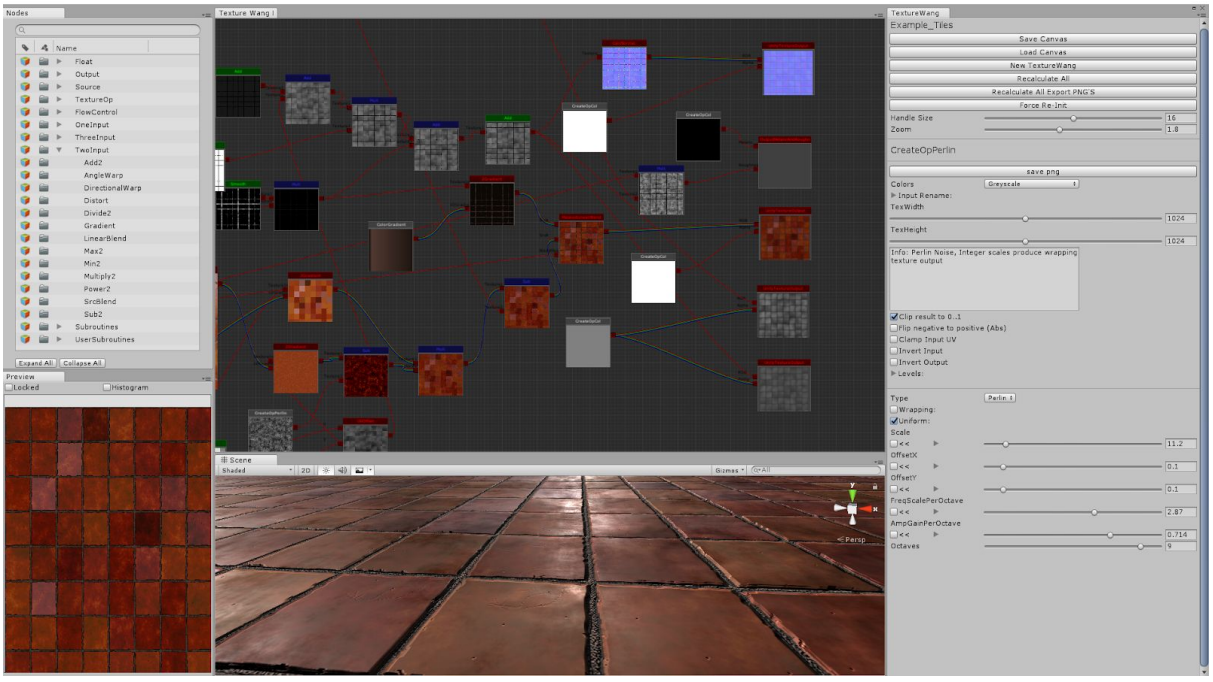
TextureWang



Why is it called textureWang?

https://www.youtube.com/watch?v=0obMRztklqU

With that out of the way..

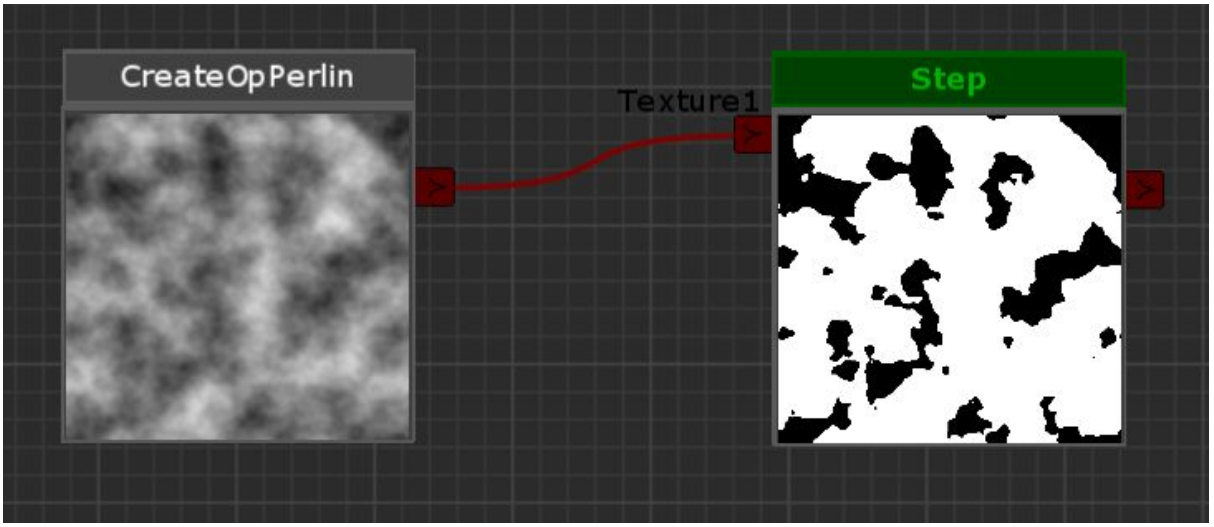
What is Texture Wang.

