# 👻 Ultimate Horror Framework

Complete Horror Game Framework for Unreal Engine 5

**v2.0 — 123 C++ Classes** **UE 5.6 / 5.7** **Multiplayer Ready**

**100% Blueprint Friendly**

---

## Quick Start — Working Horror Game in 5 Minutes

Zero Blueprint work required. Set 4 classes in your GameMode and press Play. Everything auto-creates: HUD, widgets, full player movement with WASD + mouse look, sprint, crouch, jump, interact, flashlight, lean, inventory — all input bindings are built-in.

### 1 Enable the Plugin

Go to **Edit** → **Plugins**, search for **"Horror Framework"**, enable it, and restart the editor.

### 2 Set Up GameMode

Create or open your GameMode Blueprint. Set these 4 classes in the Details Panel:

| 📋 GameMode — Details Panel (Classes) | |
|---|---|
| Default Pawn Class | `HF_HorrorCharacter` |
| Player Controller Class | `HF_HorrorPlayerController` |
| HUD Class | `HF_HorrorHUD` |
| Game State Class | `HF_GameState` |

### 3 Hit Play!

That's it. You now have all of this working automatically:

### 🏃 Movement

WASD, mouse look, sprint (Shift), crouch (C), jump (Space). Gamepad support included.

### ❤️ Health System

HP with damage, healing, regen, invincibility frames, death, and revive.

### 😱 Fear & Sanity

Sanity drains in darkness. Fear spikes from scares. Visual effects intensify automatically.

### 🔦 Flashlight

Toggle with F key. Battery drains over time. Flickers when low. Auto-created SpotLight.

### 🖐️ Interaction

Line trace from camera. "[E] Open Door" prompts on HUD. Works with all built-in interactables.

### 🎒 Inventory

12-slot system with stacking, weight, equip/unequip. Toggle with Tab.

### 🖥️ Full HUD

HP / Sanity / Stamina / Battery bars, crosshair, interaction prompt, damage vignette.

### 📢 Noise System

Walk/run/crouch produce different noise levels. AI hears you through AwarenessComponent.

## 4 Add an Enemy (Optional)

Create a Character Blueprint. Set `HF_HorrorAIController` as AI Controller Class. Drop it in the level. The AI will automatically detect the player via sight + hearing, chase when awareness is high enough, patrol between points, and search hiding spots.

## 5 Customize Controls (Optional)

**Input works out of the box!** The PlayerController auto-creates all InputActions, MappingContext, and key bindings on BeginPlay. No manual Input setup required.

Want to change keys? Open your PlayerController Blueprint → Details → `Horror | Input | Keys`:

| 📋 HF_HorrorPlayerController — Key Bindings | |
|---|---|
| Move Forward / Back / Left / Right | `W / S / A / D` |
| Sprint | `Left Shift` |
| Crouch | `C` |
| Jump | `Space Bar` |
| Interact | `E` |
| Flashlight | `F` |
| Inventory | `Tab` |
| Lean Left / Right | `Q / R` |
| Pause | `Escape` |

## 🧩 All 16 Default Components on HF_HorrorCharacter

These are auto-created in the constructor. You get them all instantly when you use `HF_HorrorCharacter` as your Default Pawn Class.

| # | COMPONENT | WHAT IT DOES |
|---|---|---|
| 1 | CameraBoom (SpringArm) | Camera mount. First-person by default (TargetArmLength = 0). |
| 2 | FirstPersonCamera | Player camera, attached to boom. |
| 3 | HF_HealthComponent | HP, damage, healing, regen, i-frames, death, status effects, stagger. |
| 4 | HF_FearSanityComponent | Sanity drain in darkness, fear spikes from scares, visual effects. |
| 5 | HF_InteractionComponent | Line trace from camera, shows prompts on HUD when aiming at interactables. |
| 6 | HF_InventoryComponent MULTIPLAYER | Slot-based inventory with stacking, weight, equip/unequip. |
| 7 | HF_NoiseSystemComponent | Tracks player noise (walk/run/crouch). AI hears it. |
| 8 | HF_NoteReaderComponent | Collect and read notes/documents. Widget auto-shows. |
| 9 | HF_HorrorHUDComponent | Bridges character state (HP, sanity, stamina, battery) to HUD widget. |
| 10 | HF_LeanPeekComponent | Lean left/right to peek around corners without exposing your body. |
| 11 | HF_DeathManager | Handles death: respawn, permadeath, lives, or downed state. |

| | | |
|---|---|---|
| 12 | HF_CameraManager | Head bob, sprint FOV boost, fear tremble, night vision. |
| 13 | HF_AccessibilityComponent | Scare intensity scaling, subtitles, motion sickness mode. |
| 14 | HF_FlashlightComponent | Toggle flashlight on/off. Battery drains. Flickers when low. |
| 15 | HF_BreathHoldComponent | Hold breath while hiding to reduce detection chance. |
| 16 | HF_WeaponSocketComponent | Attaches visible weapon meshes to character skeleton sockets. |

> 💡 **All auto-created**
>
> You don't need to add any of these manually. They're all created as default subobjects in the constructor. Just use `HF_HorrorCharacter` and they're there.

# ❤️ Health & Damage System

Already included on `HF_HorrorCharacter` with `bIsPlayer = true`. For custom characters or enemies, add `HF_HealthComponent` manually.

## Dealing Damage

Two ways to deal damage to anything with a HealthComponent:

### Option A: Unreal's Built-in Damage (recommended)

```
    Apply Damage
◆ Base Damage: 50.0
○ Damaged Actor: Target
```

*HealthComponent auto-receives damage via OnTakeAnyDamage binding. No extra wiring needed.*

### Option B: Direct Call

```
    Get Health Component → Apply Simple Damage
◆ Amount: 50.0
○ Damage Causer: Self
```

### Properties (Details Panel)

| PROPERTY | DEFAULT | WHAT IT DOES |
|---|---|---|
| MaxHealth | 100 | Maximum HP. Horror games typically use 50–100. |
| bEnableRegeneration | false | **ON:** modern horror (RE7). **OFF:** classic survival horror (healing items only). |
| RegenRate | 1.0/s | HP per second when regenerating. 0.5 = punishing, 5.0 = forgiving. |
| RegenDelay | 5.0s | Seconds after last damage before regen starts. |
| bEnableIFrames | true | Invincibility after taking damage. Prevents stunlock. |
| IFramesDuration | 0.5s | 0 = enemies can stunlock. 0.5–1.0 = fair. 2.0+ = very forgiving. |
| Armor | 0 | Flat damage reduction. Applied before damage type multipliers. |
| bIsPlayer | false | Set `true` for player — applies difficulty DamageMultiplierToPlayer. |
| bIsImmortal | false | Cannot take any damage at all. Used by EnemySlasher. |
| bIsUnkillable | false | HP cannot drop below 1. Takes damage but never dies. |
| bEnableStagger | false | Accumulated damage triggers stagger at threshold. |
| StaggerThreshold | 50 | Damage needed to trigger stagger. Resets after triggering. |

## Functions

| FUNCTION | DESCRIPTION |
| --- | --- |
| ApplyDamage(DamageInfo) | Full damage with type, direction, armor, crits, difficulty scaling. |
| ApplySimpleDamage(Amount, Causer) | Quick damage — just amount + who caused it. |
| Heal(Amount) | Restore HP. Clamped to MaxHealth. |
| HealToFull() | Full heal to MaxHealth. |
| Kill(Killer) | Instant death (blocked by bIsUnkillable/bIsImmortal). |
| Revive(HP) | Bring back from death. Sets bIsAlive = true. |
| SetInvincible(Duration) | Timed invincibility. |
| GetHealthPercent() | 0.0–1.0 for HP bars. |

### Events (bind in Blueprint)

| EVENT | WHEN | PARAMETERS |
|---|---|---|
| OnDamageReceived | Any damage taken | FHF_DamageInfo (amount, type, instigator) |
| OnHealed | Any healing | HealAmount, NewHealth |
| OnDeath | HP reaches 0 | DeadActor |
| OnHealthChanged | Any HP change | CurrentHP, MaxHP, Delta |
| OnStaggered | Stagger threshold reached | — |

### Status Effects

HealthComponent manages 10 status effects: **Bleeding, Poisoned, Burning, Frozen, Infected, Stunned, Blinded, Slowed, Hallucinating**. Each can stack, deal tick damage, and auto-expire.

