

2D Physics API

Groups the 2D Physics classes. [More...](#)

Classes



class	Quantum.Physics2D.PhysicsEngine2D.Api 2D Physics API More...
struct	Quantum.CharacterController2D
struct	Quantum.CollisionInfo2D Info about a collision between two 2D physics colliders. More...
struct	Quantum.ExitInfo2D Info about two entities that were colliding in the 2D Physics. More...
struct	Quantum.Physics2D.Hit Information returned from a valid hit of a physics query. More...
interface	Quantum.ISignalOnCollision2D Interface for receiving callbacks once per frame while two non-trigger 2D colliders are touching. More...
interface	Quantum.ISignalOnCollisionEnter2D Interface for receiving callbacks once two non-trigger 2D colliders start touching. More...
interface	Quantum.ISignalOnCollisionExit2D Interface for receiving callbacks once two non-trigger 2D colliders stop touching. More...
interface	Quantum.ISignalOnTrigger2D Interface for receiving callbacks once per frame while a non-trigger and a trigger 2D colliders are touching. More...
interface	Quantum.ISignalOnTriggerEnter2D Interface for receiving callbacks once a non-trigger and a trigger 2D colliders start touching. More...
interface	Quantum.ISignalOnTriggerExit2D Interface for receiving callbacks once a non-trigger and a trigger 2D colliders stop touching. More...
struct	Quantum.Shape2D Defines a 2D 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.TriggerInfo2D Info about a collision between a trigger and a non-trigger 2D physics colliders. More...

Functions

int	Quantum.Physics2D.PhysicsEngine2D.Api.AddLinecastQuery (FPVector2 start, FPVector2 end, bool firstHitOnly=false, int layerMask=-1, QueryOptions options=QueryOptions.HitAll, FP verticalPosition=default, FP verticalHeight=default) Adds a broad-phase line cast query to the 2D physics scene. In order to be resolved, must be added from a system that runs prior to the Physics System. More...
int	Quantum.Physics2D.PhysicsEngine2D.Api.AddOverlapShapeQuery (FPVector2 position, FP rotation, Shape2D shape, int layerMask=-1, QueryOptions options=QueryOptions.HitAll, FP verticalPosition=default, FP verticalHeight=default) Adds a broad-phase shape overlap query to the 2D physics scene. In order to be resolved, must be added from a system that runs prior to the Physics System. More...
int	Quantum.Physics2D.PhysicsEngine2D.Api.AddOverlapShapeQuery (Transform2D transform, Shape2D shape, int layerMask=-1, QueryOptions options=QueryOptions.HitAll) Adds a broad-phase shape overlap query to the 2D physics scene. In order to be resolved, must be added from a system that runs prior to the Physics System. More...
int	Quantum.Physics2D.PhysicsEngine2D.Api.AddOverlapShapeQuery (Transform2D transform, Transform2DVertical transformVertical, Shape2D shape, int layerMask=-1, QueryOptions options=QueryOptions.HitAll) Adds a broad-phase shape overlap query to the 2D physics scene. In order to be resolved, must be added from a system that runs prior to the Physics System. More...
int	Quantum.Physics2D.PhysicsEngine2D.Api.AddRaycastQuery (FPVector2 origin, FPVector2 direction, FP distance, bool firstHitOnly=false, int layerMask=-1, QueryOptions options=QueryOptions.HitAll, FP verticalPosition=default,

