

# Index

## NUMBERS AND SYMBOLS

- | (pipe operator), 15
- 2D sprites, compression and, 17
- 3D effects, XNA Shooter game, 321–323
- 3D models
  - graphics engine, 121–122
  - loading, 6
  - loading time, 65
  - supported file formats, 63
  - XNA Shooter, 303–304
- 3D programming, 130–133
  - unit testing, 134–138
- 3D Studio Max files, support for, 64
- 32-bit operating systems, XNA development and, 11
- 64-bit operating systems, XNA development and, 11

## A

- Agile Methodology, 30–32
- alpha blending, textures, rendering, 17
- animated textures, XNA Shooter, 304–307
- Ants Profiler, 23
- application model, 4–6
- applications, Reflection (viewing dlls), 4
- ApplyGravity method, 380

- Assert helper method, 71
- Assert method, static unit tests, 36
- Asset Name property, 16

- asteroids
  - polygons, 180–181
  - textures, 181
  - unit test, 201–203
  - XNA Shooter, 304

## B

- background
  - color, setting with Draw method, 14–15
  - resolution considerations, 39
  - running game in, 5
  - Tetris, 102–103
  - textures
    - Pong game, 33
    - scrollable, 18–19
    - tiling, 15–19
- backgroundTexture variable, 17
- ball (Pong game), unit testing, 40
- billboards, XNA Shooter, 307–310
- bounding boxes, Pong game, 50–51
- Breakout game
  - coding levels, 83–85
  - collision testing, 89–90
  - Draw method, 87–89

## Breakout game (*continued*)

---

### **Breakout game (*continued*)**

game loop, coding, 85–87  
 overview, 81–82  
 unit testing, 83

### **Bullet Physics Library, 386**

### **bump value, calculating, 190**

## C

### **C#, suitability for game development, 20–21**

### **CalculateTangentFrames, 198–199**

### **cameras, 289**

SpaceCamera class, 290–293

### **car collision, XNA Racing Game, 397–401**

### **CarPhysics class, unit test, 377**

### **challenges**

DiffusePerPixel, 172  
 edge detection shader, 239–240  
 free camera class, 294  
 NextBlock game component, 113  
 parallax mapping, 206  
 road collision, 401–402  
 unit test track, 367–368  
 write game component, 145  
 write game with your engine, 330–331  
 write mod, 444–449  
 XACT project, 266

### **classes. *See also* helper classes**

ColladaCombiMode, 358  
 ColladaLoader, 358  
 ColladaTrack, 358  
 Game, 5, 94–95  
 GameComponent, 5  
 Game1.cs, 5  
 helper, usefulness of  
 Input, 99–100, 270–272  
   Update method, 272–274  
 Log, 68–69

PhysicsAsteroidManager, 394–397

ScreenshotCatcher, 141–142

SimpleCamera, 140–141

SimpleEffect, 149

SkyCubeMapping, 6

Sound, 100–101

SpaceCamera, 290–293

SpriteBatch, Pong game, 34–35

StringHelper, time stamps, 69

TangentVertex, 193–195

TextureFont, 97–99

TrackLine, 352

### **Clear method, 15**

### **coding. *See* programming**

### **Collada, 357**

helper classes, 358

### **Collada files, support for, 64**

### **ColladaCombiMode class, 358**

### **ColladaLoader class, 358**

### **ColladaTrack class, 358**

### **collision detection, XNA Racing Game, 393–394**

### **collision testing, Breakout game, 89–90**

### **collision unit testing, Pong game, 47–51**

### **color**

background, setting with Draw method, 14–15  
 correction, post-screen shaders, 230–233  
 specular color component, calculating, 190

### **ColorHelper class, 80**

### **community involvement, 6**

### **compiling**

3D models, 6  
 disadvantages of XNA, 64–65  
 shaders, 6, 167–168  
 sound files, 6  
 textures, 6  
 troubleshooting errors, 24  
 xnb files and, 7

