Fluid Animation Experiment Report Using Obi Fluid Plugin in Unity

Experimenter: Chen Yiwen, Class 2, Digital Media 18

Experiment Content: Implement fluid animation effects using the Obi Fluid plugin

Experiment Environment: Unity 2018f4.2

After searching for information online, I found that the fluid effects simulated by the Obi Fluid plugin are relatively natural. So I decided to try making a small demo with this plugin to experience it.

Experiment Steps

1. Set up the scene

After downloading Obi Fluid, import it into Unity and open it. Right-click in the blank area of the Hierarchy menu and select 3D Object / Obi / Obi Emitter (with solver).

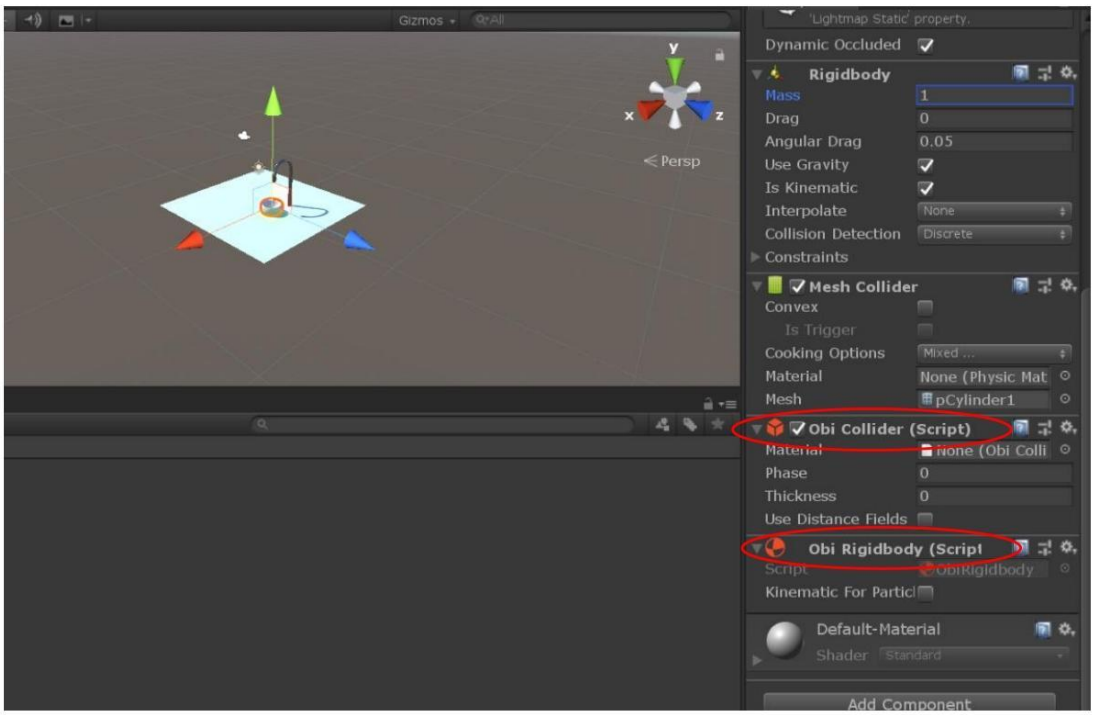
At this time, an object with the following four components will be created:

- Obi Emitter: Fluid emitter;
- Obi Emitter Shape Disk: Shape of the fluid emitter;
- Obi Particle Renderer: Fluid particle renderer;
- Obi Solver: Fluid solution.

Copy the basin and faucet models from the built-in scene of the Obi package to the newly created scene, and add a plane to build a simple faucet water flow scene.

