// MinTuts/Procedural Terrain.shader

```
float3 y = float3(p, p, p);

float r = 0;
float g = 1;
float b = 0;

return float4(y * float3(r, g, b), 1);
```

The goal of this commit to get our shader to go from green to black *instead of* white to black

We start by defining 3 **float**s

We name our 3 properties for the 3 color channels: **r** = *red*, **g** = *green*, **b** = *blue*

r and **b** are set to 0 because we're only interest in adding **g** to our **y** value

Numeric data types of the same dimension can be multiplied together; the ***** operator ensures all elements in the **type** are properly multiplied

Since **g** is 1…

as **y** approaches 1…

so does the green channel of the **float3** we multiply against **y**…

// MinTuts/Procedural Terrain.shader

```
float3 y = float3(p, p, p);

float r = 0;
float g = 1;
float b = 0;

return float4(y * float3(r, g, b), 1);
```

The goal of this commit to get our shader to go from green to black *instead of* white to black

We start by defining 3 **float**s

We name our 3 properties for the 3 color channels: **r** = *red*, **g** = *green*, **b** = *blue*

r and **b** are set to 0 because we're only interest in adding **g** to our **y** value

Numeric data types of the same dimension can be multiplied together; the ***** operator ensures all elements in the **type** are properly multiplied

Since **g** is 1…
as **y** approaches 1…
so does the green channel of the **float3** we multiply against **y**…
meaning the green channel of our color will become more intense (*aka brighter*)