TextureWang is a procedural node based texture generator, it can create albedo, normal,height, well any type of texture for any purpose.

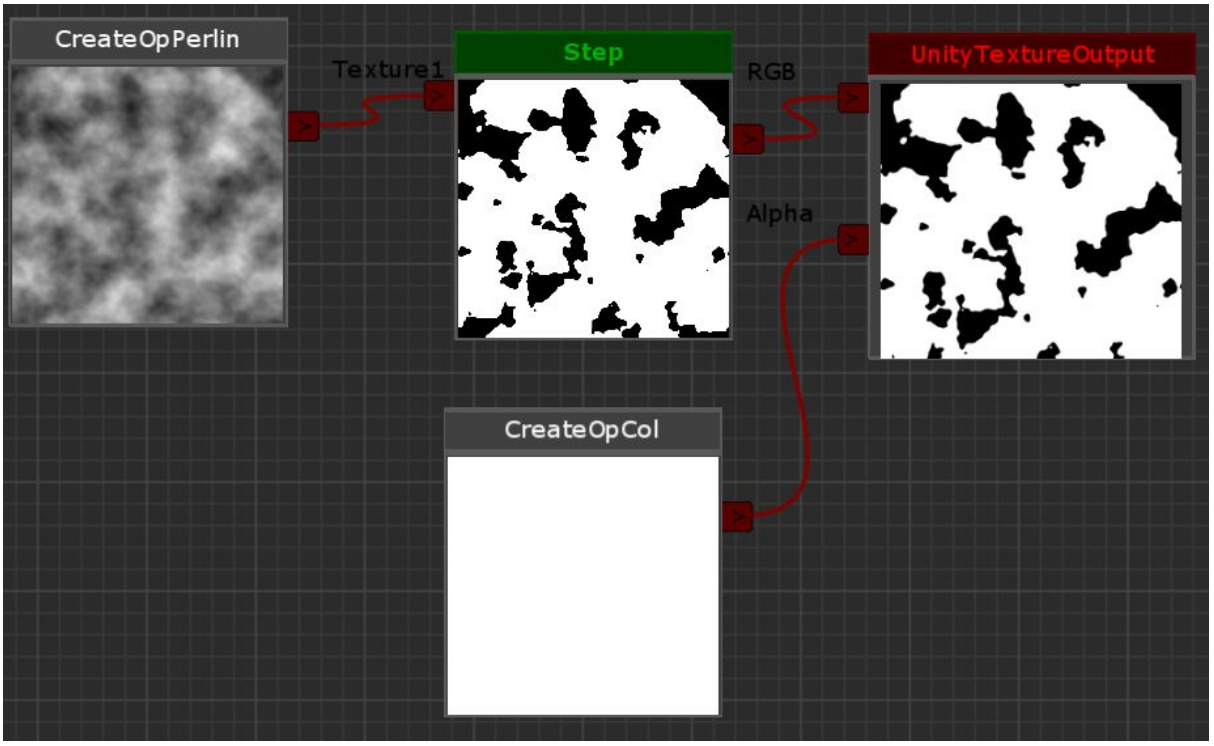
The user creates these textures by chaining together a sequence of nodes that apply various effects to every pixel of a texture.

EG perlin noise creates a cloudy sort of noise.

Connecting the output of one node into the input of another allows you to chain operations to generate a more complex effect.



