Macmillan Computer Science Series

Consulting Editor

Professor F. H. Sumner, University of Manchester

- S. T. Allworth and R. N. Zobel, *Introduction to Real-time Software Design, second edition*
- Ian O. Angell, *High-resolution Computer Graphics Using C*
- Ian O. Angell and Gareth Griffith, *High-resolution Computer Graphics Using FORTRAN 77*
- Ian O. Angell and Gareth Griffith, *High-resolution Computer Graphics Using Pascal*
- M. Azmoodeh, *Abstract Data Types and Algorithms*
- C. Bamford and P. Curran, *Data Structures, Files and Databases*
- Philip Barker, *Author Languages for CAL*
- A. N. Barrett and A. L. Mackay, *Spatial Structure and the Microcomputer*
- R. E. Berry, B. A. E. Meekings and M. D. Soren, *A Book on C, second edition*
- P. Beynon-Davies, *Information Systems Development*
- G. M. Birtwistle, *Discrete Event Modelling on Simula*
- B. G. Blundell and C. N. Daskalakis, *Using and Administering an Apollo Network ductory Guide to Silvar Lisco and Hilo Simulators*
- B. G. Blundell and C. N. Daskalakis, *Using and Administering an Appollo Network*
- T. B. Boffey, *Graph Theory in Operations Research*
- Richard Bornat, *Understanding and Writing Compilers*
- Linda E. M. Brackenbury, *Design of VLSI Systems – A Practical Introduction*
- G. R. Brookes and A. J. Stewart, *Introduction to occam 2 on the Transputer*
- J. K. Buckle, *Software Configuration Management*
- W. D. Burnham and A. R. Hall, *Prolog Programming and Applications*
- P. C. Capon and P. J. Jinks, *Compiler Engineering Using Pascal*
- J. C. Cluley, *Interfacing to Microprocessors*
- J. C. Cluley, *Introduction to Low-level Programming for Microprocessors*
- Robert Cole, *Computer Communications, second edition*
- Derek Coleman, *A Structured Programming Approach to Data*
- Andrew J. T. Colin, *Fundamentals of Computer Science*
- Andrew J. T. Colin, *Programming and Problem-solving in Algol 68*
- S. M. Deen, *Fundamentals of Data Base Systems*
- S. M. Deen, *Principles and Practice of Database Systems*
- C. Delannoy, *Turbo Pascal Programming*
- Tim Denvir, *Introduction to Discrete Mathematics for Software Engineering*
- P. M. Dew and K. R. James, *Introduction to Numerical Computation in Pascal*
- D. England *et al.*, *A Sun User's Guide*
- A. B. Fontaine and F. Barrand, *80286 and 80386 Microprocessors*
- K. C. E. Gee, *Introduction to Local Area Computer Networks*
- J. B. Gosling, *Design of Arithmetic Units for Digital Computers*
- M. G. Hartley, M. Healey and P. G. Depledge, *Mini and Microcomputer Systems*

(continued overleaf)

Roger Hatty, *Z80 Assembly Language Programming for Students*
 Roland N. Ibbett and Nigel P. Topham, *Architecture of High Performance Computers, Volume I*
 Roland N. Ibbett and Nigel P. Topham, *Architecture of High Performance Computers, Volume II*
 Patrick Jaulent, *The 68000 – Hardware and Software*
 P. Jaulent, L. Baticle and P. Pillot, *68020–30 Microprocessors and their Co-processors*
 J. M. King and J. P. Pardoe, *Program Design Using JSP – A Practical Introduction*
 E. V. Krishnamurthy, *Introductory Theory of Computer Science*
 V. P. Lane, *Security of Computer-Based Information Systems*
 Graham Lee, *From Hardware to Software – An Introduction to Computers*
 A. M. Lister and R. D. Eager, *Fundamentals of Operating Systems, fourth edition*
 Tom Manns and Michael Coleman, *Software Quality Assurance*
 Brian Meek, *Fortran, PL/1 and the Algols*
 A. Mével and T. Guéguen, *Smalltalk-80*
 R. J. Mitchell, *Microcomputer Systems Using the STE Bus*
 Y. Nishinuma and R. Espesser, *UNIX – First Contact*
 Pim Oets, *MS-DOS and PC-DOS – A Practical Guide, second edition*
 A. J. Pilavakis, *UNIX Workshop*
 Christian Queinnec, *LISP*
 E. J. Redfern, *Introduction to Pascal for Computational Mathematics*
 Gordon Reece, *Microcomputer Modelling by Finite Differences*
 W. P. Salman, O. Tisserand and B. Toulout, *FORTH*
 L. E. Scales, *Introduction to Non-linear Optimization*
 Peter S. Sell, *Expert Systems – A Practical Introduction*
 A. G. Sutcliffe, *Human-Computer Interface Design*
 Colin J. Theaker and Graham R. Brookes, *A Practical Course on Operating Systems*
 M. R. Tolhurst *et al.*, *Open Systems Interconnection*
 J-M. Trio, *8086–8088 Architecture and Programming*
 A. J. Tyrrell, *COBOL from Pascal*
 M. J. Usher, *Information Theory for Information Technologists*
 B. S. Walker, *Understanding Microprocessors*
 Colin Walls, *Programming Dedicated Microprocessors*
 I. R. Wilson and A. M. Addyman, *A Practical Introduction to Pascal – with BS6192, second edition*

