

# Index

## Numbers

### **2D games, 171**

- collision detection, 241-243
- coordinate systems, 172
- enemies, drawing, 235-241
- explosions, 247-252
- game skeletons, 217-220
- heroes, drawing, 230-235
- parallax scrolling, 220-229
- sounds, 252
- switching between states, 229-230
- text, 189-190
- transitions, 244-247
- winning and losing, 243-244

### **3D games. See tunnel vision game**

### **3D objects**

- bump mapping, 364
- cameras
  - creating, 61
  - projection, 61-62
  - view, 62
  - world, 63
- checking performance of, 75-78
- collisions
  - coefficient of restitution, 296-297
  - CollisionDemo program, 297-305
  - conservation of kinetic energy, 297
  - conservation of momentum, 296
  - final velocities, 297

- impulse, 296
- kinetic energy, 296
- momentum, 295
- overview, 295
- effects, 66-67
- GameComponent objects, 72-75
- index buffers, 69-72
- lighting
  - AdvancedTexturingDemo, 356-359
  - ambient lighting, 359-361
  - custom vertex format, 353-356
  - directional lighting, 361-364
  - overview, 353
- loading, 114-119
- matrices, 60
- motion dynamics. See kinematics
- normal mapping, 365-369
- parallax mapping, 369-372
- relief mapping, 372-376
- texture animation, 376-378
- textures, 68-69, 120-121
- transformations
  - definition of, 60
  - multiple transformations, 60-61
  - rotation, 60
  - rotation, 82-83
  - scaling, 60, 79-81
  - translation, 60
- vectors, 59-60
- vertex buffers, 63-66
- vertices, 59

***The 7 Habits of Highly Effective People*, 35**

**80-20 Rule, 36-37**

## A

### **acceleration, 290-294**

- AccelerationDemo program, 291-294
  - Draw method, 292
  - DrawModel method, 292-293
  - InitializeValues method, 292
  - LoadGraphicsContent method, 292
  - PhysicalObject class, 291
  - private member fields, 291
  - Update method, 293-294
- adding to objects, 290-291
- formula, 290
- Newton's Second Law of Motion, 295

### **AccelerationDemo program, 291-294**

- Draw method, 292
- DrawModel method, 292-293
- InitializeValues method, 292
- LoadGraphicsContent method, 292
- PhysicalObject class, 291
- private member fields, 291
- Update method, 293-294

### **acos function (HLSL), 263**

### **AddAnimation method, 198**

### **AddEnemy method, 446**

### **Additive, 213**

### **AddMissile method, 439**

### **AddService method, 87**

### **AddVerticesInformation, 160**

### **AdvancedTexturingDemo, 356-359**

- ambient lighting, 359-361
- directional lighting, 361-364
- drawing rectangle, 358-359
- InitializeIndices method, 357
- InitializeVertices method, 356-357

- loading content, 358
- private member fields, 356
- AI algorithms. See also FSM (finite state machine)**
  - advanced chase algorithm (TrackPlayer method), 343-344
  - AIDemo program, 337-342
    - Draw method, 339
    - DrawModel method, 338
    - Enemy class, 338
    - Initialization method, 339
    - KeepWithinBounds method, 341
    - Player class, 338
    - private member fields, 338
    - skybox, 337
    - sphere member field, 337
    - UpdatePlayer method, 340-341
  - evading algorithm (EvadePlayer method), 344
  - random movement algorithm (MoveRandomly method), 344-346
  - simple chase algorithm (TrackPlayerStraightLine method), 342-343
- AIDemo program, 337-342**
  - Draw method, 339
  - DrawModel method, 338
  - Enemy class, 338
  - Initialization method, 339
  - KeepWithinBounds method, 341
  - Player class, 338
  - private member fields, 338
  - skybox, 337
  - sphere member field, 337
  - UpdatePlayer method, 340-341
- AIState enumerated type, 346**
- algorithms (AI)**
  - advanced chase algorithm (TrackPlayer method), 343-344
  - AIDemo program, 337-342
    - Draw method, 339
    - DrawModel method, 338
    - Enemy class, 338
    - Initialization method, 339
    - KeepWithinBounds method, 341
    - Player class, 338
    - private member fields, 338
    - skybox, 337
    - sphere member field, 337
    - UpdatePlayer method, 340-341
  - evading algorithm (EvadePlayer method), 344
  - random movement algorithm (MoveRandomly method), 344-346
  - simple chase algorithm (TrackPlayerStraightLine method), 342-343
- all function (HLSL), 263**
- alpha blending, 173**
- ambient lighting, 359-361**
- AmbientTexture.fx file, 270-273, 359-360**
- animation**
  - cel animation. See cel animation
  - texture animation, 376-378
- ANTS Profiler, 37**
- any function (HLSL), 263**
- application data, passing to GPU, 270**
- ApplyFriction method, 303-304**
- asin function (HLSL), 263**
- assembly language, 256**
- AssemblyDescription attribute (games), 26**
- AssemblyInfo.cs file, 25**
- AssemblyTitle attribute (games), 25**
- associating Xbox 360 with computer, 22**
- atan function (HLSL), 263**
- atan2 function (HLSL), 263**