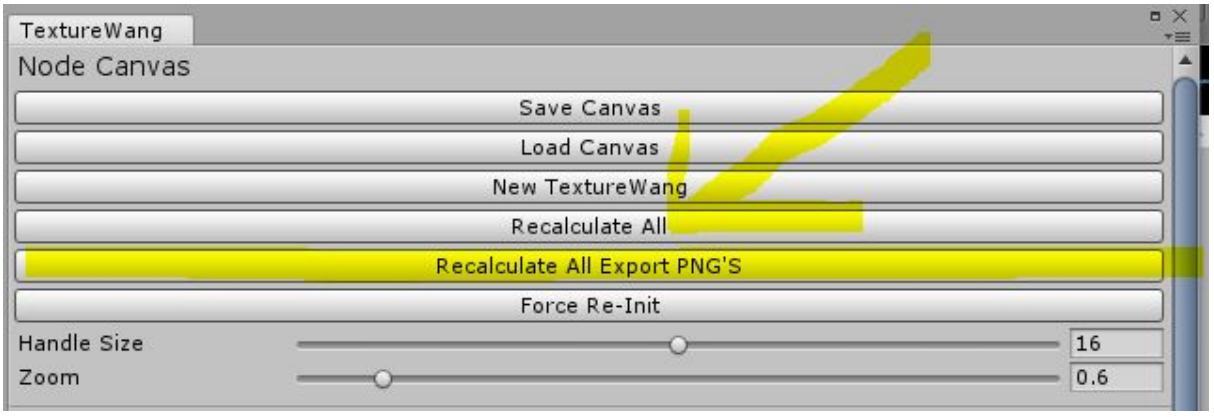
Using the UnityTextureOuput Node allows you to choose an existing texture to overwrite.



As you change options and create and edit nodes the output texture is updated in real time, if that texture is part of a material on an object in the scene view then you can see the changes happen in real time with your unity material on real game objects.

A full material would have multiple UnityTextureOutput nodes, for albedo,height,normals, metal/smoothness, occlusion…

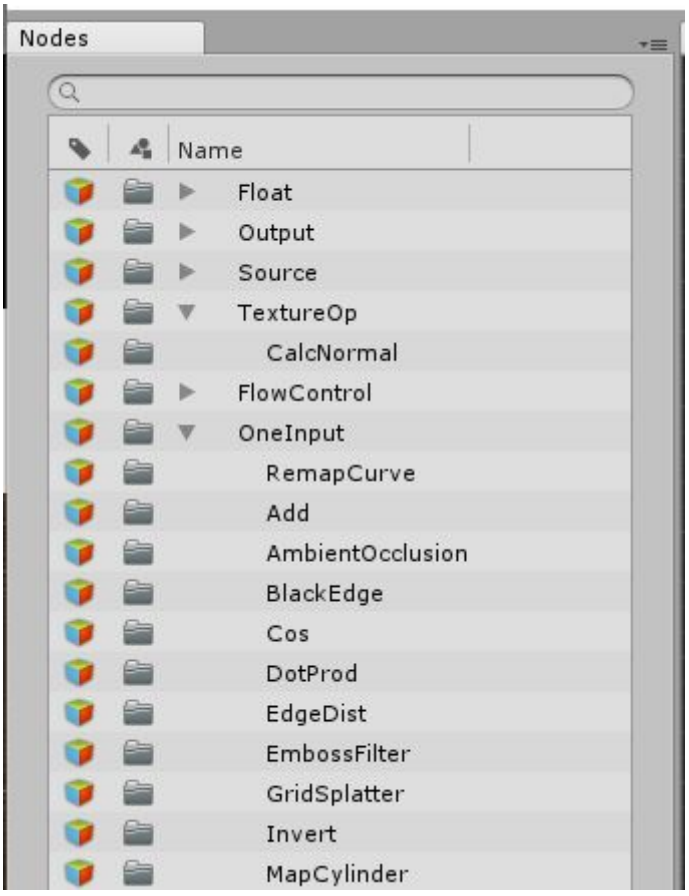
When you are happy with your material you can hit the recalculate All Export to PNG button and this will write your changes permanently to disk in the selected PNG's.



Node Types

There are many node types including perlin noise, Step,Gradient, Splatter, Grid Splatter, Edge Dist, Smooth, Transform, LinearBlend, Multiple,Add,Sub,Pow,Cos,Sharpen,Min,Max and many more.

