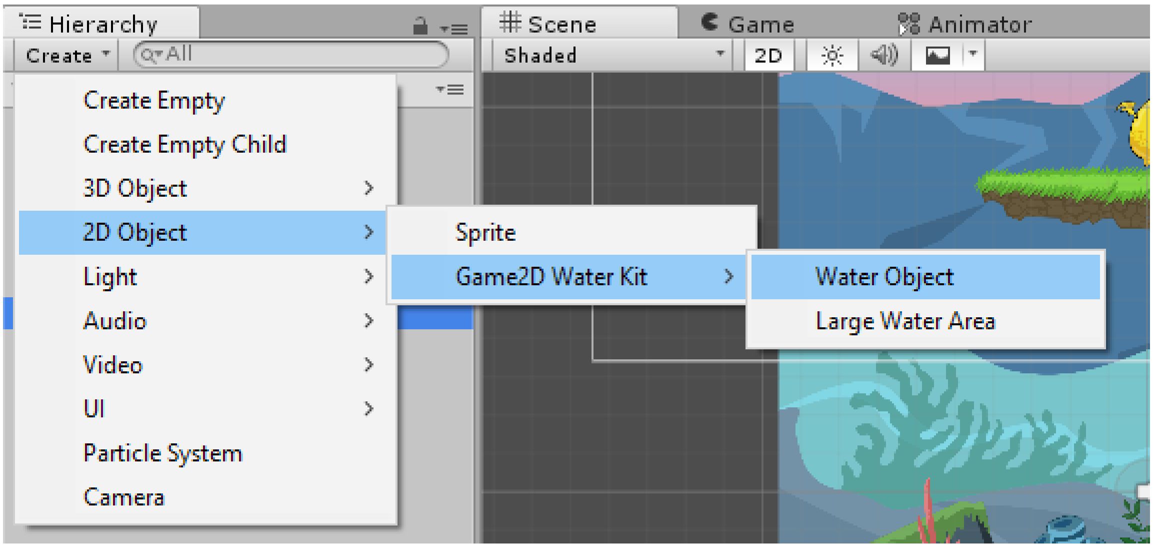
Game2D Water Kit v1.3 Documentation

Getting Started:

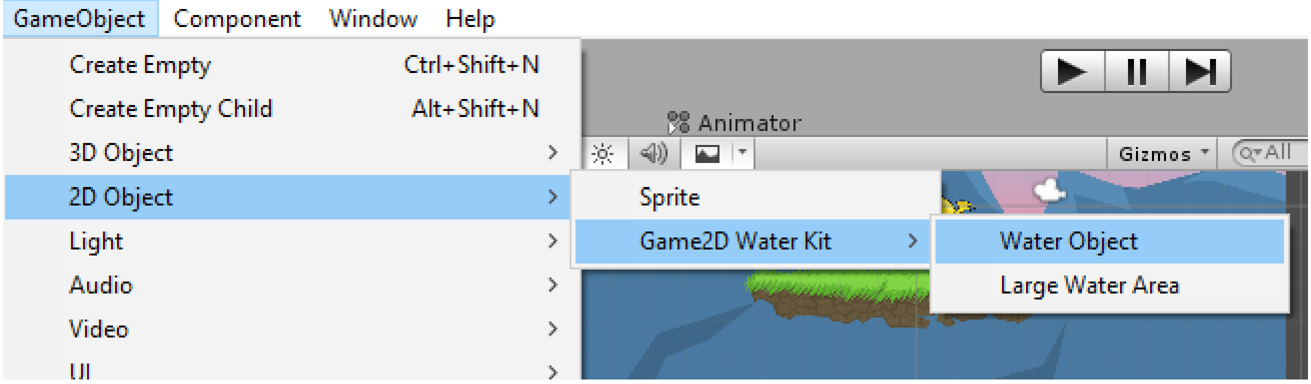
Adding a simple water object:

You can add a water object to your scene in the following ways:

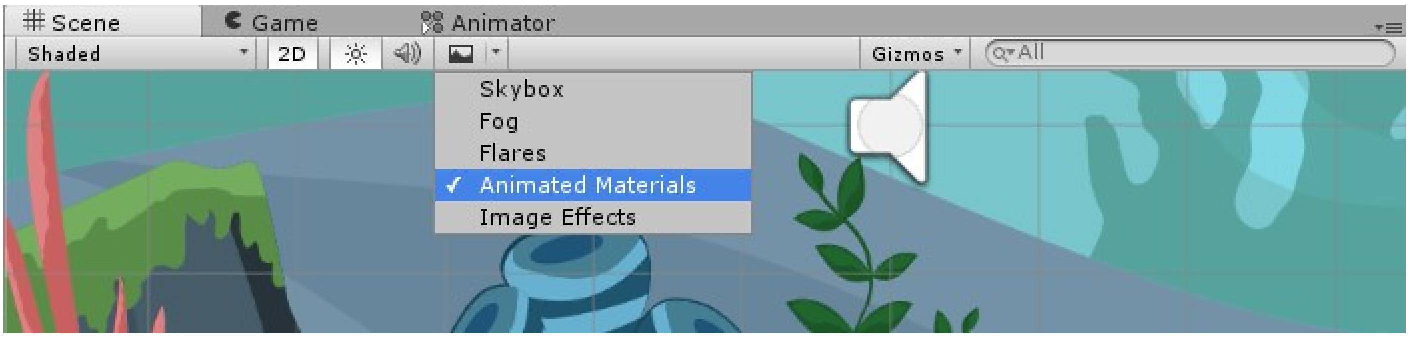
- From the hierarchy: **Create → 2D Object → Game2D Water Kit → Water Object**



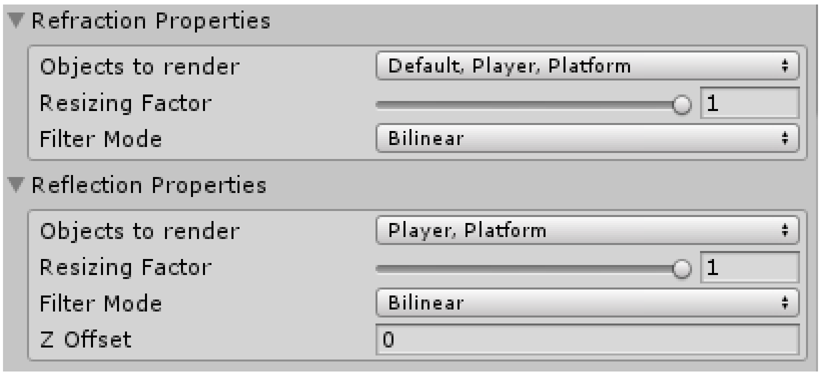
- From the top menu: **GameObject → 2D Object → Game2D Water Kit → Water Object**



The water refraction and the water reflection are not enabled by default. You can easily activate them in the material inspector. Please make sure that "Animated Materials" is checked to be able to visualize the distortion effects in the scene view.



You can then select which layers to render by the water refraction camera and the water reflection camera.



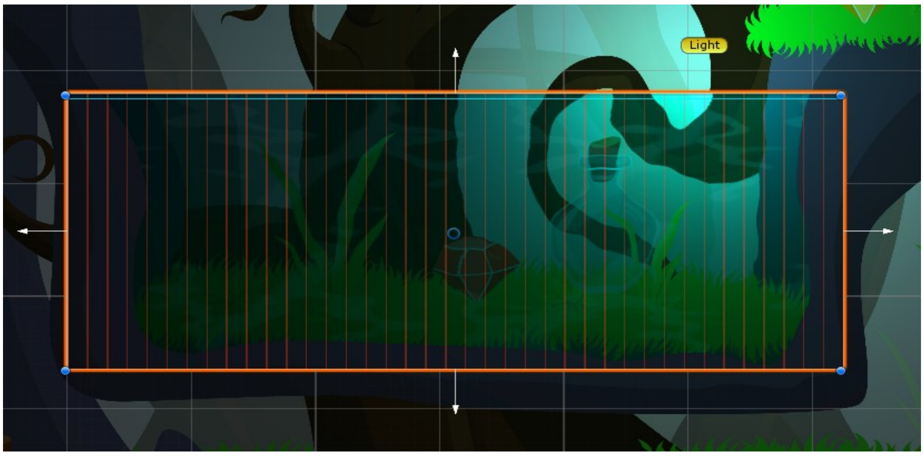
▼ Refraction Properties

Objects to render: Default, Player, Platform
Resizing Factor: 1
Filter Mode: Bilinear

▼ Reflection Properties

Objects to render: Player, Platform
Resizing Factor: 1
Filter Mode: Bilinear
Z Offset: 0

You can change the water size either in the inspector or by using the handles in the scene view.

