

UNITY GÜNLÜĞÜ

SHADER GRAPH (UNITY 2019)

SHADER NEDİR?

Shader ekran kartınıza oyununuzun çalışması sırasında ekranda olan objeleri (3B modeller, yazılar, .. vb) nasıl göstereceğini anlatan bir kod türüdür.

Pixel, Fragment, Surface gibi farklı türleri vardır. Texture da nesnelerinizi kaplayan, önceden hazırladığınız dokulardır. Bunlarda diffuse/albedo, normal, metallic gibi farklı türlerde hazırlanırlar.

Unity çeşitli shaders desteği verir. Surface shaders Fragment shaders ve vertex shaders.

Surface Shader'ı : Bir yüzeyin ışığa karşı ne şekilde tepki verdiğini belirleyen shaderlar

Displacement Shader'ı : Bir yüzey üzerindeki kıvrım ve girinti çıkıntıları.

Light Shaders : Sahnedeki ışık kaynağının yaydığı ışığın, yönü, miktarı ve rengini belirlemek için kullanılan shader tipidir.

Volume Shaders : Işığın, duman, sis gibi hacimsel bir yapıdan geçerken ne şekilde etkileneceğini .

Imager Shaders : Sonuç pikselinin çıkıştan önce, renk değişimlerini belirlemek amacıyla kullanılır.

SHADER KODU

```

SubShader
{
    Tags { "RenderType" = "Opaque" }

    CGPROGRAM
    #pragma surface surf CelShadedUsingForwardRendering
    #pragma target 3.0

    struct Input
    {
        float2 uv_MainTex;
    };

    sampler2D _MainTex;
    fixed4 _Color;

    half4 LightingCelShadedUsingForwardRendering(SurfaceOutput s, half3 lightDir, half atten)
    {
        half NdotL = dot(s.Normal, lightDir);

        //using smoothstep to soften the edges
        NdotL = smoothstep(0, 0.025f, NdotL);

        half4 c;
        c.rgb = s.Albedo * _LightColor0.rgb * (NdotL * atten * 2);
        c.a = s.Alpha;
        return c;
    }

    //surface shader function
    void surf (Input IN, inout SurfaceOutput o)
    {
        fixed4 c = tex2D(_MainTex, IN.uv_MainTex) * _Color;
        o.Albedo = c.rgb;
        o.Alpha = c.a;
    }

    ENDCG
}
FallBack "Diffuse"

```

```

Shader "MyShader"
{
    Properties
    {
        // The properties of your shaders
        // - textures
        // - colours
        // - parameters
        // ...
    }

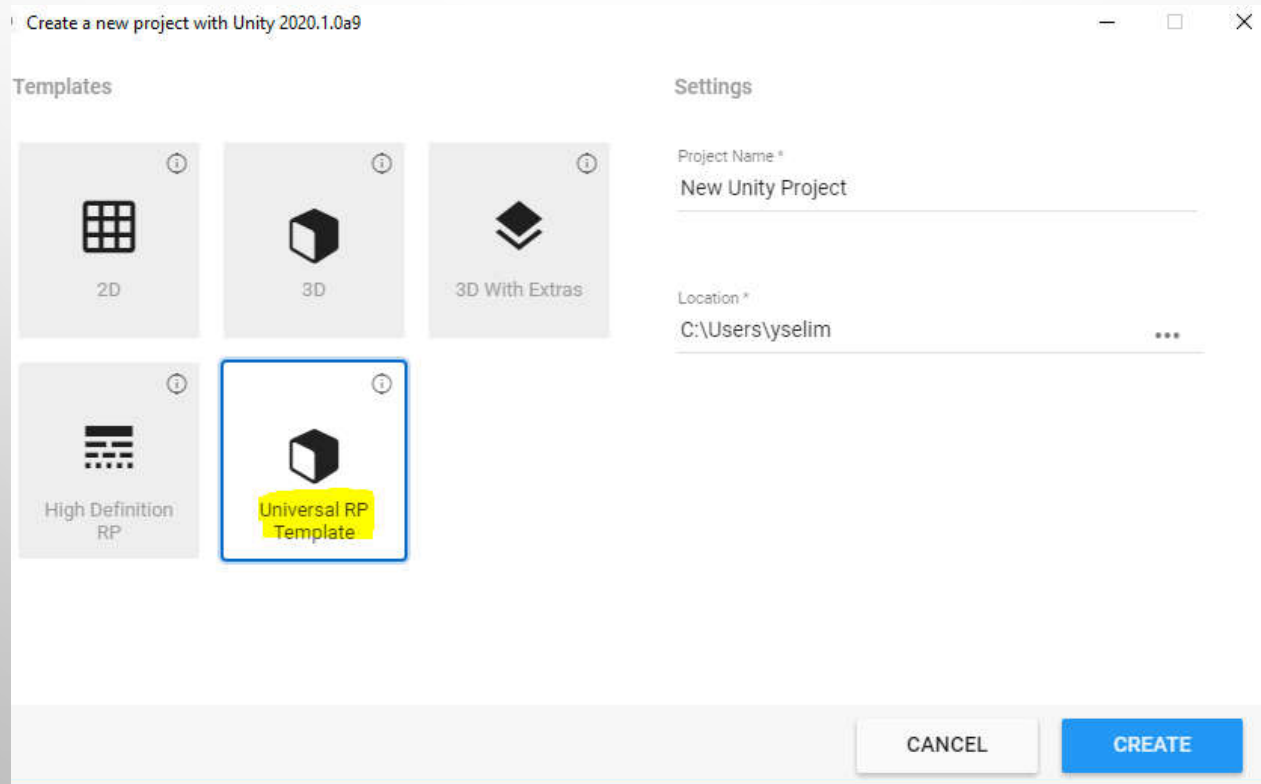
    SubShader
    {
        // The code of your shaders
        // - surface shader
        // OR
        // - vertex and fragment shader
        // OR
        // - fixed function shader
    }
}

