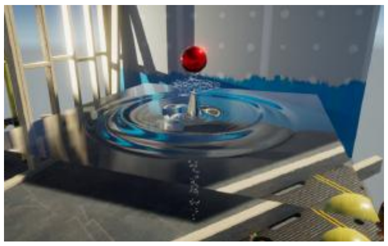
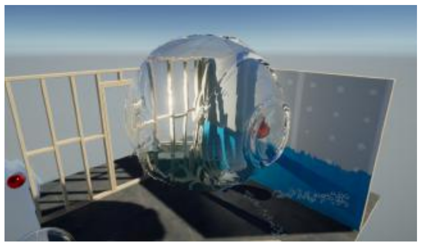
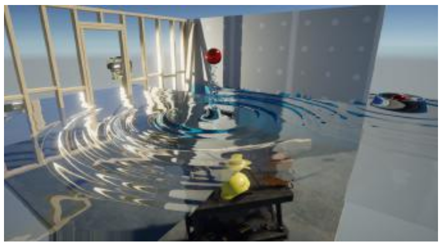
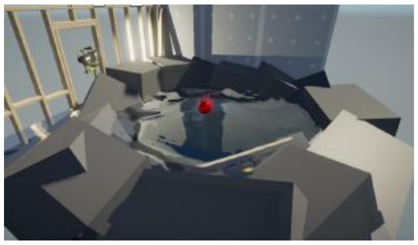
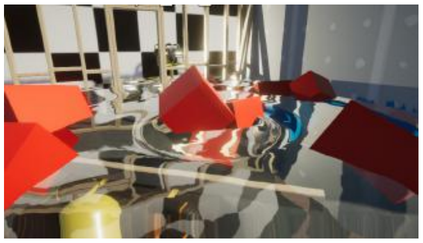
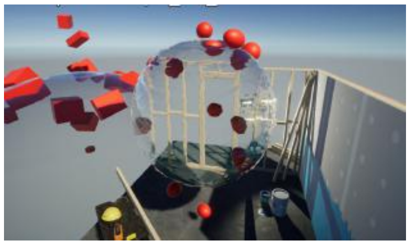
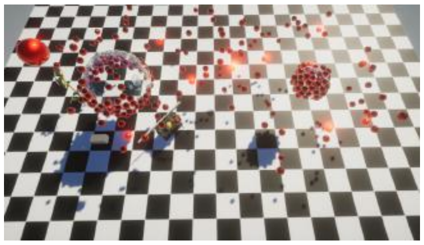
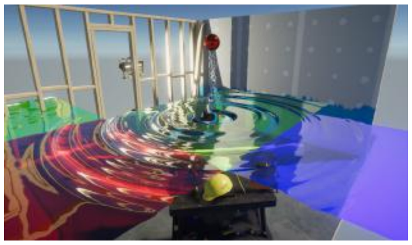
Simplestar-Game

Simple Interactive Water 3.0.2

Water refraction and wave simulation that does not break in VR

What is Simple Interactive Water?

A water surface asset that can generate waves by user operation, available in Unity's Universal RP and HDRP.by light wave calculation by compute shader and **water refraction expression that does not collapse with VR binocular vision**. In addition, when tiles are placed on the surface of the water, the waves propagate and can be used to express elongated shapes and wide lake surfaces. By drawing wave obstacle information on the mask texture, you can create a water surface that reflects waves in any shape. We also include a sample that floats on the surface of the water due to buoyancy, which is often used with the surface of the water. As an extension of the tiling idea, I added a 6-sided sphere mesh object and a sample of waves propagating on the sphere. Also added samples of gravity and buoyancy towards the center of the sphere. You can also color the surface of the water by specifying any color. I also added a refracting colored glass material, although it's not a water surface. In addition, we added a sample that moves Humanoid in VR and interacts with the water surface with the grip and trigger of the controller. The height of the water surface can be obtained in Script by specifying its position in real time.

Target Unity version

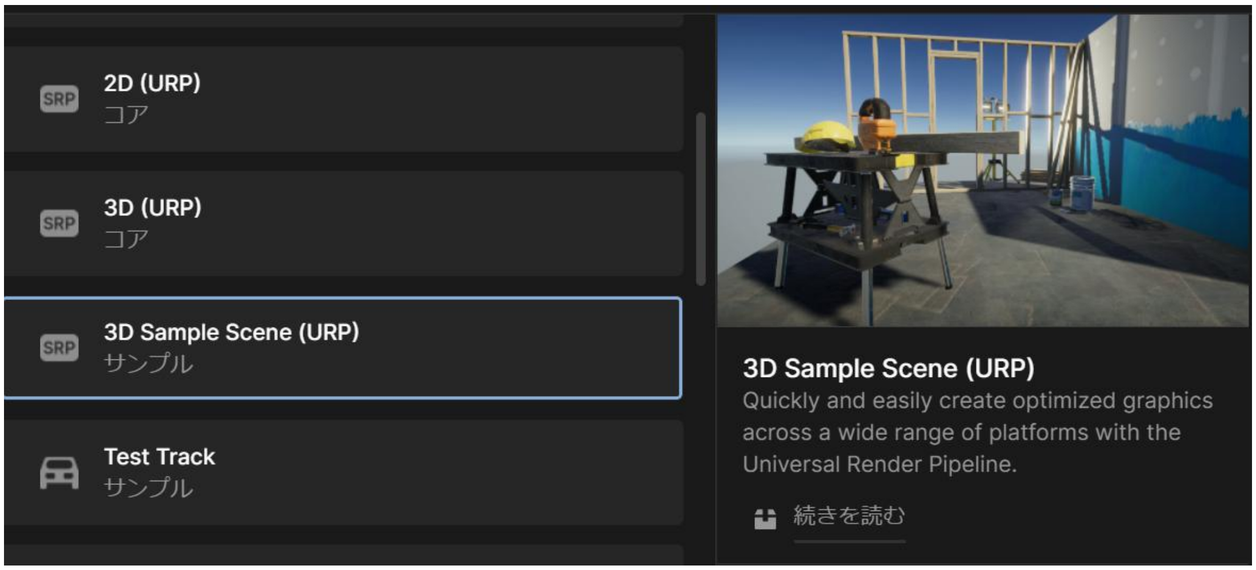
Developed with Unity 2021.3.9f1, it uses Unity's Shader Graph and is expected to work interchangeably with past and future versions.

Usage environment

It is strongly recommended to use it on a PC in general. Waves are simulated by GPU compute shaders. It works lightly in an environment with a large amount of GPU computation. You can also build for mobile devices. However, if GPU performance cannot be expected depending on the device, it will not operate at a high frame rate like a PC. WebGL will not work. This is because the compute shaders used are not supported by WebGL.

How to use (initial project settings)

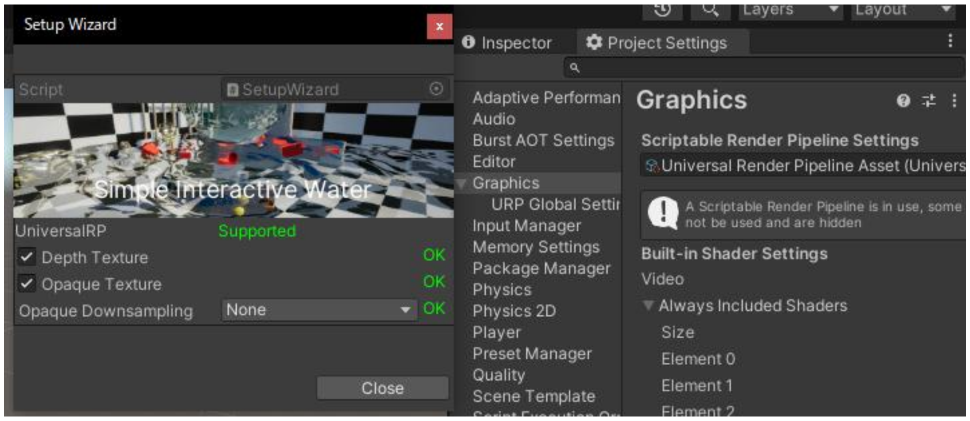
Select 3D Sample Scene (URP) from New Project to create a project.


2D (URP)
コア
3D (URP)
コア
3D Sample Scene (URP)
サンプル
Test Track
サンプル
3D Sample Scene (URP)
Quickly and easily create optimized graphics across a wide range of platforms with the Universal Render Pipeline.
続きを読む

Import this asset from the Package Manager into the URP sample project.

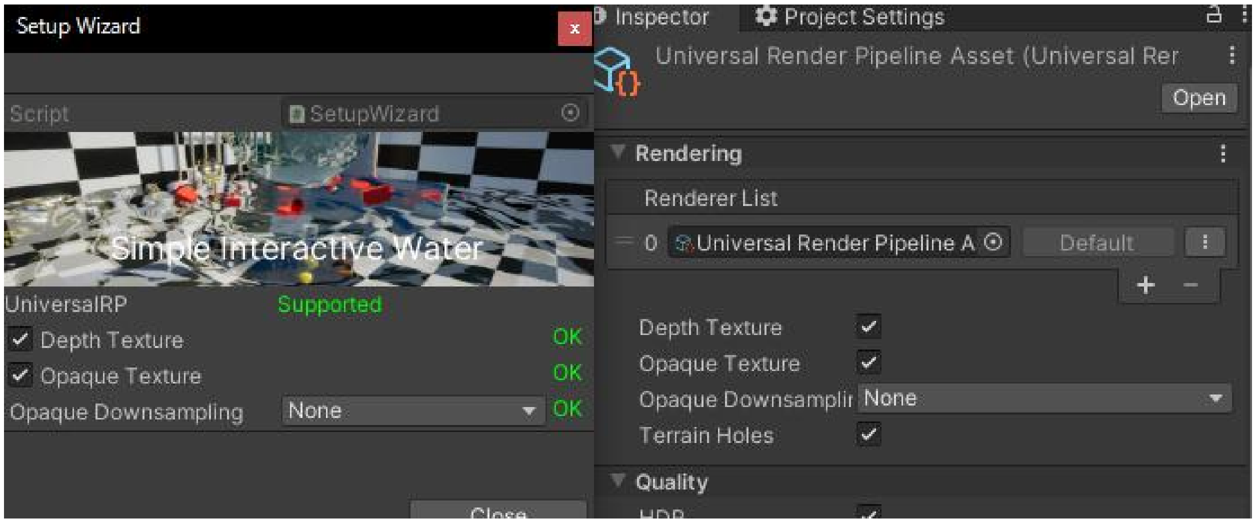
"This Unity Package has Package Manager dependencies." Press the “Install/Upgrade” button. Once the package installation is complete, you will be prompted to restart the Project for the new Input System to take effect. Please restart Project.

Import assets again after restarting the project (no second restart). Then select Window > Simple Interactive Water > Setup Wizard from the menu to open the setup wizard.



Press the Fix Now button for each item so that all items switch to OK.

The Setup Wizard edits the Universal Render Pipeline Asset under Graphics in Project Settings, Depth Texture, Opaque Texture.


Setup Wizard

Script SetupWizard

UniversalRP Supported

✓ Depth Texture OK

✓ Opaque Texture OK

Opaque Downsampling None OK

Inspector | Project Settings

Universal Render Pipeline Asset (Universal Rer

Rendering

Renderer List

0 Universal Render Pipeline A Default

Depth Texture ✓

Opaque Texture ✓

Opaque Downsamplir None

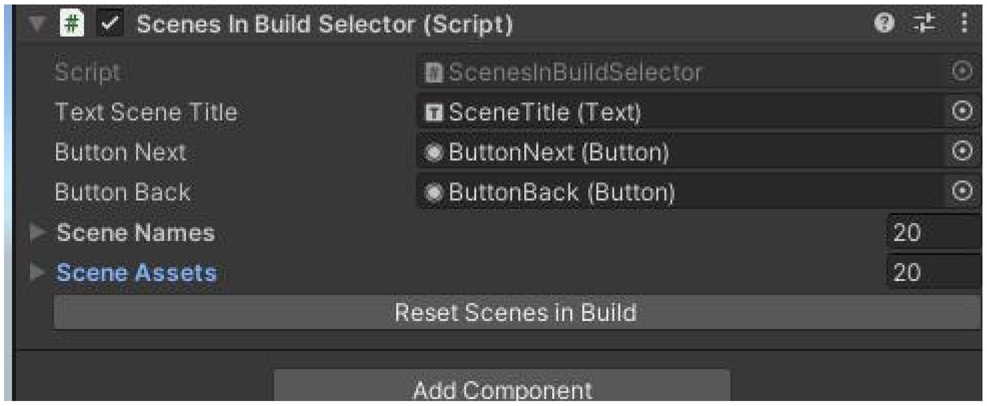
Terrain Holes ✓

Quality

Universal RP Sample Scenes

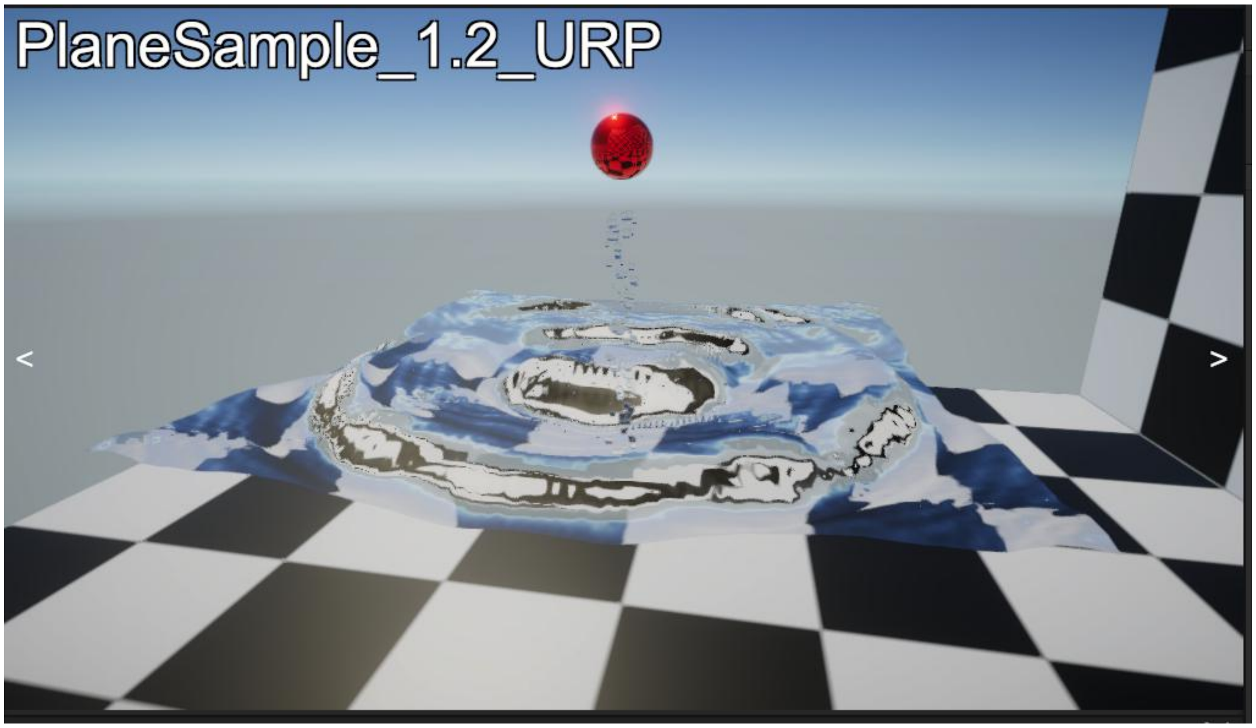
project is ready, let's check the sample scene list.

