

Index

• Symbols and Numerics •

{ } (braces) for ending functions, 41
: (colon) for ending `case` statements, 83
"" (double quotes) for outputting text, 41
= (equal) assignment operator, 67
== (equal) comparison operator, 77
! (exclamation point) literal value, 68
// (forward slashes) for indicating comments, 38
> (greater than) comparison operator, 77
>= (greater than or equal to) comparison operator, 77
< (less than) comparison operator, 77
<= (less than or equal to) comparison operator, 77
|| (logical or) character, 227
!= (not equal) comparison operator, 77
() (parentheses) for indicating conditions, 76
2-D games, 114
3-D ball, 115–116
3-D buttons, 30

• A •

accelerating objects, 344
acceleration (defined), 281, 283
access methods, 339
ActionScript
 animation, 12
 arrays, 320–321
 comparisons, 77
 conditions, 76–77
 constants, 204
 enumerations, 204
 features, 11–12
 object-oriented programming (OOP), 120
 properties, 125
 trigonometry functions, 259

`addBetter` program
 `parseInt` method, 88
 user interface, 87–90
adding
 buttons, 29–31
 elements to arrays, 321
 keyframes, 47–48, 200
 sound effects, 190–191
 states, 167–172
 values to variables, 102
 vertices (shapes), 197
adjacent side of a triangle, 251
adventure games
 buttons, 56–58
 creating, 53–59
 diagram nodes, 56–57
 diagramming, 54–55
 Green Grass game, 46–53
 introduction, 55
AI (Artificial Intelligence), 156
algorithms, 73
American Standard Code for Information Interchange (ASCII), 182
angles
 calculating, 274–275
 cosine, 252
 defined, 250
 degrees, 257
 `dx` property, 249
 `dy` property, 249
 `gravityTrace` program, 248
 measuring, 250
 radians, 256
 right triangles, 250–252
 sin, 252
 solving, 255
 tangent, 252
 vector projection, 248
 vectors, 250
animated sprites
 car sprite, 211–212
 creating, 198–201
 layers, 199
 Monster Traffic game, 194

- animated sprites (*continued*)
 - moving, 201–204, 211
 - orientation of, 200
 - properties, 204–206
 - shapes, 193–198
 - turning, 206–208
 - user-controlled, 212–217
 - animation
 - ActionScript, 12
 - Green Grass* game, 51–52
 - keyframes, 200
 - layers, 12, 23
 - Monster Traffic* game, 242–243
 - motion tween tool, 12
 - shape tween tool, 12
 - sound effects, 184–185
 - arcade font, 142
 - arcade games
 - sound effects, 183
 - vehicles, 279
 - arrayDemo program, 320
 - arrays
 - ActionScript, 320–321
 - adding elements, 321
 - code example, 320
 - counting elements, 322
 - creating, 320–321
 - for loop, 321
 - output, 322
 - retrieving elements, 321
 - arrow keys, 181
 - artStupid program, 158–160
 - Artificial Intelligence (AI), 156
 - ASCII (American Standard Code for Information Interchange), 182
 - assignment operators, 67
 - Asteroids*
 - creating, 352–353
 - gravity well, 353
 - power-ups, 353
 - shields, 353
 - sprites, 114
 - warping, 353
 - attachMovie() method, 324–325
 - Audacity open source audio editor, 186
 - audio effects
 - adding, 190–191
 - animation, 184–185
 - arcade games, 183
 - Audacity open source audio editor, 186
 - audio compression, 186–187
 - combining, 192
 - creating, 186
 - game programming, 185
 - headphones, 184, 192
 - legal issues, 185–186
 - Macromedia Flash, 183–184
 - Monster Traffic* game, 178, 234–236
 - audio files
 - audio compression, 186–187, 190
 - bit depth, 187
 - importing, 187–189, 191–192
 - MP3 format, 185
 - sampling rate, 187
 - WAV format, 185
- **B** •
- Ball.as class definition, 336–339
 - ballistics, 266
 - ballMove program, 123–127
 - balloon program
 - gravity, 284
 - helium, 284–285
 - objective of, 283
 - tied-down balloon, 286–287
 - wind, 285–286
 - binaryDice program, 81–84
 - bit depth (audio files), 187
 - board games, 373
 - boats, 312–313
 - boundary checking
 - bouncing off the walls, 134–137
 - combining behaviors, 136–137
 - ignoring borders, 136
 - stopping at the border, 132–134
 - wrapping around the screen, 128–132
 - braces ({}) for ending functions, 41
 - break statement, 83–84
 - Breakout* game
 - creating, 365–366
 - enhancements, 366
 - power-ups, 366
 - btnCalc() method, 274
 - buffer, 179
 - bugs. *See also* errors
 - incorrect names, 107–108
 - op variable, 109

