3D Physics API

Groups the 3D Physics classes. More...

Classes

class	Quantum.Physics3D.PhysicsEngine3D.Api 3D Physics API More...
struct	Quantum.CharacterController3D
struct	Quantum.CollisionInfo3D Info about a collision between two 3D physics colliders. More...
struct	Quantum.ExitInfo3D Info about two entities that were colliding in the 3D Physics. More...
struct	Quantum.Physics3D.Hit3D Information returned from a valid hit of a physics query. More...
interface	Quantum.ISignalOnCollision3D Interface for receiving callbacks once per frame while two non-trigger 3D colliders are touching. More...
interface	Quantum.ISignalOnCollisionEnter3D Interface for receiving callbacks once two non-trigger 3D colliders start touching. More...
interface	Quantum.ISignalOnCollisionExit3D Interface for receiving callbacks once two non-trigger 3D colliders stop touching. More...
interface	Quantum.ISignalOnTrigger3D Interface for receiving callbacks once per frame while a non-trigger and a trigger 3D colliders are touching. More...
interface	Quantum.ISignalOnTriggerEnter3D Interface for receiving callbacks once a non-trigger and a trigger 3D colliders start touching. More...
interface	Quantum.ISignalOnTriggerExit3D Interface for receiving callbacks once a non-trigger and a trigger 3D colliders stop touching. More...
struct	Quantum.Shape3D Defines a 3D shape with Type and data disposed in a union-like structure. All shapes have a UserTag, BroadRadius and Centroid. All non-compound shapes have a LocalTransform and their Centroid always match their local transform position. More...
struct	Quantum.TriggerInfo3D Info about a collision between a trigger and a non-trigger 3D physics colliders. More...

Functions

int	Quantum.Physics3D.PhysicsEngine3D.Api.AddLinecastQuery (FPVector3 start, FPVector3 end, bool firstHitOnly=false, int layerMask=-1, QueryOptions options=QueryOptions.HitAll) Adds a broad-phase line cast query to the 3D physics scene. In order to be resolved, must be added from a system that runs prior to the Physics System. More...
int	Quantum.Physics3D.PhysicsEngine3D.Api.AddOverlapShapeQuery (FPVector3 position, FPQuaternion rotation, Shape3D shape, int layerMask=-1, QueryOptions options=QueryOptions.HitAll) Adds a broad-phase shape overlap query to the 3D physics scene. In order to be resolved, must be added from a system that runs prior to the Physics System. More...
int	Quantum.Physics3D.PhysicsEngine3D.Api.AddOverlapShapeQuery (Transform3D transform, Shape3D shape, int layerMask=-1, QueryOptions options=QueryOptions.HitAll) Adds a broad-phase shape overlap query to the 3D physics scene. In order to be resolved, must be added from a system that runs prior to the Physics System. More...
int	Quantum.Physics3D.PhysicsEngine3D.Api.AddRaycastQuery (FPVector3 origin, FPVector3 direction, FP distance, bool firstHitOnly=false, int layerMask=-1, QueryOptions options=QueryOptions.HitAll) Adds a broad-phase ray cast query to the 3D physics scene. In order to be resolved, must be added from a system that runs prior to the Physics System. More...

