ZOMBIE ENGINE STUDIO

ULTIMATE DEVELOPER GUIDE

Complete System Documentation • Step-by-Step Tutorials • API Reference

Version 1.0 | Unreal Engine 5.x

 © 2026 dztfix - All Rights Reserved

TABLE OF CONTENTS

 © 2026 dztfix - All Rights Reserved

 © 2026 dztfix - All Rights Reserved

CHAPTER 1: INTRODUCTION

1.1 What is Zombie Engine Studio?

Zombie Engine Studio (ZESP) is a comprehensive, production-ready framework designed specifically for creating zombie survival games in Unreal Engine 5. Unlike simple asset packs that provide static meshes or animations, ZESP is a complete game system that handles everything from player controls to zombie AI, from weapon mechanics to wave-based gameplay.

The system was designed with one core philosophy: Data-Driven Design. This means that instead of creating hundreds of Blueprint classes for different weapons, zombies, or items, you create Data Assets that define their properties. A single weapon Blueprint can become a pistol, rifle, shotgun, or anything else simply by swapping its Data Asset.

1.2 Key Features

- Complete Wave-Based Survival System with configurable difficulty scaling
- Data-Driven Weapon System supporting ranged, melee, and explosive weapons
- Advanced Zombie AI with perception (sight, sound, damage) and behavior trees
- Grid-Based Inventory System with drag-and-drop support
- Wall Buy and Mystery Box mechanics for classic zombie-mode gameplay
- Modular HUD System with customizable widgets
- Full Multiplayer Support with replication-ready components

1.3 System Requirements

Requirement	Minimum	Recommended
Unreal Engine	5.7	5.7+

 © 2026 dztfix - All Rights Reserved

CHAPTER 2: INSTALLATION & SETUP

2.1 Installing the Plugin

Follow these exact steps to install ZESP into your project:

1. Locate your Unreal Engine project folder on disk.
2. If not already present, create a folder named 'Plugins' in your project root.
3. Copy the entire 'ZombieEngineStudio' folder into the 'Plugins' directory.
4. Launch your Unreal Engine project. You will be prompted to compile the plugin. Click 'Yes'.
5. Wait for the compilation to complete. This may take 2-5 minutes depending on your hardware.
6. Once complete, verify by opening Edit → Plugins and searching for 'Zombie Engine Studio'.

> □ **TIP:** If compilation fails, ensure you have Visual Studio 2022 with the 'Game Development with C++' workload installed.

2.2 Project Structure

After installation, the plugin files are organized as follows:

Path	Contents	Purpose
`Plugins/ZombieEngineStudio/`	`Root`	`Plugin root directory`
`Content/ZESP/`	`Assets`	`All Blueprint and Data Asset content`
`Content/ZESP/Blueprints/`	`Blueprints`	`Core game logic Blueprints`
`Content/ZESP/Data/`	`Data Assets`	`Weapon, Zombie, and Item definitions`
`Content/ZESP/UI/`	`Widgets`	`HUD, Menus, and Inventory UI`
`Source/ZombieEngineStudio/`	`C++ Code`	`Plugin source code`
`Source/.../Public/`	`Headers`	`Public API headers (.h files)`

 © 2026 dztfix - All Rights Reserved

`Source/.../Private/`	`Implementation`	`Private implementation (.cpp files)`

2.3 Creating Your First Level

To set up a level that uses ZESP:

7. Create a new level (File → New Level → Empty Level).
8. Add a NavMeshBoundsVolume that covers your entire playable area.
9. Press 'P' to visualize the navigation mesh. It should appear green.
10. Place at least one Player Start actor in the level.
11. Create a Blueprint child of 'BP_ZombieSpawnPoint' and place several in the level.
12. Open World Settings and set the GameMode Override to 'BP_ZombieWaveGameMode'.

> **WARNING:** Zombies CANNOT MOVE without a valid NavMesh. Always add a NavMeshBoundsVolume first!

 © 2026 dztfix - All Rights Reserved

CHAPTER 3: WAVE GAME MODE

3.1 Understanding the Wave System

The wave-based game mode is controlled by AZE_ZombieWaveGameMode. This class manages the entire lifecycle of a survival match, from spawning zombies to tracking player deaths.

