Making Macros

A Macro is used to check if a player has enough energy to jump.

Macros are essentially the same as collapsed graphs of nodes. They have an entry point and exit point designated by tunnel nodes. Each tunnel can have any number of execution or data pins which are visible on the macro node when used in other Blueprints and graphs.

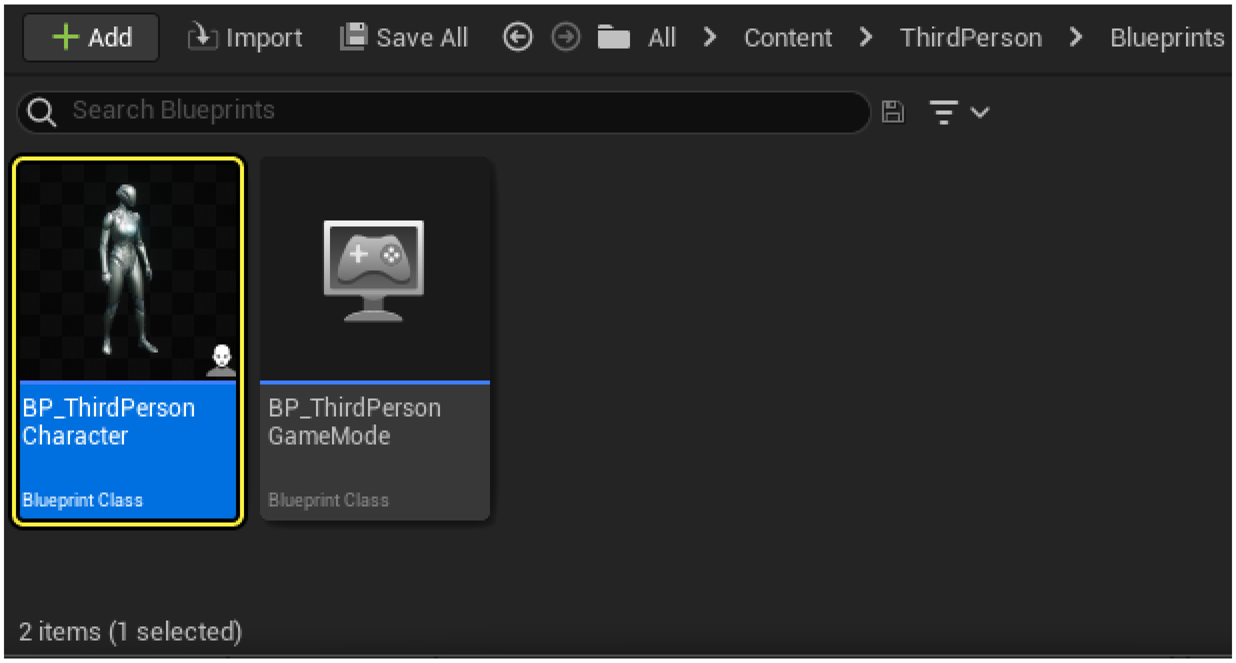
Creating Macros

In this tutorial, you will create a **Macro** to check if a Character has enough energy to jump. If they do have enough energy, then your macro will deplete energy from the player, print their current value to the screen, then call the jump function. If they do not have enough energy, your macro will print to the screen that "they are out of energy", and prevent them from jumping.

