### Introduction ......................................................................................... ix

About the Game Development Essentials Series ................................................ x

About Game Development Essentials: Game Artificial Intelligence ....................... x

Who Should Read This Book? ..................................................................... xi

How Is This Book Organized? ...................................................................... xii

How To Use This Text ............................................................................... xiv

Key Chapter Questions ............................................................................ xiv

Notes ........................................................................................................ xiv

Sidebars .................................................................................................... xiv

Quotes ...................................................................................................... xv

Tips ............................................................................................................ xv

Case Studies .............................................................................................. xv

Profiles ..................................................................................................... xvi

Chapter Review........................................................................................ xvi

About the Companion DVD ....................................................................... xvi

About the Instructor’s Guide ....................................................................... xvii

About the Author ...................................................................................... xvii

Acknowledgments ...................................................................................... xviii

Questions and Feedback ........................................................................... xx

Dedication .................................................................................................. xxi

### Part I: Creating Behaviors ................................................................. 1

#### Chapter 1 History and Concepts: how did we get here (and where is here, anyway?) .......... 3

Key Chapter Questions ............................................................................ 3

Game AI vs. Computer Science AI ................................................................. 4

A Brief Overview of Computer Science AI ..................................................... 4

Mechanical Automata .................................................................................. 5

Programmable Systems ............................................................................... 6

AI Development .......................................................................................... 7

Fuzzy Logic ................................................................................................ 8

Knowledge-Based and Expert Systems ......................................................... 9
Chapter 2 AI Agents: Creating behavior using a finite state machine

Key Chapter Questions

Creating and Controlling AI Behaviors
Behavior: A Sequence of Actions
Dynamic Sequences
Finite State Machine
Representative AI States
SAGE State Machine
MoveTo: A Single-State Behavior
Attack: A Multiple-State Behavior
Matching States to Game Design
Designing an FSM

Transitions
Internal
External
Deferred
Immediate

An AI System Controlled by a Finite State Machine
Sample Application
Agent Design
AI Module
Speculative Features
Attraction
Hidden Cost

Chapter Review
Chapter 5  Expert Systems: capturing high-level knowledge and improving behavioral control

Key Chapter Questions

Expert Systems Defined

Expert System Components

Knowledge Base

Working Memory

Inference Engine

Capturing High-Level Knowledge

Analysis

Empirical Study

The Power of Expert Systems

Challenges

Embedded Expert Systems

Sample Rule Set

Modifying the Game State

Combining Data-Driven and Rule-Based Systems

Structure Location

Order of Creation

Additional Base-Building Rules

Chapter Review

Chapter 6  Pathfinding: allowing agents to properly find and follow paths

Key Chapter Questions

Pathfinding Systems
Index

A
A* search algorithm, 170, 181–184, 193–195, 204
abstractions, 83
actions
defined, 83, 87
dynamic sequences, 41
of goals, 87–88
multiple, 91–93
selection of, 90–93
sequence of, 40–41
sequential, 91–93
simultaneous, 91–93
active state generated transitions, 72–75, 82
advantages and disadvantages, 73–74
in Unreal 3 engine, 72–73
agents
avoiding mechanical behavior in, 228–229
building, from modules, 56–62
characters, 11, 12, 14–16
choice-making by, 230–235
control of, using FSMs, 56–63
creating, with aggregation, 32–33
design of, 58–60
early, 13–17
group pathfinding by, 206–209
groups of smart, 247
history of game, 11–17
impression of intelligence in, 217–241
intelligent, 24–25
path following by, 184–188
path planning for, 166–167
retreating, 209
sample application, 57–58
simple, 246–247
solo, 248
in stuck position, 196
in unexpected locations, 196
using multiple types, 13
virtual players, 11, 12, 16–17, 249–250
AgentStateMachine class, 61–62
Age of Empires, 184, 185
aggregation, 31–34, 56
AI. See artificial intelligence (AI)
AI agents. See agents
AI behaviors
abstractions of, 83
AgentStateMachine class, 61–62
avoiding mechanical, 228–229
controlling
by considering multiple states, 76–82
creating and, 40–55
in data-driven design, 110–111
with embedded expert systems, 152–154
by FSMs, 56
in rule-based systems, 111
with scripting systems, 126
dividing into states, 50
dynamic sequences, 41
finite state machines for, 41–44, 49–50
goal-oriented model, 87–95
handling complex, 94
high-level, 265
implying, 241
low-level, 264
matching, to game design, 49
multiple-state, 46–49
options for, 230–235
parent state control, 80–81
random, 231
repetitive, 105, 228–229
retaliation behavior, 80
as sequence of actions, 80
single-state, 45–46
states of. See states
suspending, 85–86
AI core, 251
AI designers. See designers
AI module, 61–62
Aim state, 46–49, 55
AI programmers. See programmers
AI systems
See also expert systems
Black Box, 100–103, 105–107, 109, 256–257, 260
defining feature set, 254–255
design of, 255–263, 271
implementation of, 263–266
rule-based, 111
Titanium (case study), 250–270
White Box, 100–103, 106–107, 109, 256, 259, 260
algorithms, 26
See also search algorithms
Allen, 10
Alone in the Dark 360, 258
alternate states, 85–86
analog programming system, 5
analysis, 140
analytical engine, 6
- **Index**

