Index

Note: Page numbers ending in "e" refer to equations. Page numbers ending in "f" refer to figures. Page numbers ending in "t" refer to tables.

A

academic AI, 3–5, 8, 66. *See also* Game AI
action potential, 15–21, 16f, 17f, 18f
agent threat response, 391–398
AI Game Programming Wisdom, 437
AI Techniques for Game Programming, 391
alibi generation
  background of, 460
  choosing alibis, 465–466
  computing, 464e
  deleting characters, 466–467
  distribution, 463, 465
  fooling players, 459–467
  generating, 464–467
  hybrid generation, 466
  ideal world, 460–462, 462f, 465
  initial generation, 462–463
  introduction to, 459–460
  maintaining characters, 466–467
  necessity of, 463–464
  options for, 465–466
  perturbation, 466
  population size, 462–463
  position selection, 463
Anguelov, Bobby, 297
animal behavior
  control systems for, 532–537
  control theory, 532–533
  control-based architecture for, 531–538
  game extensions, 536–537
  gosling control system, 537
  hierarchical control system, 535–536, 535f
  introduction to, 531–532
  negative feedback controller, 533–537, 533f, 534f
  perceptual control system, 533–535
  purposeful behavior, 531–532
animation-driven locomotion. *See also* locomotion
  action-stack, 328–330, 329f
  commercial implementation, 334
  executing actions, 331–333
  implementing, 331–332, 334
  introduction to, 325
  inverse kinematic (IK) controllers, 333
  locomotion planning, 325–334
  movement architectures, 326
  navigation paths, 327–328, 328f
  nontransfer actions, 331–332
  performance of, 333–334
  preparation for, 326–327
  transfer actions, 325, 330–333
architectures
  for animal behavior, 531–538
  for background characters, 453–455
  for blackboards, 61, 66–67, 214–215
  for character-rich social simulation, 515–530
  CiF architecture, 516–529
  common architectures, 62–63
  for Game AI, 5–7, 6f, 45–60

for hierarchical AI, 378–379, 379f
hybrid architecture, 146
integrated architectures, 231–232
movement architectures, 326
pathfinding optimizations, 241–252
for racing games, 471–479
structural architecture, 61–71
for tactical positions, 338–339, 339f
three-layer architecture, 146f
Argyle, Michael, 448
artificial intelligence (AI), 3–4. *See also* Game AI
Artificial Intelligence for Games, 52, 391
artificial sound designer (ASD)
database for, 551–553
for dynamic sound, 549–555
explanation of, 550–551
generating events, 551
play-request events, 554
rule set for, 553
updating, 554–555
attribute checking, 226–228
authorial control, 5, 6f
autonomy
approximations of, 130
behavior trees and, 130, 136
in CiF, 515–516, 525–528
in collision system, 565–566
in Game AI, 6f, 99–102, 138
in OpenCL, 541
in robots, 383
in squad AI, 383, 515
avoidance point (AP), 302–304

B

background characters. *See also* characters
actions for, 454–455
architecture for, 453–455
breathing life into, 451–458
game objects, 454–455
improvements on, 457–458
introduction to, 451
schedule entries for, 453–454
schedule system for, 452–453
schedule templates for, 456
techniques for, 452
Bednarz, Tomasz, 539
behavior, representing, 12
behavior asset management, 96
behavior class improvements, 97
behavior observers, 88
behavior scheduler, 88–90
behavior selection algorithms
behavior trees, 52–53
chaining search, 56
finite-state machines (FSMs), 48–52
goal-oriented action planning (GOAP), 7, 55–59
hierarchical finite-state machines (HFSMs), 50–52
introduction to, 47
solutions for, 57–60, 149–150, 163, 167
for tactical position selection, 342–343
utility systems for, 53–55
behavior trees
actions for, 52–53, 75–76, 99–111
advanced behavior trees, 84–90
autonomy and, 130, 136
behavior concepts, 75–76
behavior selection algorithms, 52–53
building blocks of, 75–84
C++ and, 214–215
completion status for, 76
composites of, 78–79
conditions for, 75–78
constraints for, 100, 103–106
for decision-making, 130–132, 137
decorators, 78
evaluation and, 134–136
event-driven behavior, 88–89
examples of, 74–76, 100f, 129–130, 129f, 130f
execution and, 134–136
execution hints, 76
explanation of, 52–53
extensibility, 53
filters, 80
finite-state machines (FSMs), 63
first-generation trees, 84–85
flexibility of, 73, 98
hybrid approach to, 99–111
initialization of actions, 77
introduction to, 73, 99–100, 127–128
leaf behaviors, 77, 90, 94, 134
leaf nodes, 75–77, 102, 134–135, 222
memory access performance, 86–87
modular behaviors, 91
monitoring, 77–78, 83
node implementations, 85–87
overview of, 74–75
performance of, 6
planner hybrid, 102–103

planners, 101–103, 101f, 111
popularity of, 128
preconditions, 52–53, 80
propagating utility, 134
real-world behavior trees, 93–98
return statuses, 76–78
Runtime Compiled C++ and, 214–215
second-generation trees, 84–85
selectors, 63, 80–81, 83–84, 102–103, 129–133, 132f
sequences, 79–80
sharing between entities, 85–86
shutdown of actions, 77
simplicity of, 53
simulating, 99–111
starter kit for, 73–91
strength of, 53
transient data allocation, 88
updating, 74–75
utility decisions for, 127–136, 131e
utility selector, 132–133, 132f
utility theory, 130
utility value, 131–135, 135f
variations of, 84–91
Behavioral Mathematics for Game AI, 55
behaviors, debugging, 97–98
Beij, Arjen, 389
Belgian AI, 370, 373–374
bell curves, 30. *See also* Gaussian randomness
Binks, Doug, 201
Bjore, Stephen, 289
blackboard architecture, 61, 66–67, 214–215
Bodily Communication, 448
Boolean flag, 226, 247, 345, 517, 529
Boolean questions, 53–55, 54e
Bourke, Conan, 539
break in realism (BIR), 187–188, 191–193
Bresenham line algorithm, 311
Brewer, Daniel, 361
Brink, 343, 344
"bucketing," 121–122, 121f
Bulletstorm, 84, 325, 334, 342, 349, 357, 423–424, 430–431

