NEW · V1.0

Game Juice

20 drop-in C++ components for instant game feel. Camera shake, hitstop, slow-mo, rumble, screen flash, coyote time, hit reactions, and more. Drop a component on your character, call one function, ship it.

v1.0.0 | UE 5.4 — 5.7 | 100% C++ | Multiplayer Ready | Blueprint Friendly

GETTING STARTED

- 📦 Installation
- 🚀 Quick Start (3 min)
- 💡 How It Works

COMPONENTS

- 📷 Camera (×5)
- ⏱️ Time (×3)
- 🎮 Rumble & Input (×2)
- ✨ Camera Effects (×4)
- 🏃 Movement Feel (×3)
- 💥 Hit Feedback (×3)

EXAMPLES

- 🎯 Add juice to a shooter
- 🦘 Better platformer feel
- ⚔️ Punchy melee combat

ADVANCED

- 🌐 Multiplayer
- 🎛️ Presets & Curves
- 🎥 Custom Camera Setup

HELP

- 🔧 Troubleshooting
- ❓ FAQ
- 💬 Support

// 01 / SETUP

📦 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. Pick the ZIP that matches your engine version.

1 Download & Extract

After purchasing on FAB, download the ZIP file matching your engine version (e.g. `GameJuice_UE5.7.zip`). **Right-click → Extract All.**

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

2 Copy to Plugins Folder

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

```
📁 YourProject/
├── 📁 Content/
├── 📁 Config/
├── 📁 Plugins/ ← Create this if missing
│   └── 📁 GameJuice/
│       ├── 📁 Source/
│       └── 📄 GameJuice.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.

> 💡 If you don't see the "Rebuild" prompt, right-click your `.uproject` → **Generate Visual Studio project files**, then open the `.sln` and build in Visual Studio.

4 Verify

Create any Actor Blueprint, click **Add Component**, and search `Juice`. You should see all 20 components in the dropdown:

- `Juice Camera Shake Component`
- `Juice Hitstop Component`
- `Juice Slow Mo Component`
- ...and 17 more

🎉 If you see them, you're ready!

// 02 / FIRST STEPS

🚀 Quick Start (3 Minutes)

> 🎯 **Goal:** Add screen-shake-on-hit, hitstop-on-hit, and gamepad-rumble-on-hit to your character. Three lines of Blueprint, no C++.

1 Add Components to Your Character

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

Click **Add Component** and add these three:

- 🔍 Search `Juice Camera Shake` → add it
- 🔍 Search `Juice Hitstop` → add it
- 🔍 Search `Juice Gamepad Rumble` → add it

That's all the setup. Each component picks reasonable defaults automatically.

> ⚠️ **Important:** After adding each component, select it in the Components panel and make sure `Auto Activate` is set to **True** in the Details panel. Components ship with Auto Activate disabled by default — enable it or the component won't do anything at runtime.

2 Trigger Them on Hit

Find (or create) the event where your character takes damage. Common name: `Event Any Damage` or your custom `OnHit` event.

From that event, drag three function nodes:

- `Shake` → on the Camera Shake component (pick preset `Hit`)
- `Trigger Hitstop` → on the Hitstop component
- `Rumble` → on the Gamepad Rumble component (pick preset `Heavy`)

> 💡 Each component exposes simple functions like `Shake`, `Trigger Hitstop`, `Rumble`, `Flash`, etc. Just drag from the component reference and type the verb.

3 Press Play and Take a Hit

Compile, save, hit Play. Take damage. You should see:

- ✅ Camera shakes briefly
- ✅ Game freezes for 1/8 of a second (hitstop)
- ✅ Gamepad rumbles (if using a controller)

🎉 **Done.** That's your character now feeling 10x better. Repeat for whatever events you want to juice up.

// 03 / MENTAL MODEL

💡 How It Works

Game Juice is built on a simple pattern. Every component follows the same three rules, so once you learn one, you know them all.

1. Add the component

Click **Add Component** on any actor and pick the juice you want. Then select the component and **set Auto Activate to True** in the Details panel.

2. Configure in the Details panel

Every component has tunable parameters in the **Details** panel. Click the component, scroll the Details panel, tweak intensity / duration / curves to taste.

3. Call the function from Blueprint

Each component exposes one or two simple functions. Call them from any event. Done.

> 💡 **Every component has a** `Global Intensity Scale` **property** (0 – 3). If you want to tune ALL juice up or down at once (e.g. accessibility setting), drive this from a slider.

The base pattern

Pseudo-Blueprint, but the real thing looks identical:

```
OnHit (Event)
  ↳ Shake Component       → Shake (Preset: Hit)
  ↳ Hitstop Component     → Trigger Hitstop
  ↳ Rumble Component      → Rumble (Preset: Heavy)
  ↳ Flash Component       → Flash (Preset: Damage)
  ↳ Vignette Component    → Set Health Percent (CurrentHP / MaxHP)
```

That's it. No event dispatchers, no manual networking, no Niagara setup. The components handle replication, dedicated server cosmetic skipping, and cleanup automatically.

// 04 / COMPONENTS

📷 Camera (×5)

Five camera-driven juice components. All work in 1st person, 3rd person, or any custom camera setup.

🎬 Juice Camera Shake `cosmetic · multiplayer-aware`

Server-multicast camera shake with per-observer distance falloff. Built-in presets: Explosion, Footstep, Hit, Recoil, Earthquake, plus Custom for your own UCameraShakeBase classes.

Functions

FUNCTION	WHAT IT DOES
`Shake`	Play a shake on the local camera (pass a preset)
`Shake At Location`	Play a shake on EVERY player within radius (with falloff)

Properties

PROPERTY	WHAT IT DOES
`Custom Shake Class`	If preset = Custom, use this UCameraShakeBase class
`Falloff Curve`	Custom distance falloff curve (optional)
`Default Radius`	Default falloff radius for ShakeAtLocation
`Global Intensity Scale`	Multiplier applied to all shakes (0 – 3)

🔍 Juice Camera FOV Punch `cosmetic · local`

One-shot field-of-view kick with curve-driven decay. Perfect for sprint-start, dash, or weapon zoom.

Functions

FUNCTION	WHAT IT DOES
`Punch`	Trigger an FOV punch at default scale
`Punch Explicit`	Custom delta degrees, duration, and curve override

🌀 Juice Camera Tilt `cosmetic · local`

Continuous camera roll based on movement, turning, velocity, or manual control. Frame-rate independent.

Modes

MODE	EFFECT
`OnMovement`	Tilts opposite to lateral input (strafe lean)
`OnTurning`	Tilts proportional to mouse/stick yaw delta
`OnVelocity`	Tilts based on character velocity vector
`Manual`	You drive the tilt yourself via SetManualTilt

👁️ Juice Camera Lookahead `cosmetic · local`

Celeste-style velocity-based camera offset — camera "looks ahead" of the player's movement direction. Best for 1st person / 2D side-scrollers.

💨 Juice Camera Impulse `cosmetic · local`

One-shot directional camera push with exponential decay. Use for explosions, big hits, environmental impacts.

Functions

FUNCTION	WHAT IT DOES
`Apply Impulse`	Push camera in a world-space direction

⏱️ Time (×3)

Time manipulation components. All three handle the dilation = 0 edge case correctly using the engine's real-time core ticker (world timers don't fire when time stops).

⏸️ Juice Hitstop `multiplayer · server-multicast`

Brief time-freeze on impact. The classic Smash Bros / Super Meat Boy / Hyper Light Drifter feel.

Functions

FUNCTION	WHAT IT DOES
`Trigger Hitstop`	Run a hitstop with default duration & severity
`Trigger Hitstop Explicit`	Custom duration and severity (0 – 1)

🐢 Juice Slow Mo `multiplayer · server-multicast`

Sustained slow motion with smooth ramp-in / hold / ramp-out. Optional curve-driven blend for non-linear feel.

Functions

FUNCTION	WHAT IT DOES
`Start Slow Mo`	Default slow-mo with target dilation, blend in/hold/out
`Stop Slow Mo`	Stop early (works on clients via server RPC)

⏰ Juice Time Dilation `multiplayer · scoped`

Smooth time-dilation blend with two scopes:

- **Global** — slows down the entire world (uses WorldSettings.TimeDilation)

- **Owner Only** — slows down just this actor (uses CustomTimeDilation)

Use Owner Only for "bullet time on enemy" or speed buffs without affecting the player.

🎮 Rumble & Input (×2)

Force feedback and PS5 DualSense adaptive trigger support.

📳 Juice Gamepad Rumble `cosmetic · local`

Force feedback rumble with five built-in presets and a fallback dynamic rumble path.

Presets

PRESET	USE FOR
`Light`	UI clicks, light footsteps, item pickup
`Medium`	Standard hit, moderate impact
`Heavy`	Big hit, explosion, boss attack
`Recoil`	Quick punchy weapon recoil
`Heartbeat`	Low-frequency tense pulse (boss intro, low HP)

> 💡 You can assign a custom `UForceFeedbackEffect` asset to any preset slot in the Details panel. If left empty, the dynamic fallback path is used (works automatically).

🕹️ Juice Adaptive Trigger `cosmetic · local · PS5 only`

