Farouq Adepetu's Physics Engine

Generated by Doxygen 1.9.4

1 Class Index 1

1.1 Class List . . . 1

2 File Index 3

2.1 File List . . . 3

3 Class Documentation 5

3.1 FAPhysics::RigidBody Class Reference . . . 5

3.1.1 Constructor & Destructor Documentation . . . 5

3.1.1.1 RigidBody() . . . 6

3.1.2 Member Function Documentation . . . 6

3.1.2.1 AddForce() . . . 6

3.1.2.2 AddTorque() . . . 6

3.1.2.3 GetAngularMomentum() . . . 6

3.1.2.4 GetAngularVelocity() . . . 6

3.1.2.5 GetBodyInertiaTensor() . . . 6

3.1.2.6 GetCenterOfMass() . . . 7

3.1.2.7 GetInverseBodyInertiaTensor() . . . 7

3.1.2.8 GetInverseMass() . . . 7

3.1.2.9 GetLinearMomentum() . . . 7

3.1.2.10 GetLinearVelocity() . . . 7

3.1.2.11 GetMass() . . . 7

3.1.2.12 GetNetForce() . . . 7

3.1.2.13 GetNetTorque() . . . 8

3.1.2.14 GetOrientation() . . . 8

3.1.2.15 InitializeRigidBody() . . . 8

3.1.2.16 Integrate() . . . 8

3.1.2.17 ResetForce() . . . 8

3.1.2.18 ResetTorque() . . . 8

3.1.2.19 SetAngularMomentum() . . . 9

3.1.2.20 SetAngularVelocity() . . . 9

3.1.2.21 SetBodyInertiaTensor() . . . 9

3.1.2.22 SetCenterOfMass() . . . 9

3.1.2.23 SetLinearMomentum() . . . 9

3.1.2.24 SetLinearVelocity() . . . 9

3.1.2.25 SetMass() . . . 10

3.1.2.26 SetOrientation() . . . 10

3.2 FAPhysicsShapes::RigidBox Class Reference . . . 10

3.2.1 Constructor & Destructor Documentation . . . 11

3.2.1.1 RigidBox() . . . 11

3.2.2 Member Function Documentation . . . 11

3.2.2.1 GetDepth() . . . 11

3.2.2.2 GetHeight() . . . 11

3.2.2.3 GetRigidBody() [1/2] . . . 11
3.2.2.4 GetRigidBody() [2/2] . . . 12
3.2.2.5 GetShape() [1/2] . . . 12
3.2.2.6 GetShape() [2/2] . . . 12
3.2.2.7 GetWidth() . . . 12
3.2.2.8 InitializeRigidBox() . . . 12
3.2.2.9 SetDepth() . . . 13
3.2.2.10 SetHeight() . . . 13
3.2.2.11 SetPosition() . . . 13
3.2.2.12 SetWidth() . . . 13
3.2.2.13 UpdateModelMatrix() . . . 13
3.2.2.14 Volume() . . . 14
3.3 FAPhysicsShapes::RigidCone Class Reference . . . 14
3.3.1 Constructor & Destructor Documentation . . . 14
3.3.1.1 RigidCone() . . . 15
3.3.2 Member Function Documentation . . . 15
3.3.2.1 GetHeight() . . . 15
3.3.2.2 GetRadius() . . . 15
3.3.2.3 GetRigidBody() [1/2] . . . 15
3.3.2.4 GetRigidBody() [2/2] . . . 15
3.3.2.5 GetShape() [1/2] . . . 15
3.3.2.6 GetShape() [2/2] . . . 16
3.3.2.7 InitializeRigidCone() . . . 16
3.3.2.8 SetHeight() . . . 16
3.3.2.9 SetPosition() . . . 16
3.3.2.10 SetRadius() . . . 17
3.3.2.11 UpdateModelMatrix() . . . 17
3.3.2.12 Volume() . . . 17
3.4 FAPhysicsShapes::RigidCylinder Class Reference . . . 17
3.4.1 Constructor & Destructor Documentation . . . 18
3.4.1.1 RigidCylinder() . . . 18
3.4.2 Member Function Documentation . . . 18
3.4.2.1 GetHeight() . . . 18
3.4.2.2 GetRadius() . . . 18
3.4.2.3 GetRigidBody() [1/2] . . . 18
3.4.2.4 GetRigidBody() [2/2] . . . 19
3.4.2.5 GetShape() [1/2] . . . 19
3.4.2.6 GetShape() [2/2] . . . 19
3.4.2.7 InitializeRigidCylinder() . . . 19
3.4.2.8 SetHeight() . . . 20
3.4.2.9 SetPosition() . . . 20
3.4.2.10 SetRadius() . . . 20

3.4.2.11 UpdateModelMatrix() . . . 20
3.4.2.12 Volume() . . . 20
3.5 FAPhysicsShapes::RigidPyramid Class Reference . . . 21
3.5.1 Constructor & Destructor Documentation . . . 21
3.5.1.1 RigidPyramid() . . . 21
3.5.2 Member Function Documentation . . . 22
3.5.2.1 GetDepth() . . . 22
3.5.2.2 GetHeight() . . . 22
3.5.2.3 GetRigidBody() [1/2] . . . 22
3.5.2.4 GetRigidBody() [2/2] . . . 22
3.5.2.5 GetShape() [1/2] . . . 22
3.5.2.6 GetShape() [2/2] . . . 23
3.5.2.7 GetWidth() . . . 23
3.5.2.8 InitializeRigidPyramid() . . . 23
3.5.2.9 SetDepth() . . . 23
3.5.2.10 SetHeight() . . . 24
3.5.2.11 SetPosition() . . . 24
3.5.2.12 SetWidth() . . . 24
3.5.2.13 UpdateModelMatrix() . . . 24
3.5.2.14 Volume() . . . 24
3.6 FAPhysicsShapes::RigidSphere Class Reference . . . 25
3.6.1 Constructor & Destructor Documentation . . . 25
3.6.1.1 RigidSphere() . . . 25
3.6.2 Member Function Documentation . . . 25
3.6.2.1 GetRadius() . . . 25
3.6.2.2 GetRigidBody() [1/2] . . . 26
3.6.2.3 GetRigidBody() [2/2] . . . 26
3.6.2.4 GetShape() [1/2] . . . 26
3.6.2.5 GetShape() [2/2] . . . 26
3.6.2.6 InitializeRigidSphere() . . . 26
3.6.2.7 SetPosition() . . . 27
3.6.2.8 SetRadius() . . . 27
3.6.2.9 UpdateModelMatrix() . . . 27
3.6.2.10 Volume() . . . 27

4 File Documentation **29**

4.1 FAForceFunctions.h . . . 29
4.2 FARigidBody.h . . . 29
4.3 FARigidBox.h . . . 30
4.4 FARigidCone.h . . . 31
4.5 FARigidCylinder.h . . . 32
4.6 FARigidPyramid.h . . . 32

4.7 FARigidSphere.h 33
4.8 PolyhedralMassProperties.h 33

Index **35**

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

FAPhysics::RigidBody 5
FAPhysicsShapes::RigidBox 10
FAPhysicsShapes::RigidCone 14
FAPhysicsShapes::RigidCylinder 17
FAPhysicsShapes::RigidPyramid 21
FAPhysicsShapes::RigidSphere 25