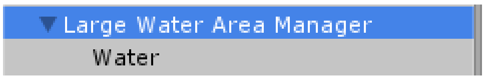


Now, you only need to tweak the other water properties to customize the feel and look of your 2D water.

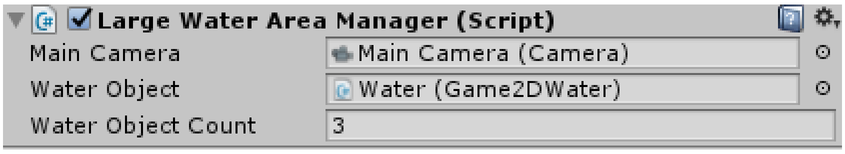
Adding a large water area:

You can add a large water area to your scene in the following ways:

- From the hierarchy: **Create → 2D Object → Game2D Water Kit → Large Water Area**
- From the top menu: **GameObject → 2D Object → Game2D Water Kit → Large Water Area**



▼ Large Water Area Manager
Water



☑ Large Water Area Manager (Script)

Main Camera: Main Camera (Camera)
Water Object: Water (Game2DWater)
Water Object Count: 3

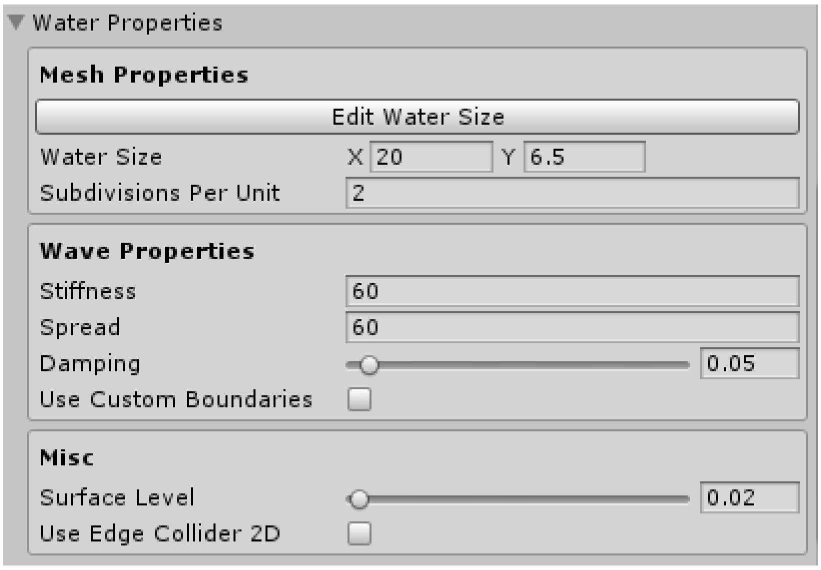
A new object is created with a Large Water Area Manager component attached to it and having a child water object.
The Large Water Area Manager component has 3 properties:

- **Main Camera:** is assigned by default to your scene main camera. This camera will be used to determine the visibility of each spawned water object. When a water object is no longer visible to this camera, it is respawned.
- **Water Object:** is assigned by default to the child water object. Please make sure that the water object width is at least half of the main camera viewing frustum width. It's also good to note that the water object doesn't necessarily have to be a child of the water manager.
- **Water Object Count:** has a default value of 3. This sets the number of water object to spawn when the game starts.

That's basically all what you have to do to setup the large water area manager.

Game2D Water Component Inspector:

★ Water Properties:



Water Properties

Mesh Properties

Water Size X 20 Y 6.5

Subdivisions Per Unit 2

Wave Properties

Stiffness 60

Spread 60

Damping 0.05

Use Custom Boundaries □

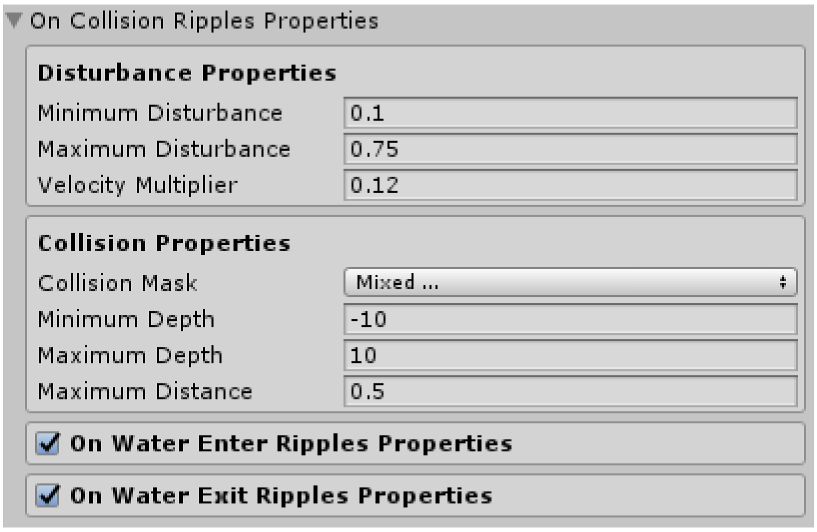
Misc

Surface Level 0.02

Use Edge Collider 2D □

- **Mesh Properties:**
  - **Water Size:** Sets the water size, the width and the height respectively.
    - Script: .MainModule.WaterSize [Vector2 get;]
    - Script: .MainModule.SetWaterSize(newSize:vector2,recomputeMesh:bool) [range: > 0]
  - **Subdivisions Per Unit:** Sets the number of water's surface vertices within one unit.
    - Script: .MeshModule.SubdivisionsPerUnit [int get;set] [range: > 0]
- **Wave Properties:**
  - **Stiffness:** Controls the frequency of wave vibration. A low value will make waves oscillate slowly, while a high value will make waves oscillate quickly.
    - Script: .SimulationModule.Stiffness [float get;set] [range: > 0]
  - **Spread:** Controls how fast the waves spread.
    - Script: .SimulationModule.Spread [float get;set] [range: > 0]
  - **Damping:** Controls how fast the waves decay. A low value will make waves oscillate for a long time, while a high value will make them oscillate for a short time.
    - Script: .SimulationModule.Damping [float get;set] [range: 0..1]
  - **Use Custom Boundaries:** Enables/Disables using custom wave boundaries. When a wave reaches a boundary, it bounces back.
    - Script: .SimulationModule.IsUsingCustomBoundaries [bool get;set]
    - ✔ **First Boundary:** The location of the first custom boundary.
      - Script: .SimulationModule.FirstCustomBoundary [float get;set] [range: -hww..hww]
    - ✔ **Second Boundary:** The location of the second custom boundary.
      - Script: .SimulationModule.SecondCustomBoundary [float get;set] [range: -hww..hww]
        - Note: hww in the range expression refers to “**h**alf **w**ater **w**idth”
- **Misc:**
  - **Surface Level:** Sets the surface location of the buoyancy fluid. When an object is above this line, no buoyancy forces are applied. When an object is intersecting or completely below this line, buoyancy forces are applied.
    - Script: .AttachedComponentsModule.BuoyancyEffectorSurfaceLevel [float get;set] [range 0..1]
  - **Use Edge Collider 2D:** Adds/Removes an EdgeCollider2D component. The water script takes care of updating the edge collider points.

★ On Collision Ripples Properties:



▼ On Collision Ripples Properties

Disturbance Properties

Minimum Disturbance 0.1
Maximum Disturbance 0.75
Velocity Multiplier 0.12

Collision Properties

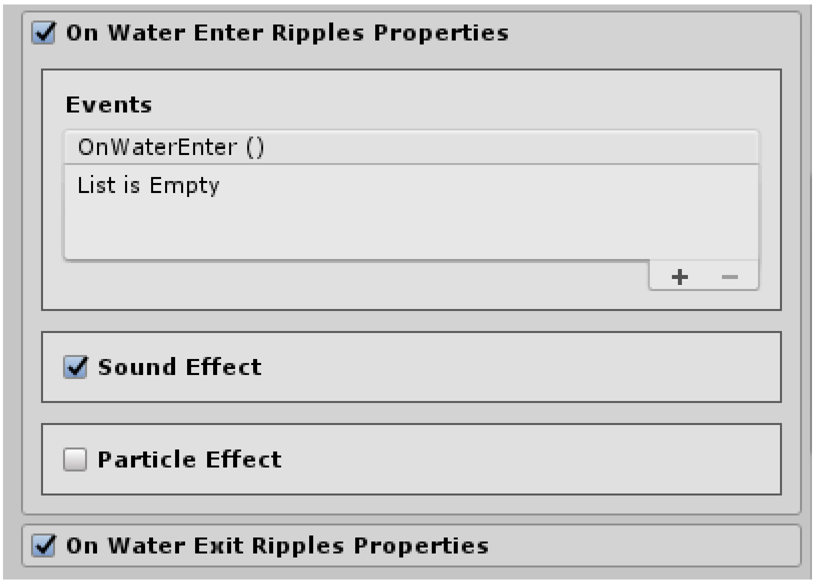
Collision Mask Mixed ...
Minimum Depth -10
Maximum Depth 10
Maximum Distance 0.5

