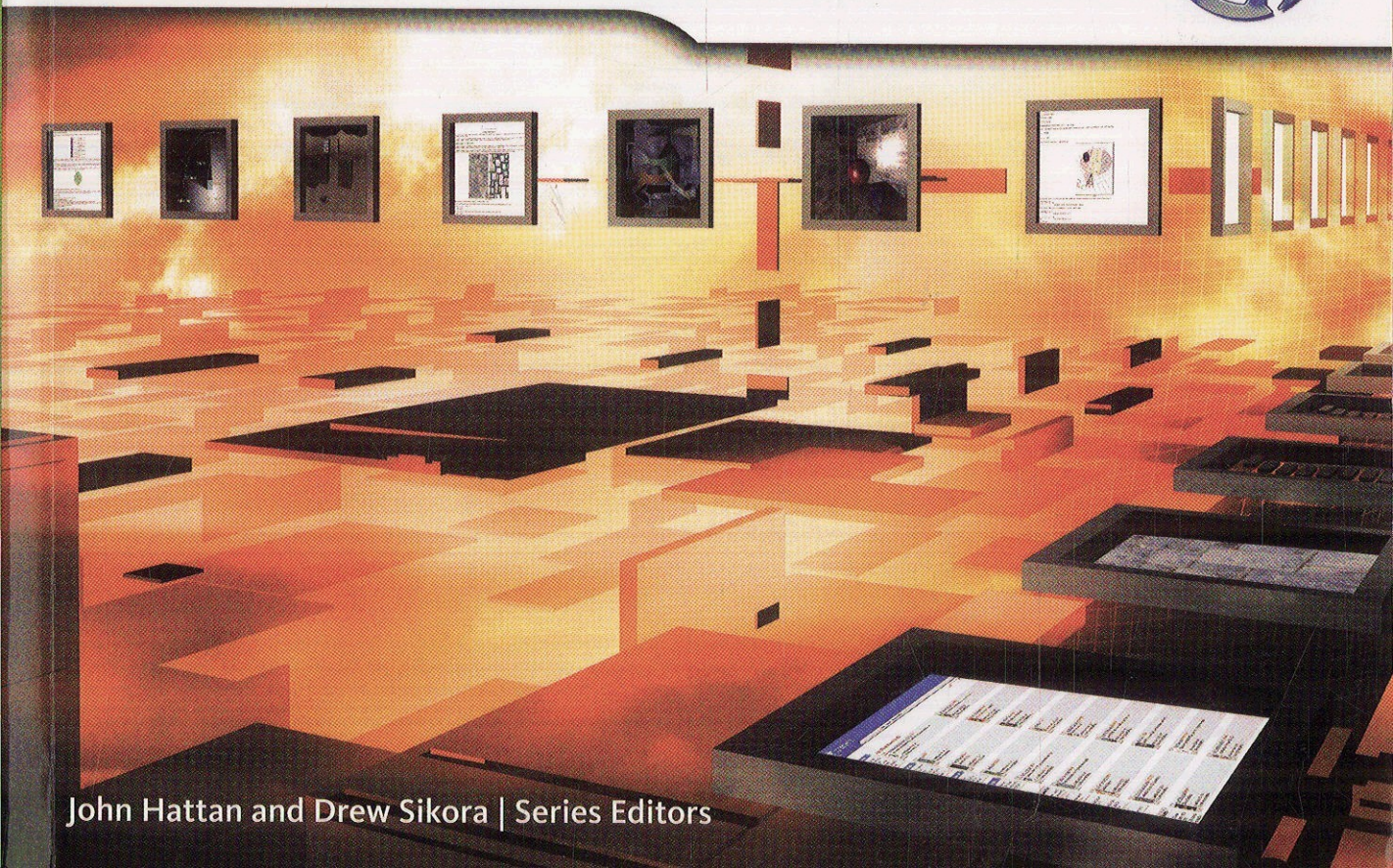


Advanced Game Programming

a gamedev.net collection



John Hattan and Drew Sikora | Series Editors

CONTENTS

	Foreword	xviii
PART 1	ALGORITHMS	1
Chapter 1	2D Car Physics	3
	<i>Matt Kincaid</i>	
	Math Requirements	3
	Phase One: Main Rendering Function	4
	Forms Wiring	6
	The Timer	6
	Conclusion of Phase One	7
	Phase Two: Rigid Body Simulation	11
	Code Dump	13
	Testing	16
	Phase Three: The Vehicle	16
	Force Calculation	18
	Almost Done!	19
	Conclusion	19
Chapter 2	3D Matrix Math Demystified	21
	<i>Seumas McNally</i>	
	Extending to Affinity	24
	Going in Reverse	25

