Vulkan – By Example

Explore the revolutionizing next-generation high-performance 3D graphics Vulkan API with {N} mind-blowing real-time graphics recipes.

[Tag #1] Learn Vulkan API through examples and build high performance applications in a step-by-step manner.

[Tag #2] Learn basics to build rich 3D media, VR applications and 4K video processing in real time.

[Tag #3] Get hands on experience with Vulkan debugging and performance tuning tools

# Audience

*This book is especially designed for the beginner’s and intermediate professionals to wants step into Vulkan programming. We expect the target readers of this book has some experience with 3D Graphics programming before and this book help them to transition to a new next-gen graphics API in step-by-step manner. The readers must have practical C/C++ programming knowledge and should be familiar with building and debugging an application.*

# Mission

*The goal of this book is to introduce the core concepts of next generation graphics API – Vulkan. The book is divided into three sections – Beginner, Intermediate and Advanced.*

*This book walks through beginner level with baby steps teaching the reader the basics of Vulkan API to build the first Vulkan application.*

*We will quickly moves to the intermediate level where we will learn in-depth concept of Vulkan programming by building several examples. This section introduces practical usage of 3D scene, advance shader, GPU programming and graphics techniques all applied to solve various stages of an end to end graphics application development to deliver stunning visual experience.*

*In the third section, the approach become more focused application development using Vulkan API. Here, we will utilize the knowledge gained in the first two section to build 3 real time 3D graphics and compute applications that includes a 3D canvas application, an immersive 360 degree view application and last but not least a real-time VR application that offers 3D media content navigation in a virtual world. This section also teaches to debug, profile, improve and tune Vulkan application using Vulkan tools.*

*This book takes the reader on a journey to learn and build stunning graphics application using Vulkan. The reader learns concepts and techniques to build an end-to-end application and keep adding additional features as they progress from one chapter to the next. At the end of the book, the reader will have a fully functional 3D application developed using Vulkan that harness the full potential of GPU to deliver stunning visuals.*

# Objectives and achievements

* *Learn the essentials of Compute and Graphics in Vulkan* *with step-by- step installation and building process.*
* *Vulkan basics – Devices, Queues, Resources, Render pass, Graphics and Compute pipeline.*
* *Learn various types in Vulkan and various GPU memory operation on them.*
* *Scratch the shaders with ample of examples covering Vertex, Fragment, Tessellation and Compute shaders.*
* *Learn scene graphs: 2D/3D scene management, viewport management, transformation, lighting, meshes.*
* *Build a 360 degree virtual VR application and learn to publish and interact on Youtube and Facebook.*
* *Building real time VR application with OpenVR and build 360 degree navigation application.*
* *Learn to add responsive user interface that allows users to navigate inside the graphics application*
* *Learning to debug Vulkan with debugging layers, tune and improve performance with Vulkan tools.*

# Detailed outline

# **Part 1- The Vulkan API basics and first hands-on experience**

The first part of this book cover the basic Vulkan API programming. We begin with a brief introduction to the Vulkan system and show you its distinct features with the successor to the OpenGL API. We will see how to establish a connection with hardware devices to query the available queues, memory types, and capabilities offered. You’ll get a grip on command buffers and acquire the knowledge to record various operation commands into command buffer and submit it to a proper queue for GPU processing. We’ll take a detailed look at memory management and demonstrate the use of buffer and image resources to create drawing textures and image views for the presentation engine and vertex buffers to store geometry information. You'll get a brief overview of SPIR-V, the new way to manage shaders, and you'll learn Render pass and build graphics and compute pipeline.

# Chapter 1: Getting started with Vulkan and build your first graphics pipeline, 45 pages

## Description

*Chapter 1, takes you through the evolution of the Vulkan API, we would understand the Vulkan ecosystem and its architecture. We will also understand the SPIRV and it’s role in the Vulkan ecosystem. In the end of the chapter, we will setup the environment, install the drivers and build a simple example of using Vulkan API to understand overall working of Vulkan application and its execution model.*

## Level

*BASIC*

## Topics covered

1. *Understand the key benefits of Vulkan API compared to other graphics APIs*
2. *Understanding Vulkan’s Architecture, Driver Installation [NVidia], Setup of application with CMake and Execution model*
3. *Understand Devices in Vulkan, queues, presentation window and command buffer management*
4. *Creating Render pass, Frame buffer management and building Graphic pipeline.*
5. *Loading, compiling and linking a shader program with SPIR-V*
6. *First example – Clearing background color.*
7. *Second example – Vulkan Hello world application*

## Skills learned

* *A big picture of what Vulkan is? How it works? What comes under Vulkan ecosystem and it Eco system*
* *Basic setup, installation and execution of Vulkan program.*
* *Basic first two examples shows how to clear the background and draw a simple primitive shape like a Square with each edge with different color interpolating nicely across.*

