

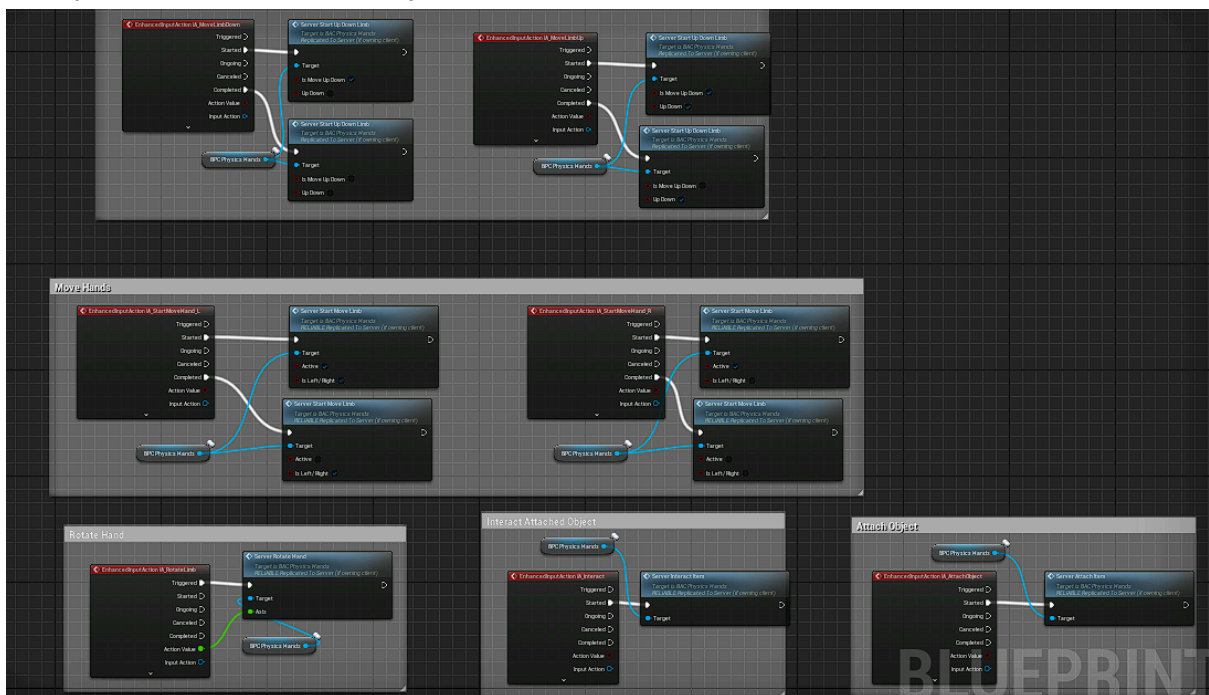
Multiplayer Physics Hands v1.0 - Documentation

This project consists for the most part of 2 components and actors, other blueprints designed to show how they work.

- **BAC_PhysicsHands** - the main component where the character physics setting is set and the logic of hand rotation is located.
- **BAC_ObjectProperties** - component responsible for the characteristics of objects that can be taken in the hand.

Character Blueprint

Here you can set the control keys:

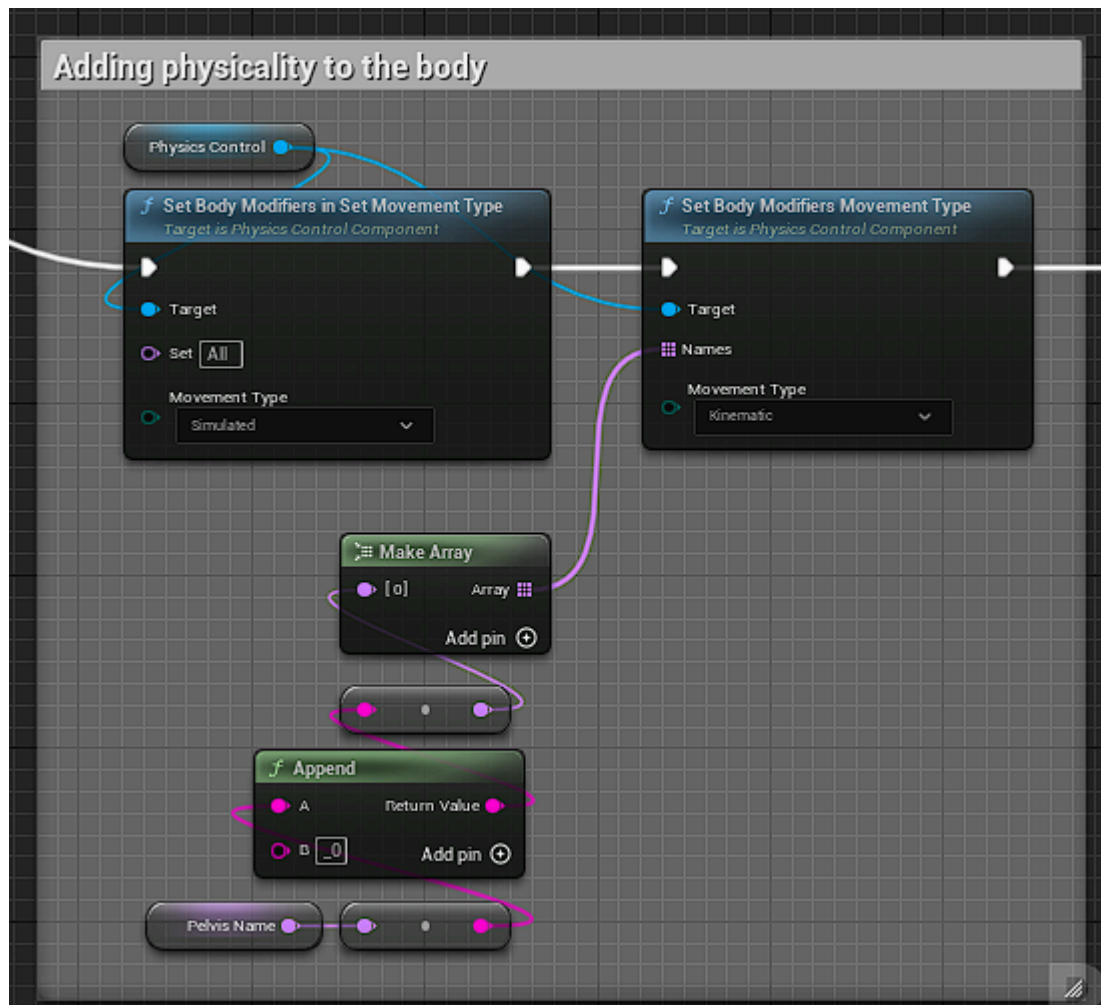


In the character in the component **BAC_PhysicsHands** you must define the name of the bones and other parameters.

