

Water rendering

Description

Water rendering consist of many different techniques that compose final water visual look. This effect includes but not limited to: water caustic, reflection/refraction, water splashes rendering and water surface ripples.

Associated classes and structures

class WaterBase

Source: Water.h, Water.cpp

Shaders: ripples_ps.hls, ripples_vs.hls, OCeanV2high_ps.hls, OceanV2high_vs.hls, OCeanV2low_ps.hls, OceanV2low_vs.hls, OCeanV2normal_ps.hls, OceanV2normal_vs.hls, OCeanV2uH_ps.hls, OceanV2uH_vs.hls, OCeanV2uL_ps.hls, OceanV2uL_vs.hls, OCeanV2uM_ps.hls, OceanV2uM_vs.hls, rainRipples_ps.hls

This class implements water rendering, and used by several object classes (see below). Class contains many water quality setup variables. Water rendering can be performed in two modes:

1. Water surface rendering (default mode).
2. Underwater rendering (used when camera is below water surface).

Important members:

r3dTexture* WaterColor[25]	Water color textures array. This array represents seamless and contiguous wave animation sequence. Each texture has wave shape normal map in rgb channels and underwater caustic animation in alpha channel.
r3dTexture* ColorTexture	Water color tint texture. This texture is tiled in all directions.
r3dColor shallowColor	Shallow water color value.
r3dColor attenColor	Light color for water shading.
float shallowDepth	Coastline depth value.
r3dColor deepColor	Deep water color value.
float attenDist	Water color attenuation distance.
float refractionIndex	Water refractivity index.
float bumpness	Water bumpiness.
float tileSize	Water tile size.
float specularCosinePower	Amount of specular contribution that water have.
float specularTiling	Specular tiling.
float causticStrength	Strength of caustic effect.
float causticDepth	Depth value where caustic effect should become visible.
float causticTiling	Tiling factor of caustic effect.

float waterColorTile	Water color tiling factor.
float waterColorBlend	Water color blending factor.
float refractionPerturbation	Refraction effect perturbation amount.
float uRefractionIndex	Underwater mode refraction index.
float uRefractionPerturbation	Underwater mode refraction effect perturbation coefficient.
float uAttenDist	Underwater mode light color attenuation factor.
r3dColor uShallowColor	Shallow color for underwater mode.
r3dColor uDeepColor	Deep color for underwater mode.
r3dColor uAttenColor	Underwater water color.
r3dScreenBuffer* ripplesRT[2]	Two render targets that used in ping-pong style water ripples accumulation.

Important methods:

void ReloadTextures()

Reload all water textures from disk.

void ReadSerializedData(**pugi::xml_node**& node)

Load WaterBase class settings from given xml node.

void WriteSerializedData(**pugi::xml_node**& node)

Write class settings into given xml node.

float SetNormalTextures(**unsigned int** t0, **unsigned int** t1) **const**

Set current water animation textures into samplers defined by t0 and t1 input indices.

void CreateRefractionBuffer()

Create render target texture used for refraction image rendering.

void RenderBegin(**const** **r3dCamera**& Cam, **float** waterLevel, **bool** followTerrain)

Begin rendering water. This function will set all needed vertex and index constants and shaders.

Parameters:

Cam – camera to render with.

waterLevel – water height value.

followTerrain – make water surface shape repeat covered terrain shape.

void RenderEnd()

End water rendering.

float DrawPropertyEditorWater(**float** scrx, **float** scry, **float** scrw, **float** scrh, **float** DEFAULT_CONTROLS_WIDTH, **float** DEFAULT_CONTROLS_HEIGHT)

Show all water setup GUI controls. This function cannot be used outside editor.

Parameters:

scrx – Current x drawing position.

scry – Current y drawing position.

scrw – screen width in pixels.

scrh – screen height in pixels.

DEFAULT_CONTROLS_WIDTH – default GUI controls width.

DEFAULT_CONTROLS_HEIGHT – default GUI controls height.

void OnCreateBase();

Create all required internal data.

void OnDestroyBase();

Destroy internal data.

void DeleteRippleBuffers();

Destroy two ripple accumulation buffers.

class obj_WaterPlane

Derived from WaterBase, this class represents water object entity in game world. Water surface defined by obj_WaterPlane consist of grid cells, each of them contains property flag – is it contains water or not. Using this technique and with sufficient grid resolution is possible to make complex water surface shapes.