```

Örnek bir Shader kodu

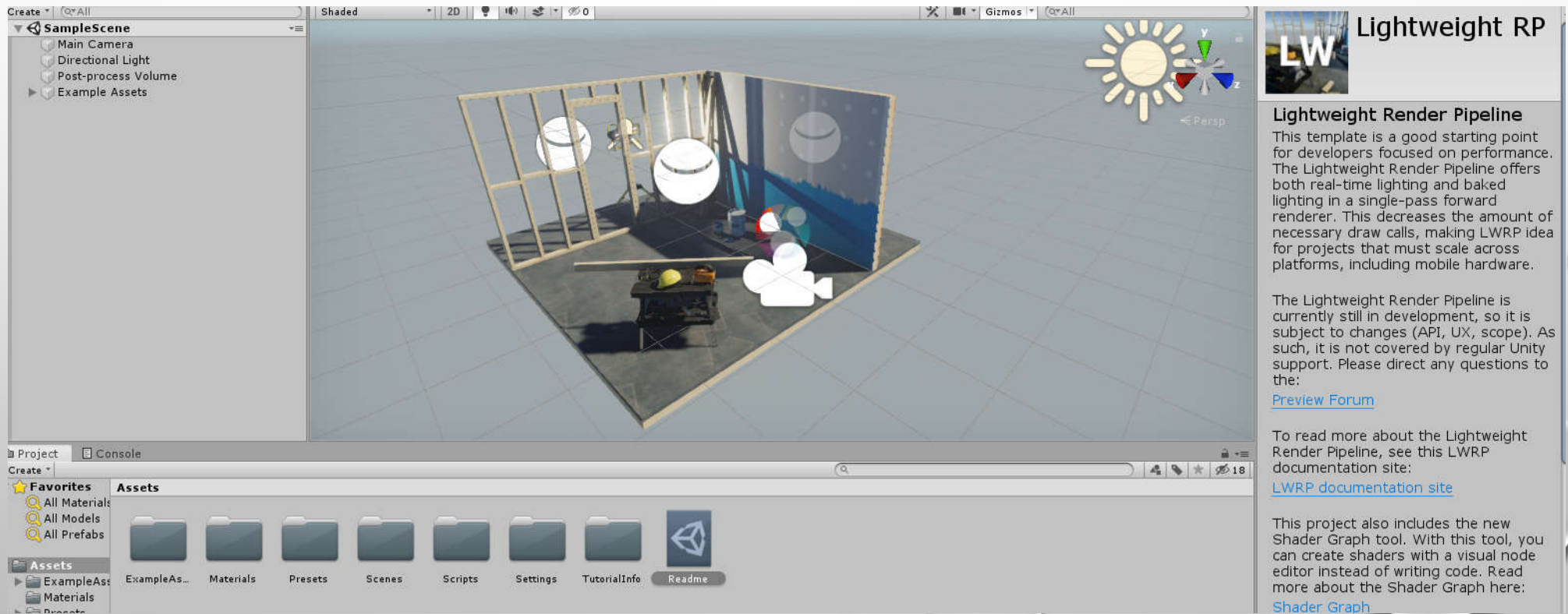
SHADER GRAPH (Kurulum)

Unity 2019 ve 2020 ile beraber projeleri oluřtururken **Render-pipeline Lightweight/HD** řeklinde proje Bařlatılırsa Shader-Graph paket iinde gelmektedir. 2018'de olduėu řekilde **Windows/Pack Manager** iinden kurulmasına gerek yok.



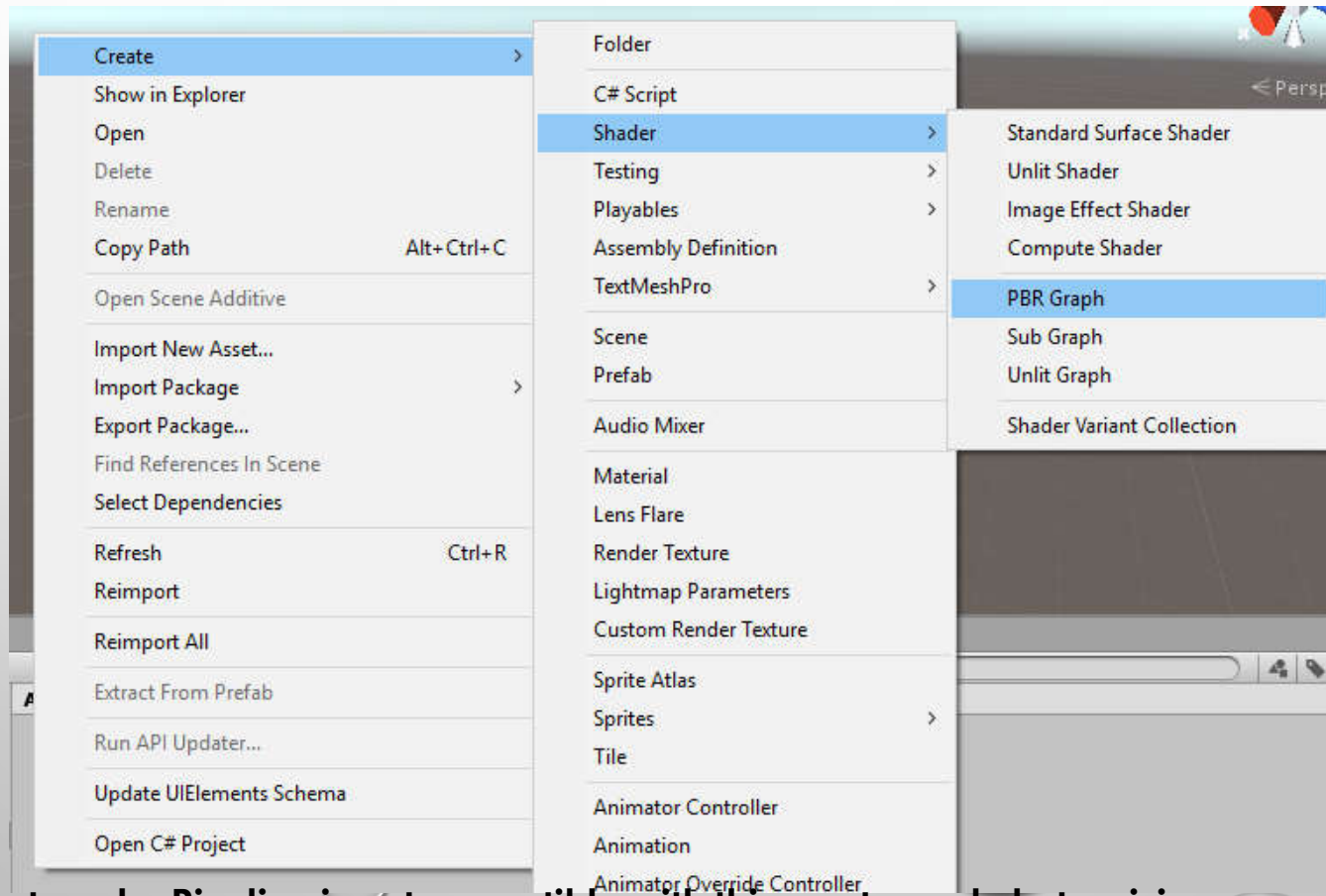
SHADER GRAPH (dissolve)

Sahneye bir nesne ekledik. **Shader graph** kullanarak bu nesneye *dissolve* efekti uygulayacağız. Shader graph'te düğümler oluşturup. Çözülme rengi, kenarlık rengi , çözülme süresi/hızı ve texture ayarları vs yapacağız.



