

Documento de Diseño

Componente: Inventario y Habilidades

Universidad Complutense de Madrid

Máster Ingeniería Informática

Desarrollo de Videojuegos

Erik Velásquez

Introducción	4
Componente Inventario	5
Estructuras	5
Item Structure:	5
Slot Structure:	6
Componentes	7
Items	8
Item Árbol:	9
Item Roca:	9
Contenedores	10
Interfaz Gráfica	11
Interfaz gráfica general del Inventario:	11
Interfaz gráfica del Slot:	11
Mecánicas	12
Visualizar inventario:	12
Interactuar con Ítems:	12
Usar un Ítem:	13
Querys con ítems:	13
Componente Habilidades	14
Estructuras	15
Estructura Habilidad:	15
Interfaz Gráfica	16
Barra de Mana y vida:	16
Barra de habilidades:	16
Habilidad general	16
Algunas habilidades	17
Bola de fuego:	17
Circulo Sagrado:	18
Curse:	19
Personaje Principal	19
Atributos:	19
Funciones:	20
Mecánicas	21
Definir y ejecutar una acción:	21
Realizar las habilidades mediante un Click:	21
Documentación y Tutoriales	22
Inventario: Implementación	22
Begin Play:	22

Agregar componente Inventario:	23
Call Inventario:	23
Habilidades: Implementación	24
Referencia al HUD en begin Play:	24
Definir atributos a Actor principal:	24
Definir la ejecución de la habilidad:	25
Tutoriales	26
Videos del componente de Inventario:	26
Videos del componente de habilidades:	26

Introducción

Uno de los caminos más claros para convertirse en un diseñador de juegos sólido es aprender de los triunfos y fracasos de otros. Dicho esto, antes de diseñar su propio sistema de inventario para ello es imperativo que al menos se familiarice con las prácticas más aceptadas.

Ahora, eso no significa que esté encasillado en el uso de uno de los esquemas más utilizados, todo lo contrario. En todo caso, debería estudiarlos, conocer sus fortalezas y debilidades, y quizás usarlas como una plantilla para su propio diseño innovador.

Uno de los primeros sistemas de inventario real implementado ampliamente por los desarrolladores, los inventarios basados en la "Regla de 99" se asocian más comúnmente con JRPG clásicos como Final Fantasy VI, Chrono Trigger y las primeras entradas en la serie Pokémon. El paradigma de diseño detrás de este sistema es extremadamente simple: los jugadores pueden esconder una cantidad innumerable de artículos en su inventario, pero solo un número fijo de cada uno. Y mientras que ese número fijo suele ser 99, puede ser fácilmente igual a 50, 100 o 1,000. Además, los inventarios de la "Regla de 99" son generalmente universales, lo que significa que se comparten entre todos los miembros del partido.

En los juegos de rol las habilidades son valores estadísticos que los directores de juego y los jugadores utilizan para resolver las acciones emprendidas por sus personajes, ya sean personajes jugadores o no jugadores, o incluso criaturas, animales u otros seres vivos susceptibles de ver sus acciones representadas en la ficción del juego. En la inmensa mayor parte de juegos de rol las habilidades de un personaje jugador están representadas por valores numéricos anotados en una hoja fotocopiable o imprimible, la hoja de personaje.

A diferencia de las características, las habilidades no representan los valores esenciales definitorios de un personaje. Mientras que las características son reducidas en número, prácticamente invariables y representan los aspectos más determinantes del personaje, las habilidades son numerosas, varían con la experiencia y no representan lo que el personaje es sino lo que sabe o es capaz de hacer. Las habilidades cumplen por tanto una función esencial en los juegos de rol: son un modo artificial de representar aquellas acciones en las que los personajes de los jugadores corren el riesgo de fallar, lo que en muchas ocasiones acarrea consecuencias radicalmente diferentes en el curso de una partida según uno o varios personajes hayan obtenido éxito o hayan fallado en la puesta en práctica de algunas de sus habilidades, y a veces en una sola.

El inventario y las habilidades con características fundamentales en casi todos los juegos modernos, por lo que su implementación es crucial. Este proyecto trata del desarrollo de ambos componentes buscando la reusabilidad de los mismos y un complemento educativo para los que lo utilicen.

Componente Inventario

El componente inventario permite guardar ítems generales en el juego, recogerlos, usarlos y otras opciones necesarias durante la jugabilidad deseada.



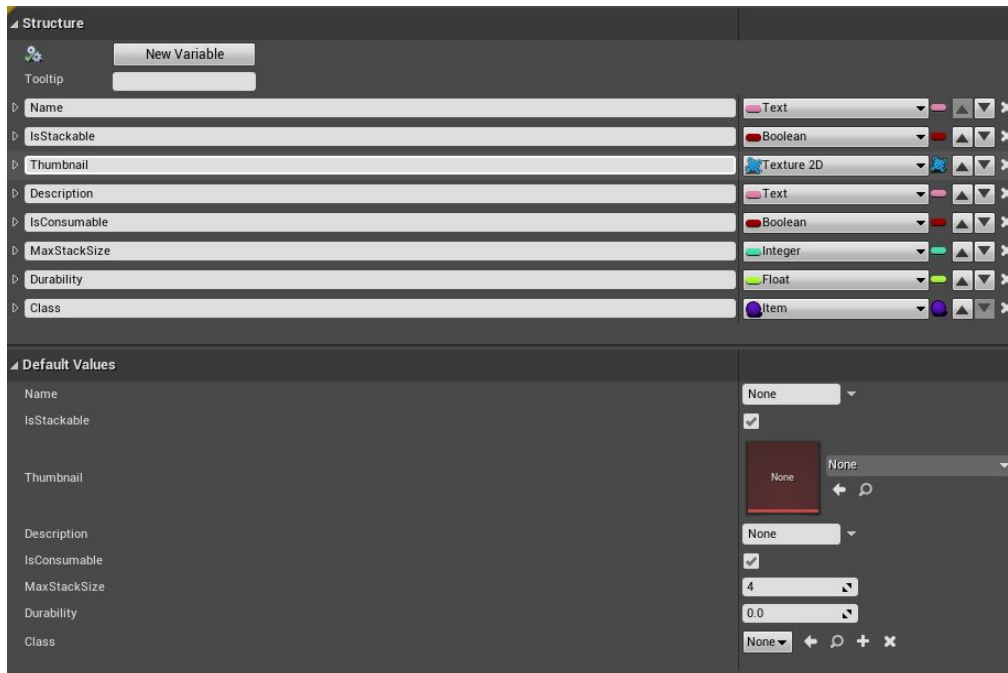
Visión general del Inventario en Unreal Engine.

Estructuras

Item Structure:

Es la estructura general para almacenar la información de los posibles ítem a ser guardados en un inventario, estos atributos son los siguientes:

- Name: Nombre que poseerá el ítem.
- IsStackable: Verificar si el ítem se puede o no guardar en inventario.
- Thumbnail: Icono.
- Description: Descripción general que poseerá el ítem.
- IsConsumable: Si el ítem puede ser consumido o no.
- MaxStackSize: Máximo tamaño permitido para acumular el mismo ítem.
- Durability: La durabilidad general del ítem.
- Class: Tipo de objeto que será.

