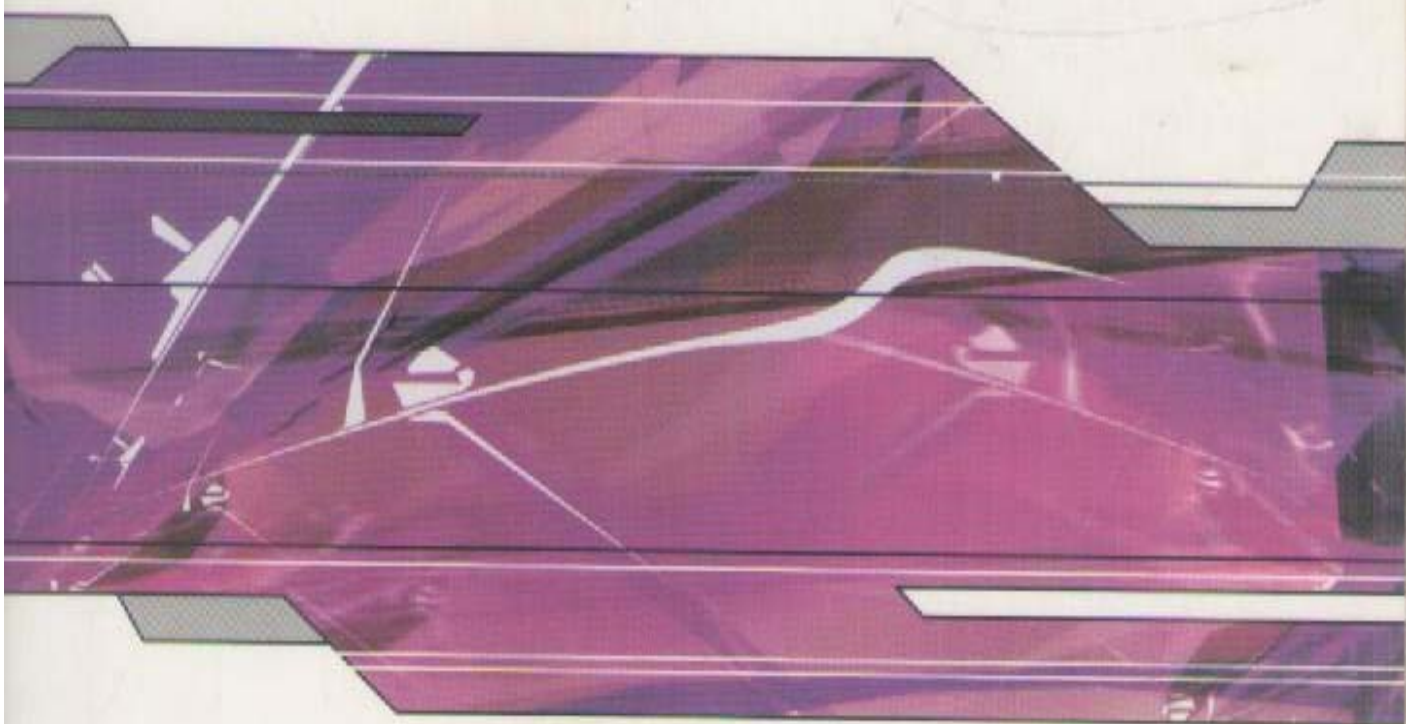


Developing Games in Java™



David Brackeen

with Bret Barker

and Laurence Vanhelsuwé

New
Riders

NR3

2-005-499-1

2-005-499-1

Developing Games in Java™

David Brackeen
with Bret Barker
and Laurence Vanhelsuwé



**New
Riders**

800 E. 96th Street, Suite 200, Indianapolis, Indiana 46240

An Imprint of Pearson Education

Boston • Indianapolis • London • Munich • New York • San Francisco

Table of Contents

Introduction	xxv
--------------	-----

Part I Java Game Fundamentals

Chapter 1	Java Threads	3
	What Is a Thread?	3
	Creating and Running Threads in Java	4
	Extending the Thread Class	5
	Implementing the Runnable Interface	5
	Using Anonymous Inner Classes	6
	Waiting for a Thread to Finish	6
	Sleepy Threads	6
	Synchronization	7
	Why Synchronize?	7
	How to Synchronize	8
	What to Synchronize	10
	What Not to Synchronize	10
	Avoiding Deadlock	11
	Using wait() and notify()	12
	The Java Event Model	14
	When to Use Threads	14
	When Not to Use Threads	14
	Sum It Up: Thread Pools	15
	Summary	22
Chapter 2	2D Graphics and Animation	23
	Full-Screen Graphics	24
	Screen Layout	24
	Pixel Color and Bit Depth	25
	Refresh Rate	26
	Switching the Display to Full-Screen Mode	27
	Anti-Aliasing	33
	Which Display Mode to Use	34