Multiplying Matrices	26
The End, For Now	27

Chapter 3 Multi-Threaded and Distributed Computing with the Actor Model 29

Michael Sikora

Multi-Threaded and Distributed Computing: Brothers or Second Cousins?	30
The Actor Model	31
When to Apply This Model	32
Implementation Details	34
Example Problem: Using Client Actors as Computer Players	38
Extending the Framework with an Address Space	41
Dynamic Load Balancing	42
Conclusion	43
References	44

Chapter 4 Grid-Based Collision Detection and Raycasting 45

Metanet Software Tutorial

SECTION 0: General Introduction	45
SECTION 1: Basic Tile Grid	46
Collision vs. a Tilemap Using the Grid	47
SECTION 2: Advanced Tile Grid	48
Grid Improvements	48
Edge Info	48
SECTION 3: Object Grid	50
More Details	52
SECTION 4: Raycasting	53
Broad Phase	53
Narrow Phase vs. Tiles	54
Narrow Phase vs. Objects	54
SECTION 5: Conclusion	55
Source Code	55
References	56

Chapter 5 In-Memory Data Compression and Decompression 57

Lee Millward

Part One: Compression	57
Part Two: Decompression	59

Chapter 6 Finite State Machines and Regular Expressions 61

Eli Bendersky

Finite State Machines—What Are They? 61

 Finite? 63

 Employing State Machines 64

Regular Expressions 64

 Recognizing Strings with Regexes 65

 FSMs to the Rescue 66

 Coding It the FSM Way 68

DFA + NFA = FSM 71

 NFA 71

 DFA 72

 Recognizing Regexes with DFAs and NFAs 72

Construction of an NFA from a Regular Expression 74

 An Example 76

Implementation of a Simple NFA 77

Implementing Thompson’s Construction 81

 The Simplest Regular Expression 81

 Some Changes to the NFA Class 82

 Implementing Alternation: `alb` 84

 Implementing Concatenation: `ab` 85

 Implementing Star: `a*` 85

 Specialty of NFAs Constructed by Thompson’s Construction 86

 A Complete NFA Construction Implementation 86

 Even Closer to Complete Automation 86

 From a Parse Tree to an NFA 87

 From a Regex to a Parse Tree 89

Converting NFAs to DFAs 89

eps-closure 91

 move: A New NFA Member Function 91

 Keeping Track of the Input Language of an NFA 92

 DFA Implementation 92

 Subset Construction 93

Conclusion 96

Chapter 7 Memoization: A Technique to Speed Up Recursive Computations 99

Eli Bendersky

Fibonacci Numbers 99

Memoized Fibonacci 102

 Alternative Fibonacci Implementations 103

	Counting Change	104
	Memoized Change Counting	107
	Wrapping Up	109
PART 2	AI	111
Chapter 8	Application of Genetic Programming to the Snake Game . . .	113
	<i>Tobin Ehlis</i>	
	Introduction and Overview	113
	Background	114
	Statement of the Problem	116
	Methods	117
	Terminals	117
	Functions	118
	Fitness Cases	121
	Fitness Measure	122
	Parameters	122
	Designating a Result and Criterion for Terminating a Run . . .	122
	Crossover and Mutation Rates	122
	Computer Equipment and Runtime	123
	Schemata	123
	Results	124
	Zig-Zagger	125
	Wall-Slitherer	128
	Circler	131
	Pattern-Following Solution	133
	Conclusion	135
	Future Work	136
	References	136
Chapter 9	Precalculated Pathfinding	137
	<i>Richard Fine</i>	
	Initial Concept	138
	Foundations	139
	Dynamic Worlds	141
	Education	146
	Progressive Nets and Subnets	148
	Echoes of the Past	148
	Conclusion	149