☑ **On Water Enter Ripples Properties**

☑ **On Water Exit Ripples Properties**

- **Disturbance Properties:**
  - **Minimum Disturbance:** Sets the minimum displacement of the water’s surface.
    - Script: .OnCollisionRipplesModule.MinimumDisturbance [float get;set] [range: >0]
  - **Maximum Disturbance:** Sets the maximum displacement of the water’s surface.
    - Script: .OnCollisionRipplesModule.MaximumDisturbance [float get;set] [range: >0]
  - **Velocity Multiplier:** When an object falls into water or leaves the water, the amount of water’s surface displacement is determined by multiplying the object’s rigidbody velocity by this factor and clamping the result between the minimum and the maximum disturbance values.
    - Script: .OnCollisionRipplesModule.VelocityMultiplier [float get;set] [range: >0]

- **Collision Properties:**
  - **Collision Mask:** Only objects on these layers will disturb the water’s surface and will trigger the OnWaterEnter and the OnWaterExit events when they get into or out of the water.
    - Script: .OnCollisionRipplesModule.CollisionMask [LayerMask get;set]
  - **Minimum Depth:** Only objects with Z coordinate (depth) greater than or equal to this value will disturb the water’s surface.
    - Script: .OnCollisionRipplesModule.CollisionMinimumDepth [float get;set]
  - **Maximum Depth:** Only objects with Z coordinate (depth) less than or equal to this value will disturb the water’s surface.
    - Script: .OnCollisionRipplesModule.CollisionMaximumDepth [float get;set]
  - **Maximum Distance:** The maximum distance from the water's surface over which to check for collisions (Default: 0.5)
    - Script: .OnCollisionRipplesModule.CollisionRaycastMaximumDistance [float get;set] [range >0]



☑ On Water Enter Ripples Properties

Events

OnWaterEnter ()

List is Empty

☑ Sound Effect

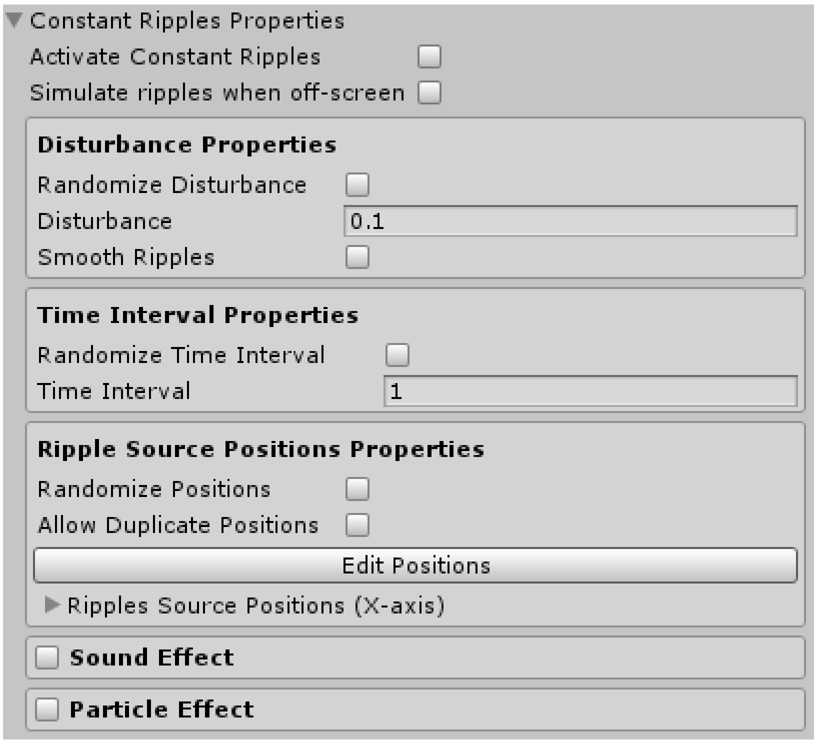
□ Particle Effect

☑ On Water Exit Ripples Properties

- **On Water Enter Ripples Properties:**
  - Activates/Deactivates generating ripples when an object falls into water.
    - Script: .OnCollisionRipplesModule.IsOnWaterEnterRipplesActive [bool get;set]
  - **OnWaterEnter:** This Unity Event is triggered when an object falls into water.
    - Script: .OnCollisionRipplesModule.OnWaterEnter [UnityEvent get;set]
  - **Activate Sound Effect:**
    - ✔ Activate/Deactivates playing the sound effect when an object falls into water.
      - Script: .OnCollisionRipplesModule.OnWaterEnterRipplesSoundEffect.IsActive [bool get;set]
    - ✔ For more information about ripples sound effect properties, see page 11
  - **Activate Particle Effect:**
    - ✔ Activate/Deactivates playing the particle effect when an object falls into water.
      - Script: .OnCollisionRipplesModule.OnWaterEnterRipplesParticleEffect.IsActive [bool get;set]
    - ✔ For more information about ripples particle effect properties, see page 12

- **On Water Exit Ripples Properties:**
  - Activates/Deactivates generating ripples when an object gets out of the water.
    - Script: .OnCollisionRipplesModule.IsOnWaterExitRipplesActive [bool get;set]
  - **OnWaterExit:** This Unity Event is triggered when an object gets out of the water.
    - Script: .OnCollisionRipplesModule.OnWaterExit [UnityEvent get;set]
  - **Activate Sound Effect:**
    - ✔ Activate/Deactivates playing the sound effect when an object gets out of water.
      - Script: .OnCollisionRipplesModule.OnWaterExitRipplesSoundEffect.IsActive [bool get;set]
    - ✔ For more information about ripples sound effect properties, see page 11
  - **Activate Particle Effect:**
    - ✔ Activate/Deactivates playing the particle effect when an object gets out of water.
      - Script: .OnCollisionRipplesModule.OnWaterExitRipplesParticleEffect.IsActive [bool get;set]
    - ✔ For more information about ripples particle effect properties, see page 12

★ Constant Ripples Properties:



▼ Constant Ripples Properties

Activate Constant Ripples □

Simulate ripples when off-screen □

Disturbance Properties

Randomize Disturbance □

Disturbance 0.1

Smooth Ripples □

Time Interval Properties

Randomize Time Interval □

Time Interval 1

Ripple Source Positions Properties

Randomize Positions □

Allow Duplicate Positions □

Edit Positions

▶ Ripples Source Positions (X-axis)

□ **Sound Effect**

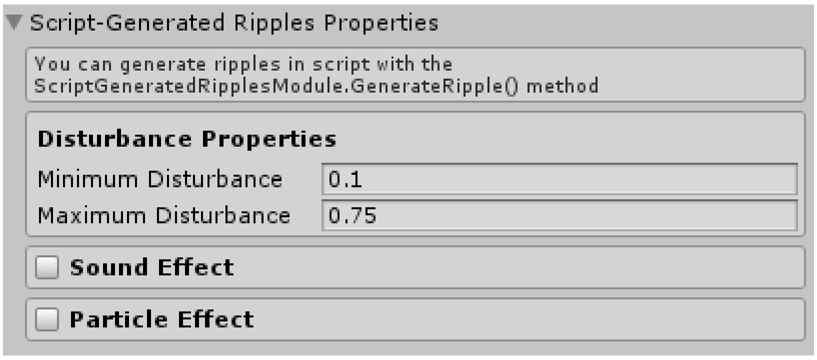
□ **Particle Effect**

- **Activate:** Activates/Deactivates generating ripples at regular time intervals.
  - Script: .ConstantRipplesModule.IsActive [bool get;set]
- **Update When off-screen:** Generate constant ripples even when the water is not visible to any camera in the scene.
  - Script: .ConstantRipplesModule.UpdateWhenOffscreen [bool get;set]
- **Disturbance Properties:**
  - **Disturbance:** Sets the displacement of the water's surface when generating constant ripples.
    - Script: .ConstantRipplesModule.Disturbance [float get;set] [range: >0]
  - **Randomize:** Randomize the disturbance (displacement) of the water's surface.
    - Script: .ConstantRipplesModule.RandomizeDisturbance [bool get;set]
  - **Minimum Disturbance:** Sets the minimum displacement of the water's surface.
    - Script: .ConstantRipplesModule.MinimumDisturbance [float get;set] [range: >0]
  - **Maximum Disturbance:** Sets the maximum displacement of the water's surface.
    - Script: .ConstantRipplesModule.MaximumDisturbance [float get;set] [range: >0]
  - **Smooth Ripples:** Disturb neighbor surface vertices to create a smoother ripple.
    - Script: .ConstantRipplesModule.SmoothRipples [bool get;set]
  - **Smoothing Factor:** The amount of disturbance to apply to neighbor surface vertices.
    - Script: .ConstantRipplesModule.SmoothingFactor [float get;set] [range: 0..1]
- **Interval Properties:**
  - **Interval:** Generate constant ripples at regular time intervals (expressed in seconds).
    - Script: .ConstantRipplesModule.TimeInterval [float get;set] [range: >0]
  - **Randomize:** Randomize the time interval.
    - Script: .ConstantRipplesModule.RandomizeTimeInterval [bool get;set]
  - **Minimum Interval:** Sets the minimum time interval.
    - Script: .ConstantRipplesModule.MinimumTimeInterval [float get;set] [range: >0]
  - **Maximum Interval:** Sets the maximum time interval.
    - Script: .ConstantRipplesModule.MaximumTimeInterval [float get;set] [range: >0]
- **Ripple Source Positions Properties:**