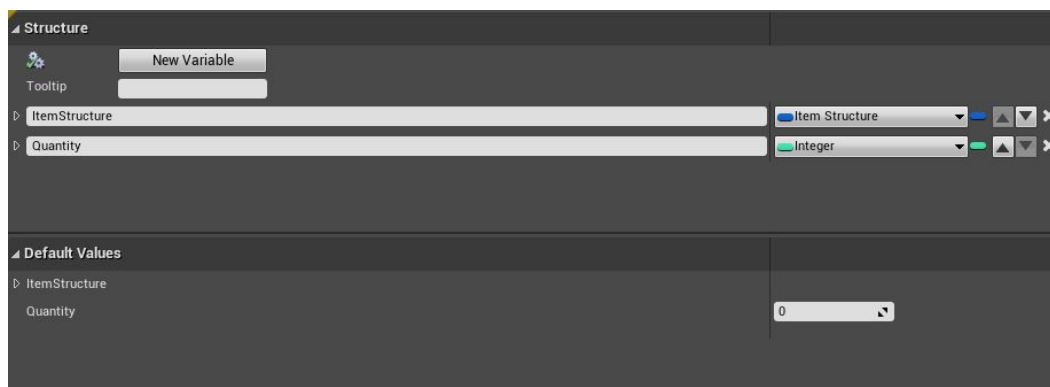


Estructura del Ítem en Unreal Engine.

Slot Structure:

Es la estructura del inventario que almacenará el ítem. Un inventario está formado por un conjunto de slots, los cuales a su vez almacenan a los ítems. La información general que posee es la siguiente:

- ItemStructure: la estructura general del ítem, con toda su información.
- Quantity: La cantidad almacenada de ese ítem en el slot.



Estructura del Slot en Unreal Engine.

Estas son las dos estructuras principales de datos y con ellas podemos almacenar la información general de una gran variedad de ítems, para ser almacenados en el inventario.

Componentes

El componente inventario es la parte más importante, es la que permite la asociación de inventarios a los distintos personajes y elementos en el juego para poder almacenar los ítems. El componente “Inventory Component” es lo que podemos llamar exactamente el inventario, y permite asociarlo a los actores que definamos para su uso.

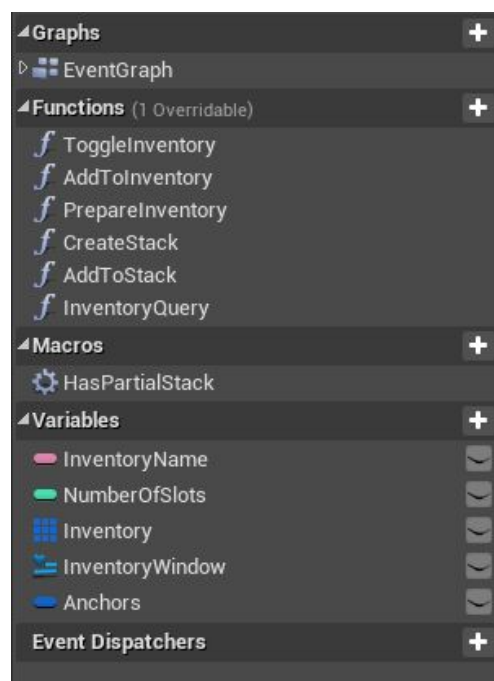
Las variables para formarlo son las siguientes:

- InventoryName: Nombre que queremos darle al inventario
- NumberOfSlots: es el número de casillas que queremos que poseen el inventario.
- Inventory: Es el arreglo de Slots, el cual contiene todos los ítems de nuestro inventario.
- InventoryWindows: La ventana asociada para la visualización del inventario.
- Anchors: La posición en la que se quiere mostrar el inventario.

De igual manera cuenta con unas funciones principales para definir su comportamiento, como lo son:

- Toggle: realiza la visualización del inventario.
- Add: agrega un elemento al inventario.
- Prepare: Inicializa el inventario antes de comenzar el juego.
- Query: permite verificar la existencia de elementos en el inventario, muy útil cuando ciertos ítems son necesarios para ejecutar una acción en el juego.

Hay algunos otros métodos que se pueden verificar y observar en el código del componente.

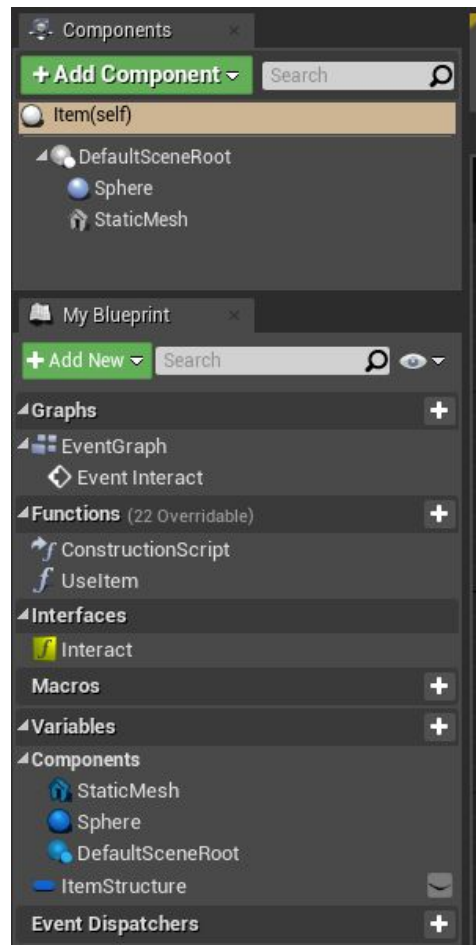


Información del componente.

Items

El ítem como su nombre lo indica es el elemento que manejaremos y almacenaremos en nuestro inventario.

Este, está formado por un actor que posee el ItemStructure como su variable principal. De igual forma posee un método de “Interact” que define la forma de interacción de cada ítem con su ambiente.



Información del Ítem.

Una de las características principales del Ítem es que posee una interfaz la cual permite interactuar con distintos actores en el juego (incluyendo el componente inventario), lo que hace muy sencillo su uso.