PS5 DualSense trigger effects via `UInputDeviceSubsystem`. On Xbox / PC the calls become silent no-ops, so it's safe to wire up across all platforms.

Effect types

EFFECT	USE FOR
`Bow`	Drawing a bow, charging an attack
`Gun`	Pulling a heavy trigger
`Brake`	Vehicle brake / heavy lever
`Vibration`	Engine running, machinery
`Wall`	Hard stop / wall (great for blocked weapons)

✨ Camera Effects (×4)

Screen-space and post-process effects that drive your existing camera. No separate Niagara systems, no extra rendering passes.

Juice Screen Flash `cosmetic · local · UMG`

Full-screen color flash with linear fade. Built-in presets: Damage (red), Heal (green), Pickup (yellow), Death (white), Custom (any color).

Provides a runtime-built UMG widget out of the box (no setup needed). For a custom design, assign your own `UUserWidget` class.

Juice Speedlines `cosmetic · local · post-process`

Anime-style speedlines via post-process material injection.

> ⚠️ **Requires:** a post-process material with an `Intensity` scalar parameter (parameter name configurable). Drop your material in the Details panel; the component creates a Material Instance Dynamic and drives it.

Juice Damage Vignette `cosmetic · local · post-process`

HP-driven vignette that pulses red when health is low. Just call `Set Health Percent` with your current/max HP each frame (or on damage).

🎞️ Juice Chromatic Aberration

`cosmetic · local · post-process`

Impact-driven RGB fringing on the camera's post-process. Triggered with one call, decays automatically.

🏃 Movement Feel (×3)

Industry-standard feel improvements that make platformers and action games not feel sluggish.

🦘 Juice Coyote Time `gameplay · local`

Lets the player jump for a configurable grace period after walking off a ledge. Distinguishes "walked off" from "jumped off" using vertical velocity heuristic.

Functions

FUNCTION	WHAT IT DOES
`Try Consume Coyote Jump`	Returns true if the player can still jump (and consumes the window)

🎯 Juice Jump Buffer `gameplay · local`

Remembers a jump press for a brief window before landing. If the player presses jump 100ms before touching the ground, they jump on landing — no missed inputs.

Functions

FUNCTION	WHAT IT DOES
`Buffer Jump Input`	Call this on jump press
`Try Consume Buffered Jump`	Call on landing — returns true if a buffered jump should fire

⏳ Juice Anticipation `gameplay · local`

Animation-principle scheduler: **Windup** → **Action** → **Recovery**. Use for melee swings, charging attacks, big abilities. Fires three delegates so you can hook anim notifies, sounds, and gameplay logic to each phase.

Events (Delegates)

EVENT	WHEN IT FIRES
`OnStarted`	Sequence begins (good place to play windup anim)
`OnAction`	Windup ended, action moment (do damage / spawn projectile here)
`OnRecovered`	Recovery ended (player can act again)
`OnCancelled`	Sequence was cancelled mid-flight

💥 Hit Feedback (×3)

Three components that make taking and dealing hits feel meaty.

👊 Juice Hit Reaction `multiplayer · server-multicast`

Knockback + 4-direction hit event. Server applies LaunchCharacter (movement replicates to clients automatically). Fires `OnHitReceived` with hit direction enum (Front / Back / Left / Right) so you can play correct hit animation.

🟣 Juice Material Flash `multiplayer · server-multicast`

Flashes ALL of a mesh's materials white (or your color) on hit. Drives a `FlashAmount` scalar parameter on Material Instance Dynamics.

> ⚠️ **Requires:** Your mesh's materials must have a `FlashAmount` scalar parameter (and optionally `FlashColor` vector). The component creates MIDs automatically at BeginPlay.

🔊 Juice Audio Pitch Variation `cosmetic · local`

Plays a USoundBase with randomized pitch (e.g. 0.95 – 1.05) and randomized volume. Stops "machine gun click syndrome" where the same sound plays identically each time.

// 05 / RECIPES

🎯 Example: Add juice to a shooter

You have a basic shooter. Player can shoot, take damage. Goal: make every shot AND every hit feel impactful.

Components to add to player character

- `Juice Camera Shake` — for shooting + getting hit
- `Juice Camera FOV Punch` — for ADS / sprint
- `Juice Hitstop` — for landing a kill
- `Juice Gamepad Rumble` — for shooting + getting hit
- `Juice Screen Flash` — for taking damage
- `Juice Damage Vignette` — for low HP
- `Juice Chromatic Aberration` — for big impacts

Wire it up

