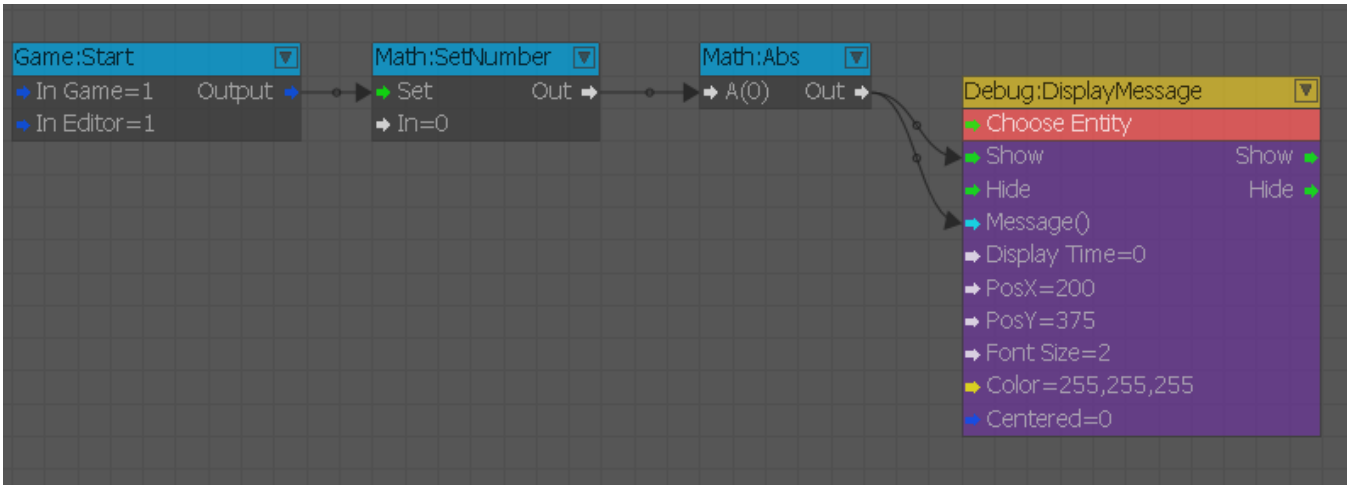
- Math:Abs
- Math:Add
- Math:AnglesToDir
- Math:ArcCosinus
- Math:ArcSinus
- Math:ArcTangens
- Math:Auto3DNoise
- Math:AutoNoise1D
- Math:BooleanFrom
- Math:BooleanTo
- Math:Calculate
- Math:Ceil
- Math:Clamp
- Math:Cosinus
- Math:CosinusInverse
- Math:Counter
- Math:DirToAngles
- Math:Div
- Math:Equal
- Math:EqualCheck
- Math:Floor
- Math:Geometry
    - SwitchCoordinateSpace
    - TransformSpace
- Math:InRange
- Math:Less
- Math:LessCheck
- Math:Mod
- Math:Mul
- Math:Noise1D
- Math:Noise3D
- Math:PortCounter
- Math:Power
- Math:Random
- Math:Reciprocal
- Math:Remainder
- Math:Round
- Math:SetColor
- Math:SetInteger
- Math:SetNumber
- Math:SinCos
- Math:Sinus
- Math:SinusInverse
- Math:Sqrt
- Math:Sub
- Math:Tangent
- Math:TangentInverse
- Math:UpDownCounter
- Math:Wrap

Math:Abs

This node is the maths operation (Absolute) that will convert the input number from negative to positive.



In the above example we have the negative number of -14.6 & after it has been through the "Math:Abs" node, the result (positive number) is outputted to the hud.

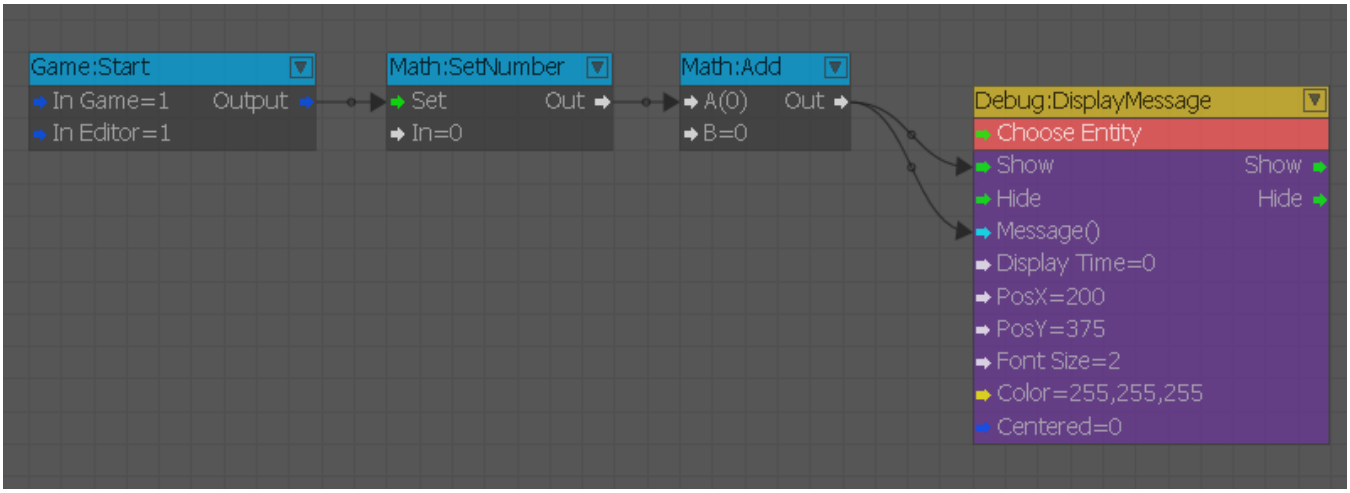
Port Description

Input	Description
Float *"A"*	The input number to be calculated by the Abs function
Output	**Description**
Float *"out"*	Outputs the result

Math:Add

This is a simple operation of adding Input B to Input A, then outputting the result. You can directly set a number into the node for A or B, or input a value from somewhere else. (Both are show in the picture).

In the following example, we have a flowgraph to output the result of the "Math:Add" to the HUD.



Input	Description
Float *"A"*	The first number being added to
Float *"B"*	The second number to add the first
Output	**Description**
Float *"out"*	Outputs the result

Math:AnglesToDir

Used to convert the input angle to a unit vector direction.

Inputs

Port	Type	Description
Angles	Vec3	Input angle

Outputs

Port	Type	Description
Dir	Vec3	Direction unit vector
Roll	Float	Roll output

Math:ArcCosinus

Used to calculate the inverse cosine of the input.

Inputs

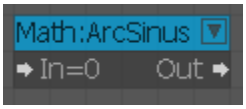
Port	Type	Description
In	Float	Input angle

Outputs

Port	Type	Description
Out	Float	Output angle

Math:ArcSinus

Used to calculate the inverse sine of the input.



Inputs

Port	Type	Description
In	Float	Input angle

Outputs

Port	Type	Description
Out	Float	Inverse sine (Arcsinus) of the input

Math:ArcTangens

Used to calculate the inverse tangent of the input.

Inputs

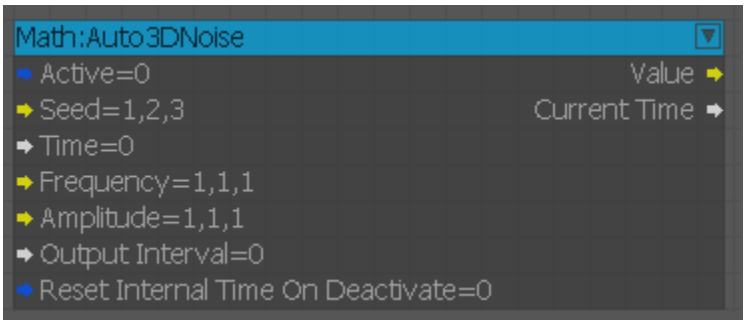
Port	Type	Description
In	Float	Input angle

Outputs

Port	Type	Description
Out	Float	Inverse tangent (ArcTangens) of the input

Math:Auto3DNoise

Generates continuous 3D noise, smoothly interpolating between a series of randomized 3D coordinates (Vec3 or X,Y,Z values). Output values are randomized based on a Vec3 seed value input and output as a Vec3 value.

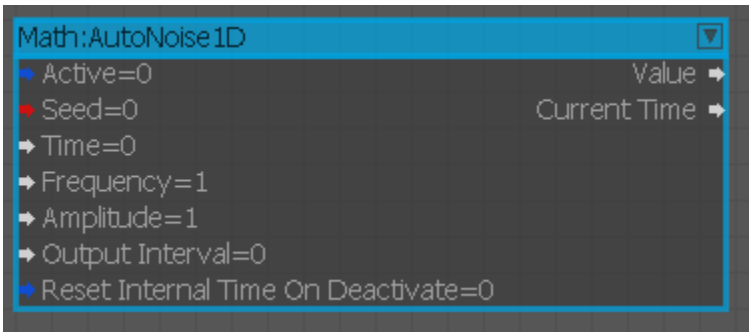


Input	Type	Description
Active	Boolean	If true, the node will output values constantly at specific intervals.
Seed	Vec3	A Vec3 seed value from which randomization will occur, allowing each axis to be set independently.

Time	Float	Specifies the time interval at which noise must be sampled. If Active port is TRUE it will output. This is also the initial sample time if the Active port is FALSE
Frequency	Vec3	Scale factor for input value. Out = amplitude * Noise(frequency * x).
Amplitude	Vec3	Scale factor for noise values. Out = amplitude * Noise(frequency * x).
Output Interval	Float	How often the value outputs if active (in seconds).
Rest Internal Time On Deactivate	Boolean	If the node is deactivated (by the Active port), internal time resets to 0.0f.
Output		**Description**
Value *"False"*	Vec3	Randomized Vec3 values.
Current Time *"True"*	Float	Floating point time value.

