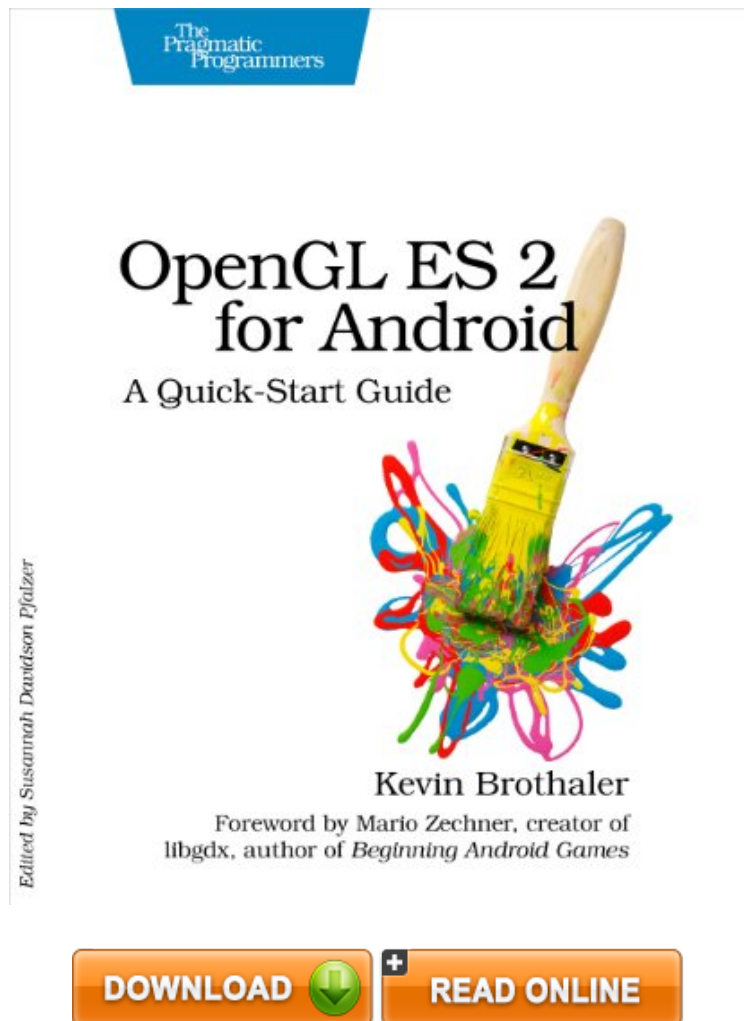


[Free] OpenGL ES 2 for Android: A Quick-Start Guide (Pragmatic Programmers)

OpenGL ES 2 for Android: A Quick-Start Guide (Pragmatic Programmers)

Von Kevin Brothaler

audiobook / *ebooks / Download PDF / ePub / DOC



Produktinformation - Verkaufsrang: #319939 in eBooks Veröffentlicht am: 2013-06-26 Erscheinungsdatum: 2014-02-04 File Name: B00IDEVIK6 | File size: 49.Mb

Von Kevin Brothaler : OpenGL ES 2 for Android: A Quick-Start Guide (Pragmatic Programmers) before purchasing it in order to gauge whether or not it would be worth my time, and all praised OpenGL ES 2 for Android: A Quick-Start Guide (Pragmatic Programmers):

Kundenrezensionen Hilfreichste Kundenrezensionen 1 von 1 Kunden fanden die folgende Rezension hilfreich. Sehr zu empfehlen Von Florian Westreicher Als langjähriger Programmierer hatte ich nun die Aufgabe eine UI Komponente in GLES zu erstellen. Dieses Buch hat mir dafür alles an Werkzeug in die Hand gegeben was ich brauche. Es ist sehr gut geschrieben, leicht verständlich und gut aufgebaut. Absolute Kaufempfehlung. 0 von 0 Kunden fanden die folgende Rezension hilfreich. Fantastisch Von Der Leser Ich habe noch nie mit einem Buch derart produktiv arbeiten und lernen können. Alles wird sehr verständlich erklärt. Jeder, der halbwegs Ahnung von der Android - Programmierung und ausreichende Englisch - Kenntnisse hat, wird nach der Lektüre dieses Buches problemlos OpenGL ES 2.0 -

Anwendungen erstellen können. Versprochen.

Kurzbeschreibung This book will teach you everything you need to know to create compelling graphics on Android. You'll learn the basics of OpenGL by building a simple game of air hockey, and along the way, you'll see how to initialize OpenGL and program the graphics pipeline using shaders. Each lesson builds upon the one before it, as you add colors, shading, 3D projections, touch interaction, and more. Then, you'll find out how to turn your idea into a live wallpaper that can run on the home screen. You'll learn about more advanced effects involving particles, lighting models, and the depth buffer. You'll understand what to look for when debugging your program, and what to watch out for when deploying to the market. OpenGL can be somewhat of a dark art to the uninitiated. As you read this book, you'll learn each new concept from first principles. You won't just learn about a feature; you'll also understand how it works, and why it works the way it does. Everything you learn is forward-compatible with the just-released OpenGL ES 3, and you can even apply these techniques to other platforms, such as iOS or HTML5 WebGL.

What You Need Preferably an Android phone or tablet that supports OpenGL ES 2.

