GPU Raytracing Collisions

Learn how to enable GPU Raytracing Collisions in Niagara.
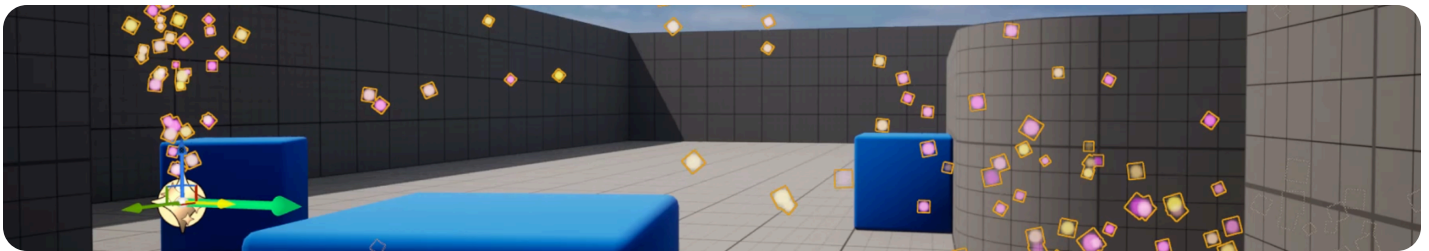
In Unreal Engine, you can set up a particle system to collide with objects in the level using the **Collision** module.

In previous versions of Unreal Engine, when using a GPU emitter, you had several options in this module. Typically, most used the **Depth Buffer** option, which generates a simulation of the environment. This is a low-cost, but also low-accuracy solution. The shapes are not accurately portrayed, and if a particle goes off the screen it disappears immediately.