Math:AutoNoise1D

Generates continuous noise as floating point values, smoothly interpolating between a series of randomized floating point values based on a seed value input.

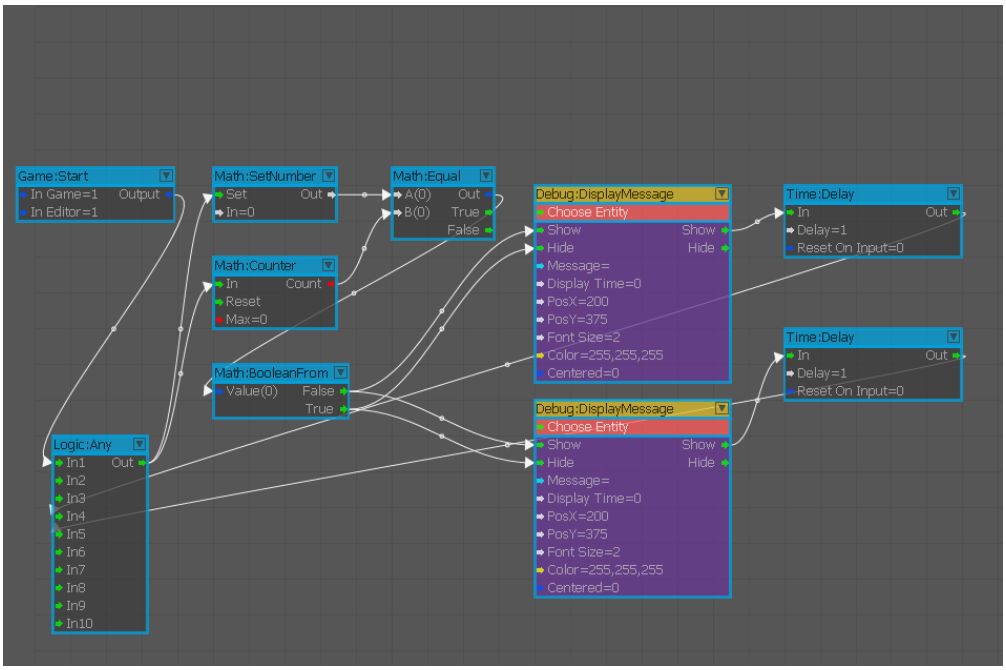


Input	Type	Description
Active	Boolean	If true, the node will output values constantly in a certain interval.
Seed	Vec3	A Vec3 seed value from which randomization will occur, allowing each axis to be set independently. More information on seed values here.
Time	Float	Time at which to sample noise. If Active port is TRUE it will output. This is also the initial sample time if the Active port is FALSE
Frequency	Vec3	Scale factor for input value. Out = amplitude * Noise(frequency * x).
Amplitude	Vec3	Scale factor for noise values. Out = amplitude * Noise(frequency * x).
Output Interval	Float	How often the value outputs if active (in seconds).
Rest Internal Time On Deactivate	Boolean	If the node is deactivated (with the active port), internal time resets to 0.0f.
Output		**Description**

Value *"False"*	Float	Randomized floating point value.
Current Time *"True"*	Float	Floating point time value.

Math:BooleanFrom

This node will split a boolean input into a true / false (1/0) output.



In the above example, the flowgraph loops around switching the input of the "Math:FromBoolean" from true to false.

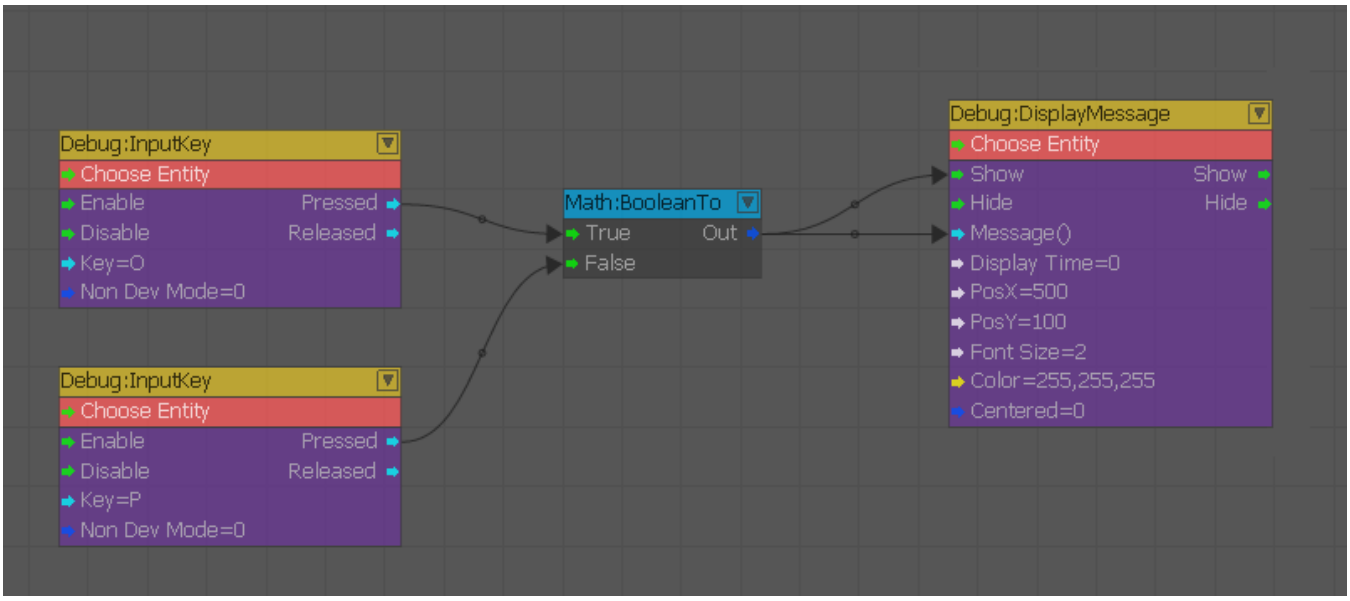
The main part of this example is the section from the "Math:Equal" onwards. The OUT of the Math:Equal is a boolean output which the "Math: FromBoolean" reads and then forwards the signal onto the HUD message nodes.

This then loops back to cycle the number from 1 to 0 (true / false) via the "Math:Counter". The result is that it will cycle between the 2 HUD messages every second.

Input	Description
Boolean *"Value"*	
Output	**Description**
Any *"False"*	Outputs the result if False (0)
Any *"True"*	Outputs the result if True (1)

Math:BooleanTo

This node will take the input of true or false, and convert it into a boolean output.



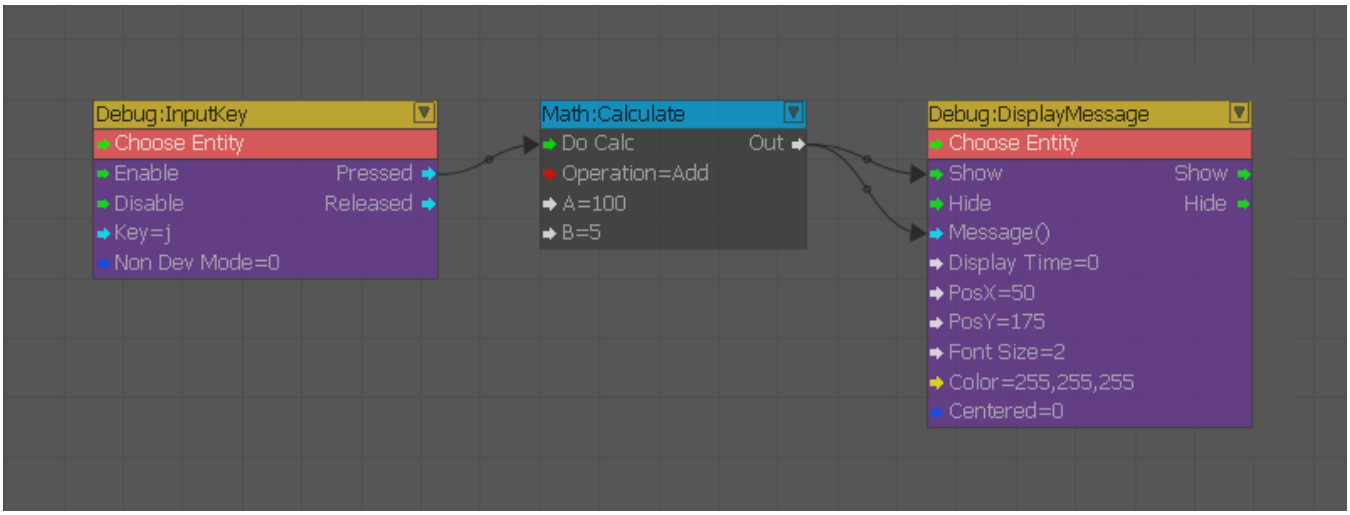
In the above example upon press the key "O" to trigger the true input, it will make the Math:ToBoolean output a 1 to the HUD. And then upon pressing the key "P" to trigger the false input, the Math:ToBoolean will convert the signal to output a 0 to the HUD.

