MOBILE WATER SHADER BY ELEKVAULT

INTRODUCTION

Although there are many water shaders available all around that look amazing, I couldn't find any that suited my needs concerning look, control, and, specially, performance on mobile platforms, so I decided to make my own shader to meet my expectations. After many fiddling and tweaking, this is the result and I hope it can satisfy your needs too. It features movement, waves, bumpiness, transparency, cubemap reflection, shadows, all in a single pass, and it also allows for more control from scripts, so you can adjust every parameter dynamically and find the best result for your project.

See the shader in action on https://youtu.be/sg1iwNKYTUM

PACKAGE CONTENTS

- The shader itself: "MobileWaterShader_EKV.shader"
- A material with the same name: "MobileWaterShader_EKV.mat"
- A script with the same name: "MobileWaterShader_EKV.cs"
- Three textures for the shader:
    - Water_Texture_A.jpg
    - Water_Texture_B.jpg
    - Water_HeightMap.png
- A script for additional functions: "WaterControl_EKV.cs"
- A demo scene with the necessary assets.

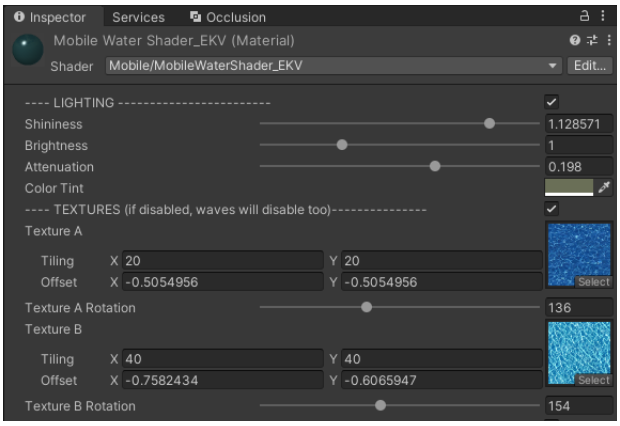
HOW TO USE

- Create your water object.
- Create a material and assign it the shader "Mobile/MobileWaterShader_EKV" (or use the material provided).
- Assign the material to your water object.
- Set the parameters of the shader as needed. Assign your own textures, or use the ones provided.
- (Optional but recommended) Assign the "WaterControl_EKV" script to your water object, and set the parameters as needed.
- (Optional) Assign the "MobileWaterShader_EKV" script to any object on your scene to easily access the shader parameters via script.

SHADER PARAMETERS

This shader contains several values distributed in five sections, which are: Lighting, Textures, Waves, Reflection, and Blending.

When you assign the shader to a material, the Inspector tab will display all the shader parameters as shown in the next screenshot:



Inspector   Services   Occlusion

Mobile Water Shader_EKV (Material)

Shader   Mobile/MobileWaterShader_EKV   Edit...