**Components property, 95**

GameComponent class, 6

**compression**

DTX format, 16–17  
music data, 254–256  
sprites and, 17  
textures, 67

**ComputeTangentMatrix method, 188****Configuration, xbox 360, 11–13****content pipeline, 4, 6**

compiling content, disadvantages, 64–65  
content directories, 65–66  
dlls, 7  
loading time, 65  
overview, 6, 62  
supported file formats, 63–64

**contnet files, importing, 66–68****Contnet Importer property, 16****Contnet Processor, textures, 66–67****Contnet Processor property, 16****controllers. See game controllers****CPU, system requirements, XNA Game Studio Express, 8****custom file formats, support for, 64****D****dds files, compression format, 16–17****debugging. See also troubleshooting**

content pipeline and, 6  
error messages, logging, 68–69  
starting projects, 14  
unit tests and, 72  
Xbox 360 and, 13

**development**

C# suitability, 20–21  
tools, 22–23

**development history (XNA), 1–2****diffuse textures, 182****DiffusePerPixel shader challenge, 172****directional lights, 158****DirectSound, 246–247****DirectX**

obtaining, 9  
SNA Game Studio Express and, 8

**DirectX 8, shaders and, 148****DirectX 9, shaders and, 148****DirectX SDK, Game Studio Express and, 9****displaying, models, 68****dlls**

application model, 4  
content pipeline, 7  
NUnit.Framework.dll, 70  
viewing, 4

**Draw method**

background color, setting, 14–15  
Breakout game, 87–89  
Game class, 5  
textures, rendering, 17

**DrawSprites method, Pong game, coding, 39****drivers, Xbox 360, using controller on PC, 10****DTX compression format, 16–17****DXT compression algorithms, 182, 184****DxTexture tool, sky cubes, 211****dynamic unit tests, compared to static, 28****E****edge detection shader, writing, 239–240****Effect class, XNA Shooter game, 322–323****Effect Importer property, 16****EffectManager, 297****effects, 441****entering text, 276–279**

## EnumHelper class

---

**EnumHelper class, 79–80**

**error messages, logging, 68–69**

## F

**F5, launching projects, 14**

**.fbx files, support for, 64**

**FBX Model Importer, 67**

**file formats, supported, 63–64**

**file naming, content files, 65–66**

**filenames, extracting, StringHelper class,  
75–76**

**firewall rules, Xbox 360 testing, 9**

**fixed function pipeline, support for, 8**

**floating-point numbers, render methods  
and, 42**

**fun racing games, 337**

**functionality (XNA), overview of, 1–2**

**functions, TransformPosition, 186**

**FX Composer, shaders and, 153–155**

**fx files, layout, shaders and, 155–156**

**.fx files, Effect Importer and, 16**

## G

**Game Application Model, 4**

**Game class, 5, 94–95**

**Game1 class, methods, 14**

**game components**

advantages and disadvantages, 5–6  
drag and drop, 96

**game concept, 405–409**

additional features, 412–414  
game idea, 409–411

**game controllers**

capturing input, Update method, 18  
Rumble support. adding, 57  
Xbox 360, using with PC, 10

**game data, loading, 62–63**

**game design**

Agile Methodology, 30–32  
basic idea formulation, 29  
initial difficulties, identifying and solving,  
32–33  
overview, 28  
planning development technology, 30  
planning features and game play, 29–30  
planning phase, 28–29  
sound files, 34  
textures, 33

**game development, C# suitability, 20–21**

**game loop, Breakout game, coding, 85–87**

**game modes, 442–443**

**game screens, 279–281, 414–415**

GameScreen class, 420–422  
Help screen, 282  
Highscores screen, 422–424  
in-game user interface, 283–286  
main menu, 418–419  
splash screen, 415–418

**Game Studio Express**

obtaining, 9  
system requirements, 8  
Xbox 360, 12

**GameComponent class, 5, 95**

**Game1.cs class, 5**

**Gamepad controller, Rumble support, adding  
for Pong game, 57**

**gamepad input, capturing, Update method, 18**

**glass shader, 204**

**Gran Turismo racing game, 337–338**

**Graphic Engine, 4**

**graphics card, XNA Game Studio Express  
requirements, 8**

**graphics engine**

- capabilities, 118–119
- 3D models, 121–122
- line rendering, 125–129
- texture rendering, 123–125
- unit testing, 119–120