- **Randomize Positions:** Randomize constant ripples source positions. When checked, random surface vertices are disturbed each time the constant ripples are generated.
    - Script: .ConstantRipplesModule.RandomizeRipplesSourcePositions [bool get;set]
- **Ripples Source Count:** When Randomize Positions is checked, this sets the number of random surface vertices to disturb when generating constant ripples.
    - Script: .ConstantRipplesModule.RandomRipplesSourceCount [int get;set] [range: >0]
- **Allow Duplicate Positions:** Allow generating multiple ripples in the same position and at the same time.
    - Script: .ConstantRipplesModule.AllowDuplicateRipplesSourcePositions [bool get;set]
- **Edit Positions:** Edit constant ripples sources positions when "Randomize Positions" *is toggled off. When this button is active, you can choose, in the scene view, the surface vertices that are disturbed each time the constant ripples are generated.

- **Activate Sound Effect:**
    - Activate/Deactivates playing the sound effect when generating constant ripples.
        - Script: .ConstantRipplesModule.SoundEffect.IsActive [bool get;set]
    - For more information about ripples sound effect properties, see page 11
- **Activate Particle Effect:**
    - Activate/Deactivates playing the particle effect when generating constant ripples.
        - Script: .ConstantRipplesModule.ParticleEffect.IsActive [bool get;set]
    - For more information about ripples particle effect properties, see page 12

★ Script-Generated Ripples Properties:



You can generate ripples in script with the ScriptGeneratedRipplesModule.GenerateRipple() method

Disturbance Properties

Minimum Disturbance 0.1

Maximum Disturbance 0.75

□ **Sound Effect**

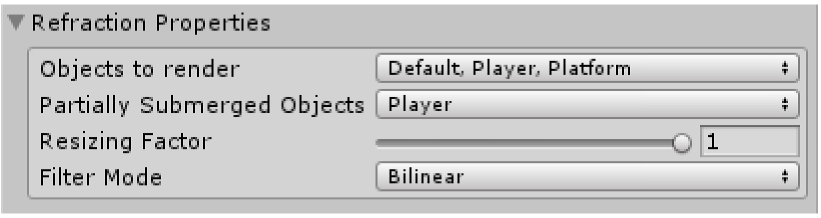
□ **Particle Effect**

- **Disturbance Properties:**
    - **Minimum Disturbance:** Sets the minimum displacement of the water's surface.
        - Script: .ScriptGeneratedRipplesModule.MinimumDisturbance [float get;set] [range >0]
    - **Maximum Disturbance:** Sets the maximum displacement of the water's surface.
        - Script: .ScriptGeneratedRipplesModule.MaximumDisturbance [float get;set] [range >0]
- **Activate Sound Effect:**
    - Activate/Deactivates playing the sound effect.
        - Script: .ScriptGeneratedRipplesModule.SoundEffect.IsActive [bool get;set]
    - For more information about ripples sound effect properties, see page 11
- **Activate Particle Effect:**
    - Activate/Deactivates playing the particle effect.
        - Script: .ScriptGeneratedRipplesModule.ParticleEffect.IsActive [bool get;set]
    - For more information about ripples particle effect properties, see page 12

➔ **To generate a ripple in code, you need to call:**

- .ScriptGeneratedRipplesModule.GenerateRipple([parameters])
  - [Parameters]:
    - ✔ **position [Vector2]:** The position where to generate the ripple. The nearest surface vertex to this position is disturbed.
    - ✔ **disturbanceFactor [float]:** Range: [0..1]: The disturbance to apply to the water surface is linearly interpolated between the minimum disturbance and the maximum disturbance by this factor.
    - ✔ **PullWaterDown [bool]:** When set to true, the water surface is pulled down as if an object falls into water, and when set to false, the water surface is pushed upward as if an object gets out of the water.
    - ✔ **playSoundEffect [bool]:** When set to true, the sound effect specified in the script-generated ripples properties is played.
    - ✔ **playParticleEffect [bool]:** When set to true, the particle effect specified in the script-generated ripples properties is played.
    - ✔ **Smooth [bool]:** When set to true, neighbor surface vertices are disturbed to create a smoother ripple.
    - ✔ **SmoothingFactor [float]:** Range: [0..1]: The amount of disturbance to apply to neighbor surface vertices.

★ Refraction Properties:

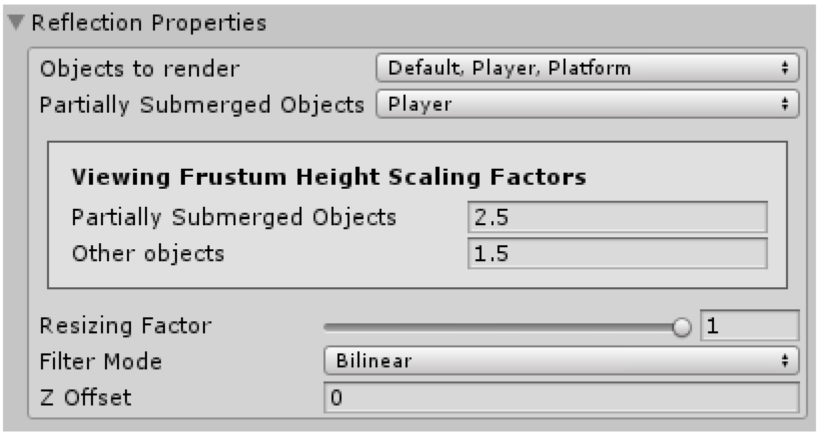


Refraction Properties

Objects to render	Default, Player, Platform
Partially Submerged Objects	Player
Resizing Factor	1
Filter Mode	Bilinear

- **Objects to render:** Only objects on these layers will be rendered by the water refraction camera.
  - Script: .RenderingModule.Refraction.CullingMask [LayerMask get;set]
- **Partially Submerged Objects:** Objects on these layers will be rendered as partially submerged into water when they intersect the submerge level (This option is only available when submerge level is toggled on in the material editor).
  - Script: .RenderingModule.RefractionPartiallySubmergedObjects.CullingMask [LayerMask get;set]
- **Resizing Factor:** Specifies how much the refraction RenderTexture is resized.
  - Script: .RenderingModule.Refraction.RenderTextureResizingFactor [float get;set] [range: 0..1]
  - Script: .RenderingModule.RefractionPartiallySubmergedObjects.RenderTextureResizingFactor [float get;set] [range: 0..1]
- **Filter Mode:** Sets the refraction RenderTexture filter mode.
  - Script: .RenderingModule.Refraction.RenderTextureFilterMode [FilterMode get;set]
  - Script: .RenderingModule.RefractionPartiallySubmergedObjects. RenderTextureFilterMode [FilterMode get;set]

★ Reflection Properties:



Reflection Properties

Objects to render: Default, Player, Platform

Partially Submerged Objects: Player

Viewing Frustum Height Scaling Factors

Partially Submerged Objects: 2.5

Other objects: 1.5

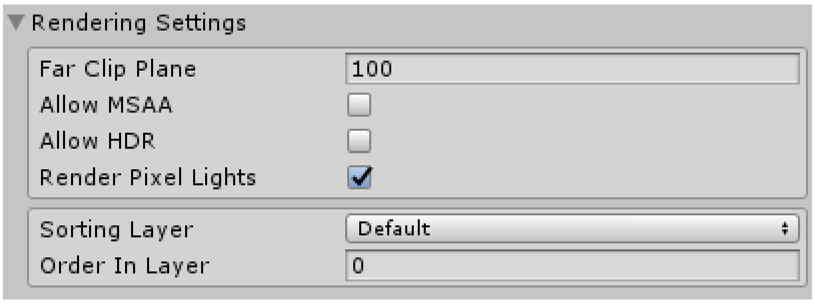
Resizing Factor: 1

Filter Mode: Bilinear

Z Offset: 0

- **Objects to render:** Only objects on these layers will be rendered by the water reflection camera.
  - Script: .RenderingModule.Reflection.CullingMask [LayerMask get;set]
- **Partially Submerged Objects:** Objects on these layers will be rendered by the partially submerged objects reflection camera (This option is only available when submerge level is toggled on in the material editor).
  - Script: .RenderingModule.ReflectionPartiallySubmergedObjects.CullingMask [LayerMask get;set]
- **Viewing Frustum Height Scaling Factor:**
  - **Partially Submerged Objects:** Sets how much to scale the partially submerged objects reflection camera viewing frustum height. The default viewing frustum height is equal to the distance between the surface level and the submerge level.
    - Script: .RenderingModule.ReflectionPartiallySubmergedObjects. ViewingFrustumHeightScalingFactor [float get;set] [range: >0]
  - **Other objects:** Sets how much to scale the reflection camera viewing frustum height when rendering other objects (all objects specified in ‘Objects to render’ layers except those specified in ‘Partially Submerged Objects’ layers). The default viewing frustum height for the reflection camera is equal to the surface thickness.
    - Script: .RenderingModule.Reflection.ViewingFrustumHeightScalingFactor [float get;set] [range: >0]
- **Resizing Factor:** Specifies how much the reflection RenderTexture is resized.
  - Script: .RenderingModule.Reflection.RenderTextureResizingFactor [float get;set] [range: 0..1]
  - Script: .RenderingModule.ReflectionPartiallySubmergedObjects.RenderTextureResizingFactor [float get;set] [range: 0..1]
- **Filter Mode:** Sets the reflection RenderTexture filter mode.
  - Script: .RenderingModule.Reflection.RenderTextureFilterMode [FilterMode get;set]
  - Script: .RenderingModule.ReflectionPartiallySubmergedObjects. RenderTextureFilterMode [FilterMode get;set]
- **Z Offset:** Controls where to start rendering the reflection relative to the water object position.
  - Script: .RenderingModule.Reflection.ZOffset [float get;set]
  - Script: .RenderingModule.ReflectionPartiallySubmergedObjects. ZOffset [float get;set]

★ Rendering Settings:



Rendering Settings

Far Clip Plane 100

Allow MSAA ☐

Allow HDR ☐

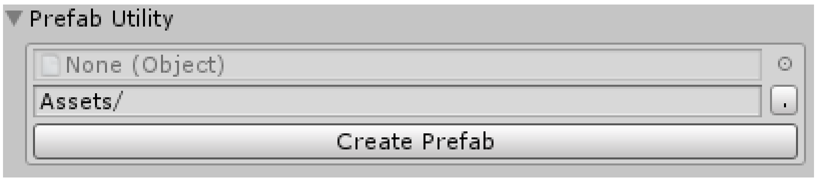
Render Pixel Lights ☑

Sorting Layer Default

Order In Layer 0

- **Far Clip Plane:** Sets the furthest point relative to the water that will be rendered by the refraction and/or the reflection cameras.
  - Script: .RenderingModule.FarClipPlane [float get;set] [range > 0.001]
- **Render Pixel Lights:** Controls whether the rendered objects will be affected by pixel lights. Disabling this parameter could increase performance at the expense of visual fidelity.
  - Script: .RenderingModule.RenderPixelLights [bool get;set]
- **Allow MSAA:** Allow multi-sample anti-aliasing rendering.
  - Script: .RenderingModule.AllowMSAA [bool get;set]
- **Allow HDR:** Allow high dynamic range rendering.
  - Script: .RenderingModule.AllowHDR [bool get;set]
- **Sorting Layer:** The name of the water mesh renderer sorting layer.
  - Script: .RenderingModule.SortingLayerID [int get;set]
- **Order In Layer:** The water mesh renderer order within a sorting layer.
  - Script: .RenderingModule.SortingOrder [int get;set]

★ Prefab Utility:



Prefab Utility

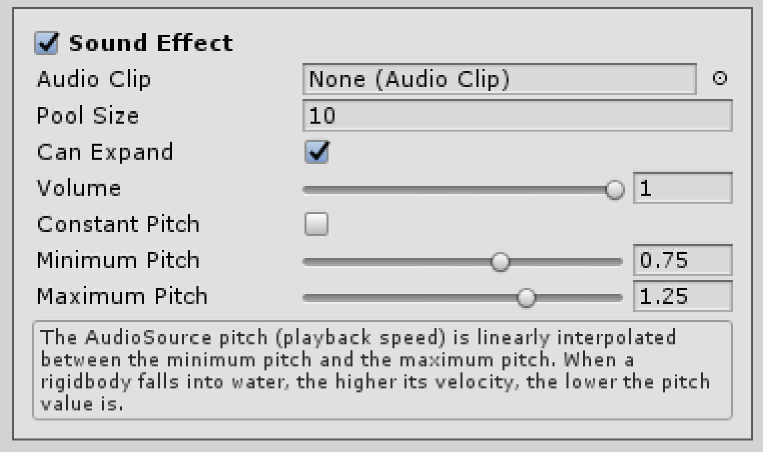
None (Object)

Assets/

- **Prefabs Path:** Specifies the prefabs destination path.
- **Create Prefab:** Creates a prefab in the specified path.
- **Unlink Prefab:** Breaks the prefab connection.
- **Relink Prefab:** Restores the prefab connection.

Note: *As of Unity 2018.3, disconnecting (unlinking) and relinking a Prefab instance are no longer supported. Alternatively, you can now unpack a Prefab instance if you want to entirely remove its link to its Prefab asset and thus be able to restructure the resulting plain object as you please.*

★ Sound Effect Properties:



☑ Sound Effect

Audio Clip: None (Audio Clip)

Pool Size: 10

Can Expand: ☑

Volume: 1

Constant Pitch: □

Minimum Pitch: 0.75

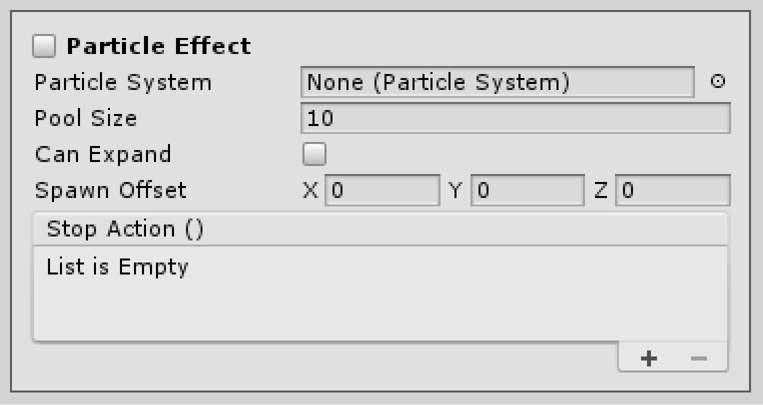
Maximum Pitch: 1.25

The AudioSource pitch (playback speed) is linearly interpolated between the minimum pitch and the maximum pitch. When a rigidbody falls into water, the higher its velocity, the lower the pitch value is.

- **Audio Clip:** The AudioClip asset to play.
  - Script: .OnCollisionRipplesModule.OnWaterEnterRipplesSoundEffect.AudioClip [AudioClip get;set]
  - Script: .OnCollisionRipplesModule.OnWaterExitRipplesSoundEffect.AudioClip [AudioClip get;set]
  - Script: .ConstantRipplesModule.SoundEffect.AudioClip [AudioClip get;set]
  - Script: .ScriptGeneratedRipplesModule.SoundEffect.AudioClip [AudioClip get;set]
- **Pool Size:** Sets the number of audio source objects that will be created and pooled when the game starts.
  - Script: .OnCollisionRipplesModule.OnWaterEnterRipplesSoundEffect.PoolSize [int get;set] [range: >0]
  - Script: .OnCollisionRipplesModule.OnWaterExitRipplesSoundEffect.PoolSize [int get;set] [range: >0]
  - Script: .ConstantRipplesModule.SoundEffect.PoolSize [int get;set] [range: >0]
  - Script: .ScriptGeneratedRipplesModule.SoundEffect.PoolSize [int get;set] [range: >0]
- **Can Expand:** Enables/Disables increasing the number of pooled audio source objects at runtime if needed.
  - Script: .OnCollisionRipplesModule.OnWaterEnterRipplesSoundEffect.CanExpandPool [bool get;set]
  - Script: .OnCollisionRipplesModule.OnWaterExitRipplesSoundEffect.CanExpandPool [bool get;set]
  - Script: .ConstantRipplesModule.SoundEffect.CanExpandPool [bool get;set]
  - Script: .ScriptGeneratedRipplesModule.SoundEffect.CanExpandPool [bool get;set]
- **Volume:** Sets the audio clip's volume.
  - Script: .OnCollisionRipplesModule.OnWaterEnterRipplesSoundEffect.AudioVolume [float get;set] [range: 0..1]
  - Script: .OnCollisionRipplesModule.OnWaterExitRipplesSoundEffect.AudioVolume [float get;set] [range: 0..1]
  - Script: .ConstantRipplesModule.SoundEffect.AudioVolume [float get;set] [range: 0..1]
  - Script: .ScriptGeneratedRipplesModule.SoundEffect.AudioVolume [float get;set] [range: 0..1]
- **Constant Pitch:** Apply constant audio clip playback speed.
  - Script: .OnCollisionRipplesModule.OnWaterEnterRipplesSoundEffect.IsUsingConstantAudioPitch [bool get;set]
  - Script: .OnCollisionRipplesModule.OnWaterExitRipplesSoundEffect.IsUsingConstantAudioPitch [bool get;set]
  - Script: .ConstantRipplesModule.SoundEffect.IsUsingConstantAudioPitch [bool get;set]

