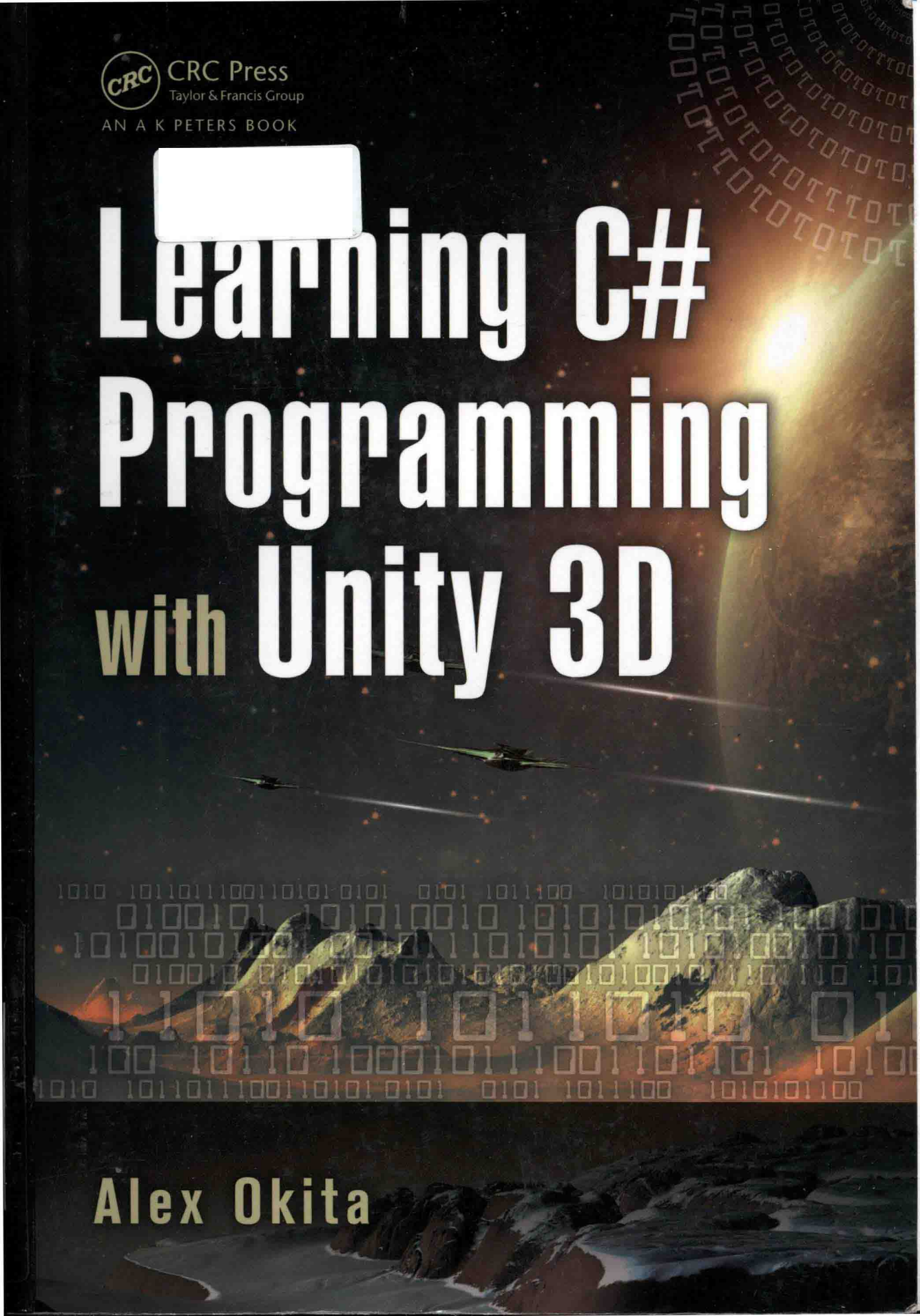


 CRC Press
Taylor & Francis Group
AN A K PETERS BOOK



Learning C# Programming with Unity 3D

Alex Okita

Learning C# Programming with Unity 3D

Alex Okita



CRC Press

Taylor & Francis Group

Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business



CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

© 2015 by Taylor & Francis Group, LLC
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works

Printed on acid-free paper
Version Date: 20140707

International Standard Book Number-13: 978-1-4665-8652-9 (Paperback)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright.com ([http://www.copyright.com/](http://www.copyright.com)) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data

Okita, Alex.

