

Foundation

Game Design

with ActionScript 3.0

SECOND EDITION

Rex van der Spuy

friendsof 
an Apress® company

*Learn the skills to design any conceivable
2D game using Flash and ActionScript*

Foundation Game Design

with ActionScript 3.0

Second Edition

Rex van der Spuy



Copyright © 2012 by Rex van der Spuy

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher

ISBN-13 (pbk): 978-1-4302-3993-2

ISBN-13 (electronic): 978-1-4302-3994-9

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logos, or image we use the names, logos, or images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com.

For information on translations, please e-mail rights@apress.com or visit www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales—eBook Licensing web page at www.apress.com/bulk-sales.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in the work.

Any source code or other supplementary materials referenced by the author in this text is available to readers at www.apress.com. For detailed information about how to locate your book's source code, go to www.apress.com/source-code/.

Credits

President and Publisher: Paul Manning
Copy Editor: Mary Behr

Lead Editor: Ben Renow-Clarke
Composer: Apress (Brigid Duffy)

Technical Reviewers: Joshua Freenev
Production Editor: Brigid Duffy

Editorial Board:

Steve Anglin, Mark Beckner, Ewan Buckingham, Gary Cornell, Morgan Ertel, Jonathan Gennick, Jonathan Hassell, Robert Hutchinson, Michelle Lowman, James Markham, Matthew Moodie, Jeff Olson, Jeffrey Pepper, Douglas Pundick, Ben Renow-Clarke, Dominic Shakeshaft, Gwenan Spearing, Matt Wade, Tom Welsh

Indexer:

BIM Indexing & Proofreading Services

Artist:

SPi Global

Cover Image Artist:

Corné van Dooren

Coordinating Editor: Kelly Moritz

Cover Designer: Anna Ishchenko

*Dedicated to my father, Mani van der Spuy,
without whom this book would not have been written.
Thanks, Dad, for always believing in me.*

Contents

About the Author

About the Technical Reviewer

About the Cover Image Designer

Acknowledgments

Introduction

Chapter 1: Programming Foundations: How To Make A Video Game

Basic requirements

1. A computer
 2. Graphic design software
 3. Computer programming software
- Adobe Flash Professional

Things you need to know

And the things you don't need to know

It's all about programming

Programming? But I'm terrible at math!

I already know how to program!

What kind of games can I make?

Learning new terms

Laying the foundation

Using Flash Builder

Setting up the work environment in Flash Builder

Writing your first program

ActionScript files and the .as file extension

I'll take that to go!

A little magic with sprites

Don't skip class!

Using the constructor method

Aligning code

What's your directive?

Adding comments to your code

Running your program and compiling the SWF file

What happened when you compiled the program

It didn't work?

Some common error messages

Confirm your program with the original source files

More about Flash Builder

Editing window quirks and features

Perspectives

A little more about AS3.0 and the Flash Player

Naming conventions

Summary

Chapter 2: Making Game Graphics

Create a game world background

Create a new file in Photoshop

Draw the sky

Drawing hills

Drawing with the pen tool

Duplicating, modifying, and arranging layers

Bevel and emboss

Making boxes with the rectangular selection tool

Making a game character

Create a new file

Draw your character

Making buttons

Create a new file

Draw the button

Using the type tool

Making more buttons

Save your work and make the PNG files

Checking your work

More about Photoshop

Summary

Chapter 3: Programming Objects

But I'm a bit scared of programming!

Loading and displaying images

Setting up a new AS3.0 project

Understanding the setup file

Set up the Flash Player with a metadata tag

Loading and displaying the background.png image

How did that work?

Variables

Variable types

Creating empty boxes

Creating instances and objects

Displaying the image on the stage

Understanding the code structure

Positioning Sprites on the stage

Programming buttons

Understanding dot notation

Methods

Using method calls

Using function definitions

Using a basic method

Understanding events and event listeners

Making your buttons work

Controlling Sprite objects with properties

Going up and down

Moving incrementally

Tracing the output

Using increment and decrement operators

Limiting movement

Making it bigger and smaller

Vanishing!

Spinning around

More properties?

