UI Spline Renderer

using Unity Spline Package

by olivecrow

1. Overview

This is an asset for rendering splines on UI using Unity's spline package.

It has a simple MonoBehaviour component, UISplineRenderer

UISplineRenderer is inherited from UnityEngine.UI.MaskableGraphic class so the rendered spline is selectable as other UI graphic components like Button or Toggle. And it also is maskable.

You can make various styles by using various properties like width curve, color gradient, texture, start and end images.

1-1. Dependency

It uses next unity packages and all of these packages are automatically installed when this asset is imported.

- Splines (com.unity.splines)

- Mathematics (com.unity.mathematics)

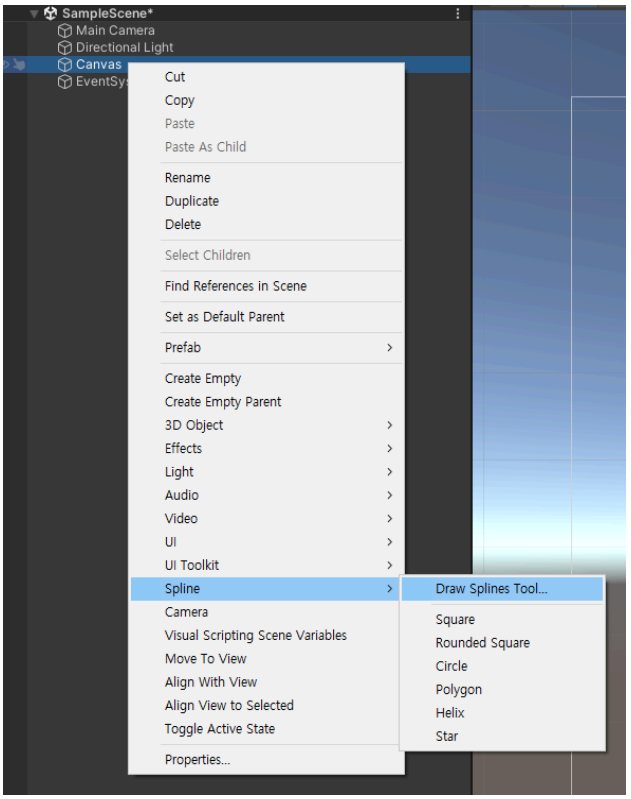
- Collections (com.unity.collections)

- Burst (com.unity.burst)

