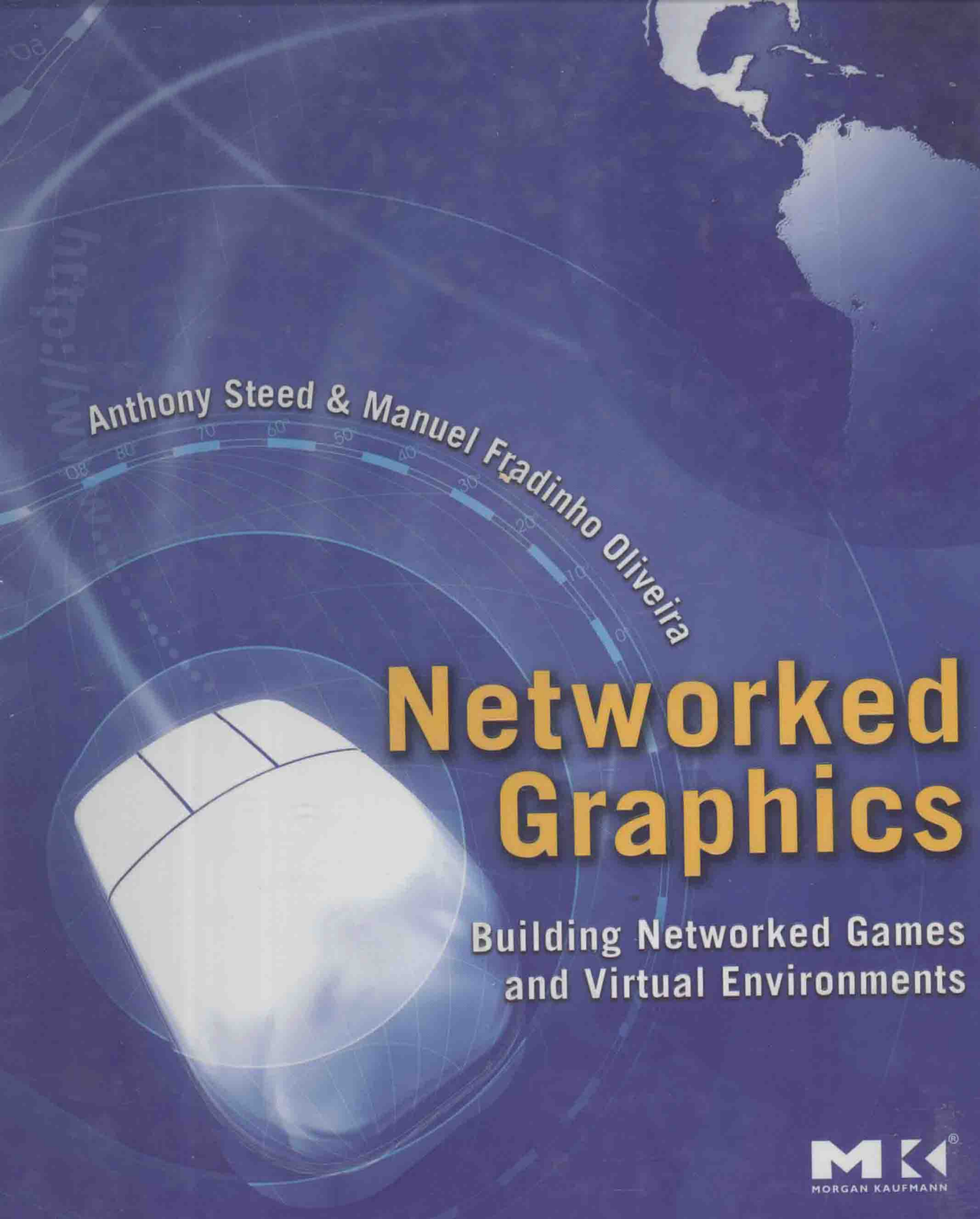




Anthony Steed & Manuel Fradinho Oliveira

Networked Graphics

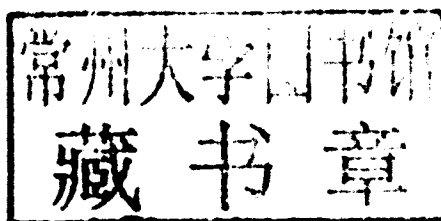
Building Networked Games
and Virtual Environments

Networked Graphics

Building Networked Games and Virtual Environments

Anthony Steed

Manuel Fradinho Oliveira



AMSTERDAM • BOSTON • HEIDELBERG • LONDON • NEW YORK
OXFORD PARIS • SAN DIEGO • SAN FRANCISCO • SINGAPORE •
SYDNEY • TOKYO

Morgan Kaufmann Publishers is an imprint of Elsevier



Morgan Kaufmann Publishers is an imprint of Elsevier.
30 Corporate Drive, Suite 400, Burlington, MA 01803, USA

This book is printed on acid-free paper.