C

C++ for rapid AI, 201–217
camera system
activation rules, 561
behavior of, 561–563
bones for, 564
camera size, 565
collision flags, 565
collision reaction, 565–566
collision system, 565–566
competing cameras, 559
configuration of, 560–561
data driven approach, 561
debugging, 558–559
distance from cameras, 562, 563f
explanation of, 557–558
features of, 563–565
first-person camera, 562
fixed cameras, 562
input transform, 560
introduction to, 557
managing multiple cameras, 559–560
object references, 561
occlusion reaction, 566
orbit cameras, 562–564, 564f
over-the-shoulder cameras, 562
priorities, 561
on rails, 563
spline for, 563–564, 564f
sweet spot for, 564–565
third-person camera, 562
tips for, 557–566
tracking cameras, 562
transitions, 559–560
view manager, 559
Carlisle, Phil, 433, 441
central limit theorem, 30–31, 31f
Champandard, Alex J., 73, 377
character blackboards, 66–67
character criticality, 188–189
character-rich social simulation. *See also* characters
character representation, 520–521, 520t
CiF architecture, 516–529
explorer interface, 518, 519f
interaction approaches, 527–528
introduction to, 515–516
related work, 528–529
social exchanges, 517, 518f, 523–527, 523t
social facts knowledge base (SFKB), 522–523
social state, 521–523
story goals, 517, 519f
characters
background characters, 451–458
breathing life into, 451–458
components of, 441–449, 448f

social dynamics system, 441–450
social simulation of, 515–530
Ciupinski, Jaroslaw, 325
collision avoidance
agent collision radius (ACR), 300
animation-driven locomotion, 297–299
introduction to, 297
for locomotion, 299–300
path modification system, 302–304, 303f
reciprocal velocity obstacle technique, 297–299, 304–305
time per movement, 300
velocity vector, 300
collision detection, 299–302, 300f
collision resolution, 299–304, 304f
collision sphere motion, 301
collision system, 565–566
combat maneuvers
examples of, 173t, 174f, 175f, 176f
introduction to, 169
multi-unit planning, 170–171, 170f, 172f, 181f
planning for, 169–183
plan-space planning, 172–182, 181f
Comme il Fault (CiF) architecture, 515–529
Company of Heroes, 246
Computational Brain, The, 12
control system
adaptive control, 497–498
for animal behavior, 532–537
closed-loop control, 492–493, 492f
control theory, 492–494
explanation of, 532–533
hierarchy of, 535–536, 535f
introduction to, 491–492
negative feedback controller, 533–537, 533f, 534f
open-loop control, 492–493, 492f
other racing applications, 498–500
perceptual control system, 533–535
PID controllers, 494–500
predictive control, 498
for racing games, 491–500
cost data, 314
cost fields, 308–309
cost integration pass, 311–312
cost stamps, 314–315
Côté, Carle, 137
Crackdown 2, 406
Creatures, 5
critical thinking skills, 11–12, 26
criticality, measuring, 189
crowd pathfinding
algorithm for, 311
Bresenham line algorithm, 311
cost data, 314
cost fields, 308–309
cost integration pass, 311–312
cost stamps, 314–315
CPU usage and, 315–316
dynamic environments, 313–314
dynamic flow field tiles, 307, 323
field types, 307–309, 323
flow field cache, 313
flow field pass, 312–313, 313f
flow fields, 307–318, 313f, 319f, 320–323
integration fields, 308–310
integration steps, 310–311
integrator, 310
introduction to, 307
island fields, 315
line of sight (LOS) pass, 310–311, 312f
motivation for, 307–308
movement types, 314–315
path costs, 308–309, 311–312
path requests, 310
world layout, 308–309, 309f
crowd simulation
algorithm for, 318–319, 322–323
flow fields, 318–319, 319f, 322–323
flow function, 318–319
future of, 322–323
grid for, 317–318
introduction to, 317
steering behaviors, 315–316, 319–321
units for, 319–321, 320f
CryENGINE, 203, 337, 406, 411
CryENGINE 3, 203, 404, 411
CryEngine Free SDK, 341, 411
Crysis 1, 348
Crysis 2, 128, 214, 215, 337, 343–344, 348, 352, 356, 358, 404

D

Da Vinci, Leonardo, 403
"danger awareness," 144–145
Dawe, Michael, 47, 93, 369
decision-making systems
action selector, 143f, 144f, 145f
behavior trees for, 130–132, 137

behavioral artifacts, 141f, 142f, 144f
conceptual model for, 144–147, 145f
deliberation and, 137–147
finite-state machines (FSMs), 137–143, 140f, 143f, 146
goal-oriented action planning (GOAP), 137
graphical modeling language-based models, 137
hierarchical finite-state machines (HFSMs), 137
introduction to, 137
multiple decision models, 142
pitfalls of, 139–147
reactivity and, 137–147, 138f, 139f
selector switch, 143f
symbolic planning language-based decision models, 137
three-layer architecture, 146f
timeline of, 138f, 139f, 141f, 142f, 144f, 145f
utility theory for, 130–132
deduplication technique, 226–227, 234
Delaunay Triangulation Decompositions, 256, 260, 265
deliberation
in decision-making systems, 137–147
examples of, 145f, 146f
explanation of, 139
depolarization, 16
Deus Ex: Human Revolution, 55
Dicken, Luke, 47
diffusion coefficient D, 14e
diffusion equation, 14e
diffusion technique, 13–15
Dijkstra algorithm, 247, 280–281, 284–285, 318–319, 322–323
Dill, Kevin, 3, 61
disbelief, suspension of, 4
Disney, Walt, 3
domain-specific language (DSL), 227–231, 340–341
Dragon Age: Origins, 246, 255, 274, 285
Dragon Quest, 122
Dragon Warrior, 122
dual utility AI, 121–122
Dunstan, Philip, 73
dynamic environments, supporting, 313–314, 352, 423
dynamic flow field tiles, 307, 323
dynamic link libraries (DLLs), 203–204