**gravity, 379–382**

- Tetris, 107–109

**grid, Tetris, 103–104****H****HandleKeyboardInput method, 277****Havok, 384–385****height map, landscapes, 313–315****helper classes**

- ColorHelper, 80
- EnumHelper, 79–80
- Input class, 99–100
- RandomHelper
  - overview, 73–74
  - random vector generation, 74
- Sound class, 100–101
- SpriteHelper, 77–79
- StringHelper
  - filenames, extracting, 75–76
  - lists, creating, 76–77
  - overview, 74–75
- TextureFont class, 97–99
- usefulness of, 61

**helper methods**

- Assert, 71
- IsInList, 71

**highscore list, online, 440–441****Highscores screen, 422–424****HLSL (High Level Shader Language), 148****I****IDEs (Integrated Development Environments)**

- game concept planning and, 30
- Visual Studio 2005 Professional, 10–11
- XNA Game Studio Express, 2–4
- XNA Game Studio Pro, 3
- XNA Studio, 3

**importing**

- content files, 66–68
- 3D models, 6
- models, 67–68
- shaders, 6
  - to engine, 166–172
- sound files, 6
- textures, 6

**Initialize method**

- Game class, 5
- Pong game, textures, 34–35
- textures, loading, 17

**Input class, 99–100, 270–272**

- keyboard access, adding, 15
- mouse rectangles, 274–276
- Update method, 272–274

**installation**

- XNA, Xbox 360, 11
- XNA Game Studio Express, 9–10

**interface, user. See UI (user interface)****interpolating splines, XNA racing game, 355–356****IsInList helper method, testing, 71****Item class, XNA Shooter game, 328–329****K****keyboard**

- accessing, Input class, 15
- input, capturing with Update method, 18

## LandscapeNormalMapping.fx shader

### L

**LandscapeNormalMapping.fx shader, 347**

#### landscapes

- base texture, 311–312
- height map, 313–315
- normal map, 311–312
- objects, adding, 315–317
- optimizing, 348–350
- rendering, 342–347
- textures, 341–342

#### lights

- calculation, 188
- directional, 158
- point, 158
- spot, 158

**lightVec variable, 188**

**line rendering, graphics engine, 125–129**

**Load method, 355–356**

**LoadGraphicsContent method, Game1.cs class, 5**

#### loading

- game data, 62–63
- textures, 17, 22, 68

**Log class, 68–69**

**logging, error messages, 68–69**

**Log.txt file, 69**

**loopings, physics, 387–390**

### M

**main menu, 418–419**

**Managed DirectX. See MDX (Managed DirectX)**

**Material class, 193**

**matrices, vertex shaders and, 187–188**

**md3 files, support for, 64**

**md5 files, support for, 64**

**MDX (Managed DirectX), migrating to XNA, 22**

**memory, system requirements, XNA Game Studio Express, 8**

**menu effects, post-screen shaders, 233–234**

**menu sounds, 265–266**

#### menus

- game planning and, 29–30
- Pong game, unit testing, 36, 38–39

#### methods

- ApplyGravity, 380
- Assert, static unit tests, 36
- Clear, 15
- ComputeTangentMatrix, 188
- DrawSprites, 39
- Game class, 5
- Game1.cs class, 5
- HandleKeyboardInput, 277
- IsInList helper method, testing, 71
- render, floating-point numbers and, 42
- RenderBall, 41
- RenderPaddles, 42
- RenderSprite, 37
- ShaderEffect SetParameters, 192
- ShowLives, 41
- SpriteHelper class, 78–79
- StringHelper class compared to Path class, 76
- TestBallCollisions, 48
- TestMenuSprites, 37–38
- VertexElement, 195

