
Engine::Components
Component
Attributes
- m_GameObject : Engine::GameObject*
- m_CachedRB : Rigidbody*
- m_CachedTransform : Transform*
Operations
+ GetRigidbody() : Rigidbody*
+ GetTransform() : Transform*
+ GetGameObject() const : GameObject*
# Added() : virtual void
# Removed() : virtual void
# Draw() : virtual void
# DrawGizmos() : virtual void
# Update(deltaTime : float) : virtual void
Transform
Attributes
+ Scale : glm::vec3
+ Position : glm::vec3
+ Rotation : glm::vec3
- m_Parent : Transform*
- m_Children : std::vector<Transform*>
Operations
+ AddChild(child : Transform*) : void
+ SetParent(parent : Transform*) : void
+ GetGlobalScale() : glm::vec3
+ GetGlobalPosition() : glm::vec3
+ GetGlobalRotation() : glm::vec3
+ GetGlobalRotationMatrix() : glm::mat4
+ GetParent() : Transform*
+ GetChildren() : std::vector<Transform*>
+ FillShader(shader : Graphics::Shader*) : void
OrbitCameraController
Attributes
+ MoveSpeed : float
+ RotateSpeed : float
+ SprintSpeed : float
- m_Camera : Camera*
MeshRenderer
Attributes
+ Meshes : std::vector<MeshInfo>
Light
Attributes
+ Colour : glm::vec3
+ Radius : float
PhysicsComponent
Operations
FixedUpdate(timestep : float) : virtual void
ApplyForces(timestep : float) : virtual void
ApplyWorldForces(timestep : float) : virtual void
SolveConstraints(timestep : float) : virtual void
Particle
Attributes
+ UseGravity : bool
+ IsTrigger : bool
- m_Radius : float
- m_IsStatic : bool
- m_Collider : SphereCollider*
- m_Force : glm::vec3
- m_Velocity : glm::vec3
- m_Mass : float
- m_Restitution : float
- m_Friction : float
- m_PreviousPosition : glm::vec3
Operations
+ Particle()
+ GetSystem() : Physics::PhysicsSystem&
+ ApplyForce(force : glm::vec3, mode : ForceMode) : void
+ GetMass() : float
+ SetMass(value : float) : void
+ GetFriction() : float
+ SetFriction(value : float) : void
+ IsStatic() : bool
+ SetStatic(value : bool) : void
+ GetVelocity() : glm::vec3
+ GetRestitution() : float
+ SetRestitution(value : float) : void
+ SetCollisionRadius(value : float) : void
+ InverseMass() : float
Camera
Attributes
+ FieldOfView : float
+ ClipNear : float
+ ClipFar : float
+ Orthographic : bool
+ OrthoSize : float
+ RenderTarget : Graphics::RenderTexture*
- s_MainCamera : static Camera*
Operations
+ Camera()
+ GetViewMatrix() : glm::mat4
+ GetProjectionMatrix() : glm::mat4
+ FillShader(shader : Graphics::Shader*) : void
+ GetUpDirection() : glm::vec3
+ GetRightDirection() : glm::vec3
+ GetForwardDirection() : glm::vec3
+ SetMainCamera() : void
+ GetMainCamera() : static Camera*
<<Abstract>>
Collider
Attributes
+ IsTrigger : bool
- m_TriggerExitEvent : std::function<void(Collider*)>
- m_TriggerEnterEvent : std::function<void(Collider*)>
- m_CollisionEvent : std::function<void(Collider*, Rigidbody*)>
- m_CurrentTriggerEntries : std::vector<Collider*>
- m_PreviousTriggerEntries : std::vector<Collider*>
- m_DefaultInverseTensor : glm::mat4
Operations
+ GetBounds() = 0 : virtual Physics::OBB&
+ LineTest(line : Physics::Line&) = 0 : virtual bool
+ IsPointInside(point : glm::vec3) const = 0 : virtual bool
+ Raycast(ray : Physics::Ray&, outResult : Physics::RaycastHit*) = 0 : virtual bool
+ CheckCollision(other : Collider*) = 0 : virtual bool
+ CheckCollision(other : BoxCollider*) = 0 : virtual bool