- syntax errors, 106–107
- troubleshooting, 109
- building
 - adventure games, 53–59
 - animated sprites, 198–201
 - arrays, 320–321
 - Asteroids* game, 352–353
 - Breakout* game, 365–366
 - buttons, 31–32
 - class definition, 333–334
 - Egg Cannon*, 355–357
 - Green Grass* game, 48–51
 - keyframes, 200
 - Lunar Lander*, 353–355
 - movie clips, 324–325
 - multiple copies of sprites, 328–331
 - objects, 332–334
 - Orbit Matcher*, 369–370
 - platform scrolling games, 362–364
 - Pong*, 143–144
 - projects, 19–20
 - shapes, 193–195
 - sound effects, 186
 - Space Invaders*, 367–368
 - spaceships, 303
 - sprites, 113, 323–326
 - tile-based world games, 371–373
 - Zelda*, 357–361
- bullet trajectories, 266
- `bullet.dir` property, 261
- `bullet.speed` property, 261
- buttons
 - adding, 29–31
 - adventure games, 56–58
 - code, 37–41
 - creating, 31–32
 - events, 40–41
 - labeling, 36–37
 - math quiz game, 96–98
 - mouseover effects, 34–36
 - naming, 32, 36–37, 57
 - Onionskin buttons, 57
 - `onRelease()` event, 40
 - states, 34–36
 - symbols, 31–33
 - text, 39
 - 3-D buttons, 30

• C •

- C language, 12
- calculating
 - angles, 274–275
 - distance, 344–345
 - gravity, 347
 - vectors, 272
- calculator, 87–90
- camel-case, 24
- Cannon* game
 - ballistics, 266
 - `bullet.dir` property, 261
 - `bullet.speed` property, 261
 - charging, 263
 - `checkKeys()` method, 262–263
 - controls, 263
 - direction, 260
 - direction control, 263
 - firing bullets, 263
 - gravity simulation, 268–269
 - `gun.charge` property, 261
 - `gun.dir` property, 261, 263
 - keyboard input, 262–263
 - `move()` method, 262–264
 - moving bullets, 263
 - turn method, 264–265
 - turning bullets, 264
 - visual elements, 260–261
- cars
 - `carMove` program, 201–204
 - `carParam` program, 295–300
 - `carVector` program, 288–291
 - `carVectorOOP` program, 291–294
 - keyboard input, 290–291
 - laws of motion, 279–282
 - motion, 349
 - traction property, 313–315
 - turning, 291
- case statement, 83–84
- `ceil` function, 72
- character sprites
 - animation, 193–201
 - boundary checking, 128–137
 - code, 117–119
 - creating, 113, 323–326
 - creating many copies, 328–331

- character sprites (*continued*)
 - custom sprites, 334–340
 - defined, 113
 - destroying, 323, 326–328
 - examples, 114
 - killSprite program, 326–327
 - lotsOfSprites program, 328–331
 - makeASprite program, 323–326
 - Monster Traffic* game, 201
 - motion, 349
 - MovieClip object, 114, 116–117
 - moving, 201–204, 211
 - naming, 117
 - properties, 204–206
 - registration point, 117, 345
 - suicidal sprites, 326–327
 - 3-D ball, 115–116
 - turning, 206–208
 - user-controlled, 212–217
- checkBoundaries() method, 129–136
- checkCollisions method, 231–232
- checkKeys method
 - Cannon* game, 215, 262–263
 - carParam program, 298
 - carVector program, 290–291
 - how it works, 214–215
 - Monster Traffic* game, 216–217
 - space program, 304
- Civilization III* game, 114
- class definition
 - Ball.as, 336–339
 - characteristics, 332
 - creating, 333–334
- class keyword, 332
- classes
 - constructors, 338
 - methods, 332
 - naming, 332
 - properties, 332
- clear() method, 271
- cloning movie clips, 221
- code
 - buttons, 37–41
 - comments, 38
 - game programming, 15–16
 - math quiz game, 95–106
 - pseudocode, 39
 - sprites, 117–119
 - style conventions, 41
- colon (:) for ending case statements, 83
- combining
 - boundary checking behaviors, 136–137
 - sound effects, 192
 - vectors, 348–349
- comments, 38
- comparison operators, 77
- competition, 11
- computer game starters
 - Asteroids*, 352–353
 - Breakout*, 365–366
 - Egg Cannon*, 355–357
 - Lunar Lander*, 353–355
 - Orbit Matcher*, 369–371
 - Pac-Man*, 45
 - platform scrolling games, 362–364
 - Space Invaders*, 367–369
 - tile-based world games, 371–373
 - Whack-an-Author*, 373–376
 - Zelda*, 357–362
- computer opponents
 - artStupid program, 158–160
 - Pong*, 156–160
- concatenation, 68–69
- conditions
 - ActionScript, 76–77
 - break statement, 83–84
 - case statement, 83–84
 - default clause, 83–84
 - else clause, 79–81
 - false conditions, 78–79
 - if statement, 75–78
 - parentheses, 76
 - switch statement, 82–84
 - true conditions, 78
- constants
 - ActionScript, 204
 - Math.PI, 259
- constructors, 338
- controls
 - Cannon* game, 263
 - Monster Traffic* game, 178
- converting
 - degrees to radians, 257
 - strings to integers, 89
 - radians to degrees, 257
- cosine of an angle, 252
- counting elements in arrays, 322
- counting variable for loops, 319