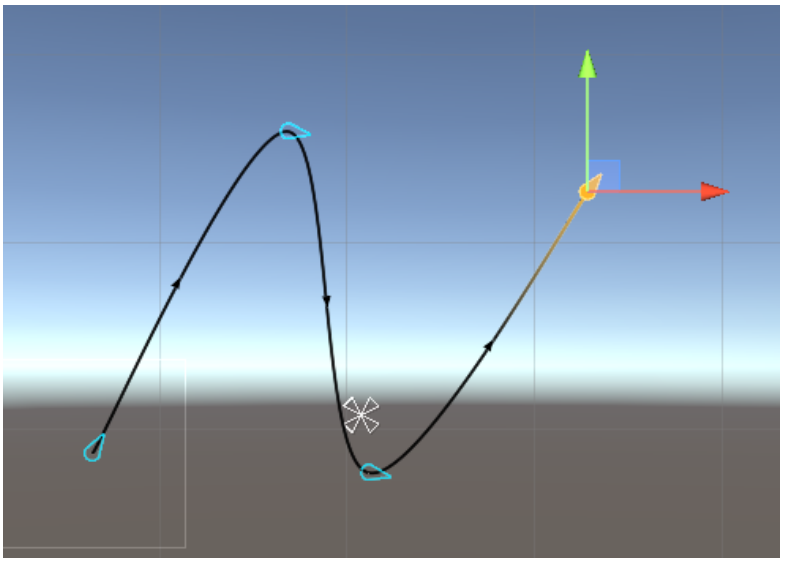
2. How to use

2-1. Edit Mode

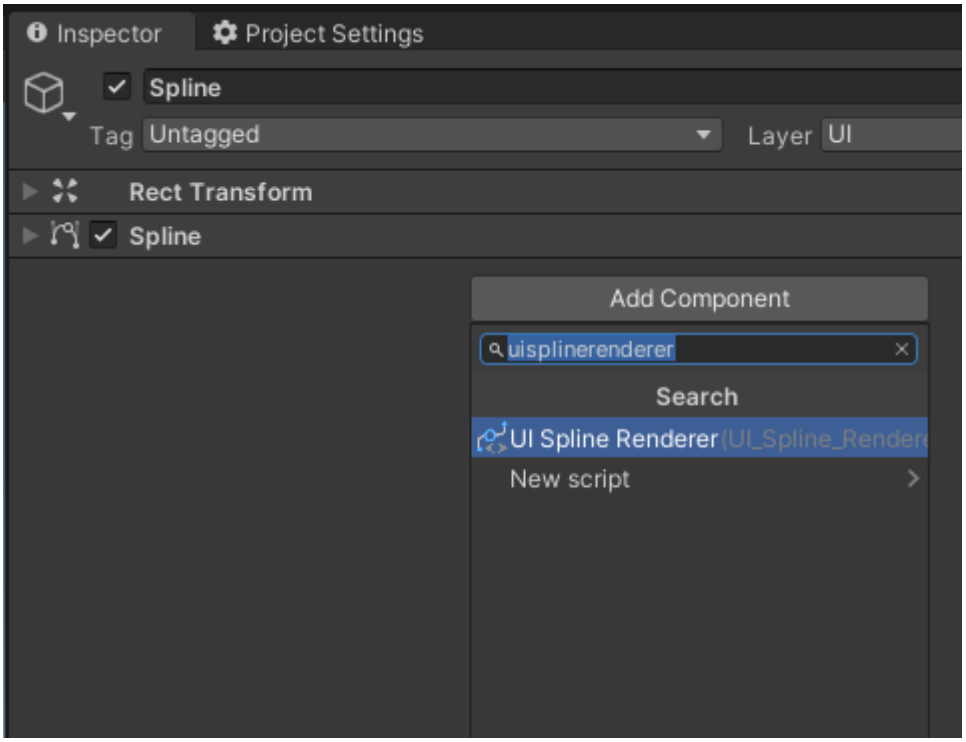
1. Create Spline under a UI object like canvas.



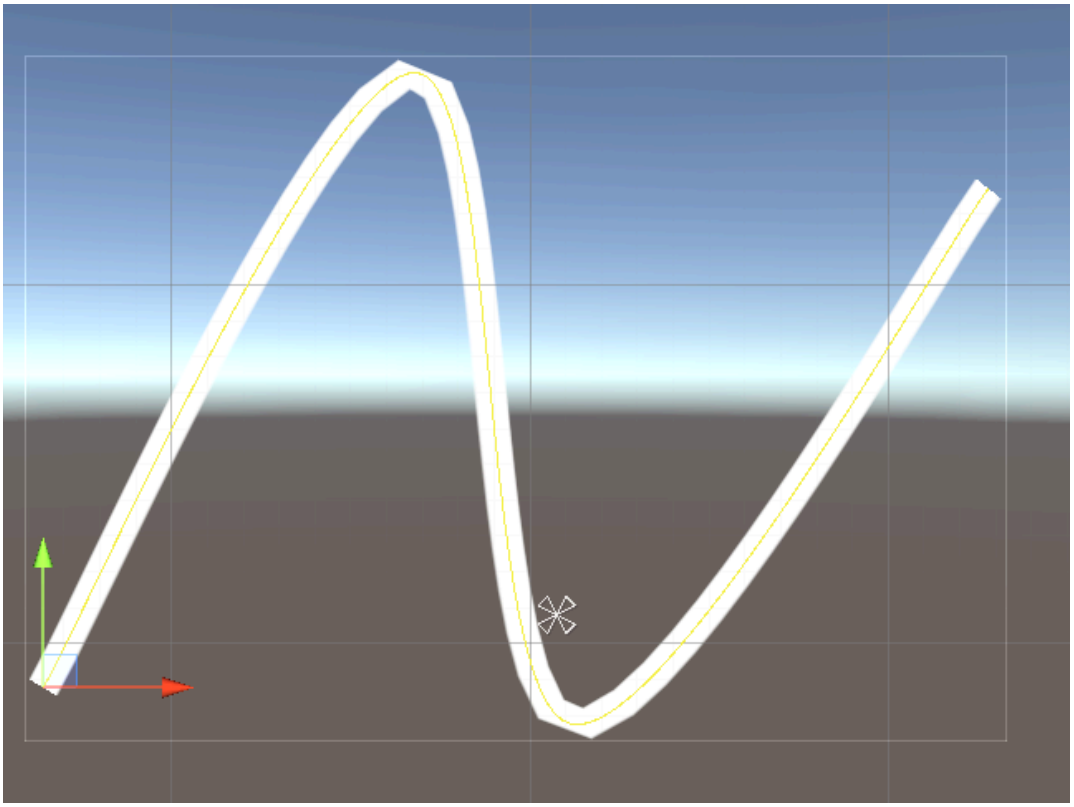
2. Edit the spline as you like.