+ CheckCollision(other : PlaneCollider*) = 0 : virtual bool
+ CheckCollision(other : SphereCollider*) = 0 : virtual bool
+ InverseTensor() : virtual glm::mat4&
+ SetTriggerExitEvent(callback : std::function<void(Collider*)>) : void
+ SetTriggerEnterEvent(callback : std::function<void(Collider*)>) : void
+ SetCollisionEvent(callback : std::function<void(Collider*, Rigidbody*)>) : void
- ProcessTriggerEntries() : void
BoxCollider
Attributes
+ Offset : glm::vec3
- m_Bounds : Physics::OBB
- m_InverseTensor : glm::mat4
- m_PreviousScale : glm::vec3
- m_Extents : glm::vec3
Operations
+ GetExtents() : glm::vec3&
+ SetExtents(value : glm::vec3) : void
+ GetOBB() : Physics::OBB&
- CalculateInverseTensor() : void
SphereCollider
Attributes
+ Offset : glm::vec3
- m_Radius : float
- m_InverseTensor : glm::mat4
- m_Bounds : Physics::OBB
- m_Sphere : Physics::Sphere
Operations
+ GetRadius() : float
+ SetRadius(value : float) : void
+ GetSphere() : Physics::Sphere&
- CalculateInverseTensor() : void
PlaneCollider
Attributes
- m_Bounds : Physics::OBB
- m_Plane : Physics::Plane
Operations
+ GetDistance() : float&
+ SetDistance(value : float) : void
+ GetNormal() : glm::vec3&
+ SetNormal(value : glm::vec3) : void
+ GetPlane() : Physics::Plane&
Spring
Attributes
- m_Point1 : Particle
- m_Point2 : Particle
- m_Stiffness : float
- m_Dampening : float
- m_RestingLength : float
Operations
+ GetPoint1() : Particle*
+ GetPoint2() : Particle*
+ SetBodies(a : Particle*, b : Particle*) : void
+ SetConstants(stiffness : float, dampening : float) : void
+ SetRestingLength(length : float) : void
+ GetRestingLength() : float
DistanceJoint
Attributes
m_Body1 : Particle*
m_Body2 : Particle*
m_Length : float
Operations
+ Initialise(a : Particle*, b : Particle*, length : float) : void
+ GetLength() : float
+ SetLength(length : float) : void
Rigidbody
Attributes
+ CanSleep : bool
+ UseGravity : bool
+ IsTrigger : bool
- m_IsStatic : bool
- m_Force : glm::vec3
- m_Torque : glm::vec3
- m_Velocity : glm::vec3
- m_Mass : float
- m_AngularVelocity : glm::vec3
- m_Sleeping : bool
- m_CoR : float
- m_Friction : float
Operations
+ Rigidbody()
+ GetSystem() : Physics::PhysicsSystem&
+ ApplyForce(force : glm::vec3, mode : ForceMode) : void
+ ApplyForce(force : glm::vec3, position : glm::vec3, mode : ForceMode) : void
+ AddRotationalImpulse(point : glm::vec3, impulse : glm::vec3) : void
+ GetMass() : float
+ SetMass(value : float) : void
+ GetFriction() : float
+ SetFriction(value : float) : void
+ IsStatic() : bool
+ SetStatic(value : bool) : void
+ GetVelocity() : glm::vec3
+ GetRestitution() : float
+ SetRestitution(value : float) : void
+ InverseMass() : float
+ KineticEnergy() : float
+ PotentialEnergy() : float
- InverseTensor() : glm::mat4&
- CheckSleeping() : void
- ApplyImpulse(other : Collider*, manifold : Physics::CollisionManifold, contactIndex : int) : void
- ApplyImpulse(other : Rigidbody*, manifold : Physics::CollisionManifold, contactIndex : int) : void
Cloth
Attributes
+ Material : Graphics::Material
# m_Mesh : Graphics::Mesh*
# m_ClothSize : unsigned int
# m_Vertices : std::vector<Particle*>
# m_BendSprings : std::vector<Spring*>
# m_ShearSprings : std::vector<Spring*>
# m_StructuralSprings : std::vector<Spring*>
Operations
+ Clear() : void
+ Initialise(size : unsigned int, spacing : float) : void
