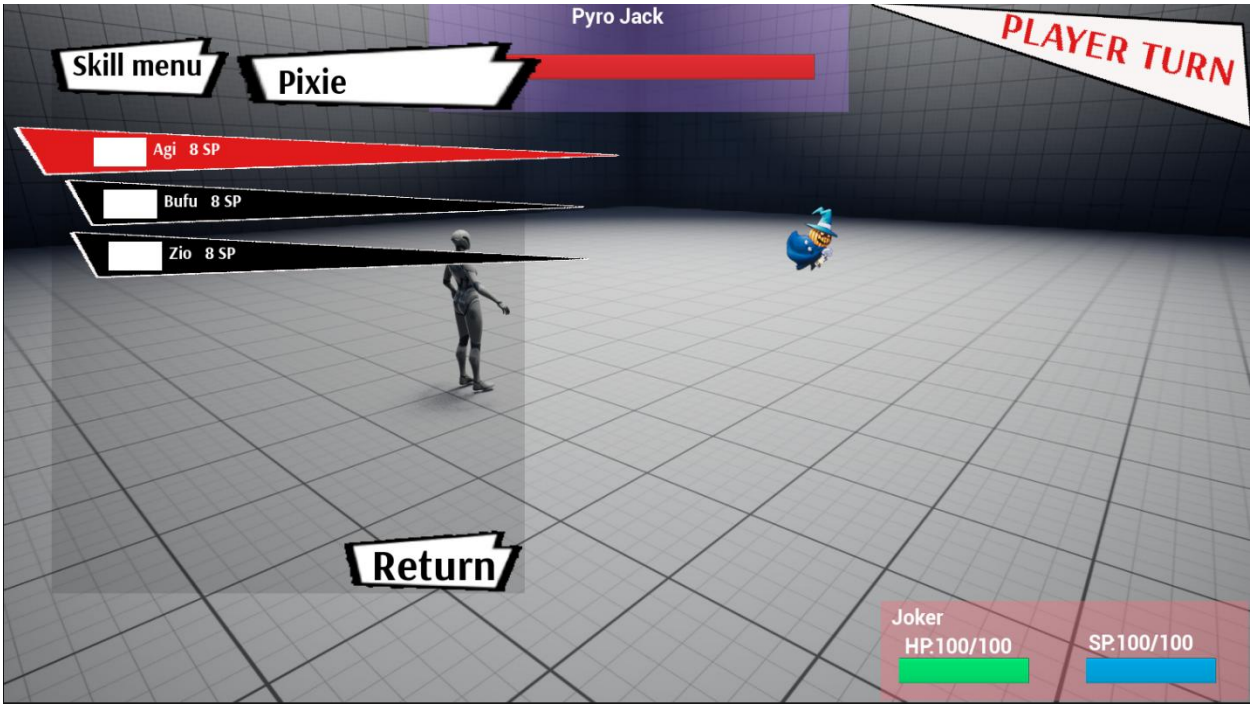
# How To: Turn-Based Combat System

Ruben 245015




## Table of Contents

How To: Turn-Based Combat System ..... 1
General Information ..... 2
Introduction ..... 2
The Core Loop ..... 3
Features Overview ..... 3
Intended Use Cases ..... 3
Limitations ..... 3
File Structure ..... 4
Quick Start Guide ..... 4
Adding the System to Your Project ..... 6
Migration ..... 6
Project Settings ..... 6
Game Instance Setup (if applicable) ..... **Error! Bookmark not defined.**

Creating Content ..... 6
Creating a Character ..... 6
Creating a Skill ..... 7
Creating an Element Type ..... 9
Creating an Encounter Trigger ..... 10
Creating a Combat Arena ..... **Error! Bookmark not defined.**
Tweaking the System ..... 11
Battle Logic ..... 11
BP_CombatManager Functions ..... 11
BP_OverworldEncounter ..... 11
Damage Formula ..... 12
Weakness Multipliers ..... 12
UI Customization ..... 13
CommonUI Widget Structure ..... 13
Adding Visuals to Skill Buttons ..... **Error! Bookmark not defined.**
Customizing Combat HUD ..... 14
Troubleshooting ..... 14
Common Issues ..... 14
Version History ..... 15

# General Information

## Introduction

The turn-based combat system is a Persona-inspired combat plugin for Unreal Engine 5. It features standard JRPG turn based combat mechanics you are probably familiar with including: elemental weaknesses/affinity system, the "One More" system (Exploiting a weakness lets the attacker take another turn), Guard (Reduce damage and immune to weaknesses) and fully customizable data tables that let you create characters, skills and element types.