# Chapter 2: VULKAN BUFFER RESOURCES - 35 pages

## Description

*Chapter 2, provides you with a detailed description of the further basic concepts of the Vulkan programming. This chapter introduces Buffer resource in Vulkan and its management. We will build understanding on Buffers through a number of examples step-by-step such as Buffer view, constants, uniform buffer objects, etc.*

## Level

*MEDIUM*

## Topics covered

1. *Introduction to Buffer Resource in Vulkan, build and using buffer and buffer views.*
2. *Using Descriptors and Descriptor pools in Vulkan and learn to build first vertex buffer in Vulkan*
   1. *Example shows how vertex buffer and used to supply geometry information from CPU to GPU.*
3. *Implementing Uniform Buffer Object in Vulkan*
   1. *Example showing how to club uniforms into a structure and uploaded to GPU.*
4. *Updating constant variable in shader using a fast path*
   1. *Example showing how constant buffers are used in Vulkan*
5. *Introduction to Meshes:*
   1. *Introduction to depth buffer and demonstrating how to build multiple objects with separate pipelines with depth testing enabled. We will demonstrate how to use different depth testing, under a procedurally generate alpha texture.*
   2. *In this example: render a dense mesh and apply Light shading on it*
6. *Rendering millions of meshes object using Vulkan Instancing.*

## Skills learned

* *Buffer resources in Vulkan, buffer and buffer views*
* *Role of descriptor and descriptor pool and their usage*
* *Uniform buffer object in Vulkan*
* *Constant buffer usage*
* *Playing with dense images.*
* *Depth buffer for real 3D perception*
* *Pipeline management in Vulkan*
* *Geometric instancing technique*

# Chapter 3: VULKAN IMAGE RESOURCES - 40 pages

## Description

*Chapter 3, introduce the texture and related operation on it. For more details please follow up Topics covered below:-*

## Level

*MEDIUM*

## Topics covered

1. *Introduction to Texture resources in Vulkan, build and using image and image views.*
2. *Creating linear texture memory in Vulkan*
   1. *Example showing how to use texture in Vulkan and building Linear texture example*
3. *Optimal texture in Vulkan*
   1. *Optimal texture example showing the difference between linear and optimal texture and the difference of performance. Where to use when.*
4. *Buffer and Image operations.*
   1. *Creating Pitch memory*
   2. *Copying data from Buffer to image and Image to buffers on GPU.*
5. *Creating Frame Buffer Object in Vulkan*
   1. *Example showing how to create offline image buffers.*
6. *Synchronization in Vulkan*
   1. *An example showing synchronization in Vulkan through various synchronization primitive in Vulkan – Such as Fence, Semaphore, Barrier etc.*

## Skills learned

* *Texture in Vulkan and it type*
* *Buffer and texture blitting operations*
* *Offline texture generation and processing.*
* *Synchronization in Vulkan*

# **Part 2- Adding eye candies to your application**

*Part 2 of the book goes in depth into exploring various rendering techniques using shader and management of complex scenes that will be helpful in creating the final application product in the last section of this book.*

# Chapter 4: SHADERS – UNDERSTANDING GPU’S LANGUAGE - 50 pages

## Description

*Chapter 4, gives you an in-depth understanding on the shaders programming technique. It discusses various techniques that can be implemented using the vertex, fragment, compute, tessellation and geometry shader, revealing their capabilities. Some of the example are reused to demonstrate the performance benchmark when implemented under different shaders. Apart from Vertex and Fragment shader this chapter would provide some detailed explanation to other shaders like compute, tessellation and geometry.*

## Level

*MEDIUM*

## Topics covered

1. *Vertex Shader (3 pages)*
   1. *Implementing the wobble and ripple effect*
2. *Fragment Shader (3 pages)*
   1. *Blur, Edge detection.*
3. *Tessellation Shader (5 pages)*
   1. *Example showing terrain mesh tessellation based on camera distance from terrain.*
4. *Compute Shader: This chapter requires introduction to compute in general. Like threads, grid, local and global grid index. (15-20 pages)*

***Note: This shader is highly important as it can do many things that other shader can do. Also, compute is the crux of Vulkan the makes it different of older APIs. We tried in the chapter to build several example to reach all different aspect of this shader.***

* 1. *Building you first compute pipeline with simple example*
     1. *Simple buffer copy example showing the GPU bandwidth*
  2. *Another simple Pixelate example showing how to use learning buffer memory to pixelate and present using optimal texture on compute shader.*
  3. *Image processing with compute shader – Blur, Edge detection*
  4. *Ripple and wobble effect etc. Height map examples*
  5. *Compute shade based ray tracing example*