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

C:/Users/Work/Desktop/First Game Engine/First-Game-Engine/FA Physics Engine/Header Files/FAForceFunctions.h
29
C:/Users/Work/Desktop/First Game Engine/First-Game-Engine/FA Physics Engine/Header Files/FARigidBody.h
29
C:/Users/Work/Desktop/First Game Engine/First-Game-Engine/FA Physics Engine/Header Files/FARigidBox.h
30
C:/Users/Work/Desktop/First Game Engine/First-Game-Engine/FA Physics Engine/Header Files/FARigidCone.h
31
C:/Users/Work/Desktop/First Game Engine/First-Game-Engine/FA Physics Engine/Header Files/FARigidCylinder.h
32
C:/Users/Work/Desktop/First Game Engine/First-Game-Engine/FA Physics Engine/Header Files/FARigidPyramid.h
32
C:/Users/Work/Desktop/First Game Engine/First-Game-Engine/FA Physics Engine/Header Files/FARigidSphere.h
33
C:/Users/Work/Desktop/First Game Engine/First-Game-Engine/FA Physics Engine/Header Files/PolyhedralMassProperties.h
33

Chapter 3

Class Documentation

3.1 FAPhysics::RigidBody Class Reference

Public Member Functions

- RigidBody ()
- void InitializeRigidBody (float massDensity, const FAMath::Quaternion &initialOrientation, const std::vector< FAShapes::Triangle > &triangles)
- float GetMass () const
- float GetInverseMass () const
- const FAMath::Matrix3x3 & GetBodyInertiaTensor () const
- const FAMath::Matrix3x3 & GetInverseBodyInertiaTensor () const
- const FAMath::Vector4D & GetCenterOfMass () const
- const FAMath::Vector4D & GetLinearVelocity () const
- const FAMath::Vector4D & GetLinearMomentum () const
- const FAMath::Quaternion & GetOrientation () const
- const FAMath::Vector4D & GetAngularVelocity () const
- const FAMath::Vector4D & GetAngularMomentum () const
- const FAMath::Vector4D & GetNetForce () const
- const FAMath::Vector4D & GetNetTorque () const
- void SetMass (float mass)
- void SetCenterOfMass (const FAMath::Vector4D &centerOfMass)
- void SetLinearVelocity (const FAMath::Vector4D &velocity)
- void SetLinearMomentum (const FAMath::Vector4D &linearMomentum)
- void SetBodyInertiaTensor (const FAMath::Matrix3x3 &bodyInertia)
- void SetAngularVelocity (const FAMath::Vector4D &angularVelocity)
- void SetAngularMomentum (const FAMath::Vector4D &angularMomentum)
- void SetOrientation (const FAMath::Quaternion &orientation)
- void ResetForce ()
- void ResetTorque ()
- void AddForce (const FAMath::Vector4D &force)
- void AddTorque (const FAMath::Vector4D &force, const FAMath::Vector4D &point)
- void Integrate (float dt)

3.1.1 Constructor & Destructor Documentation

3.1.1.1 RigidBody()

```
FAPhysics::RigidBody::RigidBody ( )
```

brief Default Constructor. Initializes all scalar member variables to 1.0f and all vectors to the zero vector.

3.1.2 Member Function Documentation

3.1.2.1 AddForce()

```
void FAPhysics::RigidBody::AddForce (
            const FAMath::Vector4D & force )
```

brief Adds the specified force to the net force of a rigid body.

3.1.2.2 AddTorque()

```
void FAPhysics::RigidBody::AddTorque (
            const FAMath::Vector4D & force,
            const FAMath::Vector4D & point )
```

brief Adds the computed torque to the net torque. Computes the torque being applied to the point using the equation torque = force x (point - center of mass).

3.1.2.3 GetAngularMomentum()

```
const FAMath::Vector4D & FAPhysics::RigidBody::GetAngularMomentum ( ) const
```

brief Returns the angular momentum of the rigid body.

3.1.2.4 GetAngularVelocity()

```
const FAMath::Vector4D & FAPhysics::RigidBody::GetAngularVelocity ( ) const
```

brief Returns the angular velocity of the rigid body.

3.1.2.5 GetBodyInertiaTensor()

```
const FAMath::Matrix3x3 & FAPhysics::RigidBody::GetBodyInertiaTensor ( ) const
```

brief Returns the inertia tensor in body coordinates.

3.1.2.6 GetCenterOfMass()

```
const FAMath::Vector4D & FAPhysics::RigidBody::GetCenterOfMass ( ) const
```

brief Returns the center of mass of the rigid body.

3.1.2.7 GetInverseBodyInertiaTensor()

```
const FAMath::Matrix3x3 & FAPhysics::RigidBody::GetInverseBodyInertiaTensor ( ) const
```

brief Returns the inverse of the inertia tensor in body coordinates.

3.1.2.8 GetInverseMass()

```
float FAPhysics::RigidBody::GetInverseMass ( ) const
```

brief Returns the inverse mass of the rigid body.

If the inverse mass equals to 0 that means the mass is infinity.

3.1.2.9 GetLinearMomentum()

```
const FAMath::Vector4D & FAPhysics::RigidBody::GetLinearMomentum ( ) const
```

brief Returns the linear momentum of the rigid body.

3.1.2.10 GetLinearVelocity()

```
const FAMath::Vector4D & FAPhysics::RigidBody::GetLinearVelocity ( ) const
```

brief Returns the linear velocity of the rigid body.

3.1.2.11 GetMass()

```
float FAPhysics::RigidBody::GetMass ( ) const
```

brief Returns the mass of the rigid body.

3.1.2.12 GetNetForce()

```
const FAMath::Vector4D & FAPhysics::RigidBody::GetNetForce ( ) const
```

brief Returns the net force acting on the rigid body.

3.1.2.13 GetNetTorque()

```
const FAMath::Vector4D & FAPhysics::RigidBody::GetNetTorque ( ) const
```

brief Returns the net torque acting on the rigid body.

3.1.2.14 GetOrientation()

```
const FAMath::Quaternion & FAPhysics::RigidBody::GetOrientation ( ) const
```

brief Returns the orientaiton quaternion of the rigid body.

3.1.2.15 InitializeRigidBody()

```
void FAPhysics::RigidBody::InitializeRigidBody (
            float massDensity,
            const FAMath::Quaternion & initialOrientation,
            const std::vector< FAShapes::Triangle > & triangles )
```

brief Initializes the properties of a rigid body.

If you want the rigid body to have infinite mass, so it can't be moved, pass in 0.0f for the mass density and the inverse mass will be set to 0.0f to indicate infinite mass.

If the specified mass density is negative, the mass and inverse mass will be set to 0.0f.

Computes the center of mass and inertia tensors from the given triangles that make up a solid polyhedron if the object does not have infinite mass.

3.1.2.16 Integrate()

```
void FAPhysics::RigidBody::Integrate (
            float dt )
```

brief A numerical integrator using semi-implicit Euler method. Uses semi-implicit Euler method to compute the new position and orientation of a rigid body.

3.1.2.17 ResetForce()

```
void FAPhysics::RigidBody::ResetForce ( )
```

brief Sets the net force of a rigid body to the zero vector.

3.1.2.18 ResetTorque()

```
void FAPhysics::RigidBody::ResetTorque ( )
```

brief Sets the net torque of a rigid body to the zero vector.

3.1.2.19 SetAngularMomentum()

```
void FAPhysics::RigidBody::SetAngularMomentum (
            const FAMath::Vector4D & angularMomentum )
```

brief Sets the angular momentum of the rigid body to the specified vector.

3.1.2.20 SetAngularVelocity()

```
void FAPhysics::RigidBody::SetAngularVelocity (
            const FAMath::Vector4D & angularVelocity )
```

brief Sets the angular velocity of the rigid body to the specified vector.

3.1.2.21 SetBodyInertiaTensor()

```
void FAPhysics::RigidBody::SetBodyInertiaTensor (
            const FAMath::Matrix3x3 & bodyInertia )
```

brief Sets the body inertia tensor to the specified matrix.

