

SIGNAL CALIBRATOR BLUEPRINT

An idea coming for the project has been the work to process data and send it from the base to earth or another location and has been so far the most efective by radio wave signals going acros the space by the same speed of the light and still going even by the void of space, thanks to the report of the NASA by the link down below:

https://science.nasa.gov/ems/05_radiowaves#:~:text=Radio%20telescopes%20look%20toward%20the,composition%2C%20structure%2C%20and%20motion.

For the project we think the idea of using the joysticks on an average controller, the build itself use gears to detect the input of the coordinates on the coordinates (X , Y) and goes from the origin (0 , 0) and range from (-1 , -1) to (1 , 1)



With this and inspired by the game Batman Arkham Knight using one of the gadgets that reads the controller joysticks we made a blueprint that will get the input coordinates of both joysticks (left (x,y) and right (x,y)) to simulate a calibration of radio signal to be adjusted by the player.



First thing on the build is to be in range to interact on the device and enable the input of the joysticks by using a collision box would read the interaction button (X button for xbox controller) then start reading the current position of the buttons then evaluate the position of the objective range where is supposed to be and when is right end the process.

For the reading of the joysticks coordinates and the objectives to be assigned was made two variables that will be an array of float type value following the next logic on the table down below

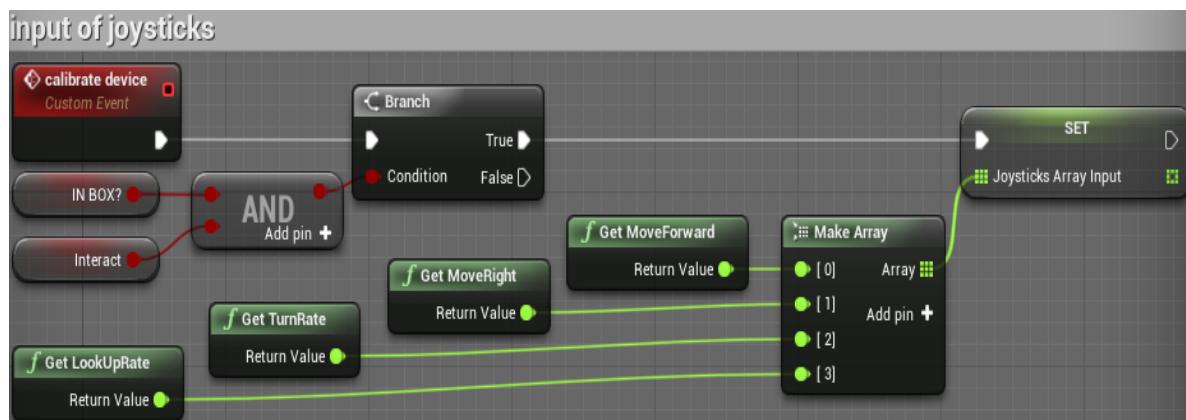
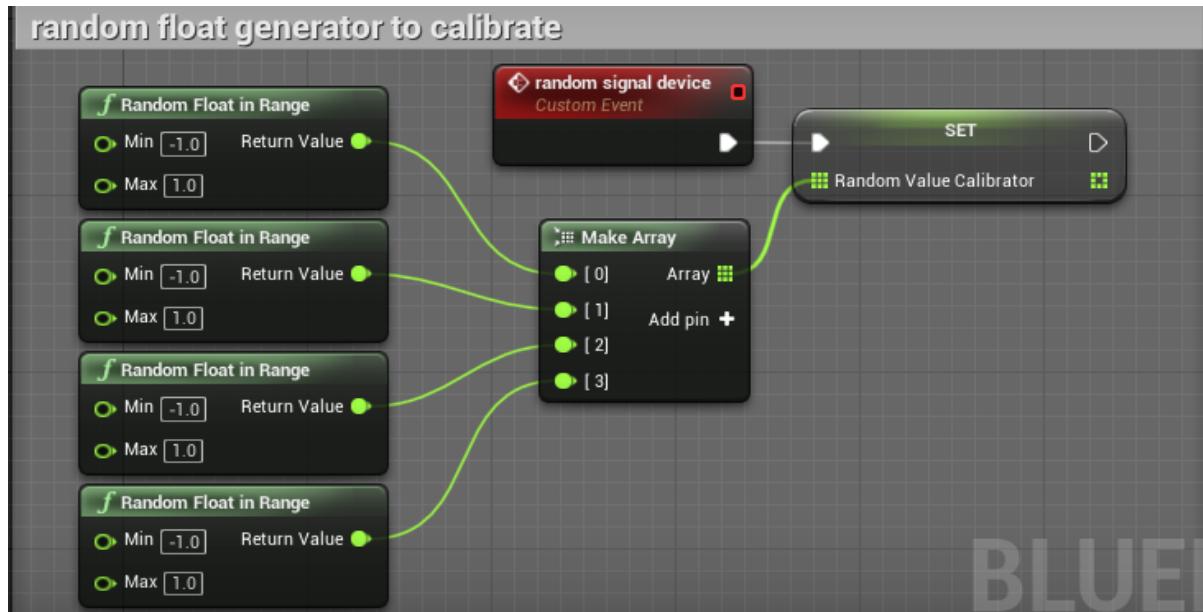
The term floating point refers to the fact that a number's radix point (decimal point also known on computers binary point can “float”)