# The Core Loop

- **Encounter:** Player triggers combat through overworld encounters
- **Parse information:** AC_CombatComponent values get read from both player and enemy to load in equipped characters, HP/SP values from player and HP value from enemy
- **Turn Order:** If player triggers combat, player always starts first. If there are multiple party members the equipped character determines turn order based on speed stat
- **Action Selection:** Player chooses to use skills or guard
    - **Skill Execution:** Skills are executed with damage calculations and weakness checks
    - **Guard:** Player guards, receives x0.4 less damage, is immune to weakness being exploited and skips their turn
- **One More:** Exploiting a combatants weakness grants an additional turn (Yes the enemy can one more you too!)
- **Resolution:** Combat ends when all enemies or party members are defeated

# Features Overview

- **BP_ Combat Manager:** Central blueprint handling all combat logic, turn flow, and state management
- **AC_Combatant Component:** Actor component storing character stats, skills, and combat data
- **Skill System:** Data-driven skills with elemental types, costs, and effects
- **Weakness System:** Elemental weakness/resistance chart with "One More" bonus turns
- **Defensive Mechanics:** Guard system for damage mitigation
- **CommonUI Integration:** Modern UI framework for menus and combat interface
- **Overworld Encounters:** Trigger system for initiating battles in the overworld

# Intended Use Cases

This system is designed for developers who want to add Persona-style turn-based combat to their projects. You can replace element type names, interactions with each other, sprites and looks however you seem fit. The current plugin is obviously more akin to Persona 5 style UI (Stylized) and theming (Character sprites, skill names, resource cost; HP/SP).

# Limitations

- **Single encounters:** Currently supports 1v1 Player vs enemy, (*Even though Combatants get loaded into an array time constraint unfortunately did not let me test for multiple combatants function but the architecture is there)

- **No VFX/SFX:** Visual and audio effects not included (Besides Images popping up in UI)
- **No multiplayer:** Single-player only
- **Maximum 8 skills**

---

# File Structure

| Folder | Contents |
|---|---|
| Blueprints | BP_CombatManager, AC_CombatantComponent, BPI_Combatant<br>BP_OverWorldEncounter |
| DataTables | DT_Skills, DT_Characters |
| Structures | FCharacterBase, FSkillBase |
| Enums | E_ElementTypes, E_AttackCost, E_SkillID |
| UI | WBP_BattleUI, WBP_VictoryScreen |
| UI/CommonUI | Style_EmptyColor, WBP_CombatChoice, WBP_SkillButtons, WBP_SkillsSelection, WBP_SelectionButton |
| Characters/Enemies | BP_Enemy_Shadow, BP_EvilSnowMan BP_PyroJack |
| Materials/Sprites | All the sprites used for UI and SpriteRenderers |

---

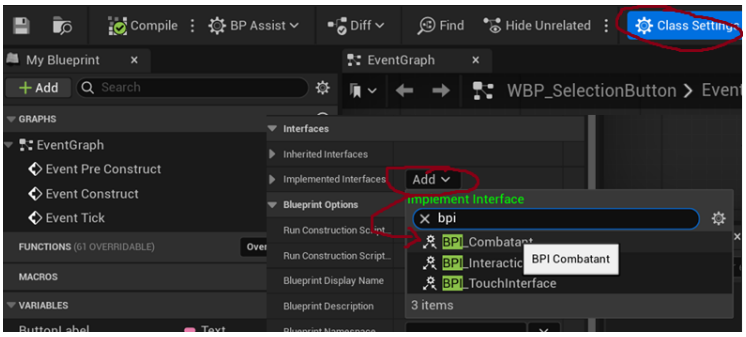
# Quick Start Guide

Get combat running in your project in minimal steps:

**Step 1:** Migrate the combat plugin folder to your project.

**Step 2:** Have your player character implement **BPI_Combatant** interface.

(YourPlayerBlueprint -> Class settings -> Interfaces -> Implemented interfae -> Add -> BPI_Combatant)



