

# COMPUTER GRAPHICS

---

## Systems & Concepts



Rod Salmon and Mel Slater

# COMPUTER GRAPHICS

---

## Systems & Concepts

**Rod Salmon**

Computer Scientist Consultant

**Mel Slater**

Queen Mary College, University of London



**ADDISON-WESLEY PUBLISHING COMPANY**

Wokingham, England · Reading, Massachusetts · Menlo Park, California  
New York · Don Mills, Ontario · Amsterdam · Bonn · Sydney  
Singapore · Tokyo · Madrid · Bogota · Santiago · San Juan

© 1987 Addison-Wesley Publishers Limited  
© 1987 Addison-Wesley Publishing Company, Inc.

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 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.

Cover graphic by kind permission of Dicomed.  
Illustrations by Chartwell Illustrators.  
Typeset by Columns, Reading, Berks.  
Printed in Great Britain by The Bath Press, Avon.

First printed 1987.

**British Library Cataloguing in Publication Data**

Salmon, Rod

Computer graphics : systems and concepts.

1. Computer graphics

I. Title II. Slater, Mel

006.6 T385

ISBN 0-201-14656-8

**Library of Congress Cataloguing in Publication Data**

Salmon, Rod, 1947-

Computer graphics.

Bibliography: p.

Includes index.

1. Computer graphics. I. Slater, Mel. II. Title.

T385.S24 1987 006.6 86-32073

ISBN 0-201-14656-8

# **COMPUTER GRAPHICS**

---

## **Systems & Concepts**

**To Judith  
and Lynette and Simon**

**To Freda and Michael**

---

# PREFACE

---

*"I have told you, you have to have an unbending intent in order to become a man of knowledge."*

Carlos Casteneda, *A Separate Reality*, Penguin, 1973, p. 132.

Computer graphics is a major force in a revolution that is changing the way people perceive and use computers. Moreover, its impact is far beyond the confines of computer science, reaching out to touch many aspects of everyday life, as illustrated by a glance through the colour plates in this book. Computer graphics not only plays an important role in applications such as advertising and entertainment but also in engineering design, medicine, architecture and aerospace. The power of modern computing equipment to generate realistic 3D pictures has also been exploited in the generation of images of often unreal entities. The synthetic production of hitherto only imagined scenarios has found a major application in science fiction films, as well as in medicine and real-time simulation and control.

## Aims and Objectives

This book offers a practical guide to the construction and implementation of computer graphics systems. It does this by:

- demonstrating the power and range of applications of computer graphics,
- presenting the basic principles involved in constructing a computer graphics system capable of these applications, and
- introducing the reader to existing computer graphics systems, in particular the international standard Graphical Kernel System (GKS).

A detailed description, evaluation and critique of GKS as an example of a graphics system and its comparison with other graphics systems, including PostScript, is a major theme throughout the book.

The book aims to provide the reader with a detailed knowledge of the *fundamental programming tools* of computer graphics, making it suitable as a basic tutorial text on computer graphics for computing, mathematics and science undergraduates. It also provides a complete tutorial text on GKS, and the reader should become something of an expert in the understanding of the *design* and *use* of the GKS standard. This will suit students in a range of disciplines, as well as commercial and industrial professionals. Attention is also paid to 'state-of-the-art' approaches such as PostScript, and the

application of graphics in the *design* and *implementation* of *modern user interfaces*. As well as becoming proficient in graphics programming systems, the reader should gain insight into the *design* and *organization* of *graphics hardware*. Here, 'hardware' not only means the computer equipment used to display and interact with images, but also the human visual system that perceives and interprets them.

## An Overview

The *graphics algorithms* discussed in this book are initially developed in an abstract programming notation based on a *functional language*, which can be understood independently of any actual programming language. The proposed *Pascal language binding* for GKS is also used. It is assumed throughout that the reader has some familiarity with basic geometry and computing terminology.