## B

**BaseGameState class, 312, 320**

**BasicEffect class, 66**

### benchmarks

creating, 37-39

micro-benchmark testing, 43-53

    CheckPerformance class, 44-48

    Program.cs file, 48-52

    time measurement relationships, 52-53

**bitmap fonts, 189**

### blend modes

Additive, 213

sprite batch demos, 173, 180-184

RotateAndScaleDemo, 203-204

**blurring images, 282**

**bubbles, creating with particle effects, 423-424**

### buffers

index buffers, 69-72

vertex buffers, 63-66

**bump mapping, 364**

## C

calculations. *See* formulas

**Camera class, 89**

**cameraReference field, 91**

### cameras

creating, 61

first person cameras

    flying camera, 105

    Update method, 103-105

game-specific cameras, creating for tunnel vision game, 457-461

input devices

    game pads, 95-98

    keyboards, 91-95

    mouse, 98-99

projection, 61-62

spin rate, 93-95

split screens

    Draw method, 109

    DrawScene method, 109

    player index, 107-109

    viewports, defining, 106-107

stationary cameras

    cull mode, 100

    Update method, 100-103

view, 62

world, 63

**capacity of particle systems, 402**

**ceil function (HLSL), 263**

**cell animation, 191-201**

    AddAnimation method, 198

    Draw method, 199

**Cg language, 256**

**chalk effect, 284**

**ChangeState method, 318**

### chase algorithms

    TrackPlayer method, 343-344

    TrackPlayerStraightLine method, 342-343

**CheckCollisions method, 449**

**CheckPerformance class, 44-48**

**clamp function (HLSL), 263**

### classes

    BaseGameState, 320

    BasicEffect, 66

    BasicGameState, 312

    Camera, 89

    CheckPerformance, 44-48

    DrawableGameComponent, 311

    Enemy (AIDemo), 338

    FadingState, 330-331

- FirstPersonCamera, 103-105
- GameState, 312-315
- GameStateManager, 315-316
  - ChangeState method, 318
  - ContainsState method, 319
  - PopState method, 316
  - PushState method, 317
  - RemoveState method, 316
- InputHandler, 87, 332
- LostGameState, 330
- OptionsMenuState, 324-326
- PhysicalObject, 291, 298
- Player (AIDemo), 338
- PlayingState, 327-329
- StartLevelState, 326-327
- StartMenuState, 322-324
- System.Diagnostics, 52
- Texture2D, 69
- TitleIntroState, 321-322
- WonGameState, 330
- YesNoDialogState, 327
- clip function (HLSL), 263**
- CLR Debugger 2.0, 167**
- coefficient of restitution, 296-297**
- collections, performance optimization, 53-54**
- CollisionDemo program, 297-305**
  - ApplyFriction method, 303-304
  - collision distance calculations, 302
  - DrawModel method, 300
  - final velocities, 302-303
  - Initialize method, 298
  - InitializeValues method, 299
  - PhysicalObject class, 298
  - private member fields, 297
  - Update method, 300-301
- CollisionDetection method, 440**
- collisions**
  - 2D games, 241-243
  - coefficient of restitution, 296-297
  - CollisionDemo program, 297-305
    - ApplyFriction method, 303-304
    - collision distance calculations, 302
    - DrawModel method, 300
    - final velocities, 302-303
    - Initialize method, 298
    - InitializeValues method, 299
    - PhysicalObject class, 298
    - private member fields, 297
    - Update method, 300-301
  - conservation of momentum, 296
  - final velocities, 297
  - impulse, 296
  - kinetic energy
    - conservation of, 297
    - formula, 296
  - momentum, 295
  - optimizing, 242
  - overview, 295
- color**
  - colorful effects, creating with particle effects, 429-430
  - fading to, 204-206
  - switching RGB values, 281-282
- commands**
  - Debug menu, Deploy Solution command, 23
  - Settings menu, Generate Connection Key, 22
- compiling**
  - Spacewar Windows Start Kit project, 17
  - tunnel vision game, 435
- conditions (HLSL), 267**
- connecting Xbox 360 to computer, 22**
- conservation of kinetic energy, 297**

conservation of momentum, 296  
**ContainsState** method, 319  
**Content Compiler**, 113  
**Content DOM**, 113  
**Content Importer**, 113  
**Content Manager**, 114  
**Content Pipeline**  
     Content Compiler, 113  
     Content DOM, 113  
     Content Importer, 113  
     Content Manager, 114  
     custom importers, 166  
     loading 3D models, 114-119  
     overview, 113-114  
     texturing 3D models, 120-121  
**Content Pipeline extensions, debugging**, 167  
**coordinate systems**  
     2D, 172  
     right-handed coordinate systems, 64  
**cos** function (HLSL), 263  
**cosh** function (HLSL), 263  
**Covey, Stephen**, 35  
**CreateCameraReferenceWithProperty** method, 46  
**CreateCameraReferenceWithValue** method, 47  
**CreateLookAt** method, 62  
**CreateParticle** method, 410  
**CreatePerspectiveFieldOfView** method, 61  
**CreateRotationY** method, 121  
**CreateScale** method, 81-83  
**CreateTexture** method, 392  
**CreateTranslation** method, 80-81, 276  
**Creators Club**, subscribing to, 20-21  
**cos** function (HLSL), 263  
**cross-fade** (dissolve transition), 384-385  
**crosshairs**, creating for tunnel vision game, 456-457

cull mode (stationary cameras), 100  
 custom Content Pipeline importer, 166  
 custom vertex format, creating, 353-356

## D

**D3DCOLORtoUBYTE4** function (HLSL), 263  
**ddx** function (HLSL), 263  
**ddy** function (HLSL), 263  
**Debug menu commands, Deploy Solution command**, 23  
**debugging**  
     Content Pipeline extensions, 167  
     Xbox 360, 25  
**degrees** function (HLSL), 263  
**demo program (Xbox 360)**, 25-28  
**Deploy Solution command (Debug menu)**, 23  
**deploying to Xbox 360**, 24  
**design, object-oriented**, 308-309  
**determining** function (HLSL), 263  
**directional lighting**, 361-364  
**directional wipes**, 385-390  
**DirectionalTexture.fx** file, 363-364  
**DirectX**, 7-9  
     FFP (Fixed Function Pipeline), 255  
     installing, 14-15  
**dissolve transition (cross-fade)**, 384-385  
**distance** function (HLSL), 263  
**distributing games**, 24-25  
**DOS**, 8  
**dot** function (HLSL), 264  
**downloading Visual C# Express**, 10  
**Draw** method, 27, 80, 292  
     AIDemo program, 339  
     cell animation, 199  
     GameStateDemo project, 334

- HLSLFireDemo project, 394-395
- particle system demo, 419-420
- particle system engines, 411-412
- radar, creating, 464
- split screens, 109
- TransitionsDemo project, 382-383
- tunnel vision game, 441-442, 450, 455

**DrawableGameComponent class, 311**

**drawing**

- 2D text, 189-190
- enemies (2D games), 235-241
- heroes (2D games), 230-235
- multiple sprites from one texture demo, 178-180

**DrawModel method, 118, 120, 292-293**

- AI Demo program, 338
- CollisionDemo program, 300

**DrawRectangle method, 79, 468**

**DrawScene method, 109**

**DrawUserIndexedPrimitives method, 71**

**dual platform programming, 28-32**

**dxwebsetup.exe file, 14**

## E

**effects, 66-67**

- blurred images, 282
- cell animation, 191-201
  - AddAnimation method, 198
  - Draw method, 199
- chalk, 284
- embossed images, 283
- explosions, 214-216
- fading to color, 204-206
- fire, 206-214
- grayscale, 283-284

