The background of the book cover is a 3D-rendered scene from a game. It depicts a futuristic, industrial environment with a large, domed structure on the left and several large, mechanical vehicles or tanks in the foreground. The sky is a dramatic, orange-hued sunset or sunrise with heavy, dark clouds. A large, dark circular area in the upper right corner contains the title and author information.

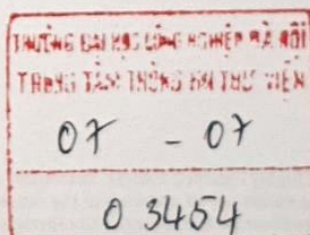
C# GAME PROGRAMMING COOKBOOK for **UNITY 3D**

Jeff W. Murray

 **CRC Press**
Taylor & Francis Group
AN A K PETERS BOOK

C# GAME PROGRAMMING COOKBOOK for **UNITY 3D**

Jeff W. Murray



CRC Press

Taylor & Francis Group

Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an Informa business
AN A K PETERS BOOK

Contents

Acknowledgments	xiii
Introduction	xv
1. Making Games the Modular Way	1
1.1 Important Programming Concepts.....	2
1.1.1 Manager and Controller Scripts.....	2
1.1.2 Script Communication.....	3
1.1.3 Using the Singleton Pattern in Unity.....	5
1.1.4 Inheritance.....	6
1.1.5 Where to Now?	8
2. Building the Core Game Framework	9
2.1 Controllers and Managers.....	11
2.1.1 Controllers.....	11
2.1.2 Managers	11
2.2 Building the Core Framework Scripts.....	11
2.2.1 BaseGameController.cs	12
2.2.1.1 Script Breakdown	14

2.2.2	Scene Manager	17
2.2.2.1	Script Breakdown	17
2.2.3	ExtendedCustomMonoBehavior.cs	19
2.2.4	BaseUserManager.cs	20
2.2.4.1	Script Breakdown	22
2.2.5	BasePlayerManager.cs	22
2.2.5.1	Script Breakdown	23
2.2.6	BaseInputController.cs	24
2.2.6.1	Script Breakdown	26
3. Player Structure		29
3.1	Game-Specific Player Controller	31
3.2	Dealing with Input	32
3.3	Player Manager	35
3.3.1	Script Breakdown	36
3.4	User Data Manager (Dealing with Player Stats Such as Health, Lives, etc.)	37
3.4.1	Script Breakdown	39
4. Recipes: Common Components		41
4.1	Introduction	41
4.2	The Timer Class	43
4.2.1	Script Breakdown	45
4.3	Spawn Scripts	48
4.3.1	A Simple Spawn Controller	49
4.3.1.1	Script Breakdown	52
4.3.2	Trigger Spawner	56
4.3.3	Path Spawner	57
4.3.3.1	Script Breakdown	61
4.4	Set Gravity	66
4.5	Pretend Friction—Friction Simulation to Prevent Slipping Around	66
4.5.1	Script Breakdown	68
4.6	Cameras	68
4.6.1	Third-Person Camera	69
4.6.1.1	Script Breakdown	71
4.6.2	Top-Down Camera	74
4.6.2.1	Script Breakdown	74
4.7	Input Scripts	75
4.7.1	Mouse Input	75
4.7.1.1	Script Breakdown	76
4.7.2	Single Axis Keyboard Input	78
4.8	Automatic Self-Destruction Script	79
4.8.1	Script Breakdown	79
4.9	Automatic Object Spinner	79
4.9.1	Script Breakdown	80

4.10	Scene Manager	81
4.10.1	Script Breakdown	82
5.	Building Player Movement Controllers	85
5.1	Shoot 'Em Up Spaceship	85
5.2	Humanoid Character	91
5.2.1	Script Breakdown	96
5.3	Wheeled Vehicle	106
5.3.1	Script Breakdown	109
5.3.2	Wheel Alignment	114
5.3.3	Script Breakdown	116
6.	Weapon Systems	121
6.1	Building the Scripts	122
6.1.1	BaseWeaponController.cs	122
6.1.1.1	Script Breakdown	127
6.1.2	BaseWeaponScript.cs	134
6.1.2.1	Script Breakdown	138
7.	Recipe: Waypoints Manager	143
7.1	Waypoint System	143
8.	Recipe: Sound Manager	157
8.1	The Sound Controller	158
8.1.1	Script Breakdown	160
8.2	The Music Player	163
8.2.1	Script Breakdown	165
8.3	Adding Sound to the Weapons	167
9.	AI Manager	169
9.1	The AI State Control Script	171
9.2	The Base AI Control Script	172
9.2.1	Script Breakdown	185
9.3	Adding Weapon Control to the AI Controller	206
9.3.1	Script Breakdown	210
10.	Menus and User Interface	215
10.1	The Main Menu	215
10.1.1	Script Breakdown	223
10.2	In-Game User Interface	231

