

OpenGL[®] ES 2.0 Programming Guide



Aaftab Munshi ■ Dan Ginsburg ■ Dave Shreiner
Foreword by Neil Trevett, President, Khronos Group

OpenGL[®] ES 2.0

Programming Guide

OpenGL ES 2.0 Programming Guide

Table of Contents

Contents

List of Figures

List of Examples

List of Tables

Foreword

Preface

- Intended Audience

- Organization of the Book

- Example Code and Shaders

- Errata

Acknowledgments

About the Authors

1. Introduction to OpenGL ES 2.0

- What Is OpenGL ES?

- OpenGL ES 2.0

 - Vertex Shader

 - Primitive Assembly

 - Rasterization

Table of Contents

Fragment Shader

Per-Fragment Operations

OpenGL ES 2.0 and OpenGL ES 1.x Backward Compatibility

EGL

Programming with OpenGL ES 2.0

Libraries and Include Files

EGL Command Syntax

OpenGL ES Command Syntax

Error Handling

Flush and Finish

Basic State Management

Further Reading

2. Hello Triangle: An OpenGL ES 2.0 Example

Code Framework

Where to Download the Examples

Hello Triangle Example

Building and Running the Examples

Using the OpenGL ES 2.0 Framework

Creating a Simple Vertex and Fragment Shader

Compiling and Loading the Shaders

Creating a Program Object and Linking the Shaders

Setting the Viewport and Clearing the Color Buffer

Loading the Geometry and Drawing a Primitive

Table of Contents

Displaying the Back Buffer

3. An Introduction to EGL

Communicating with the Windowing System

Checking for Errors

Initializing EGL

Determining the Available Surface Configurations

Querying EGLConfig Attributes

Letting EGL Choose the Config

Creating an On-Screen Rendering Area: The EGL Window

Creating an Off-Screen Rendering Area: EGL Pbuffers

Creating a Rendering Context

Making an EGLContext Current

Putting All Our EGL Knowledge Together

Synchronizing Rendering

4. Shaders and Programs

Shaders and Programs

Creating and Compiling a Shader

Creating and Linking a Program

Uniforms and Attributes

Getting and Setting Uniforms

Getting and Setting Attributes

Shader Compiler and Shader Binaries

5. OpenGL ES Shading Language

Table of Contents

OpenGL ES Shading Language Basics

Variables and Variable Types

Variable Constructors

Vector and Matrix Components

Constants

Structures

Arrays

Operators

Functions

Built-In Functions

Control Flow Statements

Uniforms

Attributes

Varyings

Preprocessor and Directives

Uniform and Varying Packing

Precision Qualifiers

Invariance

6. Vertex Attributes, Vertex Arrays, and Buffer Objects

Specifying Vertex Attribute Data

Constant Vertex Attribute

Vertex Arrays

Declaring Vertex Attribute Variables in a Vertex Shader

Table of Contents

Binding Vertex Attributes to Attribute Variables in a Vertex Shader

Vertex Buffer Objects

Mapping Buffer Objects

Performance Tip

7. Primitive Assembly and Rasterization

Primitives

Triangles

Lines

Point Sprites

Drawing Primitives

Performance Tips

Primitive Assembly

Coordinate Systems

Clipping

Perspective Division

Viewport Transformation

Rasterization

Culling

Polygon Offset

8. Vertex Shaders

Vertex Shader Overview

Vertex Shader Built-In Variables

Precision Qualifiers

ES 2.0 Vertex Shader Limitations

Table of Contents

Vertex Shader Examples

- A Simple Vertex Shader

- Lighting in a Vertex Shader

Generating Texture Coordinates

Vertex Skinning

- OpenGL ES 1.1 Vertex Pipeline as an ES 2.0 Vertex Shader

9. Texturing

Texturing Basics

- 2D Textures

- Cubemap Textures

- Texture Objects and Loading Textures

- Texture Filtering and Mipmapping

- Automatic Mipmap Generation

- Texture Coordinate Wrapping

- Using Textures in the Fragment Shader

- Example of Using a Cubemap Texture

Compressed Textures

Texture Subimage Specification

Copying Texture Data from the Color Buffer

Optional Extensions

- 3D Textures

- Ericsson Texture Compression (ETC)

