Crest - Splines API

Release 1.4.2

Wave Harmonic

Sep 04, 2025

TABLE OF CONTENTS

CHAPTER

ONE

CREST.SPLINES

1.1 AbsorptionSplineLodInputData

Class in *WaveHarmonic.Crest.Splines*, WaveHarmonic.Crest.Splines
Extends *SplineLodInputData*

Data storage for for the Spline input mode.

1.1.1 Properties

AbsorptionColor

The color of water due to absorption.

Declaration

```
public Color AbsorptionColor { get; set; }
```

1.1.2 Inherited Properties

OverrideSplineSettings

Whether to override the settings with the same name on the spline component.

Declaration

```
public bool OverrideSplineSettings { get; set; }
```

Radius

The radius of the spline.

Declaration

```
public float Radius { get; set; }
```

Spline

The *Crest Spline* to use with this input.

Declaration

```
public Spline Spline { get; set; }
```

Subdivisions

Increasing subdivision increases the geometry density.

Mostly useful for water level changes. High values can reduce staircasing effect.

Declaration

```
public int Subdivisions { get; set; }
```

Mesh

The mesh generated from the spline.

The mesh should be available by Start.

Declaration

```
public Mesh Mesh { get; }
```

1.2 AbsorptionSplinePointData

Class in *WaveHarmonic.Crest.Splines*, WaveHarmonic.Crest.Splines
Extends *SplinePointData*

Custom spline point data for scattering.

1.2.1 Properties

AbsorptionColor

The scattering color.

Declaration

```
public Color AbsorptionColor { get; set; }
```

OverrideAbsorption

Whether to override the scattering color instead of just the weight.

Declaration

```
public bool OverrideAbsorption { get; set; }
```

Weight

The weight of the scattering color.

Declaration

```
public float Weight { get; set; }
```

1.3 FlowSplineLodInputData

Class in *WaveHarmonic.Crest.Splines*, WaveHarmonic.Crest.Splines
Extends *SplineLodInputData*

Data storage for for the Spline input mode.

1.3.1 Properties

FlowVelocity

Flow velocity (speed of flow in direction of spline). Can be negative to flip direction.

Declaration

```
public float FlowVelocity { get; set; }
```

1.3.2 Inherited Properties

OverrideSplineSettings

Whether to override the settings with the same name on the spline component.

Declaration

```
public bool OverrideSplineSettings { get; set; }
```

Radius

The radius of the spline.

Declaration

```
public float Radius { get; set; }
```

Spline

The *Crest Spline* to use with this input.

Declaration

```
public Spline Spline { get; set; }
```

Subdivisions

Increasing subdivision increases the geometry density.

Mostly useful for water level changes. High values can reduce staircasing effect.

Declaration

```
public int Subdivisions { get; set; }
```

Mesh

The mesh generated from the spline.

The mesh should be available by Start.

Declaration

```
public Mesh Mesh { get; }
```

1.4 FlowSplinePointData

Class in *WaveHarmonic.Crest.Splines*, WaveHarmonic.Crest.Splines
Extends *SplinePointData*

Custom spline point data for flow

1.4.1 Properties

FlowVelocity

Flow velocity (speed of flow in direction of spline).

Can be negative to flip direction.

Declaration

```
public float FlowVelocity { get; set; }
```

1.5 FoamSplineLodInputData

Class in *WaveHarmonic.Crest.Splines*, WaveHarmonic.Crest.Splines
Extends *SplineLodInputData*

Data storage for for the Spline input mode.

1.5.1 Properties

FoamAmount

Amount of foam emitted.

Declaration

```
public float FoamAmount { get; set; }
```

1.5.2 Inherited Properties

OverrideSplineSettings

Whether to override the settings with the same name on the spline component.

Declaration

```
public bool OverrideSplineSettings { get; set; }
```

Radius

The radius of the spline.

Declaration

```
public float Radius { get; set; }
```

Spline

The *Crest Spline* to use with this input.

Declaration

```
public Spline Spline { get; set; }
```

Subdivisions

Increasing subdivision increases the geometry density.

Mostly useful for water level changes. High values can reduce staircasing effect.

Declaration

```
public int Subdivisions { get; set; }
```

Mesh

The mesh generated from the spline.

The mesh should be available by Start.

Declaration

```
public Mesh Mesh { get; }
```

1.6 FoamSplinePointData

Class in *WaveHarmonic.Crest.Splines*, WaveHarmonic.Crest.Splines
Extends *SplinePointData*

Foam tweakable param on spline points

1.6.1 Properties

FoamAmount

Amount of foam emitted.

Declaration

```
public float FoamAmount { get; set; }
```

1.7 LevelSplineLodInputData

Class in *WaveHarmonic.Crest.Splines*, WaveHarmonic.Crest.Splines
Extends *SplineLodInputData*

Data storage for for the Spline input mode.

1.7.1 Inherited Properties

OverrideSplineSettings

Whether to override the settings with the same name on the spline component.

Declaration

```
public bool OverrideSplineSettings { get; set; }
```

Radius

The radius of the spline.

Declaration

```
public float Radius { get; set; }
```

Spline

The *Crest Spline* to use with this input.

Declaration

```
public Spline Spline { get; set; }
```

Subdivisions

Increasing subdivision increases the geometry density.

Mostly useful for water level changes. High values can reduce staircasing effect.

Declaration

```
public int Subdivisions { get; set; }
```

Mesh

The mesh generated from the spline.

The mesh should be available by Start.

Declaration

```
public Mesh Mesh { get; }
```

1.8 ScatteringSplineLodInputData

Class in *WaveHarmonic.Crest.Splines*, WaveHarmonic.Crest.Splines
Extends *SplineLodInputData*

Data storage for for the Spline input mode.