**A**
- A* search algorithm, 170, 181-184, 193-195, 204
- abstractions, 83
- **actions**
  - defined, 83, 87
  - dynamic sequences, 41
  - of goals, 87-88
  - multiple, 91-93
  - selection of, 90-93
  - sequence of, 40-41
  - simultaneous, 91-93
- active state generated transitions, 72-75, 82
- advantages and disadvantages, 73-74
- in Unreal 3 engine, 72-73
- **agents**
  - avoiding mechanical behavior, in, 228-229
  - building, from modules, 56-62
  - characters, 11, 12, 14-16
  - choice-making by, 230-235
  - control of, using FSMs, 56-63
  - creating, with aggregation, 32-33
  - design of, 58-60
  - early, 13-17
  - group pathfinding by, 206-209
  - groups of smart, 247
  - history of game, 11-17
  - impact of intelligence in, 217-241
  - intelligent, 24-25
  - path following by, 184-188
  - path planning for, 166-167
  - retreating, 209
- sample application, 57-58
  - simple, 246-247
  - solo, 248
  - in stuck position, 196
  - in unexpected locations, 196
  - using multiple types, 13
  - virtual players, 11, 12, 16-17, 249-250
- AgentStateMachine class, 61-62
  - Age of Empires, 184, 185
  - aggregation, 31-34, 56
  - AI. See artificial intelligence (AI)
  - AI agents. See agents
  - AI behaviors
    - abstractions of, 83
    - AgentStateMachine class, 61-62
    - avoiding mechanical, 228-229
    - controlling
      - by considering multiple states, 76-82
      - creating and, 40-55
      - in data-driven design, 110-111
    - with embedded expert systems, 152-154
    - by FSMs, 56
    - in rule-based systems, 111
    - with scripting systems, 126
    - dividing into states, 50
    - dynamic sequences, 41
    - finite state machines for, 41-44, 49-50
    - goal-oriented model, 87-95
    - handling complexity, 94
    - high-level, 265
    - implying, 241
    - low-level, 264
    - matching, to game design, 49
  - multiple-state, 46-49
    - options for, 230-235
    - parent state control, 80-81
    - random, 231
    - repetitive, 105, 228-229
    - retaliation behavior, 80
    - as sequence of actions, 40-41
    - single-state, 45-46
    - states of. See states suspending, 85-86
  - AI core, 251
  - AI designers. See designers
  - AI module, 61-62
  - Aim state, 46-49, 55
  - AI programmers. See programmers
  - AI systems
    - See also expert systems
    - Black Box, 100-103, 105-107, 109, 256-257, 260
    - data-driven, 27-29, 58-60, 109-111, 256
    - defining feature set, 254-255
    - design of, 255-263, 271
    - implementation of, 263-266
    - rule-based, 111
    - Titanium (case study), 250-270
    - White Box, 100-103, 106-107, 109, 256, 259, 260
    - algorithms, 26
      - See also search algorithms
    - Allen, 10
    - Above in the Dark, 360, 258
    - alternate states, 85-86
    - attack behavior, 46-49, 249-250
    - Attack state, 71
    - audio cues, 223-224, 229, 236-239, 241
    - automatic fun, 129
    - automatic path nodes, 256-257, 258
    - automaton, 5
- Anantharaman, Thomas, 11
- AND operator, 9
- Babbage, Charles, 6
- Baidu, 9
- **B**
  - **Babbel, Charles, 6**
  - backward chaining, 25-26, 273
  - backward chaining inference, 148-150
  - base-building rules, 156-157
  - behaviors. See AI behaviors
    - Benjamin, Joel, 11
    - best first search, 181
    - binary number system, 5
    - binary search algorithm, 26
    - bitmask, for states, 79
    - Black Box systems, 100-103, 105-107, 109, 256-257, 260
    - blended state machines, 81
    - Body module, 60
    - Boolean logic, 5, 8
    - breadth first search, 171-175
    - Breakout, 13, 14
    - Brody, Jerry, 11
    - Brooks, Rodney, 10
    - Browder, Dustin, 278
    - Brown, Mike, 11
    - bubble sort algorithm, 26
  - C++ functions, calling from scripts, 124
  - C++ language, 19-21, 26, 29, 59, 261
  - Call of Duty 2, 16
  - Campbell, Murray, 11
  - careers, 270-279
  - Cart, 10
  - Centipede, 14, 15
  - character agents, 11, 12, 14-16
  - cheating, intentional vs. accidental, 157-158
  - chess playing computers, 11, 12
  - child states, competition among, 77-79
  - choice making, 230-235
  - Civilization, 13, 16, 17
  - Civilization IV, 45, 234, 246
  - Civilizations, 13, 16, 17
  - comes, John, 118, 134, 251
  - Command & Conquer agents in, 17, 24, 32, 63
  - agent spacing in, 186
  - AI behaviors in, 40
  - AI system in, 102, 107
  - data capture method, 262
  - data-driven base layout in, 157
  - game engine, 35
  - game engine in, 43-44, 60
  - game performance, 226
  - level of detail in, 227
  - pathfinding in, 210, 211
  - path nodes, 258
  - scripting system in, 22
  - state transitions, 48
  - complexity, handling, 94
  - computational expense, from competing child states, 78
  - computers, first programmable, 6-7
  - computer science AI vs. game AI, 4
  - overview of, 4-11
  - conditions, 137
- Brolin, Jerry, 11
- Brown, Mike, 11
Cones, John, 42–43
conflict resolution, 144, 146
control methods, 34–35, 71–82
Counter-Strike, 86
cues
for intelligence, 235–239, 241
scripting, 236
custom scripting languages, 113–114, 120–121

