# Contents


**1 Preample** **2**

1.1 Structure of this book . . . 3

1.2 Tools used throughout this book . . . 3

1.3 What is a 2D game engine? . . . 3

**2 Introduction to SpriteBuilder and Cocos2d** **4**

2.1 Introduction to Cocos2d . . . 4

2.1.1 Scenes . . . 4

2.1.2 Nodes . . . 5

2.1.3 Scene Graphs . . . 5

2.2 Introduction to SpriteBuilder . . . 5

2.2.1 The Editor . . . 5

2.2.2 CCB Files . . . 5

2.2.3 Publishing - How SpriteBuilder and Xcodework together . . . 5

2.2.4 How SpriteBuilder and Cocos2d . . . 5

2.2.5 Code Connections . . . 5

2.3 A first SpriteBuilderproject . . . 5

# 1 Preample

Welcome to **the** most compact yet detailed guide to iOS game programming. You will be guided through absolutely everything you need to know about Cocos2d and SpriteBuilder and 2d game programming in general.

While we will cover the very basics of game programming, such as scene graphs, animations and game loops - Objective-C, the language we will be using throughout the book is not in the scope of things you will learn. When starting this guide, you are expected to have a solid foundation of Objective-C knowledge.

The structure in which you will learn is the following:

- Tools: Get familiar with the very basics of Cocos2d and SpriteBuilder
- Infrastructure: Understand that on a high level a game consists of scenes. Understand how to create scenes and navigation paths through these scenes with Cocos2d and SpriteBuilder
- Action and Movement: Understand how objects in your game can be moved and animated. With Cocos2d and SpriteBuilder
- Interaction: Understand how user interaction can be captured, including Touch interaction and Accelerometer.
- Interobject Interaction: Understand how to use the delightfully integrated Chipmunk physics engine
- Beyond the Basics; Recipes and Best Practices: Once we have the basics, we will look at a ton of recipes and exciting Cocos2d classes, which you can use to create any kind of 2d game. Particle Effects, Custom Drawing, Custom Shaders, Tile Maps, Networking, Audio, cocos2d UI in depth, etc.

## 1.1 Structure of this book

This book shall function as a learning guide and a reference book. Therefore most examples will be small and self-contained. Instead of builind a game throughout the whole book, you will learn by implementing very small projects that are limited to the material we are currently discussing. That shall give you a better chance of understanding the concepts/code snippets and using them in your original game, instead starting of from an example game you have built in this book.

After we have discussed all the basics and you have a good understanding of the Cocos2d API I will point you to resources that provide example implementations for specific game types.

There are two different ways to read this book. From the front to the beginning, gaining knowledge in logical groups. Or if you aren't a beginner and would like to use this book as an example driven extension of the API reference you can look up pages by Class names or concept names. There is a special glossar in the back of this book.

## 1.2 Tools used throughout this book

The two main tools we will be using are Cocos2d and SpriteBuilder. Many of the problems that occur during game development can be solved by both of these tools. Wherever it makes sense I will point out both ways, one using only Cocos2d and one using SpriteBuilder. This will allow you to see the advantages of each approach and finally decide which tool you want to use in certain situations for your own games.

## 1.3 What is a 2D game engine?

# 2 Introduction to SpriteBuilder and Cocos2d

Now it's time to dive into 2D Game Development! For this chapter I will assume that you haven't written a game with a game engine so I will explain the relevant concepts fairly detailed.

## 2.1 Introduction to Cocos2d

To understand what Cocos2d is, it is helpful to look at the history of game development. Back in 1980 video games were written in assembler and images were drawn to the screen by manually setting colors for certain pixels. Since then a wealth of frameworks and libraries has been written to make the life of a game developer easier.

When working with a 2D game engine for the first time you will be introduced to a whole set of new terminology. Just as a framework to write desktop applications knows the concept of windows, buttons and mouse clicks a 2D game engine comes with its own set of terms and techniques. Let's get started by talking about *Scenes*.

### 2.1.1 Scenes

Scenes are a term you are most likely to now from movies. However, the term is very commonly used for game engines, too. In most game engines, including Cocos2d, scenes are used to structure your game.

In Cocos2d scenes are the highest level on which you can structure your game content. A common example would be game with following scenes:

- Main Menu Scene

- Gameplay Scene
- Leaderboard Scene
- Setting Scene

By default every scene in Cocos2d is a full-screen scene.

### 2.1.2 Nodes

### 2.1.3 Scene Graphs

## 2.2 Introduction to SpriteBuilder

As of this writing SpriteBuilder itself is a very new tool, released in early 2014.

### 2.2.1 The Editor

### 2.2.2 CCB Files

### 2.2.3 Publishing - How SpriteBuilder and Xcodework together

### 2.2.4 How SpriteBuilder and Cocos2d

### 2.2.5 Code Connections

## 2.3 A first SpriteBuilderproject