Input	Description
ANY *"true"*	The input signal to be converted
ANY *"false"*	The input signal to be converted
Output	**Description**
Boolean *"out"*	Outputs the result in Boolean format (0 or 1)

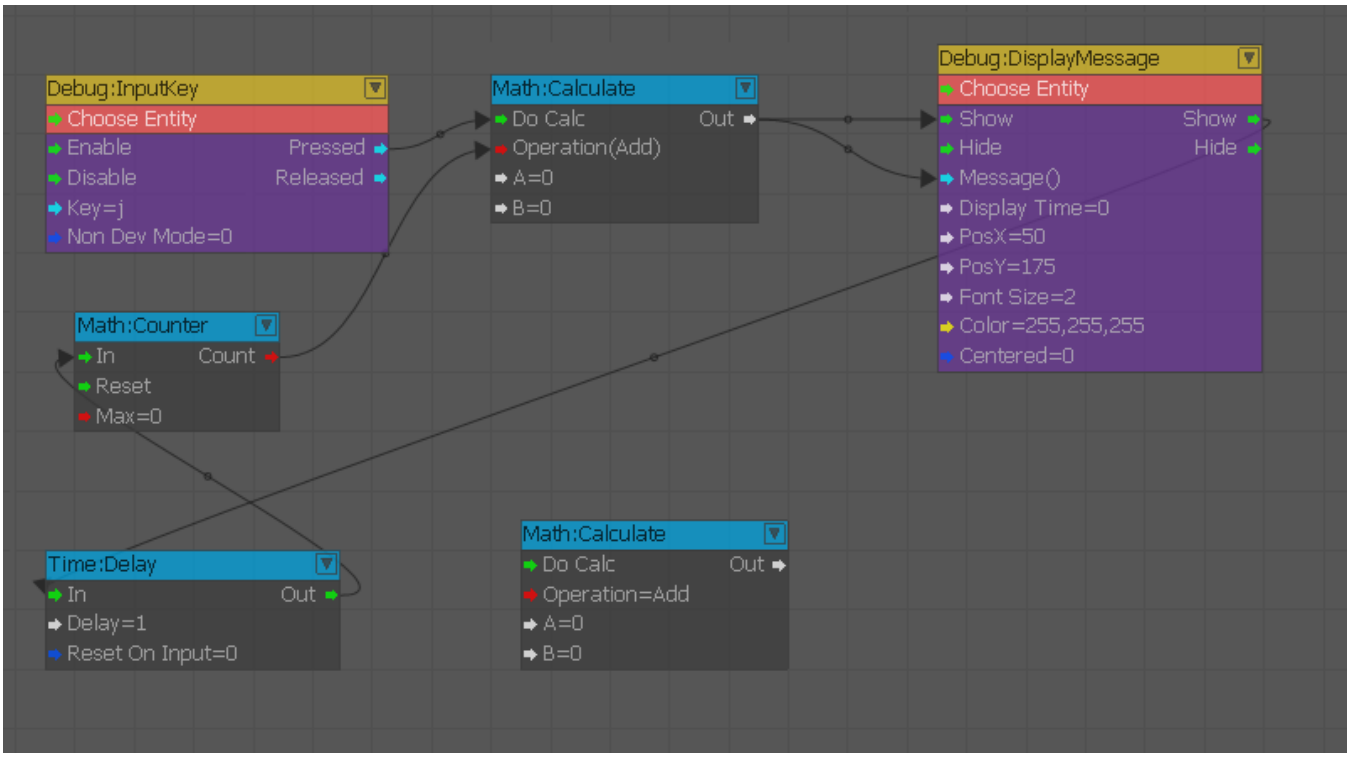
Math:Calculate

This node is an all in one basic math node. From the operation field on the left side, you can select from Add, Subtract, Multiply or Divide.

Then it will do the selected operation on A and B then output the result.



In the above example, on pressing the key "J" the Math:Calculate node will do the selected operation of "Add" on A & B.



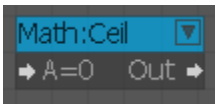
In this example the upon pressing the key"J", the Math:Calculate node will do the first operation (Add) then after it has displayed it to the HUD, it will cycle through to the next math operation after a 1 second delay and pressing "J" again.

First it will do the Add, Divide, Multiply then Subtract.

Input	Description
Any *"DoCalc"*	Executes the operation
Int *"Operation"*	The Drop down list to select the operation to preform. Select from Add, Divide, Multiply or Subtract.
Float *"A"*	The first number to use in the calculation
Float *"B"*	The second number to operate on the first
Output	**Description**
Float *"out"*	Outputs the result

Math:Ceil

Used to output the ceiling value of the input.



Inputs

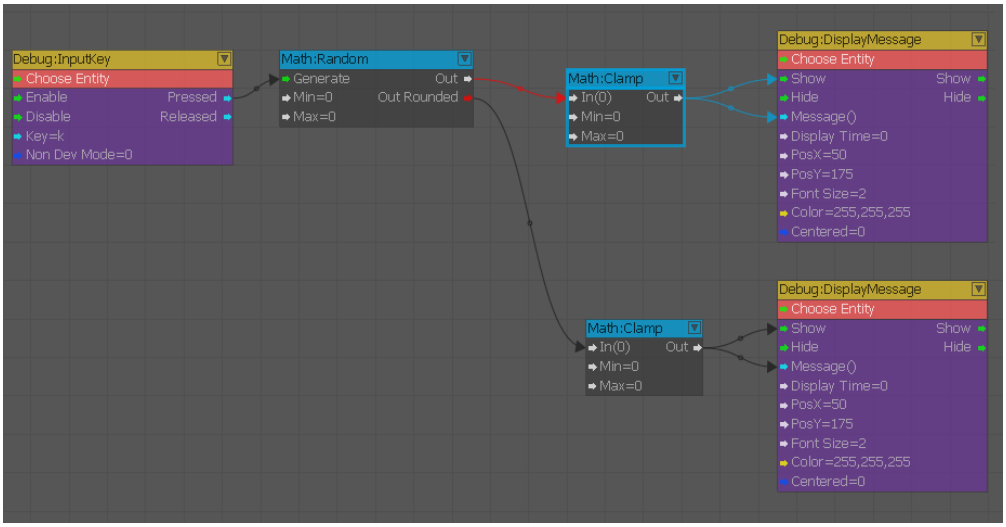
Port	Type	Description
In	Float	Input

Outputs

Port	Type	Description
Out	Float	Ceiling input value

Math:Clamp

This node will take a float number and clamp it to the specified range set within its parameters.



In the above picture, upon pressing the key "K", the "Math:Random" will generate a number between 0 and 100. The signal then passes to the two "Math: Clamp" nodes where they have been set to only output a number between 0 and 50.

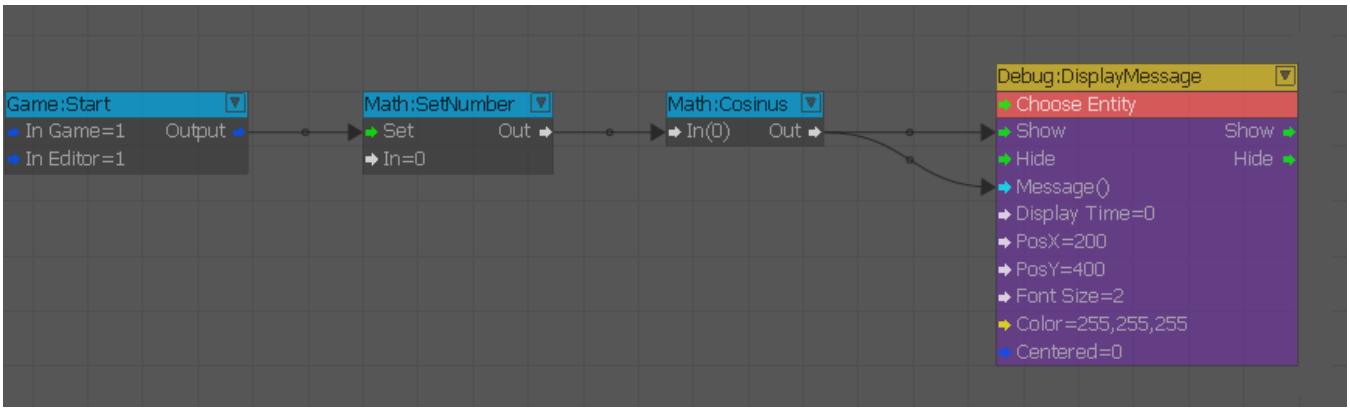
So if the number falls within the 0 -> 50 range it will pass through as normal. But if the number is higher than 50 (the maximum set), it will be clamped down to 50.

In reverse, if the minimum was set to say 20, any number lower than the minimum would be raised up to 20.

Input	Description
Float *"in"*	The input number to be clamped
Float *"min"*	The minimum of the clamp range
Float *"max"*	The maximum of the clamp range
Output	**Description**
Float *"out"*	Outputs the result

Math:Cosinus

This node will take the input of an angle in degrees and output the result in radians.



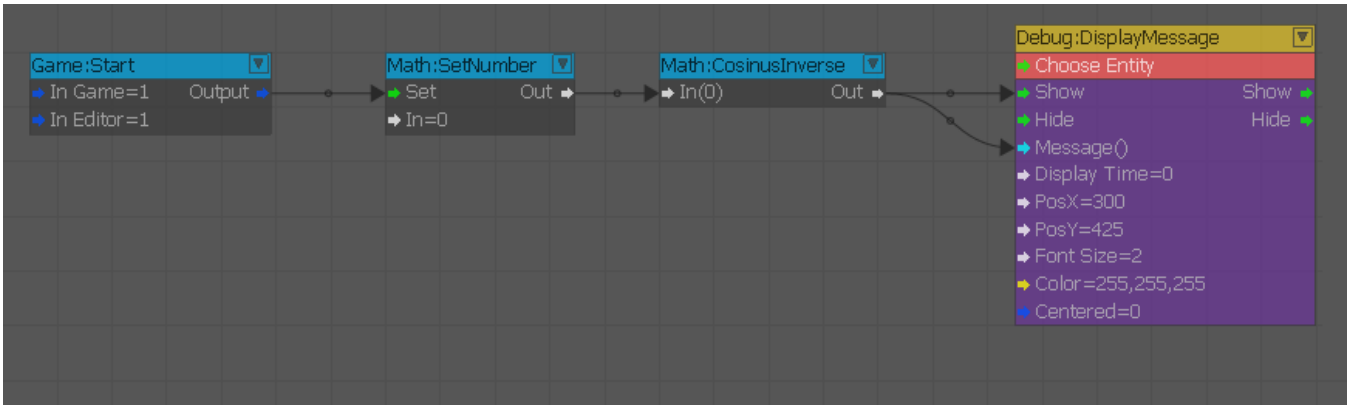
In the above example we are using a "Math:SetNumber" to input an angle (56 degrees) which passes through the "Math:Cosinus" to display the result to the HUD.