3.1.2.22 SetCenterOfMass()

```
void FAPhysics::RigidBody::SetCenterOfMass (
            const FAMath::Vector4D & centerOfMass )
```

brief Sets the center of mass the rigid body to the specified vector.

3.1.2.23 SetLinearMomentum()

```
void FAPhysics::RigidBody::SetLinearMomentum (
            const FAMath::Vector4D & linearMomentum )
```

brief Sets the linear momentum of the rigid body to the specified vector.

3.1.2.24 SetLinearVelocity()

```
void FAPhysics::RigidBody::SetLinearVelocity (
            const FAMath::Vector4D & velocity )
```

brief Sets the linear velocity of the rigid body to the specified vector.

3.1.2.25 SetMass()

```
void FAPhysics::RigidBody::SetMass (
            float mass )
```

brief Sets the mass of the rigid body to the specified float.

If you want the rigid body to have infinite mass, so it can't be moved, pass in 0.0f for the mass and the inverse mass will be set to 0.0f to indicate infinite mass.

If the specified mass is negative, the mass and inverse mass will be set to 0.0f.

3.1.2.26 SetOrientation()

```
void FAPhysics::RigidBody::SetOrientation (
            const FAMath::Quaternion & orientation )
```

brief Sets the orientation of the rigid body to the specified quaternion.

The documentation for this class was generated from the following file:

- C:/Users/Work/Desktop/First Game Engine/First-Game-Engine/FA Physics Engine/Header Files/FARigid↩ Body.h

3.2 FAPhysicsShapes::RigidBox Class Reference

Public Member Functions

- RigidBox ()

  Default Constructor. Constructs a RigidBox object.
- void InitializeRigidBox (float width, float height, float depth, float massDensity, const FAMath::Vector4D &initialPosition, const FAMath::Quaternion &initialOrientation, const FAColor::Color &color, const std::vector< FAShapes::Vertex > &vertices, const std::vector< FAShapes::Triangle > &triangles)

  Iniitalizes a rigid box that can be used to do physics simulations.
- float GetWidth () const

  Returns the width of the box.
- float GetHeight () const

  Returns the height of the box.
- float GetDepth () const

  Returns the depth of the box.
- void SetWidth (float width)

  Sets the width of the box.
- void SetHeight (float height)

  Sets the height of the box.
- void SetDepth (float depth)

  Sets the depth of the box.
- const FAPhysics::RigidBody & GetRigidBody () const

  Returns the RigidBody object.
- FAPhysics::RigidBody & GetRigidBody ()

  Returns the RigidBody object,.

- const FAShapes::ThreeDimensionalShape & GetShape () const

  Returns the ThreeDimensionalShape object.
- FAShapes::ThreeDimensionalShape & GetShape ()

  Returns the ThreeDimensionalShape object.
- void SetPosition (const FAMath::Vector4D &position)

  Sets the position of the RigidBox.
- void UpdateModelMatrix ()

  Updates the model matrix of the RigidBox.
- float Volume ()

  Returns the volume of the box.

3.2.1 Constructor & Destructor Documentation

3.2.1.1 RigidBox()

```
FAPhysicsShapes::RigidBox::RigidBox ( )
```

Default Constructor. Constructs a RigidBox object.

3.2.2 Member Function Documentation

3.2.2.1 GetDepth()

```
float FAPhysicsShapes::RigidBox::GetDepth ( ) const
```

Returns the depth of the box.

3.2.2.2 GetHeight()

```
float FAPhysicsShapes::RigidBox::GetHeight ( ) const
```

Returns the height of the box.

3.2.2.3 GetRigidBody() [1/2]

```
FAPhysics::RigidBody & FAPhysicsShapes::RigidBox::GetRigidBody ( )
```

Returns the RigidBody object,.

3.2.2.4 GetRigidBody() [2/2]

```
const FAPhysics::RigidBody & FAPhysicsShapes::RigidBox::GetRigidBody ( ) const
```

Returns the RigidBody object.

3.2.2.5 GetShape() [1/2]

```
FAShapes::ThreeDimensionalShape & FAPhysicsShapes::RigidBox::GetShape ( )
```

Returns the ThreeDimensionalShape object.

3.2.2.6 GetShape() [2/2]

```
const FAShapes::ThreeDimensionalShape & FAPhysicsShapes::RigidBox::GetShape ( ) const
```

Returns the ThreeDimensionalShape object.

3.2.2.7 GetWidth()

```
float FAPhysicsShapes::RigidBox::GetWidth ( ) const
```

Returns the width of the box.

3.2.2.8 InitializeRigidBox()

```
void FAPhysicsShapes::RigidBox::InitializeRigidBox (
            float width,
            float height,
            float depth,
            float massDensity,
            const FAMath::Vector4D & initialPosition,
            const FAMath::Quaternion & initialOrientation,
            const FAColor::Color & color,
            const std::vector< FAShapes::Vertex > & vertices,
            const std::vector< FAShapes::Triangle > & triangles )
```

Iniitalizes a rigid box that can be used to do physics simulations.

Parameters

in	*width*	The width of the box.
in	*height*	The height of the box.
in	*depth*	The depth of the box.
in	*color*	The color of the box.
in	*massDensity*	The mass density of the box.
in	*initialPosition*	The initial position of the box

3.2.2.9 SetDepth()

```
void FAPhysicsShapes::RigidBox::SetDepth (
            float depth )
```

Sets the depth of the box.

3.2.2.10 SetHeight()

```
void FAPhysicsShapes::RigidBox::SetHeight (
            float height )
```

Sets the height of the box.

3.2.2.11 SetPosition()

```
void FAPhysicsShapes::RigidBox::SetPosition (
            const FAMath::Vector4D & position )
```

Sets the position of the RigidBox.

3.2.2.12 SetWidth()

```
void FAPhysicsShapes::RigidBox::SetWidth (
            float width )
```

Sets the width of the box.

3.2.2.13 UpdateModelMatrix()

```
void FAPhysicsShapes::RigidBox::UpdateModelMatrix ( )
```

Updates the model matrix of the RigidBox.

3.2.2.14 Volume()

```
float FAPhysicsShapes::RigidBox::Volume ( )
```

Returns the volume of the box.

The documentation for this class was generated from the following file:

- C:/Users/Work/Desktop/First Game Engine/First-Game-Engine/FA Physics Engine/Header Files/FARigid↩ Box.h

3.3 FAPhysicsShapes::RigidCone Class Reference

Public Member Functions

- RigidCone ()

  Default Constructor. Constructs a RigidCone object.
- void InitializeRigidCone (float radius, float height, float massDensity, const FAMath::Vector4D &initialPosition, const FAMath::Quaternion &initialOrientation, const FAColor::Color &color, const std::vector< FAShapes::↩ Vertex > &vertices, const std::vector< FAShapes::Triangle > &triangles)

  Iniitalizes a rigid cone that can be used to do physics simulations.
- float GetRadius () const

  Returns the radius of the cone.
- float GetHeight () const

  Returns the height of the cone.
- void SetRadius (float radius)

  Sets the radius of the cone.
- void SetHeight (float height)

  Sets the height of the cone.
- const FAPhysics::RigidBody & GetRigidBody () const

  Returns the RigidBody object of the RigidCone.
- FAPhysics::RigidBody & GetRigidBody ()

  Returns the RigidBody object of the RigidCone.
- const FAShapes::ThreeDimensionalShape & GetShape () const

  Returns the ThreeDimensionalShape object.
- FAShapes::ThreeDimensionalShape & GetShape ()

  Returns the ThreeDimensionalShape.
- void SetPosition (const FAMath::Vector4D &position)

  Sets the position of the RigidCone.
- void UpdateModelMatrix ()

  Updates the model matrix of the RigidCone.
- float Volume ()

  Returns the volume of the cone.