Learning C# programming with Unity 3D / Alex Okita.
pages cm

Summary: "Ideal for game programmers with no prior experience with programming languages, this book covers the basics of programming and explains how C# is used to make a game in Unity3D. Interactive examples give C# code meaning. As more complex aspects of C# are explained the interactivity of example games gains depth. Common programming tasks are taught by way of making a game. The final result is a reader who is capable of understanding how to read and apply C# in Unity3D and apply that knowledge to other development environments that use C#"-- Provided by publisher.

Includes bibliographical references and index.

ISBN 978-1-4665-8652-9 (paperback)

1. Computer games--Programming. 2. C# (Computer program language) 3. Unity (Electronic resource) 4. Three-dimensional display systems. I. Title.

QA76.76.C672O43 2014
794.8'1526--dc23

2014006696

Visit the Taylor & Francis Web site at
<http://www.taylorandfrancis.com>

and the CRC Press Web site at
<http://www.crcpress.com>

Learning C# Programming with Unity 3D

Acknowledgments

Over the past two decades I've had the opportunity to work with many amazingly talented artists and engineers.

Those who have inspired me are Edward, JCAB, Joel, and Eric from Ronin Games. Their skill and expertise kept me on my toes and motivated me to learn new skills. Also from Ronin is Ashish, with whom I was able to get started with writing.

More recently regarding my transition from artist to engineer, I thank Dave Tin Nyo, Anka GranovSkaya, David Bennett, and Jordan Patz. The original Float Hybrid group was one of the most creative collectives I've ever had the opportunity to work with. Never before has there been such a talented group of engineers under one roof—nor will there be one such hereafter.

Along the way, I thank my mother Sandi, who has always been an amazing artist. Finally, I thank April, who was the inspiration for my getting started with this book.

Contents

Acknowledgments.....	xix
----------------------	-----

1. Introduction: What This Book Is About.....	1
1.1 Why Read a Book: Why This Book May or May Not Be for You.....	1
1.1.1 Do I Need to Know Math?.....	2
1.1.2 Programming as a Form of Expression	2
1.1.3 Games as a Stage with Lights	2
1.2 Personal Information	2
1.3 A Brief History of Computer Programming: How Programming Came to Be	3
1.3.1 Mechanical Computers	3
1.3.2 Logic	3
1.3.3 Computer Science	3
1.3.4 Software	4
1.3.5 Modern Computer Language.....	4
1.3.6 The Future of Computer Languages	4
1.4 C#: A Flexible Programming Language	4
1.4.1 C# Paradigm	4
1.4.2 Unity 3D: A Brief History of Game Engines	5
1.4.3 Why Use Unity 3D to Learn?	6
1.4.4 How Does Unity 3D Use C#?.....	6
1.4.4.1 How to Tell Unity 3D What to Do.....	6
1.5 What Is Programming?.....	7
1.5.1 What Does C# Look Like?	7
1.5.2 Learning to Copy and Paste.....	8
1.6 Compiling: Turning Words into Computer Instruction.....	9
1.7 What We've Learned	9
1.8 Leveling Up.....	9
2. Before You Begin.....	11
2.1 What Will Be Covered in This Chapter	11
2.2 Downloading and Installing: It Is Free.....	13
2.3 Unity 3D Overview: What We're Looking At.....	14
2.3.1 The Main Panel.....	15
2.3.1.1 Toolbar.....	15
2.3.1.2 Hierarchy Panel	16
2.3.1.3 Scene and Game View.....	16
2.3.1.4 Inspector Panel	17
2.3.1.5 Project and Console Panel	17
2.3.2 Creating a New Project	19
2.3.3 A New Unity 3D Project.....	20
2.3.3.1 Assets Directory	20
2.3.3.2 Library Directory	21
2.3.3.3 Project Settings Directory	22
2.3.3.4 Temp Directory.....	22
2.3.4 Summary.....	23

