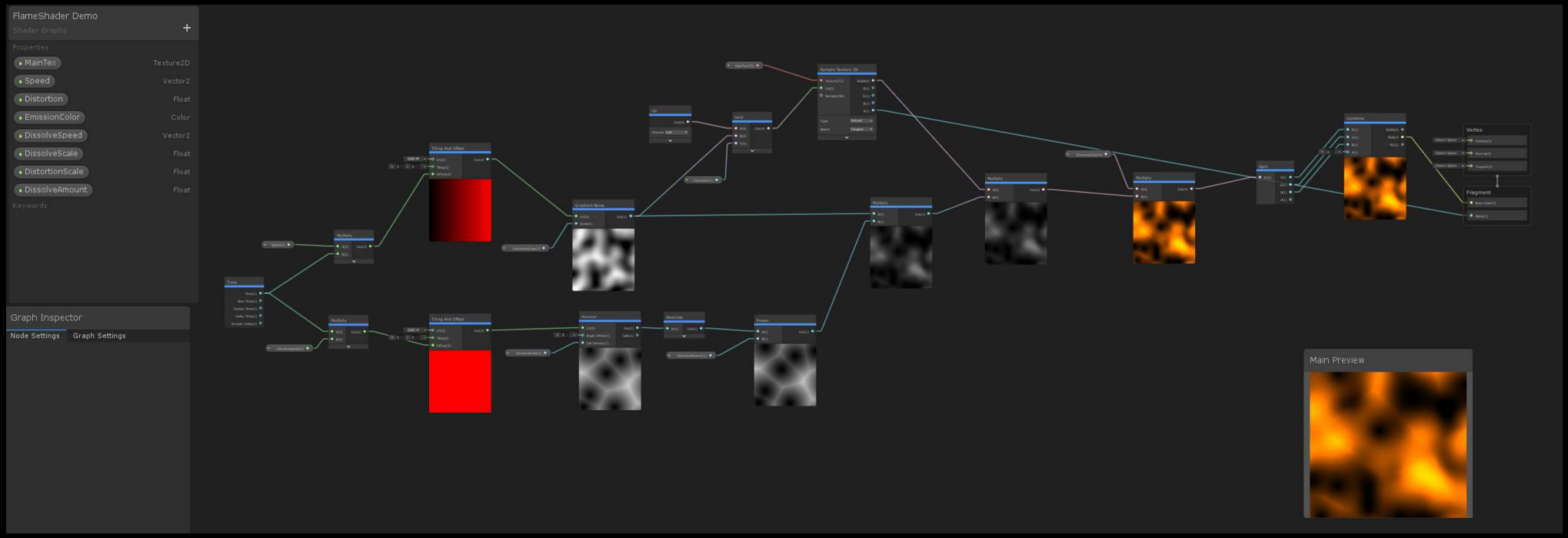
Flame Shader

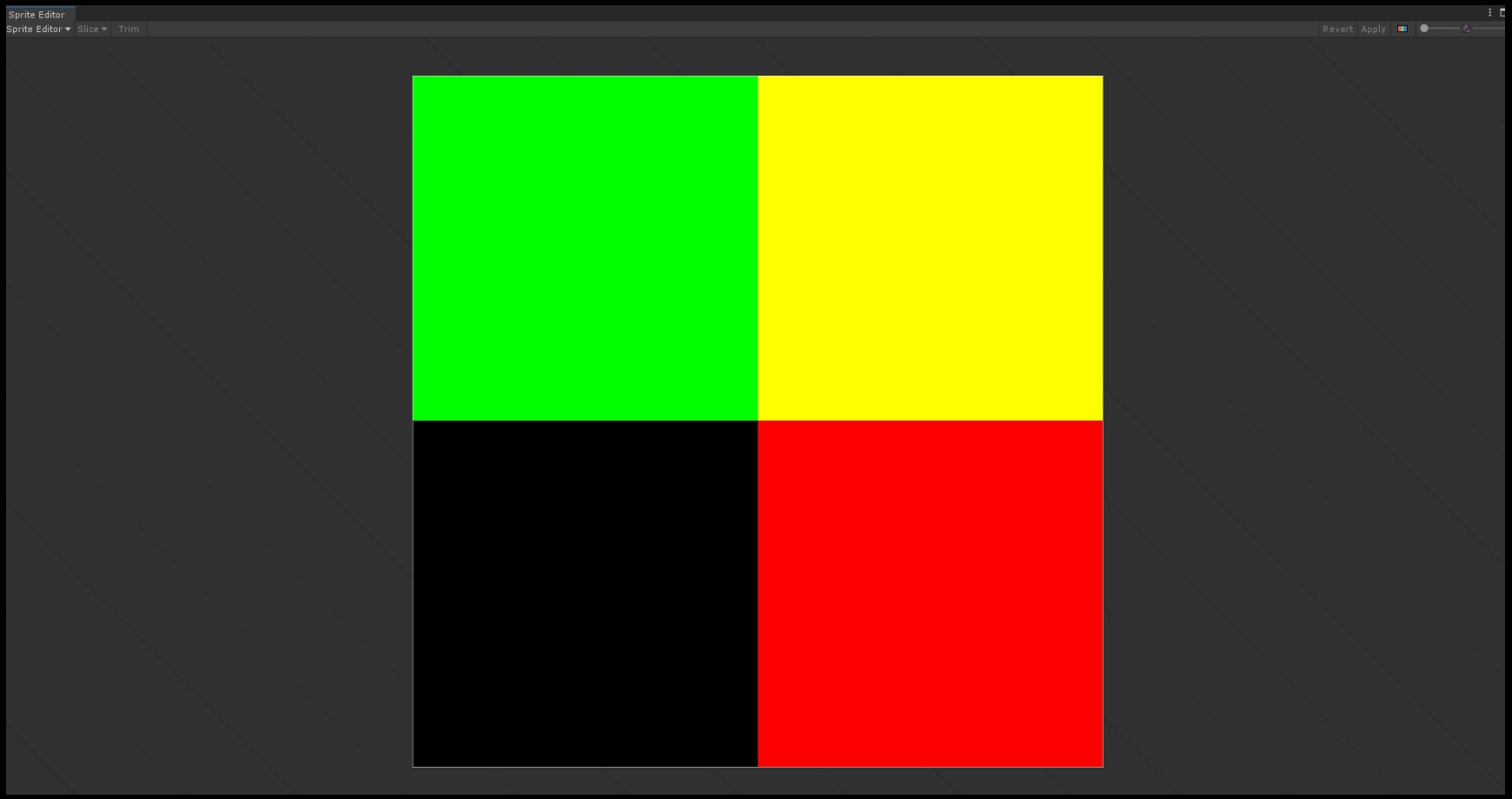
Flame Shader

Shader Graph – Flame Shader



Flame Shader

This Is Our Shader Test Colours Texture 2D Sprite



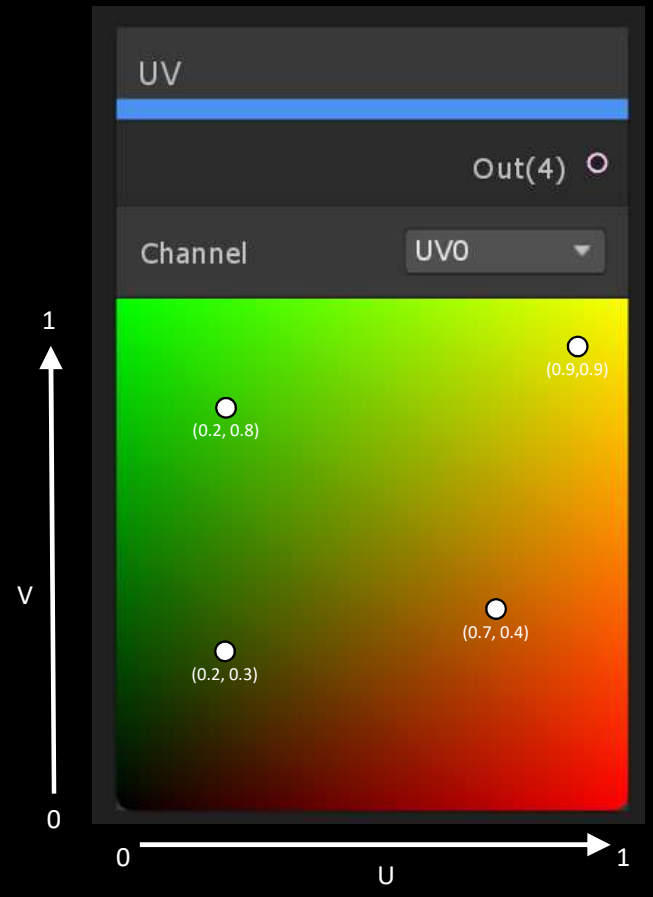
Flame Shader

Shader Graph – UV node

A UV map is used to map a textures colours to it's geometric shape. The U is the x-axis, the V is the y-axis. The U and V values go from 0 to 1.

In shader graph the UV node represents the mapping of texture colours on the sprite. Many nodes already have a UV(0) input specified that will by default pick up the original sprite UV map.

U values are shown in red, the V values are shown in green, and where both are positive they overlap and are represented by shades of yellow.

