Index

Note: Page numbers followed by f and t refer to figures and tables, respectively.

3D regular grid, 266
1000 NPCs at 60 FPS game, 403–409
  action performance, 408–409
  action queues and open-loop, 408–409
  simplified model, path finding, 409
  action selection, 405–408
  daily scripts, 406–408
  propositional planner, 405–406
  overview, 403–404, 404f

A

AAA games, 63
A* algorithms, 216
AAT. See Animation Alias Tables (AAT)
Ability cached data, 385
Ability Picker, 385–386
  and bot ability usage markup, 384–386
  ability cached data, 385
  debugging, 386
  target’s spatial density, 386
  core algorithm, 385
A/B testing, 469
Acceptance condition, 310–311
Active Life AI system, 216
Activision Treyarch, 63
Ad-hoc communication, 123
Adjusted Stack Health, 43
Affordance, 331
  path gap, 334–335, 334f

AI
  active life, 216–218
  driver personalities, 218
  vehicle paths, 217
  agent, 129
  BTs, 127, 132–138
  basic functionality tests, 74–75
  Commander, 297–298
  agent scoring, 298
  humans, 298–299
  opportunistic, 299
  design goals, 362
  developer, 331
  Driver San Francisco. See Driver San Francisco, AI
  module, 6
  opponents, difficulty of, 37t
AIAction_Subreasoner, 90
AI-controlled vehicles, 215
AICreationData, 54
  instantiate and initialize objects, 55
AIDataStore base class, 94–95
  AIActor data store, 94
  AIBlackboard_Brain data store, 94
  AIBlackboard_Global data store, 94
  AIContact data store, 94–95
  AIThreat data store, 95
AI Director, 154
AIFactoryBase’s Create() function, 61
AI Game Pro (book), 132
AI Game Programming Wisdom (book), 132
AI Graph, 146–157
  body state machine, 156–157
debbuging, 153, 153f
features, 149–152
  blackboard, 149–150, 150f
data and overrides, 151, 152f
interrupting, 150–151, 152f
parallel thinking, 150, 151f
image, 147f
implementation techniques, 149
model, 148f
rule-based AI system, 156
structure and operation principles, 147–148
tool, 148–149, 148f
Aircraft in Warfare: The Dawn of the Fourth Arm (book), 301
A la carte power, 171
AllAbilities, 386
Alleyway, 339
AlphaBeta search, 184–185
AlphaGo program, 480
Alpha parameter, 467
Alternating least squares (ALS), 463–464
AltVtx, 197
Ambient interactions, 411–421
  build chain, 420
  extensions, 420–421
  beyond agents, 421
  templating, 421
interactive acts, 413–418
  interaction language model, 413–418
  STRIPS, 413, 414
tuple space, 413, 414
overview, 411–412
script execution, 418–419
  joining late, 419
  role allocation, 418–419
  rule evaluation, 419
smart objects to smart locations, 412–413
  notification emitter, 412
  player emitter, 413
  script emitter, 413
  spawn emitter, 413
Ambient Spawn Query, 25
Animation Alias Tables (AAT), 142–143
human soldier, 142f
Animation state machines (ASMs), 127, 138–143
  aiming and shooting, 142
  ATs, 138–139
  managing animations, 140–141
  transitions, 141
Animation Tables (ATs), 138–139
  human soldier, 138t
  aim table, 139t
ANSI C, 473, 473f
Anthropologically-oriented research, 406
Anthropomorphism, 4
Apache Mahout, 463, 465–466
Apache Spark, 463, 467
A* pathfinding algorithm, 409
Approach-and-surround behavior, 310, 311f
Area-of-Effect abilities (AoE), 386
Artificial life algorithms, 309
A* search algorithm, 265, 296
  node expansion, 271f
  vanilla, 272f
  wave-front exploration pattern, 272
ASMs. See Animation state machines (ASMs)
Assassin’s Creed Syndicate, 191
  challenges, 191–192
    navigation mesh patching, 192
    obstacle avoidance, 192
    representation comparison, 192–193
  convex hull generation, 194
    building the contour, 196–197
    cases to solve, 194–196
    convex hull expansion, 197–198
    rigid body simplification, 194
  obstacle avoidance, 198
    movement around convex hull, 199–200
    movement into convex hull, 200–201
    movement out of convex hull, 201
  obstacle detection, 198–199
  triangulation from convex hull data, 201
  velocity prediction, 201–202
Assassin’s Creed Unity, 413
ATs. See Animation Tables (ATs)
Attack tower, 329, 329f
  placement, 336–337
Attack window(s), 338–339
  decay, 340–341
spatial, 338
temporal, 338
agent size, 338
Attribute saliences, 437
Attrition
differential equations, 303
order, 303
rate coefficients, 303
B
Backtracking search, 419
Backward differencing operator ($\delta^-$), 253
Bathtub-shaped power curve, 254–255
Bayesian inference, 464
BDS. See Behavior Decision System (BDS)
Behavior
moods, 383–384
snippet, 373
pseudocode, 376
Behavior Decision System (BDS), 371–378
architecture, 372
behavior evaluation, 373–376
snippets comparison, 375–376
target selection, 374–375, 375f
utility calculation, 373–374, 374f, 374t
execution step, 376–377
framework, 374
evaluation-execution, 377
modularity and opportunities for reuse, 378
movement and passive behaviors, 377–378
overview, 371–372
potential actions, 372–373
BehaviorTreeNode, 132, 134
Behavior trees (BTs), 127, 132–138
AI agent, 132–133
handling parallel conditions, 135–137
interruption events, 137–138
nodes, 133–135
pitfalls, controlling of, 115–124
language implementation, 118–123
organizing classes, creating, 117–118
overview, 115–116
routing everything through blackboard, 123–124
Behemoth, 29
Belief
mutation, 445
graph, 438, 443, 443f
oscillation, 444
Believability, 399–402
conveying urgency, 399
interesting random targets, choosing, 399–401, 400f
targeting destructible objects, 401–402, 401f
