

BossFight

BossFight

Gameplay Overview:

Combat Experience (Similar to Cuphead)

- **Movement:** WASD for directional movement.
- **Attacking :**
 - Left mouse button + attack key to shoot.
 - Mouse rotation controls shooting direction.
 - **Shift** for dashing.
- **Simple Controls:**
 - Shooting, movement, and bullet type switching.
- **Dynamic Combat Phases:**
 - Bosses transition between different attack patterns (e.g., enraged mode, stun states).

Skill System

- **Skill Activation:**
 - Skills are automatically unlocked based on **hit combos** (successful hits on valid targets).
 - **Player Customization:**
 - Players freely choose **the order of skill activation**.
-

How to use:

How to use

Player Character Configuration Guide

Users can configure the following parameters in the **BP_Player** blueprint:

▼ Player Attributes		
Fire Rate	0.2	
Max Energy	100.0	
Current Energy	100.0	
Sprint Speed Multiplier	10.0	
Damage Amount	1.0	
Max HP	100.0	
Current HP	100.0	

Combat Attributes

- **FireRate** – Shooting rate (shots per second).
- **MaxEnergy** – Maximum energy capacity.
- **CurrentEnergy** – Current energy level (initialized to match **MaxEnergy**), dynamically changes based on player actions.
- **SprintSpeedMultiplier** – Sprint speed (multiplied by the base movement speed).
- **DamageAmount** – Base attack damage.

Health System

- **MaxHP** – Maximum health points.
- **CurrentHP** – Current health (initialized to match **MaxHP**), dynamically updates during gameplay.

Usage Notes

- **Initial Values:**
 - **CurrentHP** and **CurrentEnergy** should start equal to **MaxHP** and **MaxEnergy** respectively.
- **Dynamic Adjustments:**
 - Energy depletes/replenishes based on actions (e.g., sprinting, skill usage).

- Health decreases when taking damage and can be restored via skills/items.
- **Tuning Suggestions:**
 - Balance **FireRate** and **DamageAmount** for desired combat pacing.
 - Adjust **SprintSpeedMultiplier** to control mobility trade-offs.

This configuration allows flexible customization of player stats to match gameplay design needs. 🎮

Boss Configuration Guide

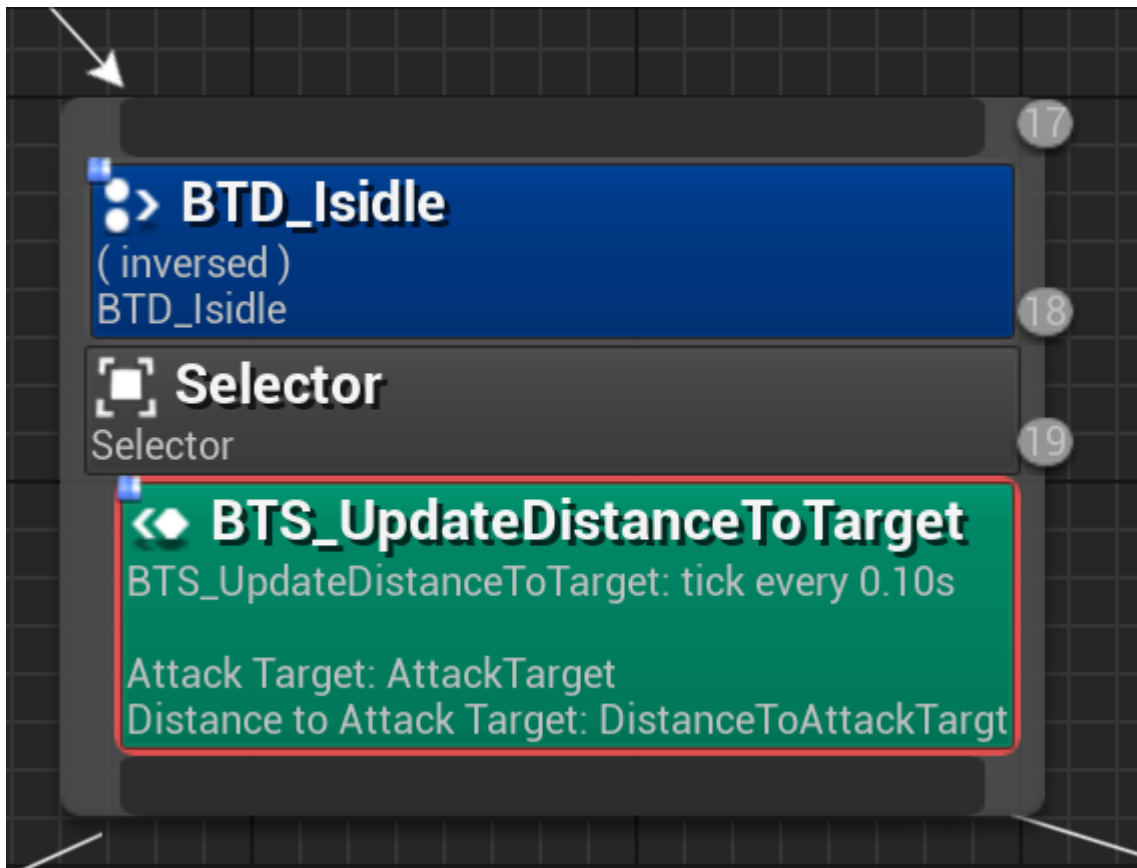
Users can configure the following in **BP_Boss_Melee**:

