Mobile App Development with Corona

Getting Started

Brian G. Burton, Ed.D.

Mobile App Development with Corona: Getting Started

SAMPLE

Brian G. Burton, Ed.D.

Mobile App Development with Corona: Getting Started- Sample
By Brian G. Burton, Ed.D.

Printed in the Abilene, Texas, United States of America

Published by Burtons Media Group.

Electronic editions are available. See
http://www.BurtonsMediaGroup.com/books for more information.

Cover images were generated using Corona Simulator and represent views of apps made in this book on the Droid®, Galaxy Tab®, iPad®, and iPhone® (from left to right).
Trademarked names and images may appear in this book. Rather than use a trademark symbol with every occurrence, we have used the name only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

While every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

Library of Congress Control Number: 2011909457
ISBN (eBook): 978-1-937336-00-4 | 1-937336-00-X
Version 1.06

Quick Table of Contents

Contents

Additional topics available in the full book:

Chapter 8
Phun with Physics

Chapter 9
Creating a Game with Corona

Chapter 10
Star Explorer Continued

Chapter 11
Media Makes the World Go Round

About the Author

Brian Gene Burton, Ed.D. is a teacher, author, and game developer. He has created game development degrees at two universities and enjoys researching and playing virtual environments. Brian presents and publishes internationally on his research and enjoys sharing what he has learned about game and mobile development. When not traveling or teaching, he can be found at his home in the Ozark Mountains of Missouri with his beautiful wife of over 25 years, Rosemary.

Dedication:
I dedicate this book to my loving wife whose support and encouragement kept me focused and writing.

A special thank you to my students and the Corona community for their support and requests for specific details and editorial comments that helped so much with the development of this book.

Ch. 5 tileset graphics from Reiner's Tilesets (http://www.reinerstilesets.de) are used with permission.
Ch. 6 sounds and music loops were graciously provided by Shaun Reed of Constant Seas. You can check out his band at http://www.constantseas.com

All other graphics (unless specified) and cover designed by Brandon Burton.

Copyediting and formatting assistance provided by Brianna Burton (http://www.LiteraryDiaries.com)

Foreword

Not long ago, circa early 2008, Walter and I decided to go into the app making business. We would create a series of mobile apps for us to sell via the app store as we saw Apple and its newly announced iPhone as the future of smartphones and app distribution.

We decided to build simple apps at first and then progress to more complicated ones. But as we dove into it, and iPhone being in the inchoate state, we weren't sure if it would take off. Being an Apple product, we knew it was going to sell and create quite a splash, but it was too early to tell. Nokia on the other hand, had a huge market share of the smartphone business but app development and distribution were lagging.

So we took a bet and knew that in order for us to be successful at writing our own apps, we needed to cover more than just one platform. On one hand, we knew iPhone was poised to be a winner, but on the other hand Nokia had quite a grip on the market with their series 60's.

As we forged ahead, and started building a framework that would allow us to cover more than one phone base and quick app production. We looked at several different options, scripting languages, and a slew of technologies that we could leverage and create our own framework for us to use.

From our very own learned experience, we knew we had to move fast, and after digesting all the options we had, we settled on Lua and started working on the framework now known as Corona.

Internally, we called this nascent framework "Ratatouille", the name was apropos because it took us back to the days of programming within a constrained memory model, small disk sizes as well as small screen sizes.

After the initial scaffolding, we started building some prototypes of the kinds of things we could do and it was, at that time for us, a glorious moment, we honestly thought we were a bit ahead of our time and we weren't even sure if it was going to work - typical engineering mentality. You work hard and after weeks all that we could show for was a rectangle being drawn on the screen.

But the ‘aha’ moment came when the same code based allowed for the app to work on the Series60 as well as on the iPhone. That quickly removed all doubt and we started adding features to Ratatouille left and right.

At one point, we had about six prototypes we had built and they all worked flawlessly, it was easy to prototype apps with this pre-alpha version of Ratatouille.

Eventually, we dropped support for Nokia and started support on Android, and decided to go knock on some VC’s and see if we could make this into a business as we saw quite an interest from our own friends and friends of friends on our product.

Fast forward to today. In typical Silicon Valley fashion, Corona was born from an idea out of our garages in order to solve our needs. Little did we know we would create a tool that would enable thousands of developers to fulfill their entrepreneurial spirit and start businesses using Corona SDK.

Today, over 20,000,000 people have played with Corona-based games and apps. These games/apps are being written all the way from 14 year olds, to teams of dedicated gamers and by ad agencies and studios. And the best part is, there is no slowdown in sight.

But Corona can’t just be successful by your apps alone. It also takes time and dedication and learning from trusted individuals like Dr. Burton, who has time and time again created some great tutorials on how to use Corona. And in his own entrepreneurial spirit, he has taken valuable time from his busy schedule to write a book on Corona.

This book is an excellent way to introduce you to our Corona SDK and will serve as the definite go to guide on how to learn and build Corona apps.

I know you will enjoy the book as much as you will enjoy building apps with Corona.

Carlos M. Icaza
Co-founder, Ansca Mobile, Inc.

Preface

Welcome

Welcome to mobile application development with Corona. This book is the result of years of developing for mobile devices. In early 2010 I began looking for a better way. I wanted a tool or set of tools that would allow me to develop more quickly and easily for multiple platforms of mobile devices. I was tired and frustrated with having to re-work everything to be able to make the same app on an iPhone, iPad, or Android device. After trying several different tools and development environments, I came across the Corona SDK by Corona Labs (http://www.CoronaLabs.com). While it was still early in the development of the SDK, it was apparent that the Corona Labs team was committed to building a quality set of tools and that a devoted community was quickly forming around this great SDK (Software Developers Kit). In the early days of my learning Corona, I focused on creating tutorials (available on my website: http://www.BurtonsMediaGroup.com/blog). After teaching Corona to several of my mobile and game development classes, and with the encouragement of my students, I began the process of creating a book that could be used as a teaching resource for the Corona SDK. You hold the fruits of that endeavor in your hand. I hope that you enjoy learning Corona as much as I have!
Best wishes,

Brian G. Burton, Ed.D.

