

Preface

Introduction

If you are reading this, then we assume that you are interested in learning how to develop your own computer games. If you do not yet know how to program, don't worry! Developing computer games is actually a perfect way to learn how to program in modern programming languages. This book will teach you how to program in C# without requiring any previous programming experience. It does so through the creation of computer games.

In our opinion, C# is the language of choice to learn how to program. The language is very well structured and avoids some of the problems of Java. It is also a more modern language. Moving from C# to C++, which is still the most important programming language in the game industry, is relatively easy. Also, for C# there is an excellent free programming environment available, Visual Studio C# Express Edition, which is used in this book.

Contrary to most programming books, we do not organize the book according to programming language paradigms, but instead we use the structure and elements of computer games as a framework. For instance, there are chapters on dealing with player input, game objects, game worlds, game states, levels, animation, physics, and intelligence.

While reading this book, you will create four games. We have chosen different types of games to show the various aspects of game development. We start with a simple shooting game, we move on to puzzle games consisting of multiple levels, and we conclude the book by developing a full-fledged platform game with animation, game physics, and intelligent enemies.

This book is not a cookbook. The book provides a thorough introduction to C# and object-oriented programming, organized by the structure of games. We introduce important aspects of programming in general, such as an overview of different programming paradigms, syntax diagrams, collections, exception handling, and more. We will also discuss various aspects of software architecture within a context of game development. By doing that, we propose a framework for managing levels, game states, as well as a hierarchy of self-sufficient game objects that together form

an interactive game world. Furthermore, we will show a number of commonly used techniques in games, such as drawing layers of sprites, rotating, scaling and animating sprites, showing a heads-up display (HUD), dealing with physics, handling interaction between game objects, and creating nice visual effects such as snow or glitters.

Throughout the book, you will find text in gray boxes. These boxes contain tips and tricks for designing parts of your game, such as adding tutorial levels, but sometimes they also discuss a particular programming issue, such as dealing with static variables or designing game engine code.

Required Materials and Tools

Along with this book, we supply various materials. All the example programs used in this book are available as Visual Studio projects, which you can open, edit, compile, and run yourself. Furthermore, we supply a set of game assets (sprites and sounds), which are used by all the examples. Next to the example programs, we also provide detailed instruction on how to download and install the necessary tools. We have created a website where you can download all the materials. The URL of this website is <http://www.csharpprogramminggames.com>.

In order to develop computer games, a few tools need to be installed on your computer. The main tool that you are going to need is the XNA Game Studio software, in combination with a development environment called *Visual Studio Express 2010*, created by Microsoft. On the accompanying website, you can find detailed instructions on how to obtain and install these tools. The Visual Studio Express 2010 environment is freely available and compatible with the latest XNA Game Studio version (4.0 when this book was printed).

Using this Book as a Basis for a Programming Course

This book is geared toward being used as a basis for a game-oriented programming course. Each part in this book is concluded by exercises and challenges. Solutions to the exercises are available through the accompanying website. The challenges are generally more complex programming exercises. These challenges can serve as practical assignments for students following the programming course. On the accompanying website, a number of additional challenges are available that can be used as a basis for practical assignments as well.

By following the structure of the book throughout the course, the students will be introduced to all the main aspects of programming in an object-oriented language. Supplementary materials for organizing such a course are available on the accompanying website. A sample schedule of a course consisting of 15 sessions with three practical assignments is given as follows:

	Topic	Chapters	Exercises, deadlines
1	Introduction	1, 2	Exercises part I
2	Game loop, types, variables	3, 4	Exercises part I
3	Player input, if , booleans	5, 6	Exercises part I
4	Classes, methods, objects	7, 8	Exercises part II
5	for , while , randomness	8, 9	Hand in practical assignment I
6	Inheritance	10, 11	Exercises part II
7	Collections, arrays, interfaces	12	Exercises part III
8	Grids, game worlds	13, 14, 15	Exercises part III
9	Time in games, recursion	16, 17, 18	Exercises part III
10	Sprite sheets, game states	19, 20, 21	Hand in practical assignment II
11	Abstract classes, switch , file I/O	21, 22	Exercises part IV
12	Libraries, game structure	23, 24, 25	Exercises part IV
13	Animation, game physics	26, 27	Exercises part V
14	Enemies, exceptions	28, 29, 30	Exercises part V
15	General questions	all	Hand in practical assignment III

Utrecht, The Netherlands

Arjan Egges
 Jeroen D. Fokker
 Mark H. Overmars



<http://www.springer.com/978-3-642-36579-9>

Learning C# by Programming Games
Egges, A.; Fokker, J.D.; Overmars, M.H.
2013, XXII, 443 p., Hardcover
ISBN: 978-3-642-36579-9