3.3.1 Constructor & Destructor Documentation

3.3.1.1 RigidCone()

```
FAPhysicsShapes::RigidCone::RigidCone ( )
```

Default Constructor. Constructs a RigidCone object.

3.3.2 Member Function Documentation

3.3.2.1 GetHeight()

```
float FAPhysicsShapes::RigidCone::GetHeight ( ) const
```

Returns the height of the cone.

3.3.2.2 GetRadius()

```
float FAPhysicsShapes::RigidCone::GetRadius ( ) const
```

Returns the radius of the cone.

3.3.2.3 GetRigidBody() [1/2]

```
FAPhysics::RigidBody & FAPhysicsShapes::RigidCone::GetRigidBody ( )
```

Returns the RigidBody object of the RigidCone.

3.3.2.4 GetRigidBody() [2/2]

```
const FAPhysics::RigidBody & FAPhysicsShapes::RigidCone::GetRigidBody ( ) const
```

Returns the RigidBody object of the RigidCone.

3.3.2.5 GetShape() [1/2]

```
FAShapes::ThreeDimensionalShape & FAPhysicsShapes::RigidCone::GetShape ( )
```

Returns the ThreeDimensionalShape.

3.3.2.6 GetShape() [2/2]

```
const FAShapes::ThreeDimensionalShape & FAPhysicsShapes::RigidCone::GetShape ( ) const
```

Returns the ThreeDimensionalShape object.

3.3.2.7 InitializeRigidCone()

```
void FAPhysicsShapes::RigidCone::InitializeRigidCone (
            float radius,
            float height,
            float massDensity,
            const FAMath::Vector4D & initialPosition,
            const FAMath::Quaternion & initialOrientation,
            const FAColor::Color & color,
            const std::vector< FAShapes::Vertex > & vertices,
            const std::vector< FAShapes::Triangle > & triangles )
```

Iniitalizes a rigid cone that can be used to do physics simulations.

Parameters

in	*radius*	The radius of the cone.
in	*height*	The height of the cone.
in	*color*	The color of the cone.
in	*massDensity*	The mass density of the cone.
in	*initialPosition*	The initial position of the cone.
in	*initialOrientation*	The initial orientation of the cone.
in	*color*	The color of the cone.
in	*vertices*	The vertex list of a unit cone.
in	*triangles*	The triangle list of a unit cone.

3.3.2.8 SetHeight()

```
void FAPhysicsShapes::RigidCone::SetHeight (
            float height )
```

Sets the height of the cone.

3.3.2.9 SetPosition()

```
void FAPhysicsShapes::RigidCone::SetPosition (
            const FAMath::Vector4D & position )
```

Sets the position of the RigidCone.

3.3.2.10 SetRadius()

```
void FAPhysicsShapes::RigidCone::SetRadius (
            float radius )
```

Sets the radius of the cone.

3.3.2.11 UpdateModelMatrix()

```
void FAPhysicsShapes::RigidCone::UpdateModelMatrix ( )
```

Updates the model matrix of the RigidCone.

3.3.2.12 Volume()

```
float FAPhysicsShapes::RigidCone::Volume ( )
```

Returns the volume of the cone.

The documentation for this class was generated from the following file:

- C:/Users/Work/Desktop/First Game Engine/First-Game-Engine/FA Physics Engine/Header Files/FARigid↩ Cone.h

3.4 FAPhysicsShapes::RigidCylinder Class Reference

Public Member Functions

- RigidCylinder ()

  Default Constructor. Constructs a RigidCylinder object.
- void InitializeRigidCylinder (float radius, float height, float massDensity, const FAMath::Vector4D &initial↩ Position, const FAMath::Quaternion &initialOrientation, const FAColor::Color &color, const std::vector< FAShapes::Vertex > &vertices, const std::vector< FAShapes::Triangle > &triangles)

  Iniitalizes a rigid cylinder that can be used to do physics simulations.
- float GetRadius () const

  Returns the radius of the cylinder.
- float GetHeight () const

  Returns the height of the cylinder.
- void SetRadius (float radius)

  Sets the radius of the cylinder.
- void SetHeight (float height)

  Sets the height of the cylinder.
- const FAPhysics::RigidBody & GetRigidBody () const

  Returns the RigidBody object.
- FAPhysics::RigidBody & GetRigidBody ()

Returns the RigidBody object.

- const FAShapes::ThreeDimensionalShape & GetShape () const

  Returns the ThreeDimensionalShape object.
- FAShapes::ThreeDimensionalShape & GetShape ()

  Returns the ThreeDimensionalShape object.
- void SetPosition (const FAMath::Vector4D &position)

  Sets the position of the RigidCylinder.
- void UpdateModelMatrix ()

  Updates the model matrix of the RigidCylinder.
- float Volume ()

  Returns the volume of the cylinder.

3.4.1 Constructor & Destructor Documentation

3.4.1.1 RigidCylinder()

```
FAPhysicsShapes::RigidCylinder::RigidCylinder ( )
```

Default Constructor. Constructs a RigidCylinder object.

3.4.2 Member Function Documentation

3.4.2.1 GetHeight()

```
float FAPhysicsShapes::RigidCylinder::GetHeight ( ) const
```

Returns the height of the cylinder.

3.4.2.2 GetRadius()

```
float FAPhysicsShapes::RigidCylinder::GetRadius ( ) const
```

Returns the radius of the cylinder.

3.4.2.3 GetRigidBody() [1/2]

```
FAPhysics::RigidBody & FAPhysicsShapes::RigidCylinder::GetRigidBody ( )
```

Returns the RigidBody object.

3.4.2.4 GetRigidBody() [2/2]

```
const FAPhysics::RigidBody & FAPhysicsShapes::RigidCylinder::GetRigidBody ( ) const
```

Returns the RigidBody object.

3.4.2.5 GetShape() [1/2]

```
FAShapes::ThreeDimensionalShape & FAPhysicsShapes::RigidCylinder::GetShape ( )
```

Returns the ThreeDimensionalShape object.

3.4.2.6 GetShape() [2/2]

```
const FAShapes::ThreeDimensionalShape & FAPhysicsShapes::RigidCylinder::GetShape ( ) const
```

Returns the ThreeDimensionalShape object.

3.4.2.7 InitializeRigidCylinder()

```
void FAPhysicsShapes::RigidCylinder::InitializeRigidCylinder (
            float radius,
            float height,
            float massDensity,
            const FAMath::Vector4D & initialPosition,
            const FAMath::Quaternion & initialOrientation,
            const FAColor::Color & color,
            const std::vector< FAShapes::Vertex > & vertices,
            const std::vector< FAShapes::Triangle > & triangles )
```

Iniitalizes a rigid cylinder that can be used to do physics simulations.

Parameters

in	*radius*	The radius of the cylinder.
in	*height*	The height of the cylinder.
in	*color*	The color of the cylinder.
in	*massDensity*	The mass density of the cylinder.
in	*initialPosition*	The initial position of the cylinder.
in	*initialOrientation*	The initial orientation of the cylinder.
in	*color*	The color of the cylinder.
in	*vertices*	The vertex list of a unit cylinder.
in	*triangles*	The triangle list of a unit cylinder.

3.4.2.8 SetHeight()

```
void FAPhysicsShapes::RigidCylinder::SetHeight (
            float height )
```

Sets the height of the cylinder.

3.4.2.9 SetPosition()

```
void FAPhysicsShapes::RigidCylinder::SetPosition (
            const FAMath::Vector4D & position )
```

Sets the position of the RigidCylinder.