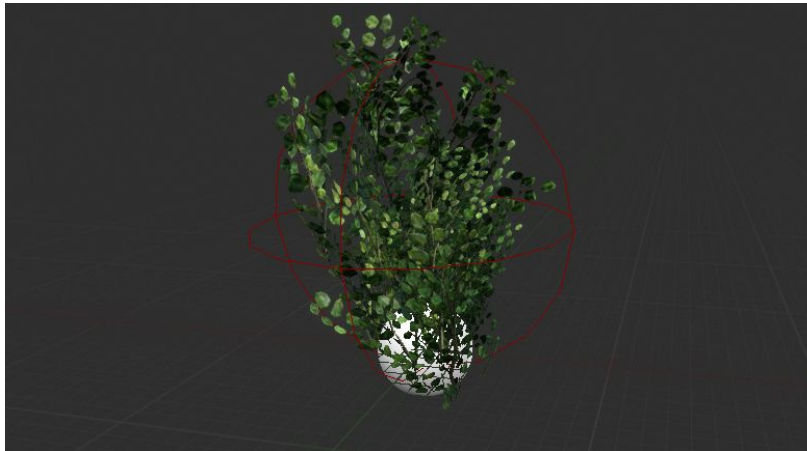
Funciones principales que se deben definir para cada ítem dependiendo de la funcionalidad deseada:

- ConstructionScript: Define cómo se realiza la construcción y almacenamiento del ítem.
- UseItem: define cuál será el uso que el ítem tendrá como actor en nuestro juego. Este es el método más importante para el ítem.

Al ser el actor principal y la clase padre, esto nos permite crear nuevos ítems que heredan las propiedades, como lo pueden ser:

Ítem Árbol:

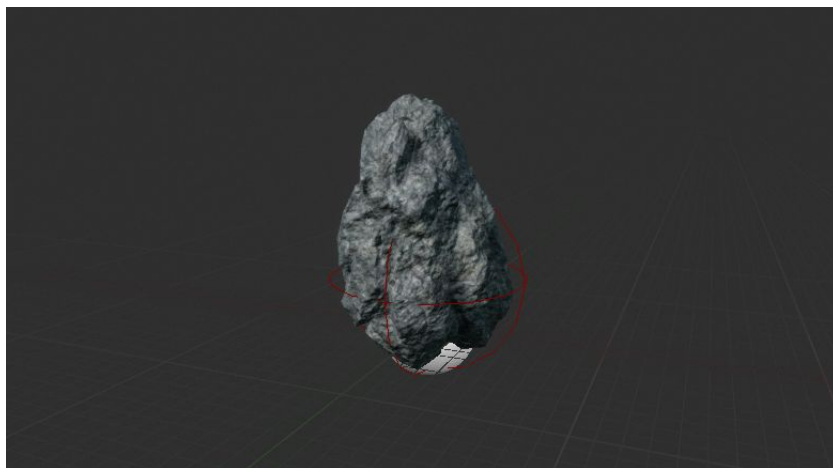
Es un actor definido a partir de la clase ítem y tiene propiedades definidas de uso. A este ítem podemos definir su método de uso y de esa forma interactuar de forma amigable o peligrosa con nuestros personajes en el juego.



Ejemplo de ítem Árbol.

Ítem Roca:

Es un actor definido a partir de la clase ítem y tiene propiedades definidas de uso. En este ítem podemos definir su método de uso para realizar acciones en el juego.

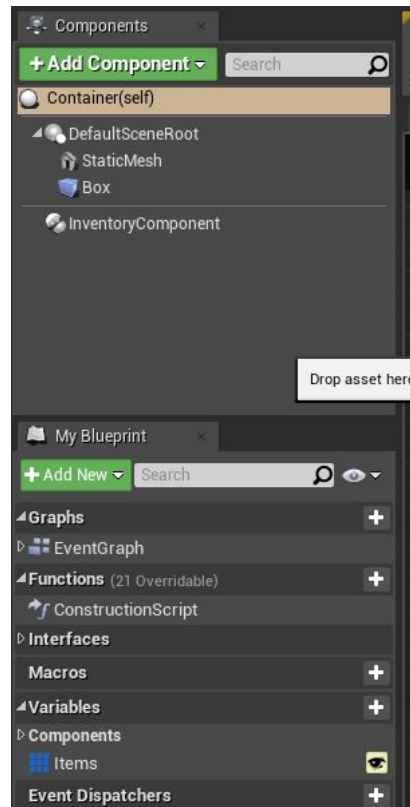


Ejemplo de Ítem Roca.

Contenedores

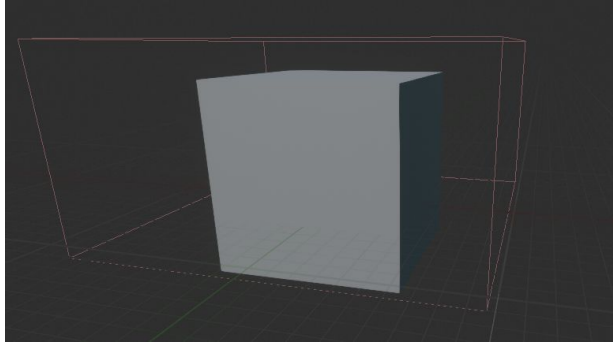
Un contenedor nos permite poseer un inventario, podemos verlo como un cofre de tesoro, el mismo personaje principal, un enemigo, un mercader, o cualquier actor en nuestro juego que deseamos que posea un inventario, ya sea para interactuar con él o dejar y recoger items.

La característica principal que debe poseer es tener asociado el componente de inventario, con esto maneja toda la funcionalidad asociada.



Información de un contenedor.

Aquí podemos observar una especie de cofre del tesoro que es manejado como un inventario, simplemente asociando el componente.

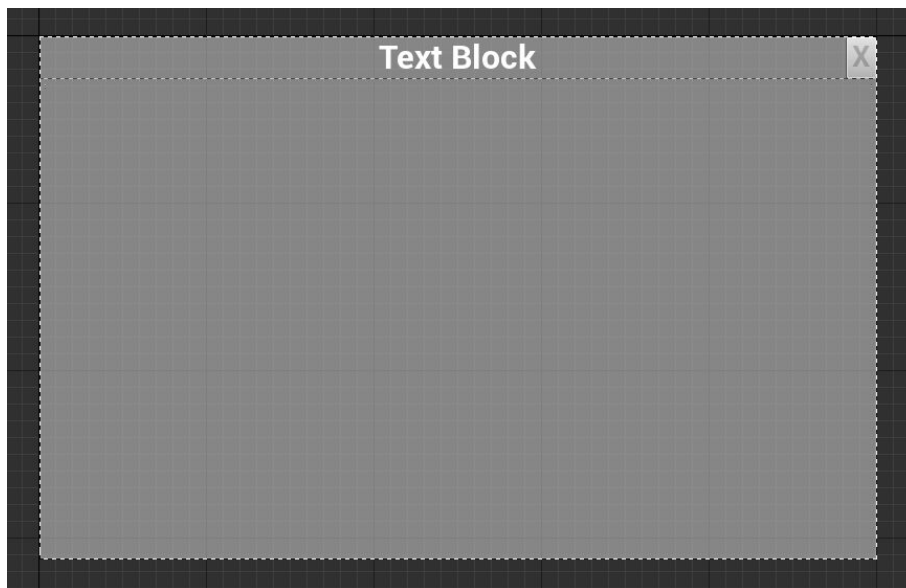


Vista de un ejemplo de cofre que permite almacenar inventario.

