FoxPhysics

Ragdoll, Forces, Constraints & Fake Physics for Blueprints

User Guide

Version 1.0

Alchemy Fox Studio

Supported Engine: UE 5.4, 5.6, 5.7, 5.8

Table of Contents

Installation

Option A: Via Unreal Engine Tab

1. Open the Epic Games Launcher and go to the Unreal Engine tab.
2. Search for FoxPhysics.
3. Click **Install to Engine**, select your Engine version, and click **Install**.
4. Open your project, go to **Edit > Plugins**, enable FoxPhysics, then restart the editor.

Option B: Via Fab Library

1. Open the Epic Games Launcher and go to **Fab**.
2. Open **My Library**, search for FoxPhysics and click on it.
3. Click **Install Plugin**, select your Engine version, and click **Install**.
4. Open your project, go to **Edit > Plugins**, enable FoxPhysics, then restart the editor.

Option C: Manual Installation

1. Copy the FoxPhysics/ folder into **YourProject/Plugins/**.
2. Right-click the .uproject file and select **Generate Visual Studio project files**.
3. Open the project, go to **Edit > Plugins**, enable FoxPhysics, then restart the editor.

All nodes appear under **FoxPhysics | <Category>** when you right-click in any Blueprint graph.

Supported Engine Version: **UE 5.4, 5.6, 5.7, 5.8**

Quick Start Tutorial

Every FoxPhysics node is a static Blueprint function — no setup, no initialization, no subsystem required (unless noted). Below are step-by-step examples showing how to use FoxPhysics in real project scenarios.

Example 1: Ragdoll on Death with Impact Impulse

Goal: Knock an enemy into a physics ragdoll at the moment of death, applying the killing blow's direction as an impulse.

Store the Hit Direction

In your TakeDamage function, calculate ImpulseDir = (HitLocation - AttackerLocation).Normalize() * DamageAmount * 80. Store in a local variable.

Enable Ragdoll

Call Enable Ragdoll with the Character reference. This enables physics, sets the Ragdoll collision profile, and disables the movement component.

Apply the Impulse

Immediately after, call Ragdoll with Impulse with BoneName="spine_02", Impulse=ImpulseDir, ImpulseStrength=1.0. The ragdoll is flung in the hit direction.

Detect Rest for Cleanup

Poll Is Ragdoll At Rest with VelocityThreshold=5.0 on a 0.5s timer. When true, start the dissolve sequence using FoxMaterial's Dissolve Actor.

Example 2: Fake Rope for a Drawbridge

Goal: Simulate a rope connecting two actors (drawbridge and anchor) without any physics asset.

Set Up Chain Points

Place 6 empty actors in the level as rope segment points: RopePoint_01 through RopePoint_06. The first is pinned to the wall anchor, the last is attached to the drawbridge edge.

Call Fake Rope Segment

On BeginPlay, call Fake Rope Segment with Points=array of the 6 actors, SegmentLength=80, Gravity=9.8, Iterations=8. Call it in the UFoxFakePhysicsComponent Tick.

Connect Spline Mesh

Use FoxSpline's Place Meshes Along Spline to render a chain mesh along the segment points each tick. This gives visual rope without a physics asset.

Play and Test

The rope droops realistically under gravity. Move the drawbridge actor — the rope follows and sags. Increase Iterations for stiffer rope, decrease for more droop.

Example 3: Area Impulse for Explosion

Goal: When a grenade explodes, launch all physics objects within 500 units away from the blast.

Get Physics Objects in Range

At the explosion point, call Get Overlapping Physics Actors with Origin=ExplosionLocation, Radius=500.

Apply Impulse Away From Point

For each returned actor, get its root UPrimitiveComponent and call Add Impulse Away From Point with Origin=ExplosionLocation, Strength=2000, Radius=500, Falloff=Linear.

Clamp Velocity

After applying impulse, call Clamp Physics Velocity with MaxSpeed=1200 so lightweight objects don't fly off to infinity.

Play and Test

Throw a grenade near barrels and crates — they scatter realistically with linear falloff. Objects closer to the blast fly faster than those at the edge of the radius.

Node Reference

Complete reference for all 41 FoxPhysics nodes organized by category. Each node includes a description, inputs, outputs, and a use case hint.

1. Ragdoll (6 Nodes)

1.1 Enable Ragdoll

Provides 1.1 Enable Ragdoll functionality.