1.8.1 Properties

ScatteringColor

The color of the scattering.

Declaration

```
public Color ScatteringColor { get; set; }
```

1.8.2 Inherited Properties

OverrideSplineSettings

Whether to override the settings with the same name on the spline component.

Declaration

```
public bool OverrideSplineSettings { get; set; }
```

Radius

The radius of the spline.

Declaration

```
public float Radius { get; set; }
```

Spline

The *Crest Spline* to use with this input.

Declaration

```
public Spline Spline { get; set; }
```

Subdivisions

Increasing subdivision increases the geometry density.

Mostly useful for water level changes. High values can reduce staircasing effect.

Declaration

```
public int Subdivisions { get; set; }
```

Mesh

The mesh generated from the spline.

The mesh should be available by Start.

Declaration

```
public Mesh Mesh { get; }
```

1.9 ScatteringSplinePointData

Class in *WaveHarmonic.Crest.Splines*, WaveHarmonic.Crest.Splines
Extends *SplinePointData*

Custom spline point data for scattering.

1.9.1 Properties

OverrideScattering

Whether to override the scattering color instead of just the weight.

Declaration

```
public bool OverrideScattering { get; set; }
```

Scattering

The scattering color.

Declaration

```
public Color Scattering { get; set; }
```

Weight

The weight of the scattering color.

Declaration

```
public float Weight { get; set; }
```

1.10 ShapeWavesSplineLodInputData

Class in *WaveHarmonic.Crest.Splines*, WaveHarmonic.Crest.Splines
Extends *SplineLodInputData*

Data storage for for the Spline input mode.

1.10.1 Properties

FeatherWaveStart

Feathers waves across the spline (ie across width). Reverse the spline to swap direction.

Declaration

```
public float FeatherWaveStart { get; set; }
```

Weight

Weight multiplier to scale waves.

Declaration

```
public float Weight { get; set; }
```

1.10.2 Inherited Properties

OverrideSplineSettings

Whether to override the settings with the same name on the spline component.

Declaration

```
public bool OverrideSplineSettings { get; set; }
```

Radius

The radius of the spline.

Declaration

```
public float Radius { get; set; }
```

Spline

The *Crest Spline* to use with this input.

Declaration

```
public Spline Spline { get; set; }
```

Subdivisions

Increasing subdivision increases the geometry density.

Mostly useful for water level changes. High values can reduce staircasing effect.

Declaration

```
public int Subdivisions { get; set; }
```

Mesh

The mesh generated from the spline.

The mesh should be available by Start.

Declaration

```
public Mesh Mesh { get; }
```

1.11 Spline

Class in *WaveHarmonic.Crest.Splines*, WaveHarmonic.Crest.Splines

Simple spline object. Spline points are child GameObjects.

1.11.1 Properties

Closed

Connect start and end point to close spline into a loop.

Requires at least 3 spline points.

Declaration

```
public bool Closed { get; set; }
```

Offset

Where generated ribbon should lie relative to spline.

If set to Center, ribbon is centered around spline.

Declaration

```
public SplineOffset Offset { get; set; }
```

Radius

The radius of the spline.

Declaration

```
public float Radius { get; set; }
```

Subdivisions

Increasing subdivision increases the geometry density.

Mostly useful for water level changes. High values can reduce staircasing effect.

Declaration

```
public int Subdivisions { get; set; }
```

1.11.2 Methods

UpdateSpline

UpdateSpline()

Applies any changes to the spline meshes.

Declaration

```
public void UpdateSpline()
```

1.12 SplineLodInputData

Class in *WaveHarmonic.Crest.Splines*, WaveHarmonic.Crest.Splines
Extends /API/WaveHarmonic.Crest/LodInputData

Data storage for for the Spline input mode.

1.12.1 Properties

OverrideSplineSettings

Whether to override the settings with the same name on the spline component.

Declaration

```
public bool OverrideSplineSettings { get; set; }
```

Radius

The radius of the spline.

Declaration

```
public float Radius { get; set; }
```

Spline

The *Crest Spline* to use with this input.

Declaration

```
public Spline Spline { get; set; }
```

Subdivisions

Increasing subdivision increases the geometry density.

Mostly useful for water level changes. High values can reduce staircasing effect.

Declaration

```
public int Subdivisions { get; set; }
```

Mesh

The mesh generated from the spline.

The mesh should be available by Start.

Declaration

```
public Mesh Mesh { get; }
```

1.13 SplineOffset

Enum in *WaveHarmonic.Crest.Splines*, WaveHarmonic.Crest.Splines

Where generated ribbon should lie relative to the *Spline*.

1.13.1 Properties

Left

Left to the spline.

Declaration

```
Left = 0
```

Center

Centered around the spline.

Declaration

```
Center = 1
```

Right

Right to the spline.

Declaration

```
Right = 2
```

1.14 SplinePoint

Class in *WaveHarmonic.Crest.Splines*, WaveHarmonic.Crest.Splines

Spline point, intended to be child of Spline object

1.14.1 Properties

RadiusMultiplier

Multiplier for spline radius.

Declaration

```
public float RadiusMultiplier { get; set; }
```

1.15 SplinePointData

Class in *WaveHarmonic.Crest.Splines*, WaveHarmonic.Crest.Splines

Base class for components which hold point-level spline data.

1.16 WavesSplinePointData

Class in *WaveHarmonic.Crest.Splines*, WaveHarmonic.Crest.Splines
Extends *SplinePointData*

Custom spline point data for waves

1.16.1 Properties

Weight

Weight multiplier to scale waves.

Declaration

```
public float Weight { get; set; }
```