

Vulkan Migration Concept

Used Opendgl:

Solution	Class
Graphics/renderer	Renderer → SscreenRect (Struct) CrenderInternal CRenderer CpostproManager struct ShadowMapInternals (GluInt) ShadowMap SilhouetteRenderer SkyManager TerrainOverlay TerrainRenderer CtexturedLineRData VertexArray VertexIndexArray CvertexBuffer CvertexBufferManager WaterManager
Graphics/graphics	CPatchRData ParticleRenderer OverlayRendererInternals OverlayRenderer ImodelDef InstancingModelRenderer ShaderModelDef struct ShaderModel ShaderModelVertexRenderer CDecalRData CCamera CCinemaManager CFontManager CLOSTexture CParticleEmitter CParticleEmitterType GLenum ParseAttribSemantics CShaderManager GLenum ParseComparisonFunc GLenum ParseBlendFunc CshaderProgramARB ShaderProgramGLSL CShaderProgram ShaderProgramFFP ShaderTechnique CTerrainTextureEntry CTerritoryTexture

	CTextRenderer CTextureManagerImpl CTexture
GUI/gui	GuiRenderer CMiniMap CInput CGUI
lowlevel/lib/	OGL functions
lowlevel/lib/res/graphics	GLCursor ogl_tex (OpenGL helper routines)
atlas/GameInterface	ActorViewerImpl
atlas/GameInterface/View.cpp	AtlasViewGame
atlas/AtlasUI/ScenarioEditor	ScenarioEditor

Used OpenGL Commands:

- Glant
- glMatrixMode
- glLoadMatrix
- ogl_HaveExtionsion
- glPixelStorei
- glDepthFunc
- glEnable
- glCullFace
- glFrontFace
- Glint
- glGetIntegerv
- glCullFace
- glDisable
- glPolygonMode
- glRenderPath
- pglActiveTextureARB
- glColor3f
- glLineWidth
- glScissor
- pglBindFramebufferEXT
- glClearColor
- ogl_WarnIfError
- glClear
- glColorMask
- glDepthMask
- glBlendFunc
- pglBlendColorExt
- glStencilFunc
- glStencilOp

- glTexEnvi
- glColor4ub
- glColor3ub
- GLSizei
- glBindTexture
- pglActiveTextureARB
- pglDeleteFramebufferEXT
- pglDeleteFramebuffersEXT
- pglDeleteBuffersARB
- glDeleteTextures
- glGenTextures
- glTexImage2D
- glTexParameterf
- pglFramebuffer2DEXT
- pglCheckFramebufferStatusEXT
- pglBindFramebufferEXT
- pglFramebufferTexture2DEXT
- pglGenerateMipmapEXT
- glDrawArrays
- GL_FLOAT
- GL_TEXTURE0
- GL_TEXTURE1
- GL_UNSIGNED_SHORT
- GL_UNSIGNED_BYTE
- GL_SHORT
- glGetInterv
- GLenum
- pglDrawBuffers
- glDrawElements
- glDepthMask
- GL_DEPTH_COMPONENT16
- GL_DEPTH_COMPONENT24
- GL_DEPTH_COMPONENT32
- GL_DEPTH_COMPONENT
- glReadBuffer
- GL_TEXTURE_CUBE_MAP_POSITIVE_X
- GL_TEXTURE_CUBE_MAP_NEGATIVE_X
- GL_TEXTURE_CUBE_MAP_POSITIVE_Z
- GL_TEXTURE_CUBE_MAP_NEGATIVE_Z
- GL_TEXTURE_CUBE_MAP_POSITIVE_Y
- GL_TEXTURE_CUBE_MAP_NEGATIVE_Y
- glTexEnvi
- glPushMatrix
- glBegin
- glPopMatrix
- glPolygonOffset
- glColor4fv
- glEnd
- glTexSubImage2D
- pglClientActiveTextureARB
- glLoadIdentity
- glTextCoordPointer