creating
 adventure games, 53–59
 animated sprites, 198–201
 arrays, 320–321
Asteroids, 352–353
Breakout, 365–366
 buttons, 31–32
 class definition, 333–334
Egg Cannon, 355–357
Green Grass, 48–51
 keyframes, 200
Lunar Lander, 353–355
 movie clips, 324–325
 multiple copies of sprites, 328–331
 objects, 332–334
Orbit Matcher, 369–370
 platform scrolling games, 362–364
Pong.fla game, 143–144
 projects, 19–20
 shapes, 193–195
 sound effects, 186
Space Invaders, 367–368
 spaceships, 303
 sprites, 113, 323–326
 tile-based world games, 371–373
Zelda, 357–361
 cursors, 137–140
 custom sprites, 334–340
 customCursor program
 code, 138–140
 replacing the mouse pointer, 137–138
 shapes, 137–140
 testing, 140

• D •

deceleration (defined), 283
 default clause, 83–84
 definitions of objects, 33–34
 depth parameter, 325
 degrees (angles), 257
 degrees-to-radians conversion, 257
 design time, 325
 destroying sprites, 323, 326–328
 diagram nodes, 56–57
 diagramming adventure games, 54–55
Diablo II, 114
 dir property, 210

direction (of movement)
 Cannon game, 260
 dir property, 210
 direction constants, 203–204
 dx property values, 210
 dy property values, 210
 rotation property, 208
 spaceships, 302
 switch statement, 208–209
 Director (Macromedia), 12
 DirectX graphics engine, 16
 distance
 calculating, 344–345
 registration points, 345
 solving, 255–256
Donkey Kong, 362
 Don't Click program, 29–32
 double quotes (“”) for outputting text, 41
 down state (of button), 36
 drawing
 movie clips, 270
 paths, 271–272
 vectors, 249–250
 dx property
 angles, 249
 defined, 123
 directions, 203, 210
 gravity, 267
 speed, 249
 values, 124, 252, 254
 vectors, 249
 dxdyToVec program, 272–274
 dy property
 angles, 249
 defined, 123
 directions, 203, 210
 gravity, 267
 speed, 249
 values, 124, 252–254
 vectors, 249
 dynamic text, 63

• E •

e (exponential notation), 258
 Easter eggs, 102
 ECMAScript, 12
 editing diagram nodes, 56–57

- Egg Cannon* game
 - creating, 355–357
 - enhancements, 357
 - else clause, 79–81
 - encapsulation, 122
 - ending condition for loops, 319
 - enumerations, 204
 - equal (=) assignment operator, 67
 - equal (==) comparison operator, 77
 - errors
 - incorrect names, 107–108
 - op variable, 109
 - Statement must appear within onClip event handler, 108
 - syntax errors, 106–107
 - troubleshooting, 109
 - events
 - buttons, 40–41
 - defined, 121
 - keyboard input, 183
 - onEnterFrame() event, 119, 121, 125–126
 - onRelease() event, 40, 121
 - exclamation point (!) literal value, 68
 - exponential notation (e), 258
 - extends keyword, 338
- F ●**
- false conditions, 78–79
 - figures. *See* sprites
 - fills
 - radial gradient, 115
 - shapes, 195
 - FLA files, 27
 - Flash
 - ActionScript, 11–12
 - audio compression, 187, 190
 - buttons, 29–41
 - FLA files, 27
 - graphic symbols, 33
 - Hello World! program, 19–20, 23–27
 - HTML files, 28
 - keyframes, 23
 - layers, 23
 - Library, 32–34
 - movie clips, 33
 - multimedia support, 11
 - panel stack, 24
 - project templates, 20–21
 - projects, 19–20
 - Properties Inspector, 22
 - publishing Web page files, 27
 - sound effects, 183–184
 - Stage, 21–22
 - SWF files, 28
 - symbols, 31–33
 - Timeline, 23
 - Tools Panel, 22–23
 - Web output, 12
 - Flash game starters
 - Asteroids*, 352–353
 - Breakout*, 365–366
 - Egg Cannon*, 355–357
 - Lunar Lander*, 353–355
 - Orbit Matcher*, 369–371
 - Pac-Man*, 45
 - platform scrolling games, 362–364
 - Space Invaders*, 367–369
 - tile-based world games, 371–373
 - Whack-an-Author, 373–376
 - Zelda*, 357–362
 - floating point real numbers, 71
 - floor() function, 72
 - followMouse program, 272–278
 - for loop, 318–321
 - force (defined), 281
 - forward slashes (/) for indicating comments, 38
 - frames
 - adding, 47–48, 200
 - creating, 200
 - defined, 23
 - sharing variables, 98
 - states (of game), 47–48
 - functions
 - attachMovie, 324–325
 - braces ({ }), 41
 - btnCalc, 274
 - checkBoundaries, 129–136
 - checkCollisions, 231–232
 - checkKeys, 214–217, 262–263, 290–291, 298, 304
 - classes, 332
 - clear, 271
 - defined, 121
 - events, 121
 - getDy, 153–155
 - gravitate, 308–309