Input	Type	Description
Character	ACharacter*	

(none)

Use Case: Ragdoll control — toggle physics simulation on a skeletal mesh.

1.2 Disable Ragdoll

Provides 1.2 Disable Ragdoll functionality.

Input	Type	Description
Character	ACharacter*	

(none)

Use Case: Ragdoll control — toggle physics simulation on a skeletal mesh.

1.3 Ragdoll With Impulse

Provides 1.3 Ragdoll With Impulse functionality.

Input	Type	Description
Character	ACharacter*	
BoneName	Name	
Impulse	Vector	
ImpulseStrength	Float	

(none)

Use Case: Ragdoll control — toggle physics simulation on a skeletal mesh.

1.4 Is Ragdoll At Rest

Provides 1.4 Is Ragdoll At Rest functionality.

Input	Type	Description
Character	ACharacter*	
VelocityThreshold	Float	

Output	Type	Description
Return Value	Bool	

Use Case: Ragdoll control — toggle physics simulation on a skeletal mesh.

1.5 Get Ragdoll Root Location

Provides 1.5 Get Ragdoll Root Location functionality.

Input	Type	Description
Character	ACharacter*	
RootBoneName	Name	

Output	Type	Description
Return Value	Vector	

Use Case: Ragdoll control — toggle physics simulation on a skeletal mesh.

1.6 Is Ragdoll Face Up

Provides 1.6 Is Ragdoll Face Up functionality.

Input	Type	Description
Character	ACharacter*	
PelvisBoneName	Name	

Output	Type	Description
Return Value	Bool	

Use Case: Ragdoll control — toggle physics simulation on a skeletal mesh.

2. Force (5 Nodes)

2.1 Add Force Towards Point

Provides 2.1 Add Force Towards Point functionality.

Input	Type	Description
Component	Primitive Component	Component reference
TargetPoint	Vector	
Strength	Float	
FalloffRadius	Float	Radius in units

(none)

Use Case: Force application — apply physics forces or impulses to an actor.

2.2 Add Impulse Away From Point

Provides 2.2 Add Impulse Away From Point functionality.

Input	Type	Description
Component	Primitive Component	Component reference
Origin	Vector	
Strength	Float	
Radius	Float	Radius in units
Falloff	EFoxImpulseFalloff	

(none)

Use Case: Force application — apply physics forces or impulses to an actor.

2.3 Apply Impulse At Bone

Provides 2.3 Apply Impulse At Bone functionality.

Input	Type	Description
Mesh	USkeletalMeshComponent*	
BoneName	Name	
Impulse	Vector	

(none)

Use Case: Force application — apply physics forces or impulses to an actor.

2.4 Add Force Along Spline

Provides 2.4 Add Force Along Spline functionality.

Input	Type	Description
Component	Primitive Component	Component reference
Spline	USplineComponent*	
Strength	Float	

(none)

Use Case: Force application — apply physics forces or impulses to an actor.

2.5 Apply Impulse In Direction

Provides 2.5 Apply Impulse In Direction functionality.

Input	Type	Description
Component	Primitive Component	Component reference
Direction	Vector	Direction vector
Strength	Float	

(none)

Use Case: Force application — apply physics forces or impulses to an actor.

3. Constraints (4 Nodes)

3.1 Create Constraint Simple

Provides 3.1 Create Constraint Simple functionality.

Input	Type	Description
ComponentA	Primitive Component	Component reference
ComponentB	Primitive Component	Component reference
ConstraintType	EFoxConstraintType	

Output	Type	Description
Return Value	UPhysicsConstraintComponent*	

Use Case: Constraint helper — create or break runtime physics constraints.

3.2 Create Spring Between

Provides 3.2 Create Spring Between functionality.

Input	Type	Description
ComponentA	Primitive Component	Component reference
ComponentB	Primitive Component	Component reference
Stiffness	Float	
Damping	Float	

Output	Type	Description
Return Value	UPhysicsConstraintComponent*	

Use Case: Constraint helper — create or break runtime physics constraints.

3.3 Break Constraint

Provides 3.3 Break Constraint functionality.

Input	Type	Description
Constraint	UPhysicsConstraintComponent*	

(none)

Use Case: Constraint helper — create or break runtime physics constraints.

3.4 Weld Actors

Provides 3.4 Weld Actors functionality.