E

electrical attraction, 13. *See also* neurology
electrical processing myth, 12–13
Embodied Conversational Agents, 443
Emerson, Elijah, 307
Environment Tactical Querying (ETQ)
anatomy of, 425–426
components of, 425
context object, 425
editor tool, 428–429, 429f
final tests, 430
goals for, 424
implementation details, 428, 431
improving, 430–431
merging tests, 430
motivation for, 423–424
multithreaded implementation, 431
overview of, 423–424
philosophy on, 424
pros/cons of, 430
query template, 425, 428–429, 429f
reversed processing, 431
test structures, 425–428, 427t, 430–431
validity tests, 426
EverSky, 433–440
exponential moving average (EMA), 189, 189e, 190e, 190f

F

Far Cry, 343–344, 348
F.E.A.R., 55
Fieldrunners 2, 317–323
filtered randomness. *See also* randomness techniques
anomalies, 35–37
binary randomness, 35–36
explanation of, 30, 34–38, 43
floating-point ranges, 36–37
generating, 34–37
implementing, 37–38
integer ranges, 36
integrity of, 37
Final Fantasy, 122
Final Fantasy XIV: A Realm Reborn, 269–271, 270f, 285
finite-state machines (FSMs)
behavior trees and, 63
for decision-making systems, 137–143, 140f, 146

examples of, 48–49, 48f, 50f, 51f, 140f, 143f
states of, 48–52, 57–58
for structural architecture, 63
transitions of, 48–52
first-person shooter (FPS), 62–63, 121–122, 241, 377, 453
Fitch, Bob, 7
flow fields, 307–318, 313f, 319f, 320–323
Forbes, Kevin, 421
formation movements
band formation, 294
calculating positions, 290–293, 290f, 291e, 292f, 293f
column formation, 294
introduction to, 289
navigating, 293–295
path generation, 289–293, 290f
Forza, 579
Franco, Simon, 549
From Neuron to Brain, 12
fundamental discontinuity (FD), 188
Funge, John, 52
Furman, Lukasz, 431

G

Game AI
architectures for, 5–7, 6f, 45–60
authorial control, 5
behavior selection algorithms, 47–60
behavior trees, 6, 73–91, 93–111, 127–136
C++ for, 62, 201–217
cheating and, 7–8
controlling, 62
creating experience with, 4–9
definition of, 3
description of, 3–9
goal of, 4, 8–9
GPGPU for, 539–547
heuristic functions of, 5–6, 63–64, 102–104
introduction to, 3–9
LOD trader, 185–200
neurology and, 11–27
N-grams for, 567–580
nondeterminism, 5, 6
purpose of, 4, 8–9
racing games for, 471–479, 481–490
randomness techniques for, 29–43
rule-based AI, 63
scalability in, 6–7
scripting and, 219–237
simplicity in, 6–7
structural architecture, 61–71
stylized AI, 567–580
utility-based AI, 63
Game Engine Architecture, 437
Game Programming Gems 2, 391
Gargolinski, Steve, 47
Gaussian randomness. *See also* randomness techniques
applications of, 32–33, 43
explanation of, 29–33
filtering, 37
generating, 31–33, 33f
uniform distribution and, 30–33, 33f
Gears of War: Judgment, 430
global blackboards, 66–67
goal stack, 65
goal-oriented action planning (GOAP)
for behavior selection algorithms, 7, 55–59
for decision-making systems, 137
for first-person shooter (FPS), 63
as single-unit planner, 170
speeding up planning, 165
structural architectures, 63
Goldblatt, Jay, 29
GPGPU
converting to, 541–542
flocking, 541–542
history of, 540
introduction to, 539–540
OpenCL, 540–542
OpenCL setup, 542–545
results for, 545–547, 546f
sharing processing, 545
Graham, David "Rez," 55, 113, 451
graphical modeling language-based decision models, 137
Gravot, Fabien, 269
Green Eggs and Ham, 577
Gregory, Jason, 437
Growth-Based Approach to the Automatic Generation of Navigation Meshes, 266

H

Hale, D. Hunter, 259
Halo 2, 84
Halo series, 128
Havok Script, 203
heat vision system
explanation of, 502

heat line, 502–503, 504f
heat signatures, 502–505, 503f
implementation, 504–505
introduction to, 501
output smoothing, 502
for racing, 501–505
track position, 502–505, 504f
Hebb, Donald, 22
Hebb rule, 22–23, 22e, 23e
Hertel–Melhlorn decompositions, 265–267
heuristic functions, 5–6, 63–64, 102–104. *See also* Game AI
hierarchical AI. *See also* multiplayer bots
in action, 388
architecture for, 378, 379f
commander AI, 378, 379f, 386–388
cover data, 383
description of, 378
future of, 388
individual layer, 378, 379f
influence maps, 384–385
introduction to, 377
level annotations for, 388
for multiplayer bots, 377–390
pathfinder for, 386f
primitive tasks, 380, 381t, 384t
squad AI, 383–384
squad layer, 378, 379f
squad tactical reasoning, 384–386, 386f
squad update loop, 383–384
strategic graphs, 384–385
tactical reasoning, 383
waypoint graphs, 383
hierarchical finite-state machines (HFSMs), 50–52, 51f, 137
hierarchical pathfinding, 246
hierarchical reasoning, 61–62, 64, 69
hierarchical task networks (HTNs)
building blocks of, 150–154
compound tasks, 58–59, 152–154
domain for, 151f, 154–155, 157f, 161f
examples of, 58–59, 149–167
exploring planners, 149–167
for first-person shooter (FPS), 63
introduction to, 149–150
method traversal record (MTR), 161
for multiplayer bots, 380
overview of, 150–151, 151f
partial plans, 165–167
plan for, 155–156
plan-space planning, 172–175, 172f, 180–182, 180f, 181f
primitive tasks, 58–59, 152–154, 380, 381t, 384t
priority plans for, 160–163, 161f
recursion for, 157–158
sensors for, 152
simultaneous behaviors, 163–165
speeding up planning, 165–167
state changers, 150–153, 155–159
tasks of, 57–59
world state changers, 150–153, 155–159
Hilburn, Daniel, 99
Hitman: Absolution, 297, 299, 305
Homeworld, 541
HTN planners. *See also* hierarchical task networks
building blocks of, 150–154
compound tasks, 152–154
domain for, 151f, 154–155, 157f, 161f
exploring, 149–167
introduction to, 149–150
method traversal record (MTR), 161
overview of, 150–151, 151f
partial plans, 165–167
primitive tasks, 152–154
priority plans for, 160–163, 161f
recursion for, 157–158
running plan, 156–157
sensors for, 152
simultaneous behaviors, 163–165
speeding up planning, 165–167
state changers, 150–153, 155–159
world state changers, 150–153, 155–159
Humphreys, Troy, 47, 149
hybrid architecture, 146
hyperpolarization, 16
hysteresis, 18–19