**Microsoft.Xna.Framework.Content.Pipeline.dll, 4, 7**

**Microsoft.Xna.Framework.Content.Pipeline.EffectImporter.dll, 7**

**Microsoft.Xna.Framework.Content.Pipeline.FBXImporter.dll, 7**

**Microsoft.Xna.Framework.Content.Pipeline.TextureImporter.dll, 7**

**Microsoft.Xna.Framework.Content.Pipeline.XImporter.dll, 7**

**Microsoft.Xna.Framework.dll, 4**

**Microsoft.Xna.Framework.Game.dll, 4**

**migration, MDX to XNA, 22**

**Model Importer property, 16**

**models**

displaying, 68

importing, 67–68

**motion blur, post-screen shaders and, 229–230**

**moveFactorPerSecond, calculating, 19**

**movement, moveFactorPerSecond,**

calculating, 19

**multiplayer mode, 443–444**

**music files, 249**

supported formats, 63

## N

**Need for Speed racing game, 338**

**.Net Framework 2.0, XNA Game Studio**

Express and, 8

**network games, player synchronization, 5**

**Newton Game Dynamics, 385**

**Newton's laws of motion, 372–373**

**Normal Mapping, unit test, 195–197**

**Normal Mapping shaders, 176**

binormal vector, 178–179

implementing, 177

pixels, 178

tangent vector, 178–179

textures, 177

**Normal Maps, DXT compression algorithms,  
182, 184**

**NormalMapCompressor, 182**

**NUnit**

running tests, 71–72

testing and, 10

**NUnit.Framework.dll, 70**

**nV Physics Engine, 386**

## O

**ODE (Open Dynamics Engine), 385**

**offset mapping, 203–204**

**online highscore list, 440–441**

**operating systems**

32-bit, XNA development and, 11

64-bit, XNA development and, 11

**optimization, pixel shader, 189–191**

## P

**parallax mapping, 206**

**parameters**

shaders, 168–169

shaders tutorial, 157–159

**physics**

gravity, 379–382

initial difficulties, identifying and solving,  
32–33

introduction, 371

loopings, 387–390

Newton's laws of motion, 372–373

Racing Game and, 374–378

spring physics, 390–393

**Physics and Math Library, 386**

**physics engines, 382–383**

Bullet Physics Library, 386

Havok, 384–385

Newton Game Dynamics, 385

nV Physics Engine, 386

ODE, 385

Physics and Math Library, 386

PhysX, 384

Tokamak Game Physics, 386

## PhysicsAsteroidManager class

---

### **PhysicsAsteroidManager class, 394–397**

#### **PhysX, 384**

#### **pipe (|) operator, 15**

#### **pipeline. See content pipeline**

#### **Pixel Shader**

optimizations, 189–191

shaders tutorials and, 163–166

#### **Pixel Shader 3.0, 175**

#### **pixels, outputting to screen, 163–166**

#### **planning. See game design**

#### **Player class, 297**

XNA Shooter game, 319–321

#### **player input, Pong game, coding, 43–47**

#### **player synchronization, network games, 5**

#### **point lights, 158**

#### **polygons, asteroids, 180–181**

#### **Pong**

Agile Methodology, 30–32

basic idea formulation, 29

coding

ball, 41

ball speed, 45–46

bounding boxes, 50–51

missed ball, 51

overview, 34

paddle speed and position, 44–45

paddles, 42

player input, 43–47

player lives, 41

rectangles, 35

sprites, 34–35

design overview, 28

history, 27

initial difficulties, identifying and solving,  
32–33

planning development technology, 30

planning features and game play, 29–30

planning phase, 28–29

Rumble support for Gamepad controller, 57

sound files, 34

adding, 52–55

textures, 33

troubleshooting, 58–59

unit testing, 35–39

ball, 40

collisions, 47–51

menus, 36

sound files, 55

Xbox 360, preparing for, 56–57

#### **post-screen shaders**

color correction, 230–233

implementation, 221

PostScreenDarkenBorder class, 223–226

RenderToTexture class, 222–223

unit testing results, 226–227

menu effects, 233–234

motion blur, 229–230

PostScreenDarkenBorder class, 223–226