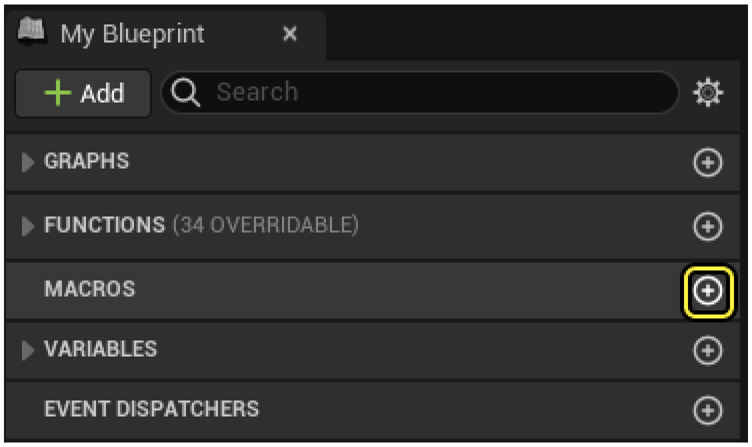
For this example, we are using the Blueprint Third Person Project with **Starter Content** enabled.

1. After **Creating** and **Launching** your project, navigate to the `Content/ThirdPerson/Blueprints` folder, and open the **BP_ThirdPersonCharacter**

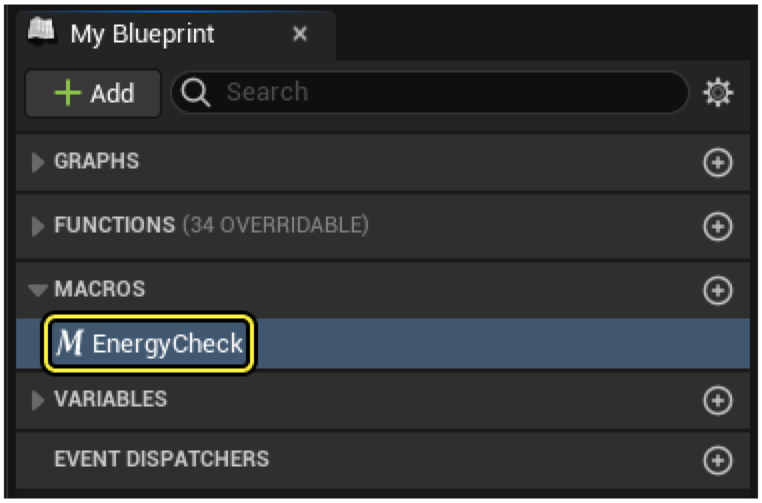
Blueprint.



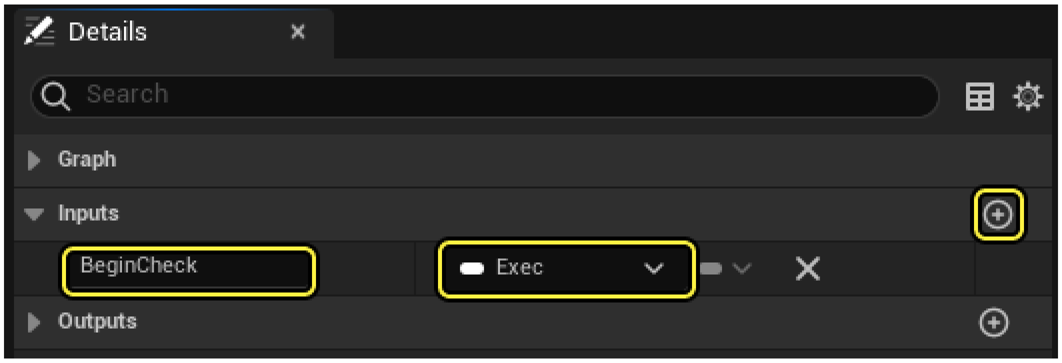
2. Navigate to the **My Blueprint** window, then click the **Add Macro** button.