- negative images, 281
- NightTime, 280-281
- postprocessing, 280-281
- rotation. See rotation
- scaling. See scaling
- sharpened images, 282
- switching RGB values, 281-282
- wavy, 284-285

**embossed images, 283**

**enemies, drawing (2D games), 235-241**

**Enemy class (AIDemo), 338**

**enumerations**

- AIState, 346
- managing game state with, 310
- PrimitiveType, 68

**EvadePlayer method, 344**

**evading algorithm (EvadePlayer method), 344**

**exiting Spacewar project, 24**

**exp function (HLSL), 264**

**exp2 function (HLSL), 264**

**explosions, creating, 214-216, 247-252**

**extensions, debugging, 167**

**extern modifier (HLSL), 259**

**ExternalForce particle systems, 402**

## F

**faceforward function (HLSL), 264**

**fading to color, 204-206**

**FadingState class, 330-331**

**FFP (Fixed Function Pipeline), 255**

**files**

- AssemblyInfo.cs, 25
- AmbientTexture.fx, 270-273, 359-360
- DirectionalTexture.fx, 363-364
- dxwebsetup.exe, 14

- Fire.fx, 395-396
- GameStateInterfaces.cs, 319-320
- interfaces.cs, 313
- NormalMapping.fx, 365-367
- ParallaxMapping.fx, 369-371
- ReliefMapping.fx, 372-375
- vcssetup.exe, 10
- xnagse\_setup.msi, 15
- .xnb file extension, 114
- final velocities, 297, 302**
- FindColorIndex method, 211**
- finite state machine. See FSM**
- fire effect, 206-214, 390**
  - constructor, 391
  - CreateTexture method, 392
  - Draw method, 394-395
  - Fire.fx file, 395-396
  - LoadGraphicsContent method, 391-392
  - private member fields, 390-391
  - UnloadGraphicsContent method, 392-394
- Fire.fx file, 395-396**
- first person cameras**
  - flying camera, 105
  - Update method, 103-105
- FirstPersonCamera class, 103-105**
- Fixed Function Pipeline (FFP), 255**
- floor function (HLSL), 264**
- fmod function (HLSL), 264**
- fonts**
  - bitmap fonts, 189
  - TrueType fonts, 188-189
- force, 294-295**
- formulas**
  - acceleration, 290
  - coefficient of restitution, 296
  - conservation of kinetic energy, 297
  - conservation of momentum, 296

- final velocities, 297
- force, 295
- impulse, 296
- kinetic energy, 296
- momentum, 295
- velocity, 290
- frac function (HLSL), 264**
- frexp function (HLSL), 264**
- FSMs (finite state machines)**
  - creating
    - AIState enumerated type, 346
    - conditions, 348-349
    - constants, 347-348
    - properties, 346
    - switch statement, 346-347
  - AIState enumerated type, 346
  - conditions, 348-349
  - constants, 347-348
  - creating, 346-349
  - overview, 307-308
  - properties, 346
  - switch statement, 346-347
- functions (HLSL), 263-267**
- fwidth function (HLSL), 264**

## G

- game components**
  - creating, 72-75
  - number of, 86
- game logic, tunnel vision game, 435-449, 452, 456**
  - AddEnemy method, 446
  - AddMissile method, 439
  - CheckCollisions method, 449
  - CollisionDetection method, 440



- Draw method, 441-442, 450, 455
- InitializeMissile method, 440
- Load method, 438
- LoadGraphicsContent method, 438, 444
- PhysicalObject class, 436
- StartLevelState, 453
- StateChanged method, 451, 454
- UnLoadGraphicsContent method, 438, 445
- Update method, 455
- UpdateEnemies method, 450
- VertexDisplacement.fx, 451
- game loops, 36**
- Game object, 86**
- game pads**
  - determining state of, 95-97
  - SetVibration method, 97-98
- game services**
  - creating, 85-87
  - GraphicsDevice, 86
- game skeletons, 217-220**
- game state. See state**
- Game Studio Express. See XNA Game Studio Express**
- game-specific cameras, creating for tunnel vision game, 457-461**
- Game1 method, 333-334**
- GameComponent objects, 72-75**
- games. See also projects**
  - 3D games. See 3D games
  - game components
    - creating, 72-75
    - number of, 86
  - game services
    - creating, 85-87
    - GraphicsDevice, 86
  - game state. See state
  - GameComponent objects, 72-75
  - Grand Theft Auto III, 309
  - improving, 499
  - postprocessing
    - blurring images, 282
    - chalk effect, 284
    - effect code, 280-281
    - embossed images, 283
    - game code, 277-280
    - grayscale images, 283-284
    - negative images, 281
    - NightTime effect, 280-281
    - overview, 277
    - sharpening images, 282
    - switching RGB values, 281-282
    - wavy effect, 284-285
  - Spacewar. See Spacewar project
  - split screens, 106-109
    - Draw method, 109
    - DrawScene method, 109
    - player index, 107-109
    - viewports, defining, 106-107
- GameState class, 312-315**
- GameStateDemo project**
  - BaseGameState class, 312, 320
  - ChangeState method, 318
  - ContainsState method, 319
  - Draw method, 334
  - FadingState class, 330-332
  - game state design, 312
  - Game1 constructor, 333-334
  - GameState class, 312-315
  - GameStateInterfaces.cs file, 319-320
  - GameStateManager class, 315-316
  - IGameState interface, 312
  - InputHandler class, 332
  - interfaces.cs file, 313
  - LostGameState class, 330

- member fields, 333
- OptionsMenuState class, 324-326
- PlayingState class, 327-329
- PopState method, 316
- PushState method, 317-318
- RemoveState method, 316-317
- StartLevelState class, 326-327
- StartMenuState class, 322-324
- TitleIntroState class, 321-322
- WonGameState class, 330
- YesNoDialogState class, 327
- GameStateInterfaces.cs file, 319-320**
- GameStateManager class, 315-316**
  - ChangeState method, 318
  - ContainsState method, 319
  - PopState method, 316
  - PushState method, 317
  - RemoveState method, 316
- garbage collection**
  - on .NET Compact Framework (Xbox 360), 42-43
  - on .NET Framework (Windows), 41-42
  - overview, 41
- Generate Connection Key command (Settings menu), 22**
- GetService method, 86**
- GetState method**
  - game pads, 95-96
  - keyboards, 91
- GPU, passing application data to, 270**
- Grand Theft Auto III, 309**
- graphics cards, 201**
- GraphicsDevice game service, 86**
- grayscale images, 283-284**
- Guid attribute (games), 26**