Who This Book Is For

While my focus and impetus for writing this book is that it be used as a textbook, I have also written it with the understanding that many (hopefully) are just interested in learning more about the Corona SDK and want to develop for multiple mobile devices at the same time. I have the expectation that anyone using this resource already has some basic programming knowledge and experience. I do not spend very much time going over programming fundamentals. There are many great books on programming, I recommend you start there and return to app development when you have the basics.

How This Book Is Organized

While writing this book, I have kept the traditional 16-week semester in mind, assuming one chapter per week. While that doesn't work for everyone, it should be enough for most people to get started with mobile development using the Corona SDK. My first draft ended up with more than 20 chapters. After reorganizing content and continuing to develop, we are now down to 16 chapters with an additional chapter on great resources and a couple of appendices that were graciously supplied by Corona Labs on the Lua scripting language.

Conventions Used In This Book

Throughout the book I will use `Courier New` font to denote code that should be typed in exactly. When you find examples that are in *Courier New, Italics* you will need to enter your own value.

Using Code Examples

This book was written to help you learn to develop applications and games with the Corona SDK. In general, you may use the code in this book in your programs and documentation. You do not need to contact us for permission for reproducing a significant portion of the code. You don't need to ask permission to write an app that uses large chunks of code.

Now, on the other extreme, if I see apps that exactly reproduce the examples from a book or tutorial, I will not be a happy camper. I don't have issues with using the examples as a starting point, but take the app much further; be original! Answering questions by citing this book or quoting examples does not require permission (but I would appreciate the citation).

I reserve all rights for selling or distributing the examples in any format provided in this book. If you're not sure if your use falls outside of the fair

use laws, please feel free to contact me at:
DrBurton@BurtonsMediaGroup.com

Why didn't I use _______ for _______

There are a lot of great products available that can help the budding programmer/developer get their work done much faster (see chapter 17 for a short list). As this book is aimed at college students and people just getting started, I tried to not use outside tools. If a tool was required to get the project done, I tried to use only free or low cost tools. If I didn't use one of your favorites, I either 1) didn't know the tool existed; 2) was unable to get an evaluation copy of the software in a timely fashion; or 3) just didn't like that tool (probably the first or second option). If you know of a great tool that can save time and money to developers, please share it with the world in the discussion board on this books site: http://www.BurtonsMediaGroup.com/books.

Appendices

Appendices A and B on the Lua programming language were supplied by Corona Labs and are included with permission. While we have performed some copy editing to (hopefully) improve readability, the original content and examples have remained as provided.

How to Contact Us

Please address any comments or questions to the books website: http://www.BurtonsMediaGroup.com/books or email DrBurton@BurtonsMediaGroup.com.

Why I Chose to Self-Publish

The decision to self-publish this book was reached after a great deal of consideration. While there were numerous publishers interested (both academic and technical), I have decided to publish at least this first edition without the use of traditional publishers. There are many reasons why I made this decision, even though it will most likely lead to fewer sells.

First among my concerns was the price of the final book. I am sick of seeing textbooks at $100+. I feel this pricing is wrong and places an undue burden upon students. While publishers have cut the price slightly with the advent of eBooks and eTextbooks, it hasn't been enough in my opinion.
My second concern was how rapidly software environments change. I personally hate having to purchase a new book for each major revision of software. I have stacks of books that are now completely useless. I decided to publish this first as an eBook, which allows me to update and provide it to you, the reader, more rapidly. I will provide the minor updates between editions to the eBook to everyone who purchases the eBook through my website: http://www.BurtonsMediaGroup.com/books/book-update/

That being said, if you received a copy of this book either through a torrent or a friend, please purchase your own copy through my website. This will provide you with the most recent version of the eBook and encourage me to continue to update it. While I am doing this to help my students, I have bills to pay, and my wife is really good at keeping my 'honey-do' list up-to-date. Help me to avoid that list by buying a legitimate copy of this book (I don't have to work on her list if I'm writing or editing).

On the downside of self-publishing, I do NOT have a team of people to proof and double check everything in this book. I am sure that typos were entered by gremlins during the night. That and I have dyslexia. I did hire a person to proof the final version of the book, but having read many books that were published by major companies and finding errors in their books, I am sure that errors remain in this one. Please let me know if you find a typo on the book's forum site: http://www.burtonsmediagroup.com/forum

Chapter 1

Hello World: Setup and Get Going!

You've got a great app/mobile game idea. Wonderful! Now, how do you create it and get it on to an iPhone/Droid/iPad, (or whatever your device of choice is)? There are so many devices to choose from. Which platform is best for my app?

With so many platforms to choose from (Android, iOS, RIM, Windows, to name a few), the choice of platform to develop for can be very difficult. Each platform uses a different language, has a different API (Application Programming Interface) and requirements. How willing should we be to get locked into one development platform? Should we choose just one?

Fortunately with the advent of tools such as Corona by Corona Labs, it is now possible to develop for multiple platforms at the same time. To write once and publish to a host of different devices is the ultimate solution in the mobile publishing world.

Corona Labs's Corona currently allows the budding developer to publish to Android and iOS (Apple) devices, be it a smart phone or tablet. This text is written to help students everywhere gain the fundamental skill set to be able to take their app idea and publish it using the Corona SDK.