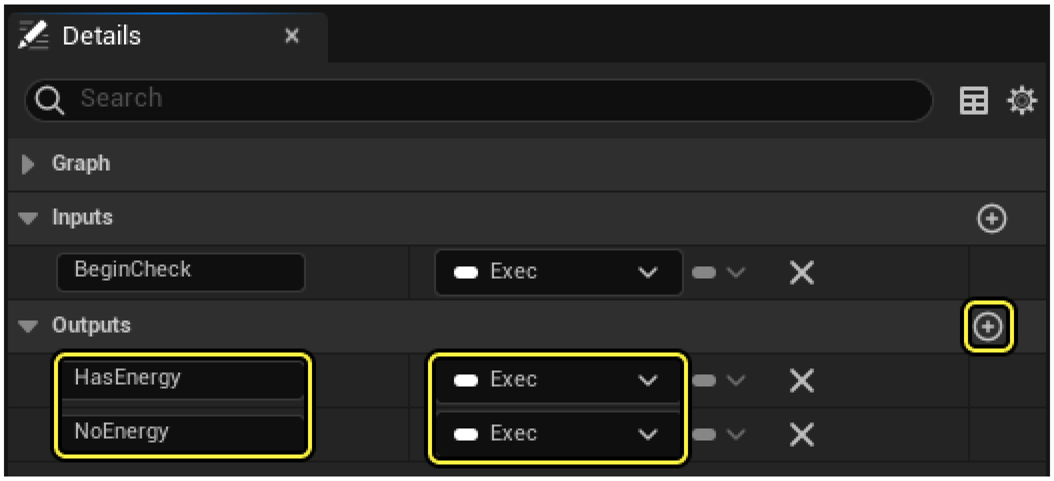
3. A new macro is created, select it and press **F2** to rename it to **EnergyCheck**.



4. With the macro selected, navigate to **Details** > Inputs**, click** Add(+) **to create a new** Input **named** BeginCheck**, then change its type to** Exec** (Execution pin)



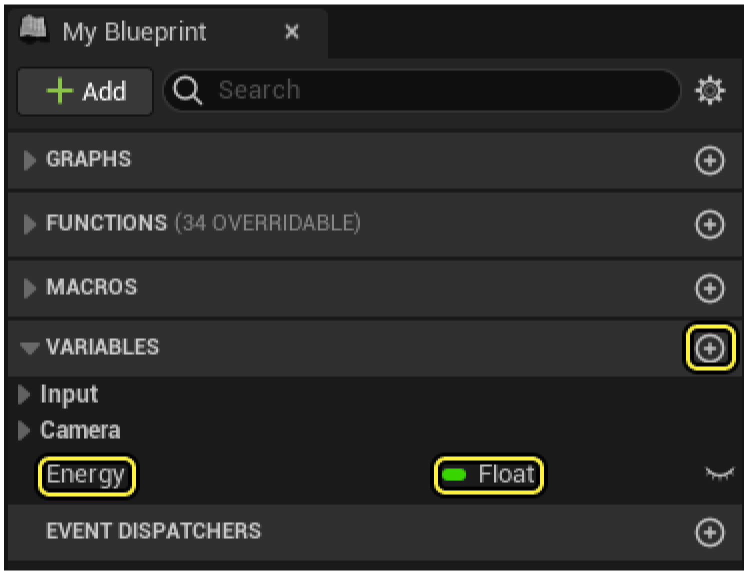
5. Navigate to **Details > Outputs**, and click **Add(+)** to create two new Outputs. Name one **HasEnergy** and the other **NoEnergy**, then set the **Exec** pin types.



This will create Input / Output nodes on the Macro Node itself that can be used to pass data to and from the Macro.

For the Input, use an Exec pin called **BeginCheck** to start the Macro. Next, create a script that checks if the player has enough energy to jump and if so, execute the **HasEnergy** pin. If the player does not have enough energy, execute the **NoEnergy** pin.

6. Inside the **My Blueprint** window, click the **Add Variable** button to create a new float variable named **Energy**.