+ SetBendSprings(stiffness : float, dampening : float) : void
+ SetShearSprings(stiffness : float, dampening : float) : void
+ SetStructuralSprings(stiffness : float, dampening : float) : void
+ SetParticleMass(mass : float) : void
Engine::Physics
Line
Attributes
+ Start : glm::vec3
+ End : glm::vec3
Operations
+ Line(start : glm::vec3, end : glm::vec3)
+ Length() : float
+ LengthSqr() : float
+ IsPointOn(point : glm::vec3&) : bool
+ GetClosestPoint(point : glm::vec3&) : glm::vec3
RaycastHit
Attributes
+ Point : glm::vec3
+ Normal : glm::vec3
+ Distance : float
+ Hit : bool
Interval
Attributes
+ Min : float
+ Max : float
Ray
Attributes
+ Origin : glm::vec3
+ Direction : glm::vec3
Operations
+ IsPointOn(point : glm::vec3&) : bool
+ GetClosestPoint(point : glm::vec3&) : glm::vec3
+ FromLine(Line& line) : static Ray
+ FromPoints(start : glm::vec3, end : glm::vec3) : static Ray
PhysicsSystem
Attributes
m_ImpulseIteration : int
m_Substeps : int
- m_Thread : std::thread
- m_VariableMutex : std::mutex
- m_CollidersMutex : std::mutex
- m_PhysicsStateMutex : std::mutex
- m_ThreadState : std::atomic_int
- m_PauseConditional : std::condition_variable
- m_Broadphase : Broadphase*
- m_PhysicsState : PhysicsPlayState
- m_Gravity : glm::vec3
- m_LastTimestep : std::chrono::milliseconds
- m_FixedTimestep : std::chrono::milliseconds
- m_Colliders : std::vector<Components::Collider*>
- m_Collisions : std::vector<CollisionFrame>
- m_Components : std::vector<Components::PhysicsComponent*>
- m_LinearProjectionPercent : float
- m_PenetrationSlack : float
- m_App : Engine::Application*
Operations
+ PhysicsSystem(app : Engine::Application*, fixedTimestep : std::chrono::milliseconds)
+ ~PhysicsSystem()
+ Start() : void
+ Stop() : void
+ DrawGizmos() : void
+ Resume() : void
+ Pause() : void
+ TogglePause() : void
+ Timestep() : std::chrono::milliseconds
+ LastTimestep() : std::chrono::milliseconds
+ SetTimestep(timestep : std::chrono::milliseconds) : void
+ SetGravity(gravity : glm::vec3) : void
+ GetGravity() : glm::vec3
+ GetState() : PhysicsPlayState
+ SetBroadphase<T>(args class...) : T*
+ LineTest(line : Line, ignoreCollider : Components::Collider*) : bool
+ Raycast(ray : Ray, outResult : RaycastHit*) : Components::Collider*
+ Raycast(ray : Ray, ignoreCollider : Components::Collider*, outResult : RaycastHit*) : Components::Collider*
+ Query(bounds : Sphere&) : std::vector<Components::Collider*>
+ Query(bounds : AABB&) : std::vector<Components::Collider*>
- PhysicsLoop() : void
- AddCollider(collider : Components::Collider*) : void
- RemoveCollider(collider : Components::Collider*) : void
- AddPhysicsComponent(component : Components::Collider*) : void
- RemovePhysicsComponent(component : Components::Collider*) : void
- NarrowPhase() : void
- ApplyImpulse() : void
- PositionalCorrect() : void
AABB
Attributes
+ Position : glm::vec3
+ Extents : glm::vec3
Operations
+ Min() : glm::vec3
+ Max() : glm::vec3
+ FromMinMax(min : glm::vec3, max : glm::vec3) : static AABB
+ Intersects(other : AABB&) : bool
+ IsPointInside(point : glm::vec3&) : bool
+ GetInterval(axis : const glm::vec3&) : Interval
+ GetClosestPoint(point : glm::vec3&) : glm::vec3
+ Raycast(ray : Ray&, outResult : RaycastHit*) : bool
+ LineTest(line : Line&) : bool
OBB
Attributes
+ Position : glm::vec3