Chapter 1 introduces computer graphics by reviewing some important applications, thus setting the scene for the rest of the book. Part One really begins with Chapter 2, which attempts to extract the basic features of a graphics library system by considering a paradigm based on the printed page. Chapter 3 discusses the architecture of device-independent software systems. Chapter 4 then discusses the fundamental algorithms needed to generate computer graphics. Chapter 5 considers the heart of any graphics system – its functions for graphical output and interaction. In this chapter, graphical output of geometric pure entities (for example, infinitesimally thin lines) is discussed, whereas Chapter 6 introduces methods for specifying style – such as line thickness and colour. Chapter 7 introduces the basic modelling tool of picture segments. Chapter 8 considers some special features of GKS which an applications programmer or systems implementor needs to understand to be aware of GKS in actual operation. Chapter 9 presents an abstract view of the GKS event input model.

Chapter 10 goes beyond GKS by discussing the essential ideas of graphics modelling. This is done by first considering functional graphics, and then widening this to a more traditional approach. In this chapter, the basics of perhaps the most far reaching, recent development in computer graphics is considered, the PostScript system. Many PostScript examples and programs are given in this chapter. There is also a brief discussion of some of the ideas of modelling as embodied in the proposed PHIGS standard.

The revolution in computing mentioned at the beginning of this Preface has at its base the introduction of bitmapped display technology and the concomitant development of powerful personal workstations with bitmapped graphics. The basic concepts of bitmapped graphics are discussed in Chapter 11, based on the Smalltalk-80 model.

Having considered alternative systems in Chapters 10 and 11, Chapter 12 compares these with GKS, as part of a critical assessment of the

standard. The chapter concludes with a strategy for the use of GKS in large-scale projects.

Part One concludes with an introduction to the basic features of 3D graphics. Sufficient information is given in this chapter to enable the reader to construct quite a powerful 3D system.

Another major aspect of the computer revolution has been the growing desire of systems designers to provide interfaces to their products which people can easily learn to use, and which people like to use. Some of the main concerns of human-computer interaction research, and user interface design, are discussed in the opening chapter of Part Two. Chapter 14 concludes with a detailed discussion of the human response to colour.

Chapter 15 moves into the area of graphics hardware, discussing the different types of display technology. Chapter 16 supplements Chapter 15 with a discussion of graphics hardware architecture and organization. Chapter 17 surveys various types of input devices – the physical basis of interactive graphics. Chapter 18 discusses hardcopy devices, including the latest laser printer technology.

Chapter 19 concludes Part Two with an integrating discussion on general ideas of building a graphics system from the first approach of a client to the delivery of the finished product.

Finally, there are two appendices. The first reviews the main ideas of the abstract programming language used in Part One, and the second presents a complete PostScript program for 3D graphics.

## Using the Book in Teaching or Training

Teaching and learning computer graphics is hard work, requiring organization and planning. To do either effectively, those new to computer graphics need to gain considerable practical experience. This requires a laboratory with modern bitmapped (ideally colour) display workstations, which can be used for several hours per week.

An excellent way of re-inforcing the major concepts of computer graphics is to design and implement a complete (although simple) graphics library for interactive graphics, based on a few primitive machine-dependent functions. The student will find the exercise of providing full documentation for the system particularly useful. Further insights can be achieved by tackling the problem of re-implementing the library on a different machine – this re-inforces at an early stage the important issues of device independence. Finally, the student should attempt to write a non-trivial applications program using only the functions of the library. Here, a choice in the type of project can be given: some students are attracted by interactive graphics, with the desire to invent good user interfaces, while others are interested in non-interactive 3D graphics.



This book provides support for a course based on practical work such as that just outlined. It is not a book with lots of programs to generate 'pretty' pictures, but it is one that gives sufficient information to enable a hard-working student to construct a respectable graphics system. It is an introductory book in the sense that many fundamental algorithms and methods are discussed; however, more advanced issues can be followed up by tracing the references.

The book also encourages self-study. A synopsis at the beginning of each chapter outlines the major topics, and links the material with previous and subsequent chapters, putting it into a meaningful context. In addition, each chapter concludes with a summary, pointing to later chapters that build on this material, and exercises to test the reader's understanding and provide relevant project work.

This book represents an integration of the authors' combined experience, and originated from a joint venture to implement and install a large graphics system (GKS). Rod Salmon has had considerable experience in the industrial sector, which has been invaluable in preparing the sections on graphics systems and hardware of Part Two. Mel Slater's experience in computer graphics teaching and research is represented mainly in the chapters forming Part One, on graphics concepts and programming systems.