(Answer = 0.559193 radians)

Input	Description
Float *"in"*	The input angle in degrees to be calculated by the Cosinus function
Output	**Description**
Float *"out"*	Outputs the result in radians

Math:CosinusInverse

This node will take the input in radians and output the result in degrees.



In the above example we are using a "Math:SetNumber" to input the radian which passes through the "Math:CosinusInverse" to display the result to the HUD.

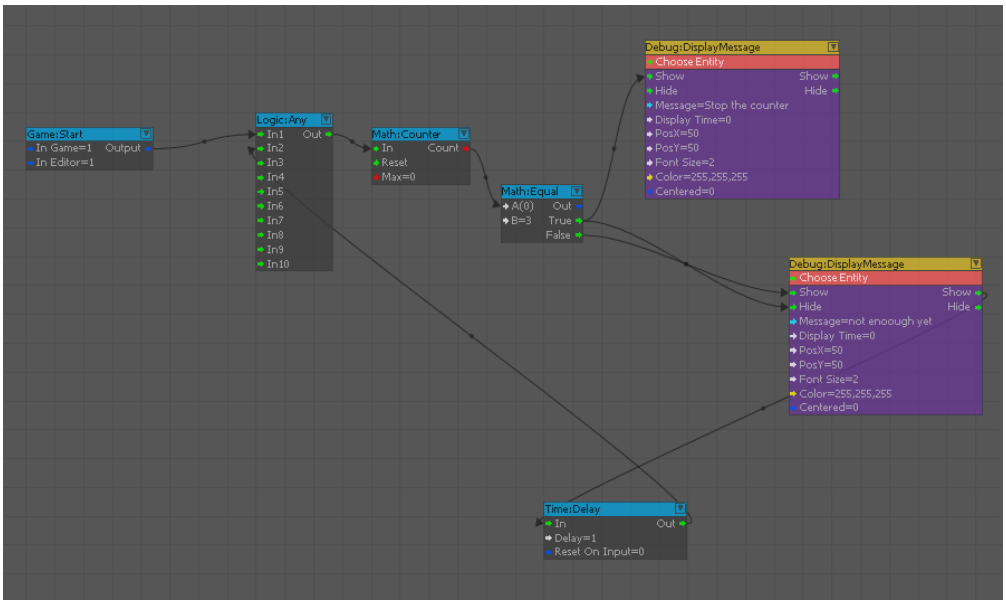
(Answer = 56 degrees)

Input	Description
Float *"in"*	The input in radians to be calculated by the CosinusInverse function
Output	**Description**
Float *"out"*	Outputs the result in degrees

Math:Counter

Every time this node receives an input, it will increase the number by 1, then forward it to the output.

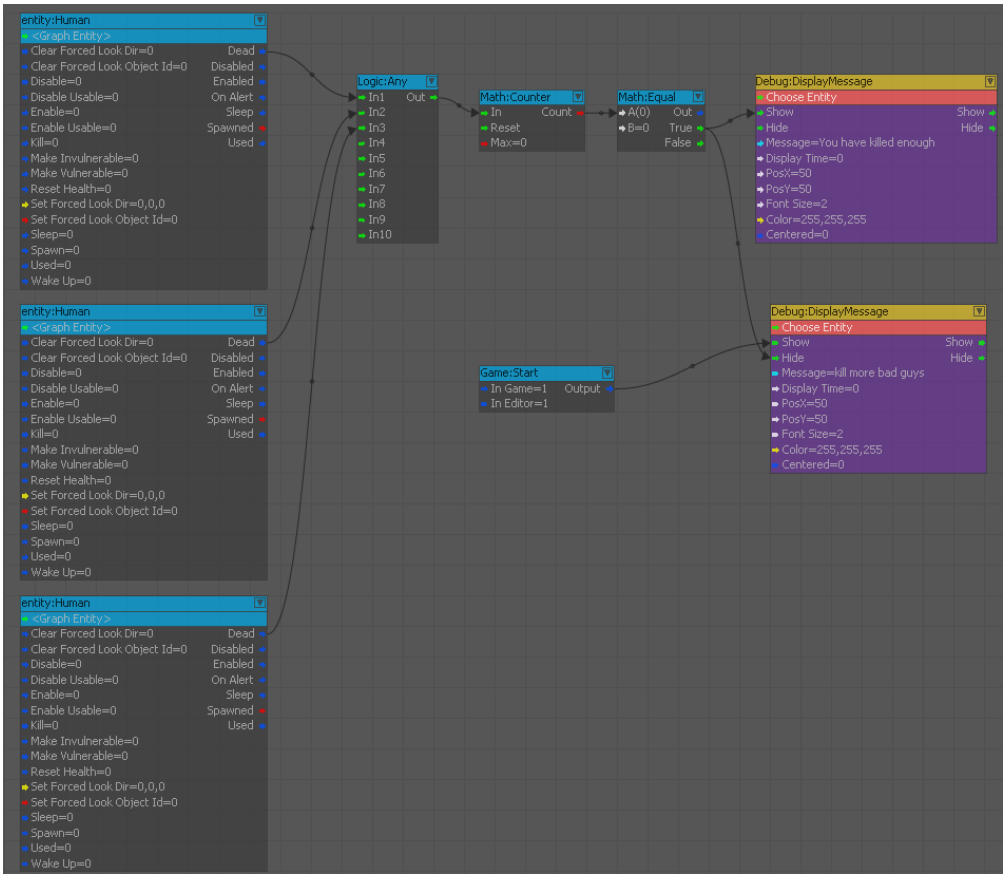
When the internal number reaches the specified number set in the "MAX" option, it will reset the counter back to zero.



Time based counter

As the signal passes through the flowgraph it increases the counter by 1, then it reaches the first Debug:DisplayMessage, then feed the signal back into the logic any after a 1 second delay.

Then it goes around again, increasing the counter by 1 each time. Once the count is equal to 3 (specified in the Math:Equal) it will hide the first HUD message then Show the second.



Bodycount counter

In this example, when any one of the human AI is killed it forwards a signal through the "Logic:Any" to the "Math:Counter". This will increase the counter by 1 each time it receives a signal. When all 3 grunts are dead, The "Math:Equal" becomes true and the HUD message is updated.

Note: In both examples, the MAX in the "Math:Counter" is set higher than the "Math:Equal". This is because if the counter reaches its set MAX it will be reset to zero and the "Math:Equal" will never become true.

Input	Description
Any *"in"*	
Any *"reset"*	To force the counter back to zero instead of letting the MAX value do it
Integer *"max"*	When MAX reached, sets the counter to zero
Output	**Description**
Integer *"count"*	Outputs the result

Math:DirToAngles

Used to convert the input vector direction to an angle.

Inputs

Port	Type	Description
Dir	Vec3	Vector direction
Roll	Float	Roll input

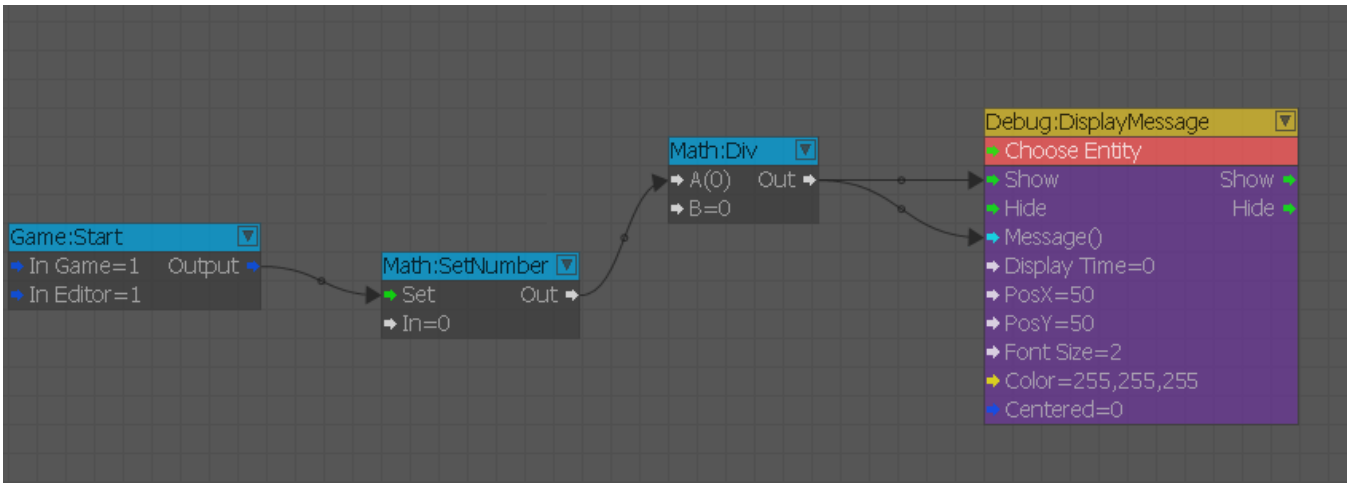
Outputs

Port	Type	Description
Angles	Vec3	Converts the direction to an angle in degrees

Math:Div

This is a simple operation of dividing Input A by Input B, then outputting the result. You can directly set a number into the node for A or B, or input a value from somewhere else. (Both are shown in the picture).

