

Raspberry Pi GPU Audio Video Programming



Jan Newmarch

Apress®

Raspberry Pi GPU Audio Video Programming

Jan Newmarch
Oakleigh, Victoria
Australia

ISBN-13 (pbk): 978-1-4842-2471-7
DOI 10.1007/978-1-4842-2472-4

ISBN-13 (electronic): 978-1-4842-2472-4

Library of Congress Control Number: 2016961518

Copyright © 2017 by Jan Newmarch

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and 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, trademarks, 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.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director: Welmoed Spahr

Lead Editor: Steve Anglin

Technical Reviewer: Chaim Krause

Editorial Board: Steve Anglin, Pramila Balan, Laura Berendson, Aaron Black, Louise Corrigan,
Jonathan Gennick, Robert Hutchinson, Celestin Suresh John, Nikhil Karkal, James Markham,
Susan McDermott, Matthew Moodie, Natalie Pao, Gwenan Spearing

Coordinating Editor: Mark Powers

Copy Editor: Kim Wimpsett

Compositor: SPi Global

Indexer: SPi Global

Artist: SPi Global

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. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

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.

Any source code or other supplementary materials referenced by the author in this text are 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/. Readers can also access source code at SpringerLink in the Supplementary Material section for each chapter.

Printed on acid-free paper

*To my wife, Linda Cai, and my daughter, Kathryn, who let me play with
my computing toys without complaint.*

Contents at a Glance

About the Author	xvii
About the Technical Reviewer	xix
Acknowledgments	xxi
Introduction	xxiii
■ Chapter 1: Introduction to the Raspberry Pi.....	1
■ Chapter 2: Khronos Group	7
■ Chapter 3: Compiling Programs for the Raspberry Pi	9
■ Chapter 4: Dispmanx on the Raspberry Pi.....	21
■ Chapter 5: EGL on the Raspberry Pi.....	27
■ Chapter 6: OpenGL ES on the Raspberry Pi	41
■ Chapter 7: OpenMAX on the Raspberry Pi Concepts	79
■ Chapter 8: OpenMAX Components on the Raspberry Pi	83
■ Chapter 9: OpenMAX on the Raspberry Pi State	113
■ Chapter 10: OpenMAX IL Client Library on the Raspberry Pi.....	129
■ Chapter 11: OpenMAX Buffers on the Raspberry Pi	137
■ Chapter 12: Image Processing on the Raspberry Pi.....	153
■ Chapter 13: OpenMAX Video Processing on the Raspberry Pi	181
■ Chapter 14: OpenMAX Audio on the Raspberry Pi.....	203
■ Chapter 15: Rendering OpenMAX to OpenGL on the Raspberry Pi.....	245

■ Chapter 16: Playing Multimedia Files on the Raspberry Pi	261
■ Chapter 17: Basic OpenVG on the Raspberry Pi	325
■ Chapter 18: Text Processing in OpenVG on the Raspberry Pi	349
■ Chapter 19: Overlays on the Raspberry Pi.....	375
Index.....	435

Contents

About the Author	xvii
About the Technical Reviewer	xix
Acknowledgments	xxi
Introduction	xxiii
■ Chapter 1: Introduction to the Raspberry Pi.....	1
Introduction	1
Programming Style	5
Conclusion	6
Resources	6
■ Chapter 2: Khronos Group	7
Role of Khronos	7
OpenGL ES	7
OpenVG	8
EGL	8
OpenMAX	8
Conclusion	8
Resources	8
■ Chapter 3: Compiling Programs for the Raspberry Pi	9
Generic Information	9
Versions of gcc.....	10
C99 or C11 Compilers	10
gcc Compile Flags	11

Includes	14
Dispmanx	14
EGL	14
OpenMAX	14
OpenGL ES	15
OpenVG	15
Libraries	15
Dispmanx	15
EGL	15
OpenMAX	15
OpenGL ES	16
OpenVG	16
Sample Makefiles	16
Dispmanx	16
EGL	17
OpenMAX	17
OpenVG	18
Using pkg-config	18
Conclusion	19
■ Chapter 4: Dispmanx on the Raspberry Pi	21
Building Programs.....	21
Accessing the GPU	21
Garbage Collection	24
Screen Capture	25
Conclusion	25
Resources	25
■ Chapter 5: EGL on the Raspberry Pi	27
Building Programs	27
Overview	27

Initializing EGL	28
Creating a Rendering Context	30
Creating an EGL Drawing Surface with Dispmanx	33
Garbage Collection	37
Conclusion	39
Resources	39
■ Chapter 6: OpenGL ES on the Raspberry Pi	41
Building Programs.....	41
OpenGL ES	42
The esUtil Functions	43
Vertices	46
Shaders	47
Minimal Vertex Shader	48
Minimal Fragment Shader	48
Loading the Shaders	49
Creating the Program Object	50
Drawing Something	51
Putting It Together	53
Drawing an Opaque Red Triangle	53
A Colored Triangle	57
Varyings	57
Passing Multiple Attributes to the Vertex Shader	58
Drawing Squares and Other Shapes	59
Which to Choose?	62
Textures	62
TGA Files	62
Mipmaps	63
Creating a Texture Object	63