Input	Type	Description
ActorA	Actor Reference	Target actor reference
ActorB	Actor Reference	Target actor reference

Output	Type	Description
Return Value	UPhysicsConstraintComponent*	

Use Case: Constraint helper — create or break runtime physics constraints.

4. State (4 Nodes)

4.1 Set Physics Enabled Batch

Provides 4.1 Set Physics Enabled Batch functionality.

Input	Type	Description
Components	Array of Primitive Component	Component reference
bSimulate	Bool	Boolean flag

(none)

Use Case: State mutator — change actor or runtime state.

4.2 Freeze Physics

Provides 4.2 Freeze Physics functionality.

Input	Type	Description
Component	Primitive Component	Component reference

(none)

Use Case: State mutator — change actor or runtime state.

4.3 Get Physics Info

Provides 4.3 Get Physics Info functionality.

Input	Type	Description
Component	Primitive Component	Component reference

Output	Type	Description
Velocity	Vector	
AngularVelocity	Vector	
Mass	Float	
Speed	Float	Speed value

Use Case: State mutator — change actor or runtime state.

4.4 Set Mass Override

Provides 4.4 Set Mass Override functionality.

Input	Type	Description
Component	Primitive Component	Component reference
MassInKg	Float	

(none)

Use Case: State mutator — change actor or runtime state.

5. Utility (6 Nodes)

5.1 Bounce Velocity

Provides 5.1 Bounce Velocity functionality.

Input	Type	Description
InVelocity	Vector	
SurfaceNormal	Vector	
Bounciness	Float	

Output	Type	Description
Return Value	Vector	

Use Case: Utility helper — supporting function for the toolkit.

5.2 Get Overlapping Physics Actors

Provides 5.2 Get Overlapping Physics Actors functionality.

Input	Type	Description
WorldContext	Object Reference	
Origin	Vector	
Radius	Float	Radius in units

Output	Type	Description
Return Value	Array of Actor Reference	

Use Case: Utility helper — supporting function for the toolkit.

5.3 Get Physics Actors In Path

Provides 5.3 Get Physics Actors In Path functionality.

Input	Type	Description
WorldContext	Object Reference	
Start	Vector	
End	Vector	
Radius	Float	Radius in units

Output	Type	Description
Return Value	Array of Actor Reference	

Use Case: Utility helper — supporting function for the toolkit.

5.4 Launch Actor

Provides 5.4 Launch Actor functionality.

Input	Type	Description
Actor	Actor Reference	Target actor reference
LaunchVelocity	Vector	
bOverrideXY	Bool	Boolean flag
bOverrideZ	Bool	Boolean flag

(none)

Use Case: Utility helper — supporting function for the toolkit.

5.5 Slow Physics Actor

Provides 5.5 Slow Physics Actor functionality.

Input	Type	Description
Component	Primitive Component	Component reference
DragPercent	Float	

(none)

Use Case: Utility helper — supporting function for the toolkit.

5.6 Clamp Physics Velocity

Provides 5.6 Clamp Physics Velocity functionality.

Input	Type	Description
Component	Primitive Component	Component reference
MaxSpeed	Float	Speed value

(none)

Use Case: Utility helper — supporting function for the toolkit.

6. Fake (16 Nodes)

6.1 Setup Hinge Joint

Provides 6.1 Setup Hinge Joint functionality.

Input	Type	Description
ActorA	Actor Reference	Target actor reference
ActorB	Actor Reference	Target actor reference
Pivot	Vector	
HingeAxis	Vector	
AngleLimit	Float	

(none)

Use Case: Fake physics setup — configure component-based mechanical setups.

6.2 Setup Ball Joint

Provides 6.2 Setup Ball Joint functionality.

Input	Type	Description
ActorA	Actor Reference	Target actor reference
ActorB	Actor Reference	Target actor reference
Pivot	Vector	
AngleLimit	Float	

(none)

Use Case: Fake physics setup — configure component-based mechanical setups.

6.3 Setup Tow Coupling

Provides 6.3 Setup Tow Coupling functionality.

Input	Type	Description
Leader	Actor Reference	
Follower	Actor Reference	
TowDistance	Float	
RotationSpeed	Float	World-space rotation

(none)

Use Case: Fake physics setup — configure component-based mechanical setups.

6.4 Setup Tow Chain

Provides 6.4 Setup Tow Chain functionality.

