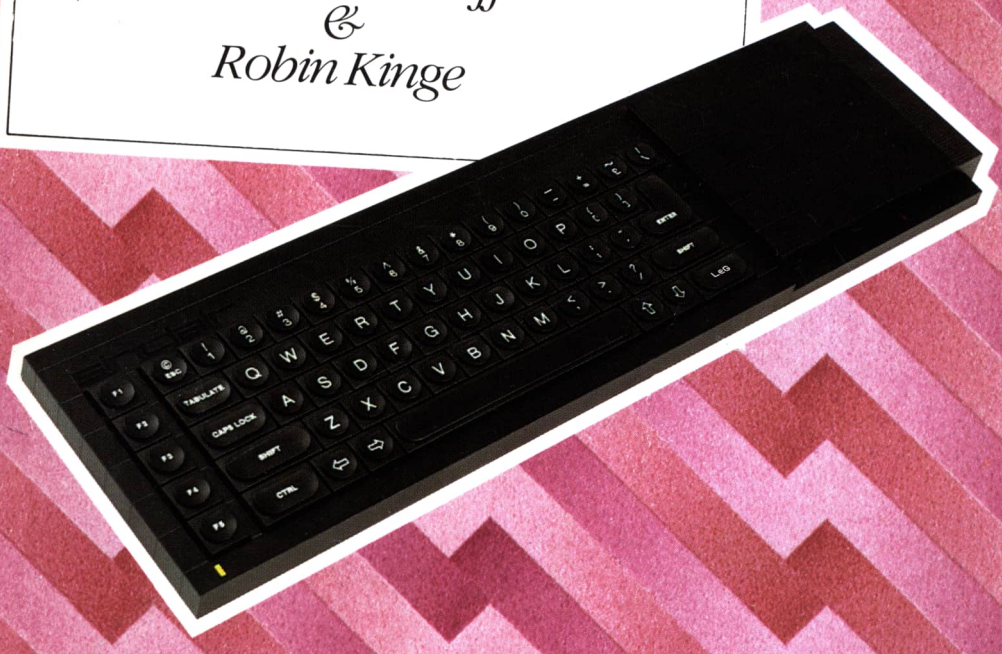
A QL Compendium

Martin Gandoff
&
Robin Kinge



A QL Compendium

A QL Compendium

Martin Gandoff and Robin Kinge



ADDISON-WESLEY PUBLISHING COMPANY
Wokingham, England • Reading, Massachusetts • Menlo Park, California
Don Mills, Ontario • Amsterdam • Sydney • Singapore • Tokyo
Mexico City • Bogota • Santiago • San Juan

The publisher gratefully acknowledges the help of *Personal Computer News* magazine for a subroutine which occurs in certain of the game programs in this book.

© 1984 Newtech Publishing Ltd

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of the publisher.

The programs presented in this book have been included for their instructional value. They have been tested with care but are not guaranteed for any particular purpose. The publisher does not offer any warranties or representations, nor does it accept any liabilities with respect to the programs.

To the best of the publisher's knowledge, the contents of this book in no way transgress any other copyrighted material. Any similarity in idea or program is purely coincidental.

Cover illustration by Chris Smithers

Print and binding by HGA Printing Company Ltd, Brentford, Middlesex

ISBN 0-201-15438-2
ABCDE 8987654

British Library Cataloguing in Publication Data
Gandoff, Martin
A QL Compendium.
1. Computer games 2. Microcomputers
I. Title II. Kinge, Robin
794.8'028'5425 GV1469.2

ISBN 0-201-15438-2

CONTENTS

Preface		vii
Chapter 1	How to Make Use of the QL	1
Chapter 2	Programming Hints and SuperBASIC	8
Chapter 3	Computer "Art"	20
	The "Pretties"	20
	Rotation	26
	The Drawing Program	29
Chapter 4	Games of Skill	35
	Wordslide	37
	Reverse	41
	Le Chop	45
	Flasher	51
	Fruit Machine	55
	Snails	65
	Pennies	70
Chapter 5	"Weapon" Games	75
	Chameleon	77
	Accuracy	83
	Depth Charge	88
	Leveller	94
	Strafe	99
	Mortar	104
Chapter 6	Modern Games	111
	Locate	113
	Batter	117
	Colours	122
	Mallet	126
	Discs	130
	Formula 1	134

Chapter 7 Traditional Games 139
Horse Race 141
Os and Xs 147

Chapter 8 Two-user Games 155
Tennis 157
Space War 163

ABOUT THE AUTHORS

Martin Gandoff has been involved with data processing since 1966 in business and technology. He has lectured at a number of State and private colleges and is a Senior Lecturer in Systems Analysis Design at Slough College of Higher Education.

Robin Kinge is a freelance programmer with a wide knowledge of computers and computer games.

PREFACE

We have been making use of the SINCLAIR QL for the last four or five months and we think it is a very nice machine in all respects, both for games and for business. It is great fun to use and has lots of facilities.

This book presents a great range of games for you to key in and play and we hope you will want to make changes to them so as to allow for your own variations. Before long, you should be able to start writing your own games programs.

The first chapter of the book will help you enter and run a particular game in the most efficient way and to sort out some of the mistakes that you are bound to make. Some of the games programs are quite long and even the most careful programmer can make a mistake.

The next chapter tells you something about the SuperBASIC statements used in the games and explains the way we have written the programs using the very powerful SuperBASIC statements and commands.

Chapter 3 introduces the machine for graphics and art. First there are some (what we call) "pretties" which show you just how nice a display you can get from a very small program. Then, for those who want to add scenes to games, there is a clever little program that allows you to generate "solid" shapes. Next, we introduce you to a drawing program that can be used to paint very detailed pictures. Once you are familiar with writing games, you might be able to use this to build up really nice backgrounds, perhaps for adding to adventure games.

The rest of the book contains lots of games grouped into different types. We hope that you will try all of them, and we expect you to get as much enjoyment out of playing the games as we did in writing them.

Finally, our thanks go to Marc Diprose for writing many of the games and for his artistic ability in making many of the games look so nice, especially Le Chop and Chameleon. He also produced the superb "painting" shown in Chapter 3, using his drawing program.

Martin Gandoff
Robin Kinge
October 1984

Chapter 1

HOW TO MAKE USE OF THE QL

Introduction

The QL is finally available and many young people are starting to make use of it. We think it will be a great success for hobbies, in education and in business and although we want you to try out the games we have written, we hope you will soon become expert in the use of SuperBASIC and quickly learn to write your own games.

To enable us to write and develop the games shown in the book, the equipment we used consisted of:

- A QL, version JM. (There are a number of different versions of the QL about. To see if yours is the same as ours, try:

  PRINT VER$

  The QL will tell you your version.)
- A Microvitec CUB colour monitor, model 1441.
- An Epson FX80 printer with a serial interface for listing the programs.

Apart from a few changes to the printer interface cable (you need to join together the blue and white wires), the QL and the accessories provided by Sinclair were used exactly as they came in the original pack.

Incidentally, the QL interfaced quite nicely with a Spectrum, using the network facility and, at one point, we were sending SuperBASIC program lines to the Spectrum so that it could print them out on its thermal printer.

A word of warning

We must make an important point about the QL. The computer was introduced over six months ago; since then, Sinclair have made many changes, both to the hardware and the software. As you can see from the screenshots, all the programs were entered and tested FOR OUR VERSION. If you have the same version and you do not make any mistakes, your games programs must run, but there is always the small chance that a later or earlier version of the machine may be slightly different. If this is the case, we obviously cannot take any responsibility. You will need to do some experiments with your machine in order to identify the problems this might cause so that you can find a way round them.

Don't be too worried however; we have only found a few items of SuperBASIC that changed for our version, so with luck later ones will not be very different.

After working with five or six different QLs, we are not a hundred per cent happy about the Microdrives, although it could be that we had a suspect batch of cartridges. Hopefully, the production versions of the QL will give no trouble.

However, we will shortly be telling you about a procedure that you should follow when using Microdrive cartridges. This should minimise the possibility of losing programs by mistake or because of Microdrive failure.

Incidentally, you may be interested to know that we used a Spectrum Microdrive connected through the expansion connector (next to the reset button). Over several days it had quite a lot of use and performed without any trouble. It was also rather faster than the QL Microdrives.

Now we will look at some facilities concerned with handling the programs you are to enter.

Entering programs

The QL keyboard

The keyboard looks nice and has a light action. After typing for 24 hours, you shouldn't have sore fingers. Don't forget that there is an auto-repeat on all keys if you press a bit harder.

We think that the keyboard is nicely laid out apart from a few exceptions. The _ and — are on the same key and you have to shift to get _. Hopefully, you will make plenty of use of this underscore character in order to make up sensible and readable variable names and it would be nice if their positions on the key were reversed.

It would also be nice if there was a single "delete" or "rub-out" key; it is a bit of a nuisance to have to use "control" and "←".

Another pity is that you can't get a space with the "shift" key down.

A word of warning - don't use the combination "control"/"alter"/7, or you will crash the machine and have to reset.

Starting up the QL

The Sinclair manual shows you how to connect the QL to the power supply and to your monitor or TV. When you are ready, press the "reset" button on the right-hand edge of the QL. When the message appears, press the F2 key in order to select Mode 8. This will give you a large screen area for entering your program.

If you press Mode F1 you will get the display set for three windows, which is very nice when writing small programs because you can see the program listing on the left and the program running on the right. However, many of you will be using TV sets, for which the three-window display is not possible. Also you will be entering fairly large programs generally; to make things easier for you both to read and enter, we have laid out the program lines so that they will fill up most of the screen.

The Microdrives

When you have set up the machine and are ready to start entering and playing the games, make use of the Microdrive cartridges. You can save your programs on these

in two ways. First of all, the whole program can be entered and then saved on a cartridge when it is working, but you may find it convenient to enter the program in parts, perhaps spread over several days. We will show you shortly how you can do this by making use of MERGE.

Before you can save programs on the cartridge you must carry out a formatting procedure. In order to do this, you place the cartridge in Microdrive 1 and then type in the command:

```
FORMAT MDV1_name
```

To format a cartridge in drive 2, you say:

```
FORMAT MDV2_name
```

Here, "name" is the name you want to give to this particular cartridge. Don't choose a silly name. Remember you may have many programs stored on many cartridges and it is nice if you can give each cartridge its own name. As a suggestion, use a name that contains either a description of what is on the cartridge (say GAMES_Chapter_2) or a name which tells you when you created the programs on the drive (perhaps a combination of the date and time).

After a short time, the QL will tell you that the format job is finished. You should then do it all over again using the same name as before. In all, run the FORMAT program three or four times in order to be safe.

Once you have formatted the cartridge, put the name on a sticky label, both on the cartridge container and on the top of the cartridge itself. This way you should not get cartridges mixed up.

If you want to see the name you have created on a cartridge put it in, say, Microdrive 1, and then enter this command:

```
DIR MDV1_
```

This command is asking the QL to tell you which programs you have stored on the cartridge you have inserted in the drive. The listing that appears on the printer is headed up by the name of the cartridge it is taking the "directory" listing from.

In this case, of course, there aren't any programs on the cartridge yet, but it will still print the cartridge name that you gave when you formatted it.

Once you have entered a program and shown that it works, to save it (say in Microdrive 2), you enter this command:

```
SAVE MDV2_name
```

"Name" is the name you want to give your program. As with the cartridges, use fairly sensible names for your programs. For example, having entered "Colours" save it as "Colours_V1". Later on when you want to make changes to the program, you load it to the same name, make your changes then, when you are happy, save it under a new name, perhaps "Colours_V2", but don't remove the earlier version until you are sure that the later one works.

To load the program that you have saved, you load the cartridge in, say, Microdrive 2, then enter:

```
LOAD MDV2_name
```

where "name" is the name of the program you want to load.

After you have been working on a program for some time, you may have a number of versions, and when you want to remove a program you enter:

```
DELETE MVDn_name
```

where n is the number of the Microdrive and "name" is the name of the program you want to delete.

If, during testing, you want to remove certain lines from the program, you can use the command:

```
DLINE n TO m
```

The n and m mean that you can delete a single line or a group of lines and the TO option means you do not have to specify the start line, or alternatively, the end line. For example:

```
DLINE 500 TO
DLINE TO 50
DLINE 200 TO 350
DLINE 100
```

The first of these deletes all the lines from line 500 onwards, the second deletes all the lines up to and including line 100 and the third removes all the lines between 200 and 350. The last just removes one line.

Another useful command allows you to key in part of a program and then merge into it some SuperBASIC that you have previously saved:

```
MERGE MDVn_name
```

This will merge the lines of program "names" with those you have keyed into memory, on a line number basis.

MERGE is much more useful when you have written a piece of program logic that can be made use of in several programs. Provided you are careful with the line numbers, having saved a subroutine you can merge it into as many programs as you like.

A safety procedure

To allow for you making mistakes (where you delete the wrong version of a program, etc.) and just in case you have trouble with the Microdrives, we suggest that you carry out the following procedure **EVERY TIME YOU ENTER A PROGRAM:**

- Put a cartridge in each drive slot but don't push them in straightaway.
- Enter as much of your program as you have time for, or 50 lines, whichever is less.
- Push in both cartridges and save what you have entered on both of them, then pull out the cartridges.

When you are ready to continue, press reset, select F2 and carry on entering lines until you run out of time or have entered another 50. Then:

- MERGE the earlier saved name from Microdrive 1, SAVE it with a new name, say "name_1" on drive 1, and pull out the cartridge.

- DELETE the old program from drive 2, SAVE the merged version again with the new name on drive 2 and remove the cartridge.

In this way, drive 1 will always contain the latest version and the earlier version. Drive 2 will contain a copy of the latest version.

This may seem rather complicated, but you should try to get into a sensible routine to save yourself trouble. So every time you add something or make a change, take a copy on both cartridges **AND** keep a copy of the previous version just in case your changes are wrong.

One other command that you might find useful is RENUM. After you have been entering program lines and then inserting lines you have forgotten, your program line numbers may become a bit messy. You will see later that we have generally tried to number our programs so that there is a gap of 10 between each line number. If your program does not look like this and you want to change it, you use the renumbering facility like this:

```
RENUM a TO b;c,d
```

This enables you to renumber all the lines from a to b giving them a starting line number of c, in steps of d.

The QL editor

Once you have entered lines of program, you will find certain mistakes; to enter "editor" mode you type in:

```
EDIT n
```

where n is the line number that you want to change.

The editor displays the line on the screen with the screen cursor lined up on the right-hand side of the last character in the line. You can then use the ← or → keys to line up on a particular character. Control/→ deletes a character to the right of the cursor; control/ ← deletes a character to the left of the cursor. If you want to add something, you just line up the cursor to the place where you want to make the addition and add the required characters. You can make use of the ←, → and the control combinations, when you are entering program lines so that if you make a mistake you can backspace with the ← key and enter what you really wanted the line to say.

Be careful with the editor. If a line is bad, unless you correct it, it will not be passed into the program. Even worse, if you are using an old AH version of the machine, if the line is bad you have to enter it all over again.

Diagnostic error messages

This is a very difficult subject because you are likely to make many different kinds and combinations of errors when entering your lines of SuperBASIC. The interpreter program examines your coding and tries to work out if you have done anything wrong. When it detects an error, it prints a diagnostic error message, which tries to tell you what you did wrong, where you did it and, if possible, how to correct it.

The QL has a fairly wide range of error messages some of which will appear very often, while others are actually quite rare.

The message format is as follows:

At line n "error message"

where n is the line number containing the error, and the error message is selected from a standard list.

The following is a selection of error messages that you are likely to generate, and an explanation of how to look for the error that is causing the message:

already exists The QL has found that the name of a program you are trying to save on a Microdrive cartridge has already been used for another program. You will either have to change the name of the program you want to save or else delete the program that is already there.

bad line This is a rather vague error message, unfortunately. It means that you have made a mistake somewhere in the line and have broken the rules of SuperBASIC; in other words, you have made an error in your syntax.

You must look to see if you have mis-spelt one of the keywords, or perhaps missed out a bracket, or put in an extra operation. Close examination of the line you have typed in should help you to find the error, but it is only really experience that counts here.

error in expression The commonest cause of this error message is when you use a variable that has not had a value placed in it, e.g. adding A to B where the statement is the first reference to A. Since A has not been assigned, the machine can't possibly know what value it is dealing with.

channel not open This means you are carrying out an operation involving a channel specification where the channel is not open. Check the line to make sure that the number after the numeric symbol is correct. It may be that you are trying to print something to channels 0, 1 or 2 and have put a different digit. (Channels 0, 1 and 2 are always open, whereas other channels always required a channel OPEN statement.)

drive full This means that you have filled up your Microdrive cartridge; you must put in an empty one if you want to save the current program. (Don't forget to FORMAT the cartridge before use.)

not complete This is the message that you get when you cancel a program that is running by entering a combination of the control key and the space bar.

not found This usually appears because you are trying to load or merge a program that should be on a cartridge but isn't! You are probably using the wrong name in your LOAD or MERGE command.

not implemented This is a funny error message which usually means that either you are using a SuperBASIC keyword that is no longer allowed (or is yet to be allowed) or you are using one in a way that is not completely wrong, but is nevertheless not quite as specified in the manual, e.g. trying to assign a value to A, when you have previously said DIM A(10).

out of range This will usually be because you have entered a subscripted variable, where the subscript is outside the range of the DIM statement which defined the bounds of the variable. It could also occur where you set up a window and then try to carry out an operation using the same channel number for a location which is outside the window.

bad or changed medium This message will often occur when you are trying to save a program; it means that either something went wrong with the SAVE operation, or possibly you have damaged the cartridge. This message is slightly different from the others in that it is preceded by the name given to the cartridge when FORMATted, instead of the words "AT LINE n".

For a more detailed list of error message, see your *QL User Guide*.

Chapter 2

PROGRAMMING HINTS AND SUPERBASIC

Program structure

The first thing you will probably notice when you look at the programs we have written is that they seem to be very long indeed. You will also notice that the reason for this is that there are very few multistatement lines.

There are two major reasons for this. First of all, people have got into the habit of making their BASIC take up as little room as possible, because it is more efficient for the translator program since many games computers have a limited amount of memory. The QL has a very large memory and we need not worry about there not being enough space available to translate our SuperBASIC into machine code.

The other reason is that people have become lazy and seem to prefer saving a little time when entering programs, at the expense of making them readable. We have made use of spaces, both in program lines and between them. We hope that you will take note of what we have done and will enter the program lines in the format in which they are shown in the book. Of course, if you are desperately short of time, you could leave out some of the spaces. For example, you will see many cases where we have "indented" lines inside an IF...THEN or a FOR...NEXT. We think that this makes the program easier to read but if you cannot be bothered, then do not put them in.

The same applies to multistatement lines, but be very careful since SuperBASIC *does* allow for particular multistatement lines and if, for example, you have a multistatement line that starts with an IF, you do not need the ENDIF.

If you are taking a group of lines that are surrounded by IF and ENDIF, you may introduce an error because SuperBASIC may take the statements on the line with the IF as being the whole of the IF complex.

An example of this might be as follows:

```
100 IF X > 10 THEN PRINT X: PRINT "TOO BIG"
120 PRINT "JUST RIGHT"
140 ENDIF
```

You will always print "JUST RIGHT", because SuperBASIC inserts an ENDIF at the end of line 100 and ignores the one in line 140.

The next major feature of the programs is that you will not see many GOTO or GOSUB statements. SuperBASIC has been designed for use in "structured programming" which is a method of writing programs that, amongst other things, does not very often need the use of GOTO.

If you have been clever in the past, you will already know about using GOSUB and RETURN for subroutines to make the logic of your programs easier to understand and, therefore, easier to program. You may also know how to handle functions with the DEF FN statement. SuperBASIC has a very powerful extension to these in which you can define a procedure in a manner something like the function, but with as many lines as you like to make up the procedure. This means that we do not need GOSUB.

The way to define a procedure in SuperBASIC is as follows:

```
200 DEFine PROCedure procedure_name (parameters)
.
.
.
290 END DEF procedure_name
```

The procedure consists of all the statements between the DEF and the END DEF.

The name in the END DEF statement is not needed but it is a good idea to put in the name that was used in the DEF since a procedure may include other groups of statements that could include an END – putting a name will help to make the end of the procedure stand out.

The parameters specified in the DEF statement are the names of the variables whose values you want to pass into the procedure. For example, in the main part of the program you may want to make use of a procedure that prints a number of circles of a given radius. You might say something like:

```
300 Draw_circles number,radius
```

This means that you want to call the procedure telling it to produce "number" circles whose radius is "radius".

The procedure may have been defined like this:

```
700 DEF PROC Draw_circles (first, second)
.
.
.
800 END DEF Draw_circles
```

When control is passed to the procedure, it will associate the number in the variable "number" with the procedure variable "first".

Similarly "radius" will have its value transferred to the procedure variable "second". This can be very useful when writing a procedure to be used in several programs where we do not necessarily want to use the same variable names in the different programs. For example, in one program we may be using the circles to print some flying saucers, perhaps after a selection to see which enemies are approaching, and the number of circles may be held in a variable called "enemies". A quite different game might need the same procedure, this time as part of a pool game, and the variable used is perhaps called "number_of_balls".

You will see extensive use made of procedures in the games programs. Be very careful when using procedure names to make sure that you spell them properly. Also, be extra careful when you enter SuperBASIC lines because a mis-spelling of a

keyword may appear to the QL like a procedure name; there is no difference between saying RETUNR and saying FIRE_GUN. Both will generate a "bad line" error.

The "graphics" statements

Now let us look at what we will call the graphics statements. We can divide these into three main groups:

- Statements relating to origins and the size of characters and drawings.
- Statements relating to the windows and INK and PAPER.
- The drawing statements themselves.

Mode

This sets the resolution of the screen and also the number of colours that can be used. When you start up the QL, F1 sets you into Mode 4, with window 0 at the bottom and windows 1 and 2 side by side above window 0. F2 gives you windows 1 and 2 superimposed, so that anything you do to one of them affects the other.

With Mode 4 only four colours are available, but the resolution allows 512 pixels across a line and 256 down the screen. Mode 8 allows eight colours, but only gives 256 pixels across. The minimum character size for Mode 8 is much larger than for Mode 4 (see CSIZE below).

Although Mode 4 only allows four colours, by making use of STIPPLE (see Colours below), the actual variation in colour can be very much extended.

Most of you will be using a TV set with your QL and although you are allowed Modes 4 or 8, you will not find it all that easy to read the characters in Mode 4.

With a colour monitor you have the choice of Mode 4 or Mode 8.

Origins

The reference point for statements that print text is the top left-hand corner of the window in which the text will appear. For graphics statements, the origin is the bottom left-hand corner.

You can change the current location of the cursor with:

```
20 CURSOR #n,x,y
```

n is the window channel (discussed shortly) and (x,y) is the location of the left-hand corner of the cursor relative to the text origin of the window you are setting it in.

A slightly different statement is **AT**. CURSOR is handled in relation to pixels. If you want to make a piece of text appear at the same place every time without having to line it up with PRINT and TO, you can say:

```
10 AT #n,a,b
```

n is the window channel and (a,b) specifies the location where the next PRINTed text will appear in the window. The (a,b) are in units determined by the CSIZE in effect and can be thought of as line and column.

Note With a JM version of the QL, the a and b are reversed, i.e. you specify the vertical before the horizontal.

CSIZE

This enables you to select the size of printing characters:

```
100 CSIZE #n,a,b
```

n is the window channel; a is the width of the character and b is the height, both in pixels. The defaults are 0,0 for Mode 4 and 2,0 for Mode 8.

The sizes are obtained thus:

a	*width*	*b*	*height*
0	6	0	10
1	8	1	20
2	12		
3	16		

Windows

These are selected areas of the screen, and a particular window can be specified in relation to the place on the screen where it starts (its left-hand corner) and its length and breadth from this starting point. The statement is:

```
100 Window #n,w,d,x,y,
```

where n is the channel that the window is associated with, w is the width of the window in pixels, d is the depth of the window in pixels, and x and y specify the location of the origin of the window in pixels, relative to the screen origin.

Just to remind you: windows 0, 1 and 2 are always open and will occupy screen space depending on whether you selected F1 or F2 after reset. You will only use the WINDOW statement if you want to change the size or location of a window.

If you want to open another channel for window use, you can say:

```
100 OPEN #n,SCR_wxhaxxy
```

This is telling SuperBASIC to open channel n as a screen window, size w by h, with window origin at (x,y). Note that you **must** put in the underlined separators.

Window statements

There are a number of statements that relate to a particular window.

PAN

This enables you to move a window or part of a window to the right or left:

```
100 PAN #n,d,part
```

where n is the number of the channel involved, d is the number of pixels you want

to pan, and "part" has the following meaning:

0 whole screen
3 the whole of the cursor line
4 the right end of the cursor line, including the cursor.

If d is positive the PAN is to the right, and if negative it will shift to the left.

Note that what we have said here about the direction of the move when d is negative/positive is the opposite of the Sinclair manual. Try for yourself.

SCROLL

This is a similar facility to PAN. It enables you to shift a window or parts of a window upwards or downwards:

```
100 SCROLL #n,d,part
```

where, as with PAN, n specifies channel, d is the number of pixels and "part" is as follows:

0 the whole window
1 the top of the window above the cursor line
2 the bottom of the window excluding the cursor line.

If d is positive then the screen scrolls down, and if negative it scrolls up.

When the PAN or SCROLL statements are used, as the window moves, the colour that comes in to fill the clear space will be controlled by the PAPER statement (covered shortly).

Note The same comment about the manual being wrong applies to SCROLL.

CLS

This enables you to clear a particular window, the background being set by the most recent PAPER statement.

```
200 CLS #n,part
```

As with all of these channel-orientated statements, n specifies the channel associated with the window you want to clear and "part" is as follows:

part = 0 whole screen is cleared
part = 1 the top of the window above the cursor line
part = 2 bottom of the window below the cursor line
part = 3 the cursor line
part = 4 the cursor line from the cursor to the right-hand end

Colours

The QL enables you to set the background colour of a window by using the PAPER statement, and the colour of the "ink" for the characters or drawing on the "paper" using the INK statement.

Before saying anything more about this group of statements, we must discuss

how the QL allows you to make use of the various colours available.

Depending on whether you are working in Mode 4 or Mode 8, the QL has a range of colours available, according to the following table of codes:

colour		*code*
Mode 4	*Mode 8*	
black	black	0
black	blue	1
red	red	2
red	magenta	3
green	green	4
green	cyan	5
white	yellow	6
white	white	7

In addition, it is possible to make use of combinations of colours, called "stipples".

Colour is specified both in the INK and PAPER statements, and has three components, of which the last two can be left out if you only want to use a solid colour. The two statements are as follows:

```
100 INK #n,b,c,s
200 PAPER #n,b,c,s
```

As before, n specifies the channel that is associated with the window to which you are assigning the colour and b is the colour selected from the code table above. If no other parameters are specified the solid colour is obtained either for the background, with PAPER, or for the colour of text or graphics (specified with INK). If you want to stipple two colours, you specify a value for c from the code table, and then a value for s, specifying the stipple as follows:



The "drawing" statements

Now let us look at the drawing statements, i.e. those that are specifically used to produce geometric shapes.

POINT

```
30 POINT #n,x,y,
```

This plots a point in the window attached to channel n, x drawing units across from the window origin and y units up.

LINE

This enables you to plot a line from and to specified points in a window, relative to the graphics origin of that window:

```
40 LINE #n,x,y TO p,q
```

Here you are plotting a line starting at location (x,y) and going to (p,q) relative to the same origin.

CIRCLE

This is a very powerful statement that enables you to draw not only circles but also ellipses, and an ellipse can be specified to appear at an angle to the horizontal axis. The statement has the form as shown below, but you will probably best be able to work out how to use it by looking at Hat (see page 20) or Globe (see page 25).

```
20 CIRCLE #n,x,y,r,e,a
```

where (x,y) is the location of the origin of the figure, and r is the radius you wish the circle to have. If you have specified e and a you are asking for an ellipse; e specifies the ratio of the major to minor areas, and a the angle (in radians) of the major axis to the horizontal. So if you plot a circle and then add an eccentricity of 2, the minor axis of the ellipse will be the same as the radius of the original circle you draw and will be in the vertical. Specifying an angle of 1 (about 60 degrees) will turn the ellipse so that its major axis is at an angle to the horizontal. (Try putting r,e and/or a in a CIRCLE statement in a loop to get some pretty patterns.)

FILL

This is an extremely powerful statement that can have some very surprising effects. Essentially, it can be used to fill plotted shapes. Before you produce the shape you want to fill, you turn the FILL option on by entering:

```
50 FILL 1
```

You might then plot a circle or a square, and this will be filled with the PAPER colour.

You turn FILL off with:

```
100 FILL 0
```

The statement is actually much more complicated, and you will find that it is possible to arrange for a filled shape by just drawing two lines in some cases. You will have to experiment with this for yourself. Try this:

```
10  INK 0
20  PAPER 7
30  CLS
31  REM screen is now white
40  INK 4
50  FILL 1
```

```
60  LINE 0,0 TO 50,50 TO 100,0
61  REM NOW YOU HAVE A TRIANGLE
70  FILL 0
```

OVER

This is another powerful statement, which does not seem to be available with any machine other than the QL. It takes the form:

```
70 OVER n
```

If n is 0 the option is turned off. When you set it to 1, it enables you to carry out a print on the screen, such that the ink characters will overwrite the screen underneath, but the new paper associated with the window will not overwrite the old paper under the new piece of text.