+ Extents : glm::vec3
+ Orientation : glm::mat3
- m_Lines : std::vector<Line>
- m_Planes : std::vector<Plane>
- m_Vertices : std::vector<glm::vec3>
- m_VertexPosition : glm::vec3
- m_VertexExtents : glm::vec3
Operations
+ OBB(position : glm::vec3, extents : glm::vec3, orientation : glm::mat3)
+ IsPointInside(point : glm::vec3&) const : bool
+ GetInterval(axis : const glm::vec3&) : Interval
+ GetClosestPoint(point : glm::vec3&) : glm::vec3
+ OverlapOnAxis(other : OBB&, axis : glm::vec3&) : bool
+ Raycast(ray : Ray&, outResult : RaycastHit*) : bool
+ LineTest(line : Line&) : bool
+ GetEdges() : std::vector<Line>&
+ GetPlanes() : std::vector<Plane>&
+ GetVertices() : std::vector<glm::vec3>&
+ ClipEdges(edges : std::vector<Line>&) : std::vector<glm::vec3>
+ ClipToPlane(plane : Plane&, line : Line&, result : glm::vec3*) : bool
+ PenetrationDepth(other : OBB&, axis : glm::vec3&, outShouldFlip : bool*) : float
Sphere
Attributes
+ Position : glm::vec3
+ Radius : float
Operations
+ IsPointInside(point : glm::vec3&) const : bool
+ GetClosestPoint(point : glm::vec3&) : glm::vec3
+ Raycast(ray : Ray&, outResult : RaycastHit*) : bool
+ LineTest(line : Line&) : bool
Plane
Attributes
+ Distance : float
+ Normal : glm::vec3
Operations
+ Plane(normal : glm::vec3, distance : float)
+ Plane(triangle : Triangle&)
+ Raycast(ray : Ray&, outResult : RaycastHit*) : bool
+ LineTest(line : Line&) : bool
+ CheckPoint(point : glm::vec3&) const : PlaneIntersection
+ GetClosestPoint(point : glm::vec3&) : glm::vec3
+ PlaneEquation(point : glm::vec3) : float
CollisionManifold
Attributes
+ IsColliding : bool
+ Normal : glm::vec3
+ PenetrationDepth : float
+ Contacts : std::vector<glm::vec3>
CollisionFrame
Attributes
+ A : Components::Collider*
+ ARigidbody : Components::Rigidbody*
+ B : Components::Collider*
+ BRigidbody : Components::Rigidbody*
+ Result : CollisionManifold
Triangle
Attributes
+ Points : glm::vec3[3]
Operations
+ Triangle()
+ Triangle(a : glm::vec3, b : glm::vec3, c : glm::vec3)
+ IsPointInside(point : glm::vec3&) const : bool
+ GetClosestPoint(point : glm::vec3&) : glm::vec3
+ Raycast(ray : Ray&, outResult : RaycastHit*) : bool
+ LineTest(line : Line&) : bool
+ Intersects(obb : OBB&) : bool
+ Intersects(abbb : AABB&) : bool
+ Intersects(plane : Plane&) : bool
+ Intersects(sphere : Sphere&) : bool
+ Intersects(triangle : Triangle&) : bool
+ GetInterval(axis : glm::vec3&) : Interval
+ OverlapOnAxis(obb : OBB&, axis : glm::vec3&) bool
+ OverlapOnAxis(abb : AABB&, axis : glm::vec3&) bool
+ OverlapOnAxis(tri : Triangle&, axis : glm::vec3&) bool
+ Barycentric(point : glm::vec3&) : glm::vec3
- SatCrossEdge(a : glm::vec3&, b : glm::vec3&, c : glm::vec3&, d : glm::vec3&) : glm::vec3
<<Abstract>>
Broadphase
Operations
+ Insert(collider : Components::Collider*) = 0 : virtual void
+ Remove(collider : Components::Collider*) = 0 : virtual void
+ Update() : virtual void
+ DrawGizmos() : virtual void
+ GetPotentialCollisions() = 0 : virtual std::vector<CollisionFrame>&
+ GetCollider(point : glm::vec3&) : virtual Components::Collider*
+ LineTest(line : Line&, ignoreCollider : Components::Collider*) = 0 : virtual bool
+ Raycast(ray : Ray, outResult : RaycastHit*) : Components::Collider*
+ Raycast(ray : Ray, ignoreCollider : Components::Collider*, outResult : RaycastHit*) = 0 : virtual Components::Collider*
