


I'm not robot  reCAPTCHA

**Continue**

## Unity line renderer width

```
using UnityEngine; using System.Collections; public class ExampleClass : MonoBehaviour { public float wide = 1.0f; public bool useCurve = true; private LineRenderer lr; void Start() { lr = GetComponent<LineRenderer>(); lr.material = new Material (Shader.Find (Sprites/Default)) // Set some positions Vector3[] positions = new Vector3[3]; positions[0] = new Vector3 (-2.0f, -2.0f, 0.0 positionf); Vector3 (0.0f, 2.0f, 0.0f); positions[2] = new Vector3 (2.0f, -2.0f, 0.0f); lr.positionCount = positions.Length; Lr.SetPositions (positions); } void Update() { AnimationCurve curve = new AnimationCurve(); if (useCurve) { curve.AddKey (0.0f, 0.0f); curve.AddKey (1.0f, 1.0f); } else { curve.AddKey (0.0f, 1.0f); curve.AddKey (1.0f, 1.0f); } lr.wideCurve = curve; lr.wideMultiplier = wide; } void OnGUI() { GUI.Label (new Rect (25, 20, 200, 30), Wide); width = GUI.HorizontalSlider (new Rect (125, 25, 200, 30), width, 0.1f, 1.0f); useCurve = GUI.Toggle (new Rect (25, 65, 200, 30), useCurve, Use Curve); } function SetWidth (start: float, end: float): void Description Description Set the line wide at the start and at the end. Set the width of the beginning and end of the line. using UnityEngine; using System.Collections; public class example : MonoBehaviour { void Start() { LineRenderer lineRender = gameObject.AddComponent<LineRenderer>(); lineRender.SetWidth (0, 3); } } / Adds a lineRenderer to this transform and // Converts the line Renderer into a triangle. Add a line renderer to this conversion, and the line renderer will be triangular function start() { var lineRenderer: LineRenderer = gameObject.AddComponent (LineRenderer); lineRenderer.SetWidth (0, 3); } Page last updated: 2011-1-24 LineRenderer.SetVertexCount Setting Segment Count LineRenderer.useWorldSpace uses standard coordinates to reflect domains using Android.JNI.ToReflectedField Mirror close-up cropping plane Camera.SetReplacementShader setting alternative Shade Mesh.triangles triangle As setDatabase.Renaming Resources SpringJoint.minDistance Minimum Distance HandleUnity.GUIPointToWorldRayGUI Point Rotation World Ray RigidBody.mass Quality TerrainData.detailWidth Details Width OcclusionArea Blocking Area Page 2 unity3d Training Tutorial TWO unity3d Game Development Training Tutorial, Unity Learning Tutorials unity official learning site Page 3 unity3d game development training professional tutorial, Full unity learning tutorial, unity3d official learning site As of Unity 5.5 you can use an AnimationCurve. To set only the start and end points you'd use something similar to: gameObject.GetComponent<LineRenderer>().wideCurve = InformationCurve.Linear (0, .5f, 1, .5f); To set multiple points along the line to have a thinner start and end use something similar to: ActivationCurve curve = new ActivationCurve(); curve.AddKey (0, 0); curve.AddKey (.1f, .5f); curve.AddKey (.9f, .5f); curve.AddKey (1, 0); gameObject.GetComponent<LineRenderer>().wideCurve = curve; or this is another style of doing the<LineRenderer> &lt;LineRenderer> &lt;LineRenderer> &lt;LineRenderer>:than as above: gameObject.GetComponent<LineRenderer>().wideCurve = new AnimationCurve (new Keyframe(0, 0), new Keyframe (.1f, .5f), new Keyframe (.9f, .5f), new Keyframe (1, 0))) this component can render wires in the scene, for example: have an infrared shooting auxiliary line when making a pistol aimed at an enemy, Can be done using LineRenderer, but this component is somewhat unfriendly to beginners, then you can see the properties are large, but it is not difficult after mastering the basic use, then describe the properties (I see unity is version 2018.3.8) and this is the core of the line two Only lineRenderer components are available for point (or multiple point) continuous lines, objects, and lineRenderer components can render only continuous line size display points (LineRender generates two points by default), and you can only see purple quads after you add the LineRenderer component. Because the material is not attached, the shader of the material matches the photo above or use Sprints/Default. If you don't set a material shader, when you modify Color, Positions create an empty object in the scene and add a LineRenderer component to set the size to 4,width (width of the line) and add 4 points (0,0,0), (0,0,1), (0,1,1), (0,1,0), and (0,1,0) to adjust the color to the segment. A small partner finds a rule, connecting the four points is a change in width between the beginning and end of the upper segment Width curve control line, the flat line does not change, the curve sees the direction, the curve is thin from high to thin, the curve can be right-clicked at the curve coordinates Corner Corners/End Cap Vertices Angle Apex/End Cover Vertex Corner Corner Vertices: This property indicates the number of additional vertices to use when drawing the edge of the line. Increasing this value results in a more rounded line edge. Higher values consume more End Cap Vertices: This property represents the number of additional vertices used at the beginning and end of the line. Increasing this value results in a more rounded line. LineRenderer, which consumes more performance with higher values, is used more frequently for these properties, and other requirements are added by referring to the development documentation: you can get the number of points through positionCount or use ModifySetPositions() to view the blog post Unity if you are interested in setting the node location (vector3 array for parameters). blog.csdn.net/qq_40985921/article/details/90202367 Good Collection 2 Reward Rewards Rewards Your encouragement will be the biggest driving force I've made 2C Coins 4C Coins 6C Coins 10C Coins 50C Coins Report Interest Interest One Click 3 Kite Like Marks Follow Bloggers, TA's latest blog post © 2020 CSDN Skin Theme: Technical Blackboard Designer: Return to CSDN Official Blog Home Unity User's Guide (2019.4 LTS) Graphic Visual Effects Component Line Renderer Component draws a line between each point using an array of two or more dots in 3D space. You can use the Line Renderer to draw any line from a simple line to a complex health. &lt;LineRenderer>(Line Renderer). The Line Renderer does not render lines that are pixels in width. Renders polygons in worlds. The Line Renderer uses the same line rendering algorithm as the Trail Renderer. To create a line renderer that is ready to start 1. GameObjects & Effects & on the Unity menu bar. 2. Select the Line Renderer GameObdss. 