IResourceInstance

ISamplerInstance

IBufferInstance

+resources

+samplers

+con+vertexBuffer

«enumeration»
**IBlendState::
BlendFunction**

«enumeration»
**IBlendState::
BlendFactor**

«enumeration»
**IDepthStencilState::
StencilFunction**

«enumeration»
**IDepthStencilState::
ComparisonFunction**

Frustum

+ Frustum()
+ GetBoundingSphere() :Spheref& {query}
+ GetInverseProjectionMatrix() :Matrix4f& {query}
+ GetViewProjectionMatrix() :Matrix4f& {query}
+ IsInside(Vector3f&, F32) :bool {query}
+ Update(Matrix4f&, Matrix4f&) :void

«enumeration»
Frustum::Point

«enumeration»
Frustum::Plane

IConstantBufferInstance

«enumeration»
**ISampler::
AddressMode**

+addressW

+alpha+blendFunction

+alpha+a+des+source+front+frontFaceStencilFunction:stencilComparisonFunction :

IVertexLayoutInstance

«struct»
ShaderStageState

+ Clear() :void

IRenderTargetInstance

IShaderInstance

IRasterStateInstance

IBlendStateInstance

IDepthStencilStateInstance

+vertexLayout

+shaderStages

+renderTarget

+shader

+rasterState

+blendState

+depthStencilState

Color

+ Color()
+ Color(I32)
+ Color(F32, F32, F32, F32)
+ Get() :F32* {query}
+ GetABGR() :U32 {query}
+ GetARGB() :U32 {query}
+ GetRGBA() :U32 {query}
+ operator!=(Color&) :bool {query}
+ operator[](size_t) :F32 {query}
+ operator[](size_t) :F32&
+ operator==(Color&) :bool {query}
+ Set(U8, U8, U8, U8) :void
+ Set(F32, F32, F32, F32) :void
+ SetABGR(U32) :void
+ SetARGB(U32) :void
+ SetRGBA(U32) :void
+ ZeroColor() :Color&

+color +borderColor

«enumeration»
**ISampler::
ComparisonFunction**

«enumeration»
ISampler::Filter

+comparisonFunction

+filter

«struct»
VertexArray

«struct»
PipelineState

+ Clear() :void

«struct»
IBlendState::Descriptor

+ Default() :Descriptor&
+ Descriptor()

«struct»
**IDepthStencilState::
Descriptor**

+ Default() :Descriptor&
+ Descriptor()

«struct»
PositionColorVertex

+ GetVertexLayoutDescriptor() :IVertexLayout::Descriptor&

«struct»
ISampler::Descriptor

+ Default() :Descriptor&
+ Descriptor()

IResource
+ GetAccessFlags() :U32 {query}
+ GetResourceType() :ResourceType {query}
+ ~IResource()

«enumeration» VertexPrimitive
+vertexPrimitive

IConstantBuffer
+ GetBuffer() :IBufferInstance& {query}
+ ~IConstantBuffer()
+ Set(U32, Matrix4f&) :void
+ Set(U32, Vector4f&) :void
+ Set(U32, Vector3f&) :void
+ Set(U32, Vector2f&) :void
+ Set(U32, F32, U32) :void
+ Set(U32, F32, U32) :void
+ Set(U32, F32, F32, F32, F32) :void