What this book is not

While this book is designed to teach the basic of mobile application development, it is not designed to teach programming fundamentals. I am making the assumption that you already know the basics of computer programming. If you don't know how to use an "if then" statement, a loop or a function, you're probably not ready for this book.

While I have made every attempt to cover the basics that most students want to learn during a 1st semester course in mobile app development, due to space and time issues, only so much could be included. There is already a second volume in development that will cover more advanced mobile application development and a volume that is just on game design with Corona.

Getting Started

Corona: Some background

Corona Labs was created in 2008 as a venture-backed company in Palo Alto, California. Before Corona, the Corona Labs team was responsible for creating many of the industry standard tools that I am sure you are familiar with. In the time that I have been developing apps with Corona, I have found Corona Labs to be one of the friendliest and helpful businesses that I have had the pleasure of working. In addition, online community is unusually friendly and supportive. If you decide to join the Corona community, be sure to continue this great spirit of helpfulness!

Software: Corona

It's no surprise that you will need the Corona SDK to get started. For learning, I recommend downloading the trial version. If you are ready to become a full subscriber, just head over to the on the Corona Labs website http://www.CoronaLabs.com/. Click on the download button and register (whether you are purchasing the subscription or downloading the trial). If you are a student or faculty, you can get a discount on your subscription by going to http://developer.CoronaLabs.com/forms/educators-and-students.

Software: Android

To get started developing apps for Android devices with Corona, you do not need to download any additional android software. However, you will need the Java SDK (typically referred to as JDK) to be able to do device builds if you are on a Microsoft Windows system. Go to http://www.oracle.com/technetwork/java/javase/downloads/index.html and download the Java Platform Standard Edition 32 bit JDK 6 by clicking on the "Download JDK" button. Note: You only need to download the JDK if you are on a Microsoft Windows system. If you are using a Macintosh with OS X, it is already installed.

Examples and Graphics

One last download that you can take care of right now: the code examples, graphics and other tools that you might want to use with the projects that are listed in this book. They are all available at http://www.BurtonsMediaGroup.com/books.

Software: iOS

For straight app development on the simulator, you don't have to download anything from Apple. However, when it is time to deploy to your test device or prepare the app for the iTunes store, you will need the Apple iOS SDK. If you are already an Apple developer, then you should be ready to deploy. If you are not a current developer ($99 per year for a standard iOS developer), you can register for a free developer account and download Xcode through the Apple App store for $4.99).

Development Hardware: Corona

Corona isn't too demanding on your development computer. As long as you are running at least OSX 10.6 or later on the Mac side, or Windows XP with a 1 GHZ processor on the PC side, you will be fine.

If you are planning to develop and deploy to iPhone, iPod Touch, and/or iPad, then you will need a Mac of some type to develop your apps. This is an Apple requirement. To keep in everyone's good graces, Corona will only publish for an iOS device if you are using a Mac computer to deploy the app. You will also be able to develop and deploy your Android based app from a Mac.

If you only have a windows system, you will be able to develop and deploy for Android based devices. You will also be able to develop for iOS devices. You just cannot deploy your finished app to an iOS device (or the iTunes store). I use both a Mac laptop and a PC, regularly switching back and forth during the app development process.

Development Hardware Matrix:

Development Hardware	Android OS		Apple iOS	
	Develop	Deploy	Develop	Deploy
Macintosh	X	X	X	X
Windows PC	X	X	X	

Development Hardware: Test Devices

If you are going to develop and sell apps for mobile devices, you should have a mobile device to test your creation. I have been on projects where I was required to develop for hardware that I didn't have. It was like herding cats. Using just the app simulator will get you 75% of the way home, but it won't allow you to spot all potential problems. On one of the fore-mentioned projects, the app worked fine on the simulator, but crashed on the mobile device and was rejected by Apple. The experience was more than just a little frustrating and taught me a valuable lesson: If you are developing for a platform, have test devices!

Android

Corona only builds for Android OS 2.2 and newer. Any devices that you plan to develop for must use the ARM V7 processor. There are plenty of devices that meet this requirement, so you shouldn't have any problem finding one to perform your tests.

iOS

For developing on iOS, you will need a developers license and either an iPhone, iPod Touch, or iPad. Obviously, having an older phone or iPad is a good idea for testing FPS (Frames Per Second) for graphically intensive apps. It is recommended that you use the newer iOS on your devices. To be able to deploy to an iOS device, you will need a Mac computer system and a Standard, Enterprise, or University developers account from Apple.

Publishing Considerations

I am sure that you are already envisioning how you will spend that first big check from your app sales. But before you can sell your app, you will need to decide with whom you will publish your app. There are several considerations specific to each publisher that you need to keep in mind.

Android (Google)

The Android market is very different from the Apple iTunes store. With the Android app market you have a number of different vendors available for selling your apps, including the Google's Android market (http://market.android.com), Amazon, Barnes and Noble, and a host of vendors. You will need to create an account with each vendor that you wish to sell through.

For $25 dollars (US) you can setup a developer account for Android with Google. You do NOT have to sign up for any account until you are ready to begin selling your apps. To get started visit setting up your personal account, visit http://developer.android.com/index.html.

Once you have your account setup you will need to decide if your app will be free or if you will charge for it. Throughout most of the mobile app industry the split is 70/30 in your favor. In other words, if you charge 99 cents for an app, you will walk away with 69.3 cents on each sell.

iOS (Apple)

One of the biggest advantages of the iOS market is that there is just one market to belong. To develop apps for the iOS market it costs $99 per year for a standard developer's license. An enterprise developer's license is also available, but unless you are developing for a major company that will only deploy your app internally, you will want the standard license. On a student budget, $99 can seem pretty expensive, so I recommend waiting as long as you can before getting your standard license since it is only good for one year. You can explore the developer's license options and the iOS SDK at: http://developer.apple.com/programs/register/. Apple also follows the 70/30 split on app sales.