I

"illusion of intelligence," 3–4
"illusion of life," 3
inertia, 65, 122
instant check mode, 77–78
intelligence, illusion of, 3–4
intelligence, placing, 67–68
interest sources
for agent scripting, 420
definition of, 416–417
examples of, 414–417, 415f

group behavior, 417–418
improvements on, 420–421
investigation, 419–420
limitations of, 420–421
prioritizing interests, 418–419
rediscoverability, 419–420
updates from, 417–418
Introduction to Utility Theory, An, 55
inverse kinematic (IK) controllers, 333
ion channels, 15–17
ion concentration, 13–17, 14f. *See also* neurology
Isla, Damian, 6
island fields, 315

J

Jack, Matthew, 201, 337
James, William, 532
Jedi AI
actions for, 103–111
changes to, 108–111
constraints for, 103–106
Jedi behavior tree, 103–107
Jedi memory, 103–107
Jedi mistakes, 110–111
Jedi simulation, 107–108
Jedi skill levels, 109–110
Johansson, Jonas, 358
Johnson, Tatham, 421
Just Cause 2, 55

K

Kernighan, Brian, 7
KillZone 2, 57
Killzone 3, 343, 377–390
Kinect Star Wars, 103–111
Kingdoms of Amalur: Reckoning, 93–98, 369–374
Kirst, Kevin, 358, 404
Kleve, Hylke, 377
knowledge management, 66–69
knowledge representation
accessing knowledge, 439
agent awareness and, 401–440
blackboards, 437, 439
character behaviors, 433–434
character relationships, 435–436
classifications, 435
debugging scripts, 439
design requirements, 433–434
entity attributes, 434–435
event handling, 439
events, 436, 439
execution, 438
existence, 435
implementation, 437–438
introduction to, 433
of others, 436
semantic knowledge, 436
sensing, 438
serialization, 439
set membership, 435–436
social dynamics system, 441–450
system for, 433–440
Kohan 2, 8
Kore Script, 203
Kung-Fu Circle, 369–370

L

Laming, Brett, 11
Laplacian operator, 14
level-of-detail (LOD) control, 185–200. *See also* LOD Trader
Lewis, Mike, 219
life, illusion of, 3
lifelike characters, 451–458. *See also* characters
linear probability, 187e, 187f
locomotion
animation-driven locomotion, 297–299, 325–334
collision avoidance for, 299–300
collision sphere motion, 301
introduction to, 297
movement and, 297–305, 325–334
performance of, 304–305
planning, 325–334
preplanned locomotion, 297–305
reciprocal velocity obstacle technique, 297–299, 304–305
time per movement, 300
velocity vector, 300
visual quality of, 304–305
LOD Trader
algorithm for, 192–199
break in realism (BIR), 187–188, 191–193
character criticality, 188–189
defining problems, 186–187
expansion heuristic, 196–197, 197e
expansion queue, 195–196, 198
exponential moving average (EMA), 189, 189e, 190e, 190f
extensions to, 198–199

features of, 193–194
introduction to, 185–186
memory model, 191e
modeling costs, 191
multiple resources, 197–198
observability, 189–190
in practice, 199–200
probability and, 186–188
resource multiplier vector, 197e
return time, 191e
transitions, pruning, 196–197, 196e
logarithmic probability, 187e, 187f
Lorenz, Konrad, 537
Lua scripting language, 94–97, 202, 215, 232, 340–341, 439, 553, 561

M

Mark, Dave, 47, 55
Mark of the Ninja, 413–421
Martel, Eric, 557
Martins, Márcio, 358
Mateas, Michael, 515
Matrix, The, 403
McCoy, Josh, 515
Melder, Nic, 471, 491, 501, 507
memory access performance, 86–87
Merrill, Bill, 127
method traversal record (MTR), 161
Miles, Brook, 413
Millington, Ian, 52
Miyake, Youichiro, 269
MMO games
algorithm for, 278–281, 284–285
alternative uses, 285
component approach, 274–276, 277f
component center, 281–284, 282f
connectivity for, 275, 275f, 277f, 278
convexity problems, 282
distances between polygons, 279, 279f
falling meshes, 278
flow chart of, 271f
hierarchical levels, 277, 277f
hierarchical table builder, 280, 280f
inconsistencies in, 281–282, 282f
introduction to, 269–270
invalid sublinks, 283–284, 283f
maps for, 285, 286f
mesh generation, 271–273, 272f, 273f, 274f
navigation meshes, 271–273, 272f, 273f, 274f, 285, 286f
navigation system, 271, 271f, 284–285
overview of, 270–271
pathfinding for, 269–287
pathfinding requests, 276–277, 276f
precomputed solutions for, 269–287
problematic components, 281–284, 282f, 283f
results for, 284–285
screenshot of, 270f
smoothed distances, 284
table builder, 279–280, 280f
table builder algorithm, 284–285
table connectivity, 275, 275f, 277f
table generation, 274–276
table generation algorithm, 278–281
table inconsistencies, 281–282, 282f
table size, 285
table use on server, 284
tool chain, 271
MMORPG, 255
mobile games, 317–323. *See also* crowd simulation
modularity, 61, 68–69, 128, 150, 162–163, 167
monitoring mode, 77–78
Morcus, Robert, 377
morphology, 19–20, 20f
Mortal Kombat, 567
movement. *See also* locomotion
agent collision radius (ACR), 300
animation-driven locomotion, 297–299, 325–334
collision avoidance for, 297–305
collision resolution, 302–303
crowd simulation for, 317–323
pathfinding and, 239–252
reciprocal velocity obstacle technique, 297–299, 304–305
time per movement, 300
velocity vector, 300
multilayer perceptrons (MLPs), 392–396. *See also* neural networks
multiplayer bots. *See also* hierarchical AI
architecture for, 378–379
bot AI, 378, 379f
commander AI, 386–388
cover data, 383
description of, 378
hierarchical AI for, 377–390
individual layer, 379–380, 379f, 380f
individual planner, 380–383
individual tactical reasoning, 383
influence maps, 384–385

introduction to, 377
pathfinder for, 386f
squad AI, 383–384
squad tactical reasoning, 384–386, 386f
squad update loop, 383–384
strategic graphs, 384–385
strategy for, 377–390
strategy layer for, 378, 379f
waypoint graphs, 383