SHADER GRAPH (dissolve)

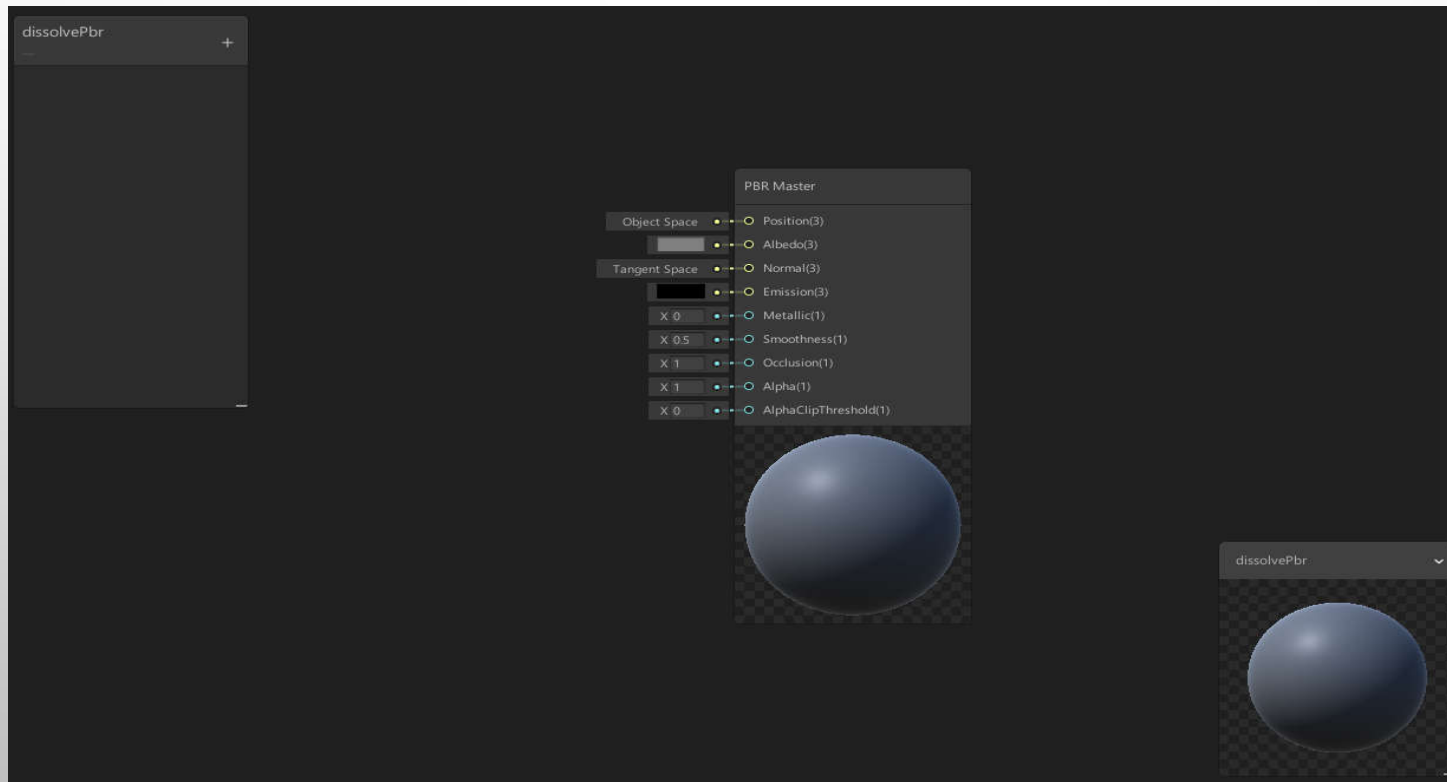
Shader graph için proje ekranında **Create /shader /PBR Graph** seçeneği ile «**dissolve**» ismini verdiğimiz yeni bir shader atılır. Shader tıklanıp Shader Graph UI açılır.



The Current render Pipeline is not compatible with this master node hatası için Project Settings /Graphics kısmına oluşturulan Shader dosyası atanacak!!

SHADER GRAPH (dissolve)

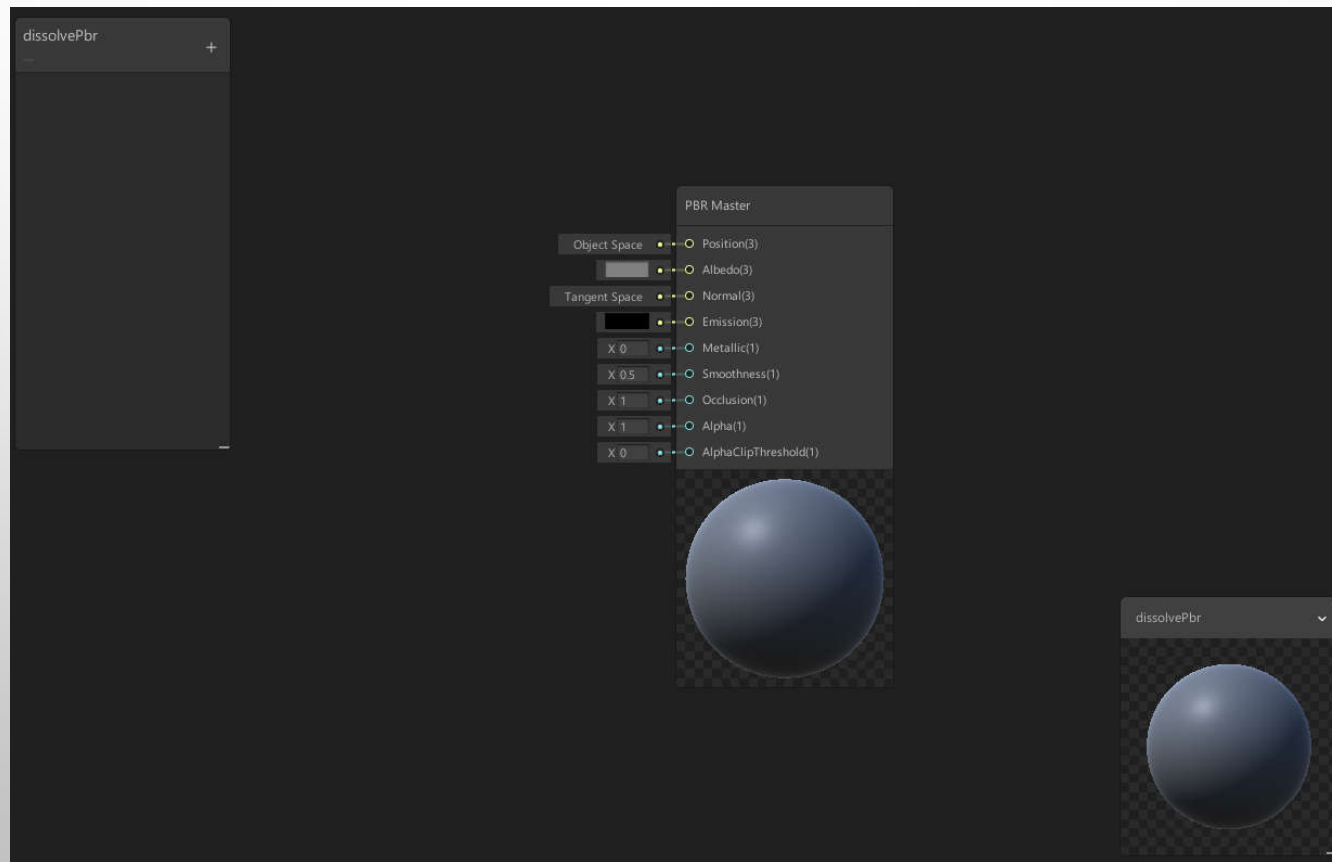
Shader Graph UI açıldığında bir **Master node** mevcuttur. **Albedo**, **Normal**, **Emission** gibi parametreleri değiştirerek efektimizi oluşturabiliriz. Master düğüm ile materyalin rengini, metalikliğini, alfa kanalını , yüzeyin tipini (opak yada transparan gibi), culling (kullanıcının görmediği poligon yüzeylerin render edilmemesi) ayarları yapılandırılır.