IRenderManager
+ Bind(IBlendStateInstance&) :void
+ Bind(IConstantBufferInstance&, IShader::Stage, U32) :void
+ Bind(IDepthStencilStateInstance&) :void
+ Bind(IRasterStateInstance&) :void
+ Bind(IRenderTargetInstance&) :void
+ Bind(IResourceBufferInstance&, IShader::Stage, U32) :void
+ Bind(ISamplerInstance&, IShader::Stage, U32) :void
+ Bind(IShaderInstance&) :void
+ Bind(ITexture2DInstance&, IShader::Stage, U32) :void
+ Bind(IVertexArray&) :void
+ Draw(DrawState&) :void
+ Finalize() :void
+ GetBlendState(IBlendState::Descriptor&) :IBlendStateInstance&
+ GetConstantBuffer(U32) :IConstantBufferInstance&
+ GetDepthStencilState(IDepthStencilState::Descriptor&) :IDepthStencilStateInstance&
+ GetFrameBuffer(U32) :IRenderTargetInstance& {query}
+ GetRasterState(IRasterState::Descriptor&) :IRasterStateInstance&
+ GetResourceBuffer(U32) :IResourceBufferInstance& {query}
+ GetSampler(ISampler::Descriptor&) :ISamplerInstance&
+ GetSettings() :Settings& {query}
+ GetShader(String&) :IShaderInstance& {query}
+ GetTexture2D(FilePath&) :ITexture2DInstance&
+ GetVertexLayout(U32) :IVertexLayoutInstance& {query}
+ GetViewport(U32) :IViewportInstance& {query}
+ Initialize() :bool
+ ~IRenderManager()
+ IsReady() :bool {query}
+ SetResourceBuffer(U32, size_t) :void
+ SetShader(String&, IShader::Descriptor&) :void
+ SetVertexLayout(U32, IVertexLayout::Descriptor&) :void
+ SetViewport(U32, Ptr, U32, U32, bool) :void
+ Unbind(IResourceBufferInstance&) :void
+ Unbind(ITexture2DInstance&) :void
+ Update(IConstantBufferInstance&) :void
+ Update(IResourceBufferInstance&) :void
+ Update(VertexArray&) :void

IResourceBuffer
+ AcquireIndex() :U32
+ GetBuffer() :IBufferInstance& {query}
+ ~IResourceBuffer()
+ ReleaseIndex(U32) :void
+ Set(U32, void*) :void

«enumeration» IBuffer::Type
+type

«enumeration» IBuffer::GpuWriteAccess
+gpuWriteAccess

«enumeration» IRasterState::CullMode
+cullMode

«enumeration» IRasterState::FillMode
+fillMode

ITexture2DInstance
+depthTarget

«struct» IShader::CompilationDescriptor
+stages

«enumeration» ITexture2D::Type
+type

«enumeration» ITexture2D::Format
+format

«enumeration» IVertexLayout::Element::Format
+format

«enumeration» IVertexLayout::Element::Type
+type

IBlendState
+ GetDescriptor() :Descriptor& {query}
+ ~IBlendState()

IBuffer
+ Add(void*, U32) :void
+ Clear() :void
+ GetCount() :U32 {query}
+ GetDescriptor() :Descriptor& {query}
+ Remove(U32, U32) :void
+ Set(void*, U32, U32) :void
+ Update() :bool

ITexture2D
+ GetDescriptor() :Descriptor& {query}

«struct» DrawState
+ Clear() :void
+ DrawState()

ConstantBuffer
+ ConstantBuffer()
+ GetBuffer() :IBufferInstance& {query}
+ Initialize(IDeviceInstance) :void
+ Set(U32, Matrix4f&) :void
+ Set(U32, Vector4f&) :void
+ Set(U32, Vector3f&) :void
+ Set(U32, Vector2f&) :void
+ Set(U32, F32, U32) :void
+ Set(U32, F32, F32, F32, F32) :void