PostScreenGlow, 228

RenderToTexture class, 222–223

sky cube mapping, 210–216

writing, 216–219

#### **PostScreenDarkenBorder class, 223–226**

#### **PostScreenDarkenBorder.fx, 216**

#### **PostScreenGlow, 228**

#### **PPUs (physics processing units), 371**

#### **Program.cs file**

contents, 14

unit testing and, 71

#### **programming**

Breakout game

game loop, 85–87

levels, 83–85

Pong

ball, 40–41

bounding boxes, 50–51

- menu testing, 36
  - missed ball, 51
  - overview, 34
  - paddles, 42
  - player input, 43–47
  - player lives, 41
  - rectangles, 35
  - sprites, 34–35
  - unit testing, 35–39
  - RenderSprite method, 37
  - StartTest, 37
  - TestDelegate, 36–37
  - TestMenuSprites, 37–38
  - Projectile class, 298**
    - XNA Shooter game, 325–327
  - projectiles, 326–327**
  - projects**
    - creating, 13–14
    - file properties, accessing, 16
    - Game class, 5
    - quitting, 15
    - starting, 14
    - textures, adding, 16
  - properties**
    - Asset Name, 16
    - Content Importer, 16
    - Contentnet Processor, 16
    - files in projects, accessing, 16
    - GameComponent class, Component, 6
- R**
- racin**
    - fun racers, 337
    - Gran Turismo, 337–338
    - Need for Speed, 338
    - realistic racers, 336
    - street racers, 336
    - Trackmania, 339
  - RacingGame class, LoadLevel method, 345**
  - random number generation, RandomHelper class, 73–74**
  - random vector generation, RandomHelper class, 74**
  - RandomHelper class, overview, 73–74**
  - realistic racing games, 336**
  - rectangles, Pong game**
    - coding, 35
    - unit testing menus, 36
  - reflection shader, 205**
  - Reflection tool, dlls, viewing, 4**
  - RenderBall method, Pong game, 41**
  - rendering**
    - landscapes, 310–311, 342–347
      - base texture, 311–312
      - height map, 313–315
      - normal map, 311–312
    - objects, adding, 315–317
    - methods, floating-point numbers and, 42
    - shaders and, 169–170
    - textures, SpriteBatch, 17
  - RenderPaddles method, Pong game, 42**
  - RenderSprite method, Pong game, 37**
  - RenderToTexture class, 222–223**
  - resolution**
    - background rendering considerations, 39
    - post-screen shaders, 217
  - resources**
    - general XNA online information, 19–20
    - unit testing information, 32
  - Rocket Commander, game concept**
    - planning, 30

## Rocket Commander game

---

### **Rocket Commander game**

introduction, 209–210  
 MDX, moving to XNA, 236–237  
 screenshots, 237–239  
 threads, performance and, 237  
 XNA and, 234–236

### **Rocket Motor sound, 264–265**

## S

**screens, border texture, 220**

**ScreenshotCaptor class, 141–142**

**scrolling, background textures, 18–19**

**scrollPosition variable, 19**

**Shader Model, XNA Game Studio Express support, 8**

**Shader Model 4.0, 175**

**ShaderEffect class, 191–193**

Asteroid unit test, 201–203  
 custom processors, 197–201  
 Normal Mapping unit test, 195–197  
 TangentVertex format, 193–195

**ShaderEffect SetParameters method, 192**

**shaders, 441. See also post-screen shaders**

compiling, 167–168  
 example games, 150–151  
 glass shader, 204  
 history of, 148–149  
 importing, to engine, 166–172  
 introduction, 147  
 loading, 6  
 Normal Mapping, 176  
 offset mapping and, 203–204  
 parameters, 168–169  
 reflection, 205  
 rendering with, 169–170  
 testing, 171–172

tutorials

FX Composer, 153–155  
 fx files, 155–156  
 parameters, 157–159  
 pixel shader, 163–166  
 setup, 152  
 vertex input format, 159–161  
 vertex shader, 161–163  
 vertex shaders, 187–188  
 vertices, formats, 169  
 water, 205

