Index

A

Angry Birds, 19

B

Box2D engine, 39
animation
  constraint solver, 45
  integrator, 45
  updated init() function, 46
  world.ClearForces() function, 45
  world.DrawDebugData() function, 45
  world.Step() function, 45
body definition, 41
b2World object, 41
  allowSleep, 41
  creation, 41
  gravity, 41
contact listeners
  BeginContact(), 58
  createSimplyBody(), 59
  DrawDebugData() method, 61
  EndContact(), 58
  implementation, 58
  listenForContact(), 59
  PostSolve(), 58
  PreSolve(), 58
  watching collisions, 61
createBody() method, 42
createFixture() method, 42
createFloor() method, 42
DrawDebugData() method, 43
elements
  circular body, 49
  complex body, 52
  joints, connecting bodies, 54
  polygon-shaped body, 50
  rectangular body, 47
fixture definition, 42
fundamentals, 39
init() function, 43
SetAsBox() method, 43
setting up, 40
shapes, fixture, 42
tracking collisions and damage, 57

C

Combat system, 231
aircraft
  attack case, 245
  guard mode case, 245
  hunt case, 245
  order states, 242
  patrol case, 245
  processOrders() method, 245
bullets
  animate() method, 235
  animation sequences, 234
  default moveTo() method, 235
  drawingLoop() method, modify, 236
  findFiringAngle() method, 235
  loadItem() method, 236
  processOrders() method, 235
  properties, 234
  reachedTarget() method, 235
  references, 235
fog of war
  animate() and draw() method, 257
  animate() method, 256
  deploy grid, 261
  draw() method, 256
  fog object, 255
  hiding objects, 261
  initLevel() method, 256–257
  references, 257
  unbuildable, fogged areas, 259
Combat system (cont.)
- intelligent enemy, building, 251
  - showMessage() method, 253
  - timed triggers and hunt order, 251
- turrets
  - findTargetInSight() method, 239
  - ground turrets, 238
  - guard case modification, 240
  - isValidTarget() method, 239
  - map items, update, 241
- vehicles, 246

D
drawingLoop() method, 17

E, F
End level, game elements
- Message dialog box
  - CSS styles, 222
  - to game Object, 224
  - messageboxOK() and messageboxCancel() methods, 225
  - showMessageBox() method, 225
- trigger implementation
  - clearTimeout() method, 228
  - Conditional triggers, 225
  - end() method, 228
  - initTrigger() method, 228
  - runTrigger() method, 228
- Timed triggers, 225

Entities, 125
- add() and remove() methods, 131
- aircrafts, 125
- aircraft sprite sheet, 149
- Box2D
  - adding references, 68
  - animate() and drawAllBodies(), 74
  - animation, 77
  - <head> section, 68
  - object creation, 68
  - setting up, 73
- buildings, 125
- collision damage, 86
- create() method, 70
- definition, 65, 125
- draw() and animate() methods, 132
- draw() method, 75
- game.resetArrays() method, 131
- ground turret, 142
- harvester buildings, 139
- items array, 130
- level ending screen, 84
- levels.data array
  - ground entities, 72
  - hero and villain entities, 72
  - rectangular block entities, 72
- load-next-hero state
  - ApplyImpulse() method, 83
  - countHeroesAndVillains() method, 80
  - fired, 81
  - firing, 81
  - handlePanning() method, 82
  - mouseOnCurrentHero() method, 81
  - wait-for-firing, 81
- main base
  - buildings object, 127
  - loadItem() and addItem() methods, 128
  - sprite sheet, 126
- map definition, 129
- name property, 65
- objects, 66
- references, 126
- requirements property, 130
- restartLevel() method, 91
- select entities
  - clearSelection() method, 159
  - drag selection, 158
  - drawLifeBar() method, 160, 162
  - drawSelection() method, 159, 161
  - enable click selection, 156
  - itemUnderMouse() method, 157
  - mouse.click() method, 157
  - mouseup event handler, 157
  - selectItem() method, 159
- Slingshot band, 88
- sound
  - background music, 96
  - break and bounce sounds, 93
- starport building, 136
- startCurrentLevel() method, 130
- startNextLevel() method, 91
- terrain, 125, 152
- type property, 65
- vehicles object, 125, 144