hide, 138
 hitTest, 150
 init, 148
 Key.getAsci, 182–183
 Key.isDown, 181–183
 LineStyle, 270
 lineTo, 271
 Math.ceil(), 72
 Math.floor(), 72
 Math.pow, 275
 Math.random(), 71–72
 Math.round(), 72
 methods, 121
 move, 126–127, 202–203, 211, 262–264
 moveTo, 271
 objects, 339
 parseInt(), 88–90
 removeMovieClip, 327, 340
 resetFlame, 232–234
 Sound.setPan, 192
 Sound.setVolume, 192
 Sound.start, 189
 Sound.stop, 192
 String.fromCharCode, 182
 trace, 40–41
 turn, 202–203, 206–207, 210, 265, 291, 299,
 304–305

• G •

game programming

benefits of, 9
 coding, 15–16
 competition, 11
 goal of game development, 13–14
 imagination, 10
 interactivity, 11
 Macromedia Flash, 11–12
 objects, 11
 process, 16–18
 skills required, 9–11
 sound effects, 185
 themes, 14–15

game starters

Asteroids, 352–353
Breakout, 365–366
Egg Cannon, 355–357
Lunar Lander, 353–355

Orbit Matcher, 369–371

Pac-Man, 45

platform scrolling games, 362–364

Space Invaders, 367–369

tile-based world games, 371–373

Whack-an-Author, 373–376

Zelda, 357–362

generating

random integers, 348

random numbers, 69–72

vectors, 346

getDy function, 153–155

gotSix program, 73–81

gradient fills

radial gradient, 115

shapes, 195

graphic symbols, 33

graphics engines, 16

gravitate method, 308–309

gravity

balloon program, 284

calculating, 347

characteristics, 267

dx property, 267

dy property, 267

Earth's gravitational constant, 269

gravityTrace program, 248, 267

Jupiter's gravitational constant, 270

Law of Universal Gravitation, 347

moon's gravitational constant, 270

Newton's Second Law, 347–348

projectiles, 267

simulating, 268–269

spaceships, 306–310

gravityTrace program, 248

greater than (>) comparison operator, 77

greater than or equal to (>=) comparison operator, 77

Green Grass game

animation loop, 51–52

creating, 48–51

keyframes, 47–48

states, 46–47

swapping states, 52–53

greeting program, 61–63

gun.charge property, 261

gun.dir property, 261, 263

• H •

Hamsterdance Web site, 16
 hardware polling, 179–182
 headphones, 184, 192
 height property, 131–132
 Hello World! program, 19–20, 23–27
 hexadecimal colors, 270
 hide method, 138
 hiding

- mouse pointer, 138
- vertices (shapes), 198

 hitTest() method, 150
 HTML files, 28
 hypotenuse of a triangle, 251

• I •

if statement, 75–78
 imagination, 10
 importing audio files

- MP3 files, 191–192
- step-by-step directions, 187–189

 incorrect names, 107–108
 inheritance, 122, 338
 init() function, 148
 input from keyboard

- arrow keys, 181
- buffer, 179
- Cannon* game, 262–263
- cars, 290–291
- checkKeys method, 214–215, 217
- code demo, 179–180
- event handlers, 183
- hardware polling, 179–182
- Monster Traffic* game, 178–180, 213–214
- planning, 213–214
- W, A, S, and D keys (alternative arrows), 181

 input text, 63–66
 instances of objects, 33–34, 39
 integers, 72
 interactivity, 11

• J •

JavaScript, 11

• K •

Key object, 180–181
 keyboard input

- arrow keys, 181
- buffer, 179
- Cannon* game, 262–263
- cars, 290–291
- checkKeys method, 214–215, 217
- code demo, 179–180
- event handlers, 183
- hardware polling, 179–182
- Monster Traffic* game, 178–180, 213–214
- planning, 213–214
- W, A, S, and D keys (alternative arrows), 181

 keyframes

- adding, 47–48, 200
- creating, 200
- defined, 23
- sharing variables, 98
- states (of game), 47–48

