Spline Mesh Renderer v2.0
User Manual

WSM GAME STUDIO

Doing all the hard work for you!

2019

SUMMARY

1 - Intro

Thank you for purchasing “Spline Mesh Renderer”!

This package contains all that you need to harvest the power of Splines to:

- Generate long curved meshes (Spline Mesh Renderer)
- Spawn objects along a curved path (Spline Prefab Spawner)
- Make objects follow curved paths (Spline Follower)

It’s really simple to use and customize.

Follow this tutorial or watch it on youtube if you like:

Youtube Tutorial

1.1 - How it Works

This package is composed of four main components.

- Spline
- Spline Mesh Renderer
- Spline Prefab Spawner
- Spline Follower

The Spline component is an Editor Extension for creating and editing Splines Paths on the Unity Editor. It allows you to create a reference path of any length composed by sections of any curvature.

The Spline Mesh Renderer Component is responsible for generating and exporting meshes. It works by extruding a base mesh segment along a Spline path. This process allows the procedural generation of long and curved meshes such as roads, railroads, power lines, pipes, etc.

The Spline Prefab Spawner Component is responsible for spawning objects along a Spline Path. It can be used to easily populate your scene with objects that repeats along a path, such as light poles, bridge pillars, check points, etc.

The Spline Follower Component is responsible for moving objects along one or multiple Spline Paths. It can be used to easily create dynamic moving props for visual or gameplay purposes, such as complex platforms movement, orbiting behaviour, etc.

2 - Spline

The Spline component is an editor extension for creating Bezier Splines on the Unity Editor. A Bezier Spline is a curved path (open or closed) composed by one or more Bezier Curves.

Each curve is composed by 4 points (2 control points and 2 handles). The control points are represented on the Scene Window as green circles. They mark the start and end of each bezier curve.

The handles are represented by circles attached to the control points by red lines. They are used to adjust the curvature. The handle color changes accordingly to the Handle Alignment mode selected (See the Handles section for more info)

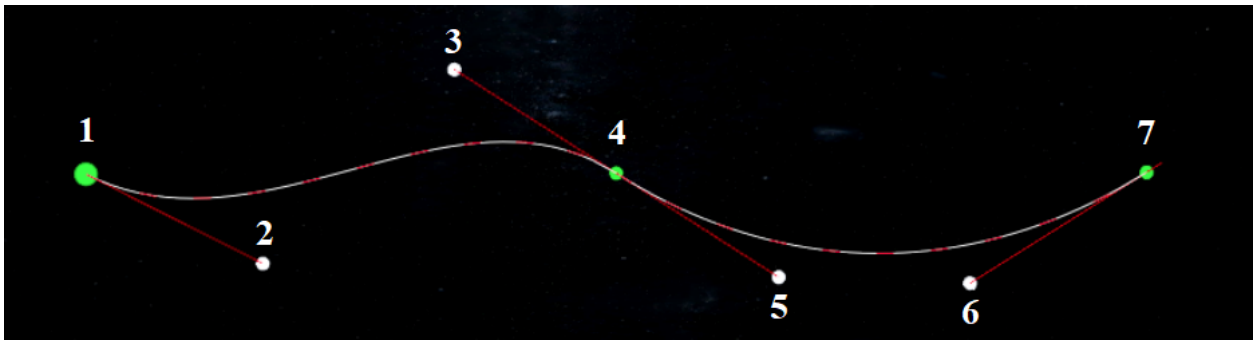

Figure 1 - spline composed by 2 bezier curves

In the example above, you can see a spline composed by 2 bezier curves. Note that, the green points marks the beginning and the end of each curve and the curvature is defined by the position of the handles (white circles connected by red lines).

To better understand how Bezier Splines works, take a look on the following affirmations about the spline shown in the sample image:

- This **spline** is **composed by** 2 **bezier curves**
  - **Curve 1** is composed by **points 1, 2, 3 and 4**
  - **Curve 2** is composed by **points 4, 5, 6 and 7**
- Points **1, 4 and 7 are control points** that defines the beginning and the end of each curve (Green points).
- The first control point is bigger than the others to mark the beginning of the spline
- Points **2, 3, 5 and 6 are handles** that defines the shape of each curve
- Each **handle is attached to** its corresponding **control point by a red line**
  - Point 2 is a handle from control point 1
  - Points 3 e 5 are handles from control point 4

    - Point 6 is a handle from control point 7
- **All control points are** positioned **inside the** generated **curve**

By using splines is possible to create curves of any shape or length. This is very useful not only for game development, but also has a large number of practical applications in the real world. For more information about the mathematics behind splines, take a look on this article.

2.1 - Creating a Spline

To create a spline, Drag & Drop one of the following prefabs to your scene.

- **SimpleSpline:** a simple spline with no other components attached to it
- **SplineMeshRenderer:** ready to use spline prefab for procedural generation
- **SplineSpawner:** ready to use spline prefab for object spawning

These prefabs are located under “Assets/WSM Game Studio/Spline Mesh Renderer/Prefabs”.

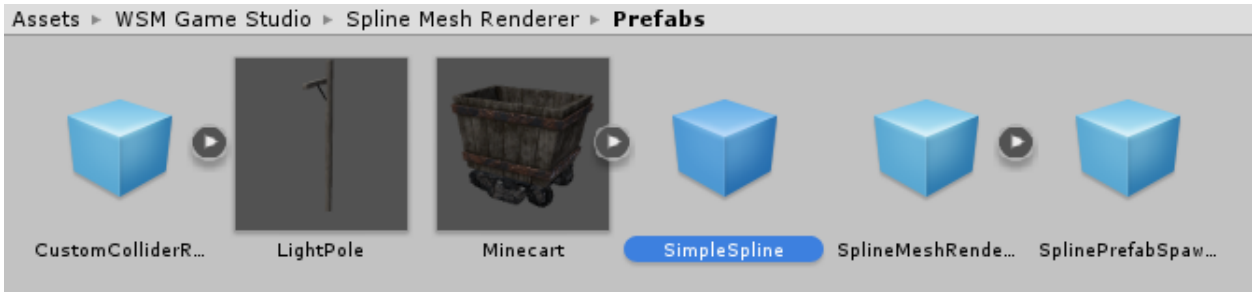

Figure 2 - Spline prefabs

NOTE: Never change these prefabs directly or save alterations made on them. They were created with the sole purpose of being used as a starting point.

Since the Spline is editor extension, it will only be visible when an object that has a Spline component attached to it is selected on the Hierarchy. By default, the newly created spline is composed by only one bezier curve.

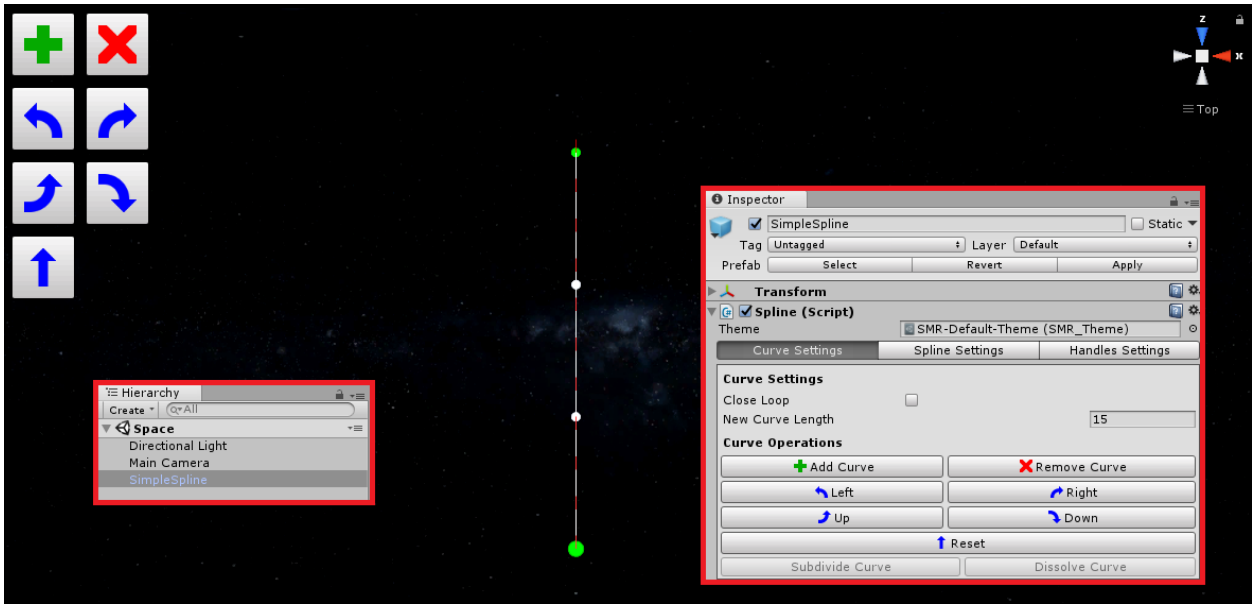

Figure 3 - Newly created spline

2.2 - Editing the Spline

As mentioned on the Spline section, the shape of the curve is defined by the position of the control points and handles.

To change the position of a control point or handle, first you need to select it with a left mouse click.

When a point is selected, its position and rotation handles will appear on the Scene View.

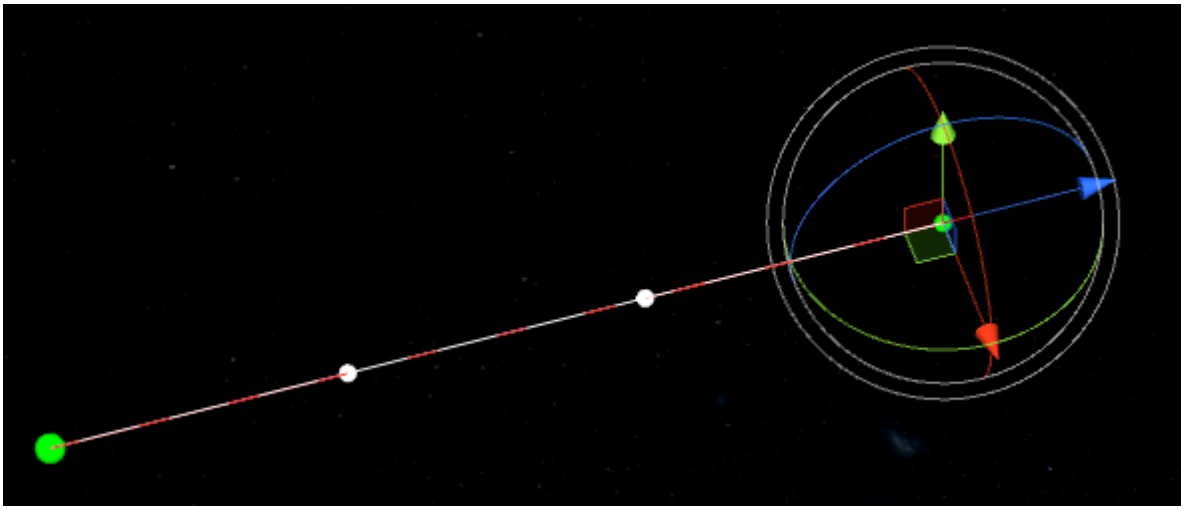

Figure 4 - Position & Rotation handles of the selected point

Use these handles to manually adjust the position and rotation of the selected point. By changing the position of the points, you can create any shape of curve you want.

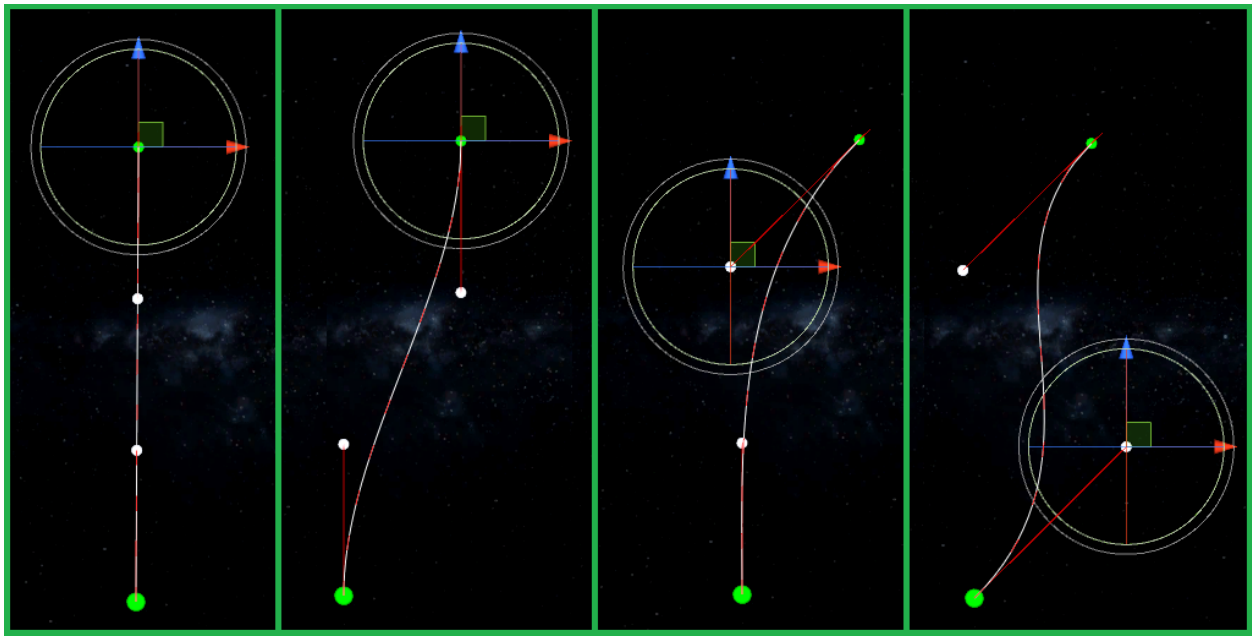

Figure 5 - Manually adjusting the curvature

The selected point position and rotation values can be seen under the Handle Settings tab of the Spline Component.

It is also possible to change these values directly on the Inspector if you wish so, this is very useful if you need mathematical precision for any purposes.

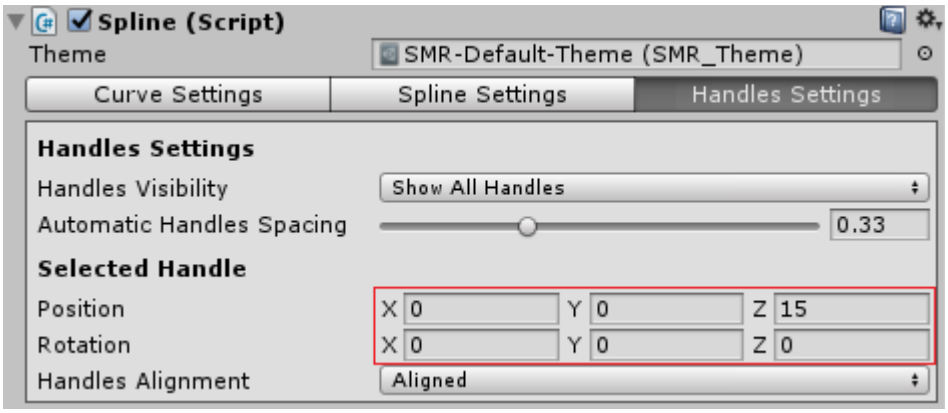

Figure 6 - Position & rotation values on the Inspector

NOTE: The rotation value of a point has different applications for each component:

- Spline Mesh Renderer
  - Applies custom rotation on a mesh segment close to the point
- Spline Follower
  - Applies custom rotation to the object movement when passing through the point
- Spline Prefab Spawner
  - Adjust the spawned object rotation for objects spawned close to the point

Now that you know how to manually adjust the curvature, let's learn how to use the spline component features to create complex splines.

