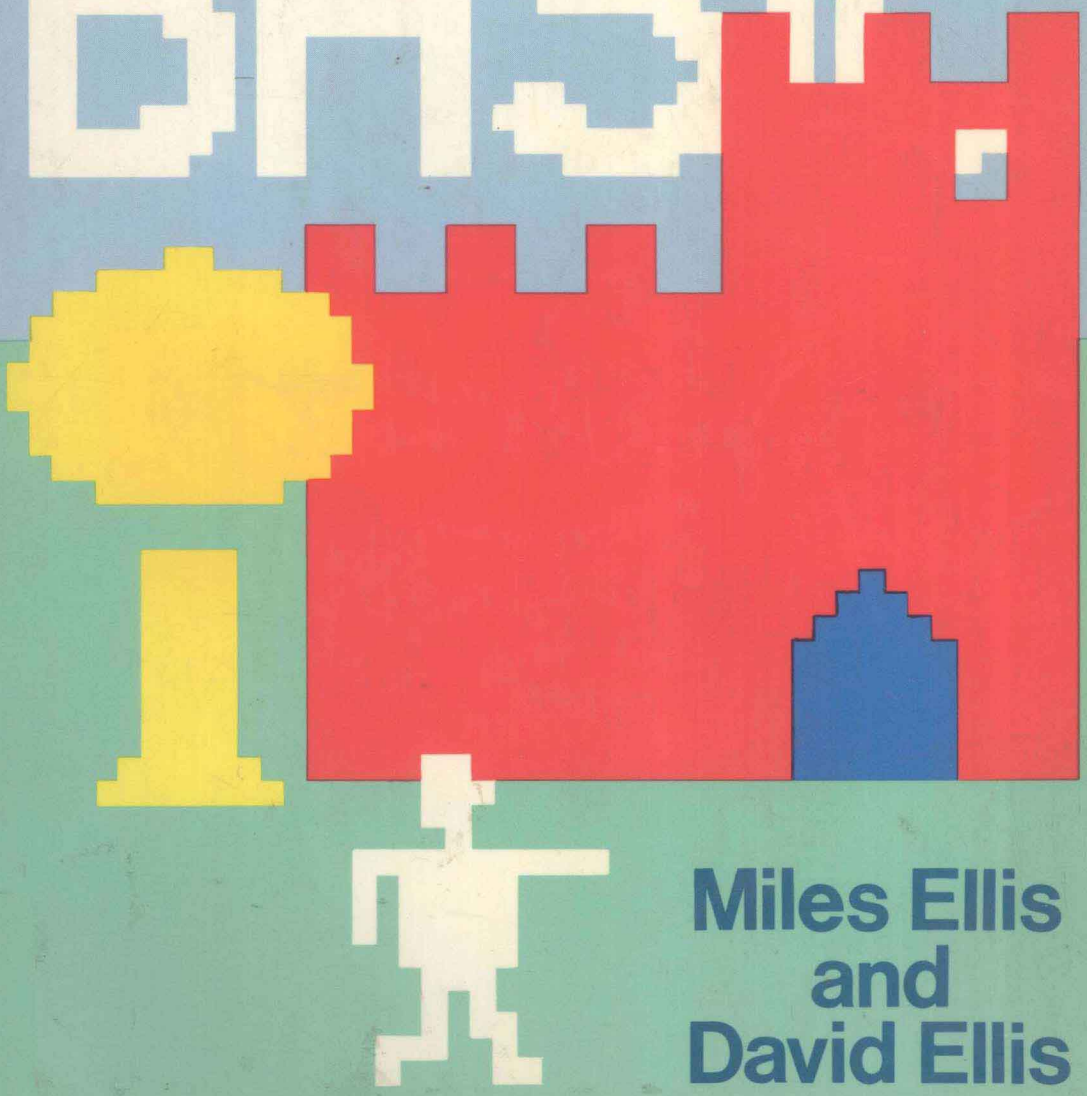
Adventure into BBC BASIC



Miles Ellis
and
David Ellis

Adventure Into
BBC Basic

Miles Ellis
Computing Services
University of Sheffield

and

David Ellis

JOHN WILEY & SONS
Chichester . New York . Brisbane . Toronto . Singapore

Library of Congress Cataloging in Publication Data:

Ellis, Miles.
Adventure into BBC Basic.