Shader Graph UI

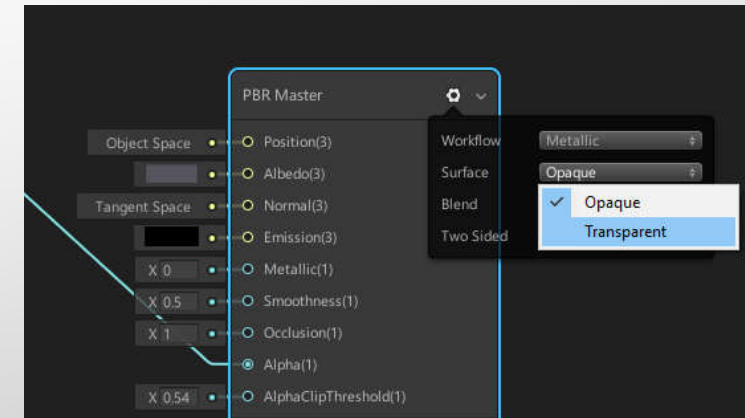
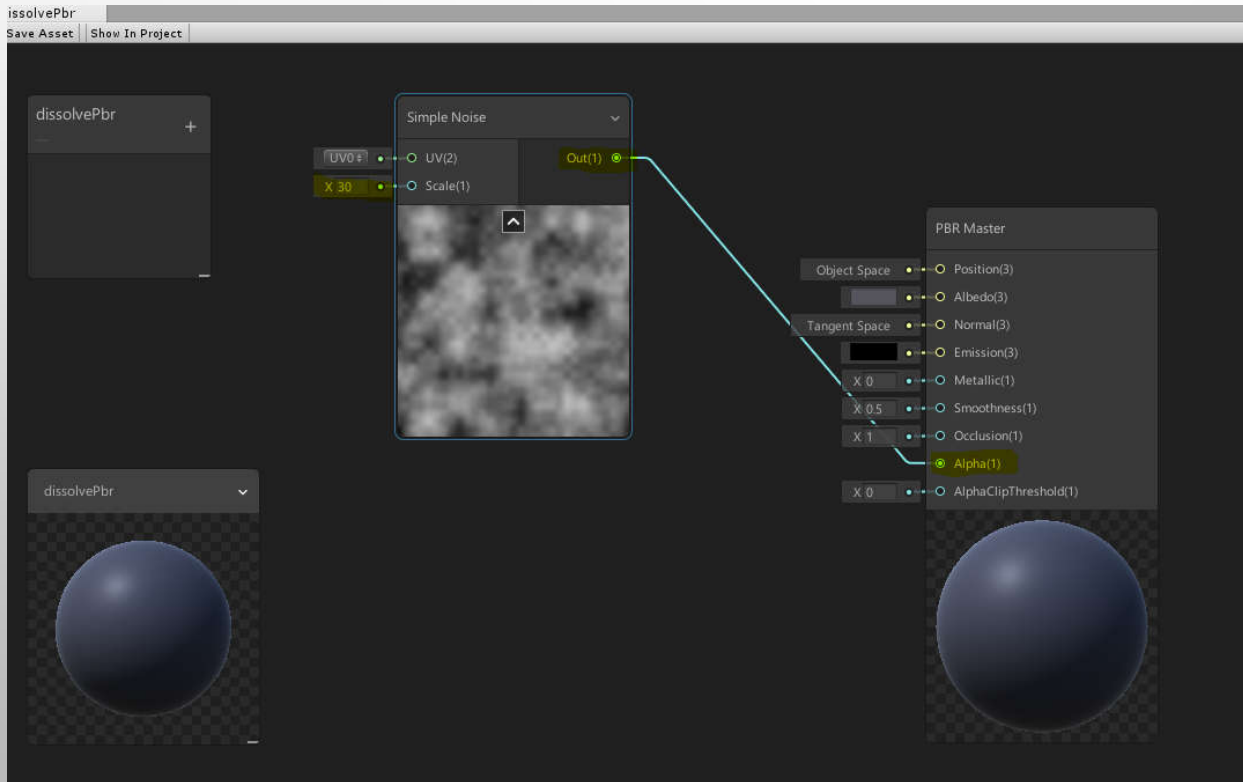
SHADER GRAPH (dissolve)

Öncelikle materyal Albedo rengi ayarlanır. Efekt iki kısımlı olacak. Transparans özelliği ile çözülme efekti ve çözülme esnasındaki parlayan kenarların kontörlerinin çizgilerinin rengi ve kalınlığı gibi özelliklerini ayarlayacağız.



SHADER GRAPH (dissolve)

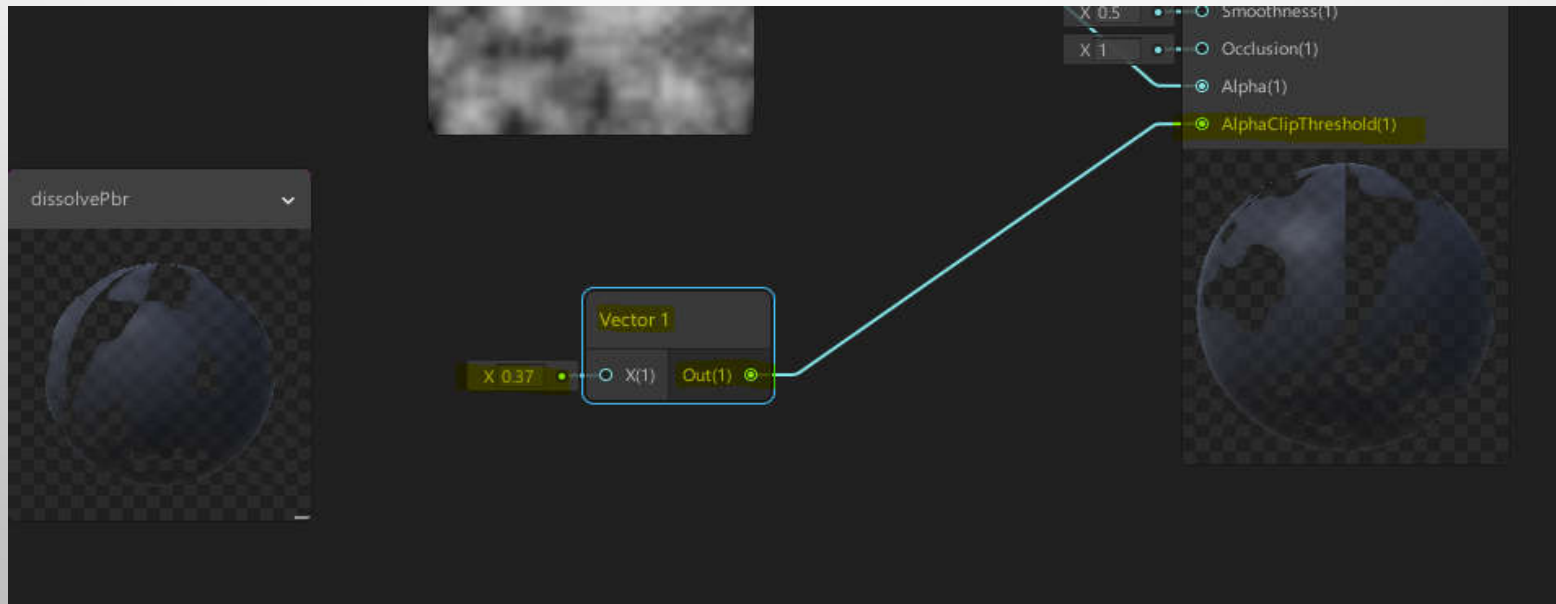
Bu iki efekti uygulayabilmek için ilk önce bir gürültü tanımlanır, gürültüyü eklemek için Shader Graph boş bir yerinde **Space** tuşuna basılarak **Simple Noise** seçeneği kullanılır, gürültü değeri 30 seçilerek gürültünün output'u Master node içindeki Alpha(1) kanalına sürüklenip bağlanır. Gürültü ön izlemesini görebilmek için Alpha kanalını Opak'tan Transparana alınarak kontrol edilebilir. Tekrar Opakta bırakılır.