+ Query(bounds : AABB&) const = 0 : virtual std::vector<Components::Collider*>
+ Query(bounds : Sphere&) const = 0 : virtual std::vector<Components::Collider*>
BasicBroadphase
Attributes
+ m_Collisions : std::vector<CollisionFrame>
+ m_Colliders : std::vector<Components::Collider*>
<<Enum>>
PlaneIntersection
None
Behind
InFront
Intersection
<<Enum>>
PhysicsPlayState : int
Stopped : 0
Playing
Paused
Engine::Graphics
Texture
Attributes
- m_ID : unsigned int
- m_Path : std::string
Operations
+ Texture()
+ Texture(path : std::string, hdr : bool)
+ ~Texture()
+ GetID() : unsigned int
+ GetPath() : std::string
+ Bind(index : unsigned int) : void
- GenerateImage(hdr : bool) : void
FramebufferSpec
Attributes
+ Resolution : glm::ivec2
+ SwapchainTarget : bool
+ Samples : unsigned int
+ Attachments : std::vector<TextureFormat>
ShaderStageInfo
Attributes
+ VertexPath : std::string
+ FragmentPath : std::string
+ ComputePath : std::string
+ GeometryPath : std::string
ShaderUniform
Attributes
+ Location : int
+ Name : std::string
+ Type : unsigned int
Shader
Attributes
- m_IsDirty : bool
- m_Program : unsigned int
- m_ShaderStages : ShaderStageInfo
- m_Uniforms : std::vector<ShaderUniform>
- m_Watchers : std::unordered_map<std::string, FileWatcher*>
Operations
+ Shader()
+ Shader(stageInfo : ShaderStageInfo)
+ ~Shader()
+ GetStages() : ShaderStageInfo&
+ UpdateStages(stageInfo : ShaderStageInfo) : void
+ Bind() : void
+ Unbind() : void
+ GetProgram() : unsigned int
+ GetUniformCount() : unsigned int
+ Set(location : int, value : int) const : void
+ Set(location : int, value : bool) const : void
+ Set(location : int, value : float) const : void
+ Set(location : int, value : double) const : void
+ Set(location : int, value : glm::vec2) const : void
+ Set(location : int, value : glm::vec3) const : void
+ Set(location : int, value : glm::vec4) const : void
+ Set(location : int, value : glm::mat3) const : void
+ Set(location : int, value : glm::mat4) const : void
+ GetUniformInfo(location : int) : ShaderUniform&
+ GetUniformInfo(locationName : std::string) : ShaderUniform&
<<Enum>>
TextureFormat
None
RGBA8
RGBA16
RGBA16F
RedInteger
Cubemap
Depth24Stencil8
Depth = Depth24Stencil8
RenderTexture
Attributes
- m_ID : unsigned int
- m_Samples : unsigned int
- m_Resolution : glm::ivec2
- m_Format : TextureFormat
Operations
+ RenderTexture(resolution : glm::ivec2, format : TextureFormat, samples : unsigned int)
+ ~RenderTexture()
+ Bind(textureIndex : unsigned int) : void
+ Unbind(textureIndex : unsigned int) : void
+ GetID() : unsigned int
+ GetSamples() : unsigned int
+ SetSamples(samples : unsigned int) : void
+ GetFormat() : TextureFormat
+ SetFormat(format : TextureFormat) : void
+ GetResolution() : glm::ivec2
+ SetResolution(res : glm::ivec2) : void
- Recreate() : void
- CreateDepthTexture() : void
- CreateColourTexture() : void
RenderPipelinePass
Attributes
+ Shader : Shader*
+ Pass : RenderPass*
+ DrawCallback : std::function<void()>
+ ResizeWithScreen : bool
RenderPass
Attributes
- m_Framebuffer : Framebuffer*
Operations
+ RenderPass(specs : FramebufferSpec&)
+ ~RenderPass()
+ GetFramebuffer() : Framebuffer*
+ Begin() : void
+ End() : void
+ GetDepthAttachment() : RenderTexture*
+ GetColourAttachmentCount() : unsigned int
