# 💀 Death & Respawn System

Complete Death/Respawn Mechanics for Unreal Engine 5

v1.0.0 | UE 5.6 - 5.7+

---

## 📖 Overview

Death & Respawn System is a comprehensive, production-ready plugin that handles all aspects of death and respawn mechanics in your Unreal Engine game. From simple instant respawns to complex checkpoint systems and arcade-style lives, this plugin has you covered.

> 💡 **Pure C++ Design:** This plugin provides only backend logic. You create your own UI widgets, visual effects, and sounds. This gives you complete creative control while we handle the complex death/respawn state management.

### What's Included

- **4 C++ Classes:** 1,869 lines of production-ready code
- **5 Respawn Modes:** Checkpoint, Instant, Lives, Permadeath, Manual
- **Complete Documentation:** 1,393 lines (Quick Start, FAQ, API Reference)
- **Blueprint-Friendly:** 30+ parameters, 15+ events
- **No Dependencies:** Works with any health/inventory system

### Perfect For

🎮

#### Action Games

Souls-like combat with checkpoint-based respawns

🏃

### Platformers

Quick respawns with lives system for arcade feel

### Roguelikes

Permadeath with death markers and item drops

### RPGs

Complex respawn logic with XP penalties

## ✨ Key Features

### Death Detection

- **Automatic Detection:** Monitors health and triggers death at threshold
- **Manual Triggering:** Call death from any Blueprint/C++ code
- **Ragdoll Physics:** Optional physics-based death animations
- **Death Penalties:** Item drops, XP loss, progress reset
- **Death Markers:** Dark Souls-style corpse run mechanics

### 5 Respawn Modes

### 1. Checkpoint Mode

Respawn at last activated checkpoint. Perfect for action games and Souls-likes.

- Place checkpoint actors in your levels
- Auto-activation on player overlap
- One-time use or reusable checkpoints
- Save system integration hooks

### 2. Instant Mode

Respawn instantly at death location. Great for fast-paced games.

- Configurable respawn delay
- Post-respawn invulnerability
- Automatic health restoration

### 3. Lives Mode

Arcade-style lives system with game over on zero lives.

- Configurable max lives (default: 3)
- Extra life pickups support
- Global or per-player tracking
- Game over event handling

### 4. Permadeath Mode

No respawn - true roguelike experience.

- Single life gameplay
- Game over on death
- Perfect for hardcore modes

### 5. Manual Mode

Complete control - respawn only when you call it.

- Custom respawn logic
- Cutscene integration
- Story-driven respawns

### Checkpoint System

- **Drag & Drop:** Place checkpoint actors in levels
- **Auto-Activation:** Triggers when player enters zone
- **Visual Feedback:** Blueprint events for particles/sounds
- **Save Integration:** Hooks for save game systems
- **Editor Icons:** Easy to see checkpoints in editor

### Lives System

- **Arcade-Style:** Traditional lives counter (3 lives default)
- **Global Tracking:** Lives persist across level loads
- **Shared Lives:** Co-op mode with shared lives pool
- **Per-Player:** Individual lives for multiplayer
- **Extra Lives:** Easy to add 1-up pickups

### Advanced Features

- **Invulnerability Periods:** Post-respawn protection (default: 2 seconds)
- **Death Location Markers:** Spawn markers for corpse run mechanics
- **Item Drops:** Drop specified items on death
- **XP/Currency Loss:** Configurable penalty percentage

- **Progress Tracking:** Respawn countdown progress (0.0 - 1.0) for UI
- **Health Restoration:** Automatic full health on respawn

### Blueprint Integration

- **30+ Parameters:** All exposed to Blueprints
- **15+ Events:** Complete customization hooks
- **Pure Functions:** Easy queries (IsDead, GetLives, etc)
- **No UI Included:** You create UI using provided events

# 🚀 Quick Start Guide

## Step 1: Install Plugin

1. Extract plugin to `YourProject/Plugins/DeathRespawnSystem/`
2. Open project in Unreal Engine
3. Go to **Edit** → **Plugins**
4. Search for "Death & Respawn System"
5. Enable plugin and restart editor

## Step 2: Add Component to Character

1. Open your character Blueprint
2. Click **Add Component**
3. Search for "Death Respawn Component"
4. Add it to your character

> ✅ **That's it!** The component is now attached to your character and ready to use.