## H

- heroes, drawing (2D games), 230-235**
- High Level Shader Language. See HLSL**
- high score screens, creating, 487-491**
- history of XNA Framework, 8-9**
- HLSL (High Level Shader Language)**
  - AmbientTexture.fx demo, 270-273
  - application data, passing to GPU, 270
  - conditions, 267
  - intrinsic functions, 263-267
  - loops, 267
  - modifiers, 259
  - overview, 255-256
  - passes, 269-270
  - pixel shaders, 256, 268-269
  - postprocessing
    - blurring images, 282
    - chalk effect, 284
    - effect code, 280-281
    - embossed images, 283
    - game code, 277-280
    - grayscale images, 283-284
    - negative images, 281
    - NightTime effect, 280-281
    - overview, 277
    - sharpening images, 282
    - switching RGB values, 281-282
    - wavy effect, 284-285
  - semantics, 260-262
    - pixel input semantic list, 262
    - pixel output semantic list, 262
    - vertex input semantic list, 261
    - vertex output semantic list, 261
  - shader process flow, 257-258
  - structs, 260-262
  - techniques, 269
  - variable types, 258-260

- vertex displacement, 275-277
    - CreateTranslation method, 276
    - SetValue method, 276
    - timers, 275-276
  - vertex shaders, 256, 267-268
  - HLSLFireDemo project**
    - constructor, 391
    - CreateTexture method, 392
    - Draw method, 394-395
    - Fire.fx file, 395-396
    - LoadGraphicsContent method, 391-392
    - private member fields, 390-391
    - UnloadGraphicsContent method, 392-394
  - HUD, creating, 471-472**
- I**
- IGameState interface, 312**
  - IInputHandler interface, 87**
  - images**
    - blurring, 282
    - chalk effect, 284
    - color, switching RGB values, 281-282
    - embossed effect, 283
    - grayscale effect, 283-284
    - negative images, 281
    - sharpening, 282
    - wavy effect, 284-285
  - importing TrueType fonts, 188-189**
  - improving games, 499**
  - impulse, 296**
  - index buffers, 69-72**
  - Initialization method (AIDemo program), 339**
  - Initialize method**
    - CollisionDemo program, 298
    - Particle class, 400-401
    - InitializeCamera method, 61, 89
    - InitializeIndices method, 71, 357
    - InitializeMissile method, 440
    - InitializeTransformWithCalculation method, 45
    - InitializeTransformWithConstant method, 45
    - InitializeTransformWithConstantReferenceOut method, 46
    - InitializeTransformWithDivision method, 46
    - InitializeTransformWithPreDeterminedAspectRatio method, 46
    - InitializeValues method, 292
      - CollisionDemo program, 299
      - TransitionsDemo project, 382
    - InitializeVertices method, 65, 356-357
  - input devices**
    - game pads
      - determining state of, 95-97
      - SetVibration method, 97-98
    - keyboards
      - camera reference direction, 91
      - determining state of, 91-93
      - spin rate, 93-95
    - mouse, 98-99
  - input handlers, updating, 134-140**
  - Input namespace, 87**
  - InputHandler class, 87, 332**
  - installing**
    - DirectX 9 runtime, 14-15
    - Visual C# Express, 10-14
    - XNA Game Studio Express, 15-16
  - interfaces**
    - IGameState, 312
    - IInputHandler, 87
  - interfaces.cs file, 313**
  - intrinsic functions (HLSL), 263-267**
  - isfinite function (HLSL), 264**
  - isinf function (HLSL), 264**
  - isnan function (HLSL), 264**

## J-K

joining XNA Creators Club, 20-21

**KeepWithinBounds method, 341**

### keyboards

- camera reference direction, 91
- determining state of, 91-93
- spin rate, 93-95

### keywords

- matrix, 259
- ref, 41
- vector, 258

### kinematics

- acceleration
  - AccelerationDemo program, 291-294
  - adding to objects, 290-291
  - formula, 290
- force, 294-295
- overview, 289
- velocity, 290

### kinetic energy

- conservation of, 297
- formula, 296

## L

laser scanners, creating with particle effects, 426-427

laser shields, creating with particle effects, 424-426

**ldexpfunction (HLSL), 264**

**length function (HLSL), 264**

**lerp function (HLSL), 264**

level timers, 469-470

### libraries

- creating, 88-90
- referencing, 90

### lighting

- AdvancedTexturingDemo
  - drawing rectangle, 358-359
  - InitializeIndices method, 357
  - InitializeVertices method, 356-357
  - loading content, 358
  - private member fields, 356
- ambient lighting, 359-361
- custom vertex format, creating, 353-356
- directional lighting, 361-364
- overview, 353

**LineList member (PrimitiveType enumeration), 68**

**LineStrip member (PrimitiveType enumeration), 68**

**lit function (HLSL), 264**

**Load method, 438**

**Load3DObject project, 115-119**

**LoadGraphicsContent method, 26-27, 116-117, 120, 226, 278, 292**

- HLSLFireDemo project, 391-392
- particle system demo, 418
- particle system engines, 411
- TransitionsDemo project, 380
- tunnel vision game, 438, 444

### loading

- 3D models, 114-119
- screen demos, 176-178

**log function (HLSL), 264**

**log10 function (HLSL), 264**

**log2 function (HLSL), 264**

**loops, 36, 267**

**losing 2D games, 243-244**

**LostGameState class, 330**

**low function (HLSL), 265**

## M

### managing game state

- enumerated types, 310
- GameStateDemo project
  - BaseGameState class, 312, 320
  - ChangeState method, 318
  - ContainsState method, 319
  - Draw method, 334
  - FadingState class, 330-332
  - game state design, 312
  - Game1 constructor, 333-334
  - GameState class, 312-315
  - GameStateInterfaces.cs file, 319-320
  - GameStateManager class, 315-316
  - IGameState interface, 312
  - InputHandler class, 332
  - interfaces.cs file, 313
  - LostGameState class, 330
  - member fields, 333
  - OptionsMenuState class, 324-326
  - PlayingState class, 327-329
  - PopState method, 316
  - PushState method, 317-318
  - RemoveState method, 316-317
  - StartLevelState class, 326-327
  - StartMenuState class, 322-324
  - TitleIntroState class, 321-322
  - WonGameState class, 330
  - YesNoDialogState class, 327
- goals of, 309-310
- memory, 40-41
- stacks, 310-311

### mapping

- bump mapping, 364
- normal mapping, 365-369
- parallax mapping, 369-372
- relief mapping, 372-376