Array Index	Joystick Array Input (Float)	Random Value Calibrator (Float)
0	Left joystick X input (-1 to 1)	Random Float with range (-1 to 1)
1	Left joystick Y input (-1 to 1)	Random Float with range (-1 to 1)
2	Right joystick X input (-1 to 1)	Random Float with range (-1 to 1)
3	Right joystick Y input (-1 to 1)	Random Float with range (-1 to 1)

With this the slightest movement of both joysticks can be as precise x.xxx in both positive and negative values of the range between one and minus one

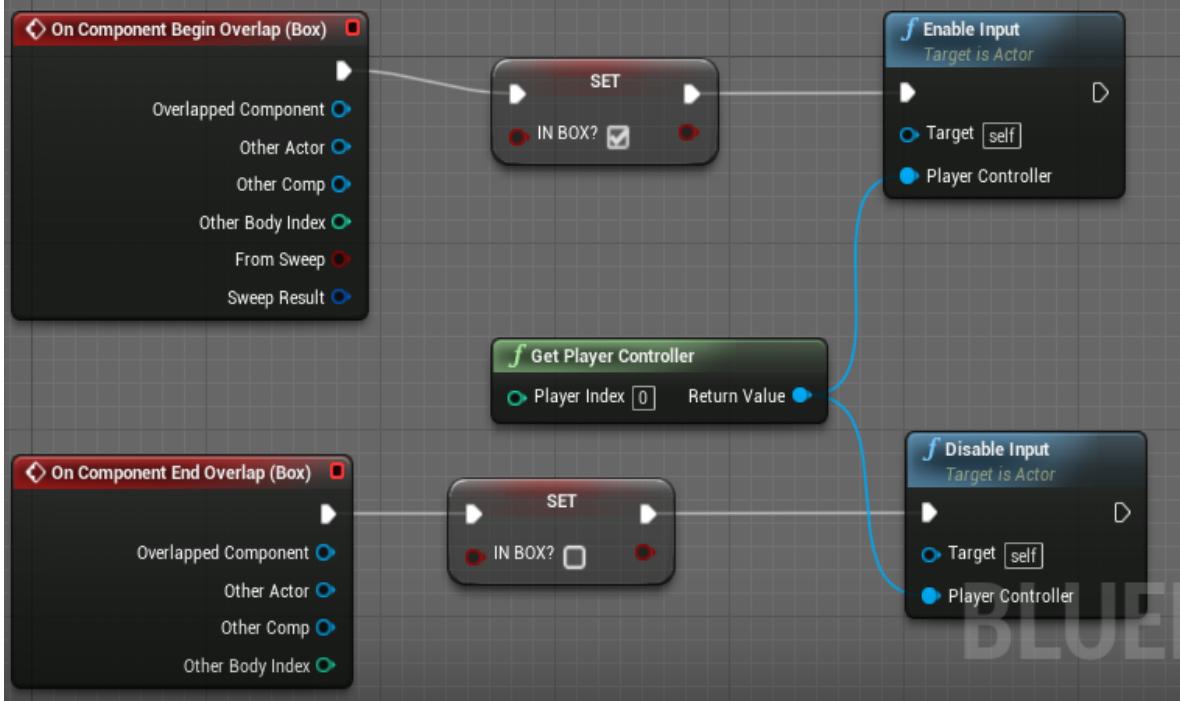
SCREEN CAPTURE OF BUILD OF THE CALIBRATOR BY BLUEPRINT

As mentioned before the build of projects in Unreal Engine 4 by the blueprints save time and process of logic, build an test to make the development of the projects an the components, in this part of the report will be showing the creation of the calibrator by it's own blueprint system to improve the understanding of how is working.