	<p>FP verticalHeight=default)</p> <p>Adds a broad-phase raycast query to the 2D physics scene. In order to be resolved, must be added from a system that runs prior to the Physics System. More...</p>
int	<p>Quantum.Physics2D.PhysicsEngine2D.Api.AddShapeCastQuery (FPVector2 start, FP rotation, Shape2D *shape, FPVector2 translation, bool firstHitOnly=false, int layerMask=-1, QueryOptions options=QueryOptions.HitAll, FP verticalPosition=default, FP verticalHeight=default)</p> <p>Adds a broad-phase shape cast query to the 2D physics scene. In order to be resolved, must be added from a system that runs prior to the Physics System. More...</p>
HitCollection *	<p>Quantum.Physics2D.PhysicsEngine2D.Api.AllocatePersistentHitCollection (int defaultCapacity=64)</p> <p>Allocates a persistent HitCollection on the heap with an initial capacity. To free an allocated persistent collection of 2D hits, use FreePersistentHitCollection. More...</p>
void	<p>Quantum.Physics2D.PhysicsEngine2D.Api.FreePersistentHitCollection (HitCollection *collection)</p> <p>Free a persistent HitCollection previously allocated on the heap. To allocate a persistent collection of 2D hits, use AllocatePersistentHitCollection. More...</p>
bool	<p>Quantum.Physics2D.PhysicsEngine2D.Api.GetAllQueriesHits (out HitCollection *queriesHits, out int queriesCount)</p> <p>Gets all the results for all the broad-phase queries added to the 2D physics scene this frame. More...</p>
HitCollection	<p>Quantum.Physics2D.PhysicsEngine2D.Api.GetQueryHits (int index)</p> <p>Gets the results of a broad-phase query added to the 2D physics scene. More...</p>
Hit?	<p>Quantum.Physics2D.PhysicsEngine2D.Api.Linecast (FPVector2 start, FPVector2 end, int layerMask=-1, QueryOptions options=QueryOptions.HitAll, FP verticalPosition=default, FP verticalHeight=default)</p> <p>Queries the 2D physics scene with a line cast and returns the closest hit to the line start, if any. More...</p>
void	<p>Quantum.Physics2D.PhysicsEngine2D.Api.Linecast (HitCollection *collection, FPVector2 start, FPVector2 end, int layerMask=-1, QueryOptions options=QueryOptions.HitAll, FP verticalPosition=default, FP verticalHeight=default)</p> <p>Queries the 2D physics scene with a line cast and adds the closest hit to the line start, if any, to a persistent collection of hits. More...</p>
HitCollection	<p>Quantum.Physics2D.PhysicsEngine2D.Api.LinecastAll (FPVector2 start, FPVector2 end, int layerMask=-1, QueryOptions options=QueryOptions.HitAll, FP verticalPosition=default, FP verticalHeight=default)</p> <p>Queries the 2D physics scene with a line cast, returning all hits. More...</p>
void	<p>Quantum.Physics2D.PhysicsEngine2D.Api.LinecastAll (HitCollection *collection, FPVector2 start, FPVector2 end, int layerMask=-1, QueryOptions options=QueryOptions.HitAll, FP verticalPosition=default, FP verticalHeight=default)</p> <p>Queries the 2D physics scene with a line cast, adding all hits to a persistent collection. More...</p>
void	<p>Quantum.ISignalOnCollision2D.OnCollision2D (Frame f, CollisionInfo2D info)</p> <p>Called once per frame while two non-trigger 2D colliders are touching. More...</p>
void	<p>Quantum.ISignalOnCollisionEnter2D.OnCollisionEnter2D (Frame f, CollisionInfo2D info)</p> <p>Called once two non-trigger 2D colliders start touching. More...</p>
void	<p>Quantum.ISignalOnCollisionExit2D.OnCollisionExit2D (Frame f, ExitInfo2D info)</p> <p>Called once two non-trigger 2D colliders stop touching. More...</p>
void	<p>Quantum.ISignalOnTrigger2D.OnTrigger2D (Frame f, TriggerInfo2D info)</p> <p>Called once per frame while a non-trigger and a trigger 2D colliders are touching. More...</p>
void	<p>Quantum.ISignalOnTriggerEnter2D.OnTriggerEnter2D (Frame f, TriggerInfo2D info)</p> <p>Called once a non-trigger and a trigger 2D colliders start touching. More...</p>
void	<p>Quantum.ISignalOnTriggerExit2D.OnTriggerExit2D (Frame f, ExitInfo2D info)</p> <p>Called once a non-trigger and a trigger 2D colliders stop touching. More...</p>
HitCollection	<p>Quantum.Physics2D.PhysicsEngine2D.Api.OverlapShape (FPVector2 position, FP rotation, Shape2D shape, int layerMask=-1, QueryOptions options=QueryOptions.HitAll, FP verticalPosition=default, FP verticalHeight=default)</p> <p>Queries the 2D physics scene with a shape overlap. More...</p>
void	<p>Quantum.Physics2D.PhysicsEngine2D.Api.OverlapShape (HitCollection *collection, FPVector2 position, FP rotation, Shape2D shape, int layerMask=-1, QueryOptions options=QueryOptions.HitAll, FP verticalPosition=default, FP verticalHeight=default)</p> <p>Queries the 2D physics scene with a shape overlap. More...</p>
void	<p>Quantum.Physics2D.PhysicsEngine2D.Api.OverlapShape (HitCollection *collection, Transform2D transform, Shape2D shape, int layerMask=-1, QueryOptions options=QueryOptions.HitAll)</p> <p>Queries the 2D physics scene with a shape overlap. More...</p>
void	<p>Quantum.Physics2D.PhysicsEngine2D.Api.OverlapShape (HitCollection *collection, Transform2D transform, Transform2DVertical transformVertical, Shape2D shape, int layerMask=-1, QueryOptions options=QueryOptions.HitAll)</p> <p>Queries the 2D physics scene with a shape overlap. More...</p>
HitCollection	<p>Quantum.Physics2D.PhysicsEngine2D.Api.OverlapShape (Transform2D transform, Shape2D shape, int layerMask=-1, QueryOptions options=QueryOptions.HitAll)</p>



