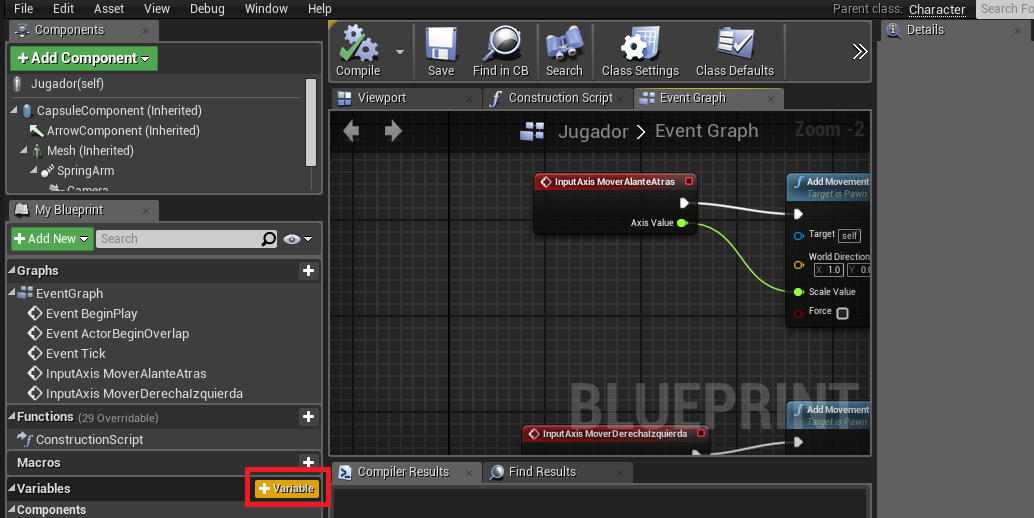
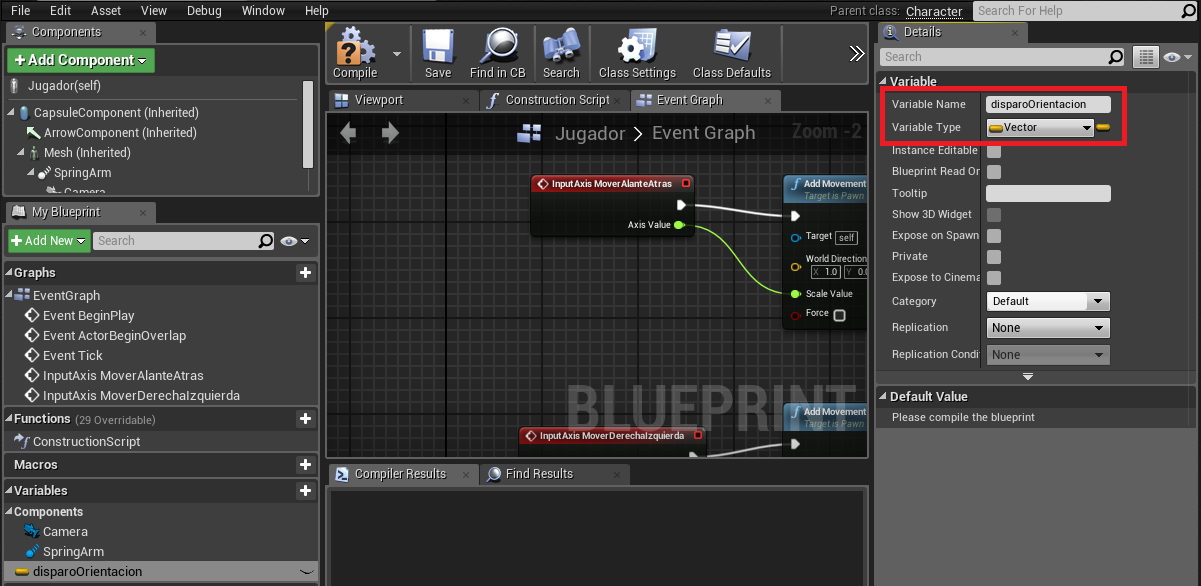
## Girar al jugador

El Jugador deberá orientarse en la dirección en la que disparamos, abrimos la clase **Jugador** y creamos une nueva variable



En la ventana de detalles de la izquierda editamos su Nombre (la llamamos **disparoOrientacion**) y la cambiamos a tipo **Vector**.



Esta variable de tipo **Vector**, tiene dentro 3 valores **X, Y y Z**, vamos a usar la X y la Y para guardar hacia donde nos pide el usuario que disparemos.