Centering Sprites

Code summary

Summary

Chapter 4: Decision Making

Using text

Setting up the project

The classes you have to import

Creating TextFormat and TextField objects

Setting the format of text

Configuring and displaying the output text

Configuring and displaying the input field

Building a simple guessing game

- Understanding the program structure

Learning more about variables

- Listening for key presses

Making decisions

- Displaying the game status

Using postfix operators to change variable values by one

- Tying up strings

- Hey, why use the gameStatus variable, anyway?

- Using uint vs. int variables

- Winning and losing

- Modular programming with methods

Polishing up

- Tackling random numbers

- Disabling the Enter key

Making really nice buttons

- Understanding the concepts

- Loading the images and displaying the button

- Understanding the mouse events

Adding the button to the number guessing game

- Disabling the button at the end of the game

- Managing complex code

Project extensions

- Make a Play again button

- Tracking guesses

- Adding a visual display

- Turning the tables

A quick guide to embedding fonts

Summary

Chapter 5: Controlling A Player Character

Controlling a player character with the keyboard

- Controlling with the keyboard'the wrong way!

- Controlling the keyboard'the right way!

- Moving with velocity

- Using the new keyDownHandler

- Using the keyUpHandler

- Using the enterFrameHandler

- Setting stage boundaries

Blocking movement at the stage edges

Screen wrapping

Embedding images

Scrolling

Looking at the code

Better scrolling

Even better scrolling

Taking it further

Parallax scrolling

Summary

Chapter 6: Bumping Into Things

Basic collision detection with hitTestObject

Using hitTestObject

Triggering a change of state

Reducing a health meter

Using scaleX to scale the meter based on a percentage

Updating a score

Picking up and dropping objects

The bad news about hitTestObject

Detecting collisions with the bounding box

Learning to live with it

Creating subobjects

Advanced collision detection strategies

Working with vector based collision detection

Preventing objects from overlapping

Programming with the Collision class

Another look at methods, arguments, and parameters

Taking a look at the Collision.block method

Summary

Chapter 7: Making Games

Making the game graphics

Making the textured box

Making the bomb

Making a game character

Designing the playing field

Making the Game Over graphic

Programming the game

Dealing with many objects

What is a class, really?

Looking at the TimeBombPanic application class

Adding objects to the game

Stage boundaries

Collisions between the character and the boxes

Defusing the bombs

Using a timer

Ending the game

Creating a scrolling game environment

Preparing your artwork for scrolling

Adding objects to the game

Scrolling and stage boundaries

Summary

Chapter 8: Making Bigger Games

Understanding the game structure

Using private variables and methods

Programming the game

Giving LevelOne access to the stage

Making sure that LevelOne is actually on the stage

The LevelOne class

Programming the game character

Programming the monsters

Programming the star weapon

Checking for the end of the level

Ending the level

Using removeChild to completely get rid of objects

Communicating between classes by dispatching events

Event bubbling

Finishing off level one

Firing stars in four directions in level two

More Monster Mayhem!

Moving objects in a scrolling game world

Intelligent monsters

Obstacles and pathfinding

A little more about game structure

Summary

Chapter 9: Physics, Loops, Arrays, and Sounds

Natural motion using physics

A game object class for playing with physics

Acceleration

Friction

Bouncing

Gravity

Jumping

Adding sounds to games

Integrating sound effects in a game

Looping music and setting volume and pan levels

Working with many objects

Using arrays

Making many boxes

Collisions with many objects

Case studies

Summary

Chapter 10: Advanced Object And Character Control

Moving objects with the mouse

Fixing an object to the mouse's position

Moving an object with easing

Case study: Firing projectiles in all directions

Making the game objects

Rotating the wand around the fairy

Firing stars in 360 degrees

Advanced player control systems

Enemy AI systems

Running away from the player

Rotating and shooting toward the mouse

Using a timer to fire bullets

Case study: Killer Bee Pandemonium!

Flash animation and publishing your game

Index

ABOUT THE AUTHOR

Rex van der Spuy is a freelance interactive media designer specializing in Flash/AS3.0 game design, interface design, and ActionScript programming. He currently divides his time between Canada, India, and South Africa. When not coding video games, he wanders around the world playing the sitar.