Includes index.
1. BBC Microcomputer – Programming. 2. Basic (Computer program language) I. Ellis, David. II. Title.
III. Title: Adventure into BBC Basic.
QA76.8.B35E44 1984 001.64'2 83-16998
ISBN 0 471 90171 7

British Library Cataloguing in Publication Data:

Ellis, Miles
Adventure into BBC Basic.
1. BBC Microcomputer – Programming
2. Basic (Computer program language)
I. Title II. Ellis, David
001.64'24 QA76.8.B3

ISBN 0 471 90171 7

Typeset by Pintail Studios, Ringwood, Hampshire.
Printed in Great Britain by the Pitman Press, Bath, Avon.

Adventure Into
BBC Basic

Preface

This is an unusual book for several reasons.

The most obvious way in which it is unusual can be best appreciated by looking at the contents list and comparing it with almost any other book on Basic programming. You will find that the order in which the various topics are discussed in this book is very different from most others. You will understand why this is so once we have discussed two more unusual features of the book.

The least obvious way in which it is unusual is that one of the authors (David Ellis) was only 12 years old when the book was written and was, therefore, much less experienced than the other author (his dad!). He knew enough about the BBC microcomputer, though, to suggest the fundamental approach which is used throughout this book and which makes it different from other books on programming.

This leads us to the third unusual feature, which is that the book has an accompanying cassette (and disc) containing a series of Basic programs which, together, create an enjoyable and challenging Adventure game to play on your BBC computer while you are learning to write your own programs. It isn't essential to buy the cassette or disc – the programs are listed in full in Appendix A – but it will save you a lot of typing if you do.

The fundamental philosophy of this book is that programming should be an enjoyable activity. It is almost always intellectually challenging, but that certainly doesn't mean that it has to be dry, theoretical and boring. We therefore start with an enjoyable game and show you how it produces different sound effects. From here it is only a small step to writing programs to play tunes.

But sound without pictures can be rather boring, so we then start to explore the world of graphics. The BBC computer has remarkable facilities for both sound and picture generation, and you will find that you can produce your own "personalised" version of the Adventure program and write a number of interesting demonstration programs without having even started on what most books consider to be the essential first steps. Once we do start to discuss the more conventional aspects of Basic, therefore, you will find that you can write really enjoyable and interesting programs – and not only to play games!

As we mentioned above, one of us was relatively inexperienced when this book was started. The idea for the approach used came from him, however, as did a number of the details of both the Adventure program and other example programs throughout the book.

He was also responsible for verifying that the book was easy to use and to understand while it was being written.

Having two members of one family involved in preparing and writing this book has, inevitably, had an effect on family life, and we would both like to apologise to Maggie (wife and mother), Sarah (daughter and sister) and Richard (son and brother) for the disruption caused to a normally chaotic household by our involvement with both computer and book!

Finally we should like to add a word of appreciation to Ian Shelley, our editor at John Wiley, for his support for the rather unusual approach used in this book, and for all his efforts in sorting out the various problems in producing a book, program cassette and disc. The text of the book was typed by Judith Denton directly onto a "floppy disc" using the Wordstar word-processing program on a SuperBrain microcomputer, and we are very grateful to her for all her hard work. The discs were then used by John Wiley to drive a photo-typesetting machine without the need for any further typing. The programs in the book were all written and tested on a BBC model B microcomputer and then listed directly from the computer to a Diablo printer – so you can be sure that they all work!

We hope that you enjoy learning to write programs for your BBC computer and that this book helps you to make the most of what is an extremely versatile and powerful machine.

MILES ELLIS
DAVID ELLIS
Sheffield

Contents

1

What's it all about?

1.1 THE BOOK

One of the most astonishing success stories of the last few years has been the personal computer. Until the mid-1970s, computers were large (or at any rate substantial) boxes of electronic and mechanical equipment that were used almost exclusively for commercial, industrial, academic or other "professional" purposes (see figures 1.1 and 1.2).