2. Add components to the basin

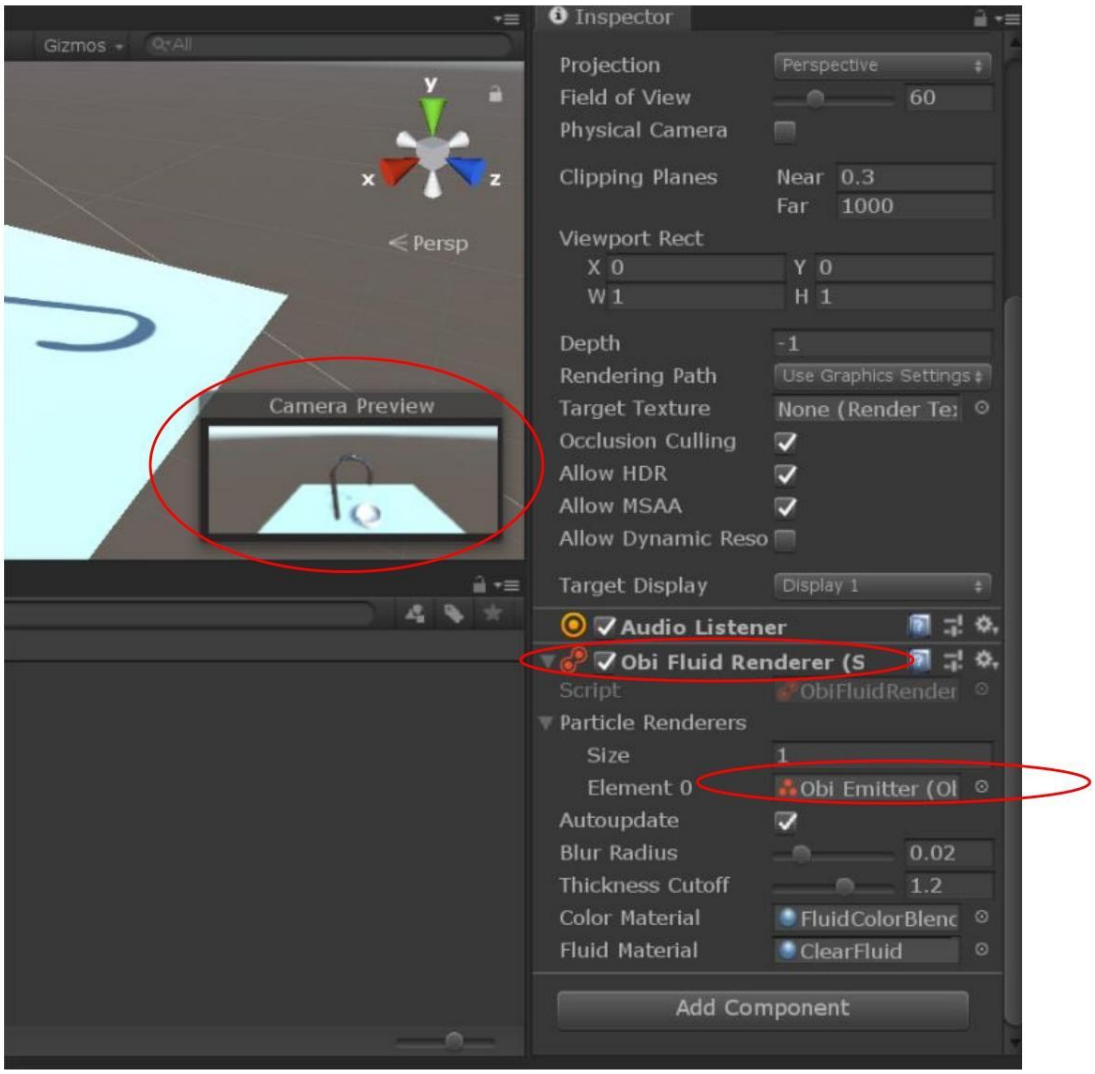
Add a Mesh Collider component to the basin and check Convex. Also, attach the Obi Collider and Obi Rigidbody components. If the Obi Collider is not added, the fluid will pass through the basin directly.



3. Adjust camera parameters

Click the play button at this time. You will see that the fluid effect is obviously in the form

of small balls. So, attach the Obi Fluid Renderer component to the camera and assign the Particle Renderers element to the Obi Emitter created above. Also, uncheck the Renderer option of the Obi Particle Renderer component on the Obi Emitter object. Additionally, adjust the position parameters of the camera to zoom in the camera lens. The effect is as follows:



4. Modify fluid color

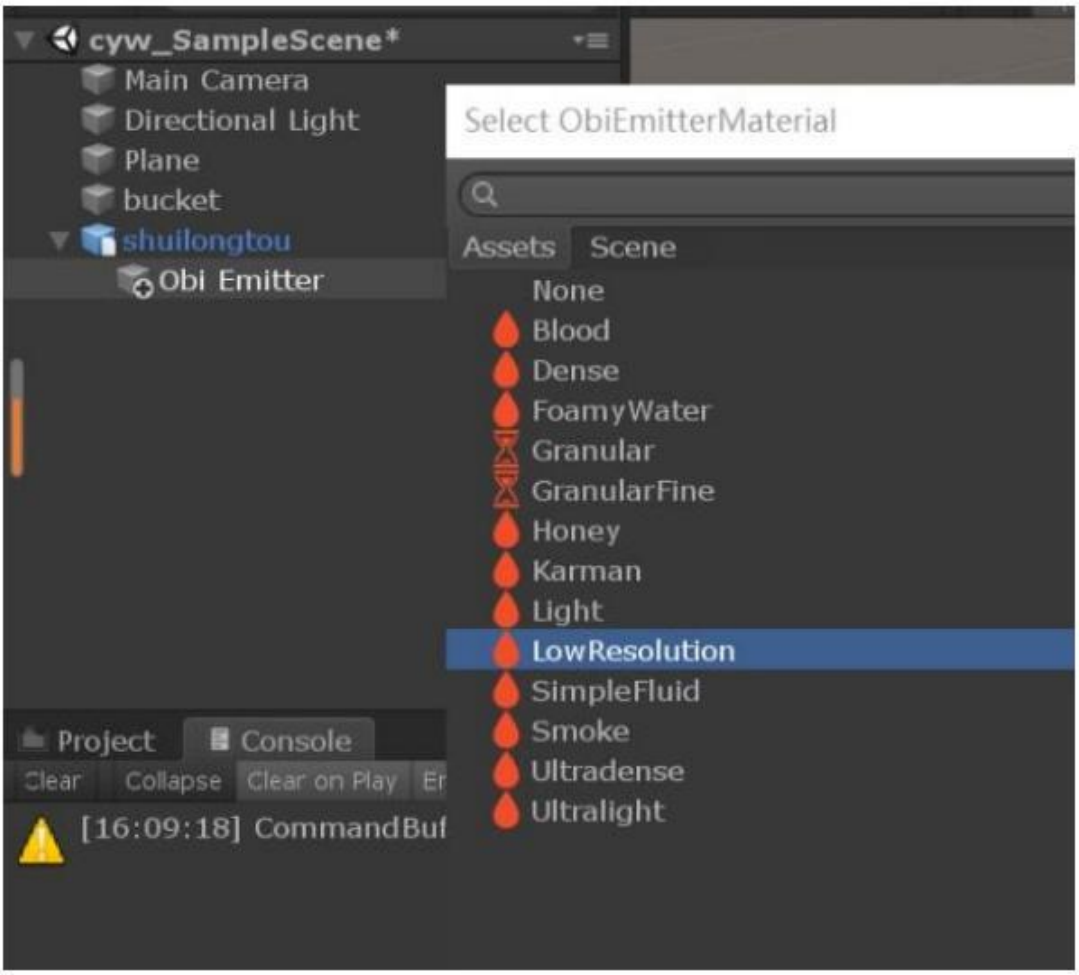
The fluid color can be modified through the Particle Color property of the Obi Particle Renderer.

5. Modify fluid particle size

The size of the fluid particles can be adjusted through the Radius Scale of the Obi Particle Renderer. A value of 1.5 is more appropriate.

6. Change fluid material

The fluid material can be set through the Emitter Material of the Obi Emitter component. Here, I chose Low Resolution.