This means that you can carry out some printing, say in black on red, and then set new INK and PAPER for the window, say green on white. If OVER is set on and you print some text, the green printing where it hits the screen will overwrite the previous black on red, but the white paper will not appear with the green.

If you set OVER to –1, again the paper is not affected, but this time the ink colour for the new text is altered in a very subtle way. Again, try this for yourself.

Miscellaneous SuperBASIC statements

Now let us look at a number of statements that are new and unique to SuperBASIC or different from other versions of BASIC.

Before doing so, a word of advice. When you have been running a game and want to start entering or loading another one, **make sure that you press reset before continuing.** If you do not, then a number of options may have been set by the earlier game which you will not want. The commands that you may need to enter to "clean things up" before you carry on could include any of the following:

```
CLEAR
INK
NEW
OVER
PAPER
WINDOW
```

All of these will be "cleared" when you reset.

Now for the variations. The first one to look at is the way that the QL requires the use of RESTORE before a READ. With most versions of BASIC the pointer to DATA statements lines up at the beginning of the list of DATA items you have included in your DATA statements. The QL lines up at the end and you must enter a RESTORE in order to make the pointer align at the beginning. If you forget to RESTORE you will get an "end of file" error message.

The next statement is SELECT. You may be familiar with versions of BASIC that contain ON... GOTO or ON... GOSUB. These are not needed in SuperBASIC. The

best way to show why they are not necessary is to look at the program for Snails on page 65.

Have a look at the procedure called "display" starting at line 1000. The program is working out the next sum for you to do and, before you execute the procedure, lines 540 to 560 produce three random numbers which will be used for the numbers that make up the sum and whether it is an add, subtract or multiply. Line 1030 contains a SELECT statement and, as you will be able to see, we are selecting a variable "f".

If f = 1, we carry out the effect of "GOSUB PLACE 1" where we add a and b together to form the correct sum, and then print the rest of the sum. If f = 2 we "GOSUB PLACE 2", where we calculate the correct number and then print out the fact that the sum is a subtraction, and similarly if f = 3, we sort out the multiply.

This piece of logic could have been achieved without using SELECT but we would have needed three GOSUBs and three subroutines together with their RETURNS.

This way it is all nicely packed together within the procedure.

RND

As you probably know, this is used to generate a random number. SuperBASIC has a different syntax for RND than most versions of BASIC.

You can say:

```
10  RND
```

in which case you get a floating point number in the range 0 and 1. Or you can say:

```
10  RND (a)
```

which generates an integer random number in the range 0 to a, or:

```
10  RND (a TO b)
```

which gives an integer random number in the range a to b.

For example, if you look in the demonstration called Saturn on page 23, line 220 is being used to generate a random ink colour in the range 4 to 7 and the next line is printing stars at random positions on the screen.

FOR. . .END FOR/NEXT

This is slightly different in that a FOR loop can be ended with a NEXT or with END FOR. SuperBASIC has an EXIT statement that allows exit from a loop which you can use in a structure like this:

```
100 FOR J = 1 TO 20
110 —
120 —
130 IF X > Y THEN EXIT
140 —
150 —
160 END FOR J
```

```
170 —
180 —
190 NEXT J
```

The statements after the END FOR are only carried out if the loop ends normally, i.e. when J is greater than 20. If you EXIT, control goes to the statement after the NEXT J.

Sound

We can't finish this chapter without saying something about the sound facility on the QL.

The operative keyword is BEEP. It is very complicated and the best way to find out about it is to experiment with the program below.

The actual format of the statement is:

```
60 BEEP duration,pitch,pitch_2,grad_x,grad_y, wraps
```

You use ←, → to select the parameter to be altered, then the ↑ to increase the parameter and ↓ to decrease it.

Here is the program:

```
100 REMark ************
110 REMark * Beep Edit *
120 REMark ************
130 :
140 MODE 8 : PAPER 0
150 INK 6 : CLS : CSIZE 3,1
160 PRINT TO 7,"Beep Edit"
170 OVER -1 : INK 1 : AT 0,0
180 PRINT TO 7,"____ ____"
190 OVER 0 : INK 5 : RESTORE
200 :
210 DATA "Duration",0,32767,100
220 DATA "Pitch"   ,0,255  ,1
230 DATA "Pitch_2" ,0,255  ,1
240 DATA "Grad_x"  ,0,1000 ,10
250 DATA "Grad_y"  ,0,1000 ,1
260 DATA "Wrap"    ,0,1000 ,1
270 DATA "Fuzzy"   ,0,15   ,1
280 DATA "Random"  ,0,32767,10
290 :
300 DIM lab$(8,8),min(8),max(8)
310 DIM inc(8),act(8)
320 :
```

```
330 FOR loop = 1 TO 8
340   READ lab$(loop)
350   READ min(loop)
360   READ max(loop)
370   READ inc(loop)
380   act(loop) = 0
390 END FOR loop
400 :
410 FOR loop = 1 TO 8
420   AT 1+loop,6
430   PRINT lab$(loop)
440   AT 1+loop,16
450   PRINT act(loop);"      "
460 END FOR loop
470 :
480 po = 1
490 :
500 AT 1+po,6
510 INK 2
520 PRINT lab$(po)
530 AT 1+po,16
540 INK 4
550 PRINT act(po);"      "
560 :
570 a$ = INKEY$
580 IF a$ = "" THEN GO TO 570
590 a = CODE(a$)
600 :
610 AT 1+po,6
620 INK 5
630 PRINT lab$(po)
640 AT 1+po,16
650 PRINT act(po);"      "
660 :
670 IF a = 216 THEN
680   IF po < 8 THEN
690     po = po + 1
700   END IF
710 END IF
720 :
```

```
730 IF a = 208 THEN
740   IF po > 1 THEN
750     po = po - 1
760   END IF
770 END IF
780 :
790 IF a = 192 THEN
800   IF act(po) > min(po) THEN
810     act(po) = act(po) - inc(po)
820   END IF
830 END IF
840 :
850 IF a = 200 THEN
860   IF act(po) < max(po) THEN
870     act(po) = act(po) + inc(po)
880   END IF
890 END IF
900 :
910 IF a = 32 THEN
920   BEEP
930   IF INKEY$="" THEN GO TO 930
940 END IF
950 :
960 a1=act(1):a2=act(2):a3=act(3)
970 a4=act(4):a5=act(5):a6=act(6)
980 a7=act(7):a8=act(8):REMark PHEW!
990 :
1000 BEEP a1,a2,a3,a4,a5,a6,a7,a8
1010 :
1020 GO TO 500
```

Now we can start looking at the programs.

Chapter 3

COMPUTER "ART"

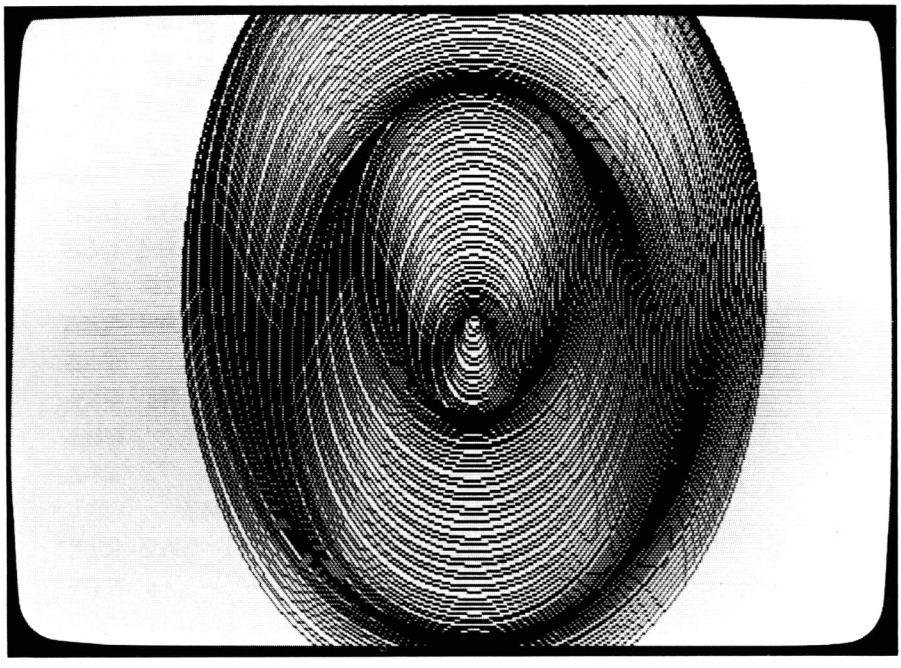
In this chapter, we will first show how nice you can make displays appear with some quite simple programs.

The "Pretties"

Just enter and run HAT, HELIX, COIL, SATURN, SPIRO or GLOBE, and then sit back and watch.

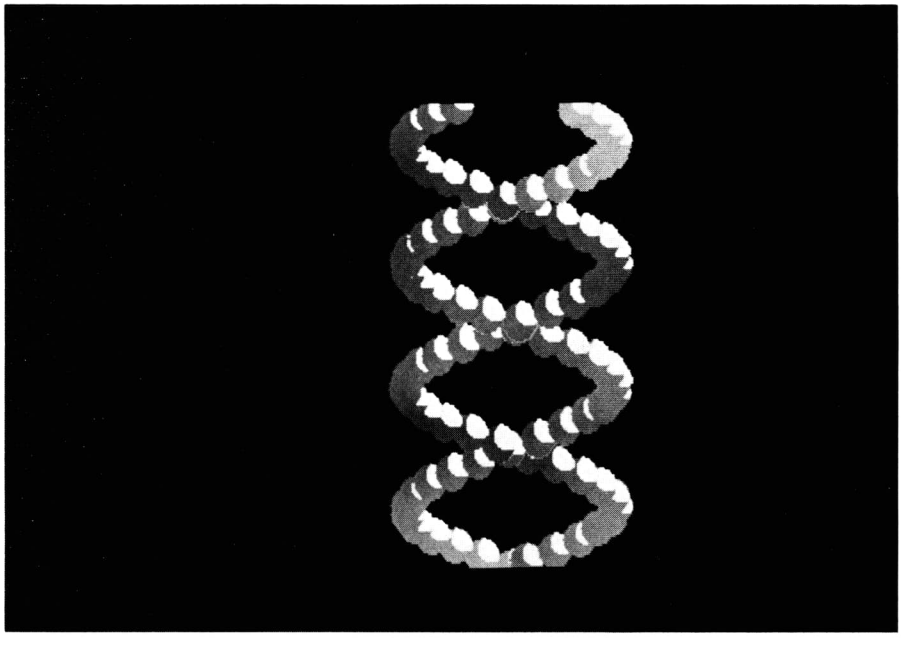
Then look at the programs to see how it is done. You will learn a lot about the "drawing" statements this way.

```
100 REMark ***********
110 REMark * Hat art *
120 REMark ***********
130 REMark
140 MODE 4
150 SCALE 256,0,0
160 PAPER 0
```



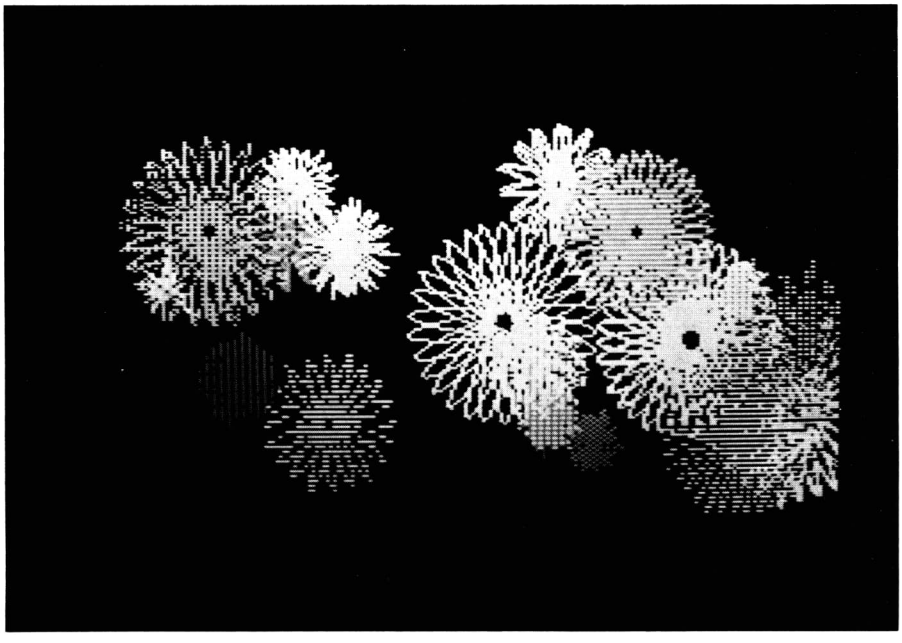
```
170 INK 7
180 CLS
190 FOR n= 0 TO 700 STEP 8
200   z = SIN ( RAD(n) )
210   CIRCLE 180,120+20*z,n/4,.9,0
220 NEXT n
```

```
100 REMark *************
110 REMark * Helix art *
120 REMark *************
130 REMark
140 ic=4
150 MODE 4
160 SCALE 256,0,0
170 FOR n = 0 TO 256 STEP 3
180   INK ic
190   FILL 1
200   CIRCLE 255+ 60* SIN (n),n,10
210   FILL 0
220   INK 7
230   FILL 1
240   ARC (255+60*SIN(n)),n+10
      TO  (255+60*SIN(n))+10,n,PI
```



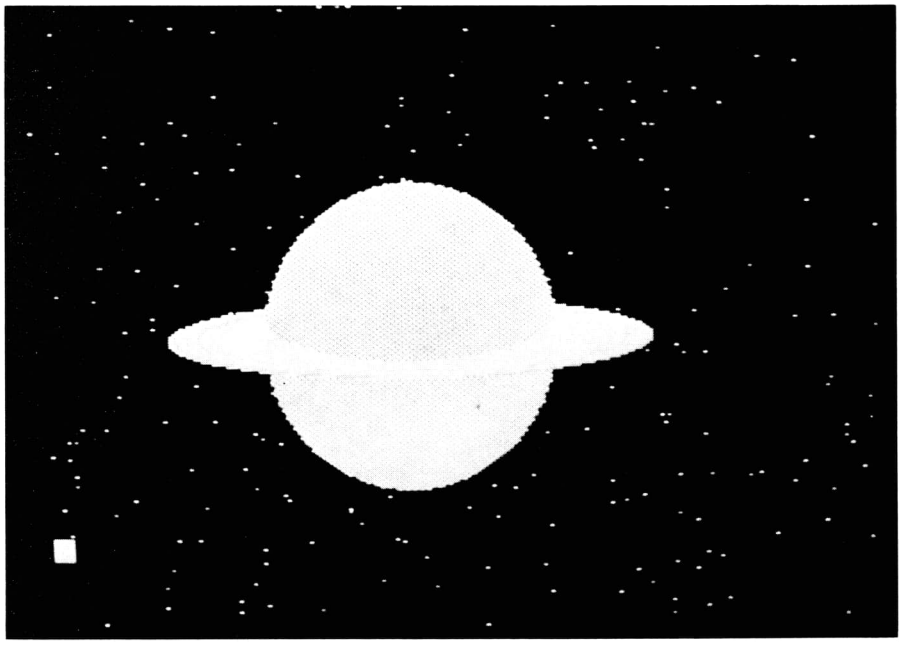
```
250    ARC TO(255+60*SIN(n)),n+10,PI/2
260    FILL 0
270    IF ic =3 THEN ic = 4: GO TO 290
280    ic =3
290 NEXT n

100 REMark ************
110 REMark * Coil Art *
120 REMark ************
130 REMark
140 ic =0
150 MODE 4
160 PAPER 0:CLS
170 SCALE 512 ,-200 ,-100
180 FOR n= 0 TO 900 STEP 8
190    z1=SIN(RAD(n))
200    z2=COS(RAD(n))
210    INK 3+ic
220    FILL 1
230    a = n/3
240    CIRCLE 255+a*z1,80+a*z2,(n/8)
250    FILL 0
260    INK 0
```



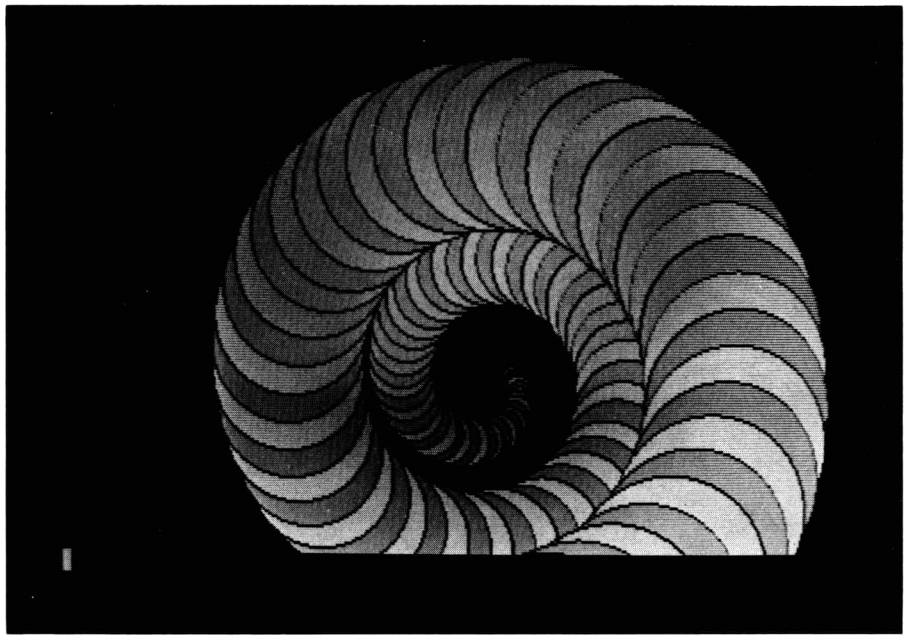
```
270   CIRCLE 255+a*z1,80+a*z2,(n/8)
280   ic=1-ic
290 NEXT n

100 REMark ***************
110 REMark * Saturn art *
120 REMark ***************
130 REMark
140 MODE 8
150 PAPER 0 : CLS
160 WINDOW #2 ,512 ,256 ,0 ,0
170 WINDOW #1 ,512 ,256 ,0 ,0
180 PAPER #1,0 : CLS #1
190 PAPER #2,0 : CLS #2
200 SCALE 256 , 80 , 0
210 FOR n = 1 TO 300
220   INK RND ( 4 TO 7 )
230   POINT RND ( 511) , RND ( 255)
240 NEXT n
250 INK 2 ,7 ,3
260 FILL 1
270 CIRCLE 255 ,120 ,60
280 FILL 0
```



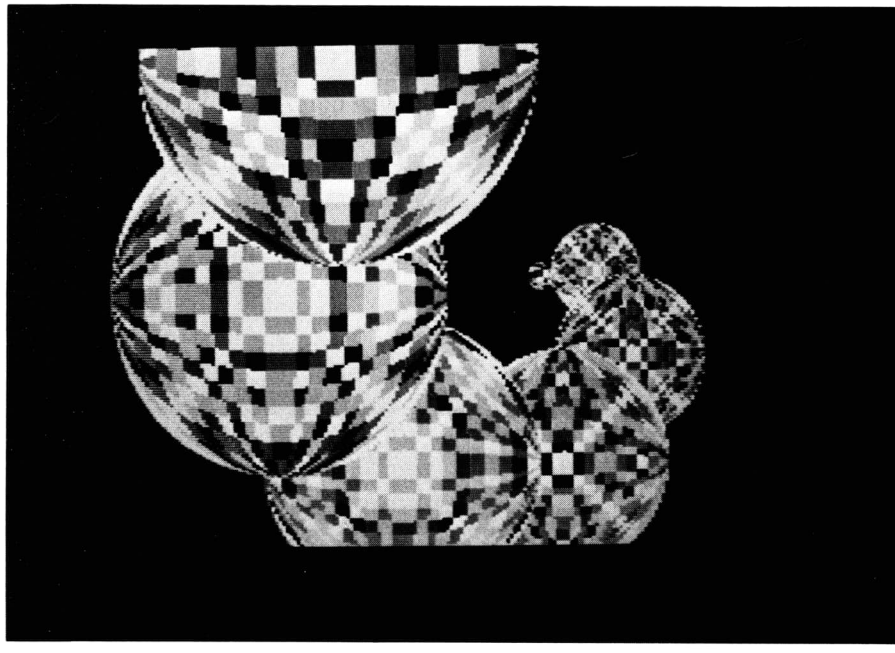
```
290 FILL 1
300 FOR n = 100 TO 75 STEP - 2
310   INK RND ( 1 TO 7 )
320   CIRCLE 255 ,120 ,n ,.15 ,PI/2
330 NEXT n
340 FILL 0
350 FILL 1
360 INK 2 ,7 ,3
370 CIRCLE 255,130,60,.3,PI/2
380 FILL 0
```

```
100 REMark **************
110 REMark * Spiro art *
120 REMark **************
130 REMark
140 MODE 8
150 PAPER 0
160 SCALE 256 ,0 ,0
170 CLS
180 X= RND (50 TO 462)
190 Y= RND (50 TO 206)
200 SIZE = RND (10 TO 50)
210 ROTATE = RND (10 TO 30)
```



```
220 ECC= 1/RND (5 TO ROTATE)
230 INK RND (1 TO 255)
240 FOR ANGLE=1 TO 360 STEP ROTATE
250   CIRCLE X,Y,SIZE,ECC,RAD(ANGLE)
260 NEXT ANGLE
270 GO TO 180
```

```
100 REMark **************
110 REMark * Globe art *
120 REMark **************
130 REMark
140 size = 10
150 CLEAR
160 MODE 8
170 PAPER 0
180 CLS
190 FOR a = 20 TO 360 STEP 50
200   z1=SIN ( RAD(a) )
210   z2=COS ( RAD(a) )
220   x= 255 + ( a/2 ) * z1
230   y= 127 + ( a/2 ) * z2
240   size = a/3
250   INK 7
```



```
260    FILL 1
270    CIRCLE x,y,size
280    FILL 0
290    OVER -1
300      FOR n = 1 TO 0 STEP -.1
310        FILL 1
320        INK RND (1 TO 7)
330        CIRCLE x,y,size,n,0
340        FILL 0
350        FILL 1
360        CIRCLE x,y,size,n,PI/2
370        FILL 0
380      NEXT n
390    OVER 0
400 NEXT a
```

Rotation

This is a program that can produce some startling effects. The idea is to generate a figure using lines or curves made up from points and then to tell the program to rotate the figure a certain number of times to make up a solid figure.

First, you enter the number of points that will be needed to create your basic shape. Then you enter pairs of x,y coordinates in order to map out the shape so as to control the level of detail in the final display.

The program will then ask you how many rotation steps you want. It is a good idea to make this somewhere between 6 and 15 since the screen gets rather crowded with many more.

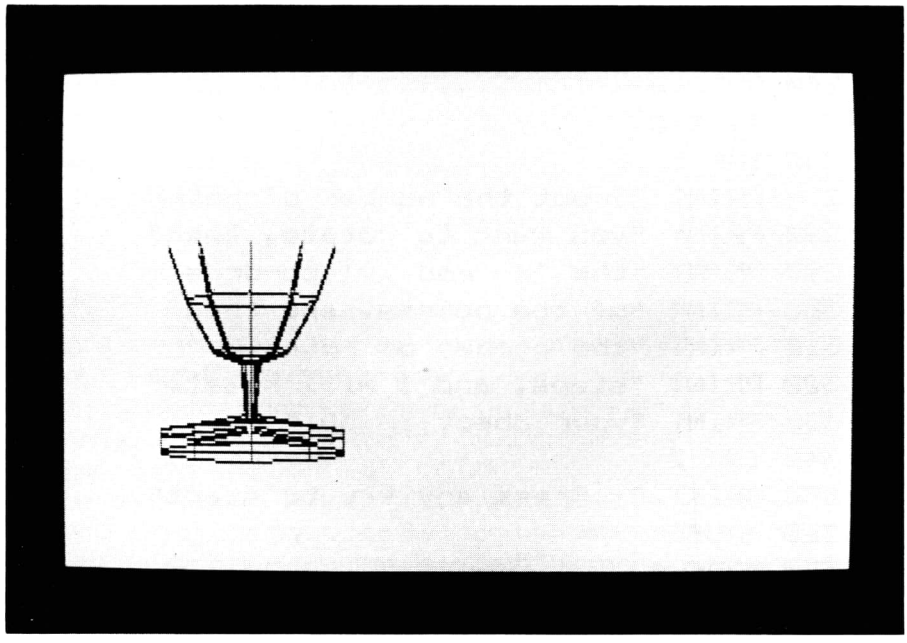
The program keeps a small copy of your original figure in the bottom left so that you don't have to remember it.

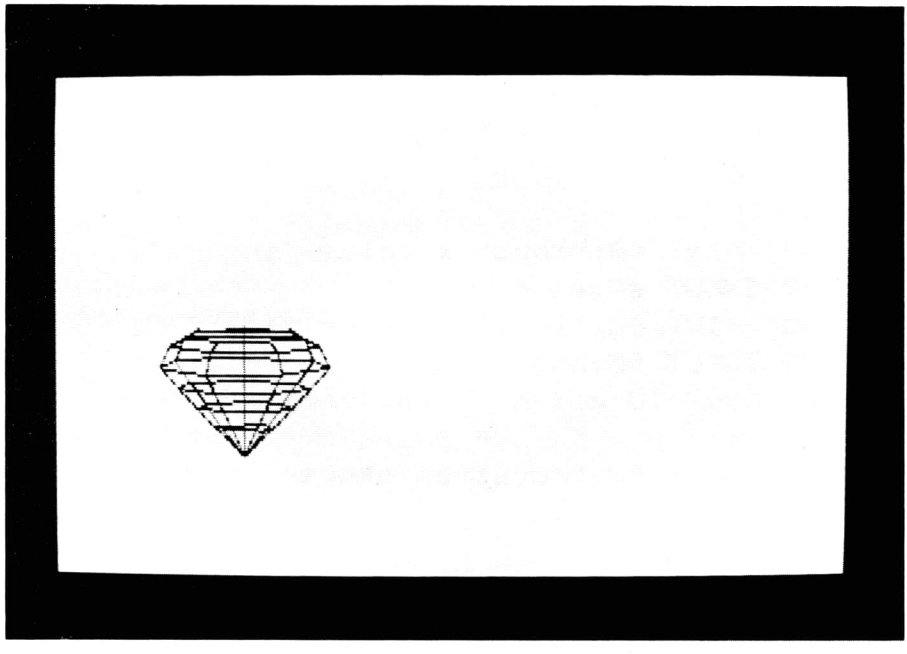
We can show you a couple of examples of what is possible.

The first screen shot shows a "wine glass" and the second shows you how to create a jewel with many facets.

You can have lots of fun with this program.

```
100 REMark ************
110 REMark * Rotation *
120 REMark ************
130 REMark
140 MODE 8
150 PAPER 0
160 CLS
170 INK 6
```





```
180 CSIZE 3,1
190 PRINT ,"Rotation"
200 OVER -1
210 AT 0,0
```

```
220 INK 1
230 PRINT ,"________"
240 OVER 0
250 AT 1,0
260 INK 5
270 PRINT "Input the number of points"
280 PRINT "you want to rotate, then"
290 PRINT "the 'x' and 'y' co-ords"
300 PRINT "of the points, and then"
310 PRINT "the number of rotation"
320 PRINT "steps, and I will rotate"
330 PRINT "your obect in 3D."
340 INK 2
350 PRINT "  Press any key to start"
360 PAUSE
370 MODE 4
380 PAPER 0
390 CLS
400 SCALE 500,0,0
410 INPUT #0 , "how many points : ";p
420 DIM a(p),b(p)
430 POINT 0,0
440 FOR n = 1 TO p
450 CLS #0
460 PRINT #0;"input x value ";n;"  ";
470 INPUT #0,a(n)
480 PRINT #0;"input y value ";n;"  ";
490 INPUT #0,b(n)
500 LINE TO a ( n ) , b ( n )
510 NEXT n
520 INPUT #0,"rotation steps ? ";s
530 CLS #0 : CLS
540 :
550 FOR n = 1 TO p - 1
560 POINT 300,120+b(n)+(a(n)/10)
570 :
580 FOR q = 360/s TO 360 STEP 360/s
590 :
600 Z1=SIN(RAD(q)):Z2=COS(RAD(q))
610 z=a(n):z3=z/10:z4=300+a(n+1)*Z1
```

```
620 LINE TO 300+z*Z1,120+b(n)+z3*Z2
630 LINE TO z4,120+b(n+1)+z3*Z2
640 LINE TO 300+z*Z1,120+b(n)+z3*Z2
650 :
660 NEXT q
670 NEXT n
```

The Drawing Program

We could say a lot about what to use this program for, but all you need is how to control it. You will soon think up loads of ideas.

The program has facilities for drawing circles, rectangles and lines. You fix a point using the F key for the centre of the circle, an edge of the rectangle or the start of a line.