Variable Name	BPC_PhysicsHands		
Tooltip			
Category	Default		
Editable when Inherited	☒		
▼ Sockets			
Parent Socket	None  		
▼ Initial			
▼ Limb Setup Data	6 Array elements		  
▶ Index [0]	3 members	▼	
▶ Index [1]	3 members	▼	
▶ Index [2]	3 members	▼	
▶ Index [3]	3 members	▼	
▶ Index [4]	3 members	▼	
▶ Index [5]	3 members	▼	
▼ Limbs Names	4 Map elements		  
hand_l	hand_l	▼	
hand_r	hand_r	▼	
foot_l	foot_l	▼	
foot_r	foot_r	▼	
▼ Limbs Attach Names	2 Map elements		  
hand_l	hand_lAttach	▼	
hand_r	hand_rAttach	▼	
Max Weight Of Object	100,0		
Clamp Hands XY	60,0		
▶ Clamp Hands Z	100,0	200,0	
▶ Clamp Hands Z World	100,0	200,0	
Pelvis Name	pelvis		

- **Limb Setup Data** - Adds physics to the assigned limbs
- **LimbsNames** - Sets the names of your character's palms and feet
- **LimbsAttachNames** - Socket names for attaching objects to the arm.
- **MaxWeightOfObject** - The maximum possible weight of an object.
- **ClampHandsXY** - X, Y boundary of hand movement relative to the mesh
- **ClampHandsZ** - Z boundary of hands movement relative to the mesh
- **PelvisName** - Name of the initial back bone

Also if you don't need physics of the whole body you can disable this block in Event Graph
BAC_PhysicsHands



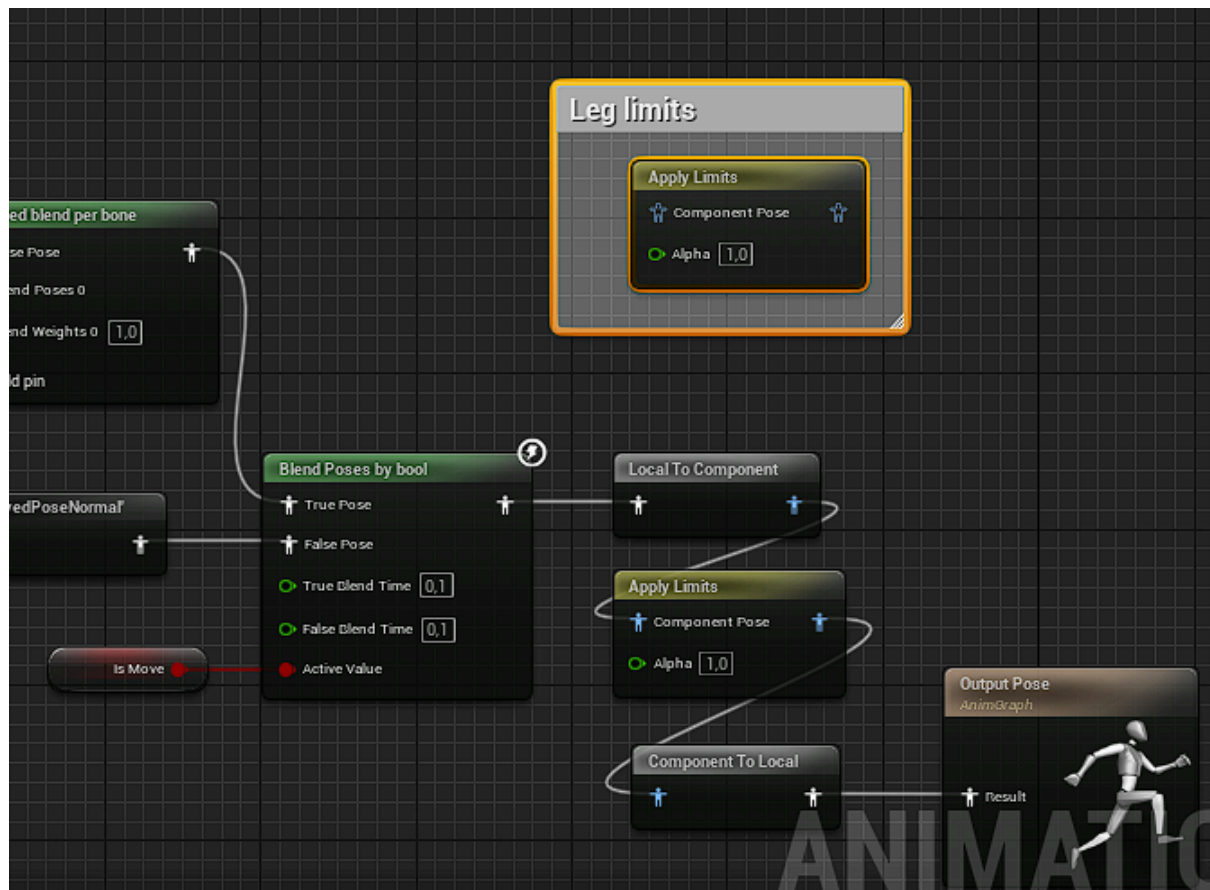
If you don't need limb physics in principle, you can disable this block. Moving hands, grabbing objects will also work, but without ragdoll.

Attaching items

In enum **EAttachModes** several modes of attaching objects are configured:

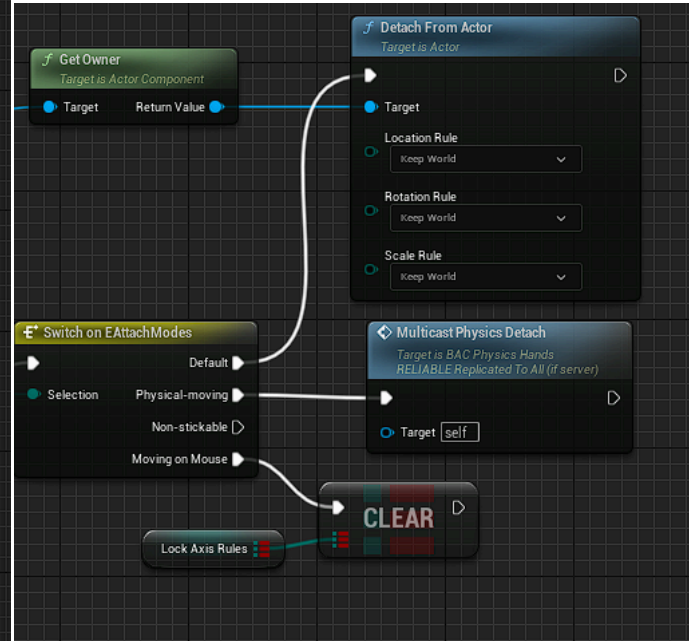
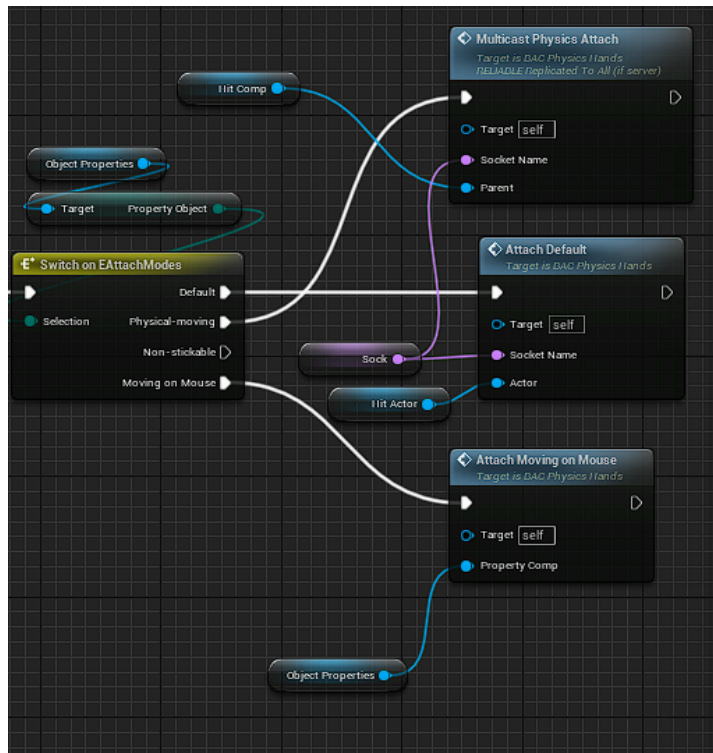
- **Non-stickable** - Item cannot be attached to the hand
- **Default** - Standard attachment to the hand, suitable for most objects.
- **Physical-moving** - Suitable for various moving platforms. The character attaches to the item and ragdoll physics is enabled.
- **MovingOnMouse** - Suitable for levers if you need to handle arm movements along certain axes, or block movement along those axes.

The character's animation blueprint is customized with limb rotation limits and logic to connect the normal animation and the required hand position.



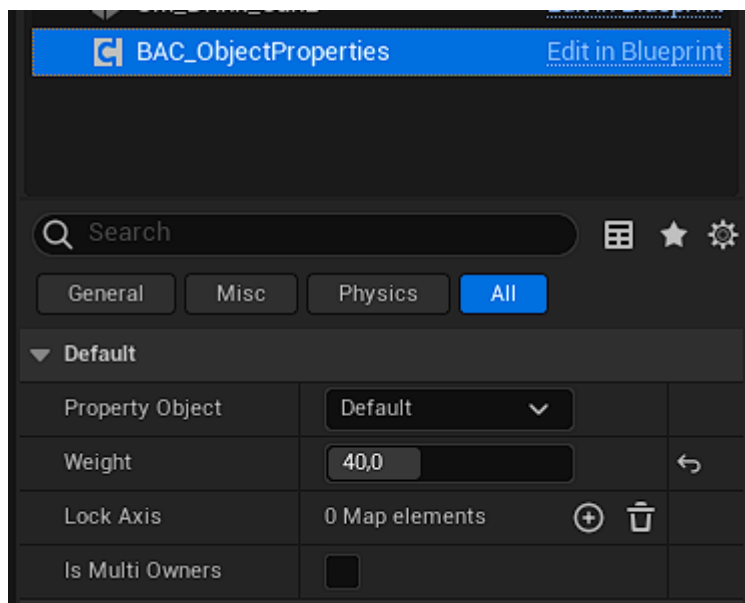
Adding your own item pickup logic and setuping actors

To add the new required logic when taking an object, add the appropriate category to **EAttachModes** and add the required functions to the **Attach Object** and **Detach Object** methods.



In order for an actor to be able to be taken in the hand you need to add the **BAC_ObjectProperties** component to it and subscribe to its dispatchers **InteractDispatcher**, **AttachDispatcher**, **DetachDispatcher** if necessary.

And also customize the item's attachment category, the need to lock the axes of arm movement, whether multiple characters can attach to this item, and the weight. The heavier the item, the heavier the arms will move.



Examples of use can be found on the initial map.

PhysicsHandsPlugin

Setup guide and Blueprint reference

Version 2.0.0 | G-Star

Copyright G-Star, 2026. All Rights Reserved.

In this guide [Installation and Character setup](#) • [Input and behavior](#) • [Targets activation and mechanisms](#) • [All editable settings](#) • [Blueprint API](#) • [Quick troubleshooting](#)

PhysicsHandsPlugin adds physical hands, object/surface grips, humanoid ragdoll manipulation and replicated levers to an existing Character. The server validates interactions; clients simulate articulated bodies locally. Full ragdoll bone streaming is not used, so small local differences in limb motion are expected.

1 Installation and Character setup

1. Put the plugin in YourProject/Plugins/PhysicsHandsPlugin and enable PhysicsHandsPlugin, Physics Control and Enhanced Input. Restart the editor when requested.
2. Open your existing Character Blueprint. Keep its normal Character parent, movement and camera logic.
3. Assign the standard Character Mesh, AnimBP and a valid Physics Asset. The asset needs real bodies on pelvis, upper/lower arms and hands, plus any other independently grabbable bones.
4. Add exactly one PhysicsHandsComponent, including inherited components. Do not add a second Physics Control Component.
5. Leave Mesh Component Name empty to use the standard Character Mesh. Map the seven bone settings for your skeleton; mannequin-style names are the defaults.
6. Leave Profile unset to edit Body/Hands locally, or assign a PhysicsHandsPluginProfile Data Asset and edit its Body/Hands instead. Input stays on the component.
7. Configure before play, consistently on server and clients. Leave component relative transform at identity; it is not the hand target origin. No Initialize graph or preset button is required.
8. Compile/save the Blueprint. Validate Character Setup reports problems but is not required to start initialization.