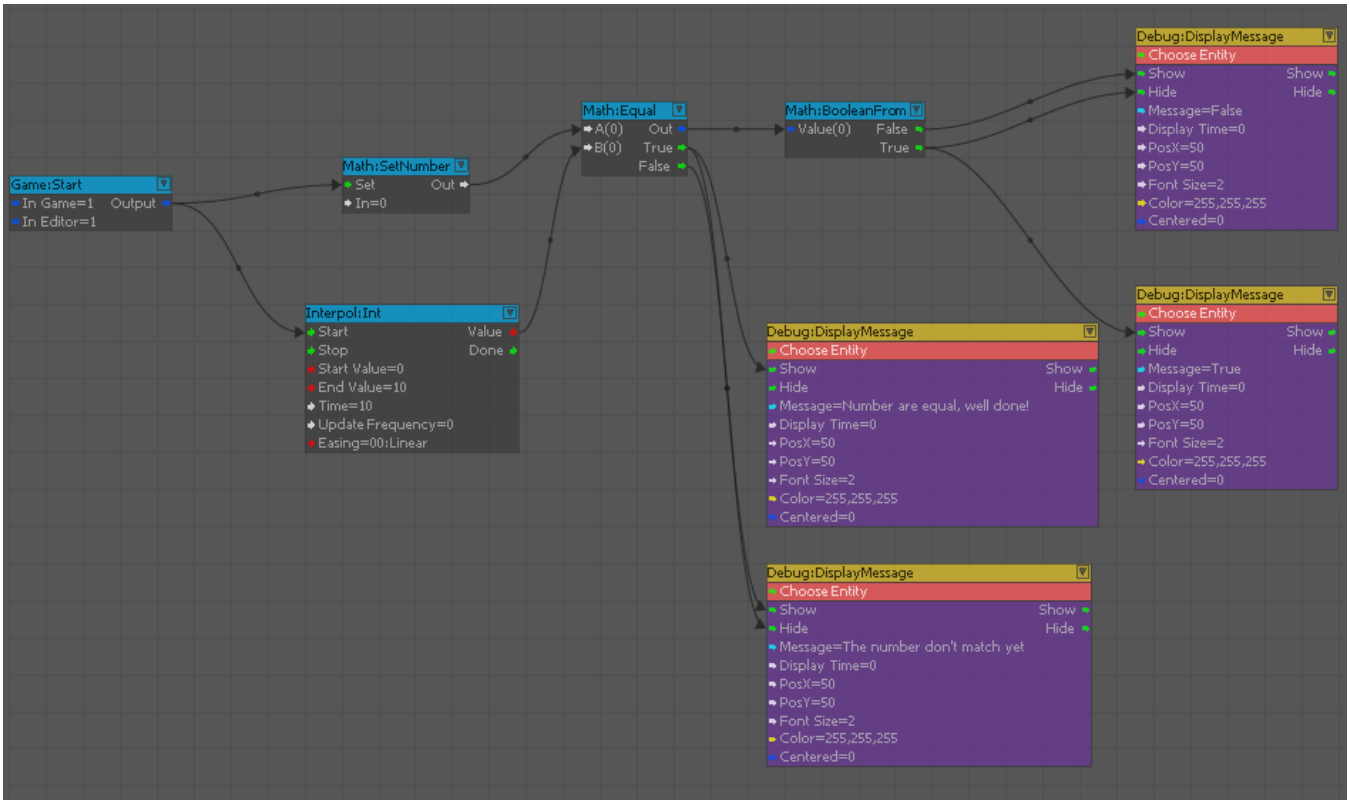
In the following example, we have a flowgraph to output the result of the "Math:Div" to the HUD.



Input	Description
Float *"A"*	The first number to be divided
Float *"B"*	The second number doing the divide
Output	**Description**
Float *"out"*	Outputs the result

Math:Equal

The "Math:Equal" node tests to see if input B is equal to input A. This will then output the answer in boolean form.



In the above example, we set a number to 10 (A), and also interpolate another number from 0 - 10 (B). These both feed into the "Math:Equal" node.

Upon start, they are not equal so the signal is passed to the false port. Until (B) has reached 10, then (B) is equal to (A) and it can the pass the signal to the true output port.

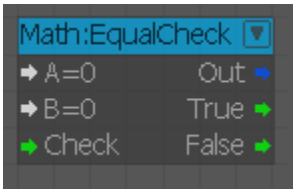
There is also an (OUT) port on the "Math:Equal" node. This is a boolean output. In the example above, the OUT acts the same as the true / false ports, but is combined into one output.

This is feed into a "Math:FromBoolean" node which then splits it into a true / false output.

Input	Description
Float *"A"*	The first number to test
Float *"B"*	The second number to test against the first
Output	**Description**
Boolean *"Out"*	Outputs the result as a boolean
Any *"True"*	Signal only passes to this output when A and B match
Any *"False"*	Signal goes to this output if A and B do not match

Math:EqualCheck

[Out] is true when [A]==[B], false otherwise. The check is only performed when the 'Check' input is triggered



Math:Floor

Used to output the floor of the input.

Inputs

Port	Type	Description
A	Float	Input

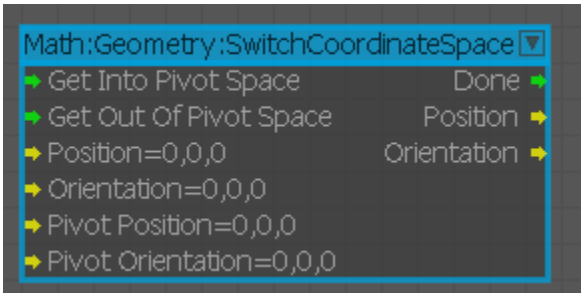
Outputs

Port	Type	Description
Out	Float	Floored input

Math:Geometry

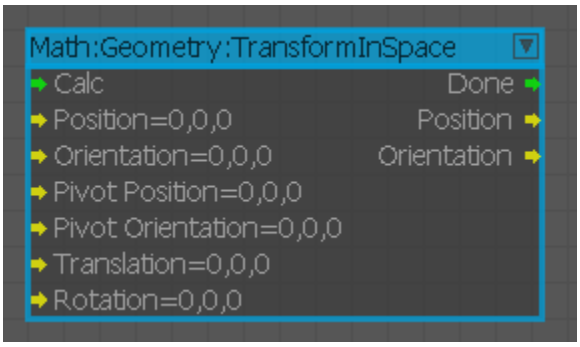
SwitchCoordinateSpace

Used for conversion between spaces (coordinate systems).



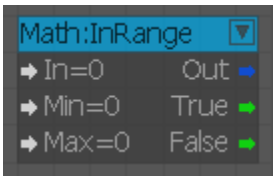
TransformSpace

Performs a transformation inside the space of a pivot.



Math:InRange

Used to check if the input is within the Min and Max value range.



Inputs

Port	Type	Description
In	Float	Input
Min	Float	Minimum value of the range
Max	Float	maximum value of the range

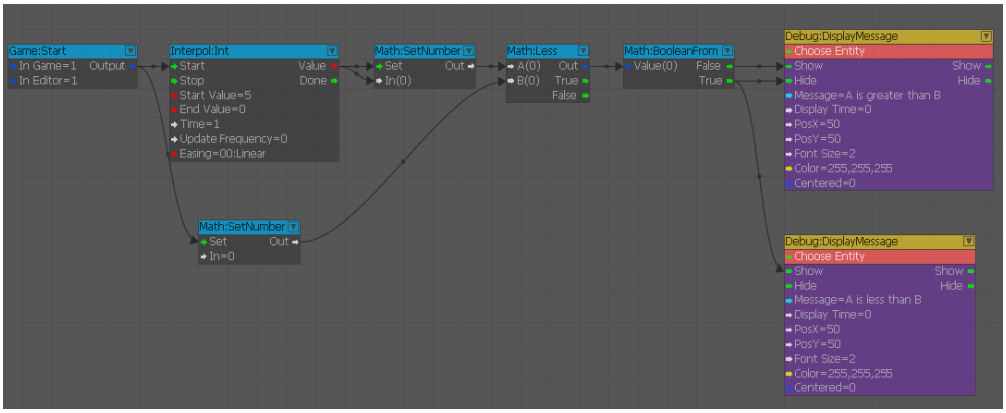
Outputs

Port	Type	Description
Out	Boolean	True if the input is within the range

True	Any	Triggered if the input is within the range
False	Any	Triggered if the input is outside of the range

Math:Less

This node is a simple calculation of, is (A) less than (B)?

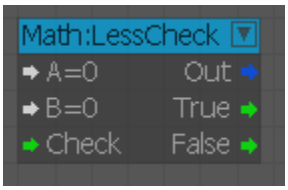


In the above example, at the start the value (A) is higher than value (B). As the seconds tick down via the "Interpol:Int", once (B) is at zero the "Math:Less" becomes true and the HUD message is updated.

Input	Description
Float *"A"*	The number doing the testing
Float *"B"*	The number to test against
Output	**Description**
Boolean *"Out"*	
Any *"False"*	Outputs the result if False (0)
Any *"True"*	Outputs the result if True (1)

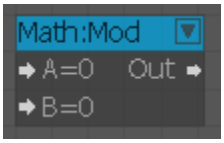
Math:LessCheck

[Out] is true when [A] < [B], false otherwise. The check is only performed when the 'Check' input is triggered



Math:Mod

Used to calculate the modulus of the two inputs.



Inputs

Port	Type	Description
A	Float	First operand
B	Float	Second operand

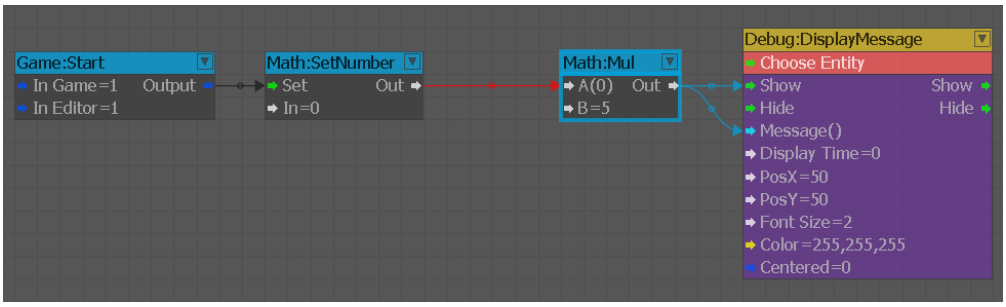
Outputs

Port	Type	Description
Out	Float	Modulus of the two inputs

Math:Mul

This is a simple operation of multiplying Input A by Input B, then outputting the result. You can directly set a number into the node for A or B, or input a value from somewhere else (both are show in the picture).