The spline component is segmented into three main categories, each corresponding to which elements of the spline are affected by its settings and operations. These elements are:

- Curves
- Spline
- Handles

2.3 - Curve Settings

All the curve related settings and operations can be found under the "Curve Settings" tab.

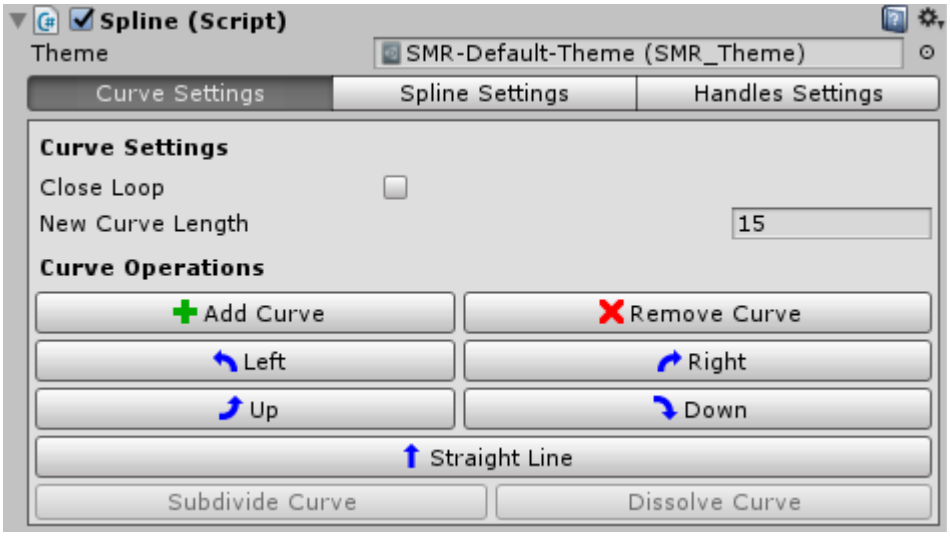

Figure 7 - Curve Settings & Operations

This settings and operations allows you to easily control the length and shape of the bezier curves that compose your spline.

2.3.1 - Close Loop

The “Close Loop” property, defines if the spline consist on an open or closed path. By default, this option is disabled, when enabled, it will automatically adjust the last curve of your spline to create a closed path.

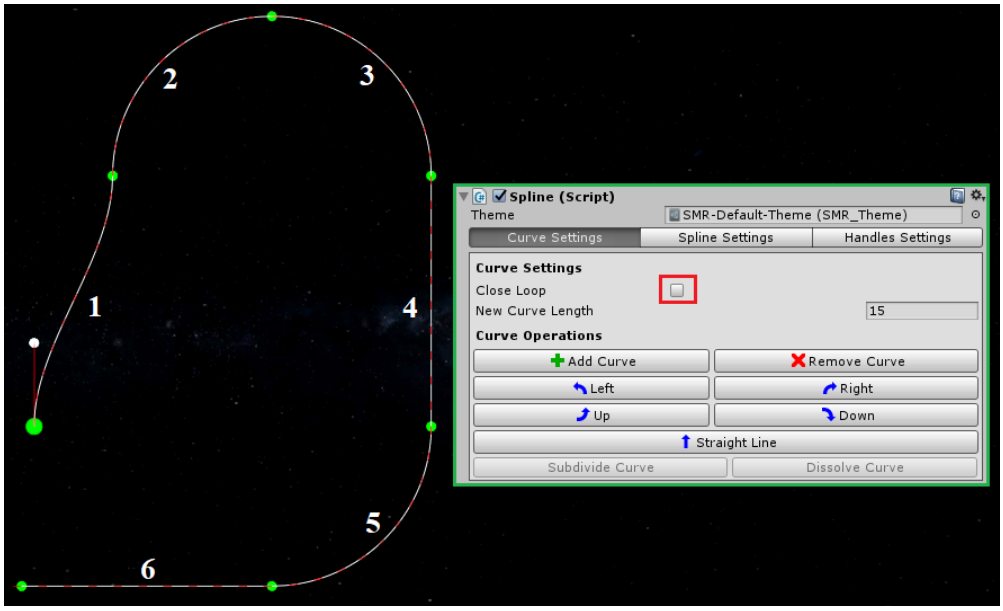

Figure 8 - Open spline path

In the sample image above, you see an open spline path composed by 6 bezier curves. Now, let’s see what happens when the “Close Loop” property is enabled.

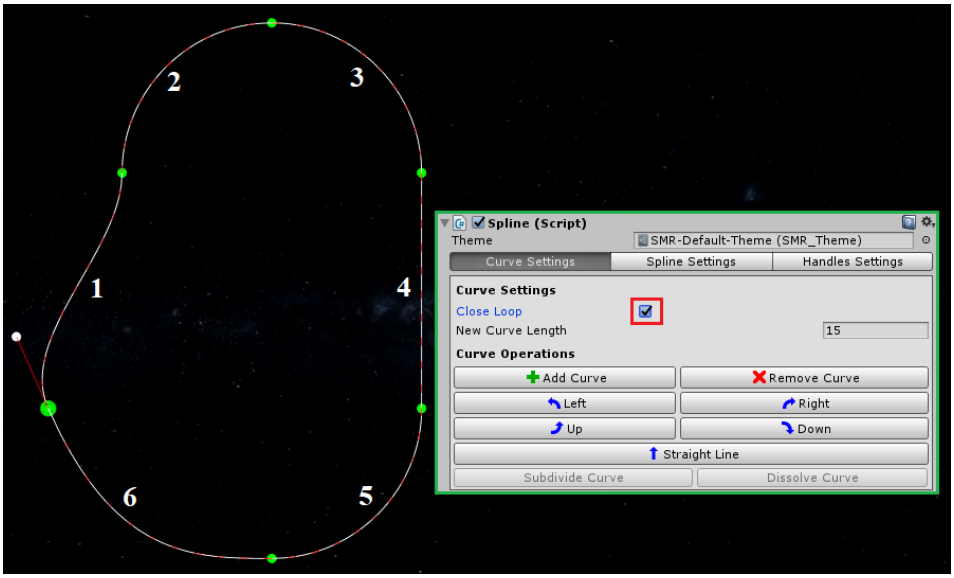

Figure 9 - Closed spline path

As you can see, the last curve (number 6), was adjusted to create a closed spline path.

NOTE: As you can see in the sample above, the first curve was also a slightly adjusted to create a smooth connection. If you don't want this to happen, it is possible to avoid the deformation of the first curve, by setting the first control point Handle Alignment to "Free" (See the Handles section for more details)

2.3.2 - New Curve Length

The "New Curve Length" property, defines the length of newly created curves and curves shaped using any of the predefined curves operations.

2.4 - Curve Operations

There are nine curve operations in total. These operations can be separated in three groups:

- **Curve Creation**
  - Add Curve
  - Subdivide Curve
- **Curve Removal**
  - Remove Curve
  - Dissolve Curve
- **Curve Shaping**
  - Left
  - Right
  - Up
  - Down
  - Straight Line

The curve operations can be found under the "Curve Settings" tab. The most used operations are also available on the top left corner of the Scene View window of the Unity Editor for quick access.

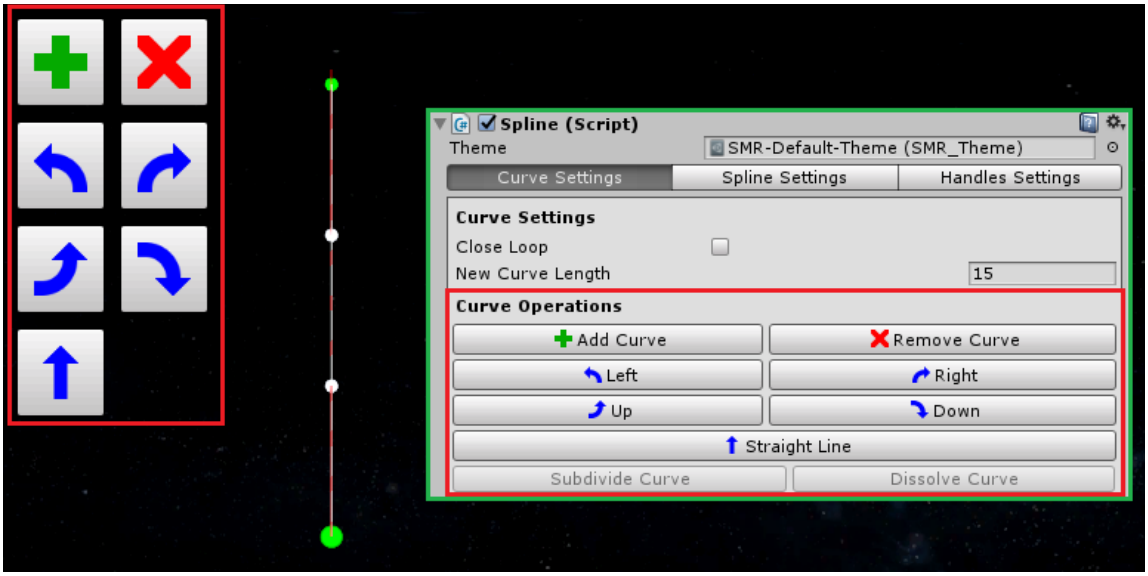

Figure 10 - Curve operations

NOTE: All the quick access operations available on the Scene View are mapped to the corresponding operation on the Unity Inspector by icons. Only quick access operations buttons has icons assigned to them.

2.4.1 - Add Curve

The “Add Curve” button, creates a new curve at the end of the spline, pointing in the same direction of the tip of the last curve. As mentioned before, the length of the newly added curve is defined by the “New Curve Length” property

2.4.2 - Delete Curve

The “Remove Curve” button deletes the last curve of the spline.
This operation only applies for splines composed by 2 or more curves

2.4.3 - Curve Shaping Operations

The “Left” button reshapes the last curve of the spline into a perfect 45º degrees curve pointing to the left.

The “Right” button reshapes the last curve of the spline into a perfect 45º degrees curve pointing to the right.

The “Up” button reshapes the last curve of the spline into a perfect 45º degrees curve pointing upwards.

The “Down” button reshapes the last curve of the spline into a perfect 45º degrees curve pointing to down.

The “Straight Line” operation reshapes the last curve of the spline into a straight line pointing to the direction of the first handle of the last curve. It is very useful to create straight segments pointing at any direction.

2.4.4 - Subdivide Curve

The “Subdivide Curve” operation divides the selected curve in half, creating two distinct curves. To select a curve, just select the first control point of the curve. It is very useful to create new curves on the middle of the spline.

2.4.5 - Dissolve Curve

The “Dissolve Curve” operation dissolve the selected control points, transforming two curves into one. It is very useful to remove curves in the middle of the spline.

Now, let’s take a look at some examples.

2.4.6 - Operation Examples

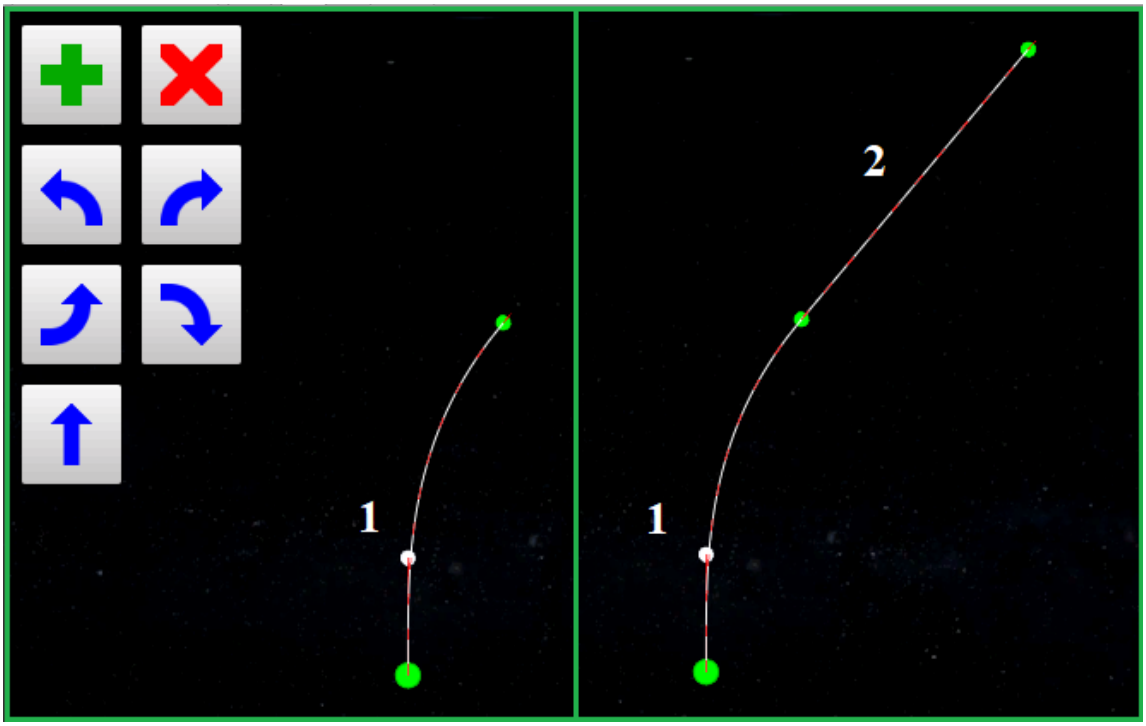

Figure 11 - “Add Curve” operation example

On the left side of the sample image above, you can see the spline before the "Add Curve" operation, on the right side you can see the spline after the operation. Note how the newly added curve (2) follows the direction from the previous curve (1).

By inference, we can affirm that by using the "Remove Curve" operation at this point, would revert the spline to the previous stage by removing the last curve (2).

By using the “Straight Line” operation we can reset the curve shape to a line pointing at any direction. This can be very useful to build ramps for example.

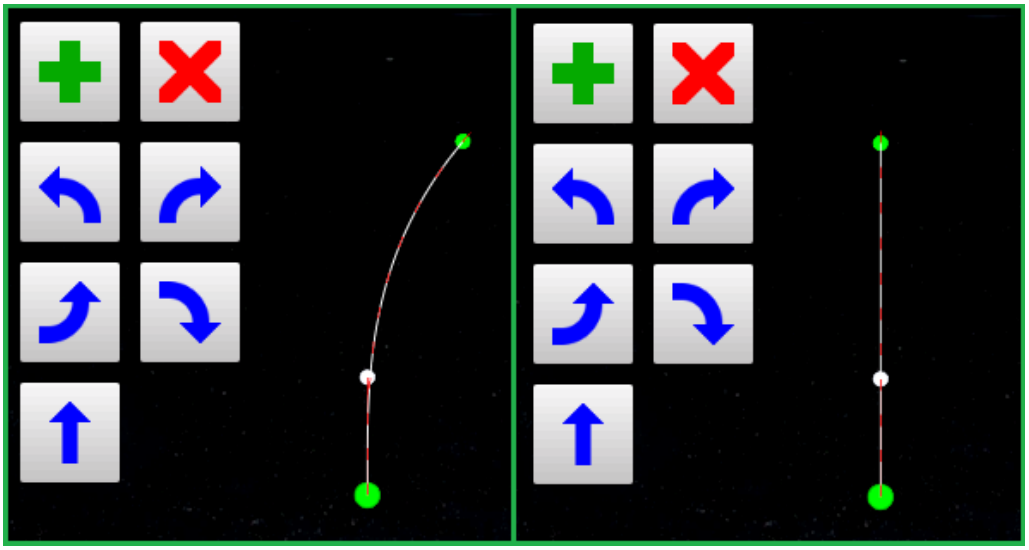

Figure 12 - “Straight Line” operation example

> **NOTE:** All curve shaping operations are affected by the “New Curve Length” property. If the length is set to 15 for example, the curve length will be 15 meters after reshaping.