Assets/SimplestarGame/SimpleInteractiveWater/Example/Scenes/URP/Sample Scenes for URP.unity and press the Reset Scenes in Build button in the inspector of the ScenesInBuildURP object that exists in the hierarchy. Confirm that Scene Names have changed from 0 to 20.


Scenes In Build Selector (Script)

Script	ScenesInBuildSelector
Text Scene Title	SceneTitle (Text)
Button Next	ButtonNext (Button)
Button Back	ButtonBack (Button)
Scene Names	20
Scene Assets	20

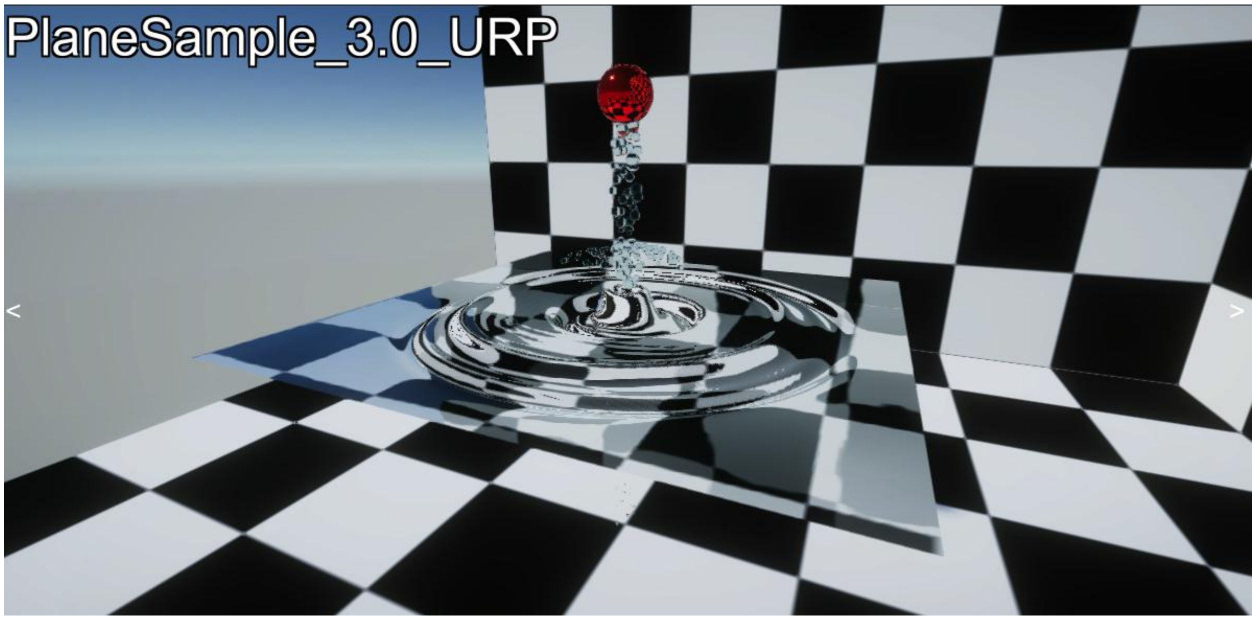
In this state, press the Play button to play the game. Playback of the "PlaneSample_1.2_URP" scene, which is the top element of Scene Names, will start.



You can switch scenes by pressing the left and right arrow keys or the left and right buttons on the screen. If you want to check individual scenes, open each scene under Assets/SimplestarGame/SimpleInteractiveWater/Example/Scenes/URP individually and press the Play button on each to check the operation.

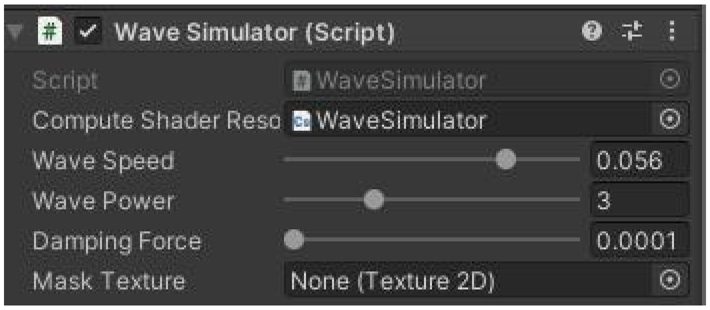
In the following, we will explain each sample scene.

0_Plane



This is a basic sample scene of this asset, where ripples spread with splashes when a scene with 3 materials applied, including the water surface material of the past version, is prepared. The only difference is the material attached to the surface of each water surface, otherwise they have the same configuration.

Describe the components of the scene. Check out the inspector for the Wave Simulator script component attached to the QuadPlaneX.X_URP object in the hierarchy.



Wave Simulator (Script)

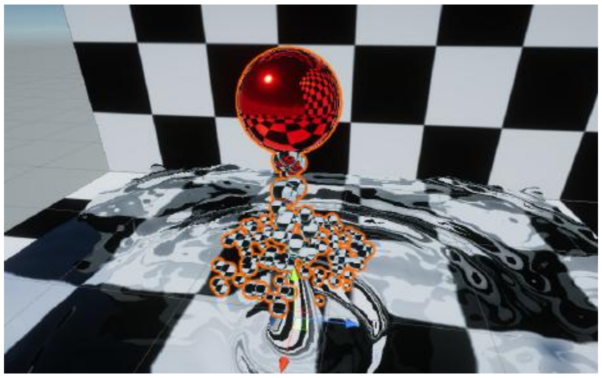
Script	WaveSimulator
Compute Shader Reso	WaveSimulator
Wave Speed	0.056
Wave Power	3
Damping Force	0.0001
Mask Texture	None (Texture 2D)

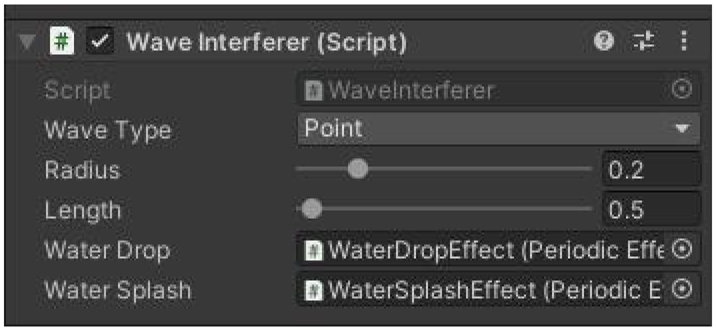
Wave Speed: Adjusts the wave speed. (Because the wave will resonate and diverge if it exceeds the acquisition frequency of the surrounding height, there is a limit to the speed. Please be careful.)
Wave Power: Adjust the size of the generated wave.

Damping Force: A factor of the speed at which the wave height converges. Small values do not easily converge.
Mask Texture is obstacle information used for waves to bounce off obstacles on the water surface. (**3_Mask** I will explain how to make it in)
Disable Auto Resolution: This option disables the automatic adjustment of the water surface resolution when the localScale of the game object is increased. If checked, the resolution is fixed at 128 x 128 even if the localScale is increased.

The water surface object corresponds to the change of localScale. If the water surface is enlarged as it is, the speed of the waves will be too fast and the resolution of the waves will look rough, so the resolution of the HeightMap texture of the water is automatically adjusted internally to match the localScale. If it is too large, processing will be performed at a high resolution, which will degrade performance. If you want to fix the resolution, please attach the texture of the fixed resolution to Mask Texture. The water surface will be created using the resolution of the Mask Texture. Alternatively, check the Disable Auto Resolution checkbox as described above.

Wave Interferer (wave generator gimmick)





Wave Interferer (Script)

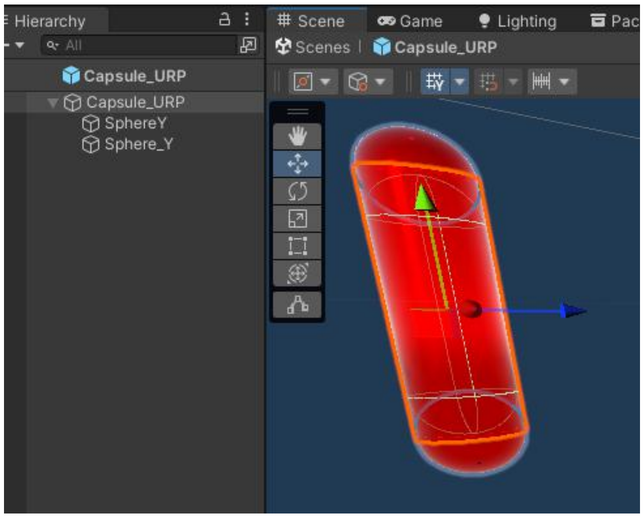
Script	WaveInterferer
Wave Type	Point
Radius	0.2
Length	0.5
Water Drop	WaterDropEffect (Periodic Effe
Water Splash	WaterSplashEffect (Periodic E

The Wave Interferer script component attached to the ball is interactively generating waves on the surface of the water. An OnTrigger event is fired when it hits a Collider on the water surface, generating waves on the water surface with the speed and direction at that time.

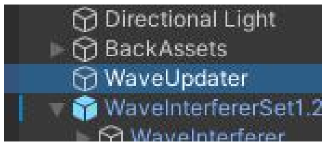
Water Drop: A particle effect of water dripping from a ball.
Water Splash: A particle effect that makes water splash on the surface of the water.
You can turn off these effects by setting each reference to None. (Omitting the staging improves performance.)

Attach this Wave Interferer script component to any object that has a Rigidbody to generate waves on the surface of the water.

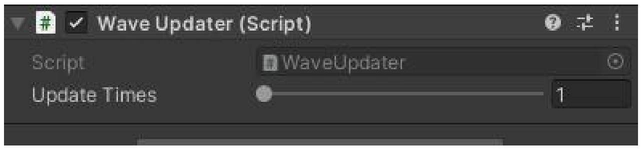
WaveType: You can choose between Point and Line. Point will generate a wave at the point of contact. Line generates waves on the line segment that connects the child object's SphereY and Sphere_Y points. (*WaveType cannot be switched from Point to Line if there is no child object named SphereY, Sphere_Y)
Radius: The radius of the object that plunges into the water. If you adjust it, the object's localScale will also switch accordingly. (* Conversely, do not adjust localScale. If you want to cut the linkage, comment out the localScale application code in the OnValidate function.)
Length: Indicates the length of the line in Line Type. If you adjust it, the object's localScale will also switch accordingly. Please refer to the implementation example for Line Type.
Assets/SimplestarGame/SimpleInteractiveWater/Prefabs/URP/Capsule_URP.prefab
Please use Line Type by referring to



In addition, it describes the GameObjects required for the scene. A WaveUpdater object must be placed in the scene to synchronize all wave calculations. All Wave Simulators do wave calculations on this WaveUpdater update event. In the WaterUpdater inspector, setting the Update Times value from 1 to 2 doubles the amount of waves calculated in the scene and doubles the speed of wave propagation.



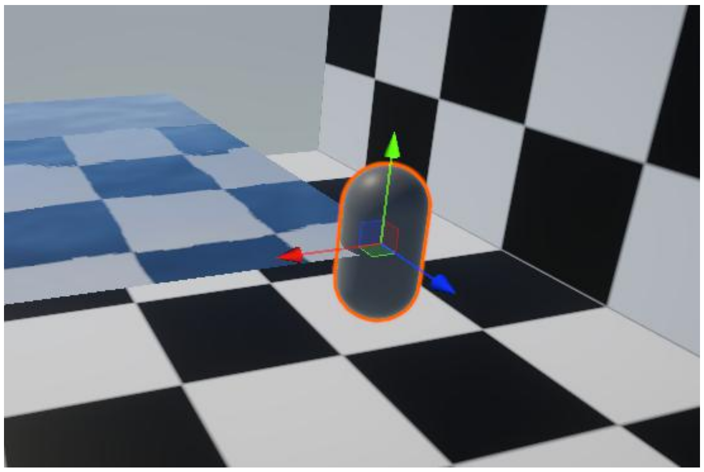
Directional Light
BackAssets
WaveUpdater
WaveInterfererSet1.2
WaveInterferer



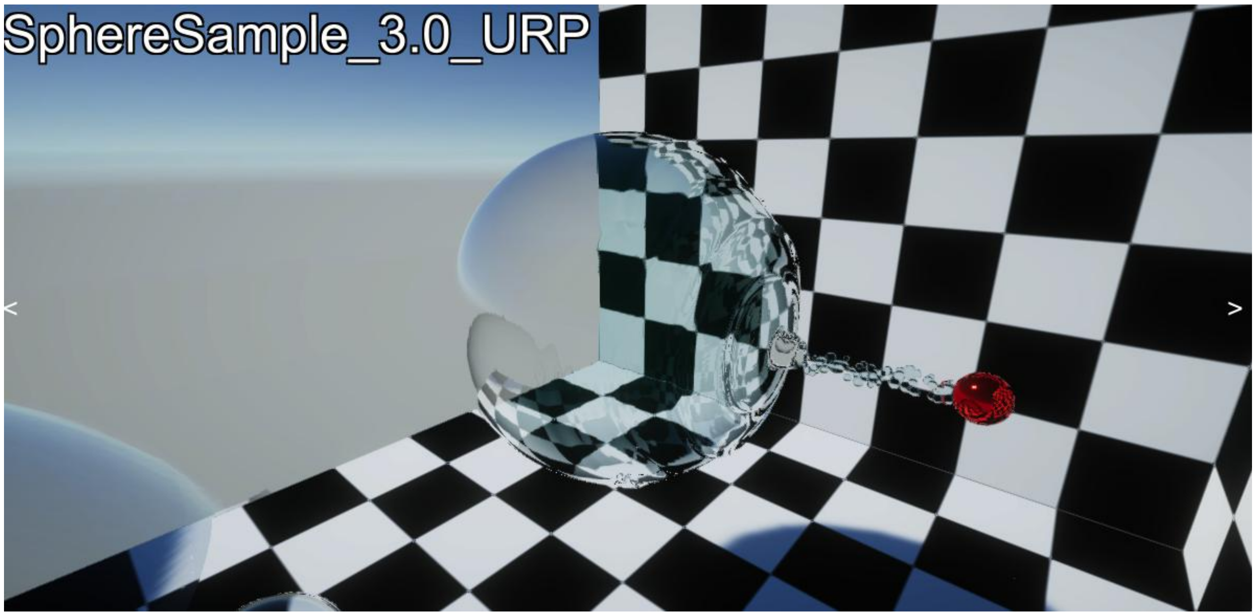
Wave Updater (Script)

Script WaveUpdater
Update Times 1

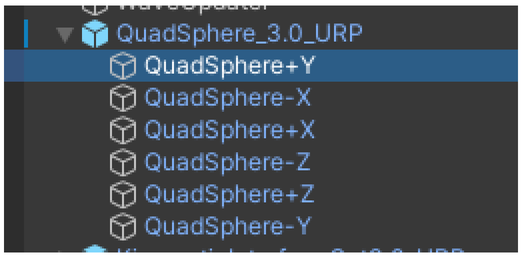
In PlaneSample_3.0_URP, there is a capsule stuck in the water surface, to which the WaveHeightChecker script component is attached. The height is obtained by entering the Raycast Hit point in the script.



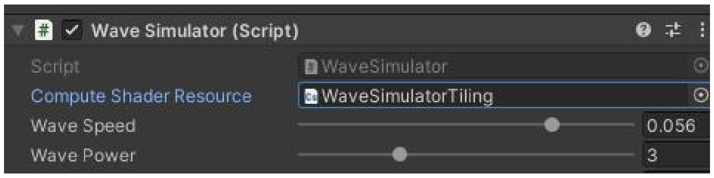
1_Sphere



This is a spherical water surface represented by a hexahedron using 6 sheets. Waves propagate in surfaces that meet each other and circulate around the sphere. WaveSimulator has a WaveSimulator.compute file for Tiling. Please note that the wave will not propagate around if it is not specified.