▼ Boss Attributes		
Over Drive Attack Damage Amount	15.0	
Over Drive Attack Power	10.0	
Damage Amount	10.0	
Jump Attack Damage Amount	30.0	
Jump Attack Power	30.0	
Over Drive Attack AOERadius	180.0	
Jump Attack AOERadius	300.0	
Over Drive Attack Time	8.0	
Max HP	150.0	↩
Current HP	150.0	↩

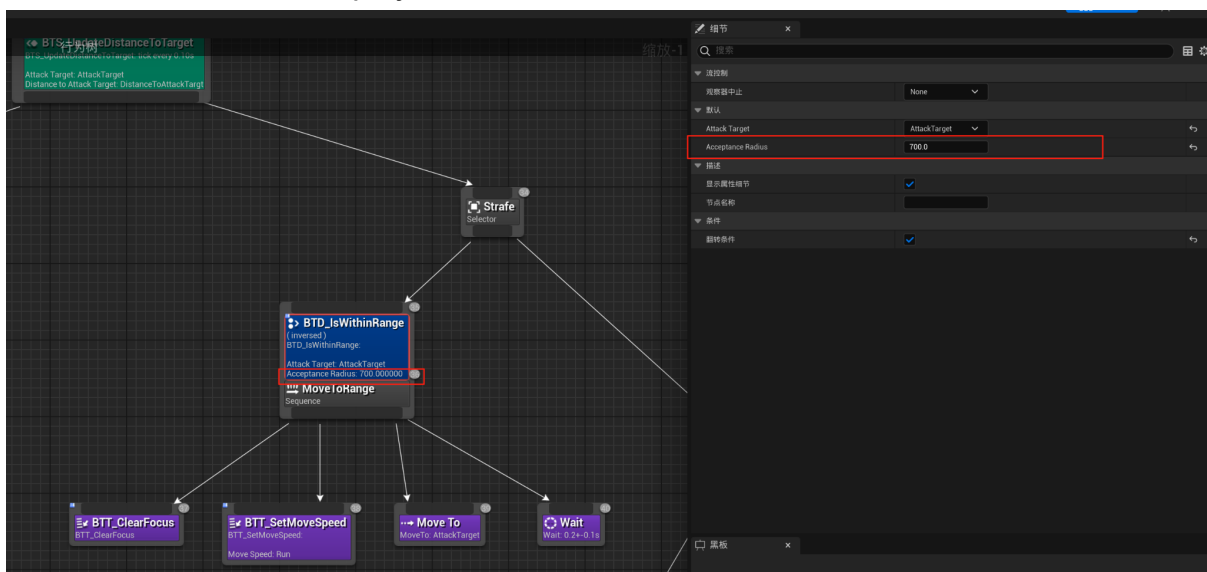
- **Health (HP), Attack Power, and Skill Damage**
 - **Over Drive Attack Damage Amount:** Damage value of the enraged attack.
 - **Over Drive Attack Power:** Knockback force of the enraged attack (pushes players away).
 - **OverDriveAttackAOERadius:** Damage range of the enraged attack.
 - **OverDriveAttackTime:** Duration of the enraged attack.
 - **Damage Amount:** Damage value of normal attacks.
 - **Jump Attack Damage Amount:** Damage value of the jump attack.
 - **Jump Attack Power:** Knockback force of the jump attack (launches players away).
 - **JumpAttackAOERadius:** Damage range of the jump attack.
 - **Max HP:** Maximum health points.