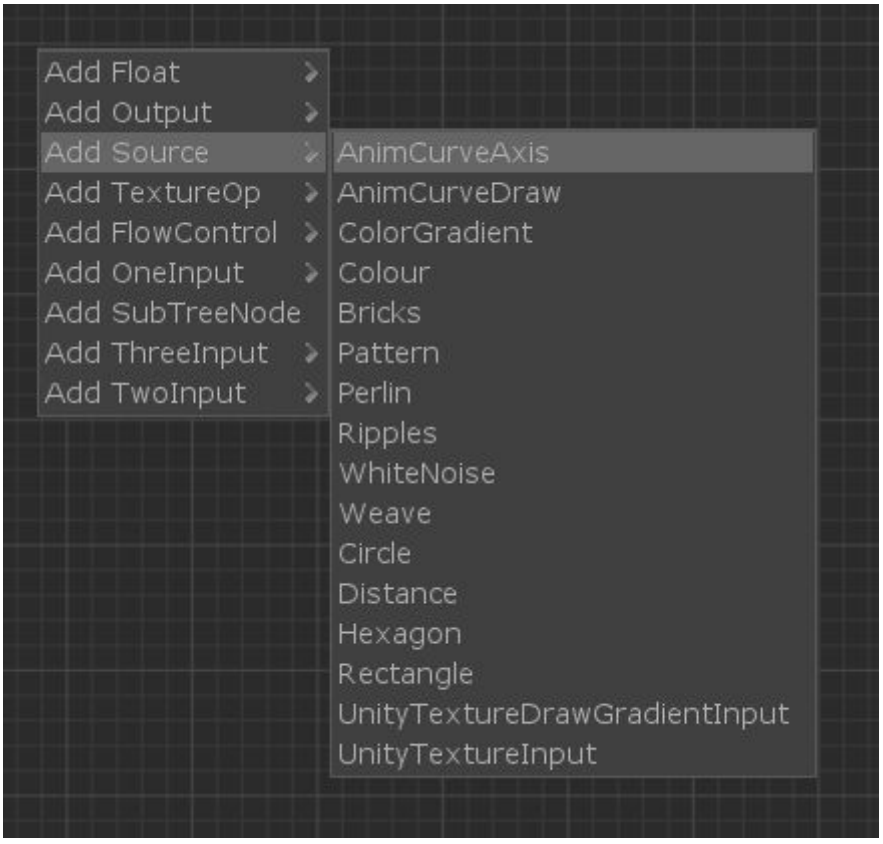
They are available to place by using the Nodes Window.



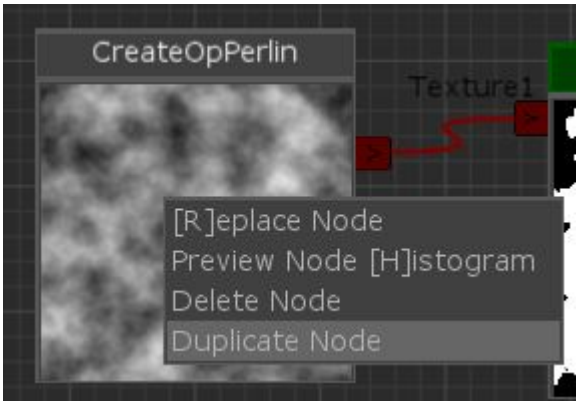
You select a node and drag it into the Node canvas window.

The Nodes are grouped by Number of Inputs, or special characteristics, such as being output related, float based rather than texture, or a texture source i.e they have no input textures such as circle, or perlin noise.

Right clicking in the node canvas brings up an alternative node menu, that allows you to place nodes where you right clicked



Right clicking a node gives the option to duplicate, delete, replace or open a large window allowing you pan and zoom the texture or see a histogram.



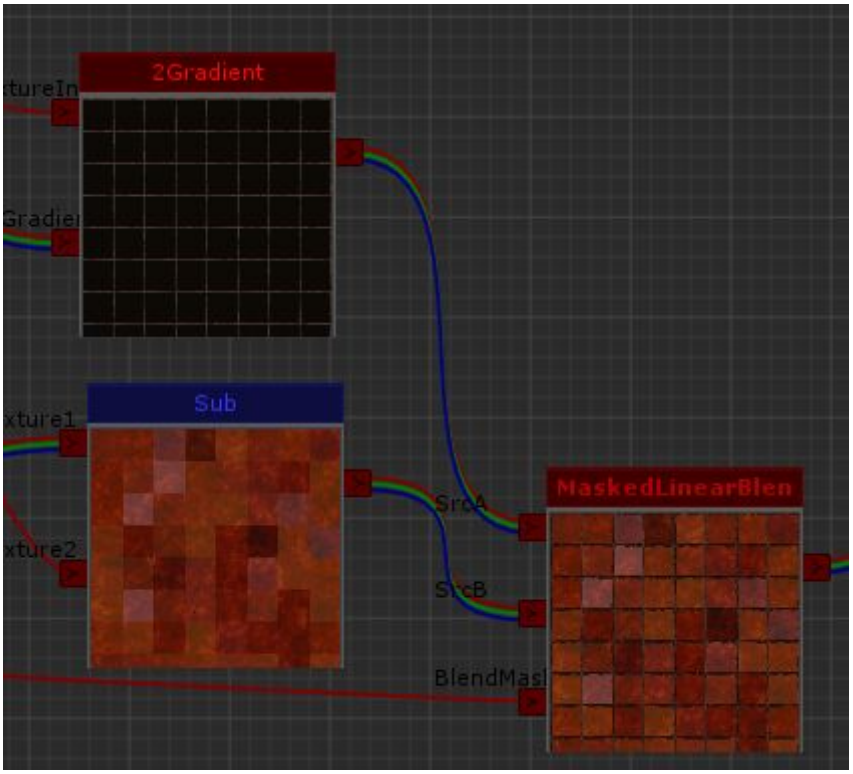
Controls

Hold middle mouse to pan the node canvas

Left Click on a node in the canvas, and hold and drag to move the node, hold alt to drag all child nodes.