The wave system operates on a simple state machine with the following states:

State	Description	Transitions To
`WaitingToStart`	`Initial state before first wave`	`WaveInProgress`
`WaveInProgress`	`Zombies are spawning and active`	`WaveCompleted or GameOver`
`WaveCompleted`	`All zombies killed, shop phase`	`WaveInProgress (next wave)`
`GameOver`	`All players dead`	`N/A (Restart)`
`Won`	`Final wave completed`	`N/A`

3.2 Wave Settings Data Asset

All wave behavior is configured via UZE_WaveGameModeSettings. Create one by:

13. Right-click in Content Browser → Miscellaneous → Data Asset.
14. Select 'ZE_WaveGameModeSettings' as the class.
15. Name it 'DA_MyWaveSettings'.
16. Configure the settings as described below.

Wave Settings Properties

Property	Type	Description
`FirstWaveDelay`	`Float`	`Seconds before the first wave starts after match begin.`

 © 2026 dztfix - All Rights Reserved

TimeBetweenWaves	Float	Seconds of 'shop time' between waves.
BaseZombiesPerWave	Int	Number of zombies in wave 1.
ZombieCountMultiplier	Float	Each wave multiplies zombie count by this value. Example: 1.2 = +20% per wave.
MaxConcurrentZombies	Int	Maximum zombies alive at once. Prevents lag on weak hardware.
SpawnRate	Float	Seconds between each zombie spawn during a wave.
ZombieTypes	Array	List of UZE_ZombieDataAsset to spawn. Random selection per spawn.
ZombieClass	Class	The Blueprint class to spawn (e.g., BP_ZombieCharacter).
bDirectChaseMode	Bool	If true, zombies always chase nearest player without needing perception.
StartingWeapons	Array	Weapon Data Assets given to players at match start.
StartingGold	Int	Gold given to players at match start.

3.3 Difficulty Scaling

Each wave can automatically scale zombie difficulty. The following multipliers are applied:

Multiplier	Effect	Example (Wave 5, 1.1x Multiplier)
HealthMultiplier	Increases zombie MaxHealth	100 HP → 161 HP
DamageMultiplier	Increases zombie AttackDamage	15 DMG → 24 DMG
SpeedMultiplier	Increases zombie RunSpeed	450 → 726

 © 2026 dztfix - All Rights Reserved

CHAPTER 4: ZOMBIE AI SYSTEM

4.1 AI Controller Overview

Every zombie is controlled by AZE_ZombieAIController. This controller handles perception (detecting players) and drives the Behavior Tree that controls zombie actions.

Perception Senses

Sense	Purpose	Key Property
Sight	Detects players in line of sight	SightRadius (from ZombieStats)
Hearing	Detects gunfire and player noise	HearingRange (from ZombieStats)
Damage	Wakes up when shot	Always active

4.2 Zombie Data Asset

Zombie types are defined by UZE_ZombieDataAsset. Each asset represents a unique zombie variant.

17. Right-click in Content Browser → Miscellaneous → Data Asset.
18. Select 'ZE_ZombieDataAsset' as the class.
19. Name it (e.g., 'DA_Zombie_Walker', 'DA_Zombie_Runner').

Zombie Stats Properties

Property	Type	Range	Description
MaxHealth	Float	1+	Total hit points before death.
bIsInvincible	Bool	–	If true, takes no damage. Use for boss phases.
CorpseLifespan	Float	0+	Seconds before ragdoll body disappears.

 © 2026 dztfix - All Rights Reserved

WalkSpeed	Float	0+	Patrol/idle movement speed.
RunSpeed	Float	0+	Chase movement speed.
MeleeDamage	Float	0+	Damage dealt to player per attack hit.
AttackRange	Float	0+	Distance from player to trigger attack.
AttackCooldown	Float	0.1+	Seconds between attacks.
AttackSphereRadius	Float	0+	Radius of the attack hit detection sphere.
SightRadius	Float	0+	How far the zombie can see.
LoseSightRadius	Float	0+	Distance to lose track of player.
ViewAngle	Float	0–360	Field of view in degrees.
HearingRange	Float	0+	How far the zombie can hear gunshots.
MinGoldReward	Int	0+	Minimum gold on kill.
MaxGoldReward	Int	0+	Maximum gold on kill.
LootDropChance	Float	0–1	Probability to roll loot table on death.
LootTable	Array	–	List of possible item drops.

