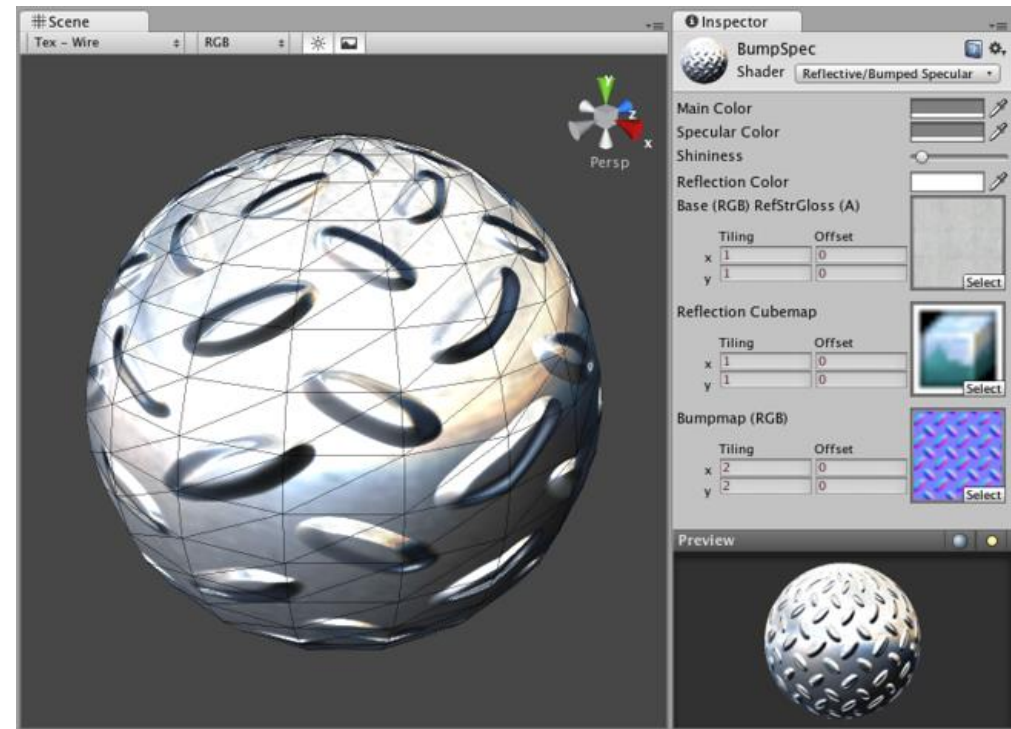


재질과 셰이더

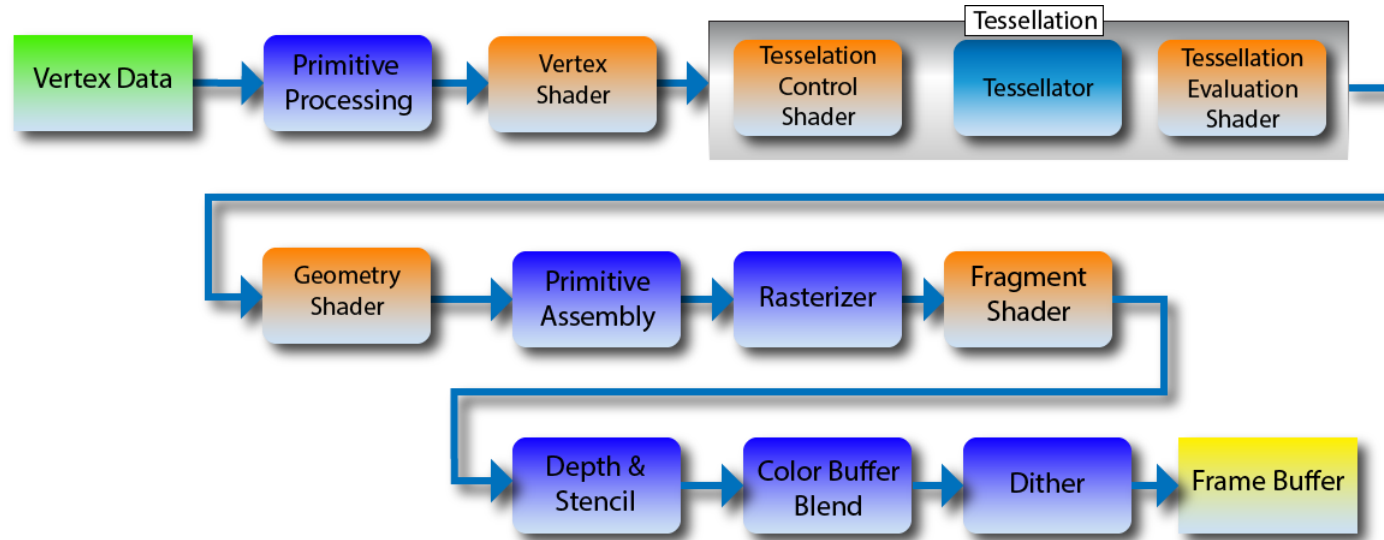
재질

- 재질(Material)
 - 화면에 표현될 게임 오브젝트의 렌더링 속성
 - 예) 나무, 스테인리스,
 - 셰이더에 프로그래밍된 내용에 의해 렌더링
 - 셰이더에서 사용할 애셋들과 셰이더 연산 시 필요한 속성 값들로 구성