Then you move away from the point you have fixed using the space bar. The program asks you how big will be the steps you want to move. You might say 6. Then you use the control keys actually to do the move and every one you hit will move you 6 pixels in the direction you want to go. The move will be blank, i.e. you won't actually draw anything. (To make a line appear, use the cursor keys without using the space bar.)

You select circle, rectangle or line with F2, F3 or F4 and the figure is drawn. Then you are asked if you are happy with your figure. (Reply y/n.) If you are not, the figure is erased.

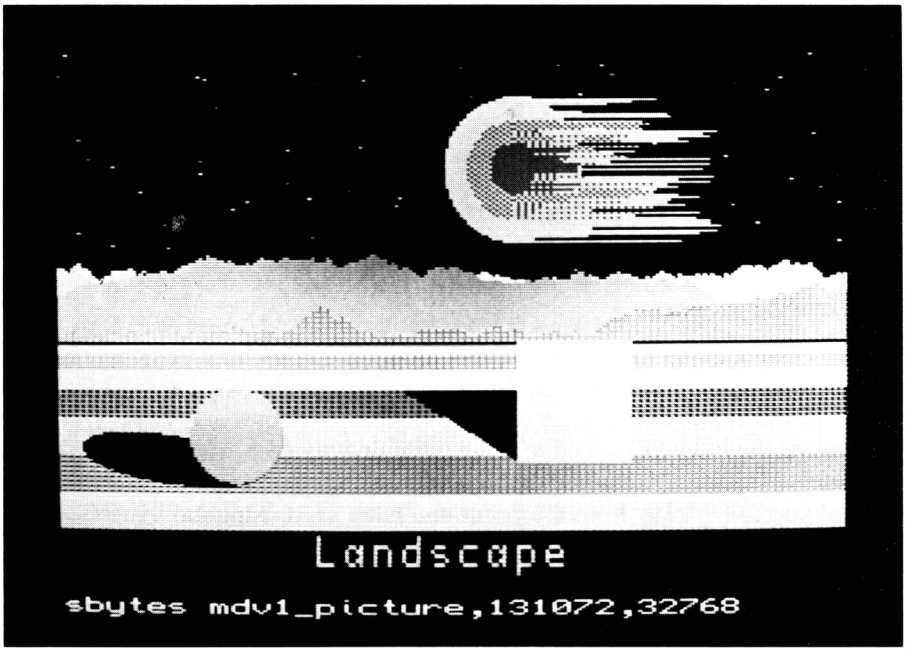
To change the ink colour, you use F1 and enter the ink colour (0-255).

After fixing a point and then moving elsewhere, F4 gives you the option to connect back to the fixed point with a straight or curved line. When it says "BEND?" you enter the curvature (in the range 0 to + or - 3).

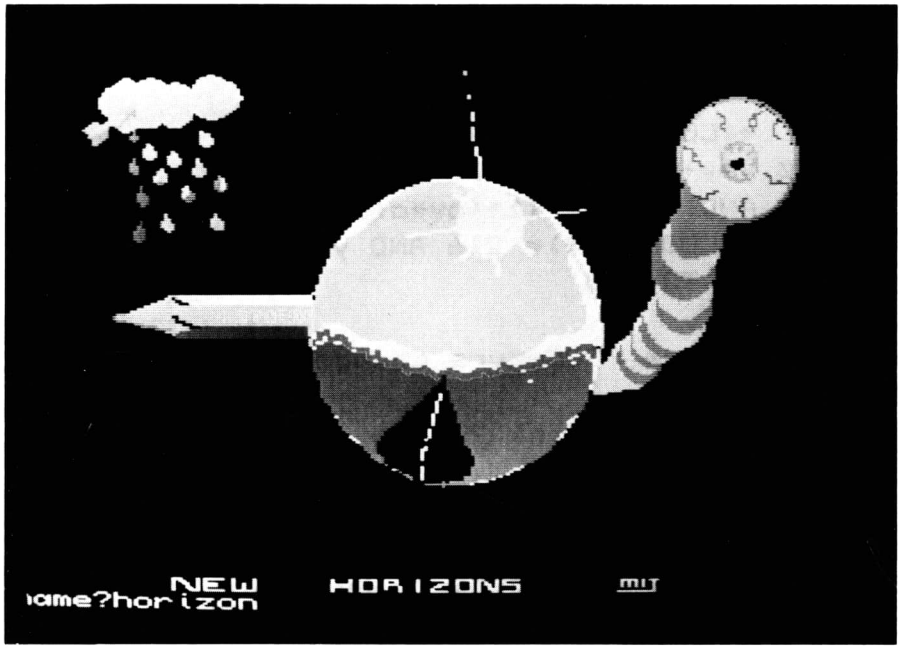
Now you are ready to go.

We can show you some of our efforts. The first is a nice landscape and the second is a surrealist painting by Marc Diprose.

```
100 REMark ***********
110 REMark * Drawing *
120 REMark ***********
130 REMark
140 FILL 0
150 MODE 8
160 SCALE 256,0,0
170 INK 7
180 PAPER 1
190 INPUT #0,"Paper ? ";p
200 INPUT #0,"Ink ? ";ic
210 WINDOW #1,512,256,0,0
```




```
220 PAPER p
230 CLS
240 PAPER #0,0
250 CLS #0
260 INK ic
270 WINDOW #0,512,21,0,235
280 x = 190
290 y = 127
300 fix = 0
310 ov = 0
320 ff = 0
330 ofd = 0
340 INK 7
350 OVER -1
360 CIRCLE x,y,4
370 OVER 0
380 INK ic
390 OVER 0-ov
400 POINT x,y
410 POINT x,y
420 OVER 0
430 k$ = INKEY$
```




```
440 IF fix = 0 AND k$="f" THEN
450   fix = 1
460   fy = y
470   fx = x
480 END IF
490 IF k$="" THEN GO TO 430
500 IF CODE(k$)=232 THEN changeink
510 IF k$ =" " THEN
520   ov = 1 - ov
530   IF ov THEN
540     INPUT #0,"How far ? ";ofd
550     CLS #0
560   END IF
570 END IF
580 IF fix = 0 THEN GO TO 620
590 IF CODE(k$)=236 THEN round
600 IF CODE(k$)=240 THEN square
610 IF CODE(k$)=244 THEN aline
620 INK 7
630 OVER -1
640 CIRCLE x,y,4
650 OVER 0
```

```
660 INK ic
670 x=x+(1+ov*(ofd))*(CODE(k$)=200
    AND x<380) - (1+(ov*ofd))
    *(CODE(k$)=192AND x>0)
680 y=y+(1+(ov*ofd))*(CODE(k$)=208
    AND y<255)-(1+(ov*ofd))
    *(CODE (k$)= 216 AND y>0)
690 GO TO 340
700 :
710 DEFine PROCedure changeink
720 CLS #0
730 INPUT #0;"ink ? ";ic
740 INK ic
750 END DEFine
760 :
770 DEFine PROCedure round
780 size = SQRT((fx-x)^2+(fy-y)^2)
790 CLS #0
800 INPUT #0;"filled ? ";f$
810 IF f$<>"y"AND f$<>"n"THEN GO TO 790
820 CLS #0
830 INK 7:OVER -1:CIRCLE x,y,4: OVER 0
840 IF f$="n" THEN GO TO 870
850 FILL 1
860 ff = 1
870 INK ic
880 OVER -1
890 CIRCLE fx,fy,size
900 FILL 0
910 CLS #0
920 INPUT #0,"   OK ? ";f$
930 IF f$<>"y"AND f$<>"n"THEN GO TO 910
940 IF f$="y" THEN OVER 0
950 IF f$="n" THEN OVER -1
960 FILL ff
970 CIRCLE fx,fy,size
980 FILL 0
990 OVER 0
1000 fix =0:ff= 0
1010 GO TO 340
```

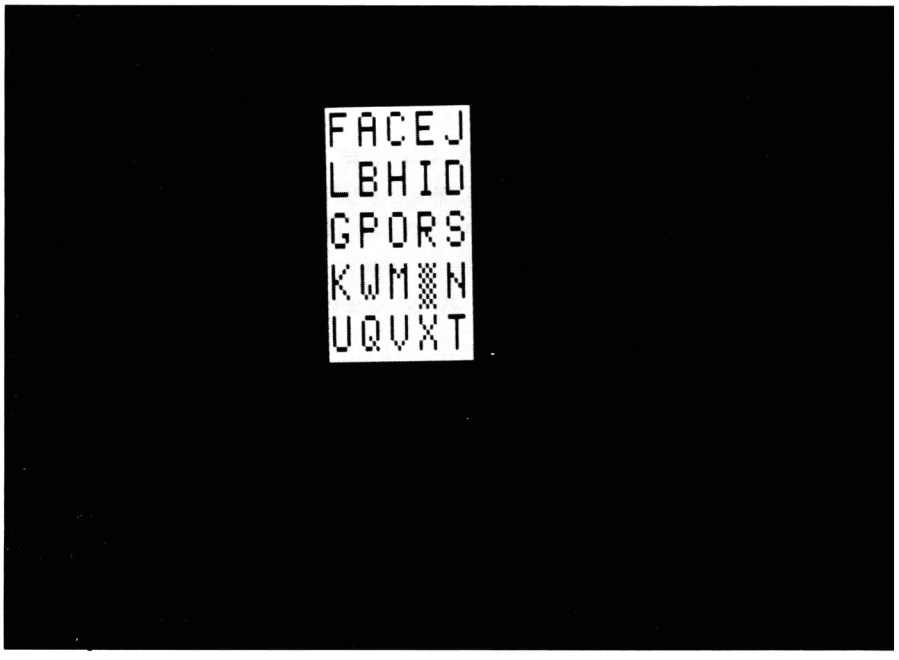
```
1020 END DEFine
1030 :
1040 DEFine PROCedure square
1050 INK 7
1060 OVER -1
1070 CIRCLE x,y,4
1080 OVER 0
1090 OVER -1
1100 FILL 1
1110 INK ic
1120 LINE x,y TO x,fy
1130 LINE fx,y TO fx,fy
1140 FILL 0
1150 CLS #0 : INPUT #0; "ok ? ";o$
1160 IF o$ = "" THEN GO TO 1150
1170 IF o$="n" THEN GO TO 1190
1180 OVER 0
1190 FILL 1
1200 LINE x,y TO x,fy
1210 LINE fx,y TO fx,fy
1220 FILL 0
1230 OVER 0
1240 fix = 0
1250 GO TO 340
1260 END DEFine
1270 :
1280 DEFine PROCedure aline
1290 INK 7
1300 OVER -1
1310 CIRCLE x,y,4
1320 OVER 0
1330 INK ic
1340 INPUT #0;"bend ? ";b
1350 OVER 1
1360 IF b = 0 THEN
1370   LINE fx,fy TO x,y
1380   GO TO 1410
1390 END IF
1400 ARC fx,fy TO x,y,b
1410 CLS #0
```

```
1420 INPUT #0;"  OK ? ";o$
1430 IF o$ = "" THEN GO TO 1410
1440 IF o$<>"y"AND o$<>"n"THEN GO TO 1410
1450 IF o$ = "n" THEN
1460   OVER -1
1470   GO TO 1510
1480 END IF
1490 OVER 0
1500 fix = 0
1510 IF b = 0 THEN
1520   LINE fx,fy TO x,y
1530   GO TO 1560
1540 END IF
1550 ARC fx,fy TO x,y,b
1560 OVER 0
1570 GO TO 340
1580 END DEFine
```

Chapter 4

GAMES OF SKILL

Wordslide



For this game, you have to rearrange the letters in alphabetical order using the cursor control keys. The difficulty level controls the number of shuffles that the letter board goes through before you can start your work.

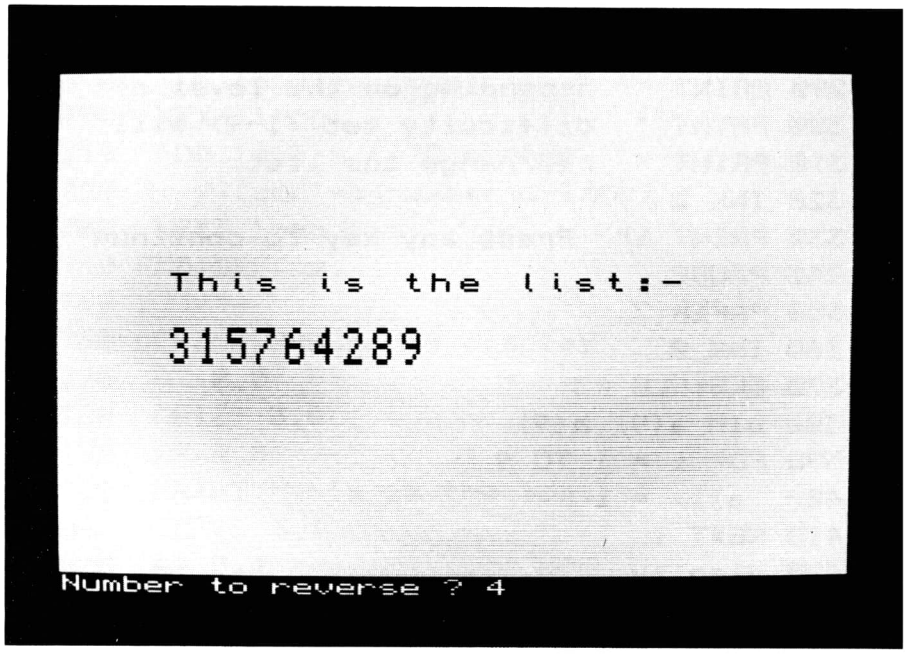
```
100 REMark **************
110 REMark * Wordslide *
120 REMark **************
130 REMark
140 MODE 8
150 BORDER 0
160 PAPER 0
170 CLS
180 CSIZE 3,1
190 INK 6
200 PRINT ,"Wordslide"
210 OVER -1
```

```
220 AT 0,0
230 INK 1
240 PRINT ,"_________"
250 OVER 0
260 INK 5
270 AT 2,0
280 PRINT "     I will shuffle up"
290 PRINT "  the puzzle. 'All' you"
300 PRINT "  have to do is to use "
310 PRINT "  the arrow keys to get"
320 PRINT "  it back in order."
330 INK 2
340 PRINT \"  Press any key to start"
350 PAUSE
360 :
370 DIM a$(5,5)
380 DIM B$(5,5)
390 RESTORE
400 FOR n = 1 TO 5
410   READ a$(n)
420   B$(n) = a$(n)
430 NEXT n
440 DATA "ABCDE" , "FGHIJ" , "KLMNO"
450 DATA "PQRST" , "UVWX" & CHR$(0)
460 :
470 MODE 8
480 CLS
490 DISPLAY
500 X = 5 : Y = 5
510 PAPER #0,0
520 CLS #0
530 INPUT #0;"Level (1-99)? ";le$
540 IF le$ = "" THEN GO TO 520
550 IF le$ <> INT(le$) THEN GO TO 520
560 IF le$<1 OR le$>99 THEN GO TO 520
570 CLS #0
580 :
590 FOR L = 1 TO le$ + 10
600   SLIDE RND(1 TO 4)
610 NEXT L
```

```
620 :
630 DISPLAY
640 K = CODE(INKEY$)
650 IF K = 208 THEN SLIDE 1
660 IF K = 216 THEN SLIDE 2
670 IF K = 200 THEN SLIDE 3
680 IF K = 192 THEN SLIDE 4
690 CO = 0
700 FOR n = 1 TO 5
710   IF a$(n) = B$(n) THEN CO=CO+1
720 NEXT n
730 IF CO = 5 THEN
740   DISPLAY
750   glory
760 END IF
770 DISPLAY
780 GO TO 640
790 :
800 DEFine PROCedure DISPLAY
810  INK 0
820  PAPER 6
830 CSIZE 3,1
840  FOR n = 1 TO 5
850    AT n,10
860    PRINT a$(n)
870  NEXT n
880 END DEFine
890 :
900 DEFine PROCedure  SLIDE (A)
910  BEEP 100,A*2
920  IF A = 1 AND Y > 1 THEN
930   a$(Y,X) = a$(Y-1,X)
940   Y = Y - 1
950  END IF
960 :
970  IF A = 2 AND Y < 5 THEN
980    a$(Y,X) = a$(Y+1,X)
990    Y = Y + 1
1000  END IF
1010 :
```

```
1020  IF A = 3 AND X < 5 THEN
1030    a$(Y,X) = a$(Y,X+1)
1040    X = X + 1
1050  END IF
1060 :
1070  IF A = 4 AND X > 1 THEN
1080    a$(Y,X) = a$(Y,X-1)
1090    X = X - 1
1100  END IF
1110 :
1120  a$(Y,X) = CHR$(0)
1130 END DEFine
1140 :
1150 DEFine PROCedure glory
1160 FOR n = 99 TO 0 STEP -5
1170   BORDER n,RND(255)
1180   BEEP 2000,n,n+20,45,45,15,n,10
1190  NEXT n
1200 STOP
1210 END DEFine
```

Reverse



You have to sort a string of nine digits into ascending numerical order. You can reverse any number of digits from 1 to 9 from the left-hand side. The machine will ask you how many you want.

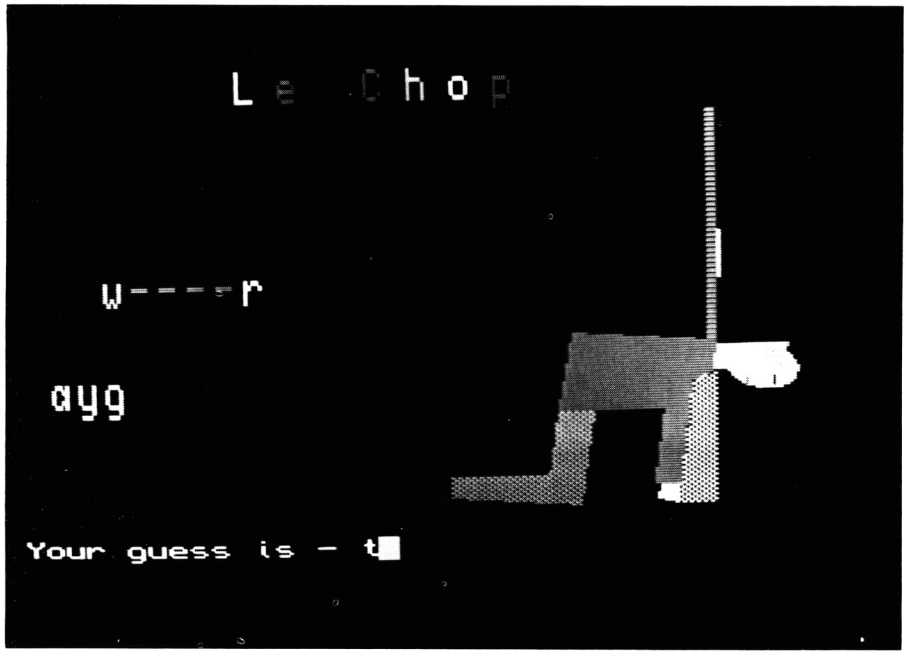
```
100 REMark ***********
110 REMark * Reverse *
120 REMark ***********
130 REMark
140 MODE 8
150 PAPER 0
160 INK 6
170 CLS
180 CSIZE 3,1
190 PRINT ,"  Reverse"
200 OVER -1
210 AT 0,0
220 INK 1
```

```
230 PRINT ,"  _______"
240 OVER 0
250 INK 5
260 AT 2,0
270 PRINT "  The computer will display"
280 PRINT "  a series of numbers. Then"
290 PRINT "  depending on the level of"
300 PRINT "  difficulty set (1-9) will"
310 PRINT "  rearrange the list. "
320 INK 2
330 PRINT \"  Press any key to continue"
340 PAUSE
350 PAPER 7
360 INK 0
370 CLS:CLS #0
380 DIM a(9),b(9)
390 FOR i = 1 TO 9
400  a(i) = i
410 NEXT i
420 display
430 CSIZE #0,2,0
440 PRINT #0,"what difficulty level"
450 PRINT #0,"( 1-9 Easy to Hard )"
460 AT #0,2,0
470 PRINT #0,"                "
480 AT #0,2,0
490 INPUT #0,r$
500 IF r$ <"1" OR r$ >"9" OR r$="" THEN
510   BEEP 1000,10
520   GO TO 460
530 END IF
540 FOR i = 1 TO r$*3
550  reverse RND (1 TO 8)
560 NEXT i
570 :
580 display
590 input_num
600 reverse number
610 check_list
620 display
```

```
630 IF done_it = 1 THEN finish
640 :
650 GO TO 580
660 :
670 DEFine PROCedure reverse (number)
680  FOR loop = 1 TO number
690    b(loop)=a(loop)
700  NEXT loop
710  FOR loop = 1 TO number
720    a(loop)=b(number+1-loop)
730  NEXT loop
740 END DEFine
750 :
760 DEFine PROCedure display
770  AT 0,0
780  CSIZE 3,0
790  AT 8,4
800  PRINT "This is the list:-"
810  CSIZE 3,1
820  AT 5,4
830  PRINT a(1 TO);
840 END DEFine
850 :
860 DEFine PROCedure input_num
870  CLS #0
880  INPUT #0;"Number to reverse ? ";r$
890  IF r$<"1" OR r$>"9" OR r$="" THEN
900    BEEP 1000,10
910    input_num
920  END IF
930  number = r$
940 END DEFine
950 :
960 DEFine PROCedure check_list
970  done_it = 0
980  FOR loop = 1 TO 9
990    IF a(loop)<> loop THEN GO TO 1020
1000  NEXT loop
1010  done_it = 1
1020 END DEFine
```

```
1030 :
1040 DEFine PROCedure finish
1050  BEEP 20000,1,100,1000,1
1060  AT 5,2
1070  PRINT "You done it!!!"
1080  INPUT "Another go ?";r$
1090  IF r$=="y" THEN
1100    PAUSE 50
1110    RUN
1120  END IF
1130  IF r$=="n" THEN STOP
1140  finish
1150 END DEFine
```

Le Chop



This is a neat variation on a game that you should recognise. The program asks you whether you are playing against it or another player. Either you can input a word for your opponent, or the machine will supply you with one. You then have to guess the letters that make up the word. If you want to change the words produced by the program, alter the DATA statements in lines 2070 to 2120. Alternatively, you may wish to change the number of possible words, in which case add some DATA statements but do not forget to change line 610 as well. If you look at it, you will see that the random number range is 1 to 10. If you have added four words, then of course this will need to be changed to 1 to 14.

```
100 REMark ***********
110 REMark * Le Chop *
120 REMark ***********
130 REMark
140 RESTORE
150 MODE 8
160 PAPER 0
```

```
170 CLS
180 INK 6
190 CSIZE 3,1
200 PRINT ,"  Le-Chop"
210 OVER -1
220 INK 1
230 AT 0,0
240 PRINT ,"  _______"
250 OVER 0
260 AT 2,0
270 INK 5
280 PRINT "  Thos game is based on "
290 PRINT "  HANGMAN , but has a new"
300 PRINT "  twist in the tail."\\
310 INK 2
320 PRINT "  Press any key to play"
330 PAUSE
340 CLS
350 found$ = ""
360 guessed$ = ""
370 wrong$ = ""
380 SCALE 100,0,0
390 WINDOW #0 ;480 ,50 ,15 ,205
400 OPEN #3 ,con
410 OPEN #4 ,con
420 WINDOW #3 ,10 ,100 ,410 ,30
430 WINDOW #4 ,100 ,65 ,411 ,130
440 CSIZE 2,1
450 FOR n = 1 TO 8
460   AT 0,5 + n*2
470   INK RND (1 TO 7)
480   PRINT " Le Chop "(n)
490 NEXT n
500 CLS #0
510 PRINT #0,"Do you want to pick";
520 INPUT #0," the word (y/n) ?";g$
530 IF g$="" THEN GO TO 500
540 IF g$ == "y" THEN GO TO 560
550 IF g$ == "n" THEN GO TO 600
560 CLS #0
```

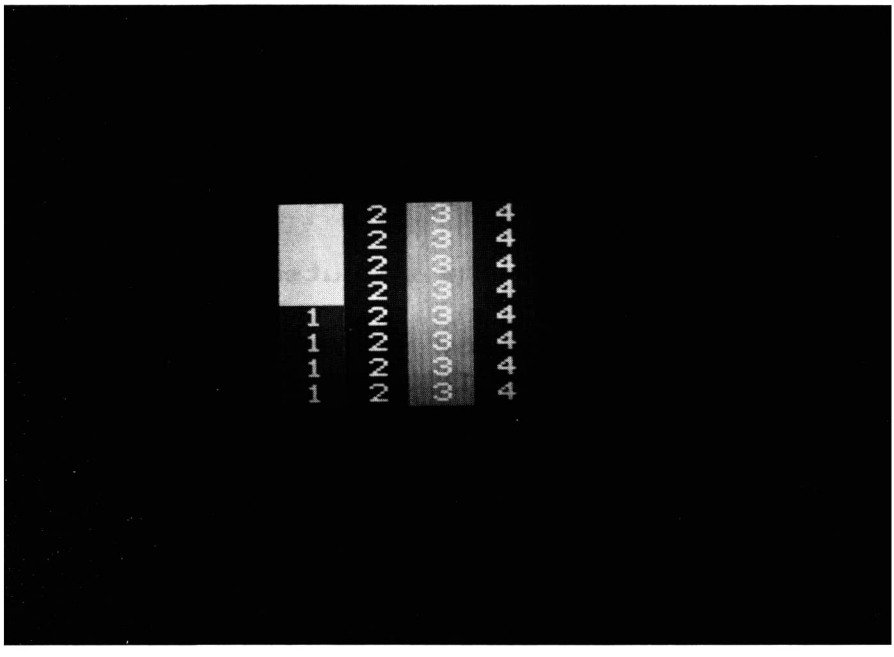
```
570 INPUT #0;"what is your word? ";word$
580 IF LEN (word$) >10 THEN GO TO 560
590 GO TO 640
600 RESTORE
610 FOR n = 1 TO RND (1 TO 10)
620   READ word$
630 NEXT n
640 CSIZE 3,1
650 right$ = FILL$("#",LEN(word$))
660 AT 4,2
670 PRINT FILL$("-",LEN(word$))
680 GO SUB 1190
690 goes = 0
700 CLS #0
710 INPUT #0;"Your guess is - ";g$
720 IF LEN (g$) <>1 THEN GO TO 700
730 IF g$ INSTR guessed$ <> 0 THEN
740   PRINT #0;"You have tried that!"
750   PAUSE 60
760   GO TO 700
770 END IF
780 guessed$ = guessed$ & g$
790 IF g$ INSTR word$ = 0 THEN
800   wrong$ = wrong$ & g$
810   AT 6,0
820   PRINT wrong$
830   BEEP 4000,40,150,3,45,0,6,0
840   PAUSE 20
850   BEEP 6000,240,180,3,45,0,6,0
860   FOR n = 1 TO 8
870     SCROLL #3,-1
880   NEXT n
890   goes = goes +1
900   IF goes = 9 THEN GO TO 1830
910 END IF
920 FOR n = 1 TO LEN (word$)
930   IF word$(n) = g$ THEN
940     right$(n) = g$
950     AT 4,1+n
960     PRINT g$
```

```
970   END IF
980 NEXT n
990 IF right$ = word$ THEN
1000   INK 2
1010   LINE 146,35 TO 147,37
1020   FOR a = 60 TO 30 STEP -10
1030     FOR n = a TO a-20 STEP -4
1040       BEEP 2500,n,n,6,44,5,1,0
1050       PAUSE 4
1060     NEXT n
1070   NEXT a
1080   GO TO 2020
1090 END IF
1100 GO TO 700
1110 FOR n = 1 TO 9
1120   PAUSE
1130   FOR a = 1 TO 8
1140     SCROLL #3,-1
1150   NEXT a
1160 NEXT n
1170 GO TO 1830
1180 STOP
1190 FILL 1
1200 INK 7 ,0 ,3
1210 LINE 140,37 TO 135,35
1220 LINE TO 132,11 TO 140,11
1230 LINE TO 140,37
1240 FILL 0
1250 FILL 1
1260 INK 4 ,2 ,1
1270 LINE 140,39 TO 140,90
1280 LINE 138,41 TO 138,90
1290 FILL 0
1300 INK 3 ,7 ,3
1310 FILL 1
1320 LINE 133,15 TO 132,11
1330 LINE TO 128,12 TO 129,15
1340 FILL 0
1350 FILL 1
1360 INK 2
```

```
1370 LINE 140,43 TO 110,45
1380 LINE TO 107,30 TO 130,37
1390 LINE TO 127,15 TO 133,15
1400 LINE TO 139,32 TO 141,43
1410 FILL 0
1420 FILL 1
1430 INK 7 ,0 ,3
1440 LINE 140,37 TO 135,35
1450 LINE TO 132,11 TO 140,11
1460 LINE TO 140,37
1470 FILL 0
1480 PAPER #3,0
1490 CLS #3
1500 PAPER #4,0
1510 CLS #4
1520 FILL 1
1530 INK 2 ,4 ,3
1540 LINE 107,30 TO 105,17
1550 LINE TO 85,17 TO 84,13
1560 LINE TO 112,11 TO 115,30
1570 FILL 0
1580 INK 2 ,0 ,3
1590 FILL 1
1600 LINE TO 141,53
1610 LINE 85,17 TO 82,18
1620 LINE TO 80,11 TO 84,11
1630 LINE TO 85,17
1640 FILL 0
1650 FILL 1
1660 INK 3 ,7 ,3
1670 LINE 140,42 TO 139,37
1680 CIRCLE 150,38,8,.54,1.4
1690 FILL 0
1700 INK 2
1710 LINE 146,34 TO 146,36
1720 INK 0
1730 LINE 152,34 TO 152,36
1740 FILL 1
1750 INK 3 ,1 ,2
1760 LINE 150,40 TO 150,42
```

```
1770 LINE TO 156,42 TO 154,37
1780 LINE TO 150,40
1790 FILL 0
1800 INK 7
1810 LINE 140,53 TO 140,43
1820 RETurn
1830 INK 2
1840 LINE 146,35 TO 145,37
1850 FOR n = 1 TO 28
1860   SCROLL #3,3
1870   BEEP 2000,n*2,n*2,2,5,15,0,0
1880 NEXT n
1890 FOR n = 1 TO 15
1900   SCROLL #4,3
1910   PAN #4,2
1920 NEXT n
1930 FOR n = 90 TO 380 STEP 10
1940   SCROLL #4,-2 * SIN (RAD(n))
1950   PAN #4,4
1960 NEXT n
1970 RECOL 0,0,0,0,0,0,0,0
1980 PRINT "The word was ";
1990 UNDER 1
2000 PRINT word$
2010 UNDER 0
2020 CLS #0
2030 INPUT #0;"Another go? ";ag$
2040 IF ag$ == "n" THEN STOP
2050 IF ag$ == "y" THEN GO TO 160
2060 GO TO 2020
2070 DATA "nobleman","sinclair"
2080 DATA "newtech","addison"
2090 DATA "wesley","robin"
2100 DATA "dunk","pansy"
2110 DATA "summer","winter"
2120 DATA "autumn","spring"
```

Flasher



With the Flasher, you have to repeat a sequence of flashing colours. You carry on following the Flasher until eventually he beats you into the ground. There are four possible flashing boxes numbered 1 to 4 and you have to repeat the sequence in terms of these numbers. Our record is a sequence of 17 (without using pen and paper). If you are a genius, the machine will take you up to 32767 and the best of luck!