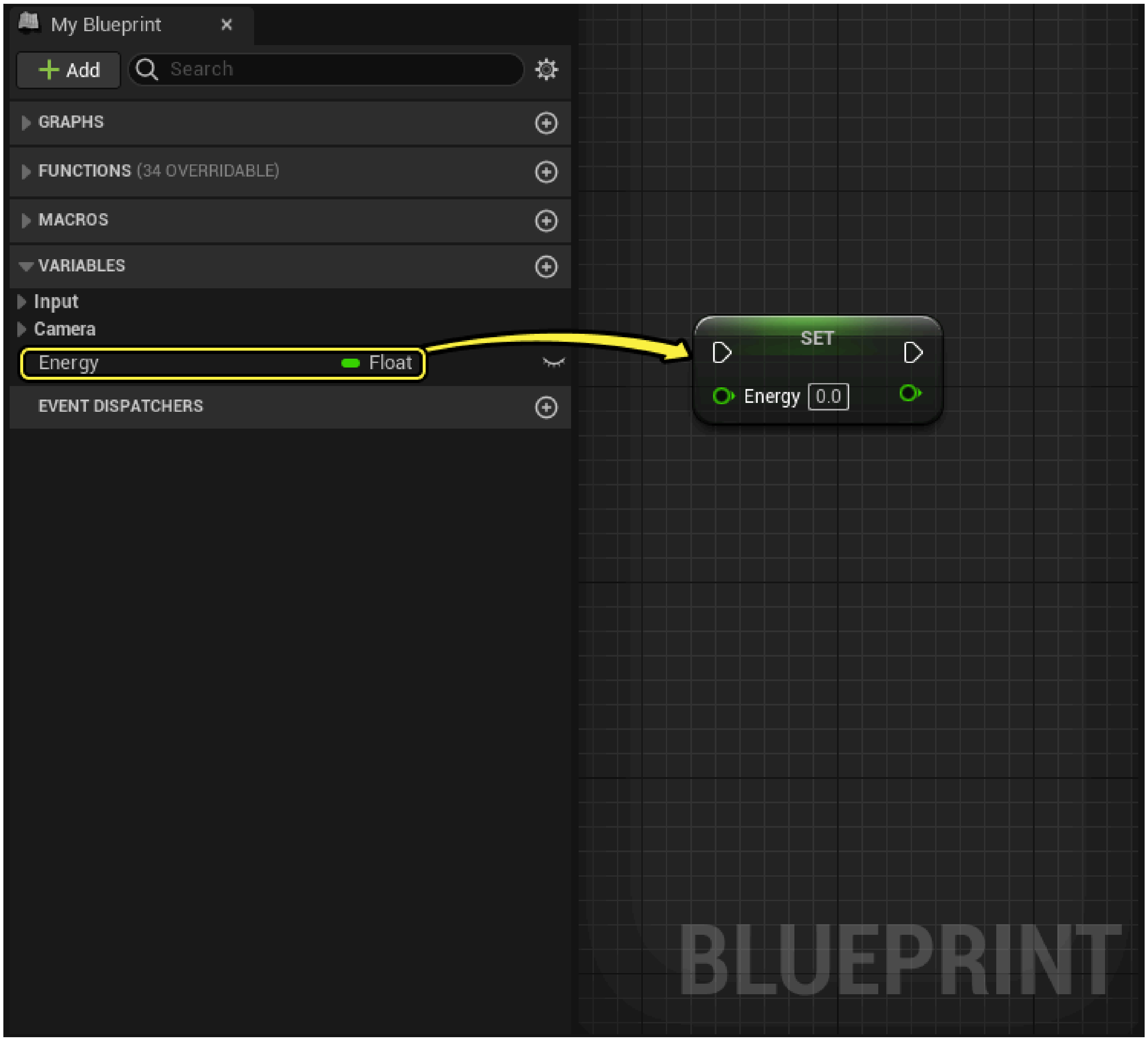


7. On the Toolbar, click **Compile**, then select **Energy** and navigate to the **Details** panel to set its value to **100**.