셰이더

- 셰이더(Shader)
 - GPU가 렌더링 파이프라인을 거치는 동안 처리해야 할 일들을 프로그래밍하기 위한 스크립트



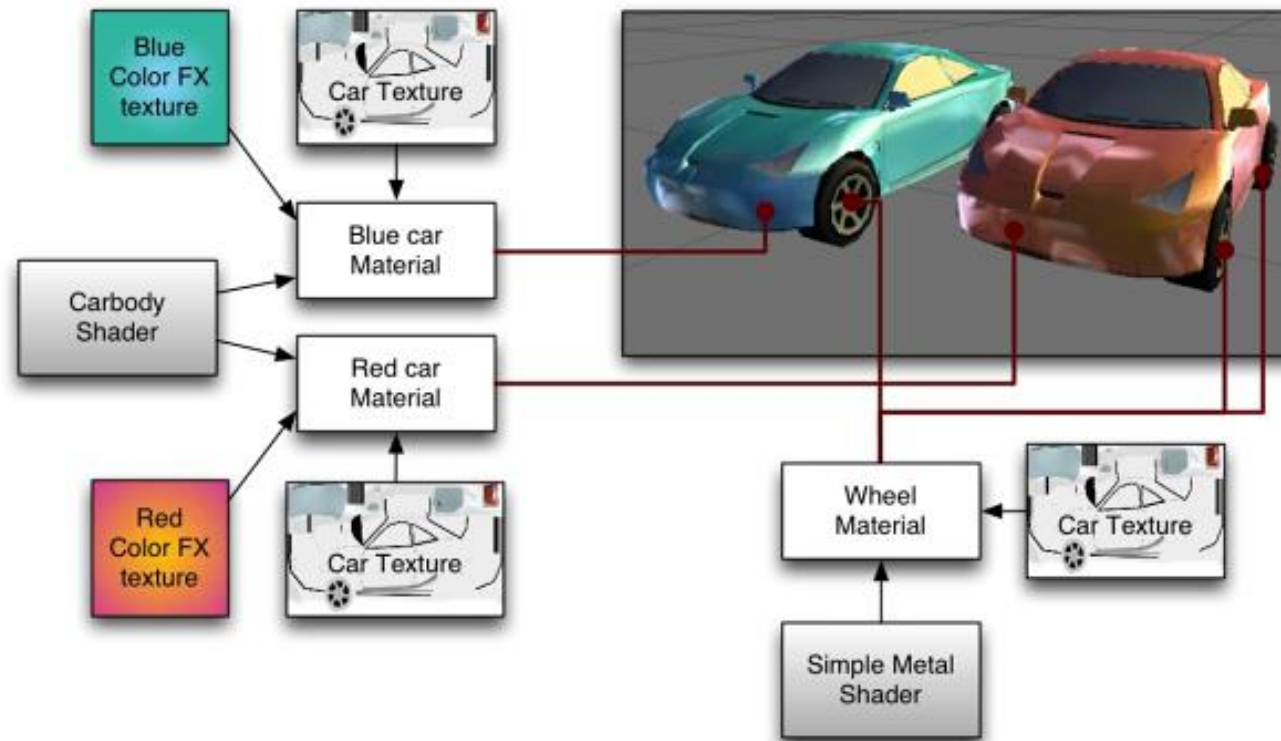
유니티 셰이더

- 유니티의 모든 객체들은 셰이더를 이용해 렌더링
- 유니티에서 사용되는 모든 재질은 셰이더를 사용
- 유니티의 셰이더는 "ShaderLab" 으로 작성 됨
 - Properties - 셰이더에서 사용을 설정 값
 - Subshaders - 실제 렌더링 파이프라인에서 처리할 명령 또는 연산
 - Fallback - Subshader를 수행할 수 없을 경우 해야할 처리
 - CustomEditor - 따로 셰이더 전용 UI를 사용할 경우

```
Shader "name"  
{  
    [Properties]  
    Subshaders  
    [Fallback]  
    [CustomEditor]  
}
```