3. Add a UISplineRenderer component on the same gameObject of the spline.



4. Now the spline is rendered as UI.



2-2. Runtime

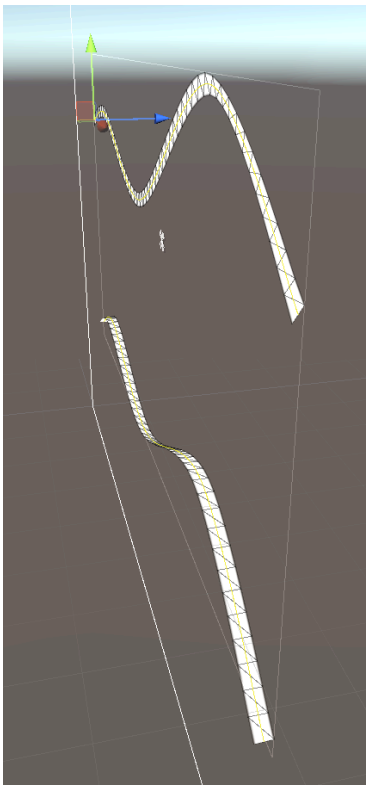
You can simply create spline renderers in runtime using UISplineRenderer.Create.

It automatically rotates each knot to be perpendicular to the XY plane. So you don't need to worry about which rotation to set.

```
public static UISplineRenderer Create(
    IEnumerable<Vector3> positions,
    RectTransform parent,
    bool isLocal = false,
    LineTexturePreset lineTexture = LineTexturePreset.Default,
    StartEndImagePreset startImage = StartEndImagePreset.None,
    StartEndImagePreset endImage = StartEndImagePreset.None)
```

```
public static UISplineRenderer Create(
    IEnumerable<Vector3> positions,
    RectTransform parent,
    bool isLocal,
    Texture lineTexture,
    Sprite startImage,
    Sprite endImage)
```

Notice!



Rotations of knots in splines should look in the UI camera direction (0, 0, -1). When you edit a spline in the scene view (2D view) it is automatically oriented on the XY plane.

But when you make spline in runtime via script, if you don't set rotation value, the normal is

(0, 1, 0). Then, the spline is not rendered properly because all the quads of the spline look upward.

The upper spline is a well-oriented spline for UI rendering, and the lower spline is a bad spline that does not render properly. Check the normal of each knot in the spline.

If you want to use other tangent mods, you can call ReorientKnots(). This is an extension method for SplineContainer included in this package, and you can also call it from the UISplineRenderer instance. Detailed description about these extension methods are below.

3. UISplineRenderer Component

3-1. Inspector

Most properties have tooltips. You can see it by hovering your mouse in the inspector.

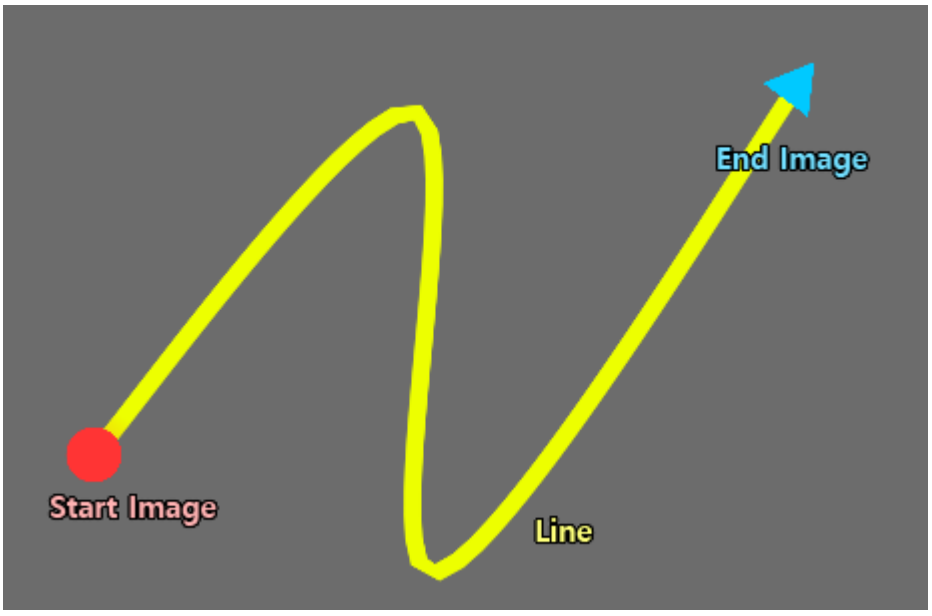
Spline Container	Splines to render. This field is automatically set OnEnable(). The component should belong to the same gameObject with UISplineRenderer.
Material	The material to this renderer. You can use your custom material. Default is null.
Recursive Material	If true, this renderer's maskable value will also be applied to startImages and endImages.
Color	Color of the renderer.
Color Gradient	Color tint multiplied to the color. 0 is the first point and 1 is the last point.
Width	Width of the renderer.
Width Curve	Multiplier for the width. 0 is the first point and 1 is the last point.
Keep Zero Z	Keep all z positions and transform.localPosition.z as 0. This keeps the spline flatten on canvas.
Keep Billboard	Keep all normals of the vertices pointing towards the screen direction. This ensures that the spline is rendered as a billboard, even if you set odd rotations to knots.
Line Texture Preset	Presets for texture property.
Resolution	The value that determines the number of quads to use to render spline. Notice that the vertex count should be lower than 65,000. This is Unity's vertex limit to render UI.
Clip Range	Range to render splines.
UV Mode	Line texture’s UV mode. Texture is applied in the Y-direction.
UV Offset	Line texture’s UV offset. You can make simple texture animations using this property.
Tile	Texture is tiled along spline length. The wrap mode of the texture should be Repeat.
Repeat Per Segment	Texture is placed per segment(spline knot). The wrap mode of the texture should be Repeat.
Stretch	Texture is stretched along splines.
UV Multiplier	Multiplier to UV size.
Start Image Preset	Presets for startImageSprite property. The first knot is located at the center of the sprite.
Start Image Size	Size of the start image.