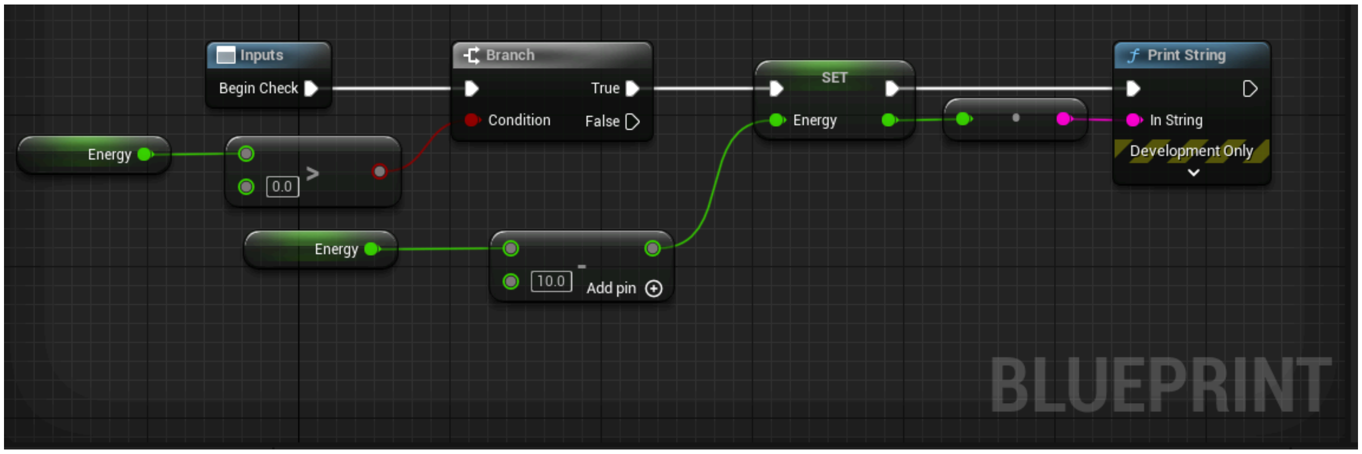
8. In the **Energy** graph, click **B** + **Left-click** to create a **Branch** node.

Copy code

9. While holding **Ctrl,** drag the **Energy** float variable from the **My Blueprint** tab into the macro graph, click and drag off the output pin to search for a **Greater** operator node, then connect the output pin to the **Branch**.
10. While holding **Alt**, drag in the **Energy** variable to add a **Set** node.



11. **Ctrl** drag in **Energy** again and connect it to a **Subtact**(**-**) node, set to **10** and connect it to the **Set** node. With this script, we define if Energy is > 0, subtract 10 from the current Energy value.
12. **Right-click** in the graph and add a **Print String** node, then connect the **Set Energy** node to the **In String** pin.