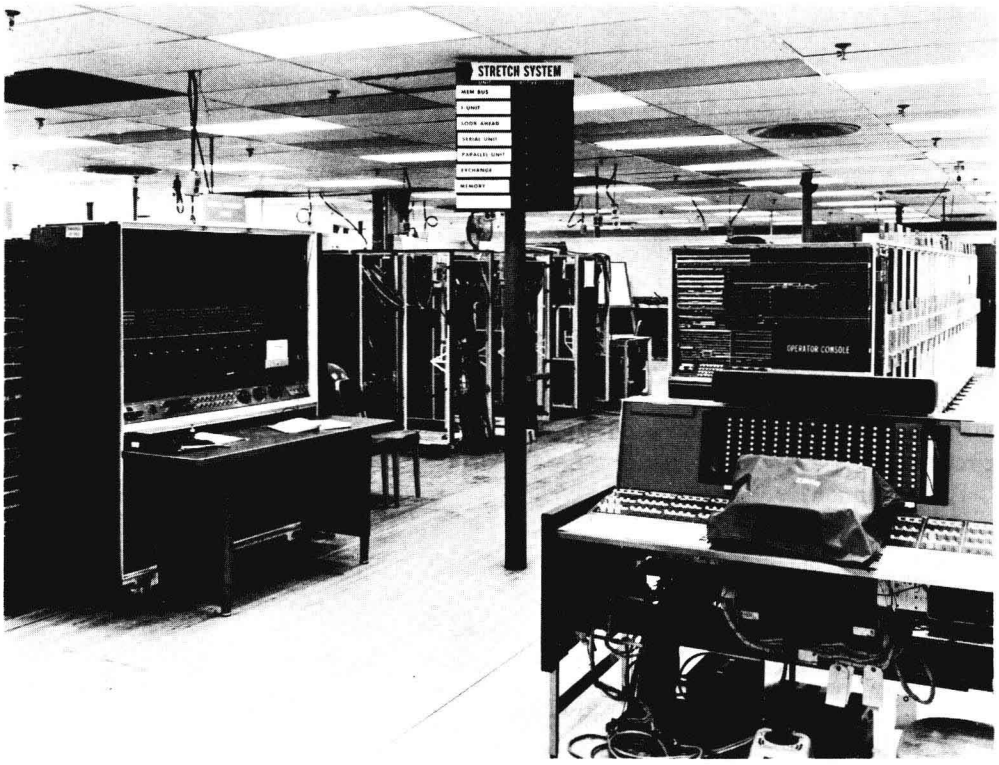


Figure 1.1 A large computer from the 1960s (courtesy of IBM United Kingdom Ltd.)