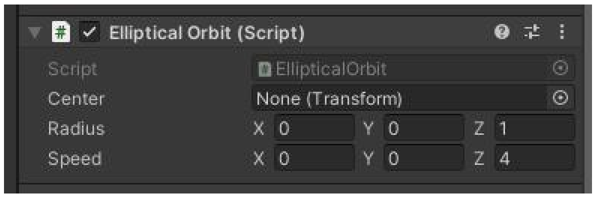
QuadSphere_3.0_URP
QuadSphere+Y
QuadSphere-X
QuadSphere+X
QuadSphere-Z
QuadSphere+Z
QuadSphere-Y



Wave Simulator (Script)

Script	WaveSimulator
Compute Shader Resource	WaveSimulatorTiling
Wave Speed	0.056
Wave Power	3

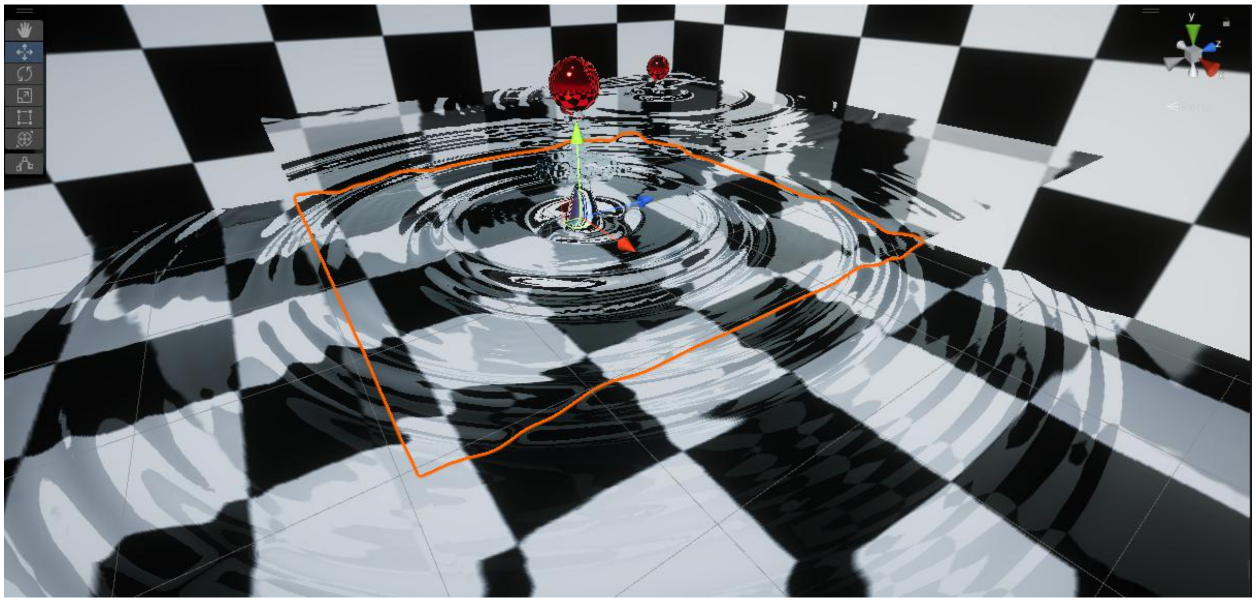
The horizontally oscillating ball has an Elliptical Orbit script component attached. This is a gimmick for demonstration purposes. Please use it for testing by specifying the radius and velocity for each axis and adjusting the trajectory.



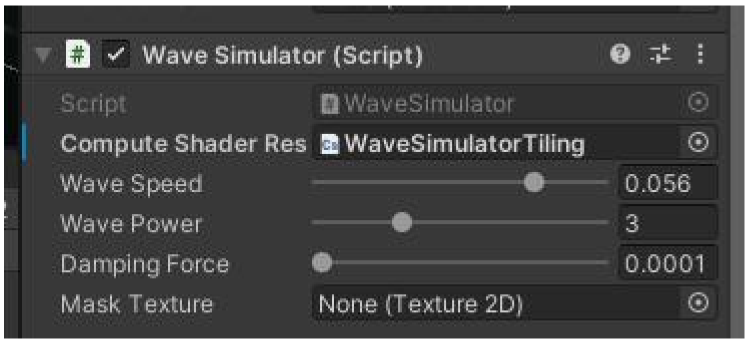
Elliptical Orbit (Script)

Script	EllipticalOrbit
Center	None (Transform)
Radius	X 0 Y 0 Z 1
Speed	X 0 Y 0 Z 4

2_Tiling



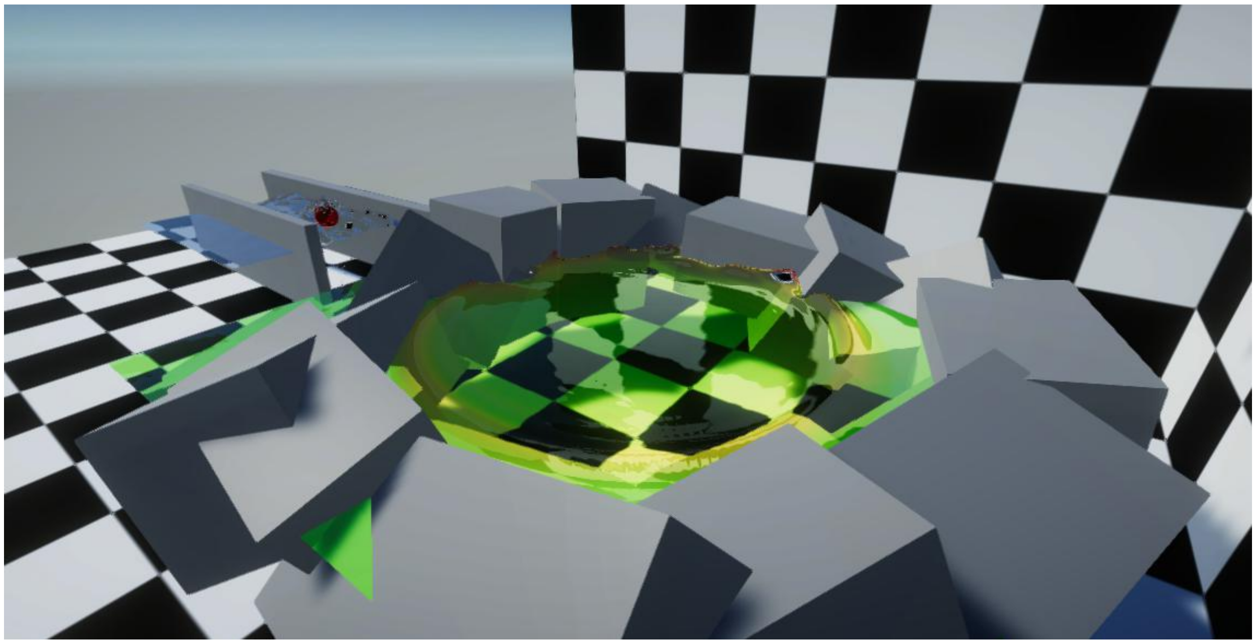
Tiling expands the range of water surfaces that can be expressed, such as long and narrow irrigation canals and wider lake surfaces. One thing to keep in mind for tiling is that all Plane objects must have the same localScale. Additionally, you need to switch the ComputeShader Resource to WaveSimulatorTiling for tiling. Refer to the sample scene and switch resources as necessary.

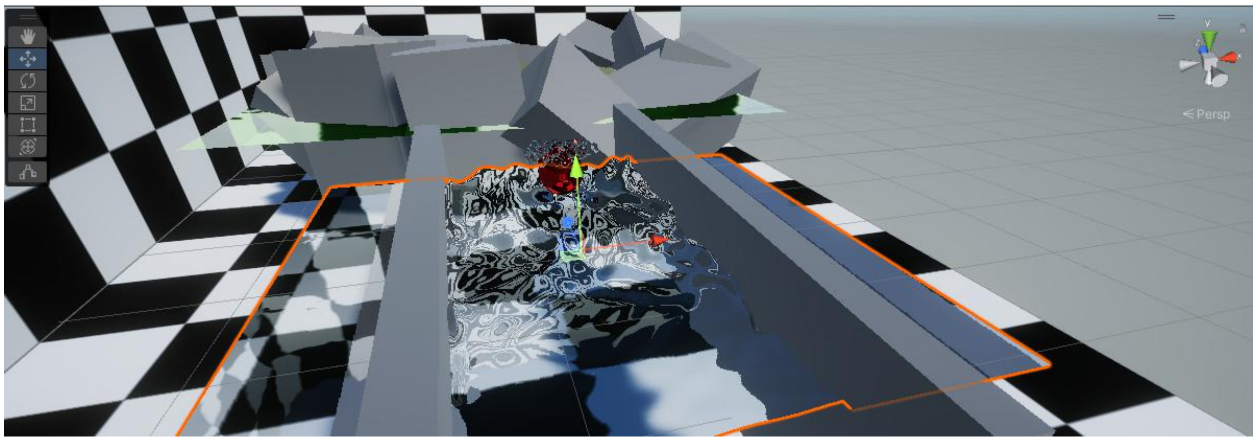


Wave Simulator (Script)

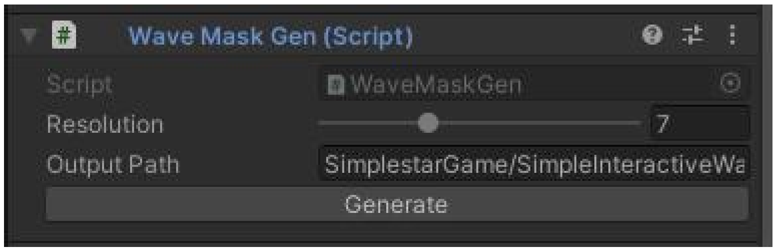
Script	WaveSimulator
Compute Shader Res	WaveSimulatorTiling
Wave Speed	0.056
Wave Power	3
Damping Force	0.0001
Mask Texture	None (Texture 2D)

3_Mask





This is a sample scene where waves are reflected by obstacles placed on the surface of the water. Waves are reflected by assigning a Texture that depicts a wave obstacle to the Mask Texture of the Wave Simulator script component.



Wave Mask Gen (Script)

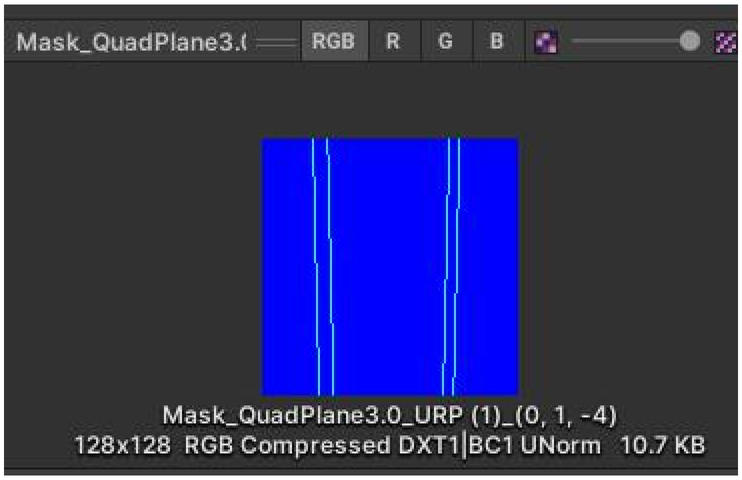
Script	WaveMaskGen
Resolution	7
Output Path	SimplestarGame/SimpleInteractiveWa

When you press the Generate button of the WaveMaskGen script component attached to the QuadPlane object, the Mask Texture file is output with the QuadPlane object name to the destination specified in the Output Path. By setting this output Texture file to the Mask Texture of the WaveSimulator, waves will be reflected by obstacles.

You can also create your own Mask Texture.

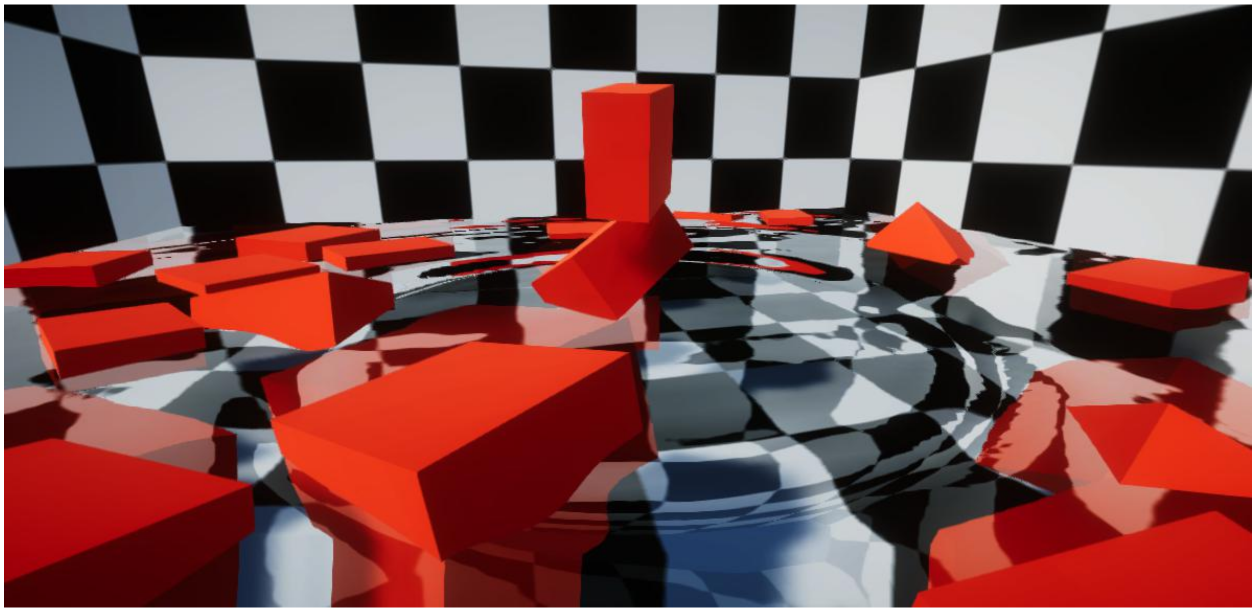
How it works: If you set the G value in 16bit RG format to something other than 0, the wave will reflect there. The Resolution value represents a power of 2 (7 is 128 x 128), select it according to the target water surface, and press the Generate button. The water surface automatically sets the resolution according to the Scale, but if Mask Texture is specified, the resolution is fixed according to the Mask Texture resolution.