11. Dish: Lazer Blast Survival	233
11.1 Main Menu Scene.....	235
11.2 Main Game Scene.....	236
11.3 Prefabs.....	237
11.4 Ingredients.....	238
11.5 Game Controller.....	239
11.5.1 Script Breakdown.....	243
11.6 Player Controller.....	250
11.6.1 Script Breakdown.....	253
11.7 Enemies.....	259
11.7.1 Script Breakdown.....	260
11.8 Wave Spawning and Control.....	261
11.8.1 Script Breakdown.....	263
11.9 Wave Properties.....	265
11.10 Weapons and Projectiles.....	266
11.11 User Interface.....	266
11.11.1 Script Breakdown.....	267
12. Dish: Metal Vehicle Doom	271
12.1 Main Menu Scene.....	272
12.2 Main Game Scene.....	272
12.2.1 Prefabs.....	275
12.3 Ingredients.....	275
12.3.1 Game Controller.....	276
12.3.1.1 Script Breakdown.....	282
12.3.2 Race Controller.....	291
12.3.2.1 Script Breakdown.....	297
12.3.3 Global Race Manager.....	306
12.3.3.1 Script Breakdown.....	311
12.3.4 Vehicle/Custom Player Control.....	318
12.3.4.1 Script Breakdown.....	327
12.3.5 User Interface.....	344
13. Dish: Making the Game Tank Battle	345
13.1 Main Game Scene.....	347
13.2 Prefabs.....	349
13.3 Ingredients.....	349
13.4 Game Controller.....	350
13.4.1 Script Breakdown.....	356
13.5 Battle Controller.....	361
13.5.1 Script Breakdown.....	363
13.6 Global Battle Manager.....	364
13.6.1 Script Breakdown.....	368
13.7 Players.....	373
13.7.1 Script Breakdown.....	382

13.8	AI Chasing with SetAIChaseTargetBasedOnTag.cs	383
13.8.1	Script Breakdown	385
14.	Dish: Making the Game <i>Interstellar Paranoids</i>	389
14.1	Main Menu	392
14.2	Game Scenes	392
14.3	Prefabs	393
14.3.1	Ingredients	394
14.3.2	Game Controller	395
14.3.2.1	Script Breakdown	401
14.3.3	Player Spaceship	411
14.3.3.1	Script Breakdown	415
14.3.4	Enemies	423
14.3.4.1	Script Breakdown	424
14.3.5	Waypoint Follower	426
14.3.5.1	Script Breakdown	427
	Final Note	429
	Index	431

Introduction

As I was starting out as a game developer, as a self-taught programmer my skills took a while to reach a level where I could achieve what I wanted. Sometimes I wanted to do things that I just didn't have yet the technical skills to achieve. Now and again, software packages came along that could either help me in my quest to make games or even make full games for me; complete game systems such as the Shoot 'Em-Up Construction Kit (aka SEUCK) from Sensible Software, Gary Kitchen's GameMaker, or The Quill Adventure System could bring to life the kinds of games that went way beyond anything that my limited programming skills could ever dream of building.

The downside to using game creation software was that it was tailored to create games within their chosen specific genre. If I wanted to do something outside of the limitations of the software, the source code was inaccessible and there was no way to extend or modify it. When that happened, I longed for a modular code-based system that I could plug together to create different types of games but modify parts of it without having to spend a lot of time learning how the entire system internals work—building block game development that I could actually script and modify if I needed to.

After completing my first book, *Game Development for iOS with Unity3D*, I wanted to follow up by applying a modular style of game building to Unity3D that would provide readers with a highly flexible framework to create just about any kind of game by “plugging in” the different script components. My intention was to make a more technical second book, based on C# programming, that would offer extensibility in any direction a developer might require. In essence, what you are holding in your hands right now is a cookbook

for game development that has a highly flexible core framework for just about any type of game.

A lot of the work I put in at the start of writing this book was in designing a framework that not only made sense in the context of Unity but also could easily cope with the demands of different genres.

Prerequisites

You can get up and running with the required software for the grand total of zero dollars. Everything you need can be downloaded free of charge with no catches. You may want to consider an upgrade to Unity Pro at some point in the future, to take advantage of some of its advanced features, but to get started all you need to do is grab the free version from the Unity website.

Unity Free or Unity Pro (available from the Unity store at <http://www.unity3d.com>)

Unity Free is completely free for anyone or any company making less than \$100,000 per year—it may be downloaded for no charge at all, and you don't even need a credit card. It's a really sweet deal! We are talking about a fully functional game engine, ready to make 3D or 2D games that may be sold commercially or otherwise. There are no royalties to pay, either.

Unity Pro adds a whole host of professional functionality to the engine, such as render culling and profiling. If you are a company with more than \$100,000 per year of turnover, you will need a Pro license, but if you find that Unity Free doesn't pack quite enough power, you may also want to consider going Pro. You can arrange a free trial of the Pro version right from the Unity website to try before you buy. If the trial licence runs out before you feel you know enough to make a purchase, contact Unity about extending it and they are usually very friendly and helpful about it (just don't try using a trial license for 6 months at a time, as they may just figure it out!).

C# programming knowledge

Again, to reiterate this very important point, this is *not* a book about learning how to program. You will need to know some C#, and there are a number of other books out there for that purpose, even if I have tried to make the examples as simple as possible! This book is about making games, not about learning to program.

What This Book Doesn't Cover

This is not a book about programming and it is not a book about the right or wrong way to do things. We assume that the reader has some experience with the C# programming language. I am a self-taught programmer, and I understand that there may well be better ways to do things.

This is a book about concepts, and it is inevitable that there will be better methods for achieving some of the same goals. The techniques and concepts offered in this book are meant to provide solid foundation, not to be the final word on any subject. It is the author's intention that, as you gain your own experiences in game development, you make your own rules and draw your own conclusions.

Additional material is available from the CRC Press Web site: <http://www.crcpress.com/product/isbn/9781466581401>.