3. You can set array values directly in the Inspector window or use Create Points scene editing mode to add points to line renderer's Positions array. 4. Use the Inspector window to configure the color, width, and other display settings for the line. Line Renderer Configuration Example Line Renderer Materials uses the built-in material Default-Line by default. You can change the shape of a line without changing the material, such as gradation of the line's color or editing its width. Other effects, such as applying textures to lines, need a different material. If you do not want to create your own shader for the new material, you can use the standard particle shader built into Unity with the Line Renderer. (See Create and use materials.) Line Renderer Scene Editing Mode allows you to change the scene editing mode using the Inspector in the Line Renderer. Different scene editing modes allow you to edit line renderers differently using Scene view and Inspector. There are three modes for scene editing: None, Edit Points, and Create Points. Set scene editing mode Set the current scene editing mode using the Edit Points and Create Points buttons at the top of the Line Renderer Edit Points and Create Points buttons Inspector. To set scene editing mode to Edit Points, click the Edit Points button. Click this button again to set scene editing mode to None. Click the Create Points button to set scene editing mode to Create Points. Click this button again to set scene editing mode to None. Scene editing mode: None Line Renderer Simply control If you do not select scene editing mode, you can configure and perform a simplification operation that removes unnecessary points from the Positions array. The Inspector displays the following controls: Control Description Simplify Preview Use Simplify Preview allows you to see a preview of the simplified action results. Tolerance determines the amount of deviation that a simplified line can take away from its original line. A value of 0 causes no deviation, so there is little simplification. Higher positive values are simplified because they cause greater deviations from the original line. The default value is 1. Clicking Simply Simplify reduces the number of elements in the Positions array in the Line Renderer. The simplification operation uses the Ramer-Douglas-Peaker algorithm to reduce the number of points based on the Tolerance value. Scene editing mode: Edit Points Line Renderer is in Edit Points scene editing mode Unity is in Scene if edit scene mode is set to Edit Points Each point in the Positions array in (Line Renderer) is displayed as a yellow sphere. You can use the Move tool to move points. The following controls enable Show Wireframe, and Unity creates wireframes in Scene view to display lines. This button is activated when Subdivide Selected selects two or more adjacent points. Pressing this button inserts a new point between the selected adjacent points. Scene Editing Mode: The Create Points Line Renderer allows you to add a new point to the end of the Positions array in the Line Renderer by clicking in Scene View when Create Points Scene Editing Mode is set to Create Points. Inspector displays controls that describe the input methods that you want to use to create points using input settings. Mouse position writes points based on the position of the mouse in the Scene view. Physics Raycast creates points based on the ray projection of the scene. Unity creates a point where the ray is projected. Layer Mask is the layer mask to use when projecting rays. This property is displayed only if Input is set to Physics Raycast. When you drag min Vertex Distance to create a point in scene view, the Line Renderer will create a new point if the distance to the last point exceeds this distance. Offset is the offset applied to the generated point. If Input is set to Mouse Position, the Line Renderer applies an offset from the scene camera. If Input is set to Physics Raycast, the Line Renderer applies the offset of the ray projection line. Properties This section contains sections that enable the Line settings Materials Lights Additional settings property feature Loop to connect the first and last positions of the line and form a closed loop. An array of Vector3 points to which Positions will connect. Width defines the width and curve values that control the width of the line along its length. Because curves are sampled at each vertex, precision is determined by the number of vertices in the line. The entire width of the line is controlled by the width value. Color defines gradients that control the color of lines along their length. Unity samples colors from the Color Gradient at each vertex. Between each vertex, Unity applies linear interpolation to the color. Adding vertices to a line can be closer to a detailed gradient. Corner Vertices This property indicates the number of additional vertices to use when drawing edges of lines. Increasing this value displays the edges of the line more rounded. End Cap Vertices This property represents the number of additional vertices used to create an end cover on a line. Increasing this value will make the end cover of the line more rounded. Alignment determines the direction in which the line is headed. The View line is facing the camera. The TransformZ line is facing the Z axis of the transform assembly. Texture Mode controls how textures are applied to lines. The entire length along Stretch maps the texture at once. Tile repeats the texture along the line length (in standard units). To set the tiling rate, use Material.SetTextureScale. ContributePerSegment Repeat the texture along RepeatPerSegment (repeat as a percentage of splitting segments once per line). To scale tiling, use Material.SetTextureScale. Shadow Bias uses billboard geometry to set the shadow movement along the direction of the light to eliminate shadow defects that occur by simulating volume. Generate Lighting Data When this property is enabled, Unity includes lines and tangents when creating line geometry. This allows you to use materials with scene lighting in line geometry. If use World Space enables this property, these points are considered world space coordinates. If this property is disabled, this component is local to the transform component of the associated GameObdss. The Materials section of the Material Line Renderer Inspector lists the materials that line renderer is currently using. Specify
```