## Step 3: Configure Basic Settings

Select the Death Respawn Component and configure:

### Death Settings:

- **Auto Detect Death:** Enable
- **Health Threshold:** 0.0 (death when health = 0)
- **Respawn Delay:** 3.0 seconds

### Respawn Settings:

- **Respawn Mode:** Checkpoint (or your preferred mode)
- **Restore Health On Respawn:** Enable
- **Invulnerability Duration:** 2.0 seconds

### Step 4: Connect to Health System

When player takes damage and health reaches 0:

```
On Take Damage Event
→ Subtract Health
→ If Health ≤ 0
   → Get Component (Death Respawn Component)
   → Trigger Death
```

### Step 5: Add Checkpoints (Optional)

1. In Content Browser, search for "Checkpoint Actor"
2. Drag it into your level
3. Position where you want respawn point
4. Adjust trigger zone size if needed

### Step 6: Create Death Screen UI

Create a Widget Blueprint for death screen:

```
Event Graph (in Character):
Get Component (Death Respawn Component)
→ On Death Triggered (Event)
→ Create Widget (WBP_DeathScreen)
→ Add to Viewport
```

### Step 7: Test!

1. Play in Editor (PIE)
2. Damage your character until health = 0
3. Should trigger death and respawn after delay

> 📚 **Need More Help?** Check the complete **Quick Start Guide** in `Documentation/QUICK_START.md` included with the plugin.

## ⚙️ Configuration Guide

### Death Settings

| Parameter | Type | Default | Description |
|---|---|---|---|
| **Auto Detect Death** | bool | true | Automatically detect death when health ≤ threshold |
| **Health Threshold** | float | 0.0 | Health value that triggers death |
| **Enable Ragdoll** | bool | false | Enable physics-based death |
| **Death Animation Duration** | float | 2.0 | Time before respawn starts (seconds) |
| **Destroy On Death** | bool | false | Destroy actor instead of respawning |

### Respawn Settings

| Parameter | Type | Default | Description |
|---|---|---|---|
| **Respawn Mode** | enum | Checkpoint | How actor respawns (5 modes) |
| **Respawn Delay** | float | 3.0 | Delay before respawn (seconds) |
| **Respawn Location Offset** | Vector | (0,0,100) | Offset from spawn point |
| **Invulnerability Duration** | float | 2.0 | Post-respawn invulnerability (seconds) |
| **Restore Health On Respawn** | bool | true | Full health on respawn |

### Lives System

| Parameter | Type | Default | Description |
|---|---|---|---|
| **Enable Lives System** | bool | false | Use arcade-style lives |
| **Max Lives** | int32 | 3 | Maximum lives |
| **Game Over On Zero Lives** | bool | true | Trigger game over at 0 lives |

### Death Penalties

| Parameter | Type | Default | Description |
|---|---|---|---|
| **Drop Items On Death** | bool | false | Drop configured items |
| **XP Loss Penalty** | float | 0.0 | XP loss percentage (0.0-1.0) |
| **Spawn Death Marker** | bool | false | Spawn marker at death location |

### Common Configurations

### Souls-Like Configuration:

```
Respawn Mode: Checkpoint
Respawn Delay: 3.0
Enable Ragdoll: true
Spawn Death Marker: true
XP Loss Penalty: 0.5 (50% XP loss)
```

### Arcade Platformer:

```
Respawn Mode: Lives
Enable Lives System: true
Max Lives: 3
Respawn Delay: 1.0
Enable Ragdoll: false
```

### Roguelike:

```
Respawn Mode: Permadeath
Drop Items On Death: true
Reset Progress On Death: true
```

### Fast-Paced Action:

```
Respawn Mode: Instant
Respawn Delay: 0.5
Invulnerability Duration: 3.0
Enable Ragdoll: false
```

## 💻 API Reference

### DeathRespawnComponent Functions

### TriggerDeath(bool bForceImmediate)

```
UFUNCTION(BlueprintCallable, Category = "Death & Respawn")
void TriggerDeath(bool bForceImmediate = false);
```

Manually trigger death. **This is the main way to kill actors!**

### TriggerRespawn()

```
UFUNCTION(BlueprintCallable, Category = "Death & Respawn")
void TriggerRespawn();
```

Manually trigger respawn (useful for Manual respawn mode).