N

navigation meshes (NavMeshes)
cover representation, 364–367
CoverMap, 364–367, 364f, 365f
creating, 259–267
in MMO games, 271–273, 272f, 273f, 274f
path segment calculations, 365–366, 366f
search space representations, 246f, 253, 254f, 255–267
tactical pathfinding and, 361–368
tessellating and, 363
neural networks
behavior adjustments, 397
benefits of, 397–398
debugging, 396–397
drawbacks of, 398
explanation of, 392
fitness function, 394–397
hidden nodes, 393–394
inputs for, 392–393
introduction to, 391
learning parameters for, 395–396
multilayer perceptrons (MLPs), 392–396
outputs for, 393, 395e
performance of, 397
setting up, 392–394
training, 394–397
neurology
action potential, 15–21, 16f, 17f, 18f
axon, 19–20, 20f
axon terminals, 19–23, 20f
cell location, 15e
cell membrane, 12–13
critical thinking skills, 11–12, 26
dendrites, 19–20, 20f
depolarization, 16
diffusion coefficient D, 14e
diffusion equation, 14e
diffusion technique, 13–15
efference copy, 25
electrical attraction, 13
electrical processing myth, 12–13
EPSP (excitatory postsynaptic membrane potential), 20–21
equilibrium, 15
Game AI and, 11–27
hyperpolarization, 16
hysteresis, 18–19
introspection, 11–12
ion channels, 15–17
ion concentrations, 13–17, 14f
IPSP (inhibitory postsynaptic membrane potential), 20–21
membrane channels, 15–17, 16f, 18f
membrane potential, 15–17, 16f
membraned pumps, 13
morphology, 19–20, 20f
motor neuron, 24–25
myelin sheath, 20, 20f
neurons, 12–15, 14f, 19–20, 20f, 21f
neurotransmitters, 20–21, 21f
open-loop ballistics, 25
pathways, 24–25, 24e
perceptron, 22–23, 22e, 23e, 27
perturbation, 15
plasticity, 20–22
postsynaptic receptors, 20–21, 21f
proprioceptors, 25
Ranvier nodes, 20, 20f
reflex arc, 24–26, 24e, 24f
refractory period, 16, 17f
repolarization, 16, 18f
resting potential, 13
saccades, 25
Schmitt trigger, 18–19
signal transmission, 17–18, 18f
soma, 19–20, 20f
spike activity, 15–21
stimuli on brain, 24, 24t
study of, 12–27
synaptic cleft, 20–21, 21f
synaptic transmission, 20–21, 21f
vesicles, 20–21, 21f
voltage, 12–13
neurons, 12–15, 14f, 19–20, 20f, 21f
neurotransmitters, 20–21, 21f
N-grams
changes to, 577–578
computation time, 573
corpus, 577

events and, 571–572, 572e, 573e, 574–580
explanation of, 568–571
future research on, 579–580
implementing, 571–578
introduction to, 567–568
issues with, 571
large numbers and, 571
memory for, 578
mimicking style, 576–577
new trends, 578
offline learning, 576
order of, 568–569, 569f, 570t
pattern occurrences, 572–573, 572e, 573e
player imitation, 579–580
for player prediction, 567–580
power of, 578–579
predicting with, 569–575
probability and, 568–570, 568e, 570e, 571e, 579
for procedural generation, 567–580
robustness for, 272
Rock-Paper-Scissors game, 571–575, 573e
self-accountability and, 573–574
time delays, 575, 579
understanding, 568–571
Nicholls, J. G., 12
nondeterminism, 5, 6
nontrivial collision resolution, 302–304
NPC attacks
behavior integration, 372–374
Belgian AI, 370, 373–374
capacity for, 370–371
circles approach, 371–373, 373f
concerns about, 373–374
grid assignments, 374–375
grid capacity, 370–371, 371f
grid sectors, 371–372
introduction to, 369
Kung-Fu Circle and, 369–370
managing, 369–375
restrictions on, 374–375
scaling difficulty, 373
troll, 372f
NPC awareness
examples of, 47
finite-state machines and, 48–52, 50f, 56–58
interest sources, 414–418, 415f
introduction to, 413–414
senses, 415–416
sight interest sources, 415–416
sound interest sources, 415–416, 415f
vision cones, 415–416, 415f
NPC goals, 56–59

O

OpenCL, 540–542
OpenCL setup, 542–545
open-loop ballistics, 25
optimization. *See* pathfinding optimizations
option stacks, 61–62, 65–69
Orkin, Jeff, 55

P

Paris Game/AI conference, 215–216
path modification system, 302–304, 303f
pathfinding. *See also* pathfinding optimizations
architecture optimizations, 241–252
avoidance point (AP), 302–304
crowd pathfinding, 307–316
crowd simulation, 317–323
flow field tiles, 307–316
movement and, 239–252
navigation meshes for, 259–267
on NavMesh, 361–368
path modification system, 302–304, 303f
precomputed pathfinding for MMO servers, 269–287
reconnection point (RP), 302–304
search space representations, 253–258
steering circles for, 289–295, 307–316
tactics for, 361–368
pathfinding optimizations
A* implementation, 242–250
backtracking issues, 249
bad ideas for, 249–250
better heuristics, 248
bidirectional pathfinding, 250
caching successors, 249–250
differences in, 242
Euclidean embedding, 244–245, 245f
failed paths, 250
hierarchical pathfinding, 246
high-quality heuristics, 242–245
implementing, 242–250
introduction to, 241–242
look-up table for, 243–244, 243f
lossless compression, 244
lossy compression, 244
magnitude of, 242–245
open list sorting, 248–249