Programming in Corona: Lua

In this text, the language that you will use throughout your programming experiences with Corona is Lua. Lua is a scripting language that was developed in the early 1990's. It is free, distributed under the MIT license and widely used for level scripting in major games and is a natural fit to be used in mobile application development due to the small size of the interpreter. Corona Labs has been kind enough to provide the two appendices on Lua (Appendix A & B) that are included in this book. If you would like to learn more about the Lua language, you can visit the Lua home page at http://www.lua.org. The first edition of Programming in Lua is available online at http://www.lua.org/pil. If you have ever programmed or scripted in any modern programming language, you should find Lua to be easy to learn as we progress through the following lessons.

Programming in Corona: Editors

The editor that you decide to use is a personal decision. Corona isn't impacted by the editor selection, so you need to use an editor that you are comfortable with. I recommend one that allows the integration of Lua to make your editing easier.

Some of the most popular editors in use with Corona include (but are not limited to) BBEdit, Eclipse, Notepad++, TextMate, TextWrangler, and Xcode. Of course you can ignore all of these editors and use notepad or textedit if you so desire.

BBEdit (Mac) by Bare Bones software, $99.99.
I have been using BBEdit on my Mac for quite a while and it is my editor of choice when working on my Mac. BBEdit has built in configurations (including Lua), which easily allows you to set the editor to the language you are developing in. http://www.barebones.com

Corona Project Manager (Mac/Win) by J.A. Whye, $75.
Corona Project Mangaer has a built in editor. Coupled with its ability to greatly simplify tracking your Corona project, the cost of CPM is well worth it. See Chapter 17 for a coupon code to save 30% on CPM.
http://www.coronaprojectmanager.com

Eclipse (Mac/Win) Open source, $0.
Eclipse is the editor I use when working on my PC. Eclipse has a large community of support. Though Eclipse was originally designed as a Java IDE (Integrated Development Environment), it is now the bases for many editors on the market. A Lua/Corona plugin is available. http://eclipse.org

Notepad++ (Win) Open source, $0
A popular open source language editor for the PC environment.
http://notepad-plus-plus.org/

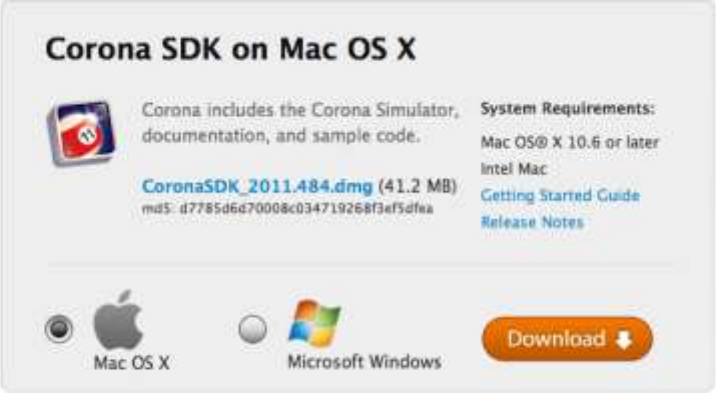
TextMate (Mac) by Micromates, €39 (about $57).
Textmate is very popular in the Corona community with a Corona plugin available on the Corona Labs website. http://macromates.com

TextWrangler (Mac) by Bare Bones Software, $0.
TextWrangler has the advantage of being a free editor for your Mac. Though it doesn't have all the bells and whistles as BBEdit, it will get the job done for those on a budget and offers integrated Lua support.
http://www.barebones.com

Xcode (Mac) by Apple, $0*.
Xcode is an integral part of the iOS SDK. If you are used to developing using Objective-C, Xcode is a natural choice. While Xcode is included with iOS SDK, it is only free if you are already a standard developer with Apple. If you register for a free account, the iOS SDK (which includes Xcode) is $4.99.

Configuring Corona

Installation of Corona SDK is a straightforward project. Just click on the download button at http://www.CoronaLabs.com, register, select whether you are downloading the Mac OS X or Microsoft Windows version of the Corona SDK, and follow the directions below based upon your operating system (images may vary depending on the version you are installing).



Corona SDK on Mac OS X

Corona includes the Corona Simulator, documentation, and sample code.

CoronaSDK_2011.484.dmg (41.2 MB)
md5: d7785d6d70008c03471926 8f3ef5dfea

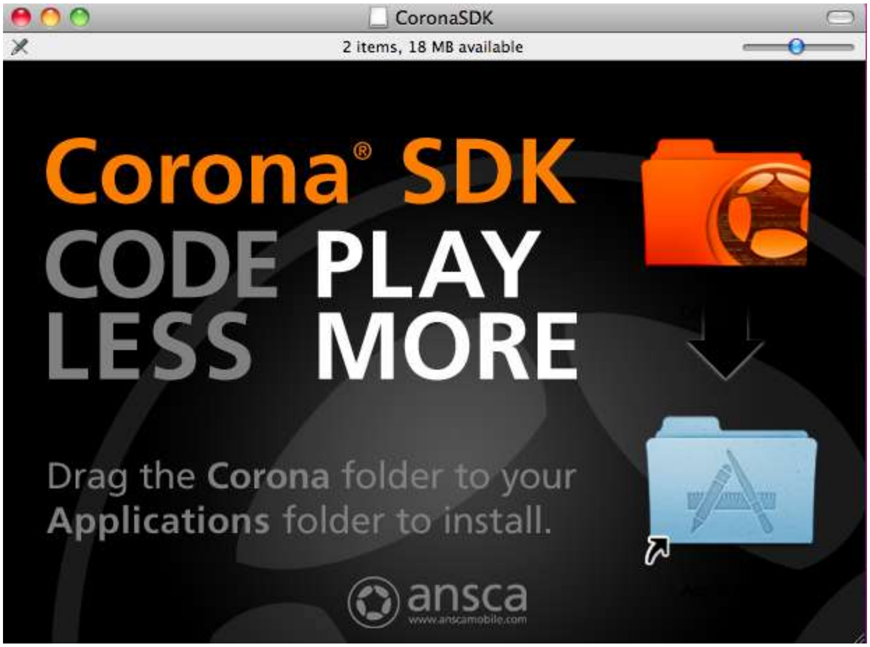
System Requirements:
Mac OS® X 10.6 or later
Intel Mac
Getting Started Guide
Release Notes