int	Quantum.Physics3D.PhysicsEngine3D.Api.AddShapeCastQuery (FPVector3 start, FPQuaternion rotation, Shape3D *shape, FPVector3 translation, bool firstHitOnly=false, int layerMask=-1, QueryOptions options=QueryOptions.HitAll) Adds a broad-phase shape cast query to the 3D physics scene. In order to be resolved, must be added from a system that runs prior to the Physics System. More...
HitCollection3D *	Quantum.Physics3D.PhysicsEngine3D.Api.AllocatePersistentHitCollection3D (int defaultCapacity=64) Allocates a persistent HitCollection3D on the heap with an initial capacity. To free an allocated persistent collection of 3D hits, use FreePersistentHitCollection3D. More...
void	Quantum.Physics3D.PhysicsEngine3D.Api.FreePersistentHitCollection3D (HitCollection3D *collection) Free a persistent HitCollection3D previously allocated on the heap. To allocate a persistent collection of 3D hits, use AllocatePersistentHitCollection3D. More...
bool	Quantum.Physics3D.PhysicsEngine3D.Api.GetAllQueriesHits (out HitCollection3D *queriesHits, out int queriesCount) Gets all the results for all the broad-phase queries added to the 3D physics scene this frame. More...
HitCollection3D	Quantum.Physics3D.PhysicsEngine3D.Api.GetQueryHits (int index) Gets the results of a broad-phase query added to the 3D physics scene. More...
Hit3D?	Quantum.Physics3D.PhysicsEngine3D.Api.Linecast (FPVector3 start, FPVector3 end, int layerMask=-1, QueryOptions options=QueryOptions.HitAll) Queries the 3D physics scene with a line cast and returns the closest hit to the line start, if any. More...
void	Quantum.Physics3D.PhysicsEngine3D.Api.Linecast (HitCollection3D *collection, FPVector3 start, FPVector3 end, int layerMask=-1, QueryOptions options=QueryOptions.HitAll) Queries the 3D physics scene with a line cast and adds the closest hit to the line start, if any, to a persistent collection of hits. More...
HitCollection3D	Quantum.Physics3D.PhysicsEngine3D.Api.LinecastAll (FPVector3 start, FPVector3 end, int layerMask=-1, QueryOptions options=QueryOptions.HitAll) Queries the 3D physics scene with a line cast, returning all hits. More...
void	Quantum.Physics3D.PhysicsEngine3D.Api.LinecastAll (HitCollection3D *collection, FPVector3 start, FPVector3 end, int layerMask=-1, QueryOptions options=QueryOptions.HitAll) Queries the 3D physics scene with a line cast, returning all hits. More...
void	Quantum.ISignalOnCollision3D.OnCollision3D (Frame f, CollisionInfo3D info) Called once per frame while two non-trigger 3D colliders are touching. More...
void	Quantum.ISignalOnCollisionEnter3D.OnCollisionEnter3D (Frame f, CollisionInfo3D info) Called once two non-trigger 3D colliders start touching. More...
void	Quantum.ISignalOnCollisionExit3D.OnCollisionExit3D (Frame f, ExitInfo3D info) Called once two non-trigger 3D colliders stop touching. More...
void	Quantum.ISignalOnTrigger3D.OnTrigger3D (Frame f, TriggerInfo3D info) Called once per frame while a non-trigger and a trigger 3D colliders are touching. More...
void	Quantum.ISignalOnTriggerEnter3D.OnTriggerEnter3D (Frame f, TriggerInfo3D info) Called once a non-trigger and a trigger 3D colliders start touching. More...
void	Quantum.ISignalOnTriggerExit3D.OnTriggerExit3D (Frame f, ExitInfo3D info) Called once a non-trigger and a trigger 3D colliders stop touching. More...
HitCollection3D	Quantum.Physics3D.PhysicsEngine3D.Api.OverlapShape (FPVector3 position, FPQuaternion rotation, Shape3D shape, int layerMask=-1, QueryOptions options=QueryOptions.HitAll) Queries the 3D physics scene with a shape overlap. More...
void	Quantum.Physics3D.PhysicsEngine3D.Api.OverlapShape (HitCollection3D *collection, FPVector3 position, FPQuaternion rotation, Shape3D shape, int layerMask=-1, QueryOptions options=QueryOptions.HitAll) Queries the 3D physics scene with a shape overlap. More...
void	Quantum.Physics3D.PhysicsEngine3D.Api.OverlapShape (HitCollection3D *collection, Transform3D transform, Shape3D shape, int layerMask=-1, QueryOptions options=QueryOptions.HitAll) Queries the 3D physics scene with a shape overlap. More...
HitCollection3D	Quantum.Physics3D.PhysicsEngine3D.Api.OverlapShape (Transform3D transform, Shape3D shape, int layerMask=-1, QueryOptions options=QueryOptions.HitAll) Queries the 3D physics scene with a shape overlap. More...
Hit3D?	Quantum.Physics3D.PhysicsEngine3D.Api.Raycast (FPVector3 origin, FPVector3 direction, FP distance, int layerMask=-1, QueryOptions options=QueryOptions.HitAll)

	Queries the 3D physics scene with a ray cast and returns the closest hit to the ray origin, if any. More...
void	Quantum.Physics3D.PhysicsEngine3D.Api.Raycast (HitCollection3D *collection, FPVector3 origin, FPVector3 direction, FP distance, int layerMask=-1, QueryOptions options=QueryOptions.HitAll) Queries the 3D physics scene with a ray cast and adds the closest hit to the ray origin, if any, to a persistent collection of hits. More...
HitCollection3D	Quantum.Physics3D.PhysicsEngine3D.Api.RaycastAll (FPVector3 origin, FPVector3 direction, FP distance, int layerMask=-1, QueryOptions options=QueryOptions.HitAll) Queries the 3D physics scene with a ray cast, returning all hits. More...
void	Quantum.Physics3D.PhysicsEngine3D.Api.RaycastAll (HitCollection3D *collection, FPVector3 origin, FPVector3 direction, FP distance, int layerMask=-1, QueryOptions options=QueryOptions.HitAll) Queries the 3D physics scene with a ray cast, adding all hits to a persistent collection. More...
void	Quantum.Physics3D.PhysicsEngine3D.Api.ResetMap () Resets 3D physics scene map asset guid. More...
void	Quantum.Physics3D.PhysicsEngine3D.Api.SetCallbacks (EntityRef entity, CallbackFlags flags) Sets which 3D physics collision callbacks will be called for the *entity* . More...
Hit3D?	Quantum.Physics3D.PhysicsEngine3D.Api.ShapeCast (FPVector3 start, FPQuaternion rotation, Shape3D *shape, FPVector3 translation, int layerMask=-1, QueryOptions options=QueryOptions.HitAll) Queries the 3D physics scene with a shape cast and returns the hit with the smallest Hit3D.CastDistanceNormalized, if any. More...
void	Quantum.Physics3D.PhysicsEngine3D.Api.ShapeCast (HitCollection3D *collection, FPVector3 start, FPQuaternion rotation, Shape3D *shape, FPVector3 translation, int layerMask=-1, QueryOptions options=QueryOptions.HitAll) Queries the 3D physics scene with a shape cast and adds the hit with the shortest Hit3D.CastDistanceNormalized, if any, to a persistent collection of hits. More...
HitCollection3D	Quantum.Physics3D.PhysicsEngine3D.Api.ShapeCastAll (FPVector3 start, FPQuaternion rotation, Shape3D *shape, FPVector3 translation, int layerMask=-1, QueryOptions options=QueryOptions.HitAll) Queries the 3D physics scene with a shape cast, returning all hits. More...
void	Quantum.Physics3D.PhysicsEngine3D.Api.ShapeCastAll (HitCollection3D *collection, FPVector3 start, FPQuaternion rotation, Shape3D *shape, FPVector3 translation, int layerMask=-1, QueryOptions options=QueryOptions.HitAll) Queries the 3D physics scene with a shape cast, adding all hits to a persistent collection. More...
bool	Quantum.Physics3D.PhysicsEngine3D.Api.TryGetQueryHits (int index, out HitCollection3D queryHits) Tries to get the results of a broad-phase query added to the 3D physics scene. More...