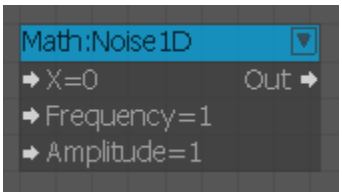
In the following example, we have a flowgraph to output the result of the "Math:Mul" to the HUD.



Input	Description
Float *"A"*	The first number to be multiplied
Float *"B"*	The second number multiplying the first
Output	**Description**
Float *"out"*	Outputs the result

Math:Noise1D

Used to multiply the scalar input by the frequency and amplitude.



Inputs

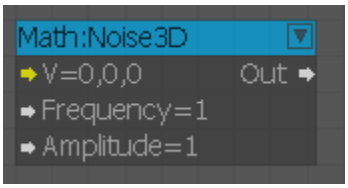
Port	Type	Description
X	Float	Scalar Input value to sample noise at
Frequency	Float	Frequency
Amplitude	Float	Amplitude

Outputs

Port	Type	Description
Out	Float	Multiplication of X by Frequency and Amplitude values

Math:Noise3D

Used to multiple the vector input by the frequency and amplitude.



Inputs

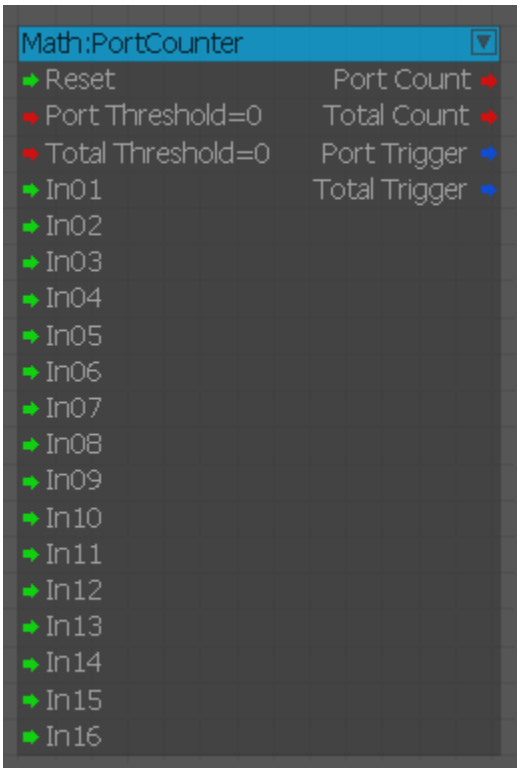
Port	Type	Description
V	Vec3	Vector input value to sample noise at
Frequency	Float	Frequency
Amplitude	Float	Amplitude

Outputs

Port	Type	Description
Out	Float	Multiplication of V by Frequency and Amplitude values

Math:PortCounter

Used to count the number of activated inputs.



Inputs

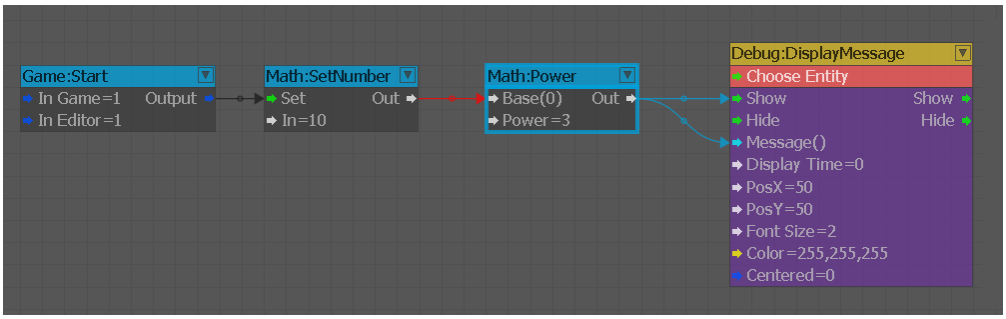
Port	Type	Description
Reset	Any	Resets PortCount and TotalCount
PortThreshold	Integer	PortCount threshold value
TotalThreshold	Integer	TotalCount threshold value
In00 - In15	Any	Inputs

Outputs

Port	Type	Description
PortCount	Integer	Number of ports that have been set
TotalCount	Integer	Sum of all times any of the input ports have been set
PortTrigger	Boolean	Triggered when PortCount reaches PortThreshold
TotalTrigger	Boolean	Triggered when TotalCount reaches TotalThreshold

Math:Power

When you use this node in a flow graph, it will calculate the base by the number set in the power input.



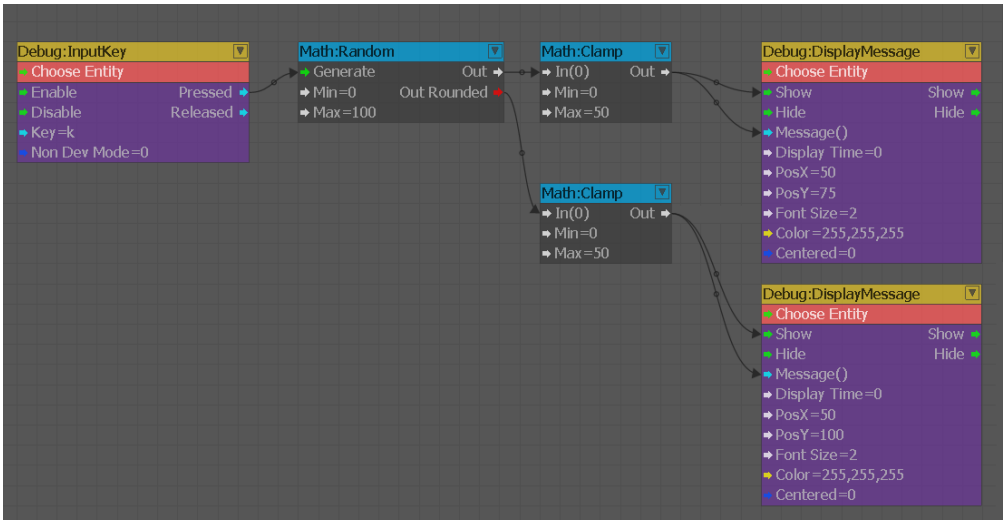
In the the above example, we set the number of the base to 10, then pass it through the "Math:Power" node with the power set to 3. This will output the result (1000) to the HUD.

Input	Description
Float *"base"*	The input number to be calculated
Float *"power"*	The power value to calculate on the base
Output	**Description**
Float *"out"*	Outputs the result

Math:Random

This node will upon receiving an input generates a random number between your specified MIN & MAX settings. It has 2 output ports, out and outRounded.

Depending on which one you select it will output the number as a float or an integer.



In the above example the upon pressing the key "K" it will make the "Math:Random" node generate a number between 0 and 100 (ignoring the clamps, the two hud messages will output the number to the screen).

The top will display the float and the bottom will display the same number generated but rounded up or down as an integer.

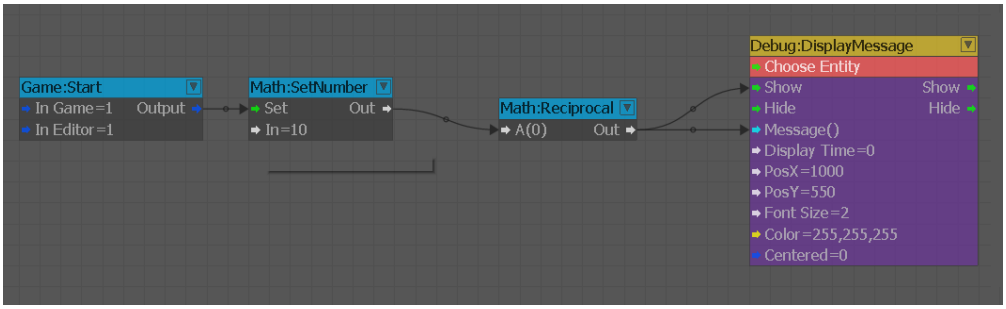
0.0 -> 0.49 = rounded down
0.5 -> 1.0 = rounded up

Input	Description
Any *"Generate"*	Upon activating will generate the random number
Float *"min"*	Minimum number specified of the range
Float *"max"*	Maximum number specified of the range

Output	**Description**
Float *"out"*	Outputs the result as a Float
Integer *"outRounded"*	Outputs the result as an Integer

Math:Reciprocal

This node will calculate the reciprocal of the input number. (To the power of -1).

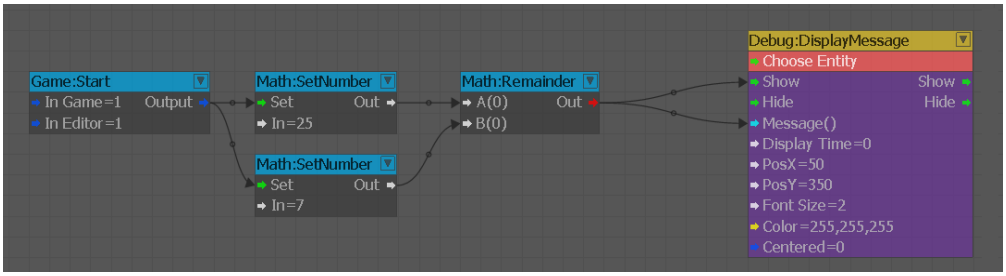


In the above example we set an input number of 10 and pass it through the "Math:reciprocal" which will then output the result (0.1) to the HUD.

