第十次实验： 着色器

学号：22920212204392 姓名：黄勖

一、 实验目的

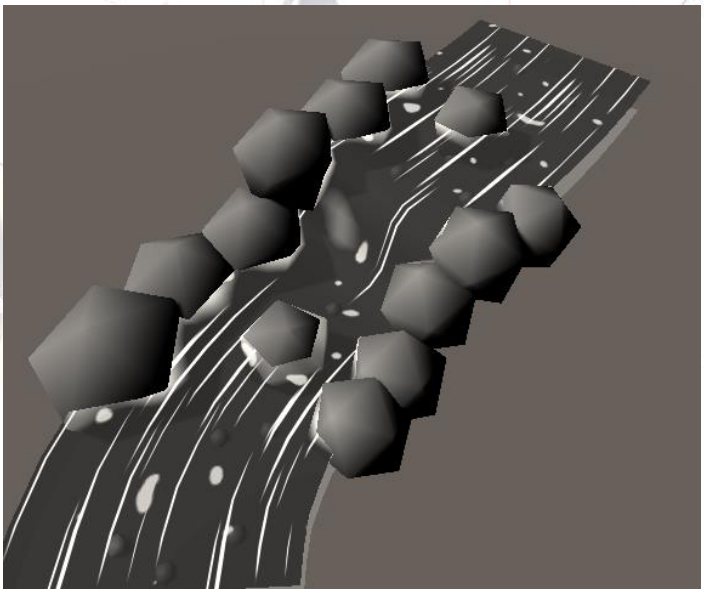
- 熟悉着色器编程
- 掌握 Shader Graph

二、 实验条件

- 系统环境：Windows 10 21H2
- 软件环境：Unity 3D 2021.3.14f1c1

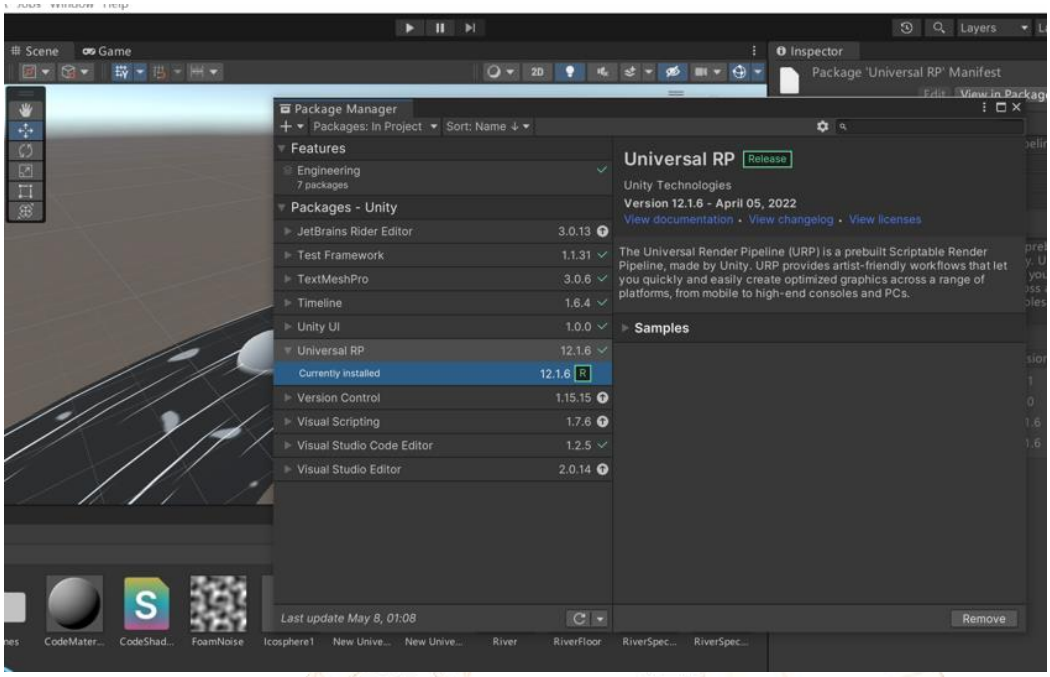
三、 实验内容

- **参考教程实现类似的流水效果**
- **https://www.ronja-tutorials.com/2018/11/03/river.html**

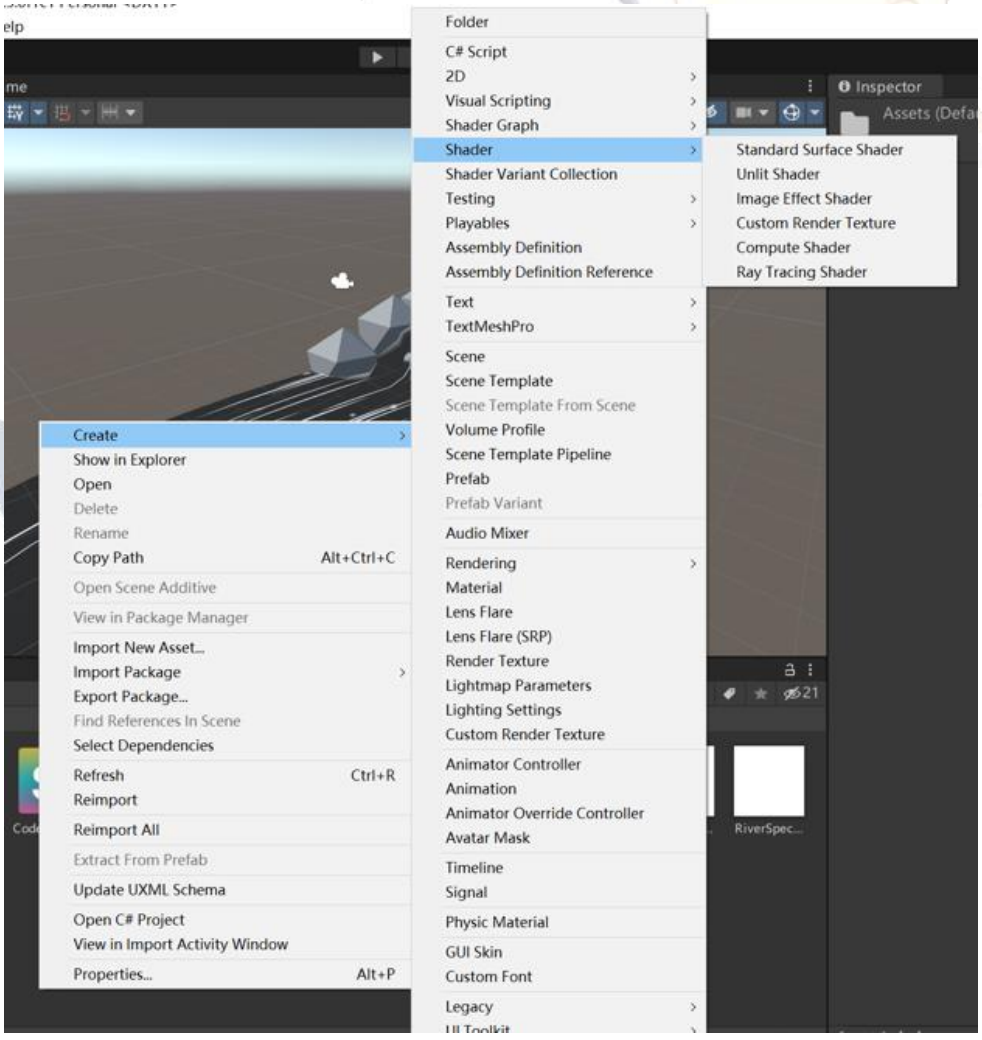


四、 实验项目步骤：

(一)搭建 shader 的环境：windows>package manager>Universal RP>import



（二）在 assets 中新建 CodeShader.shader



（三）编写 shader 脚本

CodeShader.shader

Assets > CodeShader.shader

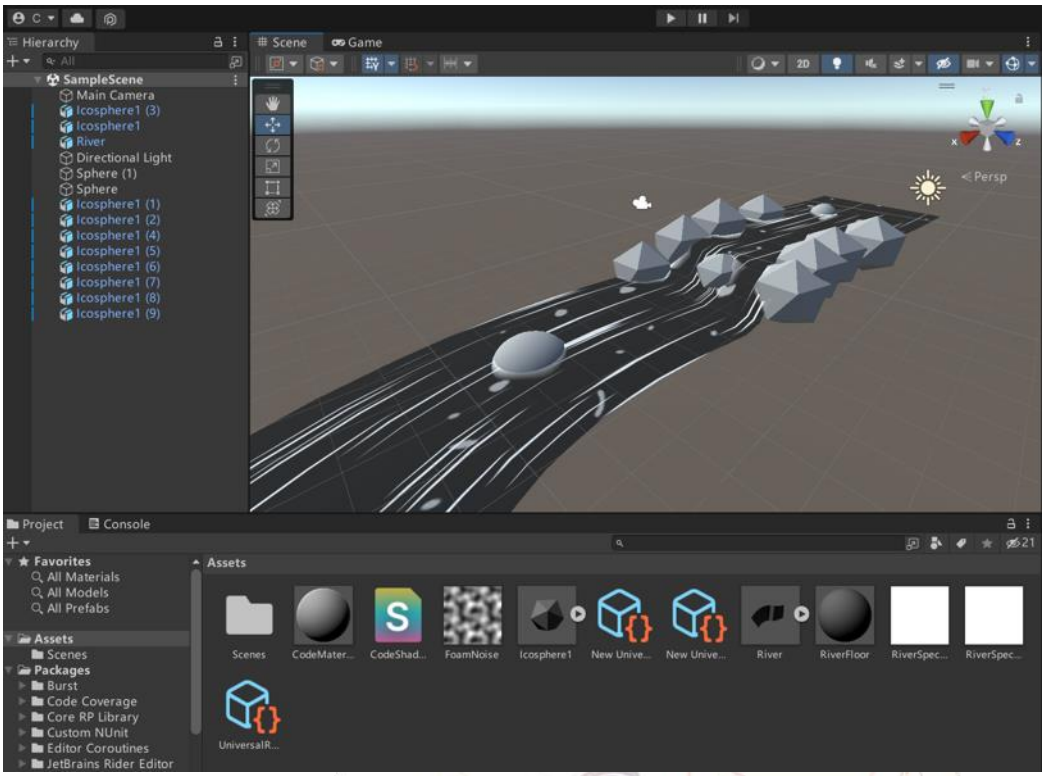
```
Shader "Unlit/CodeShader"
{
    Properties{
        _Color("Base Color", Color) = (1,1,1,1)

        [Header(Spec Layer 1)]
        _Specs1("Specs", 2D) = "white" {}
        _SpecColor1("Spec Color", Color) = (1,1,1,1)
        _SpecDirection1("Spec Direction", Vector) = (0, 1, 0, 0)

        [Header(Spec Layer 2)]
        _Specs2("Specs", 2D) = "white" {}
        _SpecColor2("Spec Color", Color) = (1,1,1,1)
        _SpecDirection2("Spec Direction", Vector) = (0, 1, 0, 0)

        [Header(Foam)]
        _FoamNoise("Foam Noise", 2D) = "white" {}
        _FoamDirection("Foam Direction", Vector) = (0, 1, 0, 0)
        _FoamColor("Foam Color", Color) = (1,1,1,1)
        _FoamAmount("Foam Amount", Range(0, 2)) = 1
    }
```

```
    SubShader{
        Tags { "RenderType" = "Transparent" "Queue" = "Transparent" "ForceNoShadowCasting" = "True"}
        LOD 200