SHADER GRAPH (dissolve)

Burada istenilen efekt **AlphaClipThreshold** deęerini kullanarak oluřturulan gürültü texture'ının animasyonunun material üzerinde oynatılmasıdır. Böylece eşik deęeri altında yada üstünde olan deęerlere göre görünürlük/görünmezlik deęerleri ayarlanır.

Denemek için; Bu eşik deęerini kontrol edecek bir düęüm space'e basılıp eklenir. **Vector1** düęümü eklenir, Vector 1 float bir deęer oluřturur. Vektör1'in out çıkışı Master Node içindeki **AlphaClipThreshold** girişine sürüklenerek atanır.



SHADER GRAPH (dissolve)

Asıl olan nedir? Transparan efekt *AlphaClipThreshold* eşik değeri altında ve üstünde kalan piksellerin gösterilmesi veya gizlenmesi prensibidir.

Eğer eşik değeri 1'e çekilirse nesne OUT, 0'a çekilirse IN efekti oluşturulmuş olur. Eşik değeri ve altındaki alanlar beyaz, eşik değeri üzerindeki pikseller siyaha çekilir. Sonuç sağdaki şekil gibi olacaktır.

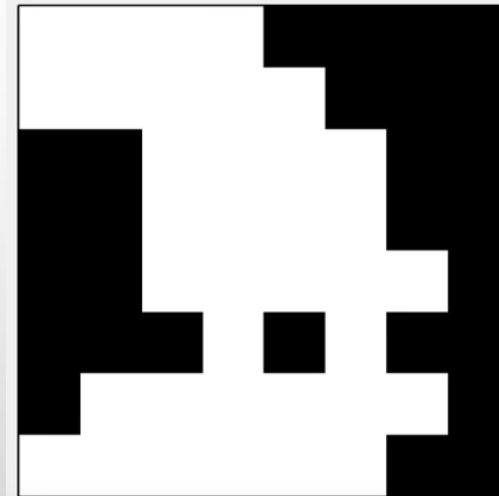
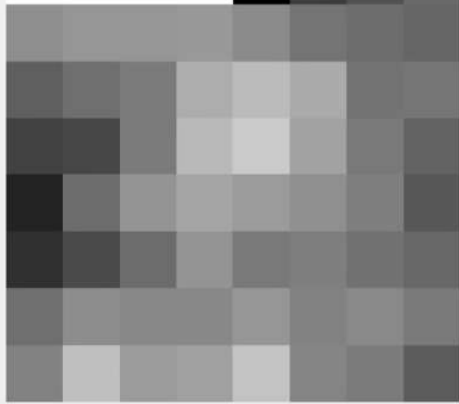
>0.7?



