Water version 1.0.0d
*My email is "**kripto289@gmail.com**"*
*Discord channel **https://discord.gg/GUUZ9D96Uq***

Getting Started

▶️ Creating Water in the Scene

To quickly get started with KWS2:

1. **Right-click** in the *Hierarchy window*
2. Navigate to: `Effects → KWS Water → Dynamic Waves Simulation Zone`

This will automatically add to your scene:

- A base **ocean** with default water settings
  A **dynamic simulation zone** (used for shallow water equations)
  A **flow source** that creates directional movement (for rivers, waterfalls, etc.)

You're ready to see dynamic water in action immediately — no setup required.

🌊 Water Simulation Zones

Each **simulation zone** represents an area where dynamic shallow-water equations are calculated in real time. You can move, resize, or rotate the zone to define where and how the water flows.

> ⚠️ **Simulation zones do not automatically blend with each other** in the current version.
> This feature is planned for a future update.

🔁 Working with multiple zones

To create long rivers or extended flowing areas:

- Add **multiple simulation zones** along the terrain
- Place a **Water Source** at the start of each zone - this continues the flow visually and physically
- Adjust direction and strength of each source to maintain natural flow between zones

🛠️ Usage tips:

- Use one zone for small areas like ponds or waterfalls
- Use multiple zones for long or complex river paths
- Ocean and zones can coexist - zones simulate local flow over static ocean base

⚙️ Performance Tips & Zone Types

KWS2 is heavily optimized, but performance depends on how you configure simulation zones and particles.

🔧 General Performance Tips

- **Smaller simulation zones** = faster performance
- **Fewer foam/splash particles** = better performance
- **Splash shadow rendering** is extremely expensive. Use **Low Quality** or **Disable** if possible
  Use the **Intersection Layer Mask** to exclude non-interactive objects from simulation calculations
- **Lower simulation resolution per meter** = higher performance. Lower values reduce quality, but improve speed significantly

🧩 Zone Types

There are **three types of simulation zones**, each optimized for specific use cases:

---

1) Static Zone

A zone designed for stationary simulations such as rivers, lakes, or local flow areas.

- At startup, it captures the **current scene depth within the zone** and uses it to compute water-surface intersections
- The water flow dynamically adapts to the captured terrain and obstacles — allowing rivers to wrap around geometry
- You can change water sources or modify the flow path, and the simulation will react accordingly
- Optionally, you can use **Precompute Start** to bake the current scene depth and flow state in advance
  → This skips the filling phase and starts the simulation already “settled”
- When precomputed, the **cached depth texture is always used** — even if terrain or objects change after restarting
  → Intersections will no longer reflect scene updates unless rebaked

  🛠️ For procedural or runtime-generated worlds, you can manually reinitialize the zone using:
  `zone.ForceUpdateZone();`

---

2) Movable Zone

- Can move or follow objects (e.g. player-triggered simulations)

- **Does not cache depth**, so it won't interact with terrain or static objects
- Still supports interaction with **moving dynamic objects**

---

3) Baked Simulation

- Works only after pressing **"Precompute Start"**
- Caches the full state including **depth intersection and flow**
- **Completely ignores any zone or flow changes after baking**
  → Unlike *Static Zone*, the flow is frozen and will not respond to changes in water sources, terrain, or obstacles
- Ideal for **static rivers**, **foam-covered waterfalls**, or any areas where flow doesn't need to change
- No CPU/GPU cost for simulation after bake
  ⚠️ **Foam and splash particles still render** and consume GPU resources

🌊 Local Water Zones

You can create **local override zones** to change specific water properties in isolated areas — such as lakes, polluted water, stylized regions, shallow pools, or underwater rooms.

Each zone blends into the main water system based on distance and falloff settings.

🧩 Override Options

1. **Color Override**
   Override base water color within the zone with smooth blending into surrounding water.

2. **Water Height Override**
   Override ocean level locally to simulate isolated bodies of water (e.g. higher lakes or sunken areas).

3. **Wind Override**
   Reduce or increase wind intensity locally to simulate calm bays, protected harbors, etc.