Properties

Map	Quantum.Physics2D.PhysicsEngine2D.Api.Map `[get]` Retrieves the map asset currently used on the physics scene. More...
Map	Quantum.Physics3D.PhysicsEngine3D.Api.Map `[get]` Retrieves the map asset currently used on the physics scene. More...

Detailed Description

Groups the 3D Physics classes.

Function Documentation

◆ SetCallbacks()

void Quantum.Physics3D.PhysicsEngine3D.Api.SetCallbacks (EntityRef entity,
CallbackFlags flags
) inline

Sets which 3D physics collision callbacks will be called for the *entity* .

The entity must have a PhysicsCollider3D component attached to be able to collide with other physics entries.

For receiving the callbacks, set the CallbackFlags and implement the corresponding signal on a system.

Parameters

entity An entity with a PhysicsCollider3D component attached.

flags The CallbackFlags of the desired collision callbacks.

```
frame.Physics3D.SetCallbacks(entity, CallbackFlags.OnDynamicCollision | CallbackFlags.OnStaticTriggerEnter);
```

◆ ResetMap()

void Quantum.Physics3D.PhysicsEngine3D.Api.ResetMap ( ) inline

Resets 3D physics scene map asset guid.

◆ AllocatePersistentHitCollection3D()

HitCollection3D*
Quantum.Physics3D.PhysicsEngine3D.Api.AllocatePersistentHitCollection3D (int defaultCapacity = 64) inline

Allocates a persistent HitCollection3D on the heap with an initial capacity. To free an allocated persistent collection of 3D hits, use FreePersistentHitCollection3D.

Parameters

defaultCapacity The initial Hit3D buffer capacity allocated for the hit collection.

Returns

A pointer to the allocated hit collection.

◆ FreePersistentHitCollection3D()

void Quantum.Physics3D.PhysicsEngine3D.Api.FreePersistentHitCollection3D (HitCollection3D * collection) inline

Free a persistent HitCollection3D previously allocated on the heap. To allocate a persistent collection of 3D hits, use AllocatePersistentHitCollection3D.

Parameters

collection A pointer to the persistent hit collection to be freed.

Exceptions

InvalidOperationException Thrown when the hit collection was not allocated as persistent.

◆ GetQueryHits()

HitCollection3D Quantum.Physics3D.PhysicsEngine3D.Api.GetQueryHits (int **index**) inline

Gets the results of a broad-phase query added to the 3D physics scene.

Parameters

index The broad-phase query index.

Returns

A HitCollection3D with the query hits.

◆ TryGetQueryHits()

bool Quantum.Physics3D.PhysicsEngine3D.Api.TryGetQueryHits (int **index**,
out HitCollection3D **queryHits**
) inline

Tries to get the results of a broad-phase query added to the 3D physics scene.

Parameters

index The broad-phase query index.

queryHits The HitCollection3D with the query hits. Default if the index is not valid.

Returns

True if the index is valid, false otherwise.

◆ GetAllQueriesHits()

bool Quantum.Physics3D.PhysicsEngine3D.Api.GetAllQueriesHits (out HitCollection3D * **queriesHits**,
out int **queriesCount**
) inline

Gets all the results for all the broad-phase queries added to the 3D physics scene this frame.

Parameters

queriesHits A buffer of HitCollection3D with *queriesCount* elements, one for each broad-phase query added.

queriesCount The number of elements in the *queriesHits* buffer, also matching the number on broad-phase queries added.

Returns

True if at least one broad-phase query has been added from a system that runs before the physics engine.