Images	34
Transparency	34
File Formats	35
Reading Images	36
Hardware-Accelerated Images	40
Image-Drawing Benchmarks	42
Animation	47
Active Rendering	51
The Animation Loop	51
Getting Rid of Flicker and Tearing	56
Double Buffering	56
Page Flipping	58
Monitor Refresh and Tearing	59
The BufferStrategy Class	60
Creating a Screen Manager	61
Sprites	71
Simple Effects	79
Image Transforms	80
Summary	87
Chapter 3 Interactivity and User Interfaces	89
The AWT Event Model	93
Keyboard Input	95
Mouse Input	100
Mouselook-Style Mouse Movement	105
Hiding the Cursor	111
Creating an Input Manager	113
Using the Input Manager	130
Pausing the Game	130
Adding Gravity	131
Designing Intuitive User Interfaces	140
User Interface Design Tips	140
Using Swing Components	141
Swing Basics	142
Tricking Swing to Work in Full-Screen Mode	143
Creating a Simple Menu	146
Letting the Player Configure the Keyboard	153
Summary	162

Chapter 4	Sound Effects and Music	163
	Sound Basics	163
	The Java Sound API	164
	Opening a Sound File	165
	Using a Line	166
	Playing a Sound	167
	Creating a Real-Time Sound Filter Architecture	174
	Creating a Real-Time Echo Filter	180
	Emulating 3D Sound	185
	The Idea Behind Creating a 3D Filter	186
	Implementing a 3D Filter	187
	Trying Out the 3D Filter	190
	Creating a Sound Manager	196
	The Sound Class	197
	The SoundManager Class	198
	Thread-Local Variables	208
	Playing Music	209
	Playing CD Audio	210
	Playing MP3 and Ogg Vorbis	210
	Playing MIDI Music	212
	Creating Adaptive Music	217
	Summary	220
Chapter 5	Creating a 2D Platform Game	221
	Creating a Tile-Based Map	222
	Implementing the Tile Map	223
	Loading Tile Maps	227
	Drawing Tile Maps	232
	Drawing Sprites	234
	Parallax Scrolling	234
	Power-Ups	237
	Simple Baddies	240
	Collision Detection	251
	Detecting a Collision	252
	Handling a Collision	253
	Sprite Collisions	257
	Finishing Things Up and Making It Fast	259
	Creating an Executable .jar File	259

	Ideas to Expand the Game	261
	Summary	262
Chapter 6	Multi-Player Games	263
	The Revolution in Java's I/O Libraries	265
	Overview of the JDK 1.4 NIO Libraries	266
	Channels	266
	Buffers	270
	Selectors and SelectionKeys	274
	ChatterBox, A Basic Multi-Player Application	275
	The Server: ChatterServer	276
	Building and Running the Server	284
	The Client: ChatterClient	286
	Building and Running the Client	288
	Multi-Player Game Server Framework	289
	Design Goals and Tactics	289
	The Design	290
	Common Classes and Interfaces	295
	Server Implementation	298
	The Client	309
	A Sample Game: RPS (Rock, Paper, Scissors)	310
	Classes	310
	Running the RPS Game	314
	Complete the Look: Building on the Framework	316
	Client GUI	316
	Persistence	317
	Buddy Lists, Lobbies, and Chat	317
	Server Administration	318
	Logging	318
	Startup/Shutdown	319
	Server Admin Consoles	320
	Game Watching	321
	Advanced Topics	321
	Disconnects and Reconnects	322
	HTTP Tunneling	323
	Testing with Bots	326
	Those Pesky Modems	327
	Profiling and Performance Stats	327
	Performance Tweaks	329
	Summary	331

Part II 3D Graphics and Advanced Techniques

Chapter 7	3D Graphics	335
	Types of 3D Rendering	335
	Don't Forget Your Math	337
	Trigonometry and Right Triangles	337
	Vector Math	338
	3D Basics	345
	3D Math	348
	Polygons	354
	3D Transforms	358
	Rotations	359
	Encapsulating Rotation and Translation Transforms	361
	Applying Transforms	366
	Rotation Order	368
	A Simple 3D Pipeline	369
	Camera Movement	374
	Solid Objects and Back-Face Removal	375
	The Dot Product	377
	The Cross Product	378
	More About the Dot and Cross Products	382
	Scan-Converting Polygons	383
	Optimizing Scan Conversion with Fixed-Point Math	389
	3D Clipping	394
	Final Rendering Pipeline	399
	Summary	409
Chapter 8	Texture Mapping and Lighting	411
	Perspective-Correct Texture Mapping Basics	412
	Deriving the Texture Mapping Equations	413
	A Simple Texture-Mapper	418
	Problems with the First Texture-Mapper	428
	Optimizing Texture Mapping	429
	Texture Storage	430
	Raw Optimization	435
	Method Inlining	440
	Fast Texture Mapping Demo	442