        CGPROGRAM
        // Physically based Standard lighting model, and enable shadows on all light types, then set it to render transparent
        #pragma surface surf Standard vertex:vert fullforwardshadows alpha

        #pragma target 4.0

        struct Input {
            float2 uv_Specs1;
            float2 uv_Specs2;
            float2 uv_FoamNoise;
            float eyeDepth;
            float4 screenPos;
        };

        sampler2D_float _CameraDepthTexture;

        fixed4 _Color;

        sampler2D _Specs1;
        fixed4 _SpecColor1;
        float2 _SpecDirection1;

        sampler2D _Specs2;
        fixed4 _SpecColor2;
        float2 _SpecDirection2;

        sampler2D _FoamNoise;
        fixed4 _FoamColor;
        float _FoamAmount;
        float2 _FoamDirection;
```

```
        void vert(inout appdata_full v, out Input o)
        {
            UNITY_INITIALIZE_OUTPUT(Input, o);
            COMPUTE_EYEDEPTH(o.eyeDepth);
        }

        void surf(Input IN, inout SurfaceOutputStandard o) {
            //set river base color
            fixed4 col = _Color;

            //add first layer of moving specs
            float2 specCoordinates1 = IN.uv_Specs1 + _SpecDirection1 * _Time.y;
            fixed4 specLayer1 = tex2D(_Specs1, specCoordinates1) * _SpecColor1;
            col.rgb = lerp(col.rgb, specLayer1.rgb, specLayer1.a);
            col.a = lerp(col.a, 1, specLayer1.a);

            //add second layer of moving specs