4.3 Loot System

Each zombie can drop loot on death. The system works as follows:

20. On zombie death, system checks LootDropChance (e.g., 0.3 = 30% chance).
21. If successful, system iterates through LootTable entries.
22. Each entry has its own DropChance. First successful roll spawns the item.
23. The loot actor spawns at the zombie's death location.

 © 2026 dztfix - All Rights Reserved

LootTable Entry Property	Type	Description
LootClass	TSubclassOf<AActor>	The actor to spawn (e.g., BP_PowerUp_MaxAmmo).
DropChance	Float (0–1)	Probability for this specific item. 0.1 = 10%.

 © 2026 dztfix - All Rights Reserved

CHAPTER 5: WEAPON SYSTEM

5.1 Weapon Types

ZESP supports multiple weapon archetypes, all controlled by UZE_WeaponDefinition:

Type	Fire Mode	Example
Rifle	Automatic / Semi-Automatic	AK-47, M4A1
Pistol	Semi-Automatic	Glock, Desert Eagle
Shotgun	Pump / Semi-Auto with Pellets	SPAS-12, Double Barrel
Sniper	Bolt-Action / Semi-Auto with Scope	AWP, Barrett
SMG	High Fire Rate Auto	MP5, Uzi
LMG	Very High Capacity Auto	M249, RPK
Melee	Trace-Based Swing	Knife, Bat, Sword
Grenade	Throwable Explosive	Frag, Molotov

5.2 Creating a New Weapon

Follow these exact steps to create a new ranged weapon:

24. Import your weapon mesh (FBX) into Content Browser.
25. Open the Skeleton and add sockets: 'Muzzle' (bullet spawn), 'Eject' (casing spawn).
26. Create an Animation Blueprint for the weapon (idle, fire, reload).
27. Right-click → Miscellaneous → Data Asset → Select 'ZE_WeaponDefinition'.
28. Name it 'DA_Weapon_MyRifle'.
29. Configure all properties as described below.

Combat Settings

Property	Type	Description

 © 2026 dztfix - All Rights Reserved

Damage	Float	Base damage per hit. Headshots use HeadshotMultiplier.
HeadshotMultiplier	Float	Damage multiplier for head bone hits (e.g., 2.0 = double damage).
FireRate	Float	Rounds per minute (RPM). 600 = 10 shots/second.
Range	Float	Maximum distance for line trace (hitscan). Beyond this, no hit registers.
bIsAutomatic	Bool	If true, holding fire continues shooting. If false, one shot per click.

Spread & Accuracy

Property	Type	Description
BaseSpread	Float	Initial inaccuracy in degrees when not moving/shooting.
SpreadPerShot	Float	Inaccuracy added per consecutive shot.
MaxSpread	Float	Maximum inaccuracy cap.
SpreadRecovery	Float	Degrees per second of accuracy recovery when not shooting.
MovementSpreadMultiplier	Float	Multiplier applied when player is moving.
AimingSpreadMultiplier	Float	Multiplier applied when aiming down sights (usually < 1.0).

Recoil

Property	Type	Description
RecoilPitch	Float	Vertical kick per shot (positive = up).
RecoilYaw	Float	Horizontal kick per shot (randomized ±).
RecoilRecovery	Float	Speed of automatic recoil compensation.

 © 2026 dztfix - All Rights Reserved

AimingRecoilMultiplier	Float	Multiplier applied when ADS (usually < 1.0).

Ammo & Reload

Property	Type	Description
bUsesAmmo	Bool	If false, weapon has infinite ammo (e.g., melee).
MagazineSize	Int	Bullets per magazine/clip.
MaxReserveAmmo	Int	Maximum ammo player can carry. -1 for infinite.
ReloadTime	Float	Duration of reload animation in seconds.
bReloadWhileAiming	Bool	If true, player can reload while ADS.