Pulsamos botón derecho sobre una superficie vacía del **Event Graph** y buscamos el evento **DispararAlanteAtras**



Continuamos el flujo de ejecución y hacemos una llamada a la función **Disparo Orientacion**

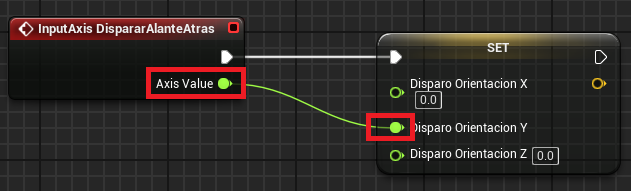




Pulsamos con el botón derecho sobre **Disparo Orientación** y lo divinos con **Split Struct Pin**

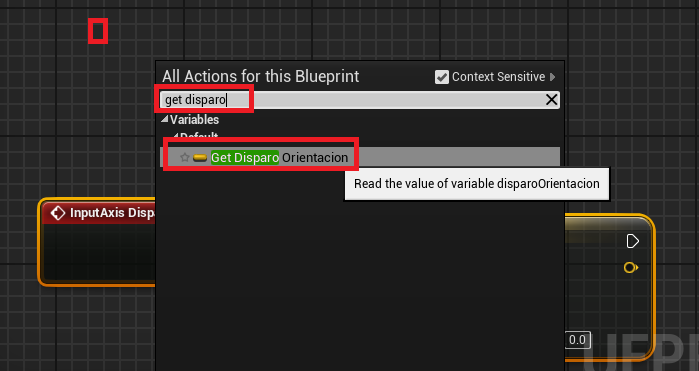


El **Axis value** se va a corresponder con la variable **Y**

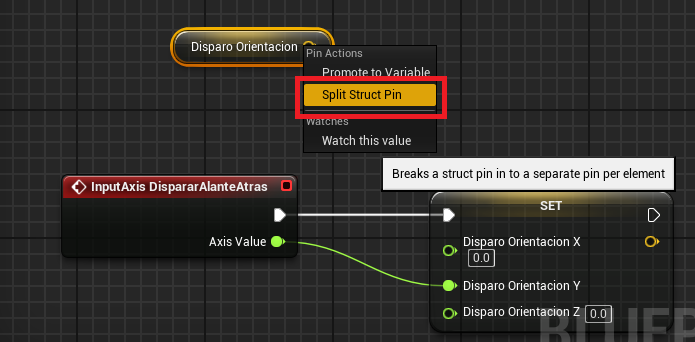


**Problema!.** Si lo dejamos así vamos a perder el componente X de la orientación X del disparo, se va a establecer a 0.

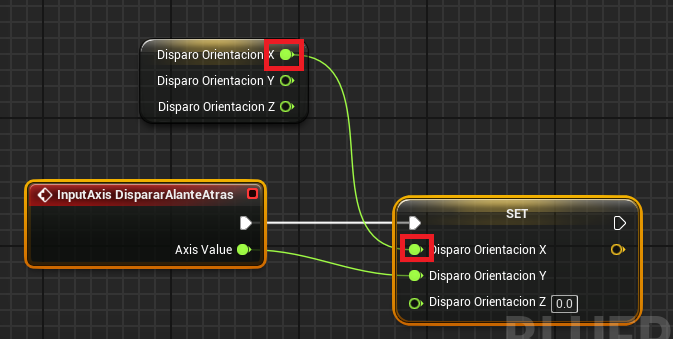
Tenemos que buscar una superficie libre y buscar la función **Get Disparo Orientación.**



Dividimos la variable con nuevamente con el **Split Struct Pin** ya que solo nos interesa el componente X.



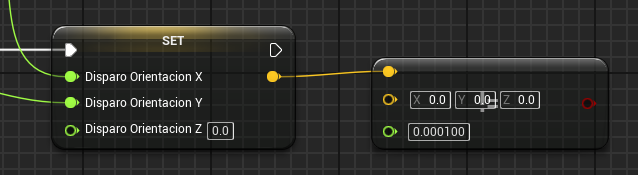
Colocamos el componente X anterior para que no se pise a 0.



Una vez guardado el dato vamos a hacer que, si el **Axis value** tiene algún movimiento distinto de 0, se rote al jugador.