GPU Raytracing Collisions is an experimental option in the Collision module, so you can use hardware ray tracing on the GPU. No matter whether the emitter and its particles are on-screen or off, or hidden behind an object, the collision will use raytracing to calculate an accurate result.
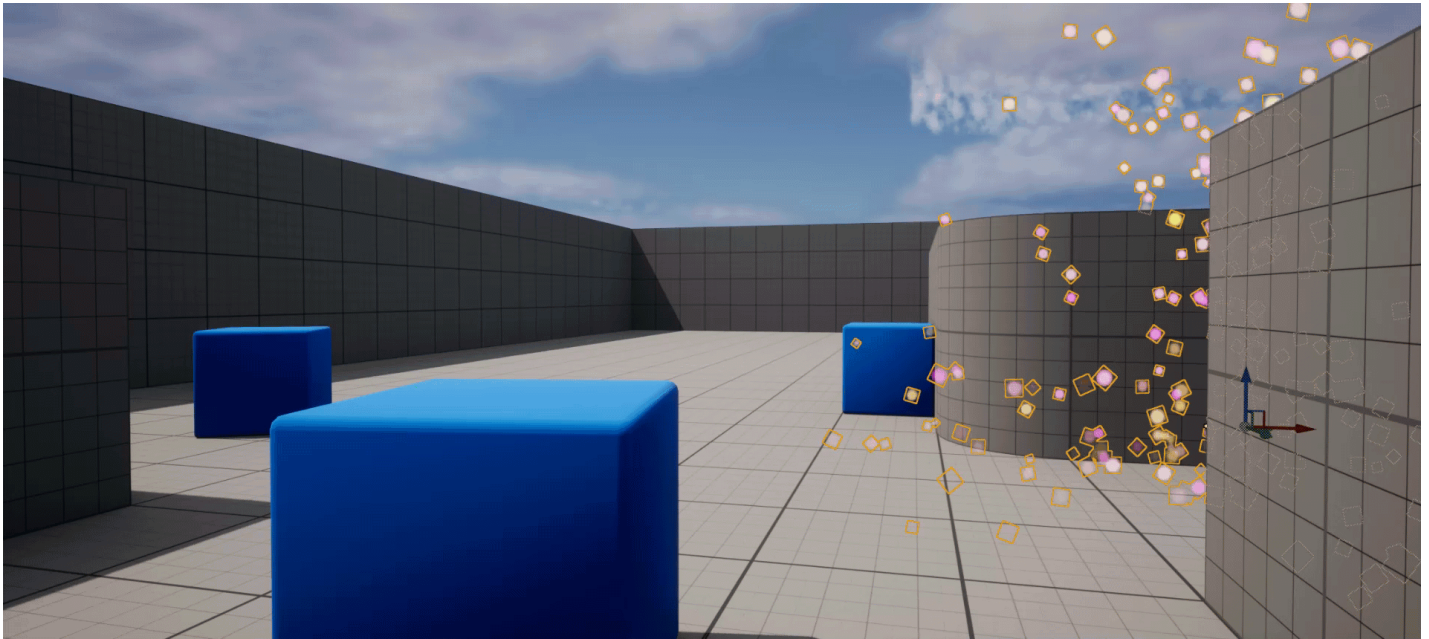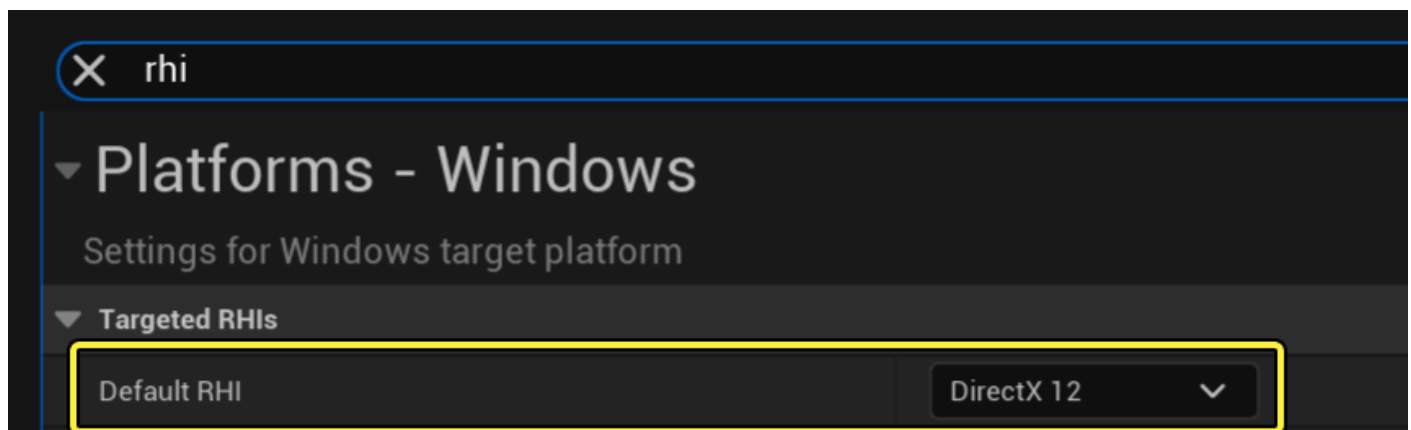
> ⓘ   The calculation is asynchronous, so the Niagara collision will be one frame behind.