.8 .9 1 .8 .6 .4 .6 .7

Alpha Clip Threshold

0.7



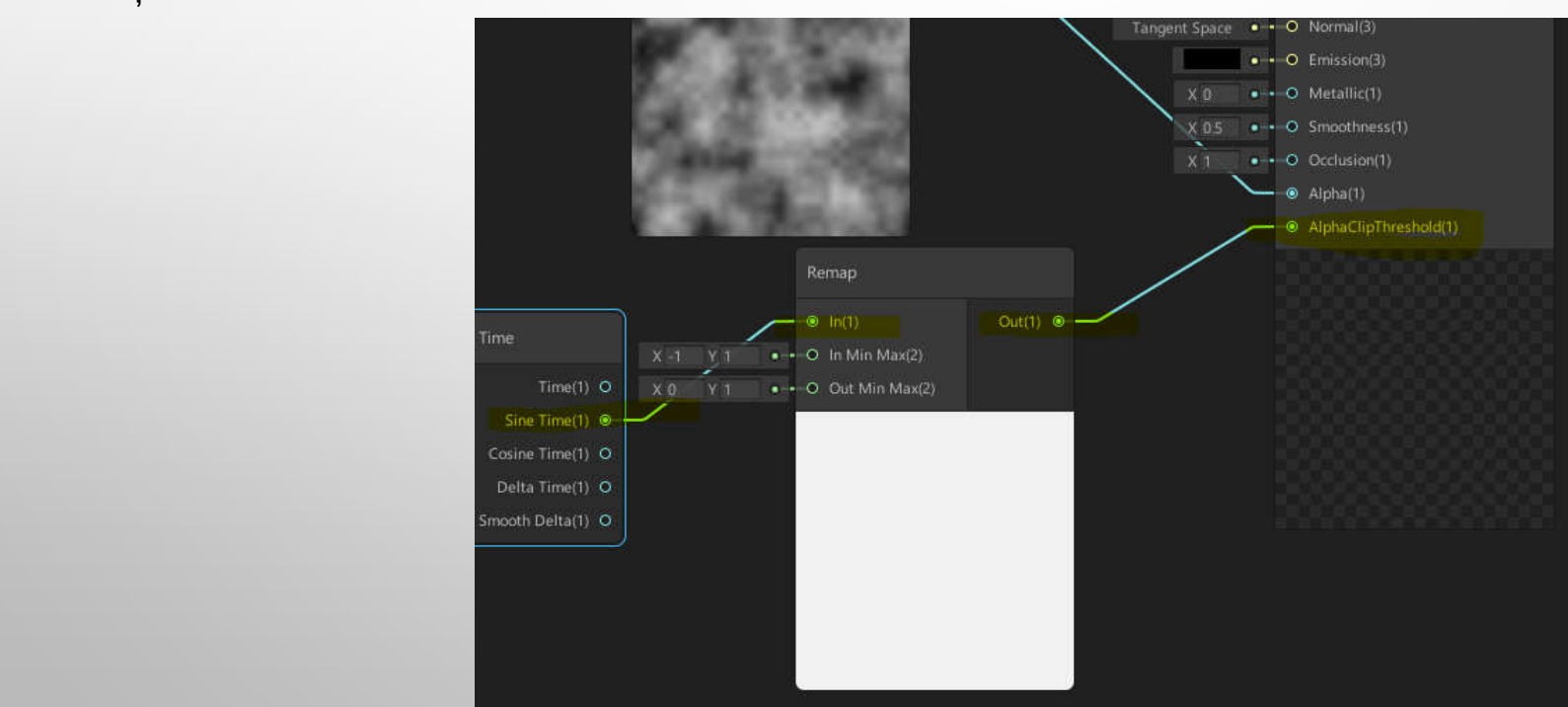
Alpha Clip Threshold

0.7

Efektin otomatik olarak 0 ile 1 arasında çalışması

Efektin otomatik olarak 0 ile 1 arasında çalışması için **Time Node** kullanılır. Sapce'e basıp Time node eklendikten **Sine Time** zamanının -1 ile +1 arası olan değeri **Remap Node** eklenir, Remap Node IN kullanılarak 0 ile 1 arasına map edilir/eşleştirilir.

Son olarak Remap çıkışı Master Node'un AlphaClipThreshold girişine sürüklenir. Zamana bağlı olarak efekt çalışmaya başlatılır. Efektin ilk kısmı tamamlanır.



SHADER GRAPH (dissolve-Kenar Efektinin uygulanması)

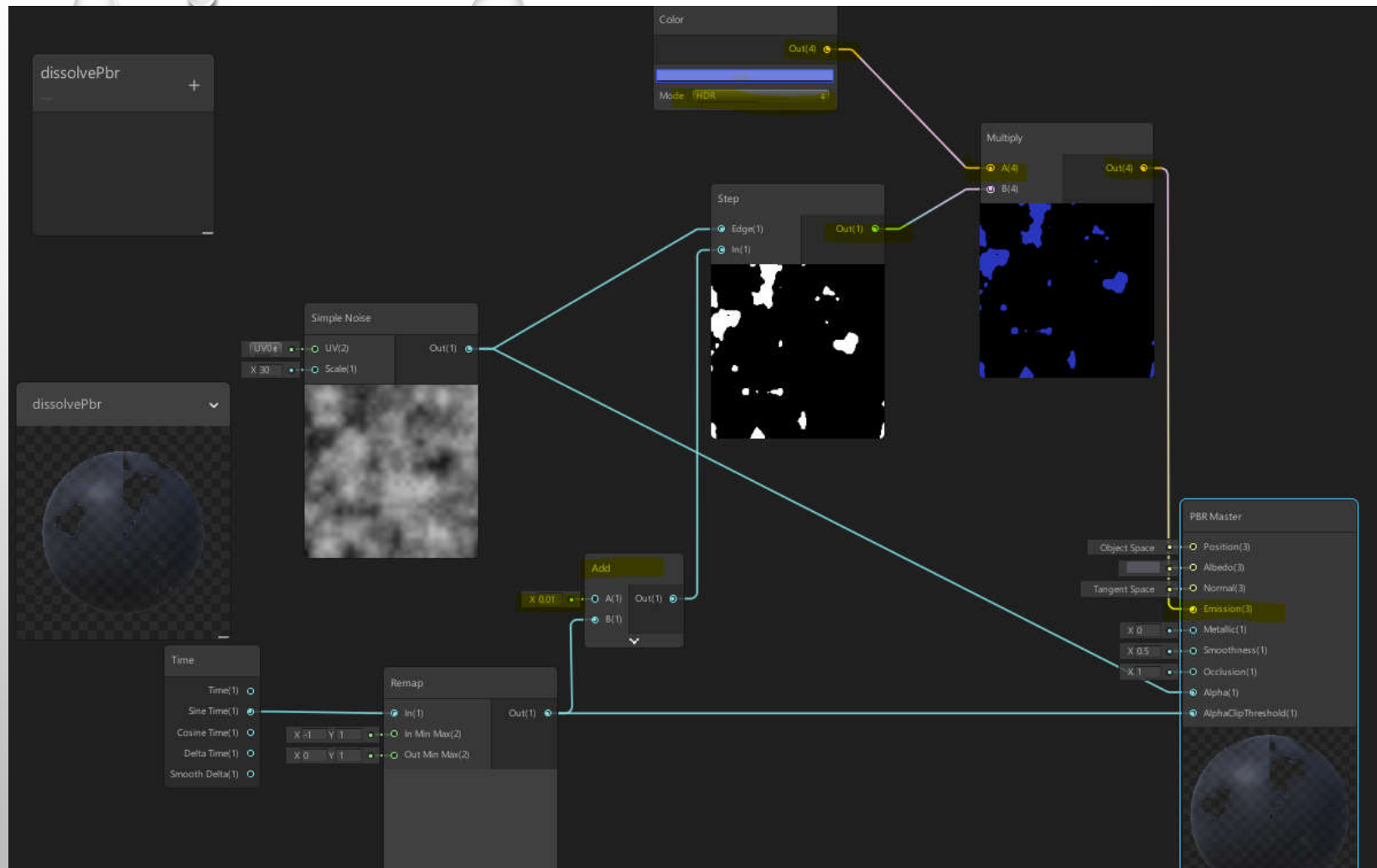
Efektin ikinci kısmı kenarlıkları eklemek, kenarlık geçişi için **Step Node Edge** eklenir. Gürültü çıktısı Step Node Edge(kenar) girişine sürüklenir, gürültüyü Time Node gibi oynatmak için kullanılır. Step Node'u da zamana bağlı kılmak için Remap çıktısını Step Node'un In girişine bağlanır.

Step Node çıktısı ise Master Node'da **Emission** girişine sürüklenir. Bu şekilde Step Node çıkışında Edge kısımlarda efekt görülebilir. Edekti görebilmek için geçici olarak Remap çıkışı iptal edilir, efekt görülür ve tekrar silinen bağlantı geri bağlanır.

Lakin iki efekt aynı anda çalıştığı için kenarlar fark edilmeyebilir. Bu sebepten kenar efektine bir gecikme uygulanabilir ve Kenarlıkların kalınlığı değiştirilebilir. Bunun için Remap çıktısını Step Node In'e bağlamadan önce araya **Add Node** eklenir ve 0.01 değeri ayarlanır.

Son olarak kenarlık rengi için önce bir **Multiply Node B** ardından **Color Node** eklenir. Color Node HDR seçilir ve renk tespiti yapılır Multiply Node çıkışı **Master Node Emissions'a** sürüklenir.