**shadow mapping, 428–434**

**ShowLives method, Pong game, 41**

**SimpleCamera class, 140–141**

**SimpleEffect class, 149**

**SimpleShader.fx, 152**

FX Composer, 153–155

**sky cube mapping, 210–216**

**SkyCubeMapping class, 6**

**software development, Agile Methodology, 30–32**

**sound**

menu sounds, 265–266  
 .wav files, 247–249  
 XNA Shooter, 299–300

**Sound Banks (XACT), creating, 53**

**Sound class, 100–101, 261–266**

**Sound Cues (XACT), 54**

**sound files**

loading, 6  
 Pong game, 34  
 XACT and, 52–55  
 supported formats, 63  
 unit testing, 55

**SpaceCamera class, 290–293**

**specular color component, calculating, 190**

**splash screen, 415–418**

**spot lights, 158**

**spring physics, 390–393**

**SpriteBatch, textures, rendering, 17**

**SpriteBatch class, Pong game, 34–35**

**SpriteHelper class, 77–79**

**sprites**

compression and, 17

Pong, coding, 34–35

**StartTest, coding, 37**

**static unit tests**

Assert method, 36

compared to dynamic, 28

**street racing games, 336**

**StringHelper class**

filenames, extracting, 75–76

lists, creating, 76–77

methods compared to Path methods, 76

overview, 74–75

timestamps, 69

**swizzle, 189**

**synchronization, network games, 5**

**system requirements**

Visual C# Express, 8

XNA Game Studio Express, 8

## T

**tangents, data, custom processors, 197–201**

**TangentVertex class, 193–195**

**TestBallCollisions method, Pong game, 48**

**TestBallCollisions unit test, Breakout game, 83**

**TestDelegate, coding, 36–37**

**TestDriven.NET**

testing and, 10

unit testing, 70–71

**TestGameSprites, Pong game, 40**

**TestGameSprites unit test, Breakout game, 83**

**testing**

frameworks for, 10

IsInList helper method, 71

NUnit and, 10

Pong, unit testing, 35–39

shaders, 171–172

static unit tests compared to dynamic unit tests, 28

TestDriven.NET and, 10

**TestMenuSprites method, Pong game, coding, 37–38**

**TestSingleplayerGame unit test**

collision testing, 47

Pong game, 43–44

**TestSounds unit test, Breakout game, 83**

**Tetris, 101**

background, 102–103

block types, 104–107

gravity, 107–109

grid, 103–104

input, 109–112

testing, 112–113

**text, entering, 276–279**

**TextureFont class, 97–99**

**textures**

background, tiling, 15–19

Content Processor, 66–67

diffuse, 182

graphics engine, 123–125

landscapes, 341–342

loading, 6, 17, 22, 68

loading time, 65

Pong game, 33

coding, 34–35

scrollable, 18–19

supported file formats, 63

XNA Shooter, 302–303

animated textures, 304–307

## tiling

---

**tiling, background, textures, 15–19**

**time stamps, log files, 69**

**Tokamak Game Physics, 386**

**tools, 22–23**

**Track class, 360–365**

**TrackLine class, 352**

**Trackmania racing game, 339**

**TransformPosition function, 186**

**Troubleshooting, pong, 58–59**

**troubleshooting, 23–24. See also debugging**

**tutorials, shaders**

FX Composer, 153–155

fx files, 155–156

parameters, 157–159

pixel shader, 163–166

vertex input format, 159–161

vertex shader, 161–163

**TV screens, UI and, 285**

## U

**UI (user interface)**

in-game, 283–286

XNA Shooter, 300–302

**Unit class, 297**

XNA Shooter game, 323–325

**unit testing**

asteroids, 201–203

Breakout game, 83

CarPhysics class, 377

3D programming, 134–138

debugging, 72

final testing

Windows, 434–436

Xbox 360, 436–439

frameworks for, 10

graphics engine, 119–120