The main component requires ACharacter and CharacterMovement. An ordinary Actor with a Skeletal Mesh uses PhysicsHandsSkeletalItemComponent. For an NPC Character, use the main component with Role = GrabbableOnly. AnimBP normally needs no extra physics node; avoid another driver controlling the same bodies.

2 Input and behavior

Built-in defaults: hold LMB/RMB to control left/right hand; aim the camera to move selected hands; wheel rotates wrists; F grabs/drops; R adds throw impulse; E uses the held Actor;

Jump/Space or G requests recovery. Releasing LMB/RMB ends selection but keeps the grip. A mixed surface/item grip can release the surface while preserving the other hand's item.

For existing Enhanced Input, choose ProjectActions and assign distinct LeftHandAction/RightHandAction plus GrabAction. Assign optional UseAction, ThrowAction, AuxiliaryAction and your existing JumpAction/CameraLookAction. Button actions are Boolean or Axis1D; look/movement actions are Axis2D; wheel/height actions are Axis1D. Activate the mapping context in your project, or enable Add Project Mapping Context If Missing.

Auto switches all plugin input to ProjectActions if any action/context reference is assigned; unassigned commands then have no raw-key fallback. Grab and Throw must use different action assets. Continuous hand actions work for held selection; one-shot Pressed actions can use Toggle Hand Selection. With Automatic Input disabled, call the public input functions on the locally controlled Character; the plugin handles networking.

CameraAim keeps ordinary camera controls active and moves hands with the view direction. Legacy horizontal/screen plane modes use deltas and can lock camera input. Gate custom camera rotation with CanRotateCamera and custom jump/launch logic with CanJumpNormally. Arm length sets target limits, not bone stretching; unreachable targets and overlapping shapes can cause jitter.

3 Targets activation and mechanisms

Rigid prop

Actor with a root Static Mesh, simple collision, Mobility = Movable, Simulate Physics on, Query and Physics collision, blocking grab trace channel, Replicates and Replicate Movement. No item component is required. Non-root dynamic bodies need real custom body replication; a CustomReplication tag alone does not provide it.

Surface

A blocking non-simulating primitive becomes an anchored grip and ragdolls the holder. Moving wheels/platforms need synchronized continuous movement and movable collision. The plugin follows the contact, not arbitrary unsynchronized Timeline logic. Releasing the final surface recovers at the current physical location when allowed.

Humanoid

Target Character needs its own grabbable main component/Physics Asset. Grab actual bodies; the navigation capsule is skipped. Different players can hold different bones, but not the same bone. Other bodies remain simulated. Release preserves momentum; eligible Throw adds impulse. After external release, recovery is manual by default; NPCs can enable automatic recovery. External or own anchored grips block standing up; an ordinary carried item does not.

Skeletal item

Add `PhysicsHandsSkeletalItemComponent` to an Actor with a Skeletal Mesh/Physics Asset. Select Mesh Component if needed; empty Root Physics Bone selects the root physics body. Enable Ragdoll On Grab defaults true. ReleaseMode: KeepRagdoll preserves physics; RestoreImmediately restores the original setup at the current location; RestoreWhenSettled waits for stable bodies. Restoration starts only after the final grip and can end newly enabled ragdoll momentum.

Lever

Add `PhysicsHandsLeverComponent` to an Actor with Base and a separate non-root single-body Arm primitive. Arm is Movable, Query and Physics, with simulation/gravity/auto-weld off. Select Base/Arm, place LocalHingePosition at the pivot, and orient hinge local X along the rotation axis; Yaw 90 gives Base X/Z motion. No additional Physics Constraint or Arm transform driver. The angle changes only while held and persists on release. Use authority-side SetValue for save-game restoration when unheld. Endpoint events can drive a door's replicated state.

Activation

Bind the player's OnHeldActorUseRequested and call your project Blueprint Interface on HeldActor. The event runs on the server after validation. For a material switch, toggle an item Boolean IsActive with RepNotify; OnRep and BeginPlay call a local ApplyVisualState function. Replicate the state, not SetMaterial itself. Apply to every required material slot. Late joiners receive the current state.

Actor component tags

`PhysicsHandsPlugin.NoGrab`, `.NoUse` and `.NoThrow` reject the respective operation; `.Surface` and `.Constrained` affect generic target classification; `.CustomReplication` declares an existing custom replication solution. Use the full `PhysicsHandsPlugin` prefix for each tag.

4 All editable settings

The tables list native property identifiers, default values and purpose. Details usually displays spaced labels. Distances are centimeters, times seconds and angles degrees unless stated otherwise. Existing Blueprints and Profiles may retain overrides. Configure settings before play; runtime reconfiguration is not replicated.

Main component

PhysicsHandsPlugin

Setting	Default	Purpose
MeshComponentName	None / empty	Managed Character mesh; None selects the standard Mesh.
Profile	Unassigned	Optional shared Body/Hands Data Asset; overrides local Body/Hands.
Role	InteractorAndGrabbable	InteractorAndGrabbable, InteractorOnly, or GrabbableOnly for NPC targets.
BodySettings	settings below	Local configuration used only when Profile is unset.
HandsSettings	settings below	Local configuration used only when Profile is unset.

Input

Setting	Default	Purpose
bAutomaticInput	true	Bind Enhanced Input automatically; disable for manual input calls.
InputSettings	settings below	Input source, actions, keys and camera/axis behavior.
bCancelWhenInputBlocked	true	Cancel selection/outgoing grips when local input becomes blocked.

Network

Setting	Default	Purpose
bAutoEnablePropReplication	true	Enable standard prop/root replication at first grab if needed.

Debug

Setting	Default	Purpose
bDrawDebugPhysicalAnimation	false	Draw live reactive-body bounds and walking capsule.

Body Settings

Skeleton

Setting	Default	Purpose
PelvisBone	pelvis	Physical anchor bone for humanoid ragdoll/recovery.
LeftUpperArmBone	upperarm_l	Upper-arm chain bone on the corresponding side.
RightUpperArmBone	upperarm_r	Upper-arm chain bone on the corresponding side.
LeftLowerArmBone	lowerarm_l	Forearm physics bone on the corresponding side.

RightLowerArmBone	lowerarm_r	Forearm physics bone on the corresponding side.
LeftHandBone	hand_l	Palm/wrist physics bone on the corresponding side.
RightHandBone	hand_r	Palm/wrist physics bone on the corresponding side.

Physical Animation

Setting	Default	Purpose
PhysicalAnimation	settings below	Optional contact-reaction setup described below.