Example of how to iterate over added broad-phase queries results.

```
if (f.Physics3D.GetAllQueriesHits(out var queriesHits, out var queriesCount)) {
  for (var i = 0; i < queriesCount; i++) {
    var queryHits = queriesHits + i;
    Log.Info($"Preemptive query {i} has {queryHits->Count} hits.");
  }
}
```

◆ AddRaycastQuery()

```
int
Quantum.Physics3D.PhysicsEngine3D.Api.AddRaycastQuery ( FPVector3    origin,
                                                        FPVector3    direction,
                                                        FP           distance,
                                                        bool         firstHitOnly = false,
                                                        int          layerMask = -1,
                                                        QueryOptions options = QueryOptions.HitAll
                                                      )
```

inline

Adds a broad-phase ray cast query to the 3D physics scene. In order to be resolved, must be added from a system that runs prior to the Physics System.

Parameters

origin The ray origin point, in a 3D world space.

direction The direction of the ray, not normalized internally.

distance The ray distance from the origin point on the specified direction.

firstHitOnly If the query results should return only the closest hit to the ray *origin* or all hits (default).

layerMask A mask that specifies which layers will be checked against. By default, all layers are considered. To create a layer mask, see LayerInfo.GetLayerMask(string) and overloads on the frame.Layers property.

options The QueryOptions used by the query. By default, all types of colliders are checked against and detailed info (hit point, penetration and normal) is not computed.

Returns

The index of the injected query, to be used when retrieving the results with GetQueryHits.

◆ AddLinecastQuery()

```
int
Quantum.Physics3D.PhysicsEngine3D.Api.AddLinecastQuery ( FPVector3    start,
                                                         FPVector3    end,
                                                         bool         firstHitOnly = false,
                                                         int          layerMask = -1,
                                                         QueryOptions options = QueryOptions.HitAll
                                                       )
```

inline

Adds a broad-phase line cast query to the 3D physics scene. In order to be resolved, must be added from a system that runs prior to the Physics System.

Parameters

start The line start point, in a 3D world space.

end The line end point, in a 3D world space.

firstHitOnly If the query results should return only the closest hit to the line *start* or all hits (default).

layerMask A mask that specifies which layers will be checked against. By default, all layers are considered. To create a layer mask, see LayerInfo.GetLayerMask(string) and overloads on the frame.Layers property.

options The QueryOptions used by the query. By default, all types of colliders are checked against and detailed info (hit point, penetration and normal) is not computed.

Returns

The index of the injected query, to be used when retrieving the results with GetQueryHits.

◆ Raycast() [1/2]

```
Hit3D? Quantum.Physics3D.PhysicsEngine3D.Api.Raycast ( FPVector3     origin,
                                                       FPVector3     direction,
                                                       FP            distance,
                                                       int           layerMask = -1,
                                                       QueryOptions  options = QueryOptions.HitAll
                                                     )
```

inline

Queries the 3D physics scene with a ray cast and returns the closest hit to the ray origin, if any.

Parameters

origin The ray origin point, in a 3D world space.

direction The direction of the ray, not normalized internally.

distance The ray distance from the origin point on the specified direction.

layerMask A mask that specifies which layers will be checked against. By default, all layers are considered. To create a layer mask, see LayerInfo.GetLayerMask(string) and overloads on the frame.Layers property.

options The QueryOptions used by the query. By default, all types of colliders are checked against and detailed info (hit point, penetration and normal) is not computed.

Returns

A nullable Hit3D with the closest hit to the ray origin, if any.

◆ Raycast() [2/2]

```
void Quantum.Physics3D.PhysicsEngine3D.Api.Raycast ( HitCollection3D *  collection,
                                                     FPVector3          origin,
                                                     FPVector3          direction,
                                                     FP                 distance,
                                                     int                layerMask = -1,
                                                     QueryOptions       options = QueryOptions.HitAll
                                                   )
```

inline

Queries the 3D physics scene with a ray cast and adds the closest hit to the ray origin, if any, to a persistent collection of hits.

Parameters

collection A previously allocated persistent HitCollection3D, to which the query hits will be added. For allocating a persistent HitCollection, see AllocatePersistentHitCollection3D.

origin The ray origin point, in a 3D world space.

direction The direction of the ray, not normalized internally.

distance The ray distance from the origin point on the specified direction.

layerMask A mask that specifies which layers will be checked against. By default, all layers are considered. To create a layer mask, see LayerInfo.GetLayerMask(string) and overloads on the frame.Layers property.

options The QueryOptions used by the query. By default, all types of colliders are checked against and detailed info (hit point, penetration and normal) is not computed.

◆ Linecast() [1/2]

```
Hit3D? Quantum.Physics3D.PhysicsEngine3D.Api.Linecast ( FPVector3     start,
                                                        FPVector3     end,
                                                        int           layerMask = -1,
                                                        QueryOptions  options = QueryOptions.HitAll
                                                      )
```

inline

Queries the 3D physics scene with a line cast and returns the closest hit to the line start, if any.

