

er			
	Timestep		
	- m_Time : float		
	+ GetSeconds() : float		
	+ GetMilliseconds(): float		

VertexArray

## VertexArray

- + LoadModelIntoVAO(string filename, sModelDrawInfo, int shaderProgramID) : bool
- + FindDrawInfoByModelName(string filename, sModelDrawInfo) : bool
- + LoadPlyFiles(string filename, sModelDrawInfo) : bool
- + Create(): shared\_ptr<VertexArray>

sModelDrawInfo

meshName : string

VAO\_ID : int

VertexBufferID : int

VertexBuffer\_Start\_Index : int

numberOfVertices : int

IndexBufferID : int

IndexBuffer\_Start\_Index : int

number Of Indices: int

numberOfTriangles : int

pVertices : sVertex\_RGBA\_XYZ\_N\_UV\_T\_BiN\_Bones

pIndices: int

maxValues : glm::vec3

minValues : glm::vec3

sVertex\_RGBA\_XYZ\_N\_UV\_T\_BiN\_Bones

Colour - RGBA : float

Position - XYZW : float

Normal - nX nY nZ nW : float

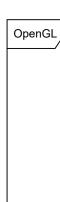
Texture UV0 UV1 :float

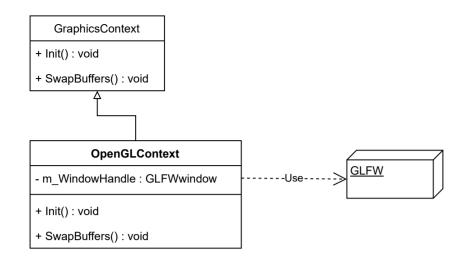
Tangent tX tY tZ tW : float

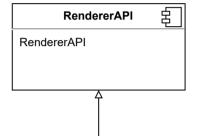
Binormal bX bY bZ bW : float

BoneID[4] : float

BoneWeight[4] : float





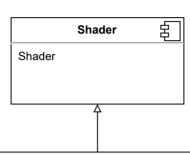


## **OpenGLRendererAPI**

- + SetClearColor(glm::vec4 color) : void
- + Clear(): void
- + DrawIndexed(sModelDrawInfo): void
- + SetViewport(int x,y,z, width, height) : void

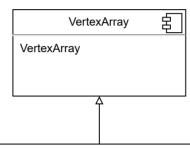
<------∳

- + SetCullFace(): void
- + EnableDepth() : void
- + SetDepthMask(bool) : void
- + SetPolygonMode(bool) : void



## OpenGLShader

- m\_basepath : string
- m\_RendererID : uint32
- m\_LastError: string
- + Bind(): void
- + UnBind(): void
- + UploadUniformInt(string uniformName, int value) : void
- + UploadUniformFloat(string uniformName, float value) : void
- + UploadUniformFloat2(string uniformName, glm::vec2 value) : void
- + UploadUniformFloat3(string uniformName, glm::vec3 value) : void
- + UploadUniformFloat4(string uniformName, glm::vec4 value) : void
- + UploadUniformMat3(string uniformName, glm::mat3 value) : void
- + UploadUniformMat4(string uniformName, glm::mat4 value) : void
- + GetID(): uint32
- Init(string vertexShaderFile, fragmentShaderFile) : bool
- m\_loadSourceFromFile(string fileName, vector<string> srcVec) : bool
- m\_compileShaderFromSource(vector<string> srcVec, GLuint id, string error) : bool
- m\_wasThereACompileError(GLuint shaderID, string error) : bool
- m\_wasThereALinkerError(GLuint progID, string error) : bool



## OpenGLVertexArray

- m\_Map\_Model\_to\_VAOID : map<string, sModelDrawInfo>
- + LoadModelIntoVAO(string filename, sModelDrawInfo, int shaderProgramID) : bool
- + FindDrawInfoByModelName(string filename, sModelDrawInfo) : bool
- + LoadPlyFiles(string filename, sModelDrawInfo) : bool

Shader	
	Shader
	+ Bind()
	+ UnBind()
	+ Create(string vertexShaderFile, FragmentShaderFile)