### mass, 294-295

#### matrices

- defining, 259
- definition of, 60
- tangent binormal normal matrix (TBN), 367

#### matrix keyword, 259

#### max function (HLSL), 264

#### measuring performance

- 80-20 Rule, 36-37
- benchmarks, 37-39
- overview, 35-36
- profiler tools, 37
- on Xbox 360, 39-41

#### membership in XNA Creators Club, purchasing, 20-21

#### memory. *See also* garbage collection

- buffers
  - index buffers, 69-72
  - vertex buffers, 63-66
- managing, 40-41

#### methods

- AddService, 87
- ApplyFriction, 303-304
- ChangeState, 318
- ContainsState, 319
- CreateCameraReferenceWithProperty, 46
- CreateCameraReferenceWithValue, 47
- CreateLookAt, 62
- CreatePerspectiveFieldOfView, 61
- CreateRotationY, 121
- CreateScale, 81-83
- CreateTexture method, 392
- CreateTranslation, 80-81, 276
- Draw, 27, 80, 292
  - AI Demo program, 339
  - GameStateDemo project, 334
  - HLSLFireDemo project, 394-395

- split screens, 109
- TransitionsDemo project, 382-383
- DrawModel, 118-120, 292-293
  - AlDemo program, 338
  - CollisionDemo program, 300
- DrawRectangle, 79
- DrawScene, 109
- DrawUserIndexedPrimitives, 71
- EvadePlayer, 344
- Game1, 333-334
- GetService, 86
- GetState
  - game pads, 95-96
  - keyboards, 91
- Initialization, 339
- Initialize, 298
- InitializeCamera, 61, 89
- InitializeIndices, 71, 357
- InitializeTransformWithCalculation, 45
- InitializeTransformWithConstant, 45
- InitializeTransformWithConstantReference-  
Out, 46
- InitializeTransformWithDivision, 46
- InitializeTransformWithPreDeterminedAspect  
Ratio, 46
- InitializeValues, 292
  - CollisionDemo program, 299
  - TransitionsDemo project, 382
- InitializeVertices, 65, 356-357
- KeepWithinBounds, 341
- LoadGraphicsContent, 26-27, 116-117,  
120, 278, 292, 380, 391-392
- MoveRandomly, 344-346
- pixelShader, 268
- PopState, 316
- PushState, 317-318
- RemoveState, 316-317
- ResolveRenderTarget, 279
- RotateElseif, 47
- RotateWithMod, 47
- RotateWithoutMod, 47
- SetRenderTarget, 279
- SetValue, 276
- SetVibration, 97-98
- StateChanged, 311
- TrackPlayer, 343-344
- TrackPlayerStraightLine, 342-343
- TransformVectorByReference, 44
- TransformVectorByReferenceAndOut, 45
- TransformVectorByReferenceAndOutVector-  
Add, 45
- UnloadGraphicsContent, 28, 392-394
- Update
  - AccelerationDemo program, 293-294
  - CollisionDemo program, 300-301
  - first person cameras, 103-105
  - stationary cameras, 100-103
- UpdateFade, 384
- UpdateInput, 381-382
- UpdatePlayer, 340-341
- UpdateWipe, 387
- VertexPositionNormalTexture, 65
- vertexShader, 267
- virtual methods, sealing, 53
- WipeDown, 387
- micro-benchmark testing, 43-53**
  - CheckPerformance class, 44-48
  - Program.cs file, 48-52
  - time measurement relationships, 52-53
- Microsoft Cross-Platform Audio Creation Tool.**  
**See XACT**
- min function (HLSL), 264**
- mod function (HLSL), 264**
- models, 3D. See 3D objects**
- modifiers (HLSL), 259**

momentum, 295-296  
 monitoring performance on Xbox 360, 39-40  
 motion dynamics. *See* kinematics  
 mouse input devices, 98-99  
 MoveRandomly method, 344-346  
 moveSpeed, 443  
 mul function (HLSL), 265

## N

namespaces, Input, 87  
 negative images, 281  
 .NET Compact Framework, 33, 42-43  
 .NET Framework, 41-42  
 Newton's First Law of Motion, 295  
 Newton's Second Law of Motion, 295  
 Newton's Third Law of Motion, 296  
 NightTime effect, 280-281  
 noise function (HLSL), 265  
 normal mapping, 365-369  
 normalize function (HLSL), 265  
 NormalMapping.fx file, 365-367  
 NProf, 37

## O

object-oriented design, 308-309  
 objects (3D). *See* 3D objects  
 optimizing performance
 

- collections, 53-54
- collision detection, 242
- micro-benchmark testing, 43-53
  - CheckPerformance class, 44-48
  - Program.cs file, 48-52
  - time measurement relationships, 52-53

overview, 43  
 virtual methods, sealing, 53  
 options menus, updating, 492-494  
 OptionsMenuState class, 324-326

## P

parallax mapping, 369-372  
 parallax scrolling, 220-229  
 ParallaxMapping.fx file, 369-371  
 Pareto, Vilfredo, 36  
 Particle class, creating, 398-399
 

- capacity, 402
- ExternalForce, 402
- Initialize method, 400-401
- ParticleSystemSettings.cs, 401-402
- randomness, 402-404
- Update method, 399

 particle effects, creating
 

- bubbles, 423-424
- colorful effects, 429-430
- laser scanners, 426-427
- laser shields, 424-426
- poisonous gas, 428-429
- rain, 420-423

 particle systems, 397, 495-497
 

- Particle class, creating, 398-399
  - capacity, 402
  - ExternalForce, 402
  - Initialize method, 400-401
  - ParticleSystemSettings.cs, 401-402
  - randomness, 402-404
  - Update method, 399
- particle effects, creating
  - bubbles, 423-424
  - colorful effects, 429-430