            float2 specCoordinates2 = IN.uv_Specs2 + _SpecDirection2 * _Time.y;
            fixed4 specLayer2 = tex2D(_Specs2, specCoordinates2) * _SpecColor2;
            col.rgb = lerp(col.rgb, specLayer2.rgb, specLayer2.a);
            col.a = lerp(col.a, 1, specLayer2.a);

            //get scene and surface depth
            float4 projCoords = UNITY_PROJ_COORD(IN.screenPos);
            float rawZ = SAMPLE_DEPTH_TEXTURE_PROJ(_CameraDepthTexture, projCoords);
            float sceneZ = LinearEyeDepth(rawZ);
            float surfaceZ = IN.eyeDepth;

            //add foam
            float2 foamCoords = IN.uv_FoamNoise + _FoamDirection * _Time.y;
            float foamNoise = tex2D(_FoamNoise, foamCoords).r;
            float foam = 1 - ((sceneZ - surfaceZ) / _FoamAmount);
            foam = saturate(foam - foamNoise);
            col.rgb = lerp(col.rgb, _FoamColor.rgb, foam);
            col.a = lerp(col.a, 1, foam * _FoamColor.a);

            //apply values to output struct
            o.Albedo = col.rgb;
            o.Alpha = col.a;
        }
        ENDCG
    }
    FallBack "Diffuse"
}
```

（四）在 assets 中新建 material。命名为 CodeMaterial，在 Shader 中选择刚编辑好的 CodeShader。修改 CodeMaterial 的参数值，并将不同材料图片拖入对应 Layer 中

（五）将 Material 拖动到 River 上，并仿照最终效果图添加石块



此时即可实现实验所要求的全部内容!

最终效果详见视频演示

五、 实验心得总结:

这次实验我学习了通过着色器编程的方法和 Shader Graph——通过图而非编程方式实现 Shader 渲染。应当注意它们的底层逻辑相似，即 shader 建立在 material 的基础上，material 建立在 3D 物体上。。