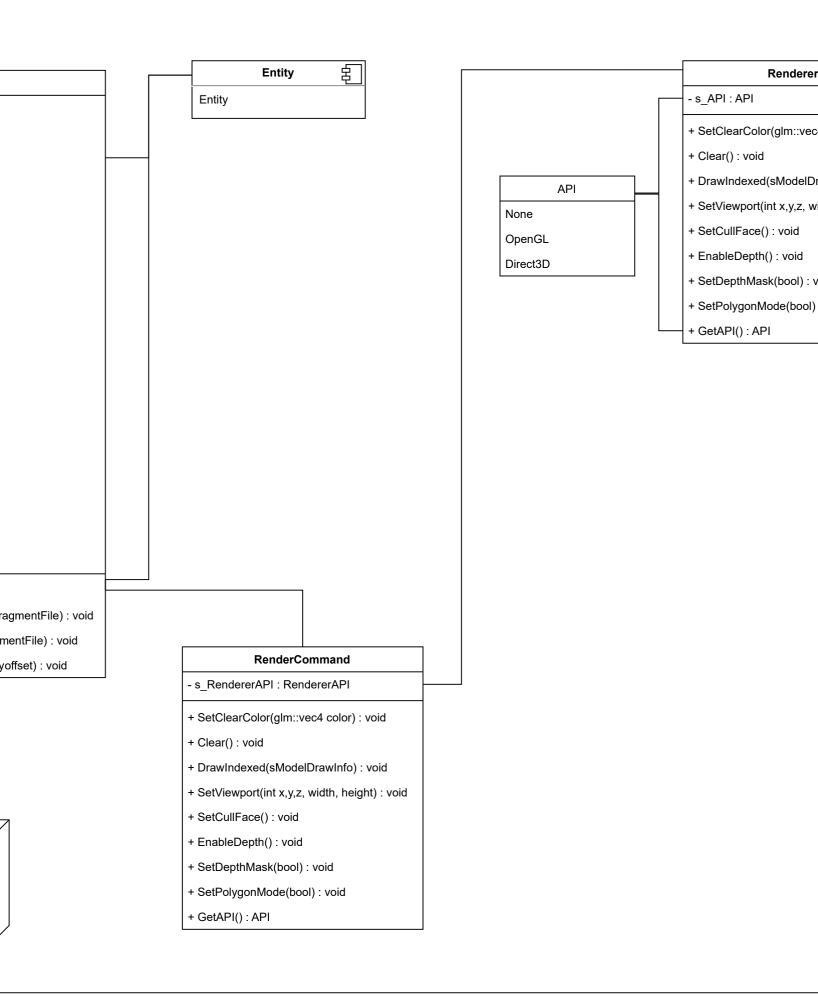


# ButtonReleasedEvent (): string

Renderer Renderer 起 Shader - m\_shader Shader - m\_VertexArray - EDITOR\_CAMERA : Entity\* - mouseHoldDown : bool - mouseClicked : bool 皂 VertexArray - cameraFront : glm::vec3 VertexArray - cameraUp : glm::vec3 - WorldUp : glm::vec3 - cameraRight : glm::vec3 - SCR\_WIDTH : int - SCR\_HEIGHT : int - lastX : float - lastY : float - firstMouse : bool - YAW: float - PITCH: float - SENSITIVITY: float - xpos : float - ypos : float + Process(vector<Entity\*>, float dt) : void + init(vector<Entity\*>, stirng VertexFile string f - CompileShader(string VertexFile, string Frag - ProcessMouseMovement( float xoffset, float Timer Use Timestep <u>OpenGL</u>