Parameters

- **start** The line start point, in a 3D world space.
- **end** The line end point, in a 3D world space.
- **layerMask** A mask that specifies which layers will be checked against. By default, all layers are considered. To create a layer mask, see LayerInfo.GetLayerMask(string) and overloads on the frame.Layers property.
- **options** The QueryOptions used by the query. By default, all types of colliders are checked against and detailed info (hit point, penetration and normal) is not computed.

Returns

A nullable Hit3D with the closest hit to the line start, if any.

◆ Linecast() [2/2]

```
void Quantum.Physics3D.PhysicsEngine3D.Api.Linecast ( HitCollection3D *  collection,
                                                      FPVector3          start,
                                                      FPVector3          end,
                                                      int                layerMask = -1,
                                                      QueryOptions       options = QueryOptions.HitAll
                                                    )
```

inline

Queries the 3D physics scene with a line cast and adds the closest hit to the line start, if any, to a persistent collection of hits.

Parameters

- **collection** A previously allocated persistent HitCollection3D, to which the query hits will be added. For allocating a persistent HitCollection, see AllocatePersistentHitCollection3D.
- **start** The line start point, in a 3D world space.
- **end** The line end point, in a 3D world space.
- **layerMask** A mask that specifies which layers will be checked against. By default, all layers are considered. To create a layer mask, see LayerInfo.GetLayerMask(string) and overloads on the frame.Layers property.
- **options** The QueryOptions used by the query. By default, all types of colliders are checked against and detailed info (hit point, penetration and normal) is not computed.

◆ RaycastAll() [1/2]

```
HitCollection3D
Quantum.Physics3D.PhysicsEngine3D.Api.RaycastAll   ( FPVector3     origin,
                                                     FPVector3     direction,
                                                     FP            distance,
                                                     int           layerMask = -1,
                                                     QueryOptions  options = QueryOptions.HitAll
                                                   )
```

inline

Queries the 3D physics scene with a ray cast, returning all hits.

Parameters

origin The ray origin point, in a 3D world space.

direction The direction of the ray, not normalized internally.

distance The ray distance from the origin point on the specified direction.

layerMask A mask that specifies which layers will be checked against. By default, all layers are considered. To create a layer mask, see LayerInfo.GetLayerMask(string) and overloads on the frame.Layers property.

options The QueryOptions used by the query. By default, all types of colliders are checked against and detailed info (hit point, penetration and normal) is not computed.

Returns

A temporary HitCollection3D with the query hits, automatically freed when the current frame simulation ends.

◆ RaycastAll() [2/2]

```
void
Quantum.Physics3D.PhysicsEngine3D.Api.RaycastAll   ( HitCollection3D *  collection,
                                                     FPVector3          origin,
                                                     FPVector3          direction,
                                                     FP                 distance,
                                                     int                layerMask = -1,
                                                     QueryOptions       options = QueryOptions.HitAll
                                                   )
```

inline

Queries the 3D physics scene with a ray cast, adding all hits to a persistent collection.

Parameters

collection A previously allocated persistent HitCollection3D, to which the query hits will be added. For allocating a persistent HitCollection, see AllocatePersistentHitCollection3D.

origin The ray origin point, in a 3D world space.

direction The direction of the ray, not normalized internally.

distance The ray distance from the origin point on the specified direction.

layerMask A mask that specifies which layers will be checked against. By default, all layers are considered. To create a layer mask, see LayerInfo.GetLayerMask(string) and overloads on the frame.Layers property.

options The QueryOptions used by the query. By default, all types of colliders are checked against and detailed info (hit point, penetration and normal) is not computed.

◆ LinecastAll() [1/2]

```
HitCollection3D
Quantum.Physics3D.PhysicsEngine3D.Api.LinecastAll ( FPVector3     start,
                                                    FPVector3     end,
                                                    int           layerMask = -1,
                                                    QueryOptions  options = QueryOptions.HitAll
                                                  )
```

inline

Queries the 3D physics scene with a line cast, returning all hits.

Parameters

start The line start point, in a 3D world space.

end The line end point, in a 3D world space.

layerMask A mask that specifies which layers will be checked against. By default, all layers are considered. To create a layer mask, see LayerInfo.GetLayerMask(string) and overloads on the frame.Layers property.

options The QueryOptions used by the query. By default, all types of colliders are checked against and detailed info (hit point, penetration and normal) is not computed.

Returns

A temporary HitCollection3D with the query hits, automatically freed when the current frame simulation ends.

◆ LinecastAll() [2/2]

```
void
Quantum.Physics3D.PhysicsEngine3D.Api.LinecastAll ( HitCollection3D *  collection,
                                                    FPVector3          start,
                                                    FPVector3          end,
                                                    int                layerMask = -1,
                                                    QueryOptions       options = QueryOptions.HitAll
                                                  )
```

inline

Queries the 3D physics scene with a line cast, returning all hits.