	Queries the 2D physics scene with a shape overlap. More...
HitCollection	Quantum.Physics2D.PhysicsEngine2D.Api.OverlapShape (Transform2D transform, Transform2DVertical transformVertical, Shape2D shape, int layerMask=-1, QueryOptions options=QueryOptions.HitAll) Queries the 2D physics scene with a shape overlap. More...
Hit?	Quantum.Physics2D.PhysicsEngine2D.Api.Raycast (FPVector2 origin, FPVector2 direction, FP distance, int layerMask=-1, QueryOptions options=QueryOptions.HitAll, FP verticalPosition=default, FP verticalHeight=default) Queries the 2D physics scene with a ray cast and returns the closest hit to the ray origin, if any. More...
void	Quantum.Physics2D.PhysicsEngine2D.Api.Raycast (HitCollection *collection, FPVector2 origin, FPVector2 direction, FP distance, int layerMask=-1, QueryOptions options=QueryOptions.HitAll, FP verticalPosition=default, FP verticalHeight=default) Queries the 2D physics scene with a ray cast and adds the closest hit to the ray origin, if any, to a persistent collection of hits. More...
HitCollection	Quantum.Physics2D.PhysicsEngine2D.Api.RaycastAll (FPVector2 origin, FPVector2 direction, FP distance, int layerMask=-1, QueryOptions options=QueryOptions.HitAll, FP verticalPosition=default, FP verticalHeight=default) Queries the 2D physics scene with a ray cast, returning all hits. More...
void	Quantum.Physics2D.PhysicsEngine2D.Api.RaycastAll (HitCollection *collection, FPVector2 origin, FPVector2 direction, FP distance, int layerMask=-1, QueryOptions options=QueryOptions.HitAll, FP verticalPosition=default, FP verticalHeight=default) Queries the 2D physics scene with a ray cast, adding all hits to a persistent collection. More...
void	Quantum.Physics2D.PhysicsEngine2D.Api.ResetMap () Resets 2D physics scene map asset guid. More...
void	Quantum.Physics2D.PhysicsEngine2D.Api.SetCallbacks (EntityRef entity, CallbackFlags flags) Sets which 2D physics collision callbacks will be called for the <i>entity</i> . More...
Hit?	Quantum.Physics2D.PhysicsEngine2D.Api.ShapeCast (FPVector2 start, FP rotation, Shape2D *shape, FPVector2 translation, int layerMask=-1, QueryOptions options=QueryOptions.HitAll, FP verticalPosition=default, FP verticalHeight=default) Queries the 2D physics scene with a shape cast and returns the hit with the shortest Hit.CastDistanceNormalized , if any. More...
void	Quantum.Physics2D.PhysicsEngine2D.Api.ShapeCast (HitCollection *collection, FPVector2 start, FP rotation, Shape2D *shape, FPVector2 translation, int layerMask=-1, QueryOptions options=QueryOptions.HitAll, FP verticalPosition=default, FP verticalHeight=default) Queries the 2D physics scene with a shape cast and adds the hit with the shortest Hit.CastDistanceNormalized , if any, to a persistent collection of hits. More...
HitCollection	Quantum.Physics2D.PhysicsEngine2D.Api.ShapeCastAll (FPVector2 start, FP rotation, Shape2D *shape, FPVector2 translation, int layerMask=-1, QueryOptions options=QueryOptions.HitAll, FP verticalPosition=default, FP verticalHeight=default) Queries the 2D physics scene with a shape cast, returning all hits. More...
void	Quantum.Physics2D.PhysicsEngine2D.Api.ShapeCastAll (HitCollection *collection, FPVector2 start, FP rotation, Shape2D *shape, FPVector2 translation, int layerMask=-1, QueryOptions options=QueryOptions.HitAll, FP verticalPosition=default, FP verticalHeight=default) Queries the 2D physics scene with a shape cast, adding all hits to a persistent collection. More...
bool	Quantum.Physics2D.PhysicsEngine2D.Api.TryGetQueryHits (int index, out HitCollection queryHits) Tries to get the results of a broad-phase query added to the 2D physics scene. More...



Detailed Description

Groups the 2D Physics classes.

Function Documentation

◆ SetCallbacks()

```
void Quantum.Physics2D.PhysicsEngine2D.Api.SetCallbacks ( EntityRef entity,
                                                         CallbackFlags flags
                                                         )
```

inline

Sets which 2D physics collision callbacks will be called for the *entity*.

The entity must have a PhysicsCollider2D 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 PhysicsCollider2D component attached.

flags The CallbackFlags of the desired collision callbacks.

```
frame.Physics2D.SetCallbacks(entity, CallbackFlags.OnDynamicCollision | CallbackFlags.OnStaticTriggerEnter);
```

◆ ResetMap()

```
void Quantum.Physics2D.PhysicsEngine2D.Api.ResetMap ( )
```

inline

Resets 2D physics scene map asset guid.

◆ AllocatePersistentHitCollection()

```
HitCollection* Quantum.Physics2D.PhysicsEngine2D.Api.AllocatePersistentHitCollection ( int defaultCapacity = 64 )
```

inline

Allocates a persistent HitCollection on the heap with an initial capacity. To free an allocated persistent collection of 2D hits, use FreePersistentHitCollection.