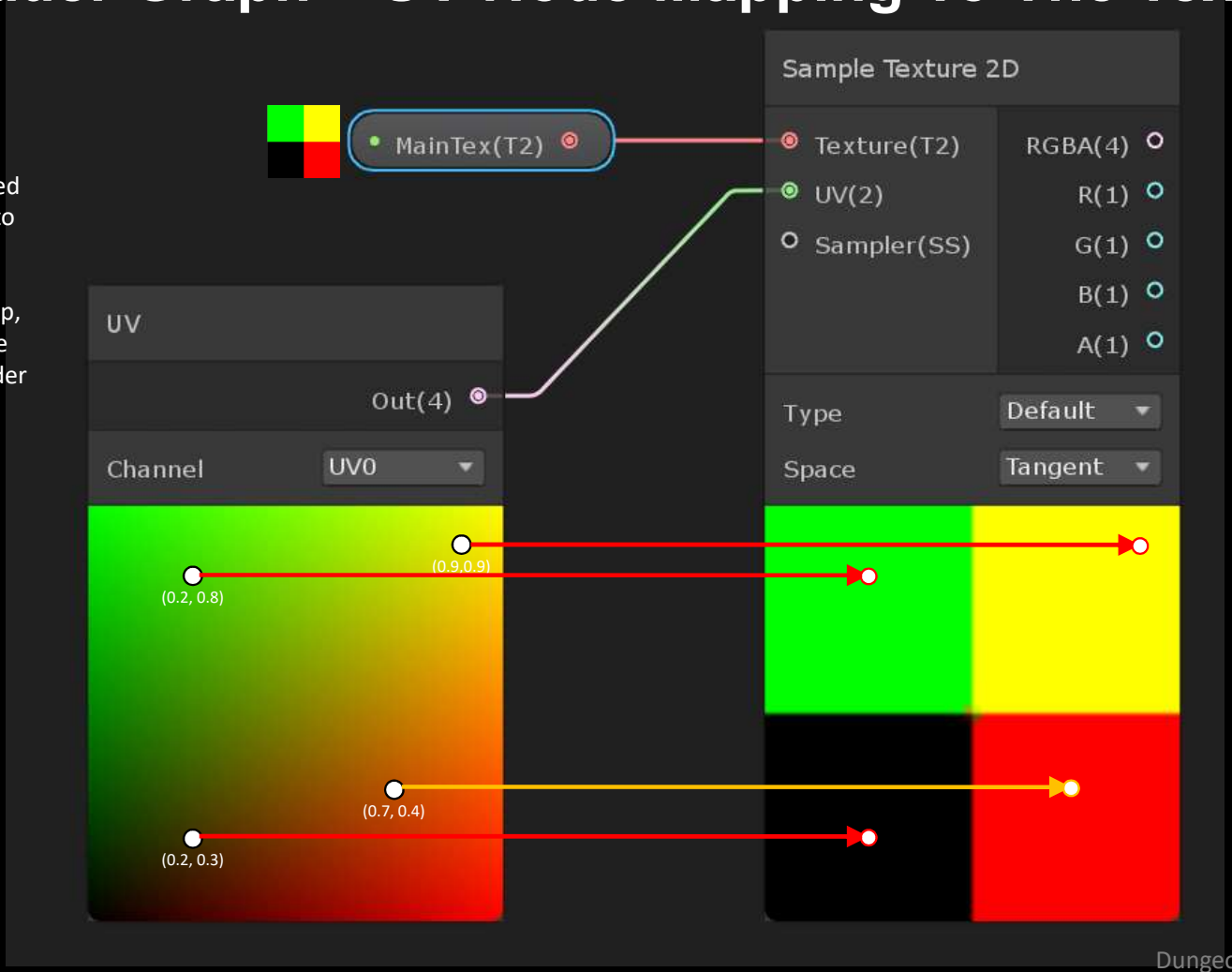


Flame Shader

Shader Graph – UV Node Mapping To The Texture 2D

Here the UV map is unaltered from the original mapping to the sprite texture.

So each point on the UV map, has a direct mapping to the equivalent point on our Shader Test Colours sprite.

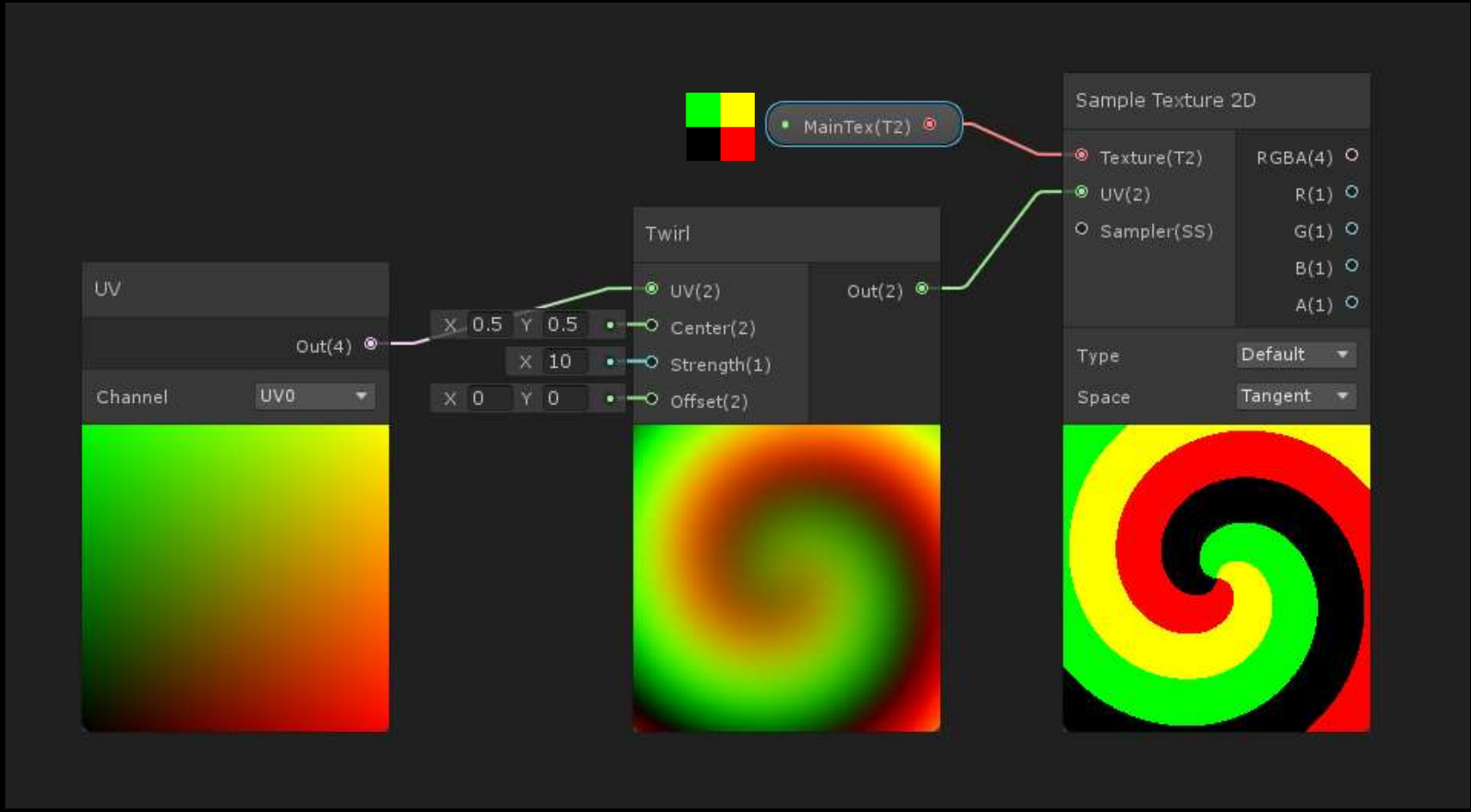


Flame Shader

Shader Graph – Manipulating The UV Map

Here we are manipulating the UV map by using the Twirl node.