- **Current HP:** Current health points (initially matches Max HP and dynamically changes based on player actions).

Behavior Logic Setup in BT_Boss_Melee

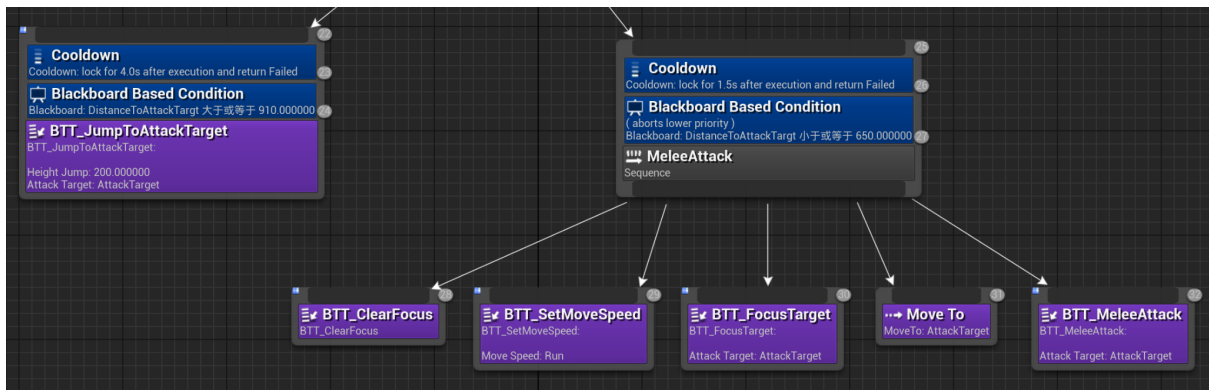


- **BTS_UpdateDistanceToTarget:** Determines the boss's behavior based on its distance from the player.

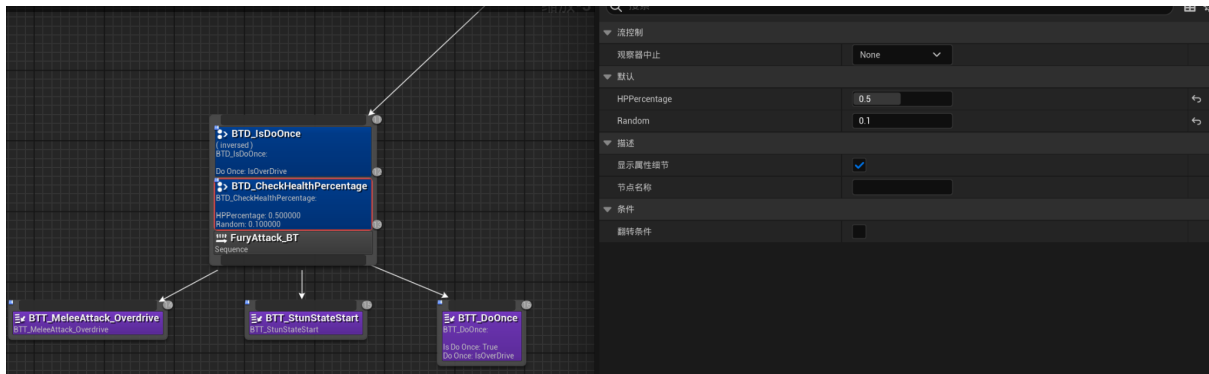


- **BTD_IsWithinRange:** If the boss's distance from the player exceeds the **Acceptance Radius**, it will run toward the player.

Skill Cooldown & Trigger Conditions



- Use **Cooldown** to control the internal cooldown of normal and jump attacks.
- Trigger skill conditions are set using the **blackboard key DistanceToAttackTarget**.



