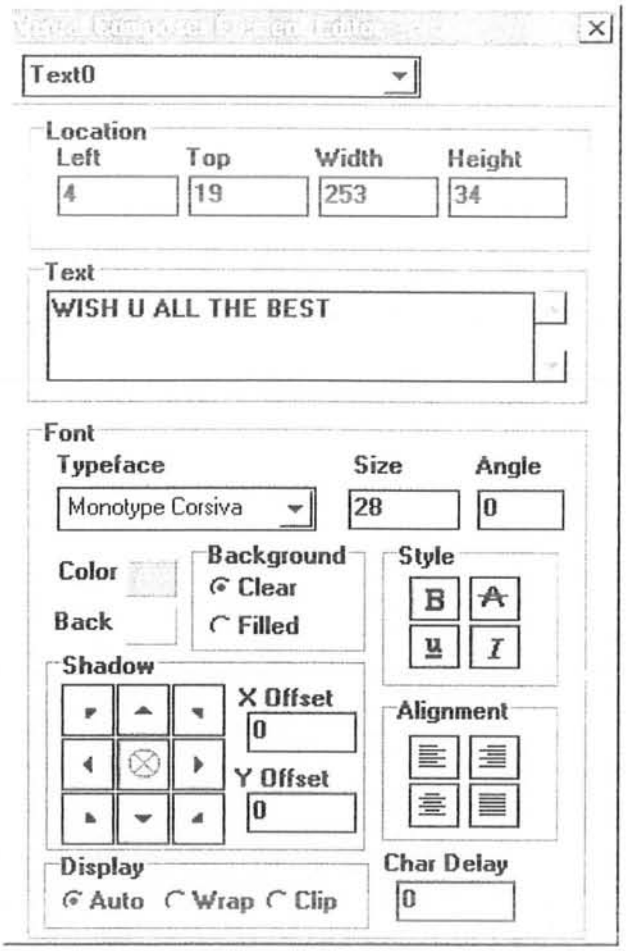
Transparency allows a color in the image to become invisible. In the example transparency is on and black is selected, so the color black does not appear when the image is displayed.

If Transparency is selected, click on the image to select the color you want transparent.

Fade selects one of 50 special effects to be applied to the image when it is displayed. Speed affects the speed of the Fade effect.

A <u>TEXT</u> Add or adjust text over background.





Text0

Location

Left: 4
Top: 19
Width: 253
Height: 34

Text: WISH U ALL THE BEST

Font

Typeface: Monotype Corsiva
Size: 28
Angle: 0

Background: ● Clear ○ Filled

Shadow: X Offset 0, Y Offset 0

Display: ● Auto ○ Wrap ○ Clip

Char Delay: 0

The controls are interactive with the text and their behavior is straightforward. But several bear special mention:

Angle rotates the text counter-clockwise. By experimenting you will notice that multi-line text does not rotate in a block in this version. Also, if text angle is greater than 0 you cannot preview delay in this release. Delay does work on angled text...you just cant preview it.

Click on Color or Back to get a color selection palette for the text and text background.

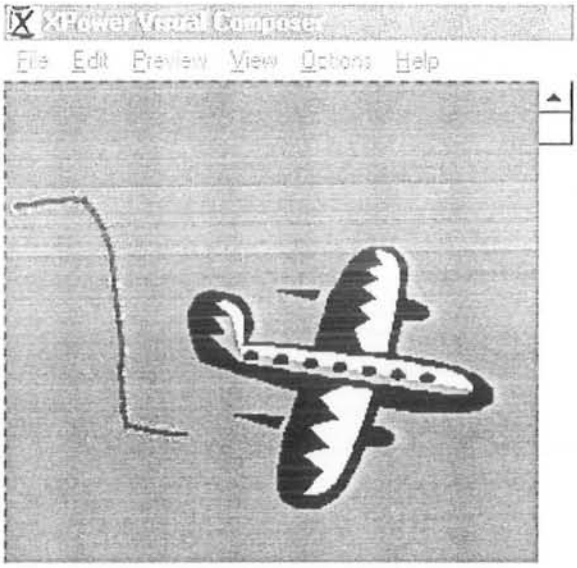
Shadow uses the Back color. Assuming a white light, the color of the shadow bears no relation to the color of the text itself; it depends upon the color of the surface the text is written over.

To achieve the best effect, the basic rule is: the shadow is a darker hue of the color the text is being written over. If the text appears over a blue background, the shadow would be a darker blue. If the text appears over a white background, the shadow would be gray. Shadows do not appear on black backgrounds make the background dark gray and make the shadow black.

Display lets you auto-adjust

Character delay causes a typing effect to be added to the text display. As noted above under Angle you cannot preview Character delay when text is angled in this release delay does work, you simply can't preview it.

FLY Draw a straight or smoothly curved path along which bitmap images are flown.



Use the mouse cursor to click on a series of points that define the critical points on the path.

OR

Drag with the left mouse button down and the path will be fitted smoothly to your freehand movements, with acceleration and deceleration of the mouse reflected in the flying behavior.

OR

Combine the two methods, clicking some sections and dragging others freehand.

THEN

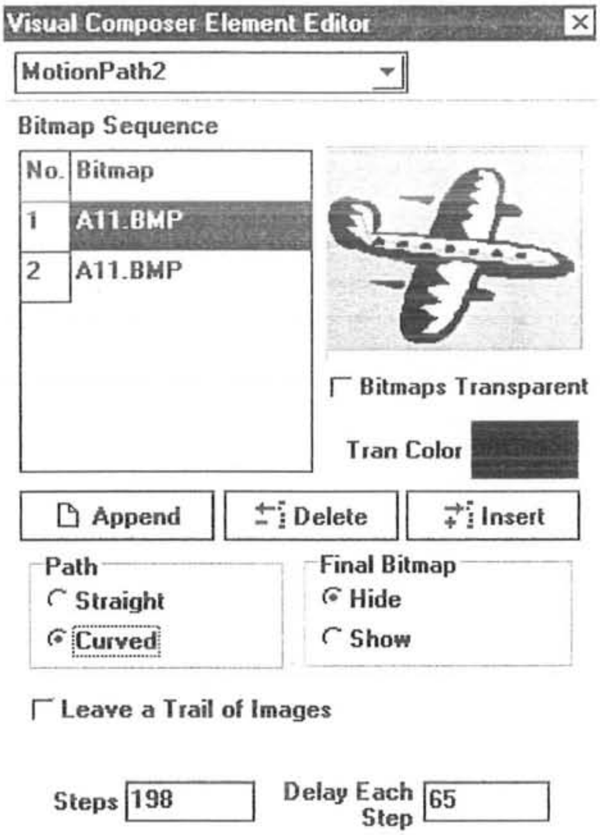
Double click to end the operation.

These are the control point handles. They let you grab and alter the shape of the path. The upper-left corner of the image(s) will pass through these points as it flies, no matter what shape the path takes

Adjust the window by clicking and dragging the square handles.

Insert a new vertex handle by double-clicking on a line joining two handles.

Delete a vertex by double clicking on a handle and answering yes to the Delete Vertex? Query.



Visual Composer Element Editor

MotionPath2

Bitmap Sequence

No.	Bitmap
1	A11.BMP
2	A11.BMP

□ Bitmaps Transparent

Tran Color ■

Path
○ Straight
● Curved

Final Bitmap
● Hide
○ Show

□ Leave a Trail of Images

Steps 198

Delay Each Step 65

Bitmap Sequence is a list of one or more images to fly. You create the list by using the Append Delete or Insert buttons. You can alter the order in the list by dragging and dropping the numbers.

Bitmaps Transparent allows a color in the image to become invisible. In the example transparency is on and black is selected, so the color black does not appear when the image is displayed.

If transparency is selected, click on the image to select the color you want transparent.

Leave a Trail of Images causes each frame to be left on the screen.

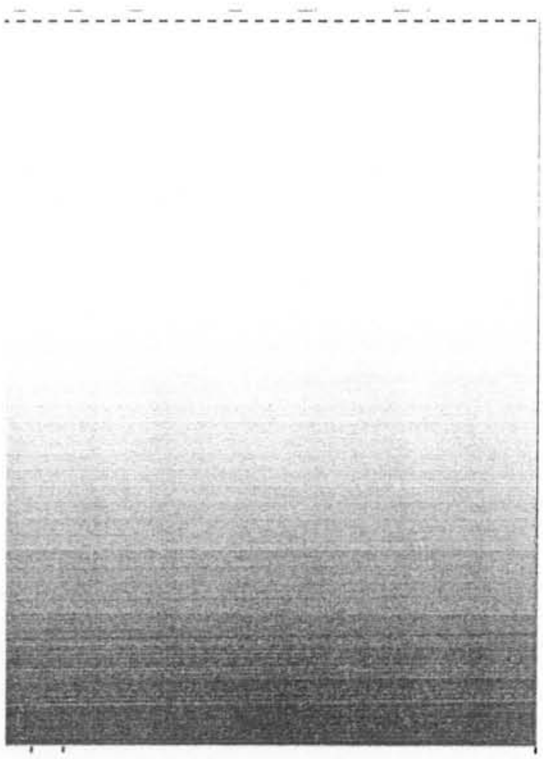
Path allows you to select whether the image(s) fly in a smoothly curved path through the control points, or straight.

Final Bitmap allows you to leave the image visible at the end of the fly, or have it vanish.

Steps lets you set the total number of frames displayed in the course of the fly. These steps are evenly divided between control points. This allows you to produce acceleration and deceleration. The farther apart the control points, the farther the image(s) will travel between frames (acceleration); the closer the control points, the less distance traveled between frames (deceleration). Evenly spaced control points produce constant, linear motion.

Delay Each Step controls the timing of the animation. The default, 65 milliseconds, produces 15 frames per second.

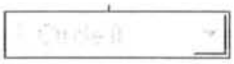
<u>Clear Screen</u> Clear the screen with color or pattern.



this illustrates the default style, Top-to-Bottom, gradient, black and white.

NOTE: When viewing in single frame mode, once a clear screen is performed, all elements are covered up and un-selectable.

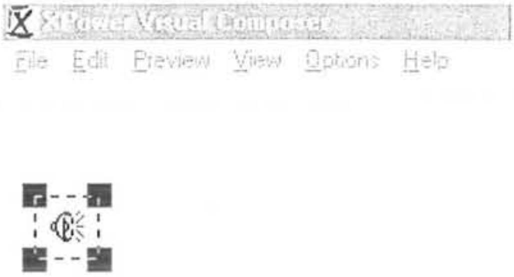
You can either switch to multi-frame view, select elements from the pull down list.



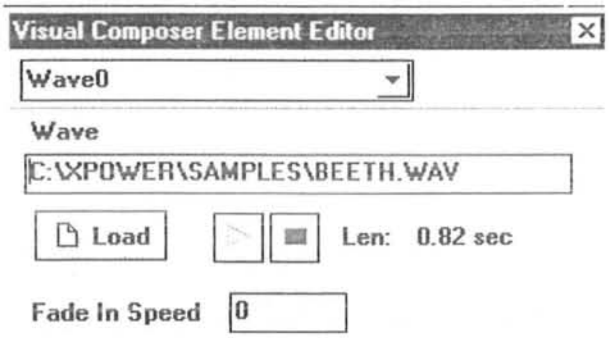
Or toggle the Select All button to let you see the other elements outlined

Right-click with the mouse to edit the Clear Screen element, or…

🔊 Play Wave Play a WAV sound file.



NOTE: the speaker icon represents the start of a .WAV audio file, but it will not appear in your actual program when it runs; its size and position are insignificant.



Wave0

Wave: C:\XPOWER\SAMPLES\BEETH.WAV

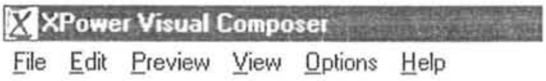
Len: 0.82 sec

Fade In Speed: 0

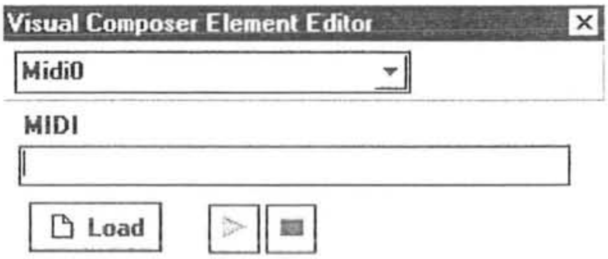
Load lets you pick the WAV file you want to play (and preview it with the VCR controls).

Fade In Speed sets the sound fade-in speed in seconds. The Visual Composer will bring the sound volume up from zero to the current volume setting.

Play Midi   Play a MIDI track a polygon.