3.4.2.10 SetRadius()

```
void FAPhysicsShapes::RigidCylinder::SetRadius (
            float radius )
```

Sets the radius of the cylinder.

3.4.2.11 UpdateModelMatrix()

```
void FAPhysicsShapes::RigidCylinder::UpdateModelMatrix ( )
```

Updates the model matrix of the RigidCylinder.

3.4.2.12 Volume()

```
float FAPhysicsShapes::RigidCylinder::Volume ( )
```

Returns the volume of the cylinder.

The documentation for this class was generated from the following file:

- C:/Users/Work/Desktop/First Game Engine/First-Game-Engine/FA Physics Engine/Header Files/FARigid↩ Cylinder.h

3.5 FAPhysicsShapes::RigidPyramid Class Reference

Public Member Functions

- RigidPyramid ()

  Default Constructor. Constructs a RigidPyramid object.
- void InitializeRigidPyramid (float width, float height, float depth, float massDensity, const FAMath::Vector4D &initialPosition, const FAMath::Quaternion &initialOrientation, const FAColor::Color &color, const std::vector< FAShapes::Vertex > &vertices, const std::vector< FAShapes::Triangle > &triangles)

  Iniitalizes a rigid pyramid that can be used to do physics simulations.
- float GetWidth () const

  Returns the width of the pyramid.
- float GetHeight () const

  Returns the height of the pyramid.
- float GetDepth () const

  Returns the depth of the pyramid.
- void SetWidth (float width)

  Sets the width of the pyramid.
- void SetHeight (float height)

  Sets the height of the pyramid.
- void SetDepth (float depth)

  Sets the depth of the pyramid.
- const FAPhysics::RigidBody & GetRigidBody () const

  Returns the RigidBody object.
- FAPhysics::RigidBody & GetRigidBody ()

  Returns the RigidBody object.
- const FAShapes::ThreeDimensionalShape & GetShape () const

  Returns the ThreeDimensionalShape object.
- FAShapes::ThreeDimensionalShape & GetShape ()

  Returns the ThreeDimensionalShape object.
- void SetPosition (const FAMath::Vector4D &position)

  Sets the position of the RigidPyramid.
- void UpdateModelMatrix ()

  Updates the model matrix of the RigidPyramid.
- float Volume ()

  Returns the volume of the pyramid.

3.5.1 Constructor & Destructor Documentation

3.5.1.1 RigidPyramid()

```
FAPhysicsShapes::RigidPyramid::RigidPyramid ( )
```

Default Constructor. Constructs a RigidPyramid object.

3.5.2 Member Function Documentation

3.5.2.1 GetDepth()

```
float FAPhysicsShapes::RigidPyramid::GetDepth ( ) const
```

Returns the depth of the pyramid.

3.5.2.2 GetHeight()

```
float FAPhysicsShapes::RigidPyramid::GetHeight ( ) const
```

Returns the height of the pyramid.

3.5.2.3 GetRigidBody() [1/2]

```
FAPhysics::RigidBody & FAPhysicsShapes::RigidPyramid::GetRigidBody ( )
```

Returns the RigidBody object.

3.5.2.4 GetRigidBody() [2/2]

```
const FAPhysics::RigidBody & FAPhysicsShapes::RigidPyramid::GetRigidBody ( ) const
```

Returns the RigidBody object.

3.5.2.5 GetShape() [1/2]

```
FAShapes::ThreeDimensionalShape & FAPhysicsShapes::RigidPyramid::GetShape ( )
```

Returns the ThreeDimensionalShape object.

3.5.2.6 GetShape() [2/2]

```
const FAShapes::ThreeDimensionalShape & FAPhysicsShapes::RigidPyramid::GetShape ( ) const
```

Returns the ThreeDimensionalShape object.

3.5.2.7 GetWidth()

```
float FAPhysicsShapes::RigidPyramid::GetWidth ( ) const
```

Returns the width of the pyramid.

3.5.2.8 InitializeRigidPyramid()

```
void FAPhysicsShapes::RigidPyramid::InitializeRigidPyramid (
            float width,
            float height,
            float depth,
            float massDensity,
            const FAMath::Vector4D & initialPosition,
            const FAMath::Quaternion & initialOrientation,
            const FAColor::Color & color,
            const std::vector< FAShapes::Vertex > & vertices,
            const std::vector< FAShapes::Triangle > & triangles )
```

Iniitalizes a rigid pyramid that can be used to do physics simulations.

Parameters

in	*width*	The width of the pyramid.
in	*height*	The height of the pyramid.
in	*depth*	The depth of the pyramid.
in	*color*	The color of the pyramid.
in	*massDensity*	The mass density of the pyramid.
in	*initialPosition*	The initial position of the pyramid.
in	*initialOrientation*	The initial orientation of the pyramid.
in	*color*	The color of the pyramid.
in	*vertices*	The vertex list of a unit pyramid.
in	*triangles*	The triangle list of a unit pyramid.

3.5.2.9 SetDepth()

```
void FAPhysicsShapes::RigidPyramid::SetDepth (
            float depth )
```

Sets the depth of the pyramid.

3.5.2.10 SetHeight()

```
void FAPhysicsShapes::RigidPyramid::SetHeight (
            float height )
```

Sets the height of the pyramid.

3.5.2.11 SetPosition()

```
void FAPhysicsShapes::RigidPyramid::SetPosition (
            const FAMath::Vector4D & position )
```

Sets the position of the RigidPyramid.

3.5.2.12 SetWidth()

```
void FAPhysicsShapes::RigidPyramid::SetWidth (
            float width )
```

Sets the width of the pyramid.

3.5.2.13 UpdateModelMatrix()

```
void FAPhysicsShapes::RigidPyramid::UpdateModelMatrix ( )
```

Updates the model matrix of the RigidPyramid.

3.5.2.14 Volume()

```
float FAPhysicsShapes::RigidPyramid::Volume ( )
```

Returns the volume of the pyramid.

The documentation for this class was generated from the following file:

- C:/Users/Work/Desktop/First Game Engine/First-Game-Engine/FA Physics Engine/Header Files/FARigid↩ Pyramid.h

3.6 FAPhysicsShapes::RigidSphere Class Reference

Public Member Functions

- RigidSphere ()

  Default Constructor. Constructs a RigidSphere object.
- void InitializeRigidSphere (float radius, float massDensity, const FAMath::Vector4D &initialPosition, const FAMath::Quaternion &initialOrientation, const FAColor::Color &color, const std::vector< FAShapes::Vertex > &vertices, const std::vector< FAShapes::Triangle > &triangles)

  Iniitalizes a rigid sphere that can be used to do physics simulations.
- float GetRadius () const

  Returns the radius of the sphere.
- void SetRadius (float radius)

  Sets the radius of the sphere.
- const FAPhysics::RigidBody & GetRigidBody () const

  Returns the RigidBody object.
- FAPhysics::RigidBody & GetRigidBody ()

  Returns the RigidBody object.
- const FAShapes::ThreeDimensionalShape & GetShape () const

  Returns the ThreeDimensionalShape object.
- FAShapes::ThreeDimensionalShape & GetShape ()

  Returns the ThreeDimensionalShape object.
- void SetPosition (const FAMath::Vector4D &position)

  Sets the position of the RigidSphere.
- void UpdateModelMatrix ()

  Updates the model matrix of the RigidSphere.
- float Volume ()

  Returns the volume of the sphere.

3.6.1 Constructor & Destructor Documentation

3.6.1.1 RigidSphere()

```
FAPhysicsShapes::RigidSphere::RigidSphere ( )
```

Default Constructor. Constructs a RigidSphere object.