Parameters

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

Returns

A pointer to the allocated hit collection.

◆ FreePersistentHitCollection()

```
void Quantum.Physics2D.PhysicsEngine2D.Api.FreePersistentHitCollection ( HitCollection * collection )
```

inline

Free a persistent HitCollection previously allocated on the heap. To allocate a persistent collection of 2D hits, use AllocatePersistentHitCollection.

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()

HitCollection Quantum.Physics2D.PhysicsEngine2D.Api.GetQueryHits (int index)

inline

Gets the results of a broad-phase query added to the 2D physics scene.

Parameters

index The broad-phase query index.

Returns

A HitCollection with the query hits.

◆ **TryGetQueryHits()**

bool Quantum.Physics2D.PhysicsEngine2D.Api.TryGetQueryHits (int index, out HitCollection queryHits)

inline

Tries to get the results of a broad-phase query added to the 2D physics scene.

Parameters

index The broad-phase query index.

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

Returns

True if the index is valid, false otherwise.

◆ **GetAllQueriesHits()**

bool Quantum.Physics2D.PhysicsEngine2D.Api.GetAllQueriesHits (out HitCollection * queriesHits, out int queriesCount)

inline

Gets all the results for all the broad-phase queries added to the 2D physics scene this frame.

Parameters

queriesHits A buffer of HitCollection 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.Physics2D.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.Physics2D.PhysicsEngine2D.Api.AddRaycastQuery ( FPVector2 origin,
                                                           FPVector2 direction,
                                                           FP distance,
                                                           bool firstHitOnly = false,
                                                           int layerMask = -1,
                                                           QueryOptions options = QueryOptions.HitAll,
                                                           FP verticalPosition = default,
                                                           FP verticalHeight = default
                                                           )
```

inline

Adds a broad-phase raycast query to the 2D 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 2D 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.
- verticalPosition** The position of the query on the vertical axis, on a 2.5D context.
- verticalHeight** The height of the query from the vertical position on the positive sense of the vertical axis.

Returns

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

If using vertical position and/or height, make sure "Use Vertical Transform" is set on the Simulation Config.

◆ AddLinecastQuery()



```
int Quantum.Physics2D.PhysicsEngine2D.Api.AddLinecastQuery ( FPVector2 start,
                                                           FPVector2 end,
                                                           bool firstHitOnly = false,
                                                           int layerMask = -1,
                                                           QueryOptions options = QueryOptions.HitAll,
                                                           FP verticalPosition = default,
                                                           FP verticalHeight = default
                                                           )
```

inline

Adds a broad-phase line cast query to the 2D 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 2D world space.
end	The line end point, in a 2D world space.
firstHitOnly	If the query results should return only the closest hit to the line <i>start</i> 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.
verticalPosition	The position of the query on the vertical axis, on a 2.5D context.
verticalHeight	The height of the query from the vertical position on the positive sense of the vertical axis.

Returns

The index of the injected query, to be used when retrieving the results with [GetQueryHits](#).

If using vertical position and/or height, make sure "Use Vertical Transform" is set on the Simulation Config.

◆ Raycast() [1/2]

```
Hit? Quantum.Physics2D.PhysicsEngine2D.Api.Raycast ( FPVector2 origin,
                                                       FPVector2 direction,
                                                       FP distance,
                                                       int layerMask = -1,
                                                       QueryOptions options = QueryOptions.HitAll,
                                                       FP verticalPosition = default,
                                                       FP verticalHeight = default
                                                       )
```

inline

Queries the 2D 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 2D 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.
verticalPosition	The position of the query on the vertical axis, on a 2.5D context.
verticalHeight	The height of the query from the vertical position on the positive sense of the vertical axis.

Returns

A nullable [Hit](#) with the closest hit to the ray origin, if any.

If using vertical position and/or height, make sure "Use Vertical Transform" is set on the Simulation Config.



◆ Raycast() [2/2]

```
void Quantum.Physics2D.PhysicsEngine2D.Api.Raycast ( HitCollection * collection,
                                                    FPVector2    origin,
                                                    FPVector2    direction,
                                                    FP            distance,
                                                    int           layerMask = -1,
                                                    QueryOptions  options = QueryOptions.HitAll,
                                                    FP            verticalPosition = default,
                                                    FP            verticalHeight = default
                                                    )
```

inline

Queries the 2D 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 HitCollection, to which the query hits will be added. For allocating a persistent HitCollection, see [AllocatePersistentHitCollection](#).
- origin** The ray origin point, in a 2D 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.
- verticalPosition** The position of the query on the vertical axis, on a 2.5D context.
- verticalHeight** The height of the query from the vertical position on the positive sense of the vertical axis.

If using vertical position and/or height, make sure "Use Vertical Transform" is set on the Simulation Config.

◆ Linecast() [1/2]

