logical device

parallel execution

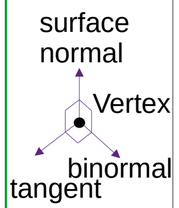
- queue to receive command buffers
- queue to receive command buffers

Command buffer

begin render-pass	**bind pipeline**	**bind descriptor set**	...	**draw Indexed**	**end render-pass**

Vertex buffer

V0	**V1**	**V2**	...	**Vn**
position	position	position		position
color	color	color		color
normal	normal	normal		normal
tangent	tangent	tangent		tangent
binormal	binormal	binormal		binormal
UV	UV	UV		UV



Index buffer

I0	**I1**	**I2**	**I0**	...	**Im**

Uniform buffer

- Mat4 ViewMatrix: transform world space to camera space
- Mat4 ProjectionMatrix: camera space to screen space

Texture (image) (per glTF spec)

- base color texture, “decal”, or also known as Albedo map or diffuse map: independent of viewing angle, the Albedo map has no shadows
- split into down-sampled textures for mipmapping to reduce aliasing at large distances. Aliasing can cause Moiré pattern (= distortion)
- Instead of down-sampling (simple mipmapping), anisotropic filtering (AF) can be used, which accounts for close objects with low grazing angle

normal map (image) (per glTF spec) aka bump map

- surface normals stored as vec3 in RGB channels
- used in lighting calculation for specular reflections

metallic and roughness map (image) (per glTF spec)

- R: Metallic, G: Roughness
- metallic value used in lighting calculation for specular reflections
- metal has no diffuse reflections
- used in lighting calculation for specular reflections: higher roughness scatters reflections more, perfect mirrors have no roughness / no scattering at all

emissive properties map (image) (per glTF spec)

- emissive value needs to be added as per render equation

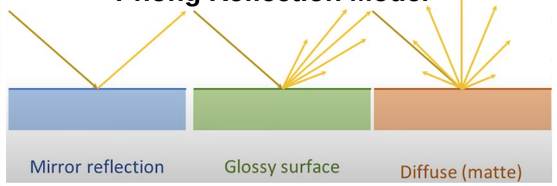
occlusion map (image) (per glTF spec)

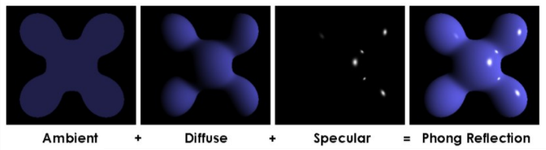
- backed in lighting offline calculated

shadow map (image) (generated in shadow pass)

- calculated in individual sub-pass before color pass
- the scene is rendered from the position of the light
- everything the light can “see” is lit, everything else is occluded

Phong Reflection Model





Display
display controller memory

Presentation
surface on desktop window

Swap Chain

- current image must be acquired
- changed with vertical sync to prevent screen tearing
- buffers/front & back image referred to as **attachments**

front image (framebuffer)
back image (framebuffer)
depth-stencil buffer

render pass for each frame with two sub-passes

pipeline with five shaders: shadow map sub-pass

Vertex shader
for each vertex or for each index:

- get vertex attributes (position, color, normal, texture coordinates “UV”, texture slot, etc.)
- transform vertices from model space to screen space (apply Model-View-Projection MVP)
- output to fragment shader

Tesselation Control Shader

Tesselation Evaluation Shader
add additional vertices (for example for procedural terrain or water simulation)

Geometry Shader

for each primitive (e.g. triangle, or line, or point): add or remove or transform primitive

Clipping: remove everything outside camera frustum

Culling: remove backsides per winding order

rasterization: generate fragments per triangle

Fragment Shader
for each fragment:

- all vertex attributes are interpolated (e.g. normals)
- determine pixel color

write shadow map: black (in shadow) or white (lit)

1

pipeline with five shaders: color sub-pass

Vertex shader
for each vertex or for each index:

- get vertex attributes (position, color, normal, texture coordinates “UV”, texture slot, etc.)
- transform vertices from model space to screen space (apply Model-View-Projection MVP)
- output to fragment shader

Tesselation Control Shader

Tesselation Evaluation Shader
add additional vertices (for example for procedural terrain or water simulation)

Geometry Shader

for each primitive (e.g. triangle, or line, or point): add or remove or transform primitive

Clipping: remove everything outside camera frustum

Culling: remove backsides per winding order

rasterization: generate fragments per triangle

scissor test: fragment in specified rectangle?

Fragment Shader
for each fragment:

- all vertex attributes are interpolated (e.g. normals)
- determine pixel color

resolve: if multiple samples have been taken per pixel, the results need to be merged

blending: merge background with new pixel

depth-stencil: discard fragment if test fails

2