Physics Performance & Optimization

Introduction

In a multiplayer game setting where real-time interactions occur, the game's performance is very important in order to provide the player with the smoothest experience possible.

While not always the culprit, physics can be a common bottleneck in these situations.

Profiling

Before doing anything further, you want to ensure that your problems are actually being caused by the physics systems.

Quantum Graph Profiler

The `Quantum Graph Profiler` is a tool for profiling all systems performance as well as graphics and Unity scripts. This is mainly useful for determining if anything Unity related is causing the performance issues.

Task profiler

This tool is used to evaluate how long each system in the Quantum Simulation takes to execute.

Generally, if you do not detect anything physics related with these tools, then the problem you are facing probably isn't the physics.

Broadphase Optimization

The physics engine is a sequence of tasks scheduled by the respective system in the System Setup. Each frame, these tasks will:

1. Create physics entries for the entities in the simulation that have physics components and collect the entries.
2. Broad-phase: Detect potentially overlapping pairs to be closely evaluated. This step is crucial to avoid a brute-force, O(n2) detection.
3. Narrow-phase: Evaluate the previous broad-phase candidates and define which ones are overlapping. This step scales linearly with the number of potential candidates filtered by the broad-phase. This step does expensive math computations.
4. Resolve velocity and position constraints using iterative solvers.
5. Integrate the forces and velocities, detect sleeping bodies and resolve previous broad-phase queries.

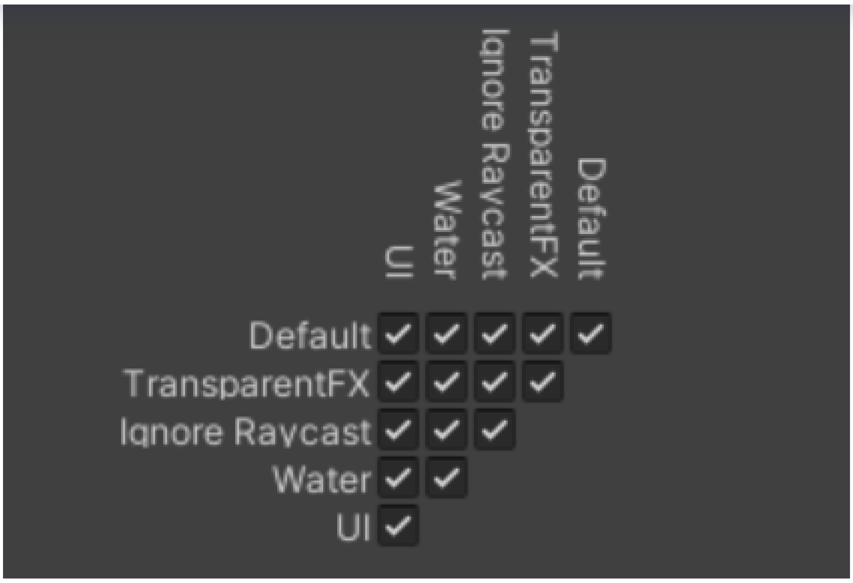
Most of these tasks scale linearly with either the number of entries or the amount of interactions between them.

Therefore, a way to reduce the amount of dynamic entries is by turning any dynamic colliders that do not move into static colliders.

For information on Static Colliders see: Static Colliders

Another way is reducing the amount of interactions between bodies, for example using physics layers or query options.

The collision matrix 3D and 2D are imported from Unity and are available in the `Simulation Config`.



	UI	Water	Ignore Raycast	TransparentFX	Default
Default	✓	✓	✓	✓	✓
TransparentFX	✓	✓	✓	✓	
Ignore Raycast	✓	✓	✓		
Water	✓	✓			
UI	✓				