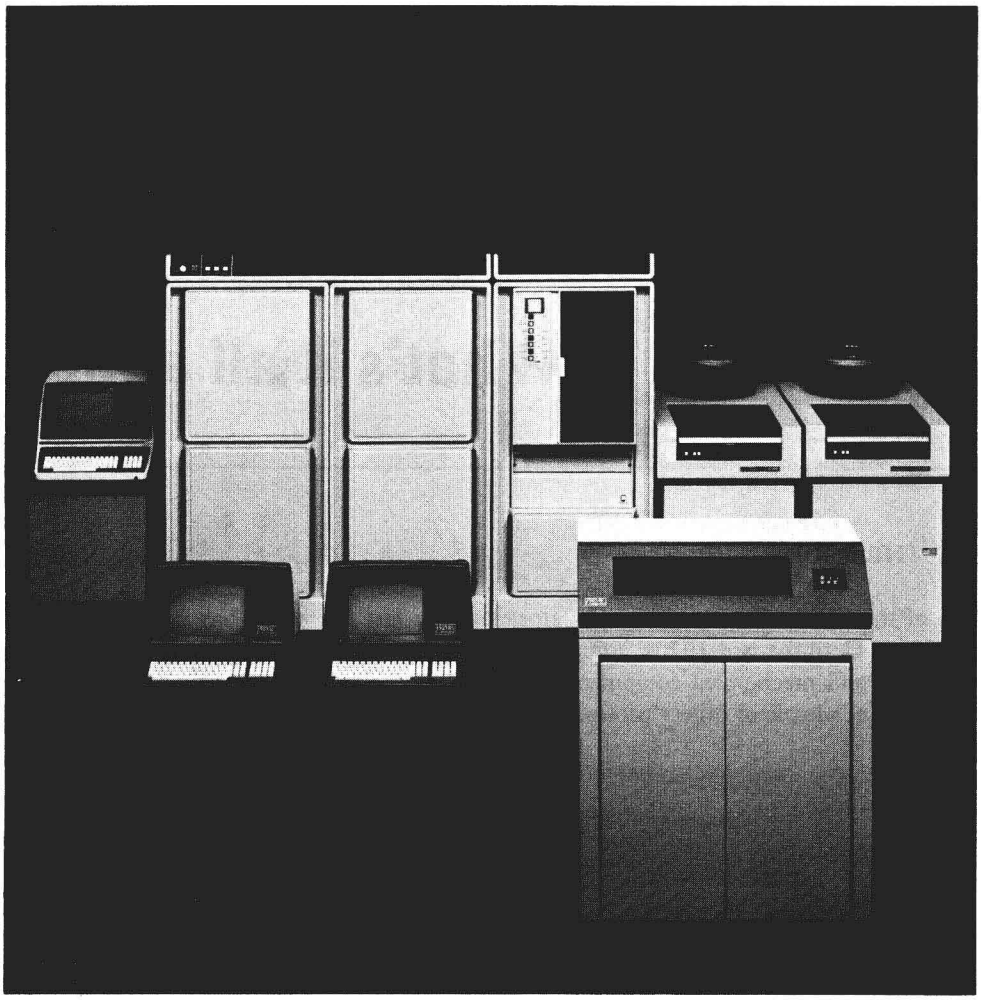


Figure 1.2 A medium-sized computer from the 1970s (courtesy of Prime Computer (UK) Ltd.)

In the mid-1970s, however, all that began to change. The development of integrated circuits which contained several transistors, resistors and capacitors on a single slice of silicon (a "chip") led inexorably to the production of a computer on a chip – or at least to the heart of a computer, the central processor, on a chip. This new type of integrated circuit was called a microprocessor.

Almost immediately the early microprocessors were incorporated into simple, self-contained, boxes containing all the other components necessary to create a desk-top sized microcomputer. One of the first of these was produced by the calculator manufacturer Commodore (figure 1.3), while the most famous of all was developed in a garage in California (figure 1.4).

At the same time as microprocessors were being used to create a new breed of computers, however, they were also being used to found a totally new form of entertainment – the video game. At first this was a simple "bat and ball" type of game (figure 1.5), but after only a brief pause came a range of highly sophisticated all-colour action games such as Defender, Pacman, and the grand-daddy of them all – Space Invaders (figure 1.6).