            - Script: .ScriptGeneratedRipplesModule.SoundEffect.IsUsingConstantAudioPitch [bool get;set]
        - **Pitch:** Sets the audio clip’s playback speed. (when ‘Constant Pitch’ is toggled on)
            - Script: .OnCollisionRipplesModule.OnWaterEnterRipplesSoundEffect.AudioPitch [float get;set] [range: -3..3]
            - Script: .OnCollisionRipplesModule.OnWaterExitRipplesSoundEffect.AudioPitch [float get;set] [range: -3..3]
            - Script: .ConstantRipplesModule.SoundEffect.AudioPitch [float get;set] [range: -3..3]
            - Script: .ScriptGeneratedRipplesModule.SoundEffect.AudioPitch [float get;set] [range: -3..3]
        - **Minimum Pitch:** Sets the audio clip’s minimum playback speed. (when ‘Constant Pitch’ is toggled off)
            - Script: .OnCollisionRipplesModule.OnWaterEnterRipplesSoundEffect.MinimumAudioPitch [float get;set] [range: -3..3]
            - Script: .OnCollisionRipplesModule.OnWaterExitRipplesSoundEffect.MinimumAudioPitch [float get;set] [range: -3..3]
            - Script: .ConstantRipplesModule.SoundEffect.MinimumAudioPitch [float get;set] [range: -3..3]
            - Script: .ScriptGeneratedRipplesModule.SoundEffect.MinimumAudioPitch [float get;set] [range: -3..3]
        - **Maximum Pitch:** Sets the audio clip’s maximum playback speed. (when constant pitch is toggled off)
            - Script: .OnCollisionRipplesModule.OnWaterEnterRipplesSoundEffect.MaximumAudioPitch [float get;set] [range: -3..3]
            - Script: .OnCollisionRipplesModule.OnWaterExitRipplesSoundEffect.MaximumAudioPitch [float get;set] [range: -3..3]
            - Script: .ConstantRipplesModule.SoundEffect.MaximumAudioPitch [float get;set] [range: -3..3]
            - Script: .ScriptGeneratedRipplesModule.SoundEffect.MaximumAudioPitch [float get;set] [range: -3..3]

★ **Particle Effect Properties:**



☐ **Particle Effect**

Particle System: None (Particle System)

Pool Size: 10

Can Expand: ☐

Spawn Offset: X 0 Y 0 Z 0

Stop Action ()

List is Empty

- **Particle System:** Sets the particle effect system to play.
    - Script: .OnCollisionRipplesModule.OnWaterEnterRipplesParticleEffect.ParticleSystem [ParticleSystem get;set]
    - Script: .OnCollisionRipplesModule.OnWaterExitRipplesParticleEffect.ParticleSystem [ParticleSystem get;set]
    - Script: .ConstantRipplesModule.ParticleEffect.ParticleSystem [ParticleSystem get;set]
    - Script: .ScriptGeneratedRipplesModule.ParticleEffect.ParticleSystem [ParticleSystem

get;set]

- **Pool Size:** Sets the number of particle system objects that will be created and pooled when the game starts.
    - Script: .OnCollisionRipplesModule.OnWaterEnterRipplesParticleEffect.PoolSize [int get;set] [range: >0]
    - Script: .OnCollisionRipplesModule.OnWaterExitRipplesParticleEffect.PoolSize [int get;set] [range: >0]
    - Script: .ConstantRipplesModule.ParticleEffect.PoolSize [int get;set] [range: >0]
    - Script: .ScriptGeneratedRipplesModule.ParticleEffect.PoolSize [int get;set] [range: >0]
- **Can Expand:** Enables/Disables increasing the number of pooled particle system objects at runtime if needed.
    - Script: .OnCollisionRipplesModule.OnWaterEnterRipplesParticleEffect.CanExpandPool [bool get;set]
    - Script: .OnCollisionRipplesModule.OnWaterExitRipplesParticleEffect.CanExpandPool [bool get;set]
    - Script: .ConstantRipplesModule.ParticleEffect.CanExpandPool [bool get;set]
    - Script: .ScriptGeneratedRipplesModule.ParticleEffect.CanExpandPool [bool get;set]
- **Spawn Offset:** Shifts the particle system spawn position.
    - Script: .OnCollisionRipplesModule.OnWaterEnterRipplesParticleEffect.SpawnOffset [Vector3 get;set]
    - Script: .OnCollisionRipplesModule.OnWaterExitRipplesParticleEffect.SpawnOffset [Vector3 get;set]
    - Script: .ConstantRipplesModule.ParticleEffect.SpawnOffset [Vector3 get;set]
    - Script: .ScriptGeneratedRipplesModule.ParticleEffect.SpawnOffset [Vector3 get;set]
- **Stop Action:** This UnityEvent is triggered when the particle system finishes playing.
    - Script: .OnCollisionRipplesModule.OnWaterEnterRipplesParticleEffect.StopAction [UnityEvent get;set]
    - Script: .OnCollisionRipplesModule.OnWaterExitRipplesParticleEffect.StopAction [UnityEvent get;set]
    - Script: .ConstantRipplesModule.ParticleEffect.StopAction [UnityEvent get;set]
    - Script: .ScriptGeneratedRipplesModule.ParticleEffect.StopAction [UnityEvent get;set]

★ Game2DWater script useful methods and properties:

- **.AnimationModule.AnimateWaterSize([parameters])**
    - **Description:** Smoothly animate the water size.
    - [Parameters]:
        - **targetSize:** [Vector2]: sets the water target size
        - **duration:** [float]: sets the animation duration in seconds.
        - **constraint:** [enum WaterAnimationConstraint]: constraints the position of one/multiple water edge(s).
            - Possible values: None, Top, Bottom, Left, Right, TopLeft, TopRight, BottomLeft, BottomRight
        - **wrapMode:** [enum WaterAnimationWrapMode]:
            - Possible values:
                - **Once:** Stops playing the animation once the target size is reached.
                - **Loop:** The water size is reset to the initial size and the animation is restarted once the target size is reached.

        - **PingPong:** When the target size is reached, the initial size will be the new target size. So the water size will ping-pong back and forth between the initial and the target size.

    - **.MaterialModule.IsUsingGradientColor:** [bool get]: Returns whether the water body color mode is set to a gradient color or a solid color.
    - **.MaterialModule.SolidColor:** [Color get;set]: Sets/Returns the water body color when the water body color mode is set to SolidColor.
    - **.MaterialModule.GradientStartColor:** [Color get;set]: Sets/Returns the water body gradient start color when the water body color mode is set to GradientColor.
    - .**MaterialModule.GradientEndColor** [Color get;set]: Sets/Returns the water body gradient end color when the water body color mode is set to GradientColor.
    - **.MaterialModule.SurfaceColor** [Color get;set]: Sets/Returns the water surface color.

★ LargeWaterAreaManager script useful methods:

- **.GetWaterObjectLocatedAt(float xPos):** returns the water object located at the specified position.