---- LIGHTING ------------------------ ☑

Shininess   1.128571

Brightness   1

Attenuation   0.198

Color Tint

---- TEXTURES (if disabled, waves will disable too)--------------- ☑

Texture A

Tiling   X 20   Y 20

Offset   X -0.5054956   Y -0.5054956

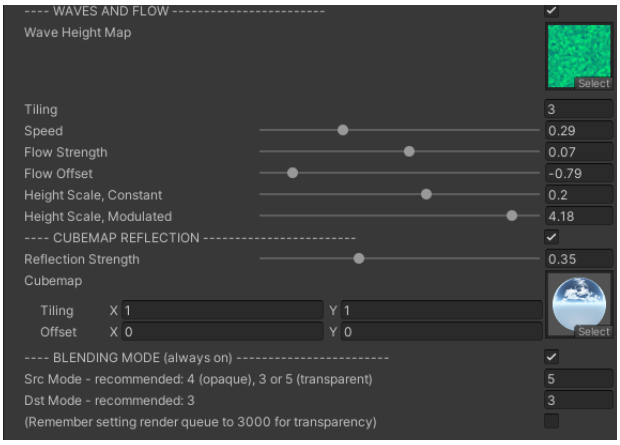
Texture A Rotation   136

Texture B

Tiling   X 40   Y 40

Offset   X -0.7582434   Y -0.6065947

Texture B Rotation   154



---- WAVES AND FLOW ------------------------ ☑

Wave Height Map

Tiling   3

Speed   0.29

Flow Strength   0.07

Flow Offset   -0.79

Height Scale, Constant   0.2

Height Scale, Modulated   4.18

---- CUBEMAP REFLECTION ------------------------ ☑

Reflection Strength   0.35

Cubemap

Tiling   X 1   Y 1

Offset   X 0   Y 0

---- BLENDING MODE (always on) ------------------------ ☑

Src Mode - recommended: 4 (opaque), 3 or 5 (transparent)   5

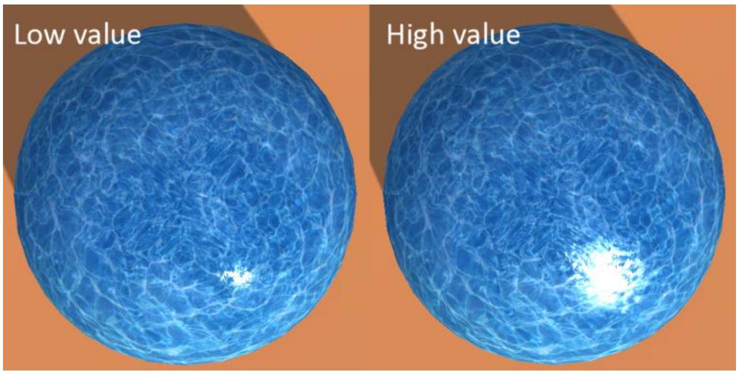
Dst Mode - recommended: 3   3

(Remember setting render queue to 3000 for transparency) □

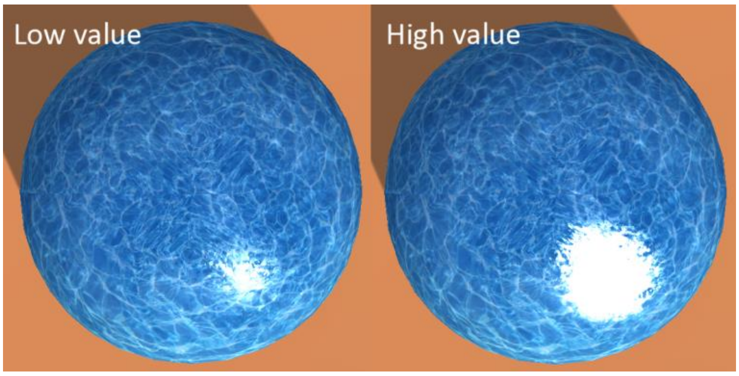
LIGHTING

Un-tick the section checkbox to disable extra light-calculations.

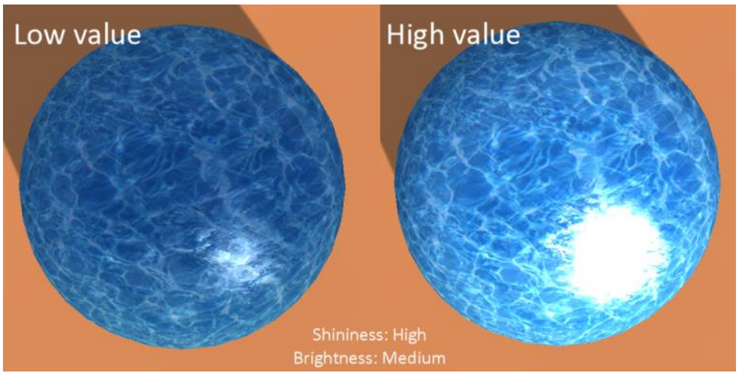
- **Shininess** (_Shininess): amount of specularity of the light on the water. High values increase the area of specularity:



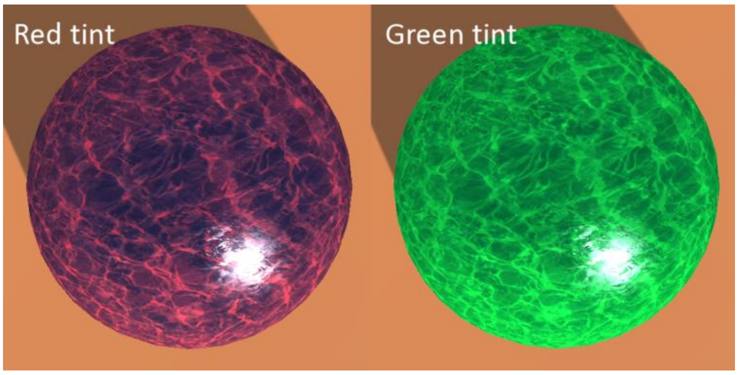
- **Brightness** (_Brightness): intensity of the specularity. High values give more intensity:



- **Attenuation** (_Attenuation): controls the overall amount of light on the object. Low values decrease the light intensity:

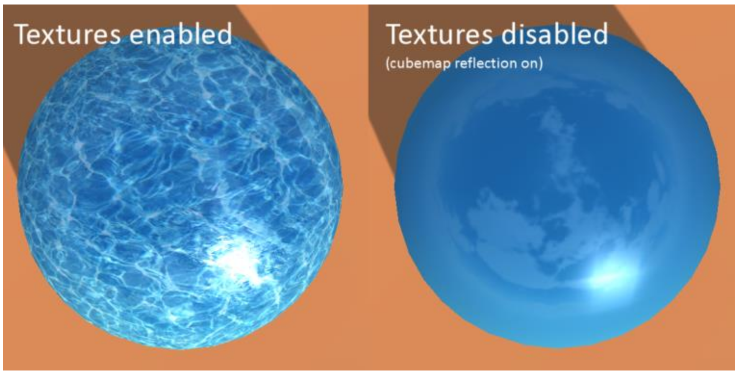


- **Color Tint** (_Color): always active; controls the color tint, additive with the texture coloring; alpha value will affect the specular brightness too:

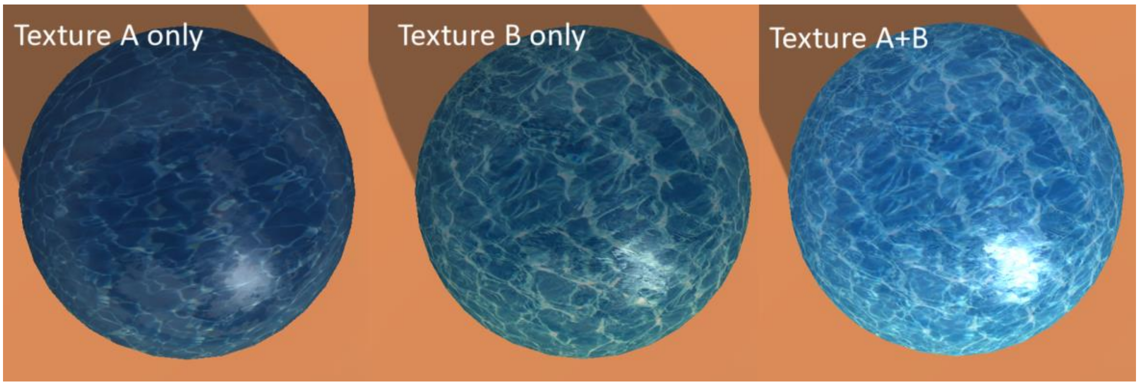


TEXTURES

Un-tick the section checkbox to disable texture drawing and all related calculations; note that even when disabled, if there are textures defined, their color values will be used to calculate the material color; otherwise, white color will be used instead, additive to the tint color:



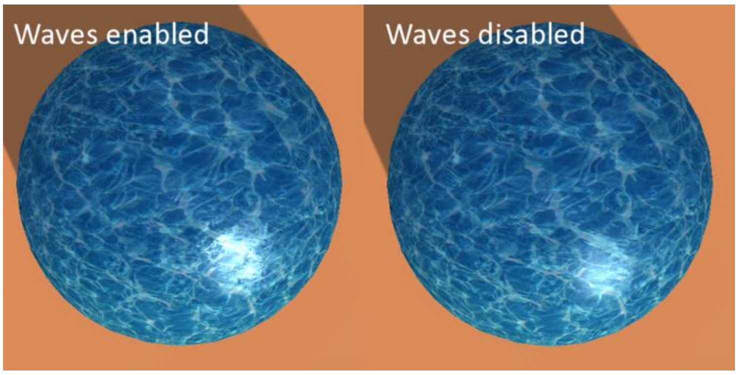
- **Texture A** (_MainTex) and **Texture B** (_DiffTex): the base textures for the shader, use both for a better result; all previous examples uses both textures. In the next comparison, the same parameters are used in all three cases:



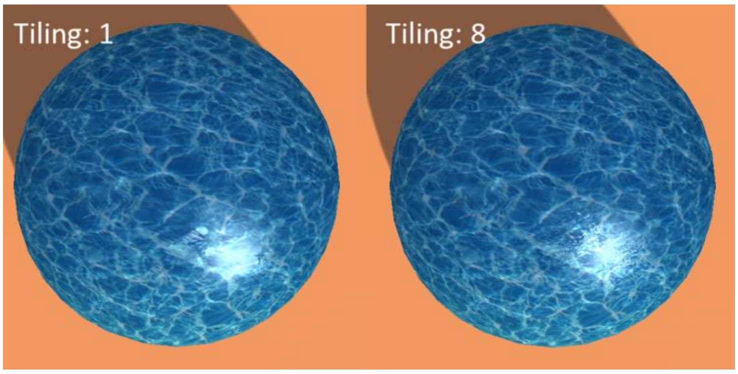
- **Texture A Rotation** (_MainTexRot) and **Texture B Rotation** (_DiffTexRot): control the rotation of both textures over the plane. Useful to offset the rotation of both textures and to properly align the texture with the direction of the flow.

WAVES AND FLOW

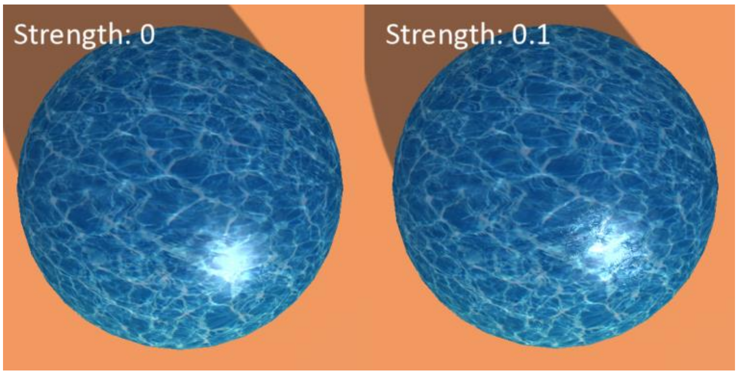
This section controls the wave and flow parameters that give the water movement and bumpy waves effects. Un-tick the section checkbox to disable the effect:



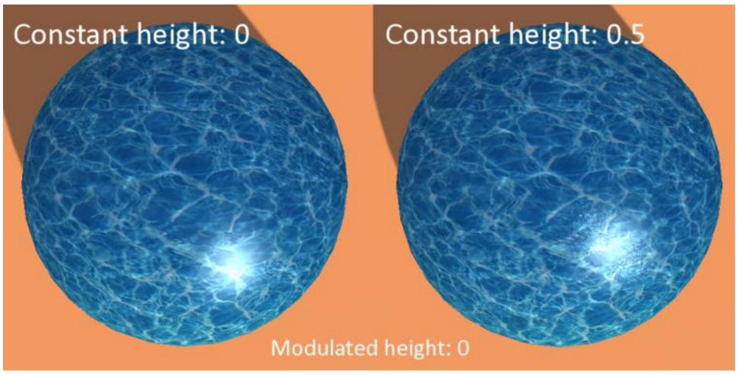
- **Wave Height Map** (_DerivHeightMap): the texture used to calculate the effect on the material. Only the Green, Blue and Alpha channels are used for this calculation.
- **Tiling** (_Tiling): the wave texture tiling. High values will make the effect smaller:



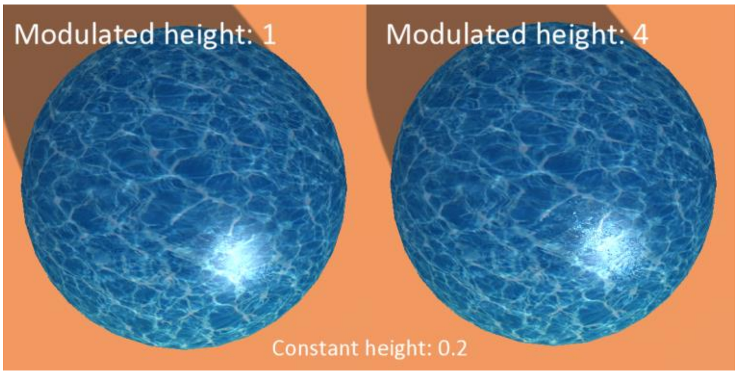
- **Speed** (_Speed): the speed of the flowing animation. Higher is faster.
- **Flow Strength** (_FlowStrength): the strength of the flow animation; it effectively displaces the center of the specularity. Values near 0 are recommended.



- **Flow Offset** (_FlowOffset): slightly displaces the wave texture when calculating the animation.
- **Height Scale, Constant** (_HeightScale): the constant value for the waves height effect; the higher, the stronger the effect:



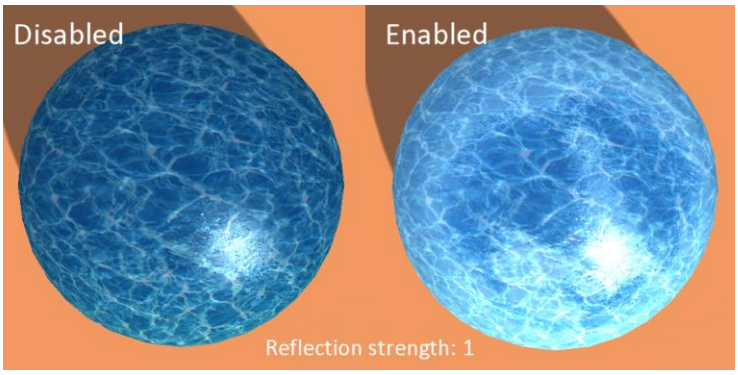
- **Height Scale, Modulated** (_HeightScaleModulated): the variable value for the waves height effect, depending on the value of texture Z over time; the higher, the stronger the effect:



CUBEMAP REFLECTION

This section defines a cubemap texture to reflect on the material surface. Un-tick the section checkbox to disable the reflection.

- **Reflection Strength** (_RefStrength): the brightness of the reflection on the material. The higher, the brighter.
- **Cubemap** (_Cube): the cubemap texture to use.

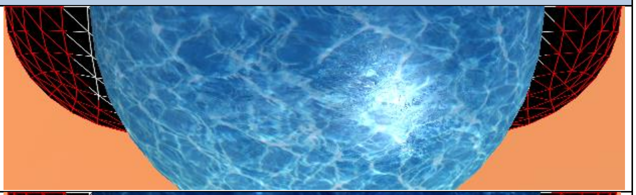
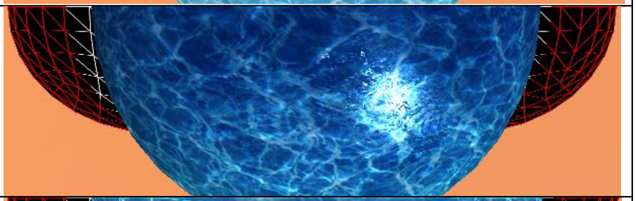
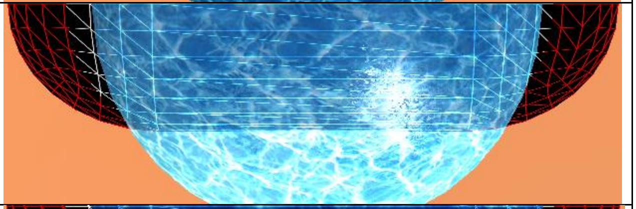
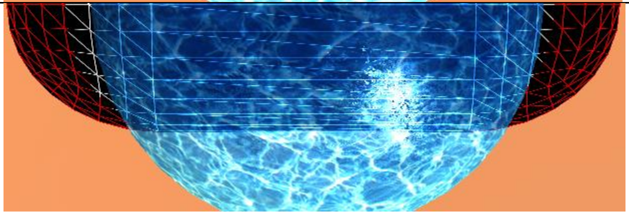
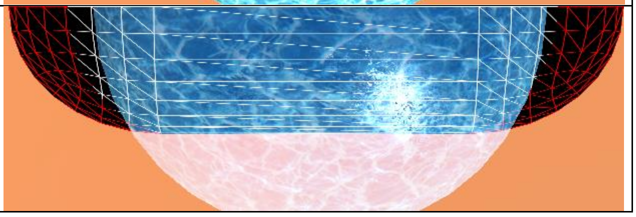


BLENDING MODE

These values set the **Source mode** and **Destination mode** for blending of the material with the rest of the scene, controlling effectively the type of transparency to apply.