- **HPPercentage**: Triggers the enraged attack when the boss's HP falls below this percentage.

Phase 2 Combat Configuration Guide

Core Attributes	
Current HP	100.0
Max HP	100.0
Side Length Max	1000.0
Spawn Rate	15.0

Core Actor Settings (BP_CoreActor)

Configure core health, orb spawn behavior, and initial distance:

- **Max HP:** Maximum health points
- **Current HP:** Current health (initialized to match Max HP, dynamically modified by player actions)
- **Side Length Max:** Maximum distance between explosion orbs and the core
- **Spawn Rate:** Orb generation interval (in seconds)

▼ Core Buttle Attributes	
Max HP	8.0
Current HP	8.0
Rotation Speed	0.3
Self- Destruct Damage Amount	30.0
Damage Amount	10.0
AOERadius	300.0
Self- Destruct AOERadius	300.0

Orb Projectile Settings (BP_AOE_CoreBullet)

Configure bullet health, rotation, collision damage, and self-destruct parameters:

- **Max HP:** Maximum health points
- **Current HP:** Current health (initialized to match Max HP)
- **Rotation Speed:** Orb rotation speed (degrees/sec)
- **Damage Amount:** Damage dealt on player contact
- **AOERadius:** Bullet spread radius after charged explosion
- **Self-Destruct Damage Amount:** Damage dealt by orb self-destruction
- **Self-Destruct AOERadius:** Blast radius of orb self-destruction

▼ 变量		+
▼ 组件		
Niagara		
Damage Amount	浮点	

Charged Explosion Settings (BP_AOE_Boom)

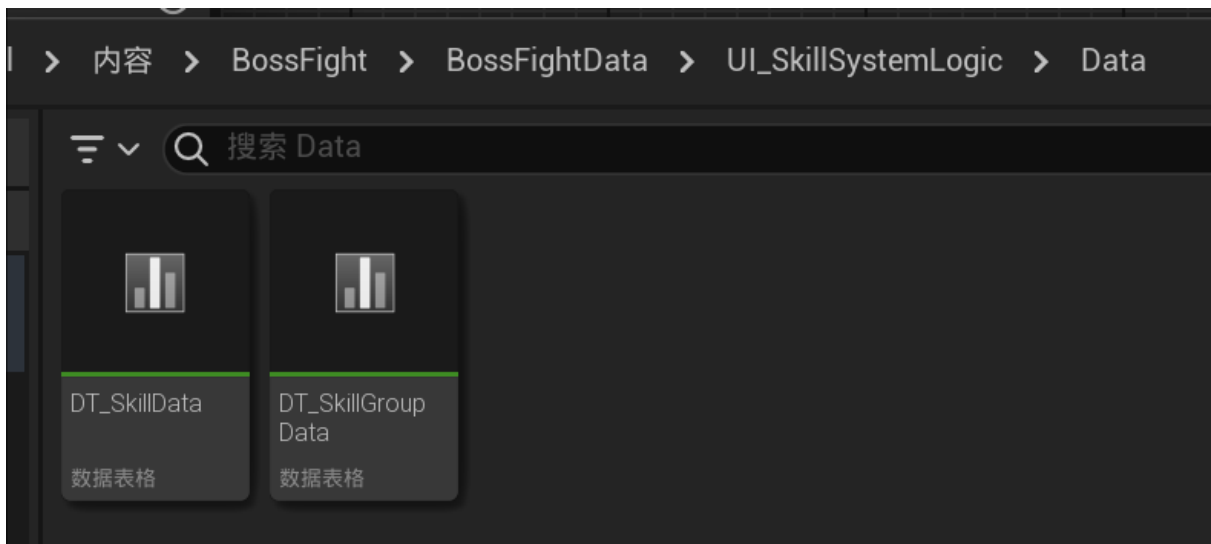
Configure radial bullet damage:

- **Damage Amount:** Damage per individual bullet in radial spread

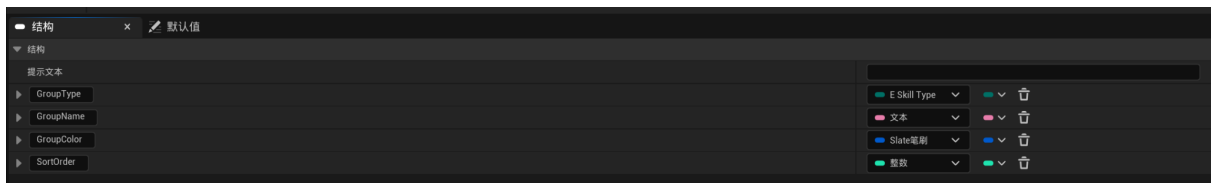
Skill Configuration Guide

Skills are pre-installed in the UI Skill Panel and are automatically triggered by the player character during combat.

UI Panel Skill Configuration



1. DT_SkillGroupData (Skill Categories)



Used to dynamically generate skill category panels in the UI.

Parameter	Description
-----------	-------------

GroupType	Skill category type (must be unique, no duplicates)
-----------	---

GroupName	Display name for the category (must be unique)
-----------	--

GroupColor UI color resource for the category panel

SortOrder Determines the order in which categories
appear (must match the number of categories)

2. DT_SkillData (Individual Skills)

结构

提示文本

SkillID

SkillName

SkillDesc

SkillIcon

GroupType

SkillLevel

bUnlocked

bInUse

InfoIcon

Grey_SkillIcon

命名

文本

文本

Slate笔刷

E Skill Type

整数

布尔

布尔

Slate笔刷

Slate笔刷

搜索

行号	SkillID	SkillName	SkillDesc	SkillIcon	GroupType	SkillLevel	bUnlocked	bInUse	InfoIcon	Grey_SkillIcon
1	001	BaseSkill001	001	001	{ "bIsDynamicallyLoaded": false, "bHasUIObject": false, "DrawAs": "Image", "Thermal": 1	1	True	False	{ "bIsDynamicallyLoaded": false, "bHasUIObject": false, "DrawAs": "Image",	{ "bIsDynamicallyLoaded": false, "bHasUIObject": false, "DrawAs": "Image",
2	002	BaseSkill002	002	002	{ "bIsDynamicallyLoaded": false, "bHasUIObject": false, "DrawAs": "Image", "Thermal": 1	True	False	{ "bIsDynamicallyLoaded": false, "bHasUIObject": false, "DrawAs": "Image",	{ "bIsDynamicallyLoaded": false, "bHasUIObject": false, "DrawAs": "Image",	
3	003	BaseSkill003	003	003	{ "bIsDynamicallyLoaded": false, "bHasUIObject": false, "DrawAs": "Image", "Thermal": 1	True	False	{ "bIsDynamicallyLoaded": false, "bHasUIObject": false, "DrawAs": "Image",	{ "bIsDynamicallyLoaded": false, "bHasUIObject": false, "DrawAs": "Image",	
4	004	BaseSkill004	004	004	{ "bIsDynamicallyLoaded": false, "bHasUIObject": false, "DrawAs": "Image", "Thermal": 1	True	False	{ "bIsDynamicallyLoaded": false, "bHasUIObject": false, "DrawAs": "Image",	{ "bIsDynamicallyLoaded": false, "bHasUIObject": false, "DrawAs": "Image",	
5	005	BaseSkill005	005	005	{ "bIsDynamicallyLoaded": false, "bHasUIObject": false, "DrawAs": "Image", "Cryonic": 1	False	False	{ "bIsDynamicallyLoaded": false, "bHasUIObject": false, "DrawAs": "Image",	{ "bIsDynamicallyLoaded": false, "bHasUIObject": false, "DrawAs": "Image",	

行编辑器

001

SkillID

SkillName

SkillDesc

SkillIcon

GroupType

SkillLevel

bUnlocked

bInUse

InfoIcon

Grey_SkillIcon

BaseSkill001

001

001

001

Thermal

1

☒

☐

☐

Defines skills under each category.