ABOUT THE TECHNICAL REVIEWER

Josh Freeney is currently an instructor for the Digital Animation and Game Design program at Ferris State University in Grand Rapids, Michigan. He teaches Flash game development classes focused on rapid agile production with maximum reusability, and he has spoken at the Michigan Flash Festival. He likes board games, hiking, sleeping in, and anything Lego. He continues to consult with anyone anywhere who has a Flash problem that needs fixing.

ABOUT THE COVER IMAGE DESIGNER



Corné van Dooren designed the front cover image for this book. After taking a break from friends of ED to create a new design for the Foundation series, he worked at combining technological and organic forms, with the results now appearing on the cover of this and other books.

Corné spent his childhood drawing on everything at hand and then began exploring the infinite world of multimedia—and his journey of discovery hasn't stopped since. His mantra has always been “the only limit to multimedia is the imagination,” a saying that keeps him moving forward constantly.

Corné works for many international clients, writes features for multimedia magazines, reviews and tests software, authors multimedia studies, and works on many other friends of ED books. If you like Corné's work, be sure to check out his chapter in *New Masters of Photoshop, Volume 2* (friends of ED, 2004). You can see more of his work (and contact him) at his web site www.cornevandooren.com.

ACKNOWLEDGMENTS

My unending gratitude to Ben, Josh, and Kelly for the vision, dedication, hard work, and encouragement that made this book a reality.

Thanks to Preston and Rahle at the world's most innovative interactive media company, PixelProject, for hosting my sojourn in Cape Town, and for all your inspiration and friendship.

This book was written in remote areas of the Himalayas with scant access to the Internet (or even electricity). I wish to send out a big thank you to the people of Ladakh, India and Nepal for their incredible generosity and graciousness while I was working on this project.

(And, for the one person out there who might find this noteworthy: Yes, it is possible to walk 9 days to the Annapurna glacier in flip-flops and program a video game 1500 meters above your laptop's 3000 meter maximum specified operating altitude.)

INTRODUCTION

So you want to make a video game. Where do you start? What do you need to learn? To whom can you look for help?

If you've ever asked any of these questions, you know how difficult it is to find the answers. I asked myself these same questions many years ago in a little village outside of Bangalore, India, where I was teaching programming and interactive media at an international school. All my students were playing games and they all wanted to create games, but there were no comprehensive books or online resources available on how to do this.

A little bit of research turned up something surprising: not only did basic game design require relatively little programming knowledge but the same set of techniques could also be used over and over again in different contexts to create completely different kinds of games. It was fun to do, the results were immediate, and it was a great creative outlet. The result of this research was an in-house textbook on game design that formed the basis of three high school—level courses and inspired the writing of this book.

That was back in the now almost prehistoric days of Flash 4 and 5, when the ActionScript programming language was still in version 1.0 and Flash had some wonderful built-in interactive tutorials that guided new users every step of the way. It seemed as if everyone was a beginner in those days, so it was relatively easy to find books and tutorials that assumed the reader had no background knowledge.

ActionScript is now in version 3.0, and things are not so easy. The ActionScript language has become much more powerful but also much more complex. Many of the resources that you'll find for AS3 are focused on that complexity, and it's harder and harder for beginners with little previous programming experience to get a comprehensive foothold to start learning. The irony of all this is that AS3.0 actually makes it much *easier* to build games than in the days of AS1.0. This book strips away the apparent complexity of AS3.0 and gets to the core of what you need to know to make games. It's fun and easy, and anyone can do it.

Game design is a fantastic thing, and what you're about to learn is as close to creating magic as the real world allows. Hang on for a wild ride—you'll be amazed by what you'll start producing very quickly.

About the second edition

The first edition of this book, *Foundation Game Design with Flash*, was all about how to make games using Flash Professional software. It used to be that using Flash Professional was the only way to make Flash games, and even when other technologies gradually became available, it was still the best way. In the almost three years since that first edition was published, AS3.0 has taken on a life of its own—it's become the best cross-platform development tool for making games for the Web and for desktop and mobile devices. It's also the easiest to learn for beginners, and it has the most comprehensive and best integrated set of tools for making games than any other single technology. This edition shows you how to build games using pure AS3.0 code and whole host of complementary technologies for the

greatest flexibility in the development process. Making games with AS3.0 has never been more fun and the potential audience that you can reach with your games has never been wider.

