Unity & Unity Pro Asset

Ultimate Toon Water V1.10

Peculiar Developer

22nd March, 2014

Last update: 13th July 2014

Important Update Notice

A more modular and optimized version of UTW was introduced in this package. This will replace the old version over time. Migration is very simple and should take no more than 10 minutes. Please read the section **Setup – Upgrade**.

1 Contents

2 Introduction

Ultimate toon water is an all-in-one solution for Cartoon/Toon water. Suitable for normal Windows/Linux/Mac/Web builds as well as mobile.

2.1 Features

- GPU rendered cartoon based water
- Multiple dynamic physics waves
- Two texture mixing
- Texture/Color mixing
- Texture size and movement
- Wave steepness coloring
- Water transparency
- Cartoon wave particles (like in Zelda – The Wind Waker)
- Floater particles
- Automatic mesh generation
- Automatic target following
- Can be placed and moved anywhere while maintaining graphics and physics
- Shoreline FX (PRO only)
- Demo boat and floaters script (including particle FX)
- Easy edit inspector
- Runtime changes possible

3 Setup

3.1 Demo scenes

There are 4 demo scenes that should work right away. Legacy shows the updated version of the non-modular setup. The floater particles and wave particles have been detached in this version to enhance mobility even in the legacy version. The Normal, Hell, Swamp demo scenes are identical with the exception of the skybox, lighting, rendersettings, and some particles **Water Config** file. The **Water Config** files have a modular setup for water config. Simply choosing another **Water Config** file allows you to completely change the appearance of the water while remaining the prefab.

3.2 Fresh Setup

Drag **Water** (UltimateToonWater) or **WaterC** (UltimateToonWaterC) from "UltimateToonWater\Media\Prefabs\" to your scene. You should see your water right away. From here you can start configuration.

You can choose the **Water Config** from the project explorer you want to use for this water with **WaterC**. Directly editing the gameobject's properties should also update the **Water Config** in the project explorer!

Optionally for a scene with a simple floating object (buoy) and a boat with controller, also drag in **buoy**, **Lil_Ship** and **Main Camera Boat** from the same folder. All these prefabs should automatically find the missing references on runtime. You can set these manually if you like.

3.3 Upgrade

Upgrading to the current version is necessary for future support. The legacy version will not receive all the updates in the future (including lighting & shadows). To upgrade, please follow the following steps:

1. Create a new **Water Config** file (Right Mouse button in the project explorer, Create, **Water Config**).
2. Rename the file (for your convenience) to something you like.
3. Open the scene with the water you want to upgrade.
4. Click on the **Water** that you want to upgrade.
5. Manually (sorry) take over the corresponding properties of the "Basic settings" and "Appearance" to the new **Water Config** you just made.
6. Note the other settings of the old **Water** (Location, Other components, Use GPU rendering, Generate Waves, etc.)
7. Remove (or disable for reference) the Water.
8. Add the **WaterC** prefab to your scene.
9. Set the "Other Settings" of the old **Water** to **WaterC** in your scene as taken in step 6.
10. Choose the **Water Config** in the **WaterC** field.
11. The basic upgrade should be complete.
12. Change **UltimateToonWater** references in classes to **UltimateToonWaterC** to upgrade your own scripts.

NOTE: Depending on the version installed. The height offset could be changed. Please change this where needed.

3.4 Configuration

The configuration section is based on the current **WaterC** version but is also applicable to the **Water** version. If you have selected "Preview water in editor." You can see the wave generation during editing. This allows you to completely configure the water while editing. Do note that Shoreline FX aren't visibile during editing, but are visible during playing (even in the editor) if you have Unity Pro.

3.4.1 Appearance

Here you can setup the color of the water, the color of the waves and shoreline FX as well as the transparency of the water. Besides that, there are also several texture options available (which textures, title, scrolling, mixing between texture and color and mixing between two textures). The last features that are available is the height coloring and the strength of the shorelines.

3.4.2 Waveforms

In this item you can setup the Wave forms of your water. The first two are rendered in the GPU and it is recommended to not use more than 2 except when strictly needed. The wave speed, length, height and offset can be determined per wave.