- glVertexPointer
- glDisableClientState
- GLfloat
- GL_ARRAY_BUFFER
- glMapBufferARB
- glBindBufferARB
- GL_DYNAMIC_DRAW
- GL_STREAM_DRAW
- GL_REPEAT
- GL_FRAMEBUFFER_COMPLETE_EXT
- glCheckFramebufferStatusEXT
- glBlendEquationEXT
- glInterleavedArrays
- glDrawRangeElementsEXT
- glPointSize
- GL_MINUS_SRC_COLOR
- GL_VERTEX_SHADER
- GL_FRAGMENT_SHADER

Classes uses OpenGL:

- SscreenRect
- CRendererInternals::SetOpenGLCamera
- CRenderer::EnumCaps
- CRenderer::Open
- CRenderer::RenderShadowMap
- CRenderer::RenderPatches
- CRenderer::RenderModels
- CRenderer::RenderTransparentModels
- CRenderer::RenderReflections
- CRenderer::RenderRefractions
- CRenderer::RenderSilhouettes
- CRenderer::RenderParticles
- CRenderer::RenderSubmission
- CRenderer::DisplayFrustum
- CRenderer::RenderTextOverlays
- CRenderer::SetViewport
- CRenderer::RenderScrene
- CRenderer::BindTexture
- CRenderer::LoadAlphaMaps
- Crender::UnloadAlphaMaps
- CPostproManager::Cleanup
- CPostprocManager::RecreateBuffers
- CPostprocManager::ApplyBlurDownscale2x
- CPostprocManager::ApplyBlurGauss
- CPostprocManager::ApplyBlur
- CPostprocManager::CaptureRenderOutput
- CPostprocManager::ReleaseRenderOutput
- CPostprocManager::ApplyEffect
- CpostProcManager::ApplyPostproc
- struct ShadowMapInternals (GluInt)

- ShadowMap::RecreateTexture
- ShadowMap::CreateTexture
- ShadowMap::BeginRender
- ShadowMap::EndRender
- ShadowMap::RenderDebugBounds
- ShadowMap::RenderDebugTexture
- SilhouetteRenderer::RenderDebugOverlays
- SkyManager::LoadSkyTextures
- SkyManager::RenderSky
- TerrainOverlay::RenderBeforeWater
- TerrainOverlay::RenderTile
- TerrainOverlay::RenderTileOutline
- TerrainOverlay::RenderAfterWater
- TerrainRenderer::RenderTerrain
- TerrainRenderer::RenderTerrainOverlayTexture
- TerrainRenderer::RenderTerrainShader
- TerrainRenderer::RenderPatches
- TerrainRenderer::RenderOutlines
- TerrainRenderer::RenderFancyWater
- TerrainRenderer::RenderSimpleWater
- CTexturedLineRData::Render
- VertexArray::AddAttribute
- VertexArray::Attribute::GetIterator<CVector3D>
- VertexArray::Attribute::GetIterator<CVector4D>
- VertexArray::Attribute::GetIterator<float[2]>
- VertexArray::Attribute::GetIterator<SColor3ub>
- VertexArray::Attribute::GetIterator<SColor4ub>
- VertexArray::Attribute::GetIterator<u16>
- VertexArray::Attribute::GetIterator<u16[2]>
- VertexArray::Attribute::GetIterator<u8>
- VertexArray::Attribute::GetIterator<u8[4]>
- VertexArray::Attribute::GetIterator<short>
- VertexArray::Attribute::GetIterator<short[2]>
- VertexArray::Layout
- VertexIndexArray::VertexIndexArray
- CVertexBuffer::CVertexBuffer
- CVertexBuffer::~CVertexBuffer
- CVertexBuffer::Bind
- CVertexBuffer::Unbind
- CVertexBuffer::UseStreaming
- CVertexBufferManager::Allocate
- WaterManager::~WaterManager
- WaterManager::LoadWaterTextures
- WaterManager::Resize
- WaterManager::ReloadWaterNormalTextures
- WaterManager::UnloadWaterTextures
- WaterManager::CreateWaveMeshes
- WaterManager::RenderWaves
- CPatchRData::RenderBases
- CPatchRData::RenderBlends
- CPatchRData::RenderStreams
- CPatchRData::RenderOutline

