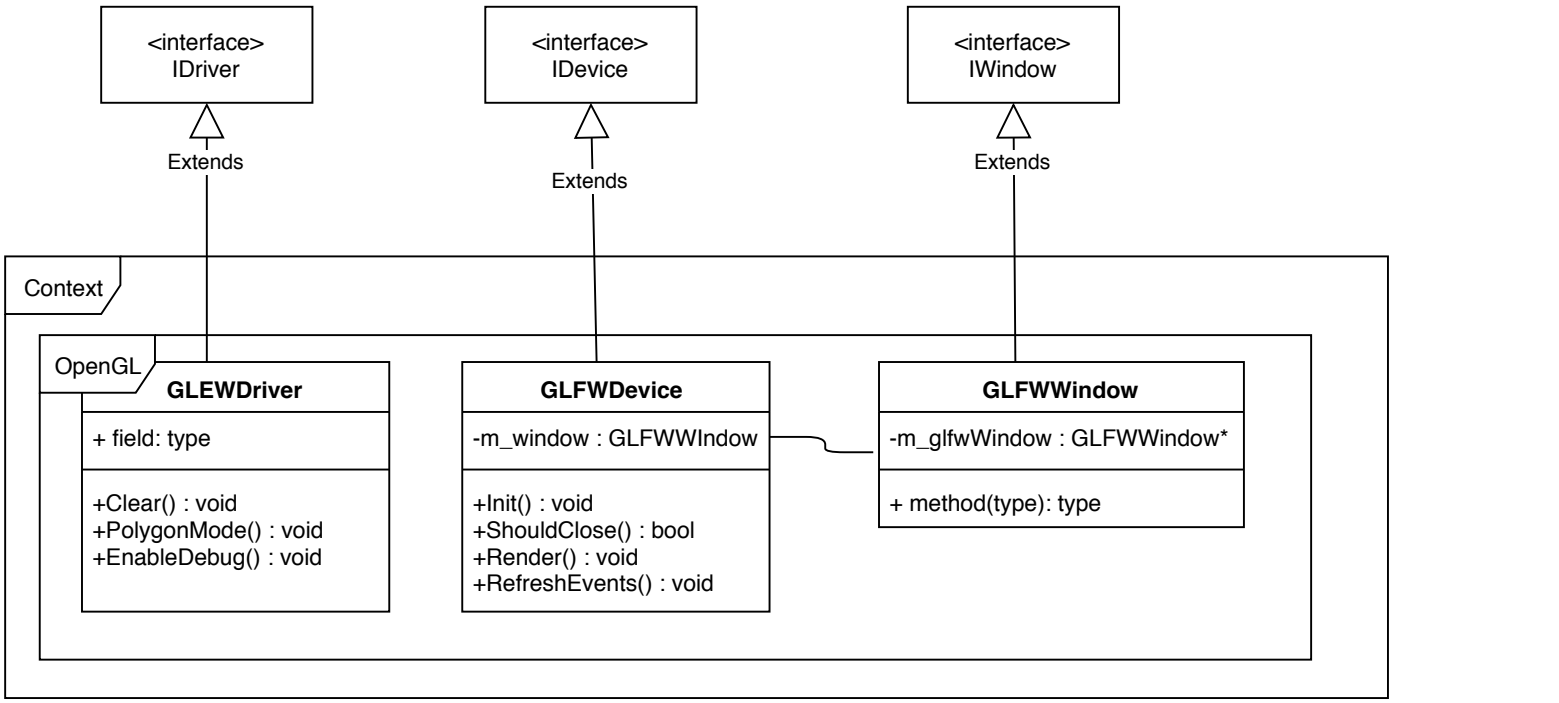
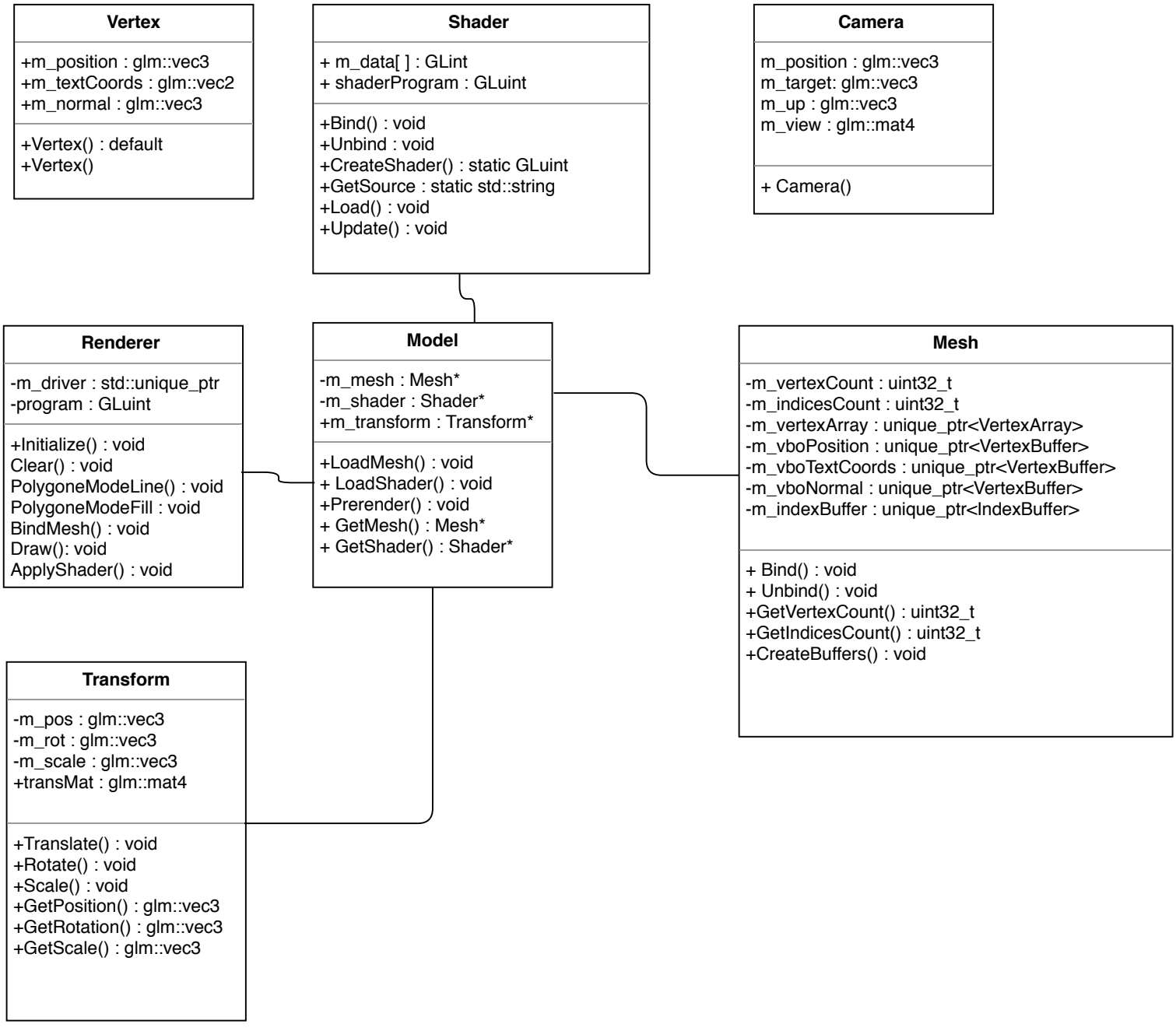
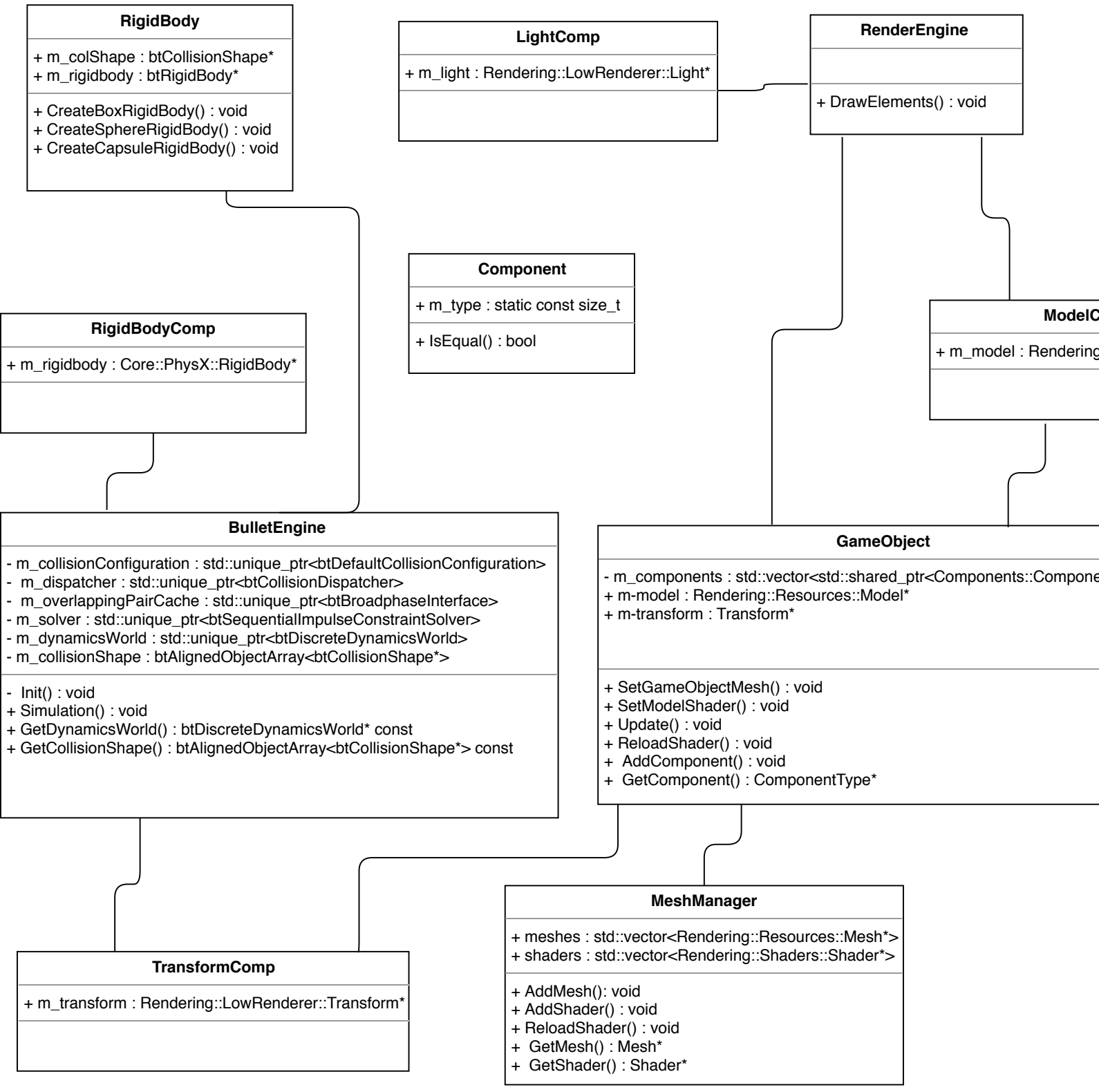
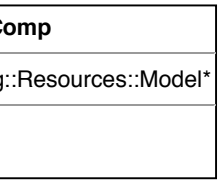
RENDERING


<interface>
IDriver
<interface>
IDevice
<interface>
IWindow
Extends
Extends
Extends
Context
OpenGL
GLEWDriver
+ field: type
+Clear() : void
+PolygonMode() : void
+EnableDebug() : void
GLFWDevice
-m_window : GLFWWIndow
+Init() : void
+ShouldClose() : bool
+Render() : void
+RefreshEvents() : void
GLFWWindow
-m_glfwWindow : GLFWWindow*
+ method(type): type


Vertex
+m_position : glm::vec3
+m_textCoords : glm::vec2
+m_normal : glm::vec3
+Vertex() : default
+Vertex()
Shader
+ m_data[ ] : GLint
+ shaderProgram : GLuint
+Bind() : void
+Unbind : void
+CreateShader() : static GLuint
+GetSource : static std::string
+Load() : void
+Update() : void
Camera
m_position : glm::vec3
m_target: glm::vec3