ShaderLab 문법

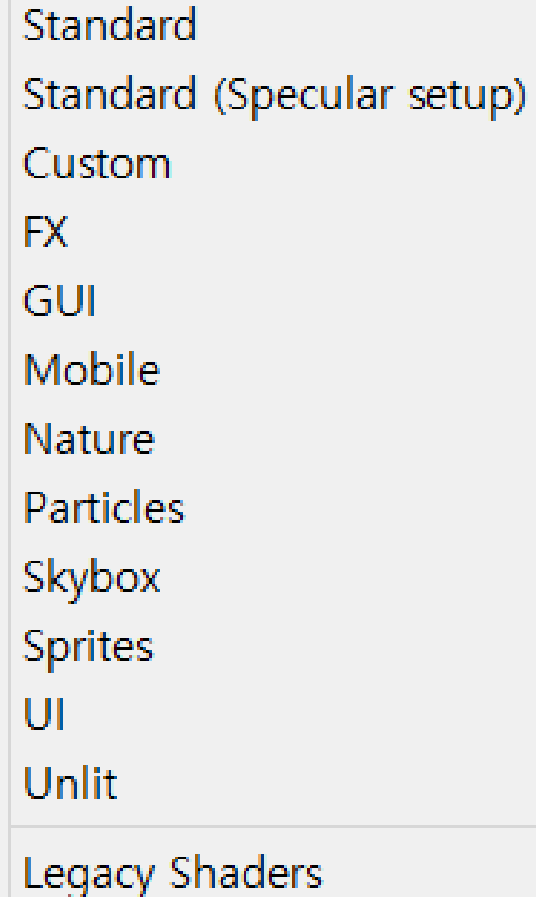
재질과 셰이더



셰이더를 이용한 재질의 적용과정

유니티 내장 셰이더

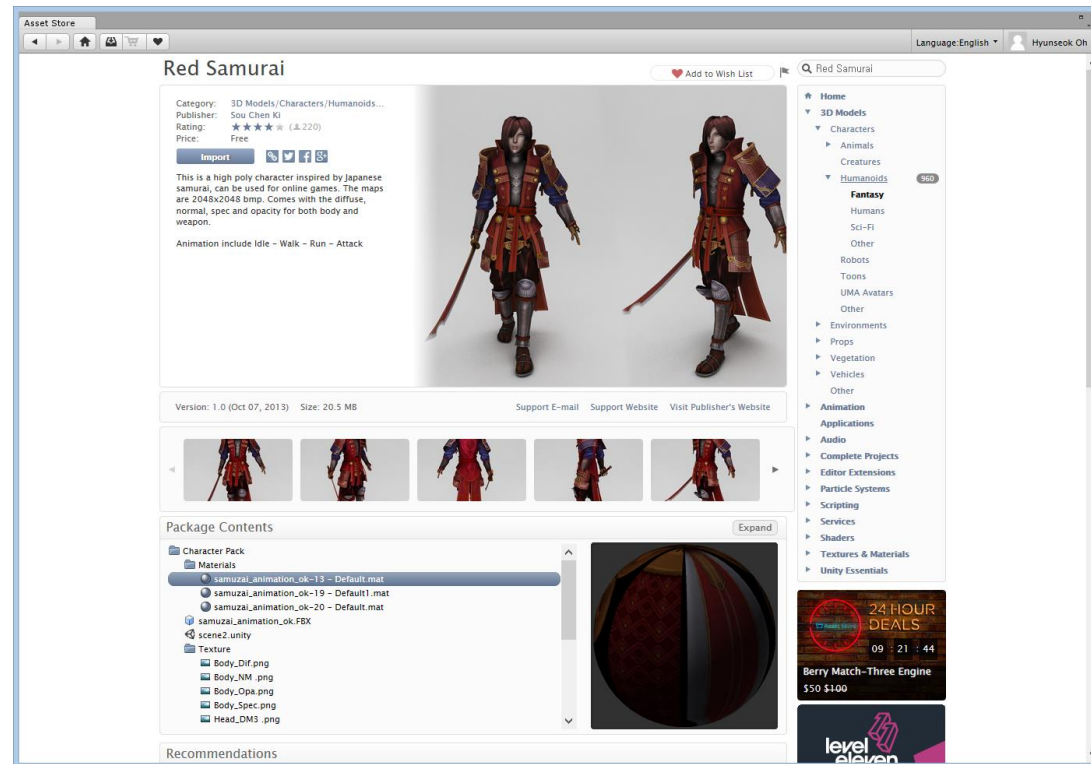
- 유니티는 다수의 내장 셰이더들을 제공한다.
- 사용될 대상의 종류에 따라 새롭게 분류
- Legacy Shaders
 - 이전 버전에서 사용하던 셰이더 모음



Standard	
Standard (Specular setup)	
Custom	▶
FX	▶
GUI	▶
Mobile	▶
Nature	▶
Particles	▶
Skybox	▶
Sprites	▶
UI	▶
Unlit	▶
Legacy Shaders	▶

