

<b>Generate rope via code</b>	<b>2</b>
RopeGenerator	2
IRopeMeshGenerator	3
RopeMeshCylinderGenerator	3
<b>Generate rope via tool.</b>	<b>3</b>
<b>Replace mesh generator</b>	<b>4</b>
<b>Example</b>	<b>4</b>

# Generate rope via code

To create rope, you need:

- 1) Create instance of mesh generator, see IRopeMeshGenerator
- 2) Create instance of RopeGenerator
- 3) Invoke method "MakeOne()".

## RopeGenerator

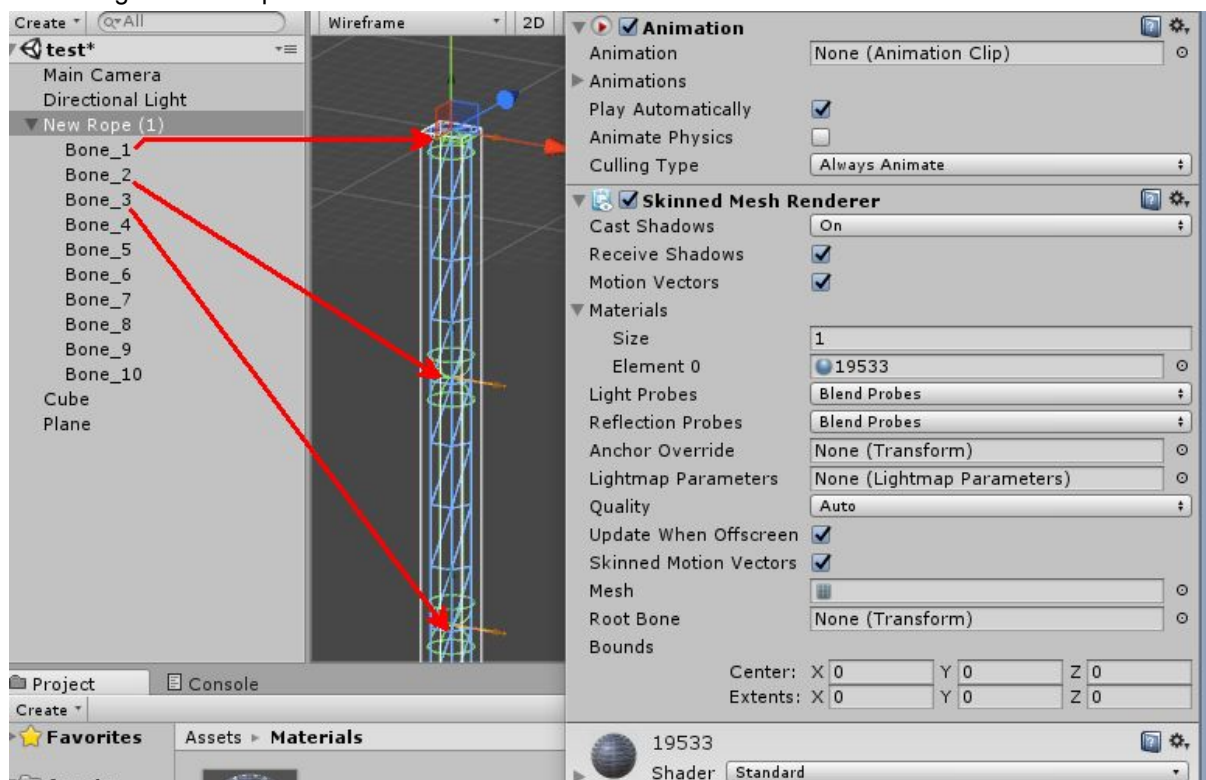
Generator have only one constructor:

[RopeGenerator](#)(IRopeMeshGenerator meshGenerator, Material material, GameObject gameObject)

Where,

- gameObject is object where Animator will be applied and bones will be added.

Result of generated rope:



You can specify some parameters before you invoke method "MakeOne()":

- JointAngleLimit - max angle between two adjacent bone
- BoneCount - count of bones in a rope
- MassOfBone - mass of each bone
- Length - total length of rope
- Radius - radius of rope
- RestrictFirstBone - if false, JointAngleLimit and skeletal weights will not be applied to first bone.

Method 'MakeOne' returns sequence of bones.