1. *Geometry Shader (3)*
   1. *Producing normal of complex geometry with geometry shader*

## Skills learned

* *Implementing and understanding Vertex Shader*
* *Implementing and understanding Fragment Shader*
* *Implementing and understanding Tessellation Shader*
* *Implementing and understanding Compute Shader*
* *Implementing and understanding Geometry Shader*

# Chapter 5: SCENE GRAPH – 60 Pages

## Description

*Chapter 5, introduces a scene graph paradigm that allows to program and manage complex scenes efficiently. This chapter will help you create a small architecture that allows you to manage multiple scenes. Each scene consists of multiple lights, cameras, and models. This chapter implements a small project where it render an animating 3D mesh object with proper light, camera, reflection and refraction. The statistics of the scene is rendering on the overlay using fonts.*

## Level

*MEDIUM*

## Topics covered

1. *Managing scene with Scene-Graph and Transformation graph (4)*
2. *Implementing orthographic and perspective scene views. (3)*
3. *How to implement Reflect and refraction mapping(Vulkan) (6)*
4. *How to show overlays over 3D rendering scene. (5)*
5. *How to select 3D objects - Ray-trace object intersection (5)*
6. *Implement multiple view ports in Vulkan (5)*
7. *Anti-aliasing technique with (10)*
   1. *MSAA enabled*
   2. *Introduction to FXAA and implementation in Vulkan*
   3. *Multiple sample frame buffer in Vulkan*
8. *Skeleton animation (10)*
9. *Adding shadows (6)*

## Skills learned

* *Understanding Scene-Graphics technique to manage complex scene.*
* *Managing camera under multiple viewports*
* *Learning how to pick objects*
* *Reflect and refraction*
* *Light and shadows*
* *Implementing animation*

# **Part 3- Building and tuning application in Vulkan**

*Part 3, this chapter reuses the knowledge gained from the first two parts of this books and implements three real time practical applications.*

* *The first application builds an endless simple 2D/3D Canvas application.*
* *The second application builds a 360 degree immersive 3D application that can be published on social media and allow to interacted real time.*
* *The second application build a rich 3D navigation VR application, it offer interactive user experience to navigate 3D enrich media contents using OpenVR.*

*At the end of the book, the reader will have a fully functional application to view their media content in 2D or 3D virtual world and navigate them using various user interfaces or navigate them automatically using user predefined path. The user should be able to save the visuals to a video and share it on social media. Hence the reader not only learns how to use Vulkan API, but also understands how to interface it with various other components that makes a complete end-to-end application.*

*In addition to that, as we reach to closing of this book discusses various techniques offered by Vulkan to improve performance as well as several customization options that allows user to customize their viewing experience with the application. For example, adding custom navigation path to navigate the virtual camera automatically in a user defined path, GUI to enable or disable features in the application etc. We will learn how to debug in Vulkan using its debugging layer and various performance tuning tools.*

# Chapter 6: Building an endless 2D/3D canvas application – 10 Pages

## Description

This is first application using Vulkan, in this application we will render an endless 2D/3D canvas application and allow to interact with that.

## Level

*BEGINNER*

## Topics covered

* 1. *Build an endless 2D/3D canvas application*
  2. *Adding personal media content in your application*
  3. *Building cube maps in Vulkan*
  4. *Setting up the OBS and capturing the real time render frame buffer*
  5. *Publishing the immersive 360 degree spherical view on Youtube and Facebook*
  6. *Separating Main and Render thread.* 
     1. *Introduction to render thread to achieve real time rendering example and different required FPS.*

## Skills learned

* *Build 360 degree immersive views on Vulkan*
* *Separating processing jobs in multiple threads.*
* *Cube maps in Vulkan*
* *Learn to capture frame buffer and seamlessly publish and interact real-time on Youtube and Facebook.*

# Chapter 7: Building a 360 degree immersive 3D application using Vulkan – 40 Pages

## Description

This is first application using Vulkan, in this application we will render a render a cube map into a spherical container.

## Level

*INTERMEDIATE*

## Topics covered

1. *Build an endless 2D/3D canvas application*
2. *Adding personal media content in your application*
3. *Building cube maps in Vulkan*
4. *Setting up the OBS and capturing the real time render frame buffer*
5. *Publishing the immersive 360 degree spherical view on Youtube and Facebook*
6. *Separating Main and Render thread.* 
   * 1. *Introduction to render thread to achieve real time rendering example and different required FPS.*

## Skills learned

* *Build 360 degree immersive views on Vulkan*
* *Separating processing jobs in multiple threads.*
* *Cube maps in Vulkan*
* *Learn to capture frame buffer and seamlessly publish and interact real-time on Youtube and Facebook.*