Left Click on a node's output (red square on right with > symbol) and hold and drag to connect it onto another nodes input (red square on left with > symbol).

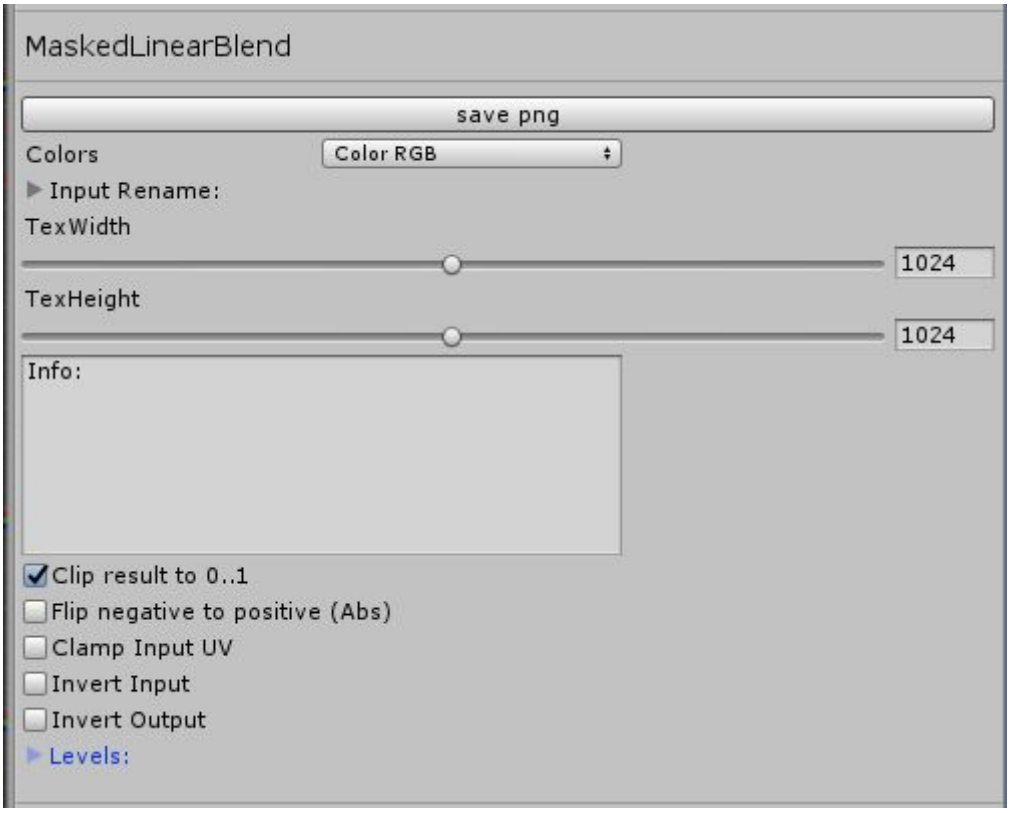
Hover mouse over a node and press D to duplicate or D to delete



The above image shows the input and output nodes including the 3 input MaskedLinearBlend node, and that two inputs are full RGB and one is only supplying info in the red channel which is treated as a greyscale.

Nodes In Detail

Selecting a node brings up its individual options, all nodes have the below options



Colors:

Lets you choose how many channles of information is stored in the texture 1 (greyscale) or 3 (color RGB)