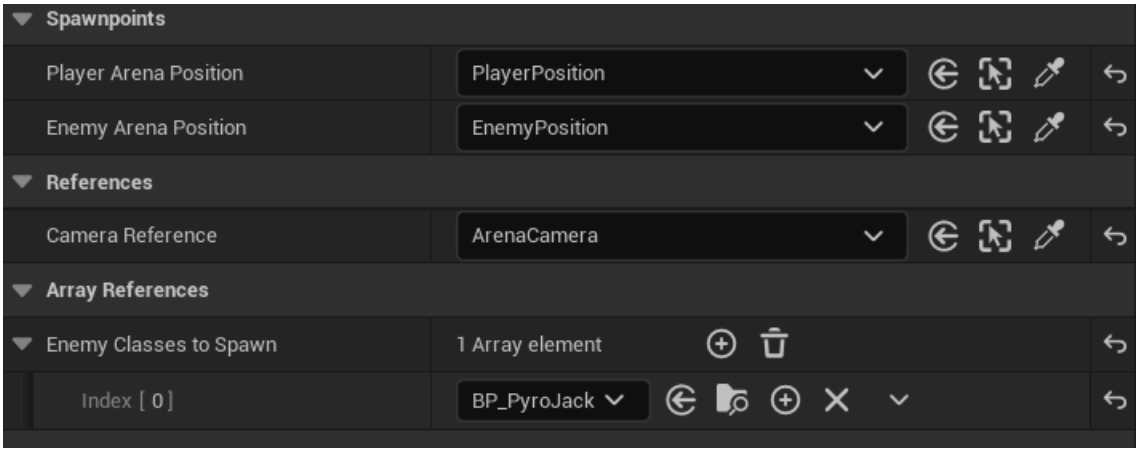

**Step 3:** Add **AC_CombatantComponent** to any actor that will participate in combat (player, enemies(To add to enemy to your liking make a child of BP_Enemy_Shadow to later load into BP_OverWorldEncounter)

Step 4: Populate the **AC_CombatantComponent**(s) with the values you want: HP, SP, EquippedCharacter *(This is the character you are equipping from **DT_Characters**, name is case and string sensitive! Make sure it's the exact same as the row name*) in both the player and enemy

**Step 4:** Add **BP_CombatManager** to your level. Only one per level is needed.

**Step 5:** Add **BP_OverworldEncounter** to trigger combat when the player overlaps.

**Step 6:** Setup **this BP_OverWorldEncounter** with:




- Two spawnpoint references [Actor transform reference, just plop down two actors
  *(This is where your player and enemy will be teleported to when combat starts)*
- ArenaCamera [Any camera, place one down near your arena/fight scene, set it up however youd like and just plug it in!)

- a **BP_Enemy_Shadow** type; this plugin comes with 3 presets (**BP_PyroJack**, **BP_EvilSnowman** and the base **BP_EnemyShadow**)

**Step 6:** Ensure your player's **AC_CombatantComponent** has a equipped character, hp and that this character has skills by double checking in DT_Skills

**Step 7:** Play and test! Walk into the encounter trigger to start combat.

---

# Adding the System to Your Project

## Migration

Right-click the combat plugin folder in your source project and select **Migrate**. Choose your target project's Content folder.

## Project Settings

**CommonUI plugin:** Top left corner of editor> Edit > Plugins > All plugins > Install CommonUI (You will have to restart)