If you wish to create a perfectly round curve, you can use the predefined curves operations (right, left, up and down). They will convert the last curve on a perfect 45º curve, as can be seen on the sample image below:

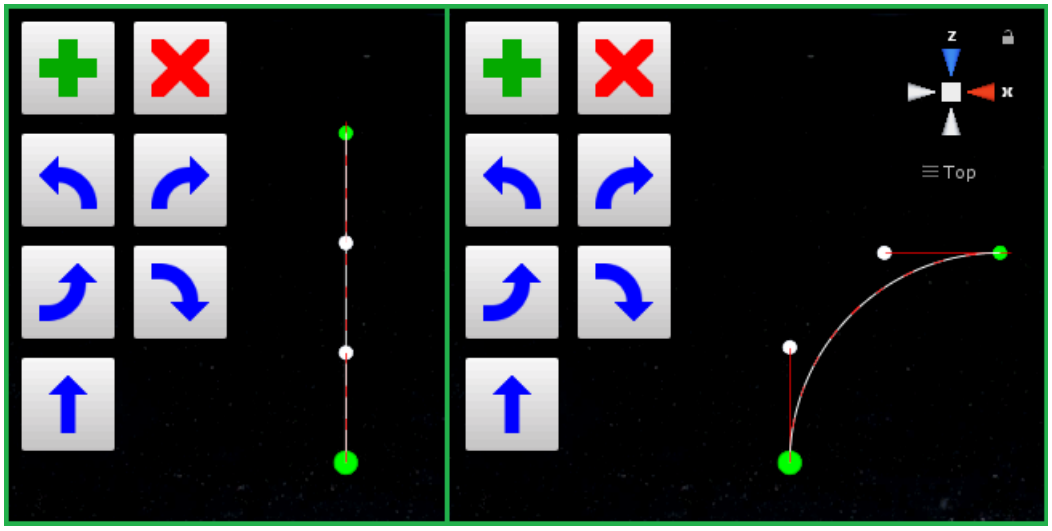

Figure 13 - “Right” curve operation example

The final curvature direction is based on the curve start direction. This allows you to create perfect curves at any direction.

If your camera is not aligned with the curve, at first it can be a little confusing to predict the curvature direction, but everything will make sense once you get used to these operations.

A good technique to understand this concept faster, is to always imagine that your camera is pointing in the same direction of the curve starting point, that way all arrow icons will be pointing to their respective curvature direction.

Alternatively, you can also look at your spline from the top while using the Left and Right operations and look at your spline from the right view while using the Up and Down operations.

For example, if you are looking at your spline from the top view, by combining the “Add Curve” and the “Right” curve shaping operation, you can create a perfect circle, as can be seen on the sample image below.

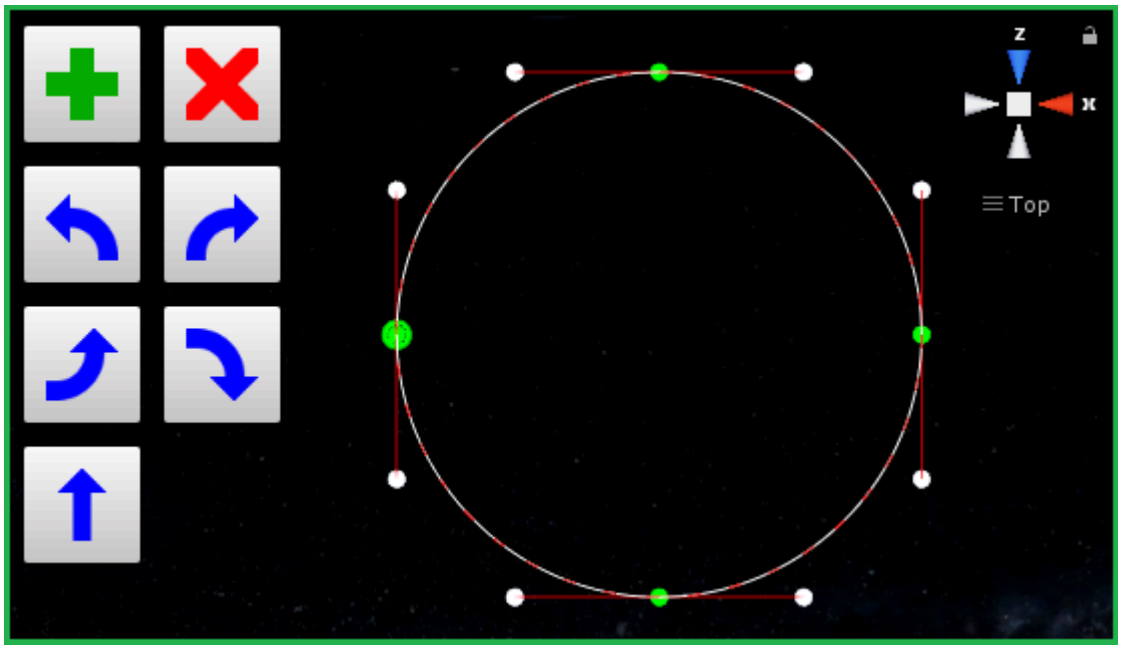

Figure 14 - Perfect circle create using “Add Curve” and “Right” curve operations

Since the predefined curves are 45º, it is very easy to create perfect curves by using this operations.

Now, let’s take a look at the “Subdivide” and “Dissolve” curve operations. Take a look at the sample image below, this spline is composed by three bezier curves. Imagine that you wish to split curve number 2 into two new curves, so you can add more detailed curvature to this segment of your spline.

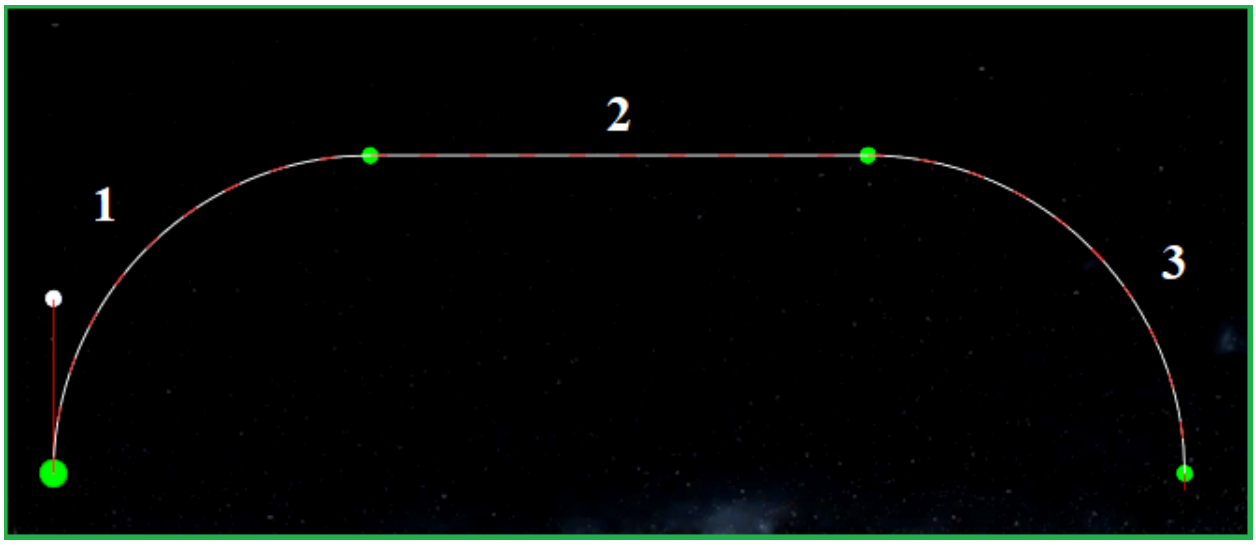

Figure 15 - Before subdividing

This can be achieved, by selecting the control point at the start of curve number 2 and clicking on the “Subdivide” button on the inspector.

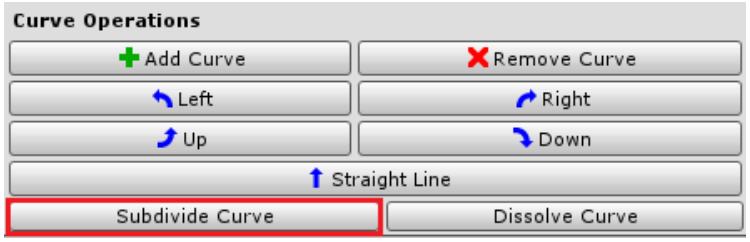

Figure 16 - Subdivide Operation

After the subdivision has taken place, this is how your spline will look like.

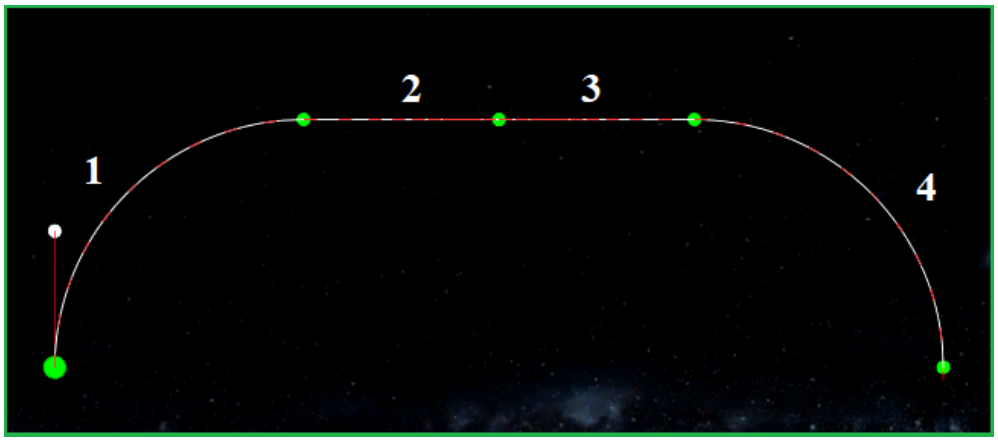

Figure 17 - Subdivided spline

Now that you have created your extra curve, you can adjust its position to create the desired shape.

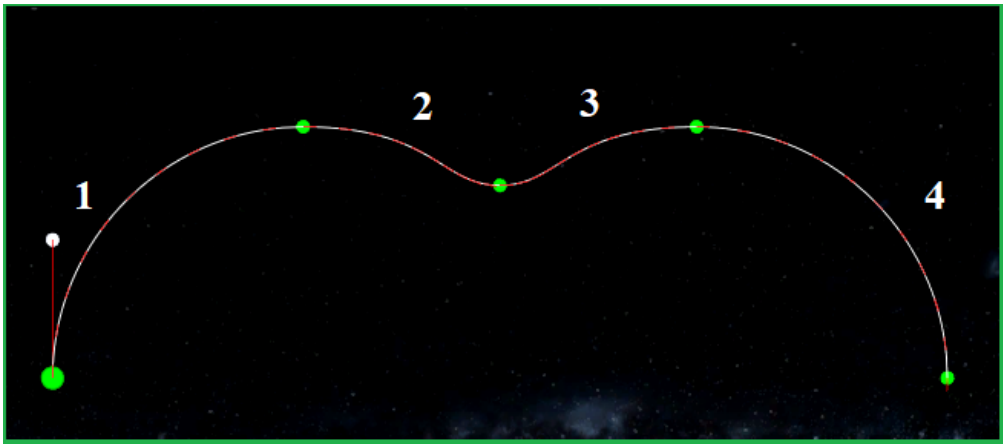

Figure 18 - Practical application of subdivision

If you change your mind and wish to remove the extra curve, you can use the “Dissolve” curve operation to get rid of it. To do that, simple select the control point that you want to dissolve and click on the “Dissolve” Button on the Inspector.

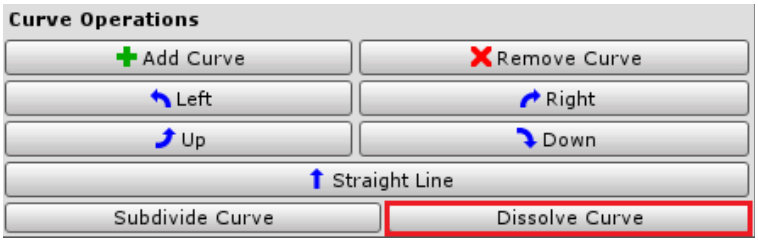

Figure 19 - Dissolve Operation

When a control point is dissolved, it will automatically connect the neighbors control points, merging two curves into a single one.

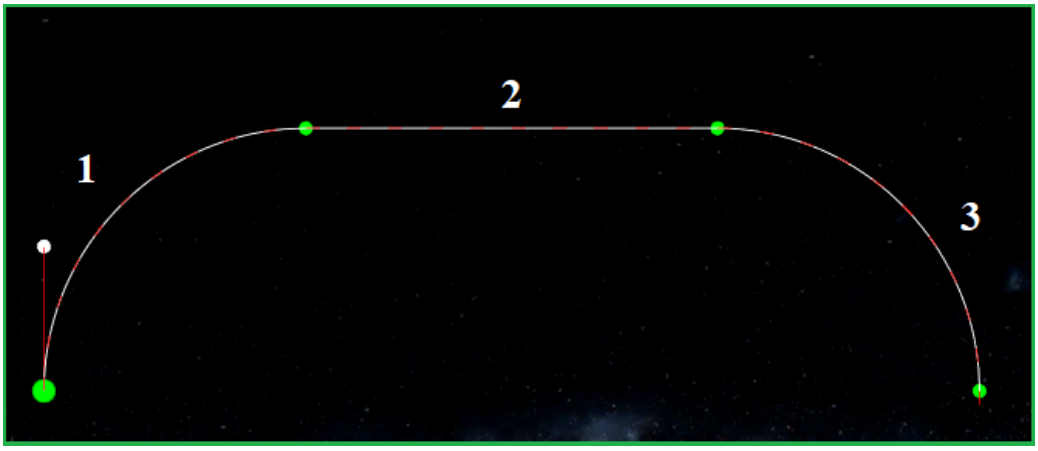

Figure 20 - After dissolving

That covers all the curve related settings and operations. Now, let's take a look at the Spline Settings tab.

2.5 - Spline Settings

All settings and operations that affects the entire spline at once are located under the "Spline Settings" tab.

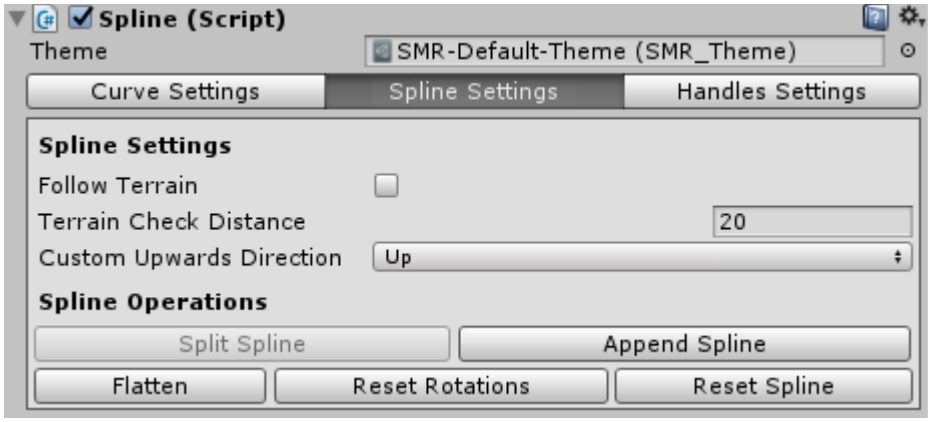

Figure 21 - Spline Settings & Operations

For a better understanding, all practical examples in this section were built using the "Spline Mesh Renderer" prefab (for more information please take a look at the Spline Mesh Renderer section).