Layout conventions

To keep this book as clear and easy to follow as possible, the following text conventions are used throughout:

Important words or concepts are normally highlighted on the first appearance in *italic type*.

Code is presented in *fixed-width font*.

New or changed code is normally presented in **bold fixed-width font**.

Pseudo-code and variable input are written in *italic fixed-width font*.

Menu commands are written in the form Menu > Submenu > Submenu.

When I want to draw your attention to something, I've highlighted it like this:

Ahem, don't say I didn't warn you.

Sometimes code won't fit on a single line in a book. When this happens, I use an arrow like this: ➡.

This is a very, very long section of code that should be written on ➡
the same line without a break.



Chapter 1

Programming Foundations: How To Make A Video Game

Congratulations on picking up this book! Video game design is one of the most interesting and creative things you can do with a computer. You're about to embark on a remarkable journey, and this book will guide you every step of the way.

So how do you make a video game? Although there are probably as many ways to make games as there are readers of this book, a good place to start is with a technology called **Flash** and **ActionScript**. Not only is it easy to learn how to make games with these technologies, but you can also use them to produce games of great complexity and professional quality if you have the time and imagination. You can also publish your games for desktop computers, laptops, the Internet, mobile phones, and tablets without having to learn any additional skills. And all the skills you'll acquire while making games with Flash and ActionScript can be directly applied to game design using other technologies if you want to take your learning further.

Learning game design with Flash and ActionScript is really a two- step process.

- You need to learn how to make graphics and illustrations for your games. This is commonly done with graphic design software such as Adobe Photoshop, Illustrator, Flash Professional, or open source software such as GIMP (GNU Image Manipulation Program).
- You also need to learn a **programming language** called **ActionScript**. A programming language is a kind of language, similar to English, that we humans use to communicate with computers.

When you've created your graphics and programmed your game, the underlying technology, which you will learn all about in this book, then puts it all together to make your game work.

There are two common ways to make games with Flash and ActionScript.

- You can use software called Flash Professional, which is made by Adobe. This software allows you to create all the visual graphics for your game and perform all the computer programming using only one piece of software.
- The second approach is to create the graphics for your game using any graphic design or illustration software you like, and then use a **computer programming code editor**, such as Flash Builder or Flash Develop to program the ActionScript code to make your game work.

This book takes the second approach. It's much more flexible because it allows you to choose any graphic design or programming software you're comfortable using. It also means that the skills you learn won't be tied to any particular version of Adobe's Flash Professional software, which is subject to frequent change, and it frees you from the considerable cost of buying or upgrading it, which can be

a significant barrier for someone just starting out in game design. It used to be that the only way to make Flash games was with Adobe's Flash Professional software, but this is fortunately no longer the case. There are now far more flexible tools available for making Flash games, and this book will show you how to use them.

Tens of thousands of people around the world have made a career out of designing games with Flash. With this book and a little bit of practice, you could become one of them.

Basic requirements

Surprisingly, video game design can be a relatively low-tech affair. Here's the basic equipment you'll need to make use of this book.

A computer

You need a reasonably up-to-date computer, either running Windows or the latest version of Mac OS X.

Graphic design software

You need graphic design software, such as Photoshop, Photoshop Elements, Illustrator, or the free open source GIMP. Photoshop is the most widely used graphic design software on the planet, so it's extremely useful to learn how to use it; however, it's very expensive. You can download **Photoshop** from Adobe's website at

www.adobe.com/products/photoshop.html

You could also make all your game graphics using **Adobe Illustrator**, and indeed most of the game graphics in this book were created using it. It's unfortunately also quite expensive, but it's definitely worth making the long term investment of purchasing it and learning how to use it if you're serious about game design because it's probably the overall best illustration software available. You can find Adobe Illustrator at

www.adobe.com/products/photoshop.html

A much cheaper alternative to Illustrator or Photoshop is **Photoshop Elements**. Elements has all the functionality of the full version of Photoshop without some of the high-end professional features. For creating game graphics, you probably won't need any of those high-end features, most of which are specialized for photography and print production. Photoshop Elements may be a very good choice. You can download and purchase it from

www.adobe.com/products/photoshopel/

Or you could save your money and download a free copy of **GIMP**. GIMP does almost everything that Photoshop does for the purposes of making game graphics. The one big disadvantage is that the menus and windows are very different to Photoshop's, so you'll find it very difficult to follow the instructions in this book. Happily, there's a good solution to this problem. You can install a software plug-in for GIMP called **GIMPshop**. GIMPshop makes GIMP look and function exactly like Photoshop. The

means that you can follow all the instructions in this book—and in any other Photoshop tutorials—without having to spend a penny on the somewhat prohibitive cost of Photoshop.