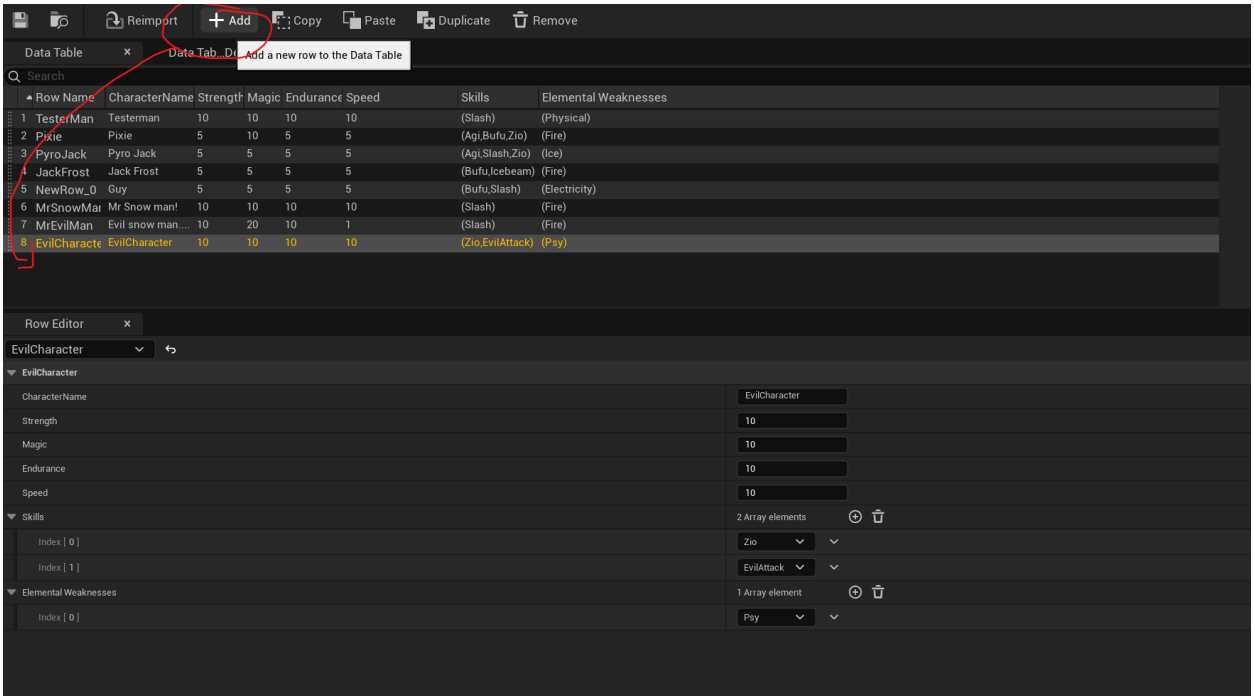
---

# Creating Content

## Creating a Character

Characters are defined in **DT_Characters** *(Blueprints>DataTables)*

**Step 1:** Open the data table and click **+ Add**.




**Step 2:** Name the row - this is the internal ID used to reference this character.

**Step 3:** Fill in the struct fields:

- **CharacterName:** Display name shown in UI
- **Stats:**
  - Strength: Determines power of attacks that cost HP to use
  - Magic: Same principle but for attacks that cost SP to use
  - Endurance: Acts as defense stat, determines how much damage you receive during damage calculation
  - Speed; Determines turn order but doesn't do anything as of (V1.0)
- **Skills:** Array of row names from DT_Skills this character can use (Pulls from E_SkillID) **Maximum of 8!!**
- **Elemental Weaknesses:** Determine what skills are calculated as super effective against this character

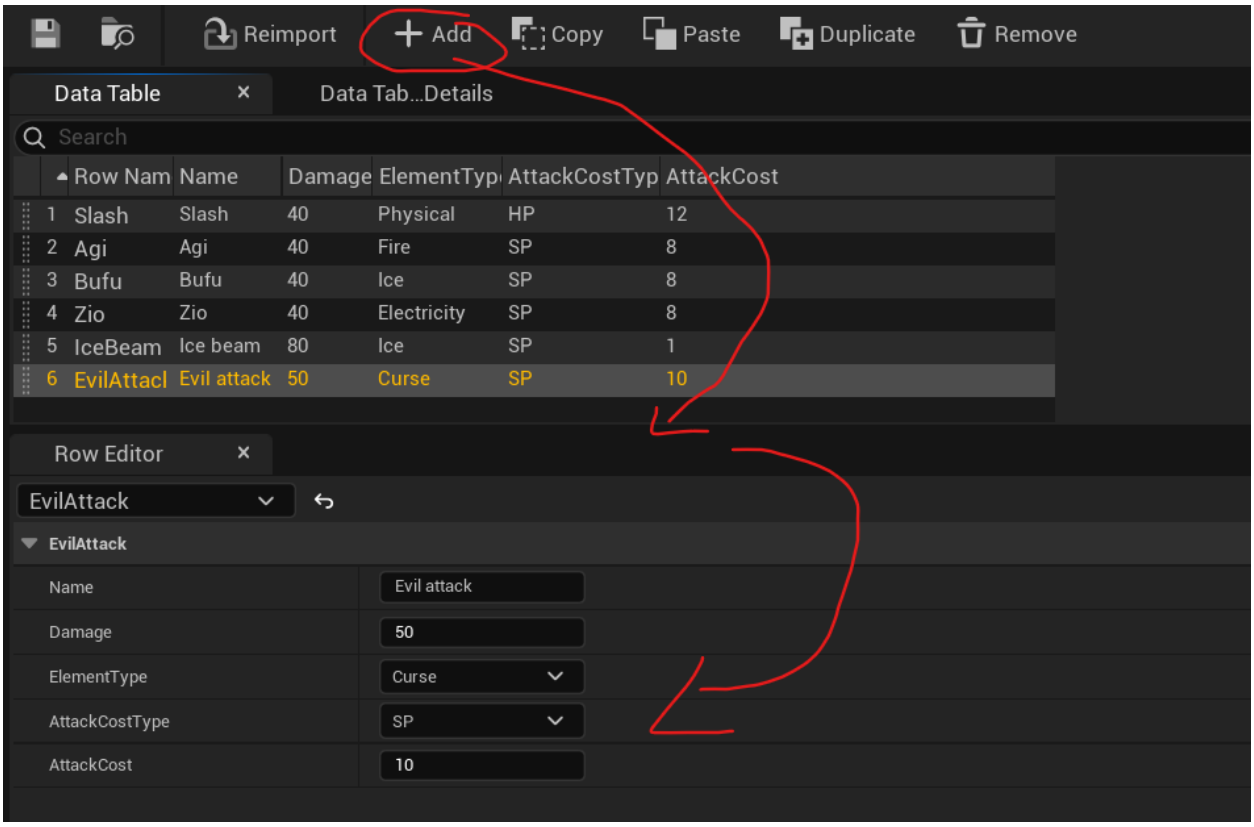
**Step 4:** Save the data table.
**Step 5**: Character is now ready to be used in any AC_CombatantComponent's "EquippedCharacter" Field