optimal search space representation, 245–246, 246f
overestimating heuristics, 247e
path generation, 243–244, 243f
preallocating memory, 246–247
Roy–Floyd–Warshall algorithm, 242–244, 243f, 248
search space representations, 245–246, 246f
simultaneous searches, 249–250
techniques for, 242–250
pathways, 24–25, 24e
Pentheny, Graham, 317
perception system
ADSR envelopes, 408–410, 409f
attack phase, 409
data driving, 410–411
decay phase, 409
introduction to, 403
modifiers, 410
overview of, 404–405, 404f
perception manager, 404, 404f, 407–408
pulses, 410
release phase, 410
sight stims, 406
sound stims, 405–406
stims, 404–407, 404f, 407t
sustain phase, 409–410
target tracks manager, 404, 404f, 408–409
threat levels, 410
perceptron, 22–23, 22e, 23e, 27. *See also* neurology
Perlin, Ken, 38
Perlin noise. *See also* randomness techniques
amplitude, 40–42
controlling, 42
demo of, 42
equation for, 40e, 42e
examples of, 38f
generating, 40–43
interpolation choice, 42
octaves of, 40–42, 41f
in one-dimension, 30, 38–43, 39f, 41f
persistence value, 40–42
purpose of, 38–39
sampling, 42
perturbation, 15, 466, 489
PHP script, 202
PID controllers
adaptive control, 497–498
braking distance, 499
channel cross-over, 499
characteristics of, 493–494, 493f, 494e, 494t
closed-loop control, 492–493, 492f
debugging, 496
derivative control, 495
designing, 496–497
determining success, 496
drifting, 499
error memory, 495
error offset, 496
gain scheduling, 497
grip loss, 499
implementing, 495–496
input data, 496
integral control, 494–495
introduction to, 491–494
K values, 496, 497
model reference adaptive control (MRAC), 498
open-loop control, 492–493, 492f
other racing applications, 498–500
predictive control, 498
priority mixing, 499–500, 500e
proportional control, 494
for racing games, 491–500
recovery behavior, 499
speed controller, 498–499
steering controller, 498
stopping distance, 499
time constants, 495
tuning, 496–497
planning combat maneuvers
examples of, 173t, 174f, 175f, 176f, 179f
introduction to, 169
multi-unit planning, 170–171, 170f, 172f
planning for, 169–183
plan-space planning
for combat maneuvers, 169–183
combinatorial explosion, 180
efficiency of, 180–182, 180f, 181f
examples of, 171f, 172f, 173f, 179f
future of, 182
introduction to, 169
partial plans, 178–180
plan of tasks, 173–175, 173t, 174f, 175f, 176f
planner main loop, 172–173, 182
planner methods, 175–178, 177t
plasticity, 20–22. *See also* neurology
play testing, 160, 236, 388
players, fooling, 459–467

players, predicting, 567–580
Pokémon Ruby Version, 580
Pokémon Sapphire Version, 580
Powers, William, 535
precomputed pathfinding, 269–270. *See also* pathfinding
procedural generation, 567–580. *See also* N-grams
procedural knowledge, 144–145
Prom Week, 515–527
Pulse, 204f, 207
Punch-Out!, 578
Python scripting language, 215, 232, 561

Q

Quake 3, 265
queries
  anatomy of, 425–426
  components of, 425
  context object, 425
  editor tool, 428–429, 429f
  final tests, 430
  goals for, 424
  implementation details, 428, 431
  improving, 430–431
  merging tests, 430
  motivation for, 423–424
  multithreaded implementation, 431
  overview of, 423–424
  philosophy on, 424
  pros and cons of, 430
  query template, 425, 428–429, 429f
  reversed processing, 431
  tactical queries, 424–431
  test structures, 425–428, 427t, 430–431
  validity tests, 426

R

Rabin, Steve, 29, 241
race tracks. *See also* racing games
  alternate lines, 489
  braking distance, 487, 487e
  branches, 483
  corner speed, 487, 487e
  curvature of, 485f, 486, 486e
  defending lines, 489
  driving, 486–488
  generating, 483, 483e
  introduction to, 481–482
  node spacing, 483–484, 485f
  open-ended tracks, 483
  piecewise linear curves, 486, 489
  racing lines, 486, 488–490
  radius of, 485f, 486, 486e
  registration of, 484–485, 484e, 485f
  representation of, 482–484, 482f
  segments of, 482f
  splines, 489–490
  steering, 486–487
  tactical information, 489
  track data calculations, 484–486
  track positioning tips, 501–505
  wall avoidance, 487–488
racing games. *See also* PID controllers; race tracks
  AI controlled vehicles, 481–490
  architecture for, 471–479
  balancing AI, 476–478
  branches, 475
  collision avoidance, 472
  defend and block state, 475
  driver persona, 472–473
  driving behaviors, 473–476
  for Game AI, 471–479, 481–490
  heat vision system for, 501–505
  interfaces for, 476
  introduction to, 471
  layers of, 472
  managing game play for, 507–512
  multiple routes, 475
  offline learning, 477–478
  overtaking mode, 474–475
  perturbation, 489
  physics of, 471–472
  race choreographer, 476
  racing behaviors, 473–476
  recovery behavior, 475–476
  rubber-banding system for, 477, 507–512
  track for, 481–490
  track positioning tips, 501–505
Ramsey, Michael, 531
randomness techniques
  filtered randomness, 30, 34–38, 43
  Gaussian randomness, 29–33, 43
  introduction to, 29–30
  Perlin noise, 30, 38–43
rapid AI development, 201–217
reactivity
  of characters, 5
  cognitive reactions, 138–139, 144–145