Start Image Offset Mode	How to calculate the offset of start images. In the case of Distance, the absolute value of the length is used as the offset. When Normalized, an offset expressed as 0..1 is used.
Start Image Offset	Offset from the first point of spline. If the value is greater than 0, it moves along the spline. Otherwise, it moves backward relative to current rotation.
Normalized Start Image Offset	Normalized offset value. 0 means the beginning of the spline, and 1 means the end of the spline.
End Image Preset	Presets for endImageSprite property. The last knot is located at the center of the sprite.
End Image Size	Size of the end image.
End Image Offset Mode	How to calculate the offset of end images. In the case of Distance, the absolute value of the length is used as the offset. When Normalized, an offset expressed as 0..1 is used.
End Image Offset	Offset from the last point of the spline. If the value is smaller than 0, it moves along the spline. Otherwise, it moves forward relative to current rotation.
Normalized End Image Offset	Normalized offset value. 0 means the beginning of the spline, and 1 means the end of the spline.

3-2. Features

A UISplineRenderer consists of 3 parts.

Start Image is located at first of the spline. This can be null. This is just another UnityEngine.UI.Image gameObject. Line is the main renderer. End Image is the same with Start Image. Start Image and End Image can move using their offset.

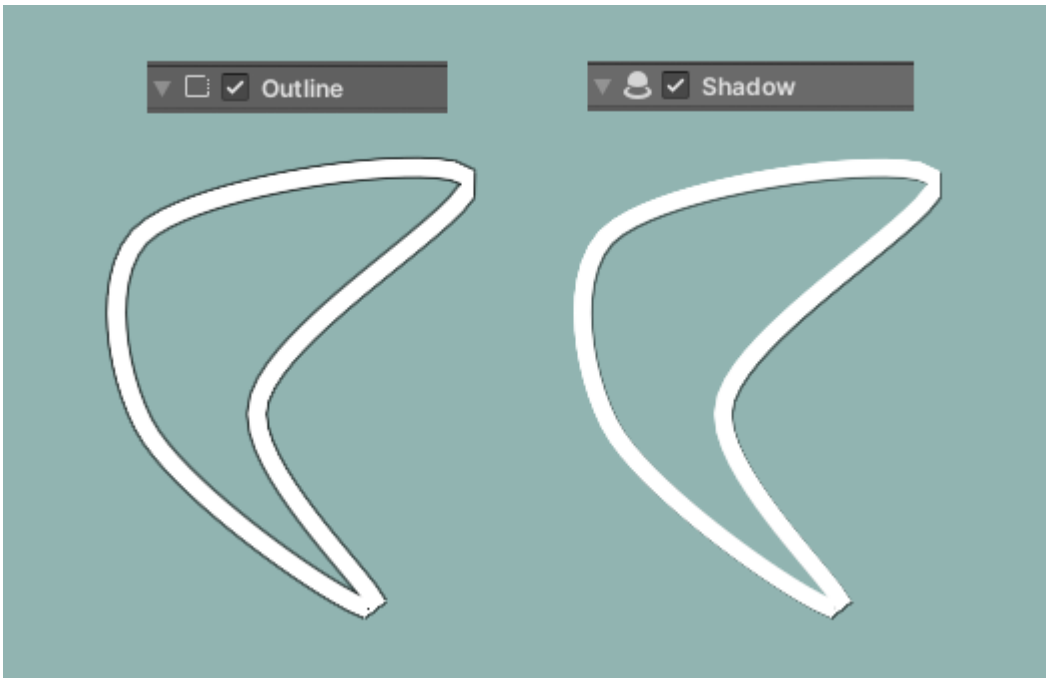


As you know, Unity's SplineContainer class can have multiple splines. And UISplineRenderer also supports multiple splines. However, keep in mind that all properties of start image and line and end image are shared with all the splines in the SplineContainer.

