

Extracted from:

# OpenGL ES 2 for Android

A Quick-Start Guide

This PDF file contains pages extracted from *OpenGL ES 2 for Android*, 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 © 2013 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

# OpenGL ES 2 for Android

A Quick-Start Guide



Kevin Brothaler

Foreword by Mario Zechner, creator of  
libgdx, author of *Beginning Android Games*



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 <http://pragprog.com>.

The Android robot is reproduced from work created and shared by Google and is used according to terms described in the Creative Commons 3.0 Attribution License (<http://creativecommons.org/licenses/by/3.0/us/legalcode>).

The unit circle image in Figure 43, from [http://en.wikipedia.org/wiki/File:Unit\\_circle.svg](http://en.wikipedia.org/wiki/File:Unit_circle.svg), is used according to the terms described in the Creative Commons Attribution-ShareAlike license, located at <http://creativecommons.org/licenses/by-sa/3.0/legalcode>.

Day skybox and night skybox courtesy of Jockum Skoglund, also known as hipshot, [hipshot@zfight.com](mailto:hipshot@zfight.com), <http://www.zfight.com>.

The image of the trace capture button is created and shared by the Android Open Source Project and is used according to terms described in the Creative Commons 2.5 Attribution License.

Copyright © 2013 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: 978-1-937785-34-5  
Encoded using the finest acid-free high-entropy binary digits.  
Book version: P1.0—June 2013

# Welcome to OpenGL ES for Android!

---

Android has just gone through an incredible period of growth, with more than 750 million devices in the hands of consumers around the world and more than 1 million activations per day.<sup>1</sup> Along with Apple, Android also has a centralized market available on every Android phone and tablet, called Google Play. With this market installed on every Android device, there's never been a better opportunity for anyone who's ever had a dream to publish his or her own game or live wallpaper.

On Android, as well as on Apple's iOS and many other mobile platforms, developers create 2D and 3D graphics through a cross-platform application programming interface called *OpenGL*. OpenGL has been used on the desktop for a while now, and the mobile platforms use a special embedded version known as *OpenGL ES*. The first version of OpenGL ES brought 3D to mobile, which was very popular with developers because it was easy to learn and because it had a well-defined feature set. However, this feature set was also limited, and it wasn't able to keep up with the latest and greatest features from the most powerful smartphones and tablets.

Enter OpenGL ES 2.0. Most of the old APIs were completely removed and replaced with new *programmable* APIs, which makes it much easier to add special effects and take advantage of what the latest devices have to offer. These devices can now produce graphics that rival consoles from just a few years ago! However, to take advantage of this power, we need to learn the new APIs that come with 2.0. In August 2012, the Khronos Group finalized the specification for the next version, OpenGL ES 3.0, which is fully compatible with 2.0, extending it with a few advanced features.

So, what can be done with OpenGL on Android? We can create amazing live wallpapers and have them downloaded by millions of users. We can create a compelling 3D game that has vivid and breathtaking graphics. With the

---

1. <http://googleblog.blogspot.ca/2013/03/update-from-ceo.html>

declining cost of hardware and the increasingly massive reach of online stores, it's a great time to begin!

## What Will We Cover?

Here's a quick look at what we're going to discuss:

- In the first part of the book, you'll learn how to create a simple game of air hockey, including touch, texturing, and basic physics. This project will teach you how to successfully initialize OpenGL and send data to the screen, as well as how to use basic vector and matrix math to create a 3D world. You'll also learn many details that are specific to Android, such as how to marshal data between the Dalvik virtual machine and the native environment and how to safely pass data between the main thread and the rendering thread.
- In the second part of the book, you'll build upon what you learned in the first part. You'll use some advanced techniques, such as lighting and terrain rendering, and then you'll learn how to create a live wallpaper that can run on your Android's home screen.

## Who Should Read This book?

If you're interested in learning how to develop more advanced graphics on Android, then this is the book for you. This book assumes that you have some programming experience, including experience with Java and Android.

### Java

If you've worked with other managed languages before, such as C#, then moving to Java will be straightforward. If you're more experienced with native languages, then one of the main differences that you'll need to keep in mind is that Java on Android is a garbage-collected language that runs in a virtual machine, which has both benefits and costs.