 Key.getAscii() method, 182–183
 Key.isDown() method, 181–183
 keywords

- class, 332
- extends, 338
- new, 320
- this, 294
- var, 338

 killSprite program, 326–327

• L •

labeling buttons, 36–37
 Law of Universal Gravitation, 347
 laws of motion

- Newton's First Law, 279–280
- Newton's Second Law, 280–282, 347–348
- Newton's Third Law, 282

 layers

- animated sprites, 199
- defined, 12, 23

 legal issues regarding sound effects, 185–186
Legend of Zelda

- creating, 357–361
- enhancements, 361–362

length of a vector, 250, 275
 less than (<) comparison operator, 77
 less than or equal to (<=) comparison operator, 77
 level editor, 373
 Library (Macromedia Flash), 32–34
 linear regression analysis, 155
`lineStyle()` method, 270
`lineStyle` property, 272
`lineTo()` method, 271
 literal value, 68
 logical or (|) character, 227
`loopDemo` program, 318
 loops
 code demo, 318
 contents, 319
 counting variable, 319
 defined, 317
 ending condition, 319
 for loop, 318–321
 while loop, 320
 losing state, 165–166
`lotsOfSprites` program, 328–331
Lunar Lander
 creating, 353–355
 enhancements, 355
 spaceship, 354–355

• M •

Macromedia Flash
 ActionScript, 11–12
 audio compression, 187, 190
 buttons, 29–41
 FLA files, 27
 graphic symbols, 33
 Hello World! program, 19–20, 23–27
 HTML files, 28
 keyframes, 23
 layers, 23
 Library, 32–34
 movie clips, 33
 multimedia support, 11
 panel stack, 24
 project templates, 20–21
 projects, 19–20
 Properties Inspector, 22
 publishing Web page files, 27
 sound effects, 183–184

Stage, 21–22
 SWF files, 28
 symbols, 31–33
 Timeline, 23
 Tools Panel, 22–23
 Web output, 12
 Macromedia Flash game starters
 Asteroids, 352–353
 Breakout, 365–366
 Egg Cannon, 355–357
 Lunar Lander, 353–355
 Orbit Matcher, 369–371
 Pac-Man, 45
 platform scrolling games, 362–364
 Space Invaders, 367–369
 tile-based world games, 371–373
 Whack-an-Author, 373–376
 Zelda, 357–362
 Macromedia Director, 21
 magnitude of vectors, 250
`makeASprite` program, 323–326
Mario Brothers, 362
 mass (defined), 281
 Math object, 71
 math quiz game
 buttons, 96–98
 calculator, 87–90
 choose page, 91–92, 96
 code, 95–106
 how it works, 85–87
 report page, 94–95, 103–106
 solve page, 92–94, 99–102
 visual design, 90–95
`Math.ceil()` function, 72
`Math.floor()` function, 72
`Math.PI` constant, 259
`Math.pow()` method, 275
`Math.random()` function, 71–72
`Math.round()` function, 72
 measuring angles, 250
 methods
 access methods, 339
 `attachMovie`, 324–325
 `btnCalc`, 274
 `checkBoundaries`, 129–136
 `checkCollisions`, 231–232
 `checkKeys`, 214–217, 262–263, 290–291, 298, 304
 classes, 332

methods (continued)

- clear, 271
- defined, 121
- gravitate, 308–309
- hide, 138
- hitTest, 150
- init, 148
- Key.getAscii, 182–183
- Key.isDown, 181–183
- lineStyle, 270
- lineTo, 271
- Math.ceil(), 72
- Math.floor(), 72
- Math.pow, 275
- Math.random(), 71–72
- Math.round(), 72
- move, 126–127, 202–203, 211, 262–264
- moveTo, 271
- objects, 339
- parseInt(), 88–90
- removeMovieClip, 327, 340
- resetFlame, 232–234
- Sound.setPan, 192
- Sound.setVolume, 192
- Sound.start, 189
- Sound.stop, 192
- String.fromCharCode, 182
- trace, 40–41
- turn, 202–203, 206–207, 210, 265, 291, 299, 304–305

modifying shapes, 195–197

Monster Traffic game

- animated sprites, 194
- animation, 242–243
- car sprite, 201, 211–212
- carMove program, 201–204
- checkCollisions method, 231–232
- collision detection, 229–230
- collision planning, 229
- collision testing, 228
- controls, 178
- direction constants, 203–204
- enemies, 221–222
- fireball sprite, 201
- instruction screen, 175–176
- introduction screen, 175–176, 236–239
- keyboard input, 178–180, 213–214
- missile firing, 223–228
- monster sprite, 201