Locomotion

Setting	Default	Purpose
AnimationLinearStrength	18	Ordinary animation-follow positional control strength.
AnimationAngularStrength	14	Ordinary animation-follow angular control strength.
LocomotionPhysicsBlend	0.35	Ordinary visual physics blend, from 0 to 1.

Collision

Setting	Default	Purpose
LocomotionBodyCollision	QueryOnly	Ordinary body collision mode outside driven/reactive overrides.
bCollideDrivenArms	true	Allow physical contact for hand-driven arm chains.
bEnableArmBodyCollision	true	Enable arm/torso collision using a transient Physics Asset copy.

Network

Setting	Default	Purpose
PelvisUpdatesPerSecond	10	Maximum humanoid pelvis snapshot rate, Hz.
CorrectionStrength	12	Free root/pelvis network correction response.
HardCorrectionDistance	250	Distance threshold for hard free-root/pelvis correction.

Recovery

Setting	Default	Purpose
MaxRecoverySpeed	180	Maximum linear speed for character recovery, cm/s.
bAutoRecoverAfterRelease	false	Allow automatic humanoid recovery after final external release.

AutoRecoveryDelay	1 s	Delay before automatic recovery can begin.
RecoveryStableTime	0.5 s	Stable duration required for automatic recovery.
RecoveryRetryInterval	0.25 s	Time between recovery-location retries.
MaxRecoveryAngularSpeed	180	Maximum character recovery angular speed, deg/s.
RecoveryGroundDistance	100 cm	Floor search distance for requested recovery.
AutoRecoveryGroundDistance	60 cm	Floor search distance for automatic recovery.
RecoverySearchRadius	120 cm	Nearby safe standing-position search radius.
bDrawDebugCapsule	false	Draw walking/recovery capsule diagnostics.

Locomotion

Setting	Default	Purpose
AnimationMaxForce	180000	Force/torque cap for ordinary animation-follow motors.
AnimationMaxTorque	120000	Force/torque cap for ordinary animation-follow motors.

Collision

Setting	Default	Purpose
RagdollCollisionProfile	Ragdoll	Project collision profile applied to ragdoll bodies.
GrabTraceChannel	ECC_Visibility	Channel used to discover and occlude grab targets.

Physical Animation

Physical Animation

Setting	Default	Purpose
bEnabled	false	Enable locomotion contact reactions before play.
LimbSetup	Empty	Empty builds automatic limbs; explicit entries map other skeletons.
CollisionLimbs	ArmLeft, ArmRight	Limb names receiving contact simulation/blend during locomotion.
ReactionPhysicsBlend	1	Reactive-limb visual physics blend, from 0 to 1.
WorldSpaceControlData	world/parent defaults below	Reactive world-space animation-follow motor parameters.
ParentSpaceControlData	world/parent defaults below	Reactive parent-relative animation-follow motor parameters.

Hands Settings

Reach

Setting	Default	Purpose
bUseOffsetBoxLimits	false	Enable additional character-local target box limits.
OffsetMinimum	(-30,-110,-70)	Minimum/maximum corners of the optional target box.
OffsetMaximum	(110,110,110)	Minimum/maximum corners of the optional target box.
LeftArmLength	80 cm	Target arm length; zero measures the scaled skeleton.
RightArmLength	80 cm	Target arm length; zero measures the scaled skeleton.
LeftHandReachRadius	0 cm	Extra per-hand reach limit; zero adds no individual override.
RightHandReachRadius	0 cm	Extra per-hand reach limit; zero adds no individual override.
MaxReach	170	Additional overall target reach ceiling.
MaxTargetSpeed	280	Maximum hand-target movement rate, cm/s.
ArmExtensionRatio	0.92	General extension fraction of effective arm length.
ForwardArmExtensionRatio	0.98	Forward extension fraction of effective arm length.

Collision

Setting	Default	Purpose
bAvoidOwnBody	true	Sweep/slide the motor target around torso collision.
HandBodyAvoidanceRadius	6 cm	Sphere radius used for torso avoidance.

Grab

Setting	Default	Purpose
GrabRadius	18	Radius of the near-palm candidate sweep.
GrabDistance	30	Distance of the near-palm candidate sweep.
LeftGripPoint	(0, 0, 0)	Contact offset in the corresponding hand-body frame.
RightGripPoint	(0, 0, 0)	Contact offset in the corresponding hand-body frame.
MaxCarryMass	120	Selected carry/mechanism body mass limit, kg; lever excluded.
BreakForce	350000	Load threshold when the applicable grip-break option is enabled.

bBreakGripUnderLoad	false	Allow ordinary grips to release under sustained overload.
---------------------	-------	---

Mechanisms

Setting	Default	Purpose
MaxAnchoredGripDistance	180 cm	Mechanism shoulder-to-contact distance limit; zero disables it.
AnchoredGripDistanceGraceTime	0.15 s	Duration beyond mechanism distance before releasing that hand.

Grab

Setting	Default	Purpose
GripOverloadGraceTime	0.4 s	Duration above load threshold before an enabled grip break.
bLockCarriedGrip	false	Use hard translation contact for ordinary rigid props.

Character Grab

Setting	Default	Purpose
bLockCharacterGrip	true	Lock only the held character/skeletal body contact.

Grab

Setting	Default	Purpose
CarriedGripApproachSpeed	150 cm/s	Hard carry contact-anchor approach speed, cm/s; no palm teleport.

Grab Rotation

Setting	Default	Purpose
bRotateCarriedObjectsWithWrist	true	Make simple rigid carry props follow palm rotation.
GripAngularStiffness	40000	Rigid carry orientation-drive stiffness.
GripAngularDamping	3000	Rigid carry orientation-drive damping.
GripAngularMaxTorque	80000	Rigid carry orientation-drive torque cap.

Grab

Setting	Default	Purpose
GripStiffness	45000	Ordinary carry linear spring stiffness; not hard-lock positioning.
GripDamping	3000	Ordinary carry linear spring damping.
GripMaxForce	300000	Ordinary carry linear spring force cap.

GripBlendTime	0.15 s	Carry/character contact and motor handover blending duration.
---------------	--------	---

Character Grab

Setting	Default	Purpose
CharacterGripStiffness	250000	Character linear spring stiffness when hard locking is disabled.
CharacterGripDamping	8000	Character spring-path damping.
CharacterGripMaxForce	2000000	Character spring-path force cap.