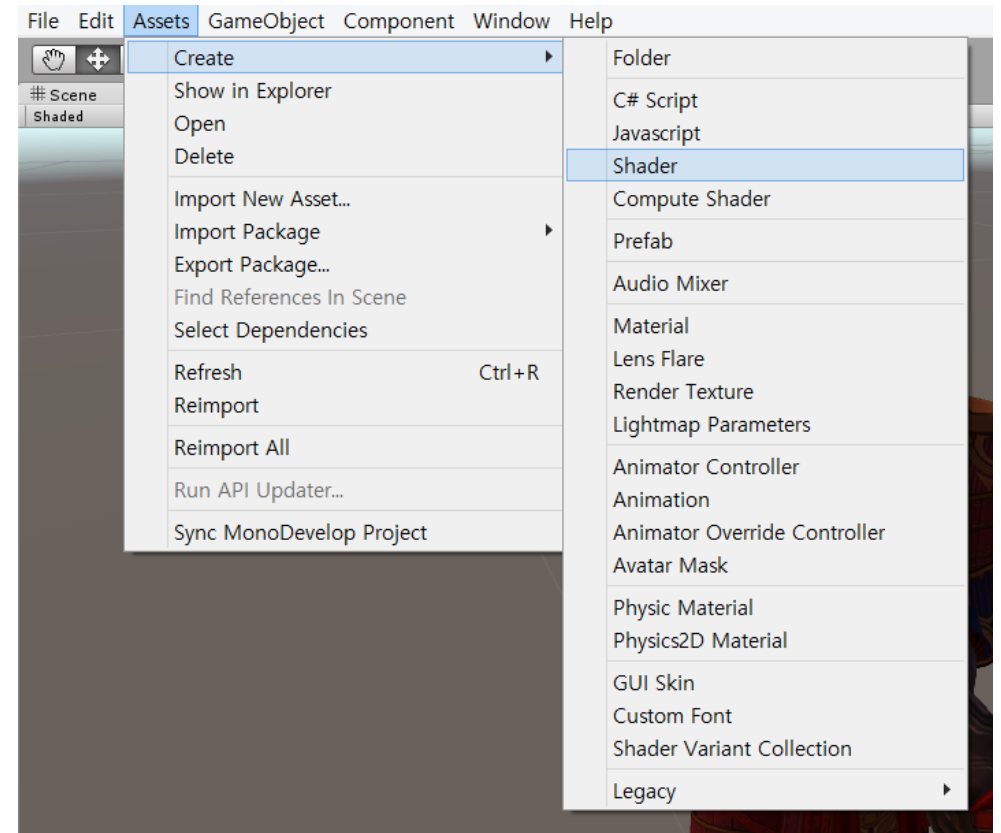
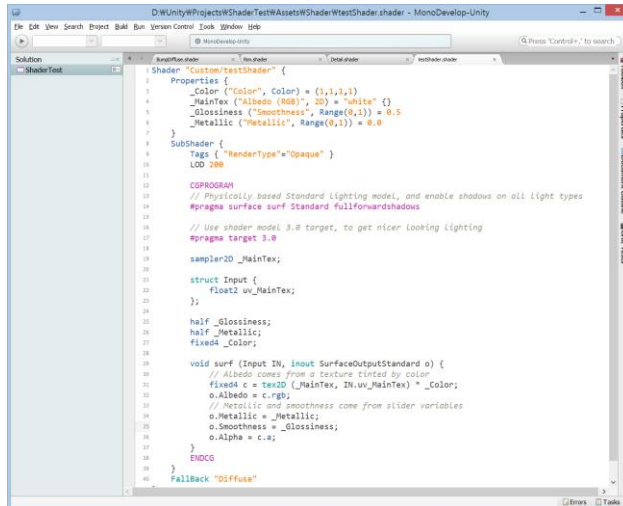
유니티 셰이더의 실습