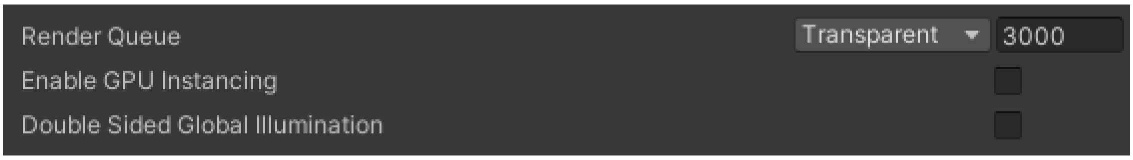
NOTE: By default, this setting is always on and can't be disabled from the Unity Inspector. Since this setting can impact performance greatly, if you're not using transparency it is highly recommended that you disable this function by commenting out the line "Blend[SrcMode][DstMode]" in the "MobileWaterShader_EKV.shader" file, and un-commenting the "Blend Off" line.

Check the following table for the recommended combinations and a preview image of the result:

Src Mode	Dst Mode	Description	Preview
1	0	Opaque (default if "Blend Off")	
4	3		
5	0		
3	0	Opaque, darker	
5	3	Transparent	
3	3	Transparent, darker	
4	1	Transparent, clear (bubble-like)	

For more information on the meaning of these values, check the documentation at https://docs.unity3d.com/Manual/SL-Blend.html

Always remember to change the Render Queue value to the appropriate one if you're using transparency:



Render Queue: Transparent ▾ 3000

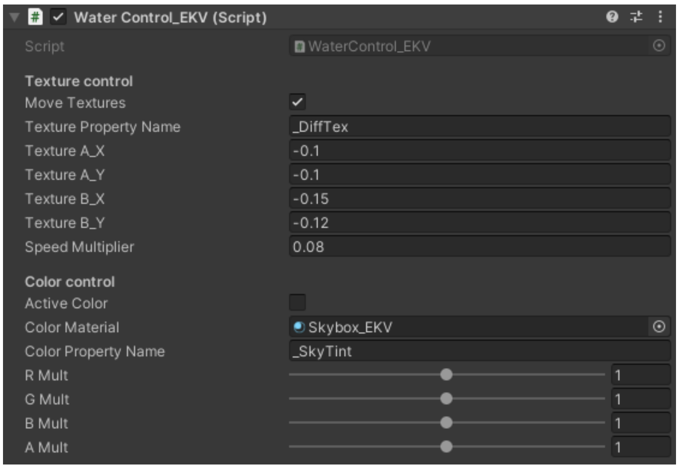
Enable GPU Instancing □

Double Sided Global Illumination □

SCRIPTS

WATERCONTROL_EKV

This script allows for some extra control over the water material properties. To use it, add the script to the water object.



Water Control_EKV (Script)

Script: WaterControl_EKV

Texture control

Move Textures: ☑

Texture Property Name: _DiffTex

Texture A_X: -0.1

Texture A_Y: -0.1

Texture B_X: -0.15

Texture B_Y: -0.12

Speed Multiplier: 0.08

Color control

Active Color: □

Color Material: Skybox_EKV

Color Property Name: _SkyTint

R Mult: 1

G Mult: 1

B Mult: 1

A Mult: 1

"Move Textures" will constantly offset the water textures A and B using the values assigned, creating a stronger flow effect.

"Active Color" will constantly update the tint color of the water material to that one in the referenced material color, applying the multiplier values. This is useful to change the apparent water color along with a changing skybox color, for example.