2.5.1 - Follow Terrain

The "Follow Terrain" property is used to add terrain elevation support **without** the need of manually adjusting your spline to fit your terrain. It works by creating a virtual projection of the spline on your terrain. The spline shape is preserved as it is, and the virtual projection can be used as a reference by the Spline Mesh Renderer, Spline Prefab Spawner and Spline Follower components.

This technique allows you to use simple flat splines to create complex level design elements that will automatically adjust to any type of terrain elevation.

For example, let's assume you wish to build a circular railroad using the Spline Mesh Renderer.

First, you build a simple flat spline on the desired shape (this can be quickly done by using the Curve Operations), but you realize that a segment of the railroad is passing through the terrain elevations, as shown in the sample image below.

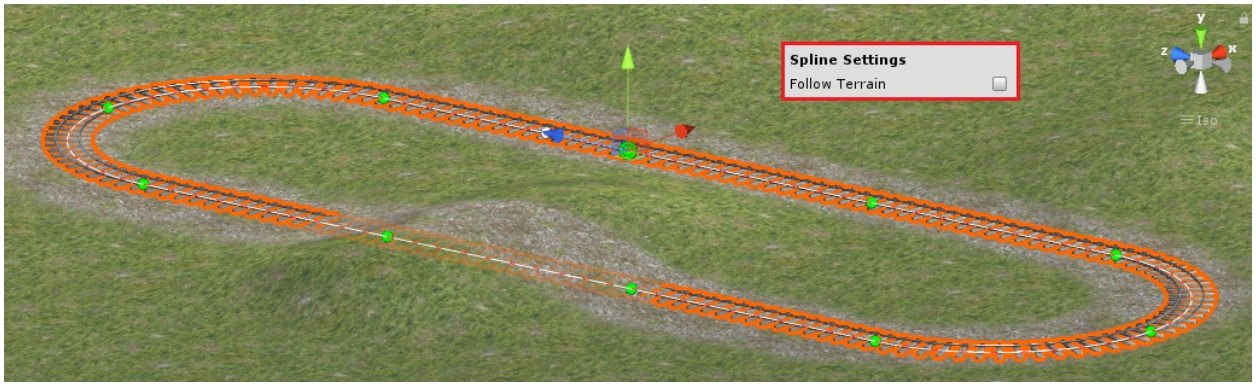

Figure 22 - Disabled Follow Terrain Feature

Since buried rails make no sense, you wish to adjust your railroad to your terrain. This can be easily achieved by enabling the “Follow Terrain” feature.

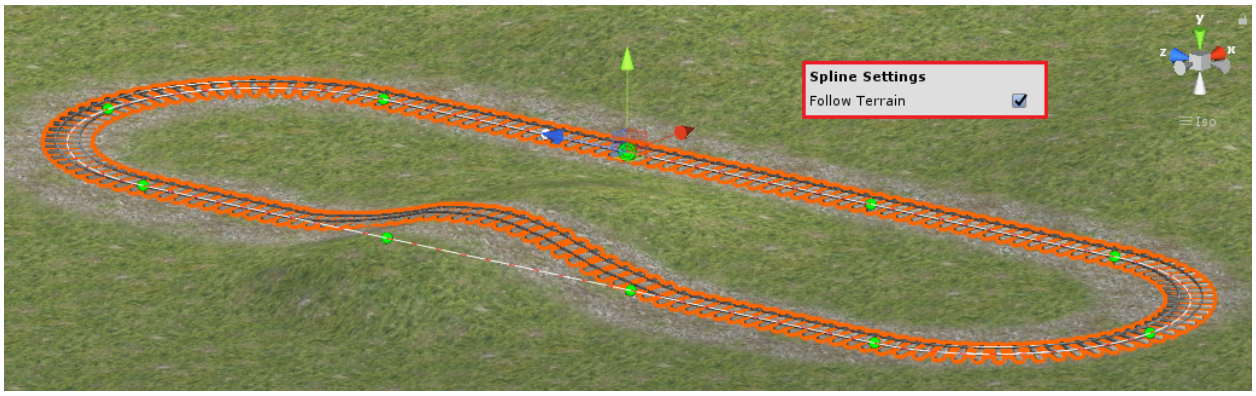

Figure 23 - Enabled Follow Terrain Feature

> **NOTE:** The spline shape remains unchanged, even though the rails mesh is now being projected on the terrain.

2.5.2 - Terrain Check Distance

The Follow Terrain is a very powerful feature, however, when working with very long splines, it can become performance heavy for the Unity Editor, since it needs to check the exact position on space that the spline vertically intersects the terrain for every new segment created.

Since the vertical intersection can be located on an infinite number of spatial points below or above the spline, to avoid performance issues this feature is limited by the "Terrain Check Distance" property.

The “Terrain Check Distance” property defines how far from the spline the “Follow Terrain” feature should look for the terrain intersection.

NOTE: By default the “Terrain Check Distance” is set to 20 meters, which means it will look for the terrain 20 meters downwards and 20 meters upwards, covering a vertical range of 40 meters.

However, if your terrain height range is greater than 40 meters, you may need to adjust this value if, and only if, your spline projection does not reach a part of your terrain.

Alternatively, you can also roughly adjust the spline to bring it closer to your terrain until it reaches the check range. This alternative is performance friendly and can be done quite quickly.

2.5.3 - Custom Upwards Direction

The “Custom Upwards Direction” property is used to easily change the default rotation of the entire spline, without losing the ability to apply custom local rotation on each control point.

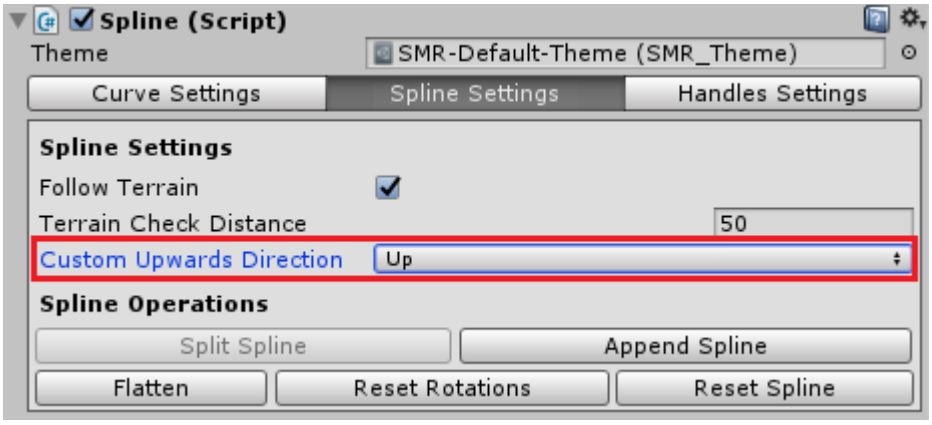

Figure 24 - Custom upwards direction property

By default it is set to “Up”, which is equivalent to the world up direction. By changing this property is possible to create some interesting effects.

For example on the demo included on this package, this property was used to tell the Spline Mesh Renderer how to correctly render the rails upside down on the top segment of the railroad loop. It also made sure that the Spline Follower attached to the minecarts correctly handles the upside down rotation while moving through the loop.

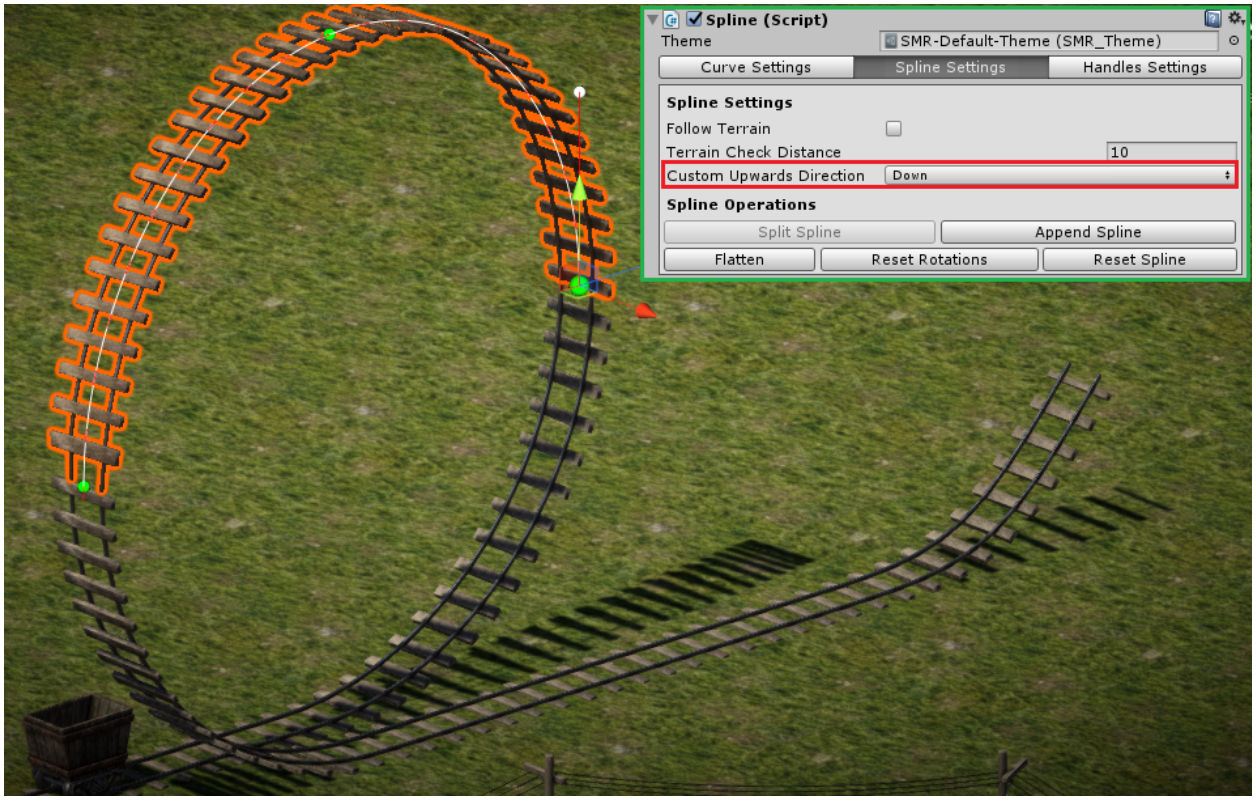

Figure 25 - Custom upwards direction practical example

2.6 - Spline Operations

2.6.1 - Split Spline

The “Split Spline” operation is used to cut one spline into two splines. This operation instantiates a new object on your scene that inherits all the characteristics and components from the original object. The newly created spline starts at the cutting point, connected at the new end point of the previous spline.

This operation is very useful for removing segments of a spline, or reducing the size of long objects to apply Occlusion Culling for example.

2.6.2 - Append Spline

The “Append Spline” operation creates a new spline at the end of the current selected one. The newly created spline inherits all the characteristics and components from the original object. However, the new spline is generated with only one straight curve segment.

This operation is useful to create complex paths composed by multiple splines.

2.6.3 - Flatten

The “Flatten” operation set all control points and handles heights at the same level of the first control point of the spline, creating a horizontal flat spline.

> **NOTE:** The Flatten spline feature is a legacy feature from v1.0 that was originally designed as a quick way to undo the old Follow Terrain feature that was deprecated.
>
> On version 2.0, the Follow Terrain feature was completely re-written for better results. The Flatten operation was kept, since it still can be useful in some situations.

2.6.4 - Reset Rotations

The “Reset Rotations” operation removes the custom rotation value of all control points and handles. It is very useful if you have rotate any points by mistake or wish to remove all custom rotations.

2.6.5 - Reset Spline

The “Reset Spline” operation completely resets the spline to its initial state: only one straight curve segment. This operation is useful if you want to recreate your spline from scratch.

2.7 - Handles Settings

All settings and operations that affects the spline control points and handles are located under the “Handles Settings” tab.

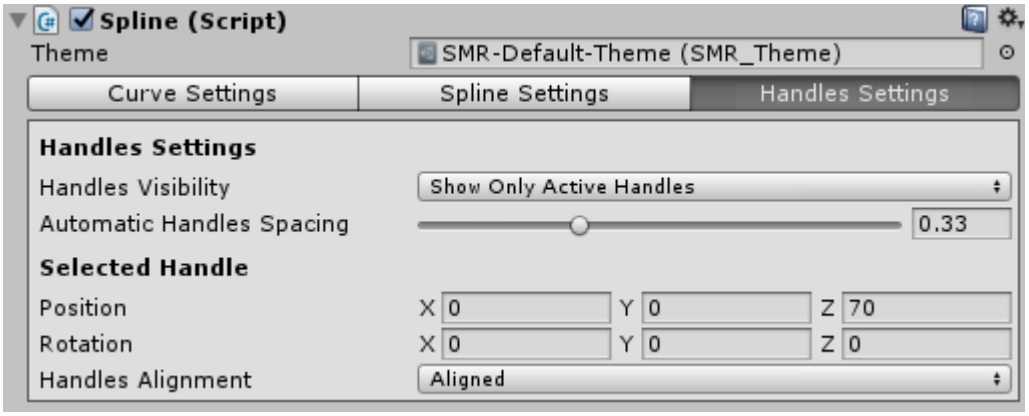

Figure 26 - Handle Settings tab

2.7.1 - Handles Visibility

The “Handles Visibility” property is used to hide/show handles on the Scene. This property is affect only handles, control points are always visible.

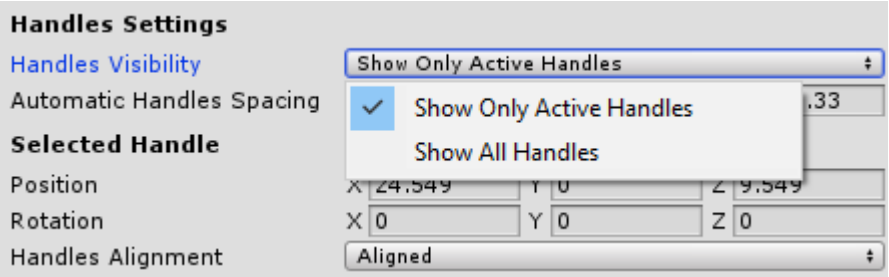

Figure 27 - Handles visibility property

You can either choose between “Show Only Active Handles” or “Show All Handles”. Active Handles are handles connected to the current selected control point.

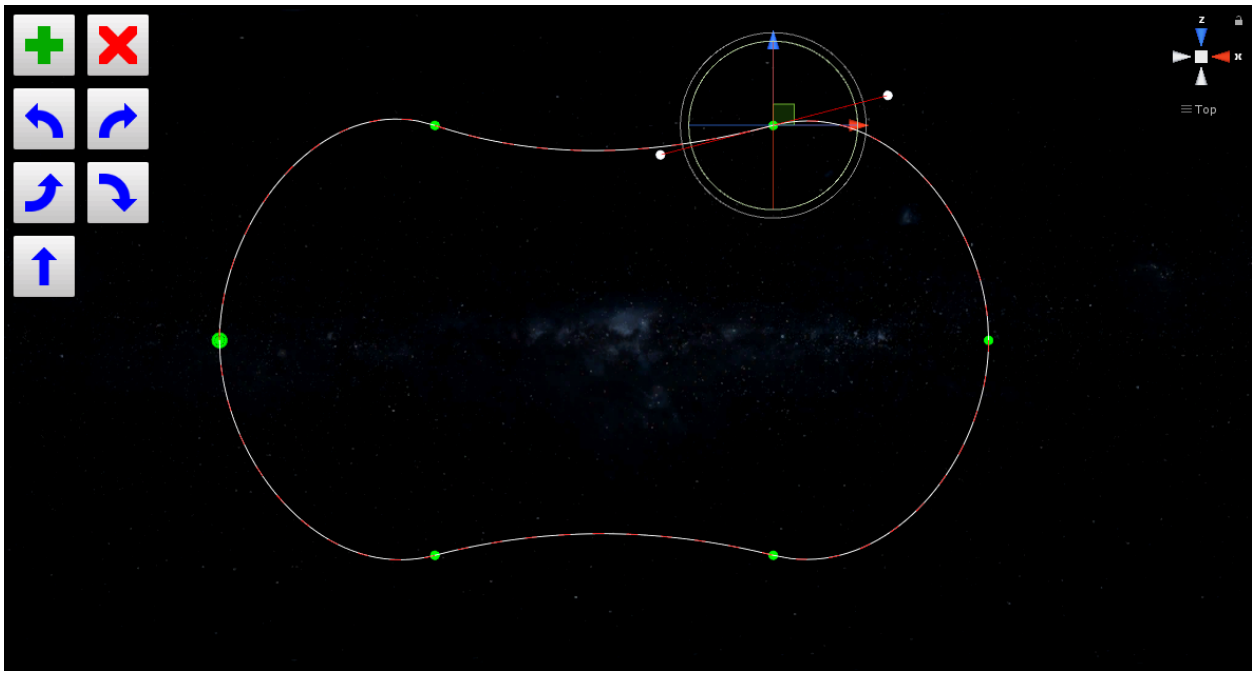

Figure 28 - Handles visibility: Show Only Active Handles

The “Show only Active Handles” is the default option, when it is selected the scene view is much more clean.