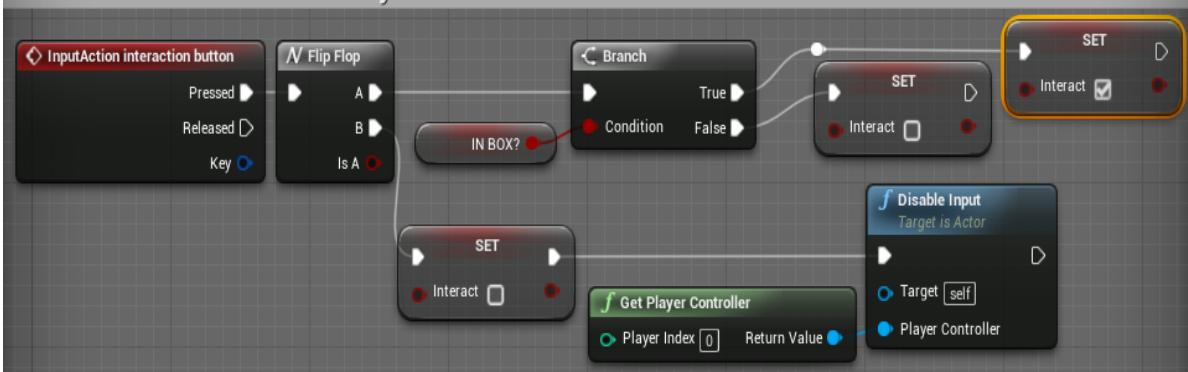


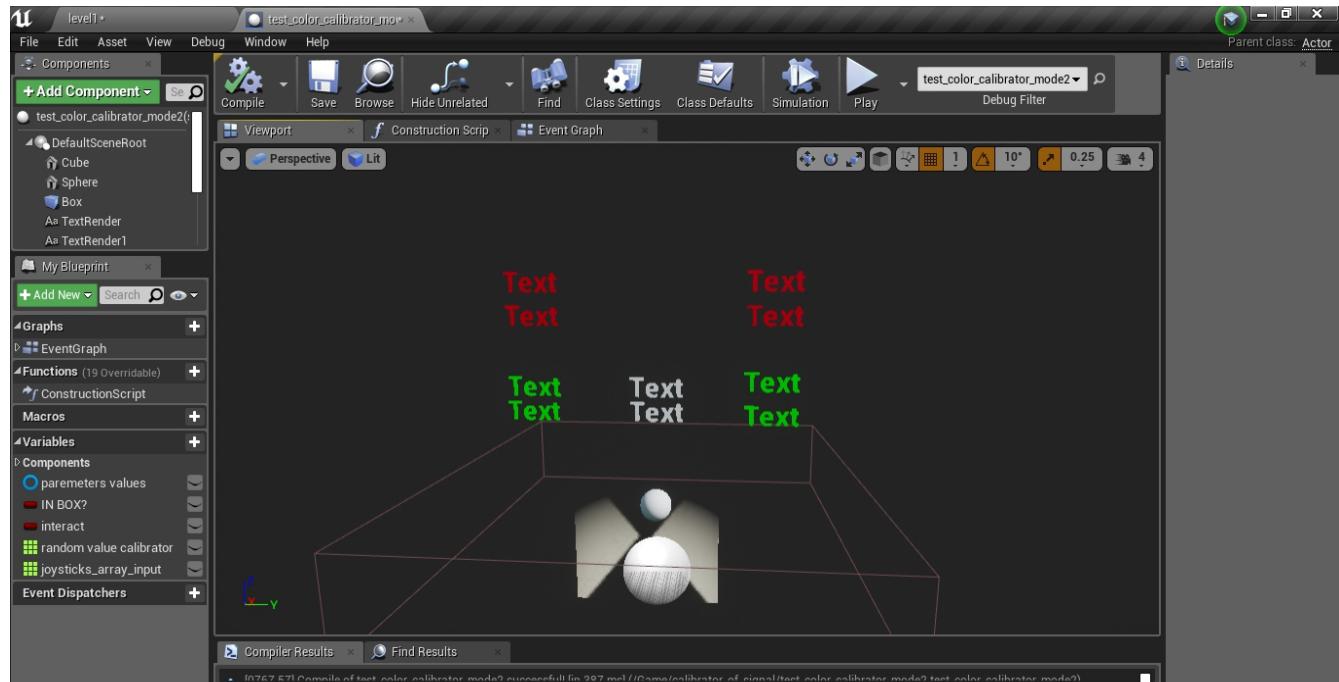
The functions (Get LookUpRate, Get TurnRate, Get MoveRight, Get MoveForward) showing on the previous capture is to read the input of both joysticks coordinates vertically and horizontally