● Mac OS X ○ Microsoft Windows

Corona SDK Download

Macintosh

After you launch the downloaded file and agree to the software license, drag the Corona SDK folder onto the Applications folder.



Installing Corona SDK on a Macintosh

This will copy all of the Corona SDK files in to your applications folder. When you open up your Corona SDK folder, you will find sample code, tools, a resource library as well as the Corona Terminal and Simulator (the primary development tools that we will be using).

Windows

Corona SDK for Windows has low hardware requirements:

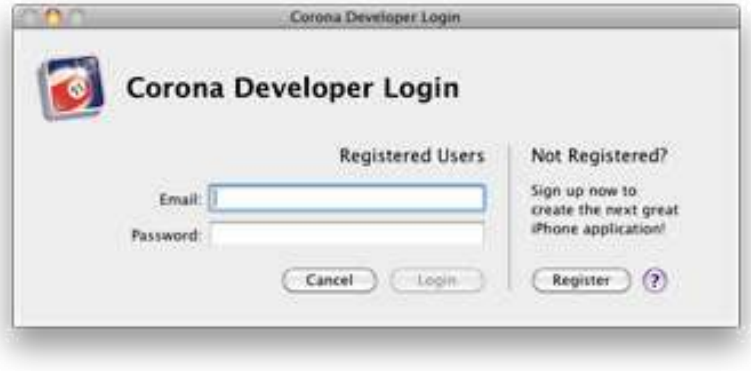
- Windows 7, Vista, or XP operating system
- 1 GHZ processor (recommended)
- 38 MB of disk space (minimum)
- 1 GB of RAM (recommended)
- OpenGL 1.3 or higher graphics system

In all of the installs that I have made of Corona, the only problem I have ever run into was when a system didn't have OpenGL 1.3 or higher. This was easily corrected by downloading newer graphics card drivers to the system. Corona SDK will run with older versions of OpenGL installed, as long as it is an application that is graphic intensive. You should be able to update your graphics card driver to correct the problem if it exists. More information about OpenGL can be found at http://www.opengl.org.

If you haven't already downloaded the Java JDK (Java Developers Kit), you should do so now. Go to http://www.oracle.com/technetwork/java/javase/downloads/index.html and download the Java Platform Standard Edition by clicking on the "Download JDK" button. On the next page, select "Windows x86" from the list of available downloads.
The JDK is required to be able to do device builds on Microsoft Windows systems. This is a free download from the Oracle website. After you have downloaded the installer, follow the normal procedure to install the JDK to your system.

Programming in Corona: Hello World (V1.0)

The first time you launch the Corona Terminal or Simulator it will ask you to login with your registration information that you used on the Corona Labs website. Complete this one time authentication and you will be ready to go.



Sign up now to create the next great iPhone application!

Corona Developer Registration

You should always launch the Corona Debugger on a Macintosh instead of the Simulator for performing application builds and testing. On a Windows system, launching the Corona Simulator also launches the Corona Simulator Output window (commonly referred to as the terminal window). The Corona

Terminal gives you important feedback when you are building your apps and allows for easier troubleshooting. The Corona Terminal will automatically launch the Corona Simulator.

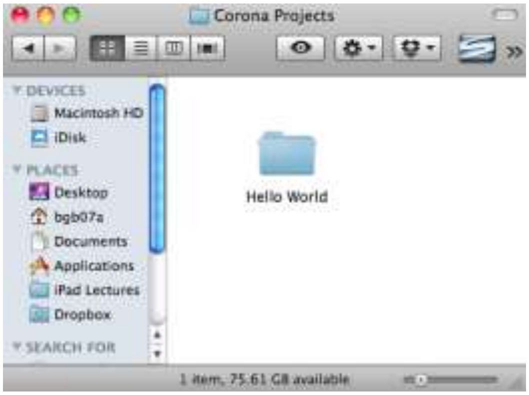
Project 1.0: Hello World

I personally always hated programming books and classes that spent the first chapter or week just getting all the details taken care of. I purchased the book or took the class because I wanted to program, not to go over some syllabus or a review of all the different ages of computer development. So let's skip all of that and do the required "Hello World" project.
Stop with the rolling of eyes! Before I lose you, let me guarantee that you will get a very valuable resource out of this Hello World project, something that you will use the rest of the time that you develop in Corona.

Was that enough to get your attention? Then let's get started!