G
Game elements, 195
- economic system
  - harvest animation state, 199
  - Loading Cash Amount, 196
  - sidebar object, 197
  - Starting Cash Amount, 195
- End level, 221, 225
  - Message dialog box, 221
  - trigger implementation, 225
purchase buildings and units
  adding sidebar buttons, 201
  at base, 214
  disable sidebar buttons, 203
  enable sidebar buttons, 203
  vehicle and aircraft construction, 206

Game world
  animation
    basic level, score bar, 33
    CSS styling, 32
    requestAnimation polyfill function, 30
    start() and animate() functions, 31
  game states
    final result, 37
    finite state machine, 35
    firing, 35
    handlePanning() method, 37
    load-next-hero, 35
    panning, 35
    panTo() method, 37
    wait-for-firing, 35
  HTML layout, 19
  images, loading
    CSS style, 26
    game.init() method, 28
    Image/Sound asset loader, 27
    loadImage()/loadSound() method, 29
    loading screen, 29
  level selection
    CSS styles, 25
    game.init() method, 24
    levels object, 23
    screen, 26
    showLevelScreen() method, 25
  levels, loading, 29
  mouse input handling, 33
  splash screen and main menu
    CSS styles, 21
    game layers, 21
    init() function, 22
    JavaScript code, 22
    jQuery hide() and show() functions, 22
    js/game.js game object, 22
    skeleton HTML file, 20
    start screen and menu options, 23

H
HTML5
  audio element
    <audio>Tag, multiple sources, 11
    autoplay, 11
    canPlaythrough, 12
    canPlayType(), 12
    file formats, 11
    loading, 12
    loop, 11
    preload, 11
  canvas element
    accessing, 2
    arc, 5
    beginPath(), 4
    closePath(), 4
    colors and transparency, 7
    creation, 2
    Drawing arcs, 5
    fill(), stroke(), 4
    full circle drawing, 6
    getgetContext() method, 2
    image drawing, 8
    lineTo, 5
    moveTo, 5
    rectangles drawing, 3
    Stroked triangle, 5
    text drawing, 6
    three-quarter arc drawing, 6
    transformation and rotation, 9
  file skeleton, 1
  image element
    image loading, 13–14
    Sprite Sheets, 14
  timer and game loops
    requestAnimationFrame, 16
    with setInterval, 15
    with setTimeout, 16

I, J, K
Intelligent unit movement, 165
  aircraft movement implementation
    animate() method, 171
    Default processOrders() method, 169
    findAngle() method, 170
    movement-related properties, 172
    moveTo() method, 170
    wrapDirection() method, 171
  collision detection and steering
    checkCollisionsObject() method, 185
    default moveTo() method, 186
    modify processOrders() method, 188
  command units
    click() method modification, 165
    sendCommand() method, 167
  Harvester vehicle deploy, 190
  pathfinding
    A* algorithm, 174
    add() and remove() methods, 179
    Dijkstra’s algorithm, 174
    mapGridWidth and mapGridHeight, 176
    properties, 175
Intelligent unit movement (cont.)
- rebuildPassableGrid() method, 178
- startCurrentLevel() method, 176
- processOrders() method, 169
- sending and receiving commands
  - getItemByUid() method, 168
  - implementation of, 167
  - processCommand() method, 168
  - Single-Player sendCommand() method, 168
- smoother unit movement, 191
- vehicle movement implementation
  - animate() method, 183
  - AStar() method, 183
  - findAngle() method, 183
  - moveTo() method, 181
  - pathfinding algorithm, 184
  - processOrders() method, 181

L
- Lock-step networking model, 319
  - network latency
    - finishMeasuringLatency() method, 320
    - latency_ping message, 321
    - measureLatency() method, 320
    - starting and finishing measurement, 321
  - sending commands
    - browsers, 327
    - from client, 324
    - handling messages, 325
    - sendCommand() method, 325
    - setInterval() method, 327
    - startGame() method, 325–326
    - tickLoop() method, 325