+ GetColourAttachment(index : unsigned int) : RenderTexture*
Material
Attributes
+ Albedo : glm::vec4
+ AlbedoMap : Texture*
+ AlphaClipping : bool
+ AlphaClipThreshold : float
+ TextureCoordinateScale : glm::vec2
+ TextureCoordinateOffset : glm::vec2
+ NormalMap : Texture*
+ Wireframe : bool
Operations
+ FillShader(shader : Shader*) : void
Framebuffer
Attributes
- m_ID : unsigned int
- m_Specs : FramebufferSpec
- m_DepthAttachment : RenderTexture*
- m_ColourAttachments : std::vector<RenderTexture*>
Operations
+ Framebuffer(specs : FramebufferSpec&)
+ ~Framebuffer()
+ Bind() : void
+ Unbind() : void
+ SetSamples(samples : unsigned int) : void
+ GetSamples() : unsigned int
+ SetResolution(res : glm::ivec2) : void
+ GetResolution() : glm::ivec2
+ CopyAttachmentTo(destination : RenderTexture*, colourAttachment : unsigned int) : void
+ BlitTo(other : Framebuffer*, bufferFlags : GLbitfield, filter : GLenum) : void
+ GetDepthAttachment() : RenderTexture*
+ GetColourAttachment(index : unsigned int) : RenderTexture*
+ GetAttachments() : std::vector<RenderTexture*>
+ GetColourAttachments() : std::vector<RenderTexture*>
+ AttachmentCount() : unsigned int
+ ColourAttachmentCount() : unsigned int
+ HasDepthAttachment() : bool
+ GetSpecs() : FramebufferSpec&
- Create() : void
- Destroy() : void
- Recreate() : void
RenderPipeline
Attributes
m_CurrentShader : Shader*
m_RenderPasses : std::vector<RenderPipelinePass>
Operations
+ Draw(camera : Engine::Components::Camera*) : void
+ GetOutputAttachment(index : unsigned int) : RenderTexture*
+ RemovePass(pass : RenderPass*) : virtual void
+ AddPass(passInfo : RenderPipelinePass&) : virtual void
+ OnResized(resolution : glm::ivec2) : virtual void
+ CurrentShader() : Shader*
ForwardRenderPipeline
Operations
+ ForwardRenderPipeline()
+ ~ForwardRenderPipeline()
- ForwardPass()
DeferredRenderPipeline
Attributes
- m_MeshPass : RenderPass*
- m_LightingPass : RenderPass*
- m_ForwardPass : RenderPass*
Operations
+ DeferredRenderPipeline()
+ ~DeferredRenderPipeline()
<<Enum>>
Mesh::DrawMode
Points
Lines
LineStrip
Triangles
TriangleStrip
TriangleFan
Quads
QuadStrip
Mesh::Vertex
Attributes
+ Position : glm::vec3
+ Normal : glm::vec3
+ TexCoords : glm::vec2
Mesh
Attributes
- m_VBO : unsigned int
- m_VAO : unsigned int
- m_EBO : unsigned int
Operations
+ Mesh()
+ Mesh(vertices : std::vector<Vertex>, indices : std::vector<unsigned int>)
+ ~Mesh()
+ Draw() : void
+ SetData(vertices : std::vector<Vertex>&, indices : std::vector<glm::vec3>) : void
+ GetVertices() : std::vector<Vertex>&
+ GetIndices() : std::vector<unsigned int>&
+ Quad() : static Mesh*
+ Cube() : static Mesh*
+ Line() : static Mesh*
+ Sphere() : static Mesh*
+ Grid(size : unsigned int) : static Mesh*
Gizmos
Attributes
- s_Material : static Material
+ Colour : static glm::vec4
Operations
+ Draw(mesh : Mesh*, position : glm::vec3, scale : glm::vec3, rotation : glm::vec3) : static void
+ DrawGrid(position : glm::vec3, gridSize : unsigned int, scale : glm::vec3, rotation : glm::vec3) : static void
+ DrawRay(ray : Ray) : static void
+ DrawLine(line : Line) : static void
+ DrawLine(start : glm::vec3, end : glm::vec3) : static void
+ DrawQuad(position : glm::vec3, scale : glm::vec2, rotation : glm::vec3) : static void
+ DrawCube(position : glm::vec3, scale : glm::vec3, rotation : glm::vec3) : static void