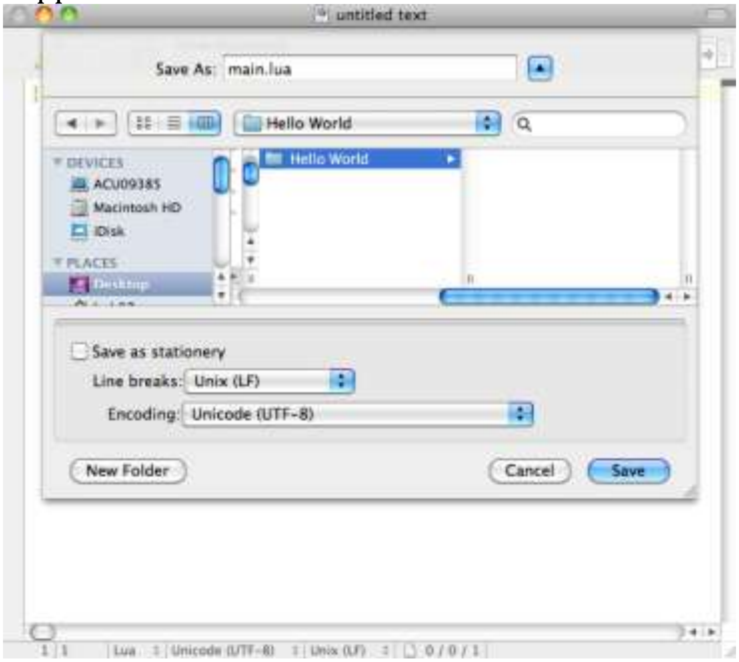
Project Setup

If you follow this process each time you start a new project, it will make your life a lot easier:
First, create a project folder called "Hello World". This can be on your desktop or wherever you like to organize your work. I keep all of my project folders together in a folder called "Corona Projects".



Create the Hello World folder for your project

Open your editor of choice (I'm using BBEdit in these initial screen shots). Create a blank file and save it as "main.lua" to your Hello World folder that you just created. The main.lua file is the first file that the Corona simulator will look for when it is run. If there is no main.lua file present, nothing will happen.



Save As: main.lua

□ Save as stationery

Line breaks: Unix (LF)

Encoding: Unicode (UTF-8)

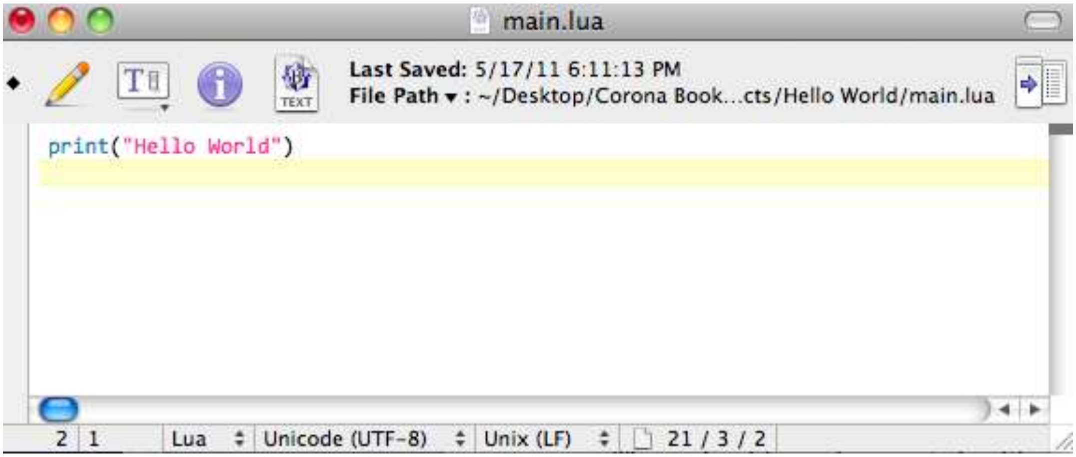
Save the main.lua file to your Hello World folder

There should now be a main.lua file in your Hello World folder.

Back in your editor type :

```
print("Hello World")
```

and save your file as main.lua.

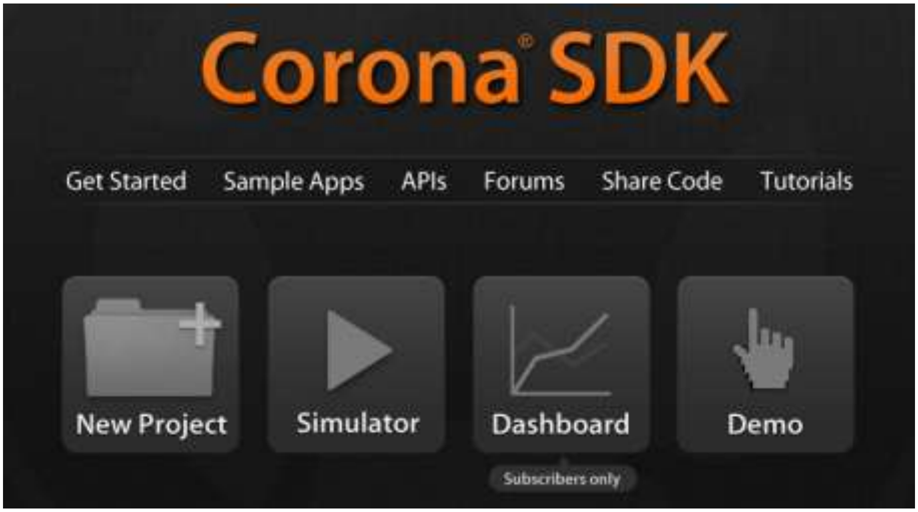


```
print("Hello World")
```

main.lua
Last Saved: 5/17/11 6:11:13 PM
File Path ▾ : ~/Desktop/Corona Book...cts/Hello World/main.lua
2 1
Lua
Unicode (UTF-8)
Unix (LF)
21 / 3 / 2