D
Daglow, Don, 42, 112, 165, 222, 230, 238, 251
daily builds, 264
data capture methods, 260–263
data definition, 272
data-driven design, 27–29, 58–60, 109–111
data-driven inference, 146–147
data-driven systems, 156–157, 256
data formats, 29
data values, 125
debugging
assembly language and, 20
custom scripting languages, 114
for FSMs, 50, 63
by programmers, 272–273
Deep Blue, 11
Deep Thought, 11
deferred transitions, 51–55
Def Jam Vendetta, 221
dendral, 10
depth first search, 171
design. See game design;
game development
designers
education needed by, 277–278
experience needed by, 278–279
relationship between programmers and, 108
    role of, 99, 104–105, 111, 270, 275–277
    scripting systems and, 118
derigner-scripted choice, 233–234
development iteration, 267–269
difference engine, 6
Difference Engine No. 2, 6
difficulty, game, 155
Dijkstra’s algorithm, 169,
175–181, 193, 194–195
DirectX, 275
distance based control, 226–227
dithering, state, 78
dodge state, 78
Doom, 15, 20
Doom 3, 247
downloadable game engines, 275
ELIZA, 7, 8
ELIZA effect, 8, 217, 219–221,
225, 235, 236
embedded expert systems, 151–155
emotion system, 236–237
Emperor, 26, 157, 207
Empire, 16
empirical study, 140
ENIAC, 5, 17
EnterState() method, 71
environment, reacting to, by character agents, 15–16
error reduction, from GUIs, 23
event-based control methods, 35
events, 137
ExitState() method, 71
expert systems, 4, 10–11
backward chaining inference, 148–150
challenges of, 142–151
combining with data-driven systems, 156–157
components of, 134–139
conflicting information and, 142–143
conflict resolution in, 144–146
defined, 134
embedded, 151–155
explanations given by, 141–142
forward chaining inference, 146–147
high-level knowledge, 139–140
inconsistent facts in, 138
inconsistent results from, 143–144
inference engines, 134,
138–139
introduction to, 133
knowledge base in, 134,
135–136
power of, 141–151
questions by, 148–150
rule sets, 134–136, 143–146
stopping criteria, 145
working memory, 134,
137–138
explanations, 141–142
external transitions, 51–55, 71–72

F
facts
conflicting, 144–146
inconsistent, 138
negation vs. absence of, 151
Far Cry, 226
F.E.A.R., 223, 233, 247, 249–250
features, speculative, 63–66
Fifth Generation project, 11
finite state machines (FSMs), 41–44
AI systems controlled by, 56–63
debugging, 63
designing, 49–50
game development team
changing roles in, 104–105
staffing considerations, 107
workload sharing by, 111–112
game difficulty, 155
game engines
See also specific engines
free downloadable, 275
indie, 275
Galt Wars, 203
Giantman Chronicles, 274

H
hacks, 71
Half-Life, 15
Half-Life 2, 12, 169, 232, 233
Halo, 15, 253
Halo 2, 81, 82, 169
hand assembly, 17–18
Heron of Alexandria, 5
heuristics, 26–27
hierarchical finite state machines
(HFSMs), 69–71, 84
vs. computer science AI, 4
game theory, 277
Gears of War, 22, 47, 48, 85, 256,
258, 263
General Problem Solver, 8
Gilson, Frank, 111–112, 237,
250, 279
glue functions, 21, 124
history of implementation
methods, 17–24
goal-oriented AI model, 87–95
goals, 83
actions to achieve, 90–93
backward chaining and, 150
defined, 87
handling multiple, 84
hierarchy of, 88–89
operation of, 87–88
selection of, 89–90
state search, 84
tools to achieve, 93–94
golems, 4
graphical user interfaces (GUI), 22–24
grids
arbitrary, 164–165
regular, 162–164
square, 180
group pathfinding, 206–209