### SetCheckpoint(FVector Location, FRotator Rotation)

```
UFUNCTION(BlueprintCallable, Category = "Death & Respawn")
void SetCheckpoint(FVector Location, FRotator Rotation);
```

Set checkpoint location for respawn.

### AddLife(int32 Amount)

```
UFUNCTION(BlueprintCallable, Category = "Death & Respawn")
void AddLife(int32 Amount = 1);
```

Add extra life (1-up pickups).

### GetCurrentLives()

```
UFUNCTION(BlueprintPure, Category = "Death & Respawn")
int32 GetCurrentLives() const;
```

Returns current lives remaining.

### IsDead()

```
UFUNCTION(BlueprintPure, Category = "Death & Respawn")
bool IsDead() const;
```

Check if actor is currently dead.

### IsInvulnerable()

```
UFUNCTION(BlueprintPure, Category = "Death & Respawn")
bool IsInvulnerable() const;
```

Check if actor is invulnerable (post-respawn).

### GetRespawnProgress()

```
UFUNCTION(BlueprintPure, Category = "Death & Respawn")
float GetRespawnProgress() const;
```

Get respawn countdown progress (0.0 - 1.0). Perfect for UI progress bars!

### Blueprint Events

| Event | Parameters | When Fired |
|---|---|---|
| **OnDeathTriggered** | DeadActor, RespawnDelay | When death occurs |
| **OnRespawnStarted** | Actor, RespawnLocation | When respawn begins |
| **OnRespawnComplete** | Actor | When respawn finishes |
| **OnLivesChanged** | NewLives, MaxLives | When lives count changes |
| **OnGameOver** | Actor | When lives reach 0 |
| **OnCheckpointSet** | Location, Rotation | When checkpoint activated |
| **OnInvulnerabilityStarted** | Duration | Post-respawn invulnerability |
| **OnInvulnerabilityEnded** | None | Invulnerability expires |

### CheckpointActor Functions

### ActivateCheckpoint(AActor* ActivatingActor)

```
UFUNCTION(BlueprintCallable, Category = "Checkpoint")
void ActivateCheckpoint(AActor* ActivatingActor);
```

Manually activate checkpoint for specific actor.

### GetRespawnLocation()

```
UFUNCTION(BlueprintPure, Category = "Checkpoint")
FVector GetRespawnLocation() const;
```

Get checkpoint's respawn location (with offset applied).

### Example: Creating Death Screen UI

```
// In Character Blueprint Event Graph:
Event Begin Play
→ Get Component (Death Respawn Component)
→ On Death Triggered (Event)
   → Create Widget (WBP_DeathScreen)
   → Add to Viewport

// In WBP_DeathScreen Widget:
Event Construct
→ Bind to Component's OnRespawnProgress
   → Update Progress Bar (0.0 - 1.0)
```

### Example: Lives Display

```
// In HUD Widget:
Event Construct
→ Get Component (Death Respawn Component)
→ On Lives Changed (Event)
   → Set Text: "Lives: " + NewLives
```

> 📚 **Complete API:** See `Documentation/API_REFERENCE.md` for all functions, properties, and events with detailed examples.