© 2010 Elsevier company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from the publisher. Details on how to seek permission, further information about the Publisher's permissions policies and our arrangements with organizations such as the Copyright Clearance Center and the Copyright Licensing Agency, can be found at our website: www.elsevier.com/permissions.

This book and the individual contributions contained in it are protected under copyright by the Publisher (other than as may be noted herein).

Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

Library of Congress Cataloging-in-Publication Data

Application submitted

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library.

ISBN: 978-0-12-374423-4

For information on all Morgan Kaufmann publications,
visit our Web site at www.mkp.com or www.elsevierdirect.com

Printed in Canada

09 10 11 12 13 5 4 3 2 1

Working together to grow
libraries in developing countries

www.elsevier.com | www.bookaid.org | www.sabre.org

ELSEVIER

BOOK AID
International

Sabre Foundation

Networked Graphics

Contents

PART I GROUNDWORK

CHAPTER 1	Introduction.....	3
1.1	What are NVEs and NGs?	4
1.2	The Illusion of a Shared Virtual Environment	5
1.3	Some History	7
1.3.1	Internet.....	8
1.3.2	Simulators	15
1.3.3	Multiuser Dungeons	19
1.3.4	Electronic Games.....	22
1.3.5	Virtual Reality Systems.....	27
1.3.6	MMORPGs	30
1.3.7	Social Spaces.....	33
1.4	Scoping the Software Architecture	36
1.4.1	Software Roles and Responsibilities	38
1.4.2	Differences to Other Network Code	39
1.5	Structure.....	40
CHAPTER 2	One on One (101)	45
2.1	Boids.....	45
2.1.1	Boid Rules	46
2.1.2	Standalone Implementation.....	48
2.2	Distributed Boids: Concepts	49
2.3	Distributed Boids: Implementation	53
2.3.1	Revised Simulation and Data Model	53
2.3.2	Network Concepts	55
2.3.3	Network Protocol	56
2.3.4	Implementation of the Network Component	56
2.3.5	Sending UDP Data Packets	58
2.3.6	Sending TCP Data Streams.....	62
2.4	Reflection	65
CHAPTER 3	Overview of the Internet	71
3.1	The Internet	71
3.2	Application Layer	75
3.2.1	Application Layer Protocols	76
3.2.2	Application QoS.....	77
3.2.3	Common Applications and Ports.....	78
3.2.4	RFCs.....	80
3.2.5	DNS.....	80
3.2.6	Telnet and HTTP.....	82

3.3	Transport Layer	83
3.3.1	Implementing UDP	84
3.3.2	Basics of TCP	87
3.3.3	Reliability in TCP	89
3.3.4	Opening and Closing TCP Connections	96
3.3.5	Flow Control and Congestion Avoidance in TCP	99
3.4	Network Layer	101
3.4.1	Network Address Translation	103
3.4.2	IP Packets	104
3.4.3	Ping, Traceroute & ICMP	109
3.4.4	Routing on the Internet	112
3.4.5	DHCP	112
3.5	Link and Physical Layer	113
3.5.1	Ethernet	115
3.5.2	Comparisons	116
3.6	Further Network Facilities	116
3.6.1	Multicast	117
3.6.2	Network QoS	118
3.7	Summary	121
CHAPTER 4	More Than Two	125
4.1	Boids	126
4.1.1	Architecture	126
4.1.2	Protocol	127
4.1.3	Codebase Refactoring	128
4.2	Simple Peer to Peer	130
4.2.1	System Architecture	131
4.2.2	Data Model	132
4.2.3	Network	134
4.2.4	Protocol	135
4.3	Peer to Peer with Master	138
4.3.1	System Architecture	139
4.3.2	Datamodel	140
4.3.3	Network	142
4.3.4	Protocol	142
4.4	Peer to Peer with Rendezvous Server	146
4.4.1	System Architecture	148
4.4.2	Datamodel	149
4.4.3	Network	149
4.4.4	Protocol	150
4.5	Client/Server	152
4.5.1	System Architecture	153
4.5.2	Datamodel	154
4.5.3	Network	155
4.5.4	Protocol	156