PART 3	GRAPHICS	151
Chapter 10	Axonometric Projections: A Technical Overview	153
	<i>Thiadmer Riemersma</i>	
	Introduction—First Attempt	153
	Introduction—Second Attempt	155
	Introduction—Third Attempt	156
	And Now, Onward.	157
	The Isometric Projection	158
	The Dimetric Projection	160
	Dimetric Projections for Computer Graphics and Games	162
	Moving Across an Axonometric Projection	164
	References and Further Information	166
Chapter 11	Real-Time Realistic Cloud Rendering and Lighting	169
	<i>Andrei Stoian</i>	
	Harris' Model for Cloud Rendering and Lighting	170
	Cloud Lighting	170
	The Phase Function: Scattering in the Eye Direction	174
	Creating Cloud Impostors	174
	Creating the Splat Texture	176
	From Impostors to Full 3D Models	177
	Conclusion	179
	References	179
Chapter 12	A Closer Look at Parallax Occlusion Mapping	181
	<i>Jason Zink</i>	
	Algorithm Overview	182
	Implementing Parallax Occlusion Mapping	185
	Algorithm Metrics	190
	Conclusion	194
Chapter 13	Quadtrees	195
	<i>Jonathan Ferraris</i>	
	Test 1	201
	Test 2	201
	Test 3	203
	Test 4	203
	Test 5	203

	Test 6	205
	Test 7	206
	Putting It Together: Coding a Quadtree	206
Chapter 14	Efficient Normal Computations for Terrain Lighting in DirectX 10	217
	<i>Jeromy Walsh</i>	
	Height-Map-Based Terrain	218
	Slope Method of Computing Heightfield Normals	220
	Implementing the Algorithm with DirectX 10	223
	Mesh-Based Terrain	228
	Grid-Mesh Smooth Shading Algorithm	228
	Implementing the Algorithm with DirectX 10 (Again)	230
	Conclusion	237
	References	238
Chapter 15	Continuous Collision Detection for Translating Ellipsoid	239
	<i>Chandan Pawaskar</i>	
	CCD for Ellipsoids	240
	Appendix A: Point on Ellipsoid Closest to Given Plane	243
	Appendix B: Point-in-Triangle Test	245
	Appendix C: Distance Between Point and Line Segment	247
	Appendix D: Ray-Ellipsoid Intersection	248
Chapter 16	Soft-Edged Shadows	251
	<i>Anirudh S. Shastry</i>	
	A Closer Look	251
	Soft Shadows	252
	So How Does It Work?	252
	Step 1: Rendering the Shadow Map	253
	Step 2: Rendering the Shadowed Scene into a Buffer	255
	Step 3: Blurring the Screen Buffer	257
	Step 4: Rendering the Shadowed Scene	262
	References	264
Chapter 17	Box Filtering Height-Maps for Smooth Rolling Hills	265
	<i>Graham Wihlidal</i>	
	Box Filtering Technique	265
	Algorithm Example	267
	Screenshots	269
	Conclusion	270

Chapter 18	Exploring Metaballs and Isosurfaces in 2D	273
	<i>Stephen Whitmore</i>	
	Overview	273
	What Are Metaballs?	274
	Creating Meta-Things	276
	A Simple 2D Implementation	276
	Equation of a Metaball	277
	Writing a 2D Implementation	279
	Other Meta-Shapes	281
	Ellipses	281
	Diamonds	282
	Donuts	282
	Optimizations and Improvements	283
	Uniform Box Division	284
	Equation Simplification (Square Root)	284
	More Optimizations and Techniques	285
	Above and Beyond	286
	3D Isosurfaces	286
	Isosurfaces in the Real World	286
	The Meta Playground	287
	References	287
Chapter 19	Dynamic 2D Soft Shadows	289
	<i>John Campbell</i>	
	Overview	289
	Light Source	289
	Shadow Caster	290
	Light Range	290
	Umbra	290
	Penumbra	290
	Core Classes	291
	Rendering Overview	292
	Simple Light Attenuation	292
	Colored Lighting	296
	Hard-Edged Shadow Casting	296
	Finding the Boundary Points	297
	Creating the Shadow Geometry	297
	Soft-Edged Shadow Casting	299
	Shadow Fins	299