- laser scanners, 426-427
- laser shields, 424-426
- poisonous gas, 428-429
- rain, 420-423
- particle system demo, 417-418
  - Draw method, 419-420
  - LoadGraphicsContent method, 418
- particle system engines, creating, 405-414
  - CreateParticle method, 410
  - Draw method, 411-412
  - LoadGraphicsContent method, 411
  - PopulatePointSprites method, 413
  - SetBlendModes method, 413
- VertexPointSprite, 404-405
- ParticleSystemSettings.cs, 401-402**
- passes (HLSL), 269-270**
- passing by reference, 41**
- passing by value, 40**
- performance**
  - of 3D objects, checking, 75-78
  - garbage collection
    - on .NET Compact Framework (Xbox 360), 42-43
    - on .NET Framework (Windows), 41-42
    - overview, 41
  - measuring
    - 80-20 Rule, 36-37
    - benchmarks, 37-39
    - overview, 35-36
    - profiler tools, 37
    - on Xbox 360, 39-40
  - memory management, 40-41
  - optimizing
    - collections, 53-54
    - micro-benchmark testing, 43-53
    - overview, 43
    - virtual methods, sealing, 53
- overview, 35
- PhysicalObject class, 291, 298, 436**
- physics**
  - collisions
    - coefficient of restitution, 296-297
    - CollisionDemo program, 297-305
    - conservation of kinetic energy, 297
    - conservation of momentum, 296
    - final velocities, 297
    - impulse, 296
    - kinetic energy, 296
    - momentum, 295
    - overview, 295
  - kinematics
    - acceleration, 290-294
    - force, 294-295
    - overview, 289
    - velocity, 290
  - mass, 294-295
  - Newton's First Law of Motion, 295
  - Newton's Second Law of Motion, 295
  - Newton's Third Law of Motion, 296
- pixel shaders, 256, 268-269**
- pixelShader method, 268, 416**
- Player class (AIDemo), 338**
- player index, defining for split screens, 107-109**
- PlayingState class, 327-329**
- point sprites, 398, 414-417**
- PointList member (PrimitiveType enumeration), 68**
- poisonous gas, creating with particle effects, 428-429**
- PopState method, 316**
- PopulatePointSprites method, 413**
- postprocessing**
  - blurring images, 282
  - chalk effect, 284
  - effect code, 280-281



- embossed images, 283
- game code, 277-280
- grayscale images, 283-284
- negative images, 281
- NightTime effect, 280-281
- overview, 277
- sharpening images, 282
- switching RGB values, 281-282
- wavy effect, 284-285

**PrimitiveType enumeration, 68**

**profiler tools, 37**

**programming for dual platforms, 28-32**

**progress bars, 184-188**

**projection, 61-62**

**projects. See also games**

- AdvancedTexturingDemo
  - ambient lighting, 359-361
  - directional lighting, 361-364
  - drawing rectangle, 358-359
  - InitializeIndices method, 357
  - InitializeVertices method, 356-357
  - loading content, 358
  - private member fields, 356
- AIDemo, 337-342
  - Draw method, 339
  - DrawModel method, 338
  - Enemy class, 338
  - Initialization method, 339
  - KeepWithinBounds method, 341
  - Player class, 338
  - private member fields, 338
  - skybox, 337
  - sphere member field, 337
  - UpdatePlayer method, 340-341
- demo program (Xbox 360), 25-28

**GameStateDemo**

- BaseGameState class, 312, 320
- ChangeState method, 318
- ContainsState method, 319
- Draw method, 334
- FadingState class, 330-331
- game state design, 312
- Game1 constructor, 333-334
- GameState class, 312-315
- GameStateInterfaces.cs file, 319-320
- GameStateManager class, 315-316
- IGameState interface, 312
- InputHandler class, 332
- interfaces.cs file, 313
- LostGameState class, 330
- member fields, 333
- OptionsMenuState class, 324-326
- PlayingState class, 327-329
- PopState method, 316
- PushState method, 317-318
- RemoveState method, 316-317
- StartLevelState class, 326-327
- StartMenuState class, 322-324
- TitleIntroState class, 321-322
- WonGameState class, 330
- YesNoDialogState class, 327

**HLSLFireDemo**

- constructor, 391
- CreateTexture method, 392
- Draw method, 394-395
- Fire.fx file, 395-396
- LoadGraphicsContent method, 391-392
- private member fields, 390-391
- UnloadGraphicsContent method, 392-394