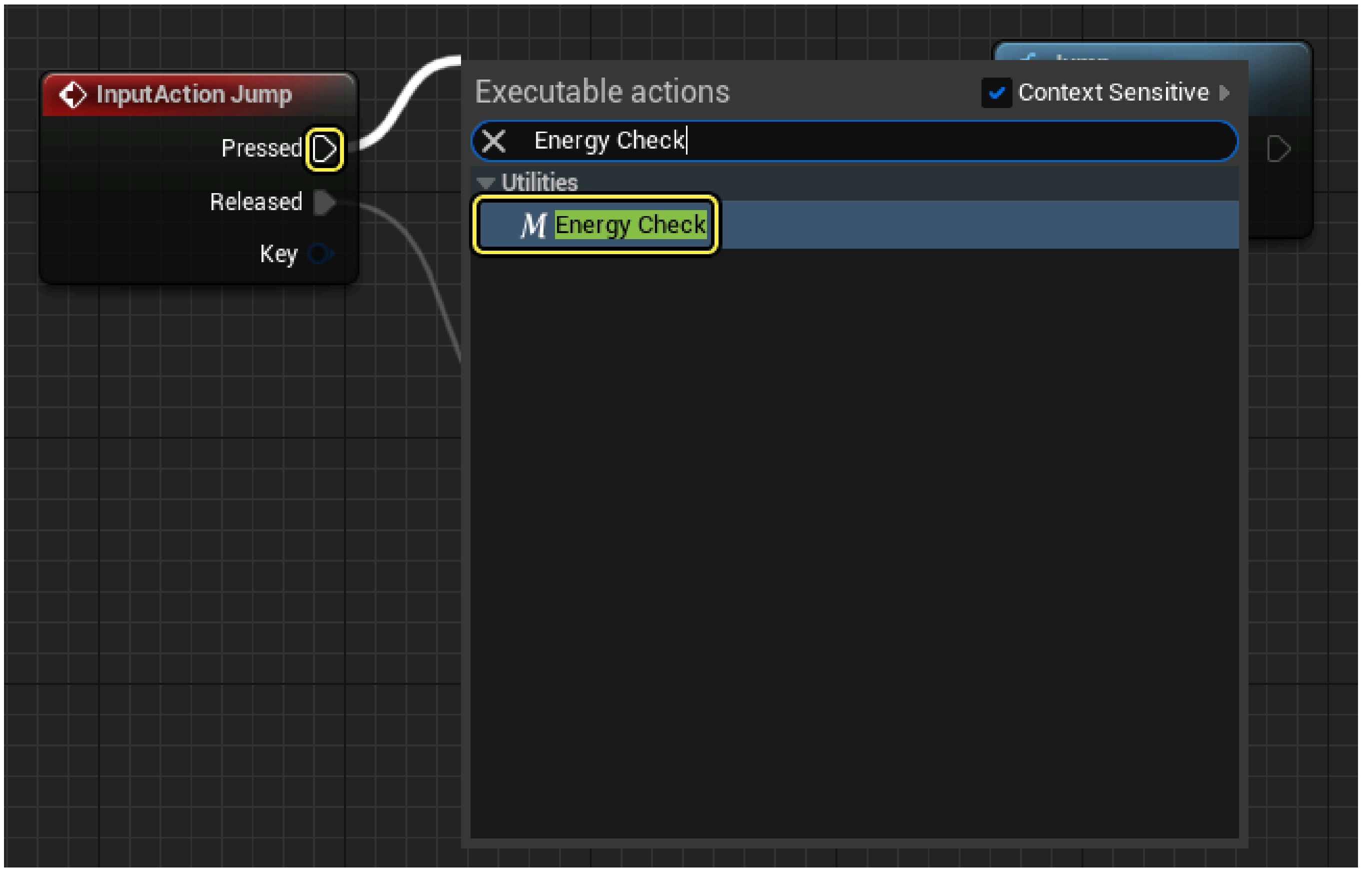
A **Conversion** node is added which converts the value of Energy into a string printed to the screen, displaying its value.

13. Off the **False** pin of the **Branch**, add another **Print String** node and enter the text **"Out of Energy!"** in the box. Then connect the first and second **Print String** nodes to the **HasEnergy** and **NoEnergy** pins, respectively.