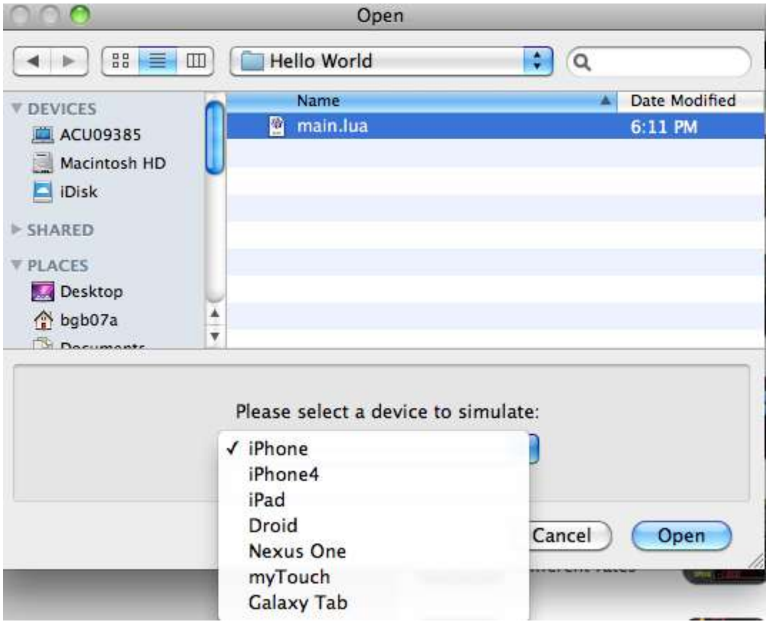
Hello World project in the editor

Next, you will need to launch Corona. If you are on a Microsoft Windows system, launch the Corona Simulator. On a Macintosh, launch Corona Terminal.



Corona at startup on a Macintosh – don't use the new project button yet!

On launch, you will see the Terminal window and the Welcome to Corona dialog box. Select "Open a Project" from the Welcome to Corona dialog and navigate to the Hello World folder that was created earlier. Your initial window might be different based upon the version of Corona that you are using.

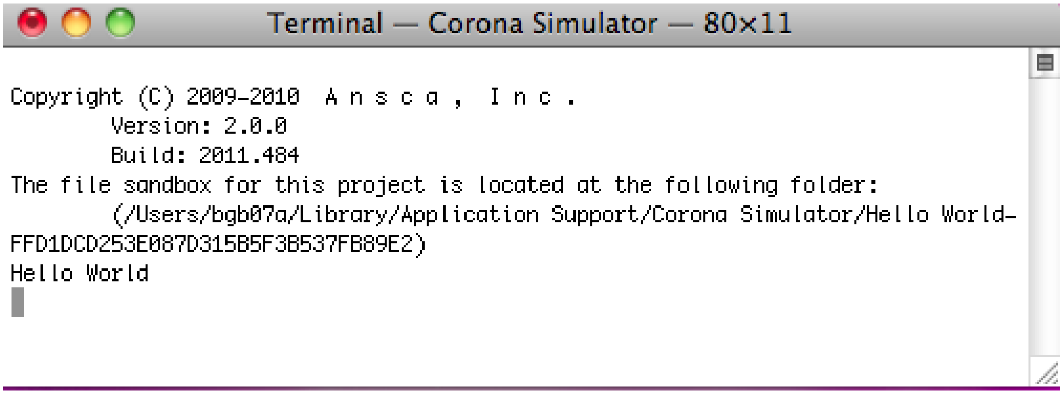


Please select a device to simulate:

Open Hello World & select device to simulate

When you are opening a project, you will be able to select which device you would like to simulate in the Corona Simulator. For now select either iPhone or Droid and click on the Open button. Selecting other devices could give you different results than what are in the screen shots that have been included.

As soon as you open the project, the simulator will run the project.
Did you notice? That's right, nothing happened…in the simulator. Look in the Terminal window.
At the bottom you will see your Hello World displayed.



```
Copyright (C) 2009-2010  A n s c a ,  I n c .
        Version: 2.0.0
        Build: 2011.484
The file sandbox for this project is located at the following folder:
        (/Users/bgb07a/Library/Application Support/Corona Simulator/Hello World-
FFD1DCD253E087D315B5F3B537FB89E2)
Hello World
```

Hello World in the Corona Terminal window

Congratulations! You just made your first Corona app! Now before you become disappointed, you just learned a very important tool for trouble shooting your applications. When something doesn't seem to be working correctly or displaying the way you want, you can send yourself messages through the Corona Terminal window. Believe me when I tell you that this one command will save you hours of troubleshooting headaches!

I am sure you also noticed that Corona generates a great deal of additional information before giving you the results of your print command. The first few lines provide information about the version of Corona and the location of the simulation files.

Note: If you didn't see anything, there are two areas that people commonly make a mistake: 1) they didn't save their main.lua file (I still make this mistake) or 2) when saving the main.lua file, it wasn't saved as a text file.

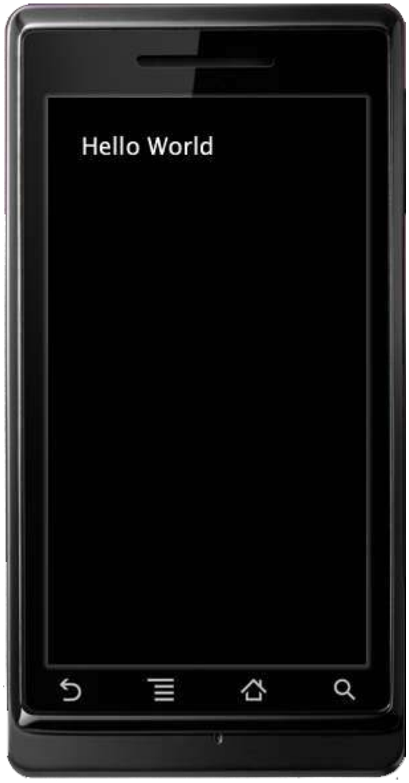
Project 1.1: Hello World (v2.0)

Back in your editor (you can use the same file)
Type:

```
local textobj = display.newText("Hello World", 50, 50,
native.systemFont, 24)
textobj:setTextColor(255, 255, 255)
```

Lua, the language behind Corona, is case sensitive so newText is different from newtext. Try newtext and look at the error that appears in the Terminal window.
Save the file, and then launch your simulator.