You can clearly see that the generated output Texture2D is a result of the original Shader Test Colours sprite colours being manipulated according to how the UV map has been manipulated.



Flame Shader

Shader Graph – Adding Noise To The UV Map

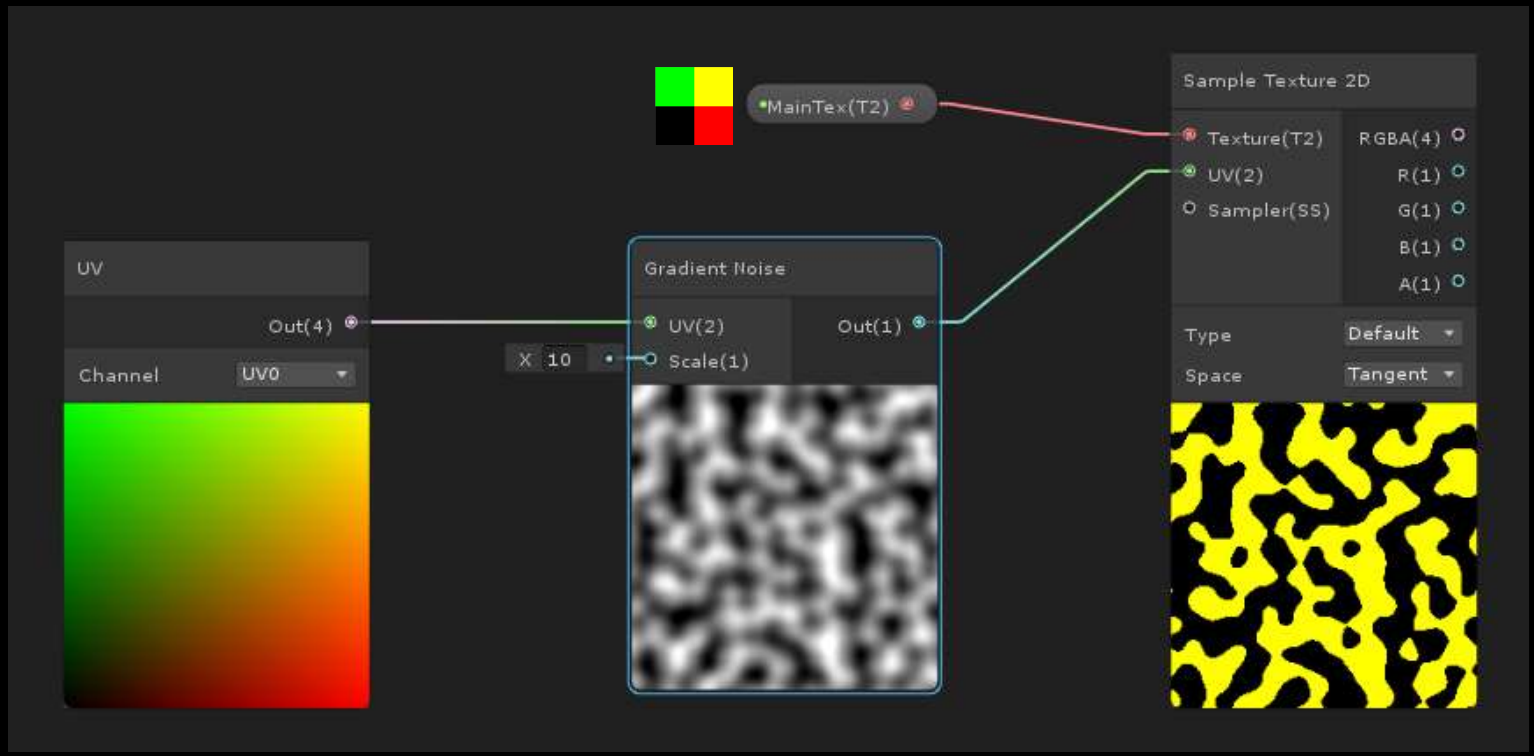
Here we are manipulating the UV map by using a Gradient Noise node.

The Gradient Noise node generates black and white values (and shades of grey). This is output as a single value so when converted to a vector2 for the UV it will always be in the form (T , T) (e.g. (0.1 ,0.1) or (0.2, 0.2) or (0.3, 0.3) etc. - never (0.2, 0.5).

The 'black' and dark grey noise values correspond to lower UV range values (e.g. U <0.5 AND V<0.5) – which map to the BLACK quadrant on the Shader Test Colours sprite.

The 'white' and light grey noise values correspond to higher UV range values (e.g. U >0.5 AND V>0.5) – which map to the YELLOW quadrant on the Shader Test Colours sprite.

The resulting texture 2D after applying the gradient noise as a UV map into the Shader Test Colours sprite using the sample texture 2D node is a black and yellow pattern .