**Status Effect Usage**

```
HealthComp → Apply Status Effect (Burning, 10s, 5dmg/tick)

HealthComp → Remove Status Effect (Burning)

◆ HealthComp → Has Status Effect (Poisoned) → true/false
```

*Burn/Poison/Bleed auto-deal HP damage if HealthComponent is on the same actor. No extra wiring.*

## Multiplayer

MULTIPLAYER CurrentHealth, MaxHealth, bIsAlive, bIsInvincible are replicated. Damage and healing only process on the server (authority check in ApplyDamage and Heal). Clients receive replicated values via OnRep.

---

# 😱 Fear & Sanity System

**Sanity** = long-term mental state. Drains passively in darkness, recovers in safe zones.
**Fear** = short-term reaction. Spikes from scare events, decays over time. Both drive visual effects, hallucinations, and gameplay penalties.

## How It Works

**Fear & Sanity Flow**

```
Player enters DarknessZone → Sanity starts draining (PassiveSanityDrain/s)

Player walks through ScareTrigger → Fear spikes (FearImpact) + Sanity drops
(SanityImpact)

Fear level rises → Visual effects intensify (vignette, chromatic, grain, camera shake)

Player enters SafeRoomZone → Sanity recovers (SafeZoneRecoveryRate/s)

Fear decays naturally over time (FearDecayRate/s)
```

## Fear Levels

| LEVEL | THRESHOLD | EFFECTS |
|---|---|---|
| Calm | 0 | Normal gameplay |
| Uneasy | 15 | Subtle vignette begins |
| Anxious | 35 | Heartbeat audio starts |
| Scared | 55 | Camera trembles, breathing gets heavy |
| Terrified | 75 | Hallucination chance, chromatic aberration |
| Panic | 90 | Maximum distortion, heavy grain, loud heartbeat |

## Properties

| PROPERTY | DEFAULT | WHAT IT DOES |
|---|---|---|
| MaxSanity | 100 | Maximum sanity value. |
| PassiveSanityDrain | 0.5/s | Drain rate in darkness. 0.1 = slow (explore freely). 2.0 = flashlight CRITICAL. |
| SafeZoneRecoveryRate | 2.0/s | Recovery in safe rooms. High = powerful safe rooms. |
| bSanityDepletionKills | false | **ON:** zero sanity = death (Amnesia). **OFF:** zero = visual effects only (Eternal Darkness). |
| FearDecayRate | varies | How fast fear naturally decays. |
| ScareIntensityMultiplier | 1.0 | Scale all scare impacts (0 = no fear, 2 = double fear). Set by AccessibilityComponent. |
| DarknessDrainMultiplier | 1.0 | Extra multiplier for darkness drain. |

## Functions

| FUNCTION | DESCRIPTION |
| --- | --- |
| DrainSanity(Amount) | Reduce sanity by amount. |
| RestoreSanity(Amount) | Restore sanity. |
| SetSanity(Value) | Set exact sanity value. |
| AddFear(Amount) | Spike fear. |
| SetFear(Value) | Set exact fear value. |
| CalmDown() | Instantly reset fear to 0. |
| GetSanityPercent() | 0.0–1.0 for sanity bar. |
| GetFearPercent() | 0.0–1.0 for fear indicator. |
| EnterDarkness() / LeaveDarkness() | Toggle darkness drain. |

## Events

| EVENT | WHEN |
| --- | --- |
| OnSanityChanged | Any sanity change — use for sanity bar |
| OnFearLevelChanged | Fear level changed (Calm → Uneasy, etc.) — use for music layers |
| OnSanityDepleted | Sanity hit zero |

MULTIPLAYER CurrentSanity, CurrentFear, and CurrentFearLevel are all replicated.

# 👋 Interaction System

`HF_InteractionComponent` fires a line trace from the camera every tick. If it hits an actor implementing `IHF_InteractableInterface`, it shows the interaction prompt on HUD.

**Interaction Flow**

```
InteractionComp traces from camera (InteractionRange units)

Hits actor with IHF_InteractableInterface? → CanInteract()?

HUD shows prompt: "[E] Open Door"

Player presses E → Interact() fires on the target actor
```

## Making an Actor Interactable

### 1 Create Actor Blueprint

Create a new Actor Blueprint.

### 2 Add Interface

Class Settings → Interfaces → Add → `HF_InteractableInterface`

### 3 Implement Events

Right-click Event Graph → Add Event from Interface: **Interact**, **Get Interaction Prompt**, **Can Interact**, **Get Interaction Type**.

Built-in interactables that already implement the interface: `HF_DoorActor`, `HF_HidingSpot`, `HF_GeneratorActor`, `HF_CombinationLock`, `HF_LightSwitchActor`, `HF_WorkbenchActor`.

| PROPERTY | DEFAULT | DESCRIPTION |
| --- | --- | --- |
| InteractionRange | 250 | Trace distance in Unreal units. Lower = more immersive. |
| TraceChannel | Visibility | Collision channel for the trace. |

# 🎒 Inventory System

MULTIPLAYER Slot-based inventory with stacking, weight limit, equip/unequip. Slots array is replicated.

## Functions

| FUNCTION | DESCRIPTION |
| --- | --- |
| AddItem(ItemData, Qty) | Returns number actually added. Inventory might be full or too heavy. |
| RemoveItem(ItemID, Qty) | Returns number actually removed. |
| UseItem(SlotIndex, Target) | Triggers OnItemUsed event. Consumes item if bConsumable. |
| HasItem(ItemID, Qty) | Check if player has at least Qty of item. |
| EquipSlot(Index) | Set active equipped item. |
| ClearInventory() | Remove all items. |
| GetCurrentWeight() | Total weight of all items. |
| IsFull() | No free slots remaining. |

## Properties

| PROPERTY | DEFAULT | WHAT IT DOES |
|---|---|---|
| MaxSlots | 12 | Lower (4–6) = RE-style inventory management. Higher (20+) = hoard everything. |
| bUseWeight | false | Enable weight limit. OFF = slot-based only. |
| MaxWeight | 50 | Maximum carry weight (only if weight enabled). |

## Item Data (FHF_ItemData)

| FIELD | TYPE | DESCRIPTION |
|---|---|---|
| ItemID | FName | Unique identifier (e.g. "Medkit", "Key_Basement") |
| DisplayName | FText | Shown in inventory UI |
| Description | FText | Item description text |
| Icon | UTexture2D | Inventory icon image |
| Weight | float | Weight per unit (if weight system enabled) |
| bStackable | bool | Can multiple units share one slot? |
| MaxStackSize | int32 | Max units per stack |
| bConsumable | bool | Consumed when UseItem is called? |
| bKeyItem | bool | Key items (doors, puzzles) |

## Events

| EVENT | WHEN |
| --- | --- |
| OnItemAdded | Item added — use for "+1 Medkit" floating text |
| OnItemRemoved | Item removed |
| OnItemUsed | Item consumed/used |
| OnInventoryChanged | Any change — use to refresh inventory UI |

# ⚔️ Weapon System

`HF_WeaponComponent` handles melee attacks, firearms with ammo/reload, and throwables. Add it to **any actor** — weapon pickups, NPCs, or the player. It handles cooldowns, stamina cost, noise generation, and sound.