SHADER GRAPH (dissolve-Kenar Efektinin uygulanması)

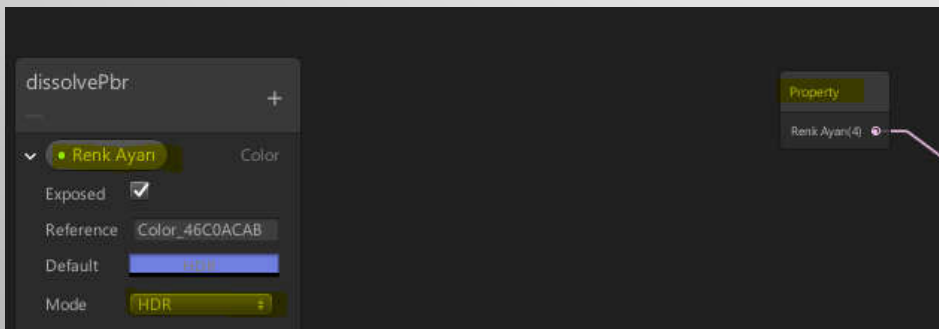
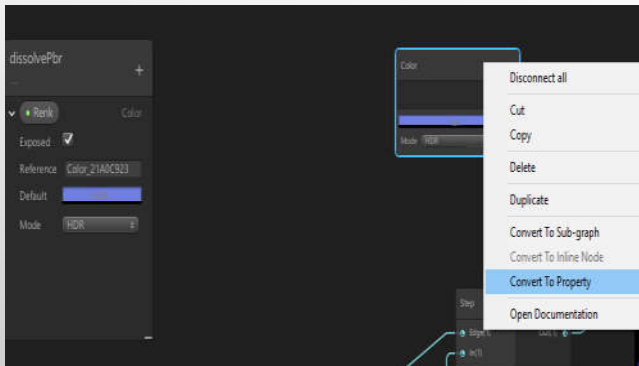


SHADER GRAPH (dissolve-Kenar Efektinin uygulanması)

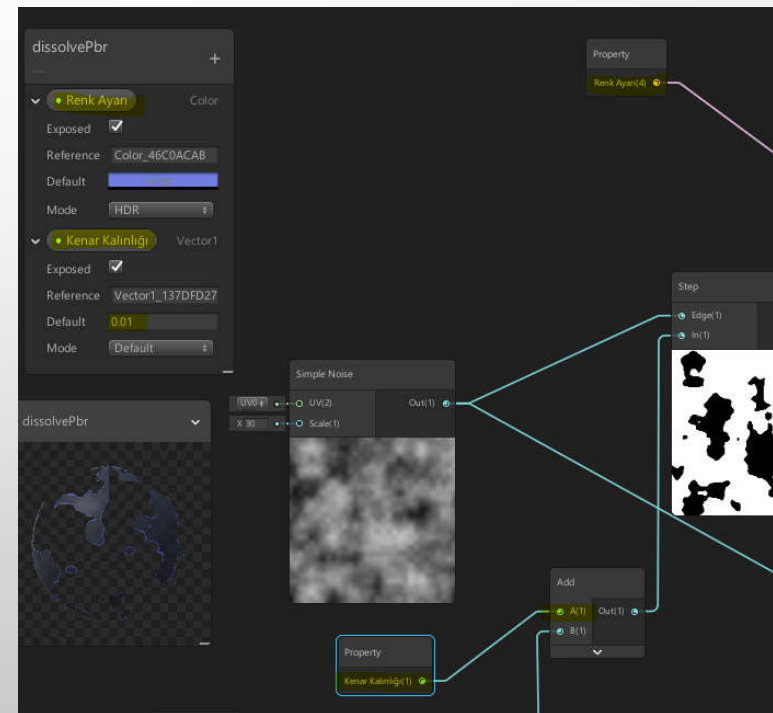
Shader Graph içinde tanımladığımız bu düğümleri bir Property şeklinde kullanılabilmesi için bunların üzerinde sağ buton yapılır **Conver To Property** seçeneği ile istenilen düğümler özelliğe çevirilir.

- **Color Prop** için «Kenar Renk» isimli property oluşturulur.
- Kenar kalınlığı; Prop ekranında değeri 0.01 olan bir Vector1 tanımlanır graph alanına sürüklenir ve çıktısı Add Node'a bağlanır. Prop ekranında «Kenar Kalinligi» ismi verilir.

Color Prop



Edge Prop

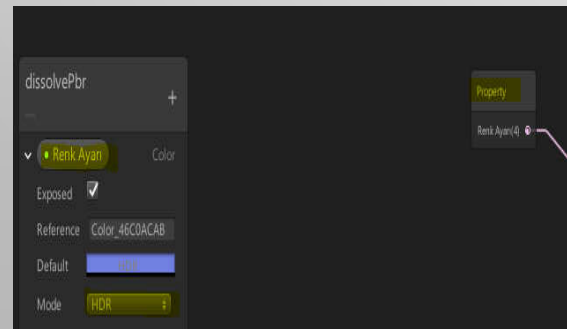
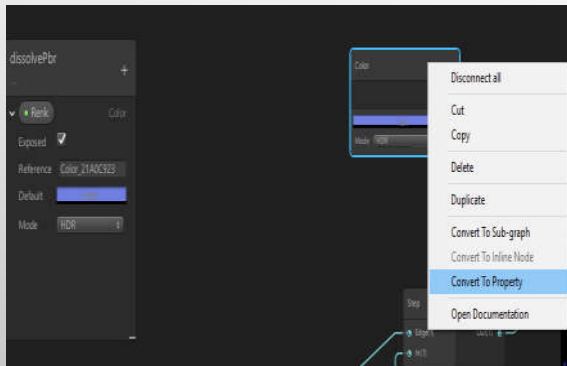


SHADER GRAPH (dissolve-Kenar Efektinin uygulanması)

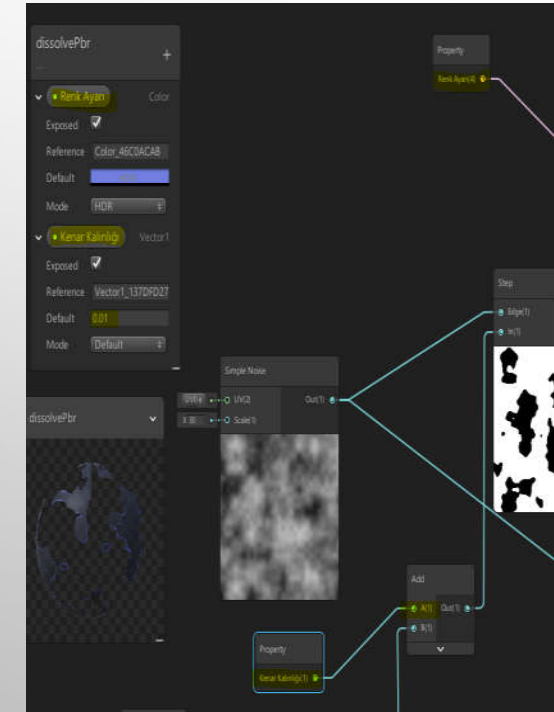
Shader Graph içinde tanımladığımız düğümleri bir Property şeklinde Unity Editorde kullanılabilmesi için bunların üzerinde sağ buton yapılır **Conver To Property** seçeneği ile istenilen düğümler özelliğe çevrilir.