```
OnFireWeapon (Event)
  ↳ Camera Shake → Shake (Preset: Recoil)
  ↳ Gamepad Rumble → Rumble (Preset: Recoil)
  ↳ Camera FOV Punch → Punch (small kick, 0.05s)

OnHitEnemy (Event)  // your weapon hit confirmation
  ↳ Hitstop → Trigger Hitstop  (only if killing blow)

OnTakeDamage (Event)
  ↳ Camera Shake → Shake (Preset: Hit)
  ↳ Gamepad Rumble → Rumble (Preset: Heavy)
  ↳ Screen Flash → Flash (Preset: Damage)
  ↳ Chromatic Aberration → Trigger
  ↳ Damage Vignette → Set Health Percent (CurrentHP / MaxHP)
```

That's it. Five minutes of wiring, and your shooter now feels professional.

🦘 Example: Better platformer feel

Add the three "feels good" tricks every modern platformer uses.

Components on player character

- `Juice Coyote Time` — forgiving ledge jumps
- `Juice Jump Buffer` — buffered jump input
- `Juice Camera Lookahead` — camera leads movement

Wire it up

```
OnJumpPressed (Input event)
  ├→ Jump Buffer → Buffer Jump Input
  └↳ Branch: Is Falling?
       True  → Coyote Time → Try Consume Coyote Jump → If true, do Jump
       False → Do Jump