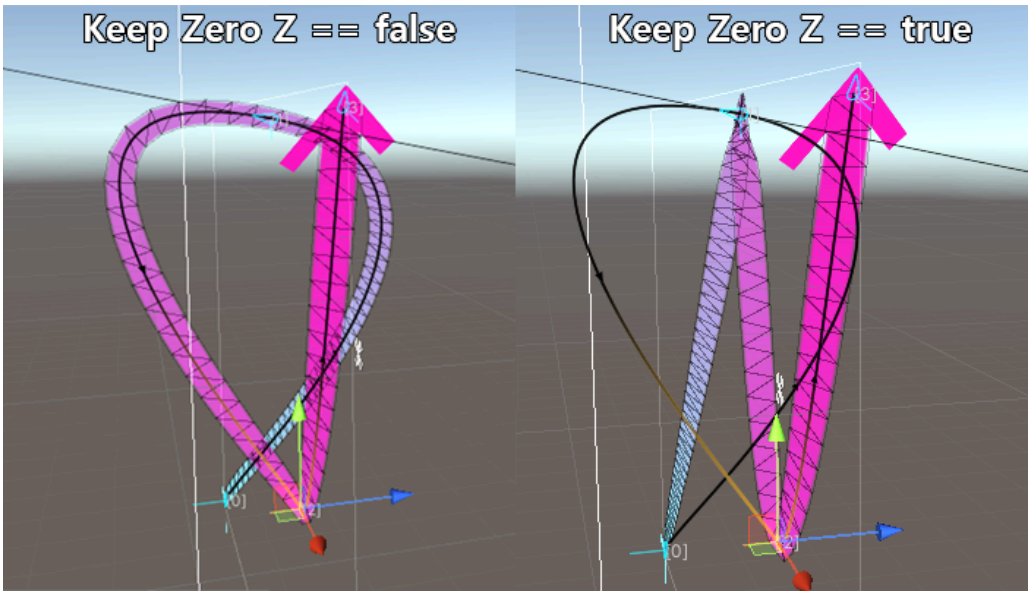
The mesh of the renderer is only created on spline changes. And It uses the Job System to optimize. But If you have too long splines with high resolution, it can cause bad performance during change splines.

Typically, 10,000+ vertices in a single UISplineRenderer is quite a lot. That can cause performance problems in mobile devices.

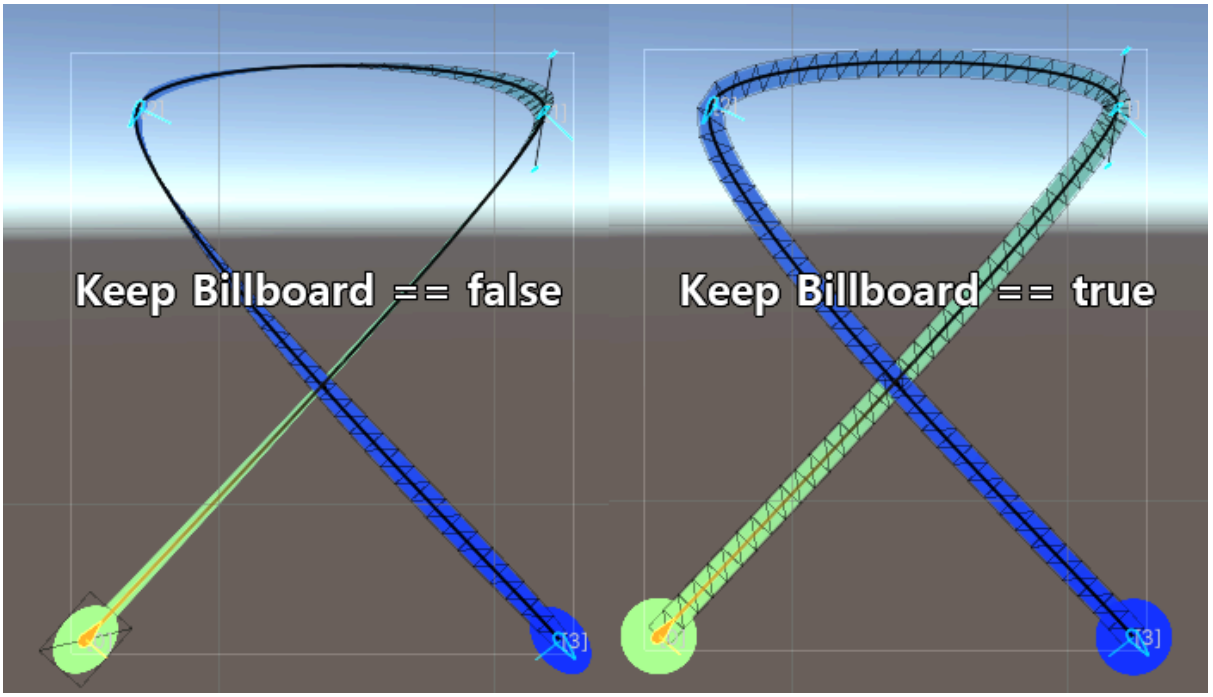
U ISplineRenderer works well with built-in Outline, Shadow component. Just add these components to it.



