Penguin Books

Creative Assembler
How To Write Arcade Games

As one of Acornsoft's games programmers, Jonathan Griffiths was responsible for such widely popular programs as Snapper and JCB Digger for the BBC Microcomputer Model B and Acorn Electron He has prepared a cassette program, which can be used in conjunction with this book, available from Acornsoft Limited, c/o Vector Marketing Ltd, Denington Industrial Estate, Wellingborough, Northants NN8 2RL.

Acknowledgements
Thanks are due to David Johnson-Davies for providing an earlier version of the first few chapters, Jim Dobson and Jeremy Bennett for helping in the writing of this book, and to Philippa Bush and Sharron Fellows for editing it Also, thanks are due to Orlando M. Pilchard (Q.C.), Chris Jordan, Paul Hudson,Peter Cockerell, Dominic Verity, Jeremy San, Robert Macmillan, Paul Fellows, John Collins, Simon Hughes and Mark Holmes for proof reading.

This book was written and prepared on a BBC Microcomputer Model B using the VIEW word processor.

This book is dedicated to all the staff at Acornsoft.

CREATIVE ASSEMBLER
How To Write Arcade Games

Jonathan Griffiths

Penguin Books

The Penguin Acorn Computer Library is a joint venture, produced by Acornsoft Limited (in association with Pilot Productions Limited), and published by Penguin Books Limited.

Penguin Books Ltd, Harmondsworth, Middlesex, England
Penguin Books, 40 West 23rd Street, New York, New York, 10010, U.S A.
Penguin Books Australia Ltd, Ringwood, Victoria, Australia
Penguin Books Canada Ltd, 2801 John Street, Markham, Ontario, Canada L3R 1~
Penguin Books (N.Z.) Ltd, 182-190 Wairau Road, Auckland 10, New Zealand

First published 1984

Set in Palatino by Repro Graphics, 61 Cromwell Road, Southampton

Colour origination by RCS Graphics Ltd, 39-40 Springfield Mills, Farsley, Pudsey, Leeds, and MRM Graphics, 61 Station Road, Winslow, Bucks

Made and printed in Spain by Printer industria grafica s.a., Sant Vicenc dels Horts, Barcelona
D.L.B. 1517O-1984

Line illustrations by Rob Shone
Original photography by Nick Wright

CONTENTS

SECTION 2

INTRODUCTION

The BASIC assembler which is available on the BBC Microcomputer and Acorn Electron is a very powerful tool for programmers. It provides a comprehensible interface between the programmer and the machine code language which the 6502 processor itself uses. Hence the programmer is able to control the machine more directly using assembler.

The main reason why people write programs in assembler rather than BASIC is probably because of the speed difference between the two. Assembler instructions can be executed extremely quickly, a program written in BASIC will take between 10 and 100 times as long. Hence assembler is particularly useful for games' programmers since it enables them to move missiles and creatures across the screen quickly and smoothly. If BASIC was used to calculate the new coordinates of each object and draw them at those positions then movement would tend to occur in jerky leaps.

However, speed is not the only factor to be taken into consideration. Assembler programming gives you more power to solve a problem than BASIC does. All high-level languages require programs to have a certain structure and this puts constraints on programs written in that language.

'Rocket Raid' — an excellent example of the use of the assembler.

Sceptics may advise against using assembler on the grounds that it is too complex. It is true that operations such as multiplication and division which are easy to perform in BASIC are not as straightforward in assembler. For what might be considered a trivial task, for example multiplying a number by three, several assembler instructions are required instead of just a single BASIC one. A further problem is that there are no FOR ... NEXT or REPEAT ... UNTIL loops in assembler; if you require a loop you must set one up yourself. The same applies to floating point arithmetic -- assembler only supports integer calculations.

My advice is that you ignore these sceptics. It isn't difficult to learn to program in assembler. The programs look much less like English than BASIC I

ones do but nevertheless to someone who knows the language they are easy to understand. Like learning to do anything else, all that is required is a certain amount of knowledge and a lot of practice. This book has been written to provide the knowledge -- the rest is up to you.

The book is divided into three sections, each of which has a different task to perform. The first part aims to introduce the more useful assembler instructions available for the 6502 processor, giving simple examples of how they can be used. The second part introduces some of the more complex programming techniques which are aimed in particular at people writing large assembler programs. The third part is aimed mainly at the games' programmer. It provides many useful routines and finally shows how these can all be linked together to produce a complete game.