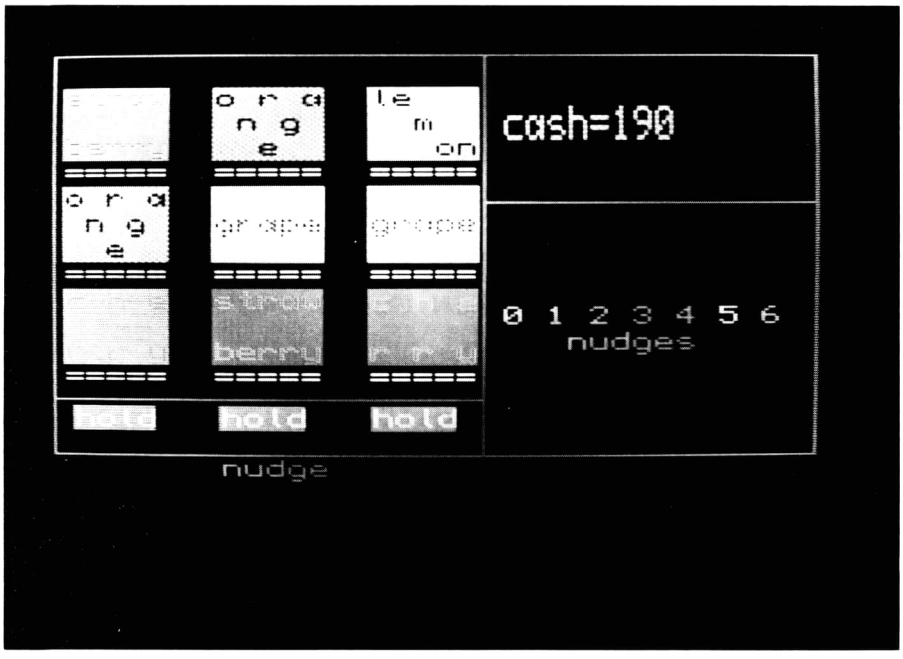
```
100 REMark ***********
110 REMark * Flasher *
120 REMark ***********
130 REMark
140 record = 17
150 MODE 8
160 PAPER 0
170 CLS
180 CSIZE 3,1
190 INK 6
200 PRINT ," Flasher"
```

```
210 OVER -1
220 AT 0,0
230 INK 2
240 PRINT ," _______"
250 OVER 0
260 INK 5
270 AT 3,2
280 PRINT "Press the number key"
290 AT 4,2
300 PRINT "of the  row  that is"
310 AT 5,2
320 PRINT "lit by the computer."
330 INK 2
340 PRINT \"Press any key to continue"
350 IF INKEY$ = "" THEN GO TO 350
360 CLS
370 INK 7
380 sc = 2
390 CSIZE 0,0
400 FOR n = 1 TO 4
410   PAPER  n
420   FOR 1 = 1 TO 8
430     AT 5+1,8+n*3
440     PRINT " ";n;" "
450   NEXT 1
460 NEXT n
470 t$ = "12"
480 sequence
490 guess
500 sc = sc + 1
510 FOR n = 20 TO 2 STEP -4
520   BEEP 2000,n,n*3,34,45,15,0,0
530   PAUSE 3
540 NEXT n
550 t$ = t$ & RND(1 TO 4)
560 GO TO 480
570 DEFine PROCedure sequence
580 FOR n = 1 TO LEN(t$)
590 x = 8+3*(t$(n))
600 PAPER 7
```

```
610 FOR l = 1 TO 8
620   AT 5+l,x
630   PRINT "    "
640 NEXT l
650 BEEP 5000,x*5
660 PAUSE 3
670 PAPER t$(n)
680 FOR l = 1 TO 8
690   AT 5+l,x
700   PRINT " ";t$(n);" "
710 NEXT l
720 NEXT n
730 END DEFine
740 :
750 DEFine PROCedure guess
760 FOR n = 1 TO LEN(t$)
770   g$ = INKEY$(-1)
780   IF g$<"1" OR g$>"4" THEN GO TO 770
790   PAPER 7
800   FOR l = 1 TO 8
810     AT 5+l,8+3*g$
820     PRINT "    "
830   NEXT l
840   IF g$ <> t$(n) THEN duffout
850   BEEP 5000,5*(8+3*(g$))
860   PAUSE 3
870   PAPER t$(n)
880   FOR l = 1 TO 8
890     AT 5+l,8+3*g$
900     PRINT " ";g$;" "
910   NEXT l
920 NEXT n
930 END DEFine
940 :
950 DEFine PROCedure duffout
960  BEEP 20000,100,140,4,43,15,13,8
970  PRINT #0;"You survived ";
980  PRINT #0,LEN(t$);" rounds."
990  IF sc > record THEN record = sc
1000  PRINT #0;"The record is ";record
```

```
1010  INPUT #0;"another go?" ;ag$
1020  IF ag$ == "y" THEN
1030    CLS #0
1040    GO TO 380
1050  END IF
1060  IF ag$ == "n" THEN STOP
1070  CLS #0
1080  GO TO 1010
1090 END DEFine
```

Fruit Machine



You have a three-window fruit machine with "random hold after win" and six nudges (upwards). You "pull the arm" with the space bar. If nudges show, you must use the space bar to pick up as many nudges as possible. Then you nudge the reels with 1, 2 or 3.

```
100 REMark ******************
110 REMark * FRUIT MACHINE *
120 REMark ******************
130 REMark
140 MODE 8
150 CSIZE 3,1
160 PAPER #1,0
170 CLS
180 INK 2
190 PRINT "         Wild Q-L"
200 INK 3
210 PRINT "     Fruit  Machine"
```

```
220 INK 4
230 PRINT
240 PRINT "Press space to start reels"
250 INK 5
260 PRINT "Use numbers"
270 PRINT "<1> <2> and <3>"
280 PRINT "For hold and nudge"
290 INK 6
300 PRINT "Press any key to get"
310 PRINT "nudge number"
320 INK 1
330 PRINT"Press any key to start"
340 IF INKEY$="" THEN GO TO 340
350 RESTORE 1770
360 money=200
370 DIM a(30),b(30),c(30)
380 FOR i=1 TO 30
390   READ a(i)
400 NEXT i
410 FOR i=1 TO 30
420   READ b(i)
430 NEXT i
440 FOR i=1 TO 30
450   READ c(i)
460 NEXT i
470 MODE 8
480 PAPER #2,0
490 PAPER #1,0
500 PAPER #0,0
510 CLS #2
520 CLS #1
530 CLS #0
540 OPEN #3,con
550 WINDOW #3,63,120,38,30
560 OPEN #4,con
570 WINDOW #4,63,120,124,30
580 OPEN #5,con
590 WINDOW #5,63,120,210,30
600 FOR i=3 TO 5
610   PAPER #i,0
```

```
620    INK #i,2
630    CLS #i
640 NEXT i
650 cancel
660 INK 1
670 AT 10,21
680 PRINT "0 1 2 3 4 5 6"
690 AT 11,21
700 PRINT "   nudges"
710 INK 2
720 LINE 0,20 TO 90,20
730 LINE TO 90,100 TO 0,100
740 LINE TO 0,20 TO 0,31
750 LINE TO 90,31 TO 90,20
760 LINE TO 160,20 TO 160,100
770 LINE TO 90,100 TO 90,70
780 LINE TO 160,70
790 a1=1
800 b1=1
810 c1=1
820 t3=RND(10 TO 15)
830 t2=RND(5 TO t3)
840 t1=RND(1 TO t2)
850 FOR j=1 TO t3
860    IF h1=1 THEN GO TO 930
870    t1=t1-1
880    IF t1<0 THEN GO TO 930
890    a1=a1+1
900    IF a1=31 THEN a1=1
910    choosereel a(a1),3
920    BEEP 200,50
930    IF h2=1 THEN GO TO 1000
940    t2=t2-1
950    IF t2<0 THEN GO TO 1000
960    b1=b1+1
970    IF b1=31 THEN b1=1
980    choosereel b(b1),4
990    BEEP 200,100
1000    IF h3=1 THEN GO TO 1070
1010    t3=t3-1
```

```
1020    IF t3<0 THEN GO TO 1070
1030    c1=c1+1
1040    IF c1=31 THEN c1=1
1050    choosereel c(c1),5
1060    BEEP 200,150
1070    REMark
1080 NEXT j
1090 IF RND(1 TO 10)=10 THEN nudge
1100 checkwin
1110 IF win<>0 THEN displaywin
1120 money=money+win
1130 cash
1140 cancel
1150 IF RND(1 TO 10)>8 THEN hold
1160 AT 16,4
1170 INK 2,7,2
1180 PAPER 0
1190 PRINT "press any key to start"
1200 IF INKEY$="" THEN GO TO 1200
1210 AT 16,4
1220 PRINT "                          "
1230 BEEP 20000,30,40,1,1
1240 money=money-10
1250 IF money=0 THEN GO TO 2960
1260 cash
1270 GO TO 820
1280 :
1290 DEFine PROCedure orange (chan)
1300  PAPER #chan,2,6
1310  INK #chan,0
1320  PRINT #chan;"o r a"
1330  PRINT #chan;" n g "
1340  PRINT #chan;"  e  "
1350 END DEFine
1360 :
1370 DEFine PROCedure lemon (chan)
1380  PAPER #chan,6
1390  INK #chan,0
1400  PRINT #chan;"le   "
1410  PRINT #chan;"  m  "
```

```
1420  PRINT #chan;"   on"
1430 END DEFine
1440 :
1450 DEFine PROCedure bar (chan)
1460  PAPER #chan,1
1470  INK #chan,7
1480  PRINT #chan;"-----"
1490  PRINT #chan;" Q-L "
1500  PRINT #chan;"-----"
1510 END DEFine
1520 :
1530 DEFine PROCedure cherry (chan)
1540  PAPER #chan,4
1550  INK #chan,2
1560  PRINT #chan;"c h e"
1570  PRINT #chan;"     "
1580  PRINT #chan;"r r y"
1590 END DEFine
1600 :
1610 DEFine PROCedure strawberry (chan)
1620  PAPER #chan,2
1630  INK #chan,4
1640  PRINT #chan;"straw"
1650  PRINT #chan;"     "
1660  PRINT #chan;"berry"
1670 END DEFine
1680 :
1690 DEFine PROCedure grape (chan)
1700  PAPER #chan,3
1710  INK #chan,0,4
1720  PRINT #chan;"     "
1730  PRINT #chan;"grape"
1740  PRINT #chan;"     "
1750 END DEFine
1760 :
1770 DATA 3,3,4,2,3,5,4,2,1,5,4,1,3,5,2
1780 DATA 5,2,4,3,5,6,5,1,3,2,4,5,2,4,2
1790 DATA 3,4,5,1,1,3,4,3,1,3,4,2,4,6,1
1800 DATA 3,5,3,4,2,5,1,3,2,5,4,2,3,4,1
1810 DATA 3,2,4,1,1,4,5,2,5,3,6,3,2,1,3
```

```
1820 DATA 4,5,1,2,3,4,3,5,3,5,1,5,2,5,1
1830 :
1840 DEFine PROCedure choosereel (l,k)
1850  IF l=1 THEN cherry k
1860  IF l=2 THEN orange k
1870  IF l=3 THEN lemon  k
1880  IF l=4 THEN strawberry  k
1890  IF l=5 THEN grape  k
1900  IF l=6 THEN bar    k
1910  INK #k,7
1920  PAPER #k,0
1930  PRINT #k;"====="
1940 END DEFine
1950 :
1960 DEFine PROCedure hold
1970  PAPER 0
1980  FOR j=1 TO 30
1990    FOR i=0 TO 3
2000      AT 16,8
2010      INK 2,6,i
2020      PRINT "hold"
2030      BEEP 10,30-j+i
2040    NEXT i
2050  NEXT j
2060  a$=INKEY$
2070  PAPER 5
2080  INK 0
2090  IF a$="1" THEN
2100    h1=1
2110    AT 14,1
2120    PRINT"held"
2130  END IF
2140  IF a$="2" THEN
2150    h2=1
2160    AT 14,8
2170    PRINT "held"
2180  END IF
2190  IF a$="3" THEN
2200    h3=1
2210    AT 14,15
```

```
2220    PRINT "held"
2230  END IF
2240  IF a$="c" THEN cancel
2250  IF a$=" " THEN GO TO 2270
2260  GO TO 2060
2270 END DEFine
2280 :
2290 DEFine PROCedure cancel
2300  PAPER 1
2310  INK 6
2320  AT 14,1
2330  PRINT "hold"
2340  h1=0
2350  AT 14,8
2360  PRINT "hold"
2370  h2=0
2380  AT 14,15
2390  PRINT "hold"
2400  h3=0
2410  INK 0
2420  PAPER 0
2430  AT 16,8
2440  PRINT "     "
2450 END DEFine
2460 :
2470 DEFine PROCedure checkwin
2480  win=0
2490  aa=a(a1)
2500  bb=b(b1)
2510  cc=c(c1)
2520  IF aa=1 AND bb=1 AND cc=1 THEN
2530    win=10
2540    GO TO 2730
2550  END IF
2560  IF aa=1 THEN
2570    win=10
2580    GO TO 2730
2590  END IF
2600  IF aa=6 AND bb=6 AND cc=6 THEN
2610    win=1000
```

```
2620    GO TO 2730
2630  END IF
2640  IF aa=6 AND bb=6 THEN
2650    win=500
2660    GO TO 2730
2670  END IF
2680  IF aa=bb AND bb=cc AND aa=cc THEN
2690    win=20*aa
2700    GO TO 2730
2710  END IF
2720  IF aa=bb THEN win=10*aa
2730  IF win<>0 THEN BEEP 30000,30,90,1,2,3
2740 END DEFine
2750 :
2760 DEFine PROCedure cash
2770  CSIZE 2,1
2780  AT 1,21
2790  INK 1,5
2800  PAPER 0
2810  PRINT"cash=";money;" "
2820  CSIZE 0,0
2830 END DEFine
2840 :
2850 DEFine PROCedure displaywin
2860  PAPER 0
2870  INK 2,7
2880  AT 5,21
2890  PRINT "you won ";win
2900  FOR j=1 TO 3000
2910  NEXT j
2920  AT 5,21
2930  PRINT "                "
2940 END DEFine
2950 :
2960 MODE 8
2970 PAPER 3
2980 INK 0
2990 CLS
3000 CSIZE 3,0
3010 AT 10,8
```

```
3020 PRINT "out of cash"
3030 STOP
3040 :
3050 DEFine PROCedure nudge
3060  AT 16,8
3070  PAPER 0
3080  INK 2
3090  PRINT "nudge"
3100  FOR j=0 TO 6
3110  AT 10,21+j*2
3120  BEEP 200,j*3+(j*4)/3,100,1,1
3130  INK 5
3140  PRINT j
3150  IF INKEY$<>"" THEN GO TO 3210
3160  AT 10,21+j*2
3170  INK 1
3180  PRINT j
3190  NEXT j
3200  GO TO 3100
3210  AT 16,8
3220  PRINT "      "
3230  nd=j
3240  IF nd=0 THEN GO TO 3510
3250  a$=INKEY$
3260  IF a$="" THEN GO TO 3250
3270  IF a$="1" THEN
3280    a1=a1+1
3290    IF a1=31 THEN a1=1
3300    choosereel a(a1),3
3310  END IF
3320  IF a$="2" THEN
3330    b1=b1+1
3340    IF b1=31 THEN b1=1
3350    choosereel b(b1),4
3360  END IF
3370  IF a$="3" THEN
3380    c1=c1+1
3390    IF c1=31 THEN c1=1
3400    choosereel c(c1),5
3410  END IF
```

```
3420  AT 10,21+nd*2
3430  INK 1
3440  PRINT nd
3450  nd=nd-1
3460  IF nd=0 THEN GO TO 3510
3470  AT 10,21+nd*2
3480  INK 5
3490  PRINT nd
3500  GO TO 3250
3510 END DEFine
```

Snails



Here we are using molluscs to teach you arithmetic. Two people play this and they each enter their name. The machine then gives you a series of arithmetic problems, and as you get one right, your snail moves forward. The winner is the one whose snail reaches the right-hand side first, that is, the one who answers most questions correctly.

```
100 REMark *********
110 REMark * Snail *
120 REMark *********
130 REMark
140 MODE 8
150 PAPER 0
160 CLS
170 CSIZE 3,1
180 INK 6
190 PRINT ,"  Snails"
200 AT 0,0
210 INK 1
```

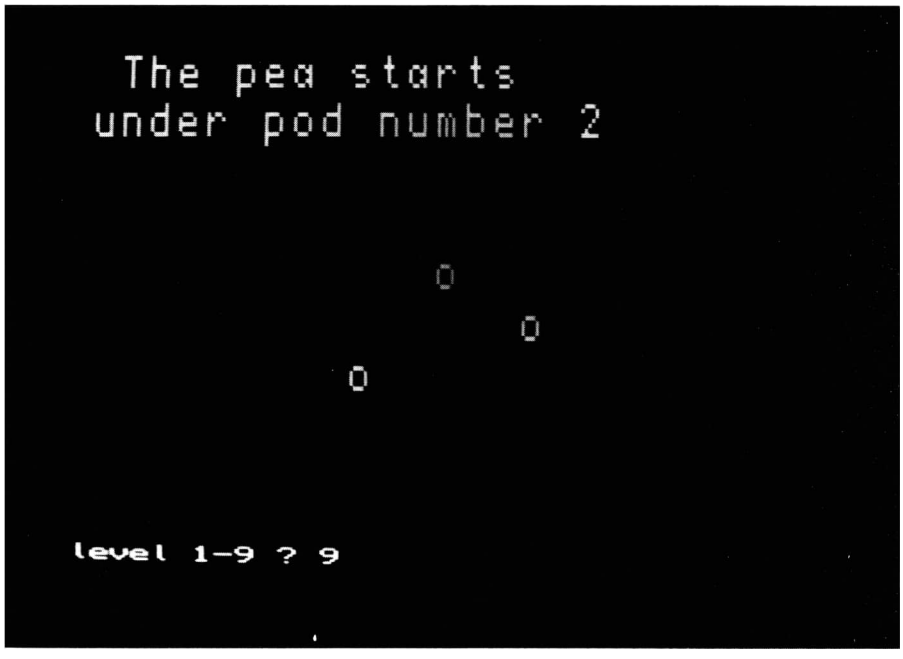
```
220 OVER -1
230 PRINT ,"  ______"
240 OVER 0
250 INK 5
260 AT 2,0
270 PRINT "  This is a race between"
280 PRINT "  two players. Points are"
290 PRINT "  gained by answering the"
300 PRINT "  questions correctly."
310 INK 2
320 PRINT \"  Press any key to start"
330 PAUSE
340 CLS
350 CSIZE 0,0
360 SCALE 256 ,0 ,0
370 DIM p$ (2,10)
380 DIM d (2)
390 d(1) = 1
400 d(2) = 1
410 OPEN #3 , con
420 OPEN #4 , con
430 WINDOW #3 ,512 ,100 ,0 ,5
440 WINDOW #4 ,512 ,105 ,0 ,105
450 snail 3
460 SCROLL - 110
470 snail 5
480 p = 1
490 PRINT #0;"What is your name ";
500 INPUT #0,"player 1 ? ";p$(1)
510 PRINT #0;"what is your name ";
520 INPUT #0,"player 2 ? ";p$(2)
530 CLS #0 : p = 1
540 a = INT (RND(1 TO 20))
550 b = INT (RND(1 TO 20))
560 f = INT (RND(1 TO 3))
570 display
580 answer
590 GO TO 540
600 :
610 DEFine PROCedure snail ( x )
```

```
620    FILL 1
630    INK 2
640    LINE 100,10 TO 10,10
650    ARC TO 100,30 ,-1
660    ARC TO 120,50 ,2
670    ARC TO 150,50 ,-PI
680    ARC TO 145,45 ,-PI
690    ARC TO 100,10 ,-2
700    LINE 120,50 TO 150,50
710    FILL 0
720    LINE 135,65 TO 140,90
730    INK 7
740    FILL 1
750    CIRCLE 141,90 ,5
760    FILL 0
770    INK 0
780    CIRCLE 142,91 ,2
790    ARC 145,45 TO 139,47 ,1
800    INK x : FILL 1
810    CIRCLE 65 , 65 , 40
820    FILL 0
830    FILL 1
840    LINE 95,40 TO 100,36
850    ARC TO 30,30 ,-1
860    LINE TO 32,40
870    FILL 0
880    INK 0
890    ARC 70,65 TO 60,65 ,-PI
900    ARC TO 75,65 ,-PI
910    ARC TO 55,65 ,-PI
920    ARC TO 80,65 ,-PI
930    ARC TO 45,65 ,-PI
940    ARC TO 90,65 ,-PI
950    ARC TO 35,65 ,-PI
960    ARC TO 91,45 ,-4
970    ARC TO 32,40 ,-1.2
980 END DEFine
990 :
1000 DEFine PROCedure display
1010    PRINT #0;p$(p);", What is - "
```

```
1020   PRINT #0; a;
1030   SELect ON f
1040   ON f = 1
1050     t = a + b
1060     PRINT #0; " + "; b
1070   ON f = 2
1080     t = a - b
1090     PRINT #0; " - "; b
1100   ON f = 3
1110     t = a * b
1120     PRINT #0; " x "; b
1130   END SELect
1140 END DEFine
1150 :
1160 DEFine PROCedure answer
1170   AT #0 ,3,0
1180 BEEP 100,10
1190   PRINT#0; "What is your guess? ";
1200   INPUT #0,an$
1210   IF an$ = "" THEN GO TO 1170
1220   an = an$
1230   IF an = t THEN
1240     FOR n = 1 TO 8
1250       PAN #(2+p) ,4
1260     NEXT n
1270     d(p) = d(p) + 1
1280     IF d(p) = 10 THEN finale
1290   END IF
1300   IF an = t THEN GO TO 1340
1310   BEEP 3000,50,70,45,45,0,0,0
1320   PAUSE 2
1330   BEEP 20000,200,240,45,45,0,0,0
1340   p = p - 1 : IF p = 0 THEN p = 2
1350   CLS #0
1360 END DEFine
1370 :
1380 DEFine PROCedure finale
1390  FOR a = 1 TO 5
1400   FOR n = 4 TO 1 STEP -1
1410     SCROLL #(2+p) ,-n
```

```
1420      tune
1430    NEXT n
1440    SCROLL #(p+2) ,1
1450    FOR n = 1 TO 3
1460      SCROLL #(2+p) ,3
1470      tune
1480    NEXT n
1490  NEXT a
1500  BEEP
1510  CLS #0
1520  PRINT #0;"Well done ";p$(p),
1530  PRINT #0;", you have finished",
1540  PRINT #0 ;ABS(d(1)-d(2));
1550  PRINT #0," moves ahead!"
1560  FOR n = 7 TO 1 STEP -1
1570    BEEP 5000,n*2,n*4,40,5,15,3,0
1580    PAUSE 2
1590  NEXT n
1600  INPUT #0 ;"another go?" ;g$
1610  IF g$=="n" THEN STOP
1620  IF g$=="y" THEN RUN
1630  GO TO 1600
1640 END DEFine
1650 :
1660 DEFine PROCedure tune
1670  BEEP 0,n*5,n*5,5,45,0,0,0
1680 END DEFine
```

Pennies



This is the old "three cups and a coin". The difficulty level controls the number of shuffles of the cups. As the difficulty level increases so does the number of shuffles, but so does the speed of the shuffling. You then have to tell the machine where the penny is by entering 1, 2 or 3.

```
100 REMark ***********
110 REMark * Pennies *
120 REMark ***********
130 REMark
140 MODE 8
150 PAPER 0
160 CLS
170 INK 6
180 CSIZE 3,1
190 PRINT ,"  Pennies"
200 OVER -1
210 AT 0,0
```

```
220 INK 1
230 PRINT ,"  _______"
240 OVER 0
250 INK 5
260 AT 2,0
270 PRINT "  I have three cups and"
280 PRINT "  under one of them is a"
290 PRINT "  pea. You must try to "
300 PRINT "  guess where it is after"
310 PRINT "  I move it around."
320 INK 2
330 PRINT \"  Press any key to start"
340 PAUSE
350 CLS
360 INK 4
370 AT 5,10 : PRINT "o"
380 AT 5,13 : PRINT "o"
390 AT 5,16 : PRINT "o"
400 DIM a(3)
410 pea = RND(1 TO 3)
420 a(pea) = 1
430 AT 0,0
440 PRINT "  The pea starts "
450 PRINT " under pod number ";pea
460 CLS #0
470 INPUT #0,"level 1-9 ? ";1$
480 IF 1$="" THEN GO TO 460
490 IF 1$<"1"OR 1$>"9" THEN GO TO 460
500 :
510 FOR i = 1 TO RND (1 TO 1$*3)
520   :
530   which = RND (1 TO 3)
540   :
550   SELect ON which
560     ON which = 1
570        move_1
580     ON which = 2
590        move_2
600     ON which = 3
610        move_3
```

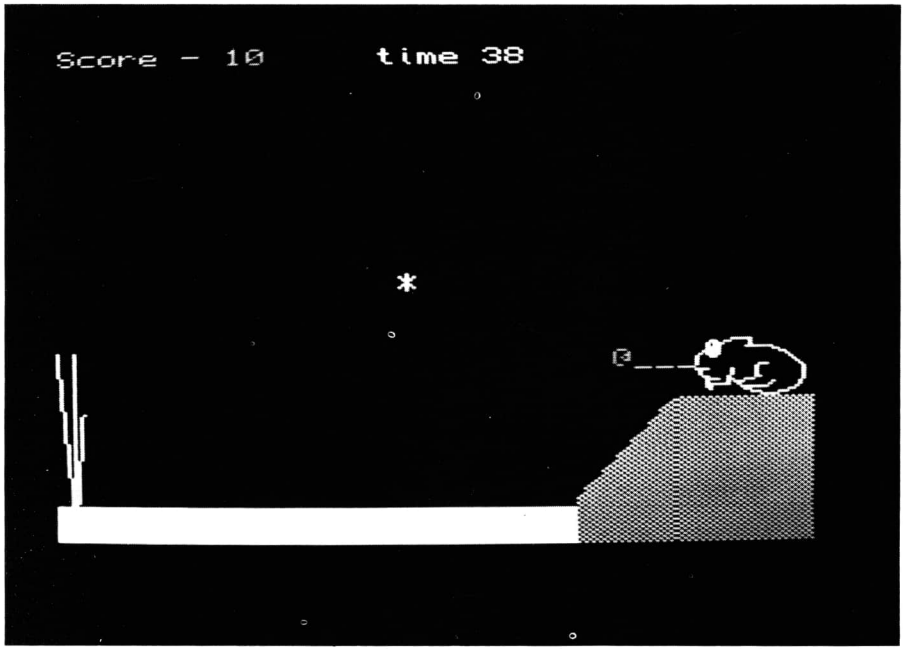
```
620   END SELect
630   PAUSE 9-1$
640   :
650 NEXT i
660 :
670 PAPER #0,0 : CLS #0
680 INPUT #0,"which one ? ";b$
690 IF b$ = "" THEN GO TO 670
700 IF b$<"1" OR b$>"3"THEN GO TO 670
710 IF a(b$) = 1 THEN
720   INK 6
730   BEEP 10000,1,100,100,1
740   PRINT "     You got it ! "
750   STOP
760 END IF
770 PRINT "    WRONG!"
780 BEEP 10000,100
790 :
800 DEFine PROCedure move_1
810  a(0)=a(1):a(1)=a(2):a(2)=a(0)
820  AT 5,10 : PRINT "     "
830  FOR loop = 0 TO 3
840    AT 4,10+loop : PRINT "o"
850    AT 6,13-loop : PRINT "o"
860    AT 4,10+loop : PRINT " "
870    AT 6,13-loop : PRINT " "
880  END FOR loop
890  AT 5,10 : PRINT "o  o"
900 END DEFine
910 :
920 DEFine PROCedure move_2
930  a(0)=a(1):a(1)=a(3):a(3)=a(0)
940  AT 5,10 : PRINT "   o    "
950  FOR loop = 0 TO 6
960    AT 4,10+loop : PRINT "o"
970    AT 6,16-loop : PRINT "o"
980    AT 4,10+loop : PRINT " "
990    AT 6,16-loop : PRINT " "
1000  END FOR loop
1010  AT 5,10 : PRINT "o  o  o"
```

```
1020 END DEFine
1030 :
1040 DEFine PROCedure move_3
1050  a(0)=a(2):a(2)=a(3):a(3)=a(0)
1060  AT 5,13 : PRINT "     "
1070  FOR loop = 0 TO 3
1080    AT 4,16-loop : PRINT "o"
1090    AT 6,13+loop : PRINT "o"
1100    AT 4,16-loop : PRINT " "
1110    AT 6,13+loop : PRINT " "
1120  END FOR loop
1130  AT 5,13 : PRINT "o   o"
1140 END DEFine
```

Chapter 5

"WEAPON" GAMES

Chameleon



The Chameleon has got to change colour to reach black, which it can do if it eats enough flies. You have a time limit in which to catch the flies. You press the space bar to make the tongue flick out. It seems easy, but the Chameleon can only make enough "sticky" for one try. Then it has to swallow its tongue before it tries again.