Input	**Description**
Float *"A"*	The input to be calculated
Output	**Description**
Float *"out"*	Outputs the result

Math:Remainder

This node will calculate the quantity left over after dividing the two inputs into each other.

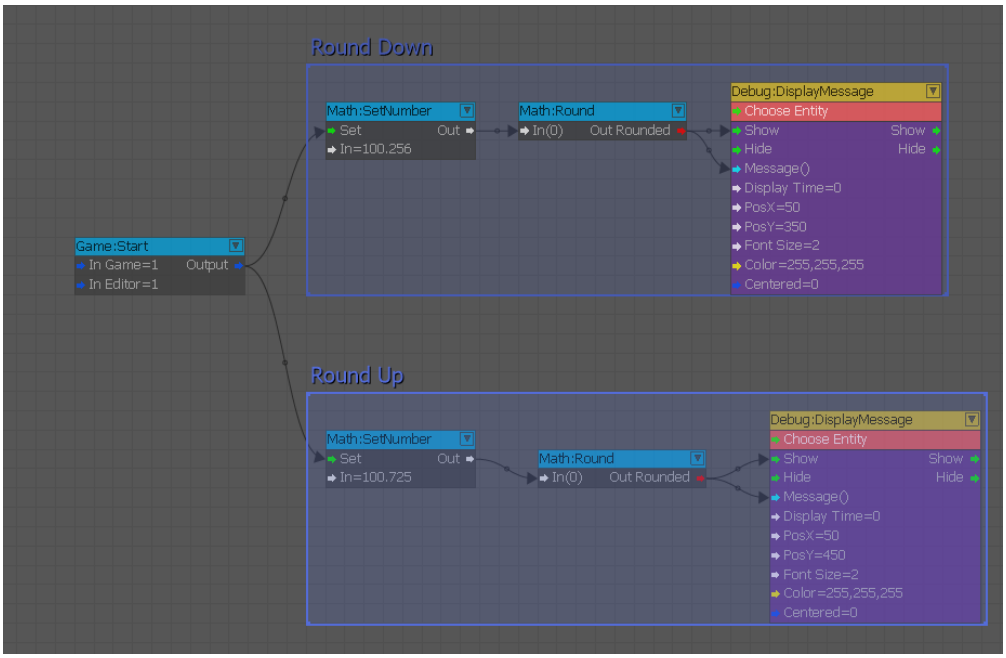


In the above example, we have 2 integers going into the inputs 25 & 7. So 25/7 = 3, remainder 4.

Input	**Description**
Float *"A"*	The first number to do the calculation on
Float *"B"*	The second number to divide against the first
Output	**Description**
Integer *"out"*	Outputs the result of the remainder

Math:Round

This node will round up or down the float input number depending on the value after the decimal point.



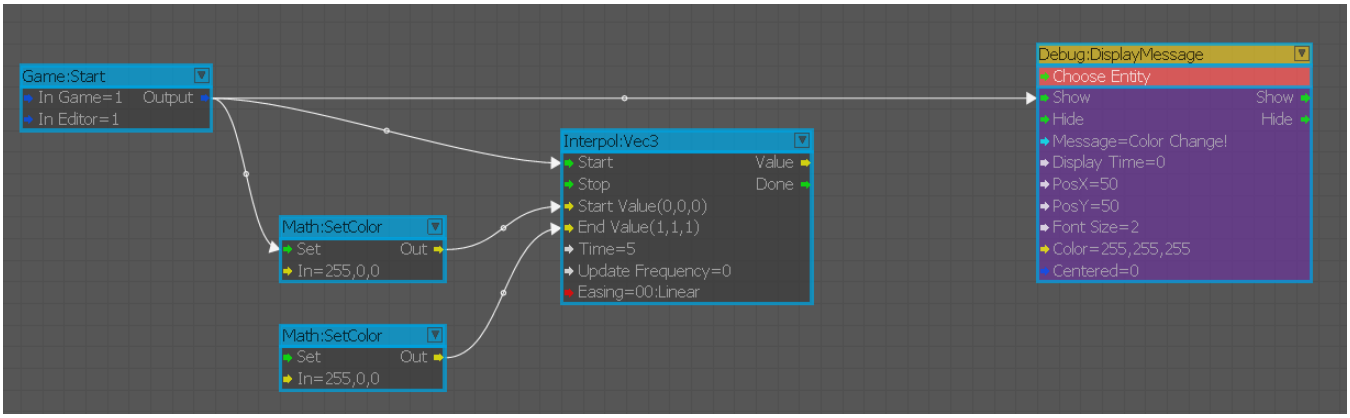
In the above example, we are rounding down the top path and rounding up the bottom path. The system it follows is...

0.0 -> 0.49 = rounded down
0.5 -> 1.0 = rounded up

Input	Description
Float *"In"*	The input number to be calculated
Output	**Description**
Integer *"outRounded"*	Outputs the result either up or down

Math:SetColor

This node sets the color in RGB format.



In the above example, we have the message "Color change!" displayed on the HUD.

We are using the node "Interpol:Vec3" to change the color of the text from Red (255,0,0) at the start, to Blue (0,0,255) over a 5 second period.

Input	Description
Any *"set"*	Activates the node

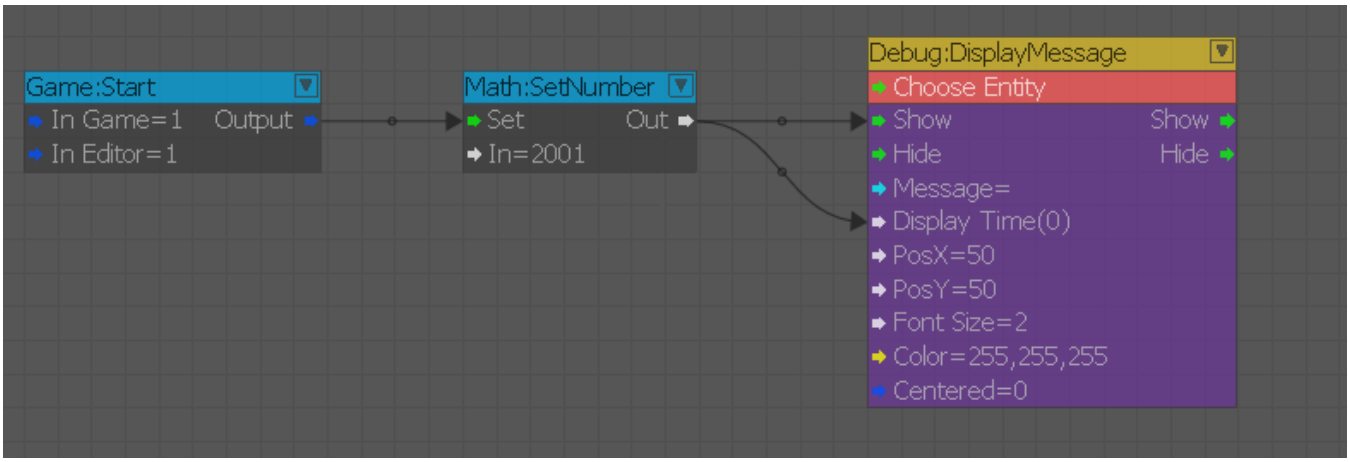
Vec3 *"in"*	Set the color in RGB format. (You can use the color picker to choose if you desire)
Output	**Description**
Vec3 *"out"*	Outputs the result

Math:SetInteger

Used to send an integer input value to the output when an event on the Set port is received.

Math:SetNumber

This node is a basic function to state a number specified within a flowgraph. This is usually used in conjunction with other nodes.

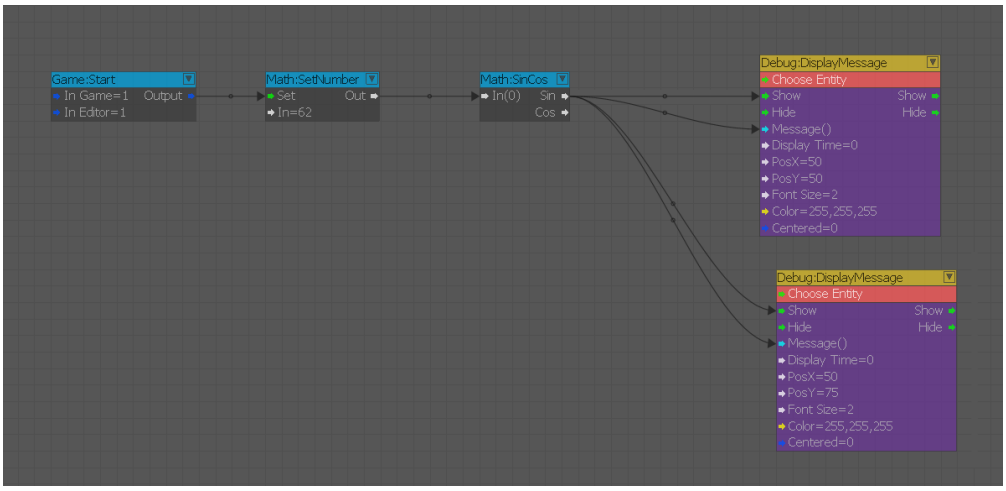


In the above example, we set the number 2001 to be displayed on the HUD.

Input	**Description**
Any *"set"*	Activates the node
Float *"in"*	Input the number you want to use
Output	**Description**
Float *"out"*	Outputs the result

Math:SinCos

This node is a combination of the "Math:Sinus" and the "Math:Cosinus" nodes. It will take the input in degrees, and then output to the two different ports in sinus & cosinus format.