- **Color Prop** için «Kenar Renk» isimli property oluşturulur.
- Kenar kalınlığı; Prop ekranında değeri 0.01 olan bir Vector1 tanımlanır graph alanına sürüklenir ve çıktısı Add Node'a bağlanır. Prop ekranında «Kenar Kalınlığı» ismi verilir.
- Noise Prop ; son olarak gürültü değeri için default değeri 30 olan bir Vector1 tanımlanır sürüklenir ve **Noise Node'** a bağlanır. En son modelin arka yüzeyinde önizlemede görebilmek için **Two-Sided** seçilir.

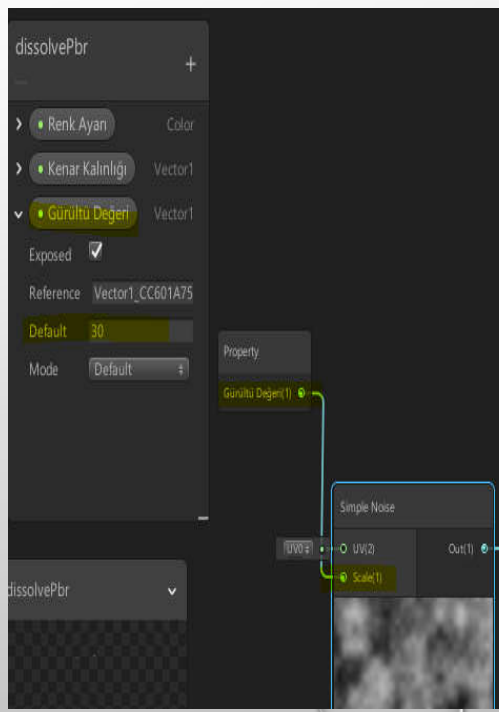
Color Prop



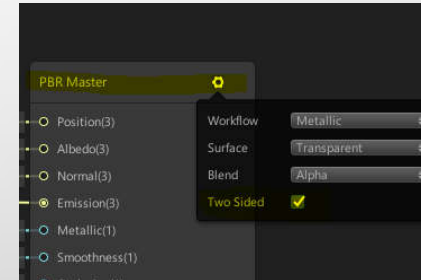
Edge Prop



Noise Prop

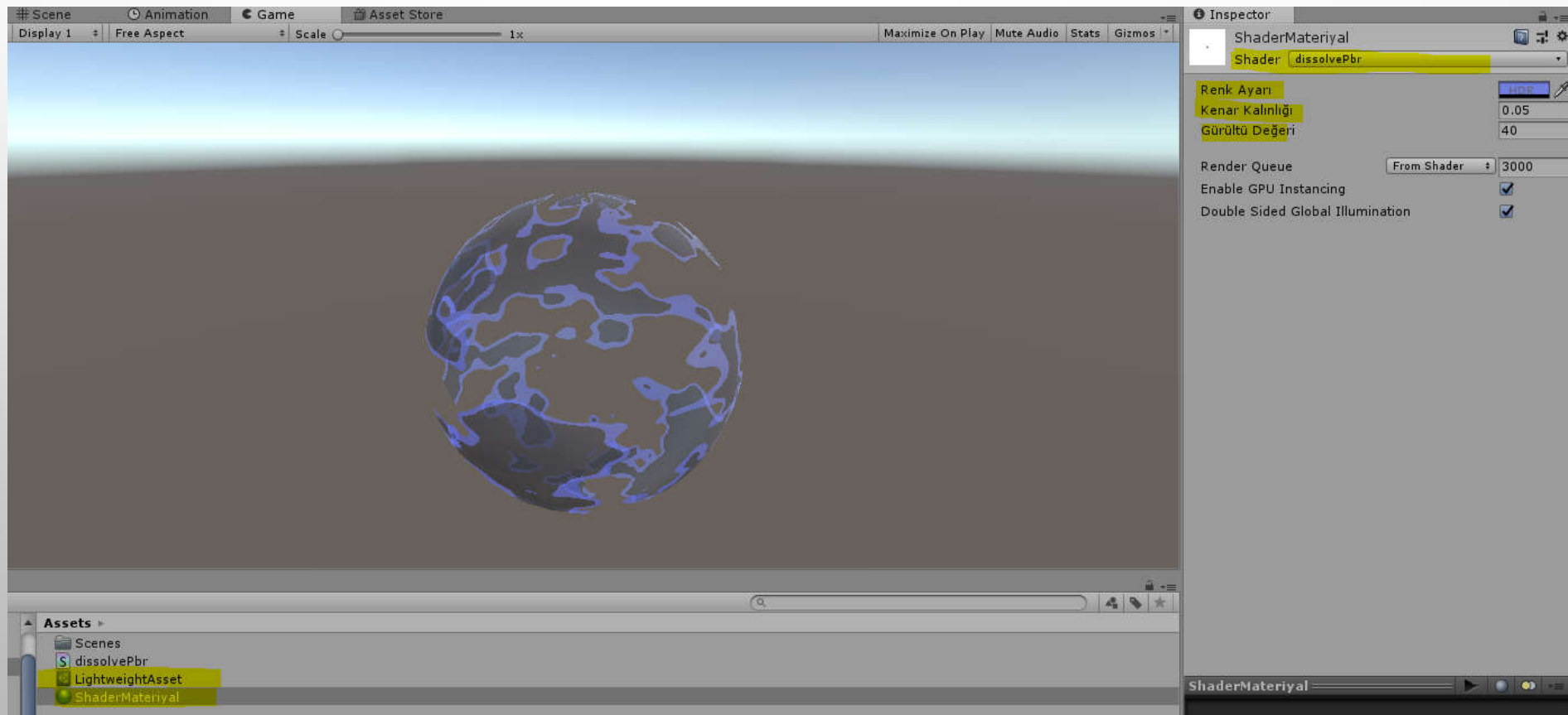


Two Sided



SHADER GRAPH (dissolve-Kenar Efektinin uygulanması)

Unity Editorde yeni bir materyal oluřturup. Bu materyalin Shader'ına oluřturduėumuz Shader sűrűklenir. Őzellikleri ile beraber Shader objeye uygulanır.



Installing ShaderGraph with Package Manager

<https://www.youtube.com/watch?v=CDbt6u4S2NE>

DISSOLVE using Unity Shader Graph

https://www.youtube.com/watch?v=taMp1g1pBeE&list=RDQM3Dnct0HQVZQ&start_radio=1

<https://killertee.wordpress.com/2018/02/28/experiments-with-unity-shader-graph-and-scriptable-render-pipeline/>

GRASS SWAY in Unity - SHADER GRAPH

https://www.youtube.com/watch?v=L_Bzcw9tqTc