Important methods:

virtual BOOL OnCreate()

Initialize water plane object internal structures. This includes WaterBase class initialization.

virtual BOOL OnDestroy()

Destroy internal structures of object.

virtual BOOL Update()

Update water plane bounding box and transformation matrix.

virtual void ReadSerializedData(pugi::xml_node& node)

Load water plane properties from given xml node.

virtual void WriteSerializedData(pugi::xml_node& node)

Write water plane object properties into given xml node.

void DoDraw(const r3dCamera& Cam)

Draw water grid using given camera.

virtual void AppendRenderables(RenderArray (& render_arrays) [rsCount], const r3dCamera& Cam)

Append water renderable objects into given input arrays. These arrays will be rendered later as part of delayed rendering process. See renderer documentation for more details.

Parameters:

render_arrays – render arrays to insert renderable entities into.

Cam – camera to render with.

virtual float DrawPropertyEditor(float scrx, float scry, float scrw, float scrh, const AClass* startClass, const GameObjects& selected)

Display GUI water configuration controls. This method available only in editor mode.

void SetCellGridSize (float fSize)

Set water cell grid size in world units.

float GetCellGridSize () const

Get cell grid size.

void SetWaterPlaneHeightOnTerrain (float fH)

Set water surface height (in world space units) above terrain, if “follow terrain” mode is used.

float GetWaterPlaneHeightOnTerrain () const

Get water surface height above terrain.

void SetWaterPlaneHeight (float fH)

Set absolute water plane height.

float GetWaterPlaneHeight () const

Get absolute water plane height.

void SetWaterPlaneCoastSmoothLevels (int iLvls)

Set how many edge cells will represent water shape slope on coastal areas. This value is used when “follow terrain” mode is used.

int GetWaterPlaneCoastSmoothLevels () const

Get number of water coastal edge slope cells.

void SetWaterPlaneFollowTerrainFlag(bool bFollow)

Set or unset “follow terrain” mode. This mode is used to closely follow terrain shape, in areas which water cover. Height above terrain is set by **SetWaterPlaneHeightOnTerrain** function.

bool GetWaterPlaneFollowTerrainFlag() const

Get “follow terrain” flag.

void SetGridDimensions (uint32_t iW, uint32_t iH)

Set water grid resolution.

void GetGridDimensions (uint32_t& iW, uint32_t& iH)

Get water grid resolution.

bool FillWaterGrid (uint32_t iW, uint32_t iH, bool bUpdateImmediately = true)

Set “is water” flag to the specific grid cell.

Parameters:

iW – x cell coordinate.

iH – y cell coordinate.

bUpdateImmediately – update water mesh now if true.

bool EraseWaterGrid (uint32_t iW, uint32_t iH, bool bUpdateImmediately = true)

Clear “is water” flag from the specific grid cell.

Parameters:

iW – x cell coordinate.

iH – y cell coordinate.

bUpdateImmediately – update water mesh now if true.

uint8_t GetWaterGridValue(uint32_t iW, uint32_t iH)

Get “is water” flag from specific cell. Return value will be nonzero if requested cell is water cell.

Parameters:

iW – x cell coordinate.

iH – y cell coordinate.

void UpdateWaterPlane ()

Update internal water mesh representation, which used for rendering. Water mesh constructed using cells grid values defined using **FillWaterGrid** and **EraseWaterGrid** functions.

void LoadWaterPlaneGrid(const char*szName)

Load water grid from file, previously stored by **SaveWaterPlaneGrid** function.

Parameters:

szName – file name to load grid from.

void SaveWaterPlaneGrid(const char*szName)

Save water grid to file.

Parameters:

szName – file name to save grid to.

bool WaterSplash(const r3dPoint3D& waterSplashPos, float height, float size, float amount, int texIdx)

Create splash on water surface. If splash with specific position and sizes will have influence on water surface, function returns true.

Parameters:

waterSplashPos – position of water splash in world coordinates.

height – splash virtual vertical height. Used as strength adjustment factor.

size – horizontal radius of splash effect.

amount – strength of splash effect.

texIdx – texture index.

void ComputeCoastSmoothInfo (r3dTL::T2DArray<uint8_t> & tInfo)

Compute coast slope using water grid cell information. Result of calculation is written into tInfo array.

void ComputeWaterVertexesPos (r3dTL::T2DArray<r3dVector> & tInfo)

Compute water vertex positions. Result is written into tInfo array.