The Graphics Pipeline

Geometry Shaders
Tesselation Shaders
Geo.
Shaders

Pipeline – we will get to that later.

There is also a Vulkan Compute Pipeline – we will get to that later.

Here’s what you need to know:
1. The Vulkan Graphics Pipeline is like what OpenGL would call “The State”, or “The Context”.
2. There’s a lot that goes into it.
3. For the most part, the Graphics Pipeline is meant to be immutable – that is, once this combination of state variables is combined into a Pipeline, that Pipeline never gets changed. To make new combinations of state variables, create a new Graphics Pipeline.
4. The shaders get compiled the rest of the way when their Graphics Pipeline gets created.

What is the Vulkan Graphics Pipeline?

Vulkan: A Pipeline Records the Following Items:

- Pipeline Layout: DescriptorSets, PushConstants
- Vertex Shader module
- Fragment Shader module
- Tesselation Shaders, Geometry Shader
- Geometry Processing
- Vertex Input attributes
- Which states are dynamic:
- Dynamic State
- Color Blending Stage
- Alpha Blending Stage
- Fragment Shader Stage
- Vertex Input Stage
- Vertex Shader Stage
- Rasterization
- Viewport
- Depth Stencil State
- Pipeline Stages and what goes into Them
- RenderPass
- Shaders
- Input Assembly
- Vertex Input Stage and what goes into Them
- Dynamic State
- Vertex Input Stage
- Rasterization
- Viewport
- Depth Stencil State
- Fragment Shader Stage
- Vertex Shader Stage
- Tesselation Shaders, Geometry Shader
- Geometry Processing
- Vertex Input Binding
- Vertex Input Attributes
- Specialization info
- Vertex Shader module
- ...
# Primitive Restart Enable

"Primitive Restart Enable" is used with:

- Indexed drawing
- Triangle Fan and "Strip" topologies

If `vpiasci.primitiveRestartEnable` is `VK_TRUE`, then a special "index" indicates that the primitive should start over. This is more efficient than explicitly ending the current primitive and explicitly starting a new primitive of the same type.

- Indexed Vertex Type
  - `VK_INDEX_TYPE_UINT16`: 0 - 65,535
  - `VK_INDEX_TYPE_UINT32`: 0 - 2,147,483,647

If your `VkIndexType` is `VK_INDEX_TYPE_UINT16`, then the special index is `0xffffffff`. If your `VkIndexType` is `VK_INDEX_TYPE_UINT32`, then the special index is `0xffffffff`.  

### Options for vpiasci.topology

- **VK_PRIMITIVE_TOPOLOGY_POINT_LIST**
- **VK_PRIMITIVE_TOPOLOGY_TRIANGLE_LIST**
- **VK_PRIMITIVE_TOPOLOGY_LINE_LIST**
- **VK_PRIMITIVE_TOPOLOGY_TRIANGLE_STRIP**
- **VK_PRIMITIVE_TOPOLOGY_LINE_STRIP**
- **VK_PRIMITIVE_TOPOLOGY_TRIANGLE_FAN**

### What is “Primitive Restart Enable”? 

- Use one `vpvisci.pVertexAttributeDescriptions` array member per element in the structure for the array-of-structures you are using.
- Declare the binding descriptions and attribute descriptions.
- Declare the vertex topology.
- Declare the tessellation shader info.
One Really Good use of Restart Enable is in Drawing Terrain Surfaces with Triangle Strips

Triangle Strip #0:

Triangle Strip #1:

Triangle Strip #2:

...
MultiSampling State

Color Blending State for each Color Attachment

Which Pipeline Variables can be Set Dynamically

Stencil Operations for Front and Back Faces

Uses for Stencil Operations
vkPipelineDepthStencilStateCreateInfo vpdssci =；
vpdssci.sType = VK_STRUCTURE_TYPE_PIPELINE_DEPTH_STENCIL_STATE_CREATE_INFO;
vpdssci.pNext = nullptr;
vpdssci.flags = 0;
vpdssci.depthTestEnable = VK_TRUE;
vpdssci.depthWriteEnable = VK_TRUE;
vpdssci.depthCompareOp = VK_COMPARE_OP_LESS;
VK_COMPARE_OP_NEVER – never succeeds
VK_COMPARE_OP_LESS – succeeds if new depth value is < the existing value
VK_COMPARE_OP_EQUAL – succeeds if new depth value is == the existing value
VK_COMPARE_OP_LESS_OR_EQUAL – succeeds if new depth value is <= the existing value
VK_COMPARE_OP_GREATER – succeeds if new depth value is > the existing value
VK_COMPARE_OP_NOT_EQUAL – succeeds if new depth value is != the existing value
VK_COMPARE_OP_GREATER_OR_EQUAL – succeeds if new depth value is >= the existing value
VK_COMPARE_OP_ALWAYS – always succeeds
vpdssci.depthBoundsTestEnable = VK_FALSE;
vpdssci.front = vsosf;
vpdssci.back = vsosb;
vpdssci.minDepthBounds = 0.;
vpdssci.maxDepthBounds = 1.;
vpdssci.stencilTestEnable = VK_FALSE;

Later on, we will Bind the Graphics Pipeline to the Command Buffer when Drawing

vkCmdBindPipeline(CommandBuffers[nextImageIndex], VK_PIPELINE_BIND_POINT_GRAPHICS, GraphicsPipeline);