```
Hit? Quantum.Physics2D.PhysicsEngine2D.Api.Linecast ( FPVector2    start,
                                                         FPVector2    end,
                                                         int           layerMask = -1,
                                                         QueryOptions  options = QueryOptions.HitAll,
                                                         FP            verticalPosition = default,
                                                         FP            verticalHeight = default
                                                         )
```

inline

Queries the 2D 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 2D world space.
- end** The line end point, in a 2D 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.
- verticalPosition** The position of the query on the vertical axis, on a 2.5D context.
- verticalHeight** The height of the query from the vertical position on the positive sense of the vertical axis.

Returns

A nullable [Hit](#) with the closest hit to the line start, if any.

If using vertical position and/or height, make sure "Use Vertical Transform" is set on the Simulation Config.

◆ Linecast() [2/2]

```
void Quantum.Physics2D.PhysicsEngine2D.Api.Linecast ( HitCollection * collection,
                                                    FPVector2 start,
                                                    FPVector2 end,
                                                    int layerMask = -1,
                                                    QueryOptions options = QueryOptions.HitAll,
                                                    FP verticalPosition = default,
                                                    FP verticalHeight = default
                                                    )
```

inline



Queries the 2D 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 HitCollection, to which the query hits will be added. For allocating a persistent HitCollection, see AllocatePersistentHitCollection .
start	The line start point, in a 2D world space.
end	The line end point, in a 2D 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.
verticalPosition	The position of the query on the vertical axis, on a 2.5D context.
verticalHeight	The height of the query from the vertical position on the positive sense of the vertical axis.

If using vertical position and/or height, make sure "Use Vertical Transform" is set on the Simulation Config.

◆ RaycastAll() [1/2]

```
HitCollection Quantum.Physics2D.PhysicsEngine2D.Api.RaycastAll ( FPVector2 origin,
                                                                FPVector2 direction,
                                                                FP distance,
                                                                int layerMask = -1,
                                                                QueryOptions options = QueryOptions.HitAll,
                                                                FP verticalPosition = default,
                                                                FP verticalHeight = default
                                                                )
```

inline

Queries the 2D physics scene with a ray cast, returning all hits.

Parameters

origin	The ray origin point, in a 2D 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.
verticalPosition	The position of the query on the vertical axis, on a 2.5D context.
verticalHeight	The height of the query from the vertical position on the positive sense of the vertical axis.

Returns

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

If using vertical position and/or height, make sure "Use Vertical Transform" is set on the Simulation Config.

◆ RaycastAll() [2/2]

```
void Quantum.Physics2D.PhysicsEngine2D.Api.RaycastAll ( HitCollection * collection,
                                                         FPVector2    origin,
                                                         FPVector2    direction,
                                                         FP           distance,
                                                         int           layerMask = -1,
                                                         QueryOptions  options = QueryOptions.HitAll,
                                                         FP           verticalPosition = default,
                                                         FP           verticalHeight = default
                                                         )
```

inline



Queries the 2D physics scene with a ray cast, adding all hits to a persistent collection.

Parameters

- collection** A previously allocated persistent HitCollection, to which the query hits will be added. For allocating a persistent HitCollection, see [AllocatePersistentHitCollection](#).
- origin** The ray origin point, in a 2D 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.
- verticalPosition** The position of the query on the vertical axis, on a 2.5D context.
- verticalHeight** The height of the query from the vertical position on the positive sense of the vertical axis.

If using vertical position and/or height, make sure "Use Vertical Transform" is set on the Simulation Config.

◆ LinecastAll() [1/2]

```
HitCollection Quantum.Physics2D.PhysicsEngine2D.Api.LinecastAll ( FPVector2    start,
                                                                    FPVector2    end,
                                                                    int           layerMask = -1,
                                                                    QueryOptions  options = QueryOptions.HitAll,
                                                                    FP           verticalPosition = default,
                                                                    FP           verticalHeight = default
                                                                    )
```

inline

Queries the 2D physics scene with a line cast, returning all hits.

Parameters

- start** The line start point, in a 2D world space.
- end** The line end point, in a 2D 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.
- verticalPosition** The position of the query on the vertical axis, on a 2.5D context.
- verticalHeight** The height of the query from the vertical position on the positive sense of the vertical axis.

Returns

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

If using vertical position and/or height, make sure "Use Vertical Transform" is set on the Simulation Config.

◆ LinecastAll() [2/2]

```
void Quantum.Physics2D.PhysicsEngine2D.Api.LinecastAll ( HitCollection * collection,
                                                         FPVector2 start,
                                                         FPVector2 end,
                                                         int layerMask = -1,
                                                         QueryOptions options = QueryOptions.HitAll,
                                                         FP verticalPosition = default,
                                                         FP verticalHeight = default
                                                         )
```

inline

Queries the 2D physics scene with a line cast, adding all hits to a persistent collection.

Parameters

collection	A previously allocated persistent HitCollection, to which the query hits will be added. For allocating a persistent HitCollection, see AllocatePersistentHitCollection .
start	The line start point, in a 2D world space.
end	The line end point, in a 2D 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.
verticalPosition	The position of the query on the vertical axis, on a 2.5D context.
verticalHeight	The height of the query from the vertical position on the positive sense of the vertical axis.

If using vertical position and/or height, make sure "Use Vertical Transform" is set on the Simulation Config.

◆ AddShapeCastQuery()