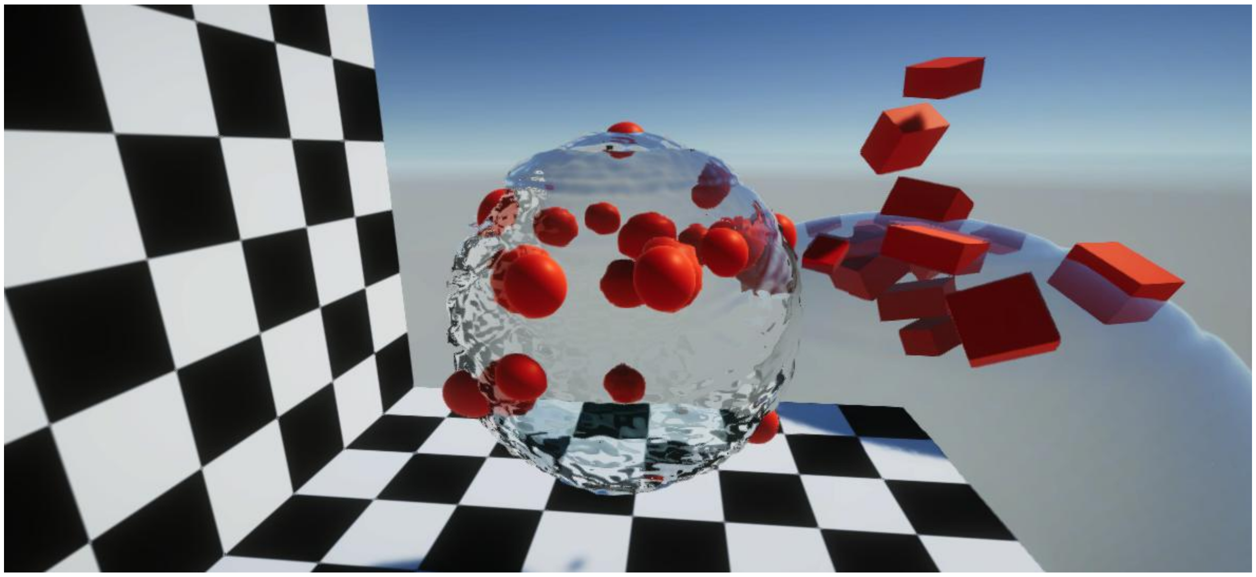
You can check the shape of the obstacle by checking the Mask Texture.



Mask_QuadPlane3.0_URP (1)_(0, 1, -4)
128x128 RGB Compressed DXT1|BC1 UNorm 10.7 KB

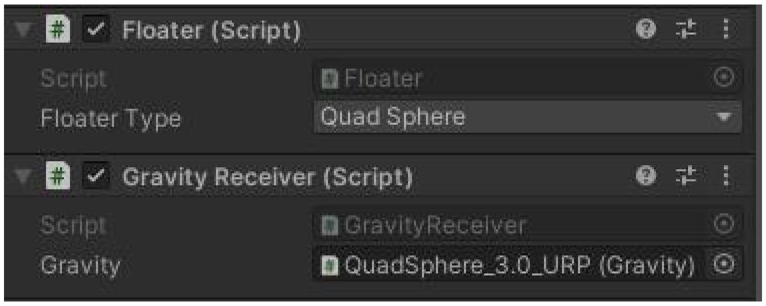
4_Floater





This is a sample scene where boxes and balls float on the surface of water and spheres. A Floater script component is attached to the floating side, and buoyancy works in proportion to the volume of how much it is submerged in the water. If you want to represent a light object that floats on water, decrease the mass of the Reigidbody, and if you want to represent a sinking object, increase the mass.

There are two types of Floater, Quad Plane and Quad Sphere, which are selected according to the surface of the water to float. The gravitational force that attracts water ball will be explained in detail in the next 5_Gravity.



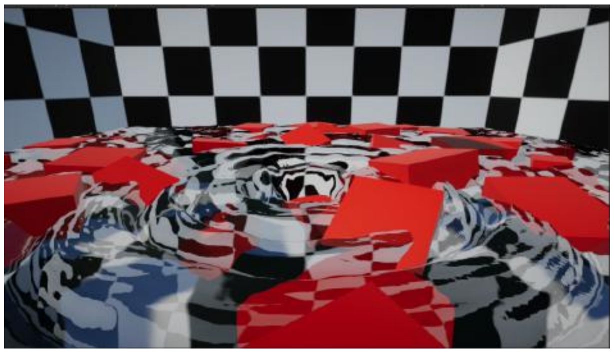
Floater (Script)

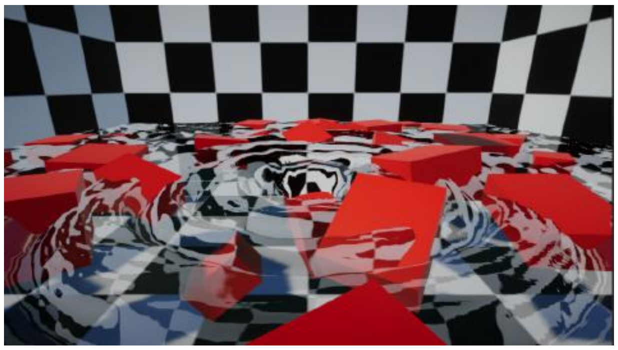
Script	Floater
Floater Type	Quad Sphere

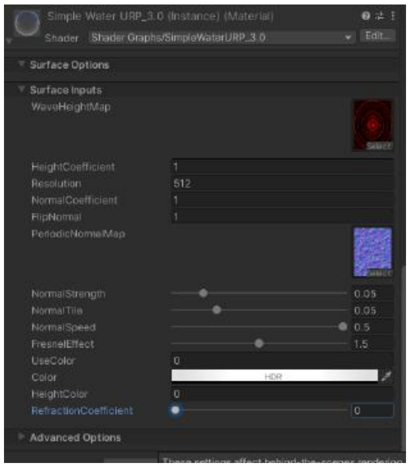
Gravity Receiver (Script)

Script	GravityReceiver
Gravity	QuadSphere_3.0_URP (Gravity)

Since the refraction representation of water is a color reference that shifts the UV, I think that many people are concerned about strange rendering results in scenes where objects float on the surface of the water. By switching the Refraction Coefficient value of the material attached to the surface of the water from 1 to 0, the refraction effect disappears, but you can get rid of the strange rendering result as a normal transparent material.







Simple Water URP_3.0 (Instance) (Material)

Shader Shader Graphs/SimpleWaterURP_3.0 Edit...

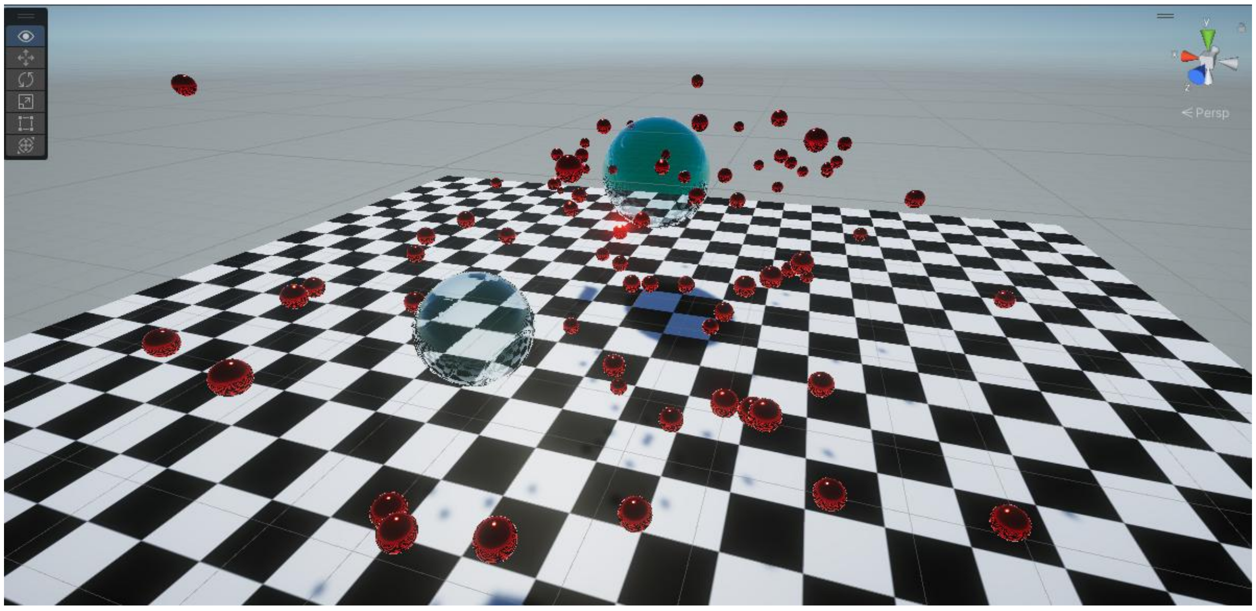
Surface Options

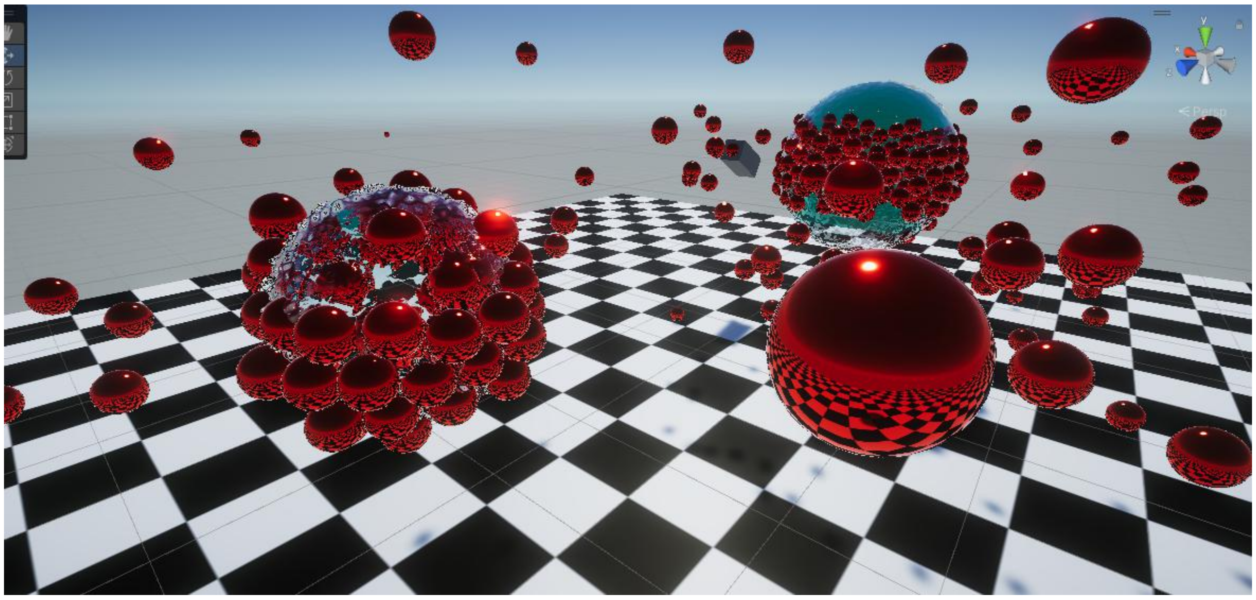
Surface Inputs

WaveHeightMap	
HeightCoefficient	1
Resolution	512
NormalCoefficient	1
FlipNormal	1
PeriodicNormalMap	
NormalStrength	0.05
NormalTile	0.05
NormalSpeed	0.5
FresnelEffect	1.5
UseColor	0
Color	HDR
HeightColor	0
RefractionCoefficient	0

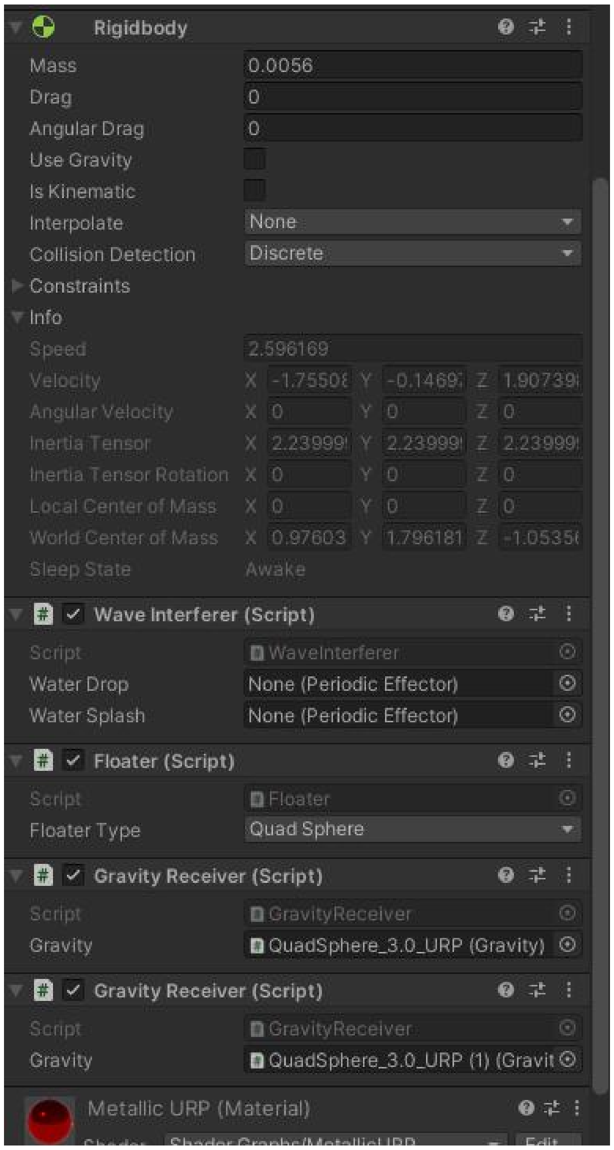
Advanced Options

5_Gravity





This is a sample scene of gravity where a red ball is attracted to two water balls. To be attracted by gravity, uncheck useGravity on the ball's Rigidbody and attach a Gravity Receiver script component. To get attracted to multiple gravity sources, attach that many Gravity Receiver script components. The gravitational force changes depending on the size of the Mass of the Gravity script component referenced by the Gravity Receiver. Also, the Mass value of the ball's own Rigidbody is used in the gravitational calculation.