Simple Lighting	443
Diffuse Reflection	443
Ambient Lighting	444
Light Source Intensity	445
Distance Falloff	445
Implementing a Point Light	445
Implementing Texture Lighting	446
Advanced Lighting Using a Shade Map	456
Finding the Bounding Rectangle	457
Applying the Shade Map	459
Building the Shade Map	461
Building the Surface	464
Caching Surfaces	469
Shaded Surface Demo	477
Additional Concepts	478
Depth Cueing	479
Fake Shadows	479
MIP Mapping	479
Bilinear Interpolation	479
Trilinear Interpolation	480
Normal Maps and Depth Maps	480
Other Types of Lights	481
Summary	481
Chapter 9 3D Objects	483
Hidden Surface Removal	483
Painter's Algorithm	484
Reverse Painter's Algorithm	485
Z-Buffer	485
Z-Buffer Using 1/2	486
Calculating the Z-Depth	490
3D Animation	492
Spatial Movement	496
Angular Movement	498
Polygon Groups	505
Iterating All the Polygons in a Group	511
Loading Polygon Groups from an OBJ File	513
The OBJ File Format	514
The MTL File Format	522

Game Objects	524
Managing Game Objects	529
Putting It All Together	532
Future Enhancements	541
Summary	542
Chapter 10 3D Scene Management Using BSP Trees	543
BSP Tree Intro	544
Binary Tree Basics	545
The One-Dimensional BSP Tree	548
The Two-Dimensional BSP Tree	551
BSP Tree-Building Example	552
BSP Tree-Traversing Example	557
Implementing a 2D BSP Tree	558
The BSP Line	560
Determining the Side of a Point Relative to a Line	561
The BSP Polygon	565
Traversing a BSP Tree	566
In-Order Traversal	568
Front-to-Back Traversal	570
Building a Tree	571
Finding the Intersection of Two Lines	577
Clipping Polygons by a Line	580
Removing T-Junction Gaps	582
Testing the BSP Tree	585
Drawing Polygons Front to Back	587
First BSP Example	598
Drawing Objects in the Scene	600
Loading Maps from a File	601
Putting It All Together	607
Enhancements	608
Summary	609

Chapter 11	Collision Detection	611
	Collision Basics	611
	Object-to-Object Collisions	613
	Eliminating Tests	613
	Bounding Spheres	616
	Bounding Cylinders	618
	The Discrete Time Issue	621
	Object-to-World Collisions	622
	Bounding Boxes for Testing Against Floors	623
	Finding the BSP Leaf for a Specific Location	624
	Implementing Floor and Ceiling Height Testing	625
	Bounding Boxes for Testing Against Walls	627
	Intersection of a Line Segment with a BSP Tree	629
	The Corner Issue	634
	Implementing Object-to-World Collision Detection	635
	Basic Collision-Detection Demo	638
	Collision Handling with Sliding	639
	Object-to-Object Sliding	639
	Object-to-Wall Sliding	643
	Gravity and Sliding Up Stairs (Object-to-Floor Sliding)	646
	Make It Jump	650
	Collision Detection with Sliding Demo	651
	Enhancements	652
	Summary	652
Chapter 12	Path Finding	653
	Path-Finding Basics	654
	Some Initial Path-Finding Attempts	654
	Breadth-First Search	657
	Basics of the A* Algorithm	660
	Applying the A* Algorithm	667
	Using the A* Algorithm with a BSP Tree	668
	Portals	668
	Implementing Portals	669
	Generic Path Finding	675
	Making a PathBot	679
	Enhancing the A* Search	684
	Summary	686

Chapter 13	Artificial Intelligence	687
	AI Basics	688
	Take Away Those Godlike Powers!	689
	Seeing	689
	Hearing	692
	State Machines and Reacting	695
	Probability Machines	698
	Some Useful Random Functions	700
	Making Decisions	702
	Patterns	704
	Dodging	705
	Attacking	708
	Running Away	711
	Aiming	713
	Firing	714
	Object Spawning	716
	Putting It All Together	717
	Brains!	718
	Health and Dying	719
	Adding a Heads-Up Display	724
	Evolution	729
	Regeneration	730
	Evolving Bots	733
	Demo Enhancements	738
	Other Game AI Ideas	739
	Team AI	739
	Summary	740
Chapter 14	Game Scripting	741
	Scripting Cookbook: What You Need	742
	Implementing Touch and Release Notifications	743
	Triggers	745
	Game Object Listeners	747
	Scripting	753
	Designing the Script	755
	Embedding BeanShell	758

