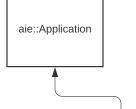
#### PhysicsScene

- +PhysicsScene()
- +virtual ~PhysicsScene()
- +addActor(PhysicsObject\*): void +removeActor(PhysicsObject\*): void
- +update(float) : void +draw() : void
- +setGravity(const glm::vec2) : void +getGravity() const : glm::vec2 +setTimeStep(const float) : void +getTimeStep() const : float
- +planeToPlane(PhysicsObject\* object1. PhysicsObject\* object2) : bool
  +planeToSphere(PhysicsObject\* object1. PhysicsObject\* object2) : bool
  +planeToBox(PhysicsObject\* object1. PhysicsObject\* object2) : bool
  +sphereToPlane(PhysicsObject\* object1. PhysicsObject\* object2) : bool
  +sphereToSphere(PhysicsObject\* object1. PhysicsObject\* object2) : bool
  +sphereToBox(PhysicsObject\* object1. PhysicsObject\* object2) : bool
  +boxToPlane(PhysicsObject\* object1. PhysicsObject\* object2) : bool
  +boxToSphere(PhysicsObject\* object1. PhysicsObject\* object2) : bool
  +boxToBox(PhysicsObject\* object1. PhysicsObject\* object2) : bool

-m\_gravity : glm::vec2 -m\_timeStep : float

-m actors : std::set<PhysicsObject\*>



# <<enum class>> ShapeType

PLANE = 0, SPHERE, BOX, LENGTH

## PhysicsGame

- +PhysicsGame()
- +virtual ~PhysicsGame()

virtual startup() : bool virtual shutdown() : void virtual update(float) : void virtual draw() : void

-m\_renderer : aie::Renderer2D\*

 $\hbox{-m\_font: aie::Font*} \\$ 

-m scene: PhysicsScene\*

## **Bootstrap Physics**

PhysicsObject

#PhysicsObject(ShapeType)

#~PhysicsObject()

+virtual fixedUpdate(glm:;vec2, float) = 0 : void

+ virtual draw() = 0 : void

+ virtual resetPosition() = 0 : void

+getShapeID(): ShapeType

-m shapeID: ShapeType

#### Plane

- +Plane(glm::vec2, float, glm::vec4)
- +~Plane()
- +virtual fixedUpdate(glm::vec2, float) : void
- +virtual draw(): void
- +resolveCollision(RigidBody\*): void

+getNormal() : glm::vec2 +getDistance() : float +getColor() : glm::vec4

-m\_normal : glm::vec2
-m\_distance : float
-m\_color : glm::vec4

## RigidBody

- +RigidBody(ShapeType, glm::vec2, glm::vec2, float, float)
- +~RigidBody()
- +virtual fixedUpdate(glm:;vec2, float) : void
- +applyForce(glm::vec2): void
- +applyForceToOther(RigidBody\*, glm::vec2): void
- +resolveCollision(RigidBody\*): void

+getPosition(): glm::vec2 +getVeolcity(): glm::vec2 +getOrientation(): float +getMass(): float

-m\_position : glm:;vec2 -m\_velocity : glm::vec2 -m\_orientation : float -m\_mass : float

Sphere

- +Sphere(glm::vec2, glm::vec2, float, float, glm::vec4)
- +~Sphere()
- +virtual draw(): void
- +getRadius() : float +getColor() : glm::vec4

-m\_radius : float -m\_color : glm:;vec4