Non-series

Roy Anderson, *Management, Information Systems and Computers*
 I. O. Angell, *Advanced Graphics with the IBM Personal Computer*
 J. E. Bingham and G. W. P. Davies, *Planning for Data Communications*
 B. V. Cordingley and D. Chamund, *Advanced BASIC Scientific Subroutines*
 N. Frude, *A Guide to SPSS/PC+*
 Barry Thomas, *A PostScript Cookbook*

High-resolution Computer Graphics Using C

Ian O. Angell

*Department of Information Systems
London School of Economics
University of London*

M
MACMILLAN

© Ian O. Angell 1990

All rights reserved. No reproduction, copy or transmission of this publication may be made without written permission.

No paragraph of this publication may be reproduced, copied or transmitted save with written permission or in accordance with the provisions of the Copyright, Designs and Patents Act 1988, or under the terms of any licence permitting limited copying issued by the Copyright Licensing Agency, 33-4 Alfred Place, London WC1E 7DP.

Any person who does any unauthorised act in relation to this publication may be liable to criminal prosecution and civil claims for damages.

First published 1990

Published by
MACMILLAN EDUCATION LTD
Houndmills, Basingstoke, Hampshire RG21 2XS
and London
Companies and representatives
throughout the world

British Library Cataloguing in Publication Data
Angell, Ian O.

High-resolution computer graphics using C. —
(Macmillan computer science series)

1. Microcomputer systems. graphic displays.
programming languages C languages

I. Title

006.6'76

ISBN 978-0-333-49873-6 ISBN 978-1-349-20759-6 (eBook)
DOI 10.1007/978-1-349-20759-6

Contents

<i>Preface</i>	viii
1 Familiarisation with Programs, Graphics Devices and Primitives	1
Structure of programs in this book. Concept of a graphics viewport. Pixels and co-ordinates. Primitives. Windows and mapping of real two-dimensional space onto rectangular matrix of pixels. Simple graphical constructions: squares, circles, ellipses, spirals	
2 Data Structures	27
Arrays and subscripted variables. Pointers. Linked lists and stacks. Graphs and networks. Topological sorting. Trees	
3 An introduction to Two-dimensional Co-ordinate Geometry	48
Two-dimensional co-ordinate systems — origin, axes, points, vectors, lines, curves, and their properties. Analytic (functional) representation and parametric forms. Polygons and convex areas. Orientation. Inside and outside	
4 Matrix Representation of Transformations in Two-dimensional Space	64
Matrices. Affine transformations: translation, rotation and scaling (reflection), and their 3 by 3 matrix representation. Inverse transformations. Transforming axes and transforming space. Combining transformations. Positions and the observer. Vertex, line and facet construction, and views of two-dimensional scenes	
5 Techniques for Manipulating Two-dimensional Objects	90
Line clipping and blanking. Line types. Use of input devices. Rubberbanding, draw, drag and delete techniques. Hatching a polygon. Area-filling. Orientation of a convex polygon. Intersection of two convex polygons. Animation	
6 Three-dimensional Co-ordinate Geometry	123
Three-dimensional co-ordinate systems. Vector representation of points, lines and planes. Properties of these objects. Direction cosines. Scalar product and vector product. Intersections, distances and reflections. Analytic representations. Orientation of convex polygons in two-dimensional and three-dimensional space	

7 Matrix Representation of Transformations in Three-dimensional Space	147
4 by 4 matrix representation of translation, rotation and scaling (reflection). Transforming axes and space. Inverse transformations. Combining transformations. Rotation about an arbitrary axis. Transforming space. Positioning objects in space. Definition of three-dimensional scenes. SETUP and ACTUAL positions	
8 The Observer and the Orthographic Projection	168
The observer, OBSERVER co-ordinate system and OBSERVED co-ordinates. General projections. The view plane and window. Orthographic projection. Drawing a scene: wire diagrams	
9 Generation of Model Data	180
Storing model data in files. Use of symmetry to minimise construction effort. Bodies of revolution and rotation. Extrusion of planar polygonal objects. Animation	
10 Simple Hidden Line and Surface Algorithms	205
Painter's algorithm: back-to-front method using raster area filling. Mathematical surfaces. Simple algorithm for convex solids using oriented facets. Discussion of general problem of hidden line and surface algorithms	
11 Perspective and Stereoscopic Projections	221
Theory of perspective projection. Cone of vision. Pyramid of vision. The perspective plane. Vanishing points and horizon. Drawing objects in perspective. Extension of previous algorithms to include perspective. Theory of stereoscopic projection. Examples	
12 A More General Hidden Line Algorithm	231
A general hidden line algorithm to be used on line-drawing graphics displays	
13 A More General Hidden Surface Algorithm	244
A general hidden surface algorithm to be used on colour graphics devices. Overlapping of oriented polygons on perspective plane. Creation of a network and topological sorting	
14 Three-dimensional Clipping	253
Dealing with objects that lie partially or totally outside the pyramid of vision	
15 Shading	264
A geometric model for a light source. Modelling the reflection of light from a surface. Intensity and colour of light. Reflective properties of a surface. Diffuse and specular reflection, ambient light. Developing a shading	

model. Implementation of shading models: random sampling, pixel patterns and RGB colour shading using constant shading. Gouraud intensity interpolation and Phong normal interpolation. Methods of colour definition