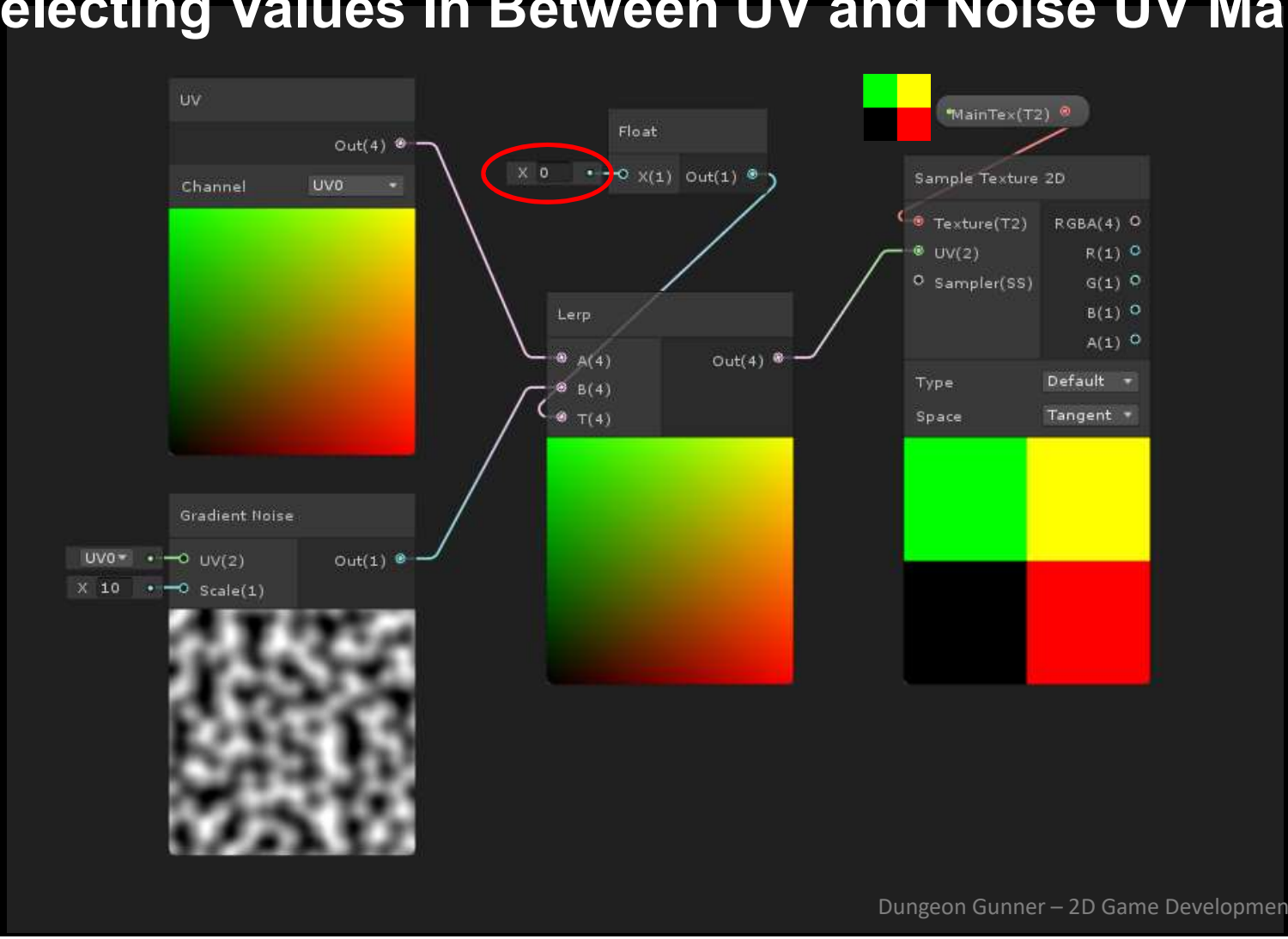


Flame Shader

Shader Graph – Selecting Values In Between UV and Noise UV Maps

Here we introduce a Lerp Node (Linear Interpolation) to select a value between the 2 inputs A and B, based on the value specified in T.

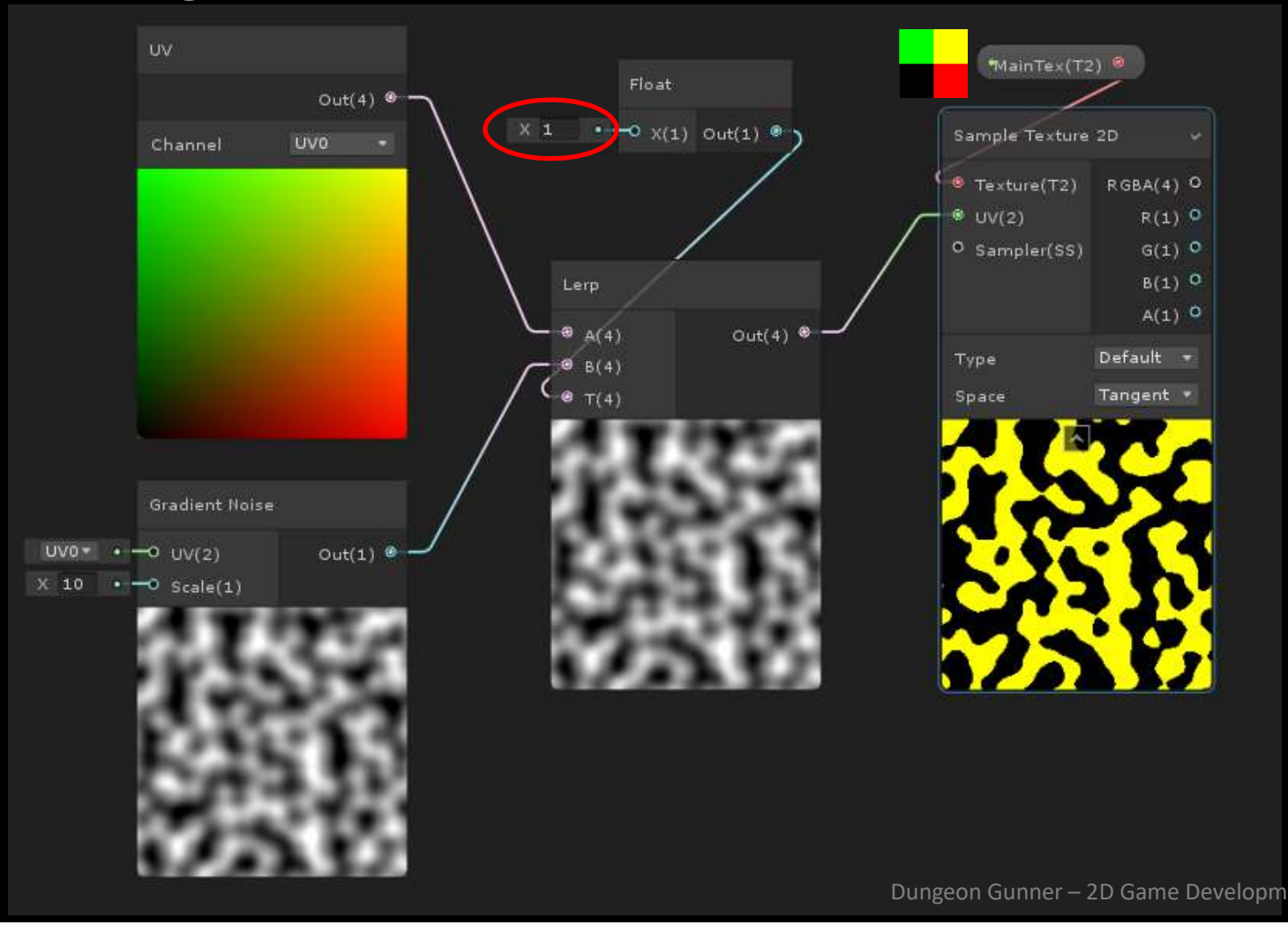
If T is 0 then all of value A is selected, which is the original UV map – so the output texture sprite is the same as our original Shader Test Colours sprite texture.