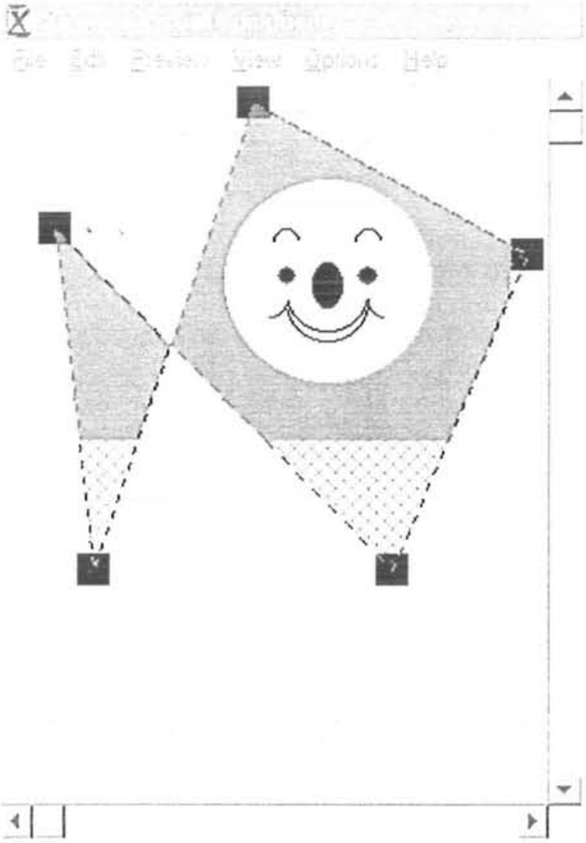
NOTE: the musical staff icon represents the start of a .MID sound file, but it will not appear in your actual program when it runs; its size and position are insignificant.



Load lets you pick the MIDI file you want to play (and preview it with the VCR controls).

WINDOW Draw polygons which define a transparent window .

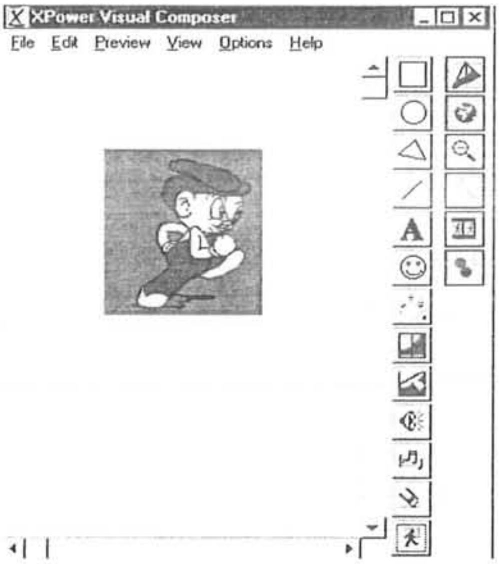
Note: windows are applied to objects drawn after the window is created. If you have more than one active window the windows are logically combined as shown below. Windows remain in force until a Window-reset is ordered, at which time the entire presentation becomes visible.



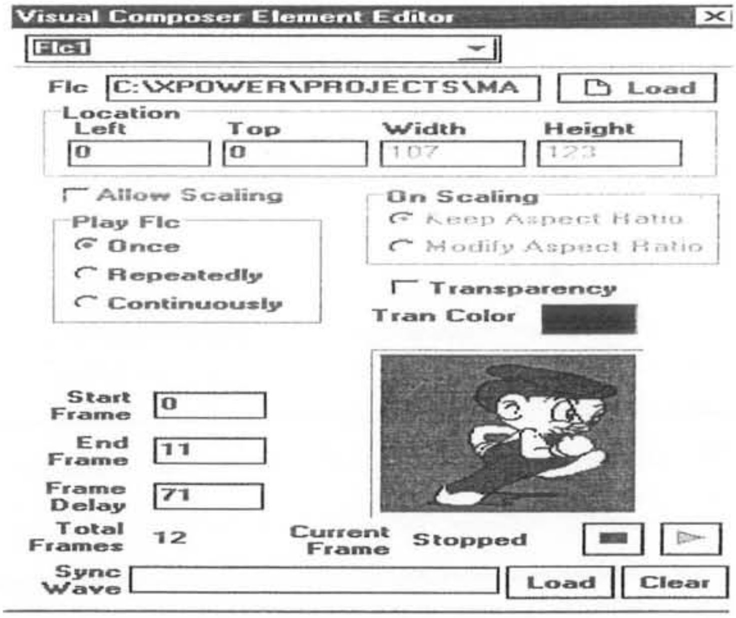
Fade Out Fade-out an object, restoring the screen.

Play FLC Animation

Play FLC Animation plays a FLC animation files. NOTE: you can play multiple FLCs but if they are large (or stretched), or there are too many of them, or you are running on a modest 486 or older Pentium machine, you may not see them run in the final application. You can improve this situation by making the Frame Delay rate higher, or cutting down the number of simultaneous FLCs or their size.



Edit current element shows the properties dialog. You can select another element from the pull down box containing the element name. You can also select the entire sequence from the same pull down.


Visual Composer Element Editor

Flc1

Flc C:\XPOWER\PROJECTS\MA Load

Location

Left	Top	Width	Height
0	0	107	123

□ Allow Scaling

Play Flc
- ● Once
- ○ Repeatedly
- ○ Continuously

On Scaling
- ● Keep Aspect Ratio
- ○ Modify Aspect Ratio

□ Transparency

Tran Color ▇

Start Frame 0

End Frame 11

Frame Delay 71

Total Frames 12

Current Frame Stopped

Sync Wave

Load Clear

Allow Scaling lets you stretch or shrink the FLC. NOTE: if you stretch the FLC to greater than 320x200 (65535 pixels) it will usually run much slower.

Play Flc determines how many times the FLC plays.

Once plays one-time only.
Repeatedly brings up a repeat counter, which you can set.

Continuously means loop forever.

Start Frame let's you begin the FLC at an arbitrary frame.

End Frame let's you end the FLC at an arbitrary frame.

Frame Delay controls the speed of playback This is set by default to the delay value stored in the FLC when it was created. Smaller numbers play faster (65 is 15 frames per second) but the size of a FLC and the power of the computer can affect playback rate and quality.

Total Frames is the number of frames in the FLC.

Sync Wave let's you choose a WAV file to play with the FLC. The sound will be synchronized with the FLC--that means if the .WAV file was recorded and timed to the action of the FLC the two will be synchronized as they would in an AVI or MOV. You Load WAV files and Clear this feature with the buttons.

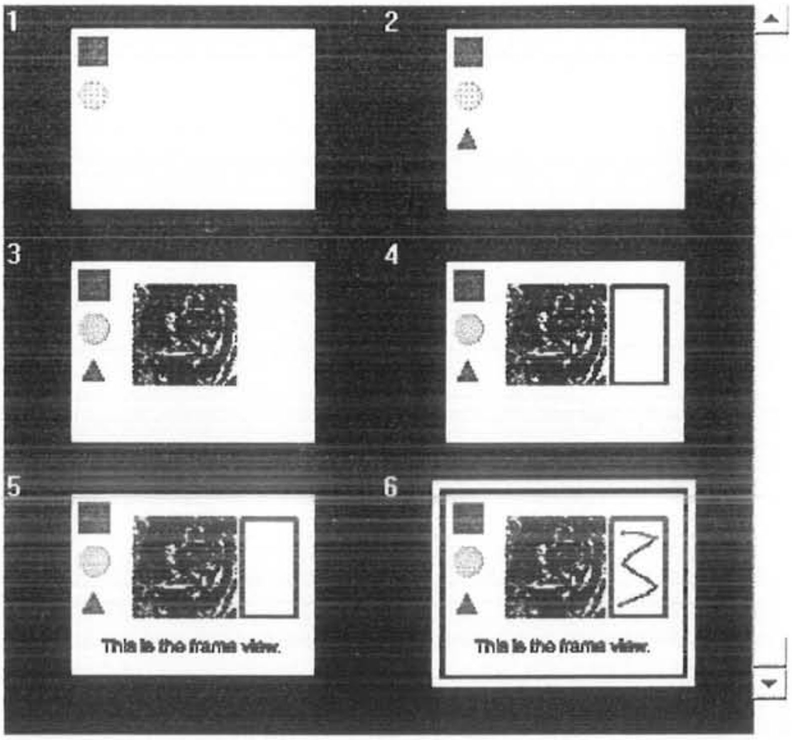
Controls

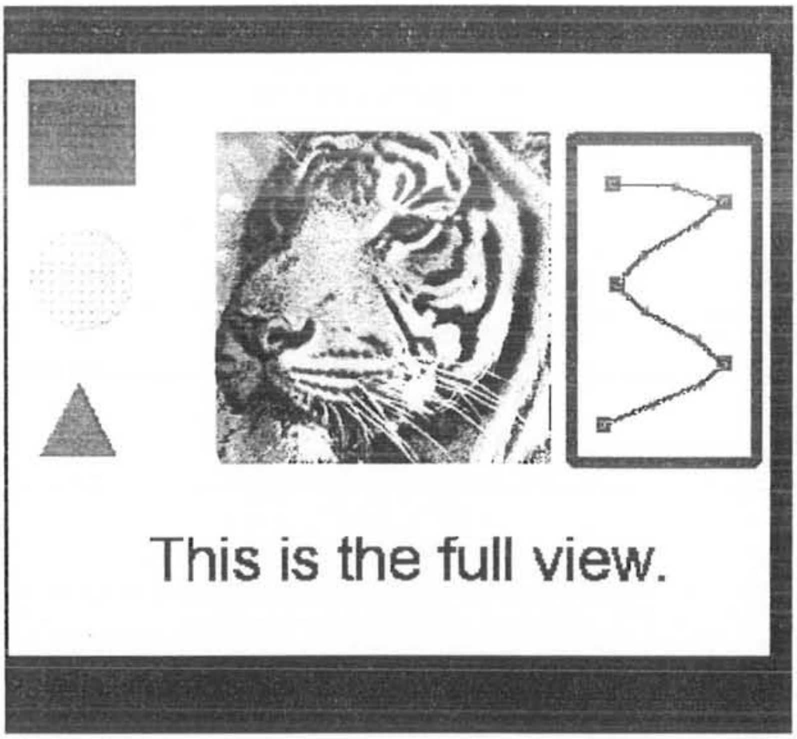
Clip Window Lets you see through Windows (if any have been created). If you click this off, you'll see everything. If clip window is on and a window is selected, element in the visible area will be represented by a crosshatched pattern.

Select all elements for viewing simultaneously. If you click this button off; you will only see elements drawn before the one you've selected all elements to be drawn later are shown as dotted outlines.

Scales the work area to let you create large presentations on small monitors.

Frame view lets you view, create, insert and adjust elements as a series of frames, with each element appearing in its own frame in the order it will be drawn.





This is the full view.

Stop Preview.

Run Preview plays the current sequence.

sequenc1 ▾

Backup file name for sequence.

Edit sequence lets you re-order elements, as well as select, modify, cut, copy and paste.

Generate Code turns the sequence into Xpower commands. Click this icon and the code will be placed on the clipboard. You can paste it into the editor at the desired location.

Preview current element by re-drawing it..

3 Circle 0 ▾

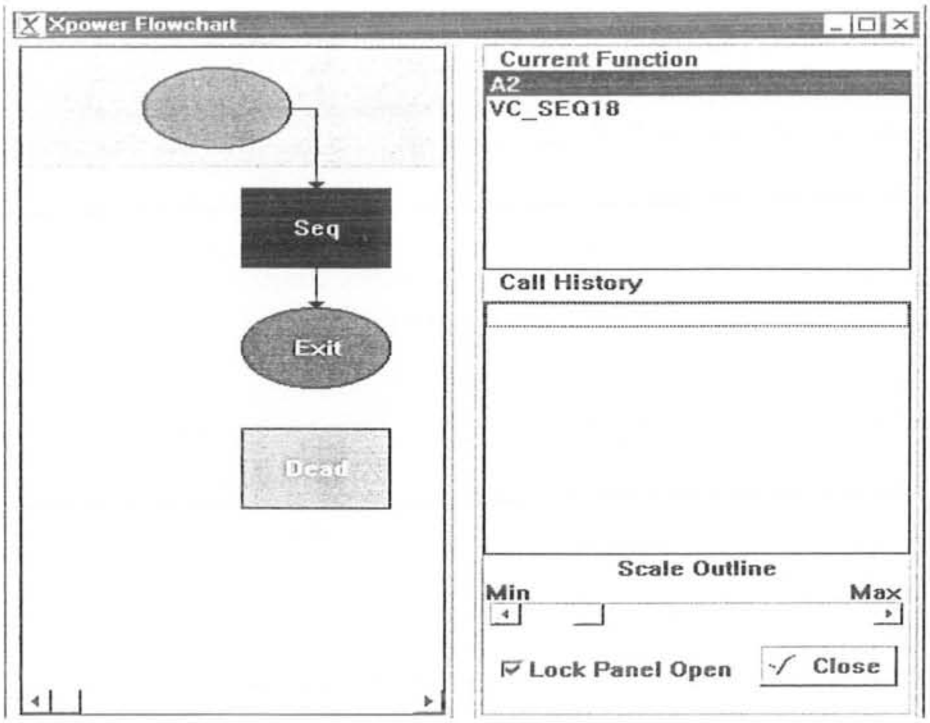
Select element to edit or move.

Edit current element shows the elements properties dialog.