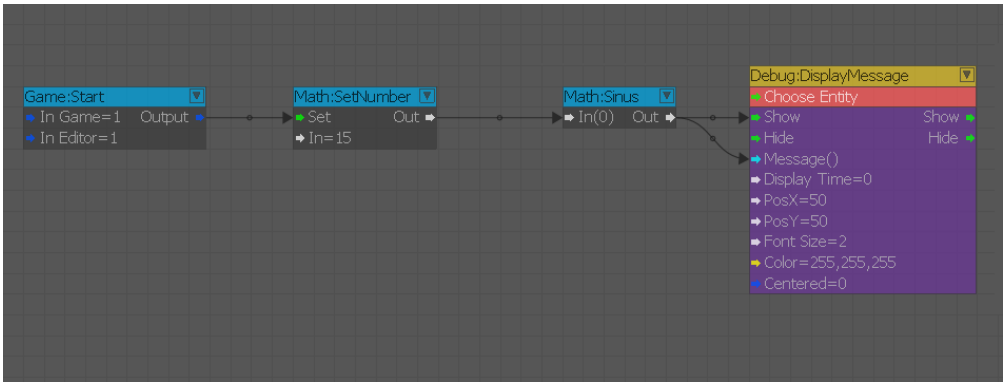


In the above example, the input in degrees (62) is fed into the "Math:SinCos" node and then outputs the results on separate ports sin & cos, which go to a different message block on the HUD.
(Answer = sin = -0.739181, cos = 0.673507)

Input	Description
Float *"in"*	The input angle in degrees to be calculated by the SinCos function
Output	**Description**
Float *"sin"*	Outputs the result in radians (sinus)
Float *"cos"*	Outputs the result in radians (cosinus)

Math:Sinus

This node will take the input of an angle in degrees and output the result in radians.



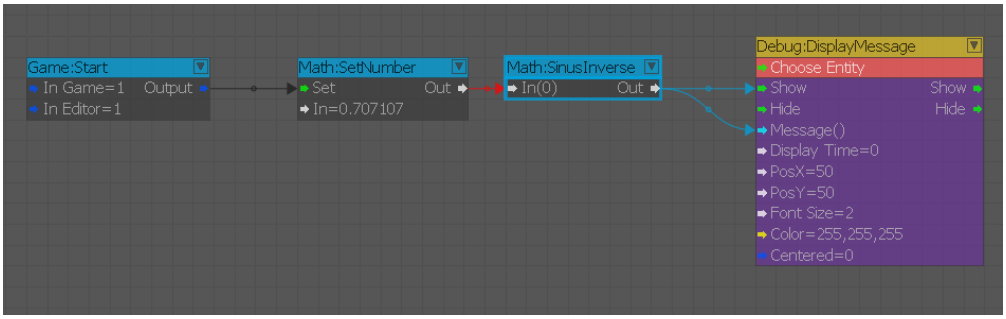
In the above example, we set the degrees to 45 and pass it through the "Math:Sinus" node to output the result in radians to the HUD.
(Answer = 0.707107)

Port Description

Input	Description
Float *"in"*	The input angle in degrees to be calculated by the Sinus function
Output	**Description**
Float *"out"*	Outputs the result in radians

Math:SinusInverse

This node will take the input in radians and convert them into degrees.



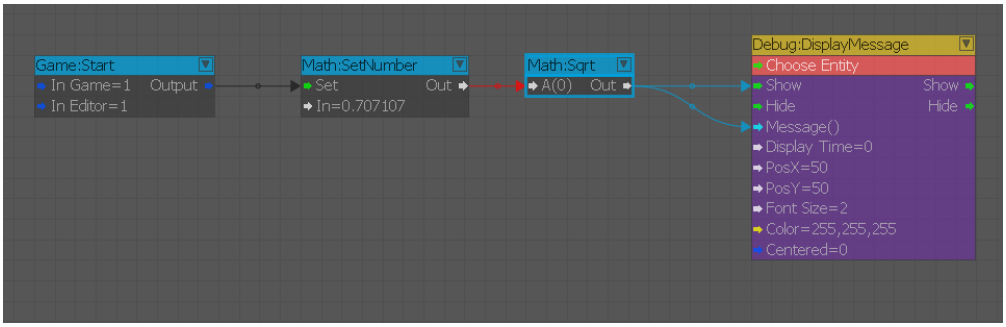
In the above example, we set the input in radians via the "Math:SetNumber", and passed it through the "Math:SinusInverse" node to get the result to the HUD in degrees.

(Answer = 45 Degrees)

Input	Description
Float *"in"*	The input angle in radians to be calculated by the SinusInverse function
Output	**Description**
Float *"out"*	Outputs the result in degrees

Math:Sqrt

This node will give you the square root of the input number.



In the above example, we state a number with the "Math:SetNumber" node which then passes through the "Math:Sqrt" node which displays the result on the HUD.

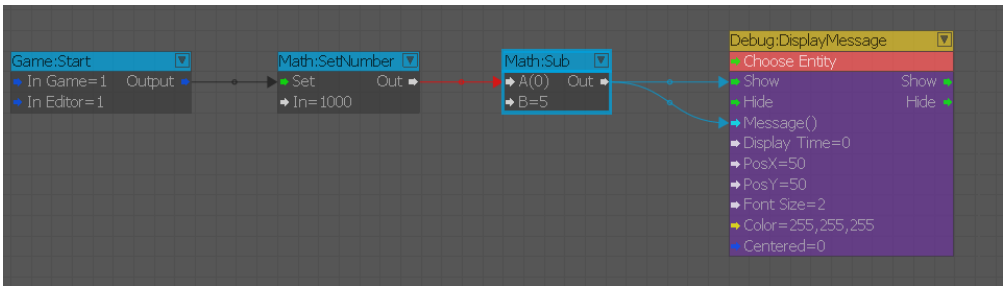
(Answer = 34.741905)

Input	Description
Float *"A"*	The input number to be calculated by the Square Root function
Output	**Description**
Float *"out"*	Outputs the result

Math:Sub

This is a simple operation of subtracting Input B from Input A, then outputting the result. You can directly set a number into the node for A or B, or input a value from somewhere else (both are shown in the picture).

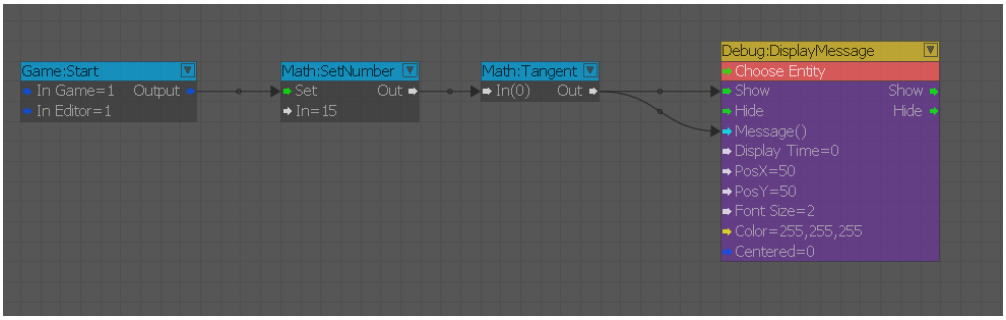
In the following example, we have made a flowgraph output the result of the "Math:Sub" to the HUD.



Input	Description
Float *"A"*	The first number to be subtracted
Float *"B"*	The second number to subtract from the first
Output	**Description**
Float *"out"*	Outputs the result

Math:Tangent

This node will take the input of degrees and output the result into radians.



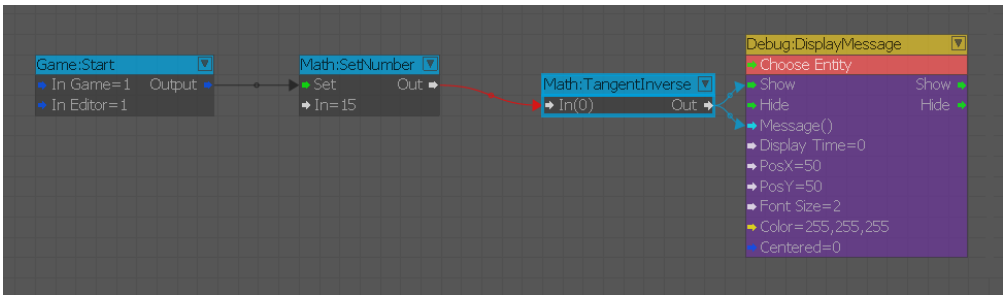
In the example above, we set the input (in degrees) to 15 via the "Math:SetNumber", then passed it through the "Math:Tangent" node and output the result to the HUD.

(Answer = 0.267949 radians)

Input	Description
Float *"in"*	The input angle in degrees to be calculated by the Tangent function
Output	**Description**
Float *"out"*	Outputs the result in radians

Math:TangentInverse

This node will take the input of radians and output the result in degrees.



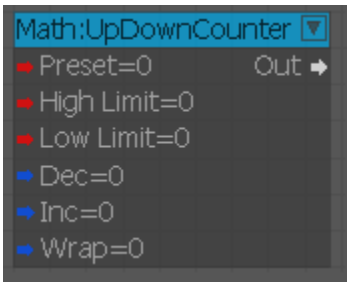
In the above example, we have set the input in radians via the "Math:SetNumber" and it passes through the "Math:TangentInverse" to output the result in degrees to the HUD.

(Answer = 15 degrees)

Input	Description
Float *"in"*	The input in radians to be calculated by the TangentInverse function
Output	**Description**
Float *"out"*	Outputs the result in degrees

Math:UpDownCounter

Used to output an up or down counter.



Inputs

Port	Type	Description
Preset	Integer	Preset input value
High Limit	Integer	Maximum counter limit
Low Limit	Integer	Minimum counter limit
Dec	Boolean	Decrements the count
Inc	Boolean	Increments the count
Wrap	Boolean	If true, the counter will wrap

Outputs

Port	Type	Description
Out	Float	Current count

Math:Wrap

Wraps Value around the interval defined by Min and Max.