Parameter	Description
SkillID	Must match <code>SkillID</code> in <code>DT_RewardTemplateData</code>
SkillName	Skill name (must be unique)
SkillDesc	Description text (used in info panels)
SkillIcon	UI icon for the skill
GroupType	Must match <code>GroupType</code> in <code>DT_SkillGroupData</code> to appear in the correct category
SkillLevel	Skill level (for progression)
bUnlocked	Whether the skill is locked (<code>True/False</code>)
bInUse	Whether the skill is currently equipped (<code>True/False</code>)
InfoIcon	UI icon for detailed info panels
Grey_SkillIcon	Disabled-state UI icon (used when locked)

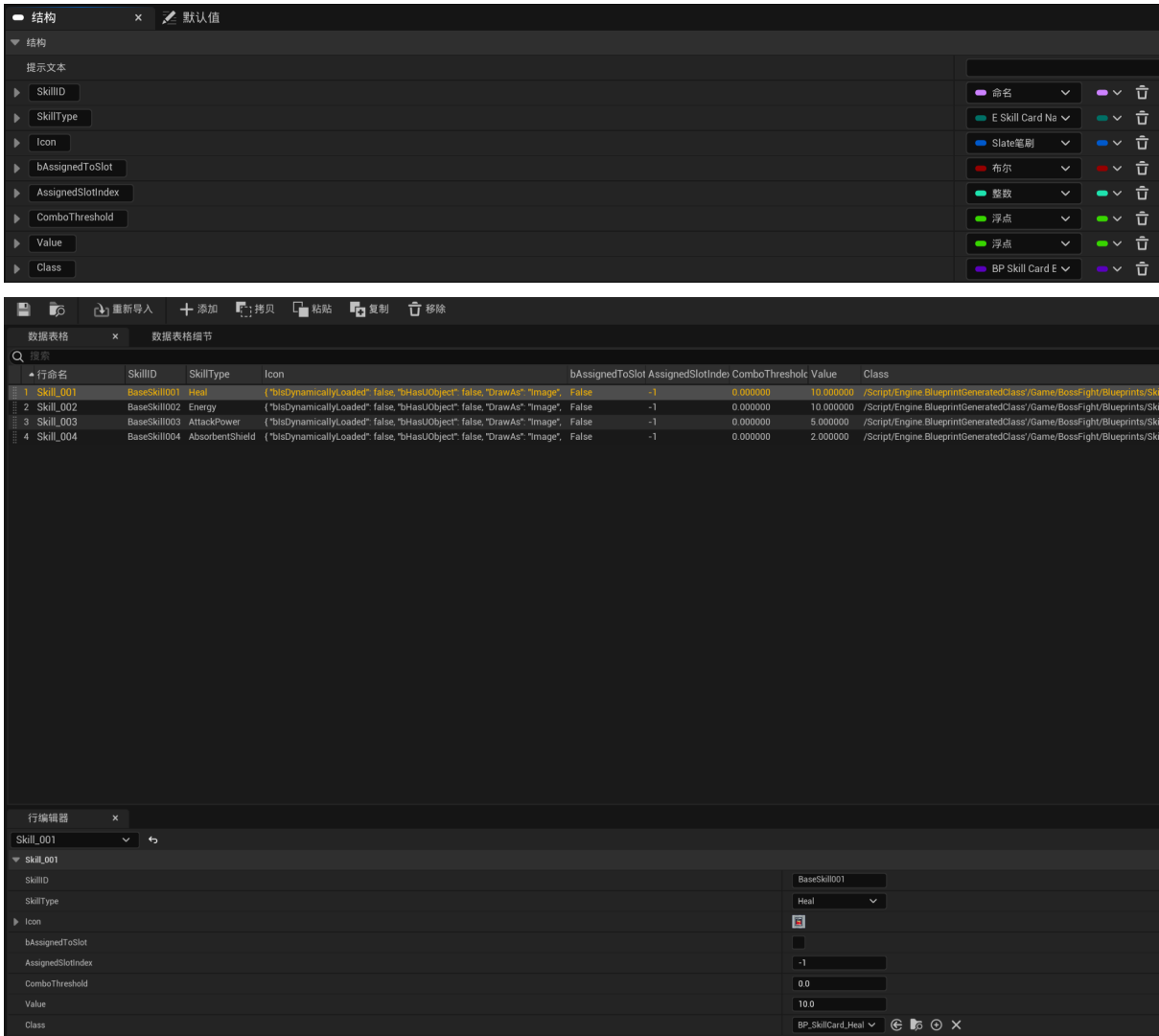
Important:

- `GroupType` in `DT_SkillGroupData` must match `SkillType` in `DT_SkillData` for proper UI display.