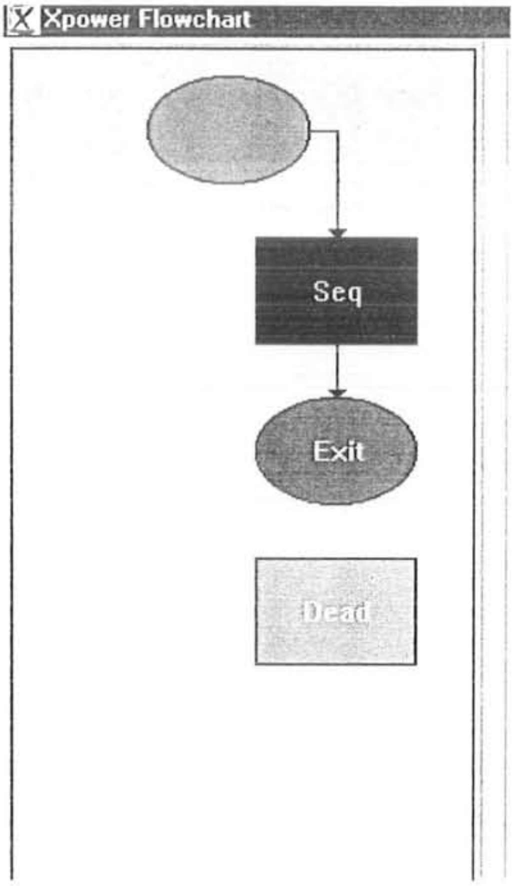
Flowchart

The Flowchart takes you a step beyond other tools. Xpower automatically diagrams the logical flow of your application. You can see what's going to happen, where you have dead code that will never execute, and how complicated or simple you are making things. The chart is linked to the source code so that clicking on a chart symbol hi-lights the associated code; and when you run the program the flowchart follows execution, with flowchart sequences showing the actual screen created by the code that underlies the logic.

FLOW CHART (Tools)



The Flowchart takes you a step beyond other tools. Xpower automatically diagrams the logical flow of your application. You can see what's going to happen, where you have dead code that will never execute, and how complicated or simple you are making things. The chart is linked to the source code so that clicking on a chart symbol hi-lights the associated code; and when you run the program the flowchart follows execution, with flowchart sequences showing the actual screen created by the code that underlies the logic.

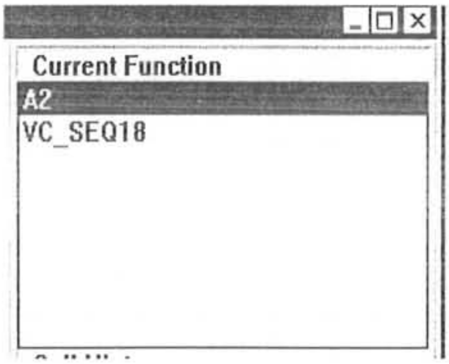


Logic Flow Diagram

Displays a flowchart diagram of your program. It depicts the logical sequence the program will follow as it executes.

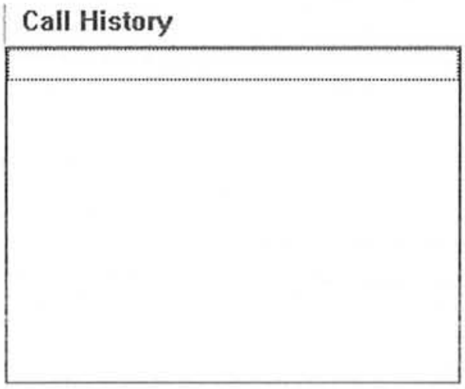
Comments to the right of the symbols reflect comments in the source code preceded by two semi-colons instead of one.

Step through the program with the debugger and you will see the symbols hi-light as they are entered and exited. The Sequence symbols will be replaced by a scaled image of the application screen actually generated by the underlying code.



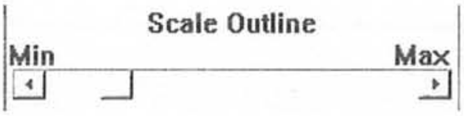
Current Function

Displays a list of user-define functions for the current project and hi-lights the current function in the flowchart. You can go to a section of the flowchart, or source code by double clicking on a function name in this list.



Call History

Displays the sequence in which user-defined functions have been called. They are listed in the order they were called. The first item listed is the original call and the last item is the function you are in now. As your code returns the list will grow shorter.

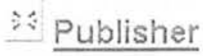


Scale

Lets you re-size the diagram to better fit the window.

Lock Panel
Forces the side panel to remain open. This prevents the annoying appearance/disappearance of the function and history windows...but it also takes up valuable screen space on small monitors.

The Debugger lets you see into the heart of your program as it runs. It is integrated into the editor so that the Edit/Debug process is seamless. You can step, pause, stop, run, walk and set breakpoints. You can watch variables change in the Watch window or you can double click on a variable in the editor and see its current value. Reset the variable and continue, or Edit in the correction and try again. Its all accomplished within a single window.



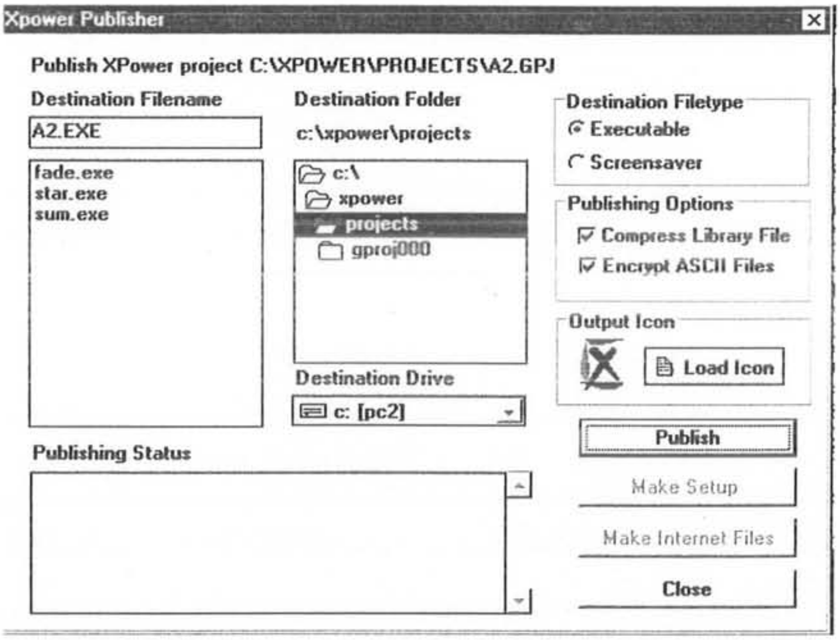
The Publisher lets you take the results of your good efforts, set your preferences, then compress and encrypt your project in one step. You get an executable, a library file, and a copy of the runtime DLL. These three files are what you distribute.

Samples for both simple and complex Xpower applications are included, with more available on web site www.pmace.com. The poet, T.S. Eliot once said Young artists borrow; the mature artist steals. Find a sample that resembles what you would like to do, load it, tinker with it, and once you understand how it works, put your name in the copyright line and turn it into your own work. That is how all great developers learn to make things. If you don't find an example of what you want to do, let us know and well offer suggestions, or make one.

A stand-alone install/uninstall program for windows is included. Called Make setup, it is not part of the IDE. While it is adequate for virtually all needs, if you are serious about distributing your work you should buy and use a commercial installation program such as Wise Install. Windows installation is a black art unto itself and these commercial installers possess magical powers. Make setup does not.

If you are looking for a windows Paint program to manipulate images, we highly recommend Paint Shop Pro from JASC. It rivals Photoshop in many features and is much more suited to creating original content. You can download a full trial version from their web site www.jasc.com. Even the registered version is under $70 and well worth it.

Publisher (Tools)



Xpower Publisher

Publish XPower project C:\XPOWER\PROJECTS\A2.GPJ

Destination Filename: A2.EXE

fade.exe
star.exe
sum.exe

Destination Folder: c:\xpower\projects

c:\
xpower
projects
gproj000

Destination Drive: c: [pc2]

Destination Filetype
● Executable
○ Screensaver

Publishing Options
☑ Compress Library File
☑ Encrypt ASCII Files

Output Icon

Publishing Status

Publisher lets you take the results of your good efforts, set your preferences, then compress and encrypt your project in one step. You get an executable, a library file, and a copy of the runtime DLL. These three files are what you distribute.

IMPORTANT NOTES:

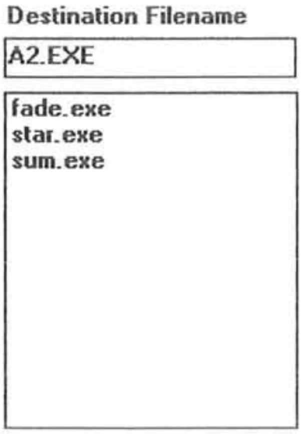
Files placed in an EL1 file must be loaded with the LOAD command before they can be used.

The Publisher does not alter the source material when encrypting and compressing. Only the resources stored inside the EL1 are actually changed.

Project File

Publish XPower project C:\XPOWER\PROJECTS\A2.GPJ

Displays the name of the project you will publish. This consists of all the resources shown in your Project Manager folders.

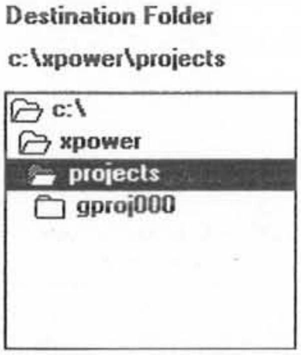


Destination Filename

A2.EXE

fade.exe
star.exe
sum.exe

Destination Filename shows the name the project will be published under. This will have either an EXE or an SCR extension, depending upon your choice of Destination File type. In either case, a corresponding library file of the same name, whose extension is EL1, will also be created. The default is the same name as your project, but you can alter it here if you wish.

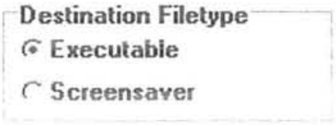


Destination Folder

c:\xpower\projects

Destination Folder

Displays the directory and drive where the distributable files (EXE, or SCR, EL1 and DLL) will be created. You can alter the target directory and drive here by selecting a new one.

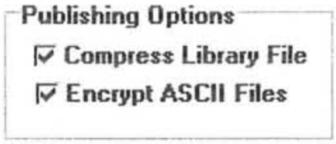


Destination Filetype

● Executable

○ Screensaver

Destination File type
Selects whether the project will build a Windows application (EXE) or a Windows Screensaver
(SCR). Selecting one or the other changes the extension of the Project File.



Publishing Options

☑ Compress Library File

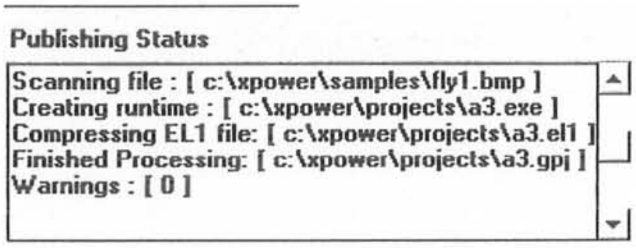
☑ Encrypt ASCII Files

Publishing Options

Compress Library File uses Microsoft Compress to squeeze the final EL1 file to minimum size. This can result in 50% to 75% reduction in size of the EL1. Xpower plays directly from compressed EL1s as well as from uncompressed EL1s.

Encrypt ASCII Files encrypts TXT and G source files so they cannot be read by someone adept at cracking open the EL1 file. Note that other resources are not encrypted.

Publish
Causes the distributable (EXE or SCR, EL1 and DLL) to be created according to your selected options.



Publishing Status

```
Scanning file : [ c:\xpower\samples\fly1.bmp ]
Creating runtime : [ c:\xpower\projects\a3.exe ]
Compressing EL1 file: [ c:\xpower\projects\a3.el1 ]
Finished Processing: [ c:\xpower\projects\a3.gpj ]
Warnings : [ 0 ]
```

Messages
Displays the results of the publishing process and warn you if there were errors in creating the necessary files.

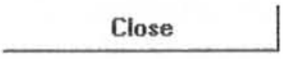
Make Setup
Creates a Windows install/uninstall that contains all your published files in a single self-installing executable. While this is adequate for most purposes, you might consider a commercial install program such as WISE Install, which offers much greater flexibility and better compression.

Make Internet Files
Extracts resources from an EL1 file as individual compressed and encrypted files for use in an Internet-based Xpower application.

NET presentations cannot run from an EL1; they need individual files.

For the presentation to be effective these files need to be compressed (and the code and text files encrypted). You can place these files in a separate target directory. For safety, it will not be allowed to inadvertently overwrite uncompressed or un-encrypted resources.

Note: The Project Manager can decompress these extracted resources, in case you lose the originals.



Close

Exit the publisher.

Getting Started Quickly

To use the IDE you must create a project, even if your project only consists of a single source file and no bitmaps or other resources however its an easy process. Your final program will bear the same name as your project.

Touch the toolbar icon to see the major menu items. Many of these choices are available elsewhere from icons and menu selections, but you can always begin here. You'll want to open the Project Manager and the Source Code Editor if they are not already open. Size and position them to suit your preference the size and position and which windows should be open are saved with the project.

If you wonder what some icon represents, point to it and pause a hint will pop up to tell you what you are pointing at.

If you have an existing .G project file from previous versions of Xpower or Expo, simply open it with the toolbar Project|Open. Resources will be automatically scanned into the project manager folders where you can examine them. Touch the right side of the Project Manager with the mouse cursor and a panel opens up to let you gather more resources or remove files you don't want. You can, if you want, check a box right there to keep this panel open.

If you have an existing G file, open the Source Code Editor. From the toolbar menu, select File|Open and pick the existing G file you want to work with. Answer the dialogs and the G Project file will be created and youll be ready to gather resources, or go to work. If your Editor detected any syntax errors, they'll be noted when the editor opens your files. If your project has multiple source files and they were not automatically scanned into, you may see errors because of unresolved references. These will go away when you add the other source files with the Project Manager Resource Gatherer.