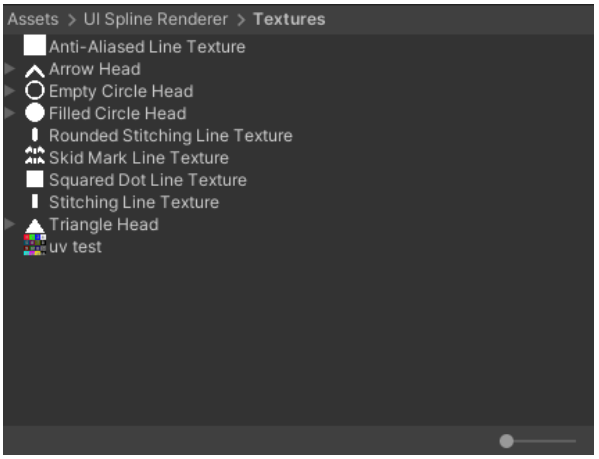
“Keep Zero Z” property is to flatten positions on canvas. It flatten the vertices as well as knot positions. The default value is true.

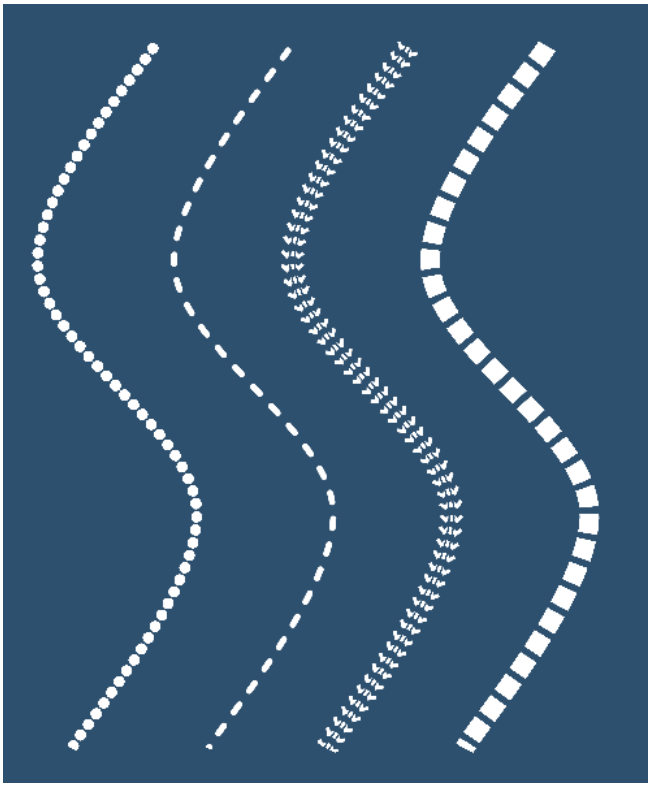


“Keep Billboard” property prevents spline twists. The default is true.



At Assets/UI SplineRenderer/Textures, basic textures are ready. Feel free to use them.