Parameters

collection A previously allocated persistent HitCollection3D, to which the query hits will be added. For allocating a persistent HitCollection, see AllocatePersistentHitCollection3D.

start The line start point, in a 3D world space.

end The line end point, in a 3D world space.

layerMask A mask that specifies which layers will be checked against. By default, all layers are considered. To create a layer mask, see LayerInfo.GetLayerMask(string) and overloads on the frame.Layers property.

options The QueryOptions used by the query. By default, all types of colliders are checked against and detailed info (hit point, penetration and normal) is not computed.

◆ AddShapeCastQuery()

```
int
Quantum.Physics3D.PhysicsEngine3D.Api.AddShapeCastQuery ( FPVector3      start,
                                                          FPQuaternion   rotation,
                                                          Shape3D *      shape,
                                                          FPVector3      translation,
                                                          bool           firstHitOnly = false,
                                                          int            layerMask = -1,
                                                          QueryOptions   options = QueryOptions.HitAll
                                                        )
```

inline

Adds a broad-phase shape cast query to the 3D physics scene. In order to be resolved, must be added from a system that runs prior to the Physics System.

Parameters

start The cast start point, in a 3D world space.

rotation The quaternion rotation of the shape.

shape The 3D shape to be casted.

translation The cast direction and distance, from the cast start point.

firstHitOnly If the query results should return only the hit with the smallest Hit3D.CastDistanceNormalized or all hits (default).

layerMask A mask that specifies which layers will be checked against. By default, all layers are considered. To create a layer mask, see LayerInfo.GetLayerMask(string) and overloads on the frame.Layers property.

options The QueryOptions used by the query. By default, all types of colliders are checked against and detailed info (hit point, penetration and normal) is not computed.

Returns

The index of the injected query, to be used when retrieving the results with GetQueryHits.

◆ ShapeCast() [1/2]

```
Hit3D? Quantum.Physics3D.PhysicsEngine3D.Api.ShapeCast ( FPVector3      start,
                                                         FPQuaternion   rotation,
                                                         Shape3D *      shape,
                                                         FPVector3      translation,
                                                         int            layerMask = -1,
                                                         QueryOptions   options = QueryOptions.HitAll
                                                       )
```

inline

Queries the 3D physics scene with a shape cast and returns the hit with the smallest Hit3D.CastDistanceNormalized, if any.

Parameters

start The cast start, in a 3D world space.

rotation The quaternion rotation of the shape.

shape The 3D shape to be casted.

translation The cast direction and distance, from the cast start point.

layerMask A mask that specifies which layers will be checked against. By default, all layers are considered. To create a layer mask, see LayerInfo.GetLayerMask(string) and overloads on the frame.Layers property.

options The QueryOptions used by the query. By default, all types of colliders are checked against and detailed info (hit point, penetration and normal) is not computed.

Returns

A nullable Hit3D with the smallest Hit3D.CastDistanceNormalized, if any.

◆ ShapeCast() [2/2]

```
void
Quantum.Physics3D.PhysicsEngine3D.Api.ShapeCast   ( HitCollection3D *  collection,
                                                    FPVector3          start,
                                                    FPQuaternion       rotation,
                                                    Shape3D *          shape,
                                                    FPVector3          translation,
                                                    int                layerMask = -1,
                                                    QueryOptions       options = QueryOptions.HitAll
                                                  )
```

inline

Queries the 3D physics scene with a shape cast and adds the hit with the shortest Hit3D.CastDistanceNormalized, if any, to a persistent collection of hits.

Parameters

collection A previously allocated persistent HitCollection3D, to which the query hits will be added. For allocating a persistent HitCollection, see AllocatePersistentHitCollection3D.

start The cast start, in a 3D world space.

rotation The quaternion rotation of the shape.

shape The 3D shape to be casted.

translation The cast direction and distance, from the cast start point.

layerMask A mask that specifies which layers will be checked against. By default, all layers are considered. To create a layer mask, see LayerInfo.GetLayerMask(string) and overloads on the frame.Layers property.

options The QueryOptions used by the query. By default, all types of colliders are checked against and detailed info (hit point, penetration and normal) is not computed.

◆ ShapeCastAll() [1/2]

```
HitCollection3D
Quantum.Physics3D.PhysicsEngine3D.Api.ShapeCastAll   ( FPVector3     start,
                                                       FPQuaternion  rotation,
                                                       Shape3D *     shape,
                                                       FPVector3     translation,
                                                       int           layerMask = -1,
                                                       QueryOptions  options = QueryOptions.HitAll
                                                     )
```

inline

Queries the 3D physics scene with a shape cast, returning all hits.

Parameters

start The cast start, in a 3D world space.

rotation The quaternion rotation of the shape.

shape The 3D shape to be casted.

translation The cast direction and distance, from the cast start point.

layerMask A mask that specifies which layers will be checked against. By default, all layers are considered. To create a layer mask, see LayerInfo.GetLayerMask(string) and overloads on the frame.Layers property.

options The QueryOptions used by the query. By default, all types of colliders are checked against and detailed info (hit point, penetration and normal) is not computed.

Returns

A temporary HitCollection3D with the query hits, automatically freed when the current frame simulation ends.