4 Programming

Every property that can be changed in the inspector can also be changed during runtime. This can be done by calling to an instance/reference of **UltimateToonWaterC** or **UltimateToonWater** to e.g. UTW. For the legacy version the properties can be changed directly (e.g. UTW.size or UTW.waterColor or UTW.waveForms[0].speed.x) and for the current version via the config (e.g. UTW.config.size or UTW.config.waterColor or UTW.config.waveForms[0].speed.x). Changes during runtime for the current version are **not** stored to the **Water Config**!

After changing certain properties it is required to update the material. **Do not directly update the material using your own code!** Instead use UTW.generateMesh() for updating the mesh size or meshpointdistance or UTW.UpdateTextureSize(), UTW.UpdateColor(), UTW. UpdateTextures() or UTW. UpdateTextureProps() for the corresponding change you have made.

4.1 Get water height.

If you have a reference to a **UltimateToonWater** instance, you can retrieve it by calling UTW. getHeightByPos(Vector2 pos) or UTW. getHeightByPos(Vector3 pos). This is the most important function if you want to implement your own physics.

5 Known limitations

5.1 GPU vs CPU rendering

Always use GPU rendering if possible (however some older mobile devices may be faster with CPU rendering). Furthermore please refrain from using more than 2 waveforms. The third waveform (and above) will be rendered in the CPU and could result in performance loss.

5.2 Size and Meshpoint Distance

The water cannot consist out of more than 65000 polygons like any other mesh. This means that for larger (sea-like) levels, a balance must be found between water size and meshpoint distance. The mesh generator is protected against meshes that are too big. Usually any meshpoint distance up to 10 gives a fair result in details and allows for a huge area of water (2500x2500 meter). If you combine that with the "Snap to target" feature it ensures an section of water wherever you go. If however you need a bigger area of water with a higher Meshpoint distance you can make a grid of water at the cost of performance. You could also consider a detail block of water and a larger block of water below that.

5.3 UTWIngameEditor

The **UTWIngameEditor** only works for the legacy version. This was done due to the fact that there was no preview function in prior versions. The IngameEditor will be removed with the removal of the Legacy version in later updates.

5.4 Lighting and Shadows

There is no full lighting and shadow support in the current version of Ultimate Toon Water. The primary reason is for maximum mobile performance. This feature will be added (targeted for the next release). Up to then pseudo-lighting can be achieved by changing the color settings.

5.5 Water does not update/load in editor/ all the same.

During editing there are several scenarios of where the water is not updated. This is due to Unity's memory policy. No materials may be leaking into the scene during runtime. This means that any preview is done by updating the sharedmaterial. However updating this is called on certain occasions and can be skipped in specific ones. Simply deselecting and selecting the water in the Hierarchy browser will automatically update the water.

Do note that during playing all the water should have it's appropriate properties.

If in case of multiple water gameobjects with different render setups, you can change the "Base Material" by a clone of the **SimpleWater** material. Duplicate **SimpleWater** and rename it to what you want it to be called. For the Legacy **Water** unfold the "Show Default Inspector" of the water in the scene you want to change. In here replace the "Base Material" to the duplicate you just made. For the current water (**WaterC**) navigate to your **Water Config** and unfold the "Show Default Inspector". In here replace the "Base Material" to the duplicate you just made.

5.6 Waveform & Height calculation

Height calculation is done up to 2 waveforms. This is done for performance reasons. This can be changed in the code if you absolutely need more than 2 waveforms for your height calculation.

6 Support

If you have any questions or suggestions, please use the Unity forum:

http://forum.unity3d.com/threads/228820-WIP-Ultimate-Toon-Water-%28Asset%29

Asset store update url can be found here: http://forum.unity3d.com/threads/ultimate-toon-water.237642/

You can also PM http://forum.unity3d.com/members/108812-Gobla , however this will not result in your question or suggestion being treated faster than the forum message.

The site http://www.peculiardeveloper.com/ will also be updated over time with the information about the asset.

7 Changelog of Manual

7.1 8th December 2014

- Updated to PDF.
- Complete revamp of the manual.

7.2 22th March 2014

- Initial version.