RenderManager
+ Bind(IBlendStateInstance&) :void
+ Bind(IConstantBufferInstance&, IShader::Stage, U32) :void
+ Bind(IDepthStencilStateInstance&) :void
+ Bind(IRasterStateInstance&) :void
+ Bind(IRenderTargetInstance&) :void
+ Bind(IResourceBufferInstance&, IShader::Stage, U32) :void
+ Bind(ISamplerInstance&, IShader::Stage, U32) :void
+ Bind(IShaderInstance&) :void
+ Bind(ITexture2DInstance&, IShader::Stage, U32) :void
+ Bind(IVertexArray&) :void
+ Draw(DrawState&) :void
+ Finalize() :void
+ GetBlendState(IBlendState::Descriptor&) :IBlendStateInstance&
+ GetConstantBuffer(U32) :IConstantBufferInstance&
+ GetDepthStencilState(IDepthStencilState::Descriptor&) :IDepthStencilStateInstance&
+ GetDevice() :IDeviceInstance& {query}
+ GetFrameBuffer(U32) :IRenderTargetInstance& {query}
+ GetRasterState(IRasterState::Descriptor&) :IRasterStateInstance&
+ GetResourceBuffer(U32) :IResourceBufferInstance& {query}
+ GetSampler(ISampler::Descriptor&) :ISamplerInstance&
+ GetSettings() :Settings&
+ GetSettings() :Settings& {query}
+ GetShader(String&) :IShaderInstance& {query}
+ GetTexture2D(FilePath&) :ITexture2DInstance&
+ GetVertexLayout(U32) :IVertexLayoutInstance& {query}
+ GetViewport(U32) :IViewportInstance& {query}
+ Initialize() :bool
+ IsReady() :bool {query}
+ RenderManager()
+ ~RenderManager()
+ SetResourceBuffer(U32, size_t) :void
+ SetShader(String&, IShader::Descriptor&) :void
+ SetVertexLayout(U32, IVertexLayout::Descriptor&) :void
+ SetViewport(U32, Ptr, U32, U32, bool) :void
+ Unbind(IResourceBufferInstance&) :void
+ Unbind(ITexture2DInstance&) :void
+ Update(IConstantBufferInstance&) :void
+ Update(IResourceBufferInstance&) :void
+ Update(VertexArray&) :void

ResourceBuffer
+ AcquireIndex() :U32
+ GetBuffer() :IBufferInstance& {query}
+ Initialize(size_t, IDeviceInstance) :void
+ ReleaseIndex(U32) :void
+ ResourceBuffer()
+ Set(U32, void*) :void

«struct» IBuffer::Descriptor
+ Descriptor()

«struct» IRasterState::Descriptor
+ Default() :Descriptor&
+ Descriptor()

«struct» IRenderTarget::Descriptor

«struct» IShader::Descriptor
+ Descriptor()

«struct» ITexture2D::Descriptor
+ Descriptor()

«struct» IVertexLayout::Element
+ Element(Type, Format)

DX11BlendState
+ DX11BlendState()
+ ~DX11BlendState()
+ Finalize() :void
+ GetDescriptor() :Descriptor& {query}
+ GetDXBlendState() :ID3D11BlendState* {query}
+ Initialize(Descriptor&) :bool

DX11Buffer
+ Add(void*, U32) :void
+ Clear() :void
+ DX11Buffer()
+ ~DX11Buffer()
+ Finalize() :void
+ GetAccessFlags() :U32 {query}
+ GetCapacity() :U32 {query}
+ GetCount() :U32 {query}
+ GetDescriptor() :Descriptor& {query}
+ GetDXBuffer() :ID3D11Buffer* {query}
+ GetDXShaderResourceView() :ID3D11ShaderResourceView* {query}
+ GetDXUnorderedAccessView() :ID3D11UnorderedAccessView* {query}
+ GetResourceType() :ResourceType {query}
+ Initialize(Descriptor&) :bool
+ Remove(U32, U32) :void
+ Set(void*, U32, U32) :void
+ Update() :bool

DX11Texture2D
+ DX11Texture2D()
+ ~DX11Texture2D()
+ Finalize() :void
+ GetAccessFlags() :U32 {query}
+ GetDescriptor() :Descriptor& {query}
+ GetDXShaderResourceView() :ID3D11ShaderResourceView* {query}
+ GetDXTexture() :ID3D11Texture2D* {query}
+ GetDXUnorderedAccessView() :ID3D11UnorderedAccessView* {query}
+ GetResourceType() :ResourceType {query}
+ Initialize(Descriptor&) :bool
+ Initialize(FilePath&) :bool
+ Initialize(IViewportInstance) :bool

IDepthStencilState
+ GetDescriptor() :Descriptor& {query}
+ ~IDepthStencilState()

