

Extracted from:

Learn Game Programming with Ruby

Bring Your Ideas to Life with Gosu

This PDF file contains pages extracted from *Learn Game Programming with Ruby*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragprog.com>.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printed versions; the content is otherwise identical.

Copyright © 2015 The Pragmatic Programmers, LLC.

All rights reserved.

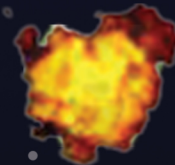
No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina

The
Pragmatic
Programmers

Learn Game Programming with Ruby



Bring Your Ideas to Life with Gosu

Credits

Mark Sobkowicz

foreword by Julian Rashke,
Lead Gosu developer

Learn Game Programming with Ruby

Bring Your Ideas to Life with Gosu

Mark Sobkowicz

The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina



Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf, PragProg and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Our Pragmatic courses, workshops, and other products can help you and your team create better software and have more fun. For more information, as well as the latest Pragmatic titles, please visit us at <https://pragprog.com>.

The team that produced this book includes:

Brian P. Hogan (editor)
Potomac Indexing, LLC (index)
Cathleen Small; Liz Welch (copyedit)
Dave Thomas (layout)
Janet Furlow (producer)
Ellie Callahan (support)

For international rights, please contact rights@pragprog.com.

Copyright © 2015 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

Printed in the United States of America.

ISBN-13: 978-1-68050-073-8

Encoded using the finest acid-free high-entropy binary digits.

Book version: P1.0—September 2015

Introduction

Hello and welcome. This is a book about making games with and for your computer—games like some of the ones you love on your computer, your phone, or your game console. By working your way through this book, you'll make four games, each of a different type. You'll learn how to open a window on your computer screen and then fill it with moving pictures. You'll make those images interact with each other, and you'll control them with your computer's mouse and keys.

The games we make will feel pretty familiar to you. They aren't copies of other games, but they do use familiar patterns and principles that you'll be able to use to make your own games. The techniques you'll learn for making a spaceship fly around by pressing keys on the keyboard can also help you move a chicken across a road full of traffic. Each chapter in this book is based on a few different elements of game creation, and each element is applicable to a wide range of games.

The goal of this book is to help you bring your own ideas to life. When you've learned these elements of game development, you'll be able to take a game you've imagined and create that game so that it runs on your computer—and also on your friends' computers.

Ruby and Gosu

Along the way, you'll level up your programming skills. Becoming a better programmer will help you make great games, and making games will help you improve your programming skills. To make the games in this book you'll use the Ruby programming language, along with a game library called Gosu. Ruby is a great language both for learning to program and for making games.

It has “an elegant syntax that is natural to read and easy to write.”¹ The Ruby language focuses on *objects*, and this makes it a great fit for creating games, as you’ll see as you work through this book.

The Gosu game library will provide the structure for your games, while leaving their design and content completely up to you. It gives you the tools you need to place images on the screen, move them around, and play sounds to spice up your game. At the same time, it doesn’t do anything you won’t understand. You and your code will always be in control of what’s happening in your game window. Gosu is also a great springboard to other platforms. In particular, working with SpriteKit, Apple’s framework for making two-dimensional games for iOS and OS X, feels like a natural step up from writing games with Gosu.

Ruby and Gosu are free, open source software that work well on both Windows and Mac OS X computers. You can download everything you need, so it’s easy to get started, and perhaps you’ll find other people willing to learn game programming along with you in your school or town. There is an online community dedicated to game programming with Ruby and Gosu, with a showcase where people share their games and a forum where they ask and answer questions.² Many people in this community have shared both finished games and the *code* for those games, and these can be great sources for learning and inspiration.

What You’ll Need

First, you’ll need a computer. It can be a Mac running OS X 10.9 or later, or it can be a Windows computer running Windows 7 or later. You need to be comfortable with the file system on your computer, so you can save files where you want them and organize them into folders.

To get the most out of this book, you’ll need a little programming experience. If you have already used Ruby, you’re ready to go. If you have experience with a different programming language, you might want to pick up a book such as [Learn to Program \[Pin09\]](#) or [Programming Ruby 1.9 & 2.0 \[FH13\]](#) and learn a little Ruby syntax before you start on the games in this book.

The Road Ahead

As you go through this book, we’ll be making some games together. The games follow a progression, and each chapter assumes that you have worked through

1. <https://www.ruby-lang.org/en/>

2. <http://www.libgosu.org/>

the preceding chapters. Here is a summary of what you'll be learning in each chapter.

- In [Chapter 2, *Get Ready*, on page ?](#), you'll set up your computer to use Ruby and Gosu to write games.
- In [Chapter 3, *Creating Your First Game*, on page ?](#), you'll make a simple game. You'll learn how a Gosu game is organized and how to use Gosu to open a window on your computer, fill it with pictures, and move those pictures around.
- In [Chapter 4, *Creating a Sprite-Based Game*, on page ?](#), you'll begin a new game, Sector Five, in which a player controls a spaceship that shoots down enemy invaders. Each thing you draw on the screen will be a *sprite*, and you'll learn to create a class for each type of sprite in the game.
- In [Chapter 5, *Managing Lots of Sprites*, on page ?](#), you'll learn how to manage many sprites by organizing them with arrays. By iterating through these arrays, you'll be able to handle the movement and interactions of many sprites in the window at once.
- In [Chapter 6, *Adding Scenes and Sounds*, on page ?](#), you'll break your game into multiple scenes, giving it a start scene with some instructions and an end scene with credits. You'll learn how to add music and sound effects to finish Sector Five.
- In [Chapter 7, *Creating a Puzzle Game*, on page ?](#), you'll create a puzzle game called Twelve. This game focuses on user interaction, and you'll learn how to write code to implement the rules of the game.
- In [Chapter 8, *Making a Platformer Game with Physics*, on page ?](#), you'll use a physics engine to make objects move naturally. In *Escape*, a hero will jump between platforms, dodging boulders that fall, spin, and bounce.
- In [Chapter 9, *Making a Side-Scrolling Game*, on page ?](#), you'll learn how to make your platformer game scroll, using a camera object to follow the hero's motion. This will allow you to have a game field that is bigger than your screen.
- In [Chapter 10, *Package and Share Your Game*, on page ?](#), you'll learn to package up a game into a single executable so you can share it with your friends.

When we're finished, you'll have a better understanding of how games are put together, as well as some new programming tools in your toolbox. You'll be ready to take your own ideas and turn them into games.

Bumps in the Road

Whether you're following a tutorial or writing your own code, it can be frustrating when things don't work. This book tries to anticipate some of these situations and give you some strategies to deal with them. At these places in the tutorial, you'll find a section with some ideas for how to solve particular problems.

What If It Doesn't Work?

We'll explore different sorts of problems that can occur and look at ways to solve them. You'll learn how to interpret some common errors and look at ways to find answers on the Internet.

Writing games is fun, and hopefully the rewards of making your programs work will outweigh the frustration you feel when they don't. Sometimes the answer will come to you only after you've walked away from the computer for a little while. Stick with it! Persistence is one of the most important assets a programmer can have.

What's Next

Before you can actually start making games, you'll need to install a few things on your computer. The next chapter will explain what you need and will take you step by step through getting ready.