	Non-Linear Shading	300
	Modifying the Umbra Generation	302
	Making It Robust	302
	Self-Intersection	302
	Eliminating “Popping”	303
	Depth Offset	303
	Emissive/Self-illumination Pass	304
	Scissor Testing	305
	Conclusion	305
PART 4	NETWORKING	307
Chapter 20	Multicasting for Multiplayer Games	309
	<i>Denis Lukianov</i>	
	The Idea Behind Multicasting	309
	How Multicasting Works	310
	Player Discovery	311
	The Darker Side	312
	Receiving Multicasts (IPv4)	313
	Sending Multicasts (IPv4)	315
	Receiving Multicasts (IPv6)	316
	Sending Multicasts (IPv6)	318
	Integrating Multicast Support into a Game	318
PART 5	C++	321
Chapter 21	Debugging in Visual Studio—Part 2	323
	<i>Steve Jones</i>	
	What Are “Bugs”?	323
	Simple Debugging Steps Defined	325
	Identify and Stabilize	325
	Gather Behavioral Data	326
	Assertions	326
	Exceptions	329
	Visual Studio Debugging	330
	Stepping Through Code	331
	Pause/Resume/Stop	331
	Step Into/Step Over/Step Out	331
	Set Next Statement	332
	Show Next Statement	333

Run to Cursor	333
Breakpoints	333
Conditional Breakpoints	334
Viewing the State of Code	336
State Monitor Windows	336
Summary	339
References	339

Chapter 22 20 Issues of Porting C++ Code on the 64-Bit Platform ... 341

Andrey Karpov and Evgeniy Ryzhkov

Introduction	341
Disabled Warnings	344
Use of Functions with a Variable Number of Arguments	344
Magic Numbers	345
Storing Integers in double Type	347
Bit-Shifting Operations	348
Storing Pointer Addresses	350
memsize Types in Unions	351
Changing an Array Type	353
Virtual Functions with Arguments of memsize Type	354
Serialization and Data Exchange	354
Using Types of Volatile Size	355
Ignoring the Byte Order	356
Bit Fields	357
Pointer Address Arithmetic	358
Array Indexing	360
Mixed Use of Simple Integer Types and memsize Types	362
Implicit Type Conversions in Functions	365
Overloaded Functions	366
Data Alignment	367
Exceptions	370
Using Outdated Functions and Predefined Constants	370
Explicit Type Conversions	371
Error Diagnosis	371
Unit Testing	371
Code Review	372
Built-In Means of Compilers	372
Static Analyzers	372
Conclusion	373
References	374

Chapter 23	Writing Endian-Independent Code in C++	375
	<i>Promit Roy</i>	
	When Does the Endian Affect Code?	376
	Writing Endian-Independent Code	377
	Step 1: Switching Endians	377
	Step 2: Set Function Pointers to Use the Correct Swap Function	379
	Step 3: Initialization	380
	A Practical Demonstration	381
Chapter 24	Test Driving Expression Template Programming	383
	<i>Kent Lai Shiao San</i>	
	Motivational Example	384
	Beginning Development on the New Object	384
	First Test Case	386
	Moving Along	387
	Back to the Scalar Multiplication Test Case and Others	391
	Rest of the List	393
	More Changes	395
	Addition Expression	396
	Generalizing the Expression Object	399
	Multiplication of Arrays	402
	Afterword	405
	References and Further Reading	407
Chapter 25	Improving Performance in C++ with Compile-Time Polymorphism	409
	<i>Ben Sunshine-Hill</i>	
	Thanks but No Thanks, C++	412
	Making Life More Difficult	413
	You Can Rebuild It—You Have the Technology	417
	Putting It All Together	420
	Limitations	424
	Summary	425
Chapter 26	Implementing a Software Cache	427
	<i>Eli Bendersky</i>	
	Software vs. Hardware Cache	427
	Basic Requirements and Definitions	428
	To Infinity and Beyond?	428

Cache Removal	429
Requirements Revisited	429
Design	430
Implementation	431
Using the Cache	433
Efficiency Revisited	434
Index	435