Player Combat Skill Configuration

(DT_RewardTemplateData)

Defines how skills behave when automatically triggered in combat.



Parameter

Description

SkillID Must match SkillID in DT_SkillData

SkillType Specific skill type (e.g., Heal, Damage Boost)

Icon Skill icon (for combat UI)

bAssignedToSlot	Whether the skill is assigned to an active slot (<code>True/False</code>)
AssignedSlotIndex	Which skill slot (0, 1, 2...) it occupies
ComboThreshold	Number of hit combos required to auto-trigger the skill
Value	Skill effect magnitude (e.g., <code>+20 HP</code> for heals, <code>+10 Damage</code> for attacks)
Class	References the BP_SkillCardBase subclass (e.g., <code>BP_SkillCard_Heal</code>) to determine ski

Key Notes:

1. Data Consistency:

- Ensure SkillID matches across **DT_SkillData** and **DT_RewardTemplateData**.

2. UI Hierarchy:

- Skills only appear in their designated category if **GroupType** (SkillData) matches **GroupType** (SkillGroupData).

3. Combat Automation:

- Skills trigger when the player reaches the **ComboThreshold** value.

This system allows flexible skill management while ensuring seamless combat integration. 🚀

Blueprint Architecture

Blueprint Architecture

All core game logic is stored in the **Blueprints** folder, categorized as follows:

Main Blueprint Classes

1. BP_Player (Player Character)

- **Movement:**
 - WASD movement + **Shift** for dash.
- **Attacks:**
 - Fires projectiles; bullet effects change with skill upgrades.
- **Skill System:**
 - Releases skills based on the **Skill Panel** configuration.
- **Interaction:**
 - Press **E** to open the **Skill Panel**.
- **Tunable Parameters:**
 - Damage, HP, stamina ("nitro"), dash speed, etc.

2. BP_Boss_Melee (Melee Boss)

- **Attacks:**
 - Standard attacks, jump attacks, enraged attacks, stun states.
- **Phase Transition:**
 - At **0 HP**, the boss enters **Phase 2**.
- **Tunable Parameters:**
 - Damage, HP, skill duration, etc.

3. Boss_Controller (Boss AI Controller)

- Initializes boss behavior.
- Detects the player.
- Controls **boss behavior switching**.

4. Boss Behavior Control

Behavior Tree (BT_Boss_Base)

- Behavior Tree Decorators (BTD_)
 - **BTD_CheckHealthPercentage`** (HP threshold checks).
 - **BTD_IsDoOnce`** (one-time skill execution).
 - **BTD_IsIdle`** (checks if boss is stationary).
 - **BTD_IsWeak`** (checks if boss is vulnerable).
 - **BTD_IsWithinRange`** (checks attack range).
- Behavior Tree Services (BTS_)

- **BTS_UpdateDistanceToTarget`** (tracks player distance).
- **Behavior Tree Tasks (BTT_)**
 - **BTT_ClearFocus`** (resets target focus).
 - **BTT_DoOnce`** (executes a skill once).
 - **BTT_FocusTarget`** (locks onto the player).
 - **BTT_JumpToAttackTarget`** (leap attack).
 - **BTT_MeleeAttack`** (standard melee).
 - **BTT_MeleeAttack_Overdrive`** (enraged melee).
 - **BTT_MoveTo`** (boss movement).
 - **BTT_OnDeath`** (death sequence).
 - **BTT_SetIdle`** (forces idle state).
 - **BTT_SetJumpAttack`** (configures jump attack).
 - **BTT_SetMoveSpeed`** (adjusts movement speed).
 - **BTT_StunStateStart`** (stuns the boss).
 - **BTT_Weak`** (triggers vulnerability).

Environment Query System (EQS)

- **BP_Boss_WarpPoint`** – Teleports boss to a predefined location after death.
- **EQS_Context_AttackTarget`** – Links to the boss's attack target.
- **EQS_RandomLocationAroundTheTarget`** – Generates random positions near the player.
- **EQS_Strafe`** – Calculates strafing positions.
- **EQS_Stun`** – Determines stun-related positions.