# API

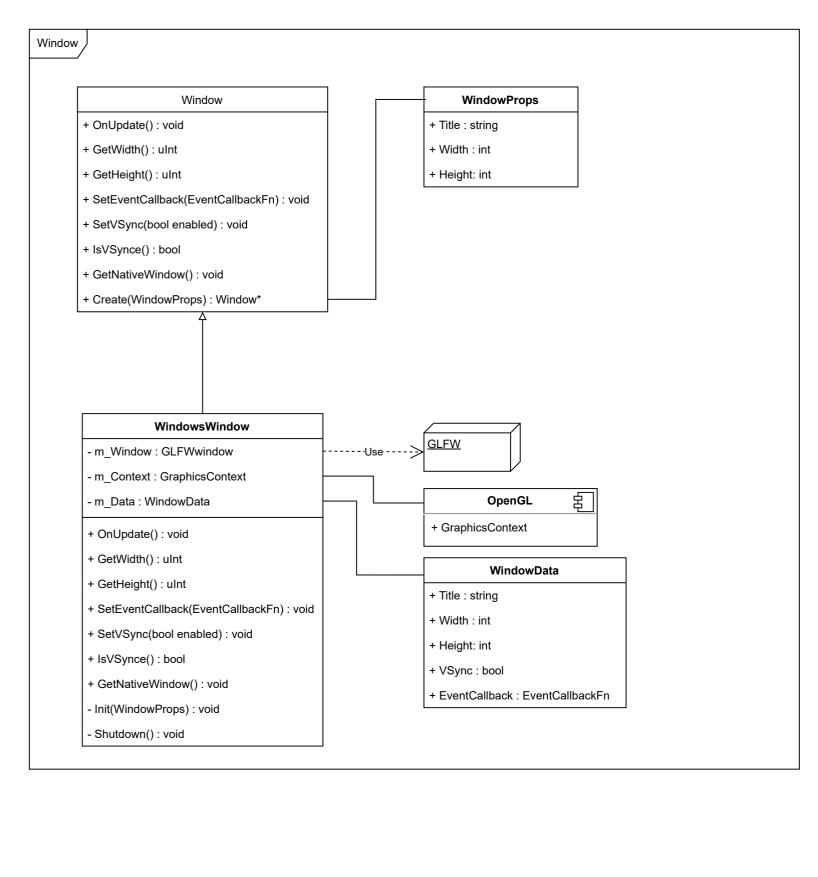
4 color) : void

awInfo) : void

dth, height): void

oid

: void



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

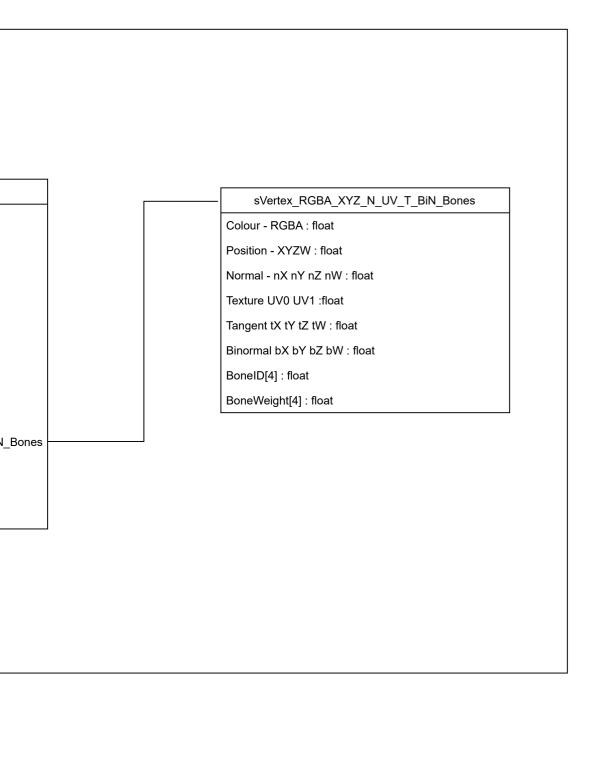
IndexBuffer\_Start\_Index : int

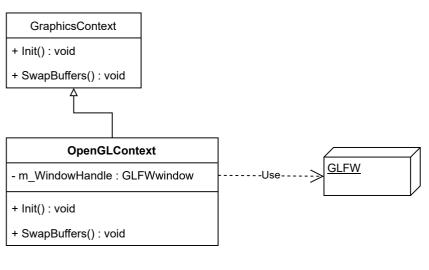
numberOfIndices : int numberOfTriangles : int

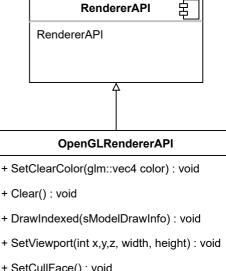
pVertices : sVertex\_RGBA\_XYZ\_N\_UV\_T\_BiN

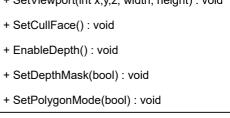
pIndices : int

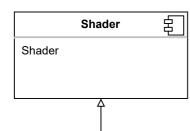
maxValues : glm::vec3 minValues : glm::vec3











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

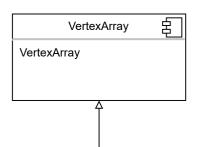
- m\_loadSourceFromFile(string fileName, vector<string> srcVec) : bool

- m compileShaderFromSource(vector<string> srcVec, GLuint id, string error) : bool

- m\_wasThereACompileError(GLuint shaderID, string error) : bool

- m\_wasThereALinkerError(GLuint progID, string error) : bool

glad Use



## OpenGLVertexArray

- m\_Map\_Model\_to\_VAOID : map<string, sModelDrawInfo>
- $+ Load Model Into VAO (string\ filename,\ s Model DrawInfo,\ int\ shader Program ID): bool$
- + FindDrawInfoByModelName(string filename, sModelDrawInfo) : bool
- + LoadPlyFiles(string filename, sModelDrawInfo) : bool

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