## Weapon Types

| TYPE | BEHAVIOR |
| --- | --- |
| Melee | No ammo. Attack within range, consumes stamina, respects cooldown. |
| Firearm | Uses ammo from clip. Must reload when empty. Plays empty-click at 0. |
| Throwable | Single-use throw. Implement throw logic in your Blueprint on OnAttack. |

## Properties

| PROPERTY | DEFAULT | DESCRIPTION |
|---|---|---|
| Damage | 25 | Damage dealt per attack |
| AttackRange | 150 | Melee range (units) |
| AttackCooldown | 0.8 | Seconds between attacks |
| StaminaCost | 10 | Stamina consumed per attack |
| NoiseLoudness | 0.7 | Noise generated — AI hears this via NoiseSystem |
| MaxAmmo | 12 | Total reserve ammo (firearm) |
| ClipSize | 6 | Max ammo per clip (firearm) |
| ReloadTime | 2.0 | Seconds to reload (firearm) |

## Socket Properties (for visible weapon mesh)

| PROPERTY | DESCRIPTION |
|---|---|
| WeaponMesh | Static Mesh to display on the character's socket (e.g. SM_Axe, SM_Pistol) |
| AttachSocketName | Per-weapon socket override (empty = use WeaponSocketComponent default) |
| SocketOffset | Position/rotation/scale offset on the socket |

> ⚠️ **Important**
>
> **Applying damage to a target is YOUR responsibility.** Bind to `OnAttack` and do a line trace, overlap, or any hit detection you want. Then call `HealthComp→ApplySimpleDamage()` on whatever you hit. The weapon handles the "can I attack?" logic — you handle the "what did I hit?" logic.

## Events

| EVENT | WHEN | PARAMS |
|---|---|---|
| OnAttack | Attack performed | float Damage |
| OnReloadComplete | Reload finished | — |
| OnAmmoEmpty | Tried to fire at 0 ammo | — |

MULTIPLAYER CurrentAmmo is replicated. Attack() and Reload() block on simulated proxies.

## Weapon Socket System

`HF_WeaponSocketComponent` attaches a visible weapon mesh to your character's skeleton socket. Already on `HF_HorrorCharacter`.

1 Add socket `weapon_r` to your skeleton (right-click hand_r → Add Socket)

2 On WeaponComponent, set `WeaponMesh` to your Static Mesh

3 Call `WeaponSocketComp→EquipWeapon(WeaponComp)`. Mesh appears on socket. `UnequipWeapon()` removes it.

Swapping: `EquipWeapon()` auto-unequips the previous weapon. Events: `OnWeaponEquipped`, `OnWeaponUnequipped`.

---

## 🔦 Flashlight

Already on `HF_HorrorCharacter`. Auto-creates a SpotLight if none assigned. Toggle with F key (auto-bound).

| PROPERTY | DEFAULT | WHAT IT DOES |
|---|---|---|
| MaxBattery | 120s | 2 minutes of light. Lower (30–60s) = scramble for batteries. |
| DrainRate | 1.0/s | Battery per second when on. Higher = faster drain, more panic. |
| bUseBatteries | true | **OFF:** infinite light, no resource anxiety. |
| FlickerThreshold | 0.15 | Battery % when flickering starts. |

| FUNCTION | DESCRIPTION |
|---|---|
| Toggle() | Switch on/off |
| TurnOn() / TurnOff() | Explicit control |
| RechargeBattery(Amount) | Add battery from pickup |
| ReplaceBattery() | Full recharge to max |

Events: `OnToggled`, `OnBatteryLow`, `OnBatteryDepleted`. **MULTIPLAYER** bIsOn and CurrentBattery are replicated.

---

# 🤖 AI Enemy System

**No animations or meshes included.** This is a C++ runtime plugin — you bring your own character mesh, animations, and sounds. The plugin provides all AI logic, awareness, pathfinding, and behaviors.

## Setting Up Your First Enemy

### 1 Create Enemy Character Blueprint

**Content Browser** → **Blueprint Class** → **Character**. Add your Skeletal Mesh.

### 2 Set AI Controller

In Details Panel: `AI Controller Class` = `HF_HorrorAIController`, `Auto Possess AI` = Placed in World or Spawned.

> **ℹ️ What happens automatically**
>
> When the enemy spawns, HF_HorrorAIController possesses it. The controller auto-creates an AwarenessComponent, auto-targets the player, applies difficulty scaling, and starts listening to events. Zero Blueprint logic needed.

### 3 Create Behavior Tree + Blackboard

Content Browser → AI → Behavior Tree. Content Browser → AI → Blackboard. Assign in AI Controller.

### 4 Add NavMesh

Place **Nav Mesh Bounds Volume** in level → **Build** → **Build Paths** → Press **P** to visualize (green = walkable).

🚨 **NavMesh required!**

Without a Nav Mesh, the AI cannot move. All enemies use Unreal's navigation for pathfinding.

## Awareness System (Detection)

`HF_AwarenessComponent` handles sight + hearing. States progress: **Unaware** → **Suspicious** → **Searching** → **Alerted** → **Chasing** → **Attacking** → **LostTarget**.

| PROPERTY | DEFAULT | DESCRIPTION |
| --- | --- | --- |
| SightRange | 1500 | How far the AI can see |
| SightAngle | 60° | Field of view. 360 = omnidirectional (Slender-style) |
| HearingRange | 2000 | How far the AI can hear noise |
| DecayRate | 5/s | Awareness drops if not detecting |

## Built-in BT Nodes

| NODE | TYPE | WHAT IT DOES |
|---|---|---|
| HF Patrol | Task | Walk between PatrolPoints array |
| HF Chase Player | Task | Chase with rubber-banding speed |
| HF Search Hiding Spots | Task | Check nearby HidingSpots for hidden player |
| HF Stalk | Task | Maintain distance, hide when player looks |
| HF Teleport Behind | Task | Teleport behind player when not looking |
| HF Check Awareness | Decorator | Branch based on AwarenessState |
| HF Update Perception | Service | Keep perception data fresh |