Interfaz Gráfica

Interfaz gráfica general del Inventario:

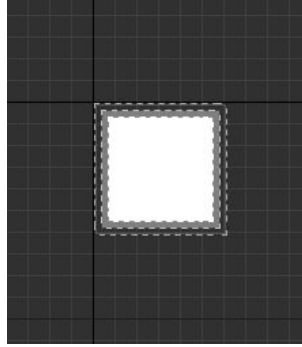
Es la interfaz principal que se mostrará de un inventario. esta interfaz puede ser modificada según las necesidades del juego.



Interfaz gráfica general del Inventario.

Interfaz gráfica del Slot:

Es el casillero que almacenará al ítem como tal. Se puede modificar según las necesidades del juego

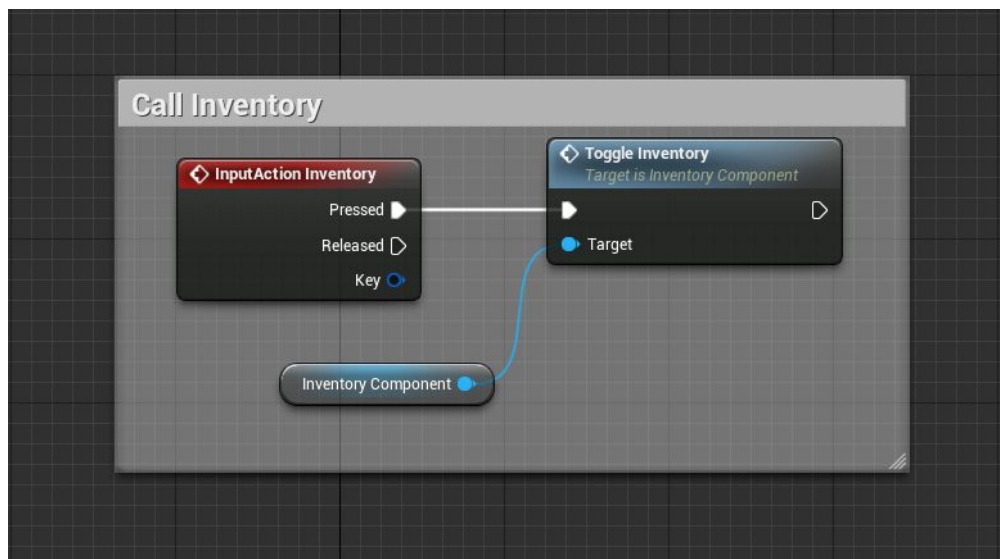


Interfaz gráfica del Slot del inventario.

Mecánicas

Visualizar inventario:

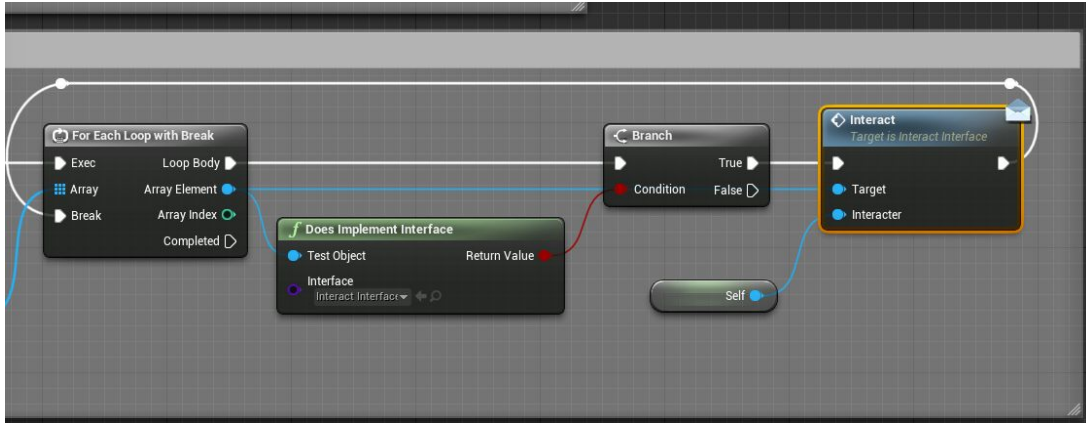
Es la acción que nos permite observar el inventario, puede llamarse como una Action Mapping en nuestro juego o sea mejor para la jugabilidad.



Llamar inventario para ser mostrado.

Interactuar con Ítems:

nos permite interactuar con los ítems o componentes de inventario repartidos y encontrados durante el juego.



Interactuar con los diferentes ítems del mundo.

Usar un Ítem:

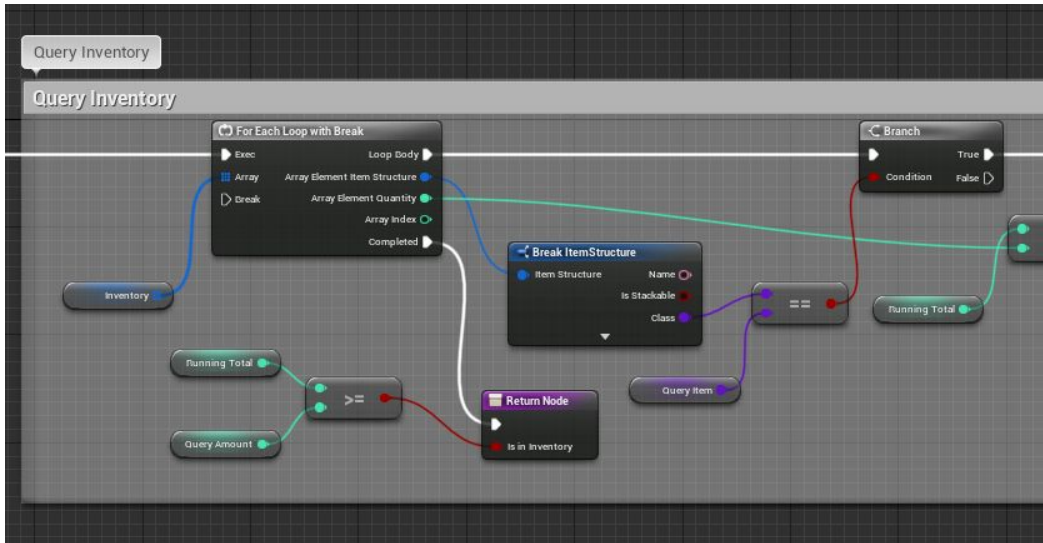
Permite usar, o ejecutar la acción de uso que el ítem posea. Aquí podemos definir cualquier acción: Subir vida, mana, comprar, cambiar, entre otras.



Usar un ítem en particular.