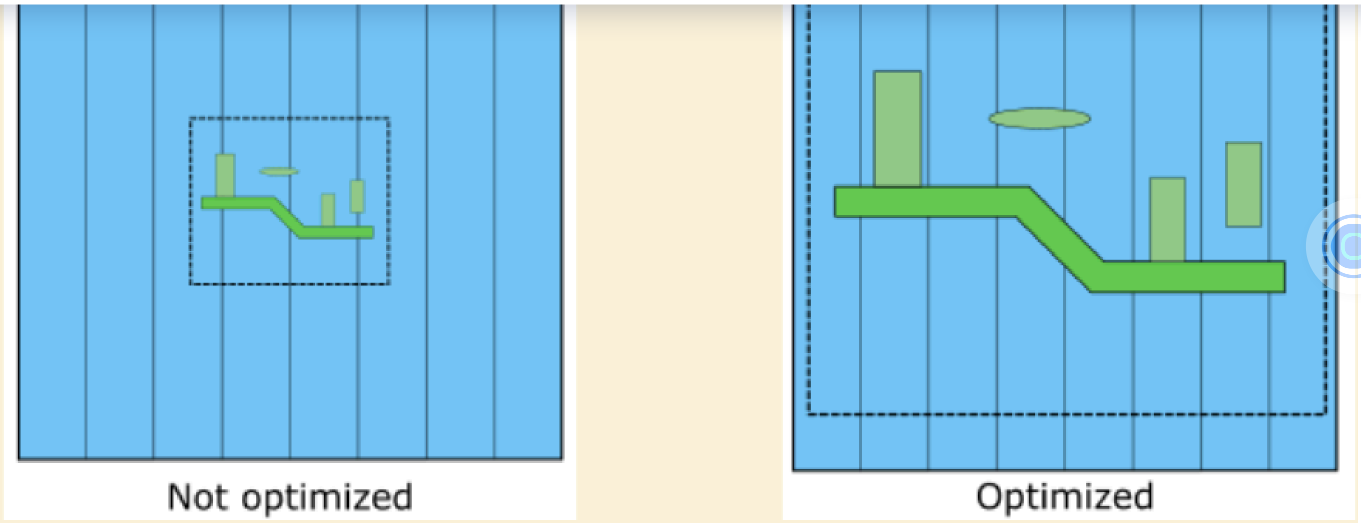
World and Bucket Size

For more in depth information on Physics Settings, see: Settings

`World Size` is a field found in the `Map` asset, which is baked by the `MapData` component. It defines the playable area in the physics engine. The map is divided into a series of sections called `Buckets`. These are used to resolve the broad and narrow phase queries.

World Size

The `World Size` should match the size of the game's playable area as close as possible in order to be efficient.



In the first image, the `World Size` is larger than the actual area used in the gameplay. The buckets are not evenly spaced and are wasted. In the second image, the objects are evenly spread in the world area.

It is important to note that adding more buckets than necessary will NOT increase the performance and too few buckets will have too many entries to evaluate. Having too many entries in the first or last bucket will greatly affect performance.

Refer to the `Task Profiler` for evaluating the performance and tweak the values based on what works best for your case.

NOTE: If something is out of the world size limits, it is considered in the first or last bucket (whichever is closest).