**(NOTE: USE ROW NAME NOT CHARACTERNAME!! IE; LittleGuy instead of Little Guy)**

# Creating a Skill

Skills are defined in **DT_Skills**. *(Blueprints>DataTables)*

**Step 1:** Open the data table and click **+ Add**.




**Step 2:** Name the row - this is how you reference the skill elsewhere.

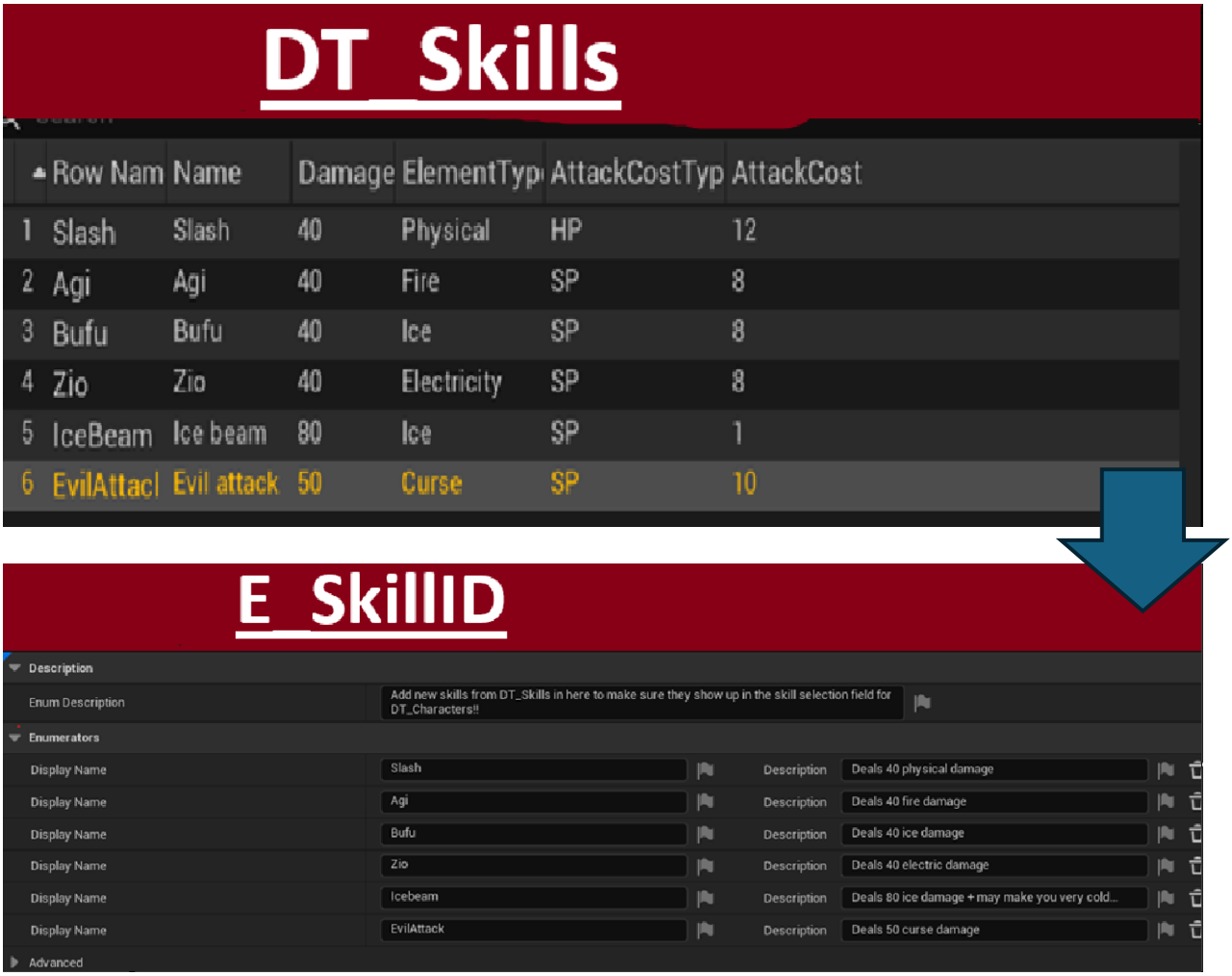
**Step 3:** Fill in the struct fields:

- **Name:** Display name shown in UI
- **Damage:** Base damage of skill
- **Element Type:** Fire, Ice, Electric, etc. (These are determined in E_Elementtypes)
- **AttackCostType:** Determines what resource combatant uses to use skill
    - **(*To repeat:** this also determines if strength or magic stat gets used for damage calculation! HP = Strength, SP = stat!)
- **Attack Cost:** How much HP/SP the attack costs

**Step 4: Go to E_SkillID (Location: Blueprint/Enums) VERY IMPORTANT!**

**Step 5**: Insert newly made skillname with the **EXACT SAME ROW NAME** as the one in DT_Skills into E_SkillID

**Step 6: Make sure the skills listed in DT_Skills match the exact same row order and name**




Step 7: If it is not the same; update accordingly. Update your newly made skill from DT_Skills into E_SkillID to update it correctly

Step 8: Save both DT_Skills and E_SkillID!

Step 9: Enjoy! You can now add this skill back at your character

# Creating an Element Type

Adding new elements  is quite simple! They are located in Blueprints>Enums>E_ElementTypes

**Step 1:** Open up E_ElementTypes

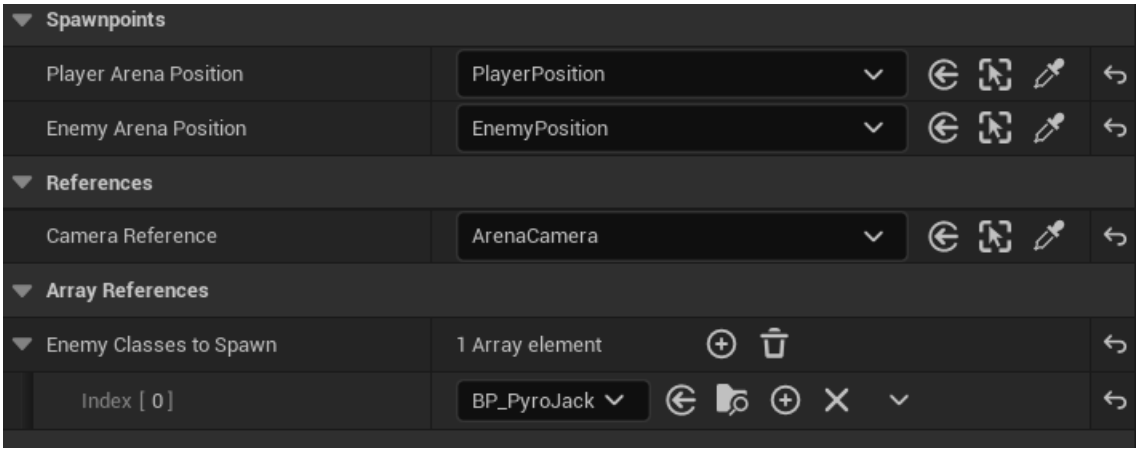
Step 2: Add your element and a description if you feel fancy

Step 3: Save and done! Update any characters weaknesses or skill's elemental attribute in their respective data tables. Told you it was easy 😊

# Creating an Encounter Trigger

**Step 1:** Place **BP_OverworldEncounter** in your level where you want combat to trigger.

**Step 2:** Configure the parameters in BP_OverworldEncounter under details panel:




- Two spawnpoint references [Actor transform reference, just plop down two actors
  *(This is where your player and enemy will be teleported to when combat starts)*
- ArenaCamera [Any camera, place one down near your arena/fight scene, set it up however youd like and just plug it in!)
- a **BP_Enemy_Shadow** type; this plugin comes with 3 presets (**BP_PyroJack**, **BP_EvilSnowman** and the base **BP_EnemyShadow**)
-

**Step 3:** Double check both combatants (BP_EnemyShadow and Player's AC_CombatantComponent) to see if everything is set properly

Step 4: You're done!

---

# Tweaking the System

## Battle Logic

### BP_CombatManager Functions

The combat manager contains several key functions you can modify:

#### InitializeCombat

- Called when combat begins (By BP_OverWorldEncounter)
- Sets up combatants, initializes turn order, transitions camera
- Modify to change combat setup behavior

#### StartTurn

- Called at the start of each turn
- Determines whose turn it is, refreshes UI
- Modify to add turn-start effects (buffs ticking down, poison damage, etc.)