# ❓ Frequently Asked Questions

## General Questions

### Q: Do I need to know C++ to use this?

**A:** No! Everything is Blueprint-friendly. You can use it entirely through Blueprints.

### Q: Does this include UI?

**A:** No. This is pure C++ logic. You create your own UI widgets using the provided events.

### Q: What does "Pure C++" mean?

**A:** We provide backend logic only. You create UI, visual effects, and sounds. This gives you full creative control.

### Q: Does it work with multiplayer?

**A:** Currently single-player only. Multiplayer support may be added in future versions.

## Setup Questions

### Q: How do I install the plugin?

**A:** Extract to `YourProject/Plugins/`, enable in Plugins menu, restart editor.

### Q: Can I use this with my existing health system?

**A:** Yes! Call `TriggerDeath()` when your health system detects health = 0.

### Q: Do I need a specific character class?

**A:** No! Works with any character or actor. Just add the component.

## Configuration Questions

### Q: Which respawn mode should I use?

**A:**

- **Action games:** Checkpoint
- **Fast shooters:** Instant
- **Arcade platformers:** Lives
- **Roguelikes:** Permadeath
- **Custom logic:** Manual

### Q: How do checkpoints work?

**A:** Drag checkpoint actor into level. When player enters trigger zone, checkpoint activates. On death, player respawns there.

### Q: Can I add extra lives (1-up)?

**A:** Yes! Call `AddLife(1)` on the component when player picks up extra life.

## Technical Questions

### Q: What engine versions are supported?

**A:** Unreal Engine 5.6 and 5.7+. Choose the correct ZIP for your version.

### Q: What platforms are supported?

**A:** Windows 64-bit currently. More platforms may be added.

### Q: Does it work with AI characters?

**A:** Yes! Add component to any actor that needs death/respawn.

### Q: Can I modify the source code?

**A:** Yes! Full C++ source included. Modify as needed.

## Troubleshooting

### Q: Death doesn't trigger automatically

**A:** Make sure:

- Auto Detect Death is enabled
- Your character has `GetHealth()` function

- OR call `TriggerDeath()` manually when health = 0

### Q: Checkpoint doesn't activate

**A:** Check:

- Auto Activate is enabled on checkpoint
- Trigger zone collision is QueryOnly
- Trigger zone size is large enough
- Player has collision enabled

### Q: Ragdoll doesn't work

**A:** Requires:

- Character has Physics Asset
- Skeletal mesh has Simulate Physics enabled
- Collision properly configured

> **More Questions?** Check `Documentation/FAQ.md` for 50+ answered questions, or email **support@adrenalinegames.pl**

## Support & Resources

### Getting Help

Need assistance with Death & Respawn System? We're here to help!

#### Email Support

**support@adrenalinegames.pl**

Response time: Within 48 hours

#### Complete Documentation

1,393 lines of comprehensive guides included

Quick Start, FAQ, API Reference

#### Bug Reports

Priority handling for critical issues

Regular updates and fixes

### What We Can Help With

- ✅ Configuration questions
- ✅ Integration with your health system
- ✅ Troubleshooting issues
- ✅ Bug reports
- ✅ Feature requests
- ✅ Blueprint examples

### Documentation Included

- **Quick Start Guide:** Step-by-step setup (365 lines)
- **FAQ:** 50+ answered questions (451 lines)
- **API Reference:** Complete function/event docs (577 lines)
- **README:** Overview and quick reference
- **CHANGELOG:** Version history and updates

### What to Include When Reporting Issues

1. **Description:** What's the problem?
2. **Steps to Reproduce:** How can we recreate it?
3. **Configuration:** Your component settings
4. **Engine Version:** UE 5.6 or 5.7?
5. **Screenshots:** If relevant

### Additional Resources

- **Website:** adrenalinegames.pl
- **Fab Marketplace:** Death & Respawn System on Fab
- **Email:** support@adrenalinegames.pl

> ⚠️ **Note:** We cannot help with general Unreal Engine tutorials, custom game-specific implementation, or other plugins/systems. Our support is specifically for Death & Respawn System.

### Planned Updates

- **v1.1:** Multiplayer replication support, additional respawn effects hooks
- **v1.2:** Advanced camera system for death, spectator mode integration
- **v2.0:** Team-based respawn modes, respawn wave system

## 🚀 Ready to Get Started?

Add professional death/respawn mechanics to your game!

$19.99 | Includes full C++ source + 1,393 lines documentation

---

**Death & Respawn System** v1.0.0

Created by **Erak Dev** | Adrenaline Games



Website Fab Marketplace Support