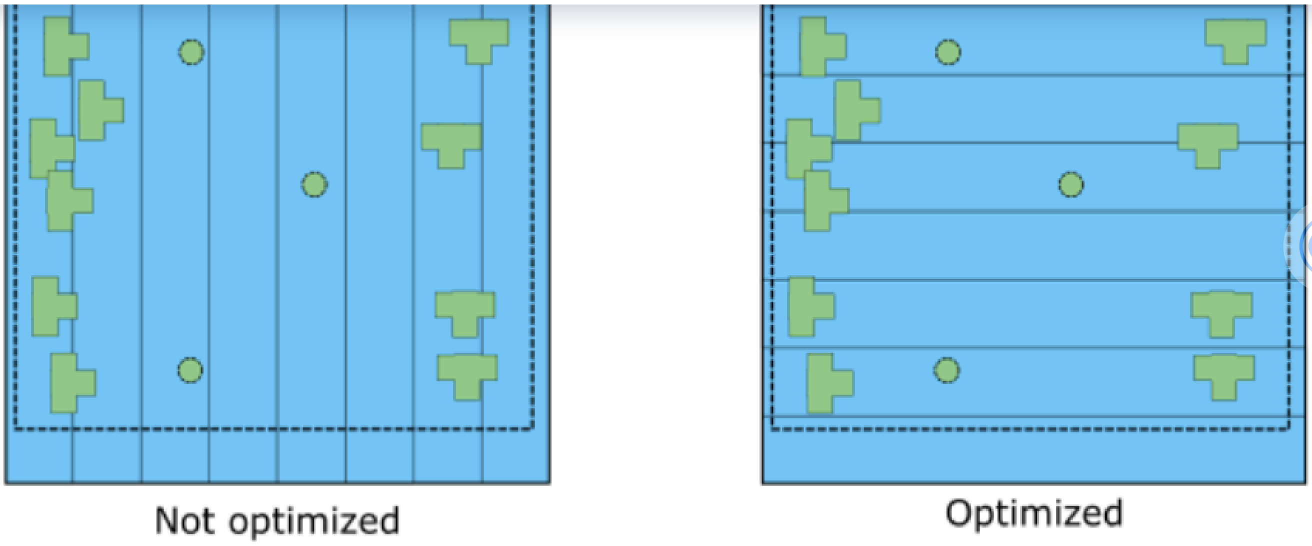
Bucket Subdivisions

Bucket Subdivisions are used to make regular physics queries more efficient. So if your game does not use normal queries, such as only using Broad-phase Queries, you may set the subdivision count to 0.

Otherwise, the default value provided in the `MapData` component is enough for most cases.

Bucketing Axis

The axis of the World Size subdivision can improve the performance depending on how the objects in your game are distributed.



In the image on the left, some buckets have 5 entries, while the others on right only have 3.

So, for this image, the `Bucketing Axis` should be set to horizontal, because the physics bodies will be more evenly spread out across the buckets.

NOTE: In this example, where there are only a few entries, the change will be minimal. However, when scaled up to hundreds or thousands of entries, it definitely matters.

Triangle Cell Size

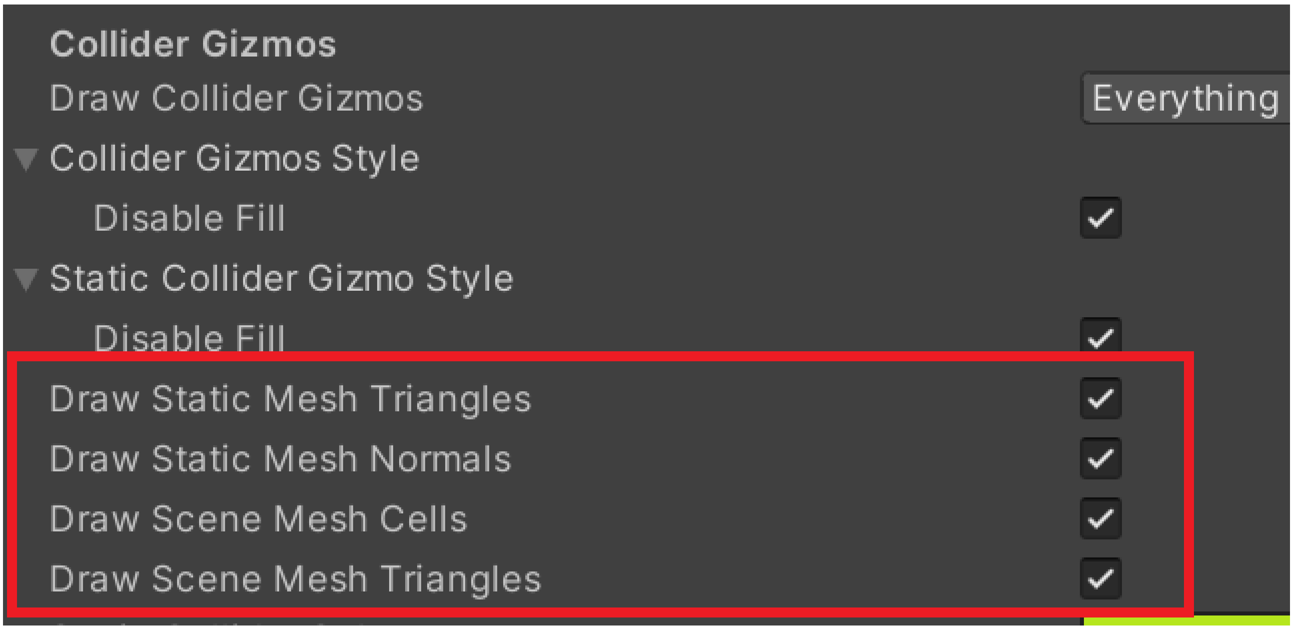
NOTE: This is only relevant for 3D games.

Triangle Cell Size defines the size of the cells into which the 3D static collider triangles are divided. Collision between the dynamic entities and the static mesh is only evaluated in the corresponding and neighboring cells.

To optimize this:

- Avoid too many triangles per cell.
- Avoid too many cells.

You should aim to have a reasonable triangle density for the terrain and mesh colliders, as they play an important role in the performance of the physcis engine. Because of this, depending on the mesh, To visualize these, enable the related fields in the `QuantumEditorSettings`.