M, N, O
- Multiplayer game, 319
  - ending the game
    - connection errors, 333
    - loseGame() and endGame() methods, 328
    - player defeats, 328
    - player disconnected, 331
    - Server endGame() method, 329
    - triggered events, 328
    - type end_game, 330
    - type lose_game, 329
  - player chat
    - Keydown events, 335
    - message event handler, 336
    - receive messages, 337
    - styles, 335
- Multiplayer lobby screen
  - CSS code, 300
  - definition, 300

P, Q
- Physics engine, 39
- Purchase buildings and units, game elements, 200
  - adding sidebar buttons, 201, 203
  - CSS styles, 201
  - enable and disable, 203
  - to gameinterfacescreen, 201
  - animate() method, 214
  - cancelDeployBuilding() method, 219
  - click Event, 214
  - deploy grid, 216
  - finishDeployBuilding() method, 219
  - mouse.click() method, 217
  - processOrder() method, 220
  - rebuildBuildableGrid() method, 215
  - vehicle and aircraft construction
    - adding the Unit, 209
    - click event, 206
    - draw() method, 211
    - init() method, 206
    - processOrder() method, 207
    - showMessage() method, 207
    - Teleport action, 211–212

R
- Real-time strategy (RTS) games, 101
- game interface screen
  - animation and drawing loops, 115
  - animationLoop() method, 116
  - background, 115
  - CSS styles, 114
  - drawingLoop() method, 116
  - gameAnimationLoop() method, 117
  - HTML markup, 113
  - init() method, 116
  - layers, 113
  - singleplayer.play() method, 117
  - startCurrentLevel() method, 117
  - start() method, 116
Game Object init() method, 105
HTML layout, 101
map images
- basic level metadata, 108
- level-designing tool, 107
- singleplayer array, 109
Tiled software, 107
map panning implementation
- calculateGameCoordinates() method, 120
- click() method, 120
- drawingLoop() method, 122
- draw() method, 120
- game.handlePanning() method, 123
- handlePanning() method, 121
- init() method, 120
- mouse.draw() method, 123
- mouse object, 118
- panningThreshold and panningSpeed variables, 122
- updated game.init() method, 121
mission screen
- advantages, 112
- background, 110
- CSS style sheet, 109
- exit() method, 112
- HTML code, 109
- missionbriefing div, 109
- singleplayer object, 110
- startCurrentLevel() method, 112
- start() method, 112
- requestAnimationFrame and asset loader, 103
splash screen and main menu
- gamecontainer and layers, 102-103
- HTML file, 102
- implementation, 103
- JavaScript and CSS files, 102
Starting Screen and Loading Screen
- singleplayer.start() and multiplayer.start() methods, 107
- style sheet, 105
- with main menu, 107

S, T, U, V
Single-player campaign
- assault
  - air support, 281
  - ending mission, implementation, 290
  - Enemy Waves, 288
  - enemy waves and reinforcements, 280
  - mission brief, 276, 285
  - starports and refineries, 285
  - triggers, 286
  - triggers array, 279

- rescue
  - characters, 272
  - conditional trigger, 275
  - enemy and convoy, 273
  - isItemDead() method, 273
  - mission briefing, 272
  - scout-tanks, life of, 272
  - triggers array, 272
  - under siege, 282
Stop calling drawingLoop(), 17

W, X, Y, Z
WebSockets, 293
- on Browser, 293
  - displayMessage() method, 295
  - elements, 295
  - handlers, 295
- initWebSocket() method, 295
multiplayer game
  - in browser windows, 317
    - handleWebSocketMessage() method, 316
    - initGame() method, 312
    - initMultiplayerLevel() method, 316
  - levels, 310
    - message event handler, modify, 313
  - spawnLocations array, 312
  - startGame() method, 312
  - teamStartingItems array, 312
  - with Node.js, 293, 296-299
    - accept() method, 299
    - client and server interaction, 299
    - connectionIsAllowed() method, 299
    - HTTP server, 296
    - reject() method, 299
    - require() method, 298
    - send() method, 299
  - WebSocket server, 297
    - sendMessage() method, 295
  - server implementations, 295
  - WebSocket client, 293
Wrapping Up, 263
- adding sound
  - combat, 268
  - commands, 265
    - init() method, 264-265
  - messages, 268
  - objects, 263
  - play() method, 264
  - references, 264
single-player campaign
  - assault, 276
  - rescue, 269
  - under siege, 282