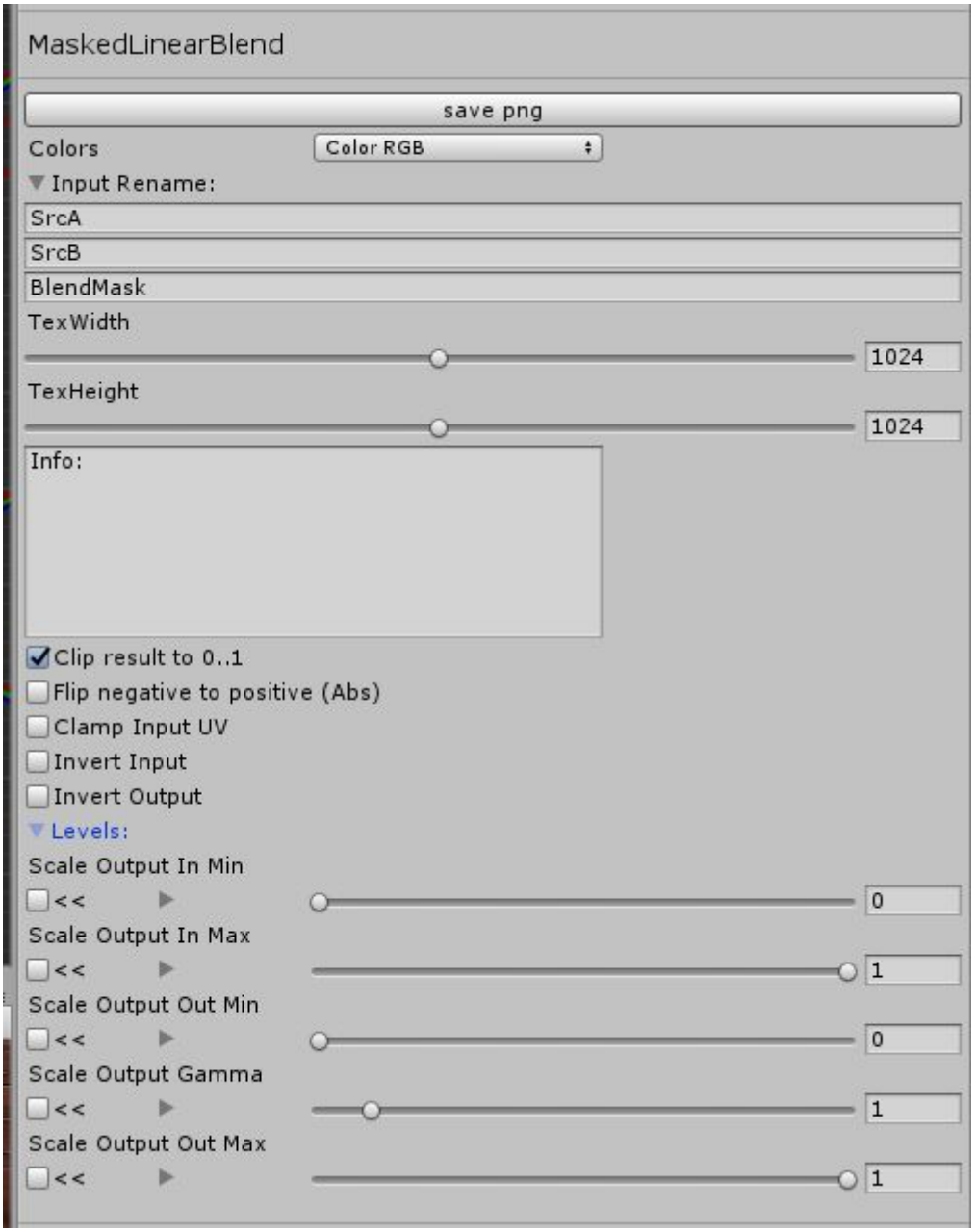
Clicking the small triangles opens up more options (see below)

Input Rename:

As it suggests you can rename the inputs of a node to something more specific to how you are using the node, for example if you were making a brick texture you might rename a nodes input to mortar color.

Levels:

This allows the output of every node to be carefully remapped , effectively the input range say 0 to 0.5 is remapped to the output range say 0..1 effectively causing all pixels to appear twice as bright. Output Gamma allows a non linear scale to be applied to the brightness.



Extra options

TexWidth/Height

Allows you to change the size of the texture, by default all textures are 1024x1024 and are full float per channel. All textures live on the GPU and use 4 MB for color and 1 MB for greyscale.

Clip Result 0..1

Texture node operations can result in values outside the range 0 to 1, only the range 0 to 1 is visible on the screen, clipping the results (on by default) will ensure the pixel values stay in that range. Sometimes it can be useful to remove the feature to have say negative values that are used in a later node, but be aware that you won't be able to see those values, but their effects can be useful as part of the math ops later on.

Clamp Input UV

Some operations such as Transform can read from an input texture outside its normal UV range (0..1) clamping causes negative UV's to read from the 0'th pixels, and UV's greater that 1 to read from the far right/bottom texel. With this off the input texture will be treated as infinitely wrapping.