★ Notes:

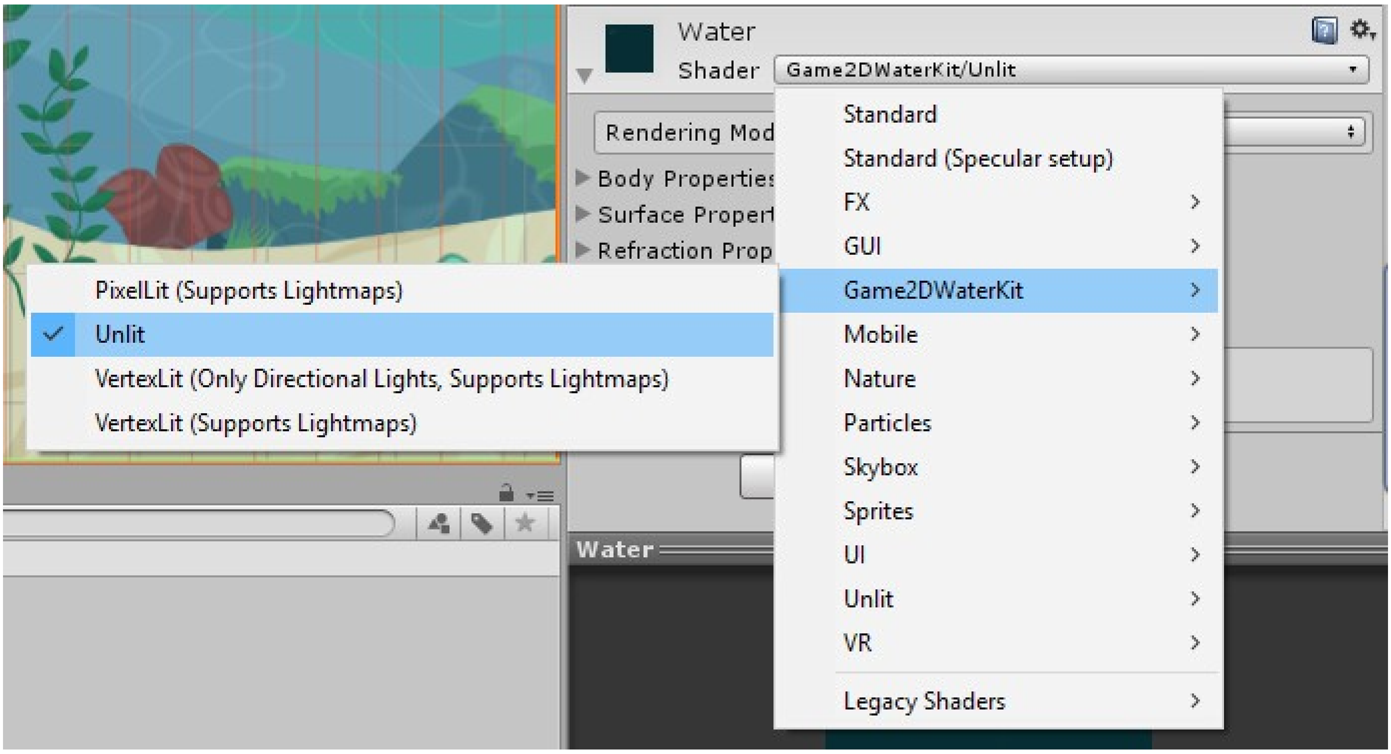
- ❑ The script folder contains a folder named “[Deprecated]”. You can safely delete this folder to get rid of deprecated functionality.
- ❑ **Only orthographic camera is currently supported.**
- ❑ *Skyboxes and Lens Flares are not rendered when rendering the water refraction and/or the water reflection.*
- ❑ *Please make sure the water object is **not** marked as Batching Static, otherwise the water refraction and the water reflection are not drawn properly.*

Game2D Water Material Inspector:

★ An overview:

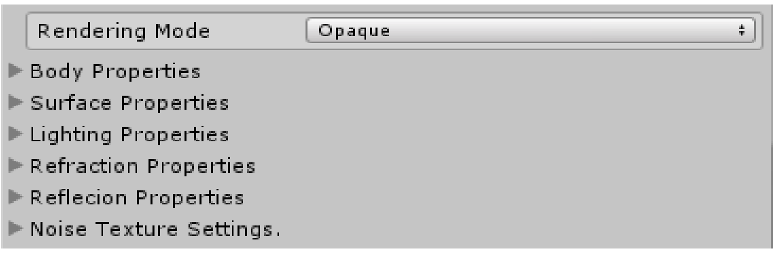
This asset comes with four shaders:

- Unlit
- VertexLit (Only Directional Lights / Supports Lightmaps)
- VertexLit (Supports Lightmaps)
- PixelLit (Supports Lightmaps)



- The Unlit shader is the fastest, no realtime lighting calculations are done.
- The VertexLit shaders are cheaper than the PixelLit shader.
- When using VertexLit shaders, all lights are rendered in a single pass and are calculated at water mesh vertices only.
- When using PixelLit shader, light is calculated at each pixel that is drawn and each light is rendered in its own pass. So, the water object may have to be drawn more than once.
- The PixelLit shader comes at performance cost and it’s not recommended to use it on mobile platforms.
- These shaders come with various features that are enabled or disabled by simply using or not using the texture slots and parameters in the material editor, so the unused shader functions are discarded at runtime automatically with no extra overhead.

★ Material Inspector:

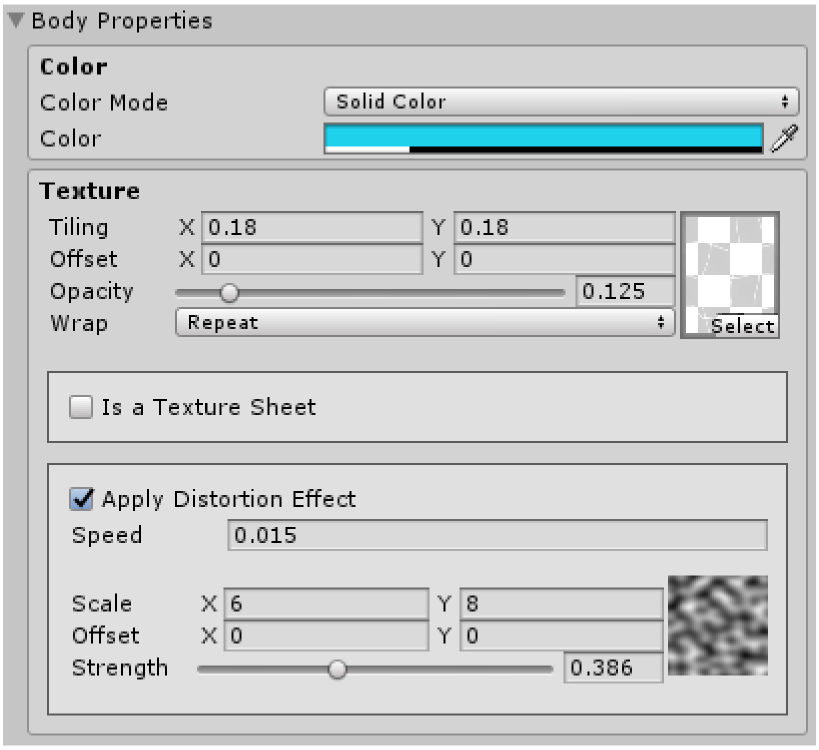


Rendering Mode: Opaque

- Body Properties
- Surface Properties
- Lighting Properties
- Refraction Properties
- Reflecion Properties
- Noise Texture Settings.

- **Rendering Mode:** Sets the rendering mode to either Opaque or Transparent. Please note that the water refraction and the water reflection are not available in the Transparent mode.

★ Water Body Properties:



Body Properties

Color

Color Mode: Solid Color

Color

Texture

Tiling X 0.18 Y 0.18

Offset X 0 Y 0

Opacity 0.125

Wrap: Repeat

Select

☐ Is a Texture Sheet

☑ Apply Distortion Effect

Speed 0.015

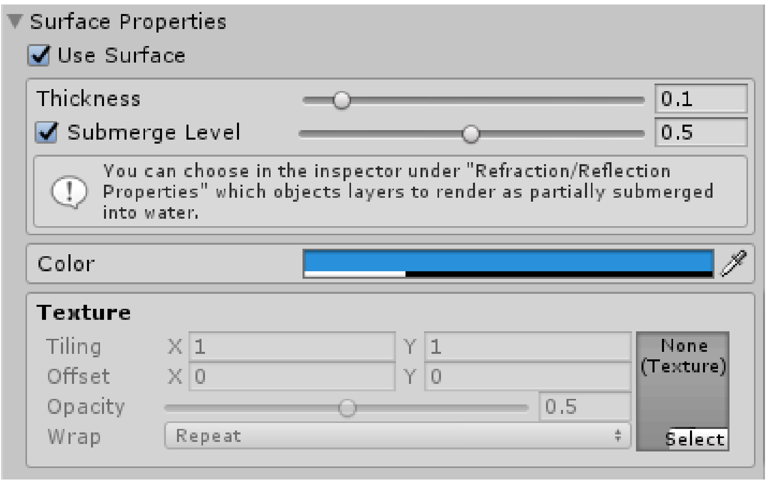
Scale X 6 Y 8

Offset X 0 Y 0

Strength 0.386

- **Color Mode:** Controls whether the water’s body is tinted with a Solid Color or a Gradient Color.
    - **Color:** Sets the water’s color when Color Mode is set to Solid Color.
    - **Start Color:** Sets the first color used to generate the gradient color when Color Mode is set to Gradient Color.
    - **End Color:** Sets the second color used to generate the gradient color when Color Mode is set to Gradient Color.
- **Texture:** Sets the texture applied over the water’s body.
    - **Is a Texture Sheet:** Specifies whether or not the provided texture is a texture sheet (texture consisting of smaller images (animation frames)).
        - **Lerp:** Interpolate between successive texture sheet frames.
        - **Frames Per Second:** The number of frames to play per second.
        - **Columns:** The number of the texture sheet columns.
        - **Rows:** The number of the texture sheet rows.
    - **Apply Distortion Effect:** Apply distortion to the specified texture. For more information about noise texture properties, see page 18.

