Настроить шейдер перехода между двумя текстурами в Unity можно с помощью Shader Graph или вручную написанного HLSL кода. Я покажу тебе оба способа. В основном, для перехода между текстурами используется параметр, который контролирует смешивание, например blendFactor (значение от 0 до 1).

**Способ 1: Использование *Shader Graph***

Если ты работаешь с *Shader Graph*, тебе понадобится создать новый Shader Graph, а затем настроить узлы (Nodes) следующим образом:

1. **Создай Shader Graph:**
   * В Unity, щелкни правой кнопкой мыши в проекте и выбери **Create > Shader > PBR Graph** или **Unlit Graph** (в зависимости от того, нужен ли тебе освещение).
   * Дай ему имя и открой двойным щелчком.
2. **Настрой параметры шейдера:**
   * **Добавь два свойства текстуры**. Щелкни правой кнопкой мыши на пустом месте и выбери **Create Node > Property**:
     + Назови первую текстуру Texture1 и установи тип Texture2D.
     + Назови вторую текстуру Texture2 и установи тип Texture2D.
   * **Добавь параметр для смешивания**:
     + Создай Property с типом Float и назови его BlendFactor.
3. **Соедини узлы:**
   * Создай два узла Sample Texture2D и подключи Texture1 и Texture2 соответственно.
   * Используй узел **Lerp (Linear Interpolation)**. Это основа для смешивания двух текстур.
     + В поле **A** подключи выход RGBA от Sample Texture2D для Texture1.
     + В поле **B** подключи выход RGBA от Sample Texture2D для Texture2.
     + В поле **T** (вес) подключи параметр BlendFactor.
4. **Выходной результат:**
   * Подключи выход узла Lerp к Base Color для PBR Graph (или к Color для Unlit Graph).
5. **Настрой материал и управляй параметром:**
   * Создай материал на основе этого шейдера и назначь его объекту.
   * Теперь можно в Inspector изменять BlendFactor, чтобы регулировать смешивание между двумя текстурами.

**Способ 2: HLSL код (ShaderLab)**

Если ты предпочитаешь писать код вручную, то вот базовый пример на HLSL:

c

Копировать код

Shader "Custom/TextureBlend"

{

Properties

{

\_Texture1 ("Texture 1", 2D) = "white" {}

\_Texture2 ("Texture 2", 2D) = "white" {}

\_BlendFactor ("Blend Factor", Range(0, 1)) = 0.5

}

SubShader

{

Tags { "RenderType"="Opaque" }

LOD 100

Pass

{

CGPROGRAM

#pragma vertex vert

#pragma fragment frag

#include "UnityCG.cginc"

struct appdata

{

float4 vertex : POSITION;

float2 uv : TEXCOORD0;

};

struct v2f

{

float2 uv : TEXCOORD0;

float4 vertex : SV\_POSITION;

};

sampler2D \_Texture1;

sampler2D \_Texture2;

float \_BlendFactor;

v2f vert (appdata v)

{

v2f o;

o.vertex = UnityObjectToClipPos(v.vertex);

o.uv = v.uv;

return o;

}

fixed4 frag (v2f i) : SV\_Target

{

fixed4 tex1 = tex2D(\_Texture1, i.uv);

fixed4 tex2 = tex2D(\_Texture2, i.uv);

return lerp(tex1, tex2, \_BlendFactor);

}

ENDCG

}

}

}

**Объяснение кода:**

* lerp — это функция линейной интерполяции, которая плавно смешивает значения tex1 и tex2 на основе \_BlendFactor.
* \_BlendFactor регулирует, насколько сильно одна текстура заменяет другую (0 — первая текстура, 1 — вторая).

**Настройка в Unity:**

1. Создай шейдер, вставь код.
2. Создай новый материал на основе этого шейдера.
3. Перетащи материал на объект и в Inspector назначь две текстуры.

Меняй BlendFactor от 0 до 1, чтобы видеть переход.