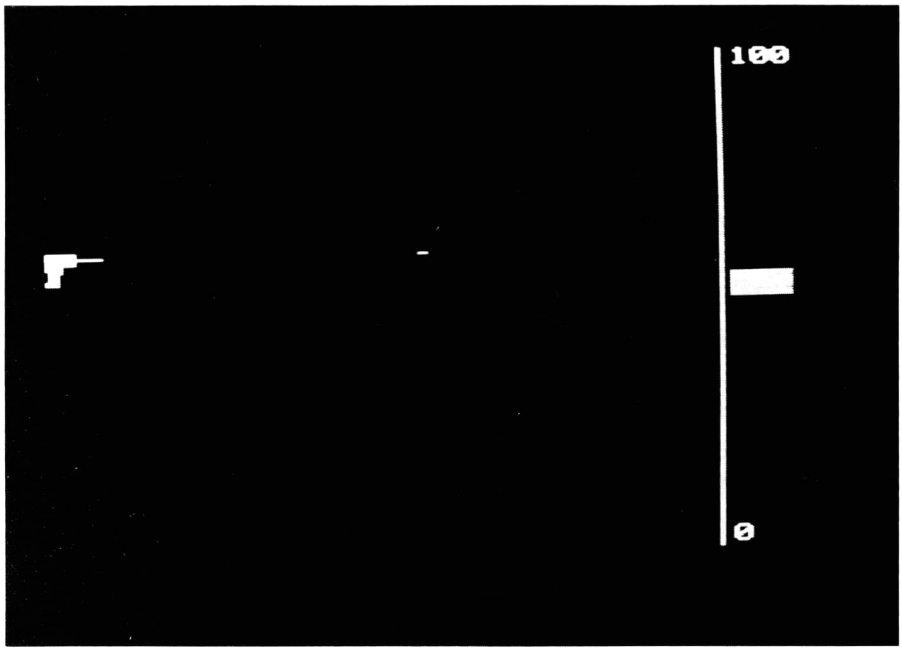
```
100 REMark **************
110 REMark * Chameleon *
120 REMark **************
130 REMark
140 MODE 8
150 PAPER 0
160 CLS
170 INK 6
180 CSIZE 3,1
190 PRINT ,"  Chameleon"
200 OVER -1
```

```
1010  i = i - 1
1020  sc = sc + 10
1030  AT 0,0
1040  PRINT "Score - ";sc
1050  IF i = 0 THEN
1060   INK i
1070   chameleon
1080   saved
1090  END IF
1100  GO TO 430
1110 END DEFine
1120 :
1130 DEFine PROCedure saved
1140  FOR n = 50 TO 20 STEP -10
1150    FOR a = n+15 TO n-10 STEP -5
1160      BEEP 5000,a,a+5,1,2,15,0,0
1170      PAUSE 1
1180    NEXT a
1190  NEXT n
1200  FOR n = 1 TO 12
1210    BEEP 5000,70/n,100/n,5,14,40,3,3
1220    PAUSE 5
1230  NEXT n
1240  INK 7
1250  i = 7
1260  GO TO 420
1270 END DEFine
1280 :
1290 DEFine PROCedure chameleon
1300  ARC 343,91 TO 350,100,-2
1310  ARC TO 370,106,-1
1320  FILL 1
1330  INK 7
1340  CIRCLE 352,100,4
1350  FILL 0
1360  INK i
1370  LINE 370,106 TO 368,100
1380  ARC TO 400,92,-1
1390  ARC TO 395,80,-1.5
1400  ARC TO 365,82,-1
```

```
1410  ARC TO 386,84,1
1420  ARC 380,93 TO 377,86,3.5
1430  LINE TO 367,84 TO 366,83
1440  LINE 367,84 TO 360,85
1450  LINE 361,90 TO 359,82
1460  LINE TO 350,82 TO 351,80
1470  ARC TO 348,85,-2
1480  LINE TO 355,85
1490  ARC TO 343,90,-1
1500  LINE TO 350,90
1510  INK 0
1520  LINE 349,102 TO 349,104
1530 END DEFine
1540 :
1550 DEFine PROCedure die
1560  FOR n=2 TO 80 STEP 4
1570    BEEP 5000,n,n+10,45,67,0,2,0
1580    INK RND (1 TO 255)
1590    CIRCLE 380,90,n
1600  NEXT n
1610  INK 0
1620  FOR n = 2 TO 80 STEP 4
1630    CIRCLE 380,90,n
1640  NEXT n
1650  INPUT #0;"Another go ? ";g$
1660  IF g$=="y" THEN RUN
1670  IF g$=="n" THEN STOP
1680  CLS #0
1690  GO TO 1650
1700 END DEFine
1710 :
1720 DEFine PROCedure Landscape
1730  LOCal i
1740  BLOCK 80,59,350,140,4,0,3
1750  INK 4,0,3
1760  FOR i = 331 TO 280 STEP -1
1770   LINE i,75-(330-i) TO i,0
1780  NEXT i
1790  INK 5
1800  FILL 1
```

```
1810  LINE TO i,75-(330-i)
1820  LINE 0,1 TO 0,20
1830  FILL 0
1840  INK 4
1850  LINE 10,10 TO 0,100
1860  LINE 10,10 TO 15,70
1870  LINE 10,10 TO 10,100
1880 END DEFine
```

Accuracy



How good a shot are you? The machine tells you how far up the wall you will find your target and you must get as close as you can. The difficulty level controls how close your shot must be. Use function key 4 to raise the gun and function key 5 to lower it. The space bar is your trigger.

```
100 REMark ************
110 REMark * Accuracy *
120 REMark ************
130 REMark
140 MODE 8
150 SCALE 100,0,0
160 PAPER 0
170 CLS
180 INK 6
190 CSIZE 3,1
200 PRINT ,"  Accuracy"
```

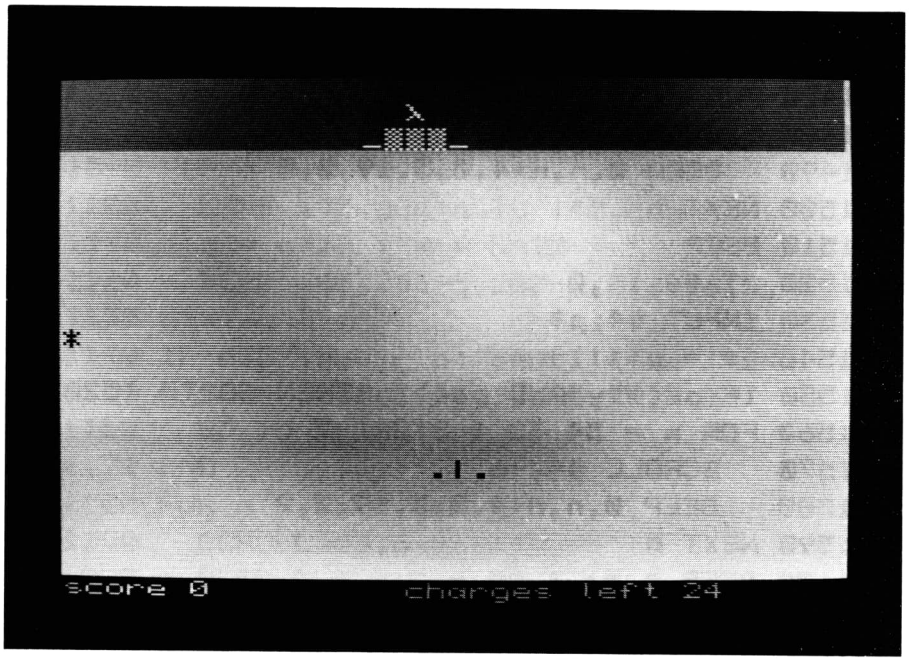
```
210 OVER -1
220 AT 0,0
230 INK 1
240 PRINT ,"  ________"
250 OVER 0
260 INK 5
270 AT 2,0
280 PRINT "  Test your judgement."
290 PRINT "  You will be given a "
300 PRINT "  percentage. Use f4 & f5"
310 PRINT "  to move the gun up and"
320 PRINT "  down. Press space to fire"
330 INK 2
340 PRINT \"  Press any key to start"
350 PAUSE
360 INK 7
370 CLS
380 CSIZE 0,0
390 PRINT "level (1-10:hard to easy)";
400 INPUT " ? ";le
410 IF le <> INT(le)THEN GO TO 170
420 IF le >100R le <1 THEN GO TO 160
430 CLS
440 INK 7
450 OPEN #3,con
460 WINDOW #3,50,255,20,0
470 OPEN #4,con
480 WINDOW #4,150,230,202,0
490 AT 19,32
500 PRINT "0"
510 AT 0,32
520 PRINT "100"
530 LINE 140,0 TO 140,100
540 percent = RND(1 TO 100)
550 AT 9,32
560 INK 2
570 PAPER 6
580 PRINT percent;"%"
590 INK 7
600 TURNTO 0
```

```
610 FILL 1
620 gun = RND (1 TO 99)
630 POINT 13,gun
640 PENDOWN
650 TURN 180
660 MOVE 6
670 TURN -90
680 MOVE 1
690 TURN 90
700 MOVE 7
710 TURN 90
720 MOVE 3
730 TURN 90
740 MOVE 1
750 TURN -100
760 MOVE 3
770 TURNTO 0
780 MOVE 3
790 TURN 80
800 MOVE 4
810 TURNTO 0
820 MOVE 3
830 TURN 90
840 MOVE 1
850 TURNTO 0
860 MOVE 6
870 FILL 0
880 a$ = INKEY$
890 IF a$ = "" THEN GO TO 880
900 IF CODE (a$)=248 AND gun>0 THEN
910   SCROLL #3,2
920   gun = gun -1
930 END IF
940 IF CODE(a$)=244 AND gun<100 THEN
950   SCROLL #3,-2
960   gun = gun + 1
970 END IF
980 IF CODE(a$) = 32 THEN GO TO 1000
990 GO TO 880
1000 BEEP 8000,240,100,5,1,190,0,15
```

```
1010 FOR n = 15 TO 135 STEP 3
1020   INK 0
1030   LINE n,gun TO n+3,gun
1040   INK 6
1050   LINE n+3,gun TO n+5,gun
1060 NEXT n
1070 INK 0
1080 LINE 135,gun TO 141,gun
1090 FOR n = 0 TO 1e
1100   LINE 140,gun+n TO 140,gun-n
1110   q = n*30 : w = n*20
1120   BEEP 300,255-q,200-w,0,3,16,0,15
1130 NEXT n
1140 IF ABS (gun - percent) <=1e THEN
1150   BEEP 3000,1,10,10,90,20,0,0
1160   GO TO 1320
1170 END IF
1180 FOR n = 1 TO gun
1190   SCROLL #3,5
1200   BEEP 0,n,n+5,5,90,4,10,2
1210 NEXT n
1220 BEEP
1230 AT 19,14
1240 INK 2,4,3
1250 PAPER 0
1260 PRINT "Bad Luck.."
1270 FOR n = 1 TO 20
1280   SCROLL #4,-4
1290 NEXT n
1300 PAUSE 20
1310 GO TO 1400
1320 AT 19,14
1330 INK 3,5,3
1340 PAPER 0
1350 PRINT "Well Done!!"
1360 FOR n= 1 TO 20
1370   SCROLL #4,-4
1380 NEXT n
1390 PAUSE 20
1400 FOR n = 1 TO 30
```

```
1410   SCROLL #4,2
1420 NEXT n
1430 AT 0,14
1440 INK 1,3,3
1450 PAPER 0
1460 PRINT "Play again ? "
1470 FOR n = 1 TO 56
1480   SCROLL #4,2
1490   BEEP 0,n,n+4,0,3,19,2,2
1500 NEXT n
1510 BEEP
1520 AT #4,14,0
1530 INPUT #4;p$
1540 p$ = p$(1)
1550 IF p$<>"y"AND p$<>"n"THEN GO TO 1520
1560 FOR n = 54 TO 1 STEP -2
1570   SCROLL #4,-6
1580   BEEP 0,n,n+4,0,3,19,2,2
1590 NEXT n
1600 BEEP
1610 IF p$(1) = "y" THEN
1620   CLS #3
1630   GO TO 160
1640 END IF
```

Depth Charge



Submarines appear underneath your ship and can travel from the left or the right. You must kill them off before you run out of depth charges. Use the ← and → to control the direction in which you want the ship to move. You use the ↑ and ↓ keys to set the depth at which your depth charges will explode, and the space bar to fire. Be careful, because the submarines are not helpless!

```
100 REMark ****************
110 REMark * DEPTH CHARGE *
120 REMark ****************
130 MODE 8
140 PAPER 0
150 PAPER #2,0
160 CLS
170 CLS #2
180 CSIZE 3,1
190 AT 0,0
```

```
200 INK 2
210 PRINT "      ____________"
220 OVER -1
230 INK 6
240 AT 0,0
250 PRINT "      DEPTH CHARGE"
260 CSIZE 1,0
270 OVER 0
280 PRINT
290 PRINT "Controls :-"
300 PRINT
310 PRINT "<left>  Move ship left"
320 PRINT "<right> Move ship right"
330 PRINT "<space> Drop depth charge"
340 PRINT "<up>    Move gauge up"
350 PRINT "<down>  Move gauge down"
360 INK 5
370 PRINT
380 PRINT "Blow up the submarines"
390 PRINT "and avoid the torpedoes"
400 PRINT
410 INK 4
420 PRINT "Press any key to start"
430 IF INKEY$="" THEN GO TO 430
440 CSIZE 0,0
450 sc=0
460 ch=25
470 dt=10
480 sf=0
490 s1$="   /   "
500 s2$=" _"&CHR$(0)&CHR$(0)&CHR$(0)&"_ "
510 x=15
520 sx=0
530 sy=10
540 sd=1
550 f=0
560 score
570 PAPER 5
580 CLS
590 PAPER 1
```

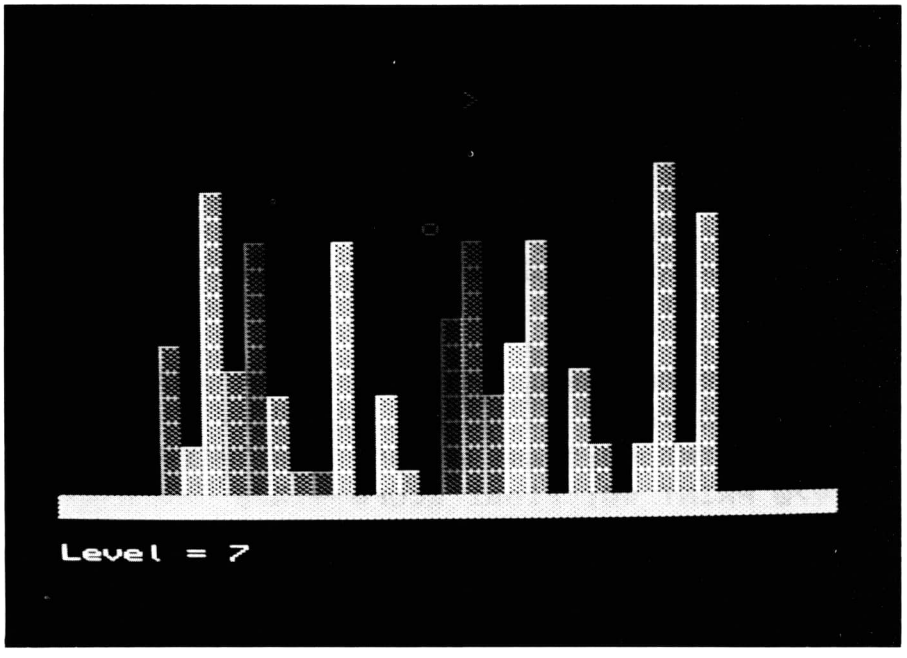
```
600 FOR i=0 TO 2
610   AT i,0
620   PRINT FILL$(" ",37)
630 NEXT i
640 PAPER 5
650 moveship
660 submarine
670 setdepth
680 IF KEYROW(1)=64 AND f=0 THEN
690   f=1
700   bx=x
710   by=3
720   ch=ch-1
730   score
740   IF ch=0 THEN GO TO 1950
750 END IF
760 IF f=1 THEN depthcharge
770 GO TO 650
780 DEFine PROCedure moveship
790 IF KEYROW(1)=2 AND x>3 THEN
800   x=x-1
810   GO TO 880
820 END IF
830 IF KEYROW(1)=16 AND x<29 THEN
840   x=x+1
850   GO TO 880
860 END IF
870 GO TO 940
880 PAPER 1
890 INK 7
900 AT 1,x
910 PRINT s1$
920 AT 2,x
930 PRINT s2$
940 END DEFine
950 DEFine PROCedure submarine
960 sx=sx+sd
970 IF sx=1 OR sx=31 THEN GO TO 1030
980 PAPER 5
990 INK 6
```

```
1000 AT sy,sx
1010 PRINT " .|. "
1020 GO TO 1130
1030 PAPER 5
1040 AT sy,sx
1050 PRINT "     "
1060 sd=RND(1 TO 2)
1070 IF sd=1 THEN sx=1
1080 IF sd=2 THEN
1090   sx=31
1100   sd=-1
1110 END IF
1120 sy=RND(3 TO 19)
1130 IF sf=0 THEN
1140   sf=1
1150   sbx=sx
1160   sby=sy
1170 END IF
1180 IF sf=1 THEN subfire
1190 END DEFine
1200 :
1210 DEFine PROCedure depthcharge
1220 AT by,bx
1230 PRINT " "
1240 by=by+1
1250 IF by>dt THEN
1260   f=0
1270 GO TO 1480
1280 END IF
1290 IF bx=sx AND (by=sy OR by=sy-1) THEN
1300  AT sy,sx
1310  PRINT "     "
1320  sc=sc+1
1330  f=0
1340  BEEP 10000,100,110,3,3,3,4
1350  sd=RND(1 TO 2)
1360  IF sd=1 THEN sx=1
1370  IF sd=2 THEN
1380  sx=31
1390  sd=-1
```

```
1400 END IF
1410  sy=RND(3 TO 19)
1420  GO TO 1480
1430 END IF
1440 INK 3
1450 AT by,bx
1460 PRINT"2"
1470 BEEP 120,10+by*10
1480 END DEFine
1490 :
1500 DEFine PROCedure score
1510 AT #0,0,0
1520 INK #0,4
1530 PRINT#0; "score ";sc
1540 INK #0,2
1550 AT #0,0,16
1560 PRINT #0,"charges left ";ch;" "
1570 END DEFine
1580 :
1590 DEFine PROCedure setdepth
1600 AT dt,0
1610 PRINT " "
1620 IF KEYROW(1)=128 AND dt<19 THEN dt=dt+1
1630 IF KEYROW(1)=4 AND dt>4 THEN dt=dt-1
1640 INK 0
1650 AT dt,0
1660 PRINT "*"
1670 END DEFine
1680 :
1690 DEFine PROCedure subfire
1700 AT sby,sbx
1710 PRINT " "
1720 sby=sby-1
1730 IF sby=2 AND (sbx>x AND sbx<x+4) THEN
1740  blow
1750  GO TO 1960
1760 END IF
1770 IF sby=2 THEN
1780   sf=0
1790   GO TO 1840
```

```
1800 END IF
1810 AT sby,sbx
1820 INK 3
1830 PRINT "!"
1840 END DEFine
1850 :
1860 DEFine PROCedure blow
1870 PAPER 1
1880 FOR i=1 TO 30
1890 BEEP 10,i,40,1,1,2,3,i
1900 AT RND(0 TO 2),RND(1 TO 6)+x
1910 INK RND(1 TO 255)
1920 PRINT "*"
1930 NEXT i
1940 END DEFine
1950 CLS #0
1960 PRINT #0;"game over"
1970 PRINT #0;"Final score = %";
1980 PRINT #0;INT(sc*(100/ch))
1990 STOP
```

Leveller



Your plane is running out of fuel and you must land. In order to do so, you must knock down every building in sight to prepare your glide path. Watch it though, some of the buildings are pretty high. If you do land, your next flight out will be tougher because the next city has even taller buildings.

```
100 REMark ************
110 REMark * Leveller *
120 REMark ************
130 REMark
140 MODE 8
150 PAPER 0
160 CLS
170 INK 6
180 CSIZE 3,1
190 PRINT ,"  Leveller"
200 OVER -1
```

```
210 AT 0,0
220 INK 1
230 PRINT ,"  ________"
240 INK 5
250 AT 2,0
260 PRINT "  Your aircraft is running"
270 PRINT "  out of fuel! So you must"
280 PRINT "  find a place to land."
290 PRINT "  The only place your fuel"
300 PRINT "  supply will get you to "
310 PRINT "  is an old city."
320 INK 2
330 PRINT\" Press any key to continue"
340 PAUSE
350 CLS
360 INK 6
370 PRINT ,"  Leveller"
380 INK 1
390 AT 0,0
400 PRINT ,"  ________"
410 OVER 0
420 AT 2,0
430 INK 5
440 PRINT "  To land you must destroy"
450 PRINT "  all the buildings. Use "
460 PRINT "  the space-bar to drop "
470 PRINT "  your bombs. Be quick as"
480 PRINT "  there is little time."
490 INK 2
500 PRINT \
510 PRINT "   Press any key to start"
520 PAUSE
530 CSIZE 0,0
540 CLS
550 SCALE 162,0,0
560 level = 0
570 s_x = 0
580 s_y = 0
590 bom = 0
600 AT #0,0,0
```

```
610 PRINT #0;"Level = ";level
620 DIM buil(40)
630 ship$ = " >"
640 AT 18,0
650 PAPER 250
660 PRINT FILL$(" ",37);
670 INK 0
680 FOR i = 5 TO 30
690   x = RND(0 TO 5+level)
700   PAPER RND (2 TO 7)
710   FOR j = 17 TO 17-x STEP -1
720     AT j,i
730     PRINT CHR$(0)
740   NEXT j
750   buil(i) = x+1
760 NEXT i
770 :
780 ship
790 bomb
800 bomb
810 :
820 GO TO 780
830 :
840 DEFine PROCedure ship
850  PAPER 0
860  INK 4
870  s_x = s_x + 1
880  IF 18-buil(s_x+1)=s_y THEN crash
890  IF s_x = 31 THEN
900    AT s_y , s_x -1
910    PRINT "  "
920    s_x = 0
930    s_y = s_y +1
940    IF s_y = 17 THEN land
950  END IF
960  AT s_y , s_x
970  PRINT ship$
980 END DEFine
990 :
1000 DEFine PROCedure bomb
```

```
1010  IF bom = 0 THEN
1020    IF INKEY$ = "" THEN GO TO 1200
1030    bom = s_y + 1
1040    a = s_x + 1
1050    buil(s_x+1) = 0
1060    BEEP 0,bom,100,1000,1
1070    GO TO 1180
1080  END IF
1090  AT bom , a
1100  PAPER 0
1110  PRINT " "
1120  bom = bom +1
1130  IF bom = 18 THEN
1140    bom = 0
1150    BEEP
1160    GO TO 1200
1170  END IF
1180  AT bom , a
1190  PRINT "o"
1200  END DEFine
1210 :
1220 DEFine PROCedure land
1230  FOR i = 1 TO 35
1240    IF buil(i)<>0 THEN
1250      FOR j = 1 TO i
1260        AT 17,j
1270        PRINT ship$
1280        BEEP 10,10
1290      NEXT j
1300      s_y = 16
1310      s_x = i
1320      crash
1330    END IF
1340  NEXT i
1350 PRINT #0;"You landed"
1360 level = level + 1
1370 PRINT #0;"Now try level";level
1380  PAUSE 200
1390  CLS #0
1400  GO TO 570
```

```
1410 END DEFine
1420 :
1430 DEFine PROCedure crash
1440  x = s_x * 7.5
1450  y = (18-s_y) * 9
1460  FOR j = 1 TO 50
1470    BEEP 10,RND (1 TO 255)
1480    INK RND (1 TO 255)
1490    x1 = RND (-20 TO 20) + x
1500    y1 = RND (-20 TO 20) + y
1510    LINE x,y TO x1,y1
1520  NEXT j
1530  game_over
1540 END DEFine
1550 :
1560 DEFine PROCedure game_over
1570  CSIZE 3,1
1580  INK 255
1590  FOR i = 0 TO 7
1600    PAPER i+1
1610    AT i,10
1620    PRINT "GAME OVER"
1630  NEXT i
1640  CSIZE #0,3,1
1650  CLS #0
1660  PRINT #0;TO 9;"ANOTHER GO?"
1670  a$ = INKEY$
1680  IF a$ = "" THEN GO TO 1670
1690  IF a$ INSTR"NynY"=0THEN GO TO 1670
1700  IF "ny"(a$ INSTR"NYny")=="y"THEN RUN
1710  STOP
1720 END DEFine
```

Strafe



You are coming in for a low-level bombing run and your orders are to knock out everything in sight. Don't forget that you are running out of fuel as well, so look for fuel dumps.

```
100 REMark **********
110 REMark * Strafe *
120 REMark **********
130 REMark
140 sc = 0
150 fu = 299
160 cr = 0
170 MODE 8
180 PAPER 0
190 CLS
200 INK 6
210 CSIZE 3,1
220 PRINT ,"  Strafe"
```