danger awareness, 144–145
in decision-making systems, 137–147
examples of, 145f, 146f
explanation of, 138–139
involuntary reactions, 138–139, 144
procedural knowledge, 144–145
timeline of, 138f, 139f
realism
break in, 187–188, 191–193
fundamental discontinuity (FD), 188
unrealistic long-term behavior (ULTB), 189
unrealistic state (US), 188
visual realism, 449
real-world behavior trees. *See also* behavior trees
behavior asset management, 96
behavior class improvements, 97
debugging behaviors, 97–98
defining, 94–97
enhancements of, 96–98
flexibility of, 98
integration of, 95
introduction to, 93
Lua scripting language, 94–97
overview of, 93–94
previously running behaviors, 97–98
in script, 93–98
script behaviors, 94–95
script concerns, 95–98
selectors and, 98
reasoning, hierarchical, 61–62, 64, 69
reciprocal velocity obstacle technique (RVO), 297–299, 304–305
Reckoning, 93–98, 369–374
reconnection point (RP), 302–304
Red Dead Redemption, 67
reflex arc, 24–26, 24e, 24f
refractory period, 16, 17f
repolarization, 16, 18f
Reunion, 224
Reynolds, Craig, 39, 319–320, 541–542
Robbins, Michael, 391
Roccucci, Francesco, 358
Rock-Paper-Scissors, 571–575, 573e
Rodriguez, Benito, 358
role-playing games (RPGs), 55, 68, 115, 122, 221, 245, 369, 452–453, 528
Rosenblatt, Frank, 22
Rosenblatt perceptron, 22–23, 22e, 23e, 27
Roy–Floyd–Warshall algorithm, 242–244, 243f, 248
RTS games, 245, 309, 316, 384, 541, 545–546, 575, 578
rubber-banding system
difficulty-based rubber-banding, 508–509, 509f
disabling, 510
first place rubber-banding, 511
implementation of, 507, 508f
improvements in, 510–511
introduction to, 507–508
issues, 509–510
for long races, 512
for multiplayer modes, 510
parameters for, 511
power-based rubber-banding, 508–510, 509f
race pace system, 512
for racing games, 477, 507–512
rule-based AI, 63. *See also* Game AI
Runtime Compiled C++ (RCC++)
approaches to, 202–205
behavior trees and, 214–215
blackboards and, 214–215
code optimizations for, 213–214, 216
code-state preservation of, 212–213
Crytek case study, 215–216
debugging, 202, 205
dynamic link libraries (DLLs), 203–204
error recovery, 210–212
examples of, 204f, 207–210
future of, 216
goals with, 216
header file handling of, 212–213
implementing, 205–207
introduction to, 201–202
multiple processes of, 203
performance-critical code, 213–214, 216
Runtime Compiler, 205
Runtime Object System, 206
scripting languages, 202–203
shared objects, 203–204
structured exception handling (SEH), 210
Visual Scripting and, 203
Visual Studio and, 203–205, 204f, 211, 216
Rutkowski, Adam, 216

S

Sakata, Shinpei, 285
Schmitt, Otto, 18
Schmitt trigger, 18–19

"scripted AI," 5–6, 219, 227, 232–236. *See also* scripting
scripting. *See also* scripting languages
- attribute checking, 226–228
- behavior trees and, 222
- building scripting system, 225–232
- case studies, 224–225
- cross-pollination, 223–224
- deduplication technique, 226–227
- delegation of, 221–222, 236–237
- domain-specific language (DSL), 227–231, 340–341
- flexibility of, 221–223, 235–236
- Game AI and, 219–237
- as "glue," 220, 235–236
- implementation techniques, 225–232, 236–237
- integrated architectures, 231–232
- introduction to, 219–220
- iteration of, 233, 236
- master-style systems, 220–225
- narratives, 235–236
- observation system, 225–226
- overusing, 233
- play testing, 236
- prioritization of, 221–222
- reaction system, 225–226
- "scripted AI," 5–6, 219, 227, 232–236
- sequencing technique, 226, 228
- servant-style systems, 220–225
- space-simulation games, 224–225
- surprises in, 235
- transitions in, 233–234
- triggers for, 222–223
- "tripwire" technique, 219, 223–228, 234–235
- variety in, 234–235
- working with, 220–225
- writing scripts, 232–236
- "X Series," 224–225

scripting languages
- C++ approach, 201–205, 232, 340
- domain-specific language (DSL), 227–231, 340–341
- embedded scripts, 553
- exposed scripts, 561
- Havok Script, 203
- Kore Script, 203
- Lua scripting language, 94–97, 202, 215, 232, 340–341, 439, 553, 561
- PHP script, 202
- Python scripting language, 215, 232, 561
- Runtime Compiled C++, 201–217
- UnrealScript, 202
- Visual Studio, 203–205, 210–211, 216

S-curve function, 40, 42
search space representations
- choosing, 253–258
- dynamic modifications, 254
- examples of, 246f, 254f
- graphs, 246–247, 246f, 254f, 256–257
- grids, 246–247, 246f, 254–255, 254f, 257
- introduction to, 253–254
- localization, 254–257
- memory usage, 254–257
- navigation meshes, 246f, 253, 254f, 255–267
- path following, 254–257
- path planning, 254–257
- path smoothing, 254–257
- pathfinding optimizations, 246–247, 246f
- tasks, 254–255
- waypoint graphs, 246–247, 246f, 253–259, 254f, 362–364, 383–386

sequencing technique, 226, 228
Shank, 413–414
Shank 2, 413–414, 421
"shared objects," 203–204
signal transmission, 17–18, 18f
Silva, Fernando, 29
Silva, Mario, 358
Sims, The, 5, 54, 61, 67, 119–122
Sims 3, The, 528
Sims Medieval, The, 122, 452–457
social dynamics system
- for characters, 441–450
- components of, 441–449, 448f
- execution of, 449
- explanation of, 442–443
- gaze control, 442, 445–446, 445f
- gestures, 443, 447–448
- implementation of, 443–449
- introduction to, 441
- observation of characters, 441–442
- posture, 443, 446–447
- practical systems, 441–450
- proxemic control, 442, 446, 447f
- social origin, 446
- visual realism, 449