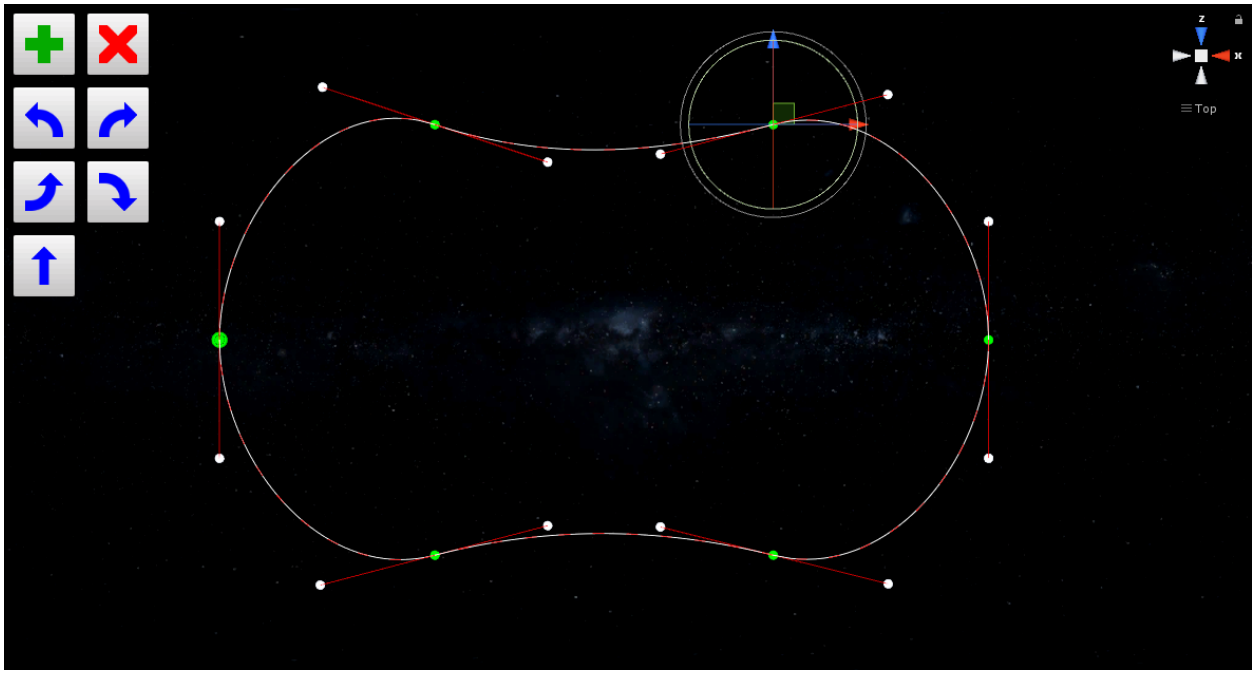

Figure 29 - Handles visibility: Show All Handles

The “Show All Handles” option is useful if you want to adjust the handles, without the need to pre-select their corresponding control point. It is also useful to check the handles alignment along your spline (See the Handles Alignment section for more details).

NOTE: Handles with “Automatic” alignment are not visible on either handles visibility modes. (see Handle Alignment section for more details)

2.7.2 - Selected Handle

The “Selected Handle” section shows the selected control point or handle properties.

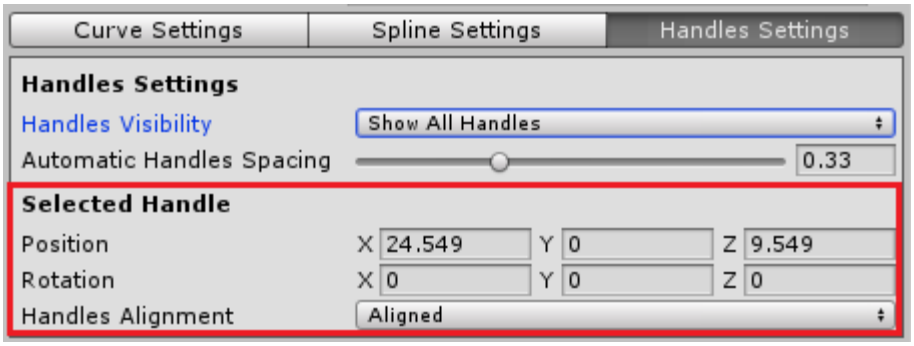

Figure 30 - Selected Handle properties

In this section you can verify and manually adjust the position, rotation and alignment of any point.

As mentioned before, the position is responsible for defining the shape of the spline, the rotation is used to customize the behaviour of other components. For example, The Spline Mesh Renderer use the rotation value to add custom bend to the mesh area affected by the point.

2.7.3 - Handles Alignment

The “Handle Alignment” property defines how the bezier curves connections are shaped and aligned.

There are four alignment modes, which mode is represented by a different color and have different effects on the curve shape.

- **Aligned**
  - Represented by white circles
  - Positioned on opposite sides of the control point
  - Distance between control point and both handles can be different
  - Can be manually adjusted
- **Mirrored**
  - Represented by yellow circles
  - Positioned on opposite sides of the control point
  - Distance between control point and both handles is always equal
  - Can be manually adjusted
- **Free**
  - Represented by blue circles
  - Can be positioned anywhere
  - Distance between control point and both handles can be different
  - Can be manually adjusted
- **Automatic**
  - Automatic handles are not visible
  - Cannot be manually adjusted
  - Automatically adjusted accordingly to the control point position

NOTE: As mentioned before at the Spline section, control points are represented on the Scene View as green circles. They mark the start and end of each bezier curve.

Handles are represented by circles attached to the control points by red lines.

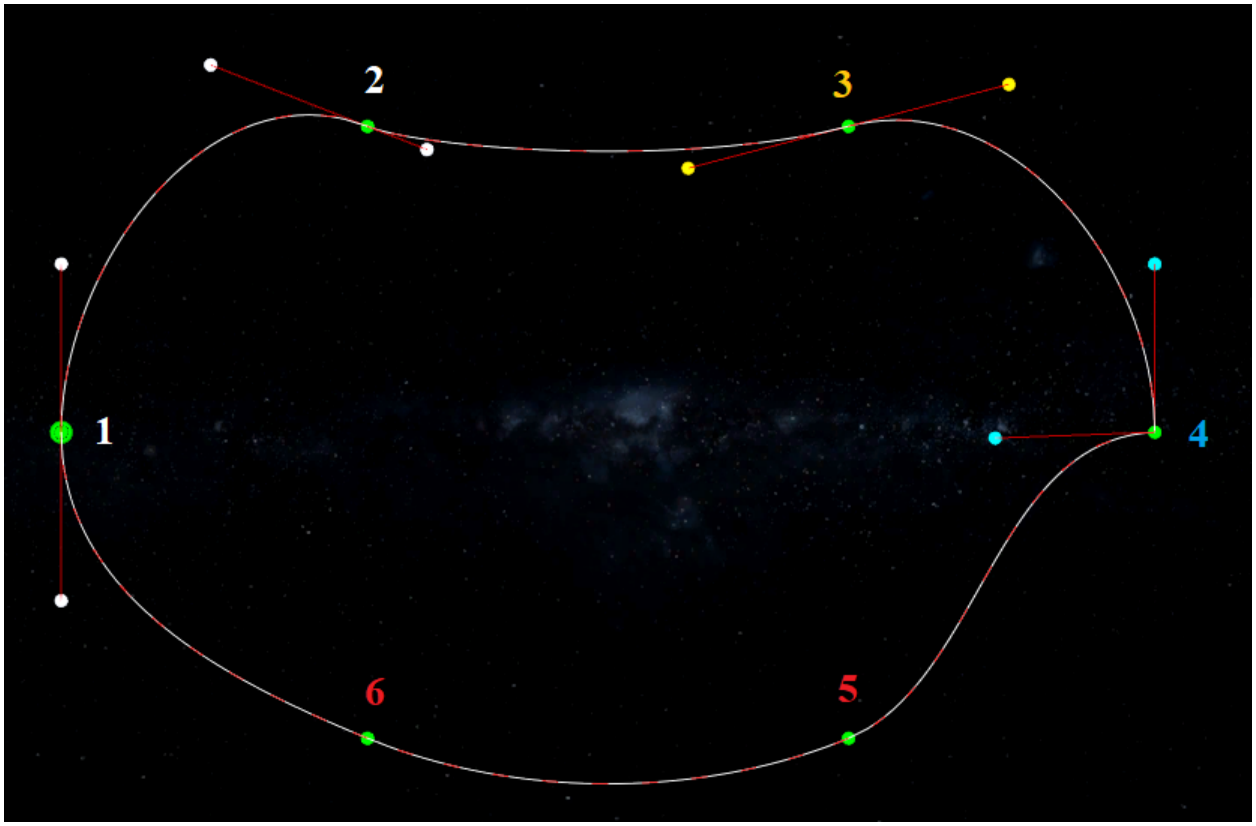

Figure 31 - Handle Alignments

Take a look at the sample image above, control points 1 and 2 handles are set to “Align”, control point 3 handles are set to “Mirrored”, control point 4 handles are set to “Free”, control points 5 and 6 handles are set to “Automatic”.

NOTE: When creating new curves, the newly created curve control point will inherit the handle alignment from the last control point. This is very useful if you want to build a spline that has the same alignment on all control points.

Align and Mirrored handles are good to create smooth curves by manually adjusting the handles positions.

Free handles can be used to create sharp curves, like can be seen in the sample image above.

Automatic handles automatically adjust themselves to create smooth curves based on the control points positions. They are useful if you want to adjust the entire spline shape at once. Automatic curves are adjusted by changing the “Automatic Handles Spacing” property.

2.7.4 - Automatic Handles Spacing

The “Automatic Handles Spacing” property affects the shape of the spline around control points that are set to the automatic handle alignment mode.

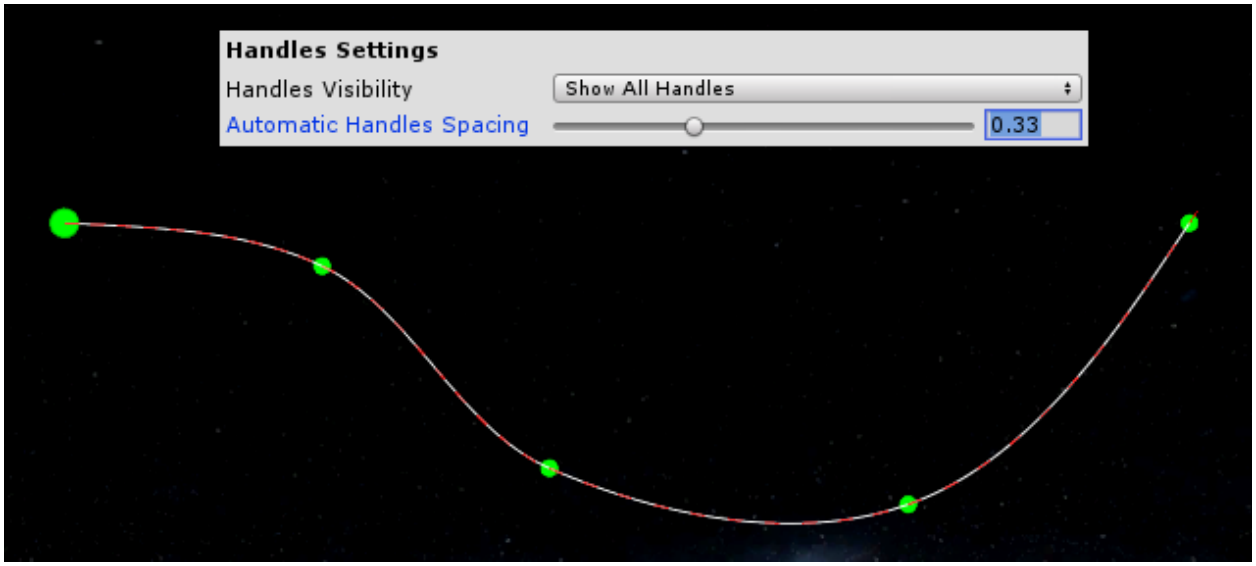

Figure 32 - Automatic Handles Spacing Sample 1

By default, it set to 0.33, by using this value the shape of the curves will automatically be adjusted to create smooth curves, based on the position and distance between the control points.

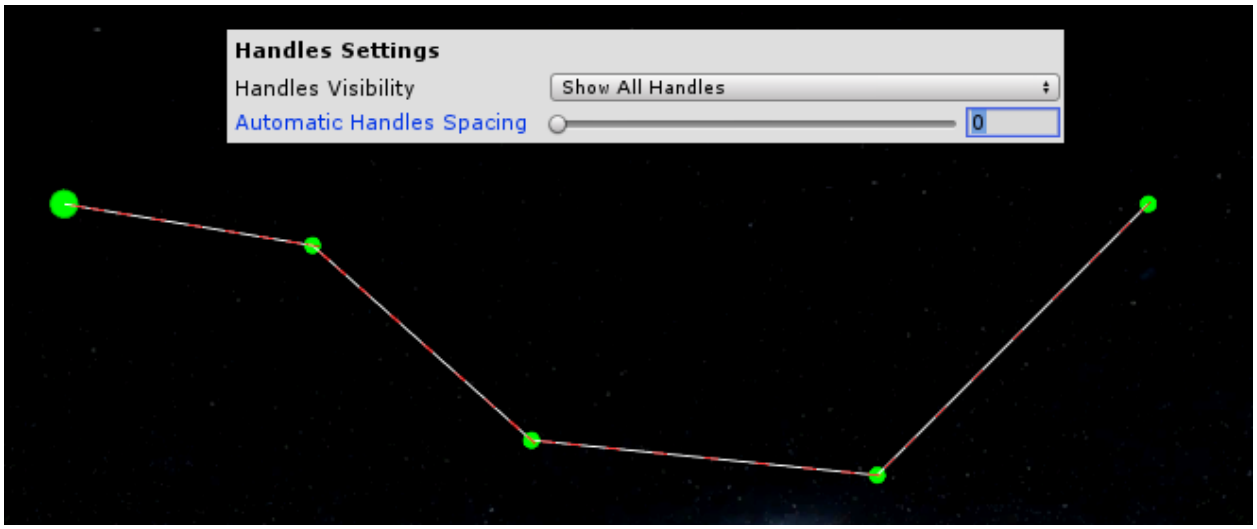

Figure 33 - Automatic Handles Spacing Sample 2

By reducing the spacing value, the spline shape will become less curved. A spacing value of zero will result on straight segments.