Texture Attributes and the Shaders	63
Drawing the Texture	65
Complete Code for Drawing an Image	67
Animation: Rotating an Image	71
Matrices	71
esUtil	71
Invoking Animation	72
Uniform Parameters: The Rotation Matrix	72
Conclusion.....	78
Resources	78
■ Chapter 7: OpenMAX on the Raspberry Pi Concepts	79
OpenMAX Overview	79
OpenMAX Concepts	80
Conclusion	81
Resources	81
■ Chapter 8: OpenMAX Components on the Raspberry Pi	83
OpenMAX IL Components	83
Getting a List of Components	84
Getting a Handle to a Component	87
Port Information	88
Getting and Setting Parameters	89
Getting Port Information	90
Getting Information About Specific Types of Port	92
Detailed Audio Port Information	93
Conclusion	111
Resources	111
■ Chapter 9: OpenMAX on the Raspberry Pi State	113
Building Programs.....	113
Component States	113

Loaded to Idle	115
A Nonworking Program	115
What's Wrong with It?	118
Idle to Executing	122
When Do State Changes Occur?	122
What to Do When a State Change Occurs?	127
Conclusion	128
Resources	128
■ Chapter 10: OpenMAX IL Client Library on the Raspberry Pi.....	129
Building Programs.....	129
Public Functions	130
Creating a Client	131
Changing Component State	132
Waiting for Events	135
Debugging Clients	135
Conclusion	136
Resources	136
■ Chapter 11: OpenMAX Buffers on the Raspberry Pi	137
Building Programs.....	137
Buffers	137
OpenMAX Buffer Allocation	138
The IL Client Library	138
Writing to and Reading from Buffers	142
Sequence of Actions	143
The EOS Flag	144
A Hiccup: Port Settings Changed	145
Example Image Decoding	146
Conclusion	152
Resources	152

■ Chapter 12: Image Processing on the Raspberry Pi.....	153
Building Programs.....	153
Image Components	153
Image Formats	154
Decoding a JPEG Image	157
Tunneling	158
Rendering an Image Using Tunnelling	162
Rendering an Image Without Tunneling	168
Conclusion	180
Resources	180
■ Chapter 13: OpenMAX Video Processing on the Raspberry Pi	181
Building Programs.....	181
Video Components	181
Video Formats	182
Decoding an H.264 File	186
Rendering an H.264 Video	193
Going Full-Screen	201
Conclusion	201
Resources	201
■ Chapter 14: OpenMAX Audio on the Raspberry Pi.....	203
Building Programs.....	203
Audio Components	204
Audio Formats	204
Decoding an Audio File Using audio_decode	213
Rendering PCM Data	213
PCM Data	213
Choosing an Output Device	214
Setting PCM Format	215

Decoding an MP3 File Using FFmpeg or Avconv	221
Rendering MP3 Using FFmpeg or LibAV and OpenMAX	226
Rendering MP3 with ID3 Extensions Using FFmpeg or LibAV and OpenMAX	234
Conclusion	244
Resources	244
■ Chapter 15: Rendering OpenMAX to OpenGL on the Raspberry Pi.....	245
EGLImage	245
OpenMAX	245
OpenGL	246
eglCreateImageKHR	246
Broadcom GPU	246
Components.....	247
Threads	247
Rendering a Video into an OpenGL ES Texture	247
Conclusion	259
Resources	259
■ Chapter 16: Playing Multimedia Files on the Raspberry Pi	261
Desperate Debugging	261
Multimedia Files	261
Demuxing Multimedia Files	263
Decoding the Audio Stream Using LibAV	263
Rendering an Audio Stream	266
Rendering the Video Stream (at Full Speed).....	275
Rendering the Video Stream (with Scheduling).....	277
Rendering Both Audio and Video	299
Use the Audio Decoder?	300
Conclusion	324
Resources	324

■ Chapter 17: Basic OpenVG on the Raspberry Pi	325
Building Programs.....	325
Introduction	325
Dispmanx and EGL	326
OpenVG Pipeline	330
Drawing a Pink Triangle	331
Drawing Standard Shapes	337
Images	341
Creating a VGImage from Memory Data	342
Drawing an Image	342
Drawing an Image Using OpenVG Calls	344
Conclusion	348
Resources	348
■ Chapter 18: Text Processing in OpenVG on the Raspberry Pi	349
Building Programs.....	349
Drawing Text	350
Drawing Text Using Cairo	350
Drawing Text Using Pango	355
FreeType	361
Font Faces	361
Paths and Glyphs	362
Drawing a FreeType Outline	363
Drawing a FreeType Bitmap	367
VGFonts and Multiple Characters	368
VGDrawGlyph	368
vgDrawGlyphs	371
Glyph Metrics	371
Conclusion	373
Resources	373