Bench scenario, example rules for, 416–417
listening, 417
sitting down, 416–417
talk-action, 417
talking, 417
Berlin, 344–345, 347–349
Binary on/off switches, 169
Bioshock infinite’s game play pattern
matcher, 458
Bitwise operations, 269
Blackboard
extension, 383
system, 128–131, 146, 149–150, 150f
function-based variables, 130
human soldier archetype, 130t
implementation, 130–131
value-based variables, 129–130
Bot ability usage tags, 384, 384t
Bounding box approach, 312
Brain blackboard, 94
Brain-imaging techniques, 4–5
BTs. See Behavior trees (BTs)
Buffing, 381
Build chain, 420, 420f
Built-in library, 467
C
Camera frustum test, 323
Canonical Dijkstra, 283, 285–288
first step of, 286f
vs. JPS, 285–286
performance, 288
pseudocode, 287
Canonical ordering of paths, 284–285, 284f
black lines, 285
definition, 284
JPS in terms of, 285
paths duplication, 284
starting from state marked, 285f
Central limit theorem, 12
Chambolle–Pock Primal-Dual algorithm, 261–262, 262f
<table>
<thead>
<tr>
<th>Character</th>
<th>Combat, AI for game, 74, 74f</th>
</tr>
</thead>
<tbody>
<tr>
<td>knowledge phenomena simulation, 433–447</td>
<td>Compiler, 424</td>
</tr>
<tr>
<td>belief revision, 443–444</td>
<td>Complete digraph, 293</td>
</tr>
<tr>
<td>core procedure, 445</td>
<td>Complexity, 33</td>
</tr>
<tr>
<td>evidence, 440–441</td>
<td>Computer-generated random walk, 20</td>
</tr>
<tr>
<td>fallibility modeling, 443, 443f</td>
<td>Computer Shogi, 482</td>
</tr>
<tr>
<td>knowledge implantation, 444</td>
<td>Conceptual abstractions, 51–52</td>
</tr>
<tr>
<td>knowledge propagation, 442–443, 442f</td>
<td>Conceptual abstractions and modular components, 91, 96–105</td>
</tr>
<tr>
<td>mental models, 439–440</td>
<td>actions, 102–103</td>
</tr>
<tr>
<td>ontological structure, 438, 438f</td>
<td>considerations, 97–99</td>
</tr>
<tr>
<td>overview, 437</td>
<td>reasoners, 100–102</td>
</tr>
<tr>
<td>requirement, 437–438</td>
<td>regions, 104–105</td>
</tr>
<tr>
<td>salience computation, 441–442</td>
<td>targets, 103–104</td>
</tr>
<tr>
<td>statistics, 445</td>
<td>weight functions, 99–100</td>
</tr>
<tr>
<td>tunable parameters, 445</td>
<td>Confabulation, 440</td>
</tr>
<tr>
<td>saliences, 437</td>
<td>Connectivity mesh, 28–29, 28f</td>
</tr>
<tr>
<td>Charge attacks, 171–172</td>
<td>Connectors, 409</td>
</tr>
<tr>
<td>Chase target behavior, 216</td>
<td>Constructor, 56</td>
</tr>
<tr>
<td>Childless nodes, 269–270</td>
<td>Content-based filtering, 462</td>
</tr>
<tr>
<td>Chunk, 23</td>
<td>Content selection, 451</td>
</tr>
<tr>
<td>Circle-based obstacle avoidance, 192, 192f, 193f</td>
<td>algorithm, 453</td>
</tr>
<tr>
<td>Circular buffer, 69</td>
<td>Context sensitive, 427</td>
</tr>
<tr>
<td>Civilian Traffic AI, 216</td>
<td>Continuous query-based behaviors, 319, 320f</td>
</tr>
<tr>
<td>Client-server engine and AI System, 128–129, 129f</td>
<td>Continuous querying vs. destination validation, 319–320, 320f</td>
</tr>
<tr>
<td>Closed-loop feedback systems, 408</td>
<td>Continuous vs. sequential updates, 319–320, 319f</td>
</tr>
<tr>
<td>Closest graph node lookup grid, 294</td>
<td>query-based behaviors, 319, 320f</td>
</tr>
<tr>
<td>Collaborative filtering, 462–464</td>
<td>querying vs. destination validation, 319–320</td>
</tr>
<tr>
<td>item-based, 463–464</td>
<td>Controller’s button scheme, 64</td>
</tr>
<tr>
<td>UBCF, 463–465</td>
<td>Convex hull generation, 194</td>
</tr>
<tr>
<td>Collectible card game, 362</td>
<td>cases to solve, 194–196</td>
</tr>
<tr>
<td>CollectIntersections, 199</td>
<td>contour creation, 196–197</td>
</tr>
<tr>
<td>Collision avoidance, 211</td>
<td>convex hull expansion, 197–198</td>
</tr>
<tr>
<td>Collision-free movement, 239</td>
<td>rigid body simplification, 194</td>
</tr>
<tr>
<td>Combat AI accuracy, 393–402</td>
<td>Cooldown, 170, 170f</td>
</tr>
<tr>
<td>believability improvement, 399–402</td>
<td>Core procedure, 445</td>
</tr>
<tr>
<td>conveying urgency, 399</td>
<td>high-level pseudocode, 445</td>
</tr>
<tr>
<td>destructive objects target, 401–402, 401f</td>
<td>Corridor map method, 249–250, 250f</td>
</tr>
<tr>
<td>interesting random targets selection, 399–401</td>
<td>Craig Reynold’s historic boids program, 324</td>
</tr>
<tr>
<td>damage dynamics, 394</td>
<td>CRAN repository, 468</td>
</tr>
<tr>
<td>dynamic accuracy control, 395–397</td>
<td>CSV printer, 24</td>
</tr>
<tr>
<td>dynamic delay calculation, 397</td>
<td>CurrVtx, 197</td>
</tr>
<tr>
<td>final delay calculation, 395</td>
<td>D</td>
</tr>
<tr>
<td>multiple AIs and different archetypes, 397–398</td>
<td>Daily scripts, 406–408</td>
</tr>
<tr>
<td>dealing with changes, 398</td>
<td>Damage dynamics, 394</td>
</tr>
<tr>
<td>most relevant agent, selection, 398</td>
<td>Damage-over-time (DoT) attacks, 37</td>
</tr>
<tr>
<td>token systems, 394–395</td>
<td></td>
</tr>
</tbody>
</table>
Data
   analysis, 25
   compression, 479
   efficient, 479–484
   plausibility orderings as AI tool, 482–484
   elements, 105
   format, 24–25, 24t
   abstraction, 52–53
   and overrides, 151, 152f
   payload, 270
   primitives, recording, 65–67, 65f
   record handlers, 66–67
   structure, 373, 490–491
   grammars, 426–427
Database
   printer, 24–25
   Prolog, 488
Data-driving state machines, 163–164
Data-logging rate, 23
Daybreak Games, 466
Debugging, 386
   AI, 71
   Decision-making component, 116
   Decision-making logic, 90