Collider Gizmos

Draw Collider Gizmos: Everything

Collider Gizmos Style

- Disable Fill ☑

Static Collider Gizmo Style

- Disable Fill ☑

Draw Static Mesh Triangles ☑

Draw Static Mesh Normals ☑

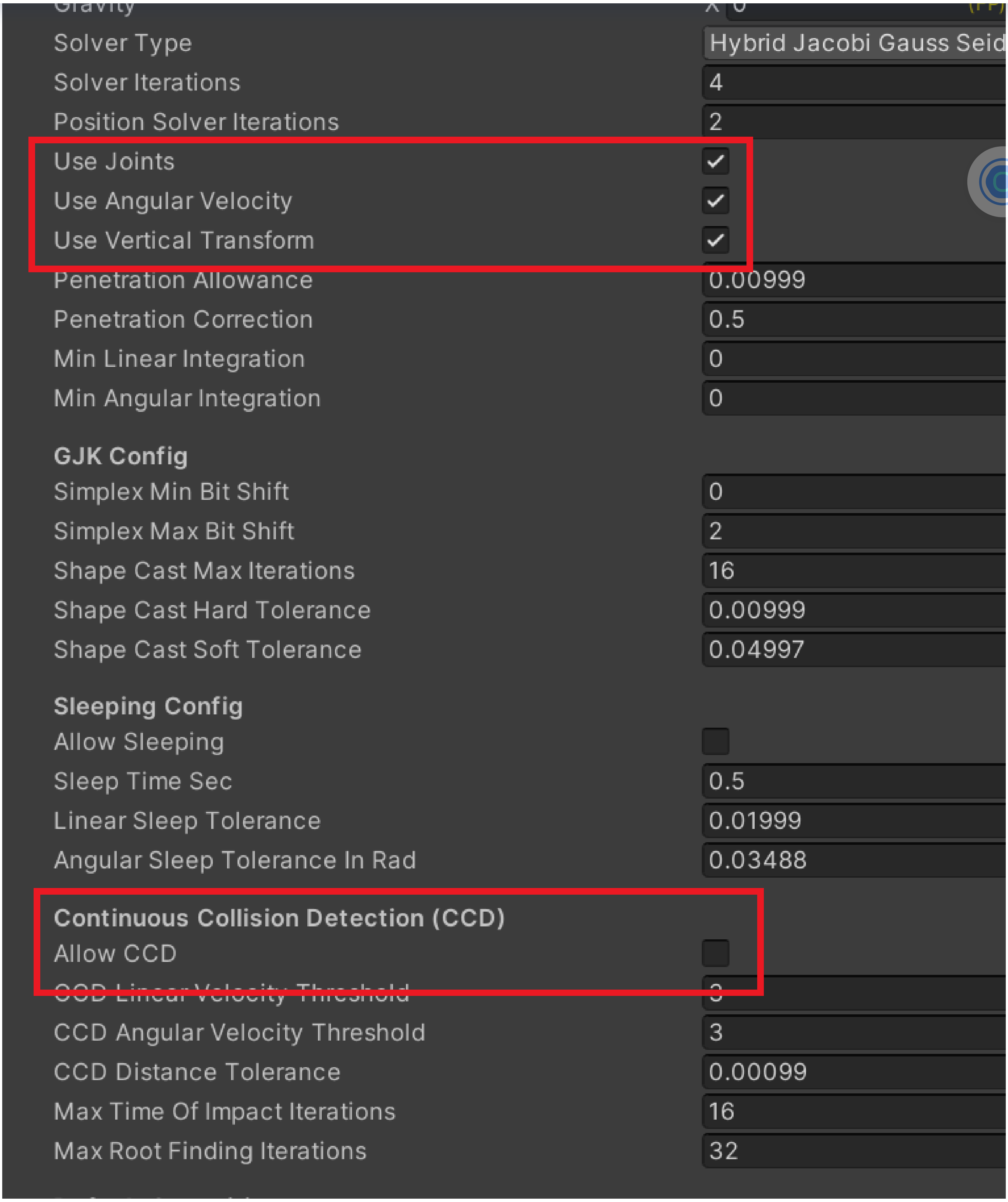
Draw Scene Mesh Cells ☑

Draw Scene Mesh Triangles ☑

Optimizing Simulation Execution

Quantum's physics engine executes multiple scheduled steps for a diversity of features, however not all of them may be needed depending on the game.