OnLanded (Character event)
  └↳ Jump Buffer → Try Consume Buffered Jump → If true, do Jump immediately
```

Now your character can jump up to 0.15s after walking off a ledge, AND if the player presses jump just before landing, they'll auto-jump on landing. The combination is what makes Celeste / Hollow Knight feel responsive.

⚔️ Example: Punchy melee combat

Make every hit land like Hollow Knight or Hyper Light Drifter.

Components on attacker

- `Juice Anticipation` — windup → strike → recovery scheduler
- `Juice Hitstop` — freeze on hit
- `Juice Camera Shake` — small shake on hit
- `Juice Audio Pitch Variation` — varied hit sounds

Components on receiver (enemy)

- `Juice Hit Reaction` — knockback + direction event
- `Juice Material Flash` — white flash on hit

Wire it up

```
OnAttackPressed
  ↳ Anticipation → Begin Explicit (windup: 0.15s, recovery: 0.4s)

Anticipation.OnAction
  ↳ Trace forward, find enemy
       ↳ Apply Damage to enemy
       ↳ Material Flash (on enemy) → Flash
       ↳ Hit Reaction (on enemy) → Apply Hit
       ↳ Hitstop (on player) → Trigger Hitstop
       ↳ Camera Shake (on player) → Shake (Preset: Hit)
       ↳ Audio Pitch Variation → Play Sound
```

The combination of **Anticipation** phases + **Hitstop** on impact + **Material Flash** on the receiver is what makes melee feel weighty.

// 06 / ADVANCED

🌐 Multiplayer

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

Game Juice components fall into two networking categories. You don't need to think about it — pick the component, call the function, replication just works.

Cosmetic / Local components

These run only on the local client. Calling them from the server has no effect on remote players.

- Camera FOV Punch, Tilt, Lookahead, Impulse
- Gamepad Rumble, Adaptive Trigger
- Screen Flash, Speedlines, Damage Vignette, Chromatic Aberration
- Coyote Time, Jump Buffer, Anticipation
- Audio Pitch Variation

World-event / Server-multicast components

These replicate via Server RPC → Multicast. All players see the effect (with appropriate falloff).

- Camera Shake (with per-observer distance falloff)
- Hitstop, Slow Mo, Time Dilation (Global scope)
- Hit Reaction (knockback applied authority-side)
- Material Flash

> ⚠️ **Dedicated server is automatically detected.** Cosmetic effects (camera, post-process, rumble) skip on dedicated server to save CPU. Server-side effects like Hit Reaction's knockback still apply.

🎛️ Presets & Curves

Most components ship with built-in presets. You can also drive any effect with a `UCurveFloat` for full custom control.

Where presets live

Each component's **Details** panel exposes its presets as enum dropdowns. For example, on the Camera Shake component:

PROPERTY	WHERE
`Default Preset`	Used when calling `Shake()` with no arguments
`Custom Shake Class`	If preset = Custom, use this UCameraShakeBase
`Falloff Curve`	Override default falloff for ShakeAtLocation

Custom curves

You can drive most effects with a `UCurveFloat` asset. Right-click in Content Browser → Miscellaneous → Curve. Set X = 0..1 (normalized time), Y = 0..1 (effect alpha).

> 💡 **Bell-shaped curve** (0 → 1 → 1 → 0) gives ramp-in, hold, ramp-out. **Ease-out curve** (1 → 0 with curve) gives strong start that fades. Pick what fits the moment.

📹 Custom Camera Setup

By default, camera-driven components auto-resolve the first `UCameraComponent` on the owning actor. If you have a non-standard setup, you can assign the target manually.

Auto-resolution

Components that drive a camera (FOV Punch, Tilt, Lookahead, Impulse, Speedlines, Damage Vignette, Chromatic Aberration) call `FindComponentByClass<UCameraComponent>` on their owner at BeginPlay.

Manual assignment

If your camera lives somewhere else (e.g. on a child actor, or you want to drive a specific camera in a multi-camera setup), call `Set Target Camera` on the component at runtime, passing the camera reference.

> 💡 If you change the camera mid-effect, the component automatically restores the OLD camera to its base state before swapping. No leftover offsets.