Resources mentioned by name will be automatically scanned into the project manager folders where you can examine them. Touch the right side of the Project Manager with the mouse cursor and a panel opens up to let you gather more resources or remove files you don't want. You can, if you want, check a box right there to keep this panel open.

If you are starting from scratch choose the toolbar Project|New and answer the dialogs. Remember, your final application will bear the name of the projects main G file. And when the projects created, the default G file will get the name you assign the project. So choose what you name things with some thoughtfulness. Now go on to the next paragraph.

No matter how you get started, if your project consists of multiple files, source files, bitmaps, sounds, animations, etc., that were not automatically scanned in, touch the right side of the Project Manager with the mouse cursor and a panel opens up to let you gather more resources or remove files you don't want. You can, if you want, check a box right there to keep this panel open.

The Project Manager window contains folders for the source code files, bitmaps, sounds, video, animation, cursors and text files that go into a project. You can actually see and preview the things that go into your project. You preview and gather content into the Project Manager with the Content Gatherer, which you can get to from the toolbar menu or by touching the right edge of the Project Manager.

The Content Gatherer lets you browse your system in search of the stuff of which Xpower applications are made of: images, FLCs and AVIs, sound files, text files, G source files, cursors. You can preview them. When you find something you want, click on it to select it, and then click the right finger button when you're ready to add it to the list of things to be put into the project manager. You can leave the files in their current directory but we don't advise it unless you are sophisticated in your ability to manage resources. Its wiser to select the Copy to Project Directory option before clicking the Add button.

Xpower must convert your images to .BMP format to make use of them. It does that right here, when you say Add. This is all transparent to you, but you should be aware of what is happening: your files are being copied into the project directory and any format conversions will take place. If you chose not to copy, The Resource Gatherer will make a BMP with the same name as the original file. When you Publish, you have the option to

Screensaver

Any Xpower application can be used as a screen saver.
A screensaver is just a Windows program, with some special internal code that lets Windows launch it when no activity has taken place for a user-defined amount of time. After that, it can do anything any program can do, from the ridiculously simple blanking of the screen, to the sublimely complex updating of database records via the Internet. It can be pure entertainment, boringly practical, or both. It does not have to be full-screen and it does not have to quit when the mouse moves. That behavior is up to the author to implement or not.

Most screensavers, however, operate according to the following logic:

Use the full screen, turn off the cursor
…And do whatever it is you do, repeatedly
…unless the mouse moves or a key is pressed
…in which event, restore the cursor and de-activate yourself!

The Xpower code that embodies this logic is:

```
Cursor off                ; turn off the cursor
escape off                : we check for this key ourselves

lmouse_x=mouse_x               ; variable holds starting x mouse coordinate
lmouse_y=mouse_y               ; variable holds starting y mouse coordinate


do          ; top of the repeating loop

 ;YOUR PROGRAM HERE
 text this is a screensaver;the Visual Composer can do this and more…


 ;THE REST IS COMMON TO ALL SCREENSAVERS

 if mouse_x==lmouse_x ; if the current mouse_x where it started…

  if mouse_y==lmouse_y        ; and if the current mouse_y is the same...
   if keypress==0         ; and if no buttons or keys were pressed…
```

```
        again                [illegible]
      endif
    endif
  endif

cursor on
exit                 [illegible]

repeat               ; bottom of the loop
```

This is a complete Windows screensaver. You can copy this code into the Xpower editor, run it and Publish it.

Select the Screen Saver option in the Publisher and it will produce an .SCR file and an .EL1 file. Place these two files, plus a copy of Xpow16.dll in the Windows directory. Your application will now be listed in the Display|Screen Saver section of Win 95. Or in Windows 3.1 it will show in the Desktop|Screen Saver section the next time you re-start Windows.

Windows will launch your Xpower Screen Saver the way it does any screen saver.

Modify the example using the VisualComposer to replace line 12:

text this is a screensaver ;the Visual Composer can do this and more...

To understand better what the sample above is doing, you need to know that Xpower always makes certain information available:

video_x and video_y always tell the width and height of the screen in pixels.

mouse_x and mouse_y always contain the current x and y locations of the mouse cursor.

keypress is always 0 if no key or mouse button was pressed since last checked.

Xpower system variables: scr_count and scr_delay are also available. The Windows Screen Saver Settings dialog lets the user change these values within a range of 0 to 100. Windows remembers between runs of the program and Xpower passes them to you to use, or not use, in your screensaver.

The current version of Xpower does not provide for screensaver Passwords.

An Xpower Screen Saver always begins by saving the existing screen to disk as XPOWSCR.BMPand erasing it when it exits. Many screensavers use this image of the screen as the starting point for the effect they want to create for example, slowly erasing the screen with a flying square. You could achieve this in Xpower by first displaying EXPOSCR.BMP, so it appears nothing has changed, then using the FLY command, for instance, with the leave a trail option turned on, to slowly erase the image.

Note that when an Xpower application executes the EXIT command, the screen saver will de-activate.

Not also that some operations, such as a FADE or FLY must be completed before the screensaver will de-activate.

More Complex Screensavers are not that difficult to create. If you load and examine the../samples/scr/logosavr.gpj you will see that it is the same basic program shown above! It has additional code to load a bitmap image, to generate random colors, and to make a random fly path for the logo to fly around.

Try making small changes in the example to get a feel for what you can accomplish.

Try substituting a bitmap of your own for the Xpower logo.

Try using the PLAYWAVE command to add sound effects just before and after the FLY.

Try changing the FLY command so it does not leave a trail.

Try making the trail switch on and of at random by using the RANDOM() function to randomly generate a 1 or a 0 for the FLY commands leave a trail argument.

Advanced ScreenSavers make use of Xpowers full feature set and Internet capabilities

The../samples/tiger/tiger.gpj example gpj is a more complex, interactive screensaver based on a puzzle game.

The../samples/netcast/netcast.gpj example is quite sophisticated both in its user interface and, with the NET command un-commented, live updating of information via the Internet.

It is worth re-emphasizing: A screensaver is just a Windows program. Special internal code lets Windows launch it when no activity has taken place for some time. Afterwards, runs like any other application. It does not have to be full-screen and it does not have to quit when the mouse moves. What your screensaver is and how it behaves is up to you.

CHAPTER#4

APPLICATION

Any Xpower application can be used as a screen saver. A screensaver is just a Windows program, with some special internal code that lets Windows launch it when no activity has taken place for a user-defined amount of time

After having introduced with the Syntax, commands & functions, I had to design some application in G-language, so that the working of all these things may become clearer.

Taking an advantage of the facility provided by X-power & with the help & guidance of my much respected teacher, I decided to make some Screensavers.

APPLICATION CODES

We being Muslims begin all our activities with the name of **ALLAH**, so my application also starts with the name of THE GREAT CREATOR. When you click "next" the index will appear. You have to enter any digit from 1-9.

When you enter "1" an introduction about the project will be displayed. When number "2" is entered you will look into the utility of X-power about text writing.

When choice "3 "is entered bitmaps in X-power will be displayed.

Choice number "4" is about Call of function from Dos- mode.

Number "5 & 6" display how you can input any data in X-power.

When you press "7" you will see FLC animation.

Number "8" will show you an AVI.