Querys con ítems:

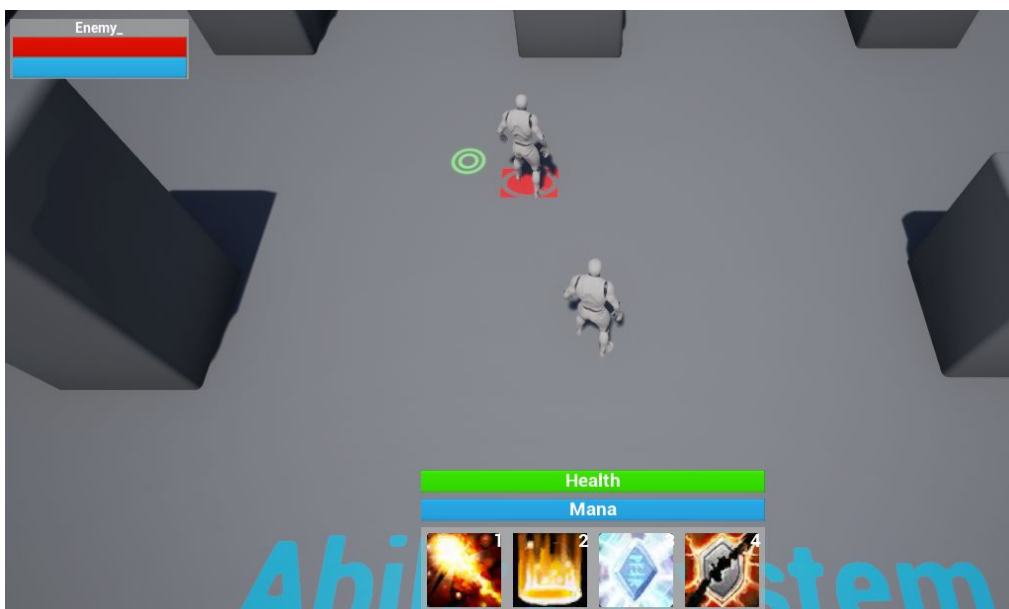
Es una acción compleja, pero nos permite definir acciones necesarias o interacción con objetos dentro de nuestro juego, ejemplo: Abrir una puerta con una llave específica, combinar elementos, vender, entre otras cosas.



Ejecución de queries o ítems necesarios.

Componente Habilidades

Este componente nos permite poseer y usar habilidades en un juego, las habilidades pueden ser tipo mágicas, que consumen acciones, que efectúen daños o cualquier otras opciones que necesitemos.



Vista general del Componente de Habilidades.

Estructuras

Estructura Habilidad:

La estructura de habilidad almacena la información de una habilidad, su coste, entre otras cosas. Es la estructura principal de información. Los atributos generales que posee son los siguientes:

- Name: Nombre de la habilidad.
- Description: descripción general de la habilidad.
- Cost: el costo que necesitará la habilidad, puede ser, coste en maná, dinero, u algún otra característica que deseamos definir.
- Cooldown: tiempo de regeneración de la habilidad.
- CastTime: tiempo de carga para ejecutar la habilidad.
- IsInterruptable: permite saber si podemos o no interrumpir la habilidad cuando está siendo ejecutada.
- RequiresTarget: define si necesita o no un objetivo para ser ejecutada.
- Icon: el icono que define a la habilidad.
- AbilityKey: define si está definida a una tecla o combinación de teclas.

The image shows a software interface for defining an ability structure. It is divided into two main sections: 'Structure' and 'Default Values'.

Structure Panel:

- At the top, there is a 'New Variable' button and a 'Tooltip' input field.
- Below, a list of attributes is shown, each with a name, a value type dropdown, and control icons (minus, up, down, and a close 'x').
- Attributes and their types: Name (Text), Description (Text), Cost (Float), Cooldown (Float), CastTime (Float), IsInterruptable (Boolean), RequiresTarget (Boolean), Icon (Texture 2D), and AbilityKey (Text).

Default Values Panel:

- This panel shows input fields for the default values of the attributes listed above.
- Fields for Name, Description, Cost, Cooldown, and CastTime are numeric inputs, all currently set to '0.0'.
- Fields for IsInterruptable and RequiresTarget are checkboxes, both currently unchecked.
- The Icon field has a 'None' button and a dropdown menu currently set to 'None'.
- The AbilityKey field is a text input.

Estructura general de la Habilidad.

Interfaz Gráfica

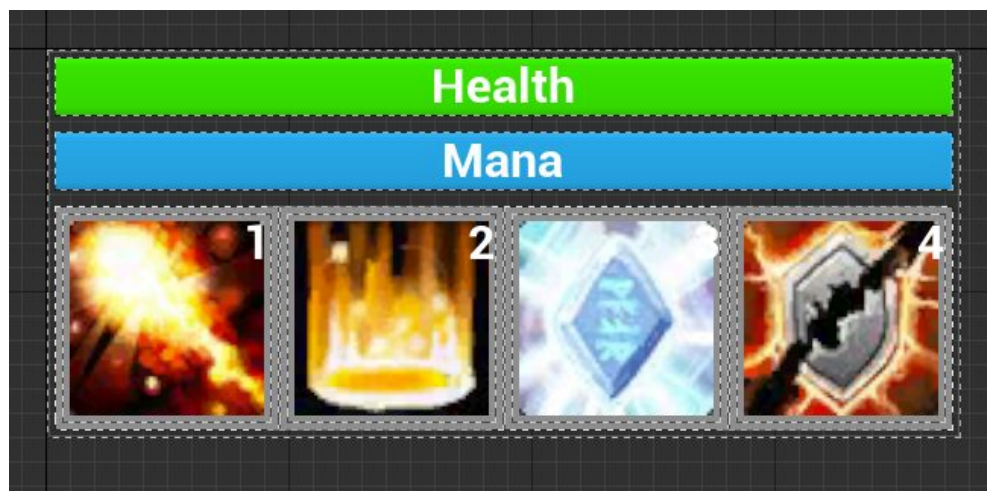
Está conformada por:

Barra de Mana y vida:

Contiene la vida y el maná para poder ejecutar las habilidades.

Barra de habilidades:

Contiene las habilidades y las acciones que podemos realizar con ellas.



Barra de habilidades.