#### ExecuteSkill

- Handles skill execution, damage calculation, weakness checks
- Triggers "One More" if weakness is hit
- Modify to change damage formulas or add skill effects

#### CheckWeakness

- Compares skill element against target's weaknesses
- Returns whether "One More" should trigger (by setting IsWeak if true)
- Modify to change weakness/resistance behavior

#### CheckForWin

- Called in-between executeskills to see if combatants are still alive
- Handles victory/defeat states, cleanup, return to overworld
- Modify to add rewards, experience, post-combat events

### BP_OverworldEncounter

#### Event ActorBeginOverlap (AmbushRadius attached to this actor)

- Triggered when player overlaps the collision volume

- Contains all Get, set player values logic and transitioning to bp_combatmanager initializecombat and arena/camera logic.
- Modify to add conditions (only trigger if player has certain item, time of day, etc.)

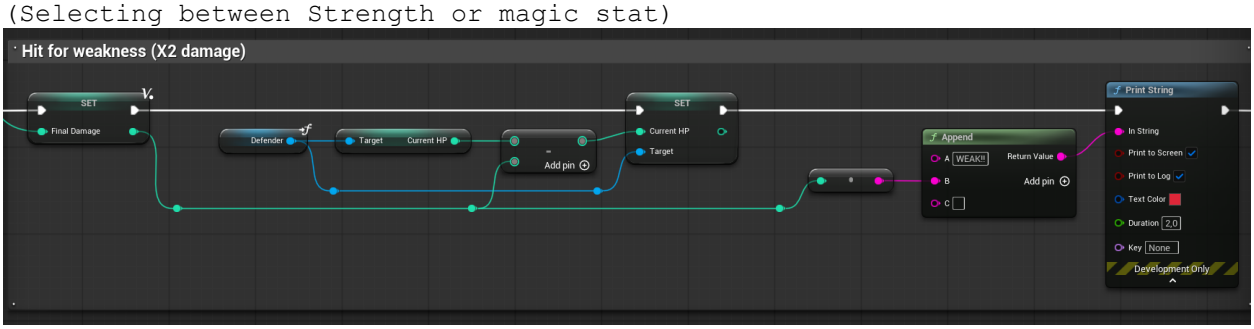
# Damage Formula

The current damage calculation is:

```
Damage = ((√ Skill Power × √Stat)
Defense =  Damage(÷)√(Endurance * 8)
```




(Selecting between Strength or magic stat)




(Hit for weakness)

You can find and modify this in the **ExecuteSkill** function within BP_CombatManager.

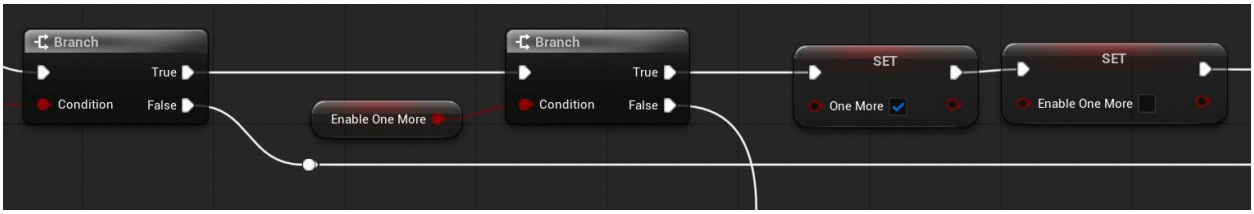
TL;DR to influence the damage at the end, change the parameters to FinalDamage, that's the damage that gets subtracted from current defenders hp (after defender calculation)

# Weakness/Resistance Multipliers

Default values:

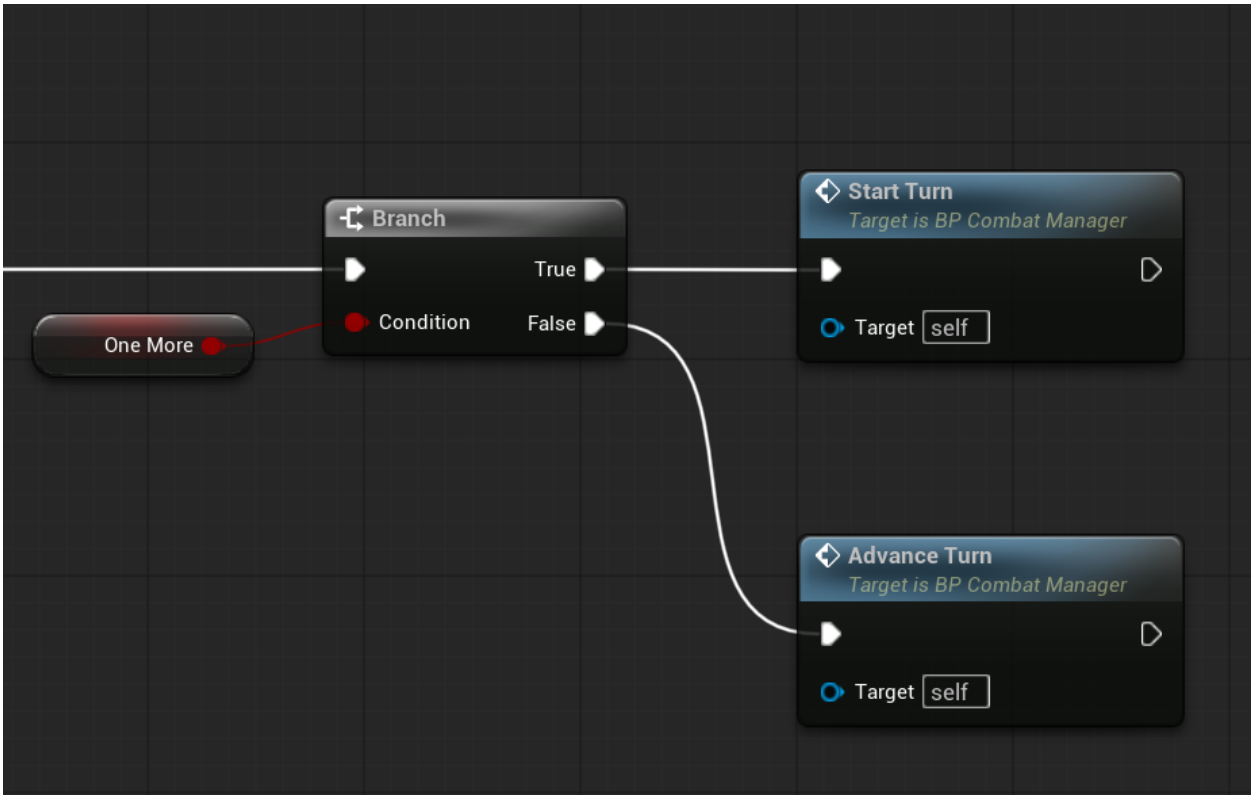
- **Weakness:** 2.0× damage + triggers One More
- **Guard:** 0.4x damage less damage

# “One more!”




When Weakness is hit and defender is not guarding -> Set OneMore to true

OneMore follows the same damage calculation path as 2x weakness but adds a branch check at the end of executeskill, determining whether turn advances (enemy turn) or turn starts (letting the player go again)




bEnableOneMore is in place to make sure either combatant can only pull off 1 one more per round

# UI Customization

## CommonUI Widget Structure

The combat UI uses CommonUI's activatable widget stack system:

- **WBP_CombatHUD:** Root widget, always visible during combat
- **WBP_CombatChoice:** Main action menu (Skill, Guard)
- **WBP_SkillSelection:** Skill list container with 8 buttons that populates buttons with equippedCharacters skills
- **WBP_SelectionButton:** Passes on what skill is chosen to combatmanager

### CommonUI Styles

The neat thing about CommonUI is that you have something called Styles, you nigh have seen it mentioned before but in commonUI there is something called Style_EmptyColor. If you apply styles to buttons you can pre-determine their behaviour, from size to color to even changing behavior when hovered or pressed

To create a new style simply

### Customizing Combat HUD

The main HUD displays:

- Player/enemy health bars
- Current turn indicator
- Action feedback text

Modify **WBP_CombatHUD** to add or rearrange elements like:

- Party member portraits
- Enemy weakness indicators (after scanning/analysis)
- Turn order display
- Combo counter

---

# Troubleshooting

## Common Issues

#### Combat doesn't start when entering trigger

- Verify BP_CombatManager exists in the level
- Check that the BP_OevrWorldEncounter's enemy list contains valid BP_EnemyShadow
- Ensure player implements BPI_Combatant interface

**Skills don't appear in menu**

- Verify the character's skill array contains valid row names from DT_Skills
- Double check if these skills are also present in E_SkillID
- Check that AC_CombatantComponent is properly initialized

**Damage seems wrong**

- Review the damage formula in ExecuteTurn
- Verify stat values in character data tables

**UI doesn't respond to input**

- Ensure CommonUI is properly configured in project settings
- Check input mode is set correctly when combat starts
- Verify widget activation states

---

# Version History

| Version | Date | Changes |
|---|---|---|
| 1.0 | [15-01] | Initial release |
| | | Changed step order in DT_Skills Step 4 -6; Is now Step 5 more detailed |
| 1.1 | [16-01] | Changed image in Step 5 for DT_Skills to be more clear comparing E_SkillID with DT_Skills |

Source*

(*Draft made by claude and revised/edited by me completely: Original chat link)