```
int Quantum.Physics2D.PhysicsEngine2D.Api.AddShapeCastQuery ( FPVector2 start,
                                                             FP rotation,
                                                             Shape2D * shape,
                                                             FPVector2 translation,
                                                             bool firstHitOnly = false,
                                                             int layerMask = -1,
                                                             QueryOptions options = QueryOptions.HitAll,
                                                             FP verticalPosition = default,
                                                             FP verticalHeight = default
                                                             )
```

inline

Adds a broad-phase shape cast query to the 2D 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 2D world space.
- rotation** The rotation of the shape, in radians.
- shape** The 2D 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 Hit.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.
- verticalPosition** The position of the query on the vertical axis, on a 2.5D context.
- verticalHeight** The height of the query from the vertical position on the positive sense of the vertical axis.

Returns

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

If using vertical position and/or height, make sure "Use Vertical Transform" is set on the Simulation Config.

◆ ShapeCast() [1/2]

```
Hit? Quantum.Physics2D.PhysicsEngine2D.Api.ShapeCast ( FPVector2 start,
                                                    FP rotation,
                                                    Shape2D * shape,
                                                    FPVector2 translation,
                                                    int layerMask = -1,
                                                    QueryOptions options = QueryOptions.HitAll,
                                                    FP verticalPosition = default,
                                                    FP verticalHeight = default
                                                    )
```

Queries the 2D physics scene with a shape cast and returns the hit with the shortest Hit.CastDistanceNormalized, if any.

Parameters

- start** The cast start, in a 2D world space.
- rotation** The rotation of the shape, in radians.
- shape** The 2D 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.
- verticalPosition** The position of the query on the vertical axis, on a 2.5D context.
- verticalHeight** The height of the query from the vertical position on the positive sense of the vertical axis.

Returns

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

If using vertical position and/or height, make sure "Use Vertical Transform" is set on the Simulation Config.

◆ ShapeCast() [2/2]



```
void Quantum.Physics2D.PhysicsEngine2D.Api.ShapeCast ( HitCollection * collection,
                                                       FVector2 start,
                                                       FP rotation,
                                                       Shape2D * shape,
                                                       FVector2 translation,
                                                       int layerMask = -1,
                                                       QueryOptions options = QueryOptions.HitAll,
                                                       FP verticalPosition = default,
                                                       FP verticalHeight = default
                                                       )
```

inline

Queries the 2D physics scene with a shape cast and adds the hit with the shortest [Hit.CastDistanceNormalized](#), if any, to a persistent collection of hits.

Parameters

- collection** A previously allocated persistent HitCollection, to which the query hits will be added. For allocating a persistent HitCollection, see [AllocatePersistentHitCollection](#).
- start** The cast start, in a 2D world space.
- rotation** The rotation of the shape, in radians.
- shape** The 2D 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.
- verticalPosition** The position of the query on the vertical axis, on a 2.5D context.
- verticalHeight** The height of the query from the vertical position on the positive sense of the vertical axis.

Returns

A nullable [Hit](#) with the smallest [Hit.CastDistanceNormalized](#), if any.

If using vertical position and/or height, make sure "Use Vertical Transform" is set on the Simulation Config.

◆ ShapeCastAll() [1/2]

HitCollection

Quantum.Physics2D.PhysicsEngine2D.Api.ShapeCastAll

```
( FPVector2 start,
  FP rotation,
  Shape2D * shape,
  FPVector2 translation,
  int layerMask = -1,
  QueryOptions options = QueryOptions.HitAll,
  FP verticalPosition = default,
  FP verticalHeight = default
)
```

inline

Queries the 2D physics scene with a shape cast, returning all hits.

Parameters

- start** The cast start, in a 2D world space.
- rotation** The rotation of the shape, in radians.
- shape** The 2D 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.
- verticalPosition** The position of the query on the vertical axis, on a 2.5D context.
- verticalHeight** The height of the query from the vertical position on the positive sense of the vertical axis.

Returns

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

If using vertical position and/or height, make sure "Use Vertical Transform" is set on the Simulation Config.

◆ ShapeCastAll() [2/2]

