In this assignment, I researched how to simulate the transparency of the water.

Firstly, after you creating the your shader, you should change the “Tag”, which is about the render queue.

Part A:
There are some frequently used default value.

1. Background: value is 1000, which can be used for skybox.
2. Geometry: value is 1000, which can be used for the objects according to the distance between object and camera.
3. Transparent: value is 1000, which can be used for the objects for the transparent objects.
4. Overlay: value is 4000, which can be used for the camera.

So, we should change the default shader as “Tags { "Queue" = "Transparent" }”

Then, we should consider one problem, how we get the depth of the water. It will decide the color and the transparency of the water.

In order to deal with it, we should set “ZWrite Off”. It controls whether pixels from this object are written to the depth buffer(default is On). Because we want to draw semitransparent effects, so we will switch ZWrite to Off.

Then, I will use “GrabPass{}”to get the mapping texture of the temporary screen. We will use it to get the rate of the riverbed and the water surface.

After that, I will simulate the flowing water, some details you can get it from my last tutorial. I will explain some changes and more new points below.

Part B:
In function “vert”, I will get the distance between water surface and the camera.

```
o.scrPos = ComputeScreenPos(o.vertex);
COMPUTE_EYEDEPTH(o.scrPos.z);
```

“ComputeScreenPos” will compute the position of every pixel from solid space to the 2D position on screen.

“COMPUTE_EYEDEPTH” will get the depth between every pixel and camera.

Part C:
In function “frag”, first, I will get the distance between riverbed and the camera.

```
float bedDepth = LinearEyeDepth(tex2Dproj(_CameraDepthTexture,
UNITY_PROJ_COORD(i.scrPos)).r)
```

“_CameraDepthTexture” is a sample2D variable of depth in shader.

“UNITY_PROJ_COORD” will get the position of every pixel on screen.

“tex2Dproj” will get the depth of every pixel in solid space.

“LinearEyeDepth” will convert the depth in solid space to the depth from the camera.

```
float depth= bedDepth- i.scrPos.z;
```

“depth” is the depth of water.

```
float4 groundcol = tex2Dproj(_GrabTexture, UNITY_PROJ_COORD(i.scrPos));
float4 col1 = tex2D(_MainTex, uv + flowSpeed + noise * float2(sin(_Time.y *
speed.x), sin(_Time.y * speed.y))*_WaveScale);
float4 col2 = tex2D(_WaveTex, i.uv2 + flowSpeed*0.2 + noise * float2(sin(_Time.y *
speed.x), sin(_Time.y * speed.y))*_WaveScale + _WaveStrength *
uvDir*sin(_Time.y*_TimeScale + dis * _WaveFactor) / 5 / dis );
```

“groundcol” is the background.

“col1” is to simulate the flowing water.

“col2” is to simulate the spray near the side.

```
float4 temp = lerp(col1, groundcol, colDepth);
```

“temp” is to compute the transparency of the water according to “colDepth”, which is variable about the depth of water. Specific procedure to calculate is in the program.

```
temp2.rgb= lerp(groundcol.rgb, col2.rgb, alpha);
temp2.w = col2.w;
col2 = temp2;
```

“temp2” is to compute the spray range according to the alpha, which can be changed in Unity 3D.

```
float4 col = lerp(temp, col2, colDepth);
```

“col” is the final output of the function “frag”.