# Creamos una habilidad

Ejemplos

* Incrementar daño que hago cuando mi vida esta baja por ejemplo
* O le doy a un botón y hago una animación

Graphical user interface, application

Description automatically generated Graphical user interface, text, application

Description automatically generated

## Activar habilidad.

Desde ClassDefaults estan los

### Triggers

Para lanzar las habilidades

A screenshot of a computer

Description automatically generated with medium confidence

Podemos por ejemplo registrarnos a eventos de la ultima clase

* GAMEPLAY EVENT (OnDie) : cuando nos llega el evento, se lanza la habilidad
* OWNED TAG ADDED: o cuando se añade un Tag
* owned tag presed (mientras estoy envenenado por ejemplo se activa la habilidad durante un tiempo)

### También se pueden activar manualmente

Graphical user interface, text, application

Description automatically generated

**Deben asegurarnos de que la habilidad está en esa clase. Con los triggers no tenemos que preocuparnos de eso. Es mas recomendable triggers**.

Enhanced Inputs lanzaran GameplayEvents. Lo bueno de hacerlo así es que si quiero lo mismo para la IA no habrá problema.

Al lanzarse una habilidad se lanza el evento

Graphical user interface

Description automatically generated

## Tasks

Vamos a hacer una habilidad que lance una animación y cuando acabe, que acabe tamb la habilidad. Hay muchas si ponemos Ability Task te sale un desplegable.

Graphical user interface, text, application

Description automatically generated

Desde el AS (animator sequence) le da click derecho y crea un Montage

Creamos un GameplayTag LightAttack

A screenshot of a computer

Description automatically generated with medium confidenceGraphical user interface, application

Description automatically generated

Y cuando se presione el click izquierdo, lanzamos el Gameplay Event

A screenshot of a computer

Description automatically generated with low confidence

En el Character (ability system component) añade la habilidad

A picture containing graphical user interface

Description automatically generated

A las 18:18 CREO QUE TOCA PARA HACER UN DEBUG

**Retrigger Instanced Ability**: si le doy muchas veces al left button va a empezar de nuevo el montage. Permite relanzar la habilidad

Graphical user interface, application

Description automatically generated

Podría poner un array de montages, y hacer un combo

## Instancing Policy

* Non Instanced: no tiene estado, solo lógica. Ejecuto la lógica, se acaba y ya está. No
* Instanced Per Actor: guardo el estado que perdura. Se guarda el estado cuando se crea el actor
* Instanced Per Execution: crea una instancia cada vez que se activa la habilidad. Cuando se termina la habilidad, eso se destruye

Graphical user interface, text, application

Description automatically generated

Creamos otra habilidad (HeavyAttack). Ojo hay que ponérsela al player

A screenshot of a video game

Description automatically generated

## Interrumpir habilidades

Se pueden interrumpir habilidades. Si hago un heavy attack y luego un light attack que pasa? Como lo gestionamos?

### AbilityTags

**Tags que usa la habilidad**.

Podemos blockear habilidades por prioridad:

Le decimos en abilityTag que ocupara todo el cuerpo (AbilityFullbody)

### Block abilities with tag:

full body (bloquea a las que necesiten todo el cuerpo). De manera que si estoy usando la habilidad (lleva FullBody) y me va a bloquear (no se lanzarán) las habilidades con Fullbody.

Graphical user interface

Description automatically generated

Otros ejemplos: si tengo el menú abierto ya no debería poder atacar por ejemplo.

### Activation Blocked Tags:

No puedes activar la habilidad, si tengo el Statuseffect.Negative.DisableAttack

Graphical user interface, application, Teams

Description automatically generated

### Activation Required Tags

Solo puedo usar esta habilidad si tengo este Tag

### Activation Owned Tags:

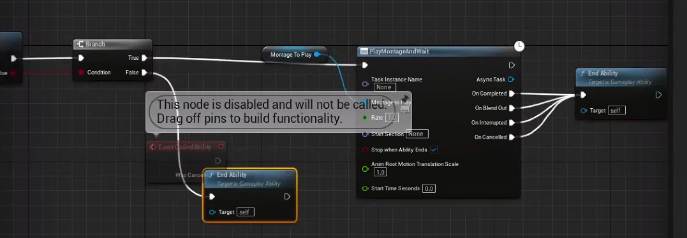
Tags que van al personaje cuando la habilidad se activa

## Cooldown

Crea un GE\_HeavyAttackCooldown y añade en el GA\_heavyAttack el cooldown en los details.

Este GE, nos da un Tag, tiene duration de 5 segs (heavyattack tag)

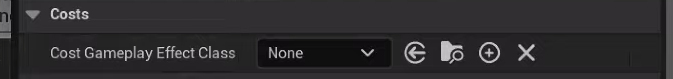
Las habilidades vienen con **cooldown**, para eso se hace un **commitability**.

Graphical user interface, application

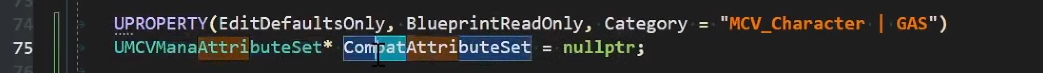
Description automatically generated

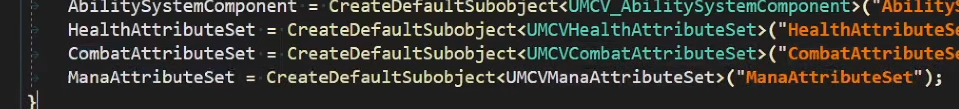
Si puede hacer el commit pues lanza la animación, sino no. Entonces ya no podemos hasta que pasen 5 segundos.

Tamb comprueba el **cost** (no solo el cooldown). Y si le metemos maná por ejemplo lo checkea.



Crea AttributeSet Mana y se lo añade al Player.





Y el include



* AbilitySystem.DebugAttribute Mana
* Showdebug abilities
* Showdebug abilitysystem (podemos ver los atributos, del player por defecto)
* AbilitySystem.Debug.NextCategory() 🡪 Vemos los efectos aplicados y si le damos otra vez las abilidades
* Slowmo 0.1

Graphical user interface

Description automatically generated with medium confidence

* Otros comandos para el log º

Crea un GE\_HeavyAttackCost y en el GA\_HeavyAttack le añadimos el GE de cost

<https://github.com/tranek/GASDocumentation>