- CPatchRData::RenderWater
- ParticleRenderer::RenderParticles
- OverlayRendererInternals::OverlayRendererInternals
- OverlayRenderer::RenderOverlaysBeforeWater
- OverlayRenderer::RenderTexturedOverlayLinux
- OverlayRenderer::RenderQuadOverlays
- OverlayRenderer::RenderForegroundOverlays
- OverlayRenderer::RenderSphereOverlays
- ImodelDef::ImodelDef
- InstancingModelRenderer::RenderModel
- ShaderModelDef::ShaderModelDef
- struct ShaderModel
- ShaderModelVertexRenderer::CreateModelData
- ShaderModelVertexRenderer::PrepareModelDef
- ShaderModelVertexRenderer::RenderModel
- CDecalRData::CDecalRData
- CdecalRData::RenderDecals
- CCamera::Render
- CCinemaManager::DrawSpline
- CCinemaManager::DrawNodes
- CCinemaManager::DrawBars
- CFontManager::ReadFont
- CLOSTexture::CreateShader
- CLOSTexture::DeleteTexture
- CLOSTexture::InterpolateLOS
- CLOSTexture::ConstructTexture
- CLOSTexture::RecomputeTexture
- CParticleEmitter::CParticleEmitter
- CParticleEmitter::UpdateArrayData
- CParticleEmitter::RenderArray
- CParticleEmitterType::LoadXML
- GLenum ParseAttribSemantics
- CShaderManager::NewProgram
- GLenum ParseComparisonFunc
- GLenum ParseBlendFunc
- CshaderProgramARB (Bind, Unbind, BindTexture, Uniform, Compile, Link, Reload, VertexPointer, NormalPointer, ColorPointer, TexCoordPointer, VertexAttribPointer, VertexAttribIPointer)
- ShaderProgramGLSL::ShaderProgramGLSL
- CShaderProgram::TexCoordPointer
- CShaderProgram::BindClientStates
- CShaderProgram::UnbindClientStates
- ShaderProgramFFP
- ShaderTechnique::Bind
- ShaderTechnique::Unbind
- CTerrainTextureEntry::LoadAlphaMaps
- CTerritoryTexture::ConstructTexture
- CTerritoryTexture::RecomputeTexture
- CTextRenderer::Render
- CTextureManagerImpl::CTextureManagerImpl
- CTextureManagerImpl::CreateTexture
- CTextureManagerImpl::LoadTexture

- CTexture::~Texture
- CTexture::Bind
- CTexture::GetWidth
- CTexture::GetHeight
- CTexture::HasAlpha
- CTexture::GetUploadedSize
- GuiRenderer::Draw
- CMiniMap::CMiniMap
- CMiniMap::DrawViewRect
- CMiniMap::DrawTexture
- CMiniMap::Draw
- CMiniMap::CreateTextures
- CMiniMap::RebuildTerrainTexture
- CInput::Draw
- CGUI::DrawText
- CGUI::Draw
- OGL functions
- GLCursor::Create
- ogl_tex (OpenGL helper routines)
- ActorViewerImpl::Render
- AtlasViewGame::Render
- AtlasViewGame::DrawCinemaPathTool
- AtlasViewGame::DrawOverlays
- ScenarioEditor::ScenarioEditor

Vulkan Shader:

<https://www.khronos.org/spir/>

supported: SPIR-V bytecode

- SPIR-V Compiler: glslangValidator
- supported Shader Languages:
 - GLSL,
 - HLSL,
 - ARB (Extension) <https://www.khronos.org/opengles/sdk/tools/Reference-Compiler/>

Example:

```
#include <iostream>
using namespace std;

// use GL or Vulkan
#define _USEGL 1

// Renderer Interface
class IRenderer
{
public:
    //bool init();
    virtual void draw() = 0;
};

#ifdef _USEGL
// OpenGL Renderer
class GlRenderer : public IRenderer
{
public:
    void draw()
    {
        cout << "GlRenderer draw method called!" << endl;
    }
};

#elif _USEVULKAN

// Vulkan Renderer
class VulkanRenderer : public IRenderer
{
public:
    void draw()
    {
        cout << "VulkanRenderer draw method called!" << endl;
    }
};

#endif

int main()
{
#ifdef _USEGL
    IRenderer *rend = new GlRenderer();
    rend->draw();
#elif _USEVULKAN
    IRenderer *rend = new VulkanRenderer();
    rend->draw();
#endif
    return 0;
}
```