■ **Chapter 19: Overlays on the Raspberry Pi..... 375**

 Introduction 375

 Dispmanx Layers 375

 Drawing OpenGL ES on Top of an OpenMAX Image 376

 Drawing OpenGL ES on Top of an OpenMAX Video 387

 Drawing OpenVG on Top of an OpenMAX Image 398

 Drawing OpenVG on Top of an OpenMAX Video 408

 Drawing Text on Top of an OpenMAX Video Using Pango 419

 Conclusion 433

Index..... 435

About the Author



Jan Newmarch is the head of the ICT (Higher Education) at Box Hill Institute, an adjunct professor at Canberra University, and an adjunct lecturer in the School of Information Technology, Computing, and Mathematics at Charles Sturt University. He is interested in more aspects of computing than he has time to pursue, but the major thrust over the last few years has developed from user interfaces under Unix into Java, to the Web, and then into general distributed systems. Jan has developed a number of publicly available software systems in these areas. For the last few years, he has been looking at sound for Linux systems and programming the Raspberry Pi's GPU. He is now exploring aspects of the IoT. He lives in Melbourne, Australia, and enjoys the food and culture there but is not so impressed by the weather.

About the Technical Reviewer



Chaim Krause currently lives in Leavenworth, Kansas, where the U.S. Army employs him as a simulation specialist. In his spare time, he likes to play PC games and occasionally develops his own. He has recently taken up the sport of golf to spend more time with his significant other, Ivana. Although he holds a bachelor's of science degree in political science from the University of Chicago, Chaim is an autodidact when it comes to computers, programming, and electronics. He wrote his first computer game in BASIC on a Tandy Model I Level I and stored the program on a cassette tape. Amateur radio introduced him to electronics, while the Arduino and the Raspberry Pi provided a medium to combine computing, programming, and electronics into one hobby.

Acknowledgments

This book was written with the aid of the extensive documentation on all topics related to computing and the helpful nature of the many thousands of programmers contributing to the Web as a knowledge source. The book is based on the engineering and development work of those behind the Raspberry Pi and of course of the organizations creating the hardware, the software, and the APIs to run programs on GPUs.

Introduction

The Raspberry Pi was created to meet a need to help younger people become involved in the IT field. As a low-cost computer, it can be used, experimented with, broken, and replaced. Initially expected to sell perhaps a few thousand, it has now sold more than 10 million units, and it is used for all sorts of activities, right up to building supercomputers held together with LEGO blocks.

A surprising and early development was to use the Raspberry Pi as a media center, where the GPU was used by software such as `omxplayer` to render 1080p movies to HDMI screens. This has led to software such as Kodi and OpenElec. My partner is Chinese, so for entertainment we sing karaoke. We have several karaoke players, and they all have different features. I wanted to combine all these features into a single application running on low-cost hardware, and the Raspberry Pi was a natural choice. But it turned out to be really, really *hard*. Some documentation about programming the Raspberry Pi's GPU was available but very inaccessible. So, as I explored what the GPU was capable of, I documented it all, and it turned into this book.

Who This Book Is For

This book is aimed at programmers familiar with C programming who want to write programs using the Raspberry Pi's GPU. It does not assume a prior background in graphics programming, but it does assume that you understand enough about the Unix/Linux programming environment to install and build software from source using tools such as `make`. Of course, it assumes you have a Raspberry Pi and are comfortable setting it up and installing software packages.

What This Book Covers

There are a multitude of APIs and programming styles for programming graphics systems. The Raspberry Pi supports a few of these: OpenGL ES for 3D programming, OpenVG for 2D vector graphics programming, and OpenMAX for video and audio. This book attempts to deal with all of these. The book deals with the basic concepts of each style of programming and illustrates them with many complete, working programs. You are fortunate in that although the Raspberry Pi has been through a number of hardware revisions, the GPU has remained the same, so the programs run unchanged on all versions of the hardware.

OpenGL ES is a well-documented API. I deal with it here in only enough depth to give simple programs showing the basic concepts of working on the Raspberry Pi. Then you can move on to other resources. The OpenMAX and OpenVG systems have not been documented anywhere outside of formal specifications and multiple PowerPoint presentations. Describing those systems form the major part of the book. In particular, the boundaries between CPU and GPU programming are discussed in detail, such as where text handling or audio decoding should occur.

The final chapter looks at the interplay between these different APIs: drawing dynamic text using OpenVG on top of a video played by OpenMAX, for example.

By the end of this book, you will be familiar with the current APIs using the Raspberry Pi's GPU. Source code for all the programs is accessible via the book's Apress.com product page, which is at www.apress.com/9781484224717.