I
IBM 701, 7
Idle state, 44, 45–46
IF THEN statements, 135
immediate transitions, 51, 54, 55
implementation methods, history
of, 17–24
implementation time, GUIs
and, 23–24
SAGE2 engine, 82
SAGE engine, 44-49, 56, 57, 60, 86
custom, 113-114
SAGE RTS engine, 35, 43
Samuel, Arthur, 7
schemas, 29
Schickard, William, 5
Schickard Calculator, 5
scripting functions, 124
scripting languages, 18, 19, 21-22. See also specific languages
advantages of, 112
benefits of, 128-129
vs. C++, 123-124
custom, 113-114
GUI, 118-123
interfacing, with game, 116-117
licensed, 115-116
open source, 114-115
programmers and, 272
text, 113-118
scripting systems, 112-130
controlling animations with, 126
for data values, 125
division of labor in, 118
events and functions, 126
interface design, 127
object-oriented, 127-128
real-world examples, 262
using, in game, 124-126
scripts, grouping, 121-122
search algorithms, 170-183
A* search, 170, 181-184, 193-195, 204
best first search, 181
 breadth first search, 171-175
Dijkstra's algorithm, 175-181, 193, 194-195
self documentation, 22-23
sequence of actions, 40-41
sequential actions, 91-93
Sims, The, 23, 24
Shaw, J.C., 8
Simon, Herbert, 8
speculative features, 63-66
spin, 54
Splinter Cell, 54
square grid networks, 180
StarCraft, 23, 24, 27, 101, 151, 262
startegic choices, 231-235
State class, 44
StateMachine class, 44, 45, 63
state machines
active state generated transitions, 72-75
advantages and disadvantages, 84-87
blended, 81
considering multiple states, 76-82
handling complex behaviors with, 94
models, 83-84
multiple, 86-87
procedural transition generation, 71-73
SAGE, 44-49
state masks, 110-111
states
alternate, 85-86
competition among child, 77-79
dealing with large numbers of, 70-71
dividing behaviors into, 50
handing complexity of, 94
hierarchies, 79
masking, 79
matching, to game design, 49
multiple-state behavior, 46-49
parent state control, 80-81
representative examples, 43-44
single-state behaviors, 45-46
spin, 54
transitions between. See transitions
state search goals, 84
state stack, 85
state suspend and continue, 85-86
state transitions. See transitions
state twitching, 78
stopping criteria, 145
strategic AI, vs. tactical AI, 42-43
strategies
choosing, 233-235
implementing, 232-233
multiple, 231-232
sub-goals, 90-91
Superville, Stephen, 240
Supreme Commander, 22, 109, 156
table lookup, 168–170
tactical AI, vs. strategic AI, 42–43
Tan, C. J., 11
technical designers, 118
temporary states, 79
tetris, 13
text-based languages. See scripting languages
3D games, early, 15
tick function, 35
tick() method, 61
time-based control methods, 34
Titanium (case study), 250–270
ai system design, 255–263
ai system implementation, 263–266
AI system setup, 254–255
development iteration, 267–268
game development environment, 252
initial game design, 252–254
production iteration, 269–270
Tony La Russa Baseball, 222
tools, 87, 89, 93–94
Torque Script, 113–117
TorqueScript, 29
tower of Hanoi, 8
transitions between. See transitions
state search goals, 84
state stack, 85
state suspend and continue, 85–86
state transitions. See transitions
state twitching, 78
stopping criteria, 145
strategic AI, vs. tactical AI, 42–43
strategies
choosing, 233–235
implementing, 232–233
multiple, 231–232
sub-goals, 90–91
Superville, Stephen, 240
Supreme Commander, 22,
109, 156
video game consoles, 7
early, 18
next-generation, 265, 279
video game technology, current state of, 24
virtual players, 11, 12, 16–17, 249–250
See also agents
smart, 246–247
waypoint paths and, 203–206
Warcraft III, 137, 164
waypoint paths, 203–206
weizenbaum, Joseph, 8, 219
White Box systems, 100–103,
106–107, 109, 256, 259, 260
working memory, 134, 137–138
World of Warcraft, 115
X
XCON/Rl expert system, 147
XML, 29
Z
Zadeh, lofti, 8
zone equivalency array, 198–201
zone equivalency tables, 202–203
zone information, 201–202
zone mapping, 197–203