 © 2026 dztfix - All Rights Reserved

CHAPTER 6: INVENTORY SYSTEM

6.1 System Architecture

The inventory system is component-based. Each player has a UZE_InventoryComponent attached, which stores a grid of items.

Items are defined by Data Assets (UZE_ItemDefinition) and instances are stored as runtime structs (FZE_InventoryItem).

6.2 Item Definitions

Property	Type	Description
DisplayName	FText	Name shown in UI.
Description	FText	Tooltip description.
Icon	TSoftObjectPtr<UTexture2D>	Inventory grid image.
Category	EZE_ItemCategory	None, Weapon, Ammo, Consumable, Armor, Material.
Rarity	EZE_ItemRarity	Common, Uncommon, Rare, Epic, Legendary.
SizeX / SizeY	Int	Grid cell dimensions.
bStackable	Bool	Can multiple instances share a slot?
MaxStack	Int	Maximum count per stack.
Weight	Float	Physical weight (for encumbrance systems).
Value	Int	Gold value for shops.

6.3 Special Item Types

Consumables (UZE_ConsumableDefinition)

Property	Type	Description
HealAmount	Float	HP restored on use.
StaminaAmount	Float	Stamina restored on use.

 © 2026 dztfix - All Rights Reserved

EffectDuration	Float	Duration of buff effects.

Armor (UZE_ArmorDefinition)

Property	Type	Description
Defense	Float	Flat defense value.
DamageReduction	Float (0–100)	Percentage damage blocked.
BiteResistance	Float (0–100)	Zombie bite resistance.

 © 2026 dztfix - All Rights Reserved

CHAPTER 7: INTERACTION SYSTEM

7.1 The Interactable Interface

Any actor can become interactable by implementing IZE_InteractableInterface. This interface exposes:

Function	Return Type	Purpose
`GetInteractText()`	`FText`	`Text shown on HUD (e.g., 'Press F to Buy').`
`CanInteract()`	`Bool`	`Returns true if interaction is currently allowed.`
`Interact()`	`Void`	`Called when player presses interact key.`
`OnFocusStart()`	`Void`	`Called when player looks at the actor (for highlighting).`
`OnFocusEnd()`	`Void`	`Called when player looks away.`

7.2 Wall Buy Tutorial

Follow these steps to create a Wall Buy that sells a weapon:

30. Create a Blueprint child of 'BP_WallBuy'.
31. In the Details panel, set 'Weapon' to your weapon Data Asset (e.g., DA_Weapon_Shotgun).
32. Set 'Cost' to the gold price (e.g., 1000).
33. Set 'AmmoCost' to the price for refilling ammo if player already owns the weapon.
34. Place the Blueprint in your level near a wall.
35. Optionally, add a static mesh (weapon silhouette) as a visual indicator.

 TIP: Use the 'InteractText' property to customize the HUD prompt (e.g., 'Buy Shotgun [1000]').

 © 2026 dztfix - All Rights Reserved

7.3 Highlight System

Method	Property	Description
Custom Depth	bUseCustomDepthHighlight	Enables Custom Depth Stencil for post-process outlines.
Overlay Material	bUseOverlayHighlight	Applies an overlay material to mesh components.
Stencil Value	HighlightStencilValue	The stencil value to use (0–255).

 © 2026 dztfix - All Rights Reserved

CHAPTER 8: USER INTERFACE

8.1 HUD Architecture

The player HUD is controlled by UZE_PlayerHUDWidget. This widget is designed for Blueprint extension.

The following widgets can be bound in your Blueprint child:

Widget Name	Type	Purpose
HealthBar	UProgressBar	Displays player health (0-100%).
Text_Hp	UTextBlock	Displays numeric HP value.
SprintBar	UProgressBar	Displays stamina / sprint energy.
AmmoText	UTextBlock	Displays 'Current / Reserve' ammo.
WeaponIcon	UImage	Displays current weapon icon.
Text_Gold	UTextBlock	Displays player gold amount.
Text_Wave	UTextBlock	Displays current wave number.
Text_ZombiesRemaining	UTextBlock	Displays zombies left in wave.
Timer_Buy	UTextBlock	Displays purchase feedback messages.