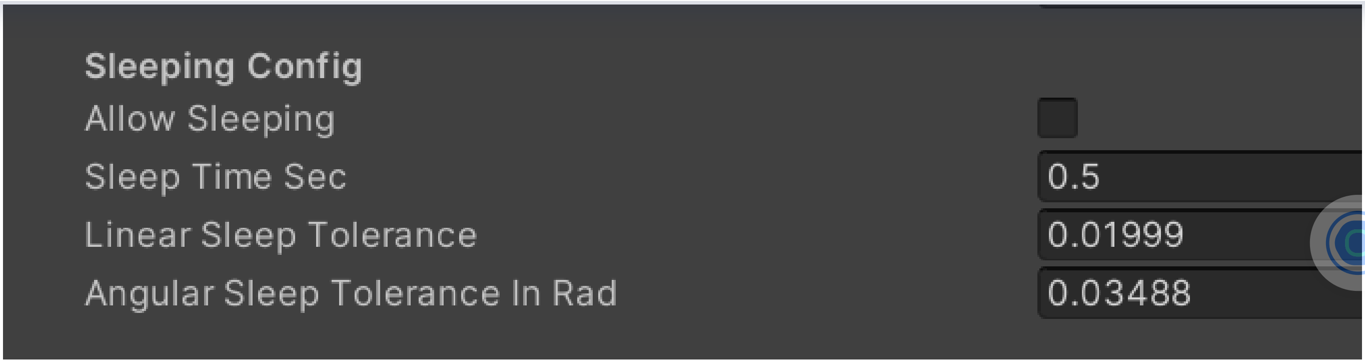
You may disable these features individually in the `SimulationConfig` asset:



Gravity	X 0 (FP)
Solver Type	Hybrid Jacobi Gauss Seid
Solver Iterations	4
Position Solver Iterations	2
Use Joints	☑
Use Angular Velocity	☑
Use Vertical Transform	☑
Penetration Allowance	0.00999
Penetration Correction	0.5
Min Linear Integration	0
Min Angular Integration	0
GJK Config	
Simplex Min Bit Shift	0
Simplex Max Bit Shift	2
Shape Cast Max Iterations	16
Shape Cast Hard Tolerance	0.00999
Shape Cast Soft Tolerance	0.04997
Sleeping Config	
Allow Sleeping	☐
Sleep Time Sec	0.5
Linear Sleep Tolerance	0.01999
Angular Sleep Tolerance In Rad	0.03488
Continuous Collision Detection (CCD)	
Allow CCD	☐
CCD Linear Velocity Threshold	3
CCD Angular Velocity Threshold	3
CCD Distance Tolerance	0.00099
Max Time Of Impact Iterations	16
Max Root Finding Iterations	32

Resting Bodies

When the game has objects that don't have to move for a long time, it will put them to sleep and skip their forces, velocity, integrations and collision detection. This option can be enabled in the



Sleeping Config

Allow Sleeping □

Sleep Time Sec 0.5

Linear Sleep Tolerance 0.01999

Angular Sleep Tolerance In Rad 0.03488

Solver Iterations

The `Solver Iterations` field in the `SimulationConfig` asset represents how many iterations are used to solve the constraints used by the physics engine, such as collisions and joints.

The default value is 4 and generally this is fine for most cases. However it can be increased or decreased depending on the accuracy required.

Raycast Optimization

If you have many raycasts happening in each frame, it is a good idea to ensure that they are correctly optimized.

Such as:

- Add a LayerMask to avoid unnecessary collision evaluations with objects that don't matter.
- Make your raycast distance as small as possible to avoid unnecessary collision evaluations.
- Use `Raycast` instead of `RaycastAll` if you only care about the first collision.
- Make use of `QueryOptions` in the Raycast method to ensure you aren't hitting things that you don't care about.

For more info on `QueryOptions` , see: Queries

Broadphase Queries

information, see: Broad-phase Queries

Back to top

Products

Fusion
Quantum
Realtime
Chat
Voice
PUN

Memberships

Gaming Circle
Industries Circle

Documentation

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

Support

Gaming Circle
Industries Circle
Circle Discord
Circle Stack Overflow

Resources

Dashboard
Samples
SDK Downloads
Cloud Status

Connect

Public Discord
YouTube
Facebook
Twitter

Languages

English
日本語
한국어
简体中文