   Decision-making system, 118
   Decision score evaluator (DSE), 168–170
   Decision tree representing script choices, 180f
Declarative programming, building custom
   static checkers, 487–496
   case studies, 495–496
   overview, 487–488
   problems to check, 493–495
   object configuration, 493–494
   type checking, 494
   uninterpreted strings, 494–495
Prolog, 488–492
   calling unity code, 491–492
   code as data and higher order code, 491
   database, 488
   data structures, 490–491
   logical deduction, 489–490
   querying database, 488–489
   rules, 489
   static checker, writing, 492–493
Decorator
   nodes, 382
   task, 118
Deep learning, 33
Deep search space, 345
Default constructor, 56
Depth-first search, 490
Desirability map, 336
Deterministic finite automatons, 428
Dialog manager, 435
DialogTree component, 494–495
Differential slowing strategy, 335, 335f
   forcing affordances and, 335
   implementation, 336–337
   spatial symmetry and, 333–337, 333f
Dijkstra search, 283–288
   canonical, 283, 285–288
   first step of, 286f
   performance, 288
   pseudocode, 287
   definition, 283–285
   JPS, 283
Directed random walk, 322–323, 323t
Distance test scoring, 314–315, 315f
djb2 hash function, 93
Domain-specific process, 234
Dot product testing techniques, 315–316
DPLL algorithms, 419
Dragon Age: Inquisition (DA:I), 371, 377
   abilities, 372
   scoring framework, 374t
Drawn to Life, 163, 165
Driver personalities, 218
Driver San Francisco, AI, 215–229
   AI, active life, 216–218
   driver personalities, 218
   vehicle paths, 217
   low-level path optimizer, 222–229
   AI path following, 227
   best path selection, 229
   dynamic potential field, 226
   search area, 223–225
   simple physics model, 226
   simulating and scoring path, 227
   static potential field, 225
   mid-level path planning, 217, 219–222, 220f
   cost generation criteria, 221
   definition, 219
   search space, 220, 220f
   speed limitations, 222, 222f
   and their costs, example, 221f
   road network, 218
   route finding, 219, 220f
   DSE (decision score evaluator), 168–170
   Dual-LCG approach, 96
Dual utility considerations, 107–111
   changing combination techniques at runtime, 110–111
   configuring, 108–110
   selecting an option, 107–108
   weight and rank, calculating, 107
Dual variable, 257
Dwarf Fortress, 433–434
Dynamic accuracy control, 395–397
dynamic delay calculation, 397
final delay calculation, 395
rules and multipliers, 395–397
distance, 395
multiplier, 395–396
velocity of player, 396
Dynamic delay calculation, 397
DynamicJoin, 419
Dynamic ordering, 348
Dynamic potential field, Driver San Francisco, 226
E
Edge-building process, 293
EdgeCostOffset variable, 293
Effective utility-based considerations, 167–177
   architecture, 168
   guiding principles, 173–177
   constructing response curves, 175–176
   interleaving decision scores, 176–177
   selecting consideration inputs, 174
   skill selection, 171–173
   charge attacks, 171–172
   side and rear attacks, 172–173
   tactical movement, 169–171
   close to melee when invisible, 170–171
   Evade Dangerous Areas, 169–170
The Elder Scrolls IV: Oblivion, 5
Engagement decision, 302–303
   definition, 302
   scripted behavior, 302
   simulations, 302–303
Enjoyment of AI Opponent, 40t
EntityExists consideration, 97–98
Entity filters, 105
Entropy learning, 346
Environment Query System (EQS), 310, 381, 383
Epsilon-greedy playout, 350
EQS (Environment Query System), 310, 381, 383
Euclidean distance, 464
Evade Dangerous Areas, 169–170
response curve, 169, 170f
runtime and cooldown, 169–170, 170f
Evaluation tree, 373, 374f
contextual nature, 374
target selector node, 374, 375f
EverQuest Landmark, 465–466
Execute ability, 376
Execution filters, 105
Execution tree, 376, 376f
Exotic elements stress tests, 81–82
Experience (XP) wells, 292, 296–297
Exploit Path Gap, 335, 335f
Extended Backus-Naur Form (EBNF), 424
natural numbers, 424t
Extrapolate random walk, 12
   problem solving using statistical methods, 12–14
Extremely parameterized behavior
trees, 382
behavior moods, 383–384
blackboard extension, 383
EQS, 383
   Vanilla UE4, 382–383
F
Façade, 6
Fast marching method, 225
Filtered meshes, 28f, 29
Final delay calculation, 395
FINAL FANTASY XV, 203, 411
   emitters
      notification, 412
      player, 413
      script, 413
      spawn, 413
logging visualization, 21–30
   architecture, 22–25
   Behemoth, 29
   data analysis, 25
   data format, 24–25, 24t
   heat map visualization, 29–30, 30f
   log aggregator, 24
   logging, 22–24
   overview, 21–22
   spatial analysis, 26–30
   statistical analysis, 25–26
   two-dimensional navigation mesh
      map, 27–29
   screenshot, 146–147, 146f, 151
FindAbilityForTarget function, 386
Finite-state machines (FSMs), 146
reasoners, 102
First-in, first-out solution, 358
First-person shooters (FPS), 127. See also 1000
NPCs at 60 FPS game
Flanking/backstab attack, 172
Floating-point value, 395
Floyd–Warshall algorithm, 281
Flyweight design pattern, 165
Formal grammars, 424, 424t
Forward differencing operator (\(\delta^+\)), 253
FPS (first-person shooters), 127
Frame playback, 70–71
controlling recorder, 71
Front-line manager, 389–390
FSMs. See Finite-state machines (FSMs)
Fully scripted game AI systems, 179
Function-based approach, 128
Function-based variables, 130
Furniture grouping, 412