Flame Shader

Shader Graph – Selecting Values In Between UV and Noise UV Maps

If T is 1 then all of value B is selected, which is the full Gradient Noise UV map – so the output texture 2D is the same as our output texture generated from the gradient noise.

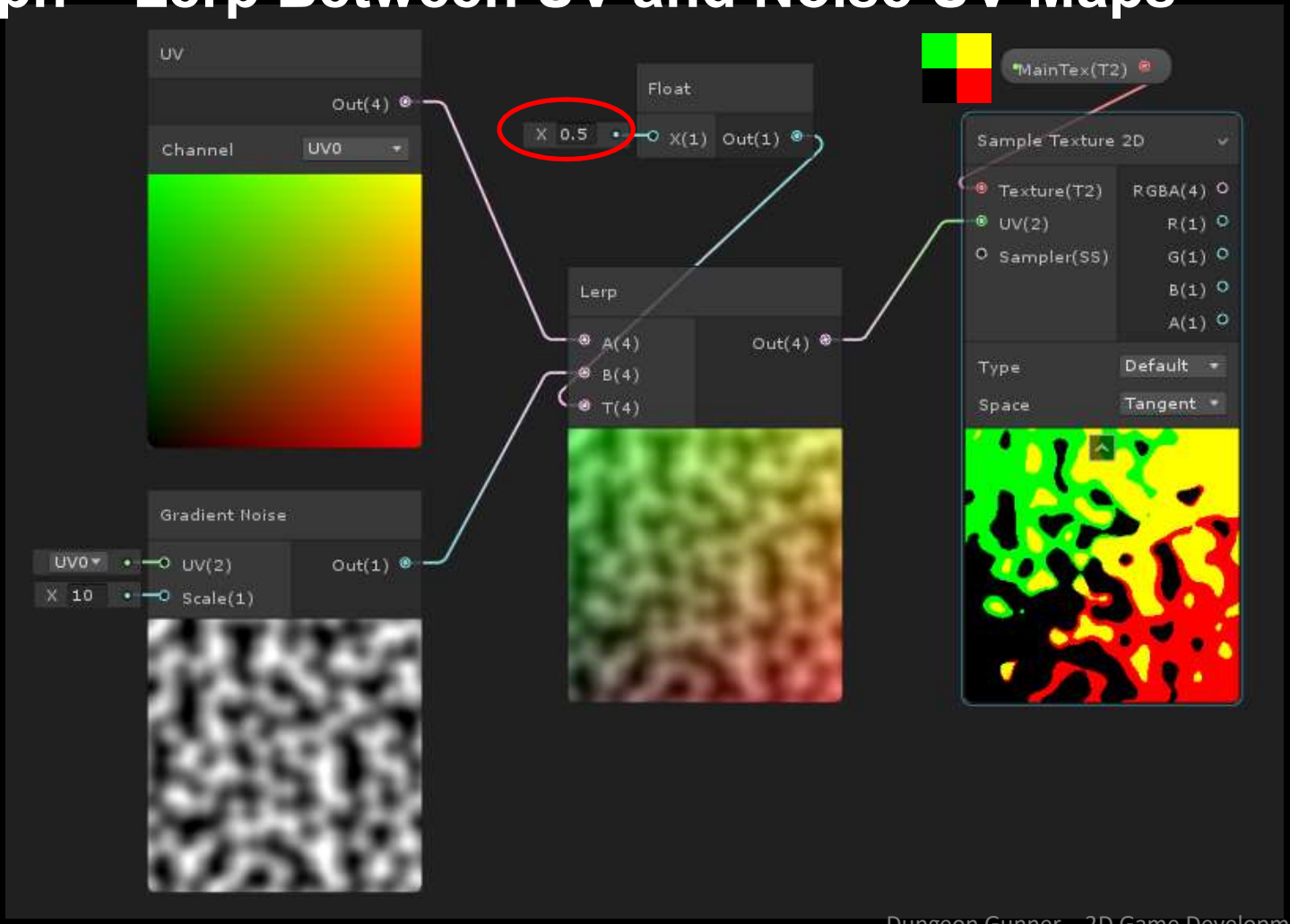


Flame Shader

Shader Graph – Lerp Between UV and Noise UV Maps

If T is 0.5 then for each position in the UV maps the average value of A and B is taken (i.e. (A + B) * 0.5.