Figure 1.3 Commodore Pet

Figure 1.4 Apple II

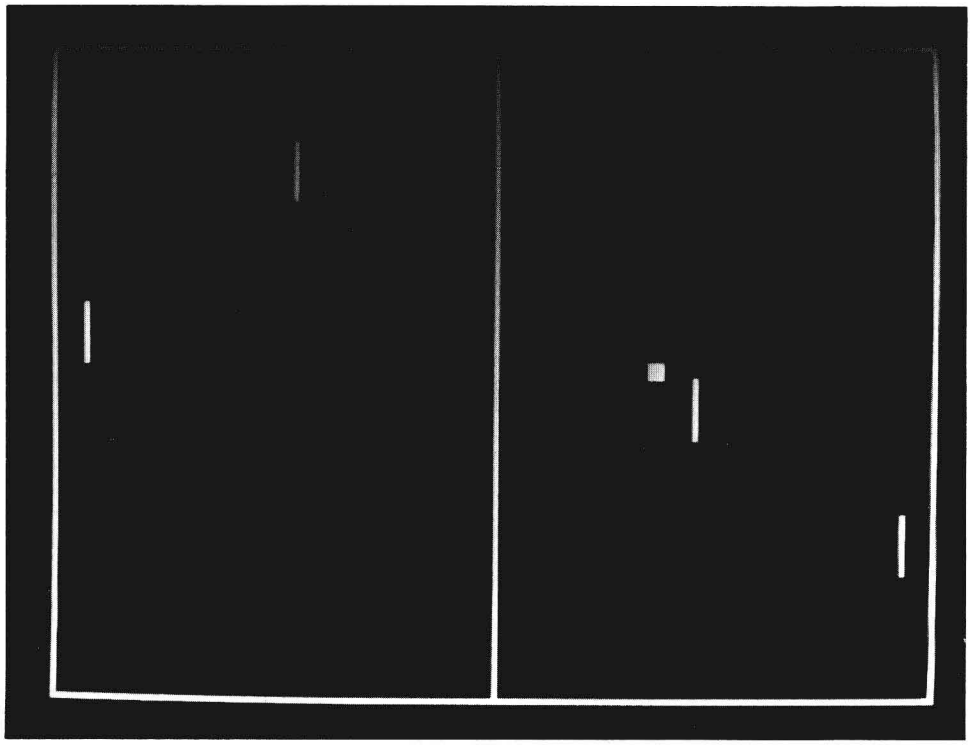


Figure 1.5 A simple "bat and ball" game

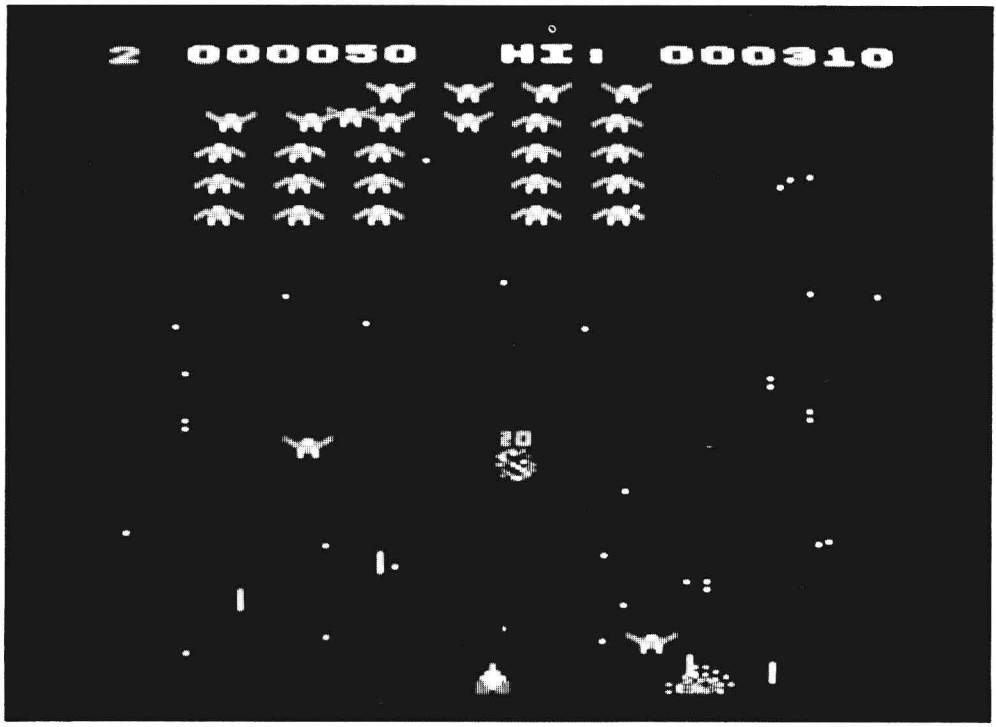


Figure 1.6 "Invaders from Space"