IDevice
+ CreateBlendState(IBlendState::Descriptor&) :IBlendStateInstance
+ CreateBuffer(IBuffer::Descriptor&) :IBufferInstance
+ CreateContext(IRenderTarget::Descriptor&) :IRenderTargetInstance
+ CreateDepthStencilState(DepthStencilState::Descriptor&) :IDepthStencilStateInstance
+ CreateRasterState(IRasterState::Descriptor&) :IRasterStateInstance
+ CreateSampler(ISampler::Descriptor&) :ISamplerInstance
+ CreateShader(IShader::Descriptor&) :IShaderInstance
+ CreateTexture2D(ITexture2D::Descriptor&) :ITexture2DInstance
+ CreateTexture2D(FilePath&) :ITexture2DInstance
+ CreateTexture2D(IViewportInstance) :ITexture2DInstance
+ CreateVertexLayout(IVertexLayout::Descriptor&) :IVertexLayoutInstance
+ CreateViewport(IViewport::Descriptor&) :IViewportInstance
+ Finalize() :void
+ GetDescriptor() :Descriptor& {query}
+ GetDeviceType() :DeviceType {query}
+ GetPipeline() :IPipelineInstance& {query}
+ ~IDevice()
+ Initialize() :bool
+ IsReady() :bool {query}

IPipeline
+ BindInput(IBufferInstance&) :void
+ BindInput(IBufferInstance&, U32) :void
+ BindInput(IVertexLayoutInstance&) :void
+ BindOutput(IRenderTargetInstance&) :void
+ BindShader(IShaderInstance&) :void
+ BindShaderConstant(IBufferInstance&, IShader::Stage, U32) :void
+ BindShaderInput(IBufferInstance&, IShader::Stage, U32) :void
+ BindShaderInput(IResourceInstance&, IShader::Stage, U32) :void
+ BindShaderInput(ITexture2DInstance&, IShader::Stage, U32) :void
+ BindShaderOutput(IBufferInstance&, U32) :void
+ BindShaderOutput(ITexture2DInstance&, U32) :void
+ BindShaderSampler(ISamplerInstance&, IShader::Stage, U32) :void
+ BindState(IBlendStateInstance&) :void
+ BindState(IDepthStencilStateInstance&) :void
+ BindState(IRasterStateInstance&) :void
+ Clear() :void
+ ~IPipeline()
+ UnbindShaderInput(IShader::Stage, U32) :void
+ UnbindShaderOutput(U32) :void

IRasterState
+ GetDescriptor() :Descriptor& {query}
+ ~IRasterState()

IRenderTarget
+ Clear(Color&, U8) :void
+ Draw(VertexPrimitive, U32, U32, U32, U32, U32, U32) :void
+ DrawIndirect(VertexPrimitive, U32, U32, U32, U32, U32, U32) :void
+ GetDescriptor() :Descriptor& {query}
+ ~IRenderTarget()

ISampler
+ GetDescriptor() :Descriptor& {query}
+ ~ISampler()

IShader
+ GetDescriptor() :Descriptor& {query}
+ ~IShader()

IVertexLayout
+ GetDescriptor() :Descriptor& {query}
+ ~IVertexLayout()

«enumeration»
ShaderStageState::
<anonymous>

«enumeration»
IRenderManager::
ResourceBufferID

IViewport
+ GetDescriptor() :Descriptor& {query}
+ ~IViewport()
+ Update() :bool

DX11DepthStencilState
+ DX11DepthStencilState()
+ ~DX11DepthStencilState()
+ Finalize() :void
+ GetDescriptor() :Descriptor& {query}
+ GetDXDepthStencilState() :ID3D11DepthStencilState* {query}
+ Initialize(Descriptor&) :bool

DX11Device
+ CreateBlendState(IBlendState::Descriptor&) :IBlendStateInstance
+ CreateBuffer(IBuffer::Descriptor&) :IBufferInstance
+ CreateContext(IRenderTarget::Descriptor&) :IRenderTargetInstance
+ CreateDepthStencilState(DepthStencilState::Descriptor&) :IDepthStencilStateInstance
+ CreateRasterState(IRasterState::Descriptor&) :IRasterStateInstance
+ CreateSampler(ISampler::Descriptor&) :ISamplerInstance
+ CreateShader(IShader::Descriptor&) :IShaderInstance
+ CreateTexture2D(ITexture2D::Descriptor&) :ITexture2DInstance
+ CreateTexture2D(FilePath&) :ITexture2DInstance
+ CreateTexture2D(IViewportInstance) :ITexture2DInstance
+ CreateVertexLayout(IVertexLayout::Descriptor&) :IVertexLayoutInstance
+ CreateViewport(IViewport::Descriptor&) :IViewportInstance
+ DX11Device()
+ ~DX11Device()
+ Finalize() :void
+ GetDescriptor() :Descriptor& {query}
+ GetDeviceType() :DeviceType {query}
+ GetPipeline() :IPipelineInstance& {query}
+ Initialize() :bool
+ IsReady() :bool {query}