Input	Type	Description
Leader	Actor Reference	
Follower	Actor Reference	
ChainLength	Float	
SlackAmount	Float	

(none)

Use Case: Fake physics setup — configure component-based mechanical setups.

6.5 Setup Wheel Axle

Provides 6.5 Setup Wheel Axle functionality.

Input	Type	Description
WheelMesh	UStaticMeshComponent*	
ParentActor	Actor Reference	Target actor reference
WheelRadius	Float	Radius in units

(none)

Use Case: Fake physics setup — configure component-based mechanical setups.

6.6 Setup Rope Segment

Provides 6.6 Setup Rope Segment functionality.

Input	Type	Description
Points	Array of Actor Reference	
SegmentLength	Float	
Gravity	Float	
Iterations	Int	

(none)

Use Case: Fake physics setup — configure component-based mechanical setups.

6.7 Setup Chain Link

Provides 6.7 Setup Chain Link functionality.

Input	Type	Description
Points	Array of Actor Reference	
SegmentLength	Float	
Gravity	Float	
Iterations	Int	
Stiffness	Float	

(none)

Use Case: Fake physics setup — configure component-based mechanical setups.

6.8 Setup Pulley

Provides 6.8 Setup Pulley functionality.

Input	Type	Description
EndA	Actor Reference	
EndB	Actor Reference	
PulleyPoint	Vector	
RopeLength	Float	

(none)

Use Case: Fake physics setup — configure component-based mechanical setups.

6.9 Setup Catapult Arm

Provides 6.9 Setup Catapult Arm functionality.

Input	Type	Description
Arm	Actor Reference	
PivotPoint	Vector	
ArmLength	Float	
LoadMass	Float	

(none)

Use Case: Fake physics setup — configure component-based mechanical setups.

6.10 Load Catapult
Provides 6.10 Load Catapult functionality.

(none)

(none)

Use Case: Fake physics setup — configure component-based mechanical setups.

6.11 Release Catapult
Provides 6.11 Release Catapult functionality.

(none)

Output	Type	Description
Return Value	Vector	

Use Case: Fake physics setup — configure component-based mechanical setups.

6.12 Setup Suspension
Provides 6.12 Setup Suspension functionality.

Input	Type	Description
Actor	Actor Reference	Target actor reference
RestLength	Float	
SpringStrength	Float	
InDamping	Float	

(none)

Use Case: Fake physics setup — configure component-based mechanical setups.

6.13 Setup Trailer Follow
Provides 6.13 Setup Trailer Follow functionality.

Input	Type	Description
Leader	Actor Reference	
Followers	Array of Actor Reference	
SegmentDistance	Float	
TurnSmoothing	Float	

(none)

Use Case: Fake physics setup — configure component-based mechanical setups.

6.14 Setup Breakable Joint

Provides 6.14 Setup Breakable Joint functionality.

Input	Type	Description
ActorA	Actor Reference	Target actor reference
ActorB	Actor Reference	Target actor reference
Pivot	Vector	
HingeAxis	Vector	
AngleLimit	Float	
BreakForce	Float	

(none)

Use Case: Fake physics setup — configure component-based mechanical setups.

6.15 Get Or Create Fake Physics

Provides 6.15 Get Or Create Fake Physics functionality.

Input	Type	Description
Actor	Actor Reference	Target actor reference

Output	Type	Description
Return Value	UFoxFakePhysicsComponent*	

Use Case: Fake physics setup — configure component-based mechanical setups.

6.16 Fake Load Weight

Provides 6.16 Fake Load Weight functionality.

Input	Type	Description
Vehicle	Actor Reference	
CargoMass	Float	
BaseSpeed	Float	Speed value
BaseAcceleration	Float	
SpeedMultiplier	Float	Speed value
AccelerationMultiplier	Float	

Output	Type	Description
AdjustedSpeed	Float	Speed value
AdjustedAcceleration	Float	

Use Case: Fake physics setup — configure component-based mechanical setups.

Implementation Notes

- Fake Physics uses UFoxFakePhysicsComponent (TickComponent) for per-frame updates.
- Rope/Chain uses Verlet integration with configurable iteration count.
- Cart systems use atan2-based steering math.
- All fake physics can be paused/resumed via component enable/disable.
- Ragdoll nodes store pre-ragdoll state (collision profile, movement mode) in a static TMap for restoration.