◆ ShapeCastAll() [2/2]

```
void
Quantum.Physics3D.PhysicsEngine3D.Api.ShapeCastAll ( HitCollection3D *  collection,
                                                     FPVector3          start,
                                                     FPQuaternion       rotation,
                                                     Shape3D *          shape,
                                                     FPVector3          translation,
                                                     int                layerMask = −1,
                                                     QueryOptions       options = QueryOptions.HitAll
                                                   )
```

inline

Queries the 3D physics scene with a shape cast, adding all hits to a persistent collection.

Parameters

collection	A previously allocated persistent HitCollection3D, to which the query hits will be added. For allocating a persistent HitCollection, see AllocatePersistentHitCollection3D.
start	The cast start, in a 3D world space.
rotation	The quaternion rotation of the shape.
shape	The 3D shape to be casted.
translation	The cast direction and distance, from the cast start point.
layerMask	A mask that specifies which layers will be checked against. By default, all layers are considered. To create a layer mask, see LayerInfo.GetLayerMask(string) and overloads on the frame.Layers property.
options	The QueryOptions used by the query. By default, all types of colliders are checked against and detailed info (hit point, penetration and normal) is not computed.

◆ AddOverlapShapeQuery() [1/2]

```
int
Quantum.Physics3D.PhysicsEngine3D.Api.AddOverlapShapeQuery ( Transform3D   transform,
                                                             Shape3D       shape,
                                                             int           layerMask = −1,
                                                             QueryOptions  options = QueryOptions.HitAll
                                                           )
```

inline

Adds a broad-phase shape overlap query to the 3D physics scene. In order to be resolved, must be added from a system that runs prior to the Physics System.

Parameters

transform	A 3D transform component, with Position and Rotation info of the shape overlap.
shape	The 3D shape to be overlapped.
layerMask	A mask that specifies which layers will be checked against. By default, all layers are considered. To create a layer mask, see LayerInfo.GetLayerMask(string) and overloads on the frame.Layers property.
options	The QueryOptions used by the query. By default, all types of colliders are checked against and detailed info (hit point, penetration and normal) is not computed.

Returns

The index of the injected query, to be used when retrieving the results with GetQueryHits.

◆ AddOverlapShapeQuery() [2/2]

```
int
Quantum.Physics3D.PhysicsEngine3D.Api.AddOverlapShapeQuery ( FPVector3     position,
                                                             FPQuaternion  rotation,
                                                             Shape3D       shape,
                                                             int           layerMask = -1,
                                                             QueryOptions  options = QueryOptions.HitAll
                                                           )
```

inline

Adds a broad-phase shape overlap query to the 3D physics scene. In order to be resolved, must be added from a system that runs prior to the Physics System.

Parameters

- **position** The position in which the shape is overlapped.
- **rotation** The quaternion rotation of the shape.
- **shape** The 3D shape to be overlapped.
- **layerMask** A mask that specifies which layers will be checked against. By default, all layers are considered. To create a layer mask, see LayerInfo.GetLayerMask(string) and overloads on the frame.Layers property.
- **options** The QueryOptions used by the query. By default, all types of colliders are checked against and detailed info (hit point, penetration and normal) is not computed.

Returns

The index of the injected query, to be used when retrieving the results with GetQueryHits.

◆ OverlapShape() [1/4]

```
void
Quantum.Physics3D.PhysicsEngine3D.Api.OverlapShape ( HitCollection3D *  collection,
                                                     Transform3D        transform,
                                                     Shape3D            shape,
                                                     int                layerMask = -1,
                                                     QueryOptions       options = QueryOptions.HitAll
                                                   )
```

inline

Queries the 3D physics scene with a shape overlap.

Parameters

- **collection** A previously allocated persistent HitCollection3D, to which the query hits will be added. For allocating a persistent HitCollection, see AllocatePersistentHitCollection3D.
- **transform** A 3D transform component, with Position and Rotation info of the shape overlap.
- **shape** The 3D shape to be overlapped.
- **layerMask** A mask that specifies which layers will be checked against. By default, all layers are considered. To create a layer mask, see LayerInfo.GetLayerMask(string) and overloads on the frame.Layers property.
- **options** The QueryOptions used by the query. By default, all types of colliders are checked against and detailed info (hit point, penetration and normal) is not computed.

Exceptions

- **InvalidOperationException** Thrown if the hit collection is not persistent.

◆ OverlapShape() [2/4]

```
void
Quantum.Physics3D.PhysicsEngine3D.Api.OverlapShape ( HitCollection3D * collection,
                                                     FPVector3         position,
                                                     FPQuaternion      rotation,
                                                     Shape3D           shape,
                                                     int               layerMask = -1,
                                                     QueryOptions      options = QueryOptions.HitAll
                                                   )
```

inline

Queries the 3D physics scene with a shape overlap.