Delayed Events	764
Creating Delayed Events in BeanShell	769
Putting It All Together	770
Enhancements	772
Summary	774
Chapter 15 Persistence—Saving the Game	775
Game-Saving Basics	775
Using Java's Serialization API for Game State Persistence	777
Introduction to Serialization	777
Serialization: The Basics	778
Serialization: The Rules	781
Serialization: The Pitfalls	784
Overriding Default Serialization Engine Behavior	790
Creating Game Screen Snapshots	795
Creating Screen Thumbnail Images	798
Saving an Image	799
Saving Games to the Right Destination	801
Summary	803

Part III Tuning and Finishing Your Game

Chapter 16 Optimization Techniques	807
Optimization Rules	808
Profiling	809
Benchmarking	809
Using the HotSpot Profiler	809
HotSpot	815
Java-Specific Optimizations	816
Dead Code Elimination	816
Loop Invariant Hoisting	817
Common Subexpression Elimination	817
Constant Propagation	818
Loop Unrolling (Server VM Only)	818
Method Inlining	819
Optimization Tricks	820
Algorithms	820
Strength Reduction: Bit Shifting	821

Strength Reduction: Modulus	822
Strength Reduction: Multiplication	823
Strength Reduction: Exponentiation	823
More Loop Invariant Hoisting	824
Lookup Tables	825
Fixed-Point Math	828
Exceptions	829
Input/Output	829
Memory-Mapped Files	830
Memory Usage and the Garbage Collector	831
The Java Heap and Garbage Collection	831
Monitoring Garbage Collection	832
Monitoring Memory Usage	834
Tuning the Heap	839
Tuning the Garbage Collector	841
Reducing Object Creation	842
Object Reuse	842
Object Pools	843
Perceived Performance	844
Timer Resolution	844
Summary	851
Chapter 17 Creating Game Art and Sounds	853
Choosing a Look and Feel	854
Looking for Inspiration	854
Staying Consistent	855
Getting Royalty-Free Game Media	855
Working with Artists and Sound Effect Engineers	856
Tools	857
Creating Sounds	857
Sound File Formats	859
Creating Textures and Sprites	860
Graphics File Formats	861
Creating Seamless Textures	861
Creating "Alternate" Textures	863
Creating Transition Textures	864
Creating Multi-Layer Textures	866

Creating Splash Screens and HUD Graphics	867
Creating UI Graphics	867
Customizing Swing	867
Creating Your Own Fonts	869
Summary	876
Chapter 18 Game Design and the Last 10%	877
The Last 10%	878
Effects	878
The Game State Machine	879
Elements of Game Design	887
Environments	887
Story Elements	888
Game Play	890
Teaching Users How to Play	892
Creating a Map Editor	893
Debugging	895
Debugging Java2D Issues	898
Logging	899
Protecting Code	902
Game Deployment	903
Game Deployment with Java Web Start	905
Retrieving Resources from .jar Files	905
Signing .jar Files	906
Creating a JNLP File	907
Setting Up a Web Server	909
Game Deployment with Native Compilation	911
Updates and Patches	911
Bandwidth Issues	912
Getting Feedback and Beta Testing	913
Making Money	914
Putting It All Together	915
Summary	916

Chapter 19 The Future	917
How Java Evolves	917
Java Community Process	918
Bug Parade	918
The Future: Java 1.5 "Tiger"	918
Generics (JSR 14)	919
Enumerations (JSR 201)	921
Static Import (JSR 201)	922
Enhanced for Loop (JSR 201)	923
Compiler API (JSR 199)	923
Network Transfer Format (JSR 200)	924
Shared VM (JSR 121)	924
What the Java Platform Needs	925
Needed: More Options for Mouse and Keyboard Input	925
Needed: Joystick Input	926
Needed: Hardware-Accelerated Translucent Images	926
Needed: Higher Resolution Timer	926
Needed: Hardware-Accelerated Graphics and Full-Screen on Linux	927
Needed: Hardware-Accelerated 3D Included with the Runtime	927
Needed: HotSpot SIMD Optimizations	928
Desired: More Font-Smoothing Options	929
Other Possibilities	931
New Devices and the Java Games Profile (JSR 134)	932
Summary	932
Index	935