4.6	Multicast	158
4.6.1	System Architecture	160
4.6.2	Datamodel	161
4.6.3	Network	162
4.6.4	Protocol	164
4.7	Extensions	166
4.7.1	Protocol design	166
4.7.2	Network Compensation	166
4.7.3	Scale	167
4.7.4	Binary Protocols	167
4.8	Conclusions	167

PART II FOUNDATIONS

CHAPTER 5 Issues in Networking Graphics 171

5.1	Architecture of the Individual System	171
5.1.1	Engines and Engine Framework	173
5.1.2	Modules	177
5.1.3	Content	177
5.1.4	Behavior	178
5.2	Role of the Network	179
5.2.1	Device I/O Sharing	180
5.2.2	Data Model Sharing	181
5.2.3	Command/Event Streaming	182
5.2.4	Coping with Scale	184
5.3	Initialization	184
5.3.1	Case Study: Disney's Toontown	185
5.4	Server and Peer Responsibilities	186
5.5	Critical and Noncritical	188
5.6	Synchronized or Unsynchronized	189
5.7	Ownership and Locking	190
5.8	Persistency	190
5.9	Latency and Bandwidth	191
5.10	Conclusions	192

CHAPTER 6 Sockets and Middleware 195

6.1	Role of Middleware	195
6.2	Low-Level Socket APIs	196
6.2.1	Main Functions	196
6.2.2	Name Helper Functions and Options	199
6.2.3	Network Byte Order	201
6.2.4	Examples	202
6.2.5	Socket Usage Patterns	207
6.2.6	Windows	207

6.2.7 Synchronous I/O.....	208
6.2.8 Critique and Discussion.....	210
6.3 C and C++ Middleware for Networking.....	210
6.3.1 HawkNL.....	211
6.3.2 SDL_Net.....	211
6.3.3 ACE.....	213
6.4 Conclusion	215
CHAPTER 7 Middleware and Message-Based Systems	217
7.1 Message-Based Systems	217
7.2 DIS.....	218
7.2.1 PDUs	219
7.3 X3D and DIS.....	220
7.3.1 Basics of X3D	222
7.3.2 Dataflow and Animation	224
7.3.3 Scripting and Interfaces.....	227
7.3.4 VRML/X3D Networking Strategies	232
7.3.5 X3D and DIS	235
7.4 X3D, HawkNL and DIS.....	237
7.4.1 Sending a DIS Packet	238
7.4.2 Receiving DIS Packets	238
7.5 Conclusions.....	241
CHAPTER 8 Middleware and Object-Sharing Systems	247
8.1 Object-Sharing Systems	247
8.2 RakNet.....	250
8.2.1 Object Lifecycles	251
8.3 Boids using Object-Sharing	252
8.3.1 Boid Definitions	253
8.3.2 Main Application	260
8.3.3 Discussion.....	264
8.4 General Object-Sharing	264
8.4.1 Sharing Policy	264
8.4.2 Visibility and Sampling Rates.....	265
8.5 Ownership	266
8.6 Scene-Graphs, Object-Sharing and Messages	271
8.7 Conclusions.....	273
CHAPTER 9 Other Networking Components.....	275
9.1 Remote Method Call.....	275
9.1.1 RPC	276
9.1.2 XML-RPC, etc.	284
9.1.3 CORBA.....	285
9.1.4 Discussion.....	287
9.2 DIVE	288

9.2.1 DIVE Overview	288
9.2.2 DIVE Entities	291
9.2.3 Events	297
9.2.4 Scripting and Event Scoping	299
9.2.5 Networking	301
9.2.6 Discussion	302
9.3 System Architectures	303
9.4 Conclusions	306