3.6.2 Member Function Documentation

3.6.2.1 GetRadius()

```
float FAPhysicsShapes::RigidSphere::GetRadius ( ) const
```

Returns the radius of the sphere.

3.6.2.2 GetRigidBody() [1/2]

```
FAPhysics::RigidBody & FAPhysicsShapes::RigidSphere::GetRigidBody ( )
```

Returns the RigidBody object.

3.6.2.3 GetRigidBody() [2/2]

```
const FAPhysics::RigidBody & FAPhysicsShapes::RigidSphere::GetRigidBody ( ) const
```

Returns the RigidBody object.

3.6.2.4 GetShape() [1/2]

```
FAShapes::ThreeDimensionalShape & FAPhysicsShapes::RigidSphere::GetShape ( )
```

Returns the ThreeDimensionalShape object.

3.6.2.5 GetShape() [2/2]

```
const FAShapes::ThreeDimensionalShape & FAPhysicsShapes::RigidSphere::GetShape ( ) const
```

Returns the ThreeDimensionalShape object.

3.6.2.6 InitializeRigidSphere()

```
void FAPhysicsShapes::RigidSphere::InitializeRigidSphere (
            float radius,
            float massDensity,
            const FAMath::Vector4D & initialPosition,
            const FAMath::Quaternion & initialOrientation,
            const FAColor::Color & color,
            const std::vector< FAShapes::Vertex > & vertices,
            const std::vector< FAShapes::Triangle > & triangles )
```

Iniitalizes a rigid sphere that can be used to do physics simulations.

Parameters

in	*radius*	The radius of the sphere.
in	*color*	The color of the sphere.
in	*massDensity*	The mass density of the sphere.
in	*initialPosition*	The initial position of the sphere.
in	*initialOrientation*	The initial orientation of the sphere.
in	*color*	The color of the sphere.
in	*vertices*	The vertex list of a unit sphere.

3.6.2.7 SetPosition()

```
void FAPhysicsShapes::RigidSphere::SetPosition (
            const FAMath::Vector4D & position )
```

Sets the position of the RigidSphere.

3.6.2.8 SetRadius()

```
void FAPhysicsShapes::RigidSphere::SetRadius (
            float radius )
```

Sets the radius of the sphere.

3.6.2.9 UpdateModelMatrix()

```
void FAPhysicsShapes::RigidSphere::UpdateModelMatrix ( )
```

Updates the model matrix of the RigidSphere.

3.6.2.10 Volume()

```
float FAPhysicsShapes::RigidSphere::Volume ( )
```

Returns the volume of the sphere.

The documentation for this class was generated from the following file:

- C:/Users/Work/Desktop/First Game Engine/First-Game-Engine/FA Physics Engine/Header Files/FARigid↩ Sphere.h

Chapter 4

File Documentation

4.1 FAForceFunctions.h

```
1 #pragma once
2
3 #include "FAMathEngine.h"
4
5 namespace FAPhysics
6 {
11     FAMath::Vector4D GravitationalForce(float mass, float gravityAcceleration, const FAMath::Vector4D&
      direction);
12
17     FAMath::Vector4D DragForce(float dragCoefficent, const FAMath::Vector4D& velocity);
18
23     FAMath::Vector4D ApplyForce(float magnitdue, const FAMath::Vector4D& direction);
24 }
```

4.2 FARigidBody.h

```
1 #pragma once
2
3 #include "PolyhedralMassProperties.h"
4 #include <vector>
5
6 namespace FAPhysics
7 {
8     class RigidBody
9     {
10     public:
14         RigidBody();
15
25         void InitializeRigidBody(float massDensity, const FAMath::Quaternion& initialOrientation, const
      std::vector<FAShapes::Triangle>& triangles);
26
29         float GetMass() const;
30
35         float GetInverseMass() const;
36
39         const FAMath::Matrix3x3& GetBodyInertiaTensor() const;
40
43         const FAMath::Matrix3x3& GetInverseBodyInertiaTensor() const;
44
47         const FAMath::Vector4D& GetCenterOfMass() const;
48
51         const FAMath::Vector4D& GetLinearVelocity() const;
52
55         const FAMath::Vector4D& GetLinearMomentum() const;
56
59         const FAMath::Quaternion& GetOrientation() const;
60
63         const FAMath::Vector4D& GetAngularVelocity() const;
64
67         const FAMath::Vector4D& GetAngularMomentum() const;
68
71         const FAMath::Vector4D& GetNetForce() const;
72
75         const FAMath::Vector4D& GetNetTorque() const;
```

```
76
84          void SetMass(float mass);
85
88          void SetCenterOfMass(const FAMath::Vector4D& centerOfMass);
89
92          void SetLinearVelocity(const FAMath::Vector4D& velocity);
93
96          void SetLinearMomentum(const FAMath::Vector4D& linearMomentum);
97
100          void SetBodyInertiaTensor(const FAMath::Matrix3x3& bodyInertia);
101
104          void SetAngularVelocity(const FAMath::Vector4D& angularVelocity);
105
108          void SetAngularMomentum(const FAMath::Vector4D& angularMomentum);
109
112          void SetOrientation(const FAMath::Quaternion& orientation);
113
116          void ResetForce();
117
120          void ResetTorque();
121
124          void AddForce(const FAMath::Vector4D& force);
125
129          void AddTorque(const FAMath::Vector4D& force, const FAMath::Vector4D& point);
130
134          void Integrate(float dt);
135
136      private:
137          float mMass;
138          float mInverseMass;
139
140          FAMath::Matrix3x3 mBodyInertiaTensor;
141          FAMath::Matrix3x3 mInverseBodyInertiaTensor;
142          FAMath::Matrix3x3 mWorldCMInertiaTensor;
143          FAMath::Matrix3x3 mInverseWorldCMInertiaTensor;
144
145          FAMath::Vector4D mCenterOfMass;
146          FAMath::Vector4D mLinearVelocity;
147          FAMath::Vector4D mLinearMomentum;
148          FAMath::Vector4D mNetForce;
149
150          FAMath::Quaternion mOrientation;
151          FAMath::Vector4D mAngularVelocity;
152          FAMath::Vector4D mAngularMomentum;
153          FAMath::Vector4D mNetTorque;
154      };
155
158      void Interpolate(const RigidBody& r1, const RigidBody& r2, RigidBody& r3, float t);
159  }
```

4.3 FARigidBox.h

```
1  #pragma once
2
3  #include "FARigidBody.h"
4  #include "FAThreeDimensionalShape.h"
5
6  namespace FAPhysicsShapes
7  {
8      class RigidBox
9
10      {
11      public:
12
16          RigidBox();
17
31          void InitializeRigidBox(float width, float height, float depth, float massDensity,
32              const FAMath::Vector4D& initialPosition, const FAMath::Quaternion& initialOrientation, const
      FAColor::Color& color,
33              const std::vector<FAShapes::Vertex>& vertices, const std::vector<FAShapes::Triangle>&
      triangles);
34
37          float GetWidth() const;
38
41          float GetHeight() const;
42
45          float GetDepth() const;
46
49          void SetWidth(float width);
50
53          void SetHeight(float height);
54
57          void SetDepth(float depth);
```

```
58
61          const FAPhysics::RigidBody& GetRigidBody() const;
62
65          FAPhysics::RigidBody& GetRigidBody();
66
69          const FAShapes::ThreeDimensionalShape& GetShape() const;
70
73          FAShapes::ThreeDimensionalShape& GetShape();
74
77          void SetPosition(const FAMath::Vector4D& position);
78
81          void UpdateModelMatrix();
82
85          float Volume();
86
87      private:
88          float mWidth;
89          float mHeight;
90          float mDepth;
91
92          FAPhysics::RigidBody mRigidBody;
93          FAShapes::ThreeDimensionalShape mShape;
94          FAMath::Vector4D mOffset;
95      };
96  }
```