- Floating-Point Textures

- Non-Power-of-Two Textures

Table of Contents

10. Fragment Shaders

Fixed Function Fragment Shaders

Fragment Shader Overview

Built-In Special Variables

Built-In Constants

Precision Qualifiers

ES 2.0 Fragment Shader Limitations

Implementing Fixed Function Techniques Using Shaders

Multitexturing

Fog

Alpha Test (Using Discard)

User Clip Planes

11. Fragment Operations

Buffers

Requesting Additional Buffers

Clearing Buffers

Using Masks to Control Writing to Framebuffers

Fragment Tests and Operations

Using the Scissor Test

Stencil Buffer Testing

Depth Buffer Testing

Blending

Dithering

Multisampled Antialiasing

Table of Contents

Reading and Writing Pixels to the Framebuffer

12. Framebuffer Objects

Why Framebuffer Objects?

Framebuffer and Renderbuffer Objects

Choosing a Renderbuffer Versus a Texture as a Framebuffer Attachment

Framebuffer Objects Versus EGL Surfaces

Creating Framebuffer and Renderbuffer Objects

Using Renderbuffer Objects

Using Framebuffer Objects

Attaching a Renderbuffer as a Framebuffer Attachment

Attaching a 2D Texture as a Framebuffer Attachment

Attaching an Image of a 3D Texture as a Framebuffer Attachment

Checking for Framebuffer Completeness

Deleting Framebuffer and Renderbuffer Objects

Deleting Renderbuffer Objects That Are Used as Framebuffer Attachments

Reading Pixels and Framebuffer Objects

Examples

Performance Tips and Tricks

13. Advanced Programming with OpenGL ES 2.0

Per-Fragment Lighting

Lighting with a Normal Map

Lighting Shaders

Lighting Equations

Environment Mapping

Table of Contents

Particle System with Point Sprites

- Particle System Setup

- Particle System Vertex Shader

- Particle System Fragment Shader

Image Postprocessing

- Render-to-Texture Setup

- Blur Fragment Shader

- Light Bloom

Projective Texturing

- Projective Texturing Basics

- Matrices for Projective Texturing

- Projective Spotlight Shaders

Noise Using a 3D Texture

- Generating Noise

- Using Noise

Procedural Texturing

- A Procedural Texture Example

- Antialiasing of Procedural Textures

- Further Reading on Procedural Textures

14. State Queries

- OpenGL ES 2.0 Implementation String Queries

- Querying Implementation-Dependent Limits

- Querying OpenGL ES State

- Hints

Table of Contents

Entity Name Queries

Nonprogrammable Operations Control and Queries

Shader and Program State Queries

Vertex Attribute Queries

Texture State Queries

Vertex Buffer Queries

Renderbuffer and Framebuffer State Queries

15. OpenGL ES and EGL on Handheld Platforms

Handheld Platforms Overview

Online Resources

C++ Portability

OpenKODE

Platform-Specific Shader Binaries

Targeting Extensions

A. GL_HALF_FLOAT_OES

16-Bit Floating-Point Number

Converting Float to Half-Float

B. Built-In Functions

Angle and Trigonometry Functions

Exponential Functions

Common Functions

Geometric Functions

Table of Contents

Matrix Functions

Vector Relational Functions

Texture Lookup Functions

Derivative Functions

C. Shading Language Grammar

D. ES Framework API

Framework Core Functions

Transformation Functions

E. OpenGL ES 2.0 on the iPhone 3GS

Getting Started with the iPhone SDK 3.0

Getting the Sample Code for the iPhone

Building the Sample Code Using Xcode

Porting the Sample Code to the iPhone

Objective C

Creating an OpenGL ES 2.0 Rendering Surface

Using a Framebuffer Object for Rendering

Discussion of a Complete Example

EAGLView.h Header File

EAGLView.m Source File

Transitioning from OpenGL ES 1.1 to OpenGL ES 2.0

Conclusion

Index