## IRopeMeshGenerator

Interface with only one method:

```
/// <summary>
/// Mash generator for rope generator
/// </summary>
public interface IRopeMeshGenerator
{
    /// <summary>
    /// Create mesh
    /// </summary>
    /// <param name="radius">Radius of rope</param>
    /// <param name="length">Length of rope</param>
    /// <param name="boneCount">Bones the mesh consist of</param>
    /// <param name="restrictFirstBone">If false, JointAngleLimit and skeletal weights will not be
    applied to first bone</param>
    Mesh Create(float radius, float length, int boneCount, bool restrictFirstBone);
}
```

This method must create mesh and set weights.

## RopeMeshCylinderGenerator

RopeMeshCylinderGenerator - is implementation of [IRopeMeshGenerator](#) that generates a simple cylinder.

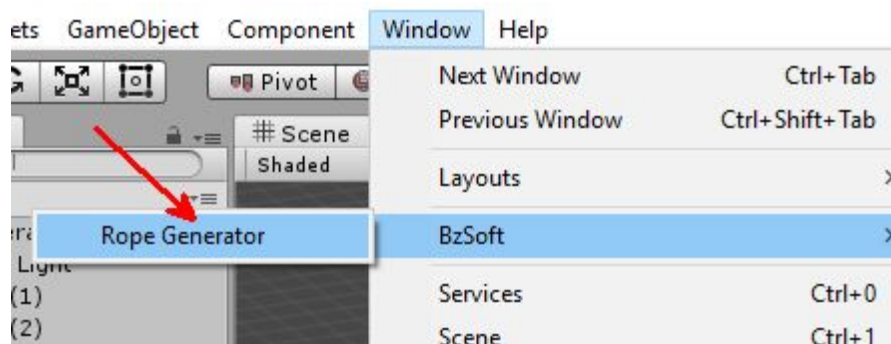
This class have only one parameterless constructor.

Some properties of [RopeMeshCylinderGenerator](#) that you can tweak:

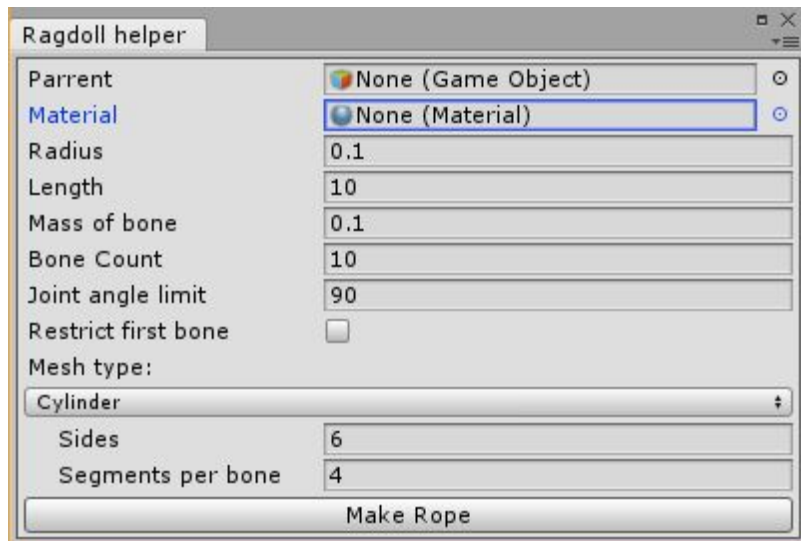
- Sides - Sets the number of sides around the cylinder.
- SegmentsPerBone - Sets the number of divisions along the bone's major axis.

## Generate rope via tool.

Open tool:



Dialog:



Here you can tweak any parameter and then click “Make rope”.  
Description of properties see in “Generate rope via code” section.

## Replace mesh generator

If you need to use your one mesh generator, you need to create new class that implements `IRopeMeshGenerator` interface and pass it to constructor of `RopeGenerator`.

## Example

```
// create main game object of your rope
var go = new GameObject("MyTestRope");
// set position of rope start point
go.transform.localPosition = new Vector3(0, 0, 0);

// prepare material for rope mesh
var material = new Material(Shader.Find("Diffuse"));

// create mesh generator
var meshGenerator = new RopeMeshCylinderGenerator();
// here you can set some properties of mesh generator.
// In line below, I set mesh to have 8 sides:
meshGenerator.Sides = 8;

// create rope generator
var rope = new RopeGenerator(meshGenerator, material, go);
// here you can set some properties of rope generator.
// In line below, I set mesh to have 8 sides:
rope.Length = 12;

// generate rope
GameObject[] bones = rope.MakeOne();
// you can use `bones` variable to get all generated bones.
// e.g., to attach sinker to the end of your rope (to last bone)
```