Parameters

collection A previously allocated persistent HitCollection3D, to which the query hits will be added. For allocating a persistent HitCollection, see AllocatePersistentHitCollection3D.

position The position in which the shape is overlapped.

rotation The quaternion rotation of the shape.

shape The 3D shape to be overlapped.

layerMask A mask that specifies which layers will be checked against. By default, all layers are considered. To create a layer mask, see LayerInfo.GetLayerMask(string) and overloads on the frame.Layers property.

options The QueryOptions used by the query. By default, all types of colliders are checked against and detailed info (hit point, penetration and normal) is not computed.

Exceptions

InvalidOperationException Thrown if the hit collection is not persistent.

◆ OverlapShape() [3/4]

```
HitCollection3D
Quantum.Physics3D.PhysicsEngine3D.Api.OverlapShape ( Transform3D  transform,
                                                     Shape3D      shape,
                                                     int          layerMask = -1,
                                                     QueryOptions options = QueryOptions.HitAll
                                                   )
```

inline

Queries the 3D physics scene with a shape overlap.

Parameters

transform A 3D transform component, with Position and Rotation info of the shape overlap.

shape The 3D shape to be overlapped.

layerMask A mask that specifies which layers will be checked against. By default, all layers are considered. To create a layer mask, see LayerInfo.GetLayerMask(string) and overloads on the frame.Layers property.

options The QueryOptions used by the query. By default, all types of colliders are checked against and detailed info (hit point, penetration and normal) is not computed.

Returns

A temporary HitCollection3D with the query hits, automatically freed when the current frame simulation ends.

◆ OverlapShape() [4/4]

```
HitCollection3D
Quantum.Physics3D.PhysicsEngine3D.Api.OverlapShape  ( FPVector3     position,
                                                      FPQuaternion  rotation,
                                                      Shape3D       shape,
                                                      int           layerMask = -1,
                                                      QueryOptions  options = QueryOptions.HitAll
                                                    )
```

inline

Queries the 3D physics scene with a shape overlap.

Parameters

- **position** The 3D position in which the shape is overlapped.
- **rotation** The quaternion rotation of the shape.
- **shape** The 3D shape to be overlapped.
- **layerMask** A mask that specifies which layers will be checked against. By default, all layers are considered. To create a layer mask, see LayerInfo.GetLayerMask(string) and overloads on the frame.Layers property.
- **options** The QueryOptions used by the query. By default, all types of colliders are checked against and detailed info (hit point, penetration and normal) is not computed.

Returns

A temporary HitCollection3D with the query hits, automatically freed when the current frame simulation ends.

◆ OnCollision3D()

```
void Quantum.ISignalOnCollision3D.OnCollision3D ( Frame            f,
                                                  CollisionInfo3D  info
                                                )
```

Called once per frame while two non-trigger 3D colliders are touching.

Parameters

- **f** The frame in which the collision happened.
- **info** The CollisionInfo3D with data about the collision.

◆ OnCollisionEnter3D()

```
void Quantum.ISignalOnCollisionEnter3D.OnCollisionEnter3D ( Frame            f,
                                                            CollisionInfo3D  info
                                                          )
```

Called once two non-trigger 3D colliders start touching.

Parameters

- **f** The frame in which the collision happened.
- **info** The CollisionInfo3D with data about the collision.

◆ OnCollisionExit3D()

void Quantum.ISignalOnCollisionExit3D.OnCollisionExit3D (Frame f,
ExitInfo3D info
)

Called once two non-trigger 3D colliders stop touching.

Parameters

f	The frame in which the entities stopped touching.
info	The ExitInfo3D with the entities that were touching.

◆ OnTrigger3D()

void Quantum.ISignalOnTrigger3D.OnTrigger3D (Frame f,
TriggerInfo3D info
)

Called once per frame while a non-trigger and a trigger 3D colliders are touching.

Parameters

f	The frame in which the collision happened.
info	The TriggerInfo3D with data about the trigger collision.

◆ OnTriggerEnter3D()

void Quantum.ISignalOnTriggerEnter3D.OnTriggerEnter3D (Frame f,
TriggerInfo3D info
)

Called once a non-trigger and a trigger 3D colliders start touching.

Parameters

f	The frame in which the collision happened.
info	The TriggerInfo3D with data about the trigger collision.

◆ OnTriggerExit3D()

void Quantum.ISignalOnTriggerExit3D.OnTriggerExit3D (Frame f,
ExitInfo3D info
)

Called once a non-trigger and a trigger 3D colliders stop touching.

Parameters

f	The frame in which the entities stopped touching.
info	The ExitInfo3D with the entities that were touching.

Properties

◆ Map [1/2]

Map [1/2]

Map Quantum.Physics2D.PhysicsEngine2D.Api.Map

get

Retrieves the map asset currently used on the physics scene.

The map currently used by the physics scene is only updated during verified simulations, so this map can be different from the frame Map during predictions, when the latter is changed.

◆ Map [2/2]

Map Quantum.Physics3D.PhysicsEngine3D.Api.Map

get

Retrieves the map asset currently used on the physics scene.

The map currently used by the physics scene is only updated during verified simulations, so this map can be different from the frame Map during predictions, when the latter is changed.