- 에셋 스토어에서 “Red Samrai” 검색 후 Import



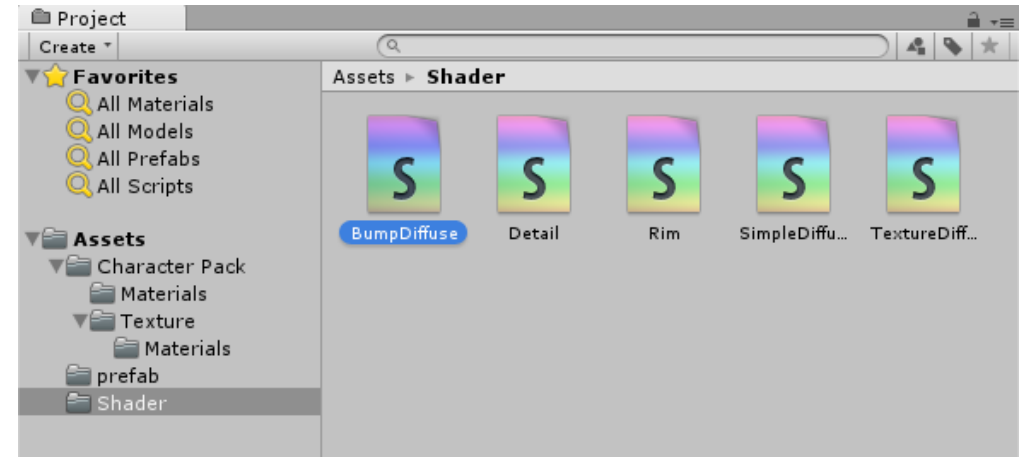
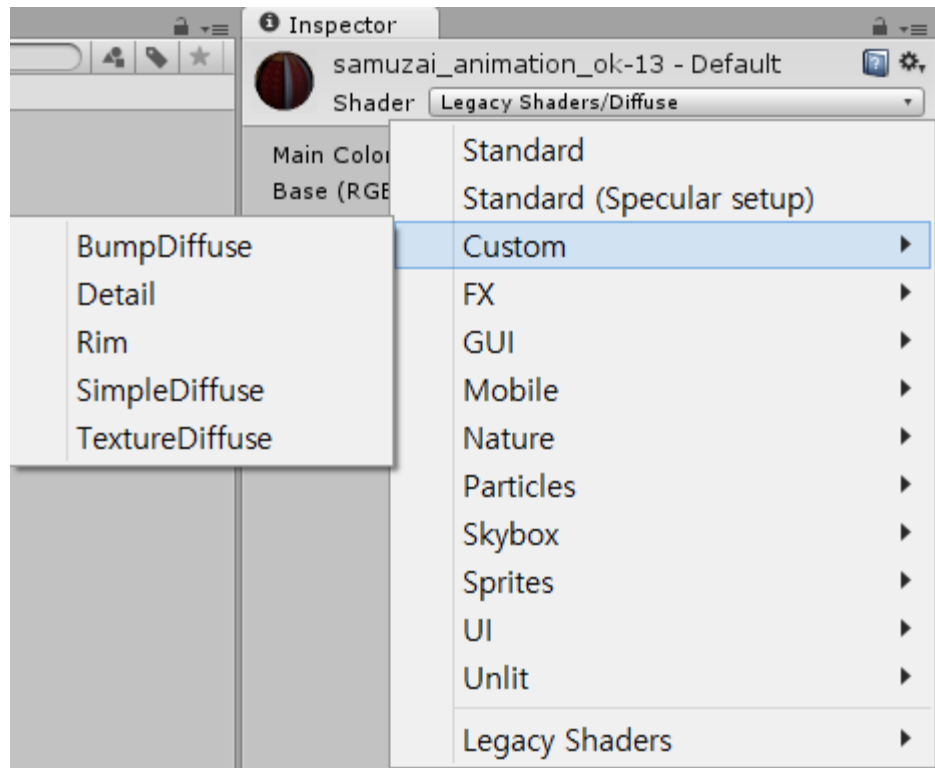
셰이더의 작성

- [메뉴 > Assets > Create > Shader]
- 프로젝트 뷰에서 생성된 셰이더 스크립트 더블클릭
- Mono Develop에서 작성



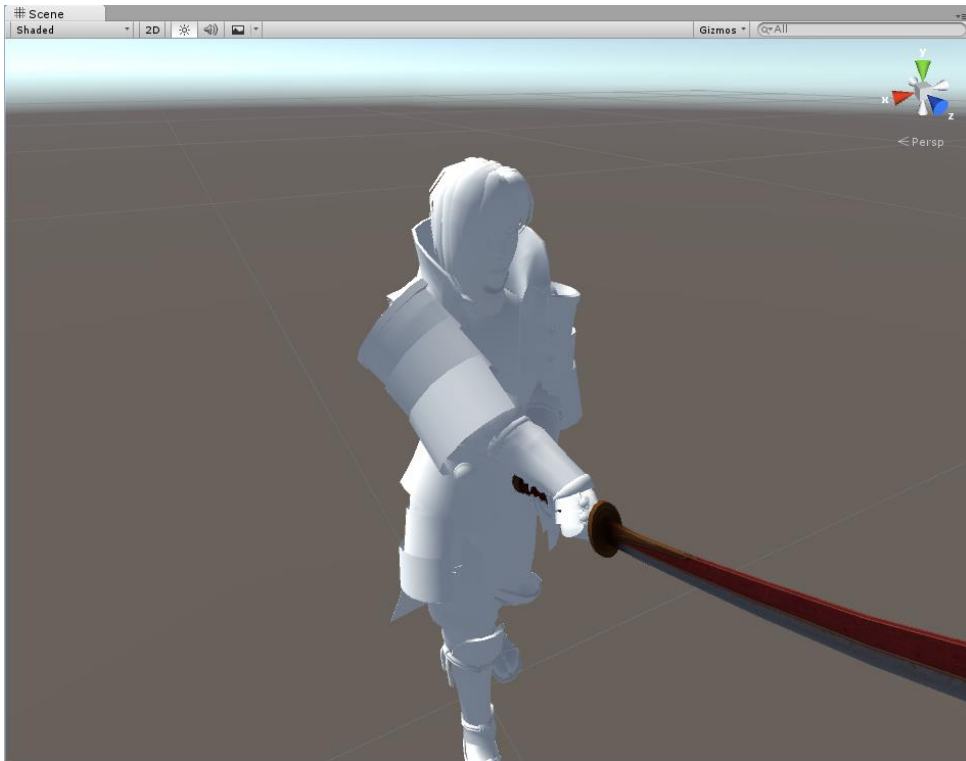
셰이더의 적용

- 작성한 셰이더는 재질의 Custom에서 선택할 수 있다.