🛠️ How to Use

- Navigate to: `Effects → KWS Water → Water Local Zone`
- Adjust zone size, falloff, and enabled properties
- Only affected water inside the zone will blend to new settings
- Zones **work in both dynamic and baked contexts**

⚠️ Known Limitations & Optimization Notes

To ensure stable performance and predictable behavior, keep in mind the following limitations and optimization guidelines:

🧩 Local Water Zones

- **Max visible zones on screen: 8**
- Using too many local override zones **can impact performance**, especially if they're large or overlap

---

🌊 Dynamic Simulation Zones

- **Best practice:** Split large simulations into **multiple smaller zones**
    - Enables **occlusion culling** for off-screen areas
    - Allows **adaptive update rates** based on camera distance
    - Particle systems (foam/splashes) **use LOD** logic tied to distance as well

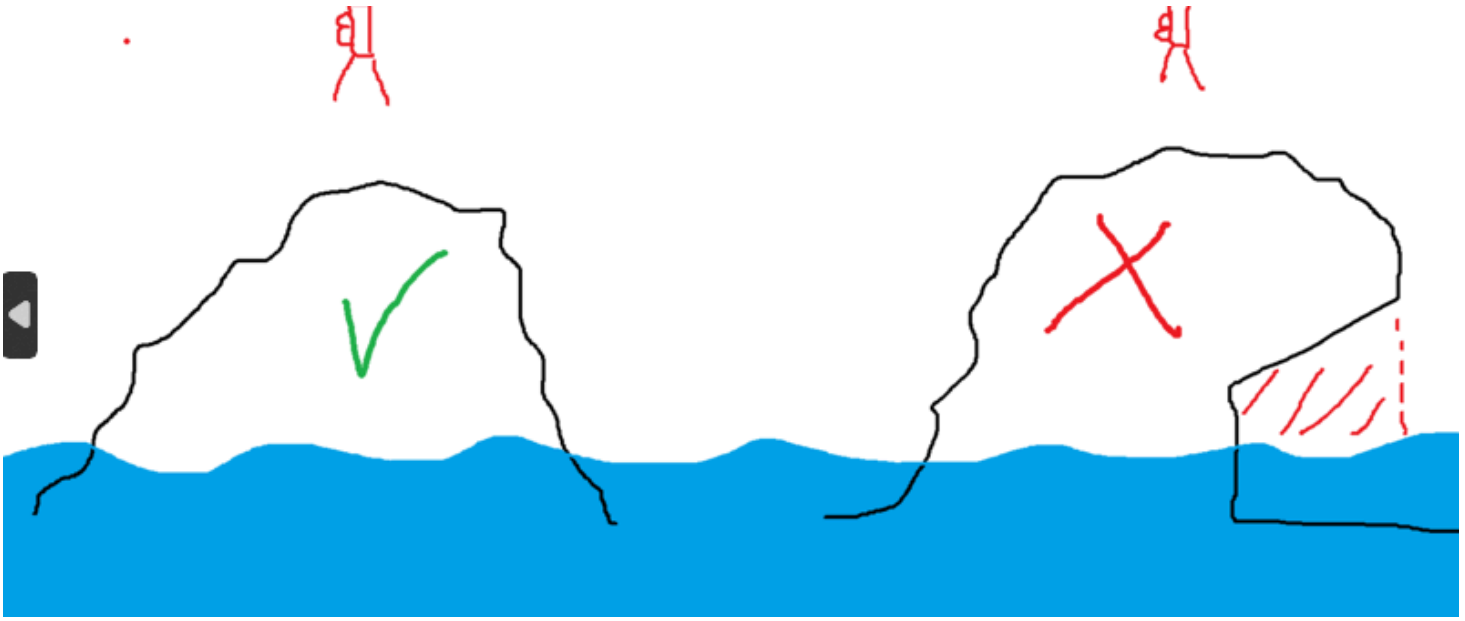
Multiple smaller zones offer significantly better performance than one massive simulation.

⚠️ Simulation Visibility & Obstacle Blocking

The dynamic water simulation works **purely from a top-down surface scan** (like a depth orthographic camera).
It **cannot detect occluded terrain under overhangs, caves, or concave geometry**.

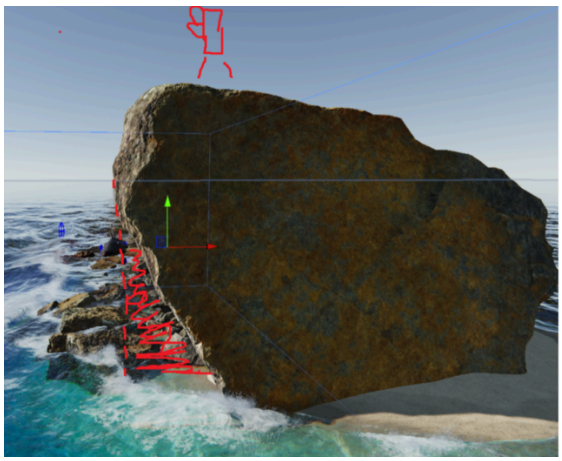
🚫 Incorrect Example: Overhead Rock Blocking Depth Sampling