2.4	Sample Code	23
2.4.1	Code Fragments	23
2.4.2	Are Errors Bad?	24
2.4.2.1	Compile-Time and Run-Time Errors.....	24
2.4.3	How to Follow Along.....	25
2.4.4	Summary	25
2.5	Working with C#: Game Engines.....	25
2.5.1	Getting Project Files	26
2.5.2	Creating and Assigning a New C# File: Now It's Your Turn	27
2.5.3	Naming Your New File	30
2.5.4	Using Your New File.....	31
2.5.5	Unity 3D Tools	36
2.5.6	Running Live Code	38
2.5.7	Saving a Scene	39
2.5.8	Opening a Scene	39
2.5.9	Summary	39
2.6	What We've Learned	40
2.7	Leveling Up.....	40
3.	First Steps	41
3.1	What Will Be Covered in This Chapter	41
3.2	Review.....	41
3.3	Tokens	42
3.3.1	Writing C#.....	42
3.3.2	Separator Tokens.....	43
3.3.3	Operator Tokens.....	44
3.3.4	Other Operator Tokens.....	44
3.3.5	Literals	44
3.3.6	Transitive and Non-Transitive Operations	44
3.3.7	Putting It All Together	45
3.3.8	What We've Learned	46
3.4	Statements and Expressions.....	46
3.4.1	Expressions	46
3.4.2	How Code Is Executed.....	47
3.4.3	Thinking in Algorithms.....	47
3.4.3.1	Wash, Rinse, Repeat.....	49
3.4.4	What We've Learned	49
3.5	Keywords	49
3.5.1	Class	50
3.5.2	What We've Learned	51
3.6	White Space	51
3.6.1	Pick a Flavor	55
3.6.2	What We've Learned	56
3.7	Code Blocks	56
3.7.1	What We've Learned	57
3.8	Classes.....	57
3.8.1	Objects	58
3.8.2	What We've Learned	61
3.9	Variables	61
3.9.1	Identifiers	62
3.9.2	Data.....	62

3.9.3	Declaring a Variable	62
3.9.3.1	A Basic Example	64
3.9.4	Dynamic Initialization	65
3.9.5	What We've Learned	65
3.10	Variable Names	66
3.10.1	Using Variables in Unity 3D	68
3.10.2	Variable Assignment	69
3.10.3	Putting It Together	71
3.10.4	What We've Learned	73
3.11	Types: A First Look	74
3.11.1	Value and Reference Types	74
3.11.2	What We've Learned	75
3.12	Strong Typing	75
3.12.1	Dynamic Typing	76
3.12.2	What We've Learned	76
3.13	Type Casting, Numbers	77
3.13.1	Explicit versus Implicit Casting	78
3.13.1.1	A Basic Example	78
3.13.2	What We've Learned	80
3.14	Comments	81
3.14.1	Line Numbers	83
3.14.2	Code Folding	84
3.14.3	Summary Comments	85
3.14.4	Navigating in Code	86
3.14.5	What We've Learned	87
3.15	Leveling Up: Moving On to Basics	87
4.	Basics: The Building Blocks of Code	89
4.1	What Will Be Covered in This Chapter	89
4.2	Review	89
4.3	Building Up a Game Idea: Working with What You Know	90
4.3.1	Design from Experience	90
4.3.1.1	Know Yourself	90
4.3.2	Primary Activity	91
4.3.3	Moment to Moment	91
4.3.4	Actions to Functions	91
4.3.5	Compromise	92
4.3.6	Starting with Controls	92
4.3.7	What We've Learned	93
4.4	Creating a Class	93
4.4.1	Class Declaration	94
4.4.1.1	A Basic Example	95
4.4.1.2	Value and Reference Types	96
4.4.2	Adding Data Fields	96
4.4.3	Access Modifiers and the Dot Operator	96
4.4.4	Class Scope	98
4.4.5	Class Members	99
4.4.6	What We've Learned	99
4.5	Directives	102
4.5.1	Libraries	103
4.5.2	Ambiguous NameSpaces	108
4.5.3	What We've Learned	108