- object of, 176–177

- opponents, 220

- resetFlame method, 232–234

- scorekeeping system, 177, 240–242

- sound effects, 178, 234–236

- states, 239–240

motion

- acceleration (defined), 281, 283

- balloon program, 283–287

- carVector program, 288–291

- deceleration (defined), 283

- force (defined), 281

- mass (defined), 281

- Newton's First Law, 279–280

- Newton's Second Law, 280–282, 347–348

- Newton's Third Law, 282

- vectors, 248

- vehicle motion, 349

- velocity (defined), 283

- motion tween tool, 12

mouse pointers

- hiding, 138

- replacing, 137–140

- shapes, 137–139

- _xmouse property, 140

- _ymouse property, 140

- mouseover effects (buttons), 34–36

move method

- ball program, 126–127

- Cannon* game, 262–264

- sprites, 202–203, 211

- moveTo() method, 271

movie clips

- cloning, 221

- creating, 324–325

- defined, 33

- drawing, 270

MovieClip object

- built-in features, 53

- creating, 114, 116–117

- sprites, 114, 116–117

moving

- objects, 343–344

- spaceships, 306

- sprites, 201–204, 211

- MP3 audio format, 185

- multi-state games, 48

• N •

naming
 buttons, 32, 36–37, 57
 classes, 332
 projects, 24
 sprites, 117
NetHack, 13–14
new keyword, 320
Newton's First Law, 279–280
Newton's Second Law, 280–282, 347–348
Newton's Third Law, 282
normalized vector, 259
not equal (!=) comparison operator, 77
numbers
 floating point real numbers, 71
 integers, 72
 random integers, 348
 random numbers, 69–72
 real numbers, 71
 rounding, 72

• O •

object-oriented programming (OOP), 120
objects
 accelerating, 344
 ballMove program, 123–127
 class definition, 332–334
 creating, 332–334
 defined, 11, 120
 definitions, 33–34
 encapsulation, 122
 inheritance, 122, 338
 instances, 33–34, 39
 Key, 180–181
 mass (defined), 281
 Math, 71
 methods, 121, 339
 MovieClip, 53, 114, 116–117
 moving, 343–344
 polymorphism, 122
 properties, 120, 339
 Sound, 185
 weight (defined), 281
 z-order, 325, 327
onEnterFrame() event, 119, 121, 125–126
Onionskin buttons, 57

onRelease() event, 40, 121
OOP (object-oriented programming), 120
oopBall program, 334–336
oopManyBalls program, 340
op variable, 109
open source audio editor, 186
OpenGL graphics engine, 16
operators
 assignment, 67
 comparison, 77
opponents
 artStupid program, 158–160
 Monster Traffic game, 220
 Pong, 156–160
opposite side of a triangle, 251
Orbit Matcher
 creating, 369–370
 enhancements, 370–371
orientation of animated sprites, 200
output window, 40
outputting arrays, 322
over state (of button), 34–36

• P •

Pac-Man, 45, 114
paddleBounce program, 152–155
panel stack (Macromedia Flash), 24
parentheses () for indicating
 conditions, 76
parse to integer, 89
parseInt() function, 88–90
paths, drawing, 271–272
planning games
 Pong example, 143–144
 step-by-step directions, 145
platform scrolling games
 creating, 362–364
 enhancements, 364
 power-ups, 364
pointers
 hiding, 138
 replacing, 137–140
 shapes, 137–139
 _xmouse property, 140
 _ymouse property, 140
polymorphism, 122

Pong

- adding states, 167–172
- arcade font, 142
- bouncing ball, 147–150
- computer opponents, 156–160
- creating, 143–144
- hitTest() method, 150
- losing screen, 165–166
- original *Pong*, 141–142
- paddle-ball collisions, 150–151
- paddleBounce program, 152–155
- plan, 143–144
- player paddle, 144–147
- risk-rewards principle, 151–152
- scorekeeping systems, 161–165
- starting screen, 165
- visual design, 144
- winning screen, 165–166

position (defined), 283

power-ups

- Asteroids* game, 353
- Breakout*, 366
- platform scrolling games, 364

process of game programming, 16–18

programming. *See* game programming

programs

- addBetter, 88–90
- arrayDemo, 320
- ballMove, 123–127
- balloon, 283–287
- buttons, 29–32
- carMove, 201–204
- carParam, 295–300
- carVector, 288–291
- carVectorOOP, 291–294
- customCursor, 137–140
- Don't Click, 29–32
- dxdyToVec, 272–274
- followMouse, 272–278
- gravityTrace, 248, 267
- greeting, 61–63
- Hello World!, 19–20, 23–27
- killSprite, 326–327
- loopDemo, 318
- lotsOfSprites, 328–331
- makeaSprite, 323–326
- oopBall, 334–336
- oopManyBalls, 340
- paddleBounce, 152–155