```
void Quantum.Physics2D.PhysicsEngine2D.Api.ShapeCastAll ( HitCollection * collection,
                                                         FPVector2 start,
                                                         FP rotation,
                                                         Shape2D * shape,
                                                         FPVector2 translation,
                                                         int layerMask = -1,
                                                         QueryOptions options = QueryOptions.HitAll,
                                                         FP verticalPosition = default,
                                                         FP verticalHeight = default
                                                         )
```

inline



Queries the 2D physics scene with a shape cast, adding all hits to a persistent collection.

Parameters

collection	A previously allocated persistent HitCollection, to which the query hits will be added. For allocating a persistent HitCollection, see AllocatePersistentHitCollection .
start	The cast start, in a 2D world space.
rotation	The rotation of the shape, in radians.
shape	The 2D 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.
verticalPosition	The position of the query on the vertical axis, on a 2.5D context.
verticalHeight	The height of the query from the vertical position on the positive sense of the vertical axis.

Returns

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

If using vertical position and/or height, make sure "Use Vertical Transform" is set on the Simulation Config.

◆ [AddOverlapShapeQuery\(\)](#) [1/3]

int

```
Quantum.Physics2D.PhysicsEngine2D.Api.AddOverlapShapeQuery ( Transform2D transform,
                                                             Transform2DVertical transformVertical,
                                                             Shape2D shape,
                                                             int layerMask = -1,
                                                             QueryOptions options = QueryOptions.HitAll
                                                             )
```



Adds a broad-phase shape overlap query to the 2D physics scene. In order to be resolved, must be added from a system that runs prior to the Physics System.

Parameters

- transform** A 2D transform component, with Position and Rotation info of the shape overlap.
- transformVertical** A vertical transform component, with Vertical Position and Height info of the shape overlap on a 2.5D context.
- shape** The 2D 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](#).

When using a vertical transform, make sure "Use Vertical Transform" is set on the Simulation Config.

◆ AddOverlapShapeQuery() [2 / 3]

int

```
Quantum.Physics2D.PhysicsEngine2D.Api.AddOverlapShapeQuery ( Transform2D transform,
                                                             Shape2D shape,
                                                             int layerMask = -1,
                                                             QueryOptions options = QueryOptions.HitAll
                                                             )
```

inline

Adds a broad-phase shape overlap query to the 2D physics scene. In order to be resolved, must be added from a system that runs prior to the Physics System.

Parameters

- transform** A 2D transform component, with Position and Rotation info of the shape overlap.
- shape** The 2D 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() [3 / 3]

```
int
Quantum.Physics2D.PhysicsEngine2D.Api.AddOverlapShapeQuery ( FPVector2    position,
                                                             FP          rotation,
                                                             Shape2D     shape,
                                                             int          layerMask = -1,
                                                             QueryOptions options = QueryOptions.HitAll,
                                                             FP          verticalPosition = default,
                                                             FP          verticalHeight = default
                                                             )
```

inline

Adds a broad-phase shape overlap query to the 2D 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 rotation of the shape, in radians.
- shape** The 2D 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.
- verticalPosition** The position of the query on the vertical axis, on a 2.5D context.
- verticalHeight** The height of the query from the vertical position on the positive sense of the vertical axis.

Returns

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

If using vertical position and/or height, make sure "Use Vertical Transform" is set on the Simulation Config.

◆ OverlapShape() [1 / 6]

```
void
Quantum.Physics2D.PhysicsEngine2D.Api.OverlapShape ( HitCollection *    collection,
                                                       Transform2D      transform,
                                                       Transform2DVertical transformVertical,
                                                       Shape2D         shape,
                                                       int             layerMask = -1,
                                                       QueryOptions    options = QueryOptions.HitAll
                                                       )
```

inline

Queries the 2D physics scene with a shape overlap.

Parameters

- collection** A previously allocated persistent `HitCollection`, to which the query hits will be added. For allocating a persistent `HitCollection`, see `AllocatePersistentHitCollection`.
- transform** A 2D transform component, with Position and Rotation info of the shape overlap.
- transformVertical** A vertical transform component, with Vertical Position and Height info of the shape overlap on a 2.5D context.
- shape** The 2D 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.

When using vertical transform, make sure "Use Vertical Transform" is set on the Simulation Config.



◆ OverlapShape() [2/6]

```
void Quantum.Physics2D.PhysicsEngine2D.Api.OverlapShape ( HitCollection * collection,
                                                         Transform2D transform,
                                                         Shape2D shape,
                                                         int layerMask = -1,
                                                         QueryOptions options = QueryOptions.HitAll
                                                         )
```

inline

Queries the 2D physics scene with a shape overlap.

Parameters

- collection** A previously allocated persistent HitCollection, to which the query hits will be added. For allocating a persistent HitCollection, see [AllocatePersistentHitCollection](#).
- transform** A 2D transform component, with Position and Rotation info of the shape overlap.
- shape** The 2D 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() [3/6]

