

Contents

Foreword	xiii
Acknowledgments	xvii
Introduction.	xix
 Chapter 1 A Math and Physics Primer	 1
Mathematics	1
Cartesian Coordinates	1
Functions and Equations	3
Exponents and Powers	5
Roots of Numbers (Radicals)	6
Simplifying Equations.	7
Trigonometry	10
Rays and Line Segments	10
Angles	11
Triangles	12
Vectors	18
Adding and Subtracting Vectors	19
Multiplying Vectors	20
Calculating the Magnitude of a Vector.	20
Normalizing Vectors	21
Resolving Vectors	22
The Dot Product	23
A Practical Example of Vector Mathematics.	24
The Vector2D Struct	25
Local Space and World Space.	26
Physics	28
Time.	28
Distance	29
Mass.	29
Position	30
Velocity	30
Acceleration.	32
Force	38
Summing Up	40
 Chapter 2 State-Driven Agent Design	 43
What Exactly Is a Finite State Machine?.	44
Implementing a Finite State Machine	45
State Transition Tables	47
Embedded Rules	48
The West World Project	50

The BaseGameEntity Class	52
The Miner Class	53
The Miner States	54
The State Design Pattern Revisited	55
The EnterMineAndDigForNugget State	60
Making the State Base Class Reusable	62
Global States and State Blips	63
Creating a State Machine Class	64
Introducing Elsa	67
Adding Messaging Capabilities to Your FSM	69
The Telegram Structure	70
Miner Bob and Elsa Communicate	71
Message Dispatch and Management	71
The MessageDispatcher Class	73
Message Handling	75
Elsa Cooks Dinner	78
Step One	78
Step Two	79
Step Three	80
Step Four	80
Step Five	81
Summing Up	82

Chapter 3 How to Create Autonomously Moving Game Agents	85
What Is an Autonomous Agent?	85
The Vehicle Model	87
Updating the Vehicle Physics	89
The Steering Behaviors	91
Seek	91
Flee	92
Arrive	93
Pursuit	94
Evade	96
Wander	96
Obstacle Avoidance	99
Finding the Closest Intersection Point	100
Calculating the Steering Force	103
Wall Avoidance	104
Interpose	106
Hide	107
Path Following	110
Offset Pursuit	111
Group Behaviors	113
Separation	115
Alignment	116
Cohesion	117
Flocking	118
Combining Steering Behaviors	119
Weighted Truncated Sum	120
Weighted Truncated Running Sum with Prioritization	121

Prioritized Dithering	123
Ensuring Zero Overlap	124
Coping with Lots of Vehicles: Spatial Partitioning	126
Smoothing	130

Chapter 4 Sports Simulation — Simple Soccer	133
The Simple Soccer Environment and Rules	134
The Soccer Pitch	135
The Goals	138
The Soccer Ball	138
SoccerBall::FuturePosition	141
SoccerBall::TimeToCoverDistance	142
Designing the AI	144
The SoccerTeam Class	145
The Receiving Player	146
The Closest Player to the Ball	146
The Controlling Player	146
The Supporting Player	146
SoccerTeam States	152
Field Players	155
Field Player Motion	155
Field Player States	156
Goalkeepers	170
Goalkeeper Motion	170
Goalkeeper States	171
Key Methods Used by the AI	176
SoccerTeam::isPassSafeFromAllOpponents	177
SoccerTeam::CanShoot	182
SoccerTeam::FindPass	184
SoccerTeam::GetBestPassToReceiver	185
Making Estimates and Assumptions Work for You	189
Summing Up	189

Chapter 5 The Secret Life of Graphs	193
Graphs	193
A More Formal Description	195
Trees	196
Graph Density	196
Digraphs	196
Graphs in Game AI	197
Navigation Graphs	198
Dependency Graphs	199
State Graphs	201
Implementing a Graph Class	203
The GraphNode Class	204
The GraphEdge Class	205
The SparseGraph Class	207
Graph Search Algorithms	209
Uninformed Graph Searches	210
Depth First Search	210

