Pico Games

Indie games by Jake

Navigate to...

QuickRopes Documentation

Current Version: 3.0.0 *(Submitted – 04/25/2015)(Released – ??/??/2015)*
Documentation: 04/25/2015

Table of Contents

Links and Other Info

You can find out more information about this and other products by following the links below. Check this section for periodic updates as I plan to add video tutorials in the near future!

Website: Check us out at picogames.com!

Contact:
– Email: reveriejake87@gmail.com
– Facebook: www.facebook.com/picogamesstudio
– Twitter: www.twitter.com/reveriejake

Documentation: Get up to date documentation (helpful if you are viewing this from a PDF)

Asset Store: Check out our other products available on the Asset Store!

What is QuickRopes?

QuickRopes is designed to take the pain out of setting up complex rope systems. The goal is to make designing and setting up these systems easier for indie and professional developers. The goal of this product is to help you make your game development easier and more streamlined.

Tools included with the QuickRopes package

- QuickRopes
- Spline Tool
- Triangulation Code

Installing QuickRopes

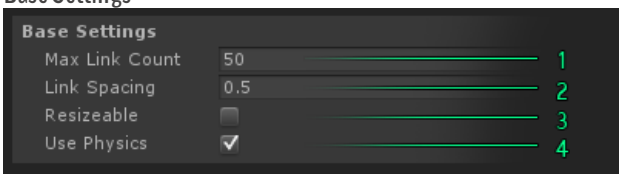
Note: If you are upgrading from QuickRopes 2 you must start fresh as the two are not compatible with each other!

1. Remove previous versions of QuickRope from your project. Please keep in mind that this is **not necessary** so if you have older versions of QuickRopes in use feel free to keep those.
2. Import the Unity Package from the Asset Store
3. Verify that the package has properly installed by ensuring the path **[GameObject -> 3D Object -> QuickRopes]** is available.

Inspector Overview

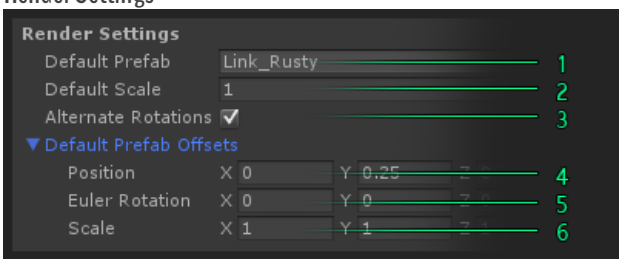
The QuickRopes inspector is where you can get down and dirty with the minor details of how your QuickRope behaves. Below you will find some descriptions of what each line of the inspector will do!

Base Settings



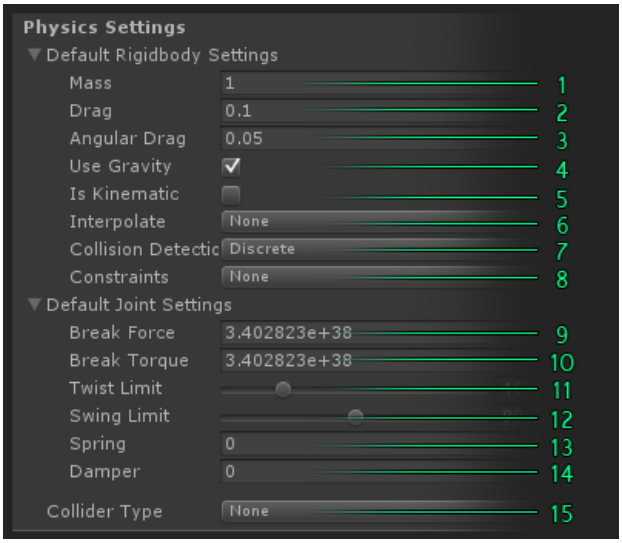
1. **Max Link Count:** Sets the maximum number of links that will spawn when creating your rope.
   [Default: 50, Min: 1, Max: *MAX_JOINT_COUNT*]
2. **Link Spacing:** Sets the space between each of the links. For less links in your chain increase this value. For more links in your chain decrease this value.
   [Default: 0.5, Min: 0.001, Max: Infinity]
3. **Resizeable:** If enabled the rope will create a cache set of joints in the size of your 'max link count' value. If disabled joints will only be generated tot he exact length of your rope.
   [Default: false]
4. **Use Physics:** If enabled the rope will generate joints and rigidbodies for each of the links. If disabled joints and rigidbodies will be destroyed from the links.
   [Default: true]