Rigidbody

Mass	0.0056
Drag	0
Angular Drag	0
Use Gravity	☐
Is Kinematic	☐
Interpolate	None
Collision Detection	Discrete
Constraints	
Info	
Speed	2.596169
Velocity	X -1.7550ŧ Y -0.1469; Z 1.90739ŧ
Angular Velocity	X 0 Y 0 Z 0
Inertia Tensor	X 2.23999 Y 2.23999 Z 2.23999
Inertia Tensor Rotation	X 0 Y 0 Z 0
Local Center of Mass	X 0 Y 0 Z 0
World Center of Mass	X 0.97603 Y 1.796181 Z -1.0535ŧ
Sleep State	Awake

Wave Interferer (Script)

Script	WaveInterferer
Water Drop	None (Periodic Effector)
Water Splash	None (Periodic Effector)

Floater (Script)

Script	Floater
Floater Type	Quad Sphere

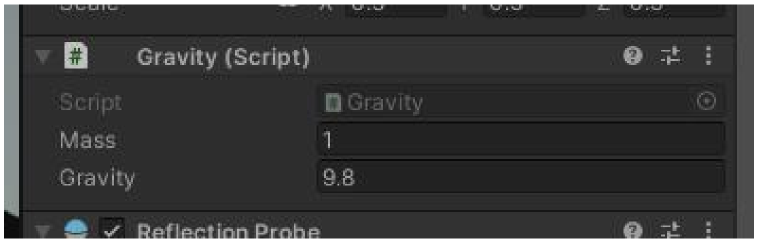
Gravity Receiver (Script)

Script	GravityReceiver
Gravity	QuadSphere_3.0_URP (Gravity)

Gravity Receiver (Script)

Script	GravityReceiver
Gravity	QuadSphere_3.0_URP (1) (Gravit

Metallic URP (Material)



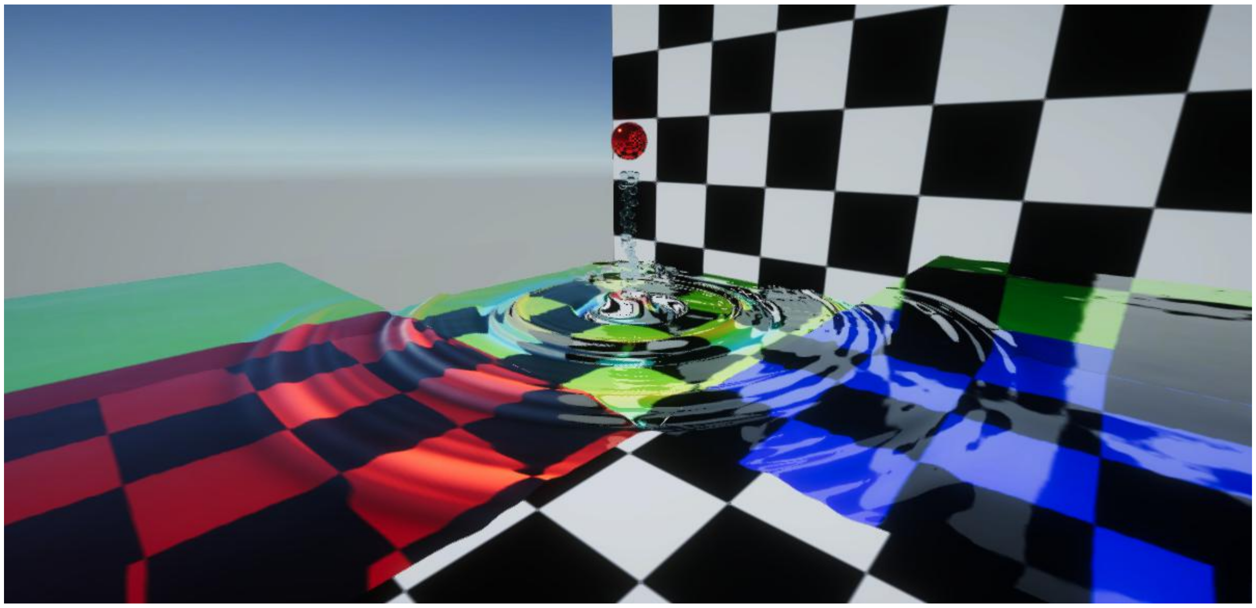
Gravity (Script)

Script	Gravity
Mass	1
Gravity	9.8

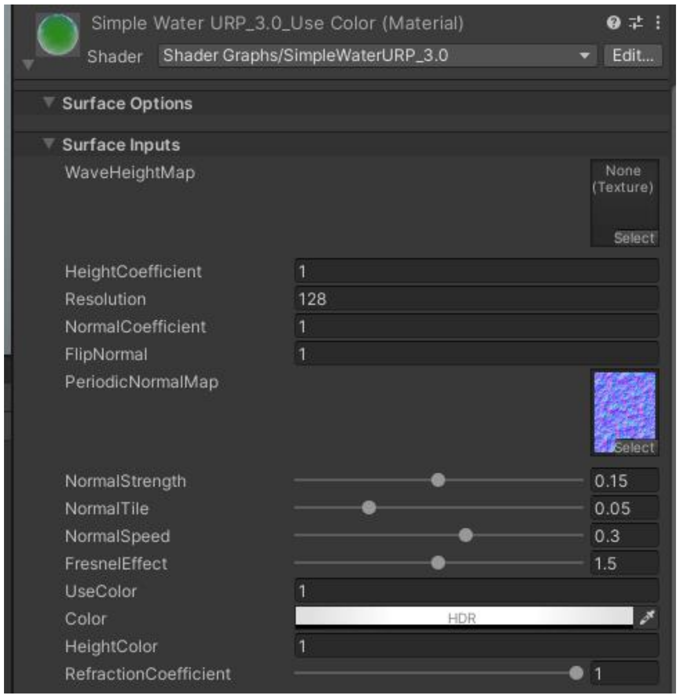
Reflection Probe

Assets/SimplestarGame/SimpleInteractiveWater/Example/Scenes/URP/5_Gravity/StarsSample_3.0_URP.unity simulates planetary motion by attaching Gravity and GravityReceiver to each other.

6_Colors



This is a sample scene that gives color to water and changes the color depending on the height of the waves. Check the Material Color of each colored water surface object. If you want the color to change with the height of the waves, increase the UseColor value from 0 to 1.

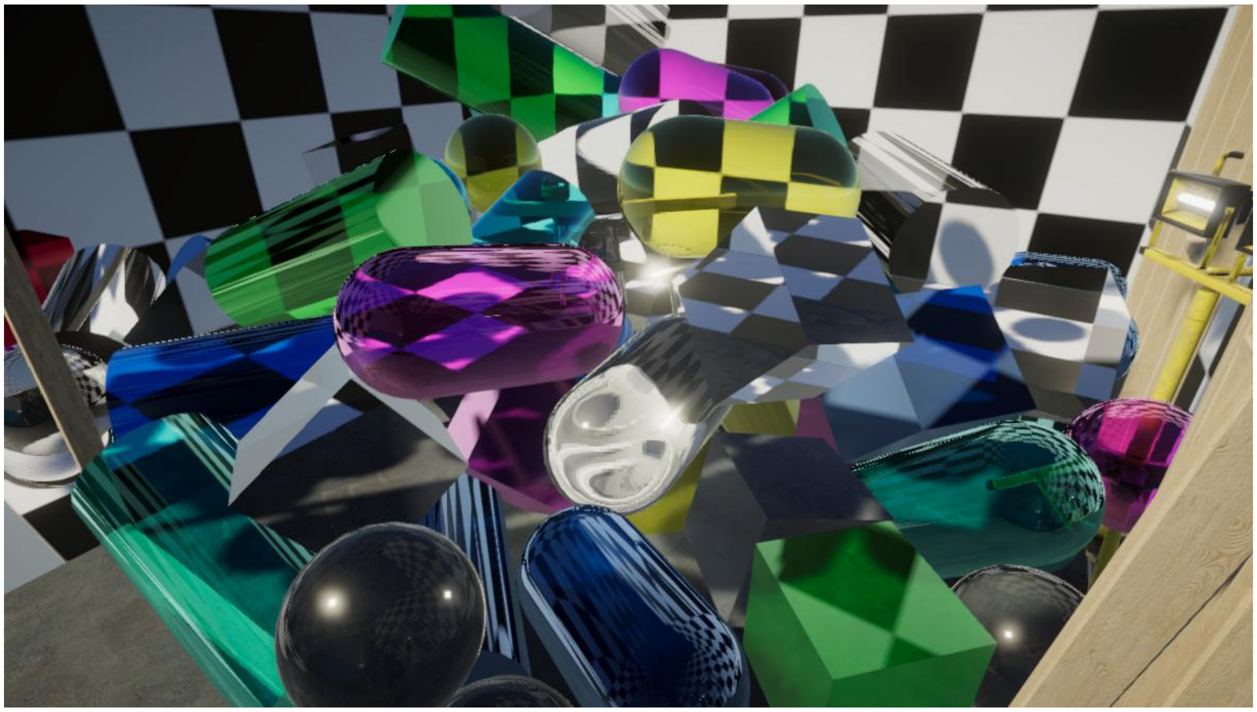


Simple Water URP_3.0_Use Color (Material)

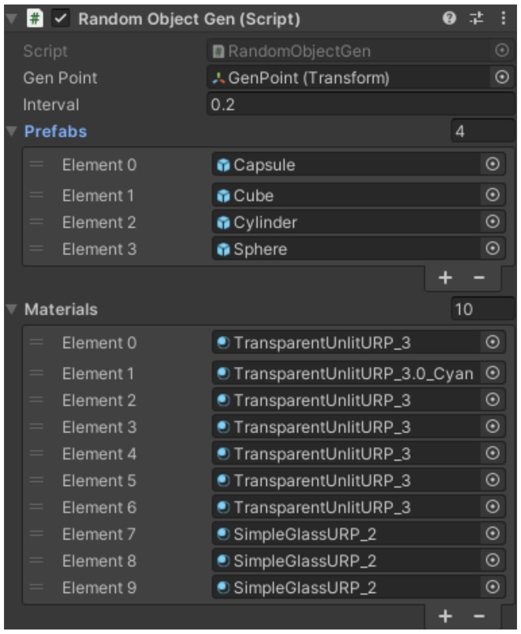
Shader: Shader Graphs/SimpleWaterURP_3.0

WaveHeightMap	None (Texture)
HeightCoefficient	1
Resolution	128
NormalCoefficient	1
FlipNormal	1
PeriodicNormalMap	
NormalStrength	0.15
NormalTile	0.05
NormalSpeed	0.3
FresnelEffect	1.5
UseColor	1
Color	HDR
HeightColor	1
RefractionCoefficient	1

7_Transparent



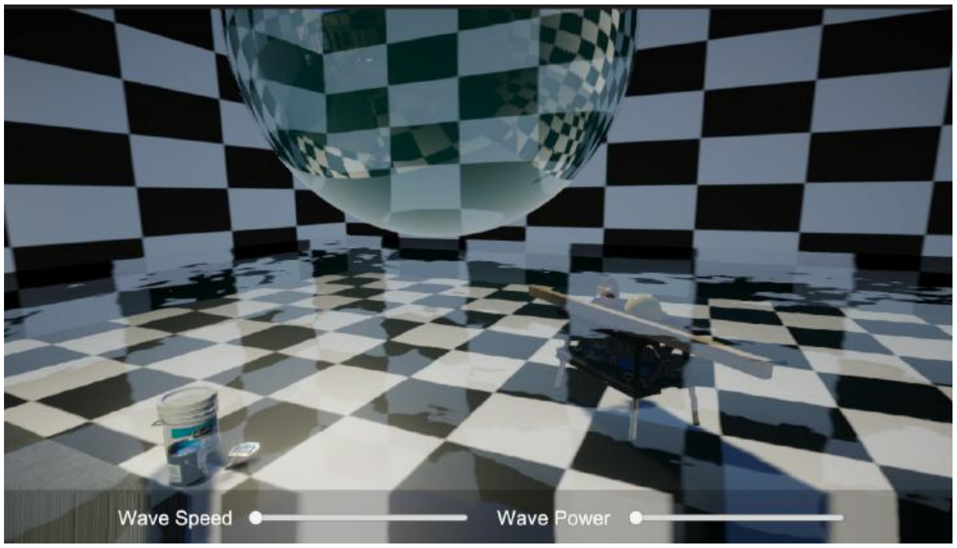
This is a sample of a colored glass material made by refraction expression of the water surface. Please use the Materials listed in the Random Object Gen script component present in the scene.

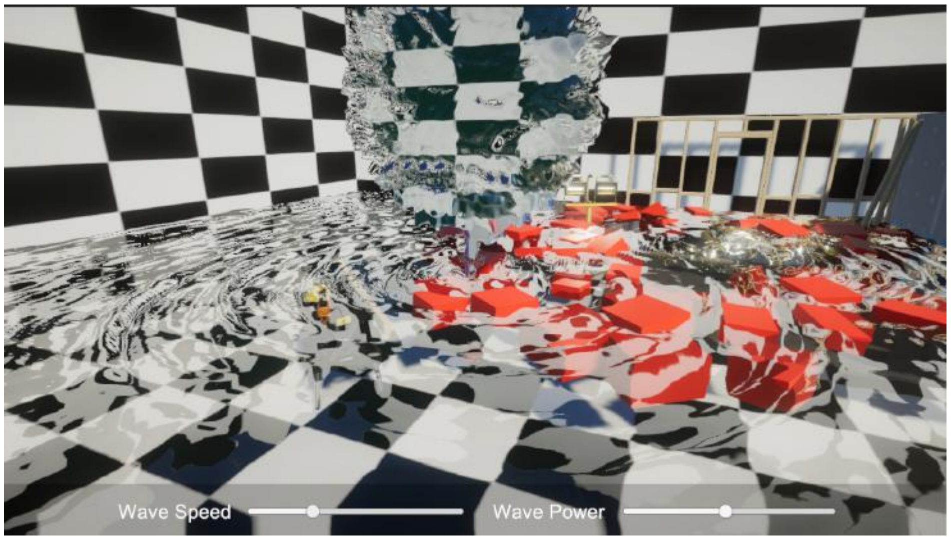


Random Object Gen (Script)

Script	RandomObjectGen
Gen Point	GenPoint (Transform)
Interval	0.2
Prefabs	4
Element 0	Capsule
Element 1	Cube
Element 2	Cylinder
Element 3	Sphere
Materials	10
Element 0	TransparentUnlitURP_3
Element 1	TransparentUnlitURP_3.0_Cyan
Element 2	TransparentUnlitURP_3
Element 3	TransparentUnlitURP_3
Element 4	TransparentUnlitURP_3
Element 5	TransparentUnlitURP_3
Element 6	TransparentUnlitURP_3
Element 7	SimpleGlassURP_2
Element 8	SimpleGlassURP_2
Element 9	SimpleGlassURP_2

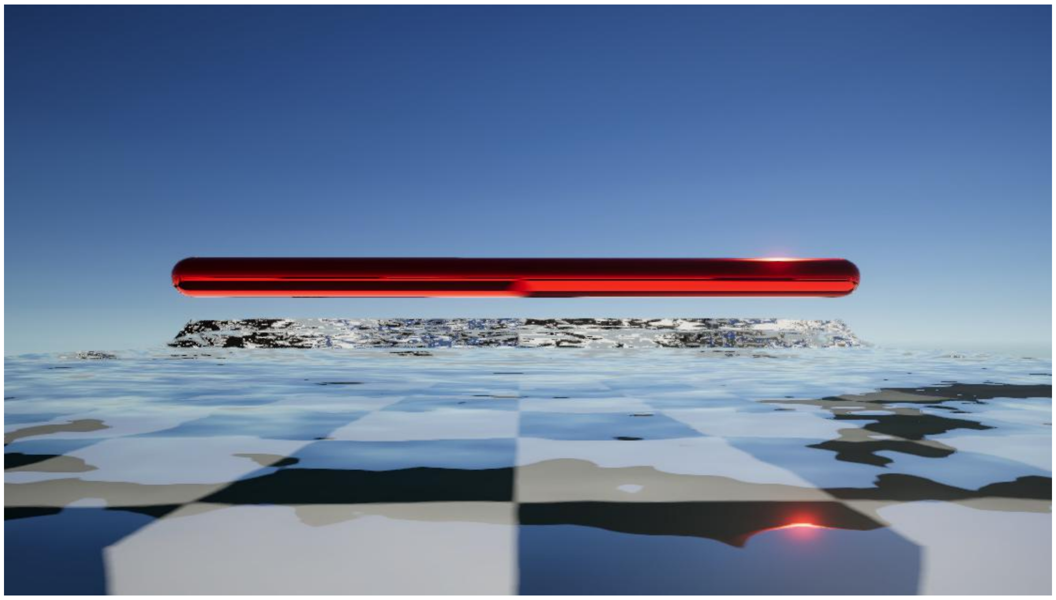
8_Params





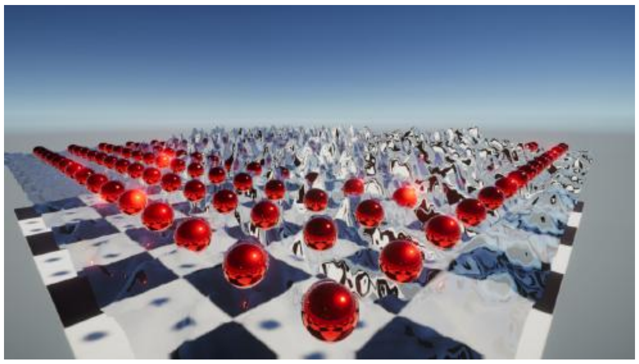
This is a sample scene where you can adjust the speed of the waves on the water surface and the size of the generated waves. By pinching the sliders in the GUI, you can see how the speed and magnitude of the waves change at run time. What you are actually adjusting are the Wave Speed and Wave Power values in the Wave Simulator script component. Damping Force is the wave damping force. A small value will not subside the waves, and a large value will subside quickly.