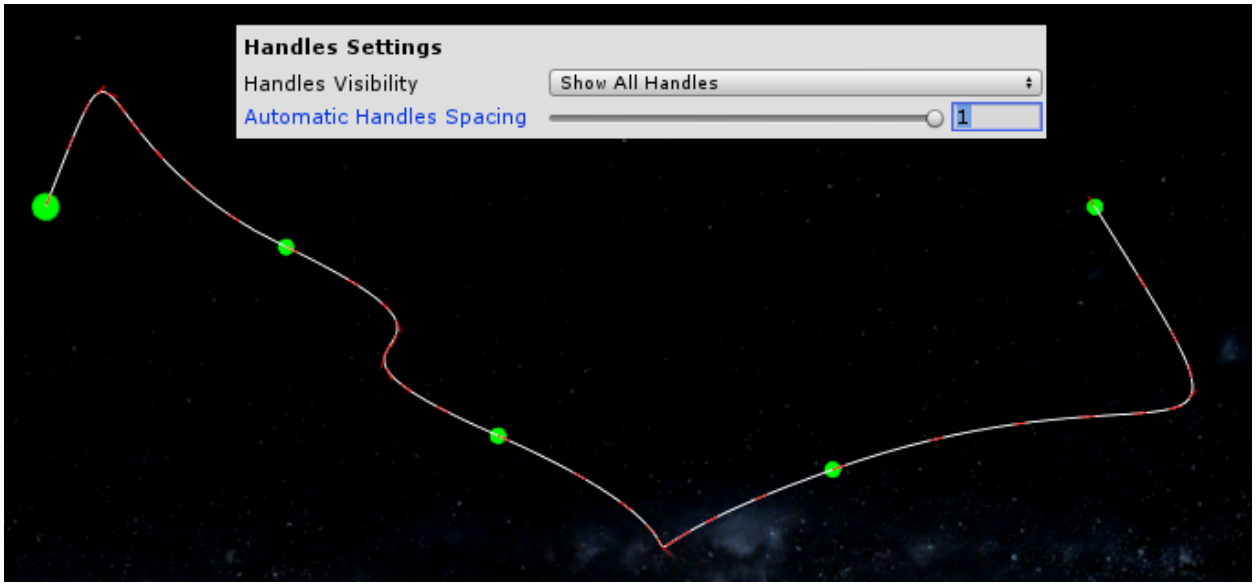

Figure 34 - Automatic Handles Spacing Sample 3

By increasing the spacing value, the spline shape will become distorted. A spacing value of one will result in completely distorted segments.

3 - Spline Mesh Renderer

The “Spline Mesh Renderer” component is a level design tool that allows you to create dynamically generated meshes that can be adjusted on the Unity Editor to fit your scene needs.

It works by extruding a base mesh segment along a Bezier Spline. This process allows the procedural generation of long and curved meshes such as roads, railroads, power lines, pipes, etc…

Therefore, all you need to create your dynamic procedural mesh is:

- Base Mesh Segment
- Base Mesh Segment Collider (optional)
- Base Mesh Materials (optional)
- Create your Bezier Spline using the Spline component

3.1 - Creating a Spline Mesh Renderer

To create a Spline Mesh Renderer, Drag & Drop the “SplineMeshRenderer” prefab to your scene.

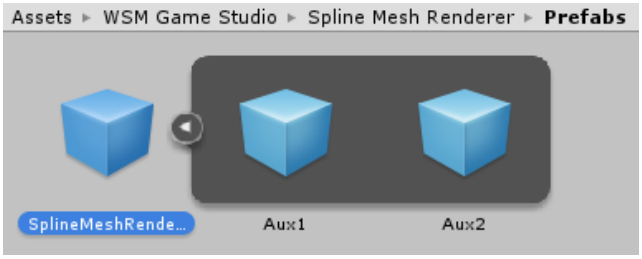

Figure 35 - Spline Mesh Renderer prefab

The “SplineMeshRenderer” prefab is all you need to start creating your mesh. It can be found at “WSM Game Studio/Spline Mesh Renderer/Prefabs”.

NOTE: Never change this prefab directly or save alterations made to it. It was created with the sole purpose of being used as a starting point.

The “SplineMeshRenderer” prefab is composed by the following components

- Mesh Filter
- Mesh Renderer
- Mesh Collider (disabled by default)
- Spline Script
- Spline Mesh Renderer Script
- Child Auxiliar Transforms (Aux1, Aux2)

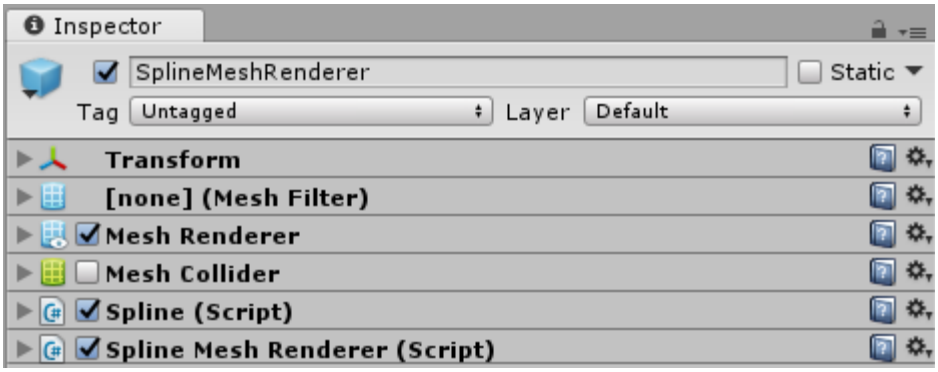

Figure 36 - SplineMeshRenderer prefab components

The mesh filter, mesh renderer and mesh collider are default Unity's components. For More information, please check the official documentation.

- Mesh Filter Official Documentation
- Mesh Renderer Official Documentation
- Mesh Collider Official Documentation

Note that the "SplineMeshRenderer" prefab already has a Spline component attached to it. By default, Spline Mesh Renderer component will use the attached spline component as reference. But, if you wish to use another spline on your scene as a reference, you can replace it on the Spline property.

Figure 37 - Reference Spline

NOTE: The auxiliar transforms (Aux1 and Aux2) are used by the "Spline Mesh Renderer" script to calculate the position and length of each extruded mesh segment along the spline. **DO NOT** rename or delete this objects.

Now that you have a Spline Mesh Renderer on your scene, all you need to create your custom mesh is to feed the Base Mesh property with your base mesh segment model and edit your Spline to shape the object as you see fit.

3.2 - Mesh Generation Settings

All settings and operations that affects the generated mesh are located under the "Mesh Generation Settings" tab.

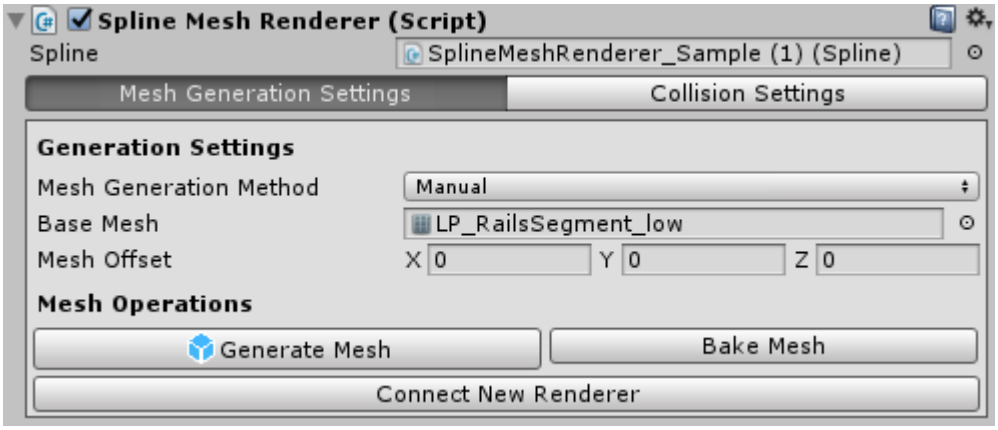

Figure 38 - Mesh Generation Setting tab

3.2.1 - Mesh Generation Method

The “Mesh Generation Method” property defines how often the mesh procedural generation is executed. You can choose between manual and real time generation mesh generation.

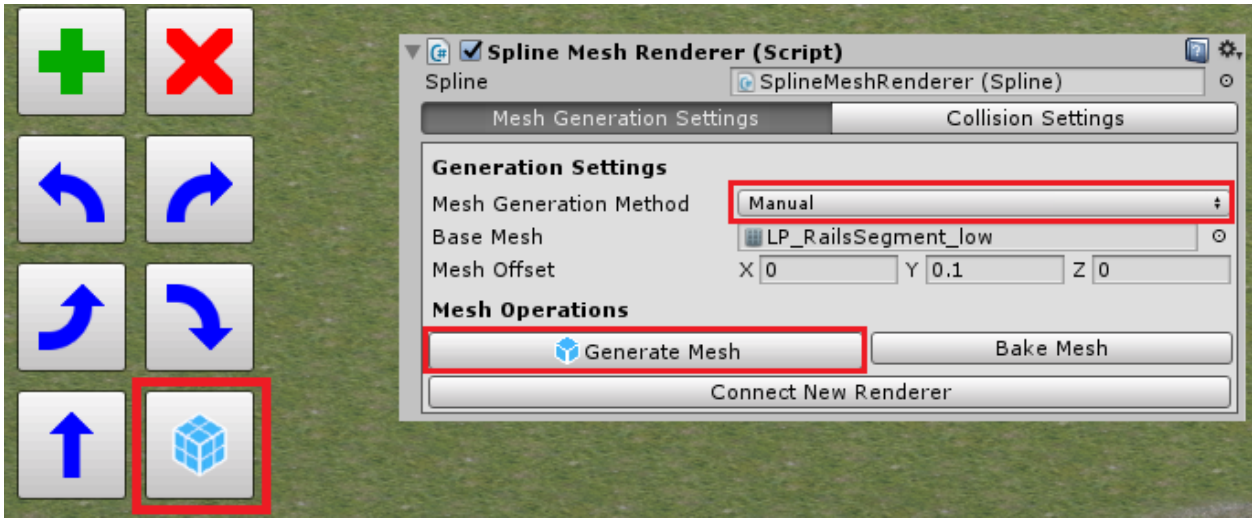

Figure 39 - Manual mesh generation

When the “Manual” generation is selected, every time you change the spline shape, you need to click the Generate Mesh button on the Inspector or on the Quick Access menu at the scene view to update the mesh.

This option is selected by default, since it allows a performance friendly workflow on any computer.

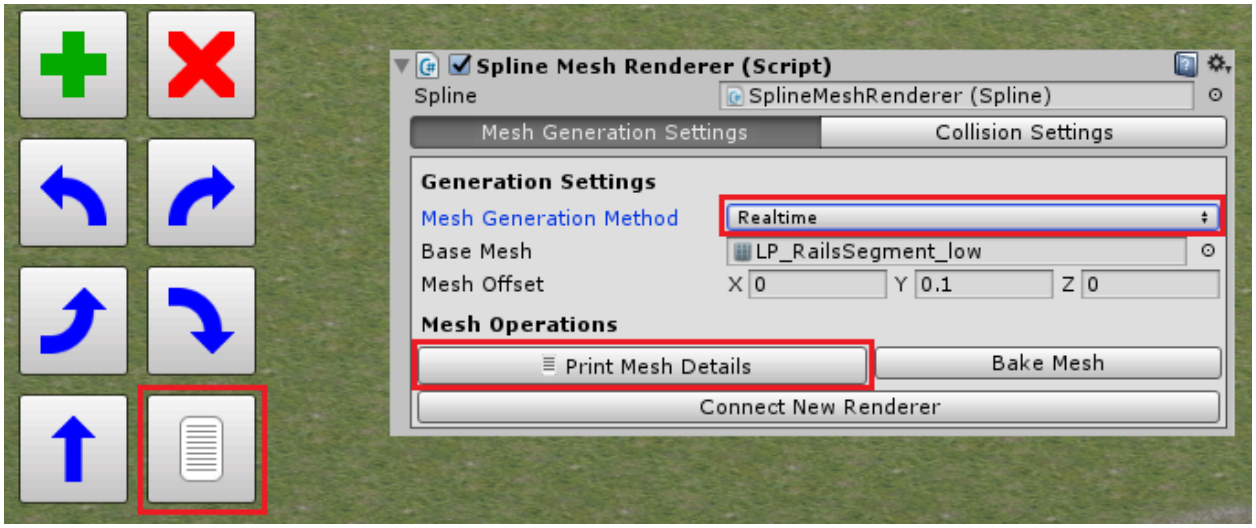

Figure 40 - Realtime mesh generation

When the “Realtime” generation is selected, all changes made to the spline shape are applied instantly to the mesh. This However it can be performance heavy when building long meshes.

When this option is selected, the Generate Mesh button is replaced by the Print Mesh Details button on the Inspector and the Quick Access menu.

The real time generation is recommended for fine tuning the mesh shape when need it or when building short mesh segments. Otherwise, always prefer using the manual generation workflow to avoid performance issues on the Unity Editor.

3.2.2 - Base Mesh

When you select a newly created “SplineMeshRenderer” object in the Hierarchy Window, you will only be able to see its first spline segment.

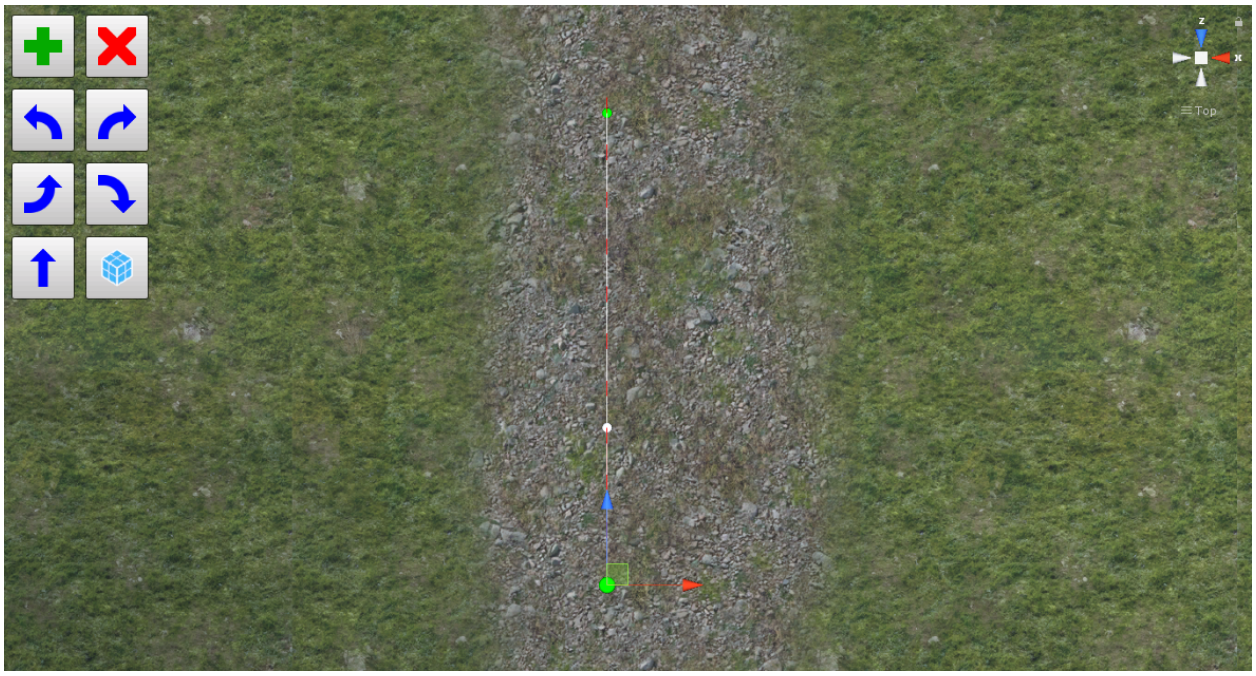

Figure 41 - Newly created Spline Mesh Renderer

That’s because it doesn’t know yet what mesh it should extrude along the spline. Therefore, it doesn’t know yet what kind of mesh you are trying to build.