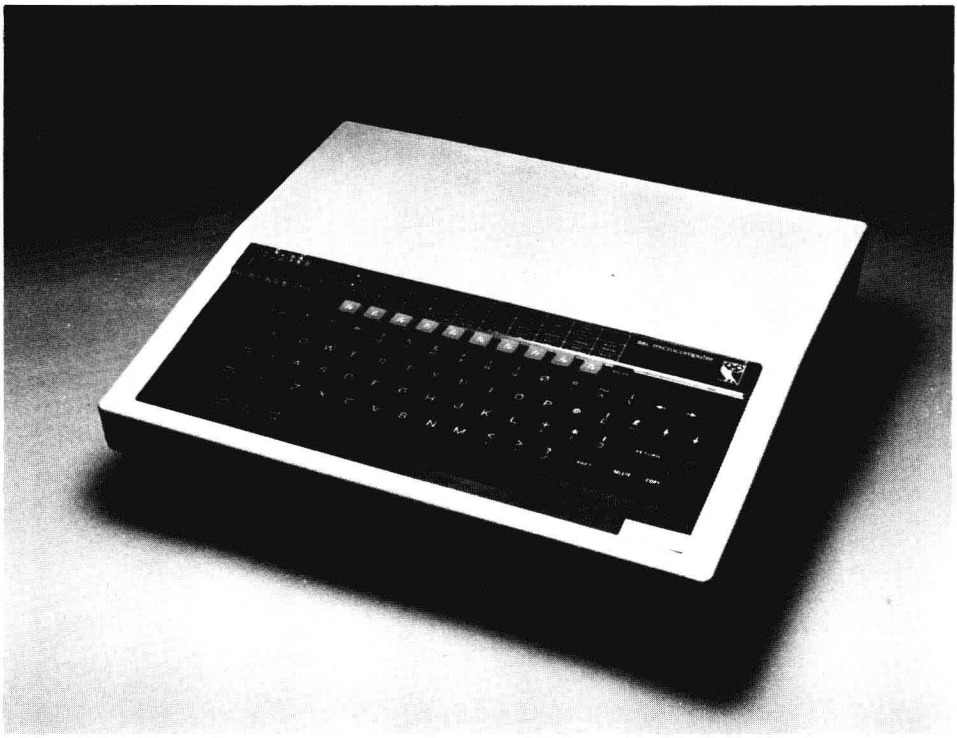


Figure 1.7 BBC computer

The modern miracle of electronics is that prices fall every year, and by 1980 prices were such that small "personal" computers could be purchased for the same as a single item of "hi-fi" equipment. Thus started the boom in demand which led in 1981 to the BBC producing a television series about computers and their uses ("The Computer Programme"), and commissioning Acorn Computers to build a low-priced, high-specification computer, to be called the BBC microcomputer (figure 1.7).

Many people buy a personal computer such as the BBC microcomputer with no intention of writing their own programs. They buy (or beg, borrow or steal!) programs written by others to play highly addictive games, or to use in education, or to convert their computer into a word-processor, or to carry out one of a multitude of different tasks. However, even the most addictive of games becomes stale in time, and someone else's view of the way to deal with a problem does not always accord with your own. So you need to learn how to write your own programs.

This book is intended to help you to learn just that. It uses a rather unusual approach however which, we feel, makes it much more suitable than the normal "text-book" approach.

We start from the premise that you have already used the computer to run other people's programs (even if these are only the programs on the Welcome tape supplied with your computer). These, especially if they are games, will probably use the special facilities of the computer (colour, sound, graphics, etc.) to provide visually interesting and stimulating results. It is therefore very undesirable for you to start by writing programs which solve trivial problems using white text on a black screen!

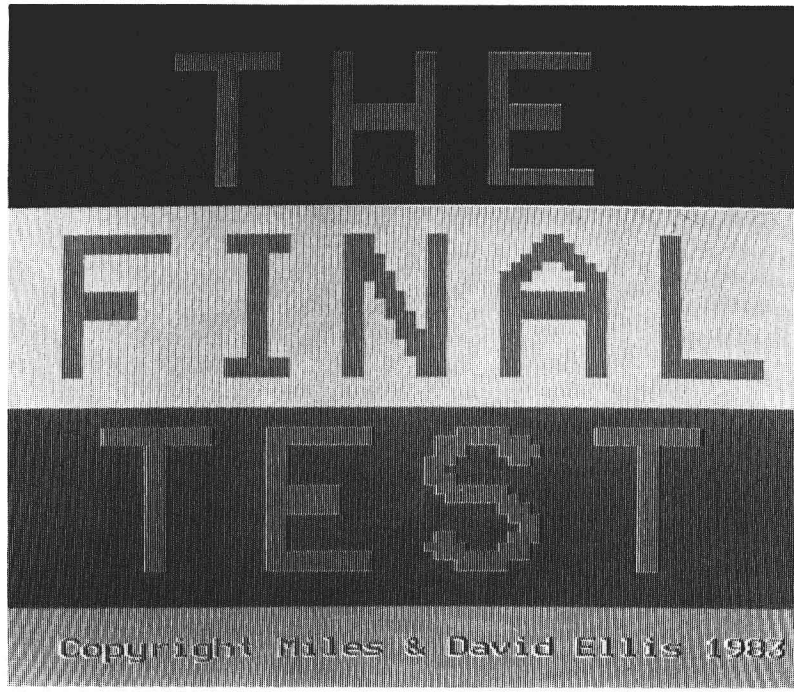


Figure 1.8 Opening title of game

At the end of this chapter we introduce a specially written program called The Final Test.