- soundDemo, 191
- space, 301–302
- twoPlanets, 310–312
- vecProj, 257–259

projectiles, 267

projecting vectors, 345–346

projects

- creating, 19–20

- naming, 24

- templates, 20–21

programming games

- benefits of, 9

- coding, 15–16

- competition, 11

- goal of game development, 13–14

- imagination, 10

- interactivity, 11

- Macromedia Flash, 11–12

- objects, 11

- process, 16–18

- skills required, 9–11

- sound effects, 185

- themes, 14–15

properties

- ActionScript, 125

- bullet.dir, 261

- bullet.speed, 261

- classes, 332

- defined, 120

- dir, 210

- dx, 123–124, 203, 210, 249, 252–254, 267

- dy, 123–124, 203, 210, 249, 252–254, 267

- gun.charge, 261

- gun.dir, 261, 263

- height, 131–132

- lineStyle, 272

- objects, 339

- rotation, 208

- sprites, 204–206

- traction, 313–315

- width, 131–132

- _xmouse, 140

- _ymouse, 140

Properties Inspector (Macromedia Flash), 22

pseudocode, 39

publishing Web page files, 27

Q

"" (quotes, double) for outputting text, 41

R

radial gradient, 115
 radians (angles), 256
 radians-to-degrees conversion, 257
 random() function, 71–72
 random integer generator, 348
 random number generator, 69–72
 real numbers, 71
 reality in games, 247
 registration point, 117, 345
 removeMovieClip() method, 327, 340
 repeated behavior. *See* loops
 replacing the mouse pointer, 137–140
 resetFlame method, 232–234
 retrieving elements from arrays, 321
 right triangles, 250–252
 risk-rewards principle, 151–152
 roll program, 69–73
 rotation property, 208
 round() function, 72
 rounding numbers, 72
 royalty-free sound effects, 186

S

sampling rate of audio files, 187
 scorekeeping systems
 Monster Traffic game, 177, 240–242
 Pong, 161–165
 scripting (ActionScript)
 animation, 12
 arrays, 320–321
 comparisons, 77
 conditions, 76–77
 constants, 204
 enumerations, 204
 features, 11–12
 object-oriented programming (OOP), 120
 properties, 125
 trigonometry functions, 259
 server requirements, 28
 shape tween tool, 12

shapes
 animated sprites, 193–198
 creating, 193–195
 gradient fills, 195
 modifying, 195–197
 mouse pointers, 137–139
 vertices, 197–198
 sharing variables, 98
 side scroller, 373
 simulating gravity, 268–269
 sin of an angle, 252
 SOHCAHTOA mnemonic, 252
 solving
 angles, 255
 distance, 255–256
 sound effects
 adding, 190–191
 animation, 184–185
 arcade games, 183
 Audacity open source audio editor, 186
 audio compression, 186–187
 combining, 192
 creating, 186
 game programming, 185
 headphones, 184, 192
 legal issues, 185–186
 Macromedia Flash, 183–184
 Monster Traffic game, 178, 234–236
 MP3 audio format, 185
 royalty-free, 186
 stereo effects, 192
 stopping, 192
 testing, 192
 time dependency, 184–185
 tips for using, 191–192
 volume, 192
 WAV audio format, 185
 sound files
 audio compression, 186–187, 190
 bit depth, 187
 importing, 187–189, 191–192
 MP3 format, 185
 sampling rate, 187
 WAV format, 185
 Sound object, 185
 soundDemo program, 191
 Sound.setPan() method, 192
 Sound.setVolume() method, 192
 Sound.start() method, 189

- Sound.`stop()` method, 192
- source code files, 5
- Space Invaders*
 - creating, 367–368
 - enhancements, 368–369
- spaceships
 - creating, 303
 - direction, 302
 - gravity, 306–310
 - keyboard input, 303–304
 - Lunar Lander* game, 354–355
 - moving, 306
 - space program, 301–302
 - states, 302–303
 - turning, 304–306
 - `twoPlanets` program, 310–312
- Spacewar*, 141
- speed
 - defined, 283
 - `dx` property, 249
 - `dy` property, 249
 - `gravityTrace` program, 248
 - vector projection, 248
- sprites
 - animation, 193–201
 - boundary checking, 128–137
 - code, 117–119
 - creating, 113, 323–326
 - creating many copies, 328–331
 - custom sprites, 334–340
 - defined, 17, 113
 - destroying, 323, 326–328
 - examples, 114
 - `killSprite` program, 326–327
 - `lotsOfSprites` program, 328–331
 - `makeASprite` program, 323–326
 - Monster Traffic* game, 201
 - motion, 349
 - `MovieClip` object, 114, 116–117
 - moving, 201–204, 211
 - naming, 117
 - properties, 204–206
 - registration point, 117, 345
 - suicidal sprites, 326–327
 - 3-D ball, 115–116
 - turning, 206–208
 - user-controlled, 212–217
- Stage (Macromedia Flash), 21–22
- STAIR analysis of programming, 236
- starters for games
 - Asteroids*, 352–353
 - Breakout*, 365–366
 - Egg Cannon*, 355–357
 - Lunar Lander*, 353–355
 - Orbit Matcher*, 369–371
 - Pac-Man*, 45
 - platform scrolling games, 362–364
 - Space Invaders*, 367–369
 - tile-based world games, 371–373
 - Whack-an-Author, 373–376
 - Zelda*, 357–362
- starting state, 165
- Statement must appear within `onClip` event handler, 108
- states (of button)
 - down state, 36
 - over state, 34–36
 - up state, 36
- states (of game)
 - adding, 167–172
 - Green Grass* game, 46–51
 - keyframes, 47–48
 - losing state, 165–166
 - Monster Traffic* game, 239–240
 - multi-state games, 48
 - Pac-Man* example, 45
 - spaceship example, 302–303
 - starting state, 165
 - swapping, 52–53
 - winning state, 165–166
- static text, 63–65
- stereo effects, 192
- stopping sound effects, 192
- strategy games, 373
- strings
 - concatenation, 68–69
 - converting to integers, 89
- `String.fromCharCode()` method, 182
- style conventions, 41
- suicidal sprites, 326–327
- swapping states, 52–53
- SWF files, 28
- switch statement
 - `binaryDice` program, 82–84
 - directions (for movement), 208–209