Adjust Your Project Settings

To use this feature, your project must be set to DirectX 12 and your GPU must have hardware ray tracing enabled. To turn on these options, follow these instructions:
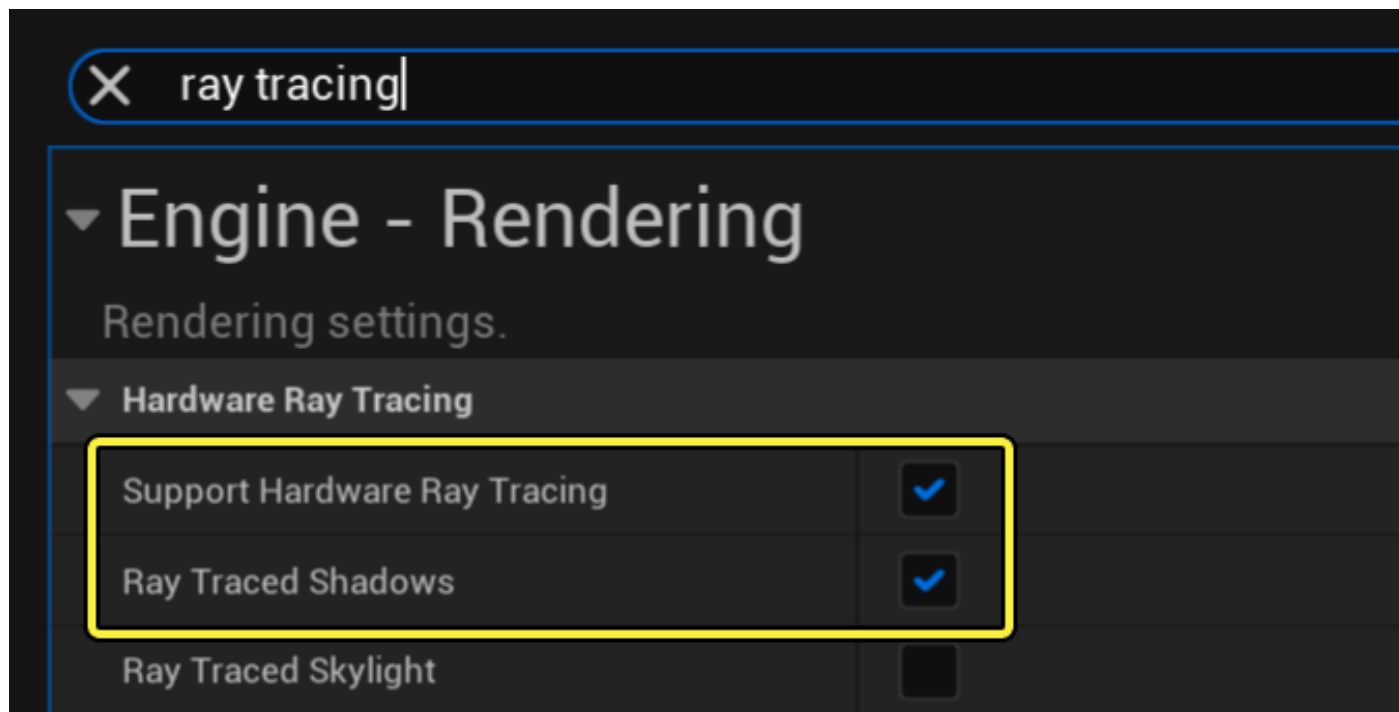
- Open **Edit > Project Settings**.
- Search for **rhi**.
- Adjust the **Default RHI** option to **DirectX 12**.

Click image for full size.

Next, enable ray tracing, also in your **Project Settings**.