This average value is then referenced on the original UV map / Shader Test Colours texture sprite to see what colour should be used.



Flame Shader

Shader Graph – Lerp Between UV and Noise UV Maps

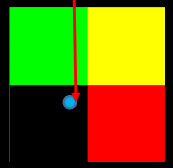
So here at the centre of this circle we have:-

Average of

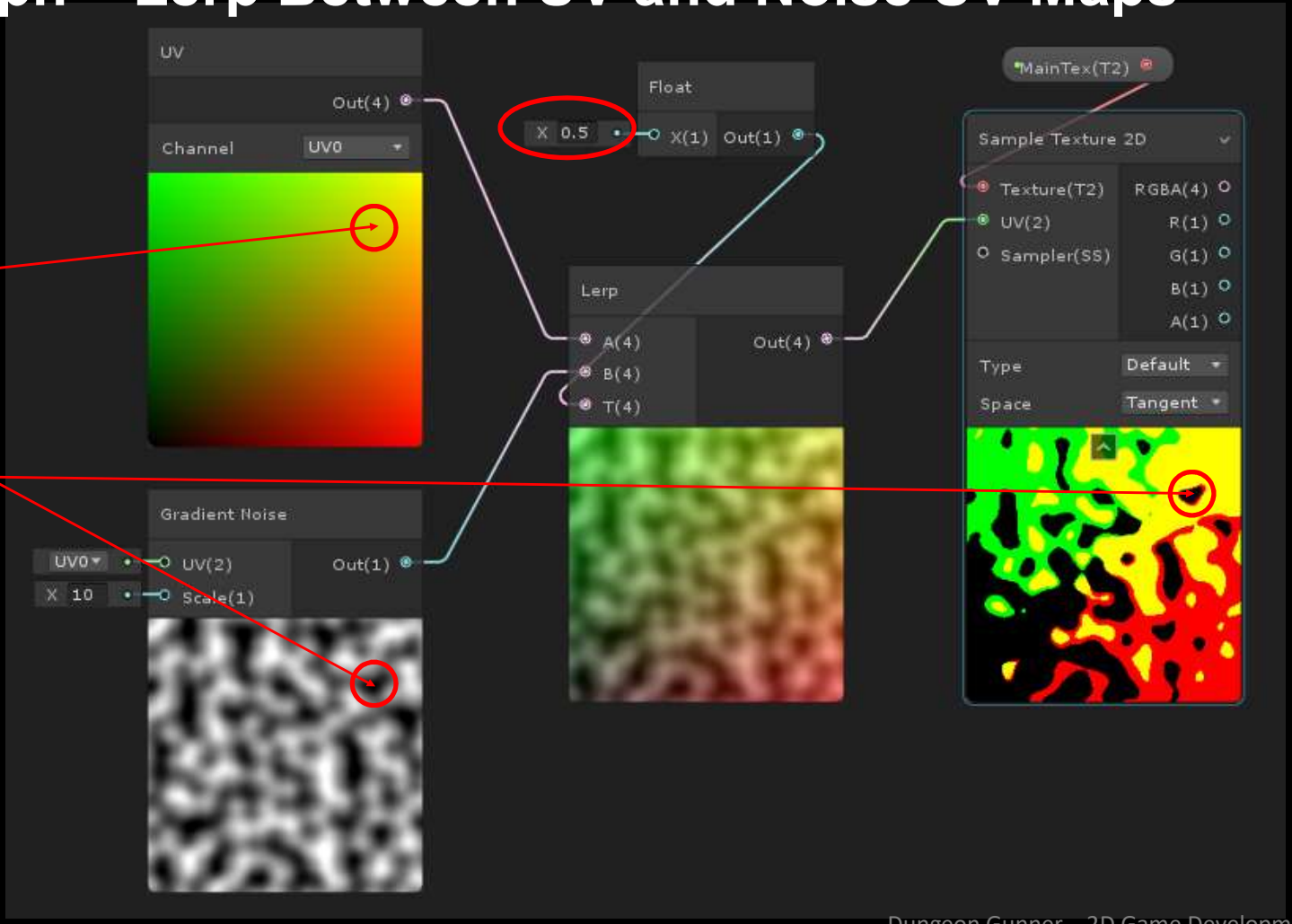
A (0.8,0.8)
&
B (0.05,0.05)

= (0.425,0.425)

which maps it just in the BLACK quadrant of the sprite



(Note: I have estimated the B value as (0.05,0.05) since it is nearly black)



Flame Shader

Shader Graph – Lerp Between UV and Noise UV Maps

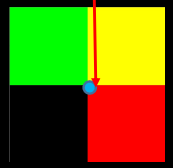
So here at the edge of this circle we have:-