```
void Quantum.Physics2D.PhysicsEngine2D.Api.OverlapShape ( HitCollection * collection,
                                                         FVector2D position,
                                                         FP rotation,
                                                         Shape2D shape,
                                                         int layerMask = -1,
                                                         QueryOptions options = QueryOptions.HitAll,
                                                         FP verticalPosition = default,
                                                         FP verticalHeight = default
                                                         )
```

inline

Queries the 2D physics scene with a shape overlap.

Parameters

- collection** A previously allocated persistent HitCollection, to which the query hits will be added. For allocating a persistent HitCollection, see [AllocatePersistentHitCollection](#).
- position** The position in which the shape is overlapped.
- rotation** The rotation of the shape, in radians.
- shape** The 2D 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.
- verticalPosition** The position of the query on the vertical axis, on a 2.5D context.
- verticalHeight** The height of the query from the vertical position on the positive sense of the vertical axis.

Exceptions

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

If using vertical position and/or height, make sure "Use Vertical Transform" is set on the Simulation Config.



◆ OverlapShape() [4 / 6]

HitCollection

Quantum.Physics2D.PhysicsEngine2D.Api.OverlapShape

```
( Transform2D    transform,
  Transform2DVertical transformVertical,
  Shape2D        shape,
  int            layerMask = -1,
  QueryOptions   options = QueryOptions.HitAll
)
```

inline

Queries the 2D physics scene with a shape overlap.

Parameters

- transform** A 2D transform component, with Position and Rotation info of the shape overlap.
- transformVertical** A vertical transform component, with Vertical Position and Height info of the shape overlap on a 2.5D context.
- shape** The 2D 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 HitCollection with the query hits, automatically freed when the current frame simulation ends.

When using vertical transform, make sure "Use Vertical Transform" is set on the Simulation Config.

◆ OverlapShape() [5 / 6]

HitCollection

Quantum.Physics2D.PhysicsEngine2D.Api.OverlapShape

```
( Transform2D    transform,
  Shape2D        shape,
  int            layerMask = -1,
  QueryOptions   options = QueryOptions.HitAll
)
```

inline

Queries the 2D physics scene with a shape overlap.

Parameters

- transform** A 2D transform component, with Position and Rotation info of the shape overlap.
- shape** The 2D 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 HitCollection with the query hits, automatically freed when the current frame simulation ends.

◆ OverlapShape() [6 / 6]

**HitCollection****Quantum.Physics2D.PhysicsEngine2D.Api.OverlapShape**

```
( FPVector2    position,
   FP          rotation,
   Shape2D     shape,
   int         layerMask = -1,
   QueryOptions options = QueryOptions.HitAll,
   FP         verticalPosition = default,
   FP         verticalHeight = default
)
```

inline

Queries the 2D physics scene with a shape overlap.

Parameters

- position** The position in which the shape is overlapped.
- rotation** The rotation of the shape, in radians.
- shape** The 2D 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.
- verticalPosition** The position of the query on the vertical axis, on a 2.5D context.
- verticalHeight** The height of the query from the vertical position on the positive sense of the vertical axis.

Returns

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

If using vertical position and/or height, make sure "Use Vertical Transform" is set on the Simulation Config.

◆ **OnCollision2D()**

```
void Quantum.ISignalOnCollision2D.OnCollision2D ( Frame    f,
                                                  CollisionInfo2D info
                                                  )
```

Called once per frame while two non-trigger 2D colliders are touching.

Parameters

- f** The frame in which the collision happened.
- info** The [CollisionInfo2D](#) with data about the collision.

◆ **OnCollisionEnter2D()**

```
void Quantum.ISignalOnCollisionEnter2D.OnCollisionEnter2D ( Frame    f,
                                                            CollisionInfo2D info
                                                            )
```

Called once two non-trigger 2D colliders start touching.

Parameters

- f** The frame in which the collision happened.
- info** The [CollisionInfo2D](#) with data about the collision.

◆ **OnCollisionExit2D()**

```
void Quantum.ISignalOnCollisionExit2D.OnCollisionExit2D ( Frame f,
                                                         ExitInfo2D info
                                                         )
```

Called once two non-trigger 2D colliders stop touching.

Parameters

- f** The frame in which the entities stopped touching.
- info** The [ExitInfo2D](#) with the entities that were touching.

◆ OnTrigger2D()

```
void Quantum.ISignalOnTrigger2D.OnTrigger2D ( Frame f,
                                              TriggerInfo2D info
                                              )
```

Called once per frame while a non-trigger and a trigger 2D colliders are touching.

Parameters

- f** The frame in which the collision happened.
- info** The [TriggerInfo2D](#) with data about the trigger collision.

◆ OnTriggerEnter2D()

```
void Quantum.ISignalOnTriggerEnter2D.OnTriggerEnter2D ( Frame f,
                                                         TriggerInfo2D info
                                                         )
```

Called once a non-trigger and a trigger 2D colliders start touching.

Parameters

- f** The frame in which the collision happened.
- info** The [TriggerInfo2D](#) with data about the trigger collision.

◆ OnTriggerExit2D()

```
void Quantum.ISignalOnTriggerExit2D.OnTriggerExit2D ( Frame f,
                                                       ExitInfo2D info
                                                       )
```

Called once a non-trigger and a trigger 2D colliders stop touching.

Parameters

- f** The frame in which the entities stopped touching.
- info** The [ExitInfo2D](#) with the entities that were touching.