Blackboard (BB_Boss_Base)

- Stores AI decision-making variables.

5. Interactive Level Blueprints

- **BP_CoreActor`** – Player's Phase 2 attack target (adjustable HP).
- **BP_AOE_BP_CoreBullet`** – Protective explosive orbs:
 - Orbit the core.
 - Knock back & damage players on collision.
 - Periodically fire bullet spreads.
 - After several attacks, charge at the player with a shockwave.
- **BP_Door`** – Closes at combat start, opens after victory (with spikes restricting movement).
- **BP_Elevator`** – Boss entrance/exit mechanics.
- **BP_Obstacle/BP_Obstacle_S`** – Spikes that limit movement space.
- **BP_SkillTerminal`** – Press ****E**** to open the skill panel.

6. Projectile Blueprints

- **BP_Bullet_Base`** – Base projectile class.
- **BP_ShotBullet`** – Standard bullets.
- **BP_ShotBullet_AttackPower`** – Upgraded bullet effects.

7. AOE Skill Blueprints

- **BP_AOE_Base`** – Base AOE class.
- **BP_AOE_Blast`** – Explosion effects.
- **BP_AOE_CoreBulletBullet`** – Bullet spread from explosive orbs.
- **BP_AOE_CoreBulletBoom`** – Orb explosion.
- **BP_AOE_BossMeteorCrash`** – Boss's meteor strike.
- **BP_AOE_OverDriveAttack`** – Boss's enraged attack.

8. Player Combat Skills

- **BP_SkillCardBase`** – Base skill class.
- **BP_SkillCard_Heal`** – Healing skill.
- **BP_SkillCard_Energy`** – Stamina recovery.
- **BP_SkillCard_AttackPower`** – Damage boost.
- **BP_SkillCard_AbsorbentShield`** – Protective shield.

9. UI Blueprints (Core Functionality)

- **Data Flow:**
 - UI widgets (WBP) are preloaded at initialization.
 - Player interactions update data → UI refreshes.
 - On closing, data is saved to ****SaveData**** and reloaded next session.

Key UI Systems

- **BP_GameInstance_SkillManager`** – Handles data:
 - Registers data at startup.
 - Converts between data types.
 - Updates UI after player input.
 - Saves to **`SaveData`**.
- **BPFL_SkillManager`** – Manages UI representation:
 - Fetches data from **`BP\_GameInstance\_SkillManager`**.
 - Handles skill-slot interactions.
 - Displays skill details.
 - Updates skill slots & skill bar.
- **WBP_SkillConfigPanel`** – Skill customization UI:
 - Dynamically generates skill types.
 - Right-click for details.
 - Drag & drop skills into slots.
 - Swap/rearrange skills.
- **WBP_PlayerCharacter_Info`** – Player HUD:
 - Displays equipped skills.
 - Shows HP & stamina.
- **WBP_Boss_Info`** – Boss HUD:
 - Tracks boss HP.

- Indicates boss state (e.g., enraged, stunned).
-

Support Systems

- **BPC_DamageSystem`** – Calculates combat damage & healing.
 - **BPC_SkillManager`** – Tracks hit combos to auto-unlock skills.
-

Interfaces

- **BPI_BossAI`** – Base boss AI functions.
 - **BPI_BossAI_Melee`** – Melee boss extensions.
 - **BPI_ComboTarget`** – Validates combo targets.
 - **BPI_Damageable`** – Universal damage handling.
 - **BPI_Interactive`** – For interactive objects (e.g., UI pop-ups).
-

Data Tables

- **DT_SkillData`** – Skill attributes.
 - **DT_SkillGroupData`** – Skill categories.
 - **DT_RewardTemplateData`** – Skill reward templates.
-

Resource Directory Structure

- **BossCharacter`** – Boss models/animations.
- **Environments`** – Level assets.
- **Materials`** – Master materials.
- **Particle`** – VFX.
- **PlayerCharacter`** – Player assets.
- **UI_Image`** – UI textures (game instructions).

This structure ensures modularity and scalability for future updates. 🚀