Copy code

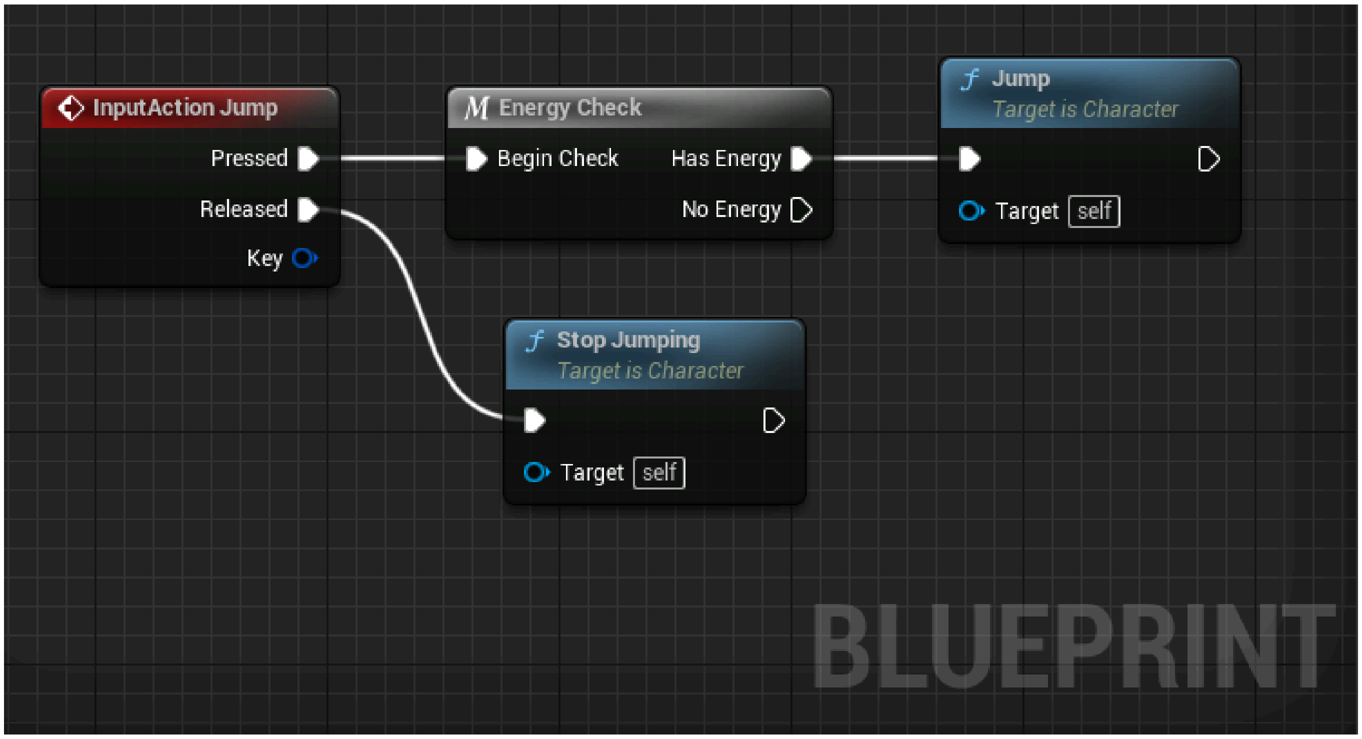
The macro is now set up to check (and if applicable, subtract from) the **Energy** variable and print if the player has enough energy or not, which is used to determine if the player can jump or not. Now, you need to implement the macro after pressing the "Jump" key, but before the Jump action is executed.

14. On the **EventGraph**, drag off the **Pressed** pin of **InputAction Jump** node and search for **EnergyCheck**.

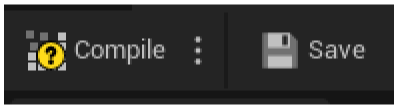


You should see that the macro you created is listed under **Utilities** and has the Macro icon next to its name.

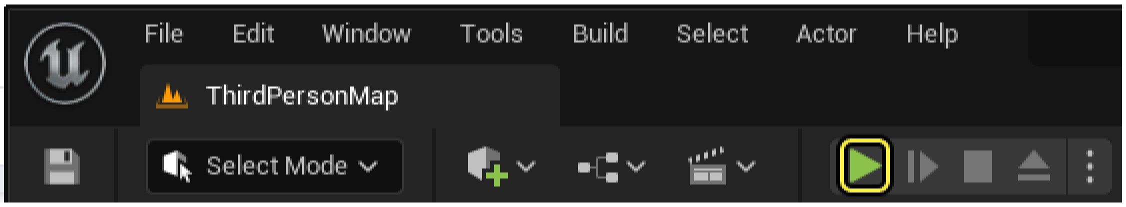
15. When the macro is added, the Jump script should look similar to below.