+ DrawSphere(position : glm::vec3, radius : float) : static void
+ DrawWireQuad(position : glm::vec3, scale : glm::vec2, rotation : glm::vec3) : static void
+ DrawWireCube(position : glm::vec3, scale : glm::vec3, rotation : glm::vec3) : static void
+ DrawWireSphere(position : glm::vec3, radius : float) : static void
DrawCall
Attributes
+ Mesh : Mesh*
+ Material : Material
+ Position : glm::vec3
+ Scale : glm::vec3
+ Rotation : glm::mat4
+ DeleteMeshAfterRender : bool
Renderer
Attributes
- s_Samples : static unsigned int
- s_App : static Application*
- s_Resolution : static glm::ivec2
- s_Wireframe : static bool
- s_VSync : static bool
- s_Pipeline : static RenderPipeline*
- s_DrawQueue : static std::vector<DrawCall>
- s_Time : static float
- s_FPS : static float
- s_DeltaTime : static float
Operations
+ Draw(clearQueue : bool) : static void
+ ClearDrawQueue() : static void
+ Submit(drawCall : DrawCall) : static void
+ Submit(mesh : Mesh*, material : Material&, transform : Components::Transform*) : st
+ Submit(mesh : Mesh*, material : Material&, position : glm::vec3, scale : glm::vec3, rotation : glm::vec3) : static void
+ Submit(mesh : Mesh*, material : Material&, position : glm::vec3, scale : glm::vec3, rotation : glm::mat4) : static void
+ SetVSync(vsync : bool) : static void
+ SetWireframe(wireframe : bool) : static void
+ SetPipeline<T>() : static T*
+ GetVSync() : static bool
+ GetFPS() : static float
+ GetTime() : static float
+ GetDeltaTime() : static float
+ GetSamples() : static unsigned int
+ GetWireframeMode() : static bool
+ GetApp() : static Application*
+ GetResolution() : static glm::ivec2
+ GetPipeline() : static RenderPipeline*
Engine
ApplicationArgs
Attributes
+ VSync : bool
+ Samples : unsigned int
+ Title : std::string
+ Resolution : glm::ivec2
+ FixedTimestep : std::chrono::milliseconds
+ Fullscreen : bool
Application
Attributes
+ AssetDir : const static std::string
- m_Scene : Scene
- m_Window : GLFWwindow*
- m_Args : ApplicationArgs
- m_WindowedResolution : glm::ivec2
- s_Instance : static Application*
Operations
+ Application(args : ApplicationArgs)
+ Run() : void
+ Exit() : void
+ CurrentScene() : Scene*
+ SetTitle(title : std::string) : void
+ GetFullscreen() : bool
+ ToggleFullscreen() : void
+ SetFullscreen(fullscreen : bool) : void
# OnDraw() : virtual void
# OnDrawGizmos() : virtual void
# OnUpdate() : virtual void
# OnStart() : virtual void
# OnShutdown() : virtual void
# OnResized(resolution : glm::ivec2) : virtual void
- SetupGizmos() : void
- CreateAppWindow() : void
- GLFW_WindowCloseCallback(window : GLFWwindow*) : static void
- GLFW_ErrorCallback(error : int, description : const char*) : static void
- GLFW_ScrollCallback(window : GLFWwindow*, xOffset : double, yOffset : double) : static void
- GLFW_MouseCallback(window : GLFWwindow*, button : int, action : int, mods : int) : static void
- GLFW_CursorPositionCallback(window : GLFWwindow*, xPos : double, yPos : double) : static void
- GLFW_FramebufferResizeCallback(window : GLFWwindow*, width : int, height : int) : static void
- GLFW_KeyCallback(window : GLFWwindow*, width : int, height : int) : static void
- GLFW_DebugOutput(source : GLenum, type : GLenum, id : unsigned int, severity : GLenum, length : GLsizei, msg : const char*, userParam : const void*) : static void