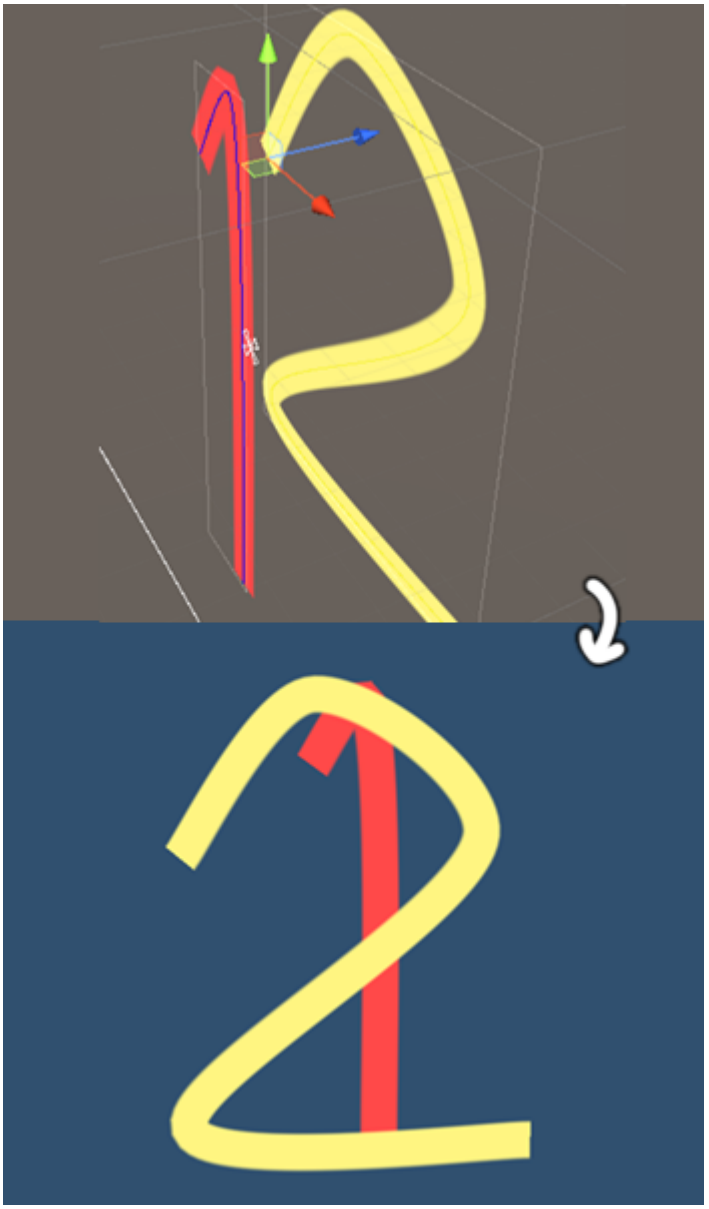


You can use your custom materials.

And if you use recursiveMaterial option, the material value will be applied to start and end images at once. Turn off the option if you want to use different materials for the image than the material for the line.

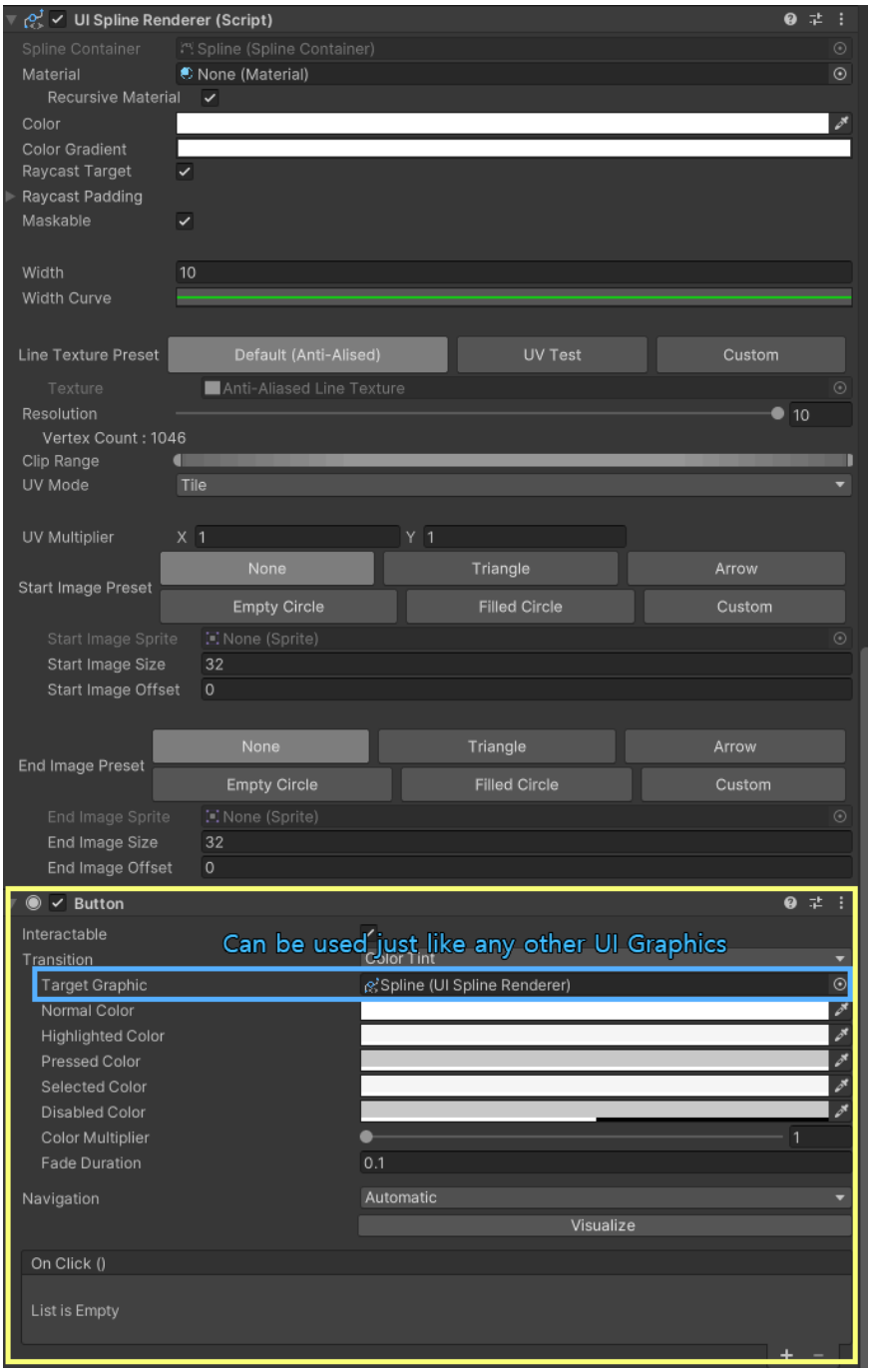
Notice!