Render Settings



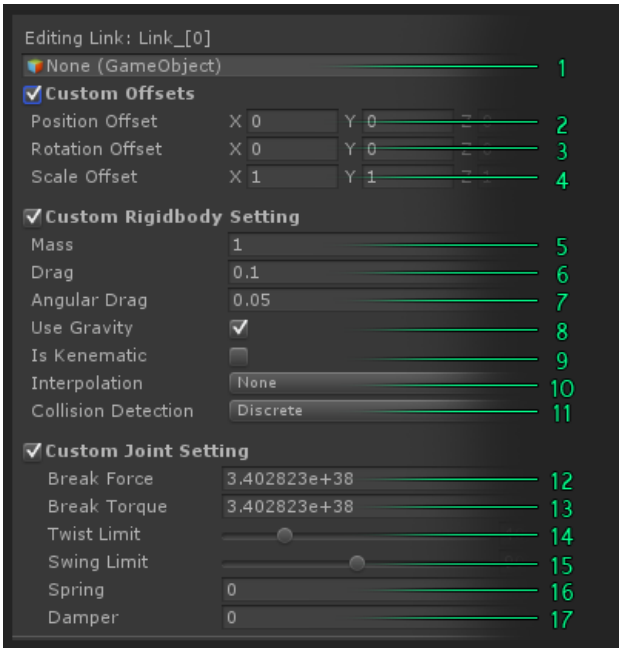
1. **Default Prefab:** Sets the default prefab to be spawned in for each link. *This value can be null and will result in an empty set of links. This way you can create custom mesh renderers for the rope.*
   [Default: Link_Rusty]
2. **Default Scale:** Sets the default scale multiplier of the set prefab. For example if your 'scale' value is (4,4,4) and this value is 2 the resulting prefab scale will be (8, 8, 8).
   [Default: 1, Min: 0.001, Max: Infinity]
3. **Alternate Rotations:** If enabled the *even* links will be rotated by 90 degrees.
   [Default: 90, Min: -Infinity, Max: Infinity]
4. **Position:** If enabled the rope will generate joints and rigidbodies for each of the links. If disabled joints and rigidbodies will be destroyed from the links.
   [Default: 0/0/0]
5. **Euler Rotation:** Default rotation of the set prefab in euler angles.
   [Default: 0/0/0]
6. **Scale:** Default scaling of the set prefab.
   [Default: 1/1/1]

Physics Settings



1. **Mass:** Sets mass.
   [Default: 1, Min: 0.001, Max: Infinity]
2. **Drag:** Sets drag.
   [Default: 0.1, Min: 0, Max: Infinity]
3. **Angular Drag:** Sets angular drag.
   [Default: 0.05, Min: 0, Max: Infinity]
4. **Use Gravity:** Sets gravity enabled.
   [Default: true]
5. **Is Kinematic:** Sets kinematic.
   [Default: false]
6. **Interpolate:** Sets interpolation.
   [Default: None]
7. **Collision Detection Mode:** Sets collision mode.
   [Default: Discrete]
8. **Constraints:** Sets constraints.
   [Default: None]
9. **Break Force:** Sets break force.
   [Default: Infinity, Min: 0, Max: Infinity]
10. **Break Torque:** Sets break torque.
    [Default: Infinity, Min: 0, Max: Infinity]
11. **Twist Limit:** Sets twist limit.
    [Default: 45, Min: 0, Max: 180]
12. **Swing Limit:** Sets swing limit.
    [Default 90, Min: 0, Max: 180]
13. **Spring:** Sets spring value.
    [Default: 0, Min: 0, Max: Infinity]
14. **Damper:** Sets damper value.
    [Default: 0, Min: 0, Max: Infinity]
15. **Collider Type:** Selects collider type. Useful when not using a prefab or when using a prefab with no collider.
    [Default: None, (None, Box, Sphere, Capsule)]