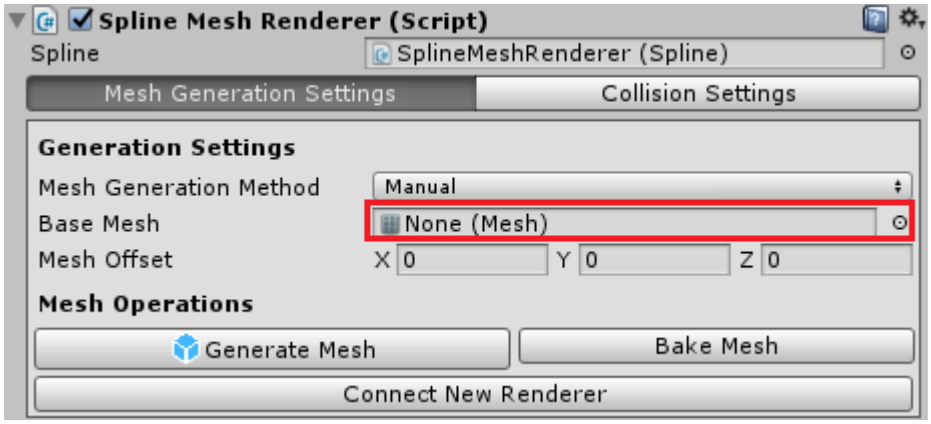

Figure 42 - Empty base mesh property

In this example, we're going to use the Spline Mesh Renderer to create a railroad. To do that, we need to feed this property with a rail segment model.

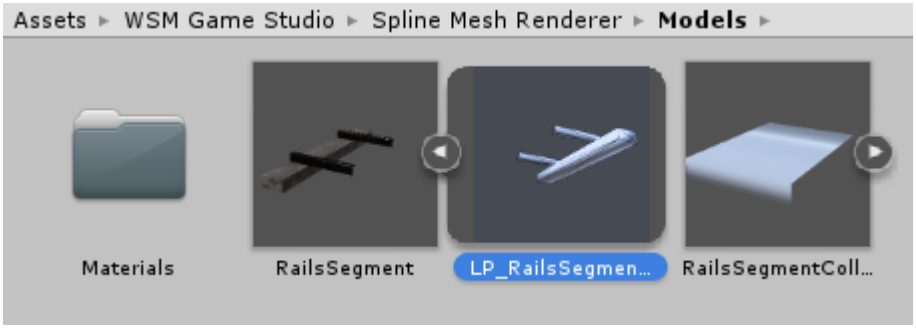

Figure 43 - Rail segment model

Drag & Drop the "LP_RailsSegment" mesh to the "Base Mesh" property of the "Spline Mesh Renderer" script on our scene.

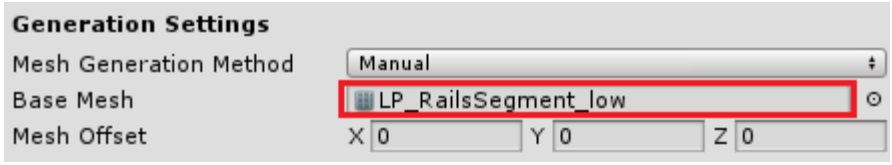

Figure 44 - Base mesh property

Now we can see our mesh, but we still need to add the rails materials.

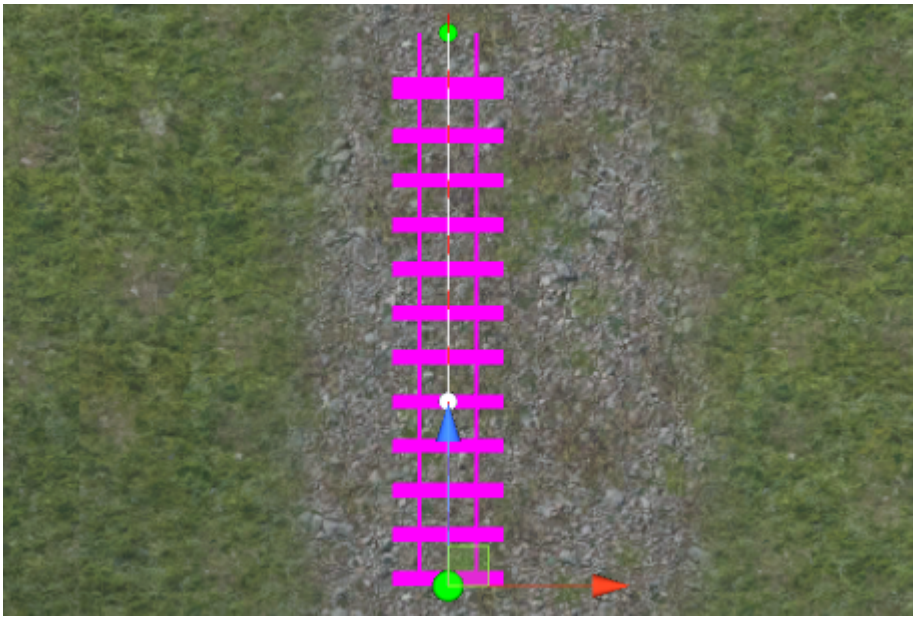

Figure 45 - Spline Mesh Renderer before adding the materials

Drag & drop the “Rails” Material on top of the mesh on the Scene View.

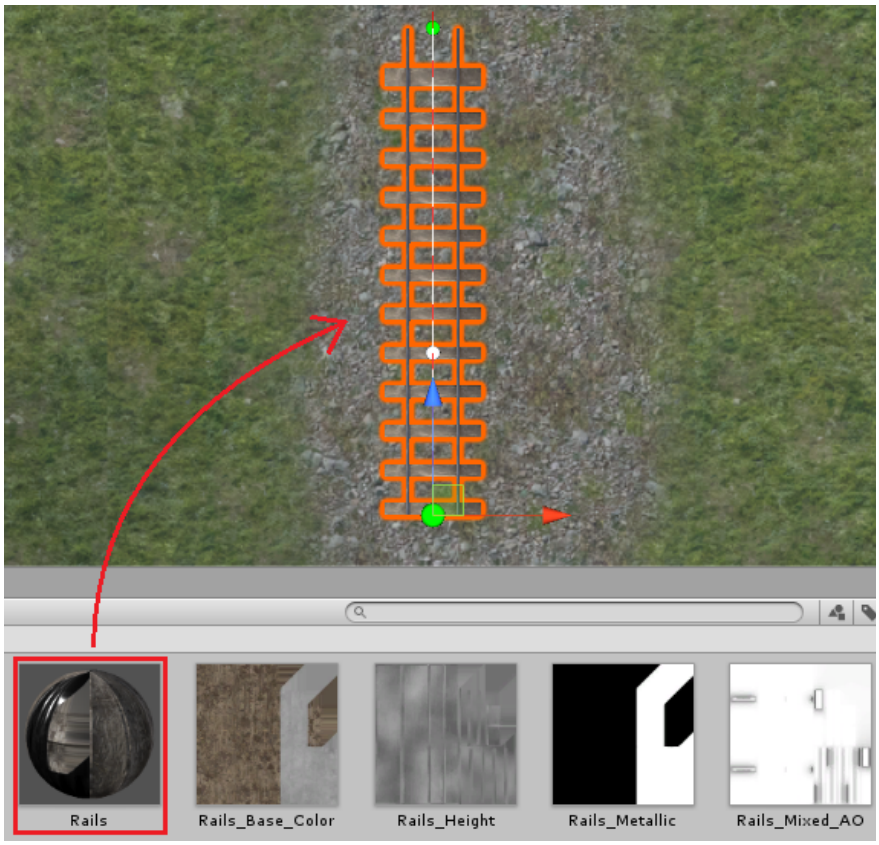

Figure 46 - Adding the materials to the Spline Mesh Renderer

Now, all you need to do to create your railroad is to edit your spline and the Spline Mesh Renderer will render the rails along the spline.

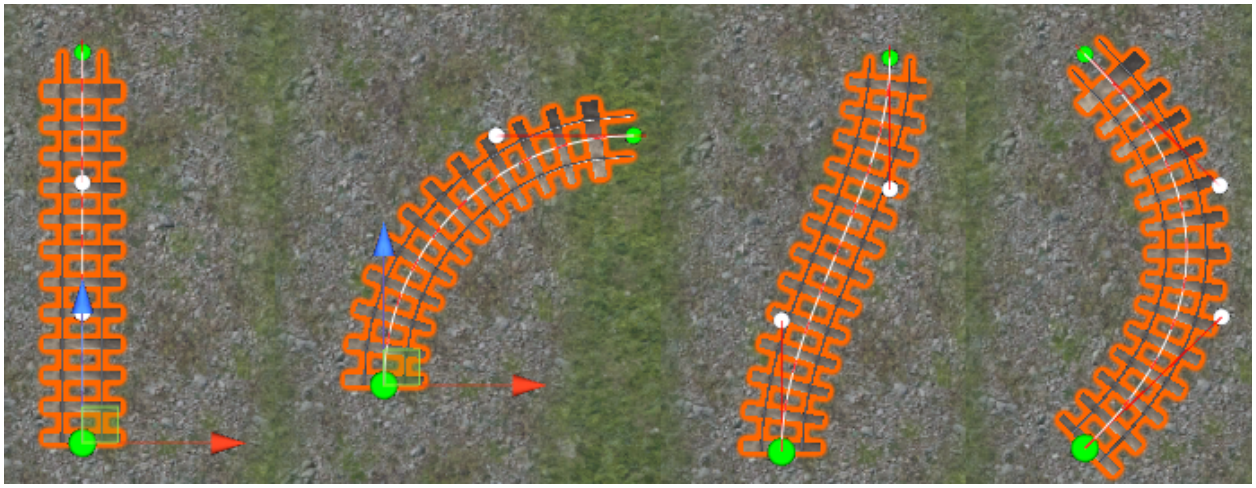

Figure 47 - Curve examples

By adjusting the spline control points and handles rotations, we can also custom local rotation along the generated mesh.

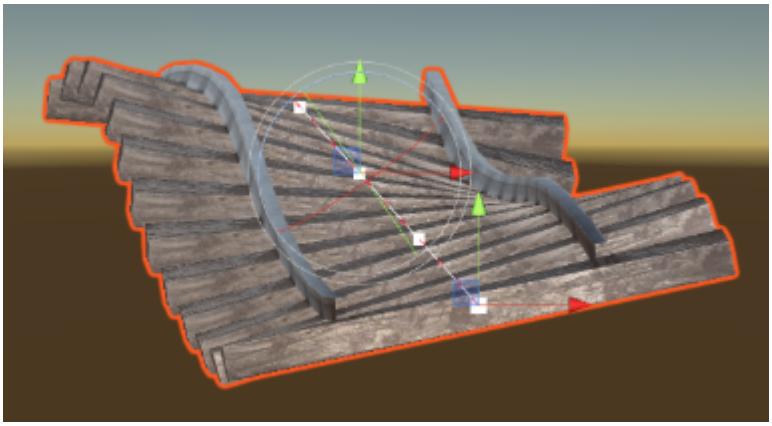

Figure 48 - Custom local rotation

This allows the creation of twisted meshes, as the example below.

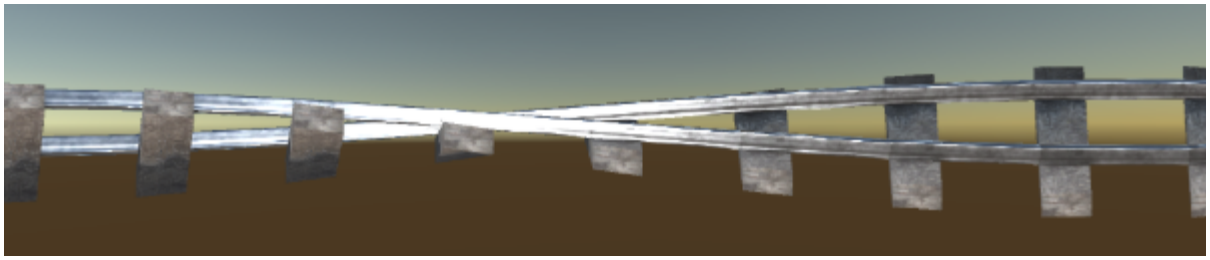

Figure 49 - Twisted mesh sample

Please take a look at the Spline section for more information on how to manipulate the spline shape.

3.2.3 - Mesh Offset

By default, the Spline Mesh Renderer will generate the mesh on top of the spline based on the base mesh model pivot position, but it is also possible to offset the mesh generation by using the “Mesh Offset” property.

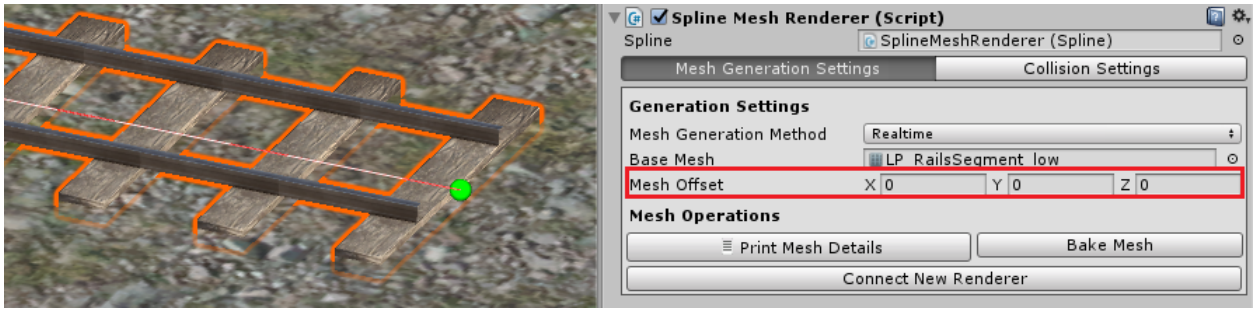

Figure 50 - Mesh offset default value

In the sample, this property was used to slightly raise the mesh to match the terrain elevation. By using this property with the Follow Terrain feature, you can fine tune the mesh position as you wish.

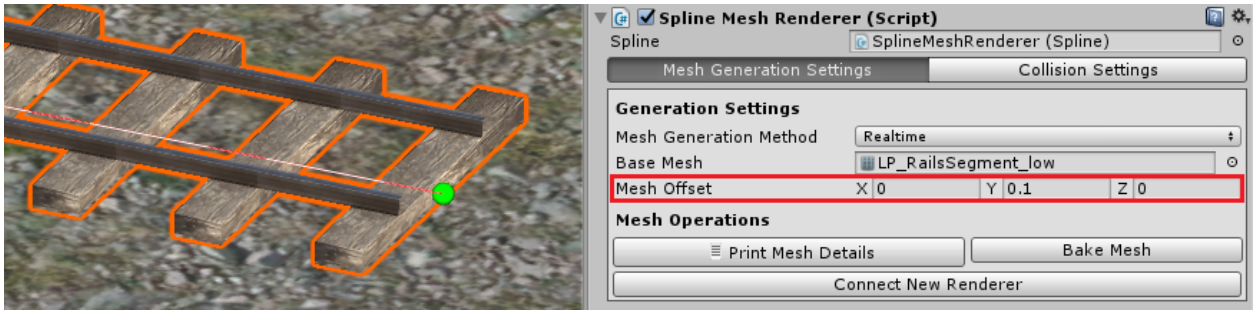

Figure 51 - Mesh offset sample

3.3 - Mesh Operations

3.3.1 - Generate Mesh

The “Generate Mesh” operation is available when the manual Mesh Generation Method is selected. This operation submits the changes made on the spline shape to the generated mesh.