Writing this book has proved to be a long, sometimes painful, but often enjoyable experience for the authors. The task of producing written material in a form that readers will find enjoyable and understandable is a humbling experience. Computer graphics is not an easy or trivial subject and readers will sometimes have to work hard to understand and apply the ideas presented. The experience of the authors suggests that the effort will be worthwhile.

*Rod Salmon*  
*Mel Slater*  
June 1987

## Acknowledgements

We are grateful to many people who have contributed directly or indirectly to making this book possible. The following people in the Department of Computer Science, Queen Mary College (QMC) have contributed through sharing of ideas about graphics, user-interface design, or programming, or materially in the form of help with equipment: Alan Ball, Don Beal, Richard Bornat, Hilary Buxton, Mike Clarke, Steve Cook, George Coulouris, Allan Davison, Kieron Drake, Ian Page (now with the Programming Research Group, University of Oxford), William Roberts, Antony Simmins, and to all the staff of the Departmental Office. In addition, the following people have critically read one or more chapters: Edwin Blake, Keith Clarke, Peter Hemmingway, Peter Johnson, Siamak Masvani and Steve Reeves.

We further thank Jacques A. C. Halè of Coopers & Lybrand Associates (London), Rob Gale of Sigmex Ltd., and Julian Gallop of the Informatics Division at Rutherford Appleton Laboratories, for critically reading many of the chapters. We are grateful to Mike Milne of Electric Image and Howard Rippiner of Tektronix UK for their invaluable cooperation.

We thank the Addison-Wesley staff, especially the editor-in-chief Sarah Mallen for her patience and encouragement, and the production editor Debra Myson-Etherington for her editorial expertise, enthusiasm and hard work.

We are grateful to all staff and students of the Department of Computer Science at QMC for making it an exciting place in which to teach and research, and to Yvonne Slater for being a patient model. We are indebted to Judith Salmon, for patience beyond the call of duty.

Finally, the help of others has been invaluable, but we alone bear the responsibility for the opinions stated, the technical details and any faults of this book.

The authors and publishers would like to thank the following for providing kind permission to reproduce figures and supplying photographs:

Rendel, Palmer and Tritton, Consulting and Designing Engineers, London (Fig. 1.1); Prime Computer, Inc. (Fig. 1.2); Mullard Ltd. (Fig. 15.2); Tektronix UK (Figs. 15.9, 18.4 and 18.8); Sinclair Research Ltd. (Figs. 15.10 and 15.11); Philips Research Laboratories (Figs. 15.12(b), 15.13 and 15.14); Barco Industries, Belgium (Fig. 15.19); Barco Electronic, N.V. (Fig. 15.20); Interstate Electronic Corporation (Fig. 15.22); GRiD Systems Corporation (Fig. 15.23); Sony UK Ltd. (Fig. 15.25); Westwood Technology Ltd. (Figs. 16.2, 17.1 and 17.5); Sun Microsystems (Figs. 16.14, 19.1 and 19.3); Summagraphics Corporation (Figs. 17.3, 17.6, 17.7 and 17.12);

Marconi Electronic Devices (Fig. 17.8); Evans & Sutherland (Fig. 17.11); Wild Heerbrugg (UK) Ltd. (Figs. 17.13(a) and (b) and 17.14); Hewlett Packard Ltd. (Figs. 18.2 and 18.3); Canon (UK) Ltd. (Fig. 18.5); Facit Ltd. (Fig. 18.6); Versatec (Fig. 18.7).

## Production of the Figures

About half of the figures in Part One, especially those illustrating the output of programs, were produced by an unusual method. Those relating to GKS were produced using a GKS implementation written at Queen Mary College (QMC), Department of Computer Science and Statistics. Initially, the images were 'screen dumped' and the output sent to an Apple LaserWriter. However, pictures that look acceptable on a screen often do not transfer well to paper by screen dumping, because all of the inadequacies of low resolution (for example, jagged lines, poor quality text) are cast in permanent form and easily noticeable (our expectations are higher for the printed page than for computer screens). Thus, an alternative strategy was adopted: a new GKS abstract workstation that delivered PostScript programs in a file was developed. Hence, the GKS program was executed and the result, as well as appearing on the screen, was also represented by a file of PostScript commands. This file was then interpreted on the LaserWriter, with much better results than before.