4.6	Functions.....	109
4.6.1	What Are Functions?	109
4.6.2	Unity 3D Entry Points.....	110
4.6.3	Writing a Function	112
4.6.3.1	A Basic Example	112
4.6.4	More on White Space and Tabs	113
4.6.5	What We've Learned	115
4.7	Order of Operation: What Is Calculated and When	115
4.7.1	Evaluating Numbers.....	116
4.7.1.1	Math.....	116
4.7.1.2	Operator Evaluation.....	118
4.7.2	What We've Learned	120
4.8	Scope: A First Look	120
4.8.1	Class Scope	121
4.8.2	Function Scope.....	124
4.8.3	What We've Learned	126
4.9	This	126
4.9.1	A Basic Example.....	127
4.9.2	When This Is Necessary	128
4.9.3	Awkward Names.....	128
4.9.4	What We've Learned	129
4.10	Turning Ideas into Code—Part 1.....	129
4.10.1	Mouse Input	131
4.10.2	GameObject	131
4.10.3	What We've Learned	133
4.11	Logic and Operators	133
4.11.1	Booleans.....	134
4.11.2	Equality Operators.....	134
4.11.2.1	A Basic Example	135
4.11.3	If and Branching	135
4.11.3.1	Not!	137
4.11.4	Flowcharts.....	137
4.11.5	Relational Operators	138
4.11.5.1	Else	139
4.11.5.2	Else If.....	140
4.11.6	Rearranging Logic	140
4.11.6.1	Flawed Logic	143
4.11.6.2	Unreachable Code.....	144
4.11.7	Another Look at Scope	145
4.11.8	What We've Learned	147
4.12	Loops	148
4.12.1	Unary Operators.....	148
4.12.2	While.....	149
4.12.3	For.....	149
4.12.4	Do-While.....	151
4.12.5	Postfix and Prefix Notation	152
4.12.6	Using Loops	153
4.12.7	Loops within Loops	155
4.12.8	Runaway Loops.....	156
4.12.9	Breakpoints: A First Look	157
4.12.10	What We've Learned	160

4.13	Scope, Again.....	160
4.13.1	Visibility or Accessibility	160
4.13.1.1	A Basic Example	161
4.13.2	Global Scope	161
4.13.3	What We've Learned	164
4.14	Warnings versus Errors.....	164
4.14.1	Warnings	165
4.14.2	Errors	165
4.14.3	Understanding the Debugger	165
4.14.4	What We've Learned	168
4.15	Leveling Up: Fundamentals.....	168
4.15.1	Style Guides	169
5.	Fundamentals	171
5.1	What Will Be Covered in This Chapter	171
5.2	Review.....	171
5.2.1	Modular Code	173
5.3	Inheritance: A First Look	175
5.3.1	Class Members.....	176
5.3.2	Instanting.....	178
5.3.2.1	A Basic Example	180
5.3.3	Parent and Child	180
5.3.4	Object.....	182
5.3.4.1	A Type Is Not an Object	183
5.3.5	!= null.....	184
5.3.6	What We've Learned	185
5.4	Instanting.....	185
5.4.1	Class Initialization	186
5.4.2	New	187
5.4.3	Constructors	187
5.4.4	What We've Learned	190
5.5	Static	190
5.5.1	A Basic Example.....	191
5.5.2	Static Variables	192
5.5.2.1	A Basic Example	193
5.5.3	Static Functions.....	195
5.5.4	Putting It All Together	197
5.5.5	What We've Learned	201
5.6	Turning Ideas into Code—Part 2.....	201
5.6.1	Input Manager.....	206
5.6.2	What We've Learned	214
5.7	Jump Statements	214
5.7.1	Return.....	215
5.7.1.1	A Basic Example	215
5.7.2	Returning Objects	216
5.7.3	A Class Is a Type	217
5.7.4	Null Is Not Void	219
5.7.5	What We've Learned	219
5.8	Operators and Conditions	219
5.8.1	Conditional Operators && and 	220
5.8.1.1	A Basic Example	220
5.8.2	What We've Learned	223

5.9	Arrays: A First Look.....	224
5.9.1	Fixed-Sized Arrays	224
5.9.1.1	A Basic Example	224
5.9.2	Foreach.....	229
5.9.2.1	A Basic Example	229
5.9.3	Dynamic Initialization.....	229
5.9.4	Using the While Loop with Arrays.....	230
5.9.4.1	Setting Array Values	230
5.9.4.2	Getting Array Values.....	231
5.9.5	What We've Learned	231
5.10	Jump Statements: Break and Continue	232
5.10.1	A Basic Example.....	232
5.10.1.1	Continue.....	232
5.10.2	ZombieData.....	232
5.10.3	Foreach—Again.....	234
5.10.4	What We've Learned	235
5.11	Multidimensional Arrays	235
5.11.1	Columns and Rows	236
5.11.1.1	A Basic Example	236
5.11.2	A Puzzle Board	238
5.11.3	Checking Range.....	242
5.11.4	What We've Learned	244
5.12	Array List.....	245
5.12.1	A Basic Example.....	246
5.12.2	ArrayList.Contains()	248
5.12.3	Remove.....	249
5.12.4	Sort and Reverse	250
5.12.5	What We've Learned	251
5.13	Strings	251
5.13.1	Declaring a String.....	251
5.13.1.1	A Basic Example	251
5.13.2	Escape Sequences	253
5.13.3	Verbatim Strings: @.....	254
5.13.4	String Format.....	255
5.13.5	What We've Learned	255
5.14	Combining What We've Learned	255
5.14.1	Timers	255
5.14.2	Adding in Classes	261
5.14.3	What We've Learned	262
5.15	Source Version Control.....	262
5.15.1	Modern Version Control	263
5.15.2	The Repository	263
5.15.3	GitHub.....	264
5.15.4	What We've Learned	264
5.15.5	Project Files	264
5.16	Setting Up a Repository	264
5.16.1	Push.....	271
5.16.2	Gitignore	271
5.16.3	Pull	273
5.16.4	Contributors	274
5.16.5	What We've Learned	275
5.17	Leveling Up: On to the Cool Stuff.....	276