UISplineRenderer is not sorted by z position because it's UI graphic. It is only sorted by hierarchy level of the gameObject.



3-3. Interaction

Each part of the renderer can be interacted with the graphic raycast. Start Image and End Image are just Unity.UI.Image components. And Line is also an inherited class from UnityEngine.UI.MaskableGraphic. So you can use UISplineRenderer like any other UI graphic components.



4. C# API

4-1. UISplineRenderer.cs

UISplineRenderer is a inherited class from UnityEngine.UI.MaskableGraphic so you can use all APIs of that UI graphic class like as Raycast, CrossFadeAlpha, CrosFadeColor, Maskable…

And UISplineRenderer has creation methods for quick start. You can see this at 2-2. Runtime.

Next methods are additional public methods.

```
public Color GetColorAt(float t)
```

- Get color * gradient at the given time [0..1]

```
public float GetWidthAt(float t)
```

- Get with * width curve at the given time [0..1]

```
public void SetWidthCurve(AnimationCurve curve)
```

- Change AnimationCurve for width. Use this method to change the width curve and rebuild mesh at once. Because the AnimationCurve class doesn't have a change callback and UISplineRenderer doesn't update every frame for performance.

```
public void SetColorGradient(Gradient gradient)
```

- Change ColorGradient for width. Use this method to change the color gradient because of the same reason for the SetWidthCurve().

```
public void ChangeColorGradientAlphaKey(int index, GradientAlphaKey key)
```

- Change a single alpha key of the color gradient and update the spline renderer.

```
public void ChangeColorGradientAlphaKey(int index, GradientColorKey key)
```

- Change a single colorkey of the color gradient and update the spline renderer.

```
public void ForceUpdate()
```

- Rebuild the spline renderer. Use this method if the UISplineRenderer is not updated even when properties are changed.

```
public void ReorientKnots()
```

- Rotate all knots to the screen direction. This is a simple wrapper method of the SplineExtensions.ReorientKnots(). This is an extension method for SplineContainer. It forces the rotation of the knots, so it can cause changes of the shape of the spline.

4-2. SplineExtensions.cs

Next methods are extension methods for the Spline and SplineContainer class.

```
public static void ReorientKnot(this Spline spline, int index, bool withoutNotify = false)
```

- Rotate a single knot to the screen direction.

```
public static void ReorientKnots(this SplineContainer container, bool withoutNotify = false)
```

- Rotate all knots to the screen direction. UISplineRenderer.ReorientKnots() wraps this method.

```
public static void ReorientKnotsAndSmooth(this SplineContainer container)
```

- Change all knots to screen direction oriented and auto smooth applied knots. UISplineRendeer.Create() methods are using this method.

5. Third Party Support

UISplineRenderer can be used with the TrueShadow asset.

It automatically update TrueShadow's custom hash when splines are changed.

If you have any assets that you want UISplineRenderer to support, just send me an e-mail (olivecrow.games@gmail.com).

6. Optimization

The most important thing to keep in mind when using UISplineRenderer is that it is much faster when built than in the editor.

For a long spline that draws about 2000 vertices, although it takes about 3 FPS (300 ms) in the editor, it only takes about 140 FPS (7 ms) when building.

UISplineRenderer is highly optimized by default. The renderer is only updated when splines change, and all drawing operations use Unity's Job System and Burst Compiler.

The biggest impact on performance is how often the spline is updated, followed by the number of vertices rendered. The number of vertices can be adjusted through the Resolution property in the inspector.