libraries, creating, 88-90

Load3DObject, 115-119

programming for dual platforms, 28-32

Spacewar project

compiling and running, 17

creating, 17-20

deploying to Xbox 360, 24

distributing, 24-25

exiting, 24

TransitionsDemo

Draw method, 382-383

InitializeValues method, 382

LoadGraphicsContent method, 380

private member fields, 379-380

UpdateFade method, 384

UpdateInput method, 381-382

XNAPerformanceChecker, 48-52

**purchasing XNA Creators Club memberships, 20-21**

**PushState method, 317-318**

## Q-R

radar, creating, 463-465

radian function (HLSL), 265

rain, creating with particle effects, 420-423

random movement algorithm (MoveRandomly method), 344-346

randomness in particle systems, 402-404

RBG values, switching, 281-282

ref keyword, 41

reference, passing by, 41

referencing libraries, 90

reflect function (HLSL), 265

refract function (HLSL), 265

relief mapping, 372-376

ReliefMapping.fx file, 372-375

RemoveState method, 316-317

**ResolveRenderTarget method, 279**

**restitution, coefficient of, 296-297**

**right-handed coordinate systems, 64**

**RotateAndScaleDemo, 201-204**

**RotateElseif method, 47**

**RotateWithMod method, 47**

**RotateWithoutMod method, 47**

**rotation**

definition of, 60

example, 82-83

RotateAndScaleDemo, 201-204

**round function (HLSL), 265**

**rsqrt function (HLSL), 265**

**running Spacewar Windows Start Kit project, 17**

## S

**saturate function (HLSL), 265**

**save state modes, 174-175**

**scaling objects**

definition of, 60

example, 79-81

RotateAndScaleDemo, 201-204

sprites, 172

**scoring**

adding to tunnel vision game, 472-473

high score screens, creating, 487-491

tracking high scores, 474-478

**scrolling, parallax, 220-229**

**sealing virtual methods, 53**

**semantics (HLSL), 260-262**

pixel input semantic list, 262

pixel output semantic list, 262

vertex input semantic list, 261

vertex output semantic list, 261

- services (game)**
  - creating, 85-87
  - GraphicsDevice, 86
- SetBlendModes method, 413**
- SetRenderTarget method, 279**
- Settings menu commands, Generate Connection Key, 22**
- SetValue method, 276**
- SetVibration method, 97-98**
- shaders**
  - AmbientTexture.fx demo, 270-273
  - application data, passing to GPU, 270
  - definition of, 255
  - passes, 269-270
  - pixel shaders, 256, 268-269
  - process flow, 257-258
  - techniques, 269
  - vertex displacement, 275-277
    - CreateTranslation method, 276
    - SetValue method, 276
    - timers, 275-276
  - vertex shaders, 256, 267-268
- sharpening images, 282**
- sign function (HLSL), 265**
- sin function (HLSL), 265**
- sincos function (HLSL), 265**
- sinh function (HLSL), 265**
- size of textures, 208**
- skyboxes**
  - adding to tunnel vision game, 434
  - AlDemo program, 337
  - creating, 153-154
    - skybox compilers, 162-163
    - skybox content objects, 154
    - skybox processors, 154-162
    - skybox readers, 164-165
  - using in games, 165-167
- smoothstep function (HLSL), 265**
- software**
  - DirectX runtime, installing, 14-15
  - Visual C# Express
    - downloading, 10
    - installing, 10-14
- sort modes**
  - sprite batch demos, 180-184
  - sprites, 173-174
- sound managers, 140-147**
- sounds, 147-150**
  - adding, 252, 497-499
  - sound banks, 125
  - sound demos, 147-150
  - sound managers, 140-147
  - variations, 125-132
  - wave banks, 124-125
  - XACT (Microsoft Cross-Platform Audio Creation Tool), 124
- Spacewar project**
  - compiling and running, 17
  - creating, 17-20
  - deploying to Xbox 360, 24
  - distributing, 24-25
  - exiting, 24
  - creating, 17
- special effects**
  - fire
    - constructor, 391
    - CreateTexture method, 392
    - Draw method, 394-395
    - Fire.fx file, 395-396
    - LoadGraphicsContent method, 391-392
    - private member fields, 390-391
    - UnloadGraphicsContent method, 392-394

- transitions
  - cross-fade (dissolve transition), 384-385
  - directional wipes, 385-390
  - TransitionsDemo project, 379-384
- spin rate, 93-95**
- splash screen demos, 176-178**
- split screens, 106-109**
  - Draw method, 109
  - DrawScene method, 109
  - player index, 107-109
  - viewports, defining, 106-107
- sprite batches, 172**
  - blend and sort mode demos, 180-184
  - drawing multiple sprites from one texture demo, 178-180
  - progress bar demo, 184-188
  - splash or loading screen demos, 176-178
- sprites, 171-172**
  - 2D text, 189-190
  - blend mode, 173
  - fonts
    - bitmap fonts, creating, 189
    - TrueType fonts, 188
    - TrueType fonts, importing, 188-189
  - point sprites, 398, 414-417
  - save state modes, 174-175
  - scaling, 172
  - sort modes, 173-174
  - sprite batches, 172
    - blend and sort mode demos, 180-184
    - drawing multiple sprites from one texture demo, 178-180
    - progress bar demo, 184-188
    - splash or loading screen demos, 176-178
- stacks**
  - definition of, 310
- GameStateDemo project
  - BaseGameState class, 312, 320
  - ChangeState method, 318
  - ContainsState method, 319
  - Draw method, 334
  - FadingState class, 330-332
  - game state design, 312
  - Game1 constructor, 333-334
  - GameState class, 312-315
  - GameStateInterfaces.cs file, 319-320
  - GameStateManager class, 315-316
  - IGameState interface, 312
  - InputHandler class, 332
  - interfaces.cs file, 313
  - LostGameState class, 330
  - member fields, 333
  - OptionsMenuState class, 324-326
  - PlayingState class, 327-329
  - PopState method, 316
  - PushState method, 317-318
  - RemoveState method, 316-317
  - StartLevelState class, 326-327
  - StartMenuState class, 322-324
  - TitleIntroState class, 321-322
  - WonGameState class, 330
  - YesNoDialogState class, 327
  - overview, 310-311
- start menus, updating, 483-487**
- StartLevelState class, 326-327, 453**
- StartMenuState class, 322-324**
- state, managing. See also FSM (finite state machine)**
  - enumerated types, 310
  - GameStateDemo project
    - BaseGameState class, 312, 320
    - ChangeState method, 318
    - ContainsState method, 319

- Draw method, 334
- FadingState class, 330-332
- game state design, 312
- Game1 constructor, 333-334
- GameState class, 312-315
- GameStateInterfaces.cs file, 319-320
- GameStateManager class, 315-316
- IGameState interface, 312
- InputHandler class, 332
- interfaces.cs file, 313
- LostGameState class, 330
- member fields, 333
- OptionsMenuState class, 324-326
- PlayingState class, 327-329
- PopState method, 316
- PushState method, 317-318
- RemoveState method, 316-317
- StartLevelState class, 326-327
- StartMenuState class, 322-324
- TitleIntroState class, 321-322
- WonGameState class, 330
- YesNoDialogState class, 327
- goals of, 309-310
- of game pads, 95-97
- of keyboard, 91-93
- of mouse input devices, 98-99
- stacks, 310-311
- switching between states, 229-230
- tunnel vision game, 433-434
- updating states, 495
- StateChanged method, 311, 451, 454**
- static modifier (HLSL), 259**
- stationary cameras**
  - cull mode, 100
  - Update method, 100-103
- step function (HLSL), 265**

- structs (HLSL), 260-262**
  - VertexPositionNormalTexture, 354-355
- subscribing to XNA Creators Club, 20-21**
- switching**
  - between states (2D games), 229-230
  - RGB values, 281-282
- swizzling, 259**
- System.Diagnostics class, 52**

## T

- tan function (HLSL), 265**
- tangent binormal normal matrix (TBN), 367**
- tanh function (HLSL), 265**
- TBN (tangent binormal normal matrix), 367**
- techniques (HLSL), 269**
- testing, micro-benchmark testing, 43-53**
  - CheckPerformance class, 44-48
  - Program.cs file, 48-49, 51-52
  - time measurement relationships, 52-53
- tex1D function (HLSL), 265**
- tex1Dbias function (HLSL), 265**
- tex1Dgrad function (HLSL), 265**
- tex1Dlod function (HLSL), 266**
- tex1Dproj function (HLSL), 266**
- tex2D function (HLSL), 266**
- tex2Dbias function (HLSL), 266**
- tex2Dgrad function (HLSL), 266**
- tex2Dlod function (HLSL), 266**
- tex2Dproj function (HLSL), 266**
- tex3D function (HLSL), 266**
- tex3Dbias function (HLSL), 266**
- tex3Dgrad function (HLSL), 266**
- tex3Dlod function (HLSL), 266-267**
- tex3Dproj function (HLSL), 266**
- texCUBE function (HLSL), 266**