Last but not least when you press the number "9" you will get some samples of Screensavers made in X-power.

```
;Main Project File, F3.gpj
clearscr 0,255,255
load  "c:\xpower\samples\a2.bmp"
 fade "c:\xpower\samples\a2.bmp",25,10,60,0
color  128,128,255
box 540,420,590,450,3
font "times new roman",14,0,1
color  64,0,127
text "NEXT>",550,430
waitkey
top:
clearscr 0,0,128
load "c:\xpower\samples\k1.bmp"
show "c:\xpower\samples\k1.bmp" ,5,10
waitkey
if keychar == "1"
f11()
goto top
endif
if  keychar=="2"
f22()
goto top
endif
if  keychar=="3"
 f33()
goto top
endif
if keychar=="4"
f44()
goto top
endif
if keychar=="5"
f55()
goto top
endif
```

```
if keychar=="6"
f66()
goto top
endif
if keychar=="7"
f77()
goto top
endif
if keychar=="8"
f88()
goto top
endif
if keychar=="9"
f99()
goto top
endif
if keychar=="10"
goto end1
endif
;goto top

end1:
exit
f11():
clearscr 0,255,255
color 0,128,255
circle 320,215,280,90,1,1
;                                                                    FONT
name,[point],[angle],[bold_flag],[italic_flag],[uline_flag],[strikethru_fla

color 0,255,255
font "times new roman",16,0,1,0,0,0,1,3,22
text "PROJECT ABOUT THE",75,180,10
text "STUDY OF X-POWER",90,230,10
wait 1000
clearscr 255,255,255
color 0,0,255
show "c:\xpower\samples\pr.bmp",20,50
box 500,390,550,420,3
font "times new roman",14,0,1
```

```
text "NEXT",510,400
waitkey
top:
clearscr 0,255,255
color 0,0,0
load"c:\xpower\samples\dart1.bmp"
font "IMPACT",30,0,1,,1
text "FEATURES OF X-POWER.",50,10,,100
show "c:\xpower\samples\dart1.bmp",50,60
wait 50
font "arial",24,0,1
text Left
text "Text  can be written in many ways.",100,60,,100
show "c:\xpower\samples\dart1.bmp",50,95
wait 200
text " X-power provides facility to play Audio-Visuals.",100,100,,100
show "c:\xpower\samples\dart1.bmp",50,135
wait 200
text "It facilitates playing Wave files.",100,140,,100
show "c:\xpower\samples\dart1.bmp",50,175
wait 200
text "We can make our own screensavers in X-power by fade command.",100,180,,100
show "c:\xpower\samples\dart1.bmp",50,225
wait 200
text "X-power provides facility in animating bitmaps & text.",100,230,,100
box 500,400,550,430,3
font "times new roman",14,0,1
text "NEXT",510,410
waitkey
clearscr 255,255,255
color 255,0,0
font "impact",28,0,1,0,1
text "CAUTION!",100,5
font "impact",24,0,1
color 0,0,255
text " While working with X-Power you come across with:",50,60
font  "Times roman",19,0,1
color 255,0,0
```

```
text LEFT
text " * Floating point invalid  error.",50,100
text " * Generation protection error." ,50,150
text " * Unrecovered parsing error." ,50,230
text " * Array not defined or index out of bounds. ",50,290
color 0,128,0
text " -> Index is lesser than declared  in array. ",120,120
text " -> There is a logical mistake or you have  ",120,180
text "      interrupted  the under process program.",120,200
text " ->  Typing mistake in the command.",120,260
text " -> Array should be defined before using in the command.",120,320
color 0,0,255
text " Continued...." ,500,380
box 500,400,550,430,3
font "times new roman",14,0,1
text "NEXT>>",510,410
color 0,0,0
text "(1)" ,340,430
waitkey
clearscr 255,255,255
color 0,0,255
font "times new roman",19,0,1
text "....From page 1" ,50,50
color 255
text " * Un-expected symbol in simple expression.",50,100
text " * Un-expected symbol in statement." ,50,160
text " * Too few arguments.",50,220
text " *  Comma missing between arguments.",50,280
color 0,128,0
text "  ->  Un-expected symbol has been  typed in the
command.",120,130
text "  -> Unexpected  symbol has been typed  in the command.",120,190
text "  -> Syntax of the command is not followed.",120,250
text "  -> The space  between the arguments is greater than",120,310
text "      required,for example instead of writing bmp[0..0], ",90,330
 text "      you have entered bmp[0. .0].",100,360
color 0,0,0
text "(2)" ,340,430
circle 510,410,25,10,3,1
font "times new roman",14,0,1
```

```
color 255,255,255
text "NEXT>>",496,405
waitkey
clearscr 0,255,255
message "INFORMATION","You can have more details: ","at www.pmace.com",,0

if msg_reply==1
       clearscr 0,255,255

endif
wait 100
return
  f22():
clearscr 255,255,255
color 0,0,0
font "Arial",50,10,1,1,1

text "X-power facilitates to write text in different sizes & in different directions",100,150
color  128,128,255
box 540,420,590,450,3
font "times new roman",14,0,1
color  64,0,127
text "NEXT>",550,430
waitkey
clearscr 255,255,255
color 128,0,255
font arial ,30,0,1,1,1
text " TEXT IN DIFFERENT FONTS & SIZES",90,10
color 255,0,255
font arial ,22,0,1,1
text "ARIAL",50,60
font arial ,24
color 0,128,255
text "ALLAH  loveth not those who make mischief.",10,120
color 0,255,0
font arial ,22,0,1
text "(AL-QURAN)",480,150
wait 200
```

```
font arial ,24
color 0,128,255
text "GOD!  There is no good ,but He -The Living,Self
subsisting,Eternal.",10,190
color 0,255,0
font arial ,22,0,1
text "(AL-QURAN)",480,220
wait 200
font arial ,24
color 0,128,255
text "Enjoin the desirable and forbid the undesirable.",10,250
color 0,255,0
font arial ,22,0,1
text "(AL-QURAN)",480,290
wait 200
font arial ,24
color 0,128,255
text "And whatever ye spend in the way of ALLAH , will be repaid to
you ",10,330
text "and ye will not be wronged.",10,360
color 0,255,0
font arial ,22,0,1
text "(AL-QURAN)",480,360
wait 200
color 0,0,250
font arial ,16,0,1
box 500,410,555,435,3
text "NEXT>",510,415
waitkey
clearscr 255,255,255
color 255,0,255
font courier ,28,0,1,1
text "courier",60,10
color 0,128,255
font courier ,26,0,1
text "Actions shall be judged only by " ,10,70
 text  " intention and a man shall have, ",10,100
text "   what he intends",10,130
color 0,255,0
font courier ,22,0,1
```

```
text "(AL-HADITH)",480,180
wait 200
font courier ,26,0,1
color 0,128,255
text "A word of wisdom is the lost property ,",10,240

text "  of the believer so wherever he finds ",10,270
text "  it he has a better right to it." ,10,300
color 0,255,0
font courier ,22,0,1
text "(AL-HADITH)",480,360
color 0,0,250
font corier ,16,0,1
box 500,410,555,435,3
text "NEXT>",510,415
waitkey
clearscr 255,255,255
color 255,0,255
font "Monotype corsiva" ,24,0,1,0
text " Monotype corsiva ",60,10
font "Monotype corsiva ",28,0,1,1,1
color 0,0,255
text "Words of wisdom:",100,50
color 0,128,255
font " Monotype corsiva ",24
text "The joys of parents are secret , and so are their griefs and
fears.",10,120
wait 200
font "Monotype corsiva ",24
color 0,128,255
text "The real essence of work is cocentrated attention.",10,160
wait 200
font "Monotype corsiva ",24
color 0,128,255
text "A man is valued as he makes himself valuable.",10,200
wait 200
font "Monotype corsiva ",24
color 0,128,255
text "Method saves hours of wasted efforts.",10,240
wait 200
```

```
color 0,0,250
font " Monotype corsiva",16,0,1
box 500,410,555,435,3
text "NEXT>",510,415
waitkey
     clearscr 255,255,255
       color 255,0,255
      font impact ,24,0,1,0
text " Impact ",60,10
font "impact",28,0,0,1
color 0,0,255
   text "....Continued.",100,50
color 0,128,255
font " impact",24
      text "Today will be yesterday tomorrow  & we will not have it
again.",10,120
wait 200
     text "A friend is someone who knows everything about you  but
still loves you.",10,160
wait 200
        text "Many able man have failed through lack of
tacts.",10,200
wait 200
    text "Suffering is not a punishment , it is a result." ,10,240
wait 200
       color 0,0,250
        font "impact" ,16,0,1
        box 500,410,555,435,3
        text "QUIT",510,415
      waitkey
clearscr  255,255,255
color 0,0,0
      font  Impact,28,0,1
text "You can write scrollable text" ,50,10
   text " in a window ",50,40
color 0,0,0
font  Impact,18,0,1
load "c:\xpower\samples\l.txt"

  h=win_editbox(200,100,300,150, "c:\xpower\samples\l.txt",1,0,0,3,3)
```

```
    win_show(h)
    waitkey
      box 500,400,550,430,3
           font "times new roman",14,0,1
              text "NEXT>>",510,410
win_destroy(h)
clearscr 0,255,255
 font arial,28,0,1
color 0,0,255
         text  "Text can be animated also.",100,80
wait 500


color 0,0,255
x=100
y=200
           font "Ms Sanserif",30,0,1
text " L",105,y
      wait 250
text " O",120,y
       wait 250
text " N",140,y
       wait 250
text " G",160,y
       wait 250
text " L",205,y
       wait 250
text " I",220,y
       wait 250
text " V",230,y
       wait 250
text " E",245,y
       wait 250
font "TIMES ROMAN",70,0,1
text " P ",110,300
          wait 250
text " A",140,300
wait 250
text " K ",180,300
wait 250
```

```
text " I ",220,300
wait 250
text " S ",240,300
wait 250
text " T ",270,300
wait 250
text " A",300,300
wait 250
text " N ",340,300
color 0,0,250
        font " Monotype corsiva",16,0,1
            box 500,410,555,435,3
                text "QUIT",510,415
                   waitkey
return
f33():
                clearscr 0,255,255
             font "Times roman",28,1
          color 0,0,255
              text "Bitmaps can be animated in a straight line.",30,400
   load "c:\xpower\samples\a1.bmp"
; PAN fname,starting_x, starting_y, direction,#_pixels,[push]
       pan "c:\xpower\samples\a1.bmp",5,20,3,650,1

               wait 1000
clearscr 255,255,255
color 112,0,255
font "lucida handwriting",36
    text "You can animate  bitmaps along ",10,40
text "a curved path",10,80
b1[ ]="c:\xpower\samples\ball.bmp"
p1[0 ]=10
 p1[1 ]=400
p1[2 ]=150
p1[3 ]=120
p2[0 ]=150
p2[1 ]=120
p2[2]=350
p2[3 ]=200
p3[0 ]=350
```

```
p3[1 ]=200
p3[2 ]=640
p3[3 ]=420
load  b1[0]
wait 100
; FLY bmp_array[ ], path_array[ ],[steps],[float_flag],[curved_flag]
fly   b1[0..0],p1[0..3],350,0,1
fly   b1[0..0],p2[0..3],350,0,1
fly   b1[0..0],p3[0..3],350,,1
wait 1000

clearscr 0,255,0

load "c:\xpower\samples\h1.bmp"
show  "c:\xpower\samples\h1.bmp",0,0
 font "Times roman",30,1
          color 0,0,255
             text "Multiple bitmaps can be animated.",30,10
 d[]=  "c:\xpower\samples\gly4.bmp"
d1[]=  "c:\xpower\samples\fly2.bmp"
d2[]=  "c:\xpower\samples\fly3.bmp"
d3[]=  "c:\xpower\samples\fly4.bmp"
load d[0],1
load d1[0],1
load d2[0],1
load d3[0],1
wait 200
path[0]=40
path[1]=380
path[2]=0
path[3]=325
path1[0]=0
path1[1]=320
path1[2]=150
path1[3]=330
path2[0]=150
path2[1]=330
path2[2]=440
path2[3]=330
path3[0]= 440
```

```
path3[1]=330
path3[2]=640
path3[3]=320
fly d[0..0],path[0..3],400,0,0
fly d1[0..0],path1[0..3],420,0,1
fly d2[0..0],path2[0..3],450,0,1
fly d3[0..0],path3[0..3],320,0,1
color 255,128,0
    font " arial",16,0,1
        box 550,415,600,445,3
            text "QUIT",560,420
waitkey
return
f44():
     clearscr 255,255,255
color 0,0,255
font "garamond",38
text "You can execute functions from other operating system" ,10,100
wait 1000
clearscr
load "c:\tc\bin\l5.exe"
s=exec("c:\tc\bin\l5.exe",,1)
        text s,10,400

return
f55():
clearscr
clearscr 0,255,255
color 0,0,0,255,255,255
font "IMPACT",28,0,1
text "NUM-GAME",200,10
font "impact",24
text "Enter any five-digit numbers",50,40
font "arial",22
input first,140,100,,,15
input second,140,150,,,15
input third,140,200,,,15
font "courier",18
button ok,280,400,50,50,"ok"
text "FIRST NO:",0,110
```

```
input first,on
text "SECOND NO:",0,160
input second,on
text "THIRD NO:",0,210
input third,on
button ok,on
top:
input first,focus
        waitkey ,200
   x=first-2
font "monotype cosiva",30,0,1,1
text "The answer will be  2"+x  ,50,260
input second,focus
        waitkey ,200
x2=99999-second
font "monotype cosiva",20
text " first",390,280
text first,520,280
text "second ",390,310
text second ,520,310
text "computer's entry",390,340
text  x2  ,520,340
button ok,up
input third,focus
        waitkey ,200
        if ok==0, top
        button ok,up
   x3=99999-third
text "computer's entry",390,400
text x3  ,520,400
text " third",390,370
text  third, 520,370
        x4=first+second+x2+third+x3
font "monotype cosiva",24
color  255,0,0
text " ANSWER",390,430
text  x4,520,430
waitkey
        button ok,up
input first ,free
input second ,free
input third ,free
return
f66():
clearscr
clearscr 255,255,255
        color 128,0,255,255,255,255
font "IMPACT",30,0,1
       text "LET ME GUESS YOUR AGE",200,10
font "Arial",24,0,1
text "Pick a code of the day. ",50,40
font "arial",16,0,1
text "MONDAY        (1)",80,90
           text "TUESDAY        (2)",80,110
 text "WEDNESDAY  (3)",80,130
            text "THURSDAY    (4)",240,90
 text "FRIDAY          (5)",240,110
          text "SATURDAY    (6)",240,130
 text "SUNDAY          (0)",80,150
        font "IMPACT",20,0,1
text "Choice",20,200
 font "arial",20,0,1
 text "If  you have celebrated your birthday  then  enter 1753"   ,80,200
            text " else enter 1752", 50,230
 input day,140,335,,,5
           input choice,240,335,,,5
 input year,340,335,,,5
          font "courier",18
 button ok1,280,400,50,50,"ok"
      text "DAY:",140,310
 input day,on
      text "CHOICE:",240,310
 input choice,on
 text "YEAR:",340,310
       input year,on
 button ok1,on
 top:
 font "MS Outlook",20
 input day,focus
       waitkey ,200
```

```
    x=2*day
          x1=x+5
 x2=x1*50
 font "MS Outlook",20
        input choice,focus
 waitkey ,200
          x3=x2+choice
 font "arial",22,0,1
 wait 10
            text "Enter your year of birth"   ,50,280
 font "MS Outlook",20
  input year,focus
          waitkey ,200
  if ok1==0, top
          button ok1,up
  x4=x3-year
            d=day*100
  x5=x4-d
if day=="0"
day="Sunday"
endif
if day=="1"
day="Monday"
endif
if day=="2"
day="Tuesday"
endif
if day=="3"
day="Wednesday"
endif
if day=="4"
day="Thursday"
endif

if day=="5"
day="Friday"
endif
if day=="6"
day="Saturday"
endif
```

```
text"ANSWER",10,390
text "you entered", 10,410
text  " & you are of",10,430
color 255,0,255
 text day,130,410
          text  x5  ,130,430
button ok1,up
waitkey
button ok1,free
return
f77():
clearscr 255,0,0

color 0,0,255
font "monotype corsiva",40
color 0,0,255
text "Xpower allows FLCs animation.",100,10
wait 500
load "c:\xpower\samples\man.flc",1
        load "c:\xpower\samples\man1.flc",1
load "c:\xpower\samples\man2.flc",1
     load "c:\xpower\samples\man3.flc",1
load "c:\xpower\samples\man4.flc",1

font "monotype corsiva",34,0,1
text "CATCH ME IF YOU CAN! ",10,60
; PLAYFLC file,x,y,start,end,[time]

        playflc "c:\xpower\samples\man.flc",100,200,,,0,-1,65
wait 500

playflc "c:\xpower\samples\man.flc",stop
    clearscr 255,0,0
color 0,0,255
text "CATCH ME IF YOU CAN! ",10,60
        playflc "c:\xpower\samples\man1.flc",200,200,,,,-1,200
wait 500
playflc "c:\xpower\samples\man1.flc",stop
      clearscr 255,0,0
color 0,0,255
```

```
        text "CATCH ME IF YOU CAN! ",10,60
playflc "c:\xpower\samples\man2.flc",300,200,,,0,-1,200
wait 500
playflc "c:\xpower\samples\man2.flc",stop
      clearscr 255,0,0
color 0,0,255
        text "CATCH ME IF YOU CAN! ",10,60
            playflc "c:\xpower\samples\man3.flc",400,200,,,0,-1,300
wait 500
    text "CATCH ME IF YOU CAN! ",10,60
 playflc "c:\xpower\samples\man3.flc",stop
clearscr 255,0,0
    clearscr 255,0,0
color 0,0,255
        text "CATCH ME IF YOU CAN! ",10,60
            playflc "c:\xpower\samples\man4.flc",500,200,,,0,-1,300
wait 500
    text "CATCH ME IF YOU CAN! ",10,60
 playflc "c:\xpower\samples\man4.flc",stop
color 0,0,0
box 500,400,570,440,3
font "lucida cosole",22,0,1
text "QUIT",510,410
clearscr 255,0,0
waitkey
return
f88():
clearscr 255,255,255
color 0,0,128
font "Bookman old style",40,,1,1
text "X-power facilitates to play AVI",10,10
load "c:\xpower\projects\clock.avi"
        playavi "c:\xpower\projects\clock.avi",100,120,250,250,0,1000,1
        waitkey
        playavi "c:\xpower\projects\clock.avi",stop
color 0,0,128
box 500,400,570,440,3
font "lucida cosole",24,0,1
text "Quit",510,410
        waitkey
```