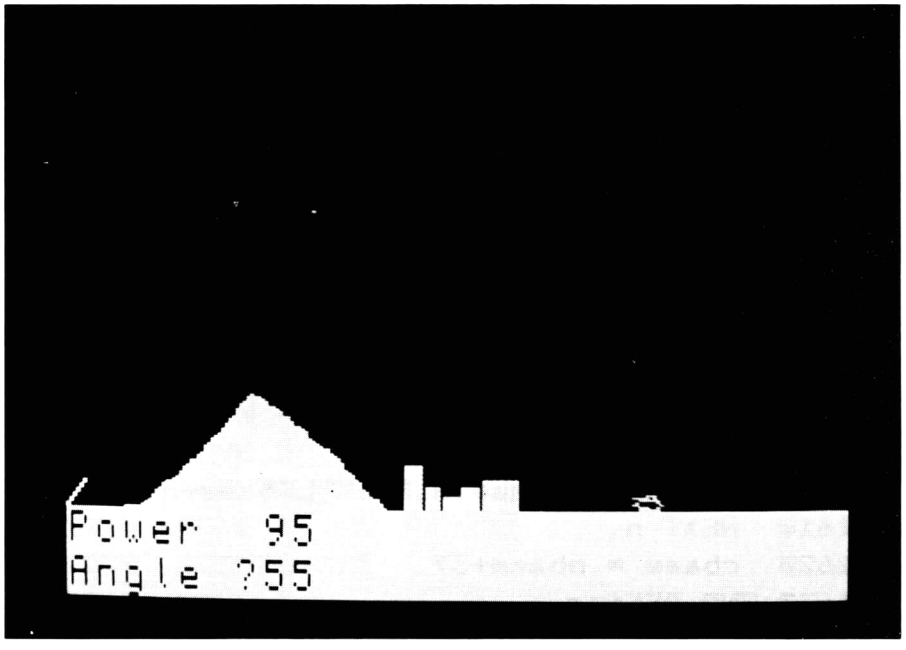
```
230 OVER -1
240 INK 2
250 AT 0,0
260 PRINT ,"  ______"
270 INK 5
280 AT 2,0
290 PRINT "    Use the space-bar to "
300 PRINT "    drop your bombs. Hit "
310 PRINT "    the fuel dumps ' for "
320 PRINT "    more fuel. GOOD LUCK"
330 udg
340 INK 2
350 PRINT
360 PRINT "   Press any key to start"
370 chars
380 PAUSE
390 OVER 0
400 CSIZE 0,0
410 SCALE 100,0,0
420 OPEN #3,con
430 WINDOW #3,400,75,40,110
440 a$="                  GET READY....."
450 o$=" #   #    $$     &   % %    '"
460 e$=")(()*+*)("
470 f = 0
480 ef = 0
490 PAPER 1
500 CLS
510 INK 7
520 PAPER 4
530 AT 18,0
540 PRINT FILL$(" ",37)
550 plane
560 AT 17,0
570 INK 7
580 PRINT a$
590 IF ef<>0 THEN
600   a$(ef) = e$(ef)
610   ef = ef - 1
620   IF ef = 2 THEN ef = 0
```

```
630 END IF
640 a$ = a$(2 TO)&o$(RND(1 TO 29))
650 IF f = 1 THEN bomb
660 IF f=0 AND cr=0 AND INKEY$<>"" THEN
670   f = 1
680   x = 14
690   y = 12
700 END IF
710 IF NOT cr THEN
720   fu = fu - 1
730   INK 1
740   PAPER 7
750   AT 0,15
760   PRINT "Fuel - ";fu;"   "
770   PAPER 1
780   IF fu = 0 THEN cr = 1
790 END IF
800 IF cr THEN crash
810 GO TO 560
820 :
830 DEFine PROCedure bomb
840  AT y,x
850  PRINT " "
860  x = x - 1
870  y = y + 1
880  IF y = 17 THEN check
890  AT y,x
900  PRINT "."
910 END DEFine
920 :
930 DEFine PROCedure check
940  f = 0
950  IF a$(9) INSTR o$=0 THEN GO TO 1060
960  IF a$(9) = "'" THEN
970    fu = fu + 10
980    BEEP 5000,20,40,45,45
990  END IF
1000  sc = sc + (CODE(a$(9))-33)
1010  AT 0,0
1020  PAPER 6
```

```
1030  INK 2
1040  PRINT "Score - ";sc;" "
1050  PAPER 1
1060  a$(9) = "("
1070  ef = 9
1080  BEEP 20000,200,230,45,45,0,15,0
1090 END DEFine
1100 :
1110 DEFine PROCedure plane
1120  POINT 75,41
1130  INK 4
1140  PENDOWN
1150  TURNTO 180
1160  FILL 1
1170  MOVE 10
1180  TURN -5
1190  MOVE 5
1200  TURNTO 90
1210  MOVE 4
1220  TURNTO 315
1230  MOVE 5
1240  TURNTO 0
1250  MOVE 10
1260  MOVE -4
1270  TURNTO 240
1280  MOVE 3
1290  TURNTO 180
1300  MOVE 2
1310  TURNTO 50
1320  MOVE 3
1330  INK 7
1340  FILL 0
1350 END DEFine
1360 :
1370 DEFine PROCedure crash
1380  PAPER #3,1
1390  SCROLL #3,2
1400  PAN #3,3
1410  cr = cr + 1
1420  IF cr = 25 THEN
```

```
1430   BEEP 32000,200,230,3,45,0,15,0
1440    FOR a = 10 TO 30 STEP 5
1450     FOR n = 0 TO 2*PI
1460      INK RND (2 TO 200)
1470      FILL RND (0 TO 1)
1480 CIRCLE 85,18,RND(a)/5,RND(10),RND(1)
1490      BEEP 5000,a,a+n,4,45,15,0,10
1500     NEXT n
1510    NEXT a
1520    STOP
1530   END IF
1540 END DEFine
1550 :
1560 DEFine PROCedure udg
1570  base = PEEK_L (167722)
1580  nbase = RESPR(875)
1590  FOR n=0 TO 875 STEP 4
1600    POKE_L nbase+n,PEEK_L(base+n)
1610  NEXT n
1620  cbase = nbase+37
1630 END DEFine
1640 :
1650 DEFine PROCedure chars
1660  RESTORE
1670  FOR n = 1 TO 81
1680    READ x
1690    POKE cbase+n,x
1700  NEXT n
1710  POKE_L 167722,nbase
1720  DATA 0,0,8,56,124,84,124,84,116
1730  DATA 0,0,0,0,48,80,92,124,36
1740  DATA 0,0,0,64,32,24,124,56,56
1750  DATA 0,4,72,56,8,28,28,60,60
1760  DATA 56,68,92,76,92,92,56,68,68
1770  DATA 0,0,0,0,0,0,0,40,16
1780  DATA 0,0,0,40,40,16,56,16,16
1790  DATA 0,20,88,56,60,24,88,48,56
1800  DATA 56,124,124,124,56,16,48,48,56
1810 END DEFine
```

Mortar



Try to lob a mortar shell at the oncoming tank. Use the ↑ and ↓ keys to control the angle of the mortar, and the space bar to fire. Just one small problem though. Knock out any protected buildings and you are in trouble.

```
100 REMark **********
110 REMark * mortar *
120 REMark **********
130 MODE 8
140 SCALE 200,0,0
150 PAPER #0,0
160 PAPER #1,0
170 CLS # 1
180 CLS # 0
190 CSIZE 3,1
200 AT 0,10
210 INK 2
```

```
220 UNDER 1
230 PRINT "MORTAR"
240 INK 1
250 PRINT
260 PRINT "Controls"
270 UNDER 0
280 PRINT
290 INK 6
300 PRINT "up arrow   raise mortar"
310 PRINT "down arrow lower mortar"
320 PRINT "space bar  fire  mortar"
330 INK 4
340 PRINT
350 PRINT " Press any key to start"
360 IF INKEY$ = "" THEN GO TO 360
370 PAPER #0,4
380 INK #0,0
390 CLS # 0
400 CSIZE #0,3,1
410 g = 32.2
420 PAPER 0
430 CLS
440 INK 4
450 mountain
460 ss = x
470 village ss
480 d = 320
490 tank d,2
500 CLS # 0
510 v = RND(75 TO 105)
520 a = 45
530 AT #0,0,0
540 PRINT #0,"POWER  ";v
550 AT #0,1,0
560 PRINT #0;"ANGLE  ";a;" "
570 a$=INKEY$
580 IF a$="" THEN GO TO 570
590 INK 0
600 POINT 0,0
610 b1=SIN(RAD(90-a))*10
```

```
620 b2=COS(RAD(90-a))*10
630 LINE TO b1,b2
640 IF KEYROW (1)=128 AND a>0 THEN a=a-1
650 IF KEYROW (1)=4 AND a<91 THEN a=a+1
660 INK 6
670 b1=SIN(RAD(90-a))*10
680 b2=COS(RAD(90-a))*10
690 LINE 0,0 TO b1,b2
700 IF a$=" " THEN GO TO 740
710 AT#0,1,0
720 PRINT #0;"angle  ";a;" "
730 GO TO 570
740 INK 0
750 LINE 0,0 TO b1,b2
760 a  = RAD(a)
770 r  = v*v*SIN(2*a)/g
780 h  = (v*SIN(a))^2/(2*g)
790 t1 = 2*v*SIN(a)/g
800 IF t1 = 0 THEN t1 = 1
810 FOR t = 0 TO t1 STEP t1/100
820   x1= x
830   y1= y
840   x = INT(v*COS(a)*t)
850   y = INT((v*SIN(a)-g*t/2)*t)
860   BEEP 20,200-y
870     IF getcol (x,y) = 4 THEN
880        hole
890        GO TO 1000
900     END IF
910   INK 5
920   POINT x,y
930   INK 0
940   POINT x1,y1
950 NEXT t
960 IF a = RAD(90) THEN GO TO 2170
970 IF x>ss+4 AND x<ss+50 THEN GO TO 2050
980 IF x>d AND x<d+15 THEN GO TO 2390
990 explosion
1000 BEEP
1010 tank d,0
```

```
1020 d = d - 10
1030 IF d < ss + 36 THEN GO TO 2110
1040 tank d,2
1050 INK 0
1060 POINT x,y
1070 GO TO 500
1080 :
1090 DEFine PROCedure explosion
1100  OVER - 1
1110  FOR i = 1 TO 14
1120    INK i
1130    CIRCLE x,y,i
1140    BEEP 0,100-i*10,100,1,1,1,14
1150  NEXT i
1160  FOR i = 14 TO 0 STEP -1
1170    INK i
1180    BEEP 0,100-i*10,100,1,1,1,14
1190    CIRCLE x,y,i
1200  NEXT i
1210  BEEP
1220  OVER 0
1230 END DEFine
1240 :
1250 DEFine PROCedure hole
1260 INK 0
1270 FOR i = 1 TO 40
1280 b1=RND(1 TO 15)
1290 b2=RND(1 TO 15)
1300 BEEP 20000,100,100,1,1,1,b1,b2
1310 POINT x+RND(-5 TO 5),y+RND(-5 TO 5)
1320 NEXT i
1330 END DEFine
1340 :
1350 DEFine PROCedure mountain
1360  x = 100
1370  y = RND (40 TO 80)
1380  x1= x
1390  y1= y
1400  y1= y1- RND * 2
1410  y = y - RND * 2
```

```
1420  IF y<0 AND y1<0 THEN GO TO 1470
1430  x = x + 1
1440  LINE x,0 TO x,y1
1450  LINE 101+(100-x),0 TO 101+(100-x),y
1460  GO TO 1400
1470 END DEFine
1480 :
1490 DEFine FuNction getcol (f,g)
1500  xx = f
1510  yy = g
1520  convert xx,yy
1530  add=INT(131072+128*yy+INT(xx/4)*2)
1540  bit=(xx MOD 4)*2
1550  mask=(2^(7-bit))
1560  grn=4*((mask && PEEK (add))<>0)
1570  mask=(2^(7-bit)*1.5)
1580  rb=(mask && PEEK (add+1))/mask*3
1590  colour = grn + rb
1600  RETurn colour
1610 END DEFine
1620 :
1630 DEFine PROCedure convert (xxx,yyy)
1640  LOCal s,w,h
1650  s    = 200
1660  w    = 448
1670  h    = 199
1680  atx = 32
1690  aty = 16
1700  xxx=INT((xxx*w/1.659/s+atx)/2)
1710  yyy = INT((s - yyy)*h/ s+ aty)
1720 END DEFine
1730 INK 0
1740 FOR i = 1 TO 40
1750   POINT x+RND(-5 TO 5),y+RND(-5 TO 5)
1760 NEXT i
1770 GO TO 510
1780 DEFine PROCedure tank (g,cl)
1790  INK cl
1800 LINE g,0 TO g+10,0
1810 LINE TO g+12,3 TO g+10,5
```

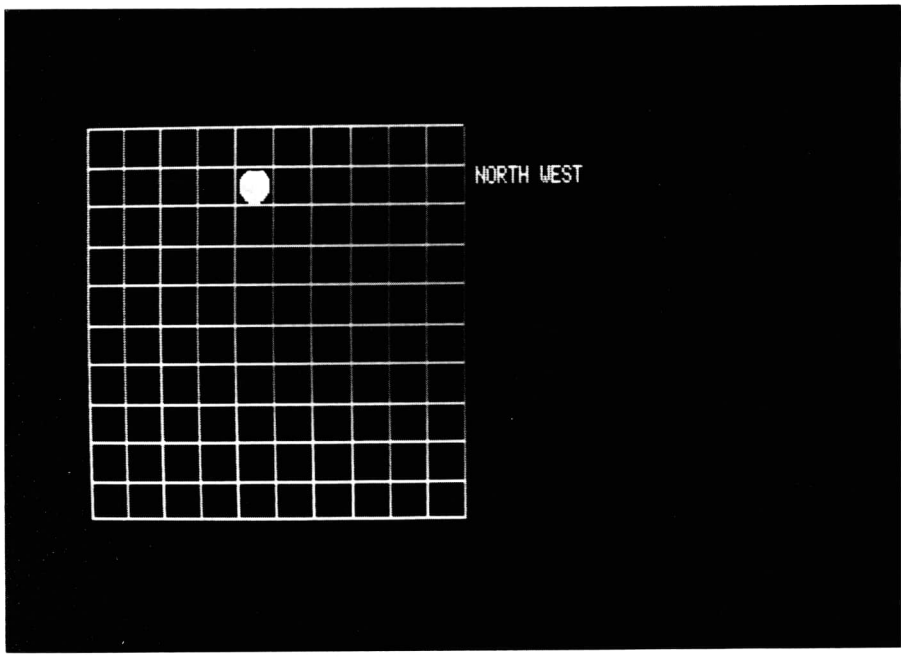
```
1820 LINE TO g,5 TO g-2,3
1830 LINE TO g,0
1840 LINE g,1 TO g+9,1
1850 LINE TO g+11,3 TO g+9,4
1860 LINE TO g,4 TO g-1,3
1870 LINE TO g,1
1880 LINE g+7,5 TO g+9,8
1890 LINE TO g+3,8 TO g+2,6.5
1900 LINE TO g+3,5
1910 LINE g + 2,6.5 TO g - 8,9
1920 END DEFine
1930 :
1940 DEFine PROCedure village (vil)
1950  FOR i = 1 TO 6
1960    f = RND (10 TO 20)
1970    cc = RND (1 TO 7)
1980    IF cc = 4 THEN GO TO 1970
1990    INK cc,RND (1 TO 3)
2000      FOR j = 1 TO 5
2010        LINE vil+i*8+j,0 TO vil+i*8+j,f
2020      NEXT j
2030  NEXT i
2040 END DEFine
2050 PAPER #0,0
2060 INK #0,6,3
2070 INK #0,6,1
2080 PRINT #0;"You just nuked the village"
2090 nuke x - 10
2100 GO TO 2430
2110 CLS # 0
2120 INK #0,4,1
2130 PRINT #0;"Tank destroyed villlage"
2140 nuke ss + 10
2150 nuke d
2160 GO TO 2430
2170 BEEP
2180 PAPER #0,2,6
2190 INK #0,0
2200 CLS # 0
2210 PRINT#0, " You just blew yourself up"
```

```
2220 nuke 1
2230 GO TO 2430
2240 DEFine PROCedure nuke (n)
2250  FOR i = 1 TO 25
2260    BEEP 0,100+i,100+i*2,2,2,2,i,i
2270    INK i
2280    LINE n,i TO n + 20,i
2290  NEXT i
2300  FOR i = 180 TO 0 STEP -3
2310    INK i
2320    x = SIN(RAD(i)) * 30
2330    y = COS(RAD(i)) * 15
2340    LINE n-x+20,40+y TO n+x,40+y
2350    BEEP 0,ABS(x),ABS(y),3,3,2,15,15
2360  NEXT i
2370  BEEP
2380 END DEFine
2390 PRINT #0;"You got the tank"
2400 CSIZE #0,0,0
2410 PRINT #0;"SCORE =";d - ss
2420 nuke d
2430 PRINT #0;"PLAY AGAIN ?";
2440 INPUT #0;a$
2450 IF a$ == "y" THEN RUN
2460 IF a$=="n" THEN STOP
2470 GO TO 2430
```

Chapter 6

MODERN GAMES

Locate



Try and find me. You are given directions showing where I am in relation to you and you can use all four cursor control keys to try and find me. You must be quick because my location from you is given at one particular instant. Don't expect me to stand still and wait for you to catch me.

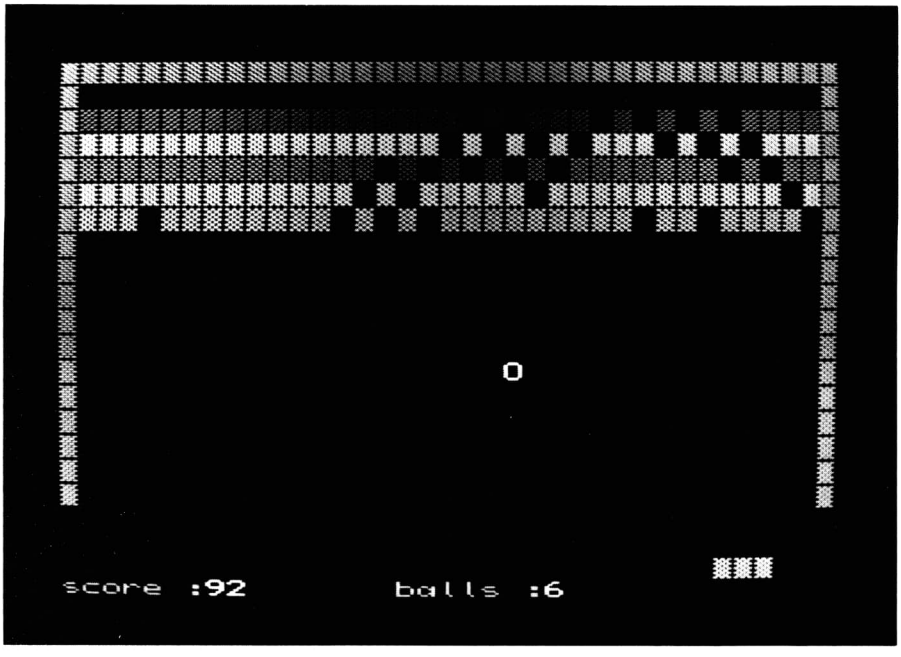
```
100 REMark **********
110 REMark * Locate *
120 REMark **********
130 REMark
140 MODE 8
150 PAPER 0
160 CLS
170 CSIZE 3,1
180 INK 6
190 PRINT ,"  Locate"
200 OVER -1
```

```
210 AT 0,0
220 INK 1
230 PRINT ,"  ______"
240 OVER 0
250 AT 2,0
260 INK 5
270 PRINT "  Use the cursor keys to"
280 PRINT "  move around the grid."
290 PRINT "  Follow the directions"
300 PRINT "  given to find me."
310 INK 2
320 PRINT \"  Press any key to start"
330 PAUSE
340 MODE 4
350 PAPER 0
360 CLS
370 SCALE 256,0,0
380 INK 2
390 OPEN #3,con
400 WINDOW #3,100,10,270,60
410 FOR n = 20 TO 220 STEP 20
420   LINE 20,n TO 220,n
430   LINE n,20 TO n,220
440 NEXT n
450 hx = RND(1 TO 10)
460 hy = RND(1 TO 10)
470 x = RND(1 TO 10)
480 y = RND(1 TO 10)
490 IF x=hx OR y=hy THEN GO TO 470
500 FILL 1
510 INK 4
520 CIRCLE 10+20*x,10+20*y,8
530 FILL 0
540 INK 7
550 FILL 1
560 CIRCLE 7+20*x,13+20*y,3
570 FILL 0
580 FILL 1
590 CIRCLE 14+20*x,13+20*y,3
600 FILL 0
```

```
610 INK 0
620 POINT 14+20*x,13+20*y
630 POINT 7+20*x,13+20*y
640 INK 2
650 q = 20*x : w = 20*y
660 ARC 5+q,8+w TO 15+q,8+w,1
670 IF hy > y THEN PRINT #3;"NORTH ";
680 IF hy < y THEN PRINT #3;"SOUTH ";
690 IF hx < x THEN PRINT #3;"WEST";
700 IF hx > x THEN PRINT #3;"EAST";
710 PRINT #3
720 a$ = INKEY$
730 k = CODE (a$)
740 nx=x+(k=200AND x<10)-(k=192AND x>1)
750 ny=y+(k=208AND y<10)-(k=216AND y>1)
760 IF nx = hx AND ny = hy THEN
770   GO SUB 900
780   GO TO 980
790 END IF
800 a = RND (2)
810 b = RND (2)
820 hy=hy+(b>1AND hy<10)-(b<1AND hy>1)
830 hx=hx+(a>1AND hx<10)-(a<1AND hx>1)
840 IF a$ = "" THEN
850   PAUSE 80
860   GO TO 670
870 END IF
880 GO SUB 900
890 GO TO 500
900 INK 0
910 FILL 1
920 q = 20*x : w = 20*y
930 LINE 1+q,2+w TO 1+q,19+w
940 LINE 19+q,2+w TO 19+q,19+w
950 FILL 0
960 x = nx : y = ny
970 RETurn
980 BEEP 30000,40,25,0,15,19,0,0
990 FILL 1
1000 INK 4
```

```
1010 CIRCLE 10+20*nx,10+20*ny,8
1020 FILL 0
1030 INK 7
1040 FILL 1
1050 CIRCLE 7+20*nx,13+20*ny,5
1060 FILL 0
1070 INK 7
1080 FILL 1
1090 CIRCLE 14+20*nx,13+20*ny,5
1100 FILL 0
1110 BEEP 30000,140,125,10,15,1,12,0
1120 INK 0
1130 FILL 1
1140 CIRCLE 7+20*nx,13+20*ny,3
1150 FILL 0
1160 INK 0
1170 FILL 1
1180 CIRCLE 14+20*nx,13+20*ny,3
1190 FILL 0
1200 INK 2
1210 FILL 1
1220 CIRCLE 10+20*nx,6+20*ny,5,.4,PI/2
1230 FILL 0
1240 IF NOT BEEPING THEN GO TO 1260
1250 GO TO 1240
1260 INK 0
1270 FILL 1
1280 FOR n = 1 TO 360 STEP 30
1290   CIRCLE 10+20*nx,10+20*ny,n
1300 NEXT n
1310 FILL 0
1320 CLS #0
1330 INPUT #0;"play again ? ";p$
1340 IF p$ = "" THEN GO TO 1320
1350 IF p$ == "y" THEN RUN
1360 IF p$ == "n" THEN STOP
1370 GO TO 1320
```

Batter



You have seven balls to try to batter your way through my wall. Use the ← and → keys to control the bat. If you get through, you get an extra ball each time. Try to build a big score.

```
100 REMark **********
110 REMark * BATTER *
120 REMark **********
130 CSIZE 0,0
140 MODE 8
150 DIM a(5,35)
160 PAPER 0
170 PAPER #2,0
180 CLS
190 CLS #2
200 CSIZE 3,1
210 AT 0,0
```

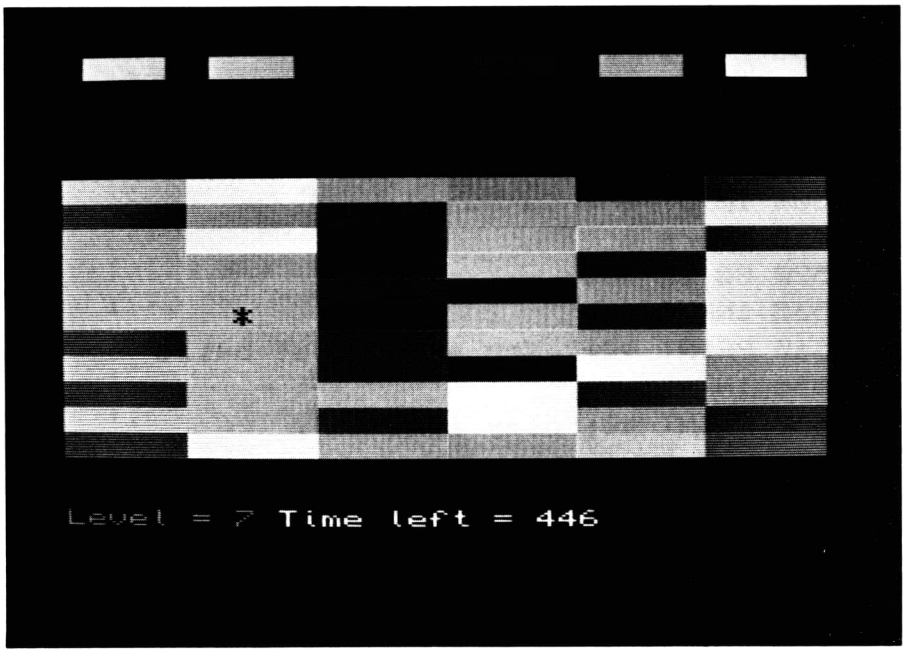
```
220 INK 4
230 PRINT "          Batter"
240 AT 0,0
250 INK 2
260 OVER 1
270 PRINT "          ______"
280 OVER 0
290 INK 1
300 PRINT
310 PRINT "Controls :-"
320 INK 6
330 PRINT "<Left>  Move bat left"
340 PRINT "<Right> Move bat right"
350 PRINT "Knock bricks out of wall"
360 PRINT "by directing ball with bat"
370 INK 3
380 PRINT " Press <1> For batter"
390 PRINT " Press <2> For batthrough"
400 PRINT " Press <3> For 'who cares?'"
410 a$=INKEY$
420 IF a$="" THEN GO TO 410
430 flag=1
440 f=0
450 IF a$="1" THEN flag=1
460 IF a$="2" THEN flag=0
470 IF a$="3" THEN f=1
480 CSIZE 0,0
490 CLS
500 INK 1
510 FOR i=17 TO 0 STEP -1
520   AT i,0
530   PRINT CHR$(0)
540   AT i,36
550   PRINT CHR$(0)
560   BEEP 20,i
570 NEXT i
580 BEEP 10000,20,50,2,1
590 AT 0,0
600 PRINT FILL$(CHR$(0),38)
610 bl=7
```

```
620 sc=0
630 cn=0
640 wall
650 x=15
660 bx=RND(2 TO 34)
670 by=10
680 bxd=-1
690 byd=-1
700 sc=0
710 bat
720 ball
730 GO TO 710
740 :
750 DEFine PROCedure bat
760  IF KEYROW(1)=2 AND x>0 THEN x=x-1
770  IF KEYROW(1)=16 AND x<32 THEN x=x+1
780  AT #0,0,x
790  INK 7
800  PRINT #0," ";CHR$(0);CHR$(0);CHR$(0);" "
810 END DEFine
820 :
830 DEFine PROCedure ball
840  x1=bx
850  y1=by
860  dn=RND/1000
870  bx=bx+bxd+dn
880  IF bx<2 OR bx>35 THEN bxd=-bxd
890  by=by+byd
900  IF by<2 THEN byd=-byd
910  IF by>18 AND (bx<x OR  bx>x+4) THEN GO
     TO 1480
920  IF by>18 THEN byd=-byd
930  IF by>1 AND by<7 THEN
940    xx=7-by
950    update
960  END IF
970  AT y1,x1
980  PRINT " "
990  AT by,bx
1000  PRINT "O"
```

```
1010 END DEFine
1020 :
1030 DEFine PROCedure wall
1040  FOR i=2 TO 6
1050    FOR j=1 TO 35
1060      BEEP 30,j,j*(i/2),1,1
1070      a(i-1,j)=1
1080      INK i
1090      AT i,j
1100      PRINT CHR$(0)
1110    NEXT j
1120  NEXT i
1130 END DEFine
1140 :
1150 DEFine PROCedure update
1160  yy=bx
1170  IF a(xx,yy)=0 THEN GO TO 1330
1180  IF f=1 THEN flag=RND(0 TO 1)
1190  IF flag=1 THEN
1200    byd=-byd
1210    GO TO 1230
1220  END IF
1230  sc=sc+xx
1240  a(xx,yy)=0
1250  BEEP 200,xx,1,1,1
1260  cn=cn+1
1270  IF cn=175 THEN
1280    out
1290    bl=bl+1
1300    GO TO 630
1310  END IF
1320  score
1330 END DEFine
1340 :
1350 DEFine PROCedure score
1360  INK #0,2
1370  AT #0,1,0
1380  PRINT #0;"score :";
1390  INK #0,3
1400  PRINT #0;sc
```

```
1410  AT #0,1,16
1420  INK #0,4
1430  PRINT #0;"balls :";
1440  INK #0,5
1450  PRINT #0;bl
1460 END DEFine
1470 :
1480 bl=bl-1
1490 out
1500 IF bl=0 THEN GO TO 1530
1510 score
1520 GO TO 650
1530 INK #0,2
1540 PRINT #0;" You are out of balls"
1550 INK #0,3
1560 PRINT #0;"play again ?";
1570 INK #0,6
1580 INPUT #0;a$
1590 IF a$="y" THEN RUN
1600 IF a$="n" THEN STOP
1610 GO TO 1550
1620 :
1630 DEFine PROCedure out
1640  BEEP 10000,30,60,1,2,1
1650  AT #0,0,x
1660  PRINT #0;"      "
1670  AT by-byd,bx-bxd
1680  PRINT " "
1690  AT by,bx
1700  PRINT " "
1710  AT 19,0
1720  PRINT FILL$(" ",37)
1730  AT 18,0
1740  PRINT FILL$(" ",37)
1750 END DEFine
```

Colours



This one will really try you out. You have to get through a set of coloured bricks. You can only go if the path is the same colour as the brick at the top. It all gets faster with difficulty level but as well as the colour of the top bricks changing, now they are all different colours.