4.4 FARigidCone.h

```
1  #pragma once
2
3  #include "FARigidBody.h"
4  #include "FAThreeDimensionalShape.h"
5
6  namespace FAPhysicsShapes
7  {
8      class RigidCone
9      {
10      public:
11
15          RigidCone();
16
29          void InitializeRigidCone(float radius, float height, float massDensity,
30              const FAMath::Vector4D& initialPosition, const FAMath::Quaternion& initialOrientation, const
       FAColor::Color& color,
31              const std::vector<FAShapes::Vertex>& vertices, const std::vector<FAShapes::Triangle>&
       triangles);
32
35          float GetRadius() const;
36
39          float GetHeight() const;
40
43          void SetRadius(float radius);
44
47          void SetHeight(float height);
48
51          const FAPhysics::RigidBody& GetRigidBody() const;
52
55          FAPhysics::RigidBody& GetRigidBody();
56
59          const FAShapes::ThreeDimensionalShape& GetShape() const;
60
63          FAShapes::ThreeDimensionalShape& GetShape();
64
67          void SetPosition(const FAMath::Vector4D& position);
68
71          void UpdateModelMatrix();
72
75          float Volume();
76
77      private:
78          float mRadius;
79          float mHeight;
80
81          FAPhysics::RigidBody mRigidBody;
82          FAShapes::ThreeDimensionalShape mShape;
83          FAMath::Vector4D mOffset;
84      };
85  }
```

4.5 FARigidCylinder.h

```
1 #pragma once
2
3 #include "FARigidBody.h"
4 #include "FAThreeDimensionalShape.h"
5
6 namespace FAPhysicsShapes
7 {
8     class RigidCylinder
9     {
10     public:
11
15         RigidCylinder();
16
29         void InitializeRigidCylinder(float radius, float height, float massDensity,
30             const FAMath::Vector4D& initialPosition, const FAMath::Quaternion& initialOrientation, const
    FAColor::Color& color,
31             const std::vector<FAShapes::Vertex>& vertices, const std::vector<FAShapes::Triangle>&
    triangles);
32
35         float GetRadius() const;
36
39         float GetHeight() const;
40
43         void SetRadius(float radius);
44
47         void SetHeight(float height);
48
51         const FAPhysics::RigidBody& GetRigidBody() const;
52
55         FAPhysics::RigidBody& GetRigidBody();
56
59         const FAShapes::ThreeDimensionalShape& GetShape() const;
60
63         FAShapes::ThreeDimensionalShape& GetShape();
64
67         void SetPosition(const FAMath::Vector4D& position);
68
71         void UpdateModelMatrix();
72
75         float Volume();
76
77     private:
78         float mRadius;
79         float mHeight;
80
81         FAPhysics::RigidBody mRigidBody;
82         FAShapes::ThreeDimensionalShape mShape;
83         FAMath::Vector4D mOffset;
84     };
85 }
```

4.6 FARigidPyramid.h

```
1 #pragma once
2
3 #include "FARigidBody.h"
4 #include "FAThreeDimensionalShape.h"
5
6 namespace FAPhysicsShapes
7 {
8     class RigidPyramid
9
10     {
11     public:
12
16         RigidPyramid();
17
31         void InitializeRigidPyramid(float width, float height, float depth, float massDensity,
32             const FAMath::Vector4D& initialPosition, const FAMath::Quaternion& initialOrientation, const
    FAColor::Color& color,
33             const std::vector<FAShapes::Vertex>& vertices, const std::vector<FAShapes::Triangle>&
    triangles);
34
37         float GetWidth() const;
38
41         float GetHeight() const;
42
45         float GetDepth() const;
46
49         void SetWidth(float width);
50
```

```
53          void SetHeight(float height);
54
57          void SetDepth(float depth);
58
61          const FAPhysics::RigidBody& GetRigidBody() const;
62
65          FAPhysics::RigidBody& GetRigidBody();
66
69          const FAShapes::ThreeDimensionalShape& GetShape() const;
70
73          FAShapes::ThreeDimensionalShape& GetShape();
74
77          void SetPosition(const FAMath::Vector4D& position);
78
81          void UpdateModelMatrix();
82
85          float Volume();
86
87      private:
88          float mWidth;
89          float mHeight;
90          float mDepth;
91
92          FAPhysics::RigidBody mRigidBody;
93          FAShapes::ThreeDimensionalShape mShape;
94          FAMath::Vector4D mOffset;
95      };
96  }
```

4.7 FARigidSphere.h

```
1  #pragma once
2
3  #include "FARigidBody.h"
4  #include "FAThreeDimensionalShape.h"
5
6  namespace FAPhysicsShapes
7  {
8      class RigidSphere
9      {
10      public:
11
15          RigidSphere();
16
28          void InitializeRigidSphere(float radius, float massDensity,
29              const FAMath::Vector4D& initialPosition, const FAMath::Quaternion& initialOrientation, const
       FAColor::Color& color,
30              const std::vector<FAShapes::Vertex>& vertices, const std::vector<FAShapes::Triangle>&
       triangles);
31
34          float GetRadius() const;
35
38          void SetRadius(float radius);
39
42          const FAPhysics::RigidBody& GetRigidBody() const;
43
46          FAPhysics::RigidBody& GetRigidBody();
47
50          const FAShapes::ThreeDimensionalShape& GetShape() const;
51
54          FAShapes::ThreeDimensionalShape& GetShape();
55
58          void SetPosition(const FAMath::Vector4D& position);
59
62          void UpdateModelMatrix();
63
66          float Volume();
67
68      private:
69          float mRadius;
70
71          FAPhysics::RigidBody mRigidBody;
72          FAShapes::ThreeDimensionalShape mShape;
73          FAMath::Vector4D mOffset;
74      };
75  }
```

4.8 PolyhedralMassProperties.h

```
1  #pragma once
```

```
2
3 #include "FATriangle.h"
4 #include <vector>
5
6 namespace FAPhysics
7 {
10     void SubExpressions(double w0, double w1, double w2, double& f1, double& f2, double& f3, double& g0,
      double& g1, double& g2);
11
18     void ComputeMassProperties(const std::vector<FAShapes::Triangle>& triangles, double& mass,
      FAMath::Vector4D& cm,
19         FAMath::Matrix3x3& bodyInertia);
20 }
```

Index

AddForce
    FAPhysics::RigidBody, 6
AddTorque
    FAPhysics::RigidBody, 6

C:/Users/Work/Desktop/First Game Engine/First-Game-Engine/FA Physics Engine/Header Files/FAForceFunctions.h, 29
C:/Users/Work/Desktop/First Game Engine/First-Game-Engine/FA Physics Engine/Header Files/FARigidBody.h, 29
C:/Users/Work/Desktop/First Game Engine/First-Game-Engine/FA Physics Engine/Header Files/FARigidBox.h, 30
C:/Users/Work/Desktop/First Game Engine/First-Game-Engine/FA Physics Engine/Header Files/FARigidCone.h, 31
C:/Users/Work/Desktop/First Game Engine/First-Game-Engine/FA Physics Engine/Header Files/FARigidCylinder.h, 32
C:/Users/Work/Desktop/First Game Engine/First-Game-Engine/FA Physics Engine/Header Files/FARigidPyramid.h, 32
C:/Users/Work/Desktop/First Game Engine/First-Game-Engine/FA Physics Engine/Header Files/FARigidSphere.h, 33
C:/Users/Work/Desktop/First Game Engine/First-Game-Engine/FA Physics Engine/Header Files/PolyhedralMassProperties.h, 33