social facts knowledge base (SFKB), 522–523
SoftCode, 215–216

soundscapes
artificial sound designer, 550–551
changing, 554
creating dynamic sound, 549–555
database for, 551–553
generating events, 551
introduction to, 549–550
play-request events, 554
rule set for, 553
updating, 554–555
source cost data, 314
space-simulation games, 224–225
spatial problems, solving, 19
squad AI, 383–384, 515
Stanford Research Institute Problem Solver (STRIPS), 55, 128, 165
state stack, 65
state-space planning, 171f
stealth platformer, 413–421
steering behaviors
benefits of, 322
flow fields, 319f, 320–323
future of, 322–323
limitations of, 321–322
performance considerations, 321–322
units for, 319–321, 320f
steering circles
calculating positions, 290–293, 290f, 291e, 292f, 293f
formation movement for, 289–295
introduction to, 289, 307
navigating, 293–295
path generation, 289–293, 290f
pathfinding, 289–295, 307–316
Straatman, Remco, 377
strategy
for multiplayer bots, 377–390
strategic graphs, 384–385
strategy layer, 378, 379f
tactics and, 335–359
Street Fighter II, 575–576
Street Fighter IV: 3D Edition, 579
structural architecture
behavior trees, 63
blackboards, 61, 66–67
common architectures, 62–63
definitions for, 62
finite-state machines (FSMs), 63
first-person shooter (FPS), 62–63
goal stack, 65
goal-oriented action planning (GOAP), 63
hierarchical reasoning, 61, 62, 64, 69
inertia in, 65
introduction to, 61
knowledge management, 66–69
modularity, 68–69
option stacks, 61–62, 65–69
rule-based AI, 63
scripting basics, 63
state stack, 65
subsumption, 65
tricks for, 61–71
utility-based AI, 63
structured exception handling (SEH), 210
Sturtevant, Nathan R., 241, 253
Sunshine-Hill, Ben, 185, 459
Super Street Fighter IV: 3D Edition, 579
Supreme Commander 2, 307–309, 309f, 314–316, 391–398
Supreme Commander 3, 398
"suspension of disbelief," 4
symbolic planning language-based decision models, 137
synaptic plasticity, 21–22
synaptic transmission, 20–21, 21f. *See also* neurology

T

tactical pathfinding
for 3D applications, 366–367, 368f
cover representation, 364–367
introduction to, 361
methods for, 361–366
navigation meshes and, 361–368
path segment calculations, 365–366, 366f
raycasts, 362
tessellating and, 363
tactical position selection
architecture for, 338–339, 339f
behavior selection algorithms, 342–343
database for, 339, 343
directness of, 346–348, 347e
domain-specific language (DSL), 340–341
evaluation of, 353–357
filters for, 344–345, 353–354
fundamentals of, 338
future of, 358
generation of points, 341–344, 351–352, 358
group techniques, 349–351
hidespot contention, 350

introduction to, 337
line of sight for, 341
motivation for, 338
performance of, 351–357
query language for, 339–343, 358
query library for, 342, 358
raycasts, 352, 358
scheduler for, 339
spatial coherency, 349–350
squad center, 349–350
squadmates, 350
timeslicing, 356–357
tools for, 357–358
weights, 345–349
tactics
for NPC attacks, 369–375
for pathfinding, 361–368
position selections, 337–359
strategy and, 335–359
Terran Conflict, 224–225
Threat, The, 224
threat response, 391–398. *See also* neural networks
time per movement (TPM), 300
Tomlinson, Simon, 471, 491
Transformers: Fall of Cybertron, 57–58, 156, 159, 167
Transformers: War for Cybertron, 156
Trapezoidal Cell Decomposition, 265–267
tripwire systems, 219, 223–228, 234–235

U

Unreal Engine 3, 341, 428
Unreal Engine 4, 202, 205
Unreal Kismet, 203, 216
unrealistic long-term behavior (ULTB), 189
unrealistic state (US), 188
UnrealScript, 202
utility curves
attack desire curve, 122–123, 123f
health desire curve, 123–124, 124e, 124f
linear curves, 117, 117e, 117f, 119–120, 120f
piecewise linear curves, 119–120, 120f
quadratic curves, 117–118, 118e
run desire curve, 124–125, 124f, 125f
threat curve, 123–124, 124f
utility decisions, 127–136
utility scores, 114–115, 115e, 116f
utility systems, 53–55
utility theory
actions for, 120–125
attack desire curve, 122–123, 123f
behavior trees and, 130
bucketing, 121–122, 121f
calculating utility, 116–120
concept of, 113–114, 130
cool-downs, 122
cubed utility, 135f
decision factors, 115–117, 122–127, 123e
decision-making systems, 125–126, 130–132
demos, 122–125
dual utility, 121–122
expected utility, 115, 115e
health desire curve, 123–124, 124f
inertia and, 122
introduction to, 113
linear curves, 117, 117e, 117f, 119–120, 120f
logistic function, 118–119, 119e, 119f
piecewise linear curves, 119–120, 120f
quadratic curves, 117–118, 118e, 118f
run desire curve, 124–125, 124f, 125f
scores, 114–115, 115e, 116f
threat curve, 123–124, 124f
utility curves, 117–118, 122–126
utility-based AI, 63. *See also* Game AI

V

Van der Sterren, William, 169
Vasquez, Joseph, II, 567
Verweij, Tim, 377
visual realism, 449
Visual Scripting, 203
Visual Studio, 203–205, 204f, 211, 216

W

war games
examples of, 173t, 174f, 175f, 176f
introduction to, 169
multi-unit planning, 170–171, 170f, 172f, 181f
planning for, 169–183
plan-space planning, 172–182, 181f
Warzone, 377, 378, 388
wavefront edge expansions
algorithm for, 261–267
classification of, 261–263
comparisons, 265–267
decomposition, 261–266, 262f, 266f
Delaunay Triangulation Decompositions, 256, 260, 265

Hertel–Melhlorn decompositions, 265–267
introduction to, 259–260
navigation meshes, 259–267
obstructions, 260–265, 266f
PASFV, 260–261, 264
postdecomposition, 264–265
reseeding, 264
runtime, 260, 265
seeding, 261–264
space-filling volumes (SFVs), 260–261, 264
spatial decomposition, 261–266, 262f, 266f
Trapezoidal Cell Decomposition, 265–267
VASFV, 260–261, 264
waypoint graphs, 246–247, 246f, 253–259, 254f, 362–364, 383–386
Weizenbaum, Joseph, 4
Welsh, Rich, 403
Wilson, Will, 201
World of Warcraft, 5, 7
World of Zoo, 532

X

X2: The Threat, 224
X3: Reunion, 224
X3: Terran Conflict, 224–225

Y

Yokoyama, Takanori, 269
Youngblood, G. Michael, 259

Z

Zielinski, Agata, 431
Zielinski, Mieszko, 423
Zoo Tycoon, 5
Zoo Tycoon 2, 67