- Search for **ray tracing**.
- Enable the options **Support Hardware Ray Tracing** and **Ray Traced Shadows**.
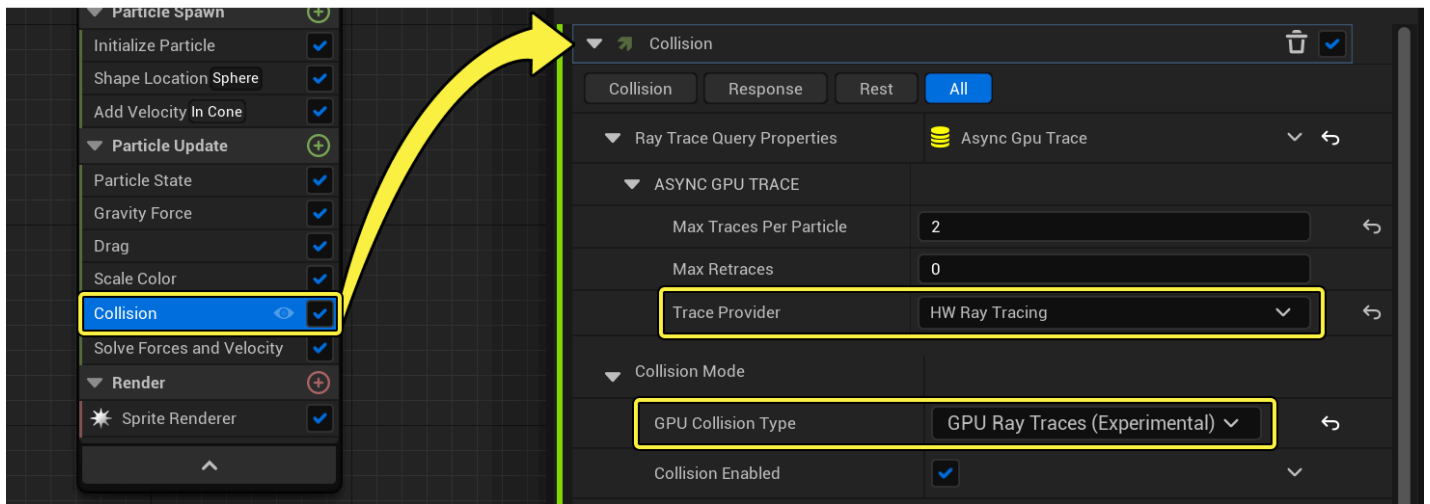

Click image for full size.

Set Up the Collision Module

In the **Collision** module, adjust the settings to use this experimental feature. Set the **GPU Collision Type** to **GPU Ray Traces (Experimental)**.

> ⓘ The **Trace Provider** is set to **Project Settings** by default, which picks the best from an array of options. However, you can also manually set this to **HW Ray Tracing**.

Set Up A Fallback (Optional)

You can set up a fallback in case ray tracing is not available. To do this, open **Edit > Project Settings**.

In the **Plugins > Niagara** section of the settings, there is a setting called **Async Gpu Trace DI**. It has an array set up with two options in it, **HW Ray Tracing** and **Global Signed Distance Fields**. When your options are set up like this, the system will first try to use ray tracing, but if ray tracing is not available then it will fall back to use distance fields instead. These are the default settings.
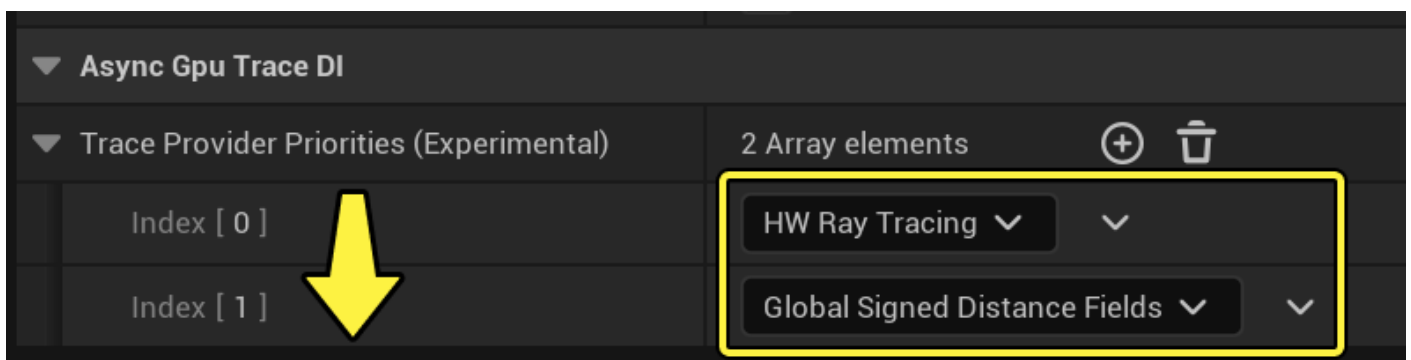
You can add elements to the array or rearrange as needed. However, in most cases, the default settings will be sufficient.