Link Settings



1. **Prefab:** Sets the custom prefab of the selected link. If this is left null then the default prefab from the 'Base Settings' pane will be used.
   [Default: Null]
2. **Position Offset:** Sets the custom position offset of this link. This value will always be used when the 'Custom Offsets' toggle is enabled.
   [Default: 0/0/0]
3. **Rotation Offset:** Sets the custom Euler rotation offset for this link. This value will always be used when the 'Custom Offsets' toggle is enabled.
   [Default: 0/0/0]
4. **Scale Offset:** Sets a custom scale for the selected link. This value will always be used when the 'Custom Offsets' toggle is enabled.
   [Default: 1/1/1]
5. **Mass:** Sets the custom mass for this link. This value will always be used when the 'Custom Rigidbody Settings' toggle is enabled.
   [Default: 1, Min: 0.001, Max: Infinity]
6. **Drag:** Sets the custom drag for this link. This value will always be used when the 'Custom Rigidbody Settings' toggle is enabled.
   [Default: 0.1, Min: 0, Max: Infinity]
7. **Angular Drag:** Sets the custom angular drag for this link. This value will always be used when the 'Custom Rigidbody Settings' toggle is enabled.
   [Default: 0.05, Min: 0, Max: Infinity]
8. **Use Gravity:** Sets the custom gravity is enabled for this link. This value will always be used when the 'Custom Rigidbody Settings' toggle is enabled.
   [Default: true]
9. **Is Kinematic:** Sets the custom kinematic for this link. This value will always be used when the 'Custom Rigidbody Settings' toggle is enabled.
   [Default: false]
10. **Interpolate:** Sets the custom interpolation for this link. This value will always be used when the 'Custom Rigidbody Settings' toggle is enabled.
    [Default: None]
11. **Collision Detection Mode:** Sets the custom collision mode for this link. This value will always be used when the 'Custom Rigidbody Settings' toggle is enabled.
    [Default: Discrete]
12. **Break Force:** Sets break force.
    [Default: Infinity, Min: 0, Max: Infinity]
13. **Break Torque:** Sets break torque.
    [Default: Infinity, Min: 0, Max: Infinity]
14. **Twist Limit:** Sets twist limit.
    [Default: 45, Min: 0, Max: 180]
15. **Swing Limit:** Sets swing limit.
    [Default 90, Min: 0, Max: 180]
16. **Spring:** Sets spring value.
    [Default: 0, Min: 0, Max: Infinity]
17. **Damper:** Sets damper value.
    [Default: 0, Min: 0, Max: Infinity]

Spline Settings

1. **Is Looped:** Sets whether the spline is to be looped or not. If only 2 points are on your spline this will add a third point to the spline.
   [Default: false]
2. **Evenly Distribute:** If this is set to true then the spline point algorithm will attempt to evenly distribute its points along the spline so that the joints fill the spline up