This is a game of the type known as Adventure games, in which you, the player, have to find your way through a computer-generated maze, overcoming various obstacles on the way, in order to achieve some final goal. In this particular case the task involves a multi-storeyed castle populated by dragons, evil spirits and other unpleasant creatures. Your objective is to find your way to the top of the tower, and to rescue the beautiful princess imprisoned there. The problem is that there are a great many doors and tunnels to go through, enemies to fight off (by both magical and physical means) and tests of initiative and speed of reaction to overcome.

This program is available in cassette form, although a full listing is also included in this book as Appendix A.

The game itself is an interesting and challenging one, but more important is the fact that it is written entirely in Basic. It can therefore be used as an example of how to carry out various types of operations on the computer, and equally importantly as a means whereby you can create your own personalised sophisticated game program at a very early stage in your programming career.

Of course there are plenty of other examples and programming experiments in the book, but the use of this program enables us to demonstrate and explain all the features of Basic in an easily understandable, and yet totally non-trivial, way. We hope and believe that you will find this approach easy and enjoyable to follow.

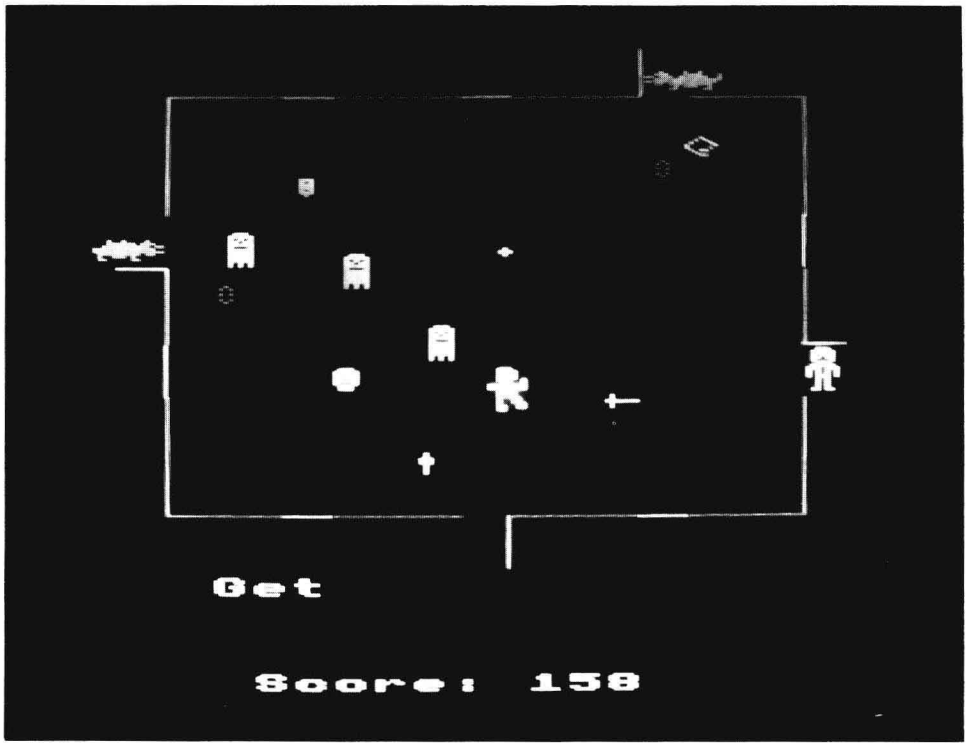


Figure 1.9 Screen during game (in Castle)

We should give one word of warning though. This book is intended to be read in the order in which it is written. The order in which items are introduced is very different from that employed in most books on programming because, as explained above, we feel that the conventional approach is not suitable for those who have already *used* computers but have not yet *programmed* them. Once you have reached the end of the book, you will find that the index will enable you to readily remind yourself of how some aspect of Basic works, but don't be tempted to use it to jump ahead to items in later chapters or you may find that you have missed some important point or experiment.

Incidentally, we have included a number of what we call "experiments" throughout the book for you to try. We have called them experiments rather than "exercises", "questions" or "problems" because that is what they are. We want you to *experiment* with the facilities of your computer – to make the dragons yellow instead of green, or to flash on and off the screen, or to be a different shape – to write a program to play a tune, or even to use the keys like the keys of a piano – to draw graphs in which colours are used to highlight particular aspects – to create abstract dynamic visual or aural effects – etc. In some cases we shall include an example of a typical result of the experiment (i.e. a program), in other cases we shall leave it to you to obtain a satisfactory conclusion by yourself. At the end of each chapter is a section called "Experimental Hints" which you may find useful to look at after you have completed a particular experiment, or group of experiments.