G
GAIA. See Game AI Architecture (GAIA)
GAIA_EXECUTE_FACTORY_MACRO, 60
Game
function, HPS, 365
loop, 205f
phase decomposition, 365
record and compressed record, 482, 483t–484t
theory, 33
tree search technique, 363–364
Game AI Architecture (GAIA), 88, 90
data driven, 90
infrastructure, 93–96
AIDataStore base class, 94–95
AIString class, 93
factories, 93
singletons, 95–96
overview, 90–93
control flow, 90–91
implementation concepts, 91–92
sniper, 92–93, 92f
six factory system tricks. See Six factory system tricks, GAIA

Game AI Pro, 122
Game-world connections, 293
Generic clustering approach, 409
Genetic algorithms, 33
Geometric containers, 282
Global blackboard, 96, 149
Global object configurations, 60–61
Goal bounding
A* algorithm with, 279
concept, 277, 278f
constraints, 275–276
map, 275
memory, 276
memory at runtime, 276
precomputation, 276
precomputed offline, 276
definition, 275
table, 276
empirical results, 281
flexibility, 276
JPS+, applying to, 281
map, 277f
precomputation, 279–280, 280f
runtime, 278–279
Goal-Oriented Action Planner (GOAP)
reasoner, 102
Google robotics project, 117
GopherTD, 328, 329f
Gossip system, 433
Gradient
descent algorithm, 241
methods, 241, 244–246
modifying weights, 244
substepping, 244–245
time horizon, 245–246, 246f
Graph, Paragon bots, 292–294
closest graph node lookup grid, 294
edges, 293–294
nodes, 293
uses, 295–296
Graph search algorithm, 270
Greedy algorithm, 372
Grid-based approach, 313
Grid-Based Path Planning Competition (GPPC), 281
Group and timing stress tests, 82

H
Halo 2 AI, 115–116
Handcrafted rule, 464
Handcrafted schema, 443
Hard-coded logic, 169
Hash moves, 185
Heart of Thorns, 167–168, 172–173
Heat map visualization, 29–30, 30f
Hero, 382
Heroes of Might & Magic, 32
Heuristic-based strategy, 350
Heuristic function, 345
Hierarchical expansion, 347, 347f
Hierarchical navigation graph, 295
Hierarchical nested structure, 147
Hierarchical Portfolio Search (HPS) in Prismata, 361–368
AI design goals, 362
algorithm, 365–366
components, 363–364
final, 364
game function, 365
gameplay overview, 362–363
multiple difficulty settings, creating, 366–367, 367t
Negamax algorithm, 365–366
overview, 361
playing strength, evaluation, 367–368
portfolio creation, 364–365, 365t
state evaluation, 365
HighestStackAdjustedHealth, 36
HighestStackAdjustedHealthManaged, 36
HighestStackCost, 36
HighestStackCostManaged, 36
HighestStackHealth, 36, 39
HighestUnitHealth, 36
High-level arbitration, 234
HPS. See Hierarchical Portfolio Search (HPS) in Prismata
Huffman encoding, 481
Human-player-centric elements, 381
Hustling, 35
Hybrid reciprocal velocity obstacles (HRVO), 237

IfValidTarget function, 386
Independent code library, 52
Index movement, 196
Indicator function ($I_C$), 254
IndividualRevenge attacks, 36, 44t, 45
Infinite Axis Utility System, 168
Infinite-resolution influence map, 169
In-game recorder architecture, 64–65
high-level class diagram, 65, 65f
Inside-outside algorithm, 428–429
Intelligence, illusion, 3–8
selling, 5–8
AI, quality, 5
animation and dialog, 5–6
existence purpose, 7
react emotionally on demand, 8
robot, 7
strong personality, 7–8
works, 4–5
anthropomorphize, 4
expectations, 4–5
players, 4
Interaction-centric approach, 412
Interaction language model, 413–418
defered
addition, 415
deletion, 415
roles, 415–416
syntactic sugar, 416
termination type, 415
Interactive acts, 413–418
interaction language model, 413–418
STRIPS, 413, 414
tuple space, 413, 414
Interactive fiction (IF), 454
Interesting random targets, 399–401
target behind
full cover, 400, 400f
half cover, 400–401, 401f
target out in open, 401, 401f
Interrupts, 128, 137
Intuitive/modular design, AI, 362
Irrational games, 457
IsBetterScore function, 386
Item-based collaborative filtering, 463–464
Mahout (Java), 463, 466
pseudocode, 463
Item-based recommender, 466
Iterator-based expansion, 349–350
Iterator pattern, 349–350

J
Java, 465–466
JavaScript IF system, 454
JPS. See Jump Point Search (JPS)
JPS+, applying goal bounding to, 281
Jump points, 284
Jump Point Search (JPS), 272–273, 283–285
vs. Canonical Dijkstra, 285–286
and canonical orderings, 285
Jungle creeps’ camps, 292

K
Killer moves technique, 185
Kill zone position, 336
Klondike Puzzle Database, 483t–484t
Klondike Solitaire App (game), 480
  space requirements, 480–482
    bit packing, 480–481
    plausibility + and run-length encoding, 481–482
    plausibility + Morse/Huffman encoding, 481
    table lookup, 481
Knowledge implantation procedure, 444

L

Lambda parameter, 467
Lanchester’s attrition models, 303–304
  parameters, 304–306
    choosing strength value, 304–305
    learning strength value, 305–306
    logistic regression, learning with, 306
Lanchester’s Law of Ancient Warfare, 303–304
Lanchester’s Linear Law, 303
Lane(s), 381
  progress, 389
  space, paragon bots, 388–390
    front-line manager, 389–390
  lane progress, 389
Last known position (LKP), 78–79
  propagation tests, 78–79, 79f
  updating tests, 78
Last mysteries, case study, 452–454
  content exclusion, 454
  missing rooms, dealing with, 452–453
  optional tags, 453
  sparse content, 453
Latent factors, 464
LCG. See Linear congruential random number generator (LCG)
Leaflet visualization, 26–27, 28f
Leap-ahead-and-back-fill problem, 271, 272f
Legendre–Fenchel (LF) transform, 258–259
Lehmer, Derrick, 472
Levels of detail (LODs), vehicle path, 217
Light-weight finite-state machine, 159–165
  architecture, 160–161, 160f
  performance and memory improvements, 164–165
  state machines
    data-driving, 163–164
    and metastates, 163
    state transitions, 161–163
Lightweight mechanism, 494
Lindenmayer-systems (L-systems), 428
Linear congruential random number generator (LCG), 471–472
  combined, 476–477, 477f
  linear and repetitive behavior, 473f
  minimum standard and best known, 476f
  vintage, 475–476, 476f
Linear scoring, 317, 317f
Line-of-sight AI component, 54
Liquid AI system, 5
LKP. See Last known position (LKP)
Local blackboard, 149
Lockheed Martin Rotary and Mission Systems, 88
Log
  aggregator, 22, 24
  header, 24
Logging, 22–24
  elementary data chunk, 23, 23t
  library implementation pipeline, 23f
Logical/comparison operators, 457
Logical deduction, Prolog, 489–490
Logical operations, 106
LongestName, 36
Look-ahead algorithms, 302
Look-ahead search, 181–182
Loose coupling, 89
Low-level combinator approach, 234
Low-level path optimizer, 222–229
Low-LOD vehicle paths, 217
Low-potential areas, 223
Low-resolution rasterization, 269
LUMINOUS STUDIO system, 146