★ Water Surface Properties:



Surface Properties

☑ Use Surface

Thickness 0.1

☑ Submerge Level 0.5

You can choose in the inspector under "Refraction/Reflection Properties" which objects layers to render as partially submerged into water.

Color

Texture

Tiling X 1 Y 1

Offset X 0 Y 0

Opacity 0.5

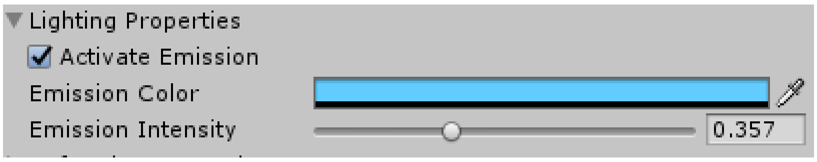
Wrap: Repeat

None (Texture)

Select

- **Use Surface:** Enables/Disables rendering the water's surface line.
- **Thickness:** Sets the water's surface line thickness.
- **Submerge Level:** The toggle box activates/deactivates using the submerge level. The slider sets the submerge level position. Objects specified in 'Partially Submerged Objects' layers are rendered as partially submerged into water when they intersect the submerge level (fake 3D perspective).
- **Color:** Sets the water's surface color.
- **Texture:** Sets the texture applied over the water's surface.
    - **Is a Texture Sheet:** Specifies whether the provided texture is a texture sheet (texture consisting of smaller images (animation frames)) or not.
        - **Lerp:** Interpolate between successive texture sheet frames.
        - **Frames Per Second:** The number of frames to play per second.
        - **Columns:** The number of the texture sheet columns.
        - **Rows:** The number of the texture sheet rows.
    - **Apply Distortion Effect:** Apply distortion to the specified texture. For more information about noise texture properties, see page 18.

★ Lighting Properties:



▼ Lighting Properties

☑ Activate Emission

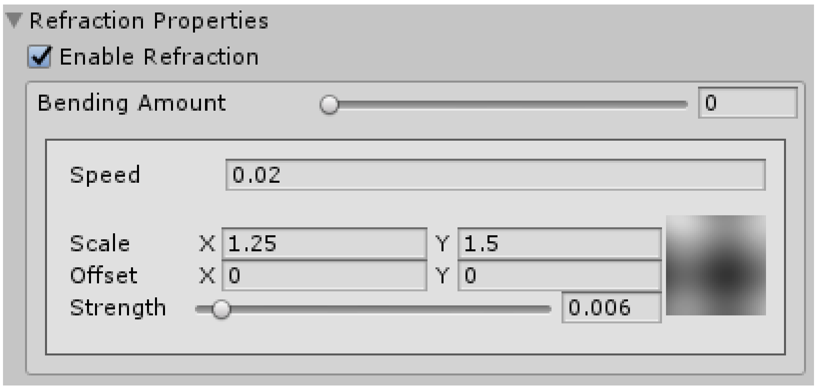
Emission Color

Emission Intensity 0.357

- **Activate Emission:** Activates/Deactivates adjusting the brightness of the water.
- **Emission Color**: Sets the emission color.
- **Emission Intensity:** Sets the brightness level.

Note: *The water object appears as self-illuminated but it does not act as an actual source of light (It does not affect other surrounding objects).*

★ Water Refraction Properties:



▼ Refraction Properties

☑ Enable Refraction

Bending Amount 0

Speed 0.02

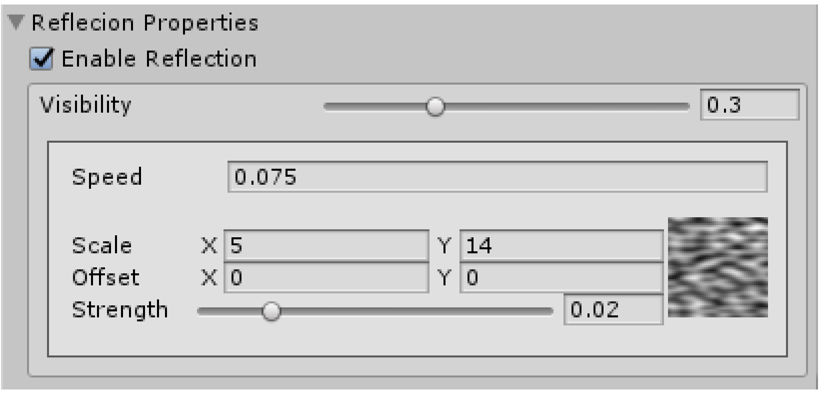
Scale X 1.25 Y 1.5

Offset X 0 Y 0

Strength 0.006

- **Enable Refraction:** Enables/Disables rendering the water refraction.
- **Bending Amount:** Controls how much the portion of the object under the water is shifted relative to the other portion above the water.

★ Water Reflection Properties:



▼ Reflecion Properties

☑ Enable Reflection

Visibility 0.3

Speed 0.075

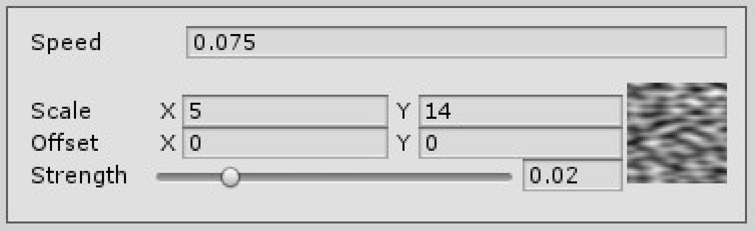
Scale X 5 Y 14

Offset X 0 Y 0

Strength 0.02

- **Enable Reflection:** Enables/Disables rendering the water reflection.
- **Visibility:** Controls the visibility of the reflection when both reflection and refraction are enabled.

★ Noise texture generation properties:



Speed 0.075

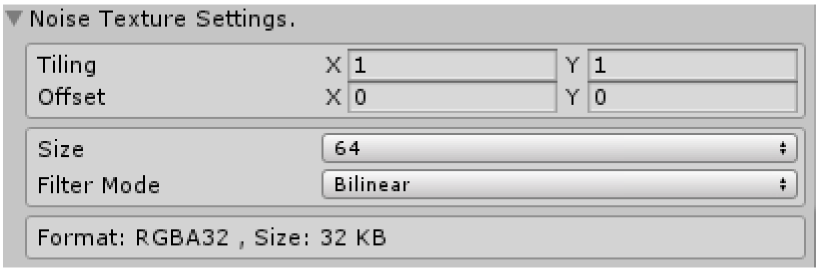
Scale X 5 Y 14

Offset X 0 Y 0

Strength 0.02

- **Speed:** Sets the noise texture scroll speed.
- **Scale:** Sets the noise scale in x and y axes.
- **Offset:** Sets the noise offset in x and y axes.
- **Strength:** Sets the the distortion strength.

★ Noise texture settings:



▼ Noise Texture Settings.

Tiling X 1 Y 1

Offset X 0 Y 0

Size 64

Filter Mode Bilinear

Format: RGBA32 , Size: 32 KB

- **Tiling:** Sets the noise texture tiling.
- **Offset:** Sets the noise texture offset.
- **Size:** Sets the noise texture size.
- **Wrap Mode:** Sets the noise texture's wrap mode to Repeat or Mirror.
- **Filter Mode:** Sets the noise texture's filter mode to Point, Bilinear or Trilinear.

Note: *The Wrap Mode field is only available in Unity version 2017.x or newer. Unity versions prior to 2017.x don't support setting texture wrap mode to Mirror, so the noise texture wrap mode is always set to Repeat in those versions.*