



Search

This document is about: **QUANTUM 2**

SWITCH TO



Continuous Collision Detection (CCD)

Overview

Continuous Collision Detection is used to fast moving physics entities collide with other physics colliders instead of tunnelling through them.

There are two common approaches for CCD algorithms, *speculative* and *sweep based*. Quantum implements a *speculative Continuous Collision Detection* due to the performance considerations tied to its stateless physics engine. The speculative CCD approach is better suited for parallelism while also handling angular motion well; the former is needed for performance and the latter is necessary in many gameplay scenarios.

The speculative CCD algorithm increases the minimum bounding box used during the broad-phase for an entity based on its **PhysicsBody** component linear **Velocity** and **AngularVelocity**. It is called *speculative* because it "speculates" the entity may collide with any of the other physics objects in that area and feeds all these candidates into the solver. This speculation ensures all contact constraints are taken into account when solving the collision thus preventing tunnelling.

Set-up

Two simple steps are required to set up the CCD; both of which can be done at edit-time and / or runtime.

N.B.: Given the performance impact CCD has on the simulation, the CCD functionality is enabled on a *per-entity* basis and *NOT globally*!

Edit-Time

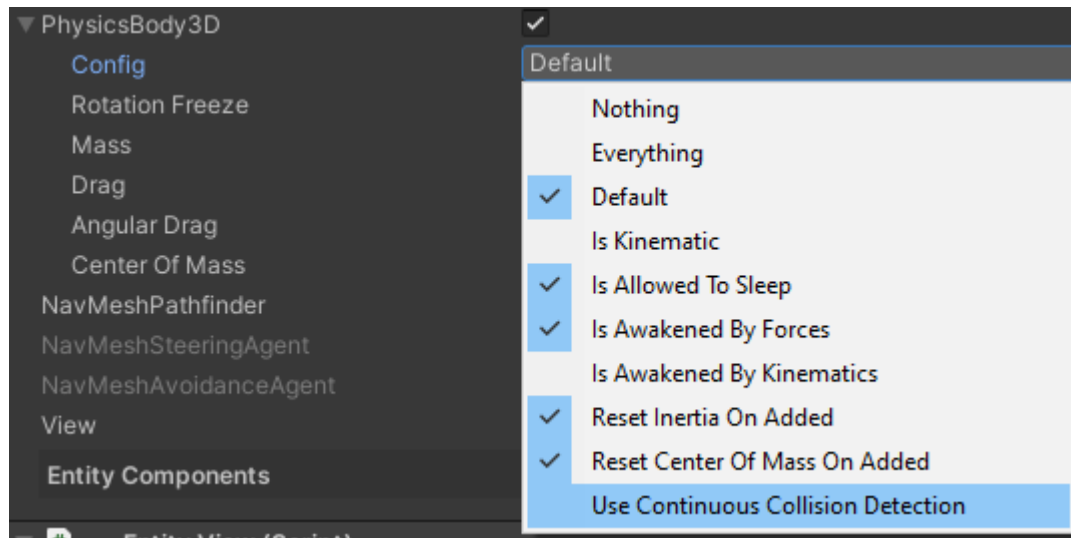


Continuous Collision Detection (CCD)

Allow CCD ☒

Enable the CCD in the Simulation Config.

Step 2: Enable the **Use Continuous Collision Detected** flag in the **Config** found on the **PhysicsBody** component of the Entity Prototype\$.



Select the CCD Flag in the PhysicsBody Config.

Runtime

Should the CCD only be necessary in particular situation or moments of the game, it is possible to dynamically toggle the CCD and entities using it on and off.

Step 1: Toggle the **AllowCCD** property in the current game state's **PhysicsSceneSettings**. The **PhysicsSceneSettings** are part of the frame and initialized with the Physics values found in the **SimulationConfig** asset. **IMPORTANT:** Do **NOT** modify the **SimulationConfig** asset at runtime, this is undeterministic and will result in desynchronization!

C#

```
frame.PhysicsSceneSettings->CCDSettings.AllowCCD = true;
```



C#

```
var physicsBody = f.Get<PhysicsBody3D>(myEntityRef);  
physicsBody.UseContinuousCollisionDetection = true;
```



Config

The **SimulationConfig** assets hold the default values for initializing the physics engine; including the aspects related to the CCD. The default values found in the **Continuous Collision Detection (CCD)** section are optimal for most games and should only be tweaked with care if edge cases were to arise.

- **AllowCCD** : Allows CCD to be performed if the Physics Body has CCD enabled on its Config flags.
- **CCDLinearVelocityThreshold** : If CCD is allowed, it will be performed on all Physics Bodies that have it enabled and have a linear velocity magnitude above this threshold.
- **CCDAngularVelocityThreshold** : If CCD is allowed, it will be performed on all Physics Bodies that have it enabled and have an angular velocity magnitude above this threshold.
- **CCDDistanceTolerance** : The absolute distance value below which the Physics Bodies under CCD check can be considered as touching.
- **MaxTimeOfImpactIterations** : The maximum number of iterations performed by the CCD algorithm when computing the time of impact between two Physics Bodies.
- **MaxRootFindingIterations** : The maximum number of iterations performed when computing the point in time when the distance between two Physics Bodies in a given separation axis is below the tolerance.

Known Limitations

Although the *speculative CCD* is feature complete, one needs to be aware of the known limitations of the speculative approach.

The current algorithm runs a single CCD iteration alongside the regular physics collision resolution. In other words, after a CCD collision is detected and resolved, the remaining delta-time for that entity



[Back to top](#)



We Make Multiplayer Simple



Products

Fusion
Quantum
Realtime
Chat
Voice
PUN

Memberships

Gaming Circle
Industries Circle

Support

Gaming Circle
Industries Circle
Circle Discord
Circle Stack Overflow

Connect

Public Discord
YouTube
Facebook
Twitter
Blog

Documentation

Fusion
Quantum
Realtime
Chat
Voice
PUN
Bolt
Server
VR | AR | MR

Resources

Dashboard
Samples
SDK Downloads
Cloud Status

Languages

English
日本語
한국어
简体中文
繁体中文



[Terms](#) [Regulatory](#) [Privacy Policy](#) [Privacy](#) [Code of Conduct](#) [Cookie Settings](#)