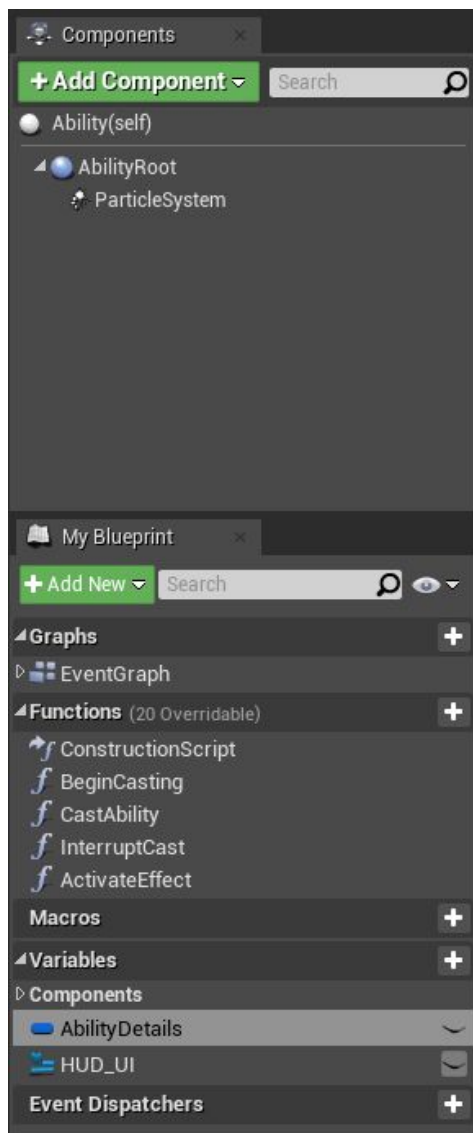
Habilidad general

Esta es la estructura general de un actor de tipo habilidad en el juego. Posee una serie de atributos importantes como son:

- AbilityDetails: estructura de los detalles específicos de la habilidad.
- HUD: La interfaz general asociada a la habilidad.

De igual manera están los métodos principales para su ejecución:

- BeginCasting: define las acciones necesarias cuando se comienza a ejecutar la habilidad, verificar que se cumplan todos los requerimientos mínimos para su ejecución.
- CastAbility: ejecuta la habilidad.
- InterruptCast: permite interrumpir la ejecución de la habilidad, siempre que esto sea permitido.
- ActivateEffect: Activa distintos efectos que sean necesarios para la habilidad.

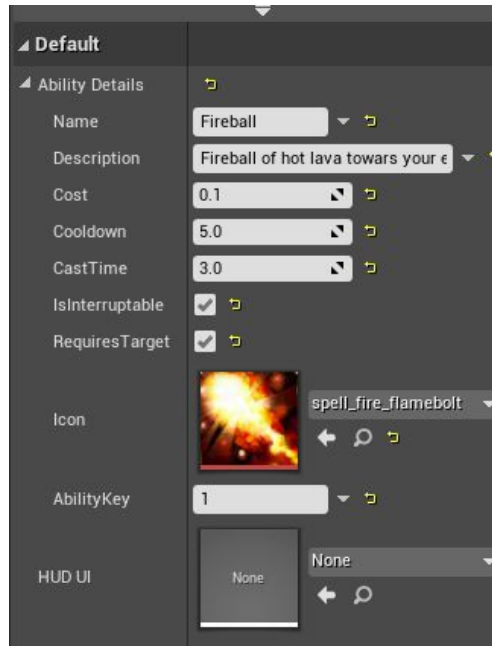


Actor de Habilidad.

Algunas habilidades

Bola de fuego:

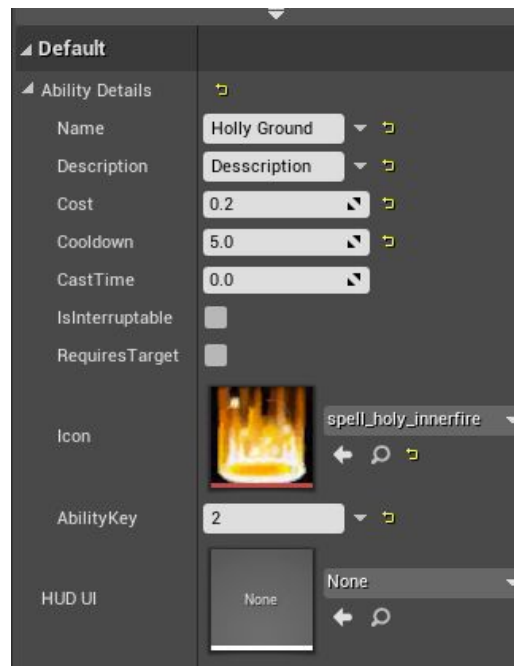
Es una bola de fuego que puede ser dirigida a nuestros enemigos. Esta habilidad permite seleccionar enemigos para ser atacados.



Habilidad de bola de fuego.

Circulo Sagrado:

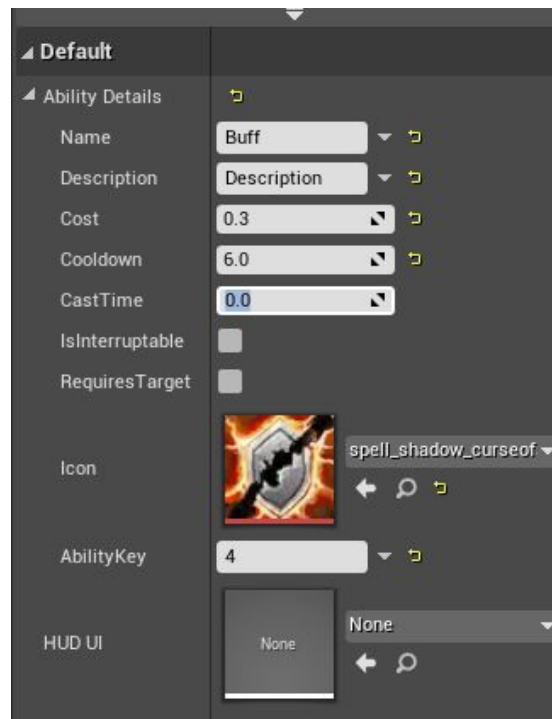
Genera un círculo alrededor del personaje y todos los enemigos que se encuentren en el círculo recibirán un porcentaje de daño.



Habilidad de círculo sagrado.

Curse:

Ejecuta una maldición sobre un enemigo.



Habilidad de Curse.

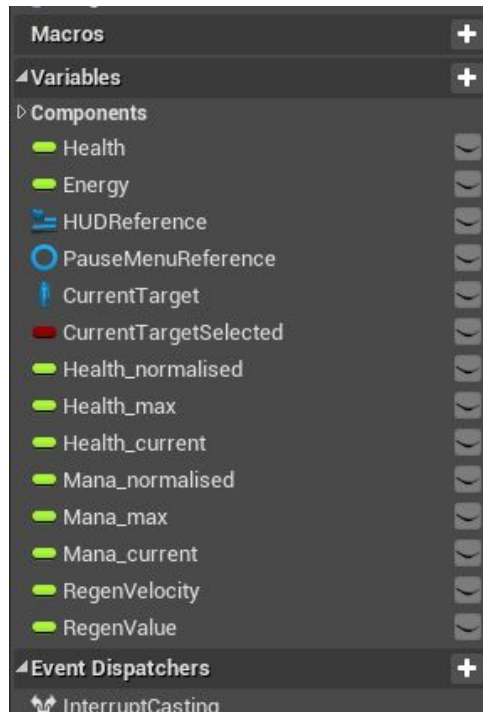
Personaje Principal

En este apartado definimos algunas de las acciones y atributos que nuestro personaje debe poseer para poder usar este componente de habilidades.