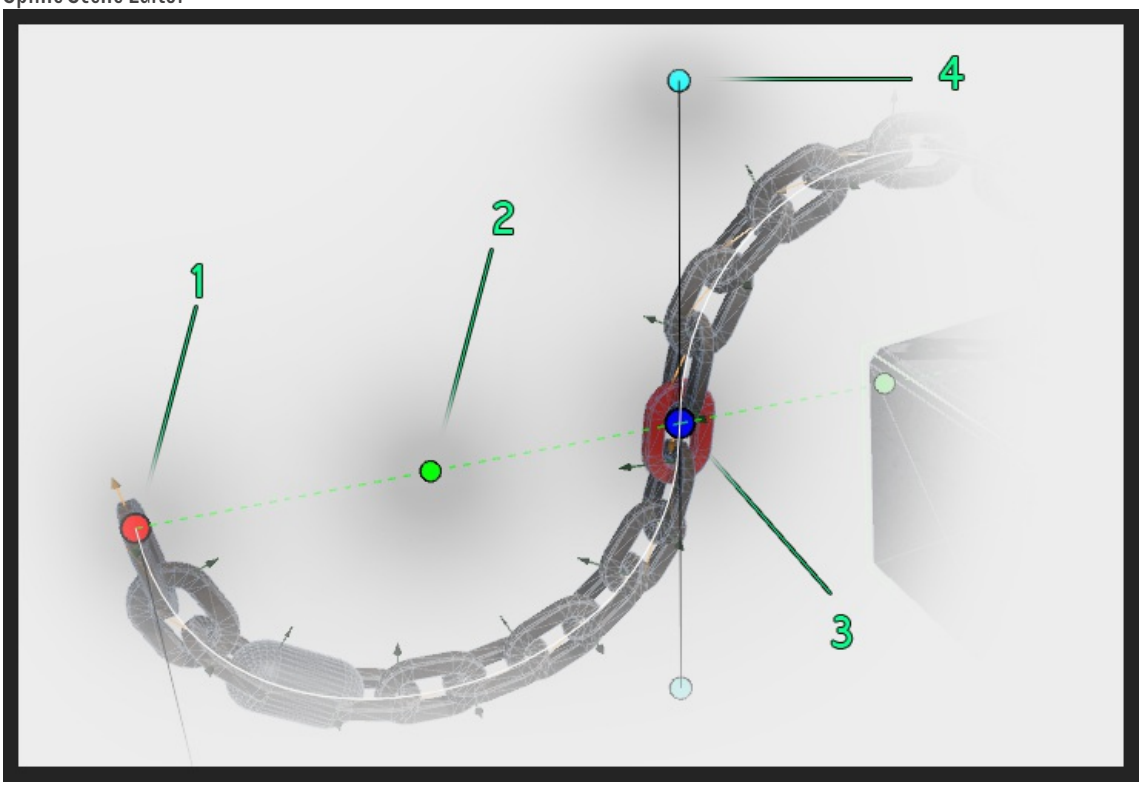
completely. A good reason to have this set to false is if you are making a grapple and need the joints to flow out of the origin smoothly.
[Default: true]

3. **Output Resolution:** This value is the 'search' value for the point distribution. By default it is set to 1000 which means the spline will iterate through itself 1000 times in an attempt to distribute points along itself. A very low value will make the joints spread out and be placed less accurately.
[Default: 1000]
4. ***Recommended Min. Output Resolution:*** *This is a read-only display of what the above resolution should be set to at minimum. You do not have to follow this value but to get the best positioning possible it is recommended that you keep the Output Resolution value above this suggestion.*
5. **Edit In Play Mode:** When this is set to true it allows you to see the spline editor in the scene view during play mode (Editor only).
[Default: false]

Spline Editor Overview

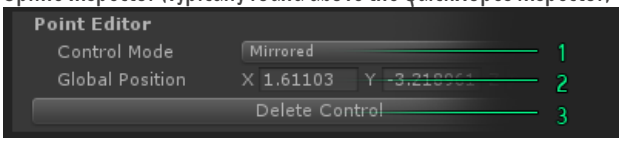
QuickRopes 3 features a brand new spline editor. Below you will find some basic information about what each component of this new editor will do. *This new powerful feature has been separated into its own monobehaviour to allow you to use it with your own game scripts!*

Spline Scene Editor



1. **Root:** The root control will move the entire rope gameobject. This is the only control point that cannot be moved away from the gameobjects origin.
2. **Add Control Point:** A point that when clicked will add a new control point. The added point will be placed at the origin of this point.
3. **Control Point:** A movable point at which the spline will intersect.
4. **Control Tangent:** A movable point which controls the curve of the spline between control points.

Spline Inspector (Typically found above the QuickRopes inspector)



1. **Control Mode:** Sets how the tangent controls behave. When set to 'Mirrored' the tangent opposite to the selected one will mirror its angle and distance to the selected tangent. When in 'Aligned' mode the tangents will be mirrored but will not maintain equal distance from the control point. When in 'Free' mode the tangents will operate independently of each other.
[Default: Mirrored, (Mirrored / Aligned / Free)]
2. **Global Position:** Sets the global position of the selected control point. If you have a tangent selected it will set the selected tangents position and follow the control mode rules.
3. **Delete Control:** This will delete the currently selected control point. It is inactive when there are only 2 points or when the root control is selected.