Surface

Setting	Default	Purpose
bStabilizeSurfaceRagdoll	true	Temporarily stabilize ragdoll solver/joints during surface holds.
SurfacePositionIterations	32	Temporary ragdoll position solver iterations while surface-held.
SurfaceVelocityIterations	8	Temporary ragdoll velocity solver iterations while surface-held.
bLockSurfaceGrip	true	Hard linear surface attachment with gradual anchor approach.
SurfaceLockTolerance	2 cm	Contact-distance tolerance for kinematic lever locking.
SurfaceGripApproachSpeed	150 cm/s	Surface contact-anchor approach speed, cm/s.
SurfaceGripAngularDamping	8	Relative palm rotational damping; zero disables it.
SurfaceGripAngularMaxTorque	40000	Relative surface rotational-damping torque cap.
bBreakSurfaceGripUnderLoad	false	Allow sustained-load release of surface grips.
SurfaceGripStiffness	200000	Surface linear spring stiffness when hard locking is disabled.
SurfaceGripDamping	10000	Surface spring-path damping.
SurfaceGripMaxForce	2000000	Surface spring-path force cap.
SurfaceGripBlendTime	0.12 s	Minimum surface contact approach blend duration.

Throw

Setting	Default	Purpose
ThrowSpeed	650	Additional velocity-scale impulse for eligible released bodies.

Use

Setting	Default	Purpose
---------	---------	---------

MaxUseDistance	80 cm	Maximum palm-to-held-anchor distance for item activation.
----------------	-------	---

Drive

Setting	Default	Purpose
HandStrength	22	Hand-position Physics Control strength.
HandMaxForce	220000	Hand-position drive force cap.
HandDampingRatio	1.2	Hand-position drive damping ratio.
MaxHandDriveVelocity	800 cm/s	Maximum hand-drive target velocity, cm/s.

Wrist

Setting	Default	Purpose
bEnableWristRotation	true	Enable selected-hand wrist control.
LeftWristRotationAxis	(1, 0, 0)	Twist axis in the corresponding hand bone local frame.
RightWristRotationAxis	(1, 0, 0)	Twist axis in the corresponding hand bone local frame.
WristAngleMinimum	-75 deg	Minimum/maximum desired wrist twist, degrees.
WristAngleMaximum	75 deg	Minimum/maximum desired wrist twist, degrees.
MaxWristRotationSpeed	180	Maximum wrist target angle rate, deg/s.
WristStrength	8	Wrist angular Physics Control strength.
WristDampingRatio	1.2	Wrist motor damping ratio.
WristMaxTorque	40000	Wrist motor torque cap.
WristBlendTime	0.2 s	Wrist drive blending duration.

Network

Setting	Default	Purpose
TargetUpdatesPerSecond	15	Maximum hand-intent transmission rate, Hz.
InputLeaseSeconds	1.5	Timeout for stale active-hand input intent.

Input Settings

Input

Setting	Default	Purpose
InputSource	Auto	Auto selects ProjectActions if any action/context is assigned.
bToggleHandSelection	false	Press again to deselect; useful with one-shot button triggers.

Enhanced Input

Setting	Default	Purpose
ProjectMappingContext	Unassigned	Context to track/use; not added automatically by default.
bAddProjectMappingContextIfMissing	false	Allow component-owned activation of an absent project context.
LeftHandAction	Unassigned	Distinct Boolean/Axis1D hand-selection actions; continuous while held.
RightHandAction	Unassigned	Distinct Boolean/Axis1D hand-selection actions; continuous while held.
GrabAction	Unassigned	Boolean/Axis1D grab and ordinary-release action.
ThrowAction	Unassigned	Separate optional Boolean/Axis1D extra-impulse action.
UseAction	Unassigned	Optional Boolean/Axis1D held-actor activation action.
RecoveryAction	Unassigned	Optional Boolean/Axis1D recovery action.
JumpAction	Unassigned	Existing Boolean/Axis1D jump action used for ragdoll recovery.

Input

Setting	Default	Purpose
bRecoverOnJump	true	Request recovery through jump input while ragdolled.
JumpKey	SpaceBar	Built-in recovery-on-jump key; normal jump remains project-owned.

Enhanced Input

Setting	Default	Purpose
HandMoveAction	Unassigned	Optional Axis2D hand delta in legacy plane modes.
CameraLookAction	Unassigned	Existing Axis2D look action; Camera Aim leaves host camera active.
AuxiliaryAction	Unassigned	Optional Axis1D wrist/depth command.
HeightAction	Unassigned	Optional Axis1D elevation command.

Input

Setting	Default	Purpose
HandMovementMode	CameraAim	CameraAim, CameraHorizontalPlane, or CameraScreenPlane.

Camera Aim

Setting	Default	Purpose
AimReachRatio	0.95	Camera Aim initial length fraction before other target clamps.
AimHandSpacing	6 cm	Per-hand lateral camera-aim spacing.
AimPitchMinimum	-75 deg	Minimum/maximum pitch used to aim hand targets, degrees.
AimPitchMaximum	85 deg	Minimum/maximum pitch used to aim hand targets, degrees.

Input

Setting	Default	Purpose
ElevationModifierKey	LeftShift	Built-in horizontal-plane modifier for raising/lowering the hand.
PlaneForwardSensitivity	3	Horizontal-plane forward/retract delta scale.
HeightSensitivity	3	Explicit/elevation-modified vertical delta scale.
LeftHandKey	LeftMouseButton	Built-in selection button for the corresponding hand.
RightHandKey	RightMouseButton	Built-in selection button for the corresponding hand.
GrabKey	F	Built-in grab/drop button.
ThrowKey	R	Built-in additional-impulse throw button.
UseKey	E	Built-in held-item activation button.
RecoveryKey	G	Built-in separate recovery button.
HorizontalAxis	MouseX	Built-in horizontal/vertical delta input axis.
VerticalAxis	MouseY	Built-in horizontal/vertical delta input axis.
ForwardAxis	MouseWheelAxis	Built-in auxiliary wheel axis: wrist rotation or depth.
AuxiliaryAxisMode	WristRotation	WristRotation or MoveDepth interpretation.
WristRotationSensitivity	12	Degrees per auxiliary unit; negative reverses direction.
HorizontalSensitivity	3	Sideways delta sensitivity in legacy modes.
VerticalSensitivity	3	Vertical delta sensitivity in legacy modes.
ForwardSensitivity	10	Depth-adjustment delta sensitivity.
bScaleAxesByDeltaSeconds	false	For rate-based sticks; normally false for mouse deltas.
MappingPriority	100	Priority of a context added by this component.
bConsumeButtons	true	Consume handled automatic input commands.

bLockCameraWhileDrivingHands	true	Camera lock in legacy plane modes; Camera Aim ignores it.
------------------------------	------	---

Lever component

Components

Setting	Default	Purpose
BaseComponent	Unassigned	Lever base scene component on the same Actor.
ArmComponent	Unassigned	Separate non-root, non-skeletal single-body lever primitive.
BasePrimitiveName	Base	Fallback component name when the corresponding picker is empty.
ArmPrimitiveName	Arm	Fallback component name when the corresponding picker is empty.