enable input when close on collision box



enable and disable interaction by button



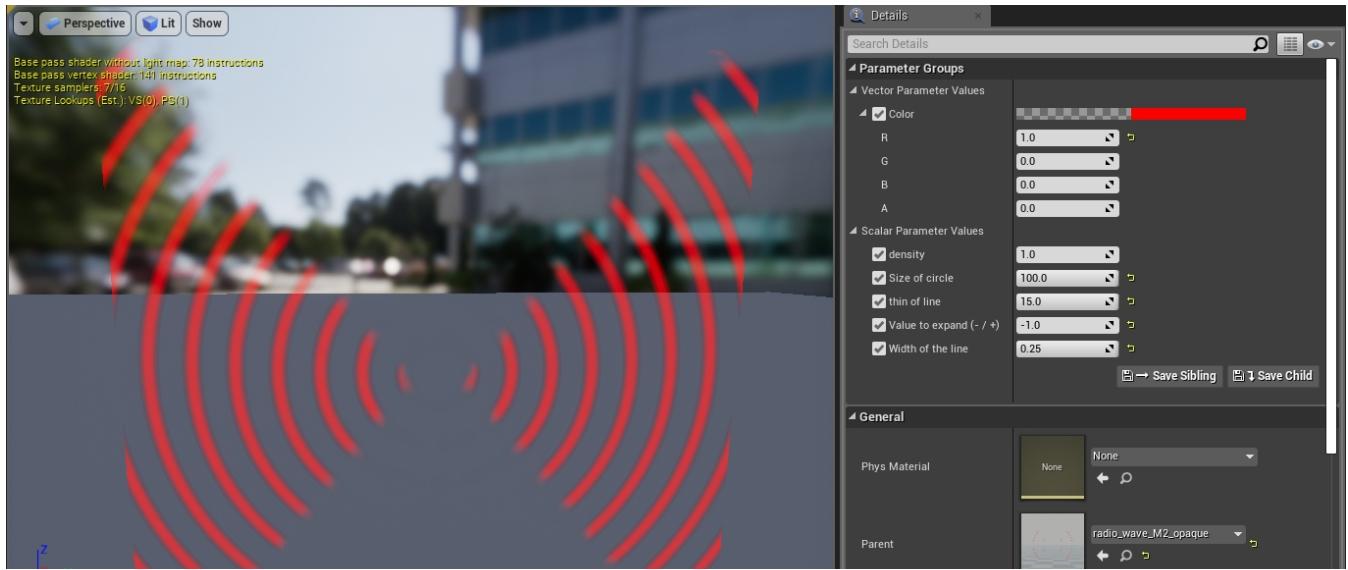
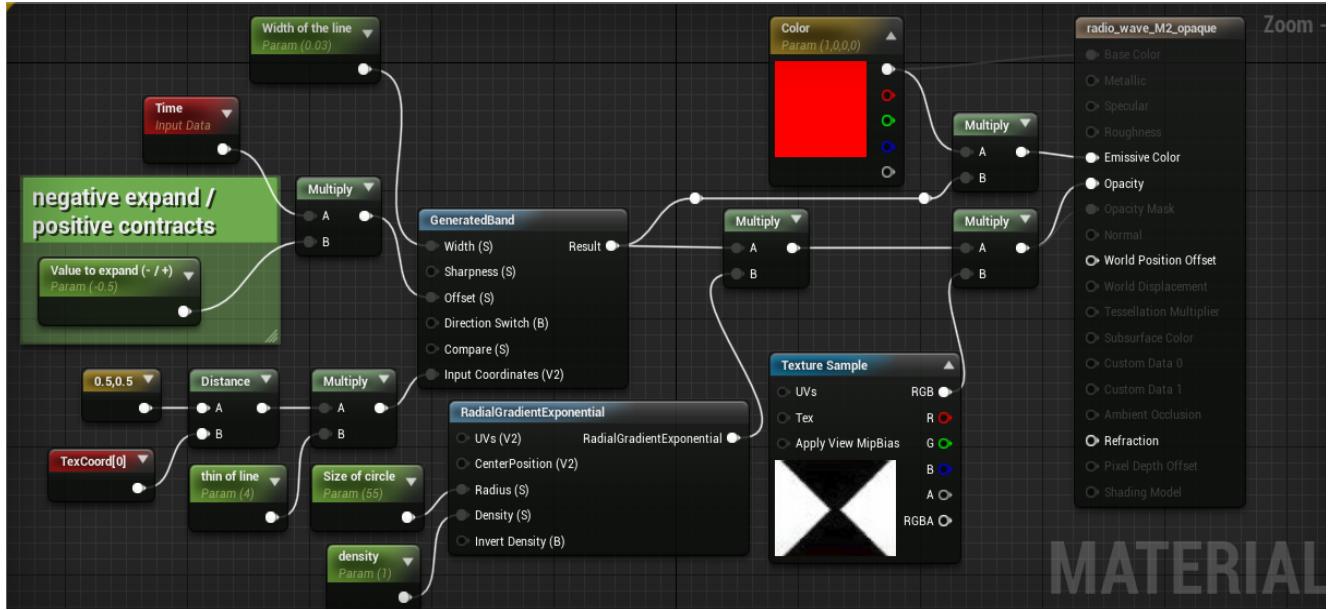