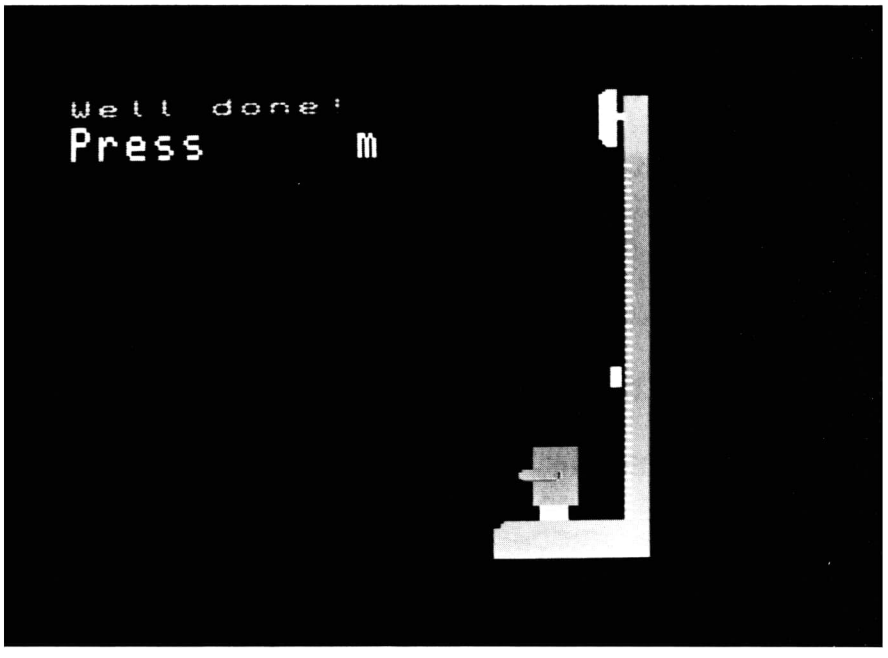
```
100 REMark ***********
110 REMark * Colours *
120 REMark ***********
130 REMark
140 MODE 8
150 PAPER 0
160 CLS
170 CSIZE 3,1
180 INK 6
190 PRINT ,"  Colours"
200 OVER -1
210 AT 0,0
```

```
220 INK 1
230 PRINT ,"   _______"
240 OVER 0
250 INK 5
260 AT 2,0
270 PRINT "  Use the arrow keys to"
280 PRINT "  get to the top.  You"
290 PRINT "  can only climb up if "
300 PRINT "  the top is the same "
310 PRINT "  colour as the square "
320 PRINT "  above you. NOT EASY."
330 INK 2
340 PRINT \"  Press any key to start"
350 PAUSE
360 CSIZE 0,0
370 DIM a(5,11)
380 DIM up(5)
390 level = 0
400 ti = 0
410 PAPER 0
420 CLS
430 FOR i = 5 TO 15
440   FOR j = 0 TO 5
450     a(j,i-5) = 0
460   NEXT j
470   FOR j = 0 TO 5
480     po = RND(0 TO 5)
490     IF a(po,i-5)>0 THEN GO TO 480
500     a(po,i-5) = j+2
510     AT i,po*6
520     PAPER j+2
530     PRINT FILL$(" ",6);
540   NEXT j
550 PRINT
560 NEXT i
570 level = level + 1
580 PAPER 0
590 INK 4
600 AT 18,0
610 PRINT "Level = ";level
```

```
620 count = 1000-level * 50
630 top
640 x = 15 : y = 15
650 AT y,x
660 p=a(INT(x/6),y-6)
670 q=a(INT(x/6),y-5)
680 topcol = up(INT(x/6))
690 PAPER q
700 INK 0
710 PRINT "*"
720 k$ = INKEY$
730 ti = ti + 1
740 count = count - 1
750 IF count = 0 THEN dead
760 AT 18,10
770 PAPER 0
780 INK 7
790 PRINT "Time left = ";count;"    "
800 IF ti >200-level*20  THEN
810 ti = 0
820 top
830 END IF
840 IF k$ = "" THEN GO TO 720
850 PAPER q
860 AT y,x
870 PRINT " "
880 k = CODE(k$)
890 IF k=192 AND x>0 THEN x=x-1
900 IF k=200 AND x<35 THEN x=x+1
910 IF k=208 AND p=topcol THEN y=y-1
920 IF y = 5 THEN
930   BEEP 15000,1,100,1000,1
940   IF BEEPING THEN GO TO 940
950   GO TO 400
960 END IF
970 GO TO 650
980 :
990 DEFine PROCedure top
1000  IF level>5 THEN
1010    FOR i = 0 TO 5:up(i)=0
```

```
1020     FOR i = 0 TO 5
1030       pos = RND (0 TO 5)
1040       IF up(pos)<>0 THEN GO TO 1030
1050       up(pos) = i+2
1060     NEXT i
1070     GO TO 1130
1080   END IF
1090   col = RND (2 TO 7)
1100   FOR i = 0 TO 5
1110     up(i) = col
1120   NEXT i
1130   FOR i = 0 TO 5
1140     PAPER up(i)
1150     AT 0,i*6+1
1160     PRINT "     "
1170   NEXT i
1180 END DEFine
1190 :
1200 DEFine PROCedure dead
1210   BEEP 20000,1,255,1000,2
1220   CLS #0
1230   PRINT #0;"You ran out of time"
1240   STOP
1250 END DEFine
```

Mallet



This one is a fairground reaction test. You are trying to ring the bell and the weight of the mallet on the button depends on your speed in hitting a given character on the keyboard.

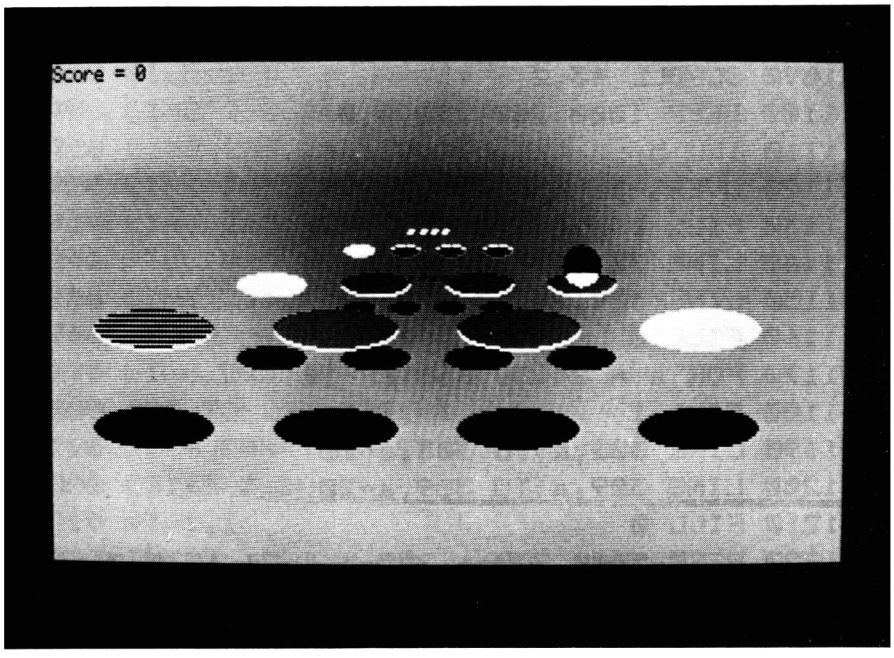
```
100 REMark **********
110 REMark * Mallet *
120 REMark **********
130 REMark
140 MODE 8
150 CSIZE 3,1
160 PAPER 0
170 CLS
180 INK 6
190 PRINT ,"  Mallet"
200 INK 1
210 OVER -1
220 AT 0,0
```

```
230 PRINT ,"  ______"
240 OVER 0
250 INK 5
260 AT 2,0
270 PRINT "  This is based  upon the"
280 PRINT "  old  fair  game to test"
290 PRINT "  your strength. Only now"
300 PRINT "  it  will  test  your"
310 PRINT "        REACTIONS."
320 INK 2
330 PRINT \"  Press any key to start"
340 PAUSE
350 MODE 8
360 PAPER 0
370 CLS
380 SCALE 256,0,0
390 OPEN #3,con
400 WINDOW #3,100,180,235,10
410 INK 6
420 FILL 1
430 LINE 265,32 TO 265,25
440 LINE 280,32 TO 280,25
450 FILL 0
460 INK 1 : FILL 1
470 LINE TO 248,24
480 ARC TO 244,20,PI/2
490 LINE TO 240,20 TO 240,5
500 LINE TO 324,5 TO 324,240
510 LINE TO 310,240 TO 310,24
520 FILL 0
530 FILL 1
540 INK 6
550 ARC 307,245 TO 297,240,PI/2
560 LINE TO 297,220
570 ARC TO 307,215,PI/2
580 LINE TO 307,245
590 FILL 0
600 FILL 1
610 LINE 307,232 TO 307,228
620 LINE 310,232 TO 310,228
```

```
630 FILL 0
640 INK 5
650 FOR n = 28 TO 204 STEP 4
660   LINE 310,n TO 313,n
670 NEXT n
680 INK 4
690 FILL 1
700 LINE 260,240 TO 260,210
710 LINE 285,240 TO 285,210
720 LINE 255,229 TO 255,224
730 LINE 260,229 TO 260,224
740 FILL 0
750 LINE 253,225 TO 253,229
760 INK 2
770 LINE 255,224 TO 270,224
780 ARC TO 273,229,PI
790 a$ = CHR$(RND(97 TO 122))
800 CSIZE 3,1
810 AT 1,1
820 INK 7
830 PRINT "Get ready"
840 AT 1,1
850 FOR i = 1 TO RND (200 TO 400)
860 NEXT i
870 PRINT "Press      ";
880 PRINT a$
890 FOR n = 300 TO 20 STEP -1
900 g$ = INKEY$
910 IF g$<>"" AND g$<>a$ THEN GO TO 940
920 IF g$ = a$ THEN woopee
930 NEXT n
940 FOR n = 1 TO 17
950   SCROLL #3,8
960   PAN #3,-2
970   BEEP 2000,n*2,n*3,4,45,15,4,0
980 NEXT n
990 CSIZE 3,0
1000 AT 1,1
1010 INK 2,6,3
1020 PRINT "Yah missed it!"
```

```
1030 GO TO 1390
1040 DEFine PROCedure woopee
1050 FOR 1 = 1 TO 34
1060   SCROLL #3,4
1070   BEEP 2000,1*2,10+1*2,4,23,15,0,0
1080 NEXT 1
1090 SCROLL #3,2
1100 BEEP 1000,200,220,0,0
1110 INK 5
1120 OVER -1
1130 FILL 1
1140 LINE 303,24 TO 303,34
1150 LINE 309,24 TO 309,34
1160 FILL 0
1170 FOR a = 24 TO n STEP 10
1180 FILL 1
1190 LINE 303,a TO 303,a+20
1200 LINE 309,a TO 309,a+20
1210 FILL 0
1220 BEEP 5000,250-a,255-a,4,24,15,0,0
1230 IF a >= 185 THEN GO TO 1260
1240 NEXT a
1250 GO TO 1310
1260 BEEP 30000,10,15,4,1000
1270 CSIZE 3,0
1280 INK 2,6,3
1290 AT 1,1
1300 PRINT "Well done!"
1310 INK 5
1320 OVER -1
1330 FOR n = a TO 24 STEP -10
1340 FILL 1
1350 LINE 303,n TO 303,n+20
1360 LINE 309,n TO 309,n+20
1370 FILL 0
1380 NEXT n
1390 INPUT #0;"play again ? ";p$
1400 IF p$ = "" THEN GO TO 1390
1410 IF p$ == "n" THEN STOP
1420 IF p$ == "y" THEN GO TO 350
1430 GO TO 1390
```

Discs



Not so easy. You are a frog jumping on lily-pads trying to avoid a French monster. A leaf flips from red to white when you jump onto it. When you have changed all the leaves, you get points. Then you try to change them back again, and again, and again. Get the highest score before you are finally devoured. We got 960.

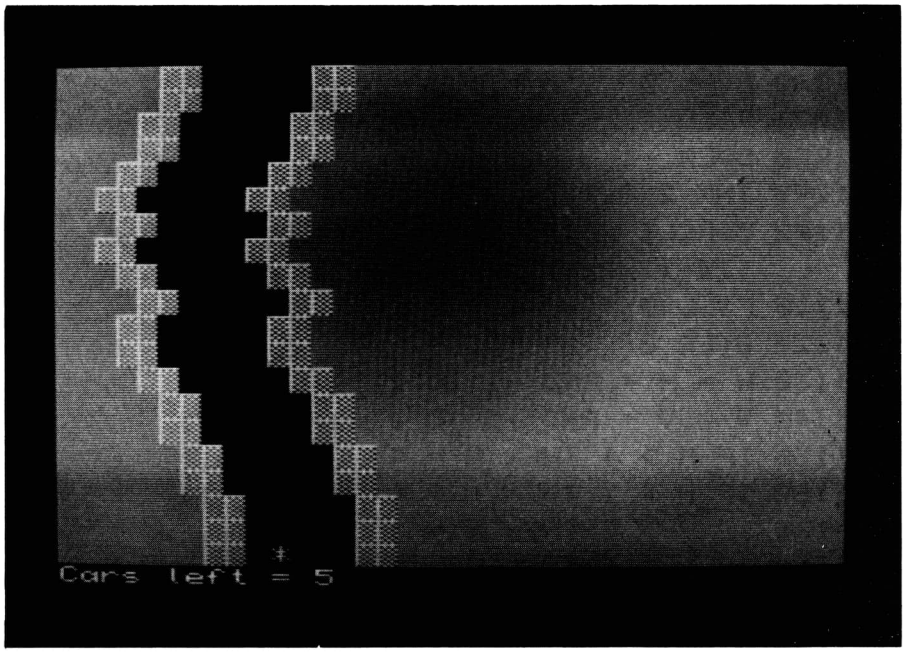
```
100 REMark *********
110 REMark * Discs *
120 REMark *********
130 REMark
140 MODE 8
150 INK 6
160 PAPER 0
170 CSIZE 3,1
180 CLS
190 PRINT ,"  Discs"
200 OVER -1
```

```
210 INK 1
220 AT 0,0
230 PRINT ,"  _____"
240 OVER 0
250 INK 5
260 AT 2,0
270 PRINT "  Try to work this out."
280 PRINT "  Use the arrow keys to"
290 PRINT "  get around. But look"
300 PRINT "  I am out to get you!"
310 INK 2
320 PRINT \"  Press any key to start"
330 PAUSE
340 CSIZE 0,0
350 MODE 4
360 PAPER 4
370 CLS
380 DIM S(4,4)
390 CO = 0
400 SCALE 256 ,0 ,0
410 FOR 1 = 0 TO 7 STEP 7
420 INK 1
430 FOR n = 1 TO 4
440   y = (150+3*1)-(5-(1/4))*(n^2)
450   FOR a = -6 TO 6 STEP 4
460     x = 1.5 * a * (n^2)
470     FILL 1
480     CIRCLE 200+x,y,2*(n^2),.3,PI/2
490     FILL 0
500   NEXT a
510 NEXT n
520 NEXT 1
530 V = 0
540 x = 1
550 y = 4
560 H = RND(1 TO 4)
570 SC = 0
580 AT 0,0
590 INK 0
600 PRINT "Score = ";SC
```

```
610 yp = (169+y*3)-3.1*(y^2)
620 xp = 200+1.5*((-10+4*x))*(y^2)
630 FILL 1
640 OVER -1
650 INK 4
660 q = y^2
670 CIRCLE xp,172-(3.25*q),2*q,.3,PI/2
680 FILL 0
690 OVER 0
700 S(y,x) = 1-S(y,x)
710 CO = (CO-1)+2*S(y,x)
720 IF CO = 16 THEN
730   SC = SC + 160
740   AT 0,0
750   INK 0
760   PRINT "Score = ";SC
770   CO = 0
780   FOR n = 1 TO 4
790     FOR a = 1 TO 4
800       S(n,a) = 0
810       BEEP 100,a*n,3*a-n,23,45,0,0,0
820     NEXT a
830   NEXT n
840 END IF
850 FILL 1
860 OVER -1
870 INK 2,4,0
880 CIRCLE 200+1.5*((-10+4*H))*(V^2),
    172-(3.25*(V^2)),2*(V^2),.3,PI/2
890 FILL 0
900 OVER 0
910 INK 4
920 OVER -1
930 FILL 1
940 CIRCLE xp,yp,1+(y^2)
950 FILL 0
960 IF V = y AND H = x THEN
970   BEEP 30000,200,150,260,1000,0,0,0
980   GO TO 1230
990 END IF
```

```
1000 key$ = INKEY$
1010 IF key$ = "" THEN GO TO 1000
1020 INK 4
1030 FILL 1
1040 CIRCLE xp,yp,1+(y^2)
1050 OVER 0
1060 FILL 0
1070 FILL 1
1080 OVER -1
1090 INK 2,4,0
1100 CIRCLE 200+1.5*((-10+4*H))*(V^2),
     172-(3.25*(V^2)),2*(V^2),.3,PI/2
1110 FILL 0
1120 OVER 0
1130 H = H + RND(-1 TO 1)
1140 V = V + RND(-1 TO 1)
1150 IF H > 4 THEN H = 4
1160 IF H < 1 THEN H = 1
1170 IF V > 4 THEN V = 4
1180 IF V < 1 THEN V = 1
1190 k = CODE (key$)
1200 x=x+(k=200 AND x<4)-(k=192 AND x>1)
1210 y=y+(k=216 AND y<4)-(k=208 AND y>1)
1220 GO TO 610
1230 STOP
```

Formula 1



Have you got what it takes to become a top racing driver? Zip round the course as fast as you can and as the difficulty level goes up so does your speed - but the road gets narrower. The controls are ← and →.

```
100 REMark **************
110 REMark * Formula_1 *
120 REMark **************
130 REMark
140 MODE 8
150 PAPER 0
160 CLS
170 INK 4
180 CSIZE 3,1
190 AT 0,9
200 PRINT "Formula 1"
210 AT 0,9
220 INK 1
```

```
230 OVER -1
240 PRINT "_________"
250 OVER 0
260 INK 6
270 AT 2,5
280 PRINT "Use left and right"
290 AT 3,5
300 PRINT "arrow keys to move"
310 AT 4,5
320 PRINT "your car,Avoid the"
330 AT 5,5
340 PRINT "edge of the track."
350 AT 8,3
360 INK 2
370 PRINT "Press any key to start"
380 PAUSE
390 CLS
400 AT 3,0
410 INPUT "Difficulty level (1-5)? ";d$
420 IF d$ = "" THEN GO TO 390
430 IF d$<1 OR d$> 5 THEN GO TO 390
440 CLS
450 CSIZE 2,0
460 wall$ = CHR$(0)&CHR$(0)
470 track$ = FILL$(" ",8-d$)
480 metres = 0
490 cars = 5
500 across = 10
510 car = 15
520 CLS #0
530 AT #0,0,0
540 PRINT #0,"Cars left = ";cars
550 AT 0,0
560 PRINT FILL$(CHR$(0),37);
570 :
580 AT 0,across
590 INK 2
600 PAPER 6
610 PRINT wall$;
620 PAPER 0
```

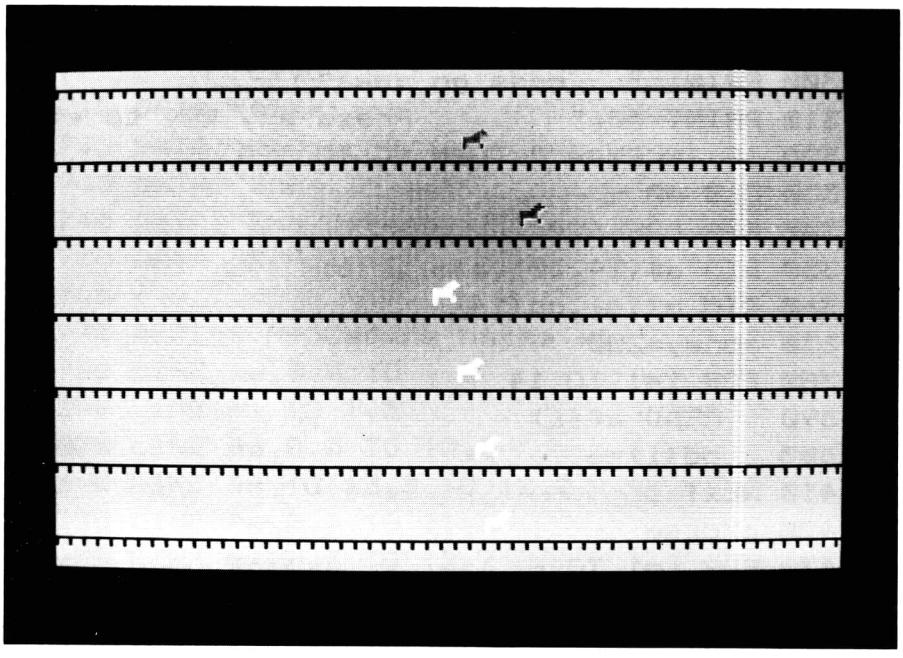
```
630 PRINT track$;
640 PAPER 6
650 PRINT wall$
660 PAPER 4
670 SCROLL 10
680 AT 19,car
690 INK 6
700 PAPER 0
710 PRINT "*"
720 :
730 key = KEYROW (1)
740 :
750 IF key = 2 THEN
760   IF car > 0 THEN
770     car = car - 1
780   END IF
790 END IF
800 :
810 IF key = 16 THEN
820   IF car < 36 THEN
830     car = car + 1
840   END IF
850 END IF
860 :
870 IF PEEK(157064+car*3.1) <> 0 THEN
880   AT 18,car-3
890   PRINT "Crashed";
900   BEEP 10000,1,100,1000,2
910   IF BEEPING THEN GO TO 910
920   cars = cars -1
930   IF cars = 0 THEN game_over
940   CLS #0
950   AT #0,0,0
960   PRINT #0,"Cars left = ";cars
970 END IF
980 :
990 movement = RND (-1 TO 1)
1000 metres = metres + 1
1010 :
1020 IF movement = -1 THEN
```

```
1030    IF across > 1 THEN
1040      across = across - 1
1050    END IF
1060  END IF
1070  :
1080  IF movement = 1 THEN
1090    IF 34-(across+LEN(track$)) > 1 THEN
1100      across = across + 1
1110    END IF
1120  END IF
1130  :
1140  GO TO 580
1150  :
1160  DEFine PROCedure game_over
1170   CSIZE 3,1
1180    FOR i = 0 TO 7
1190      AT i,9
1200      PAPER i+1
1210      INK i+3
1220      PRINT "Game Over"
1230    NEXT i
1240   CLS #0
1250   CSIZE #0,3,1
1260   PRINT #0,"         Another go ?"
1270  CSIZE #0,0,0
1280  AT #0,3,8
1290  PRINT #0,"Miles travelled = ";
      INT(metres/16)
1300   a$ = INKEY$
1310   IF a$ = "" THEN GO TO 1300
1320   IF a$ == "y" THEN RUN
1330   IF a$ == "n" THEN STOP
1340   GO TO 1300
1350  END DEFine
```

Chapter 7

TRADITIONAL GAMES

Horse Race



A chance to earn some money from your friends. Back your favourite depending on its form and odds.

```
100 REMark ***************
110 REMark * horse race *
120 REMark ***************
130 MODE 8
140 CSIZE 3,1
150 PAPER 0
160 CLS
170 INK 2
180 AT 0,0
190 PRINT "        HORSE RACE"
200 PRINT "        ==========="
210 INK 6
220 PRINT "    Bet on horse using"
230 PRINT "    the odds provided"
```

```
240 PRINT "    Then enter amount to"
250 PRINT "    bet"
260 INK 3
270 PRINT "    If a horse looses "
280 PRINT "    odds go up"
290 PRINT "    If a horse wins"
300 PRINT "    odds go down"
310 PRINT #0;"          Press any key to start"
320 IF INKEY$="" THEN GO TO 320
330 CLS #0
340 RESTORE 620
350 DIM a$(14,20),a(6),r(6)
360 DIM b(14),od(14)
370 money=200
380 FOR i=1 TO 14
390   READ a$(i)
400   od(i)=6
410 NEXT i
420 choose
430 FOR i=1 TO 6
440   a(i)=0
450 NEXT i
460 INK 148
470 PRINT
480 PRINT"Which horse to bet on 1-6 ?";
490 INPUT h
500 IF h<1 OR h>6 THEN h=RND(1 TO 6)
510 CSIZE 0,1
520 PRINT "Horse ";h;" choosen, ";
530 PRINT "You have `";money
540 PRINT"How much to bet 1 to 25 ?";
550 INPUT bet
560 IF bet<0 OR bet>25THEN GO TO 540
570 IF money-bet<0 THEN GO TO 540
580 CSIZE 0,0
590 PAPER 4
600 INK 0
610 CLS
620 DATA "Discovery"
630 DATA "Tubular Bells"
```

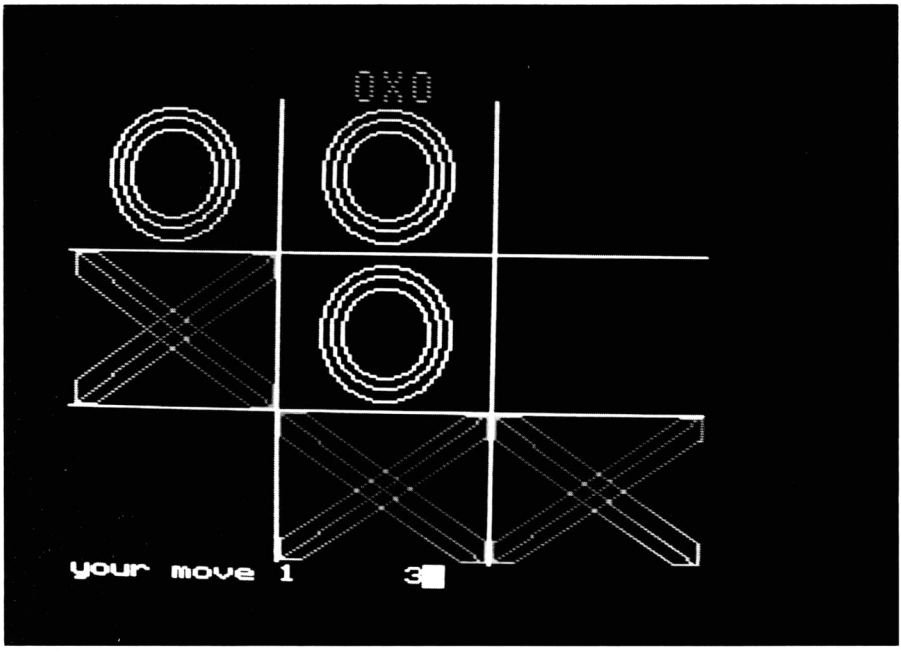
```
640 DATA "Flash  AHHH!"
650 DATA "The N Team"
660 DATA "Old Smokey"
670 DATA "Red Rum"
680 DATA "Vector Twin Turbo"
690 DATA "Ferrari"
700 DATA "Oxygene"
710 DATA "Atarian"
720 DATA "Newtech"
730 DATA "Bone Shaker"
740 DATA "The Defender"
750 DATA "Sinclair"
760 :
770 DEFine PROCedure horse(w)
780 LINE #w,0,0 TO 1,2
790 LINE #w TO 1,5
800 LINE #w,0,5 TO 10,5
810 LINE #w TO 12,3 TO 12,0
820 LINE #w,0,6 TO 11,6
830 LINE #w,0,7 TO 12,7
840 LINE #w,6,8 TO 12,8
850 LINE #w,8,9 TO 13,9
860 LINE #w,10,10 TO 14,10
870 LINE #w,11,11 TO 16,11
880 LINE #w,14,10 TO 17,10
890 LINE #w,16,9 TO 17,9
900 LINE #w,12,12 TO 12,13
910 END DEFine
920 :
930 c=0
940 FOR i=1 TO 6
950   OPEN #2+i,con
960   c=c+1
970   IF c=4 THEN c=5
980   INK #2+i,c
990   WINDOW #2+i,20,20,100,i*30
1000   SCALE #2+i,28,0,0
1010   PAPER #2+i,4
1020   CLS #2+i
1030   horse i+2
```

```
1040 NEXT i
1050 :
1060 DEFine PROCedure railings
1070  BLOCK #2,8,200,380,0,60
1080 INK #2,0
1090  FOR i=6 TO 115 STEP 15
1100    LINE #2,0,i TO 170,i
1110    FOR j=0 TO 170 STEP 3
1120      LINE #2,j,i TO j,i-2
1130    NEXT j
1140  NEXT i
1150 END DEFine
1160 :
1170 railings
1180 REMark ***race
1190 FOR i=1 TO 6
1200   x=RND (1 TO 3)
1210   x=x*2
1220   a(i)=a(i)+x
1230   IF a(i)=300 THEN GO TO 1580
1240   PAN #2+i,x
1250   WINDOW #2+i,20,20,100+a(i),i*30
1260 NEXT i
1270 GO TO 1190
1280 :
1290 DEFine PROCedure choose
1300  PAPER #2,0
1310  CLS #2
1320  CSIZE 0,0
1330  PAPER 0
1340  INK 6
1350  CLS
1360  FOR i=1 TO 14
1370    b(i)=0
1380  NEXT i
1390  FOR i=1 TO 6
1400    x=INT(RND(1 TO 14))
1410    IF b(x)=1 THEN GO TO 1400
1420    b(x)=1
1430    r(i)=x
```

```
1440  NEXT i
1450  CLS #2
1460  c=0
1470  FOR i = 1 TO 6
1480    c=c+1
1490    IF c=4 THEN c=5
1500    INK c
1510    AT i*2,1
1520    PRINT i;" ";a$(r(i));
1530    AT i*2,22
1540    PRINT od(r(i));"-1"
1550  NEXT i
1560 END DEFine
1570 :
1580 REMark *** end of race
1590 CLS #2
1600 PAPER 0
1610 INK 5
1620 CLS
1630 CSIZE 3,1
1640 AT 0,0
1650 PRINT a$(r(i));" Won"
1660 PRINT "You bet on ";a$(r(h))
1670 IF h=i THEN
1680   PRINT "You Won ";od(r(i))*bet
1690 END IF
1700 IF h=i THEN
1710 money=money+od(r(i))*bet
1720 END IF
1730 money=money-bet
1740 IF money <=0 THEN
1750   PRINT "Out of money"
1760   STOP
1770 END IF
1780 FOR j=1 TO 6
1790   od(r(j))=od(r(j))+1
1800 NEXT j
1810 od(r(i))=od(r(i))-2
1820 IF od(r(i))<1 THEN od(r(i))=1
1830 INPUT"Race again ?";d$
```

```
1840 IF d$="y" THEN GO TO 420
1850 IF d$<>"n" THEN GO TO 1830
1860 STOP
```