Other figures were produced by writing PostScript programs directly, or PostScript programs produced by ML. Still others were produced by traditional means, by an illustrator.

The authors would like to thank Siamak Masvani, a postgraduate student at QMC, for writing the GKS programs, in addition to the Department of Computer Science and Statistics, QMC, for use of the equipment – a Sun 3, a Whitechapel Workstations MG1 and the Apple LaserWriter.

## TRADEMARKS

Apple, Macintosh and LaserWriter are trademarks of Apple Computer, Inc. NeWS<sup>TM</sup> and SunView<sup>TM</sup> are trademarks of Sun Microsystems, Inc.

POSTSCRIPT<sup>®</sup> is a registered trademark of Adobe Systems Incorporated.

Smalltalk-80<sup>TM</sup> is a trademark of Xerox Corporation.

UNIX<sup>TM</sup> is a registered trademark of AT & T Information Systems, Inc.

---

# CONTENTS

---

Preface

vii

## PART ONE GRAPHICS PROGRAMMING

<b>Chapter 1</b>	<b>Introducing Computer Graphics and Its Applications</b>	<b>3</b>
1.1	A Brief History	4
1.2	The Applications and Uses of Computer Graphics	5
1.3	Picture Definition – The Graphics Pipeline	20
1.4	The Image-Generating Process	21
1.5	Computers, Computer Systems and Software	23
1.6	Graphics Standards	24
1.7	Image Processing and Image Analysis	27
1.8	The Geometry of Computer Graphics	29
1.9	Programming Notation	29
<b>Chapter 2</b>	<b>A Framework for Computer Graphics</b>	<b>31</b>
2.1	Introduction	32
2.2	Picture Layout	32
2.3	The Basic Elements of an Image	39
2.4	Style in Graphics	45
2.5	Interacting with the Picture	46
<b>Chapter 3</b>	<b>Design Principles for Device-Independent Graphics</b>	<b>53</b>
3.1	Introduction	54
3.2	Graphics Devices	54
3.3	Virtual Graphics Devices	56
3.4	Specification of a Simple Graphics Library	58
3.5	A Virtual Graphics Device for the Library	62
3.6	The Library Implementation	63
3.7	Device Independence	73
3.8	The GKS Workstation Concept	73
<b>Chapter 4</b>	<b>Some Fundamental Algorithms</b>	<b>83</b>
4.1	Mapping Windows to Viewports: Normalization Transformations	84
4.2	The Device Transformation	86
4.3	Clipping Lines	89
4.4	The GKS Viewing Pipeline	92
4.5	The Role of Clipping in GKS	97

4.6	Generating Lines	100
4.7	Polygon Fill	105
4.8	Clipping Polygons	111
<b>Chapter 5</b>	<b>GKS for Interactive Graphics</b>	<b>121</b>
5.1	Graphical Output Primitives	122
5.2	A Model for Graphical Output	127
5.3	Initialisation of Input Devices	134
5.4	The Six GKS Input Classes	137
5.5	GKS Graphical Input in Pascal	145
<b>Chapter 6</b>	<b>GKS for Style</b>	<b>155</b>
6.1	Choosing Colours	156
6.2	Bundles	157
6.3	The Geometric Attributes of Text	166
6.4	Dynamic and Static Binding of Attributes	170
6.5	The Pascal Representation	173
6.6	A Simple User Model for the Drawing Program	176
<b>Chapter 7</b>	<b>Picture Segments</b>	<b>189</b>
7.1	Introduction	190
7.2	Segments	191
7.3	Segments in GKS	198
7.4	Interacting with Segments	214
7.5	The Workstation-Independent Segment Storage	218
7.6	Structured Interactive Drawing – Without Hidden States	221
<b>Chapter 8</b>	<b>GKS Performance and Additional Features</b>	<b>227</b>
8.1	Introduction	228
8.2	Dynamic Modification	228
8.3	Controlling Dynamic Modifications in GKS	239
8.4	Additional Workstation Control Functions	244
8.5	Metafiles: Filing a Picture	247
8.6	Inquiry Functions	250
8.7	Errors in GKS	254
8.8	The Levels of GKS	256
<b>Chapter 9</b>	<b>The GKS Event Input Model</b>	<b>261</b>
9.1	Introduction	262
9.2	Sequential Input and Simultaneous Input	262
9.3	How Events Join the Queue	264
9.4	How Events Leave the Queue	267