셰이더 살펴보기

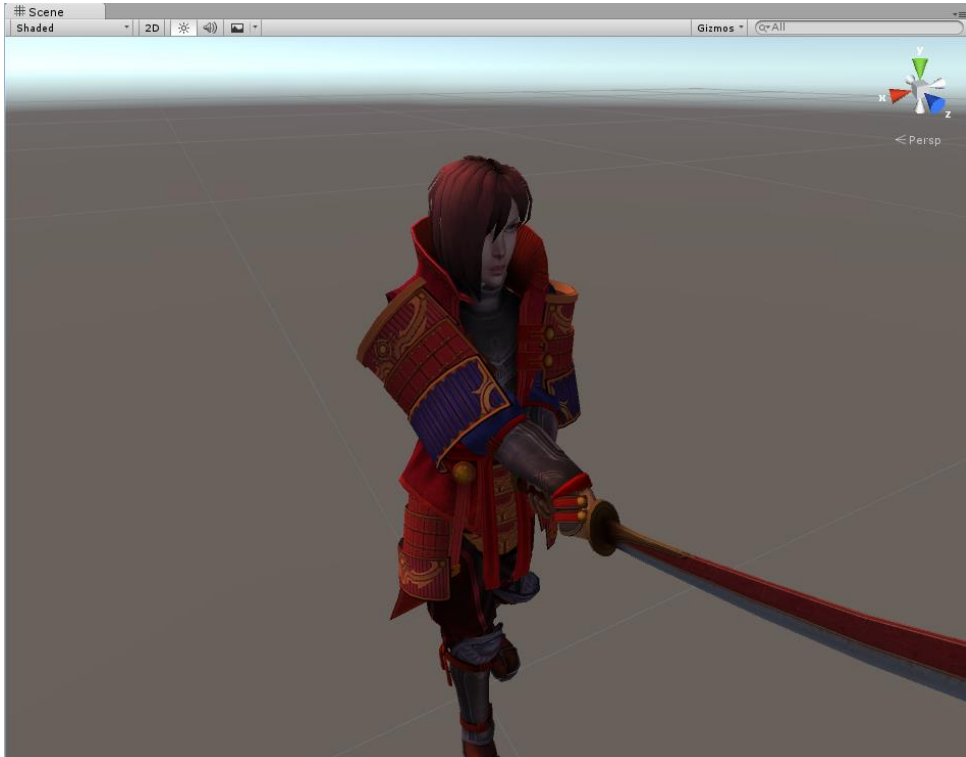
- SimpleDiffuse



```
1 Shader "Custom/SimpleDiffuse" {
2     SubShader {
3         Tags { "RenderType"="Opaque" }
4
5         CGPROGRAM
6         #pragma surface surf Lambert
7         struct Input {
8             float4 color : COLOR;
9         };
10
11         void surf (Input IN, inout SurfaceOutput o) {
12             o.Albedo = 1;
13         }
14         ENDCG
15     }
16     FallBack "Diffuse"
17 }
```

셰이더 살펴보기

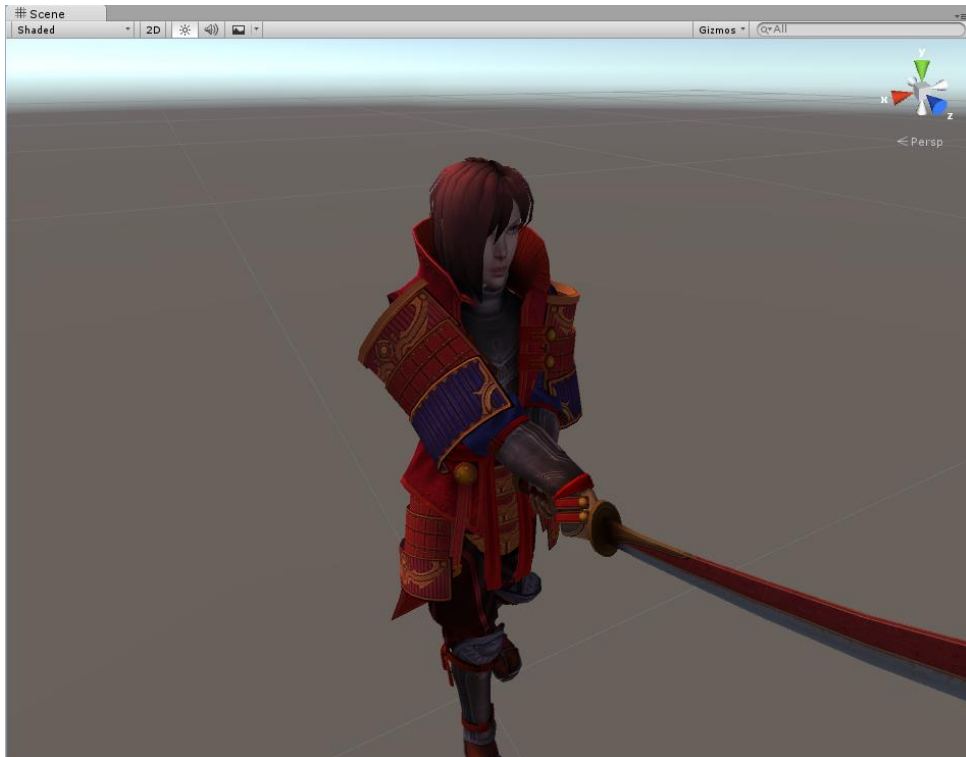
- TextureDiffuse



```
1 Shader "Custom/TextureDiffuse" {
2     Properties {
3         _MainTex ("Texture", 2D) = "white" {}
4     }
5     SubShader {
6         Tags { "RenderType"="Opaque" }
7
8         CGPROGRAM
9         #pragma surface surf Lambert
10
11         sampler2D _MainTex;
12
13         struct Input {
14             float2 uv_MainTex;
15         };
16         void surf (Input IN, inout SurfaceOutput o) {
17             o.Albedo = tex2D (_MainTex, IN.uv_MainTex).rgb;
18         }
19         ENDCG
20     }
21     FallBack "Diffuse"
22 }
```