You should now see Hello World displayed in the simulator.



Hello World on the Droid simulator

What did you just do? Here's the run down:
First we created a local variable called textobj. We do not have to use the variable name textobj, we could use fred for the variable name but after a couple of days we might forget what fred represents. Use variable names

that make sense. It might mean more typing, but you will really appreciate it when you go to revise or update the program at a later date.

We set textobj equal to the object that we create by calling display.newText method, passing it the text “Hello World”, the X & Y location of the top left corner of the text, font, and 24 (the size of the text to be created).

The display.newText parameters are:

```
display.newText(text, X, Y, font, text size)
```

In the font parameter, you can use the system default of native.systemFont or native.systemFontBold. You can also enter the font type in as a string such as “arial” or “arial black”. If you set this parameter to *nil*, it will default to the native.systemFont.

In the second command line, we set the color of the textobj that was just created using the R, G, B color system (each color (red, green, blue) having a value between 0 – 255) to white:

```
textobj:setTextColor(R, G, B)
```

By default, the text object is white, so we didn't really accomplish anything by setting the textobj to white. I want to get you in the practice of setting the text color when you create a text object. Later we will look at how to fade the text object out (or in).

Now you have made your first REAL Corona app!

Warning: If you copy code from a website (or even from this book), sometimes the quotation marks will change from straight quotation marks to smart quotes. This WILL cause an error in Corona. Make sure your quotes are always "" and not “”.

Objects

You may have noticed the use of the term object sprinkled throughout the text thus far. When I use the term ‘object’ it is to denote anything that is used in our project; text, buttons, or sounds, they are all objects. Just as in the real, physical world, I can move or interact with an object (a lamp, table, or car), an object in your software is anything that can be interacted with.

Real world objects all have properties that help to describe the object’s location, color, or anything that can be changed about the object. If I have a car, I might describe the car’s location by its longitude and latitude.
In programming (including Corona), we are able to interact with each objects properties to make changes; such as when the textobj was created, we set the x, y, font, and size properties as well as the string that would be displayed.

Summary

This has been a busy chapter! Corona should now be installed on your system, you have been introduced to editors, hardware considerations, and publishing information. We even managed to develop two apps! The first introducing the critically important print command, the second actually displaying text to the simulator. Finally, the concept of an object in programming was briefly introduced.

Chapter 2

Buttons and Text

We are going to combine a few things with this Chapter:

- First, we will learn about creating an object that is interactive
- Second, we examine how to set or change the location of an object
- Third, we will add an event listener to the object so that we can interact with the app
- Fourth, we will make use of the math.random function to help us move an object around the screen.
- Finally, we will look at the Relaunch feature of Corona Simulator, which makes it so easy to tweak your code!

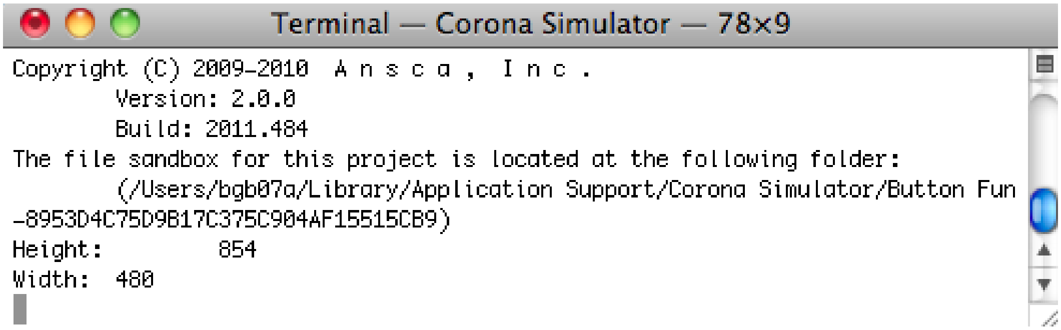
Know your Boundaries

When developing for multiple types of devices it is important to automate the placement of objects on the screen. In Corona, this can be done easily with display.contentWidth and display.contentHeight.

For our second project, let's start by finding the height and width of your device. This can easily be accomplished using the print command that was discussed in chapter 1. Start by creating new folder and main.lua for your app. Add the following code:

```
print("Height: ", display.contentHeight)
print("Width: ", display.contentWidth)
```

Launch Corona as you did in the previous chapter (Corona Terminal for Mac, Corona Simulator for Windows), open main.lua in the Button Fun folder, then select the type of device to be simulated. Your terminal windows will show the results.



```
Copyright (C) 2009-2010  A n s c a ,  I n c .
        Version: 2.0.0
        Build: 2011.484
The file sandbox for this project is located at the following folder:
        (/Users/bgb07a/Library/Application Support/Corona Simulator/Button Fun
-8953D4C75D9B17C375C904AF15515CB9)
Height:         854
Width:  480
```

Display Content Height and Width

In this example I selected Droid for my simulator. As you can see in Figure 1, the height is 854 pixels and width is 480 pixels.

We hope that you have found this sample of *Mobile App Development with Corona: Getting Started* valuable and informative.

The full book, programming resources, forums as well as a textbook version for use in colleges and schools can be found at: http://www.BurtonsMediaGroup.com/books

To stay up-to-date on what's happening at Burtons Media Group, Like us on Facebook, follow @DrBrianBurton on twitter, or check out our tutorials on game and mobile app development.