16. Click the **Compile** and **Save** buttons, then close the Blueprint.

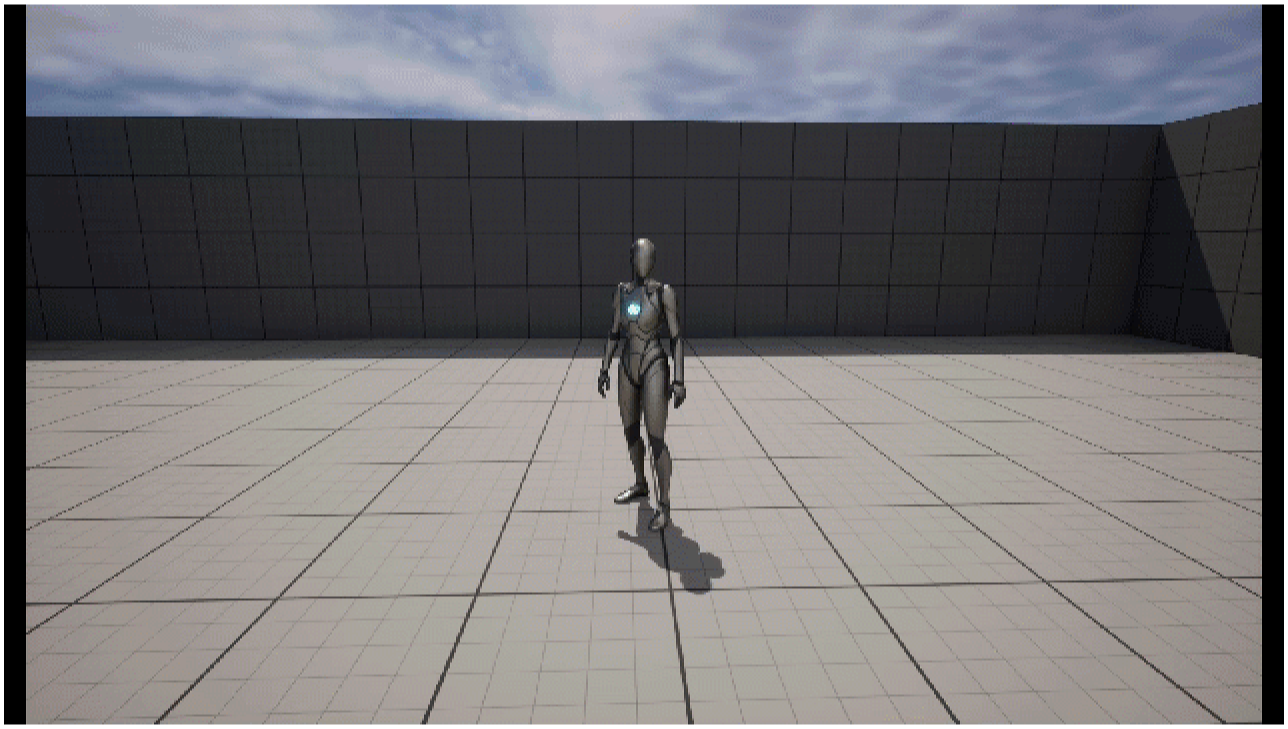


17. Click **Play** on the main **Toolbar** in the **Editor**.



When you press the **Spacebar** to jump, in the upper left corner of the screen the value of **Energy** is printed to the screen. When **Energy** is 0, you should no longer be able to jump.

End Result



This is a basic example of using a Macro to execute and consolidate a script into a single node, improving the readability of our Event Graph and main character script. Additionally, you could call this Macro in other instances. For example, if you have some other action that reduces the player's energy, and you want to check if they have enough energy to perform the action (like attacking), you could run this Macro to check if the player has enough energy to Attack after pressing your Attack key then execute an attack off of the **HasEnergy** exec pin.