

Easy Interaction System

Add interactions to any actor in 2 minutes. Two components, zero Blueprint wiring, full multiplayer support.

v1.0.0

UE 5.4 - 5.7

100% C++

Multiplayer Ready



Installation



Before you start: Make sure you have Unreal Engine 5.4 or newer. This plugin works with UE 5.4, 5.5, 5.6, and 5.7.

1 Download & Extract

After purchasing, download the ZIP file for your engine version. **Right-click** → **Extract All**.



Do NOT open files inside the ZIP directly. Always extract first!

2 Copy to Plugins Folder

Put the `EasyInteractionSystem` folder into your project's `Plugins` folder.

```

└─ YourProject/
  └─ Content/
  └─ Config/
  └─ Plugins/ ← Create this if missing
    └─ EasyInteractionSystem/
      └─ Source/
      └─ Content/
      └─ EasyInteractionSystem.uplugin
  └─ YourProject.uproject
```

 **WRONG:** Don't put it in `Engine/Plugins` or inside `Content` !

3 Open Project & Compile

Double-click your `.uproject` file. If asked to rebuild, click **Yes**. Wait 1-3 minutes.

4 Verify

Create any Actor Blueprint, click **Add Component**, and search for:

- `Interactable Component` (goes on objects)
- `Interaction Component` (goes on player)



If you see both components, you're ready!



Quick Start (5 Minutes)



Goal: You'll have a working pickup item with an on-screen prompt in under 5 minutes. No C++ needed.

1 Add Interaction Component to Your Player

Open your **Player Character Blueprint** (e.g. `BP_ThirdPersonCharacter`).

Click **Add Component** → search `Interaction Component` → add it.

That's it for the player. Input is auto-bound to the `E` key. No wiring needed.



The component auto-creates an Enhanced Input Action and Mapping Context at runtime. If you already have your own `IA_Interact`, assign it in the Details panel under **Interaction | Input**.

2 Create an Interactable Object

Create a new **Actor Blueprint** (e.g. `BP_HealthPotion`).

Add a **Static Mesh** (any mesh, e.g. a cube or sphere).

Click **Add Component** → search `Interactable Component` → add it.

3 Configure the Interactable

Select the **Interactable Component** in your Blueprint. In the Details panel:

SETTING	SET TO
Interactable Type	Pickup
Interaction Name	Health Potion

The **Interaction Verb** auto-fills to "Pick Up" based on the type you chose.

4 Add Logic with Events

Still in your `BP_HealthPotion` Blueprint:

1. Select the **Interactable Component**
2. In Details panel, scroll to **Events**
3. Click the **+** next to `On Interacted`
4. This creates an Event node in your Event Graph
5. From that event, call **Destroy Pickup** on the Interactable Component



The `On Interacted` event gives you the **Interactor** pin (the player who pressed E). Use it to add health, give items, play sounds, etc.

5 Place in Level & Play!

Drag `BP_HealthPotion` into your level. Press **Play**.

Walk up to it and press **E**. You should see:

- ☒ Custom depth highlight when you look at it
- ☒ On-screen prompt: **[E] Pick Up Health Potion**
- ☒ Object destroyed when you press E



Done! You have a working interaction system. Read on to learn about doors, switches, hold interactions, and more.



HUD Setup

The plugin includes a built-in Canvas prompt (the "[E] Pick Up" text you see on screen). To enable it:

1 Set the HUD Class

Open your **GameMode Blueprint** (or create one).

In the Details panel, find **HUD Class** and set it to `InteractionHUD`.

2 Make Sure Your GameMode is Active

Go to **World Settings** (in your level) → **GameMode Override** → select your GameMode.

Or set it as default in **Project Settings** → **Maps & Modes** → **Default GameMode**.



Already have a custom HUD? If your game already uses a custom HUD class, you can set

`Prompt Mode = Custom Widget` on any Interactable Component and handle the UI yourself. Or set

`Prompt Mode = None` to disable prompts entirely.