8.2 Customizing the HUD

36. Create a Blueprint child of 'WBP_PlayerHUD' (in Content/ZESP/UI/).
37. Open the Widget Designer.
38. Add/remove widgets as needed. Bind them using 'Is Variable' checkbox.
39. In your BP_PlayerController, set the 'HUDWidgetClass' to your new widget.

 © 2026 dztfix - All Rights Reserved

CHAPTER 9: CONTROL PANEL

9.1 Accessing the Control Panel

The ZESP Control Panel provides quick access to all global settings without editing Data Assets.

40. In the Editor, go to Edit → Project Settings.
41. Scroll down to 'Plugins' section.
42. Find 'Zombie Engine Studio' subsections.

9.2 Player Settings

Setting	Type	Default	Description
WalkSpeed	Float	400	Default movement speed.
SprintSpeed	Float	600	Speed when sprinting.
CrouchSpeed	Float	200	Speed when crouching.
MaxHealth	Float	100	Starting health.
MaxStamina	Float	100	Starting stamina.
StaminaDrainRate	Float	20	Stamina consumed per second while sprinting.
StaminaRegenRate	Float	15	Stamina recovered per second.
DeathWidgetClass	TSubclassOf	–	Widget to show on player death.

9.3 Combat Settings

Setting	Type	Default	Description
DefaultFireMode	Enum	Automatic	Default fire mode for new weapons.
HeadshotBoneName	FName	head	Bone name for headshot detection.

 © 2026 dztfix - All Rights Reserved

FriendlyFire	Bool	False	If true, players can damage each other.

9.4 Inventory Settings

Setting	Type	Default	Description
GridWidth	Int	10	Inventory grid columns.
GridHeight	Int	6	Inventory grid rows.
QuickbarSlots	Int	4	Number of quickbar weapon slots.
bShowItemTooltips	Bool	True	Show tooltips on hover.
bAllowDragDropToWorld	Bool	True	Allow dragging items out to drop.

 © 2026 dztfix - All Rights Reserved

CHAPTER 10: BLUEPRINT API REFERENCE

10.1 Player Character Functions

Function	Inputs	Outputs	Description
AddGold	Int Amount	Void	Adds gold to the player.
RemoveGold	Int Amount	Bool Success	Removes gold. Returns false if insufficient.
GetGold	None	Int	Returns current gold amount.
Heal	Float Amount	Void	Restores HP (clamped to MaxHealth).
TakeDamage	Float Damage	Void	Applies damage to the player.
IsAlive	None	Bool	Returns true if HP > 0.
Respawn	None	Void	Respawns the player at a random start point.

10.2 Inventory Component Functions

Function	Inputs	Outputs	Description
TryAddItem	ItemDef, Count	Int Overflow	Adds item. Returns leftover count if full.
RemoveItem	ItemDef, Count	Bool Success	Removes item from inventory.
HasItem	ItemDef	Bool	Checks if player has at least one of this item.
GetItemCount	ItemDef	Int	Returns total count of this item in inventory.
EquipWeapon	SlotIndex	Bool	Equips the weapon in the specified quickbar slot.

 © 2026 dztfix - All Rights Reserved

10.3 Game Mode Functions

Function	Inputs	Outputs	Description
StartWave	None	Void	Begins the current wave immediately.
EndWave	None	Void	Forces current wave to end.
GetCurrentWave	None	Int	Returns the current wave number.
GetZombiesRemaining	None	Int	Returns zombies left to kill.
GetWaveState	None	Enum	Returns WaitingToStart, WaveInProgress, etc.

10.4 Zombie Character Functions

Function	Inputs	Outputs	Description
TakeDamage	Float, Instigator	Void	Deals damage to the zombie.
Kill	None	Void	Instantly kills the zombie.
GetCurrentHealth	None	Float	Returns current HP.
IsAlive	None	Bool	Returns true if not dead.