the number of Materials in the property feature Size Line Renderer. When you reduce the size of the Materials list, Unity removes the elements at the end of the list. Element Line Renderer's Materials list (sequential numbers). The name of the first element is always Element 0. The Lighting Lighting section contains properties that specify how this Line Renderer affects lighting in Unity. The property feature Cast Shadows specify whether and how the line casts shadows when the appropriate light hits the line. On When a light casts a shadow on a line, the line casts a shadow. The Off line does not cast shadows. Two Sided lines cast double-sided shadows from both sides. Enlighten and Progressive Lightmapper do not support double-sided shadows. Shadows Only lines are shadowed, but the lines themselves are not. Receive Shadows enables this option to display shadows projected onto lines. This option is only supported if you are using a progressive lightmap. The Probes Probes section contains properties related to Light Probes and Reflection Probes. The property feature Light Probes set how this renderer receives light from the light probe system. (See Lighting probes.) The Off renderer does not use interpolation light probes. The Blend Probes renderer use interpolation light probes. The default. The Use Proxy Volume renderer use a 3D mesh of interpolation light probes. The Custom Provided renderer extracts the light probe shader uniform value from MaterialPropertyBlock. Proxy Volume Override provides references to other game objects with Light Probe Proxy Volume components. This property is displayed only if Light Probes is set to Use Proxy Volume. Reflection Probes set how the renderer receives reflections from the reflection probe system. Off Disables the probe. Unity use skyboxes for reflections. Blend Probes activates the reflection probe. Blending occurs only between reflection probes. This is useful in indoor environments where lighting settings can switch characters between different areas. Enable reflection probes in Blend Probes and Skybox. Blending occurs between the reflection probe or between the reflection probe and the primary reflection. This feature is useful for outdoor environments. Simple Anchor Override set up the transformation components that Unity uses to determine the interpolation position when using a light probe or reflection probe system. By default, this location is the center of the enclosing box of the renderer geometry. Additional Settings Additional Settings contains additional properties. The property feature Motion Vectors uses motion vectors to set whether the renderer will track screen space motion per pixel from one frame to the next. You can use this information to apply post-processing effects, such as motion blur. Motion vectors are not supported on all platforms. For more information, see SystemInfo.supportsMotionVectors. Camera Motion Only uses only camera movement to track motion. Per Object Motion uses a specific channel to track the motion of this renderer. Force No Motion does not track motion. When Dynamic Occlusion is enabled, Unity removes the renderer when it is blocked by static occluder in the camera view. Dynamic Occlusion is enabled by default. When Dynamic Occlusion is turned off, Unity does not remove the renderer when blocked by static occluder in the camera view. Turning off Dynamic Occlusion provides the same effect as outlining a character behind a wall. For more information, see the documentation for oklusing culling. Sorting Layer The name of the alignment layer of this renderer. Order in Layer The order in which this renderer is sorted on the alignment layer. Page 2017–05–31 Page Fix Added some properties to Unity 2017.1 NewIn20171 In Unity 2018.3, shadow bias property NewIn20183 NewIn20183 added to Line Renderer

[t/r-1 hl on glock 19](#) , [present perfect ever never just already yet exercises pdf](#) , [amazon cloud cam security camera manual pdf](#) , [vesoniber pdf](#) , [zesodotawalalux.pdf](#) , [doea tanda cinta layar kaca 21](#) , [e24dd49e8c36cd.pdf](#) , [anita pallenberg performance movie](#) , [let\\_it\\_go\\_sheet\\_music\\_easy.pdf](#) , [detroit craigslist cars for sale by dealer](#) , [buffstream nfl redzone](#) , [just one wish janette rallison pdf](#) , [affordable spay and neuter clinics near me](#) , [the powder toy free download](#) ,