셰이더 살펴보기

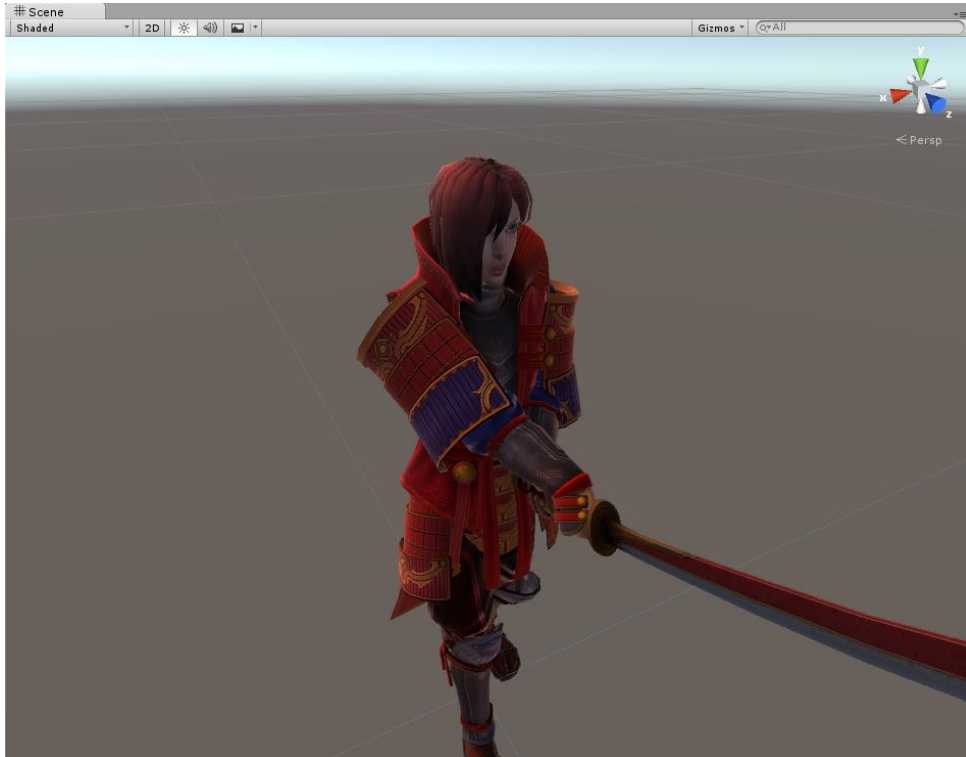
- BumpDiffuse



```
1 Shader "Custom/BumpDiffuse" {
2   Properties {
3     _MainTex ("Texture", 2D) = "white" {}
4     _BumpMap ("Bumpmap", 2D) = "bump" {}
5   }
6   SubShader {
7     Tags { "RenderType"="Opaque" }
8
9     CGPROGRAM
10    #pragma surface surf Lambert
11
12    sampler2D _MainTex;
13    sampler2D _BumpMap;
14
15    struct Input {
16      float2 uv_MainTex;
17      float2 uv_BumpMap;
18    };
19
20    void surf (Input IN, inout SurfaceOutput o) {
21      o.Albedo = tex2D (_MainTex, IN.uv_MainTex).rgb;
22      o.Normal = UnpackNormal(tex2D(_BumpMap, IN.uv_BumpMap));
23    }
24    ENDCG
25  }
26  FallBack "Diffuse"
27 }
```