Top Five OpenGL ES Tips

- 1) Avoid expensive operations on the UI and rendering threads. To avoid frame rate stutters or the dreaded Application not responding dialog from appearing, expensive operations should be run asynchronously in a background thread. This is especially important when it comes to the rendering thread, as rendering at 30 frames per second means that each frame has to complete in about 33 milliseconds to avoid frame stutters. Garbage collection is an expensive operation that happens non-deterministically and can cause rendering stalls, so to avoid this, you'll also want to minimize object allocation during a frame to reduce pressure on the garbage collector. You may even see benefits to doing a manual GC at certain points in the application.
- 2) Do expensive texture and shader loads during intermissions. Some expensive operations need to be done on the rendering thread, such as texture loads and shader compilations. These operations can cause noticeable stuttering if they happen in the middle of rendering, so to avoid this, preload these resources at opportune times, such as when the game / application is loaded, or at the beginning of each level.
- 3) Take advantage of the libraries out there. There are many libraries out there that support OpenGL ES 2.0 development without boxing you into a framework or a closed-source middleware solution. Libgdx is one of these libraries, and by using a library like libgdx, you can more easily port your code to other platforms, as well as take advantage of the library's math classes and asset loading management.
- 4) Read the documentation. The OpenGL specs and manuals are available for free from Khronos.org. Each GPU vendor also shares a wealth of free information on how to use their GPUs, and as you read the documentation, you'll find that they generally share a lot of advice in common, such as "minimize state switches" and "avoid discard in a fragment shader." Reading these documents and specs will give you a lot more insight into how the GPUs work at a lower level, and will help you understand what the pitfalls are, and how to avoid premature pessimization when writing your OpenGL code.
- 5) Focus on what will impress the end user. Sometimes it's easy to lose sight of the end goal, and to become focused on optimizing areas that won't make much of a difference to the end user. Does it matter if we get this shader to execute 2% faster? Or will we make much more of a visual impact by finding a better artist? Find out what the most important areas are, and then focus on improving those first.

Kurzbeschreibung This book will teach you everything you need to know to create compelling graphics on Android. You'll learn the basics of OpenGL by building a simple game of air hockey, and along the way, you'll see how to initialize OpenGL and program the graphics pipeline using shaders. Each lesson builds upon the one before it, as you add colors, shading, 3D projections, touch interaction, and more. Then, you'll find out how to turn your idea into a live wallpaper that can run on the home screen. You'll learn about more advanced effects involving particles, lighting models, and the depth buffer. You'll understand what to look for when debugging your program, and what to watch out for when deploying to the market. OpenGL can be somewhat of a dark art to the uninitiated. As you read this book, you'll learn each new concept from first principles. You won't just learn about a feature; you'll also understand how it works, and why it works the way it does. Everything you learn is forward-compatible with the just-released OpenGL ES 3, and you can even apply these techniques to other platforms, such as iOS or HTML5 WebGL.

What You Need Preferably an Android phone or tablet that supports OpenGL ES 2.

Top Five OpenGL ES Tips

- 1) Avoid expensive operations on the UI and rendering threads. To avoid frame rate stutters or the dreaded Application not responding dialog from appearing, expensive operations should be run asynchronously in a background thread. This is especially important when it comes to the rendering thread, as rendering at 30 frames per second means that each frame has to complete in about 33 milliseconds to avoid frame stutters. Garbage collection is an expensive operation that happens non-deterministically and can cause rendering stalls, so to avoid this, you'll also want to minimize object allocation during a frame to reduce pressure on the garbage collector. You may even see benefits to doing a manual GC at certain points in the application.
- 2) Do expensive texture and shader loads during intermissions. Some expensive operations need to be done on the rendering thread, such as texture loads and shader compilations. These operations can cause noticeable stuttering if they happen in the middle of rendering, so to avoid this, preload these resources at opportune times, such as when the game / application is loaded, or at the beginning of each level.
- 3) Take advantage of the

libraries out there. There are many libraries out there that support OpenGL ES 2.0 development without boxing you into a framework or a closed-source middleware solution. Libgdx is one of these libraries, and by using a library like libgdx, you can more easily port your code to other platforms, as well as take advantage of the library's math classes and asset loading management.

4) Read the documentation. The OpenGL specs and manuals are available for free from Khronos.org. Each GPU vendor also shares a wealth of free information on how to use their GPUs, and as you read the documentation, you'll find that they generally share a lot of advice in common, such as "minimize state switches" and "avoid discard in a fragment shader." Reading these documents and specs will give you a lot more insight into how the GPUs work at a lower level, and will help you understand what the pitfalls are, and how to avoid premature pessimization when writing your OpenGL code.

5) Focus on what will impress the end user. Sometimes it's easy to lose sight of the end goal, and to become focused on optimizing areas that won't make much of a difference to the end user. Does it matter if we get this shader to execute 2% faster? Or will we make much more of a visual impact by finding a better artist? Find out what the most important areas are, and then focus on improving those first.

ber den Autor und weitere Mitwirkende Kevin Brothaler is the founder of Digipom, a mobile software development shop. He has extensive experience in Android development, and he also manages -Learn OpenGL ES-: <http://www.learnopengles.com/>, an online set of OpenGL tutorials for Android and WebGL.