- texCUBEbias function (HLSL), 266**
- texCUBEgrad function (HLSL), 267**
- texCUBEproj function (HLSL), 267**
- Texture2D class, 69**
- textures, 68-69**
  - 3D models, 120-121
  - AdvancedTexturingDemo, 356-359
    - ambient lighting, 359-361
    - directional lighting, 361-364
    - drawing rectangle, 358-359
    - InitializeIndices method, 357
    - InitializeVertices method, 356-357
    - loading content, 358
    - private member fields, 356
  - animation, 376-378
  - bump mapping, 364
  - normal mapping, 365-369
  - parallax mapping, 369-372
  - relief mapping, 372-376
  - size of, 208
- time measurement relationships, 52-53**
- timers**
  - for vertex displacement, 275-276
  - level timers, 469-470
- title screens, updating, 479-483**
- TitleIntroState class, 321-322**
- TrackPlayer method, 343-344**
- TrackPlayerStraightLine method, 342-343**
- transformations**
  - definition of, 60
  - multiple transformations, 60-61
  - rotation
    - definition of, 60
    - example, 82-83
    - RotateAndScaleDemo, 201-204
  - scaling
    - definition of, 60
    - example, 79-81
    - RotateAndScaleDemo, 201-204
    - translation, 60
- TransformVectorByReference method, 44**
- TransformVectorByReferenceAndOut method, 45**
- TransformVectorByReferenceAndOutVectorAdd method, 45**
- transitions**
  - adding in 2D games, 244-247
  - cross-fade (dissolve transition), 384-385
  - directional wipes, 385-390
  - TransitionsDemo project, 379-384
    - Draw method, 382-383
    - InitializeValues method, 382
    - LoadGraphicsContent method, 380
    - private member fields, 379-380
    - UpdateFade method, 384
    - UpdateInput method, 381-382
- TransitionsDemo project, 379-384**
  - Draw method, 382-383
  - InitializeValues method, 382
  - LoadGraphicsContent method, 380
  - private member fields, 379-380
  - UpdateFade method, 384
  - UpdateInput method, 381-382
- translation, 60**
- transpose function (HLSL), 267**
- TriangleFan member (PrimitiveType enumeration), 68**
- TriangleList member (PrimitiveType enumeration), 68**
- TriangleStrip member (PrimitiveType enumeration), 68**
- TrueType fonts, 188-189**
- tunnel vision game, 433**
  - compiling, 435
  - crosshairs, 456-457

- game logic, 435-449, 452, 456
  - AddEnemy method, 446
  - AddMissile method, 439
  - CheckCollisions method, 449
  - CollisionDetection method, 440
  - Draw method, 441-442, 450, 455
  - InitializeMissile method, 440
  - Load method, 438
  - LoadGraphicsContent method, 438, 444
  - PhysicalObject class, 436
  - StartLevelState, 453
  - StateChanged method, 451, 454
  - UnLoadGraphicsContent method, 438, 445
  - Update method, 455
  - UpdateEnemies method, 450
  - VertexDisplacement.fx, 451
- game states, 433-434
- game-specific cameras, 457-461
- HUD, 471-472
- level timers, 469-470
- radar, 463-465
- scoring
  - adding, 472-473
  - tracking high scores, 474-478
- skyboxes, 434
- tunnels, 465-469
- tunnels, creating, 465-469**
- types (HLSL), 258-260**

## U

- uniform modifier (HLSL), 259**
- UnloadGraphicsContent method, 28**
  - HLSLFireDemo project, 392-394
  - tunnel vision game, 438, 445

## Update method

- AccelerationDemo program, 293-294
- CollisionDemo program, 300-301
- first person cameras, 103-105
- Particle class, 399
- progress bar, 186
- stationary cameras, 100-103
- tunnel vision game, 455
- UpdateEnemies method, 450**
- UpdateFade method, 384**
- UpdateInput method, 381-382**
- UpdatePlayer method, 340-341**
- UpdateWipe method, 387**
- updating**
  - input handlers, 134-140
  - options menus, 492-494
  - start menus, 483-487
  - states, 495
  - title screens, 479-483

## V

- value abs function (HLSL), 263**
- value sqrt function (HLSL), 265**
- value, passing by, 40**
- variables**
  - passing by reference, 41
  - passing by value, 40
  - types, 258-260
- variations, 125-132**
- vcsetup.exe file, 10**
- vector keyword, 258**
- vectors**
  - defining, 258-259
  - overview, 59-60
- velocity, 290**
  - final velocities, 297, 302

**vertex buffers, 63-66**

**vertex displacement, 275-277**

- CreateTranslation method, 276
- SetValue method, 276
- timers, 275-276

**vertex shaders, 256, 267-268**

**VertexDisplacement.fx, 451**

**VertexPointSprite, 404-405**

**VertexPositionNormalTexture method, 65**

**VertexPositionNormalTexture struct, 354-355**

**vertexShader method, 267**

**vertices**

- custom vertex format, creating, 353-356
- overview, 59
- vertex buffers, 63-66
- vertex displacement, 275-277
  - CreateTranslation method, 276
  - SetValue method, 276
  - timers, 275-276
- vertex shaders, 256, 267-268

**view (camera), 62**

**viewports, defining for split screens, 106-107**

**virtual methods, sealing, 53**

**Visual C# Express**

- downloading, 10
- installing, 10-14

**volatile modifier (HLSL), 259**

## W

**wave banks, 124-125**

**wavy effect, 284-285**

**Windows**

- garbage collection, 41-42
- mouse input devices, 98-99
- Windows 95, 8

**winning 2D games, 243-244**

**WipeDown method, 387**

**WonGameState class, 330**

**world, 63**

## X-Y-Z

**XACT (Microsoft Cross-Platform Audio Creation Tool), 123-124**

### Xbox 360

- connecting to computer, 22
- debugging, 25
- deploying to, 24
- garbage collection, 42-43
- monitoring performance on, 39-40
- .NET Compact Framework on, 33
- test demo program, 25-28

**XNA Creators Club, subscribing to, 20-21**

### XNA Framework

- Content Pipeline
  - Content Compiler, 113
  - Content DOM, 113
  - Content Importer, 113
  - Content Manager, 114
  - loading 3D models, 114-119
  - overview, 113-114
  - texturing 3D models, 120-121
- definition of, 9-10
- influence of DirectX on, 8-9

**XNA Game Studio Express, 10, 15-16**

**.xnb file extension, 114**

**xnagse\_setup.msi file, 15**

**XNAPerformanceChecker project, 48-52**

**YesNoDialogState class, 327**