You can customize every aspect of the prompt on the **Interaction Component** (on your player):

SETTING	WHAT IT DOES
Prompt Position	Top / Center / Bottom / Above Object
Prompt Scale	Size of the prompt (0.5 - 3.0)
Fade Speed	How fast prompt fades in/out
Background Color	Prompt background
Text Color	Verb and name text
Key Border Color	The [E] key box color
Frame Style	None / Thin / Rounded / Solid
Key Button Style	Bordered / Filled / Minimal / Circle



Interaction Types

The **Interactable Type** is a preset that auto-fills the verb and mode. It does NOT limit what you can do; it just saves time.

TYPE	DEFAULT VERB	DEFAULT MODE	MULTI-USE	BEST FOR
Pickup	Pick Up	Instant	No	Items, weapons, ammo
Collectable	Collect	Instant	No	Coins, notes, keys
Switch	Activate / Deactivate	Toggle	Yes	Levers, buttons, power
Door	Open / Close	Toggle	Yes	Doors, gates, hatches
Container	Open	Instant	No	Chests, crates, lockers
Custom	(empty)	Instant	Yes	Anything you want!



You can always override the verb, mode, and multi-use setting manually after selecting a type. The type is just a convenient starting point.

Interaction Modes

Instant

Player presses E → interaction fires immediately. Use for: pickups, buttons, simple interactions.

Hold

Player holds E for a duration → progress bar fills → interaction fires when full. Use for: reviving, hacking, planting bombs.

Set **Hold Duration** (seconds) in the Interactable Component. The HUD automatically shows a progress bar.

Toggle

Player presses E → state flips ON/OFF. Verb auto-swaps (e.g. "Open" becomes "Close"). Use for: doors, switches, lights.

Enable **Auto Swap Verb** and set the **Toggled Verb** to get automatic text switching.



Doors & Containers

The plugin includes built-in door and container animation nodes. No Timeline or animation Blueprint needed!

Rotating Door

In your door actor's **On Interacted** event, call **Rotate Door** on the Interactable Component.

PARAMETER	DEFAULT	WHAT IT DOES
Angle	90	How far the door opens (degrees)
Speed	2.0	Animation speed (higher = faster)
Toward Player	true	Door opens away from the player

Sliding Door

Call **Slide Door** instead. Set the **Direction** vector and **Distance**.

Close Door

Call **Close Door** to return the door to its original position. It automatically knows whether to rotate or slide back.

Container Lid

Call **Animate Lid** and pass in the lid's **Scene Component** reference. The lid rotates open. Call **Close Lid** to close it back.



All animations are replicated in multiplayer! Every player sees the door open/close.



Conditions & Locks

Lock System

Call **Set Locked** on the Interactable Component to prevent interaction. When locked, the player sees a "Locked" message (customizable) and the `On Interaction Failed` event fires.

Tag Conditions

Add entries to the **Conditions** array in the Interactable Component:

FIELD	WHAT IT DOES
Condition Tag	The Actor Tag the player must have (e.g. <code>HasKeycard</code>)
Fail Message	Text shown when condition fails (e.g. "Requires Keycard")
Invert Condition	If true, player must NOT have the tag



How to add a tag to your player: In your player Blueprint, call `Add Tag` with the tag name (e.g. `HasKeycard`) when the player picks up a key.



Events (Delegates)

Bind these in Blueprint to react to interactions. All events fire on the **server** in multiplayer.

