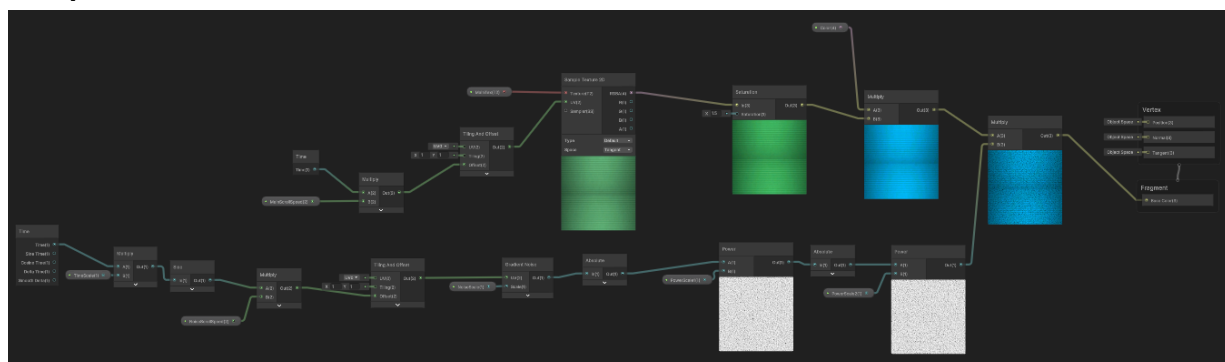


ComputerTerminal



El shader graph se usa una textura aplicando movimiento hacia abajo sumandole un ruido con tiempo con coseno para hacerlo que parezca randomizado.

TextToonShader

Instanciación de las variables para el toon shader.

```
Properties
{
    _Color ("Color", Color) = (1,1,1,1)
    _MainTex ("Albedo (RGB)", 2D) = "white" {}
    _ColorTex ("Texture", 2D) = "white" {}
    _brightness ("Brightness", float) = 0
    _strength ("Strength", float) = 1
}

sampler2D _MainTex;
sampler2D _ColorTex;
float _brightness;
float _strength;

struct Input
{
    float2 uv_MainTex;
};
fixed4 _Color;
```

metodo surface donde multiplicamos las texturas por el color de la textura.

```
void surf (Input IN, inout SurfaceOutput o)
{
    // Albedo comes from a texture tinted by color
    fixed4 c = tex2D (_MainTex, IN.uv_MainTex) * _Color;
    o.Albedo = c.rgb;
    o.Alpha = c.a;
}
```

metodo donde mapeamos la textura de niveles de toon y calculamos de acuerdo a la dirección de la luz.

```
half4 LightingSimpleLambert(SurfaceOutput s, half3 lightDir, half atten)
{
    half NdotL = dot(s.Normal, -lightDir);
    float2 uv = float2((NdotL + 1) / 2, 0);
    float4 levels = tex2D(_ColorTex, uv);

    half4 color;

    color.rgb = s.Albedo * _LightColor0.rgb * (levels * atten);
    color.rgb += _brightness;

    float3 intensity = dot(color.rgb, lightDir);
    color.rgb = lerp(intensity, color.rgb, _strength);
    color.a = s.Alpha;

    return color;
}
```