# Chapter 8: Building rich 2D/3D median content navigation VR application – 50 Pages

## Description

*Chapter 8, This chapter implements an advance project that make use of previous small implementation to produce a real time VR application using OpenVR. The application consists of a dense procedure terrains which will be tessellated on run time based on the camera distance from the ground. The terrain contains billboard that contains personal media files and visualize them in an endless terrain with each media file represented as separate tile. At this stage the user should be able to point the application to any folder in their personal computer where they stored their media files and the application automatically visualize them as set of tiles on the display devices, all rendered using Vulkan on GPU. The reader is then introduced to 3D programming using Vulkan and ways to transform their existing 2D tile based application to 3D and add responsive user interface to navigate the 3D world using mouse, keyboard or touch interfaces.*

## Level

*ADVANCE*

## Topics covered

*Building a rich media 2D/3D Navigation application*

1. *Adding billboards on the terrain.*
2. *Implementing the particle effect*
3. *Mechanism to support interactivity in graphics application*
4. *Graphical User Interface*
5. *Navigation techniques to navigate in 2D and 3D world*
6. *Loading personalize media files*
7. *Each billboard will represent a folder or content. Double clicking the folder will take into the respective directory, showing respective contents. Double clicking the content would play the files. Double click with Ctrl key would bring back to parent folder.*
8. *Adding support to take this experience from Desktop to VR using OpenVR*

## Skills learned

* *Design and implement graphics algorithms & techniques using Vulkan (environment mapping, shadow etc)*
* *Implement particle effects & 3D terrain and various configurable parameters to create custom special effects*
* *The reader learns to add GUI interface for enabling/disabling features on the fly*
* *Learns techniques to smoothly navigate the 3D world with user inputs*
* *Add support for Virtual Reality mode to view the 3D world using Open VR kit*

# Chapter 9: Providing customization options to the user - 15 pages

[To Selva] : Is this chapter required, can it be dissolved.

## Description

*Customization plays a big role in productizing an application hence this chapter introduces reader to build a scene editor that allows user to build their own custom 3D navigation path to visualize and automatically navigate the media content in the 3D world.*

## Level

*ADVANCED*

## Topics covered

* *Various customization options that can make the application more personalized for each user*
* *Step by step process to build a 3D scene editor feature in the application*

## Skills learned

* *How to save and load user settings(Parminder will help to use LUA to load and save project settings)*
* *How to smooth the navigation path using B-spline for smooth camera movements*
* *Techniques to save the 3D rendered scenes as a video file to share in social media*

# Chapter 10: Application tuning, debugging and profiling in Vulkan - 15 pages

## Description

*This chapter tunes the application performance and let the user to debug those using Vulkan tools and debugging layers.*

## Level

*ADVANCED*

## Topics covered

* *Techniques to improve performance*
* *Debugging in Vulkan*
  + *Understanding Vulkan debugging layers*
* *Tuning and profiling application performance in Vulkan*
  + *Vulkan tools to view the health of underlying application and improve the performance.*

## Skills learned

* Debugging in Vulkan
* Profiling in Vulkan
* Performance improvement techniques and tips

# Author Bio

**Selvakumar Panneer** is a Senior Graphics SW Architect at Intel with 20+ years of professional product development experience in GPU driver development, 3D graphics/gaming & interactive graphics research. His innovative GPU performance optimizations & power saving techniques are featured in AMD GPUs & Intel Processor Graphics. He started his career as a game developer and worked on 2 end to end game projects. He spent a more than a decade on GPU optimization techniques and graphics driver development. He is currently working at Intel Research labs focused on technologies & algorithms to bring immersive and interactive visual experience.

**Parminder Singh** is a senior engineer at [Black Magic Designs](https://www.blackmagicdesign.com), Singapore. As a passionate 3D architect, he has worked in the fields of network simulations, geo-modeling, navigation, automotive, infotainment system, image processing and post production. His research interests include GPU-based real-time rendering, geospatial terrain rendering, screen-spaced techniques, real-time dynamic shadows, scientific visualization, scene graphs, and anti-aliasing techniques. He is also, an author of [Learning Vulkan](https://www.packtpub.com/application-development/learning-vulkan) and [OpenGL ES 3.0 Cookbook](https://www.packtpub.com/application-development/opengl-es-30-cookbook)

He is an OpenGL ES and Vulkan trainer. Parminder loves to take up challenges related to real-time rendering. His current research and work includes futuristic implementation for next-generation scalable and manageable graphics engines to produce stunning data, user interface, and visualization effects (merging 2D and 3D concepts). His hobbies include cooking, traveling, sharing knowledge, and exploring the possibilities of applied physics and mathematics.

Feel free to reach Parminder at <https://www.linkedin.com/in/parmindersingh18>