guidelines, 73

Normal Mapping, 195–197

NUnit, 70–71

overview, 70

Pong game, 35–39

ball, 40

collisions, 47–51

menus, 36

overview, 35

player input, 43–47

post-screen shaders, 226–227

running tests, 71–72

sound files, 55

StartTest, 37

static compared to dynamic, 28

TestDelegate, 36–37

TestDriven.NET, 70–71

Tetris, 112–113

XNA racing game, 352–354

**UnitTesting.cs file, 71**

**UnloadGraphicsContent method, Game1.cs  
class, 5**

**Update method**

collision testing, Pong game, 47

Game class, 5

Input class, 272–274

scrollable background textures, 18–19

Xbox 360, 15

**using directives, unit testing, NUnit, 70–71**

## V

**variables, cue variables, 258–259**

**vertex shader**

matrices and, 187–188

shaders tutorial and, 161–163

**VertexElement method, 195**

**vertices**

- formats, 169
- input, format, 159–161
- XNA racing game, 360–365

**viewVec variable, 188****Visual C# Express, XNA Game Studio Express and, 8****Visual Studio 2005 Professional, XNA development and, 10–11****W****water shader, 205****.wav files, 247–249**

- compatibility with Xbox 360, 52

**Wave Banks (XACT), creating, 53****Windows 2003, XNA Game Studio Express and, 8****Windows Vista, XNA Game Studio Express and, 8****Windows XP SP2, XNA Game Studio Express, system requirements, 8****X****.x files, Model Importer and, 16****X Model Importer, 67****XACT**

- acquiring, 250
- challenge project, 266
- cues, variables, 258–259
- effects, 259–261
- introduction, 245–246
- music data, compressing, 254–256
- projects, creating, 251–252
- sound banks, 256–258
- sound files, 34
  - adding to Pong game, 52–55

- .wav files, 247–249

- wave banks, creating, 252–254

**.xap files, 250****Xbox 360**

- configuration, 11–13
- controller, using on PC, 10
- debugging, 13, 145
- firewall rules, 9
- games
  - deploying, 144
  - setup, 144
- Pong game, preparing for, 56–57
- tools, 145
- troubleshooting, 23–24
- Update method, quitting programs, 15
- .wav file compatibility, 52
- XNA Game Launcher, 11–12
  - downloading, 143–144
  - XNA Game Studio Express, 12

**XMA files, compression, 254–256****XNA, installation, Xbox 360, 11****XNA Content Pipeline, 4****XNA Game Application Model, 4****XNA Game Launcher, Xbox 360, 11–12****XNA Game Studio Express**

- installation, 9–10
- obtaining, 9
- overview, 2–4
- system requirements, 8
- Xbox 360, 12

**XNA Game Studio Pro, 3****XNA Graphic Engine, 4****XNA projects. See projects****XNA Racing Game**

- car collision, 397–401
- collision detection, 393–394
- complex tracks, 356–357

## XNA Racing Game (*continued*)

---

### **XNA Racing Game (*continued*)**

- importing track data, 358–360
- interpolating splines, 355–356
- landscape
  - rendering, 342–347
  - textures, 341–342
- physics and, 374–378
- tracks, 351–352
- unit testing, 352–354
- vertices, generating, 360–365

### **XNA Shooter**

- asteroids, 304
- billboards, 307–310
- Corvette, 303
- 3D models, 303–304
- files, 298
- Firebird, 304

- game, 317–319
  - 3D effects, 321–323
  - final screenshot, 330
  - Item class, 328–329
  - logic, 319–321
  - Projectile class, 325–327
  - Unit class, 323–325
- introduction, 297
- OwnShip, 303
- Rocket-Frigate, 304
- Small-Transporter, 304
- sound, 299–300
- textures, 302–303
  - animated, 304–307
- UI (user interface), 300–302

### **XNA Studio, 3**

### **XnaShooter, 298**

**xnb files, 7**