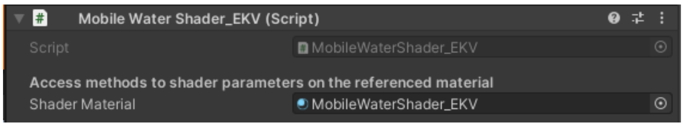
- **PUBLIC PROPERTIES**

NAME	TYPE	DESCRIPTION
Move Textures	bool	Enables the offset movement of water textures A and B
Texture Property Name	string	Property name of the texture B on the shader
Texture A_X and Texture A_Y	float	Amount of offset on texture A in X and Y axis
Texture B_X and Texture B_Y	float	Amount of offset on texture B in X and Y axis

Speed Multiplier	float	Multiplier for all the offset values
Active Color	bool	Enables color control from outside material; this will override the shader color tint, using instead the referenced material color
Color Material	Material	Reference material for color
Color Property Name	string	Property name of the color on the referenced material (so you can specify which color, in case the referenced material contains several colors)
R, G, B, A Mult	float	Multipliers for RGBA values; they apply to the referenced material color.

MOBILEWATERSHADER_EKV

This script gives easy access to all the shader parameters on the material during runtime via scripting, both for reading and setting the values. Add the script to any object on the scene, then assign the Shader Material you want access to.



Script: MobileWaterShader_EKV

Access methods to shader parameters on the referenced material

Shader Material: MobileWaterShader_EKV

- **PUBLIC PROPERTIES**:

NAME	TYPE	DESCRIPTION
Shader Material	Material	The referenced material

- **PUBLIC METHODS:**

DECLARATION	DESCRIPTION
public bool IsLightingEnabled()	Returns the activation state of Lighting check
public void EnableLighting(bool b)	Enables/disables functionality
public float GetShininess()	Returns the parameter value
public void SetShininess(float f)	Sets the parameter value
public float GetBrightness()	Returns the parameter value
public void SetBrightness(float f)	Sets the parameter value
public float GetAttenuation()	Returns the parameter value
public void SetAttenuation(float f)	Sets the parameter value
public Color GetColor()	Returns the tint color
public void SetColor(Color c)	Sets the tint color
public bool IsTexturesEnabled()	Returns the activation state of Textures check
public void EnableTextures(bool b)	Enables/disables functionality
public Texture GetTextureA()	Returns the texture
public void SetTextureA(Texture t)	Sets the texture
public float GetTextureARotation()	Gets the texture current rotation value
public void SetTextureARotation(float f)	Sets the texture rotation value
public Texture GetTextureB()	Returns the texture
public void SetTextureB(Texture t)	Sets the texture
public float GetTextureBRotation()	Gets the texture current rotation value
public void SetTextureBRotation(float f)	Sets the texture rotation value

public bool IsWavesEnabled()	Returns the activation state of Waves check
public void EnableWaves(bool b)	Enables/disables functionality
public Texture GetTextureWaves()	Returns the texture
public void SetTextureWaves(Texture t)	Sets the texture
public float GetWaveTiling()	Returns the parameter value
public void SetWaveTiling(float f)	Sets the parameter value
public float GetSpeed()	Returns the parameter value
public void SetSpeed(float f)	Sets the parameter value
public float GetFlowStrength()	Returns the parameter value
public void SetFlowStrength(float f)	Sets the parameter value
public float GetFlowOffset()	Returns the parameter value
public void SetFlowOffset(float f)	Sets the parameter value
public float GetHeightScale()	Returns the parameter value
public void SetHeightScale(float f)	Sets the parameter value
public float GetHeightScaleModulated()	Returns the parameter value
public void SetHeightScaleModulated(float f)	Sets the parameter value
public bool IsReflectionEnabled()	Returns the activation state of Reflection check
public void EnableReflection(bool b)	Enables/disables functionality
public Texture GetReflectionCubemap()	Returns the texture
public void SetReflectionCubemap(Texture t)	Sets the texture
public float GetReflectionStrength()	Returns the parameter value
public void SetReflectionStrength(float f)	Sets the parameter value
public int GetSourceMode()	Returns the parameter value
public int GetDestinationMode()	Returns the parameter value
public void SetSourceMode(int i)	Sets the parameter value
public void SetDestinationMode(int i)	Sets the parameter value
public void EnableTransparency(bool b)	Sets the blending mode to 5-3 (transparent) if true, or 4-3 (opaque) if false
public bool IsTransparent()	Returns true if blending is 5-3 or 3-3, false otherwise