6. Intermediate.....	279
6.1 What Will Be Covered in This Chapter	279
6.2 Review.....	279
6.3 Pseudocode	280
6.3.1 Thinking It Through	280
6.3.2 Class Members.....	281
6.3.2.1 A Basic Example	281
6.3.2.2 Thinking like a Programmer.....	282
6.3.3 Return.....	285
6.3.3.1 A Basic Example	285
6.3.4 Arguments aka Args (Not Related to Pirates)	286
6.3.4.1 The Basic Example.....	286
6.3.4.2 Multiple Args.....	287
6.3.4.3 Using Args.....	288
6.3.5 Assignment Operators.....	290
6.3.5.1 A Basic Example	291
6.3.6 What We've Learned	292
6.4 Class Constructors	292
6.4.1 A Basic Example.....	293
6.4.2 What We've Learned	294
6.4.3 What We've Learned	296
6.5 Arrays Revisited.....	296
6.5.1 Using Arrays in Unity 3D	297
6.5.1.1 Starting with 0	298
6.5.1.2 Mathf	301
6.5.1.3 Time.....	301
6.5.2 Instanting with AddComponent();	302
6.5.3 Type Casting Unity 3D Objects	303
6.5.4 What We've Learned	305
6.6 Enums	305
6.6.1 Using Enums	306
6.6.2 Combining What We've Learned	308
6.6.3 What We've Learned	310
6.7 Switch.....	310
6.7.1 A Basic Example.....	311
6.7.2 Default:.....	313
6.7.3 What We've Learned	315
6.7.4 Fall Through	319
6.7.5 goto Case.....	320
6.7.6 Limitations	320
6.7.7 What We've Learned	321
6.8 Structs	321
6.8.1 Structs	322
6.8.2 Struct versus Class	322
6.8.3 Without Structs	324
6.8.4 Handling Structs	324
6.8.5 Accessing Structs	326
6.8.6 Global Access.....	327
6.8.7 What We've Learned	328
6.9 Class Data	328
6.9.1 Character Base Class	329
6.9.2 Const	330

6.9.3	Readonly	330
6.9.4	What We've Learned	331
6.10	Namespaces	331
6.10.1	A Basic Example.....	331
6.10.2	Directives in Namespaces.....	332
6.10.3	Ambiguous References	334
6.10.4	Alias Directives.....	336
6.10.5	Putting Namespaces to Work.....	337
6.10.6	Extending Namespaces	338
6.10.7	What We've Learned	339
6.11	Functions Again.....	339
6.11.1	Parameter Lists	339
6.11.1.1	A Basic Example	340
6.11.2	Side Effects	340
6.11.3	Multiple Arguments.....	341
6.11.4	Useful Parameters.....	342
6.11.4.1	The Rule of Three.....	342
6.11.5	Foreach versus For	343
6.11.6	What We've Learned	345
6.12	Unity 3D Execution Order	345
6.12.1	A Basic Example.....	346
6.12.2	Component Execution Order	349
6.12.3	What We've Learned	352
6.13	Inheritance Again	352
6.13.1	Function Overrides	352
6.13.1.1	A Basic Example	352
6.13.2	Class Inheritance.....	355
6.13.2.1	Sharing Common Attributes	355
6.13.3	Object.....	358
6.13.4	What We've Learned	359
6.14	Type Casting Again	360
6.14.1	(<Type>) versus "as"	361
6.14.2	User-Defined Type Conversion	362
6.14.3	Implicit versus Explicit Type Conversion	363
6.14.4	Break	363
6.14.5	What We've Learned	365
6.15	Working with Vectors	365
6.15.1	Vectors Are Objects.....	365
6.15.2	Stepping through MonsterGenerator	366
6.15.3	Gizmos	368
6.15.3.1	A Basic Example	369
6.15.4	Using Gizmos.....	371
6.15.4.1	Building Up Parameters	375
6.15.5	Optimizing	376
6.15.6	What We've Learned	377
6.16	goto Labels.....	377
6.16.1	A Basic Example.....	378
6.16.2	Zombie State Machine	379
6.16.3	This as a Reference to Yourself	385
6.16.4	HumanState Based on ZombieState	388
6.16.5	The Is Keyword.....	389
6.16.6	What We've Learned	392