 © 2026 dztfix - All Rights Reserved

CHAPTER 11: COMPLETE FILE PATHS

11.1 Blueprint Locations

Asset	Path
Player Character	Content/ZESP/Blueprints/Player/BP_PlayerCharacter
Player Controller	Content/ZESP/Blueprints/Player/BP_PlayerController
Zombie Character	Content/ZESP/Blueprints/Zombies/BP_ZombieCharacter
Zombie AI Controller	Content/ZESP/Blueprints/Zombies/BP_ZombieAIController
Wave Game Mode	Content/ZESP/Blueprints/Core/BP_ZombieWaveGameMode
Spawn Point	Content/ZESP/Blueprints/Zombies/BP_ZombieSpawnPoint
Wall Buy	Content/ZESP/Blueprints/Interaction/BP_WallBuy
Weapon Base	Content/ZESP/Blueprints/Weapons/BP_WeaponBase
Pickup Base	Content/ZESP/Blueprints/Interaction/BP_PickupBase

11.2 Data Asset Locations

Asset Type	Path
Weapon Definitions	Content/ZESP/Data/Weapons/
Zombie Definitions	Content/ZESP/Data/Zombies/
Item Definitions	Content/ZESP/Data/Items/
Wave Settings	Content/ZESP/Data/GameMode/
Consumables	Content/ZESP/Data/Items/Consumables/
Armor	Content/ZESP/Data/Items/Armor/

11.3 C++ Header Locations

Class	Header Path
AZE_PlayerCharacter	Public/Characters/ZE_PlayerCharacter.h
AZE_ZombieCharacter	Public/Zombies/ZE_ZombieCharacter.h
AZE_ZombieAIController	Public/Zombies/ZE_ZombieAIController.h
UZE_WeaponDefinition	Public/Inventory/ZE_WeaponDefinition.h
UZE_ItemDefinition	Public/Inventory/ZE_ItemDefinition.h
UZE_InventoryComponent	Public/Inventory/ZE_InventoryComponent.h

 © 2026 dztfix - All Rights Reserved

AZE_ZombieWaveGameMode	Public/Survival/ZE_ZombieWaveGameMode.h
UZE_WaveGameModeSettings	Public/Survival/ZE_WaveGameModeSettings.h

 © 2026 dztfix - All Rights Reserved

CHAPTER 12: PUBLISHING TO FAB

12.1 Preparing Your Plugin

Before submitting to the Unreal Engine Marketplace (Fab), you must clean and package your plugin correctly.

Step 1: Clean Temporary Files

43. Delete the 'Binaries' folder inside your plugin directory.
44. Delete the 'Intermediate' folder inside your plugin directory.
45. Delete any 'Saved' folders.
46. Delete any '.vs' or '.idea' folders (IDE caches).

Step 2: Fix Redirectors

47. In Content Browser, right-click on your Content folder.
48. Select 'Fix Up Redirectors in Folder'.
49. Wait for completion.

Step 3: Validate Assets

50. Open 'Asset Audit' window (Window → Developer Tools → Asset Audit).
51. Verify no 'Missing' or 'Broken' references.

12.2 Creating Documentation

52. Export this Word document as PDF: File → Export → Create PDF.
53. Place the PDF in your plugin's 'Docs' folder: Plugins/ZombieEngineStudio/Docs/
54. Alternatively, host it on Google Drive and link it in your product description.

12.3 Packaging

 © 2026 dztfix - All Rights Reserved

55. Zip only the 'ZombieEngineStudio' folder (not the entire project).
56. Ensure the .uplugin file is at the root of the zip.
57. Name the zip: 'ZombieEngineStudio_v1.0.zip'.

 WARNING: Never include your full project. Only the plugin folder should be zipped!

12.4 Store Listing Checklist

- Featured Image: 1920x1080 PNG
- Gallery Images: At least 4 screenshots showing gameplay
- Video: Optional but highly recommended (YouTube link)
- Description: Include feature list, system requirements, and support contact
- Documentation: Link to this PDF or embedded text

 © 2026 dztfix - All Rights Reserved

END OF DEVELOPER GUIDE

Thank you for choosing Zombie Engine Studio!

For support, contact: dztfix@example.com

 © 2026 dztfix - All Rights Reserved