FAPhysics::RigidBody, 5
    AddForce, 6
    AddTorque, 6
    GetAngularMomentum, 6
    GetAngularVelocity, 6
    GetBodyInertiaTensor, 6
    GetCenterOfMass, 6
    GetInverseBodyInertiaTensor, 7
    GetInverseMass, 7
    GetLinearMomentum, 7
    GetLinearVelocity, 7
    GetMass, 7
    GetNetForce, 7
    GetNetTorque, 7
    GetOrientation, 8
    InitializeRigidBody, 8
    Integrate, 8
    ResetForce, 8
    ResetTorque, 8
    RigidBody, 5
    SetAngularMomentum, 8
    SetAngularVelocity, 9
    SetBodyInertiaTensor, 9
    SetCenterOfMass, 9
    SetLinearMomentum, 9
    SetLinearVelocity, 9
    SetMass, 9
    SetOrientation, 10
FAPhysicsShapes::RigidBox, 10
    GetDepth, 11
    GetHeight, 11
    GetRigidBody, 11
    GetShape, 12
    GetWidth, 12
    InitializeRigidBox, 12
    RigidBox, 11
    SetDepth, 13
    SetHeight, 13
    SetPosition, 13
    SetWidth, 13
    UpdateModelMatrix, 13
    Volume, 13
FAPhysicsShapes::RigidCone, 14
    GetHeight, 15
    GetRadius, 15
    GetRigidBody, 15
    GetShape, 15
    InitializeRigidCone, 16
    RigidCone, 14
    SetHeight, 16
    SetPosition, 16
    SetRadius, 16
    UpdateModelMatrix, 17
    Volume, 17
FAPhysicsShapes::RigidCylinder, 17
    GetHeight, 18
    GetRadius, 18
    GetRigidBody, 18
    GetShape, 19
    InitializeRigidCylinder, 19
    RigidCylinder, 18
    SetHeight, 20
    SetPosition, 20
    SetRadius, 20
    UpdateModelMatrix, 20
    Volume, 20
FAPhysicsShapes::RigidPyramid, 21
    GetDepth, 22
    GetHeight, 22

GetRigidBody, 22
GetShape, 22
GetWidth, 23
InitializeRigidPyramid, 23
RigidPyramid, 21
SetDepth, 23
SetHeight, 24
SetPosition, 24
SetWidth, 24
UpdateModelMatrix, 24
Volume, 24
FAPhysicsShapes::RigidSphere, 25
GetRadius, 25
GetRigidBody, 25, 26
GetShape, 26
InitializeRigidSphere, 26
RigidSphere, 25
SetPosition, 27
SetRadius, 27
UpdateModelMatrix, 27
Volume, 27

GetAngularMomentum
FAPhysics::RigidBody, 6
GetAngularVelocity
FAPhysics::RigidBody, 6
GetBodyInertiaTensor
FAPhysics::RigidBody, 6
GetCenterOfMass
FAPhysics::RigidBody, 6
GetDepth
FAPhysicsShapes::RigidBox, 11
FAPhysicsShapes::RigidPyramid, 22
GetHeight
FAPhysicsShapes::RigidBox, 11
FAPhysicsShapes::RigidCone, 15
FAPhysicsShapes::RigidCylinder, 18
FAPhysicsShapes::RigidPyramid, 22
GetInverseBodyInertiaTensor
FAPhysics::RigidBody, 7
GetInverseMass
FAPhysics::RigidBody, 7
GetLinearMomentum
FAPhysics::RigidBody, 7
GetLinearVelocity
FAPhysics::RigidBody, 7
GetMass
FAPhysics::RigidBody, 7
GetNetForce
FAPhysics::RigidBody, 7
GetNetTorque
FAPhysics::RigidBody, 7
GetOrientation
FAPhysics::RigidBody, 8
GetRadius
FAPhysicsShapes::RigidCone, 15
FAPhysicsShapes::RigidCylinder, 18
FAPhysicsShapes::RigidSphere, 25
GetRigidBody
FAPhysicsShapes::RigidBox, 11
FAPhysicsShapes::RigidCone, 15
FAPhysicsShapes::RigidCylinder, 18
FAPhysicsShapes::RigidPyramid, 22
FAPhysicsShapes::RigidSphere, 25, 26
GetShape
FAPhysicsShapes::RigidBox, 12
FAPhysicsShapes::RigidCone, 15
FAPhysicsShapes::RigidCylinder, 19
FAPhysicsShapes::RigidPyramid, 22
FAPhysicsShapes::RigidSphere, 26
GetWidth
FAPhysicsShapes::RigidBox, 12
FAPhysicsShapes::RigidPyramid, 23

InitializeRigidBody
FAPhysics::RigidBody, 8
InitializeRigidBox
FAPhysicsShapes::RigidBox, 12
InitializeRigidCone
FAPhysicsShapes::RigidCone, 16
InitializeRigidCylinder
FAPhysicsShapes::RigidCylinder, 19
InitializeRigidPyramid
FAPhysicsShapes::RigidPyramid, 23
InitializeRigidSphere
FAPhysicsShapes::RigidSphere, 26
Integrate
FAPhysics::RigidBody, 8

ResetForce
FAPhysics::RigidBody, 8
ResetTorque
FAPhysics::RigidBody, 8
RigidBody
FAPhysics::RigidBody, 5
RigidBox
FAPhysicsShapes::RigidBox, 11
RigidCone
FAPhysicsShapes::RigidCone, 14
RigidCylinder
FAPhysicsShapes::RigidCylinder, 18
RigidPyramid
FAPhysicsShapes::RigidPyramid, 21
RigidSphere
FAPhysicsShapes::RigidSphere, 25

SetAngularMomentum
FAPhysics::RigidBody, 8
SetAngularVelocity
FAPhysics::RigidBody, 9
SetBodyInertiaTensor
FAPhysics::RigidBody, 9
SetCenterOfMass
FAPhysics::RigidBody, 9
SetDepth
FAPhysicsShapes::RigidBox, 13
FAPhysicsShapes::RigidPyramid, 23
SetHeight

FAPhysicsShapes::RigidBox, 13
FAPhysicsShapes::RigidCone, 16
FAPhysicsShapes::RigidCylinder, 20
FAPhysicsShapes::RigidPyramid, 24
SetLinearMomentum
FAPhysics::RigidBody, 9
SetLinearVelocity
FAPhysics::RigidBody, 9
SetMass
FAPhysics::RigidBody, 9
SetOrientation
FAPhysics::RigidBody, 10
SetPosition
FAPhysicsShapes::RigidBox, 13
FAPhysicsShapes::RigidCone, 16
FAPhysicsShapes::RigidCylinder, 20
FAPhysicsShapes::RigidPyramid, 24
FAPhysicsShapes::RigidSphere, 27
SetRadius
FAPhysicsShapes::RigidCone, 16
FAPhysicsShapes::RigidCylinder, 20
FAPhysicsShapes::RigidSphere, 27
SetWidth
FAPhysicsShapes::RigidBox, 13
FAPhysicsShapes::RigidPyramid, 24

UpdateModelMatrix
FAPhysicsShapes::RigidBox, 13
FAPhysicsShapes::RigidCone, 17
FAPhysicsShapes::RigidCylinder, 20
FAPhysicsShapes::RigidPyramid, 24
FAPhysicsShapes::RigidSphere, 27

Volume
FAPhysicsShapes::RigidBox, 13
FAPhysicsShapes::RigidCone, 17
FAPhysicsShapes::RigidCylinder, 20
FAPhysicsShapes::RigidPyramid, 24
FAPhysicsShapes::RigidSphere, 27