Average of

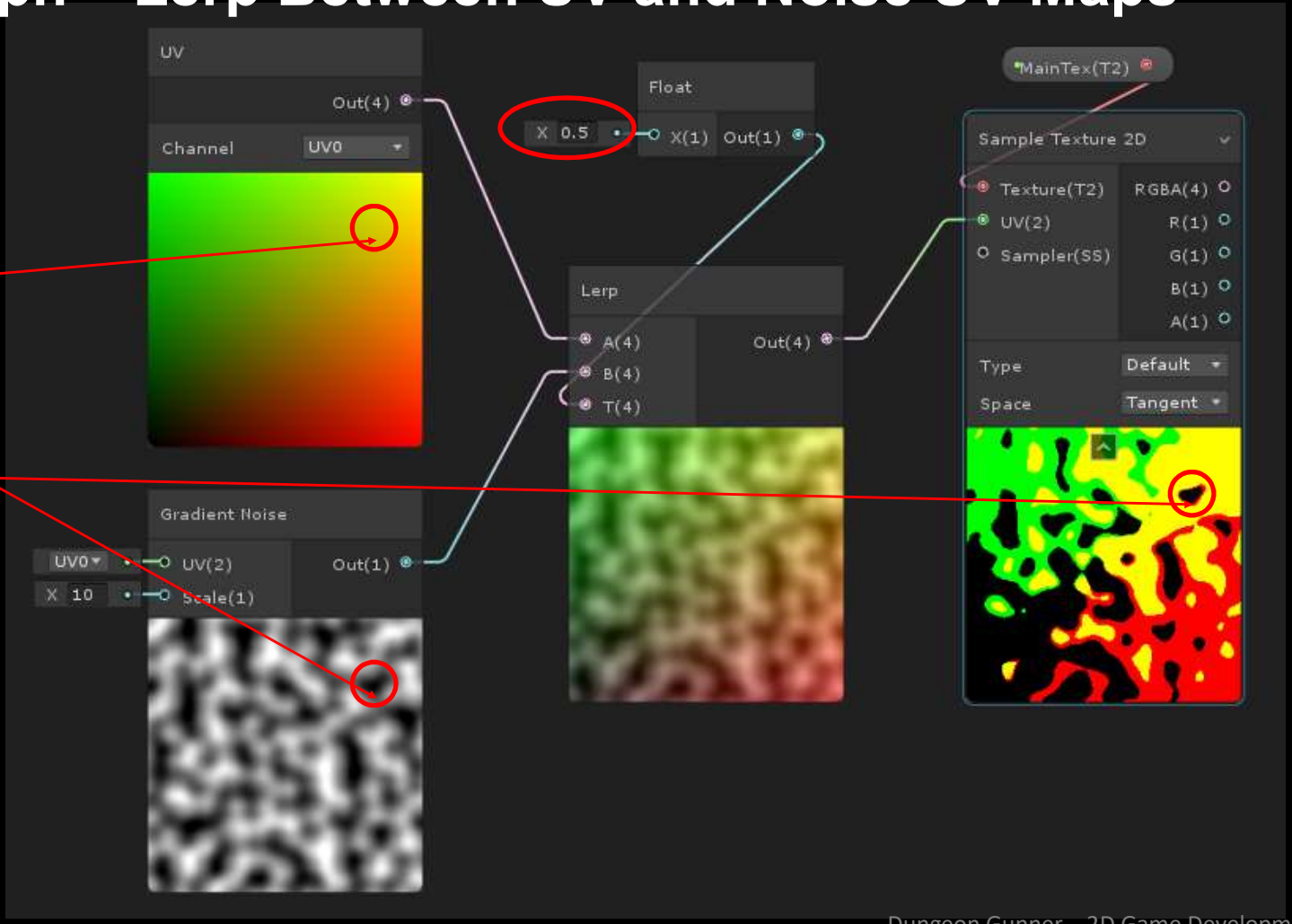
A (0.8,0.73)
&
B (0.25,0.25) (grey value)

= (0.525,0.49)

which maps it just in the RED quadrant of the sprite

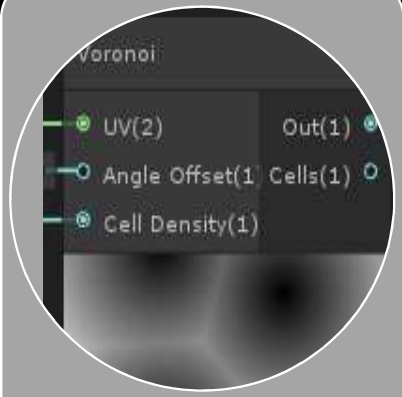


(Note: I have estimated the B value as (0.25,0.25) since it is quite a dark grey)

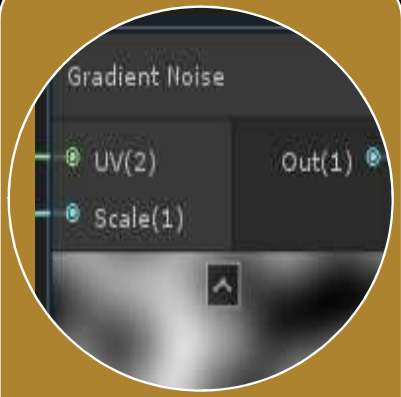


Flame Shader

Shader Graph – Creating A Flame Effect



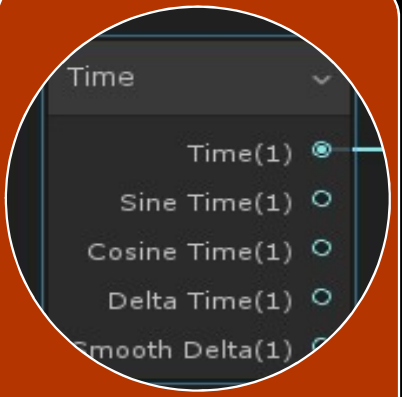
So we are going to use the Voronoi Noise Node....



... and blend it with the Gradient Noise Node to create an interesting noise effect. This will ....

... manipulate the UV map to affect any sprite renderer using a material created with this shader – such as a flame sprite. We'll also create movement in the noise using ...

.... the Tiling and Offset Node To Move / Offset The Noise UVs to simulate flame movement in conjunction with the ....



.... Time node to move the Offset amount over time to create continual movement