Breadth First Search	224
Cost-Based Graph Searches	231
Edge Relaxation	231
Shortest Path Trees	233
Dijkstra's Algorithm	233
Dijkstra with a Twist: A*	241
Summing Up	247
Chapter 6 To Script, or Not to Script, That Is the Question	249
Just What Is a Scripting Language?	249
What a Scripting Language Can Do for You	251
Dialogue Flow	253
Stage Direction	254
AI Logic	255
Scripting in Lua	255
Setting Up Your Compiler to Work with Lua	256
Getting Started	256
Lua Variables	258
Lua Types	260
Logical Operators	263
Conditional Structures	264
Rock-Paper-Scissors in Lua	265
Interfacing with C/C++	268
Accessing Lua Global Variables from within Your C++ Program	269
Accessing a Lua Table from within Your C++ Program	271
Accessing a Lua Function from within C++	273
Exposing a C/C++ Function to Lua	274
Exposing a C/C++ Class to Lua	276
Luabind to the Rescue!	276
Setting Up Luabind	276
Scopes	277
Exposing C/C++ Functions Using Luabind	278
Exposing C/C++ Classes Using Luabind	279
Creating Classes in Lua Using Luabind	281
luabind::object	282
Creating a Scripted Finite State Machine	285
How It Works	285
The States	289
GoHome	290
Sleep	290
GoToMine	291
Useful URLs	292
It Doesn't All Smell of Roses	292
Summing Up	293
Chapter 7 Raven: An Overview	295
The Game	295
Overview of the Game Architecture	296
The Raven_Game Class	297
The Raven Map	299

Raven Weapons	301
Projectiles	302
Triggers	303
TriggerRegion	304
Trigger	305
Respawning Triggers	307
Giver-Triggers	308
Limited Lifetime Triggers	309
Sound Notification Triggers	310
Managing Triggers: The TriggerSystem Class	311
AI Design Considerations	313
AI Implementation	315
Decision Making	315
Movement	315
Path Planning	315
Perception	316
Target Selection	321
Weapon Handling	323
Putting It All Together	327
Updating the AI Components	328
Summing Up	331
Chapter 8 Practical Path Planning	333
Navigation Graph Construction	333
Tile Based	333
Points of Visibility	334
Expanded Geometry	335
NavMesh	335
The Raven Navigation Graph	336
Coarsely Granulated Graphs	336
Finely Grained Graphs	339
Adding Items to the Raven Navigation Graph	341
Using Spatial Partitioning to Speed Up Proximity Queries	342
Creating a Path Planner Class	342
Planning a Path to a Position	344
Planning a Path to an Item Type	346
Paths as Nodes or Paths as Edges?	348
An Annotated Edge Class Example	350
Modifying the Path Planner Class to Accommodate Annotated Edges	350
Path Smoothing	353
Path Smoothing Rough but Quick	354
Path Smoothing Precise but Slow	358
Methods for Reducing CPU Overhead	359
Precalculated Paths	359
Precalculated Costs	361
Time-Sliced Path Planning	363
Hierarchical Pathfinding	372
Getting Out of Sticky Situations	374
Summing Up	376

Chapter 9 Goal-Driven Agent Behavior	379
The Return of Eric the Brave	380
Implementation	382
Goal_Composite::ProcessSubgoals	385
Goal_Composite::RemoveAllSubgoals	386
Examples of Goals Used by Raven Bots	387
Goal_Wander	387
Goal_TraverseEdge	388
Goal_FollowPath	391
Goal_MoveToPosition	393
Goal_AttackTarget	395
Goal Arbitration	398
Calculating the Desirability of Locating a Health Item	400
Calculating the Desirability of Locating a Specific Weapon	401
Calculating the Desirability of Attacking the Target	403
Calculating the Desirability of Exploring the Map	403
Putting It All Together	404
Spin-offs	405
Personalities	405
State Memory	406
Example One — Automatic Resuming of Interrupted Activities	407
Example Two — Negotiating Special Path Obstacles	408
Command Queuing	410
Using the Queue to Script Behavior	412
Summing Up	414
Chapter 10 Fuzzy Logic	415
Crisp Sets	417
Set Operators	418
Fuzzy Sets	419
Defining Fuzzy Boundaries with Membership Functions	419
Fuzzy Set Operators	421
Hedges	423
Fuzzy Linguistic Variables	423
Fuzzy Rules	424
Designing FLVs for Weapon Selection	425
Designing the Desirability FLV	426
Designing the Distance to Target FLV	427
Designing the Ammo Status FLV	428
Designing the Rule Set for Weapon Selection	428
Fuzzy Inference	429
Rule One	429
Rule Two	430
Rule Three	430
Defuzzification	433
From Theory to Application: Coding a Fuzzy Logic Module	437
The FuzzyModule Class	437
The FuzzySet Base Class	439
The Triangular Fuzzy Set Class	440
The Right Shoulder Fuzzy Set Class	441

Creating a Fuzzy Linguistic Variable Class	443
Designing Classes for Building Fuzzy Rules	445
How Raven Uses the Fuzzy Logic Classes	451
The Combs Method	452
Fuzzy Inference and the Combs Method	454
Implementation	455
Summing Up	455

Last Words	457
------------	-----

Appendix A C++ Templates	459
Appendix B UML Class Diagrams	465
Appendix C Setting Up Your Development Environment	475

References	477
Bugs and Errata	479
Index	481