symbols

- buttons, 31–33
- graphics, 33
- movie clips, 33
- syntax errors, 106–107

● T ●

- tangent of an angle, 252
- templates (projects), 20–21
- testing
 - customCursor program, 140
 - sound effects, 192
- text
 - buttons, 39
 - dynamic text, 63
 - input text, 63–66
 - static text, 63–65
 - string concatenation, 68–69
- text fields
 - transferring data between, 67
 - variables, 66–68
- Text tool, 64
- themes, 14–15
- this keyword, 294
- 3-D ball, 115–116
- 3-D buttons, 30
- tile-based world games
 - creating, 371–373
 - enhancements, 373
- time dependency of sound effects, 184–185
- Timeline (Macromedia Flash), 23
- tools
 - motion tween tool, 12
 - shape tween tool, 12
 - Text tool, 64
- Tools Panel (Macromedia Flash), 22–23
- trace function, 40–41
- traction property, 313–315
- trajectories of bullets, 266
- transferring data between text fields, 67
- triangles
 - adjacent side, 251
 - hypotenuse, 251
 - opposite side, 251

- right triangles, 250–252
- SOHCAHTOA mnemonic, 252
- trigonometry functions, 259
- trigonometry notation, 251–252
- trigonometry ratios, 252
- troubleshooting errors, 109
- true conditions, 78
- turn method
 - Cannon game, 264–265
 - carParam program, 299
 - carVector program, 291
 - Monster Traffic game, 202–203, 206–207, 210
 - space program, 304–305
- turning
 - cars, 291
 - spaceships, 304–306
 - sprites, 206–208
- 2-D games, 114
- twoPlanets program, 310–312

● U ●

- Unicode, 182
- unit vector, 259
- up state (of button), 36
- user-controlled sprites, 212–217
- UTF (Unicode Transformation Format), 182
- UTF-8, 182

● V ●

- values
 - adding to variables, 102
 - dx property, 124, 252, 254
 - dy property, 124, 252–254
 - literal value, 68
- var keyword, 338
- variables
 - adding values, 102
 - defined, 66
 - op variable, 109
 - sharing, 98
 - text fields, 66–68

vecProj program, 257–259

vectorAngle.fla file, 249

vectors

calculating, 272

carVector program, 288–291

combining, 348–349

defined, 247, 250

drawing, 249–250

dx property, 249

dy property, 249

generating, 346

length, 250, 275

magnitude, 250

motion, 248

normalized, 259

projecting, 345–346

right triangles, 250–252

unit, 259

vector projection, 248, 259–260

vehicles

arcade games, 279

boats, 312–313

carMove program, 201–204

carParam program, 295–300

carVector program, 288–291

carVectorOOP program, 291–294

keyboard input, 290–291

laws of motion, 279–282

motion, 349

spaceships, 301–312

traction property, 313–315

turning, 291

velocity (defined), 283

vertices (shapes), 197–198

visual design, 90–95, 144

volume, 192

• W •

W, A, S, and D keys (alternative arrows), 181

war games, 373

WAV audio format, 185

Web output with Macromedia Flash, 12

Web server requirements, 28

Web sites

Audacity open source audio editor, 186

Hampsterdance, 16

source code files, 5

weight (defined), 281

Whack-an-Author game, 373–376

while loop, 320

width property, 131–132

Wikipedia's hexadecimal color guide, 270

wind effects, 285–286

winning state, 165–166

writing guidelines, 41

• X •

_xmouse property, 140

• Y •

_ymouse property, 140

• Z •

Zelda

creating, 357–361

enhancements, 361–362

z-order, 325, 327