PART III REAL SYSTEMS

CHAPTER 10 Requirements	313
10.1 Consistency	314
10.1.1 System Perspective	315
10.1.2 Plausibility & Fairness	316
10.1.3 Consistency-Throughput Tradeoff	318
10.2 Latency and Jitter	319
10.2.1 Sources of Network Latency	319
10.2.2 Jitter	321
10.2.3 Clocks and Measurement of Latency and Jitter	324
10.2.4 User Tolerance	326
10.2.5 Latency Impact in Specific Systems	329
10.3 Bandwidth	330
10.3.1 Measurement	331
10.3.2 Traffic Patterns	331
10.3.3 Congestion	336
10.4 State of the Internet	337
10.5 Connectivity	340
10.5.1 NAT	340
10.6 Case Study: Burnout™ Paradise	343
10.6.1 Car “Mechanics”	344
10.6.2 Game Phases and Time Synchronization	345
10.6.3 Game Hosting and Peer-to-Peer Networking	347
10.7 Conclusions	348
CHAPTER 11 Latency and Consistency	355
11.1 Latency Impact	355
11.2 Dumb Client and Lockstep Synchronization	359
11.2.1 Dumb Client	359
11.2.2 Lockstep Synchronization	360
11.3 Conservative Simulations	360
11.4 Time	362
11.5 Optimistic Algorithms	365

11.6	Client Predict Ahead	367
11.7	Extrapolation Algorithms	371
11.7.1	Basic Dead-Reckoning	371
11.7.2	Dead-Reckoning Variants.....	376
11.8	Interpolation, Payout Delays and Local Lag	378
11.9	Local Perception Filters.....	383
11.10	Revealing Latency	387
11.11	Conclusions.....	388
CHAPTER 12	Scalability.....	393
12.1	Service Architectures	394
12.1.1	Awareness	394
12.1.2	Cohorts, Groups and Service Architecture.....	396
12.2	Overview of Interest Management.....	399
12.3	Spatial Models	403
12.3.1	Zone Models	404
12.3.2	Dynamic Spatial Models	406
12.3.3	Visibility Models	408
12.4	Interest Specification and Interest Management.....	411
12.4.1	Interest Expressions	411
12.4.2	Interest Management.....	413
12.4.3	Examples of Distributed Interest Management	415
12.5	Separating Interest Management from Network Architecture	418
12.5.1	Message Routing and Message Filtering	418
12.5.2	Filtering as a Pipeline	420
12.5.3	Routing and Filtering Examples	421
12.5.4	Delegating Group Communication to the Network Layer.....	423
12.6	Server Partitioning.....	424
12.6.1	Server Tasks and Static Partitionings.....	425
12.6.2	Seams and Handovers.....	428
12.6.3	Load Balancing and Dynamic Partitioning.....	430
12.7	Group Communication Services	434
12.7.1	Reflectors and Tunnels	434
12.7.2	Automatic Overlay Multicast	435
12.7.3	Gossiping for Peer-to-Peer Communications.....	436
12.7.4	Specific Group Communication Routing Service	437
12.8	Peer to Peer.....	437
12.8.1	Unstructured Peer-to-Peer Systems.....	438
12.8.2	Structured Peer-to-Peer Systems.....	439
12.8.3	Peer-to-Peer Support Using DHTs.....	442

12.8.4 Local Interest Systems	443
12.8.5 Update-Free Regions and Frontier Sets	446
12.9 Conclusions	449
CHAPTER 13 Application Support Issues	459
13.1 Security and Cheating	459
13.1.1 Client-Side Attacks	461
13.1.2 Network Level	463
13.1.3 Server Side	465
13.1.4 Social	465
13.1.5 Repercussions	466
13.2 Binary Protocols and Compression	467
13.2.1 Binary Encodings	467
13.2.2 Compression	468
13.3 Streaming	470
13.3.1 Audio and Video	470
13.3.2 Geometry and Animation	473
13.3.3 Mixing and Grouping	476
13.4 Revisiting the Protocol Decision	477
13.5 Persistent and Tiered Services	478
13.6 Clusters	479
13.6.1 Multidisplay Systems	479
13.6.2 Massive Models	482
13.7 Thin Clients	482
13.8 Conclusions	483
 Index	 489

Groundwork

I

1	Introduction	3
2	One on One	45
3	Overview of the Internet	71
4	More Than Two	125

Introduction

1

Over the last four decades the Internet has radically changed many forms of collaborative activity. Email and more recently instant messaging have enabled efficient asynchronous collaboration between remote people. The World-Wide Web (WWW) has enabled a range of different publishing models for individuals as well as large organizations. More recently though, the Internet has enabled new types of real-time synchronous communication between people. Synchronous collaboration tools include video and audio tools, but also network games (NGs) and networked virtual environments (NVEs). Within NVEs and NGs, users can share a virtual space with business partners to brainstorm or they can be immersed¹ in a fantasy world to go exploring with friends.