M
Machine learning (ML), 181
Macro programming, 59
  factory declaration, 59
Madden NFL 98, 5
Mainframe and Choba, case study, 454–455
  XML-based syntax, 454
Mana/stamina, 371
Map
  constraint, goal bounding, 275
  evaluation, 299
MapAnalyzer, 333
Markov chain Monte-Carlo simulations, 33
The Mars Game, 106
Master behavior tree, 382
Master Bot, 367
Matching (unification) process, 490
Matrix factorization problem, 463, 467
Maximum
  damage, 41
  usable range strategy, U-turn problem, 329–333
  agent, 332
  implementation, 331
  tower, 335f
Maximum transmission unit (MTU), 128
MCTS. See Monte Carlo Tree Search (MCTS)
Memory
  buffer, 69
  and fragmentation, 69, 70f
  constraint, goal bounding, 276
  management, 69–70
Mental models, Talk of the Town, 439–440
  businesses/homes, 439
  characters, 439
Meta-AI, cooperation of characters, 154–155, 155f
Metastate, 163
Microsoft’s FxCop program, 487
Mid-level path planning, 217, 219–222, 220f
  cost generation criteria, 221
  definition, 219
  search space, 220, 220f
  speed limitations, 222, 222f
  and their costs, example, 221f
Minimax search technique, 183–185, 343, 345
Minimum distance test, 314, 322
Minions, 381
Misattribution effect, 4
Mixed strategy approach, 350
MKULTRA, 488, 495
MLlib, 467
MOBA map, traditional, 389f
MOBA-type game, 381
Model-based filtering, 464
Modular AI, 87–113
  combining considerations, 105–111
  dual utility considerations, 107–111
  simple Boolean, 106–107
  conceptual abstractions and modular components, 96–105
  actions, 102–103
  considerations, 97–99
  reasoners, 100–102
  regions, 104–105
  targets, 103–105
  weight functions, 99–100
GAIA
  infrastructure, 93–96
  overview, 90–93
  overview, 87–89
  pickers, 111–113
  theoretical underpinnings, 89
Modular components, 51
Modular decomposition, 378
Module interface, 89
Modulus operation, 472, 474
MongoDB, 22
Monotonicity, 175
Monster’s attack motion, 153, 154f
Monster’s visual sensors, 155, 155f
Monte-Carlo algorithm, 37, 38
Monte Carlo Tree Search (MCTS), 185
  pitfalls and solutions. See Pitfalls and solutions, MCTS
Morse Code-like encoding, 481
Morton Code order, 266–267
  2D, 267f
Movement
  around convex hull, 199–200
  into convex hull, 200–201
  out of convex hull, 201
  systems, 232
  combining, 233
Multiplayer online battle arena (MOBA) games, 291–292
Multiple AIs and different archetypes, 397–398
  dealing with changes, 398
  most relevant agent, selection, 398
Multiplicative LCGs, 472
Multithreaded logging process, 23f
Mutation
  belief, 445
  graph, 438, 443, 443f
  rates, 445
N
Naïve approach, 425
Natural language generation (NLG)
  module, 435
Natural language understanding (NLU), 436
Natural Number rule, 424
Navigation mesh (navmesh), 232, 265–266, 275, 311–313, 312f
  generation, 29
Index

patching, 192
polygons, 311–312
Navigation stress tests, 80–81
Navmesh. See Navigation mesh (navmesh)
Negamax, 183
algorithm, 365–366
search, 184f
Neighborhood methods, 463
Nested rule, 426
Netflix, 463
Networking AI, 143–144, 143f
Neural networks (NNs), 299
Neverwinter Nights, 433–434
NFL GameDay 98, 5
Node-based graph system, 147
NodeJS (web server), 22
Nonfiltered mesh, 28, 28f
Nonplayer character (NPC), 50–51, 53, 88, 90, 94–95
Nontraffic vehicles, 216
Normalization process, 168
Notification emitter, 412
NPC. See Nonplayer character (NPC)
NPC component, 492, 494

O
Objective Graph, 299
Object-oriented bounding boxes (OOBB), 194
Obstacle avoidance, 192, 198
movement
around convex hull, 199–200
into convex hull, 200–201
out of convex hull, 201
obstacle detection, 198–199
Occam’s Razor, 117
Offensive personality, 430
Offset value, 293
Oncoming lanes, 218
One-hit kill syndrome, 394
One machine learning system, 123
One-step influence map, Paragon bots, 386–388
influence sources, 387–388
information use, 388
One-to-many mapping, 313
One-to-one mapping, 312
Ontological structure, character, 438, 438f
Open-loop action performance, 408–409
Open-source tools, 465
Optimal reciprocal collision avoidance
(ORCA), 237, 242–243
agents, 243f
constraint relaxation, 243
RVO, 237–248
Optimization algorithm, 255–262, 257f
Chambolle–Pock Primal-Dual algorithm,
261–262, 262f
LF transform, 258–259
proximity operator, 259–260
ORCA. See Optimal reciprocal collision avoidance (ORCA)
Override, 151
Overshoot definition, 210f