## Built-in Enemy Classes

| CLASS | DESCRIPTION | KEY DIFFERENCES |
|---|---|---|
| HF_EnemyCharacterBase | Base class. HP, attack sphere, death, AI-ready. | Foundation for all enemies. 5 montage slots. |
| HF_EnemyMelee | Killable. Zombie/mutant/infected. | HP=150, HeavyAttack 20% chance (2x damage), ChaseSpeed=400. |
| HF_EnemySlasher | Unkillable. Outlast/Alien:Isolation style. | bIsImmortal=true, Damage=999 (instant kill), FearOnSight=10, FearOnAttack=40. |

## AI Events

| EVENT | WHEN |
| --- | --- |
| OnChaseStarted / OnChaseStopped | Chase mode toggle — play chase music |
| OnTargetSpotted / OnTargetLost | First sight / lost player — play roar/confused |
| OnNoiseHeard | Heard a noise — look toward source |
| OnAwarenessStateChanged | Any state transition |
| OnTeleported | AI teleported — play VFX |

## 🚪 Hiding & Safe Rooms

### HF_HidingSpot

Place in level (closets, lockers, under beds). Player interacts → enters hiding. AI can search with configurable find chance.

| PROPERTY | DESCRIPTION |
| --- | --- |
| SpotType | Closet, Locker, UnderBed, Curtain, Barrel, Dumpster |
| HidingCameraOffset / Rotation | Camera position and angle when hiding |
| bCanPeek | Player can peek while hiding |
| NoiseReductionMultiplier | Noise output while hiding (0.1 = 10%) |
| CustomFindChance | AI probability of finding player (0.15 = 15%) |

### HF_BreathHoldComponent

Already on HorrorCharacter. Hold breath to become nearly silent. Drains over time. Releasing makes a gasp noise.

| PROPERTY | DESCRIPTION |
| --- | --- |
| MaxBreathHoldTime | Seconds the player can hold breath |
| WarningThreshold | % remaining when warning event fires |

### HF_SafeRoomZone

Place in level as a safe area.

| PROPERTY | DEFAULT | DESCRIPTION |
| --- | --- | --- |
| bBlocksAI | true | AI enemies cannot enter |
| bRestoresSanity | true | Sanity recovers inside |
| bAllowsSaving | true | Player can save here |

## 👻 Scare Events

| ACTOR | DESCRIPTION |
| --- | --- |
| HF_ScareTrigger | Box trigger — player enters → sound + camera shake + fear spike. Configurable: FearImpact, SanityImpact, bOneShot. |
| HF_MannequinScare | Moves toward player when not looking (SCP-173 style). |
| HF_ApparitionActor | Ghost appears, fades in/out, disappears when looked at directly. |
| HF_ObjectMoveScare | Object slides/moves when player isn't looking. |
| HF_FlashbackTrigger | Screen fades, shows flashback scene, fades back. |
| HF_RandomScareScheduler | Randomly triggers scare events at configurable intervals. |
| HF_HallucinationComponent | Spawns hallucination actors when sanity is low. |
| HF_FlickerLightComponent | Random light flicker. Attach to any light component. |
| HF_FogManager | Dynamic fog transitions based on threat level. |
| HF_ScriptedEventSequencer | Timed sequence of events (sounds, spawns, fear spikes). |
| HF_PossessionComponent | Player gets "possessed" when sanity drops below threshold. |

# 🔑 Doors, Locks & Environment

## HF_DoorActor

States: Closed → Open → Locked → Barricaded → Broken. Auto-animates rotation.

| FUNCTION | DESCRIPTION |
| --- | --- |
| OpenDoor() / CloseDoor() | Open/close with sound |
| LockDoor() / UnlockDoor() | Lock/unlock |
| TryUnlockWithKey(KeyID) | Check inventory for correct key |
| BreakDoor() | Permanently open (broken state) |
| DamageDoor(Amount) | Reduce door HP, breaks at 0 |

MULTIPLAYER DoorState is replicated.

## Other Environment Actors

| ACTOR | DESCRIPTION |
| --- | --- |
| HF_BarricadeActor | Boards to block enemies. Each board adds HP. |
| HF_WindowActor | Breakable windows with health. Barricadable. |
| HF_LightSwitchActor | Toggle connected lights. MULTIPLAYER |
| HF_GeneratorActor | Fuel-powered. Powers linked systems. |
| HF_KeyLockComponent | Requires specific key from inventory. |
| HF_DarknessZone | Triggers sanity drain when player enters. |
| HF_DayNightCycle | Directional light rotation. Day/night with colors. |
| HF_TrapActor | Damage + stun on trigger. Configurable for player/AI. |

## 🎥 Camera System

`HF_CameraManager` auto-finds the CameraComponent and applies effects.

| PROPERTY | DEFAULT | DESCRIPTION |
|---|---|---|
| bEnableHeadBob | true | Sinusoidal bob when walking/running |
| HeadBobIntensity | 0.5 | Bob strength. 1.5+ = nausea risk. |
| BaseFOV | 90 | Default field of view |
| SprintFOVBoost | 10 | Extra FOV when sprinting |
| bEnableNightVision | false | Night vision toggle with battery |

Additional: `HF_DollyZoomComponent` (Hitchcock vertigo), `HF_FixedCameraTrigger` (classic RE angles), `HF_ForcedLookComponent` (force camera at target), `HF_PostProcessFearDriver` (auto vignette/grain from fear), `HF_LeanPeekComponent` (lean around corners).

## 💀 Death & Respawn

`HF_DeathManager` auto-connects to HealthComponent::OnDeath.

| MODE | BEHAVIOR |
| --- | --- |
| Respawn | Teleport to last checkpoint after delay, full heal |
| Permadeath | Game over — death screen shows |
| Lives | Limited lives, respawn until 0, then game over |
| DownedState | Co-op: go down, teammates revive within time limit |

`HF_CheckpointActor` — place in level, player walks through → respawn point updated.
`HF_ConsequenceManager` — escalating penalties per death (enemies faster, fewer resources).
`HF_InjuryComponent` — persistent injuries: Leg (slow), Arm (slow interact), Bleeding (HP drain), Infection (worsens).

# 💾 Save System

`HF_SaveManager` is a Blueprint Function Library. Call from anywhere.

### Save / Load Functions

HF Save Manager → Save Game (SlotName, UserIndex)

HF Save Manager → Load Game (SlotName, UserIndex)

HF Save Manager → Quick Save / Quick Load

HF Save Manager → Auto Save

◆ HF Save Manager → Does Save Exist (SlotName)

HF Save Manager → Delete Save (SlotName)

## What Gets Saved

| DATA | SOURCE |
| --- | --- |
| Player position + rotation | Character transform |
| Health, MaxHealth, bIsAlive | HealthComponent |
| Active status effects (all) | HealthComponent |
| Sanity + Fear level | FearSanityComponent |
| Full inventory (items + quantities + icons + everything) | InventoryComponent |
| Equipped item | InventoryComponent |
| Flashlight battery | FlashlightComponent |
| Lives remaining + last checkpoint | DeathManager |
| Objective progress (active + completed) | ObjectiveTracker |
| Story flags (dialogue branching) | DialogueComponent |
| Game phase + difficulty | GameMode |
| Threat level, death count, play time | GameState |
| Solved puzzles, collected keys, triggered scares | GameState |

Custom data: use `SaveGame→SetModuleValue(ModuleName, Key, Value)` and `GetModuleValue()` for your own systems.