m_up : glm::vec3
m_view : glm::mat4
+ Camera()
Renderer
-m_driver : std::unique_ptr
-program : GLuint
+Initialize() : void
Clear() : void
PolygoneModeLine() : void
PolygoneModeFill : void
BindMesh() : void
Draw(): void
ApplyShader() : void
Model
-m_mesh : Mesh*
-m_shader : Shader*
+m_transform : Transform*
+LoadMesh() : void
+ LoadShader() : void
+Prerender() : void
+ GetMesh() : Mesh*
+ GetShader() : Shader*
Mesh
-m_vertexCount : uint32_t
-m_indicesCount : uint32_t
-m_vertexArray : unique_ptr<VertexArray>
-m_vboPosition : unique_ptr<VertexBuffer>
-m_vboTextCoords : unique_ptr<VertexBuffer>
-m_vboNormal : unique_ptr<VertexBuffer>
-m_indexBuffer : unique_ptr<IndexBuffer>
+ Bind() : void
+ Unbind() : void
+GetVertexCount() : uint32_t
+GetIndicesCount() : uint32_t
+CreateBuffers() : void
Transform
-m_pos : glm::vec3
-m_rot : glm::vec3
-m_scale : glm::vec3
+transMat : glm::mat4
+Translate() : void
+Rotate() : void
+Scale() : void
+GetPosition() : glm::vec3
+GetRotation() : glm::vec3
+GetScale() : glm::vec3

CORE


RigidBody
+ m_colShape : btCollisionShape*
+ m_rigidbody : btRigidBody*
+ CreateBoxRigidBody() : void
+ CreateSphereRigidBody() : void
+ CreateCapsuleRigidBody() : void
LightComp
+ m_light : Rendering::LowRenderer::Light*
RenderEngine
+ DrawElements() : void
Component
+ m_type : static const size_t
+ IsEqual() : bool
RigidBodyComp
+ m_rigidbody : Core::PhysX::RigidBody*
ModelC
+ m_model : Rendering
BulletEngine
- m_collisionConfiguration : std::unique_ptr<btDefaultCollisionConfiguration>
- m_dispatcher : std::unique_ptr<btCollisionDispatcher>
- m_overlappingPairCache : std::unique_ptr<btBroadphaseInterface>
- m_solver : std::unique_ptr<btSequentialImpulseConstraintSolver>
- m_dynamicsWorld : std::unique_ptr<btDiscreteDynamicsWorld>
- m_collisionShape : btAlignedObjectArray<btCollisionShape*>
- Init() : void
+ Simulation() : void
+ GetDynamicsWorld() : btDiscreteDynamicsWorld* const
+ GetCollisionShape() : btAlignedObjectArray<btCollisionShape*> const
GameObject
- m_components : std::vector<std::shared_ptr<Components::Compone
+ m-model : Rendering::Resources::Model*
+ m-transform : Transform*
+ SetGameObjectMesh() : void
+ SetModelShader() : void
+ Update() : void
+ ReloadShader() : void
+ AddComponent() : void
+ GetComponent() : ComponentType*
MeshManager
+ meshes : std::vector<Rendering::Resources::Mesh*>
+ shaders : std::vector<Rendering::Shaders::Shader*>
+ AddMesh(): void
+ AddShader() : void
+ ReloadShader() : void
+ GetMesh() : Mesh*
+ GetShader() : Shader*
TransformComp
+ m_transform : Rendering::LowRenderer::Transform*


Comp
g::Resources::Model*

ent>>