셰이더 살펴보기

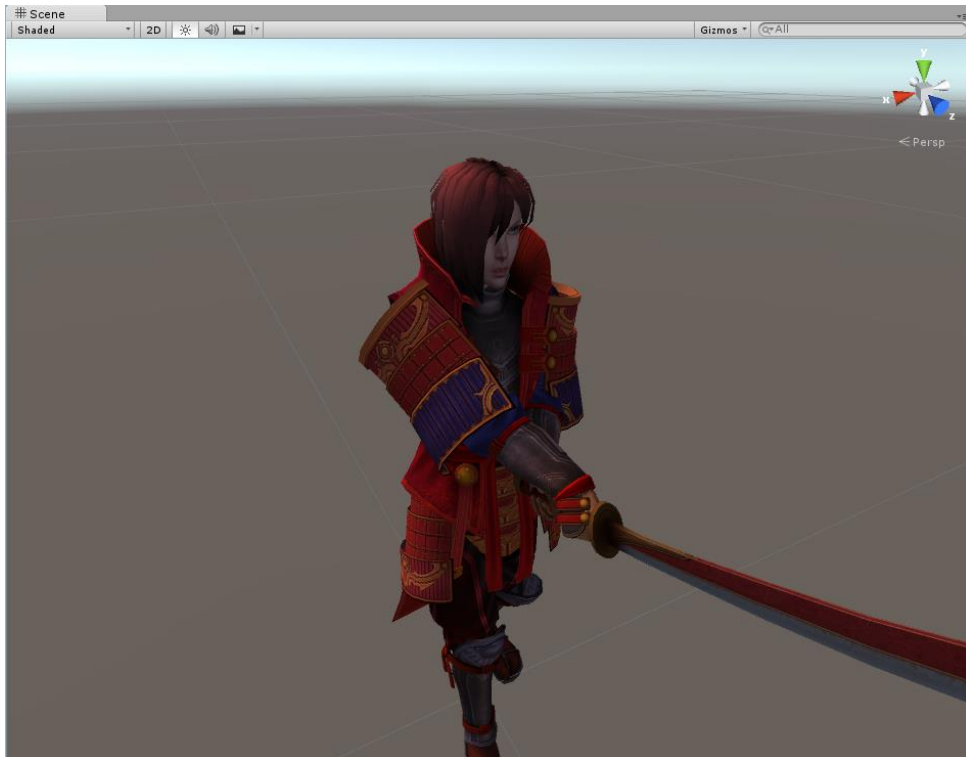
- Rim



```
1 Shader "Custom/Rim" {
2   Properties {
3     _MainTex ("Texture", 2D) = "white" {}
4     _BumpMap ("Bumpmap", 2D) = "bump" {}
5     _RimColor ("Rim Color", Color) = (0.26, 0.19, 0.16, 0.0)
6     _RimPower ("Rim Power", Range(0.5, 8.0)) = 3.0
7   }
8   SubShader {
9     Tags { "RenderType"="Opaque" }
10
11     CGPROGRAM
12     #pragma surface surf Lambert
13
14     sampler2D _MainTex;
15     sampler2D _BumpMap;
16     float4 _RimColor;
17     float _RimPower;
18
19     struct Input {
20       float2 uv_MainTex;
21       float2 uv_BumpMap;
22       float3 viewDir;
23     };
24
25     void surf (Input IN, inout SurfaceOutput o) {
26       o.Albedo = tex2D (_MainTex, IN.uv_MainTex).rgb;
27       o.Normal = UnpackNormal (tex2D (_BumpMap, IN.uv_BumpMap));
28       half rim = 1.0 - saturate(dot (normalize(IN.viewDir), o.Normal));
29       o.Emission = _RimColor.rgb * pow(rim, _RimPower);
30     }
31     ENDCG
32   }
33   FallBack "Diffuse"
34 }
```

셰이더 살펴보기

- Detail



TextureDiffuse



Detail

```
1 Shader "Custom/Detail" {
2   Properties {
3     _MainTex ("Texture", 2D) = "white" {}
4     _BumpMap ("Bumpmap", 2D) = "bump" {}
5     _Detail ("Detail", 2D) = "gray" {}
6   }
7
8   SubShader {
9     Tags { "RenderType" = "Opaque" }
10    CGPROGRAM
11    #pragma surface surf Lambert
12    struct Input {
13      float2 uv_MainTex;
14      float2 uv_BumpMap;
15      float2 uv_Detail;
16    };
17    sampler2D _MainTex;
18    sampler2D _BumpMap;
19    sampler2D _Detail;
20    void surf (Input IN, inout SurfaceOutput o) {
21      o.Albedo = tex2D (_MainTex, IN.uv_MainTex).rgb;
22      o.Albedo *= tex2D (_Detail, IN.uv_Detail).rgb * 2;
23      o.Normal = UnpackNormal (tex2D (_BumpMap, IN.uv_BumpMap));
24    }
25    ENDCG
26  }
27  FallBack "Diffuse"
28 }
```