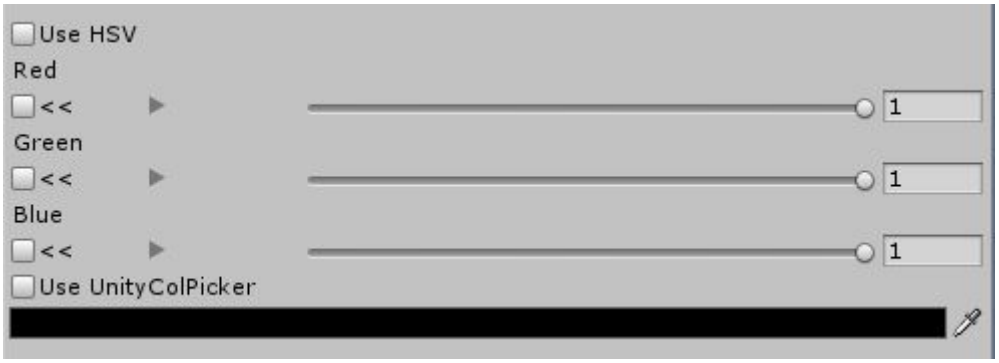
Invert Input

Inverts the data read from the input texture so white becomes black and black becomes white (math op 1-pixel color).

Invert Output

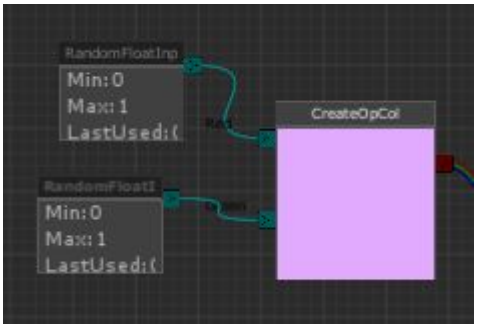
Inverts the output from this node so white becomes black and black becomes white (math op 1-pixel color).

Note the save PNG button is always available to write out the nodes current texture to an image file.



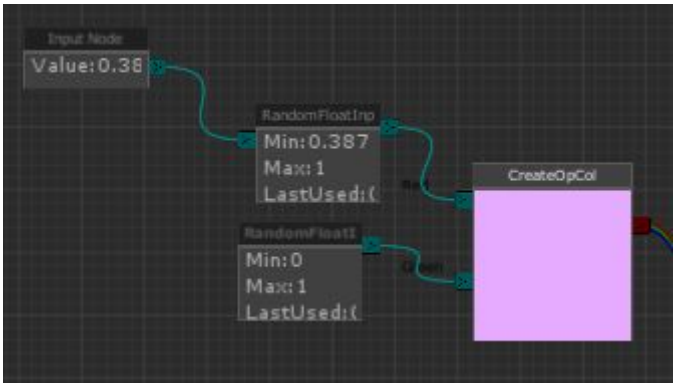
Nodes each have tailored inputs depending on what they do, the example above is for the CreateOpCol node which fills the entire texture with a single color. As seen above you can use HSV to change the meaning of the sliders or use the unity color picker.

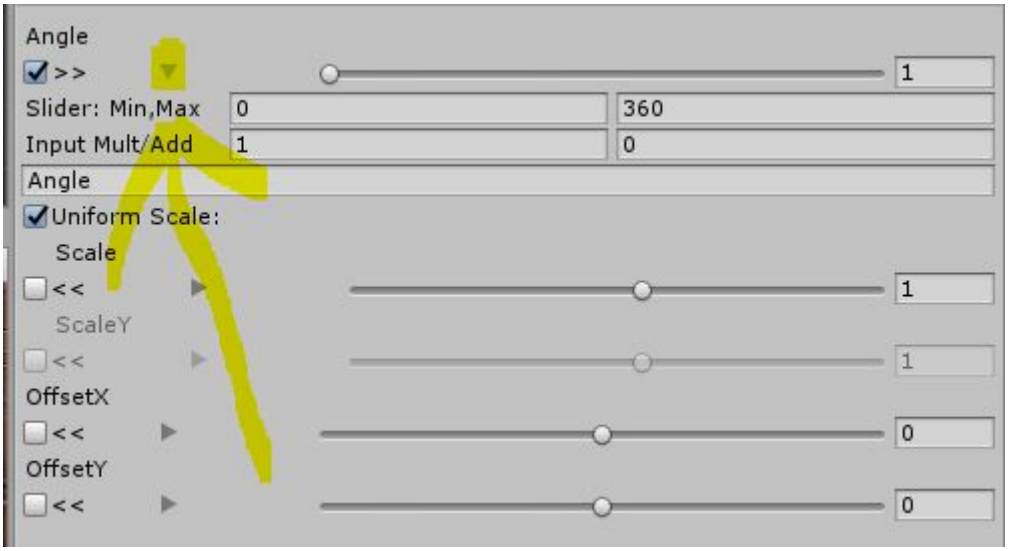
Note the box next to the '<<' under red Green and Blue, this allows any of these values to be taken instead from an input float.