Os and Xs



The moves are numbered by a simple grid reference, i.e. column and row. You decide on who has the first move. If your move is illegal, the machine tells you. Try to beat the machine without a draw (stalemate).

```
100 REMark *********
110 REMark * o x o *
120 REMark *********
130 MODE 8
140 PAPER #0,0
150 CLS #0
160 PAPER 0
170 CLS 0
180 CSIZE 3,1
190 AT 0,0
200 INK 5
210 PRINT "           OXO"
220 PRINT "           ---"
```

```
230 PRINT
240 INK 6
250 PRINT "    Try to beat me"
260 PRINT "    at Noughts and"
270 PRINT "    Crosses."
280 INK 2
290 PRINT "  Enter Co-Ordinates"
300 PRINT "  for move  E.G. 1,3"
310 PRINT "  (ALONG,UP)"
320 INK 4
330 PRINT "Press any key to start"
340 IF INKEY$="" THEN  GO TO 340
350 CLS
360 SCALE 225,0,0
370 PAPER 0
380 CLS
390 g=0
400 CM=1
410 PL=2
420 DIM A$(1),C$(1)
430 INK 73
440 CSIZE 3,1
450 PRINT "           OXO"
460 CSIZE 0,0
470 DIM a(3,3)
480 FOR i=1 TO 3
490   FOR j=1 TO 3
500     a(i,j)=0
510   NEXT j
520 NEXT i
530 board
540 CLS #0
550 PRINT #0;"NOUGHTS OR CROSSES ?";
560 INPUT #0,C$
570 IF C$="" THEN GO TO 540
580 CLS #0
590 PRINT #0;"DO YOU WANT TO GO FIRST ?";
600 INPUT #0,A$
610 IF A$="" THEN GO TO 580
620 CM=1
```

```
630 PL=2
640 IF C$=="c" THEN
650   CM=2
660   PL=1
670 END IF
680 IF A$="N" OR A$="n" THEN GO TO 860
690 CLS #0
700 INPUT #0,"your move ";b,c
710 IF c<=0 OR c>=4 OR b<=0 OR b>=4
     THEN GO TO 690
720 IF a(c,b)=0 THEN
730   a(c,b)=1
740   GO TO 780
750 END IF
760 PRINT #0;"illegal move"
770 GO TO 700
780 CLS #0
790 g=g+1
800 IF g=9 THEN
810   win
820   GO TO 2230
830 END IF
840 win
850 board
860 WINSCAN
870 IF f=0 THEN iwin
880 IF f=1 THEN GO TO 900
890 ranm
900 g=g+1
910 IF g=9 THEN
920   win
930   GO TO 2220
940 END IF
950 board
960 win
970 GO TO 700
980 :
990 DEFine PROCedure iwin
1000 f=0
1010 FOR j=1 TO 3
```

```
1020 IF a(1,j)=1AND a(2,j)=1AND a(3,j)=0
     THEN a(3,j)=2:GO TO 1170
1030 IF a(1,j)=1AND a(2,j)=0AND a(3,j)=1
     THEN a(2,j)=2:GO TO 1170
1040 IF a(1,j)=0AND a(2,j)=1AND a(3,j)=1
     THEN a(1,j)=2:GO TO 1170
1050 IF a(j,1)=1AND a(j,2)=1AND a(j,3)=0
     THEN a(j,3)=2:GO TO 1170
1060 IF a(j,1)=1AND a(j,2)=0AND a(j,3)=1
     THEN a(j,2)=2:GO TO 1170
1070 IF a(j,1)=0AND a(j,2)=1AND a(j,3)=1
     THEN a(j,1)=2:GO TO 1170
1080 NEXT j
1090 IF a(1,1)=1AND a(2,2)=1AND a(3,3)=0
     THEN a(3,3)=2:GO TO 1170
1100 IF a(1,1)=0AND a(2,2)=1AND a(3,3)=1
     THEN a(1,1)=2:GO TO 1170
1110 IF a(1,1)=1AND a(2,2)=0AND a(3,3)=1
     THEN a(2,2)=2:GO TO 1170
1120 REMark
1130 IF a(1,3)=1AND a(2,2)=1AND a(3,1)=0
     THEN a(3,1)=2:GO TO 1170
1140 IF a(1,3)=1AND a(3,1)=1AND a(2,2)=0
     THEN a(2,2)=2:GO TO 1170
1150 IF a(2,2)=1AND a(3,1)=1AND a(1,3)=0
     THEN a(1,3)=2:GO TO 1170
1160 GO TO 1180
1170 f=1
1180 END DEFine
1190 :
1200 DEFine PROCedure WINSCAN
1210 f=0
1220 FOR j=1 TO 3
1230 IF a(1,j)=2AND a(2,j)=2AND a(3,j)=0
     THEN a(3,j)=2:GO TO 1370
1240 IF a(1,j)=2AND a(3,j)=2AND a(2,j)=0
     THEN a(2,j)=2:GO TO 1370
1250 IF a(2,j)=2AND a(3,j)=2AND a(1,j)=0
     THEN a(1,j)=2:GO TO 1370
1260 IF a(j,1)=2AND a(j,2)=2AND a(j,3)=0
     THEN a(j,3)=2:GO TO 1370
```

```
1270 IF a(j,1)=2AND a(j,3)=2AND a(j,2)=0
     THEN a(j,2)=2:GO TO 1370
1280 IF a(j,3)=2AND a(j,2)=2AND a(j,1)=0
     THEN a(j,1)=2:GO TO 1370
1290 NEXT j
1300 IF a(1,1)=2AND a(2,2)=2AND a(3,3)=0
     THEN a(3,3)=2:GO TO 1370
1310 IF a(1,1)=2AND a(2,2)=0AND a(3,3)=2
     THEN a(2,2)=1:GO TO 1370
1320 IF a(1,1)=0AND a(2,2)=2AND a(3,3)=2
     THEN a(1,1)=2:GO TO 1370
1330 IF a(3,1)=2AND a(2,2)=2AND a(1,3)=0
     THEN a(1,3)=2:GO TO 1370
1340 IF a(3,1)=2AND a(2,2)=0AND a(1,3)=2
     THEN a(2,2)=2:GO TO 1370
1350 IF a(3,1)=0AND a(2,2)=2AND a(1,3)=2
     THEN a(3,1)=2:GO TO 1370
1360 GO TO 1380
1370 f=1
1380 END DEFine
1390 :
1400 DEFine PROCedure board
1410  INK 5
1420  FOR i=100 TO 200 STEP 100
1430    LINE i,0 TO i,210
1440  NEXT i
1450  FOR i=70 TO 140 STEP 70
1460    LINE 0,i TO 300,i
1470  NEXT i
1480  FOR i=1 TO 3
1490    FOR j=1 TO 3
1500      cc=a(j,i)
1510      IF cc=CM THEN dcirc
1520      IF cc=PL THEN dcross
1530    NEXT j
1540  NEXT i
1550 END DEFine
1560 :
1570 DEFine PROCedure dcirc
1580  INK 6
1590  CIRCLE (i*100)-50,(j*70)-35,30
```

```
1600  CIRCLE (i*100)-50,(j*70)-35,20
1610  CIRCLE (i*100)-50,(j*70)-35,25
1620 END DEFine
1630 :
1640 DEFine PROCedure dcross
1650  p1=(i*100)-97
1660  p2=(j*70)-69
1670  p3=(i*100)-1
1680  p4=(j*70)-1
1690  INK 2
1700  LINE p1,p2 TO p1+10,p2
1710  LINE TO p3,p4-10 TO p3,p4
1720  LINE TO p3-10,p4 TO p1,p2+10
1730  LINE TO p1,p2
1740  LINE p1,p2 TO p3,p4
1750  LINE p1,p4 TO p1,p4-10
1760  LINE TO p3-10,p2 TO p3,p2
1770  LINE TO p3,p2+10 TO p1+10,p4
1780  LINE TO p1,p4
1790  LINE p1,p4 TO p3,p2
1800 END DEFine
1810 :
1820 DEFine PROCedure ranm
1830  x=RND(1 TO 3)
1840  y=RND(1 TO 3)
1850  IF a(x,y)<>0 THEN GO TO 1830
1860  a(x,y)=2
1870  f=1
1880 END DEFine
1890 :
1900 DEFine PROCedure win
1910  FOR j=1 TO 3
1920 IF a(1,j)=1AND a(2,j)=1AND a(3,j)=1
   THEN GO TO 2030
1930 IF a(1,j)=2AND a(2,j)=2AND a(3,j)=2
   THEN GO TO 2110
1940 IF a(j,1)=1AND a(j,2)=1AND a(j,3)=1
   THEN GO TO 2030
1950 IF a(j,1)=2AND a(j,2)=2AND a(j,3)=2
   THEN GO TO 2110
```

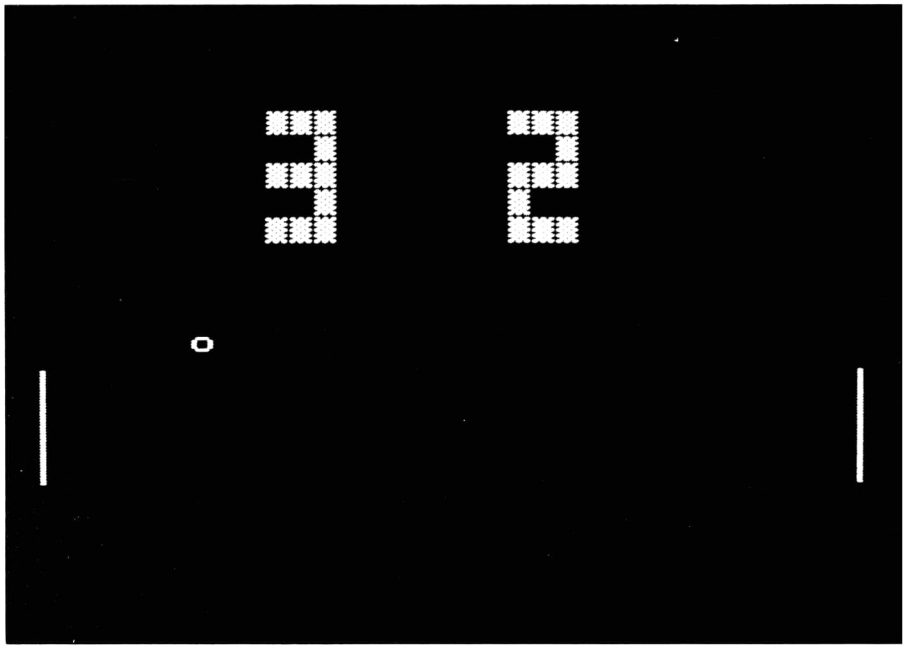
```
1960 NEXT j
1970 IF a(1,1)=1AND a(2,2)=1AND a(3,3)=1
   THEN GO TO 2030
1980 IF a(3,1)=1AND a(2,2)=1AND a(1,3)=1
   THEN GO TO 2030
1990 IF a(1,1)=2AND a(2,2)=2AND a(3,3)=2
   THEN GO TO 2110
2000 IF a(3,1)=2AND a(2,2)=2AND a(1,3)=2
   THEN GO TO 2110
2010 END DEFine
2020 :
2030 MODE 8
2040 PAPER 5
2050 INK 1
2060 CSIZE 3,1
2070 CLS
2080 AT 4,9
2090 PRINT "you won"
2100 GO TO 2300
2110 INK 1
2120 BEEP 0,100,200,2,3,4
2130 MODE 8
2140 CSIZE 3,1
2150 FOR i=1 TO 255 STEP 2
2160   INK i
2170   AT 4,10
2180   PRINT "i won"
2190 NEXT i
2200 BEEP
2210 GO TO 2300
2220 MODE 8
2230 PAPER 4
2240 INK 2
2250 CLS
2260 CSIZE 3,1
2270 AT 4,8
2280 PRINT "stalemate"
2290 GO TO 2300
2300 CSIZE 1,1
2310 INPUT #0;"play again ?";A$
```

```
2320 IF A$=="y" THEN RUN
2330 IF A$<>"n" THEN STOP
2340 CLS #0
2350 GO TO 2310
```

Chapter 8

TWO-USER GAMES

Tennis



Nothing fancy. Just the traditional computer tennis. You control the bats with F4 and F5 (left-hand player) and ↑ ↓ (right-hand player). The first player to 9 wins.

```
100 REMark **********
110 REMark * Tennis *
120 REMark **********
130 REMark
140 MODE 8
150 PAPER 0
160 CLS
170 CSIZE 3,1
180 INK 6
190 PRINT ,"Tennis"
200 OVER -1
210 INK 1
220 AT 0,0
230 PRINT ,"______"
240 AT 2,0
```

```
250 INK 5
260 PRINT " Nothing special,  just "
270 PRINT "the old blip/blop tennis."
280 PRINT "  Player 1   Player 2"
290 INK 6
300 AT 4,0
310 PRINT "  ______ _   ______ _"
320 OVER 0
330 INK 5
340 PRINT\" f4  <UP>    up arrow"
350 PRINT " f5 <down>  down arrow"
360 INK 2
370 set_score
380 PRINT\" Press any key to play"
390 PAUSE
400 score_left = 0
410 score_right = 0
420 CSIZE 2,0
430 PAPER 0
440 OVER -1
450 INK 4
460 CLS
470 p 1,0
480 p 3,0
490 OPEN #3,con
500 OPEN #4,con
510 WINDOW #3,2,256,40,0
520 WINDOW #4,2,256,453,0
530 PAPER #3,0
540 PAPER #4,0
550 CLS #3
560 CLS #4
570 PAPER #3,4
580 PAPER #4,4
590 FOR i = 1 TO 20
600   SCROLL #3,2
610   SCROLL #4,2
620 END FOR i
630 PAPER #3,0
640 PAPER #4,0
650 left = 0
660 right = 0
670 FOR i = 1 TO 10
680   SCROLL #3,10
690   SCROLL #4,10
700   left = left + 1
710   right = right + 1
```

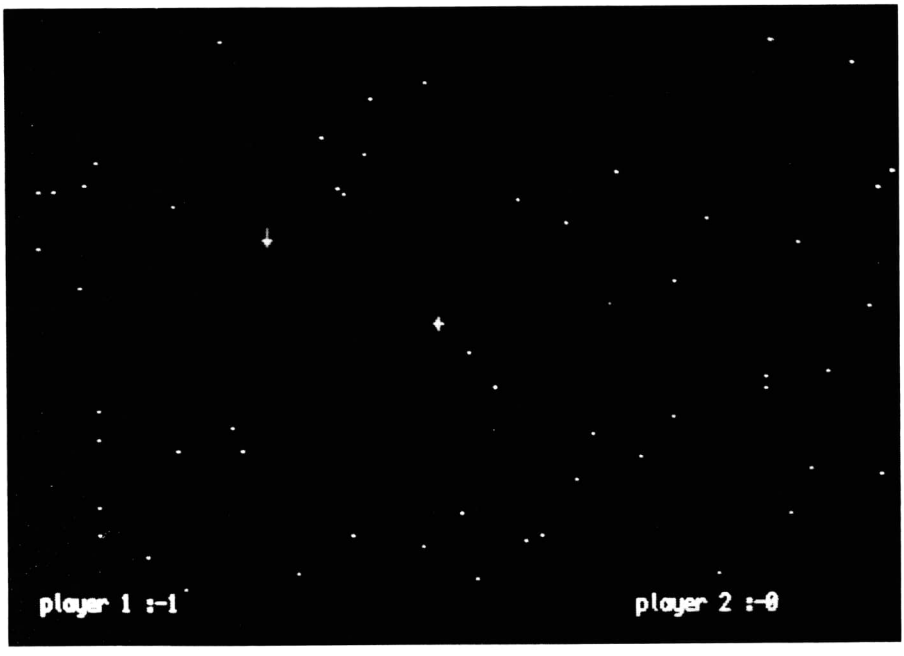
```
720 END FOR i
730 balls = 15
740 ball_x = 10+RND(1 TO 5)
750 ball_y = 15
760 AT ball_y,ball_x
770 PRINT "o"
780 q = RND (-1 TO 1)
790 IF q=0 THEN GO TO 780
800 ball_x_dir = q
810 q = RND (-1 TO 1)
820 IF q=0 THEN GO TO 810
830 ball_y_dir = q
840 temp_x = ball_x
850 temp_y = ball_y
860 IF ball_x_dir = 1 THEN
870   move_right
880   IF ball_x + 1 < 35 THEN
890     ball_x = ball_x + 1
900     GO TO 930
910   END IF
920   check_right
930 END IF
940 IF ball_x_dir = -1 THEN
950   move_left
960   IF ball_x - 1 > 0 THEN
970     ball_x = ball_x - 1
980     GO TO 1010
990   END IF
1000   check_left
1010 END IF
1020 IF ball_y_dir = -1 THEN
1030   IF ball_y - 1 > -1 THEN
1040     ball_y = ball_y -1
1050     GO TO 1100
1060   END IF
1070   ball_y_dir = 1
1080   BEEP 100,10
1090   GO TO 1190
1100 END IF
1110 IF ball_y_dir = 1 THEN
1120   IF ball_y + 1 < 20 THEN
1130     ball_y = ball_y + 1
1140     GO TO 1180
1150   END IF
1160   ball_y_dir = -1
1170   BEEP 100,10
1180 END IF
```

```
1190 AT temp_y,temp_x
1200 PRINT "o"
1210 AT ball_y,ball_x
1220 PRINT "o"
1230 GO TO 840
1240 DEFine PROCedure move_left
1250  k=KEYROW(0)
1260  IF k=0 THEN GO TO 1390
1270  IF k=1 THEN
1280    IF left > 1 THEN
1290      SCROLL #3,-10
1300      left = left -1
1310    END IF
1320  END IF
1330  IF k=32 THEN
1340    IF left < 20 THEN
1350      SCROLL #3,10
1360      left = left +1
1370    END IF
1380  END IF
1390 END DEFine
1400 DEFine PROCedure move_right
1410  k=KEYROW(1)
1420  IF k=0 THEN GO TO 1550
1430  IF k=4 THEN
1440    IF right > 1 THEN
1450      SCROLL #4,-10
1460      right = right -1
1470    END IF
1480  END IF
1490  IF k=128 THEN
1500    IF right < 20 THEN
1510      SCROLL #4,10
1520      right = right +1
1530    END IF
1540  END IF
1550 END DEFine
1560 DEFine PROCedure set_score
1570  DIM sc$(9,15)
1580  b$ = CHR$(0)
1590  RESTORE
1600  FOR i = 0 TO 9
1610    READ a$
1620    FOR j = 1 TO 15
1630      IF a$(j)="*" THEN a$(j)=b$
1640    NEXT j
1650    sc$(i)=a$
```

```
1660  NEXT i
1670  DATA "**** ** ** ****"
1680  DATA " * **  *  * ***"
1690  DATA "***  *****  ***"
1700  DATA "***  ****  ****"
1710  DATA "* ** ****  *  *"
1720  DATA "****  ***  ****"
1730  DATA "****  **** ****"
1740  DATA "***  *  * * *  "
1750  DATA "**** ***** ****"
1760  DATA "**** ****  ****"
1770 END DEFine
1780 DEFine PROCedure p(pos,sco)
1790  OVER 0
1800  PAPER 0
1810  INK 2
1820  score$ = sc$(sco)
1830  FOR i = 0 TO 4
1840    AT i,5*pos+5
1850    PRINT score$(i*3+1 TO i*3+3)
1860  NEXT i
1870  OVER -1
1880  INK 4
1890  IF sco = 9 THEN
1900    CLS #0
1910    CSIZE #0,3,1
1920    PRINT #0,,"GAME OVER"
1930    STOP
1940  END IF
1950 END DEFine
1960 DEFine PROCedure check_left
1970  hit = 0
1980  test = ball_y - left
1990  IF ABS(test) < 2 THEN hit = 1
2000  IF hit = 0 THEN
2010    score_right = score_right +1
2020    p 3,score_right
2030  END IF
2040    ball_x_dir = -ball_x_dir
2050    BEEP 100,10
2060 END DEFine
2070 DEFine PROCedure check_right
2080  hit = 0
2090  test = ball_y - right
2100  IF ABS(test) < 2 THEN hit = 1
2110  IF hit = 0 THEN
2120   score_left = score_left +1
```

```
2130    p 1,score_left
2140   END IF
2150   ball_x_dir = -ball_x_dir
2160   BEEP 100,10
2170 END DEFine
```

Space War



A battle in space between two missile launchers, both difficult to destroy. You must register 10 hits to finish off an opponent. The left-hand player uses F1–F3 for left, right and fire, while the right-hand player uses £], for left, right and fire.

```
100 REMark **************
110 REMark * Space War *
120 REMark **************
130 REMark
140 MODE 8 : PAPER 0 : CLS
150 CSIZE 3,1 : INK 2 : AT 0,8
160 FLASH 1
170 PRINT "Space War"
180 FLASH 0 : OVER -1 : INK 1
190 AT 0,8
200 PRINT "_________"
210 INK 6
220 PRINT \"Controls"
230 INK 3
240 PRINT "Player 1        Player 2"
```

```
250 PRINT "f1  left            left"
260 PRINT "f2  right       ]   right"
270 PRINT "f3  fire        ,   fire"
280 INK 7
290 PRINT " Press any key to start"
300 IF INKEY$="" THEN GO TO 300
310 MODE 4 : INK 7
320 FOR i=1 TO 100
330   POINT RND(0 TO 300),RND(0 TO 100)
340 NEXT i
350 RESTORE
360 PAPER 0
370 DIM dx(8),dy(8),sh$(8)
380 d1=3 : d2=7 : x1=0 : y1=10
390 x2=73 : y2=10 : f1=0 : f2=0
400 s1=0 : s2=0
410 FOR i=1 TO 8
420   READ dx(i),dy(i)
430 NEXT i
440 FOR i=1 TO 8
450   READ sh$(i)
460 NEXT i
470 move_player1
480 move_player2
490 IF RND(0 TO 30)>29 THEN
500   POINT RND(0 TO 300),RND(0 TO 100)
510 END IF
520 GO TO 470
530 DEFine PROCedure move_player1
540  q = KEYROW(0)
550  IF q = 2 THEN
560    d1=d1+1
570    IF d1=9 THEN d1=1
580  END IF
590  IF q = 8 THEN
600    d1=d1-1
610    IF d1=0 THEN d1=8
620  END IF
630 IF (q=16OR q=24OR q=18)AND f1=0THEN
640    f1=1
650    fx1=x1
660    fy1=y1
670    fxd1=dx(d1)
680    fyd1=dy(d1)
690  END IF
700  IF f1=1 THEN
710    fire1
```

```
720     fire1
730   END IF
740   AT y1,x1
750   PRINT " "
760   x1=x1+dx(d1)
770   y1=y1+dy(d1)
780   IF x1>73 THEN x1=0
790   IF x1<0 THEN x1=73
800   IF y1>19 THEN y1=0
810   IF y1<0 THEN y1=19
820   AT y1,x1
830   INK 2
840   PRINT sh$(d1)
850 END DEFine
860 DEFine PROCedure move_player2
870   q = KEYROW(2)
880   IF q=32 THEN
890     d2=d2+1
900     IF d2=9 THEN d2=1
910   END IF
920   IF q=1 THEN
930     d2=d2-1
940     IF d2=0 THEN d2=8
950   END IF
960 IF (q=128OR q=160OR q=129)AND f2=0THEN
970     f2=1
980     fx2=x2
990     fy2=y2
1000     fxd2=dx(d2)
1010     fyd2=dy(d2)
1020   END IF
1030   IF f2=1 THEN
1040     fire2
1050     fire2
1060   END IF
1070   AT y2,x2
1080   PRINT " "
1090   x2=x2+dx(d2)
1100   y2=y2+dy(d2)
1110   IF x2>73 THEN x2=0
1120   IF x2<0 THEN x2=73
1130   IF y2<0 THEN y2=19
1140   IF y2>19 THEN y2=0
1150   AT y2,x2
1160   INK 4
1170   PRINT sh$(d2)
1180 END DEFine
```

```
1190 DATA 0,-1
1200 DATA 1,-1
1210 DATA 1,0
1220 DATA 1,1
1230 DATA 0,1
1240 DATA -1,1
1250 DATA -1,0
1260 DATA -1,-1
1270 DATA ">","/","=","\","?","/","<","\"
1280 DEFine PROCedure fire1
1290 gy1=fy1:gx1=fx1
1300  fx1=fx1+fxd1
1310  fy1=fy1+fyd1
1320 IF fx1>72 THEN
1330    f1=0
1340    GO TO 1550
1350  END IF
1360  IF fx1<2 THEN
1370    f1=0
1380    GO TO 1550
1390  END IF
1400  IF fy1<2 THEN
1410    f1=0
1420    GO TO 1550
1430  END IF
1440  IF fy1>18 THEN
1450    f1=0
1460    GO TO 1550
1470  END IF
1480  IF fy1=y2 AND fx1=x2 THEN
1490    f1=0
1500    score1
1510  END IF
1520  INK 2
1530  AT fy1,fx1
1540  PRINT "."
1550 AT gy1,gx1
1560 PRINT " "
1570 END DEFine
1580 DEFine PROCedure fire2
1590 gx2=fx2:gy2=fy2
1600  fx2=fx2+fxd2
1610  fy2=fy2+fyd2
1620  IF fx2>71 THEN
1630    f2=0
1640    GO TO 1850
1650  END IF
```

```
1660  IF fx2<2 THEN
1670    f2=0
1680    GO TO 1850
1690  END IF
1700  IF fy2<2 THEN
1710    f2=0
1720    GO TO 1850
1730  END IF
1740  IF fy2>18 THEN
1750    f2=0
1760    GO TO 1850
1770  END IF
1780  IF fx2=x1 AND fy2=y1 THEN
1790    score2
1800    f2=0
1810  END IF
1820  INK 4
1830  AT fy2,fx2
1840  PRINT "."
1850  AT gy2,gx2
1860  PRINT " "
1870 END DEFine
1880 DEFine PROCedure score1
1890  s1=s1+1
1900  IF s1=10 THEN GO TO 2050
1910  score
1920 END DEFine
1930 DEFine PROCedure score2
1940  s2=s2+1
1950  f2=0
1960  IF s2=10 THEN GO TO 2050
1970  score
1980 END DEFine
1990 DEFine PROCedure score
2000  AT #0,0,0
2010  PRINT #0;"player 1 :-";s1
2020  AT #0,0,50
2030  PRINT #0;"player 2 :-";s2
2040 END DEFine
2050 MODE 8
2060 PAPER 0
2070 CLS
2080 CSIZE 3,1
2090 AT 0,8
2100 PRINT "Game over"
2110 INK 4
2120 AT 3,2
```

```
2130 PRINT "player 1 ";s1
2140 AT 5,2
2150 PRINT "player 2 ";s2
2160 PRINT
2170 INK 1
2180 IF s1=10 THEN PRINT "player 1 won"
2190 IF s2=10 THEN PRINT "player 2 won"
2200 PRINT
2210 INK 2
2220 PRINT "play again ?";
2230 INK 6
2240 INPUT a$
2250 IF a$="y" THEN RUN
2260 IF a$<>"n" THEN GO TO 2220
```

Programs to Delight and Dazzle!

A QL Compendium invites you, the keen computer hobbyist, to key in and play a fascinating selection of thirty games on the Sinclair QL.

Each game is presented with a complete listing in SUPERBASIC, a helpful introductory description and a screenshot. There is something to appeal to the traditional and ultramodern computer gamester alike.

Colourful graphics and art programs are included and you are introduced to a drawing program that allows you to 'paint' detailed pictures of your own creation.

The authors set out with the intention of encouraging you to develop programming skills, allowing you to experiment with the programs in the book or even to write your own original programs. But the emphasis remains strictly on the fun you can have exploring the powerful facilities of your own Sinclair QL!

A Newtech Production for
Addison-Wesley Publishing Company

GB £ NET +007.95

ISBN 0-201-15438-2

00795

9 780201 154382