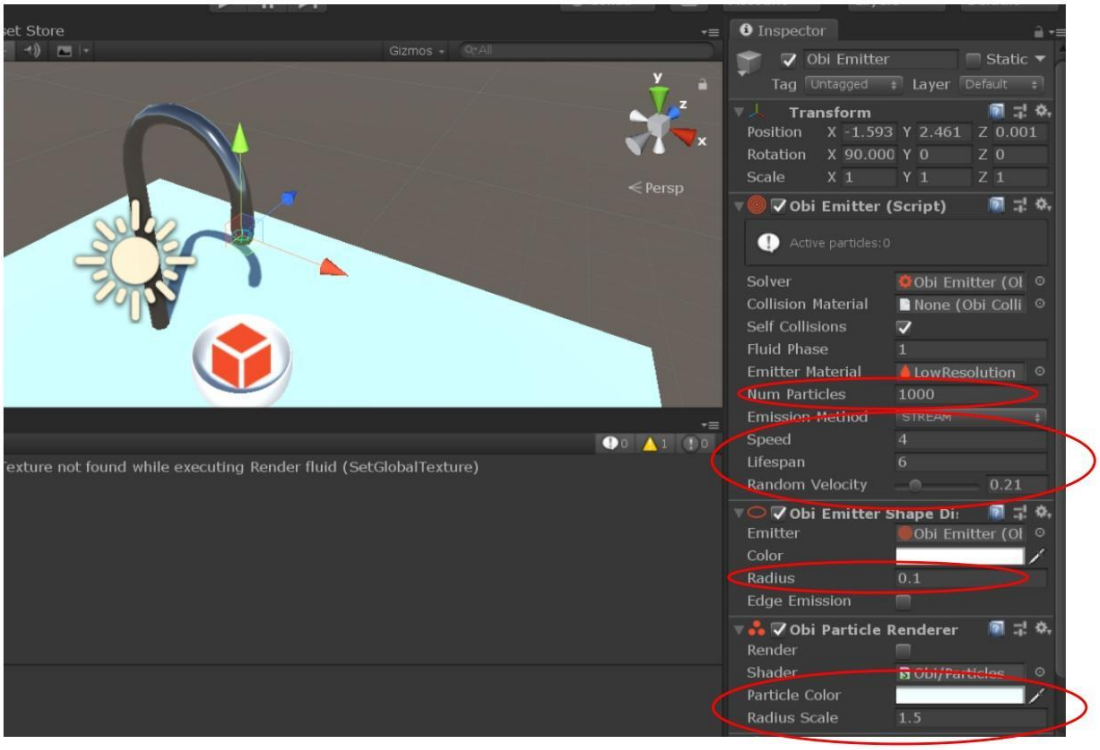


7. Adjust the nozzle

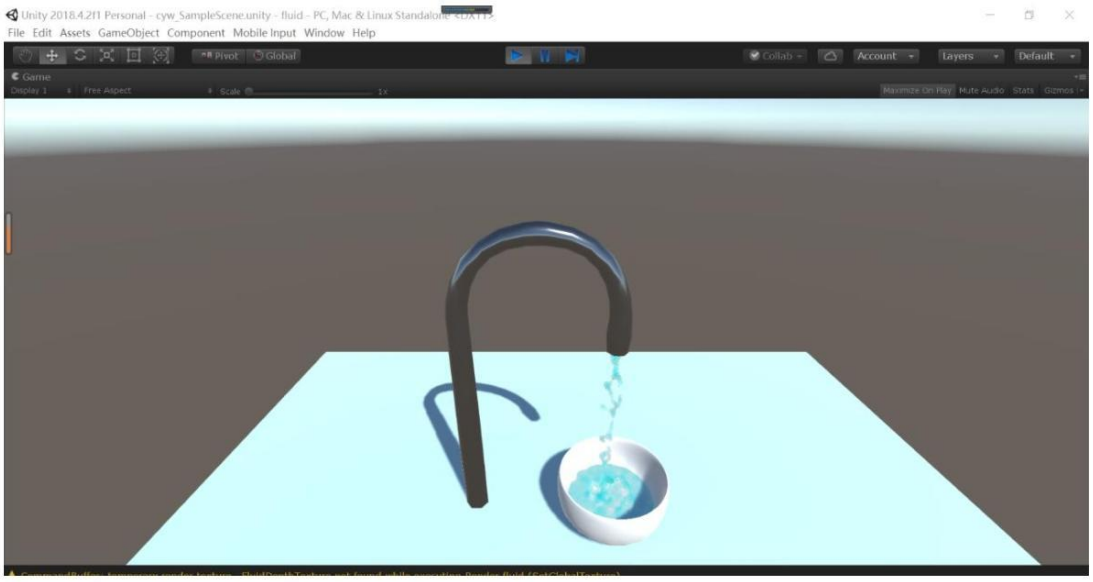
Select the Obi Emitter. You can see the shape of the nozzle in the Scene window. It is like a shower head. Since the direction of the nozzle is not facing the basin at this time, adjust the angle of the Transform to make the direction of the nozzle point to the basin. And if the nozzle is a bit large, you can adjust the size of the nozzle through the Radius of the Obi Emitter Shape. A value of 0.1 is more in line with the faucet diameter.