Spring arm / 3rd person

Camera offset components (Lookahead, Impulse) handle spring arm hierarchies correctly — they convert offsets to the camera's parent space before applying. Works in 3rd person without modification.

// 07 / HELP

🔧 Troubleshooting

Component doesn't do anything / effect doesn't trigger

1. **Check that `Auto Activate` is set to True** on the component in the Details panel. This is the #1 reason effects don't fire. Components ship with Auto Activate disabled by default.
2. Check that `Global Intensity Scale` is greater than 0 (default is 1.0).
3. Make sure you're actually calling the component's function (e.g. `Shake`, `Rumble`) from your event graph.

I don't see any of the Juice components in the Add Component menu

1. Did you copy the plugin to `YourProject/Plugins/GameJuice/`?
2. Did you compile? (Open `.uproject`, click Yes when asked to rebuild.)
3. Check **Edit → Plugins → Installed** for "Game Juice". If disabled, enable it and restart the editor.
4. Make sure your engine version matches the ZIP you extracted (UE 5.4 plugin won't load in UE 5.6, etc.).

Camera Shake works in standalone but not in multiplayer

1. Make sure you're calling `Shake At Location` from the **server**. Camera Shake's RPC fires Server → Multicast.
2. If calling from a client, the component automatically forwards to the server. But if your owning actor isn't replicated, the RPC won't reach the server.
3. Owning actor must have `Replicates = true`.

Hitstop freezes the dedicated server, breaking other players

This was an early-version bug. **v1.0+ skips dedicated server automatically**. If you see this, double-check you're on the latest version.

Speedlines don't show up

1. Did you assign a `Post Process Material` to the component?
2. Does that material have a scalar parameter named `Intensity` (or whatever you set in `Intensity Parameter Name`)?

3. Is the post-process actually rendering? (Test by directly setting your camera's post-process intensity high to confirm post-processing works.)

Material Flash doesn't flash

1. Does your mesh's material have a scalar parameter called `FlashAmount`?
2. Try the simplest possible setup: in the material, plug a Lerp into Emissive → A = your normal output, B = white, Alpha = the FlashAmount scalar parameter.
3. The component creates MIDs at BeginPlay. If you swap the mesh's material at runtime, call `Set Target Mesh` again to refresh.

Coyote Time doesn't trigger

1. The component hooks `ACharacter::MovementModeChangedDelegate`. Make sure your owner is an `ACharacter` (not an `APawn`).
2. Coyote window only opens on transition to MOVE_Falling with low vertical velocity (≤ 50 cm/s default), to distinguish "walked off" from "jumped off".

Adaptive Trigger doesn't do anything

It only works on PS5 with a connected DualSense controller. On PC / Xbox the calls are silent no-ops by design (so you can wire it cross-platform without checks).

Compilation errors after installing

1. Make sure you extracted the ZIP for your exact engine version.
2. Delete `Binaries`, `Intermediate`, and `Saved` folders, then right-click `.uproject` → **Generate Visual Studio project files** → rebuild.
3. Make sure UMG plugin is enabled (Edit → Plugins → UMG).

❓ FAQ

Do I need to know C++? +

Will it work in 1st person AND 3rd person games? +

Does it work without Niagara / Enhanced Input plugins? +

Can I use it with GAS (Gameplay Ability System)? +

How does it differ from custom UCameraShake / Force Feedback assets I could make myself? +

Can I modify the source code? +

Will future engine versions be supported? +

Is it compatible with my custom GameMode / PlayerController / Character setup? +

What happens on a dedicated server? +

Support

Email

support@adrenalinegames.pl

Response within 48 hours

AI Assistant

+48 918 918 005

Knows the docs, answers right away

Discord

Join Discord

Live chat & community

Website

adrenalinegames.pl

All plugins & docs

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

Game Juice

Version 1.0.0 · Made with ❤️ by Erak Dev