Creating a Looped Chain

One of the most highly requested features aside from the grapple hook is the ability to create looped ropes. With QuickRopes 3 creating loops has never been easier! With the new spline editor its a single click operation. Below you will find a quick tutorial on how to turn your rope or chain into a loop!

Scene View of Looped Chain

1. If you have not already done so, create a new QuickRope by going to GameObjects -> 3D Object -> QuickRope
2. Click ont he 'Spline Editor' tab on the QuickRopes inspector.
3. Finally, toggle the 'Is Looped' parameter to true!

You can also toggle the looped feature to on via script.

```
rope.IsLooped = true;
```

Connecting Rigidbodies

Unlike previous versions of QuickRopes, in version 3 connecting other rigidbodies to the rope or chain is straight forward. When a rope is generated it creates a list of links inside of the rope. You can see these links by viewing the children of the rope gameobject in edit mode or via script by accessing the Links array.

To link a rigidbody to a link on the rope simply add a joint to the gameobject you would like to connect to. And assign one of the links in the rope to the connectedbody parameter of the joint! The joint should maintain the reference as long as the referenced link isn't destroyed.

Using the Rope Renderer

The official rope renderer for version 3+ is still under development but I expect it to be finished very soon after release! For now you have the option to write your own mesh rendering code or to download SixthSensors awesome tube renderer from the asset store. If you decide to download the tube renderer please also download the extension script I provided specifically for SixthSensors asset.

Download Links:

- **"TubeRenderer" by Sixth Sensor**
- **Extension scripts for Sixth Sensors "TubeRenderer"**

Extending/Retracting Rope

Controlling the rope in QuickRopes 3 is as simple as setting a single variable!

- Setting the **ResizeVelocity** parameter to a positive number will slowly will increase the rope's length:

```
rope.ResizeVelocity = 5;
```

- Setting the **ResizeVelocity** parameter to a negative number will slowly decrease the rope's length:

```
rope.ResizeVelocity = -5;
```

- And finally, setting the **ResizeVelocity** parameter to zero will slowly stop the rope from re-sizing:

```
rope.ResizeVelocity = 0;
```

If you would like to change the acceleration or deceleration speed you can do so with the **ResizeAcceleration** parameter and the **ResizeDampening** parameter.

- Setting the **ResizeAcceleration** and **ResizeDampening** parameters:

```
rope.ResizeAcceleration = 1.0f;
rope.ResizeDampening = 0.98f;
```

Below you will find the "RopeController.cs" script which is available in the sample scripts folder.

Default RopeController.cs script

```
using UnityEngine;
using System.Collections;

// When working with Quickropes be sure you include
// the 'PicoGames.QuickRopes' namespace!
using PicoGames.QuickRopes;

[RequireComponent(typeof(QuickRope))]
public class RopeController : MonoBehaviour
{
   [Min(1)]
   public int minJointCount = 3;
   [Min(0.001f)]
   public float maxSpeed = 5;
   [Range(0, 1f)]
   public float acceleration = 1f;
   [Range(0.001f, 1)]
   public float dampening = 1f;

   private QuickRope rope = null;

   void Awake()
   {
      rope = GetComponent<QuickRope>();

      if (rope.Spline.IsLooped)
      {
         enabled = false;
         return;
      }

      if (!rope.canResize)
      {
         rope.maxLinkCount = rope.Links.Length;
         rope.canResize = true;
         rope.Generate();
      }

      rope.minLinkCount = minJointCount;
   }

   void Update()
   {
      rope.ResizeAcceleration = acceleration;
      rope.ResizeDampening = dampening;

      if (Input.GetKey(KeyCode.UpArrow))
         rope.ResizeVelocity = maxSpeed;
      else if (Input.GetKey(KeyCode.DownArrow))
         rope.ResizeVelocity = -maxSpeed;
      else
         rope.ResizeVelocity = 0;
   }
}
```

Scripting with QuickRopes

More info coming soon!