P
Paragon bots, 291, 381–390
ability picker and bot ability usage markup,
384–386
ability cached data, 385
ability picker, 385–386
debugging, 386
target’s spatial density, 386
AI Commander, 297–298
agent scoring, 298
humans, 298–299
opportunistic, 299
definition, 381
definition, 381
definition, 381
definition, 381
definition, 381
every presence, 294–295
every presence, 294–295
every presence, 294–295
every presence, 294–295
every presence, 294–295
behavior trees, 382
behavior trees, 382
behavior trees, 382
behavior trees, 382
behavior trees, 382
EQS, 383
Vanilla UE4, 382–383
graph, 292–294
closest graph node lookup grid, 294
definition, 294–295
definition, 294–295
definition, 294–295
definition, 294–295
definition, 294–295
definition, 294–295
edges, 293–294
edges, 293–294
edges, 293–294
edges, 293–294
edges, 293–294
nodes, 293
nodes, 293
nodes, 293
uses, 295–296
uses, 295–296
lane space, 388–390
lane space, 388–390
lane space, 388–390
front-line manager, 389–390
front-line manager, 389–390
lane progress, 389
lane progress, 389
map evaluation, 299
map evaluation, 299
objectives, 296
objectives, 296
objectives, 296
one-step influence map, 386–388
one-step influence map, 386–388
one-step influence map, 386–388
influence sources, 387–388
influence sources, 387–388
influence sources, 387–388
information use, 388
information use, 388
information use, 388
probabilistic “presence” propagation, 299
probabilistic “presence” propagation, 299
problem description, 292
problem description, 292
terms primer, 382
terms primer, 382
tricks, 390
tricks, 390
Parent–child links, 269
Parent–child links, 269
Parser generator, 424
Path
distance method, 312
gap affordance, 334–335, 334f
exploit, 335f
low-level path optimizer, 222–229
AI path following, 227
best path selection, 229
dynamic potential field, 226
path optimization, 228–229
search area, 223–225, 224f
simple physics model, 226
simulating and scoring path, 227
static potential field, 225
low-LOD vehicle, 217
mid-level, 217, 219–222
cost generation criteria, 221
and costs, 221f
definition, 219
search space, 220, 220f
speed limitations, 222, 222f
smoothing
Chambolle–Pock Primal-Dual Algorithm, 261–262
energy function, 253–255
vehicle, 217
Path-finding algorithm, 299
Path-planning algorithms, 232
Pause actions, 102
Pearson correlation, 464
Perceived difficulty of AI Opponents, 38t
Perceived realism of AI Opponent, 39t
Perception stress tests, AI for game, 75–76
pistol, 76
rifle, 75–76, 76f
Permitting arbitrary curves, 174
Persistency, 35
PersistentHighestStackCost, 36
PersistentHighestStackHealth, 36
Petri nets and AI arbitration, 355–360
arbiter, 359
basics, 355–357, 356f
overview, 355
places and transitions, 355–356, 356f
scenario, 358–359
Pickers, 111–113
PID controllers, 233
PingPong, 36
Pitfalls and solutions, MCTS, 343–353
algorithm, 344f
backpropagation, 351
phase, 349
expansion, 346–350
hierarchical, 347, 347f
iterator-based, 349–350
order of actions, 347–348
partial results, 349
pruning action sets, 349
search space, restructuring, 346–347, 347f
overview, 343–344
phases, 344
problem, 345–346
selection phase, 351–353
simulation, 350–351
abstract, 350–351
information, 350
Placebo effect/response, 4–5
Plain old data structure (POD), 123
Plausibility ordering, 481–482
as AI tool, 482–484
leveraging, efficient data compression, 479–484
Playback mode, 70
Player
emitter, 413
perception, 33
recognition of AI strategies, 41t
Player vs Player (PvP) games, 292
Playout parallelization, 352
Point-wise separation, 260
Polymorphic objects, 49
Polymorphism, 89
Portfolio-based search systems, 363
Positional restrictors and directors, 232
PowerCalculationFancy, 36, 37, 39, 43
PowerCalculationFancyManaged, 37
Precombat, AI for game, 74, 74f
Precomputation, goal bounding, 279–280
constraint, 276
Predicate-based system, 458
Predictability, 35
Predictive animation control, 203
accurate movement model creation, 207
deceleration, 208–209
overshoot, 209–210
rotation radius, 210–211
speed, 207–208
stopping, 208
actors, 203–204
pipeline, 212–213
runtime, 211–212
scaling up, 213
  benefits, 213–214
  content, 213
  size and playback, 213
simulations to rescue, 205
measurement output, 206–207
simulation controller, 205–206
“Primal-dual” problem, 257
Priority setting parameter, 156
Prismata, HPS, 361–368
  AI, 361–363
  gameplay overview, 362–363
  “Ranked” play mode, 367, 368f
  rules, 363
  sample portfolio, 365t
Probabilistic grammar structure, 425
Procjam, 454
Project Highrise, 403, 488
  screenshot, 404f
Prolog, 488–492
  built-in predicate, 492
  calling unity code, 491–492
  code as data and higher order
    code, 491
  database, 488
  data structures, 490–491
  logical deduction, 489–490
  querying database, 488–489
  rules, 489
Property window, 149
Propositional planner, 405–406
  bitmask, 406
Proximity operator, 259–260
Pruning action sets, 349
Pseudocode, 444
  high-level, 445
  item-based collaborative filtering, 463
  user-based collaborative filtering, 463
Python’s pandas library, 306
Q
  Quadratic Lanchester model, 303
  Quadratic relaxation, 259f
  Querying database, 488–489
R
  Radiant AI system, 5
  rand() function, 471–475
  ANSI C, 473f
    generate Boolean values, 475f
    Microsoft’s implementation, 474
    portable implementation, 474
    Visual C++ 2015, 473f
RandomBot attacks, 36
Random-feeling sequences, 424
Random walks, 11
  additive functions for player interaction, 17–18
  for dependent variable simulation, 18–19
  extrapolation problem of, 12
  solving using statistical methods, 12–14
  generation, 12
    with different probability distributions, 19–20
  manipulation and shaping with additive function, 16–17
  restriction, 14–16
  toward fixed point using interpolation, 14
Rasterization, 269
Raycast test, 323
Reaction stress tests, 76–78
  reacting to nondamage events, 77–78
  shooting at AI, 77
  spotting enemy, 76–77
Recommendation algorithms, 462–465
  collaborative filtering, 462–464
  content-based filtering, 462
  model-based filtering, 464
  selection, 465
Recommendation systems in games, 461–469
  building recommender, 465
    Apache Mahout, 465–466
    Apache Spark, 467
evaluation, 468–469
    recommenderlab (R), 467–468
  SQL, 468
deploying, 469
  overview, 461–462
  recommendation algorithms, 462–465
    collaborative filtering, 462–464
    content-based filtering, 462
    model-based filtering, 464
    selection, 465
  steam, 462