Atributos:

Los atributos principales a definir son los siguientes:

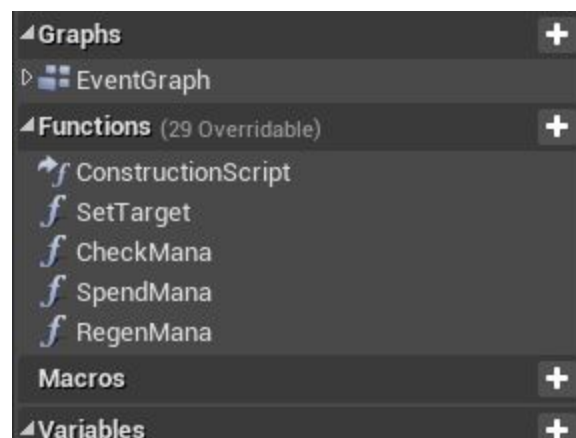
- Health: vida de nuestro personaje.
- Mana: cantidad de mana de nuestro personaje.
- CurrentTarget: los enemigos seleccionados.
- HUDReference: referencia al HUD principal del juego.



Atributos a definir.

Funciones:

- SetTarget: seleccionar un objetivo en el juego.
- CheckMana: Verificar la cantidad de maná que posee nuestro personaje.
- SpendMana: Disminuir la cantidad de maná que posee nuestro personaje.



Funciones a definir.

Mecánicas

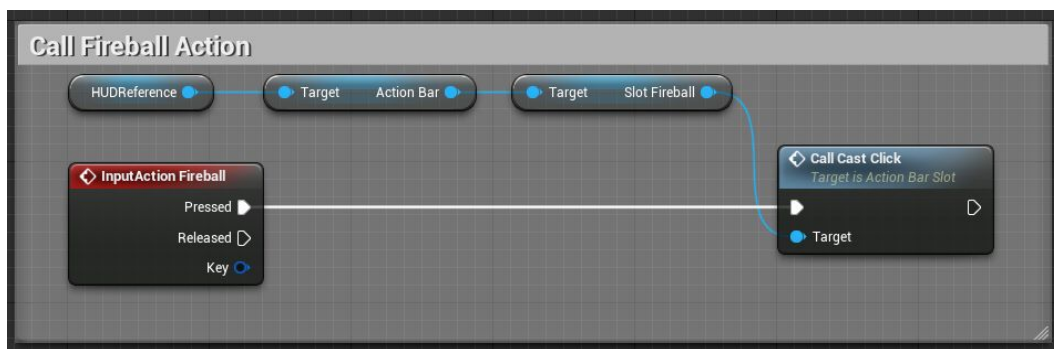
Las mecánicas que se ejecutan durante el manejo de habilidades son las siguientes:

Definir y ejecutar una acción:

Podemos definir acciones o combinaciones de teclas para ejecutar las habilidades, esto puede variar según las necesidades de nuestro juego.



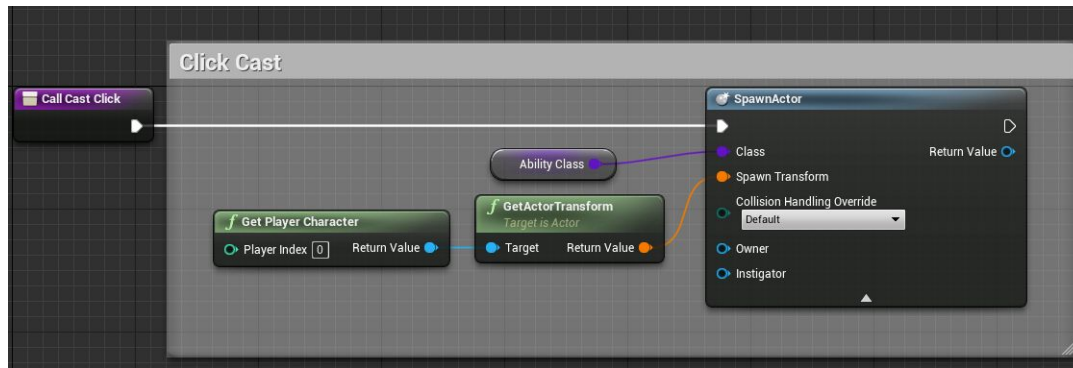
Definir acciones para habilidades.



Ejecutar acciones.

Realizar las habilidades mediante un Click:

También podemos definir ejecutar ciertas habilidades mediante pinchando el ratón, ya sea sobre nuestros enemigos o sobre un elemento de nuestro HUD.



Ejecutar habilidad pinchando el ratón.

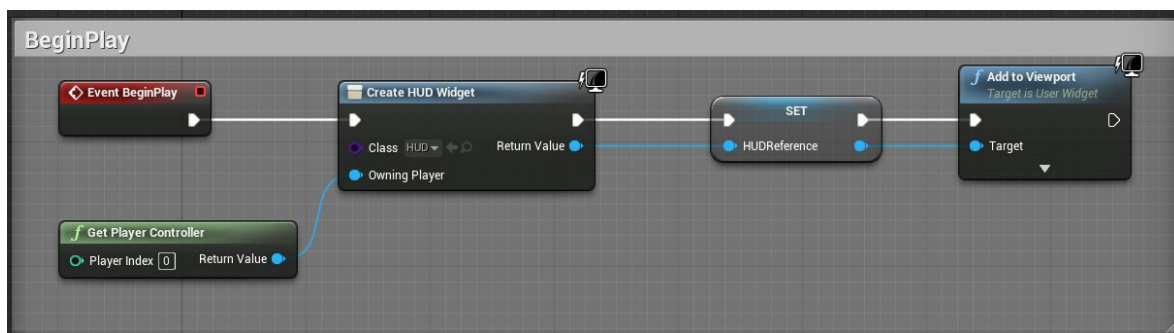
Documentación y Tutoriales

Para que el componente pueda ser funcional es necesario implementarlo en diferentes juegos. Para una mejor explicación y un tutorial detallado es recomendable ver los videos tutoriales que se desarrollaron. A continuación definimos algunos de los cambios principales para poder utilizar los componentes:

Inventario: Implementación

Begin Play:

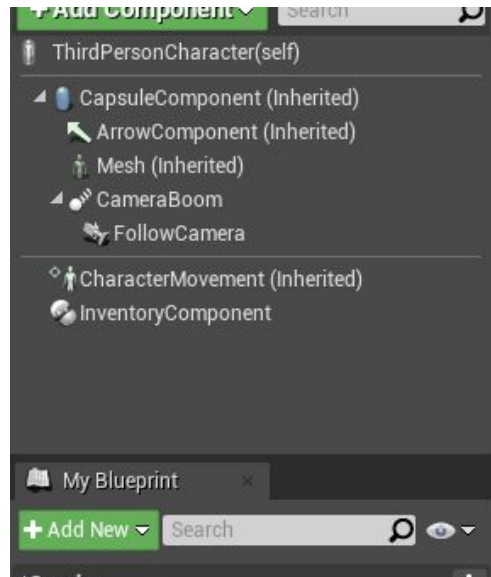
Lo primero que debemos hacer es asociar el HUD y las referencias de ese HUD para poder utilizar nuestro inventario.