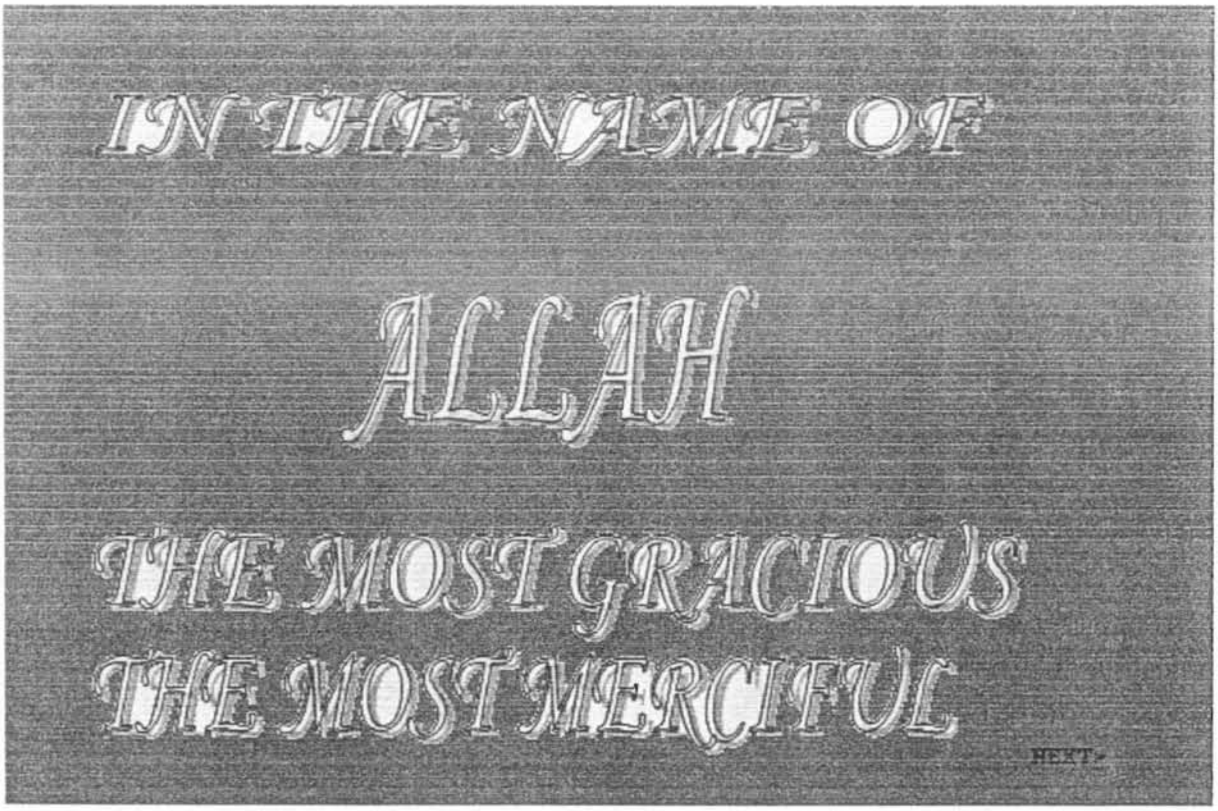
```
return
 f99( ):

clearscr 255,255,255
load "c:\xpower\samples\a3.bmp"
load "c:\xpower\samples\a5.bmp"
load "c:\xpower\samples\m11.bmp"
load "c:\xpower\samples\m12.bmp"
clearscr 255,255,255
fade  "c:\xpower\samples\a3.bmp",11,0,0,3500

        wait 500
clearscr 0,64,128

load "c:\xpower\samples\a4.bmp"
fade  "c:\xpower\samples\a4.bmp",12,0,0,3500
           wait 200
clearscr 0,255,0

show "c:\xpower\samples\m11.bmp",1,1
fade  "c:\xpower\samples\m12.bmp",37,100,150,3500
            wait 500
clearscr
fade  "c:\xpower\samples\a5.bmp",27,0,0,3500
color  0,0,255
box 540,420,590,450,3
font "times new roman",14,0,1
text "NEXT",550,430
waitkey
clearscr 0,0,0
do
x1=10
y1=10
x2=10
y2=10
dx1=2
dy1=2
dx2=2
dy2=2
x4=100
```

```
count 0,2000,1
line x1,y1,x2,y2
x4=x4+10
x1=x1+dx1
y1=y1+dy1
x2=x2+dx2
y2=y2+dy2

color x1,x4,y2
if (x1<=o)||(x1>=632)
dx1=0-dx1
endif
if (y1<=o)||(x1>=459)
dy1=0-dy1
endif
if (x2<=o)||(x2>=632)
dx2=0-dx2
endif
if (y2<=o)||(y2>=459)
dy2=0-dy2
endif
wait 20

loop
        if keypress==1
     break do
endif
repeat
color 0,0,255
box 5,400,50,430,3
font "courier",14,0,1
text "NEXT>",10,410
waitkey

clearscr 255,255,255
load "c:\xpower\samples\chick.bmp"
fade "c:\xpower\samples\chick.bmp",19,50,50,100
color 0,0,255
box 500,400,550,430,3
font "times new roman",14,0,1
```

```
text "NEXT>",510,410
waitkey
clearscr 0,0,0,0,0,0
x=0
x1=255
y1=255
z1=0
do
count 0,9,1
font "times roman",52,x,1,1

color x1,y1,z1
x1=x1-30
y1=y1+15
z1=z1+128
text "WELCOME",300,230
x=x+45
if (x==180)
x= 0-180
endif
wait  300
loop
if keypress==1
break do
endif
repeat
color 255,128,128
box 500,400,550,430,3
font "Courier",14,0,1
text "NEXT>",510,410
waitkey


clearscr 0,0,0
load "c:\xpower\samples\star.bmp"
show "c:\xpower\samples\star.bmp",250,170
fade  "c:\xpower\samples\star.bmp",14,120,170
fade  "c:\xpower\samples\star.bmp",15,385,170
fade  "c:\xpower\samples\star.bmp",15,320,260
fade  "c:\xpower\samples\star.bmp",14,175,260
fade  "c:\xpower\samples\star.bmp",13,250,350
fade  "c:\xpower\samples\star.bmp",15,320,80
fade  "c:\xpower\samples\star.bmp",14,175,80
fade  "c:\xpower\samples\star.bmp",16,250,0
color 255,255,0
box 500,400,570,440,3
font "lucida cosole",22,0,1
text "NEXT>",510,410
waitkey
do
clearscr 0,128,0
load "c:\xpower\samples\fl1.bmp"
fade  "c:\xpower\samples\fl1.bmp",12,10,10,4000
wait 200
fade  "c:\xpower\samples\fl1.bmp",21,210,150,4000
wait 200
fade  "c:\xpower\samples\fl1.bmp",12,440,10,4000
wait 200
fade  "c:\xpower\samples\fl1.bmp",12,10,290,4000
wait 200
fade  "c:\xpower\samples\fl1.bmp",12,440,290,4000
wait 100
if keypress==1
break do
endif
repeat
color 0,255,0
box 500,400,570,440,3
font "lucida cosole",22,0,1
text "NEXT>",510,410
waitkey

clearscr 0,255,255
load "c:\xpower\samples\i1.bmp"
s[]="c:\xpower\samples\s1.bmp"
clearscr 0,255,255
path[0]=1
path[1]=50
path[2]=520
path[3]=50
```

```
do
fade "c:\xpower\samples\s1.bmp",15,0,0,100
color 0,0,255
font "lucida cosole ",30,0,1,0,0,,,3,25
text "HI GUYS!", 50,20
wait 350
clearscr 0,255,255

load s[0],1
clearscr 0,255,255
; FLY bmp_array[ ], path_array[ ],[steps],[float_flag],[curved_flag]
tile "c:\xpower\samples\s1.bmp",2
fade "c:\xpower\samples\s1.bmp",15,300,34,150
fade "c:\xpower\samples\s1.bmp",15,34,32,100
fade "c:\xpower\samples\s1.bmp",15,300,225,150
fade "c:\xpower\samples\s1.bmp",15,40,225,150


if keypress==1
clearscr 0,255,255
fade "c:\xpower\samples\i1.bmp",13,10,340,200
fly s[0..0],path[0..3],320,1,0
wait 10
break do
endif
repeat
color 0,0,128
box 500,400,570,440,3
font "lucida cosole",22,0,1
text "QUIT",510,410
waitkey
return
```