Recommender construction, 465
  Apache Mahout, 465–466
  Apache Spark, 467
evaluation, 468–469
  recommenderlab (R), 467–468
  SQL, 468

RecommenderEvaluator, 466
Recommenderlab (R) package, 467–468

Recorder assert, 71
Region factory, 51
Relative direction check, 172
Relaxed Behavior, 135
Representation methods, obstacle, 192–193
Revenge, 35
Rifle perception testing, 75–76, 76f
Ring generator, 310, 314
Road
  definition, 217
  network, 218
    spline with cross section information, 218f
Robot
  control system, 123
  operating system, 117
Role Playing Game (RPG), 21, 147
Root parallelization technique, 352
Route
  definition, 217
  finding, 219, 220f
RPG (Role Playing Game), 21, 147
Rule-based sequence, 362
Rule-based systems, 458
Run-length encoding, 481–482
Runtime’s job, 169–170, 170f
Rush, 180

RVO. See Reciprocal velocity obstacles (RVO)
RVO2, 242
S
Salience computation, 441–442
SAQS (Spatial Affordance Query System), 330, 333, 336
Scala, 467–468
SchedulableTask, 117–118
Script
  as decision trees, 180
    adding choices, 180–181
    adding search, 181–182
    look-ahead search, 182
    minimax search, 183–185
    state evaluation, 182–183
  emitter, 413
    programs, 428
Scripting engine, grammars as, 428
Scrubbing, 64
Search
  AI for game, 74, 74f
    algorithm speeds, comparison, 281t
Sensors, 155–156
Sentinel, 23–24
Sequence reasoners, 101
SequenceTask, 120–122
Service node, 383
SetVariable actions, 103
Shannon Entropy, 348
ShareBear, 36
Sicheritz, Tobias, 452
Side and rear attacks, 172–173
Simulation
  animation parameters, 153–154
  frames, recording of, 67–70, 67f
    memory management, 69–70, 70f
    recorder manager, 68–69, 68f
  loop, 206f
  MCTS, 350–351
    abstract, 350–351
    information, 350
Sine-ranked dot product test, 323
Sine-scored distance test, 323
Sine scoring, 317, 317f
  techniques, 318
Single path overwatch position, 336
Single thread environment, 352
Singletons, 95–96
AIBlackboard_Global, 96
AIManager, 95
AIOutputManager, 96
AIRandomManager, 96
AISpecificationManager and
AIGlobalManager, 95–96
AITimeManager, 96
Single vs. multiple test subjects, 314
Sink transitions, 357
Six Ages, storytelling game, 457
Six factory system tricks, GAIA, 49–62
design requirements, 52–61
  consistent object construction and
  initialization, 54–56
data format abstraction, 52–53
encapsulating initialization inputs, 53–54
global object configurations, 60–61
injecting external code into AI, 56–57
standardize factory definitions,
templates and macros, 57–60
overview, 49–52
  conceptual abstractions and modular
  components, 51
  extensibility, 50, 52
  factory pattern, 50–51
  reuse, 50, 52
Skyscraper construction management, 403
Slowing
towers, 336
zone position, 336
Smart locations, 412, 413f
Smooth paths, optimization, 249–262
  algorithm, 255–262
  energy function, 253–255
  overview, 249–253, 251f
Sniper rifle’s laser, 82
Soft-bounded extrapolation equation, 19
Solitaire Free, 480
Solver code, 332
Source transitions, 357
Space-time and attack window separation
strategy, 337–341
  attack windows, 338–339
  decay, 340–341
  space-time, metric, 337–338
  U-turn problem, 339–340
Space-time equivalence, 330
Sparse Voxel Octrees (SVO), 3D flight
  navigation, 265–273, 267f
  alternative techniques, 265–266
  complications and optimizations, 271–273
creation, 268–270, 269f
high-level illustration, 267, 268f
neighbor links, 269f
overview, 265
pathfinding, 270–271, 271f
Spatial Affordance Query System (SAQS), 330,
333, 336
Spatial analysis
  heat map visualization, 29–30, 30f
two-dimensional navigation mesh
  map, 27–29
Spatial attack window, 338
Spatial geometry, 334
Spatial queries, effective auto-generated,
309–324
  behaviors, 322–324
  boids, 324, 324t
  camera, 323, 323t
  directed random walk, 322–323, 323t
  orbit, 323–324, 324t
  components, 310
  continuous vs. sequential updates,
  319–320, 319f
  query-based behaviors, 319, 320f
  querying vs. destination validation,
  319–320, 320f
  overview, 309–311, 311f
  reducing query failure, 320–322, 321f
    fallback queries, 322
    permissiveness, 321, 321f
    preserving quality, 322
    robustness, 321, 321f
  sample points generation, 311–314
    navigation mesh, 311–313, 312f
    structure, 313–314, 313f
  testing techniques and test subjects, 314–316
    distance test scoring, 314–315, 315f
    dot product, 315–316
    single vs. multiple, 314
    subject floor position, 316
  test scoring functions, 317–318
    sine scoring techniques, 318
Spatial symmetry and differential slowing
  strategy, 333–337, 333f
differential slowing strategy
  forcing affordances and, 335, 335f
  implementation, 336–337
problem, 333
temporal asymmetry and path gap
  affordance, 334–335, 334f
Spatio-temporal reasoning, time role in, 327–341
defend village, 328–329, 329f
quantifying space-time and attack window separation strategy, 337–341
spatial symmetry and differential slowing strategy, 333–337
U-Turns and maximum usable range strategy, 329–333
Spawn emitter, 413
Splines, 217
SQL, 468
Square law, 304
root scoring, 317, 317f
scoring, 317, 317f
SQUARE ENIX game engine, 148
Stack Cost, 43
Stanford Research Institute Problem Solver (STRIPS), 413–414
StarCraft, 288
StarCraft Broodwar, 304
StarCraft: Brood War, 186
State evaluation, 182–183
transitions, 161–163
StateMachine class, 160–161
data-driving, 163–164
and metastates, hierarchies, 163
State-of-the-art search algorithms, 361
Static checker, 492–493
Static checking, 487–488
Static potential field, 225, 225f
Statistical analysis, 25–26
histogram, 25, 26f, 27f
Stealth mechanics, 78
Steam recommendation system, 462
Step function, 17
number, 336
Stereotyped scripts, 407
Stochastic grammars, 423–430
analogs to behavior trees, 427–428
data structure, 426–427
formal grammars, 424, 424t
overview, 423–424
scripting engine, 428
sequences generation, 425–426, 425t, 426t
streaming sequences, 427
tuning, 428–429
utility theory, 429–430, 430t
Stochastic optimization methods, 245
Straight-line shortest path, 249, 251f
Strategic layer, 291
Streaming sequences, 427
Strings, 424
STRIPS (Stanford Research Institute Problem Solver), 413–414
Sum-differencing operator (δ), 253
Summary information, 34
SVO. See Sparse Voxel Octrees (SVO), 3D flight navigation
Symmetric game playout, 365
Synchronizing external AI tools, 71
Tactical decomposition, 364
Tactical Environment Awareness System, 79
Tactical layer, 291
Tactical movement, 169–171
close to melee, 170–171
evade dangerous areas, 169–170
Tactical Point System (TPS), 309
Tag-based content selection, procedural level and story generation, 451–458
advantage, 456
decks, 455–457
extensions and approaches, 457–458
bioshock infinite’s gameplay pattern matcher, 458
valve’s dynamic dialog system, 458
irrational games, 457
last mysteries, 452–454
excluding content, 454
missing rooms, dealing with, 452–453
optional tags, 453
sparse content, 453
mainframe and choba, 454–455
overview, 451
simplicity, advantage, 452
Six Ages, 457
Talk of the Town, 434–437
character knowledge phenomena simulation, 433–447
belief revision, 443–444
core procedure, 445
evidence, 440–441
fallibility modeling, 443, 443f
knowledge implantation, 444