Hinge

Setting	Default	Purpose
LocalHingePosition	(0,0,10)	Pivot in Base-local coordinates.
LocalHingeRotation	(0,90,0)	Its local X is the hinge axis; Yaw 90 gives X/Z motion.
MaximumAngle	65	Lever travel limit relative to authored zero pose, degrees.
InitialValue	0	Starting lever position, normalized -1 to +1.

Motion

Setting	Default	Purpose
MaximumAngularSpeed	120 deg/s	Maximum kinematic lever movement rate.
MotionResponse	12	Lever response/smoothing toward requested angle.
AngularDeadZone	0.5 deg	Ignore smaller lever angle changes.

Network

Setting	Default	Purpose
UpdatesPerSecond	10	Maximum moving lever state publication rate, Hz.

Switch

Setting	Default	Purpose
SwitchThreshold	0.95	Positive turns on, negative turns off; state persists between.

bInitiallyOn	false	Initial switch state between endpoint thresholds.
--------------	-------	---

Skeletal Item component

Skeletal Item

Setting	Default	Purpose
MeshComponent	Unassigned	Managed item Skeletal Mesh; empty selects the only mesh.
RootPhysicsBone	None / empty	Item root physics body; empty selects Physics Asset root body.
bEnableRagdollOnGrab	true	Enable item simulation on grab; false preserves current setup.
ReleaseMode	KeepRagdoll	KeepRagdoll, RestoreImmediately, or RestoreWhenSettled.

Recovery

Setting	Default	Purpose
RecoveryDelay	0.5 s	Item settling delay after final release.
SettledTime	0.5 s	Duration all simulated item bodies must satisfy settling limits.
MaxSettledSpeed	30 cm/s	Maximum item linear settling speed.
MaxSettledAngularSpeed	20 deg/s	Maximum item angular settling speed.

Network

Setting	Default	Purpose
RootUpdatesPerSecond	10	Maximum item root snapshot publication rate, Hz.
CorrectionStrength	7	Free root/pelvis network correction response.
HardCorrectionDistance	100 cm	Distance threshold for hard free-root/pelvis correction.

Platform Motion

Motion

Setting	Default	Purpose
Travel	(0,0,180)	Time-driven platform translation vector from its initial position.
Period	6	Platform movement period, seconds.

Physical Animation nested fields

LimbSetup entries: LimbName (empty) identifies a unique limb; StartBone (empty) selects its chain root; bIncludeParentBone (false) adds its parent; bCreateWorldSpaceControls, bCreateParentSpaceControls and bCreateBodyModifiers (true) create the respective controls/modifiers. Body modifiers are required. Order limbs before the final torso/spine limb and avoid duplicate body ownership. CollisionLimbs must match LimbName.

WorldSpaceControlData / ParentSpaceControlData defaults and purpose:

- **bEnabled:** true / true; enable the motor before mode/selection overrides.
- **LinearStrength:** 2 / 0; positional control strength.
- **AngularStrength:** 2 / 4; rotational control strength.
- **LinearDampingRatio, AngularDampingRatio:** 1.2 / 1.2 on both; damping relative to strength.
- **LinearExtraDamping, AngularExtraDamping:** 0 on both; damping even without strength.
- **MaxForce:** 50000 / 50000; force cap; zero means uncapped.
- **MaxTorque:** 35000 / 40000; torque cap; zero means uncapped.
- **LinearTargetVelocityMultiplier, AngularTargetVelocityMultiplier:** 1 on both; scale animation-target velocity used by damping.
- **CustomControlPoint:** (0,0,0) on both; custom control-point offset.
- **bUseCustomControlPoint:** false on both; otherwise use the automatic point.
- **bUseSkeletalAnimation:** true on both; required for animation-follow targets.
- **bDisableCollision:** false / true; disable parent-child collision where a parent exists.
- **bOnlyControlChildObject:** true on both; do not apply drive reaction to the parent.

Physical Animation is optional and off initially. Enable it in the effective Body/Profile before play. Empty LimbSetup is automatic, not an error. Inspect IsPhysicalAnimationConfigured and live diagnostics. Free-arm contact needs physical shapes and a target the arm reaches before the walking capsule blocks it. Selected/held arms and ragdoll suspend competing animation motors.

For ordinary humanoid recovery: require a suitable floor, acceptable relative speed and standing capsule clearance. RequestRecovery can wait/retry. For surfaces, locked grip tuning controls attachment; corresponding spring parameters apply only when locking is disabled. Mechanism distance release affects only Constrained grips, not carry/character/ordinary surface holds.

5 Blueprint API

Native names below identify the Blueprint nodes; the editor adds spaces. Local input calls belong on the locally controlled Character. The plugin sends its own requests. [Server] calls require actual authority; they do not automatically become client-to-server RPCs. Getter/event state initializes UI; old actions are not replayed for late joiners.

PhysicsHandsComponent setup and queries

- **ValidateSetup(Errors):** bool result and current setup error array.
- **IsReady():** initialization completed.
- **GetCharacterMesh():** managed Skeletal Mesh.

- **CanBeGrabbed():** current role/lifecycle permits humanoid grabs.
- **GetGameplayLocation():** location appropriate to current physical mode.
- **IsRagdoll():** current physics mode, not death status.
- **GetExternalGripCount():** outside grips on this humanoid.
- **GetHandBone(Hand), GetHandWorldLocation(Hand):** configured bone/current palm position.
- **IsHandControlActive(Hand):** hand selection/control state.
- **GetHandState(Hand):** observed drive/grip snapshot.
- **GetHeldActor(Hand), GetHeldComponent(Hand):** current target or none.
- **HasActiveHand(), HasAnyGrab():** any selection/any grip.
- **HasSurfaceGrab(), HasAnchoredGrab():** surface/standing-blocking anchored grip.
- **GetArmLength(Hand), GetCalculatedArmLength(Hand):** effective/reference-skeleton lengths.
- **CanRotateCamera():** gate custom camera rotation.
- **CanJumpNormally():** gate custom jump/launch behavior.