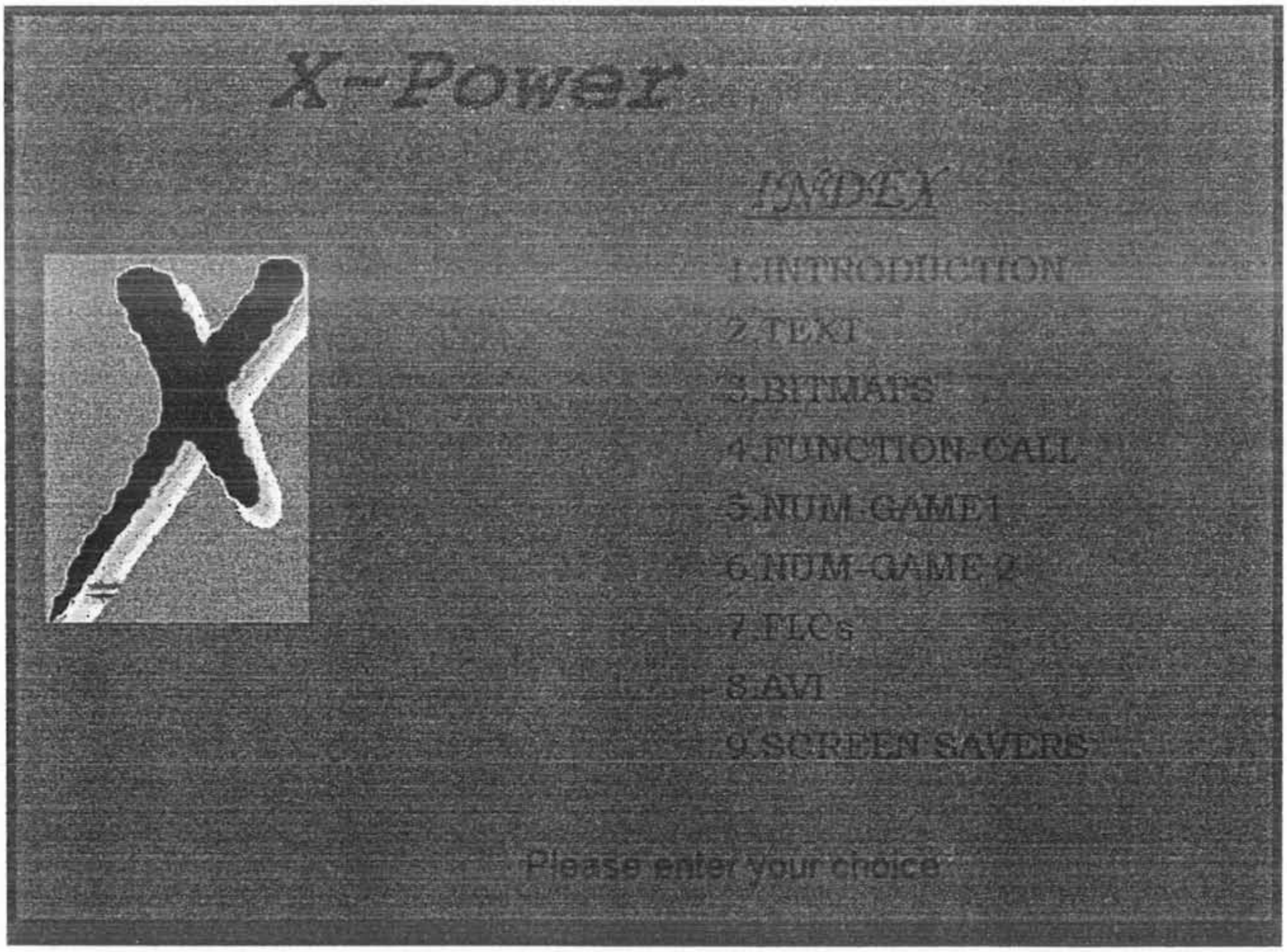
Samples
Of output



IN THE NAME OF

ALLAH

THE MOST GRACIOUS
THE MOST MERCIFUL



X-Power

INDEX

1.INTRODUCTION

2.TEXT

3.BITMAPS

4.FUNCTION-CALL

5.NUM-GAME1

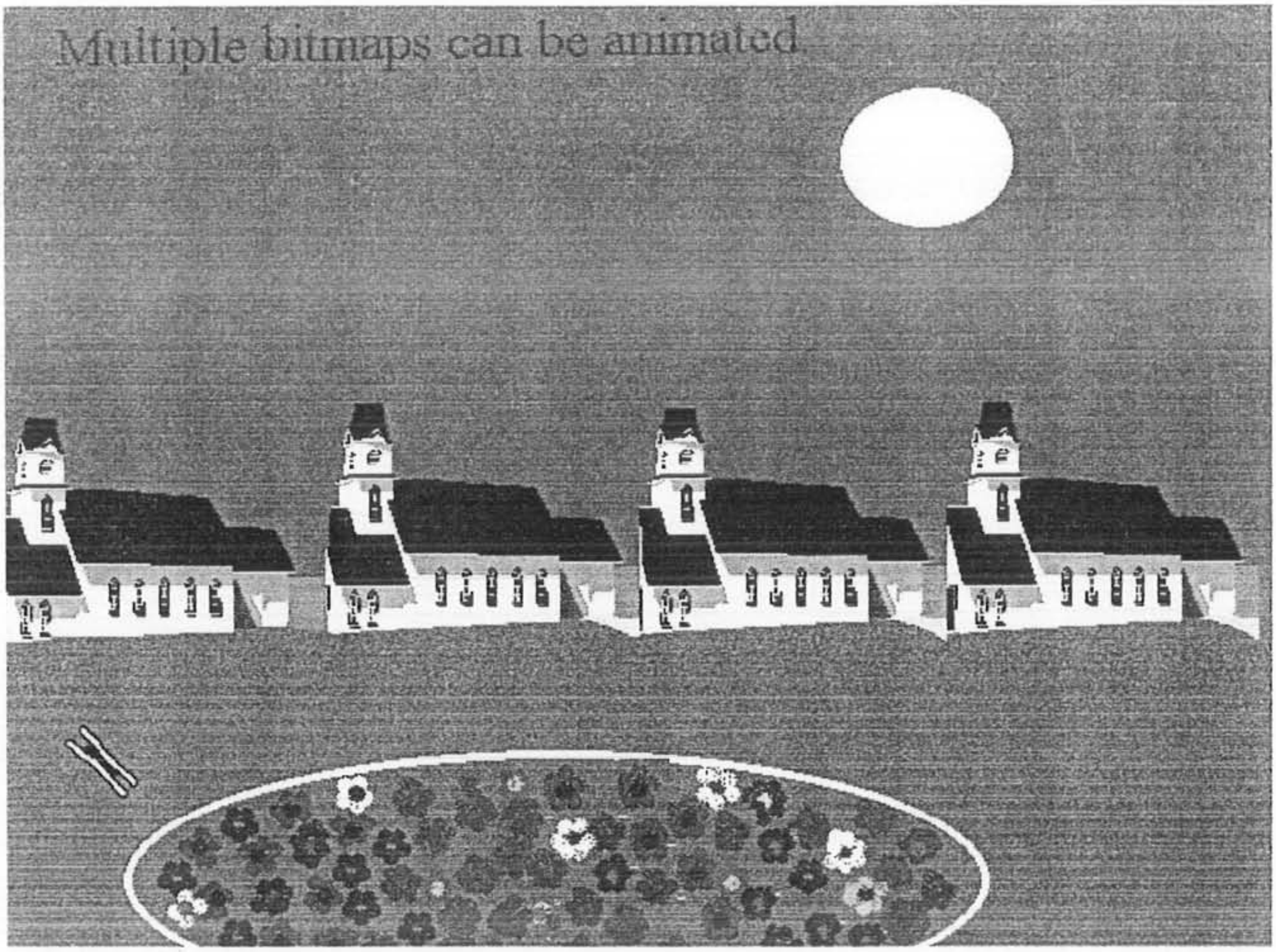
6.NUM-GAME 2

7.FLCs

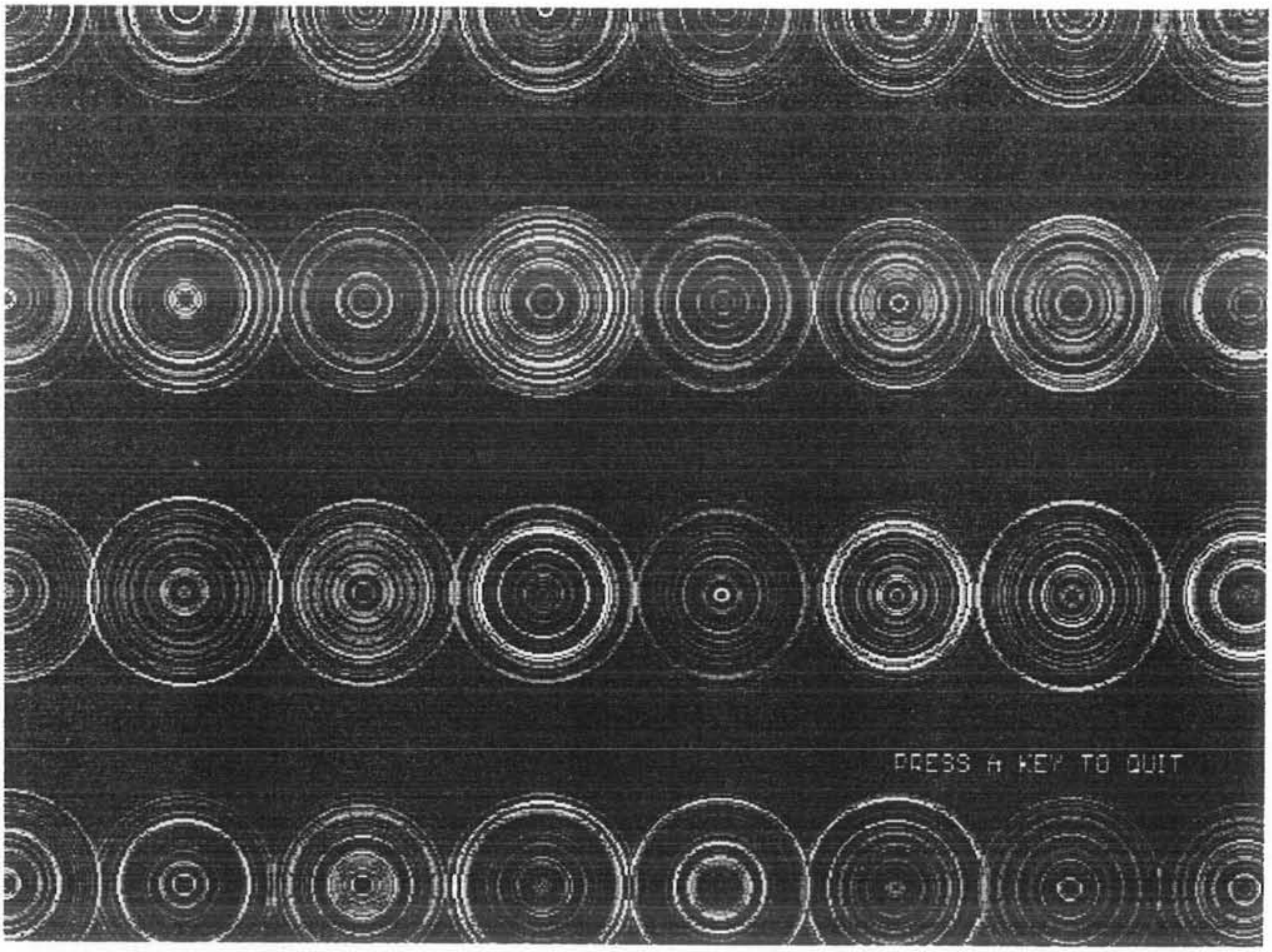
8.AVI

9.SCREEN SAVERS

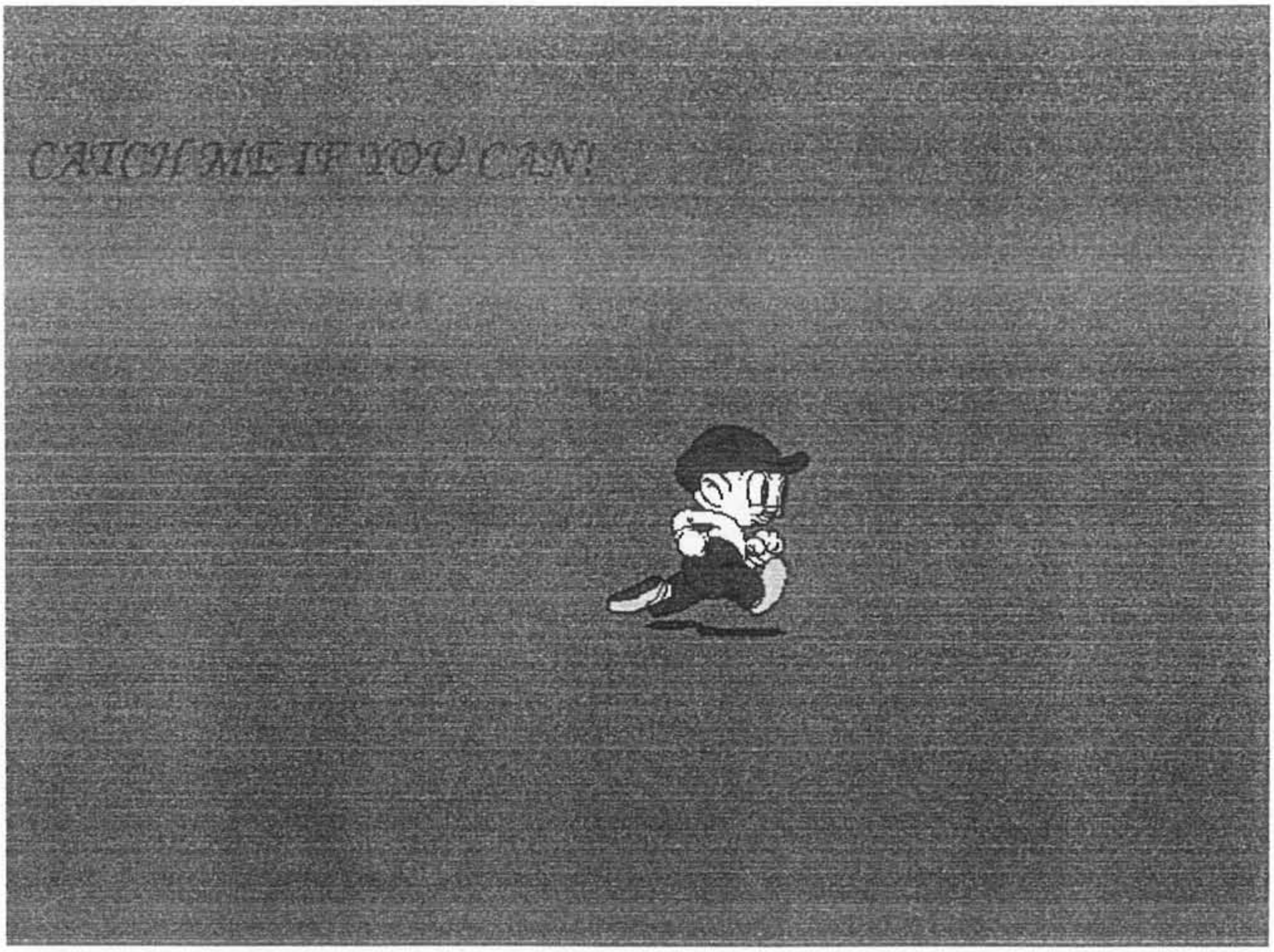
Please enter your choice



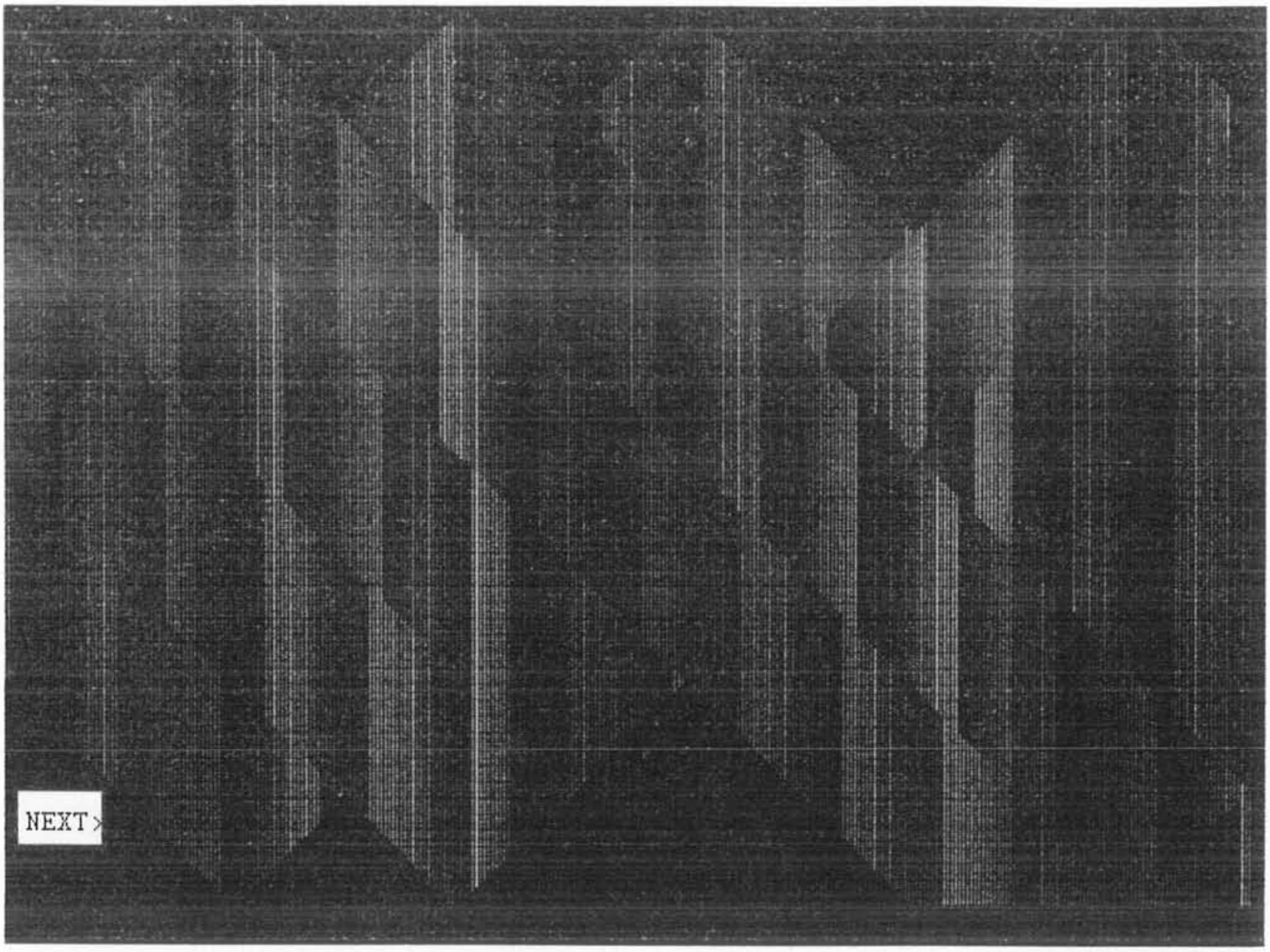
Multiple bitmaps can be animated.