This book describes the data communication technologies behind these NGs and NVEs. We focus exclusively on Internet technologies because of their pervasive nature, though we'll present some of the historical context behind the development of the Internet. The book takes a ground-up approach from basics of networking through to strategies for partitioning large numbers of players between servers. Thus we've tried to write the book for a few different audiences: students interested in networking aspects of games and simulations, software engineers getting into the field, game developers involved in the implementation of NGs, hobbyists or games enthusiasts interested in learning more about games technology and researchers interested in a general background text.

In this introduction we want to present the outline of the book and give some background history to the topic.

¹Immersive is a term used often to describe computer and video games that are rich, engaging and interactive. In the area of virtual reality systems, the term is used differently to mean displays that encompass and surround the user, see Section 1.3.5.

1.1 WHAT ARE NVES AND NGs?

With NVEs, we refer to virtual environment systems that are distributed over the network in some sense. That is, usually, there are several computers, each running a piece of software that communicates with similar software on other computers. Users interact with an interface facilitated by devices and the software interact with the virtual environment, perhaps by moving a player character through the virtual environment. If this is for recreational purposes then the whole system might be called a NG, though these would be a subset of the applications that use such technology, which range from medical simulations through education to military training. In effect, there is often little to distinguish between systems that support NVEs and systems that support NGs, other than the type of content that is created for them. Some systems are designed specifically for recreational purposes, being based on fantasy or cartoon styles such as World of Warcraft™ or Disney's Toontown. Other systems are neutral to use and leave the content creation and its application context to the user. For example, Linden Lab's SecondLife® contains both business centers and nightclubs.

The common feature of both NVEs and NGs, and an important piece of scope for this book, is that by virtual environment, we refer to a rich three-dimensional (3D), or less commonly 2D, space that depicts a real or imaginary place. The client software allows the user to move about this space to get new viewpoints on that space. The space is probably displayed to the user at real-time (30Hz plus) rate, is animated, has interactive elements to it, and the reality it represents is governed by well-defined rules (e.g. Brownian motion, gravity or refraction). Depending on the type of user interaction supported, the users may be represented in some way.

Figure 1.1 shows a typical NVE which one of the authors had a small hand in developing. This particular system, Distributed Interactive Virtual Environment (DIVE), was a research prototype built by Swedish Institute of Computer Science (SICS) (Frécon et al, 2001). Although the particular version of the software shown in Figure 1.1 was released in 1999, it has many features of the current crop of online social NVEs: it has text chat, large worlds to visit, supports up to around 30 players in a single location and has audio and video streaming. Most importantly for our introduction, the system has *avatars*, which is the term commonly used in the field to refer to the visual representation of a user in the world.² Avatars indicate your location in the world to other people and provide the user with a reference to their interaction model with the environment. Different systems support different visual styles of avatars, from abstract, such as the DIVE avatars, through cartoony to realistic-looking avatars. In DIVE audio communication is enabled by having avatars stand near each other in the virtual environment. Users can select and manipulate

²The term avatar derives from the Sanskrit word in Hindu texts, where it means incarnation. The first use in a computing context to refer the representation of a user is not well-known, but it was perhaps coined in Habitat (see later), and popularized in Neal Stephenson's novel "SnowCrash".

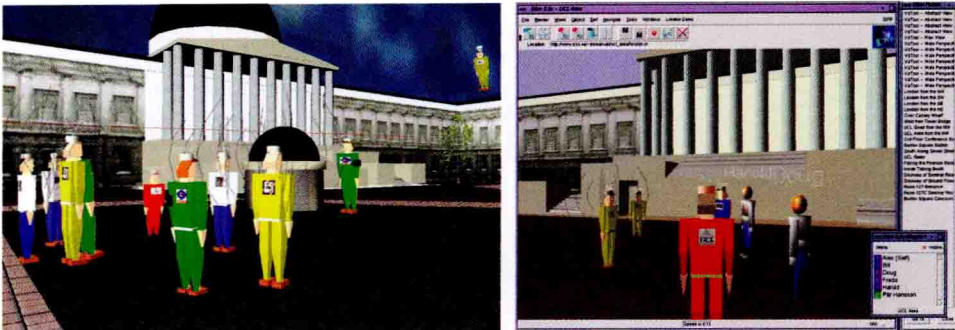


FIGURE 1.1

Two scenes from the DIVE system. Left: a first-person view of a group visiting a virtual University College London. Right: a third-person view of the same environment showing some of the desktop user-interfaces for voice, navigation and editing