---

## 📢 Noise System

`HF_NoiseSystemComponent` on HorrorCharacter. AI hears noise through AwarenessComponent.

| PROPERTY | DEFAULT | DESCRIPTION |
| --- | --- | --- |
| WalkNoise | 0.3 | Noise when walking |
| RunNoise | 0.7 | Noise when sprinting — running is dangerous! |
| CrouchNoise | 0.1 | Noise when crouching — nearly silent |
| NoisePropagationRadius | 1500 | How far noise travels (units) |

WeaponComponent automatically calls `MakeNoise(NoiseLoudness)` on attack. No extra setup needed.

---

## Dialogue, Notes & Radio

| COMPONENT | DESCRIPTION |
| --- | --- |
| HF_DialogueComponent | Node-based branching dialogue with story flags. Widget auto-shows. `SetStoryFlag("HasKey", true)` / `GetStoryFlag("MetNPC")`. |
| HF_NoteReaderComponent | Collect and read notes/documents. Widget auto-shows. |
| HF_MonologueComponent | Sequential voice lines with subtitles. Play(), PlaySingle(), Stop(). |
| HF_RadioContactComponent | Incoming radio messages with static noise and speaker name. |
| HF_WalkieTalkieComponent | Range-based walkie-talkie with signal quality (distance-based). |

## 🔨 Crafting & Resources

| COMPONENT | DESCRIPTION |
| --- | --- |
| HF_CraftingComponent | Recipe-based. CanCraft(ID), Craft(ID), DiscoverRecipe(ID). Auto-consumes from inventory. |
| HF_ConsumableComponent | Use items: Health, Sanity, Stamina, Battery, Antidote, Custom. Events: OnUsed. |
| HF_DurabilityComponent | Tools/weapons break after use. Repair at workbench. BreakSound on zero. |
| HF_FuelSystemComponent | Generic fuel/resource for generators. AddFuel(), StartConsuming(), StopConsuming(). |
| HF_WorkbenchActor | Interactable crafting station. OnWorkbenchUsed event. |

## 🤝 Companion & NPCs

| COMPONENT | DESCRIPTION |
| --- | --- |
| HF_CompanionComponent | AI companion: Following, Waiting, Hiding, Downed, Dead. Morale + Trust system. CommandFollow(), CommandWait(), CommandHide(). |
| HF_CompanionAIController | AI logic for companion behavior. Detects threats, assists player. |
| HF_NPCScheduleComponent | Time-based NPC routines. Schedule entries: start/end time, location, idle anim. |

# 🔍 Investigation Tools (Phasmophobia-style)

Ghost-hunting equipment. Each is a separate component — add only what you need.

| COMPONENT | DESCRIPTION |
| --- | --- |
| HF_EMFReaderComponent | Levels 0–5 based on proximity. BeepSound on detection. Events: OnEMFLevelChanged, OnHighEMFDetected. |
| HF_ThermometerComponent | Temperature drops near ghost. FreezingThreshold. Event: OnFreezingDetected. |
| HF_SpiritBoxComponent | Radio-frequency communication. ResponseChance per scan. Event: OnGhostResponse. |
| HF_PhotoCaptureComponent | SceneCaptureComponent2D photo evidence. MaxPhotos, ShutterSound. Event: OnPhotoCaptured. |
| HF_EvidenceComponent | Tracks collected evidence. CollectEvidence(), HasEvidence(), GetCompletionPercent(). |
| HF_ExamineComponent | Hold an object up and turn it over. The game pauses, movement and looking are taken away, the object is centred on its own bounds in front of the camera and the mouse rotates it freely in any direction. The interact key puts it back down. Properties: ExamineDistance, RotationSpeed, EyeHeight-independent, bPauseWhileExamining, bMouseRotatesObject, bReturnToOriginalPlace, DropDistance, bDropToGround. Functions: StartExamine, StopExamine, RotateExaminedObject, ResetExamineRotation. |

## Co-op Multiplayer

`HF_CoopManager` — player proximity, isolation detection, shared sanity, revive system.

| PROPERTY | DESCRIPTION |
| --- | --- |
| CoopMode | SinglePlayer, CoopSymmetric, CoopAsymmetric (1 ghost), Versus |
| MaxPlayers | Maximum players (default 4) |
| bSharedSanity | All players share lowest sanity value |
| bReviveSystem | Enable downed/revive mechanic |
| IsolationDistance | Distance to be "isolated" — AI prioritizes lone players |

`HF_SpectatorComponent` — dead players spectate living teammates. NextTarget() / PrevTarget().

### Networking model

The framework is server-authoritative. Gameplay state lives on the server and replicates down; clients own only their own camera, input and UI. Interactions performed by a client are routed to the server, and screens that a world actor opens (keypad, note reader, death screen) are sent back to the player who triggered them rather than to the host.

| AREA | WHAT REPLICATES |
| --- | --- |
| Character state | Health, sanity and fear, injuries, inventory, flashlight battery, weapon ammo, tool durability |
| World actors | Doors, light switches, generators and fuel, barricades, boarded windows, hiding spots, pickups and one-shot props, checkpoints |
| Puzzles | Solved state on `HF_PuzzleBase`, so every puzzle type inherits it. Combination locks validate the entered code on the server |
| Progression | Objectives, evidence, collected notes, story flags, endings, death consequences, game state |
| World systems | Day/night clock, scares and apparitions, scripted sequences, physics objects while carried or thrown |
| AI | Perception, noise propagation and behaviour run on the server. Enemies target the nearest player and re-evaluate as players move |

Systems that read "the player" — proximity prompts, scare triggers, stalking AI, radio signal strength — resolve against the player actually involved, not player 0. Look-away mechanics such as the mannequin and the teleporting enemy require that *no* player has them in view.

Pausing is single-player only. In a session with more than one player the pause menu and the examine screen open over a running world, since `SetGamePaused` would freeze the game for everybody.

## Session model and saving

Co-op runs on the host's session, the same way most co-op horror games do: the host owns the world and the save file, and guests join into it. World progress — solved puzzles, key items obtained, one-shot scares already triggered, difficulty, phase and death count — is stored in the save and restored for the whole session. Per-player state in the slot (transform, health, sanity, inventory, flashlight battery) is the host's; guests spawn normally when a save is loaded.

Worth knowing when you design your levels: an item's *collection* is recorded globally, but the physical item in a guest's inventory is not. If a guest is carrying the basement key when the game is saved, the world remembers the key was found and nobody is holding it after loading. Either gate progression on the recorded key item rather than on the inventory item, or place progression items where the host is expected to pick them up. Per-character persistence, as in a survival sandbox, is a different save format and is left to the project.

### Hosting a session

`HF_MainMenuWidget` can show **HOST** and **JOIN** entries — enable `bShowCoopButtons`. Hosting opens `StartLevelName` as a listen server; joining travels to the address typed into the field. No online subsystem is required, so it works on LAN and over the internet with a forwarded port, and Steam or EOS sessions can be layered on top without touching the rest of the framework. `HF_GameMode` uses seamless travel so connected players move to the next level together.