16 Shadows, Transparent Surfaces and Reflections **303**

Incorporating shadows cast by a single light source into a scene. Use of superficial facets. Use of the hidden surface algorithm to find the shadows. Extension to multiple light sources. Extending the hidden surface algorithm to deal with transparent facets. Calculating the reflection of a scene in a mirror facet

17 Analytic Representation of Three-dimensional Space **322**

Quad-tree and oct-tree algorithms. Tree representation of a scene, logical operations on primitive shapes

18 Projects **336**

Ideas for extended programming projects to help the committed reader

Appendix **347**

Primitives for some of the more popular graphics devices and standards – such as Tektronix 4100 series, G.K.S. etc.

Bibliography and References **362**

Index **365**

Index of Function Listings **379**

Note on availability of diskette **381**

Preface

Until recently, all but the most trivial computer graphics was the province of specialised research groups. Now with the introduction of inexpensive micro-computers and 'graphics-boards', the subject will reach many more users and its full potential can be realised. Computer-generated pictures involving smooth shading, shadows and transparent surfaces, for example, have made a major impact in television advertising. The 'mysterious' techniques for producing such pictures have gained a (false) reputation of complexity for computer graphics.

This book gives a practical description of these ideas and, after studying the contents and implementing the examples and exercises, the reader will be ready to attempt most tasks in graphics.

It is assumed that the reader has an elementary knowledge of the C programming language, and of Cartesian co-ordinate geometry. For those who wish to read good texts on these subjects we recommend books by Waite, Prata and Martin (1984), Kernighan and Ritchie (1978) and Cohn (1961). This knowledge will be used to produce simple diagrams, and to create the basic programming tools and routines for the more advanced colour pictures. Then, hopefully, the reader will be inspired to seek a greater understanding of geometry and also to read the more advanced journals (such as *SIGGRAPH* and *ACM Transactions*) describing recent research developments in computer graphics. A number of relevant references are given throughout the text, but for a more comprehensive bibliography readers are advised to refer to Newman and Sproull (1973) and Foley and Van Dam (1981).

The only way to understand any branch of applied computing is to study and write a large number of programs; this is why the format of this book is that of understanding through program listings and worked examples. The chapters are centred around numerous examples and the ideas that lead from them. Many students readily understand the theory behind graphics, but they have great difficulty in implementing the ideas. Hence great emphasis is placed on the program listings; over a hundred are given — some quite substantial. Total understanding of the theory given in this book will be achieved only by running these programs and experimenting with them. The listings can be thought of as an embryonic graphics package, but most importantly they are a means of describing the algorithms required in the solution of the given problems. They are readily translatable into other computer languages such as Basic, Pascal and FORTRAN. The functions given all relate to a small number of graphics primitives, which are

necessarily device or package dependent. Examples of these primitives are given for the Tektronix 4100 range, for the G.K.S. standard, and the GINO and sample microfilm packages in an appendix.

On occasions, efficiency has been sacrificed in favour of clarity in the description of algorithms. The programs are written in a modular form, which makes the interchanging of program functions relatively straightforward. A unique name is given to any function that solves a given problem, but more than one example may be given for such functions if different, perhaps more general, solutions are required. For example, function *facetfill* can be used to draw a polygon in a fixed colour, but other manifestations may include smooth shading of that polygon and even various textures.

The main purpose of this book, which is essentially a third year undergraduate and M.Sc. course at the University of London, is to set the groundwork of computer graphics. Some of the figures given in this book were produced by past students of the course. Figure 5.1 was produced by Hilary Green, figure 9.11 by Colin Ching, figure 12.5 by Andrew Pullen and figure 18.3 by Paul Mclean Thorne. All of the figures can be produced by using the listings in this book (with some extensions). I would also like to thank Digital Arts Production Ltd for the use of their computing facilities and in particular Gareth Griffith for his collaboration on the FORTRAN 77 and Pascal versions of this course, and for producing many of the diagrams. The programs given are NOT the only way of solving the problems of computer graphics: they reflect the teaching and research interests of the author. They do, however, provide a general strategy for gaining a thorough understanding of computer graphics and should lead the reader to research level in the subject. With this advanced groundwork done, readers can reorganise the approach with their own specific interests in mind. The listings are in a limited form of C in order to make the programs applicable to many of the C subsets now available on microcomputers.

The package developed in this book is for teaching purposes only. Although the author places no copyright restrictions on the use of the listings in a lecture course, no commercial exploitation is permitted without his prior agreement.

Computer graphics is fun! I hope that readers will make this discovery and spend many enjoyable and productive hours in front of their graphics console.

Ian O. Angell