Hand is Left or Right. Hand State exposes Driven, TargetOffset, WristAngleDegrees, Kind, Target, TargetBone, TargetAnchor, HandAnchor and LastReleaseWasThrow. Kind is None, Carry, Surface, Character or Constrained. Contact anchors are body-local. A deselected hand can remain held.

PhysicsHandsComponent input and lifecycle

- **SetHandActive(Hand, Active):** select/deselect; does not itself grab/drop.
- **MoveActiveHands(Horizontal, Vertical, Forward):** centimeter deltas/aim adjustments.
- **RotateActiveWrists(DeltaDegrees):** selected-hand twist delta.
- **ToggleGrab():** selected-hand grab/drop policy.
- **GrabHand(Hand):** explicit grab request; select the hand first.
- **ReleaseHand(Hand, Throw=false):** release one hand, optionally requesting extra impulse.
- **ThrowHeld():** request eligible additional-impulse throws.
- **UseHeld():** request validated item activation.
- **RequestRecovery():** request safe standing recovery; may wait/retry.
- **CancelLocalInput():** clear local interaction input/selection as required by lifecycle.
- **EnterRagdoll() [Server]:** enter physical mode.
- **TryRecover() [Server]:** immediate attempt; check returned bool.
- **ReleaseAll() [Server]:** release all outgoing grips.
- **SetInteractionEnabled(Enabled) [Server]:** enable/disable interaction lifecycle.
- **IsComponentHeld(Target) [Server]:** query the authority grip registry.

PhysicsHandsComponent diagnostics

- **IsPhysicalAnimationConfigured():** physical animation initialized successfully.
- **GetPhysicalAnimationDiagnostics(), PrintPhysicalAnimationDiagnostics():** return/log body/motor report.
- **GetSurfaceGripDiagnostics(), PrintSurfaceGripDiagnostics():** return/log surface attachment report.

- **ApplyPhysicalAnimationContactPreset(), ApplyRecommendedHandSettings():** hidden compatibility helpers for older graphs; not required for new setup.

Read-only runtime maps: LimbWorldSpaceControls, LimbParentSpaceControls, LimbBodyModifiers. Do not manually destroy their internal motors during interaction.

PhysicsHandsComponent events

- **OnReady():** initialization complete; also check IsReady when subscribing.
- **OnHandControlChanged(Hand, Active):** selection transition, not pickup.
- **OnGrabChanged(Hand, State):** observed hand/grip state changed.
- **OnInteractionFailed(Reason):** rejected/unavailable request feedback.
- **OnRagdollChanged(Ragdoll):** observed body-mode transition.
- **OnActorGrabbed(Hand, HeldActor, HeldComponent, Bone):** observed pickup.
- **OnActorReleased(Hand, HeldActor, HeldComponent, Bone):** observed release, including throw.
- **OnActorThrown(Hand, HeldActor, HeldComponent, Bone):** extra throw-impulse notification.
- **OnHeldActorUseRequested(Hand, HeldActor, HeldComponent, Bone):** server-only validated activation hook; call your item interface here.

Observational events can run on clients; authority-gate gameplay changes. Initialize UI from getters. For a hand widget, attach to GetCharacterMesh/GetHandBone after OnReady, then use OnHandControlChanged and initial IsHandControlActive. For a held-item indicator also read GetHandState/GetHeldActor.

PhysicsHandsLeverComponent

- **GetValue():** normalized position -1..+1.
- **SetValue(Value) [Server]:** set position; refused while held.
- **IsOn(), IsHeld(), IsReady():** current switch/hold/initialization state.
- **GetLeverArm():** controlled primitive.
- **OnValueChanged(Value), OnHeldChanged(Held):** position/hold observations.
- **OnSwitchChanged(On), OnSwitchedOn(), OnSwitchedOff():** endpoint switch observations.

Events run on authority/observers. Trigger doors/puzzles only on authority and replicate their resulting state. Read getters for initial indicators.

PhysicsHandsSkeletalItemComponent

- **GetMesh():** controlled Skeletal Mesh.
- **IsReady(), IsHeld(), IsRagdoll():** current readiness/hold/mode.
- **GetGripCount():** accepted holding hand count.
- **RestoreOriginalSetup() [Server]:** restore at current location; refused while held.
- **OnGrabbed(Player, Hand, Bone):** server accepted-pickup gameplay event.
- **OnReleased(Player, Hand, Bone, Thrown):** server release/drop/throw/grip-loss event.

- **OnHeldChanged(Held, GripCount), OnRagdollChanged(Ragdoll):** persistent state observations on server/clients.

Restoration waits for the final grip. Earlier server pickup/release events are not historical replay. The original fully simulated setup can remain ragdolled even after restoration.

PhysicsHandsPluginProfile

- **ValidateSettings(Errors):** validate Body/Hands values.

The profile contains Body and Hands only; Input Actions/camera remain per-character.

Advanced world hinge helpers

These legacy helpers create a simulated world-fixed hinge; it can yield under contact. Use the kinematic lever component for touch-resistant movement.

- **MakeWorldLeverSetup(Hinge, Handle) [Server]:** capture original hinge/handle frames once before movement.
- **ConfigureWorldLever(Hinge, Handle, Setup, MaximumAngle=65, HoldStiffness=150000, HoldDamping=12000, HoldMaxTorque=5000000):** bool configuration result; Handle must be root.
- **SetLeverHeld(Hinge, Held) [Server]:** disable/enable holding motor for grip.
- **CaptureLeverPosition(Hinge) [Server]:** capture angle and resume holding after the last grip.

Replicate the captured Setup to peers before configuration; do not derive a late joiner's zero from a rotated handle. Retry configuration only until success. Route server grab/release through your own interface; query IsComponentHeld before capturing a final release. These helpers do not wire your item events automatically.

6 Quick troubleshooting

Input missing: check action assignments, active context, continuous hand triggers and Auto switching.

Setup invalid: check Character ownership, duplicate inherited components, bone names and physical bodies.

No contact reaction: enable the effective Profile/settings before play, inspect live diagnostics and actual arm contact.

Shaking/short reach: inspect collision overlap, joint limits and unreachable targets before raising forces.

Recovery blocked: check grips, speed, floor and capsule clearance.

Client effects differ: replicate persistent item/door state.

Moving surfaces differ: synchronize target motion.

Verify host and remote client, two different limbs on one target, mixed item/surface release, lever endpoints and a client joining during a hold. Do not run a competing

mesh/capsule/held-body transform driver. The plugin does not supply camera, inventory, damage, animation content or save-game logic.