---

## 🖥️ UI / HUD System

**Everything is automatic.** Set `HF_HorrorHUD` as HUD Class. It auto-creates 10 widgets on BeginPlay.

| WIDGET | VISIBILITY | WHEN SHOWN |
|---|---|---|
| HorrorHUDWidget | Always | HP, sanity, stamina, battery bars, crosshair, prompt |
| InventoryWidget | Hidden | ShowInventory() / ToggleInventory() |
| PauseMenuWidget | Hidden | ShowPauseMenu() |
| DeathScreenWidget | Hidden | Auto on player death |
| DialogueWidget | Hidden | Auto when dialogue starts |
| NoteReaderWidget | Hidden | Auto when reading note |
| SubtitleWidget | Hidden | During monologue/dialogue |
| CombinationLockWidget | Hidden | ShowCombinationLock() |
| EvidenceBoardWidget | Hidden | ShowEvidenceBoard() |
| LoadingScreenWidget | Hidden | ShowLoadingScreen() |

> 💡 **Customizing Widgets**
>
> Create Widget Blueprint → Parent Class = the widget you want to replace. Customize colors/sizes in Details. Assign in HUD Blueprint under `Horror|UI|Widgets`. If a slot is `None`,

> the built-in C++ default is used automatically.

## 🎵 Audio System

| COMPONENT | DESCRIPTION |
| --- | --- |
| HF_AdaptiveMusicManager | Layered music responding to threat/fear level. SetFearLevel(), PlayStinger(). |
| HF_HeartbeatComponent | Heartbeat speeds up with fear or low HP. SetIntensity(), ForceSpike(). |
| HF_BreathingAudioComponent | 4 states: Normal, Heavy, Panicked, Holding. Auto-transitions from stamina/fear. |
| HF_AmbientSoundManager | Named audio layers. EnableLayer("Wind"), SetLayerVolume("Crickets", 0.3). |
| HF_AudioOcclusionHelper | Auto line-trace occlusion for 3D sounds. Volume reduces through walls. |
| HF_StingerSystem | One-shot scare sounds with cooldown. PlayStinger("JumpScare01"), PlayRandom(). |
| HF_ReverbZoneComponent | Per-zone reverb effects (caves echo, bathrooms tile). |

## ⚔️ Difficulty System

5 presets. Change at runtime: `GameMode→SetDifficulty(NewDifficulty)`. All systems update immediately.

| SETTING | STORY | EASY | NORMAL | HARD | NIGHTMARE |
|---|---|---|---|---|---|
| Damage to Player | 0.25x | 0.5x | 1.0x | 1.5x | 2.5x |
| Enemy Speed | 0.7x | 0.85x | 1.0x | 1.15x | 1.3x |
| Enemy Detection | 0.5x | 0.75x | 1.0x | 1.3x | 1.75x |
| Sanity Drain | 0.3x | 0.6x | 1.0x | 1.5x | 2.0x |
| Resources | 2.0x | 1.5x | 1.0x | 0.7x | 0.4x |
| Permadeath | No | No | No | No | Yes |

## 📦 Module System (23 Modules)

---

The `ActiveModules` bitmask in PluginSettings controls which systems run. Disabled modules auto-deactivate — zero CPU cost.

| MODULE | WHAT GETS DISABLED WHEN OFF |
|---|---|
| Core | GameMode, GameState, EventHub, SaveManager — **never turn off** |
| HealthDamage | HealthComponent, InjuryComponent |
| FearSanity | FearSanityComponent, PostProcessFearDriver |
| AIEnemy | HorrorAIController, all BT tasks |
| AwarenessDetection | AwarenessComponent (sight, hearing) |
| EnvironmentAtmosphere | DarknessZone, FlickerLight, Fog, DayNight |
| InteractionInventory | Interaction, Inventory, Crafting, Grabbable, Examine |
| HidingStealth | HidingSpot, BreathHold, Noise, LeanPeek |
| PuzzleProgression | PuzzleBase, CombinationLock, ObjectiveTracker |
| JumpScareEvents | ScareTrigger, MannequinScare, Apparition, ObjectMove |
| CameraPerspective | CameraManager, ForcedLook, DollyZoom |
| NarrativeStory | Dialogue, NoteReader, Monologue |
| Multiplayer | CoopManager, Spectator |
| UIHUD | HorrorHUDComponent, all 10 widgets |
| Audio | AdaptiveMusic, Heartbeat, Breathing, Reverb |
| DeathConsequence | DeathManager, ConsequenceManager |

| | |
|---|---|
| CraftingResources | Crafting, Consumable, Durability, Fuel |
| PhysicsDestruction | Destructible, Throwable, Rope |
| TimeWorldState | DayNight, WorldPhase, Timer, Trail |
| ProceduralReplayability | RandomPlacement, DifficultyMutators, MultipleEndings |
| PhotoEvidence | PhotoCapture, EMF, SpiritBox, Thermometer |
| CcompanionNPC | Companion, CompanionAI, WalkieTalkie |
| Accessibility | AccessibilityComponent |

> 💡 **Start minimal**
>
> Begin with only Core + HealthDamage + FearSanity + UIHUD enabled. Add modules as you build features. This way you see only what you need and keep performance tight.

## 🔧 Plugin Settings Data Asset

Your master control panel. One file controls everything.

1 **Create: Content Browser → Right Click → Miscellaneous → Data Asset → pick** `HF_PluginSettings`

2 **Assign in GameMode Blueprint → Details →** `HorrorSettings`

Controls: ActiveModules bitmask, player health/regen, fear/sanity settings, AI defaults, flashlight, inventory, hiding, save, audio, camera, accessibility, and difficulty presets. Without it, all systems use hard-coded defaults.

# 📡 Event Hub

`UHF_EventHub` is a GameInstance Subsystem. Central event bus — all systems broadcast through it.

### Get Event Hub & Bind Events

- Get Game Instance → Get Subsystem → HF Event Hub

*Then bind to any event:*

```
OnPlayerDeath / OnEnemyDeath

OnItemPickedUp / OnItemUsed

OnStatusEffectApplied / OnStatusEffectRemoved

OnScareTriggered / OnPlayerHealthChanged

OnGameSaved / OnGameLoaded

OnGamePhaseChanged / OnDifficultyChanged
```

> 💡 **Pro tip**
>
> Use EventHub instead of direct component references. This decouples your systems — e.g., your audio manager listens to `OnScareTriggered` without knowing about ScareTrigger directly.

## 🎮 Cheat Manager

`HF_CheatManager` — type in the UE console (press `~`):

| COMMAND | DESCRIPTION |
| --- | --- |
| HF_God | Toggle invincibility |
| HF_HealFull | Full HP |
| HF_Kill | Instant death |
| HF_MaxSanity / HF_ZeroSanity | Set sanity |
| HF_MaxFear / HF_ZeroFear | Set fear |
| HF_InfiniteBattery | Toggle infinite flashlight |
| HF_SetThreat [0–100] | Set threat level |
| HF_SetPhase [0–5] | Set game phase |
| HF_AddItem [ID] [Qty] | Add item to inventory |
| HF_ClearInventory | Remove all items |
| HF_TeleportToCheckpoint | Teleport to last checkpoint |
| HF_NoClip | Toggle fly mode |

# 🖥️ Main Menu — Working Menu Without a Single Blueprint Node

**New in 1.1.** `HF_MainMenuGameMode` turns any empty level into a working main menu. It creates the menu widget, hands input to the UI, shows the cursor and deliberately spawns no pawn, so nothing drifts around behind the interface.