HUD en Begin Play.

Agregar componente Inventario:

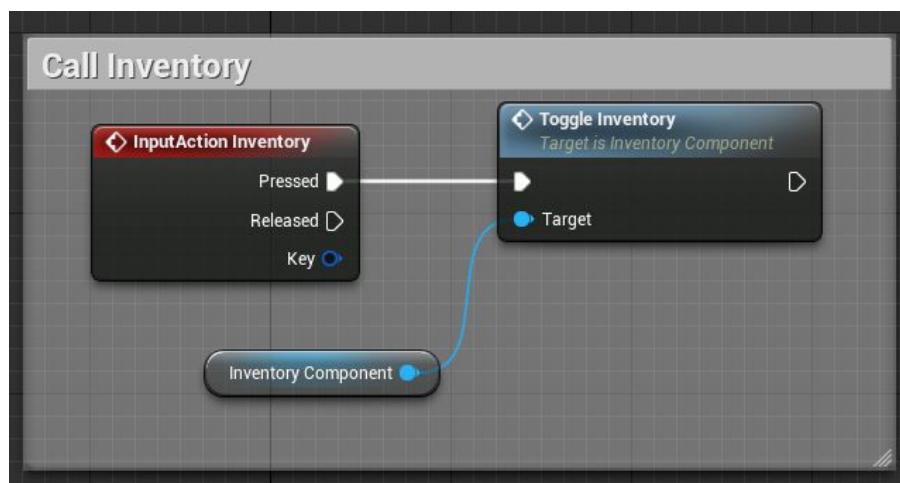
Tanto a nuestro personaje principal como a los actores que deseamos poseen el componente inventario, debemos asignar dicho componente.



Agregar componente a personales.

Call Inventario:

Debemos poder mostrar el inventario de nuestro personaje principal, por lo que debemos definir una función o método para usarlo.

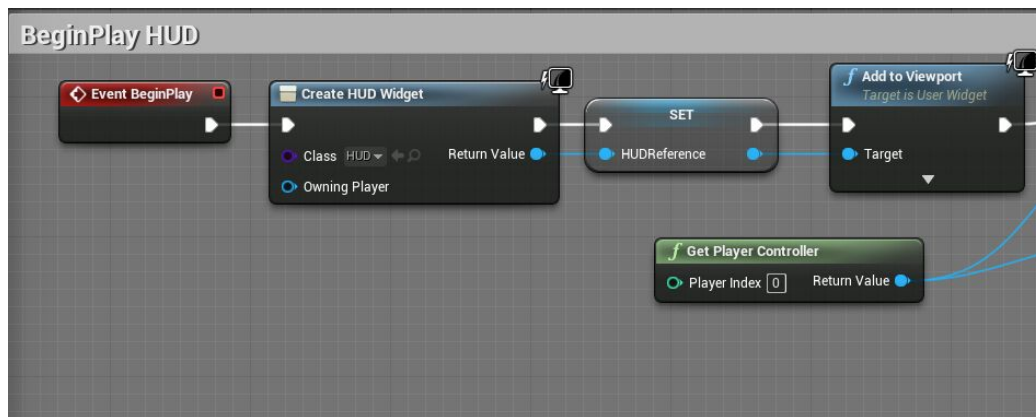


Definir el llamar a inventario.

Habilidades: Implementación

Referencia al HUD en begin Play:

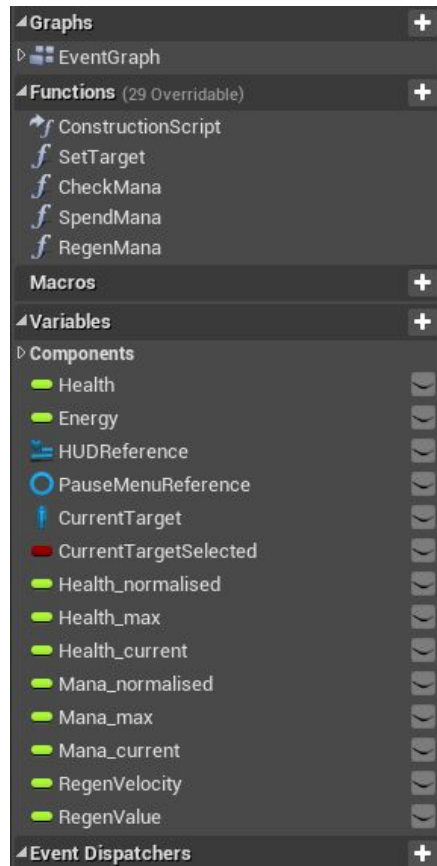
Es importante definir la referencia inicial al HUD cuando comenzamos el juego si queremos usar las habilidades.



Begin Play para habilidades.

Definir atributos a Actor principal:

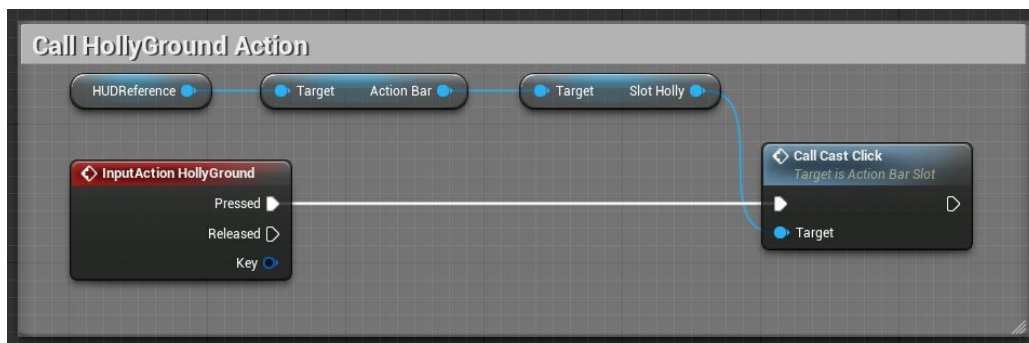
Debemos definir los atributos definidos anteriormente o los que se muestran en la imagen a nuestro personaje, para que use las habilidades sin problemas.



Definir variables a nuestro actor.

Definir la ejecución de la habilidad:

Luego de definida la habilidad o la reutilización de la misma, debemos definir cómo esta será llamada o usada en nuestro juego.



Definir la llamada de nuestras habilidades.

Tutoriales

Estas son las urls a los videos tutoriales para la definición y uso de los componentes. De igual manera esta son las urls a los repositorios de los componentes por si se quiere aportar ayudar:

Inventario:

<https://github.com/velasquezerik/master-dev-inventory>

Habilidades:

<https://github.com/velasquezerik/master-dev-ability>

Videos del componente de Inventario:

TO-DO

Videos del componente de habilidades:

TO-DO