GameObject
Attributes
- m_Name : std::string
- m_Scene : Scene*
- m_Transform : Components::Transform*
- m_Components : std::unordered_map<std::type_index, Components::Component*>
Operations
+ GameObject(scene : Scene*, name : std::string)
+ GameObject(parent : GameObject*, name : std::string)
+ GameObject(parent : Components::Transform*, name : std::string)
+ ~GameObject()
+ Draw() : void
+ DrawGizmos() : void
+ Update(deltaTime : float) : void
+ GetScene() : Scene*
+ GetName() : std::string
+ SetName(name : std::string) : void
+ GetTransform() : Components::Transform*
+ AddComponent<T>() : T*
+ RemoveComponent<T>() : void
+ GetComponent<T>(checkInheritedClass : bool) : T*
+ GetComponents<T>(checkInheritedClass : bool) : std::vector<T*>
+ GetComponentInChildren(checkInheritedClass : bool) : T*
+ GetComponentsInChildren(checkInheritedClass : bool) : std::vector<T*>
+ HasComponent<T>(checkInheritedClass : bool) : bool
<<Enum>>
KeyState
Up
Down
Pressed
Released
Input
Attributes
- s_ScrollDelta : static float
- s_MousePos : static glm::vec2
- s_LastMousePos : static glm::vec2
- s_MainWindow : static GLFWwindow*
- s_KeyStates : static std::map<int, KeyState>
- s_MouseStates : static std::map<int, KeyState>
Operations
+ IsKeyUp(key : int) : static bool
+ IsKeyDown(key : int) : static bool
+ IsKeyPressed(key : int) : static bool
+ IsKeyReleased(key : int) : static bool
+ GetMousePosition() : static glm::vec2
+ GetMouseDelta() : static glm::vec2
+ GetScrollDelta() : static float
+ IsMouseUp(button : int) : static bool
+ IsMouseDown(button : int) : static bool
+ IsMousePressed(button : int) : static bool
+ IsMouseReleased(button : int) : static bool
+ ShowMouse(show : bool) : static void
Scene
Attributes
- m_Name : std::string
- m_Root : GameObject
- m_Physics : Physics::PhysicsSystem
Operations
+ Scene(app : Application*, name : std::string)
+ Root() : GameObject&
+ GetPhysics() : Physics::PhysicsSystem&
+ Draw() : void
+ DrawGizmos() : void
+ Update(deltaTime : float) : void
Timer
Attributes
+ DefaultInterval : static const std::chrono::milliseconds
- m_IsRunning : bool
- m_Thread : std::thread
- m_Callback : sdt::function<void()>
- m_Interval : std::chrono::milliseconds
- m_StartTime : std::chrono::time_point<chrono::high_resolution_clock>
- m_StopTime : std::chrono::time_point<chrono::high_resolution_clock>
Operations
+ Timer(intervalMS : unsigned int)
+ Timer(interval : std::chrono::milliseconds)
+ Timer(callback : std::function<void()>, interval : std::chrono::milliseconds)
+ ~Timer()
+ Start() : void
+ Stop() : void
+ ElapsedTime() : std::chrono::milliseconds
+ Restart() : void
+ IsRunning() : bool
<<Enum>>
FileWatchStatus
Created
Removed
Modified
FileWatcher
Attributes
- m_Path : std::string
- m_LoopThread : std::thread
- m_Running : bool
- m_Multithread : bool
- m_Interval : std::chrono::duration<int, std::milli>
- m_Callback : std::function<void(std::string, FileWatchStatus)>
- m_WatchedPaths : std::unordered_map<std::string, std::filesystem::file_time_type>
Operations
+ FileWatcher(path : std::string, intervalMs : int, multithread : bool)
+ FileWatcher(path : std::string, multithread : bool)
+ Start(callback : std::function<void(std::string, FileWatchStatus)>) : void
+ Start() : void
+ Stop() : void