With the build of the calibrator was also placed a Material Build to simulate Signal and make dynamic parameters in order to manipulate and modify through the blueprint of the calibrator.

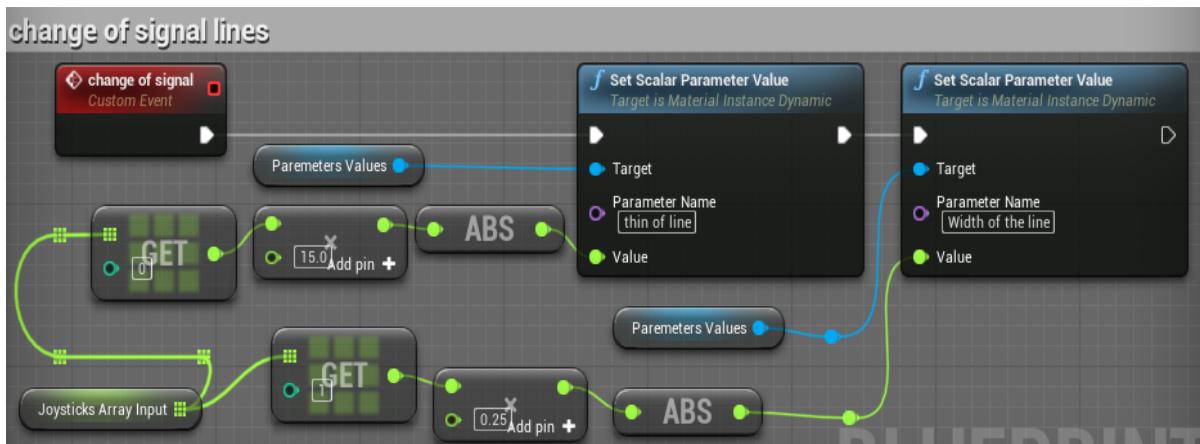
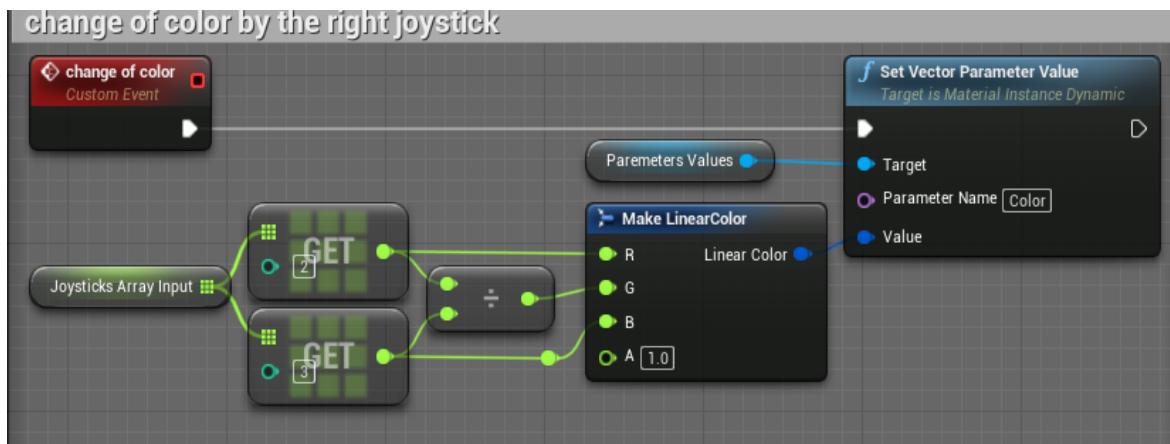
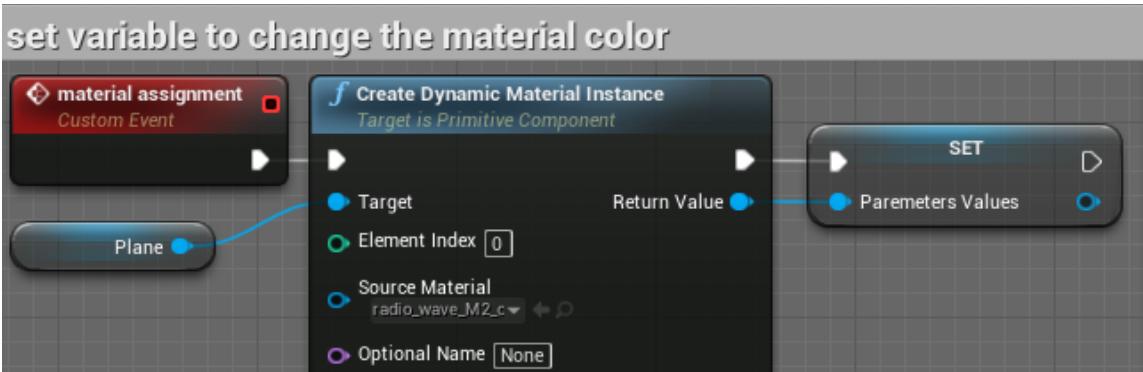
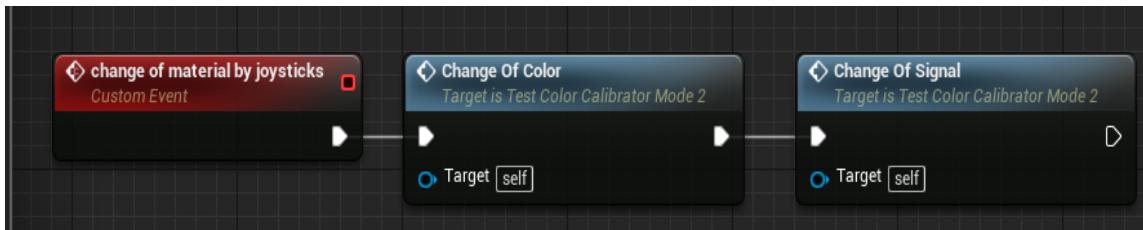
Following the tutorial on the link down below we got an idea to start:

<https://elarasystems.com/reating-radio-waves-using-generatedband-2/>

Then we made changes on the build and added paramters and other functions to make it able to adapt on the values for the Joysticks