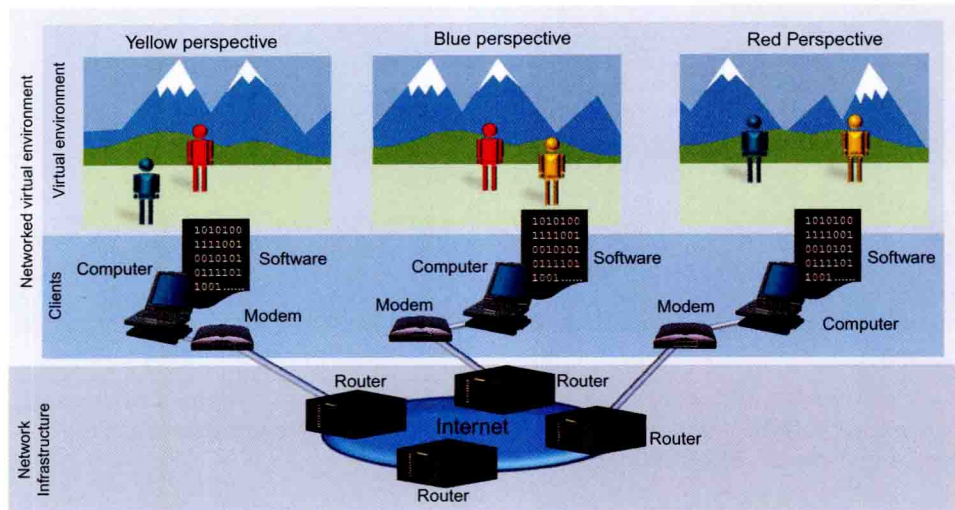
many of the objects in the world near to them. We discuss DIVE in more detail in Section 9.3.

1.2 THE ILLUSION OF A SHARED VIRTUAL ENVIRONMENT

The foundation of a NVE is to create the illusion of a virtual environment that is shared amongst all participating users. As illustrated in Figure 1.2, the NVE consists of multiple collaborating NVE *client computers* (or just *client*), where each client typically runs on a single workstation or home computer.³ A client is composed of one or more pieces of software that are responsible for key functions such as generating sensorial outputs (e.g. video), processing inputs (e.g. joysticks) and doing physics simulations. The client computers are connected together with a network infrastructure (the Internet being the most pervasive with global connectivity) through modems and/or routers. A more detailed overview of components of the NVE system itself is given in Section 1.4.

As illustrated by the diagram of Figure 1.2, the aim of a NVE system is not to create a single-user experience, but a shared experience amongst all participating users. Each user in the system is seeing a different view, but these views should be consistent. From a technical point of view this implies that each NVE system shares a common model of the virtual environment. As we will learn, this is extremely difficult over networks where there are real problems with latency, congestion, etc. Thus each client in the system has a slightly different model, and thus the renderings of the virtual environments at each client are all different. This is why this section is entitled the *illusion* of a shared virtual environment. Fortunately for us, users are

³We will talk about various types of clusters for graphics in Chapter 13.

**FIGURE 1.2**

Illusion of a shared virtual environment

sometimes unaware or at least tolerant of the discrepancies between each client's view of the virtual environment. Users see avatars moving around the virtual environment and interacting with objects, including other avatars. They can often talk to the other users or at least send text messages. Though the media might seem crude, users nonetheless interact successfully with one another. As long as users can experience the virtual environment in such a way that they can agree on the important features of the world (e.g. the placement of enemies in a game or simulation), then they can have the common understanding of sharing the same virtual environment. It is when things start to get desynchronized that illusion of a shared space becomes hard to maintain, and users spend a lot of time talking or texting in an attempt to come to a shared understanding of their situation.⁴

The network infrastructure provides the necessary support for the sharing of information to keep a consistent perspective amongst all participating users. How exactly this is done will be discussed throughout the remainder of the book: but aside from the technical details of what must be done, we can already posit that there is going to be a bottleneck in that the Internet is not sufficiently fast to simply copy all changing information to all participants. The network infrastructure needs to choose carefully what is sent over the Internet. In doing so, it will necessarily take

⁴One of the most fascinating aspects of NVEs, which we will only have space to touch on very briefly in the book, is that users seeing avatars tend to treat them as if they were real people. In social NVEs, avatars tend to form social groups in small circles, as they might do in conversation in the real world. There is a large body of work in the overlap between computer science and sociology (see Schroder, 2001; Churchill et al., 2001 and Schroeder and Axelsson, 2006 for overviews).