9_BigWave



A scene where a large Capsule Wave Interferer object generates a big wave. By increasing the scale of one water surface, you can confirm that it can be made into a vast water surface and that waves can be generated by line segments using the Line Type Wave Interferer. Please use it as a reference when you want to use the water surface widely or when you want to generate linear waves.

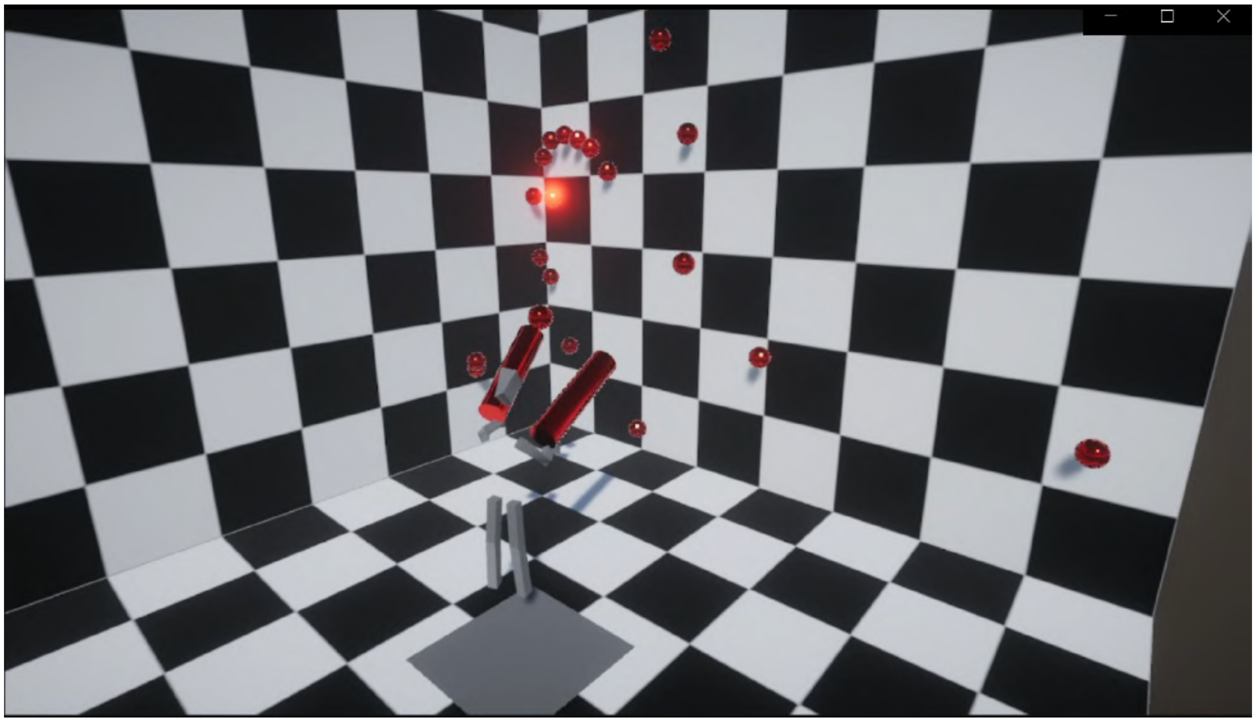
MultiWaves



Test scene to check the maximum number of waves that can be generated in one frame. Spawning waves all at once with this number does not degrade performance.

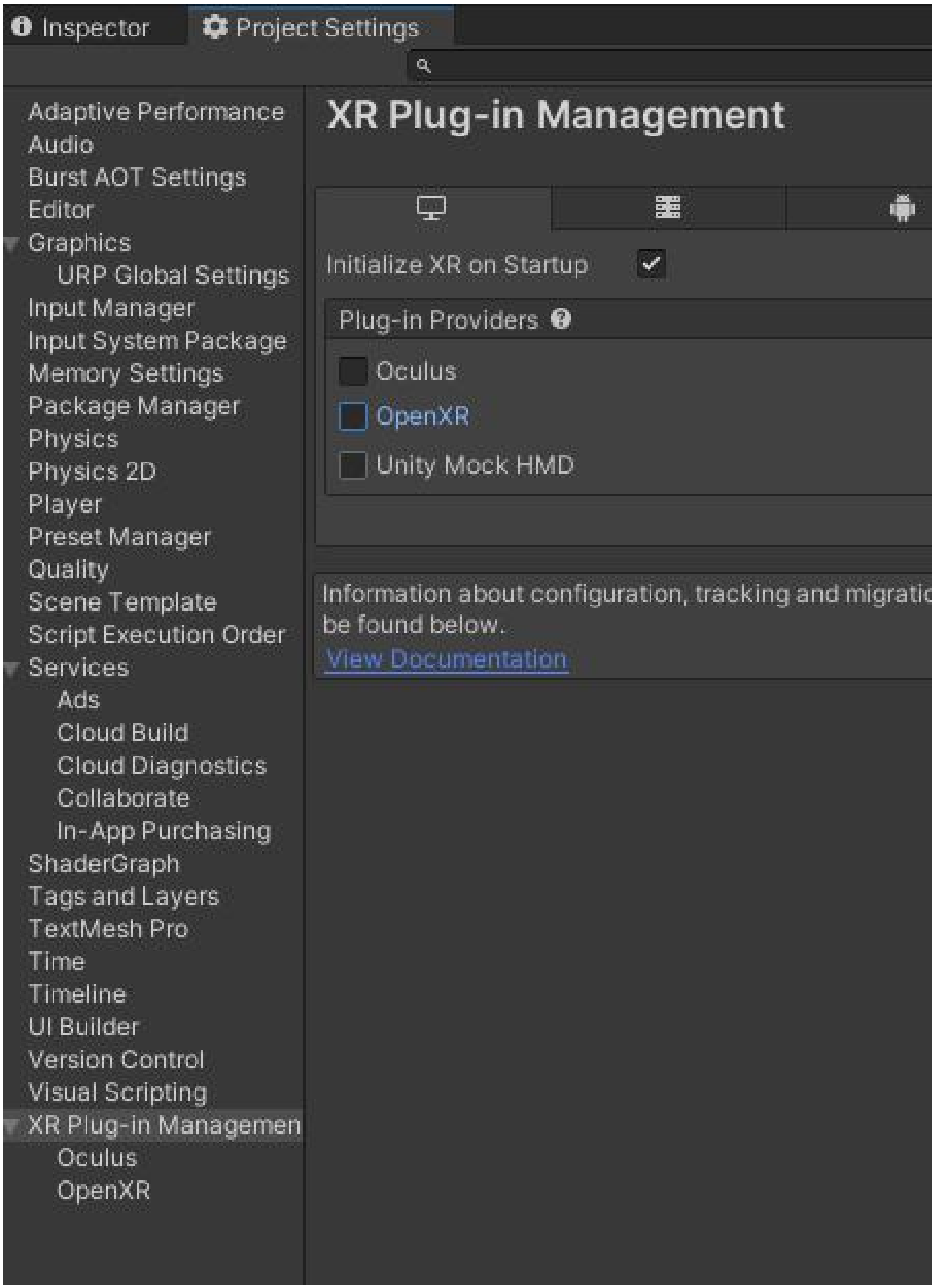
SimpleXR

XRAvatar



VR environment sample scene. (Because HDRP does not have the performance of using VR due to Unity restrictions, it is dedicated to Universal RP.)

For preparation, select your VR environment that the developer targets from Plug-in Providers of XR Plug-in Management in Project Settings. This time, check OpenXR for the HTC Vive devices.


Inspector | Project Settings

Adaptive Performance
Audio
Burst AOT Settings
Editor
Graphics
URP Global Settings
Input Manager
Input System Package
Memory Settings
Package Manager
Physics
Physics 2D
Player
Preset Manager
Quality
Scene Template
Script Execution Order
Services
Ads
Cloud Build
Cloud Diagnostics
Collaborate
In-App Purchasing
ShaderGraph
Tags and Layers
TextMesh Pro
Time
Timeline
UI Builder
Version Control
Visual Scripting
XR Plug-in Managemen
Oculus
OpenXR

XR Plug-in Management

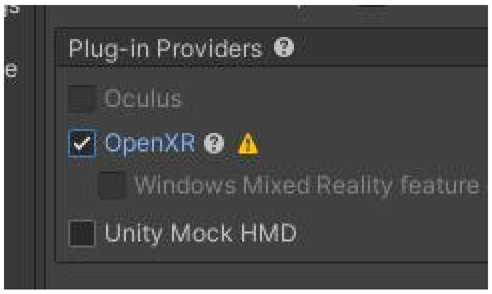
Initialize XR on Startup ✓

Plug-in Providers

- [ ] Oculus
- [ ] OpenXR
- [ ] Unity Mock HMD

Information about configuration, tracking and migratio
be found below.
View Documentation

If checked, a warning will appear.


Plug-in Providers

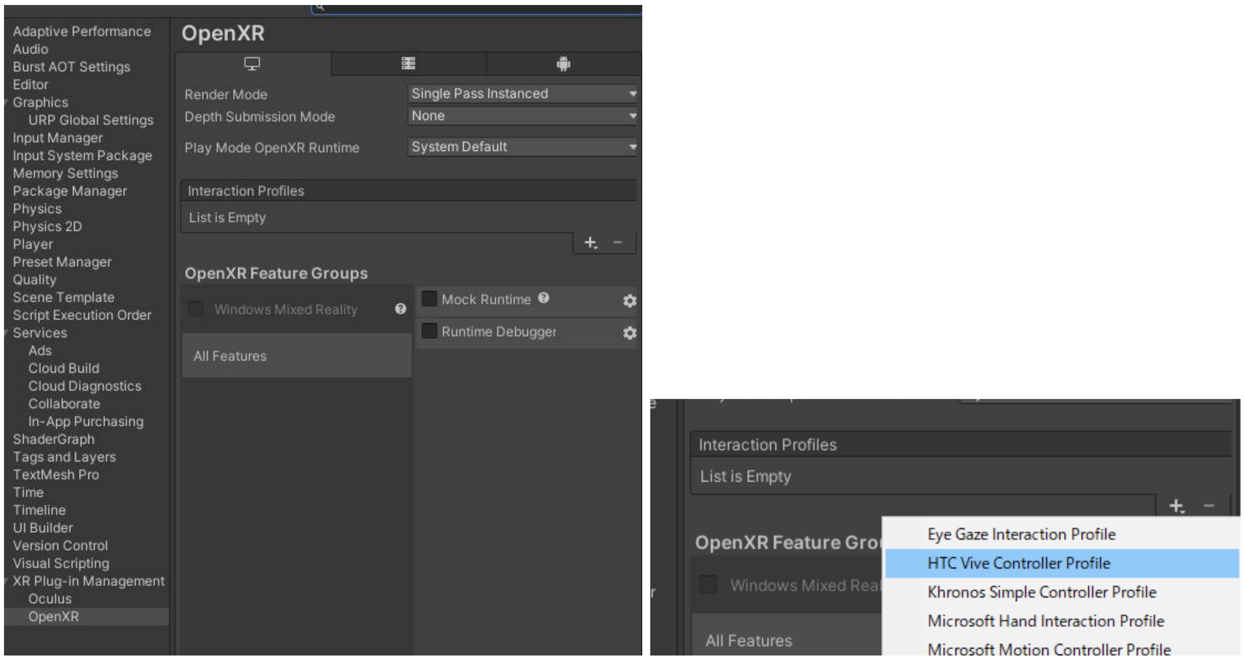
Oculus

OpenXR ✓

Windows Mixed Reality feature

Unity Mock HMD

Select OpenXR from Project Settings as shown in the image and add the HTC Vive Controller Profile to Interaction Profiles.


Adaptive Performance
Audio
Burst AOT Settings
Editor
Graphics
URP Global Settings
Input Manager
Input System Package
Memory Settings
Package Manager
Physics
Physics 2D
Player
Preset Manager
Quality
Scene Template
Script Execution Order
Services
Ads
Cloud Build
Cloud Diagnostics
Collaborate
In-App Purchasing
ShaderGraph
Tags and Layers
TextMesh Pro
Time
Timeline
UI Builder
Version Control
Visual Scripting
XR Plug-in Management
Oculus
OpenXR

OpenXR

Render Mode	Single Pass Instanced
Depth Submission Mode	None
Play Mode OpenXR Runtime	System Default

Interaction Profiles

List is Empty

OpenXR Feature Groups

Windows Mixed Reality

All Features

Mock Runtime

Runtime Debugger

Interaction Profiles

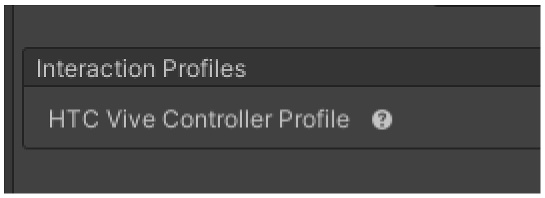
List is Empty

OpenXR Feature Groups

Windows Mixed Reality

All Features

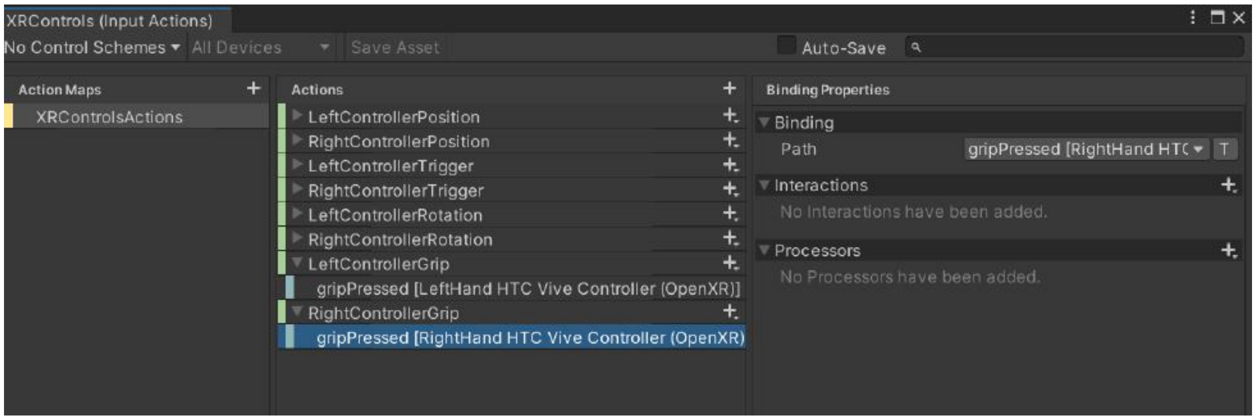
Eye Gaze Interaction Profile
HTC Vive Controller Profile
Khronos Simple Controller Profile
Microsoft Hand Interaction Profile
Microsoft Motion Controller Profile


Interaction Profiles

HTC Vive Controller Profile

Once you've picked your target VR environment, it's time to check the Input Actions.

Open **Assets/SimplestarGame/SimpleXRAvatar/Input/XRControls.inputactions** file.



In this way, you can check the input Action list of the VR equipment used in the sample scene. Open the Binding Properties for each and add bindings if your VR equipment doesn't have enough inputs. (In the case of HTC Vive's OpenXR environment, the settings have already been made.) After completing the above preparations, open

Assets/SimplestarGame/SimpleXRAvatar/Scenes/XRAvatar.unity scene.

When played, Humanoid starts moving using IK with VR HMD and left and right controller tracking, you can hold the red cylinder with the grip operation of the controller, and you can shoot the red ball by pulling the trigger.

