Water Stylized Shader Orto & Perspective Camera - Offline Documentation

Table of Contents

2. Introduction

Welcome to the **Water Stylized Shader Orto & Perspective Camera** documentation. This guide will help you install, configure, and understand how to use the asset effectively. It also contains a complete reference for the included scripts, offering details on their functionality and usage.

2. Package Contents

- Six preset materials that you can duplicate and modify to achieve the desired water aesthetic for your project.
- The original shaders and subgraphs are easy to modify and can be extended with additional features.
- One script that simulates floating behavior and wave displacement on objects using the sine function.
- Terrain data for the scene template (you can ignore this if you don't need it).
- A scene template featuring the preset materials applied in examples, along with a beautiful landscape.
- kMirrors dependency by *Matt Dean (https://github.com/Kink3d/kMirrors/wiki)*

3. Set up Guide

Follow these steps to get everything ready:

1. Import the package in an URP project (not tested in Built-In and HDRP).
2. If you want all the functionalities working properly, activate **Depth Texture** and **Opaque Texture** in the renderer settings asset that you are currently using.
3. Create a material or duplicate any of the example ones and in advanced settings of the material, change the render queue to **Transparent** (3000).
4. Create a plane and apply the material to it and everything should work fine. This is only tested with the default plane that you can add directly in the Unity Editor.
5. Get the **Reflection** object and add it to the scene to get planar reflections for your materials.

4. Material Properties Documentation

This section provides a description of every parameter found in the shader.

1 : General Settings

- **World Space UV:** A toggle that activates World Space UVs for calculations instead of local UVs. World Space UVs are useful if you want tiles that align seamlessly.
- **Perspective?:** A toggle that should be activated when the rendering camera is in perspective mode and deactivated when the camera is orthographic. This is because depth is calculated differently for these two modes.

2 : Deep

- **Depth Fade Distance:** Controls how deep the underwater silhouette appears.
- **Deep Color:** The color of the underwater silhouette.
- **Shallow Color:** The color of the water that doesn't have an underwater silhouette.
- **A short hike mode?:** Activates hard progression control by a set number of steps, instead of using a gradient, to manage the deep color.

3 : Horizon Fresnel

- **Horizon Distance:** Adjusts the horizon line for fresnel calculations.
- **Horizon Color:** The color added to the water when the player looks toward the horizon (when the angle between the surface normal and the view vector is 90º).

4 : Refraction (can cause image deformation sometimes)

- **Refraction Speed:** Controls how fast the distorted image moves.
- **Refraction & Gradient Scale:** Parameters that control the dimensions and noise applied to the distortion.
- **Refraction Strength:** Determines the intensity of the distortion.

5 : Surface Foam

- **Direction:** The direction the foam moves when speed is applied, controlled by a value between 0 and 1 (0 is down, 0.25 is right, 0.5 is up, 0.75 is left, and values in between indicate diagonal directions).
- **Speed:** Controls how fast the foam texture moves.
- **Tiling:** Adjusts the size of the foam texture.
- **Distorsion:** Adds a small amount of distortion to the texture.
- **Texture:** The main texture applied. Black-and-white textures work best.
- **Color:** The color of the applied texture. You can adjust the alpha for a softer effect on the water, and apply HDR intensity to create a bloom effect.

The same features apply to the secondary foam texture.

- **Foam UV Offset:** A small displacement applied to the secondary foam texture using the main texture's origin to generate an offset for interesting effects.

6 : Intersection Foam (is displayed where the plane is intersected)

- **Texture:** The texture applied to the intersection area.
- **Color:** The color applied to the texture, with alpha controlling transparency.
- **Depth:** Controls how far the intersection texture extends.
- **Fade:** Determines whether the effect is applied hard or soft (better controlled with the alpha of the color).
- **Tiling:** Adjusts the size of the intersection texture (typically noise scale).
- **Direction:** The direction the intersection foam takes when speed is applied.
- **Cutoff:** Controls the overall size, affecting both the texture and the effect.
- **Speed:** Adjusts how fast the foam moves in the applied direction.

7 : Normal surface & Lighting (Specular reflections)

- **Texture:** The normal map applied to simulate surface highlights.
- **Strength:** The intensity of the normal map effect.
- **Speed:** How fast the highlights change.
- **Scale:** The size of the normal map.
- **Lighting Smoothness & Hardness:** Control the properties of the specular reflections.
- **Specular Color:** The color of the specular highlights, with alpha controlling transparency.

8 : Waves (Vertex displacement)

- **Steepness and Length:** Control the height and length of the waves.
- **Speed:** How fast the waves change.
- **Directions:** The waves use **Gerstner Waves**, a complex but accurate simulation of wave displacement controlled by a Vector4. Each number in the vector takes a value from 0 to 1, indicating the direction of each small wave, which combine to form the overall wave pattern.

9 : Caves

- **Texture:** The texture applied to simulate random deep parts in the ocean.
- **Color:** Controls the color, with alpha managing transparency.
- **Scale:** Adjusts the size of the texture.
- **Distortion:** Creates a refraction effect.
- **Cave Offset:** Allows you to control which part of the texture is applied, in case you want to adjust the default result.

10 : Reflection

- **Reflections?:** Toggle to add reflections to the water (a **Reflection** object is needed to make this work).

- **Distortion:** Adjusts how distorted the reflections are.
- **Blend:** Controls the blending between the base color and the reflections. A value of 1 results in fully reflective water, while 0 means no reflection.

5. Script Reference

A reference guide for the script included in the asset. It is an extra feature: floating objects.

FloatingObject.cs

- **Description**: This script controls the behavior of a floating object in Unity, simulating buoyancy and the interaction between the object and water.

Public Variables:

- `Transform[] floaters`: An array of **Transform** components representing points on the object that will interact with the water. These are used to simulate the floatation behavior.
- `underWaterDrag` & `underWaterAngularDrag`: These values control how much drag (resistance) and angular drag the object experiences when underwater.
- `airWaterDrag` & `airWaterAngularDrag`: These values control the drag and angular drag when the object is in the air, above water.
- `floatingPower`: The force applied to the object when it is underwater to make it float.
- `baseWaterHeight`: The baseline water level. This is used to determine the average height of the water.
- `waterHeightVariation`: The amplitude of the wave height variation (how much the water height changes over time).
- `waveSpeed`: The speed at which the water height oscillates (the speed of the wave movement).
- `waterHeight`: The current water height, calculated based on time and the wave movement.

How it works:

- When a floater goes underwater, a buoyant force is applied upward, simulating the physics of floating. The more a floater is submerged, the stronger the force.
- If none of the floaters are underwater, the object is considered to be out of the water and behaves accordingly.
- The drag properties change dynamically depending on whether the object is submerged or above water, giving a more realistic simulation.

User knowledge: Attach the script to an object and set the base water height to match the water surface you are using. Adjust the height variation and speed to simulate water waves, if applicable. Modify the drag values to give the object the desired physical properties.

8. Contact Information

For support or further inquiries, please contact me at:

- **Email**: auredevgames@gmail.com
- **Website**: https://aerisway.github.io/AerisPortfolio/
- **Linkedin:** https://www.linkedin.com/in/auredevgames/