The remainder of this chapter explains a few basic principles concerning your computer and its language, and then describes how you can set up The Final Test so that you can play it and also (when you have had enough of that!) use it in conjunction with the rest of this book to learn how to write your own programs.

1.2 THE COMPUTER

When we refer to the BBC microcomputer we probably are thinking of a cream-coloured plastic box approximately 16 in × 12 in × 3 in high (40 cm × 30 cm × 7.5 cm), see figure 1.10.

In many ways we are correct, but at the same time we are partly wrong since this box can do absolutely *nothing* unless it is connected to some other pieces of equipment. In particular we must have some form of display (e.g. a television set) so that the computer can communicate with us. Figure 1.11 therefore is a more realistic view of the computer.

However, even now the picture is only partially complete because without some way of loading other people's programs and/or saving your own for another occasion the usefulness of the computer would be drastically reduced. Typically, therefore, we attach a cassette recorder to the computer for this purpose (see figure 1.12).

But even this isn't all it seems. Our original cream-coloured box actually consists of electronic and mechanical devices to carry out three very different functions, namely the input of information we wish to communicate to the computer, the (temporary) storage of that and other information, and the calculation of whatever results we require from the computer. These are performed, respectively, by the keyboard of the computer, the memory "chips" inside it and the microprocessor (or central processor unit, or CPU) which is also inside the case. Thus figure 1.13 gives a more accurate representation of our computer (or, perhaps, our computer *system*), and indicates that it consists essentially of five parts, of which three are contained within what we originally identified as the BBC microcomputer.

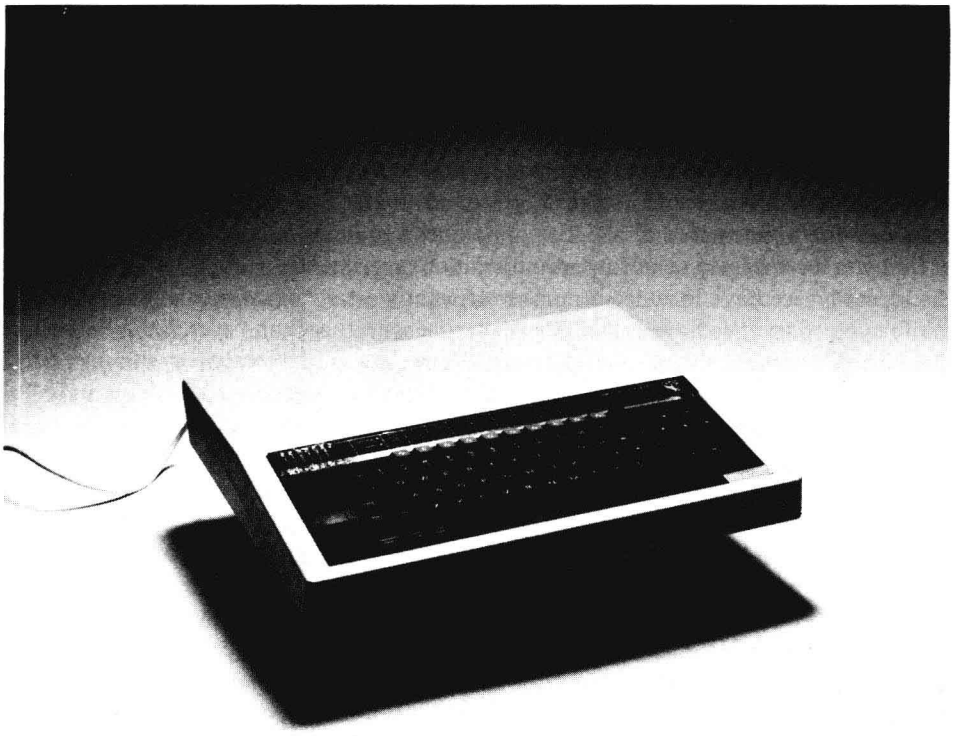


Figure 1.10 BBC computer