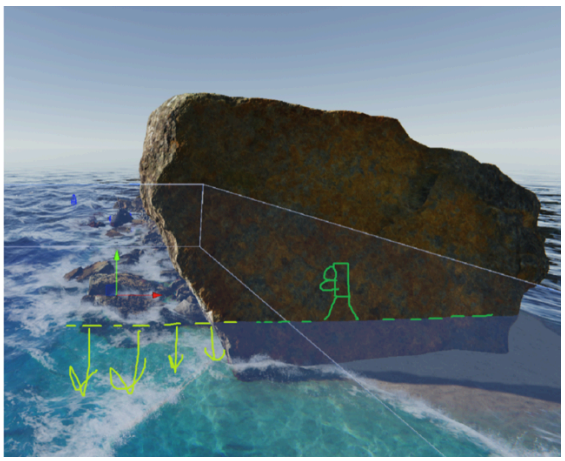
If the rock **visually hides** the ground from the simulation's perspective, water won't interact correctly.
You may see **foam or water climbing into the rock**.



✅ Correct Setup: Open Geometry or Shifted Zones

- Make sure terrain under the rock is **visible from the top** OR **move the simulation zone** to begin just beyond the obstruction





🧩 Global Water Quality Settings

All global quality settings for KWS2 are located in **Project Settings → KWS Water Settings**.
This section controls **which water features are active at different Unity Quality Levels** (Very Low → Ultra), as well as default resolution, lighting, and visual effects.

These settings allow for automatic quality scaling based on Unity's built-in quality level system

📋 Key Features of This Panel

- **Unity Quality Level Mapping**
  Toggle which water features are enabled per Unity quality tier (Low, Medium, High, etc.)
- **Reflections**
  - Enable or disable **Screen Space Reflections (SSR)** and **Planar Reflections**
  - Adjust **SSR resolution quality**
  - Toggle anisotropic reflections, skybox overrides, sunlight reflections

- **Refraction**
  - Choose between **physical approximation** or simple refraction
  - Enable **dispersion** for more realistic water distortion

- **Ocean Foam & Wetness**
  Toggle global foam and wet surface effects

- **Volumetric Lighting**
  - Control caustics strength, resolution, temporal accumulation
  - Enable additional lights if needed (note: higher GPU cost)

- **Underwater Effects**
  - Adjust internal reflection, Snell's window effect
  - Enable droplets, tension line distortion, or override transparency

- **Caustics**
  - Choose caustic simulation quality and enable high-quality filtering
  - Dispersion support for realistic colored caustics

- **Mesh Settings**
  - Control LOD and far-detailing range for water mesh generation

- **Rendering Settings**
  - Set transparent render order
  - Toggle depth writing, wide-angle compatibility, and fog integration

🧠 **Tip:** You can change the active Unity quality level at runtime or via code to dynamically adjust KWS features per platform or device.

🧠 Scripting & API Integration

KWS2 exposes a set of public APIs to help you sample the water surface, check underwater states, or override settings at runtime.

🌊 1. Sampling Water Height, Velocity, and Foam

You can sample dynamic water data using built-in request types:

private WaterSurfaceRequestArray _request = new WaterSurfaceRequestArray();

```
void Update()
{
   _request.SetNewPositions(_voxelsWorldPos);
   WaterSystem.TryGetWaterSurfaceData(_request);

   if (!_request.IsDataReady) return; //Usually ready after the first few simulation frames.

   for (int i = 0; i < _voxelsWorldPos.Length; i++)
   {
         var waterPos      = _request.Result[i].Position;
         var waterVelocity = _request.Result[i].Velocity;
         var foam          = _request.Result[i].Foam;
    }
}
```

Other supported types: WaterSurfaceRequestPoint, WaterSurfaceRequestList

⏱ 2. Time Synchronization for Multiplayer

To synchronize simulation time across clients or servers:

```
WaterSystem.UseNetworkTime = true;
WaterSystem.NetworkTime = yourTimeInSeconds;
```

📷 3. Camera & Rendering State Checks

Useful for triggering effects like diving, UI changes, post-processing:

```
bool isUnderwater = WaterSystem.IsCameraFullUnderwater;
bool isPartiallyUnder = WaterSystem.IsCameraPartialUnderwater;
```

You can also toggle rendering manually:

```
WaterSystem.Instance.WaterRenderingActive = false; // e.g., inside caves
```

Check if any point is submerged:
var isPointUnderwater = WaterSystem.IsPositionUnderWater(position);

🧪 4. Runtime Quality Overrides

Override specific features at runtime:
`WaterSystem.QualitySettings.ScreenSpaceReflectionResolutionQuality = WaterQualityLevelSettings.ScreenSpaceReflectionResolutionQualityEnum.Low;`

🔧 5. Runtime Property Control

Directly change visual or physical properties:
WaterSystem.Instance.CausticStrength = 0.5f;