PRESS A KEY TO QUIT



CATCH ME IF YOU CAN!


NEXT>

WELCOME
WELCOME
WELCOME
WELCOME
WELCOME
WELCOME
WELCOME
WELCOME

APPENDIX

Command Reference (alphabetic order)

Brackets [ ] indicate optional arguments.

AGAIN Immediate repeat of a Do/Repeat loop.

ARC x1, y1, x2, y2, x_start, y_start, x_end, y_end, [fill_ flag], [width]
Inscribes an arc within a given box.

BACKPAGE Permits the creation, moving and freeing of an off-screen drawing page.

BOX x1, y1, x2, y2, [line_width], [fill_flag], [xor_flag]
Draws a hollow or filled box.

BUILDBMPfilename, bmp_array[ ],order, dimension

Builds a larger bitmap from smaller ones.

BUTTON(see description) Turns on, off, specify a button for use.

BREAK[DO] Break out of a DO/REPEAT or COUNT/LOOP.

CIRCLE x1, y1, radiusx, [radiusy], [width], [fill_flag], [moire], [xor_flag]
Draws a circle or ellipse at center x, y.

CLEARSCR [r1,g1,b1], [r2,g2,b2],[grad]
Clears the screen to color or gradient.

COLOR r1, g1, b1, [r2, g2,b2] Specifies foreground and background color.

COPY fname Copies a bitmap to the clipboard.

COUNT start, end, step, [counter] Marks the top of a looping construction.

CURSOR (see description) Defines, activates or deactivates a mouse cursor.

DO Used with Repeat for looping

ELSE Used for multiple conditional testing of Xpower expressions.

ENDIF Ends an IF construction.

ESCAPE Turns Escape key on and off

EXIT Exits a program.

FADE fname,type,[x,y],[speed] Fades a bitmap to the screen.

FLOOD x,y,[r1,g1,b1],[r2,g2,b2],[xor_flag]

Flood fills a region with a color.

FLY bmp_array [ ], path_array [ ], [steps], [float flag], [curved flag]

Flies one or more bitmaps along a predefined path.

FONT name, [point], [angle], [bold_flag], [italic_flag], [uline_flag], [strikethru_flag], [Pitch], [family]

Specifies an installed Windows font to use.

FREE object
Frees an object from memory.

GOTO arg
Branches to a specified label.

IF exp
If a condition is met, execute following code or branch to label.

IS exp
following an IF, when condition is met,
execute
following code.

INPUT name,x,y,width,height,max_chars,[initial_string],[tab_order],[r,g,b]
Create a box for user input.

LINE x1, y1, x2, y2, [width], [style], [xor_flag], [r1, g1, b1]
Draws a line on the screen.

LOAD object, [tran_flag], [queue_flag], [pal_flag]
Loads an object into memory.

LOOP
Marks the end of a looping construction.

MESSAGE title, line1, [line2], [line3], [icon], [button], [prompt]

Creates a pop-up Windows message box.

MOVE x1, y1, width, height, x2, y2, [width, height]
Moves a region of the screen.

NET (see description)
Attempts to log-on to an FTP site at the specified domain.

PALETTE [arg]
Turns on or off universal palette.

PAN fname, starting_x, starting_y, direction, #_pixels,[push]

Pans a bitmap.

PASTE [x1,y1,y1,y2]
Pastes a bitmap from the clipboard.

PIXEL x1, y1,[r,g,b], [xor_flag]
Draws a single pixel on the screen at the specified location.

PLAYAVI Provides complete control over the playback of an AVI file.

PLAYFLC file ,x,y,x2,y2,start,end,[argument] Plays an AutoDesk FLC file.

PLAYMIDI [arg] Plays a midi file.

PLAYWAV Plays a WAV file.

POLYGON or

POLYLINEarray [start_idx...end_idx], [close_flag], [fill_flag], [width], [pen], [brush], [xor_flag]

Draws a series of connected lines.

PRINT[value] Draw to the printer and print a page.

REPEAT [if, value] Repeats a Do loop.

RETURN [value] Returns from a subroutine.

SAVE filename, [x1, y1, x2, y2], [save_flag]

Saves an area of the screen.

SCROLL Display and scroll text file in a window.

SCRNSAVER arg Controls the Windows screen saver.

SHADOW dir,offx, [offy] Turns drop shadow for the current font on.

SHOW filename, [x1, y1], [width, height] Displays a bitmap on screen at location x, y.

STORE [filename] Save a file to a remote internet site.

TEXT (see description) Writes a text string to the screen at specified coordinates or turns on/off text attributes.

TILE bitmap,[option] Covers entire screen with multiple copies of a bitmap.

TITLE [text_string] Set the text shown on Windows title bar.

TRAN [arg] Turns transparent on a color for loading.

VIDEO [x],[y],[position] Dictates size and placement of presentation on screen.

WAIT Waits for designated time.

WAITKEY[kc_value],[time] Waits for a time, keyboard or mouse event.

WINDOW coords[ ],[arg],[circle_flag]

Restricts drawing operations to a specified screen region.

Function Reference (Alphabetic)

Functions are commands that return values.
Brackets [ ] indicate optional arguments.

ABS (numb) Calculates the absolute value of the given expression.

ARRAY_END (array_name) Returns index to last element in array.
ASC (char) Calculates the number corresponding to an ASCII character.

BLUE (color) Calculates the intensity of the blue color specified.

BMPHEIGHT (fname) Calculates the height of a bitmap.
BMPWIDTH (fname) Calculates the width of a bitmap.

CHDIR (path) Changes the current drive and/or directory.

CHR (value) Calculates the ASCII character corresponding To a given value.

COPYFILE (src, dest) Copies a file from source to destination.

DATE_TIME (style) Returns date and time as string or elapsed seconds.

DELETE (fname) Deletes a file from disk.

DISKINDRIVE (drive letter) Checks for the existence of a disk in a removable drive.
DISK_SPACE (drive letter) Returns the free space on a disk drive.
DISK_VOL (drive letter) Returns the volume label of a drive.
DOS_ENVIRO (enviro_variable) Returns the DOS environment string.

DURATION (filename) Returns the total number of milliseconds in a Midi file, or total frames in FLC or AVI files.

EVENT (see details) Create autonomous object/event, such as a timer, mouse or key press.

EXEC (program,[arguments],[window type])
Executes a DOS or a Windows program.

EXISTS (fname) Checks for the existence of a file.

FACTOR (number) Returns the factorial for that number.

FAKE_EVENT (name, type,[arguments…]) Fake an autonomous event.

GETBMPX (filename) Returns embedded x coordinate of bitmap file.

GETBMPY (filename) Returns embedded y coordinate of bitmap file.

GETDIR ( ) Returns a string containing the current drive and directory.

GETDRIVETYPE ( ) Identifies the type of a specified drive.

GETPIXEL (bitmap, x, y) Gets the RGB value of a bitmap at coordinate x, y.

GREEN (color) Returns the intensity of the green color specified.

HEIGHT (string) Determines the height, in pixels, of a string.

HEX (string) Converts a string to its hexadecimal value.

HWINDOW() Returns the handle of the Xpower window.

INSTR (s1, s2, [pos]) Searches for string1 in string2 starting at specified position.

MKDIR(path) Creates a subdirectory.

POWER (base, number) Raises a base number to the power specified.

PRINT_WIDTH( ) Returns number of pixels in x dimension of printer page.

PRINT_HEIGHT ( ) Returns number of pixels in y dimension of printer page.

PRINT XDPI ( ) Returns pixels-per-inch in x dimension of printer page.

PRINT YDPI ( ) Returns pixels-per-inch in y dimension of printer page.

PREFIX (value, width) Returns a string padded with leading zeros.

RANDOM (val1, val2) Creates a random number.

READ (fname, count, buffer, [offset], [seek relative])
Reads a file into a variable.

RED (color) Returns the intensity of the red color specified.

RENAME (fname1, fname2) Renames file1 to file2.
REPLACE (s1, s2, s3) Replaces string1 with string2 in string3.

RGB(r, g, b,[mask]) Returns a 32-bit Windows value of specified R,G,B values.

RMDIR (path) Removes a specified directory and returns an error code.

SETBMPXY (filename, x, y) Set embedded x, y coordinate of bitmap file.

SIGN (number) Determines whether a number is positive, negative or zero.

SIZE (handle) Calculates the size of a file.

SQRT (value) Calculates the square root of the specified value.

STATUS (filename) Returns the current playing position in Midi, FLC and AVI files.

STRFIELD (index, string, delimiter) Returns indexed field/word/token.

STRFILL (Len, char) Makes a string of the given length with the given character.

STRLEFT (STR, len) Returns a string whose length is Len, starting at the leftmost of STR.

STRLEN (STR) Returns the number of characters in a string.

STRLOWER (STR) Returns the lower case of the specified string.

STRMID (STR) Returns a string from the middle of STR.

STRRIGHT (STR, len) Returns a string copied from the right of a specified string.

STRUPPER (STR) Calculates the upper case of the specified string.

SUFFIX (STR, char, length) Returns a string padded with trailing characters
VERSION ( ) Determines the version of Windows.
WAVE_GETVOL (device) Gets WAV audio volume.

WAVE_SETVOL (volume, device)   Sets WAV audio volume.
WEB (command)   Send command to web browser.
WIDTH (string)   Determines the width, in pixels, of a string.
WIN_DESTROY (hwnd)Destroy a window.

WIN_EDITBOX (hwnd) Create a scrollable text Window.
WIN_GETTEXT (hwnd) Get contents of a text window.
WIN_SCROLLTEXT (hwnd, vertical_count, horizontal_count)

Scroll text in a text window.

WIN_SETTEXT (hwnd) Set text in a text window.
WIN_SHOW(hwnd)   Display or hide window.
WRITE (fname, count, buffer, [offset],[seek relative])
Writes from an array to a file.

Limits

Maximum size of a single .G file is 65500 bytes.
Maximum .G files you can load is 256.
Total lines of loaded script cannot exceed 24000.

Maximum length for a single line of script is 1024 characters.
Maximum length for a variable is 80 characters.

Maximum length of a command or function argument, or INPUT box line, is 255 characters. This is also the maximum length of a quoted text string used in a TEXT command. Place long text strings in a file for loading and displaying with the TEXT command.

Maximum length of text file 32,000 bytes.

Maximum size of LOADed objects is available memory, except over the Internet, where the maximum size is 512000 bytes.

Maximum number of queued LOAD requests is 512.

Maximum Number of Global variables 512
Maximum number of local variables active at one time is 256.
Maximum number of arrays active at any time is 64

Maximum nesting levels for COUNT/LOOP is 10 deep.
Maximum nesting level for user-defined functions is 32.

Maximum number of arguments for a user-defined function is 10.

Maximum number of active BUTTONs is 32. If you hit the limit, turn some buttons off.
Maximum number of active SCROLLing text boxes is 32.
Maximum simultaneous FLCs is 10.

Maximum number of bitmaps for FLY animation is 200.
Maximum x,y points on a FLY path is 100.

Maximum number of variable WATCHes in debug mode is 32.

XV. Appendix

Fades

Number	Effect
1	Clockwise dissolve starting at 12:00.
2	Clockwise dissolve starting at 9:00.
3	Clockwise dissolve starting at 6:00.
4	Clockwise dissolve starting at 3:00.
5	Counter clockwise dissolve starting at 3:00.
6	Counter clockwise dissolve starting at 12:00.
7	Counter clockwise dissolve starting at 9:00.
8	Counter clockwise dissolve starting at 6:00.
9	Left to right, vertical blinds wipe 2 passes
10	Bottom to top, vertical blinds wipe, 2 passes
11	Bottom to top, vertical blinds wipe, 1 pass
12	Inside out, diamond aperture.
13	Slide in from bottom.
14	Slide in from left.
15	Slide in from right.
16	Slide in from top.
17	Bottom up wipe.
18	Left to right wipe.
19	Right to left wipe.
20	Top to bottom wipes.
21	Vertical blinds.
22	Horizontal blinds.
23	Vertical comb.
24	Horizontal comb.
25	Corner aperture in
26	Serpentine horizontal.
27	Cross-fade.
28	Split screen horizontal, left to right.
29	Split Screen horizontal, right to left.
30	Split screen vertical, left to right.
31	Split screen vertical, right to left.

32 Push right to left.
33 Push left to right.
34 Push top to bottom.
35 Push bottom to top.
36 Circle aperture outside-in.
37 Circle aperture center out.
38 Mirror backwards (left to right).
39 Mirror upside-down (top to bottom).
40 Mirror upside-down and backwards.
41 Cross in from edges.
42 Cross out from center.

43 Slide diagonal up and right.
44 Slide diagonal down and left.
45 Paint-drop left to right.
46 Paint-drop top to bottom.
47 Curtain edge to center horizontal.
48 Curtain edge to center vertical.
49 Curtain center to edge horizontal.
50 Curtain center to edge vertical.