# Understanding UE4 Gameplay Ability System

Some useful links to understand GAS

<https://docs.unrealengine.com/en-US/Gameplay/GameplayAbilitySystem/index.html>

<https://github.com/RandyKoiSA/GASDocumentation#concepts-ga-spec>

<https://docs.unrealengine.com/en-US/Resources/SampleGames/ARPG/index.html>

## Definitions

**Attributes:** Values like Health and Damage and are required to use the system.

**Gameplay Effects:** How Attributes are modified and are required to use the system.

**Ability System Component:** Is the component used to actually apply effects and abilities. All Actors that have this component. There is an ActionRPG subclass that handles some game-specific logic.

**Execution Calculation:** Used by Gameplay Effects to apply math buffs and debuffs when changing attributes. ARPG uses Execution Calculations for damage formula.

**Abilities:** Special Blueprints that are executed to make gameplay changes and are useful in any game that wants custom logic for specific abilities. A highly data-driven game may choose to use something other than Blueprints to execute effects but ability Blueprints work well for an action game like ARPG.

**Gameplay Events**: Gameplay tags with optional payload data that are used to communicate between different parts of the ability system. ARPG uses them to pass information between montages and abilities.

**Gameplay Cues**: Assets that are used to tie a Gameplay Tag to a spawn particle or sound. These are useful for handling client prediction or creating visual that scale in intensity.

## Gameplay Abilities

## Gameplay Attributes

## Gameplay Effects

|  |  |
| --- | --- |
| Function Name | Purpose |
| PreAttributeChange / PreAttributeBaseChange | These functions are called just before modifications to an Attribute. They are intended to enforce rules on the Attribute’s value, such as “Health must be between 0 and MaxHealth”, and should not trigger in-game reactions to Attribute changes. |
| PreGamplayEffectExecute | Just before modifying an Attribute’s value, this function can reject or alter the proposed modification. |
| PostGameplayEffectExecute | Immediately after modifying an Attribute’s value, this function can react to the change. This often includes clamping the final value of the Attribute, or triggering an in-game reaction to the new value, like dying when the “health” attribute falls to zero. |