

Collision Condition Type

In dynamic gameplay scenarios, it's often important to decide when and how objects should respond to collisions—especially during events like `OnCollisionEnter`.

Should an action trigger on every collision, or only when the colliding object has a specific tag—say, a **"Player"** tag? What if you want to trigger destruction when hit by any of several tags, like **"Player"**, **"Enemy"**, or **"NPC"**? Or maybe you want certain objects to completely ignore collisions, acting as passive elements in the environment.

The **Collision Condition Type** enum helps you define exactly how your objects behave in these situations. By customizing how tag comparisons are handled during collisions, you gain precise control over which interactions matter—and which don't.

Available Collision Condition Types:

- **None** – No action is triggered on collision. Use this when the object should remain unaffected by any physical interaction.
- **Simple** – Any collision will trigger the assigned `ActionType`, regardless of tags. Great for objects that should always respond, like breakable pots or crates.
- **Single_Tag_Comparison** – The object checks for a specific tag during `OnCollisionEnter`. If the colliding object has the correct tag—like `"Player"`—the action is triggered. Perfect for cases where only certain actors should cause a reaction.
- **Multiple_Tag_Comparison** –Allows you to define a list of valid tags. If during `OnCollisionEnter` the colliding object has any of these tags, the assigned `ActionType` is executed. Ideal for shared behaviors across multiple actor types, such as both `"Enemy"` and `"Ally"` triggering a trap.