Scene Hierarchy Description

StaticCamera : A camera separate from the VR view. Since the priority of the camera is high, it is possible to project the appearance of playing in VR on the monitor from a third-person perspective.

AvatarController: In the XRAvatarController script component, the Animator controller attached to the Humanoid, the Head, Left Controller, and Right Controller objects that follow and move by tracking, and the XRRig that is the offset posture of the start position are set. When you start the game, the AvatarActivator script component allows you to set the XRAvatarController to the specified Humanoid and make the Avatar follow the Avatar by IK.

XRActions: Describes event registration in the Input System's Action list, object grabbing by Grip, and object triggering by triggers. Objects that can be gripped must specify “Grip” as

Tag. In addition, there is a function that can adjust the position after gripping, and this is specified by the GripOffset script component attached to each object.

If the target Grip object has a class attached that inherits from ITrigger, its OnTrigger event is called in the trigger event to perform the Trigger processing of the gripped object. As an example, the ShooterL and ShooterR objects in the scene are attached with the Shooter script component, which inherits the ITrigger class.

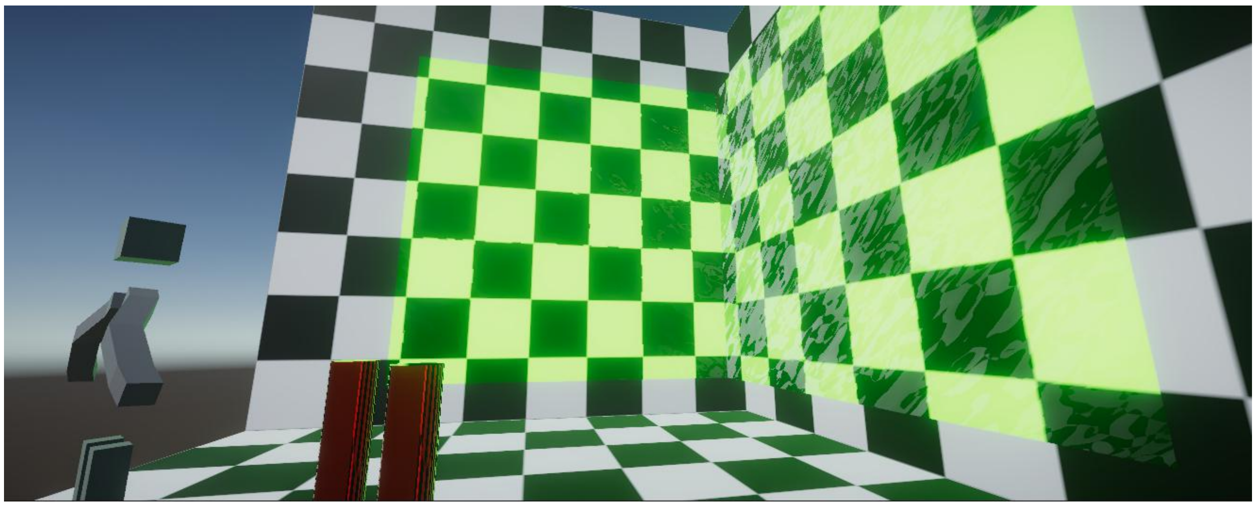
BoxHumanoid is a very simple Humanoid. Please use the scene in your own product by replacing it with the Humanoid of your own game or avatar.

XRWaterInteraction

A demo scene where you grab two balls floating on the surface of the water by gripping them and hit them on the surface of the water to generate waves. You can experience a VR scene using the most basic water surface.

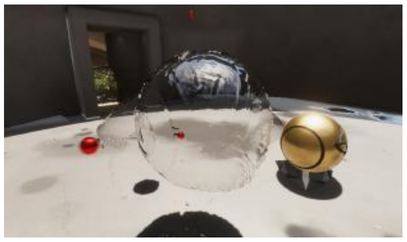
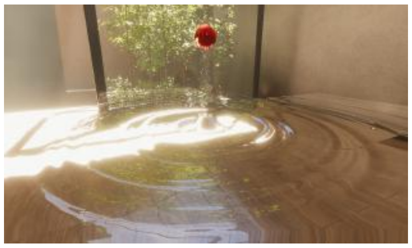
XRInterfererShooter

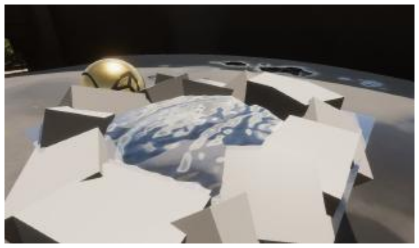
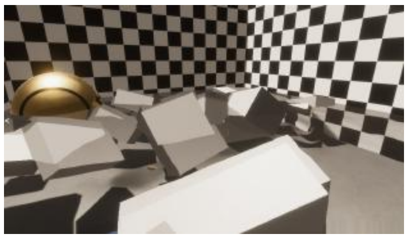
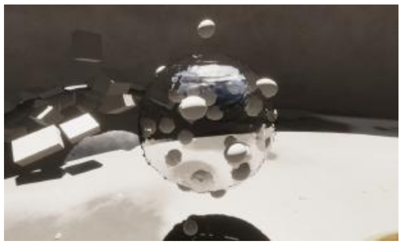
This is a sample scene in which two UseColor water surfaces are placed vertically and a Wave Interferer is shot into the water surface.

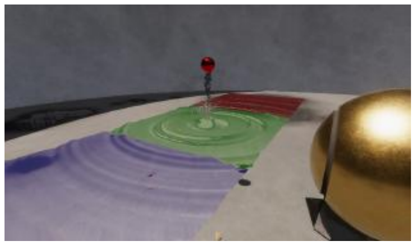
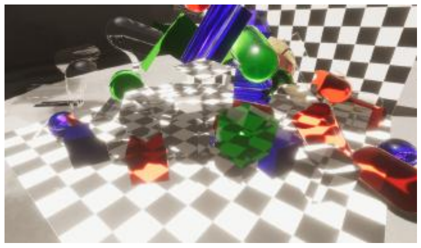
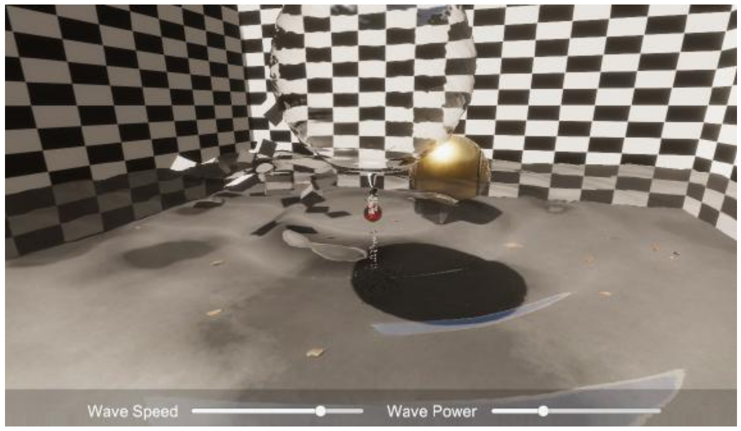


It is a test scene that works even if the water surface is placed vertically.

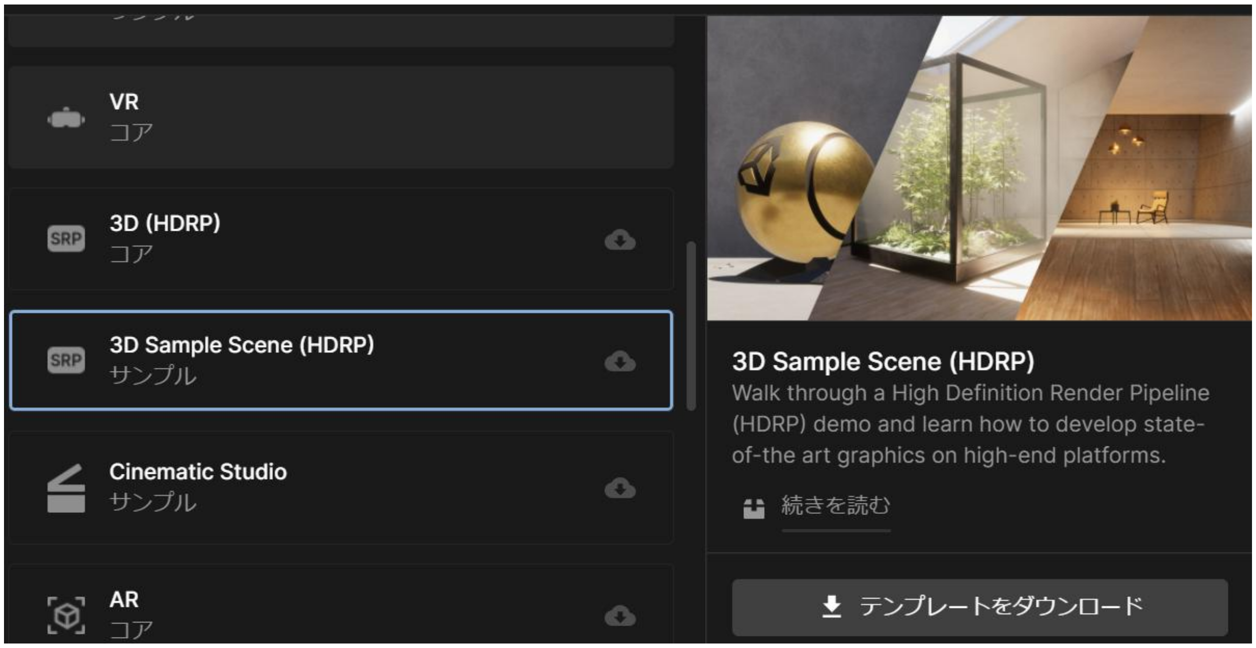
HDRP Sample Scenes

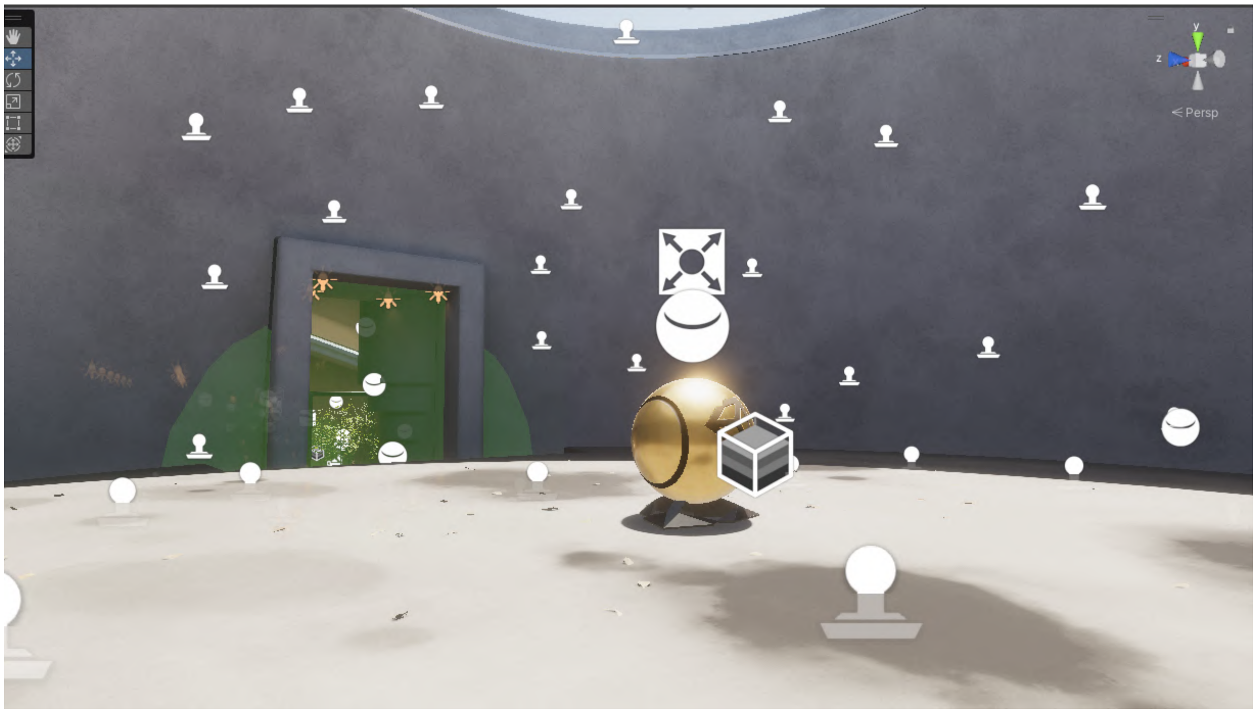




Create a new Unity project by choosing an HDRP template scene.



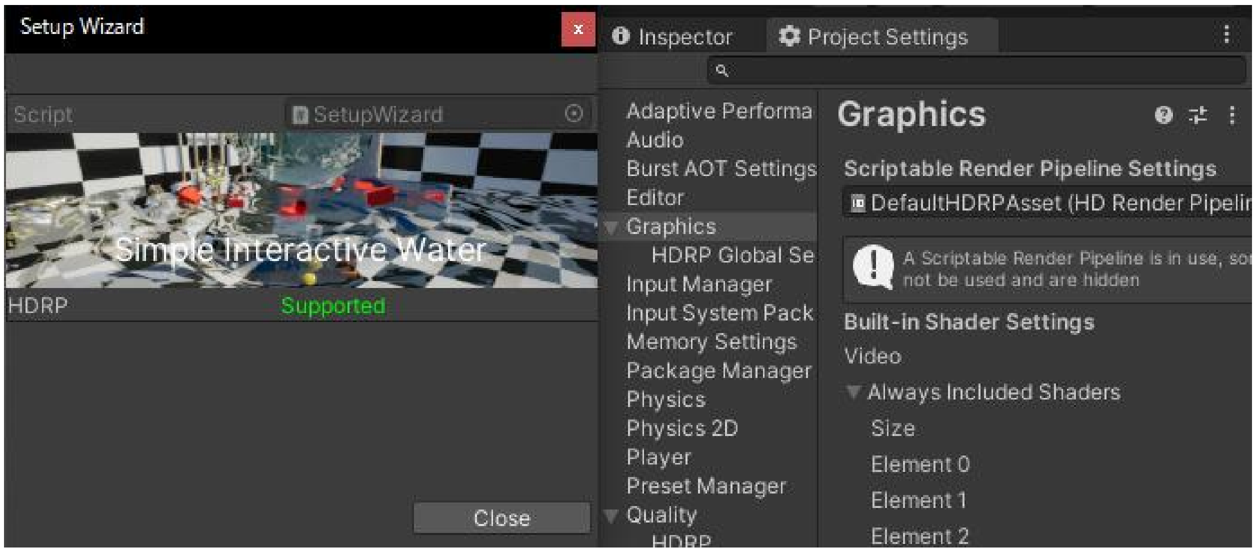
Walk through a High Definition Render Pipeline (HDRP) demo and learn how to develop state-of-the art graphics on high-end platforms.



Then import the Simple Interactive Water asset.

"This Unity Package has Package Manager dependencies." shows. Press the “Install/Upgrade” button.

The HDRP template comes with the new Input System installed from the beginning, so continue importing assets. Then select Window > Simple Interactive Water > Setup Wizard from the menu to open the setup wizard.