You'll find a free copy of GIMP at

www.gimp.org/

And you'll find GIMPshop at

www.gimpshop.com/

Another option is that you can create your game graphics using free web-based software, such as Aviary.

www.aviary.com/

Aviary has many of the same basic features as Photoshop and Illustrator, and it doesn't require you to purchase or install anything. If you don't have any other graphics design or illustration software installed on your computer, and aren't yet sure about purchasing any, Aviary may be an excellent place to start.

Computer programming software

You'll also need computer programming software, either **Flash Builder** or **Flash Develop**. The pieces of software are known as **IDEs** (integrated development environments). Don't let that term scare you! They are just text editors or word processors that are specialized to help you write and edit computer programming code. They will also put all the pieces of your finished game together for you, usually at the click of a button.

At the time of writing, Flash Builder is free if you're a student or an unemployed developer. But if you're not either of those, it may be worth paying full price for it because it will save you a great deal of time and trouble. Flash Builder currently comes in three versions: Educational, Standard, and Premium. The Standard version is all you'll need to make use of this book. You can find out if you qualify for the free Educational version of Flash Builder by visiting Adobe's educational web site and following the links.

www.adobe.com/devnet/edu.html

You can download and buy the Standard version of Flash Builder from

www.adobe.com/products/flash-builder-standard.html

If you're running Windows, consider using the completely free and superb Flash Develop. It will not be difficult for you to use the techniques in this book with Flash Develop, if that's what you prefer to use. All the game design techniques and programming code in this book are compatible with both Flash Builder and Flash Develop. However, if you do decide to use Flash Develop, you'll need to make a little more effort to install and use it. Some aspects of the installation and configuration can be a little tricky, so don't consider starting with it unless you're highly computer literate. You can download Flash Develop from

If you decide to use Flash Develop, you'll also need to download and install two additional software components. First, you need Adobe's free **Flex SDK** (software development kit). The Flex SDK is actually the brains behind what makes all your Flash games work. You can download the latest version from Adobe's web site at

<http://opensource.adobe.com/wiki/display/flexsdk/Flex+SDK>

You will also need the **debug version** of Adobe's Flash Player software. You'll find it at

<http://www.adobe.com/support/flashplayer/downloads.html>

Download the Flash Player version for your system that's called a **content debugger**. The Flash Player is the software that actually runs your games. You need the special debug version so that Flash Develop can tell you if you've made any mistakes in your programming code.

However, if you already have another version Adobe's Flash Player on your computer, it's wise to first uninstall it before installing the new debug version. Doing so can prevent some quirky bugs from occurring. You can download a Flash Player uninstaller for Windows from

http://kb2.adobe.com/cps/141/tn_14157.html

If you need a Flash Player uninstaller for Mac OSX, you'll find it at

http://kb2.adobe.com/cps/909/cpsid_90906.html

After you've uninstalled your current version of the Flash Player, install the new debug version.

There are very many other IDEs besides Flash Builder and Flash Develop that you could use to write your ActionScript programs, such as **Eclipse** (for Windows and Mac OSX), **TextMate**, or **XCode** (only for Mac OSX). Or, with a bit of effort you could also use the humble **Notepad** (Windows) or **TextEdit** (Mac OSX) applications. All of these will require a bit of research on your part to set up and use to make Flash games, but many professional Flash game designers use them exclusively. I don't recommend any of them if you're just learning to program, but you might want to look into them once you have a bit more experience.