8. Adjust the ejection speed

If the ejection speed is relatively slow, you can adjust the ejection speed through the Speed property of the Obi Emitter component. Set it to 4 at this time and give a random force Random Velocity. The unified adjusted parameters are as follows:

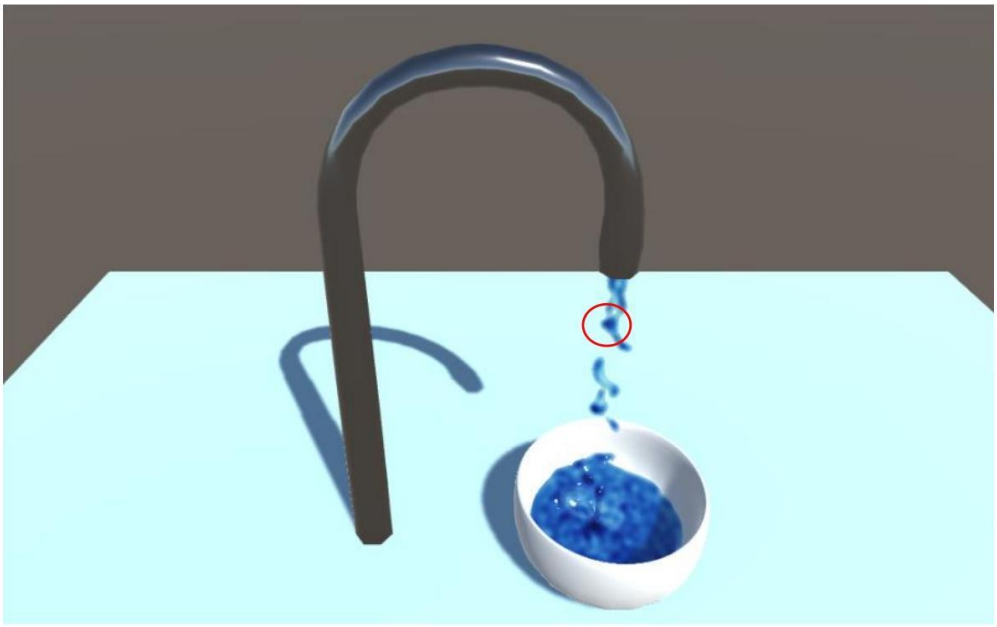


Experiment Results



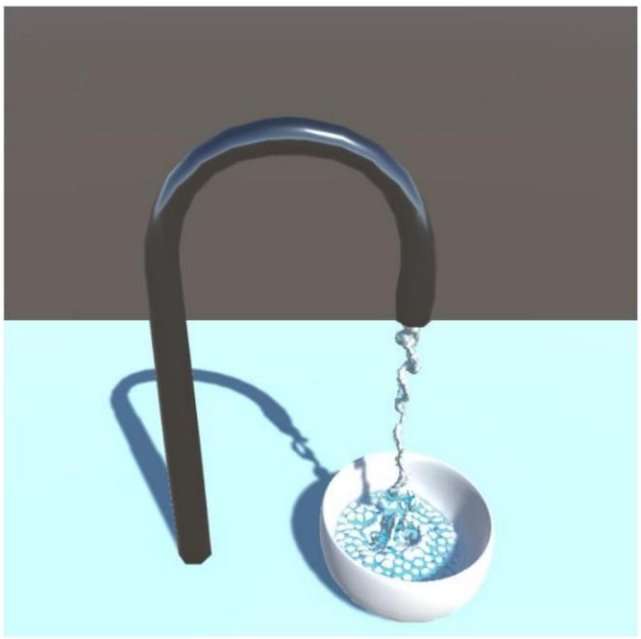
Experiment Insights

To better simulate the water flow effect of a faucet in real life, the value of each parameter is crucial. A slight mistake will make the effect less realistic. For example, the color of the particles has a great influence on the final water effect. If the color is a little darker, it will look a bit fake in the end.



Pic a

I found online that the principle of this plugin is roughly to generate a large number of small balls (points) with physical effects, and constrain the small balls through an algorithm similar to the fish swarm algorithm (simulating water pressure, tension, etc.). Finally, calculate the liquid performance through the points of the small balls. Calculate the smooth edges of the liquid by the small balls on the edge, and deepen the color in the center area according to the density of the small balls. (When the particle color is set relatively dark, you can clearly see that the color in the center of each small ball is darker, as shown in Figure a below. In addition, if the Render variable on the ObiParticleRenderer is not unchecked in step 3 above, you can see that each particle is a small ball after running, as shown in Figure b)



Pic b