Vamos a la salida del SET **disparoOrientacion** (ese punto es una referencia al Vector, lo mismo que hacer otro Get DisparoOrientación)Buscamos el nodo !=

|  |  |
| --- | --- |
|  |  |



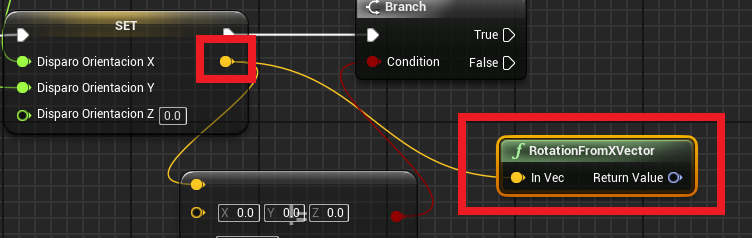
Ahora vamos a crear un nodo **Branch** (es igual que un IF)



**IMPORTANTISIMO**. Nos aseguramos de unir la entrada del **Branch** y los flujos de ejecución .



Obtenemos el **RotationFromXVector** (plano X)

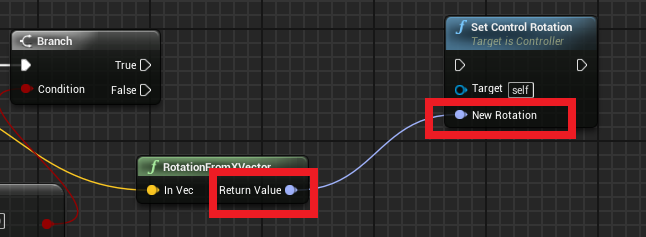


|  |
| --- |
| **\*Nota – sugerencias sensibles al contexto**  La herramienta que busca el nombre de funciones mejora las sugerencias sí partimos de un flujo de un dato.  Por ejemplo, si sacamos un flujo del Vector, solo nos mostrará funciones que pueden recibir un vector como parámetro |

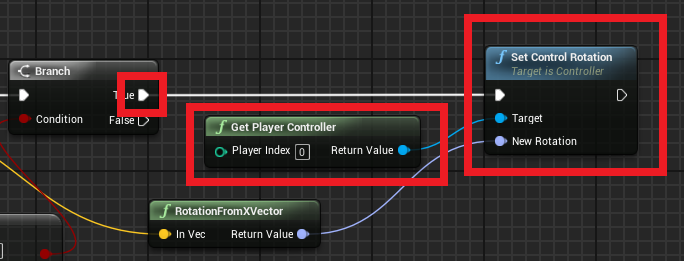
Buscamos la función **Set control Rotation (**probablemente haya que desactivar el **Context Sensitive**)

|  |  |
| --- | --- |
|  |  |

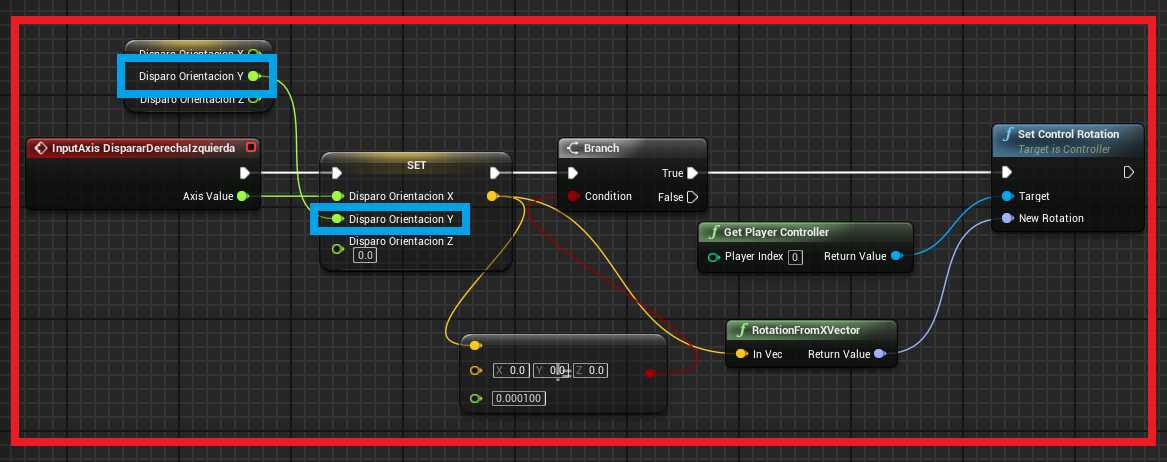
Le asignamos el vector de rotación del eje X



Solo falta asignarle el parámetro **Get Player Controller** y unir la línea de flujo True.



Faltaría por implementar el **DispararDerechaIzquierda**, el cual sería prácticamente igual al anterior. Solo cambia que modifica la orientación X y mantiene la orientación Y (al revés que el anterior).



Debemos asegurarnos bien de que todas las líneas de flujo están conectadas.

Compilamos y salvamos el Jugador.



Sí ejecutamos el juego y pulsamos el Pad derecho o las flechas del teclado el Jugador rotará.