## Setting it up

1. Create an empty level and drop in a `PlayerStart`.
2. Create a Blueprint child of `HF_MainMenuGameMode`.
3. Fill in the fields below, above all **Start Level Name**.
4. Set the level's GameMode Override to your Blueprint.
5. Set this level as **Game Default Map** in Project Settings, so it is the first thing a player sees.

| 📋 HF_MainMenuGameMode — Details Panel | |
|---|---|
| Menu Widget Class | `HF_MainMenuWidget (swap for your own subclass)` |
| Game Title | `Large text on the left` |
| Tagline | `Smaller line under the title` |
| Version Text | `Bottom right corner` |
| Start Level Name | `Full path, e.g. /Game/Maps/MyLevel` |
| Show Continue / Settings / Credits | `Hide the entries you have not built yet` |
| Studio Line | `Your studio name, bottom right. Empty by default` |
| Website Button Text | `Label of the website button` |
| Website URL | `Opens in the system browser. The button is hidden while this is empty` |

### Use a full level path

**Start Level Name** takes an `FName`, and a bare level name relies on the engine searching for it. That search is not something to depend on in a packaged build, particularly for a level that lives inside a plugin's content. Write the full path, `/Game/Maps/MyLevel`, and it behaves the same in the editor and in a shipped game.

### Button behaviour

Every button broadcasts its delegate first and then runs a built-in default underneath. Start opens **Start Level Name**, Exit quits the game, and the Website button opens the URL. Bind the delegates if you want your own logic; set `bUseDefaultActions` to false if you want the defaults gone entirely. Existing projects that bound these delegates before 1.1 keep working unchanged.

Leaving **Start Level Name** empty is a valid choice: the button then only broadcasts `OnNewGame`, which is the hook for a project that wants to run a cutscene, load a save or pick a difficulty before travelling.

> ⚠️ **Branding fields are empty on purpose**
>
> Studio line and website are blank in the shipped defaults, so nobody accidentally ships someone else's studio name in their own game. Fill them in with yours.

---

# 🏠 Demo Level Builder — Playable Level in One Click

**New in 1.1.** The plugin ships with a builder that generates a complete, playable demo level from scratch. Drop one actor into an empty map, press a button, press Play. Nothing else to set up.

## Getting the demo

1. Place `HF_DemoLevelBuilder` anywhere in a level.
2. In the Details panel press **Build Demo**.
3. Set the level's GameMode Override to `HF_GameMode` (or a Blueprint child of it).
4. Press Play.

**Important:** the generated actors are ordinary actors saved into your map. Changing builder settings does not update a level that was already built — press **Build Demo** again. Rebuilding deletes everything tagged `HF_DemoGenerated`, so hand edits to generated actors are lost on the next build. Your own actors are never touched.

> ⚠️ **The demo geometry is still being polished**
>
> The level is assembled at runtime from primitives, and placement is tuned by hand, pass by pass. In the current build some props sit slightly off their surface and some room joins are not flush, which can read as objects floating in the air. This is cosmetic. The systems driving them, interaction, puzzles, AI, inventory and objectives, all function normally. Placement fixes are landing in the next update.

> ⚠️ **The demo character has no mesh**
>
> The plugin ships as pure C++ with no content, which is what lets one download work on UE 5.5 through 5.8: a `.uasset` can only be opened by the engine version that saved it, while code compiles against whichever version you are on. The consequence is that `HF_DemoCharacter` arrives without a skeletal mesh. Assign your own, or the mannequin from

> any Epic template, in a Blueprint child of it. In a first-person game you will not see the body anyway, so this is optional for testing.

## What gets built

Seven rooms on a grid, joined by real doorways, plus a basement reached through a locked door:

```
     (0,1) Exit ---- (1,1) Workshop ---- (2,1) Office
                                                |
(0,0) Entrance -- (1,0) Corridor ----- (2,0) Generator Hall
                        |
                 (1,-1) Basement
```

Doorways are derived from where each room's neighbours sit on the grid, so moving a room moves its doors with it. Openings are cut into the walls themselves, with sill and lintel panels, and the same system cuts window apertures.

## Placement rules

The builder never uses hard-coded coordinates. Two rules keep the layout correct when anything changes:

- **Positions are anchors.** "Against the west wall, centred" is resolved against the room's real size, so resizing a room moves its contents with it.
- **Clearances come from measured bounds.** After an actor is spawned it is asked for its real size and pushed off the wall by half its own depth. Replace a placeholder with a large custom mesh and it still stands against the wall rather than inside it.

Objects that rest on other objects are attached as children, not positioned separately. A note on a table is parented to the table, so moving, rotating or replacing the table takes the note with it.

### Builder API (Blueprint callable)

| FUNCTION | PURPOSE |
| --- | --- |
| GetAnchorLocation | Resolve an anchor to a world position on the room floor |
| SpawnInRoom | Spawn and seat an actor against an anchor, with optional mount height |
| SeatInRoom | Re-measure and re-seat an actor. Call after changing its mesh or scale |
| SpawnOnTopOf | Spawn an actor resting on another, attached as a child |
| PlaceOnTopOf | Re-measure both actors and rest one on the other |
| BuildDemo / ClearDemo / ValidatePlacement | Generate, remove, and check for overlapping props |

**Wall-mounted objects need a mount height.** Pass `MountHeight` to `SpawnInRoom` for switches, windows and locks. With zero the object tries to reach the floor, which buries anything whose mesh already sits high in its own constructor.

**Size before you seat.** Placement is derived from the measured object, so a measurement taken before the object was resized describes something that no longer exists. Configure the mesh first, then call `SeatInRoom` or `PlaceOnTopOf` again.

### Room settings

| PROPERTY | MEANING |
| --- | --- |
| Content | Which props are placed: Entrance, Corridor, Generator Hall, Office, Workshop, Basement, Exit, or Empty |
| RelativeCenter / InteriorSize | Position and interior dimensions in centimetres |
| DoorwayWalls | Walls opened as passages. A passage needs an opening in *both* rooms |
| WindowWalls / WindowWidth / WindowHeight / WindowSillHeight / WindowOffset | Window apertures cut into the wall |
| bSnapToGround | Trace down and build on whatever surface is below the builder |

### Mission director

`HF_MissionDirector` is spawned with the demo. It seeds the objective tracker, keeps exactly one objective active, and advances as each is completed. Individual actors stay ignorant of the story: they report the objective they closed and nothing more.

It also seeds the data several systems need in order to do anything: crafting recipes, evidence definitions, and the map's world bounds. Each can be turned off independently.

The demo mission runs: read the shift log, restore the power, find the lock code, take the basement key, document three pieces of evidence, get out. The exit is gated on the evidence.

### Demo overlay

`HF_DemoOverlayWidget` shows the full objective list on the left and a live control legend on the right. Key names are read from the player controller, so rebinding a key updates the legend rather than

making it lie. Turn it off with `bShowDemoOverlay` on the HUD for a real game.

### Default controls

| KEY | ACTION |
| --- | --- |
| W A S D | Move |
| Shift / C / Space | Run / Crouch / Jump |
| E | Interact, and close any open screen |
| F | Flashlight. With a dead battery and a spare in the bag, swaps the cell |
| I | Inventory |
| Tab | Journal |
| M | Map |
| B | Evidence board |
| J | Reopen the last note collected |
| 1 / 2 | EMF reader / Spirit box |
| P / G | Take photo / Photo gallery |
| Q / R | Lean left / right |
| Esc | Pause, or close an open screen |

All of these are `FKey` properties on `HF_HorrorPlayerController` and can be rebound in the Details panel.