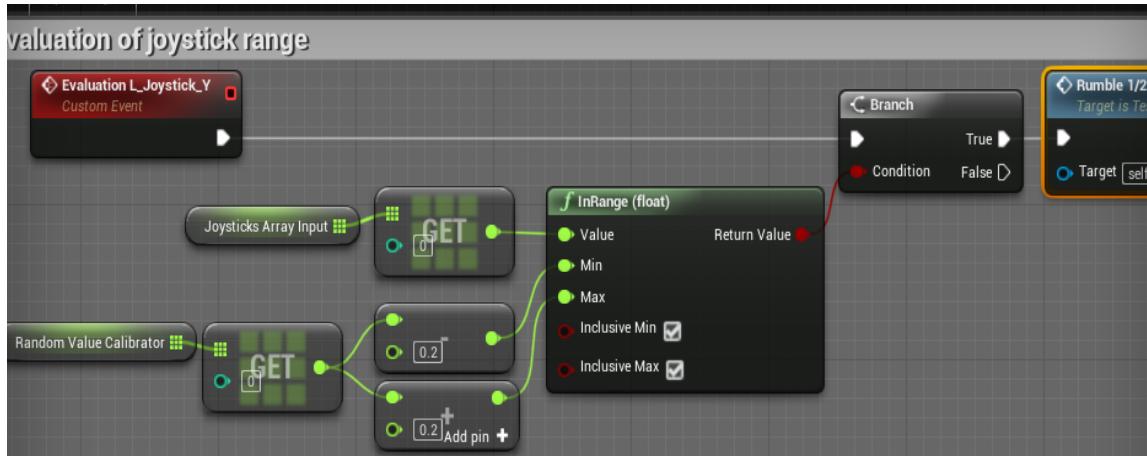


With that created is being made the new segment of the blueprint to access the and change the values of the parameters as showed on the next screen captures



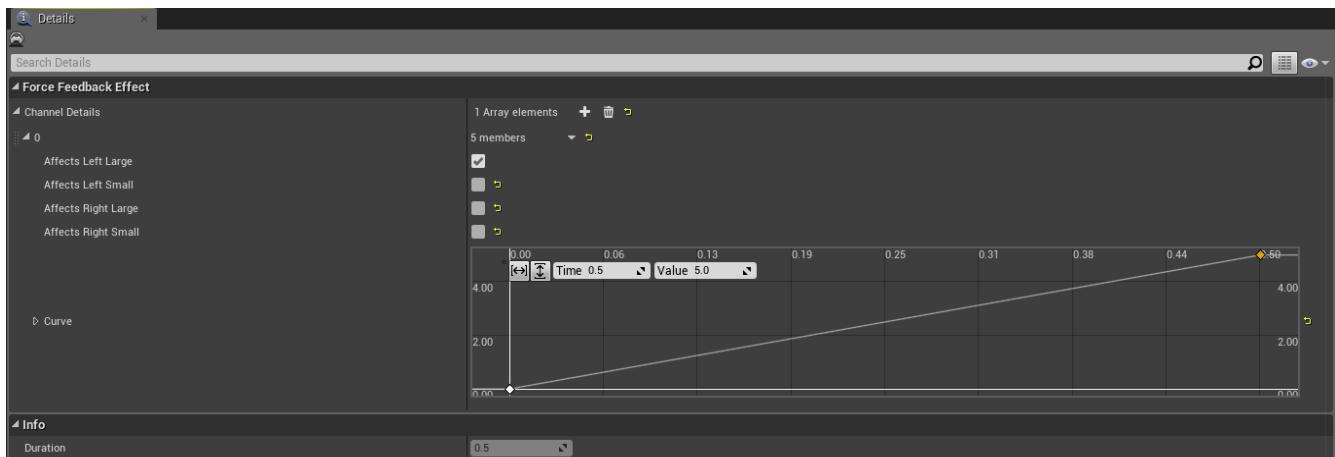
<https://youtu.be/mKGD00mh9LM>

Another concept at the moment to calibrate the coordinates to match we use a comparison function for each array index where are the values of the joysticks and the random generator, since the value of the float is so small is practically impossible to be equal completely on the level of being the value x.xxx but we make it to be close on a range of 0.1 but didn't work as expected being to high and causing to be automatically calibrated when is for example the value to match 0.0845 and the Joystick input currently on 0 (not moved) calibrated so we change the range comparison to calibrate on 0.02 and the results have been more satisfying.



Is the same build for the other inputs and random values only change is the Index value from 0 to 3 on the GET function for the array values showing on the picture.

As well to indicate is getting on the right spot is placed a function to vibrate the controller, other function is the force feedback that makes the work to make the motor of the controller to rumble



Controller Vibration