DX11Pipeline
+ BindInput(IBufferInstance&) :void
+ BindInput(IBufferInstance&, U32) :void
+ BindInput(IVertexLayoutInstance&) :void
+ BindOutput(IRenderTargetInstance&) :void
+ BindShader(IShaderInstance&) :void
+ BindShaderConstant(IBufferInstance&, IShader::Stage, U32) :void
+ BindShaderInput(IBufferInstance&, IShader::Stage, U32) :void
+ BindShaderInput(IResourceInstance&, IShader::Stage, U32) :void
+ BindShaderInput(ITexture2DInstance&, IShader::Stage, U32) :void
+ BindShaderOutput(IBufferInstance&, U32) :void
+ BindShaderOutput(ITexture2DInstance&, U32) :void
+ BindShaderSampler(ISamplerInstance&, IShader::Stage, U32) :void
+ BindState(IBlendStateInstance&) :void
+ BindState(IDepthStencilStateInstance&) :void
+ BindState(IRasterStateInstance&) :void
+ Clear() :void
+ DX11Pipeline()
+ ~DX11Pipeline()
+ UnbindShaderInput(IShader::Stage, U32) :void
+ UnbindShaderOutput(U32) :void

DX11RasterState
+ DX11RasterState()
+ ~DX11RasterState()
+ Finalize() :void
+ GetDescriptor() :Descriptor& {query}
+ GetDXRasterState() :ID3D11RasterizerState* {query}
+ Initialize(Descriptor&) :bool

DX11RenderTarget
+ Clear(Color&, U8) :void
+ Draw(VertexPrimitive, U32, U32, U32, U32, U32, U32) :void
+ DrawIndirect(VertexPrimitive, U32, U32, U32, U32, U32, U32) :void
+ DX11RenderTarget()
+ ~DX11RenderTarget()
+ Finalize() :void
+ GetDescriptor() :Descriptor& {query}
+ GetDXColorTargets() :ID3D11RenderTargetView*const * {query}
+ GetDXDepthTarget() :ID3D11DepthStencilView* {query}
+ Initialize(Descriptor&) :bool

DX11Sampler
+ DX11Sampler()
+ ~DX11Sampler()
+ Finalize() :void
+ GetDescriptor() :Descriptor& {query}
+ GetDXSamplerState() :ID3D11SamplerState* {query}
+ Initialize(Descriptor&) :bool

DX11Shader
- DX11Shader()
- ~DX11Shader()
+ Finalize() :void
+ GetDescriptor() :Descriptor& {query}
+ GetDXComputeShader() :ID3D11ComputeShader* {query}
+ GetDXDomainShader() :ID3D11DomainShader* {query}
+ GetDXGeometryShader() :ID3D11GeometryShader* {query}
+ GetDXHullShader() :ID3D11HullShader* {query}
+ GetDXPixelShader() :ID3D11PixelShader* {query}
+ GetDXVertexShader() :ID3D11VertexShader* {query}
+ Initialize(Descriptor&) :bool

DX11VertexLayout
+ DX11VertexLayout()
+ ~DX11VertexLayout()
+ Finalize() :void
+ GetDescriptor() :Descriptor& {query}
+ GetDXVertexLayout() :ID3D11InputLayout* {query}
+ Initialize(Descriptor&) :bool

DX11Viewport
+ DX11Viewport()
+ ~DX11Viewport()
+ Finalize() :void
+ GetDescriptor() :Descriptor& {query}
+ GetDXSwapChain() :IDXGISwapChain* {query}
+ Initialize(Descriptor&) :bool
+ Update() :bool

DX11Core
+ GetDXDevice() :ID3D11Device* {query}
+ GetDXDeviceContext() :ID3D11DeviceContext* {query}
+ GetDXViewportList() :Containers::List<D3D11_VIEWPORT>&