### Fully equipped character

`HF_DemoCharacter` extends `HF_HorrorCharacter` with the remaining player-side systems: flashlight, breath hold, objective tracker, map, crafting, examine, endings, EMF, spirit box, thermometer, photo capture, evidence, heartbeat, breathing audio, injury, fear post-process, hallucination, monologue and weapon socket. `HF_GameMode` uses it as the default pawn.

It is a subclass rather than a change to the base character, so a project that already added these components by hand does not end up with two of each.

---

# 📝 Changelog

## Version 1.1

### New

- **Demo Level Builder** — generates a seven-room playable demo with one button.
- **Mission Director** — objective chain, plus seeding for crafting recipes, evidence definitions and map bounds.
- **Demo Overlay** — objectives and a live control legend on screen.
- **Fully equipped demo character** with every player-side system attached.
- **Central HUD style asset** (`HF_HUDStyle`) with Classic, Minimal, Cinematic and High Contrast presets. Every widget reads from it, so the whole interface re-themes from one Data Asset.
- **Six new screens**: map, crafting, journal, save slots, photo gallery and investigation tool readout.
- **Generic interactable prop** (`HF_InteractablePropActor`) — assign a mesh, type a prompt, tick the reactions. Covers notes, pickups, keys, levers and scenery without Blueprints.
- **Examinable and level exit actors**, and a placement helper that keeps props clear of doorways while letting doors and barricades stand in them.
- **Photo history** — the capture component now keeps every photo instead of overwriting one render target, which is what makes a gallery possible.
- **Main menu game mode** (`HF_MainMenuGameMode`) — a game mode for a menu level. Add it and the menu appears, takes UI input, shows the cursor and spawns no pawn. Title, tagline, version line, studio line, website button and start level are all fields on it.
- **Menu buttons now do something by default.** Start opens `StartLevelName`, Exit quits, and a Website button appears only when a URL is filled in. The existing delegates still broadcast, so projects already bound to them keep working.
- **Examine mode rebuilt.** The game pauses while an object is held, the object is centred on its own bounds rather than its pivot, and the mouse turns it freely in any direction with no gimbal lock and no dead zone. `ResetExamineRotation()` returns it to the angle it was picked up at.

- **Adjustable eye height** (`EyeHeight` on `HF_HorrorCharacter`), default 74 above the capsule centre.
- **Input actions that need to survive a pause now do.** Interact, pause, inventory, notes, map, journal, evidence board and photo gallery all keep working while the game is paused.

### Fixed

- **The plugin did not compile.** `HF_WeaponSocketComponent` referenced three properties that did not exist on `UHF_WeaponComponent`.
- **Ten interactable actors had no components at all** — no mesh, no collision, no interaction volume. Dropping one into a level produced nothing visible or usable.
- **Hiding spots and barricades did not implement the interaction interface**, so their mechanics were unreachable without custom Blueprints.
- **Hiding did not move the player.** Entering a wardrobe set a flag while the pawn stayed standing in the open.
- **Notes and dialogue displayed placeholder text.** Opening a note showed the panel without ever passing the note's contents to it.
- **The inventory panel never showed its contents.** Nothing connected the component to the widget, and the slot's item name was accepted and discarded.
- **Inventory USE and DROP buttons were not connected.**
- **Batteries did not reach the flashlight.** `ReplaceBattery` existed and was never called.
- **The generator could not be fuelled.** `AddFuel` existed and was never called, so a generator requiring fuel could never start.
- **Locked doors ignored their key lock component**, so keys could not be used.
- **Leaning changed a number and nothing else.** The component computed an offset that was never applied to the camera.
- **The EMF reader only scanned pawns**, so apparitions and haunted objects were invisible to it.
- **The apparition had no mesh assigned**, so appearing revealed an invisible object.
- **Six screens had no way to be opened** and the investigation tools had no activation key.
- **Popups had no close binding.** Opening the combination lock left the player with no way back to the game.

- **The flashlight spotlight was never attached.** `SetupAttachment` has no effect at runtime, so the light sat at a fixed point in the world instead of in the player's hands. It now attaches to the camera and can be aimed.
- Debug placeholder text was being shown to players through the note reader.
- **Clicking an inventory slot crashed the game.** The HUD handled the slot-clicked event by calling the very function that had broadcast it, so every click recursed until the stack overflowed. The panel now refreshes instead, and the widget guards against re-entry so no listener can repeat the mistake.
- **Combination locks did nothing.** The keypad collected the code and broadcast it to a delegate nobody subscribed to, so the lock was never told. It now drives the lock: a wrong code is rejected on the readout, and the right one solves the puzzle and fires its event, which is what makes hidden keys and rewards appear.
- **Held objects drifted and shook.** The examined object was teleported to a world position once per tick, which lands a frame behind the camera transform used to render. It is now parented to the camera, so it moves with the view in the same frame.
- **Held objects could not be put down.** Fixed together with the two problems above.
- **Looking up and down could stop working entirely.** A camera fallback path detached the camera from the spring arm, and since a Character capsule only ever yaws, all pitch was silently lost. The camera now stays on the arm, and the view no longer sinks when the capsule shrinks on crouch.
- **Gameplay input was lost when a level was opened from a menu.** Mouse capture and lock live on the viewport, which outlives a level change, so arriving from a UI-only menu left the cursor loose and keys unreachable. It read as a dead flashlight and an inventory that would not open. The player controller now claims game input for itself on BeginPlay.
- **Full HUD bars looked empty.** A bar at 100% was drawn at 15% opacity, which on a dark HUD is indistinguishable from drained. That behaviour is off by default now and stays available as a property.
- **Boarded windows built their planks inside the wall** and read as if the window had been boarded from outside. They now sit on the player's face of the wall.

## Engine support

Built for UE 5.5, 5.6, 5.7 and 5.8. The plugin ships as pure C++ with no content, which is deliberate: a `.uasset` or `.umap` can only be opened by the engine version that saved it, so shipping content

would tie one download to a single version of Unreal. Code compiles against whichever version you are on, so one product covers all four.

Note for 5.8: `TWeakObjectPtr` no longer provides `operator bool`, so validity must be tested with `IsValid()`.

### Known limitations

- Demo geometry and props use engine primitives as placeholders and are meant to be replaced with your own art.

---

## ❓ FAQ / Troubleshooting

### HUD doesn't show

Make sure your GameMode has `HF_HorrorHUD` as HUD Class.

### Camera doesn't rotate with mouse

Make sure you're using `HF_HorrorPlayerController` — it auto-creates Enhanced Input.

### AI enemy doesn't move

You need a **Nav Mesh Bounds Volume** in your level. Build → Build Paths. Press P to visualize.

### Systems don't respond to difficulty changes

Call `GameMode→SetDifficulty()` — it broadcasts through EventHub.

### Module disabled but component still runs

Module checks happen in BeginPlay. Restart PIE after changing settings.

### Flashlight doesn't toggle

FlashlightComponent is auto-created on HorrorCharacter. Make sure you're using `HF_HorrorCharacter` as Default Pawn Class.

---

**Ultimate Horror Framework**

Built by **Adrenaline Games (Erak_Dev)**

123 C++ classes · 16,700+ lines · 23 modules · Multiplayer ready

adrenalinegames.pl · support@adrenalinegames.pl