Setup Wizard

Script SetupWizard

Simple Interactive Water

HDRP Supported

Close

Inspector | Project Settings

Adaptive Performa
Audio
Burst AOT Settings
Editor
Graphics
HDRP Global Se
Input Manager
Input System Pack
Memory Settings
Package Manager
Physics
Physics 2D
Player
Preset Manager
Quality
HDRP

Graphics

Scriptable Render Pipeline Settings
DefaultHDRPAsset (HD Render Pipelin

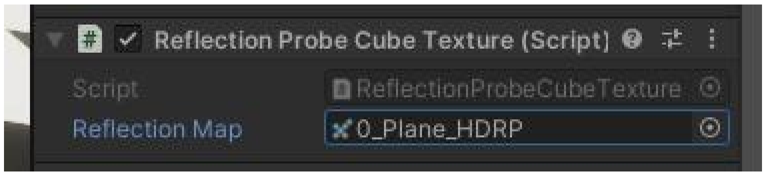
A Scriptable Render Pipeline is in use, so
not be used and are hidden

Built-in Shader Settings
Video
Always Included Shaders
Size
Element 0
Element 1
Element 2

After confirming that Supported is displayed, check the sample scene list. Open **Assets/SimplestarGame/SimpleInteractiveWater/Example/Scenes/HDRP/Sample Scenes for HDRP.unity** and press the Reset Scenes in Build button in the ScenesInBuildHDRP game object inspector that exists in the hierarchy. When the number of Scene Names reaches 14, you are ready to go.

Press the Play button and play the game to see a list of HDRP sample scenes. You can switch scenes by pressing the left and right arrow keys or pressing the left and right buttons on the screen. Since the contents are the same as those explained for URP, a detailed explanation of each scene is omitted. One point, in the HDRP version 3.0 shader, the value of the Reflection Probe cannot be obtained with the Shader Graph, so the Cube Map is written to the texture once, and the shader uses this to realize the reflection of the water surface. If you move the location of the water surface, press the Bake button of the Custom Reflection Probe attached to the Reflection Probe, which is a child object of the

Plane object. The Cube Texture referenced in the Reflection Probe Cube Texture below is updated.

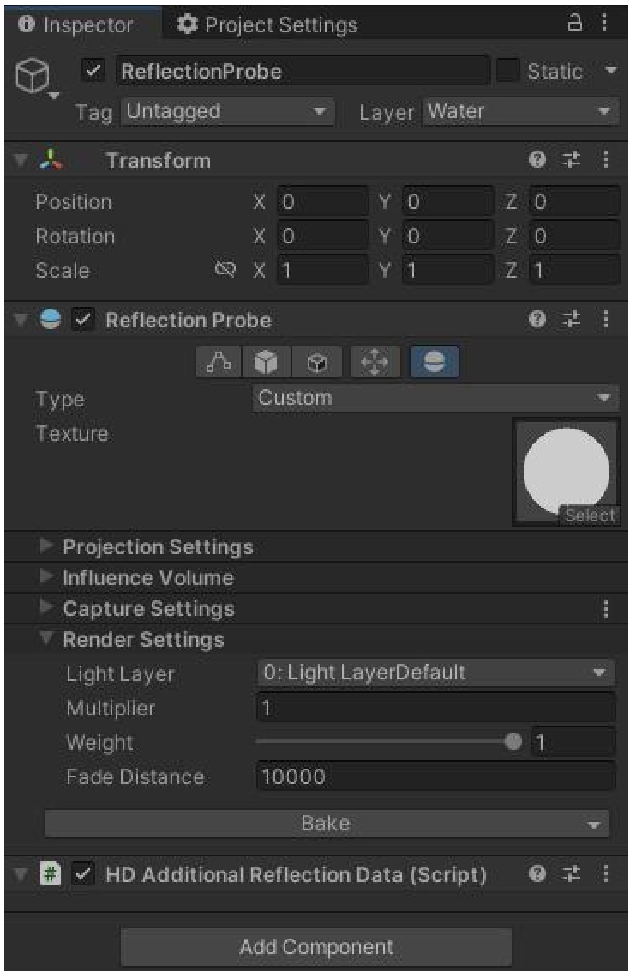


Reflection Probe Cube Texture (Script)

Script: ReflectionProbeCubeTexture

Reflection Map: 0_Plane_HDRP

It takes more time than 2.2. (Instead, you can adjust the intensity of the reflection compared to 2.2.)



ReflectionProbe — Static
Tag: Untagged — Layer: Water

Transform

Position	X 0	Y 0	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Reflection Probe

Type: Custom
Texture

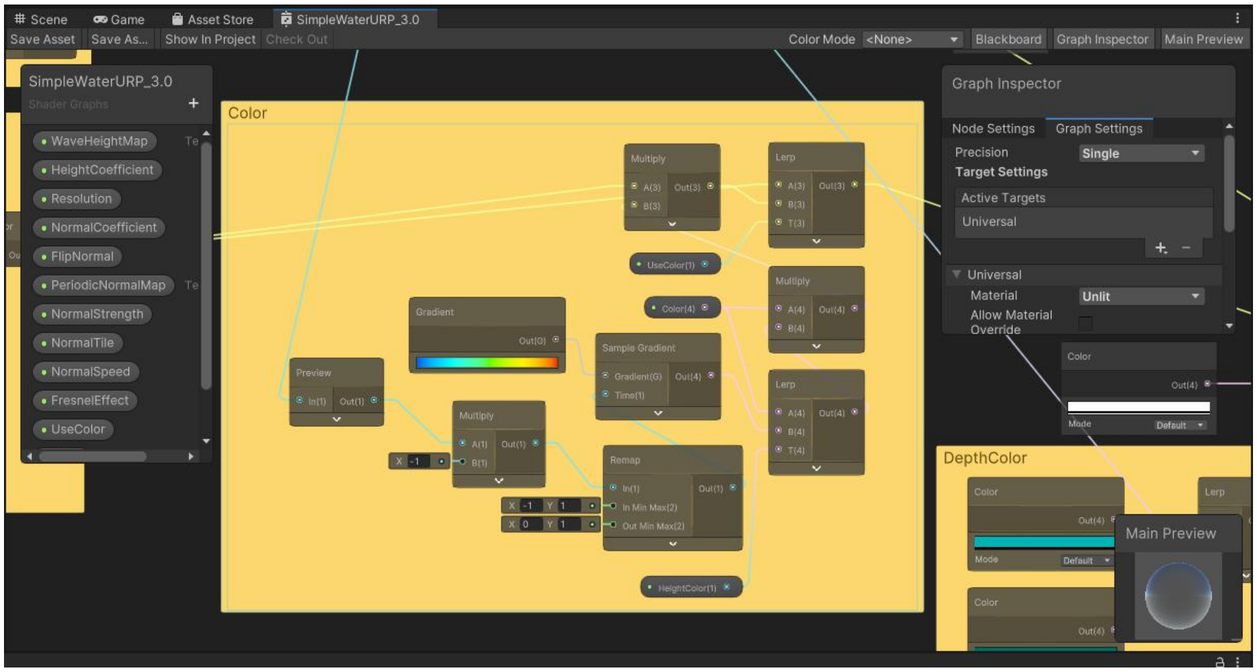
Render Settings

Light Layer: 0: Light LayerDefault
Multiplier: 1
Weight: 1
Fade Distance: 10000

HD Additional Reflection Data (Script)

Shader Graph

From here on, it's information for those who want to change the appearance of the water surface. By pressing the Edit button of the material attached to each water surface, you can open the corresponding Shader Graph edit screen. If you want to analyze the shader content and customize it to your own water surface expression, please understand the content of the Shader Graph here and edit it.



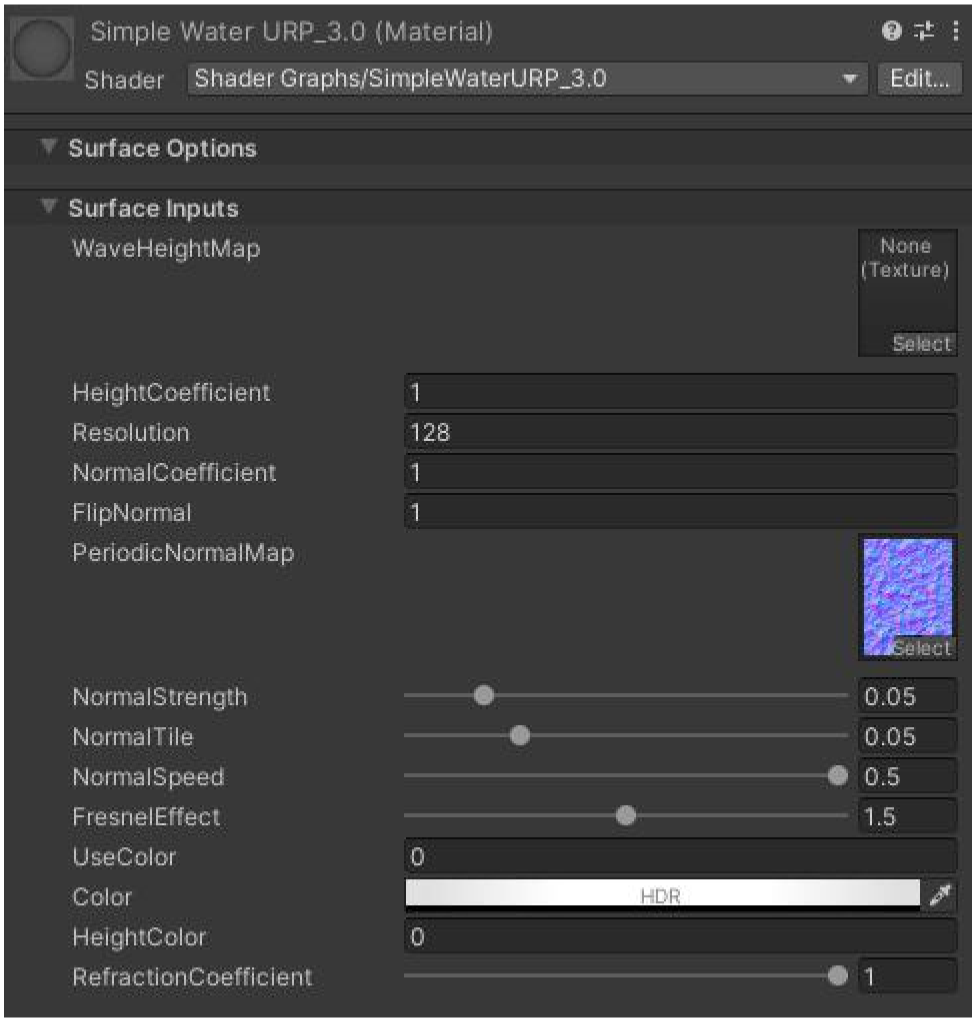
Graph Inspector
Node Settings | Graph Settings
Precision: Single
Target Settings
Active Targets
Universal
Universal
Material: Unlit
Allow Material Override

Uninstall

Uninstall is completed by deleting the Asset/SimplestarGame/SimpleInteractiveWater and SimpleXRAvatar folders.

Material Parameter

Describes the adjustment parameters for the water surface expression.



Simple Water URP_3.0 (Material)

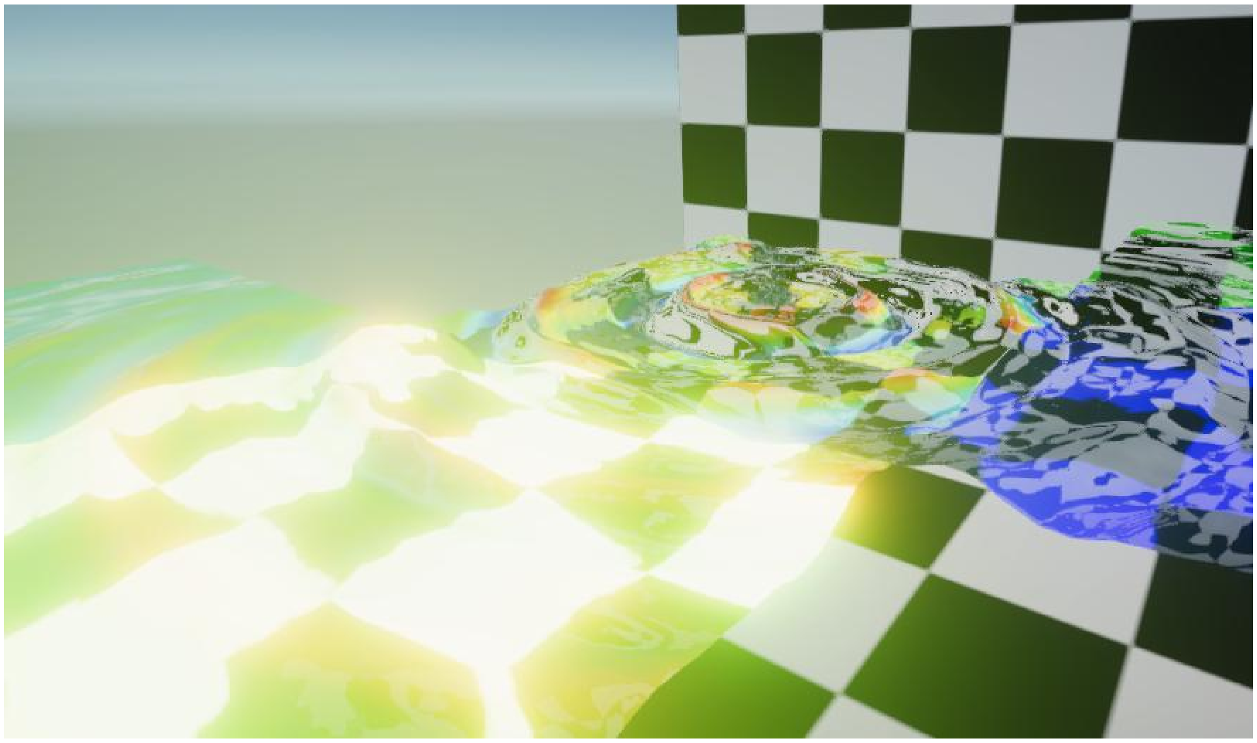
Shader: Shader Graphs/SimpleWaterURP_3.0

WaveHeightMap	None (Texture)
HeightCoefficient	1
Resolution	128
NormalCoefficient	1
FlipNormal	1
PeriodicNormalMap	
NormalStrength	0.05
NormalTile	0.05
NormalSpeed	0.5
FresnelEffect	1.5
UseColor	0
Color	HDR
HeightColor	0
RefractionCoefficient	1

WeveHeightMap: The HeightMap of the resolution waves created by Wave Simulator during gameplay is set here.
HeightCoefficient: Allows you to adjust the height of the waves for appearance, independent of the simulation.

Resolution: Displays the resolution of the HeightMap. Adjusting it will affect the calculation results, but please do not edit it as it will only corrupt the display.
NormalCoefficient: Allows adjusting the normal strength of the wave. If you want to make the water surface more visually impactful, it is effective to set a large value.
FlipNormal: Coefficient to switch to total internal reflection representation when the camera is submerged underwater. If you set it from 1 to -1, you can express the water surface with total reflection. This requires adjustments such as making the water surface a double-sided material.
PereodicNormalMap: Specifies the normal map texture used for small waves on the water surface in a steady state. A tiling texture is preferred. You can hide waves in the steady state by specifying None.
NormalStrength: Lets you adjust the normal strength of the wave in a steady state.
NormalTile: Lets you adjust the tiling density of steady state waves.
NormalSpeed: Lets you adjust the steady state wave speed.
FresnelEffect: You can adjust the reflection intensity reflected by the normal and viewing angle.

UseColor: Set to 1 to multiply the next Color value.
Color: The color to multiply the surface of the water. If you want to express shiny water, set a bright color as the HDR color.



HeightColor: Gradation will occur depending on wave height. Mainly a display option for debugging.

RefractionCoefficient: You can adjust the strength of the refraction representation of light on the water surface. Objects floating on the water surface may look strange when used to represent water refraction, so depending on the scene, setting this to 0 can avoid the strange appearance.