Adobe Flash Professional

Optionally, you could also use Adobe's **Flash Professional** to make both the graphics for your games and write the programming code in a single piece of software. The book's bonus download chapter, "Flash Animation and Publishing Your Games" will show you how to do this. Download Flash Professional from

www.adobe.com/cfusion/tdrc/index.cfm?product=flash

Making games exclusively with Flash Professional was the subject of the first edition of this book, *Foundation Game Design with Flash*. Using Flash Professional is still a great way to make Flash games, and it's still possibly the quickest and easiest. But learning how to make games without it gives you a number of advantages. It means your skills won't be tied to one particular piece of software

one version of a piece of software. It means that you're free to choose whichever software you like to program your game and make your graphics in. It also means that you'll be developing skills and work styles that will be easy to transfer to other game design or programming technologies when you're ready to take that step. This is the more flexible approach I'll be taking in this edition of the book.

Software used in this book

The examples and instructions in this book use Flash Builder, Photoshop (or Photoshop Elements) and Illustrator. These programs were picked because they're widely available, easy to use, don't require any complicated installation or configuration, and are common to both Windows and Macintosh. As a bare minimum, these are the only pieces of software you'll need to make good use of this book.

[Chapter 2](#) includes an introduction on to how to use Photoshop. You'll learn how to make game graphics using Illustrator in [Chapter 7](#).

If you choose to use any other graphic design or programming software, make sure you read its documentation carefully and are comfortable using it before you start working through the projects in this book.

Things you need to know

This book assumes that you haven't had any experience making graphics for games or any experience with computer programming. You'll go on a step-by-step journey through these fascinating worlds. If you want to learn to design games from scratch and know absolutely nothing about it, this book is a good place for you to get started.

That said, game development and the ActionScript programming language are huge topics that you could easily spend a lifetime studying. No one book can provide all the answers to all the questions you might have while you're learning. If you've never had any experience with professional graphic design software, I highly recommend that you spend a bit of extra time learning how to use Photoshop, GIMP, or Illustrator before you proceed much further in this book. You're going to be making lots and lots of graphics and illustrations for your games, and you'll have much more confidence when tackling the programming side of game design if you're comfortable making graphics. Spend a week or two tinkering around with Photoshop or GIMP, and you'll have your head well above water when you get to game programming. Here are some resources to help get you started.

If you're using Photoshop, read through the documentation and work through some of the exercises and sample projects in Adobe's online help system. (To access them, select Help ▸ Photoshop Help from Photoshop's menu.)

If you're using GIMP with the GIMPshop plug-in, you can follow any tutorials intended for Photoshop. A good place to start is Adobe's "Getting Started" tutorial site, which contains many helpful videos on basic Photoshop use. Go to

www.adobe.com/support/photoshop/gettingstarted/

A web search for "Photoshop tutorials for beginners" will also turn up a voluminous trove of tutorial sites, too numerous to mention here, but many of which are excellent.

But, hey, if you want to dive into the deep end right away, I'm with you! This book is a great place to

- [*On a Farther Shore: The Life and Legacy of Rachel Carson pdf, azw \(kindle\), epub, doc, mobi*](#)
- [**Computational Modeling of Biological Systems: From Molecules to Pathways \(Biological and Medical Physics, Biomedical Engineering\) pdf, azw \(kindle\)**](#)
- [download online The Black Pearl online](#)
- [download online The Power of Argumentation \(Poznań, Studies in the Philosophy of the Sciences and the Humanities, Volume 93; New Trends in Philosophy\) pdf, azw \(kindle\), epub](#)
- [The Pleasure of My Company: A Novel pdf, azw \(kindle\), epub](#)
- [read *Beginning iPad Application Development*](#)

- <http://www.1973vision.com/?library/Den-senila-skalbaggens-tr--st.pdf>
- <http://academialanguagebar.com/?ebooks/Dawn--Cutler--Book-1-.pdf>
- <http://rodrigocaporal.com/library/Mathematical-Methods-in-Quantum-Mechanics--Graduate-Studies-in-Mathematics-.pdf>
- <http://monkeybubblemedia.com/lib/The-Power-of-Argumentation--Pozna---Studies-in-the-Philosophy-of-the-Sciences-and-the-Humanities--Volume-93--New-Tr>
- <http://thermco.pl/library/First-Aid-for-the-Pediatrics-Clerkship--3rd-Edition-.pdf>
- <http://www.celebritychat.in/?ebooks/Beginning-iPad-Application-Development.pdf>