



Programming Guide

The Official Guide to Learning Vulkan



Graham Sellers

With contributions from John Kessenich

VulkanTM

Programming Guide

Vulkan Programming Guide: The Official Guide to Learning Vulkan

Table of Contents

Cover

Title Page

Copyright Page

Contents

Figures

Tables

Listings

About This Book

 About the Sample Code

 Errata

Acknowledgments

About the Author

1 Overview of Vulkan

 Introduction

 Instances, Devices, and Queues

 The Vulkan Instance

 Vulkan Physical Devices

 Physical Device Memory

 Device Queues

 Creating a Logical Device

 Object Types and Function Conventions

Table of Contents

Managing Memory

Multithreading in Vulkan

Mathematical Concepts

 Vectors and Matrices

 Coordinate Systems

Enhancing Vulkan

 Layers

 Extensions

Shutting Down Cleanly

Summary

2 Memory and Resources

Host Memory Management

Resources

 Buffers

 Formats and Support

 Images

 Resource Views

 Destroying Resources

Device Memory Management

 Allocating Device Memory

 Host Access to Device Memory

 Binding Memory to Resources

 Sparse Resources

Summary

3 Queues and Commands

Device Queues

Creating Command Buffers

Recording Commands

Table of Contents

Recycling Command Buffers

Submission of Commands

Summary

4 Moving Data

Managing Resource State

Pipeline Barriers

Global Memory Barriers

Buffer Memory Barriers

Image Memory Barriers

Clearing and Filling Buffers

Clearing and Filling Images

Copying Image Data

Copying Compressed Image Data

Stretching Images

Summary

5 Presentation

Presentation Extension

Presentation Surfaces

Presentation on Microsoft Windows

Presentation on Xlib-Based Platforms

Presentation with Xcb

Swap Chains

Full-Screen Surfaces

Performing Presentation

Cleaning Up

Summary

6 Shaders and Pipelines

Table of Contents

An Overview of GLSL

An Overview of SPIR-V

Representation of SPIR-V

Handing SPIR-V to Vulkan

Pipelines

Compute Pipelines

Creating Pipelines

Specialization Constants

Accelerating Pipeline Creation

Binding Pipelines

Executing Work

Resource Access in Shaders

Descriptor Sets

Binding Resources to Descriptor Sets

Binding Descriptor Sets

Uniform, Texel, and Storage Buffers

Push Constants

Sampled Images

Summary

7 Graphics Pipelines

The Logical Graphics Pipeline

Renderpasses

The Framebuffer

Creating a Simple Graphics Pipeline

Graphics Shader Stages

Vertex Input State

Input Assembly

Tessellation State

Table of Contents

- Viewport State
- Rasterization State
- Multisample State
- Depth and Stencil State
- Color Blend State

- Dynamic State

- Summary

8 Drawing

- Getting Ready to Draw

- Vertex Data

- Indexed Draws

 - Index-Only Rendering

 - Reset Indices

- Instancing

- Indirect Draws

- Summary

9 Geometry Processing

- Tessellation

 - Tessellation Configuration

 - Tessellation Variables

 - Tessellation Example: Displacement Mapping

- Geometry Shaders

 - Cutting Primitives

 - Geometry Shader Instancing

- Programmable Point Size

- Line Width and Rasterization

- User Clipping and Culling

- The Viewport Transformation

Table of Contents

Summary

10 Fragment Processing

Scissor Testing

Depth and Stencil Operations

Depth Testing

Stencil Testing

Early Fragment Tests

Multisample Rendering

Sample Rate Shading

Multisample Resolves

Logic Operations

Fragment Shader Outputs

Color Blending

Summary

11 Synchronization

Fences

Events

Semaphores

Summary

12 Getting Data Back

Queries

Executing Queries

Timing Queries

Reading Data with the Host

Summary

13 Multipass Rendering

Input Attachments

Table of Contents

Attachment Contents

Attachment Initialization

Render Areas

Preserving Attachment Content

Secondary Command Buffers

Summary

Appendix: Vulkan Functions

Glossary

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

Table of Contents

U

V

W

Index