The following books will help bring your Java up to speed:

- [\*The Java Programming Language\* \[AGH05\]](#) by Ken Arnold, James Gosling, and David Holmes
- [\*Effective Java\* \[Blo08\]](#) by Joshua Bloch
- [\*Thinking in Java\* \[Eck06\]](#) by Bruce Eckel

## Android

Once you're comfortable with Java, developing for Android just requires some experience with the appropriate libraries and methods. To cover all of the basics, I recommend reading [Hello, Android \[Bur10\]](#) by Ed Burnette. You can also follow the first two lessons of Google's Android training online:

- Building Your First App<sup>2</sup>
- Managing the Activity Lifecycle<sup>3</sup>

While it's possible to go through most of this book with the emulator, having an Android device on hand will make life much easier. We'll go into more detail about that soon, in [Section 1.1, Installing the Tools, on page ?](#).

This should be enough to get you through this book. We'll cover all of the basics from first principles, so you don't need prior experience in 3D graphics programming, and while we'll cover some math in this book, if you've taken trigonometry and linear algebra in the past, then you're well prepared! If not, no fear: everything will be explained in detail as we go along.

## How to Read This Book

Each chapter builds on the chapter before it, so this book is best read in sequence. However, all of the code samples are available online (see [Section 5, Online Resources, on page x](#)), so if you want to check out a specific chapter, you can always follow along by downloading the completed project from the previous chapter and continuing on from there. This can also help out if you ever get lost or want to start from a fresh base.

## Conventions

We'll use *OpenGL* to refer to OpenGL ES 2.0, the modern version of OpenGL for mobile and the Web.

In most parts of the book, we'll be working with the GLE20 class, which is part of the Android Software Development Kit (SDK). Since most of our OpenGL constants and methods will be in this class, I'll generally omit the class name and just mention the constant or method directly. We'll use static imports (see [Section 1.5, Using Static Imports, on page ?](#)) to omit the class name in the code as well.

---

2. <http://developer.android.com/training/basics/firstapp/index.html>

3. <http://developer.android.com/training/basics/activity-lifecycle/index.html>

## Online Resources

All of the resources for this book can be found at <http://pragprog.com/book/kbogla>, where you can find code samples and accompanying images and textures. If you have purchased the ebook, then you can also click on the hyperlink above each code extract to download that extract directly. You can also join in the discussions at the book's website and help improve the book by submitting your feedback and errata.

Please feel free to also visit *Learn OpenGL ES*, an OpenGL ES tutorial blog that I maintain.<sup>4</sup>

The following is a list of some great online resources maintained by the Khronos Group:<sup>5</sup>

- OpenGL ES 2.0 API Quick Reference Card<sup>6</sup>
- OpenGL ES 2.0 Reference Pages<sup>7</sup>
- OpenGL ES Shading Language (GLSL ES) Reference Pages<sup>8</sup>
- The OpenGL® ES Shading Language<sup>9</sup>
- OpenGL® ES Common Profile Specification Version 2.0.25 (Full Specification)<sup>10</sup>

I recommend printing out the reference card and keeping it handy, so you can quickly refer to it when needed. Android uses the EGL (a native platform interface) to help set up the display, so you may also find the Khronos EGL API Registry to be useful.<sup>11</sup>

## Let's Get Started!

There are more people with powerful cell phones and tablets than ever before, and the market continues to grow. Android's software tools make it easy to develop an application for Android, and Google's Play Store makes it easy for us to share our applications with the world. Let's head over to [Chapter 1, Getting Started, on page ?](#), and get things started!

- 
4. <http://www.learnopengles.com/>
  5. <http://www.khronos.org/>
  6. [http://www.khronos.org/opengles/sdk/docs/reference\\_cards/OpenGL-ES-2\\_0-Reference-card.pdf](http://www.khronos.org/opengles/sdk/docs/reference_cards/OpenGL-ES-2_0-Reference-card.pdf)
  7. <http://www.khronos.org/opengles/sdk/docs/man/>
  8. <http://www.khronos.org/opengles/sdk/docs/manglsl/>
  9. [http://www.khronos.org/registry/gles/specs/2.0/GLSL\\_ES\\_Specification\\_1.0.17.pdf](http://www.khronos.org/registry/gles/specs/2.0/GLSL_ES_Specification_1.0.17.pdf)
  10. [http://www.khronos.org/registry/gles/specs/2.0/es\\_full\\_spec\\_2.0.25.pdf](http://www.khronos.org/registry/gles/specs/2.0/es_full_spec_2.0.25.pdf)
  11. <http://www.khronos.org/registry/egl/>