Interactable Component Events

EVENT	WHEN IT FIRES	PARAMETERS
On Interacted	Interaction completed successfully	Interactor (Actor)
On Toggled	Toggle state changed	Interactor, New State (bool)
On Hold Progress	Every tick during hold	Interactor, Progress (0-1)
On Hover Begin	Player looks at object	Interactor
On Hover End	Player looks away	Interactor
On Interaction Failed	Locked or condition not met	Interactor, Reason (Text)
On Interaction Enabled	Interaction turned on	-
On Interaction Disabled	Interaction turned off	-

Interaction Component Events (on Player)

EVENT	WHEN IT FIRES	PARAMETERS
On Interactable Found	Player looks at an interactable	Interactable Component
On Interactable Lost	Player looks away	-
On Interaction Started	Player presses E	Interactable Component
On Interaction Completed	Interaction finished	Interactable Component



Prompt Customisation

Each interactable can override the global prompt style:

SETTING (ON INTERACTABLE)	WHAT IT DOES
Prompt Mode	Built-in / Custom Widget / None
Override Prompt Style	Enable per-object color overrides
Override Key Display Text	Show different key text (e.g. "F" instead of "E")
Override Text Color	Custom text color for this object only
Override Key Border Color	Custom key box color
Prompt Icon	Texture2D icon shown next to the name



Multiplayer



Full multiplayer support out of the box. No extra setup required.

Here's what happens automatically:

FEATURE	HOW IT WORKS
Interaction requests	Routed via Server RPC with anti-cheat validation
State (enabled, toggled, locked, collected)	Replicated to all clients
Door/Container animations	Multicast RPC - all players see them
Sound & VFX	Multicast - all players hear/see them
Hold progress	Server-authoritative, replicated to clients
Hover/Highlight	Local per-client (only you see your highlight)



Important: All gameplay functions (RotateDoor, ToggleSwitch, SetLocked, etc.) must be called on the **server**. If you call them from an `On Interacted` event, they already run on the server. If you call them from custom Blueprint logic, make sure you're on the server (use `Has Authority` or `Switch Has Authority`).



All Blueprint Functions

Interactable Component

FUNCTION	WHAT IT DOES
Try Interact	Trigger interaction (called automatically by the system)
Set Interaction Enabled	Enable/disable this interactable
Set Prompt Text	Change verb and name at runtime
Get Hold Progress	Returns 0-1 hold fill amount
Cancel Hold	Cancel active hold interaction
Rotate Door	Animate door rotation open/close
Slide Door	Animate door sliding open/close
Close Door	Return door to original position
Set Locked	Lock/unlock with optional message
Destroy Pickup	Disable + destroy actor (with optional delay)
Disable And Hide	Disable + hide actor (doesn't destroy)
Mark As Collected	Flag as collected + disable
Set Collected Visual	Dim the object visually
Toggle Switch	Flip toggle state + fire events
Activate Linked Actors	Call Activate on all Linked Actors
Set Switch State	Set toggle to specific true/false

Animate Lid	Open a container lid component
Close Lid	Close the lid back
Mark As Opened	Flag container as opened + disable
Play Interaction Sound	Play a sound at the object (replicated)
Spawn Interaction VFX	Spawn Niagara VFX at the object (replicated)
Reset Interaction	Reset everything to initial state
Change Interactable Type	Switch type at runtime

Interaction Component (Player)

FUNCTION	WHAT IT DOES
Get Current Interactable	Returns what the player is looking at
Is Interacting	True if currently in a hold interaction
Try Interact	Manually trigger interaction (usually auto-bound to E)
Stop Interact	Cancel current interaction



Troubleshooting

✗ Nothing happens when I press E

✗ Prompt not showing on screen

✖ Object highlights but nothing else works

✖ Door doesn't animate

✖ Multiplayer: interaction works for host but not client

✖ Compilation errors after installing

? FAQ

Do I need to know C++?



Can I use this with First Person AND Third Person?



Can one object have multiple interaction types?



How do I change the interact key from E to something else?



Does it work with the IActivatable interface?



Can I modify the source code?



Support

Email

support@adrenalinegames.pl

Response within 48 hours

Website

adrenalinegames.pl

All plugins & documentation



When reporting issues, include: UE version, steps to reproduce, error messages (screenshots help!), and what you've already tried.

 Easy Interaction System

Version 1.0.0 • Made with ❤️ by Erak Dev

© 2026 Adrenaline Games. All rights reserved.