9.5	Interrogation of the Queue	269
9.6	Implementation	273
<b>Chapter 10</b>	<b>Graphics Modelling</b>	<b>277</b>
10.1	Introduction	278
10.2	Some Ideas on Functional Graphics	278
10.3	Graphics Modelling in Conventional Languages	291
10.4	PostScript	294
10.5	Graphics Modelling with PHIGS	304
<b>Chapter 11</b>	<b>Bitmapped Graphics and Its Applications</b>	<b>331</b>
11.1	Introduction	332
11.2	Forms	332
11.3	<i>BitBLT</i> , <i>RasterOp</i> and Lines	334
11.4	Rubber Band Techniques	340
11.5	Pop-Up Techniques	343
11.6	Raster-Generated Text	346
11.7	Filling and Clipping	348
11.8	Simple Interactive Curve Drawing	350
11.9	Raster Control Functions in the Computer Graphics Interface	357
<b>Chapter 12</b>	<b>GKS: A Critical Assessment</b>	<b>365</b>
12.1	Introduction	366
12.2	GKS Efficiency	366
12.3	GKS Complexity	368
12.4	The Paucity of GKS Graphical Output	371
12.5	A Strategy for the Use of GKS	374
<b>Chapter 13</b>	<b>Basic Ideas of 3D Graphics</b>	<b>379</b>
13.1	Introduction	380
13.2	Coordinate Systems	381
13.3	Basic Ideas of Object Representation	382
13.4	3D Transformations	390
13.5	Viewing	397
13.6	Basic Concepts of Shading	416
13.7	Anti-Aliasing	420
13.8	Visibility	426
13.9	The World Is Not Made of Polygons	436
13.10	Standards	446

## **PART TWO   GRAPHICS SYSTEMS AND INTERFACE**

<b>Chapter 14</b>	<b>Some Aspects of the Human–Computer Interface</b>	<b>455</b>
14.1	Introduction	456
14.2	General Considerations for a Good HCI	457
14.3	Task Analysis	460
14.4	User Conceptual Models	465
14.5	Styles of Interaction	474
14.6	Some Specific Interaction Methods and Techniques	479
14.7	The User–Display Interface	484
14.8	The Display Image and the Eye	500
14.9	Display Image Composition	505
<b>Chapter 15</b>	<b>Electronic Displays</b>	<b>511</b>
15.1	Introduction	512
15.2	Cathode Ray Tubes	512
15.3	Video Monitors	527
15.4	Flat Panel Displays	536
15.5	Non-Emitter Displays	541
<b>Chapter 16</b>	<b>Graphics Image Generators</b>	<b>547</b>
16.1	Introduction	548
16.2	Graphics Image Generation	548
16.3	Vector Refresh Graphics Display Systems	555
16.4	Raster Graphics Display Systems	558
16.5	Communication with a Graphics Generator	574
<b>Chapter 17</b>	<b>Graphics Input Devices</b>	<b>579</b>
17.1	Introduction	580
17.2	Direct Input Devices	580
17.3	Cursor Drivers	583
17.4	Direct Screen Interaction	587
17.5	Special Input Devices	590
17.6	Logical Input Devices	591
17.7	Three-Dimensional Input	595
<b>Chapter 18</b>	<b>Hardcopy Devices</b>	<b>601</b>
18.1	Introduction	602
18.2	Pen Plotters	602
18.3	Character and Line Printers	604
18.4	Ink Jet Printers	606
18.5	Electrophotographic Printers	607

18.6	Laser Displays	610
18.7	Photographic Systems	611
<b>Chapter 19</b>	<b>Building a Computer Graphics System</b>	<b>613</b>
19.1	Introduction	615
19.2	Elements of a System	615
19.3	The Draughtsman's Contract	621
<b>Appendix A</b>	<b>Programming Notation</b>	<b>633</b>
<b>Appendix B</b>	<b>A PostScript Program for 3D Viewing</b>	<b>657</b>
<b>References</b>		<b>671</b>
<b>Index</b>		<b>679</b>



PART ONE

# GRAPHICS PROGRAMMING