With Red and Green input boxes ticked we can then create colours using random values, or chains of complex float nodes.

The RandFloatInput can even have its inputs for min and/or max range redirected to be taken from a float input





Clicking the arrow next to an input allows you to edit the range of the slider bar for example in the transform node the Angle option provides a slider that goes from 0 to 360 by default.

Any slider can have its default range changed, to allow easier editing in the required range for your use case.

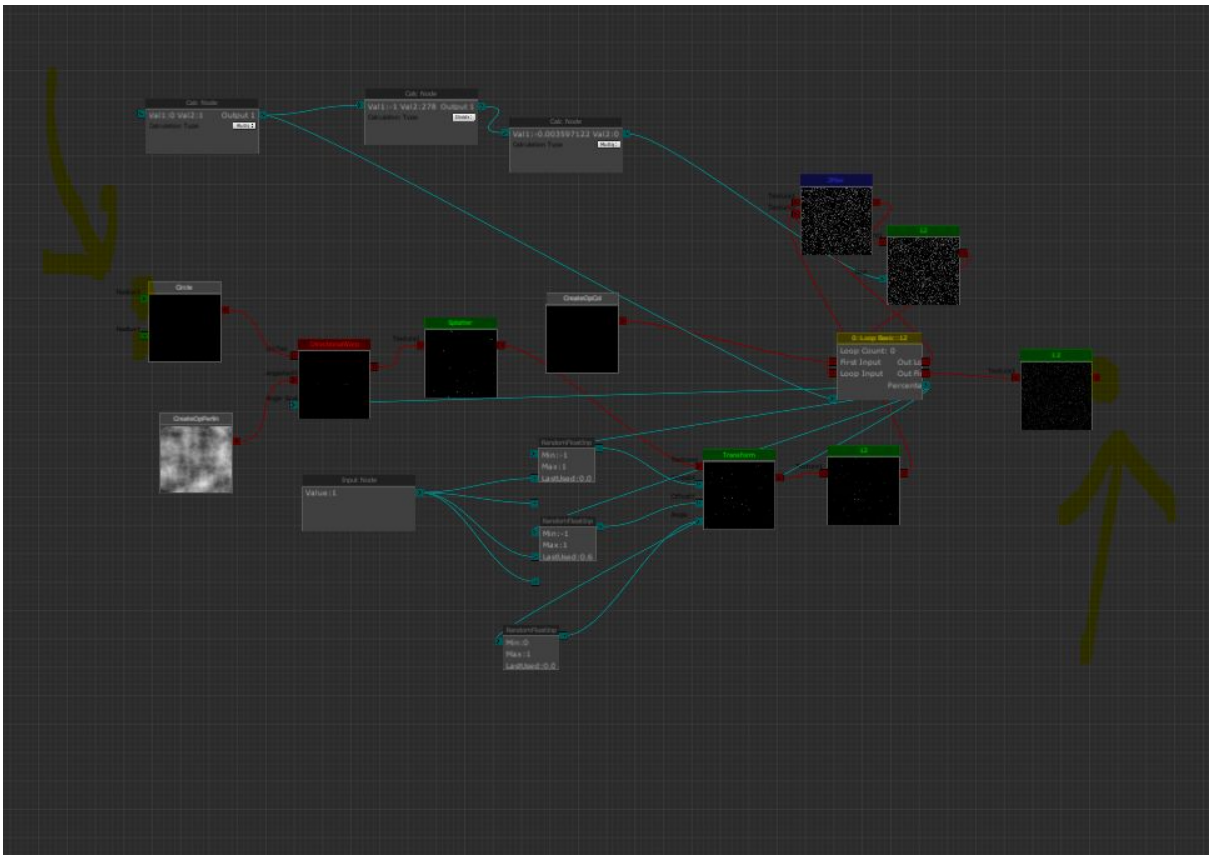
The Input Mult/Add options provide a built in scale and offset to be applied to any remapped float input, so if mult was set to 360 then a float Input in the range 0 to 1 would be automatically mapped to 0 to 360 or any other required range.

These are especially useful when creating subroutines with float values exposed to users of the subroutine.

The name of the input Angle is also an editable texture input so you can rename the input for users, again especially useful for subroutines.

Subroutines

The below shows the fibers sub routine



Any Collection of nodes can be saved as a subroutine, those saved in the userSubroutine folder will appear in the nodes list for easy drag and drop into the canvas.

And unconnected inputs will appear as input fields and any unconnected outputs will appear as outputs on the SubRoutine node that appears in the canvas.
As seen below.