512
knowledge propagation, 442–443, 442f
mental models, 439–440
ontological structure, 438, 438f
overview, 433–434, 437
requirement, 437–438
salience computation, 441–442
statistics, 445
tunable parameters, 445
gameplay, 435–437
story, 434
world generation, 434–435
Tanimoto coefficient, 464, 468
Target
density, 388
destructible objects, 401–402, 401f
selector node, 374–375, 375f
spatial density, 386
stress tests, 78–79
  LKP propagation tests, 78–79, 79f
  LKP-updating tests, 78
TeamDamageMonteCarlo, 36, 43
TeamDamageMonteCarloManaged, 36, 45
TeamRevenge, 36
Technique complexity, 34–35
Temporal asymmetry, 334–335, 334f
Temporal attack window, 338
Test scoring functions, 317–318, 317f
  sine scoring techniques, 318
Three-layered character system, 156, 156f
Three-tier path optimization approach, 216
Tic Tac Toe game, 351
Time horizon, 245–246, 246f
TiXmlElement, 52–54
Token systems, 394–395
Tower
defense game, 328
  management game, 496
Tozour, Paul, 458
TrainImplicit method, 467
Transitions, 355
  multiple tokens, 357, 357f
  places and, 355–356, 356f
  source and sink, 357
Transposition table, 184
Tray, 149
Tree-type structure, 426
Triangulation from convex hull data, 201
Triple-A game AIs, 34
Tuple space, 413, 414
Turn-based strategy game, 31, 32–33
AI design goals, 33
AI opponents, 36
AI traits, 34
  available information, 34
  hustling, 35
  persistency, 35
  predictability, 35
  revenge, 35
  technique complexity, 34–35
  explanations, 41–46
  numbers, 37–41
  test, 36–37
Turn Behavior, 135
Two-dimensional navigation mesh map, 27–29
  connectivity mesh, 28–29, 28f
  difference image, 28f, 29
  filtered meshes, 28f, 29
  nonfiltered mesh, 28, 28f
U
UAlbertaBot, 307
UBCF. See User-based collaborative filtering (UBCF)
UE4. See Unreal Engine 4 (UE4)
Unification, 419
Unified theory of locomotion, 231–234
  cascading value dependencies, 233
  combining movement systems, 233
  overview, 231–232
  positional restrictors and directors, 232
  value arbitration and combination,
  233–234
  velocity restrictors and directors, 232–233
Uninterpreted strings, 494–495
  localization, 494
UnityProlog, 491–492
Unreachability stress tests, 79–80
Unreal Engine 4 (UE4), 381
  BB, 382–383
  BTs, 382
Upper Confidence Bounds Applied to Trees (UCT), 344, 351
User-based collaborative filtering (UBCF), 463–465, 468
  Mahout (Java), 466
  MLlib (Scala), 467
  pseudocode, 463
  recommenderlab (R), 467
  SQL, 468
Utility-based approach, 105
Utility theory, 167
  grammars, 429–430, 430t
U-turn problem, 339–340, 340t
  and maximum usable range strategy, 329–333
  problem, 329–330
  space-time equivalence, 330
  strategies and affordances, 330–333
two lines of agents, 340f

V
Value arbitration and combination, 233–234
Value-based variables, 129–130, 130t
Valve's dynamic dialog system, 458
Vanilla UE4, 382–383
  AI systems, 381
Vector space, 251–252
dot product, 252
Vehicle
dynamics calculation, 226
paths, 217
Vehicle-handling code, 217
Velocity
candidate, 246
prediction, 201–202
restrictors and directors, 232–233
Velocity obstacle (VO) methods, 237, 243
  agents, 238
  original, 239f
  reciprocal, 239f
Video game, 355

  Vintage random number generators, 471–477
    LCGs
      combined, 476–477, 477f
      vintage, 475–476, 476f
      overview, 471
      rand(), 472–475, 473f, 475f
Visible error rate, 33
Vision model, 75
  upgraded, 76f
Visual C++ 2015's, 473, 473f
Visual node debugger, 153, 153f
Voxel grid, 271

W
Wave-front exploration pattern, 272
Waypoint
graph, 266
weight curve, 255, 255f
Weapon-Handling stress tests, 81
Web-based IF game, 454
Web browser visualization, data flow, 22, 22f
Weight functions, 99–100
Wide search space, 345

X
Xenonauts 2, 344–346, 348–349, 351

Z
Zero-sum games, 183