6.17	More on Arrays.....	392
6.17.1	Length and Count.....	392
6.17.1.1	A Basic Example	392
6.17.2	Foreach: A Reminder.....	393
6.17.3	Discovery	397
6.17.4	Putting It Together	397
6.17.5	What We've Learned	405
6.18	Out Parameter.....	405
6.18.1	A Basic Example.....	406
6.18.2	Simple Sort (Bubble Sort).....	407
6.18.3	Simple Sort Proof.....	410
6.18.4	What We've Learned	410
6.19	Ref Parameter	413
6.19.1	A Basic Example.....	414
6.19.2	Code Portability Side Effects.....	415
6.19.3	What We've Learned	415
6.20	Type Casting Numbers.....	417
6.20.1	Number Types	419
6.20.2	Integers.....	419
6.20.2.1	Signed Numbers	419
6.20.3	Floating Point.....	420
6.20.4	What We've Learned	420
6.21	Types and Operators	420
6.21.1	GetType()	421
6.21.2	More Type Casting.....	422
6.21.3	Type Aliasing.....	425
6.21.4	Boxing and Unboxing	425
6.22	Operator Overloading	426
6.22.1	A Basic Example.....	426
6.22.2	Overloading *.....	428
6.22.3	Overloading <	429
6.22.4	What We've Learned	430
6.23	Controlling Inheritance.....	430
6.23.1	Sealed.....	431
6.23.1.1	A Basic Example	431
6.23.2	Abstract.....	432
6.23.2.1	A Basic Example	432
6.23.3	Abstract: Abstract	436
6.23.4	Putting This to Use	438
6.23.5	What We've Learned	441
6.24	Leveling Up.....	442
7.	Advanced.....	443
7.1	What Will Be Covered in This Chapter	443
7.2	Review.....	444
7.2.1	Moving Forward	444
7.3	MonoDevelop.....	445
7.3.1	Find in Files	445
7.3.2	Word Processors.....	447
7.3.3	Mono History	448

7.4	Function Overloading	448
7.4.1	A Closer Look at Functions	448
7.4.1.1	A Basic Example	449
7.4.2	Function Signature	450
7.4.3	Different Signatures	450
7.4.4	Putting It Together	451
7.4.5	Not Quite Recursion	452
7.4.6	DrawWord	452
7.4.7	What We've Learned	456
7.5	Accessors (or Properties)	456
7.5.1	Value	457
7.5.1.1	A Basic Example	457
7.5.2	Set Event	458
7.5.3	Read-Only Accessor	459
7.5.4	Simplification	460
7.5.5	What We've Learned	460
7.6	Base Classes: Another Look	460
7.6.1	Generalization—Base Classes	461
7.6.2	Specialization	463
7.6.3	Base	464
7.6.4	Partial	466
7.6.5	Protected, Private, and Public	471
7.6.6	What We've Learned	472
7.7	Optional Parameters	472
7.7.1	Using Optionals	474
7.7.1.1	A Basic Example	474
7.7.2	Optional Arguments	475
7.7.3	Named Parameters	476
7.7.3.1	A Basic Example	477
7.7.4	Combining What We've Learned	478
7.7.5	What We've Learned	478
7.8	Delegate Functions	479
7.8.1	Delegates	479
7.8.1.1	A Basic Example	479
7.8.2	Delegate Signatures	480
7.8.3	Stacking Delegates	481
7.8.4	Using Delegates	482
7.8.5	What We've Learned	484
7.9	Interface	484
7.9.1	Early Planning	485
7.9.1.1	A Basic Example	485
7.9.1.2	Using Accessors	486
7.9.2	Interface Methods	489
7.9.2.1	Breaking a Fixing	491
7.9.3	Multiple Interfaces	491
7.9.4	IComparer	492
7.9.5	Using IComparer	493
7.9.6	What We've Learned	494
7.10	Class Constructors Revisited	494
7.10.1	A Basic Example	495
7.10.2	When to Create a New Class	496
7.10.2.1	Add in a Private zposition Offset	499