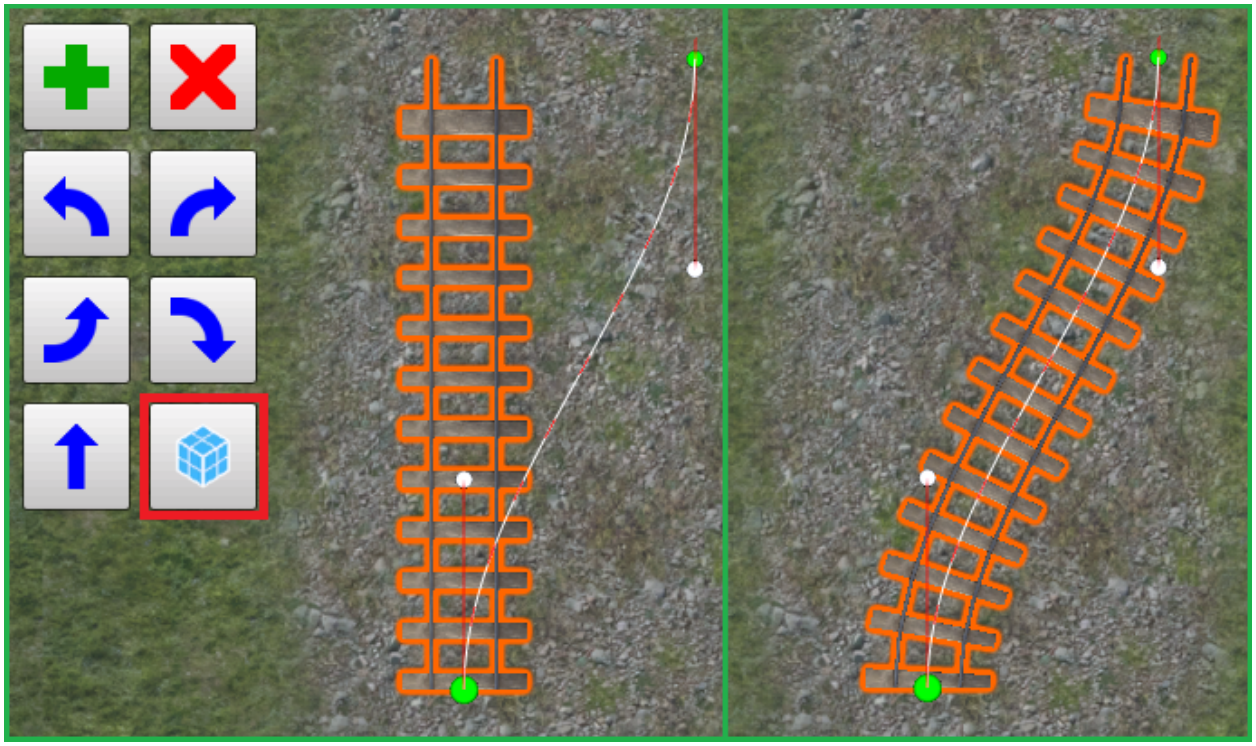

Figure 52 - Before x After “Generate Mesh” operation

In the sample imagem below, you can see the mesh before and after the Generate Mesh operation is performed to submit the changes made to the spline shape to the generated mesh.

The fastest way to used this operation is by clicking on the quick access menu as shown on the sample image above, but it is also available on the Inspector under the Mesh generation Settings tab.

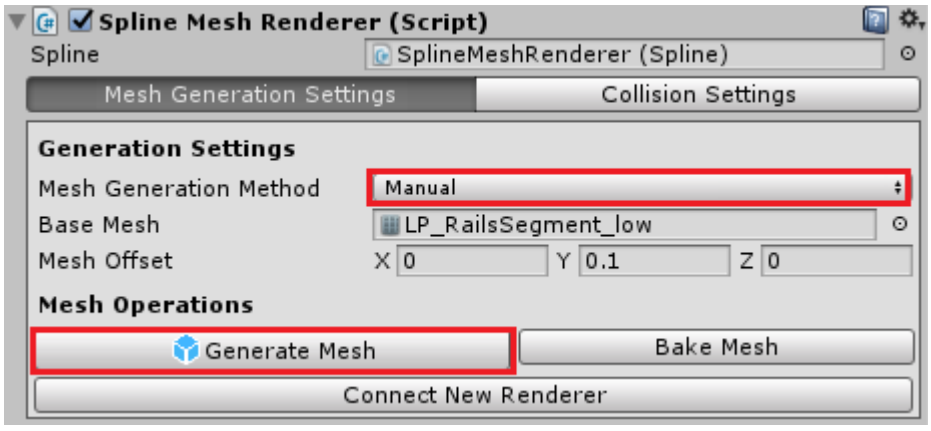

Figure 53 - Generate Mesh on the Inspector

3.3.2 - Print Mesh Details

The “Generate Mesh” operation is available when the real time Mesh Generation Method is selected. This operation prints information about the generated mesh on the Console Window.

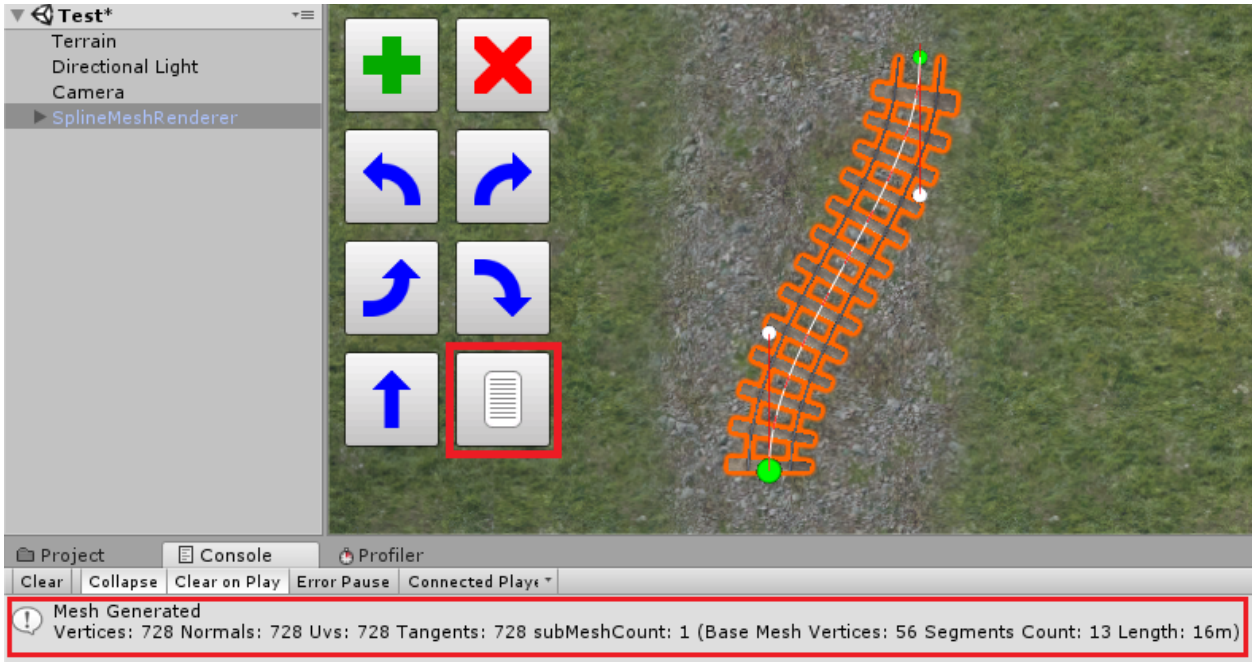

Figure 54 - Mesh details sample

This operation is also available on both the quick access menu and the Inspector.

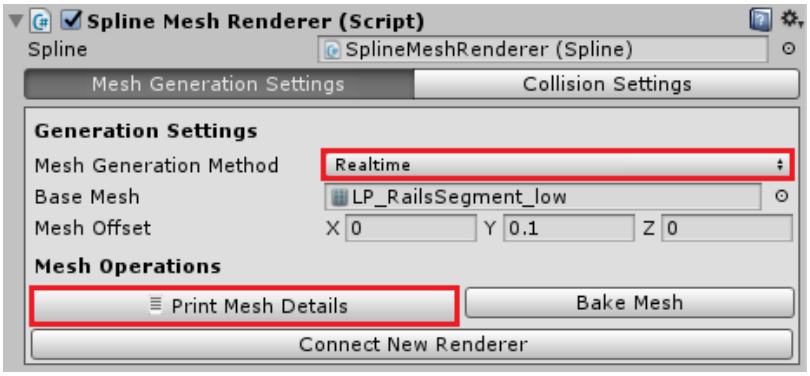

Figure 55 - Print mesh details on the Inspector

NOTE: The mesh details log message is also printed on the console every time the Generate Mesh operation is performed, when the real time generation is selected the log information must be manually printed to avoid too much information being outputted on the console window when editing the mesh.

3.3.3 - Bake Mesh

Then you finish editing your mesh, you bake a static version of it to improve the performance of your game. This feature saves the generated mesh and creates a ready to use prefab that can be used to replace the Spline Mesh Renderer on your scene.

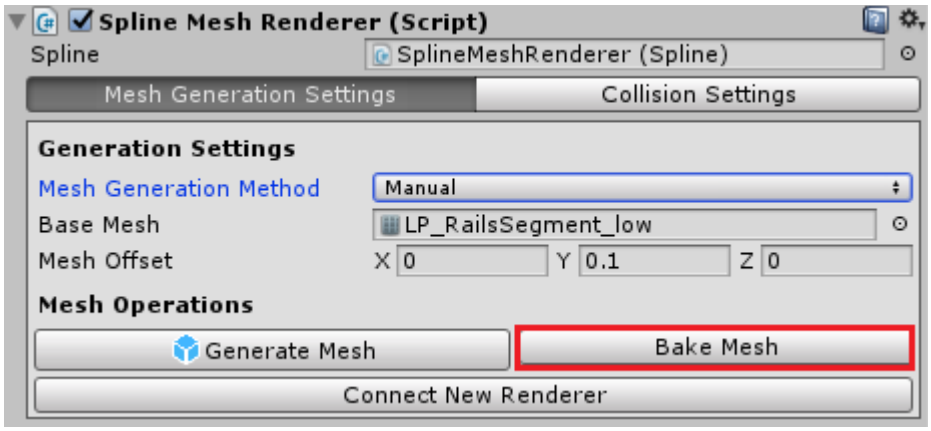

Figure 56 - Bake mesh operation

By clicking on the “Bake Mesh” button the Mesh Baker window will appear.

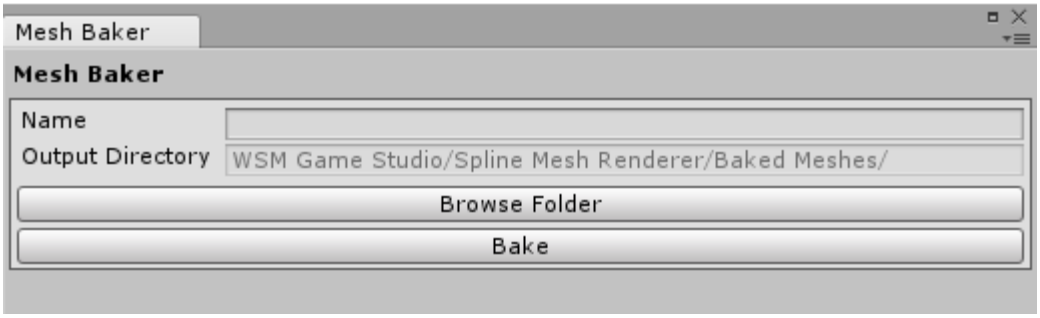

Figure 57 - Bake Mesh window

This window can also be found under the window menu tab “Window > WSM Game Studio > Mesh Baker”.

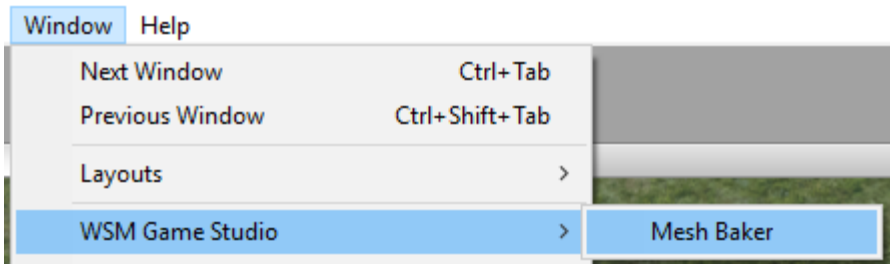

Figure 58 - Mesh Baker menu

By default, the prefabs will be baked into the “Baked Meshes” folder, but you can change the Output Directory by using the “Browse Folder” button.

To start the baking process, all you need to do is select a name for your prefab and click on the “Bake” button.

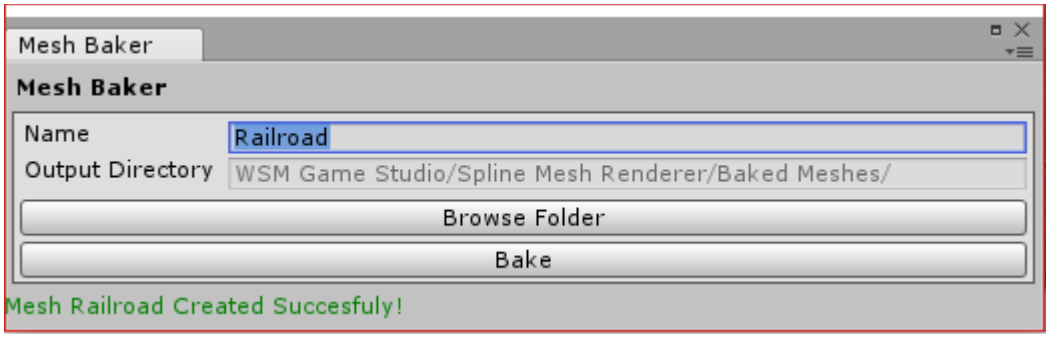

Figure 59 - Mesh baker sample

The “Bake” feature will automatically save the meshes and the prefab on the selected Output Directory.

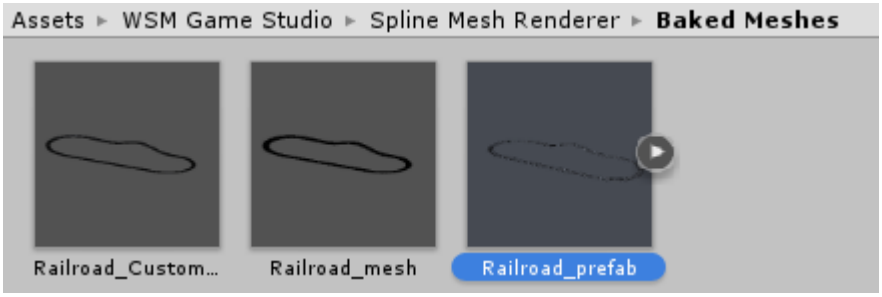

Figure 60 - Baked meshes & prefab

If collision is enabled it will also save all mesh colliders and apply them to the generated prefab (See the Collision Settings section for more details)

The generated prefab can be used to replace the Spline Mesh Renderer on your scene. This will reduce the scene loading time and increase performance.

NOTE: If you want to change your mesh in the future you can keep the Spline Mesh Renderer as a disabled object on your scene instead of deleting it. That way you can always adjustments and export the results again.

3.3.4 - Connect New Renderer

The “Connect New Renderer” operation creates a new Spline Mesh Renderer at the end of the current selected one. The newly created object inherits all the characteristics and components from the original object. However, the new spline is generated with only one straight curve segment.

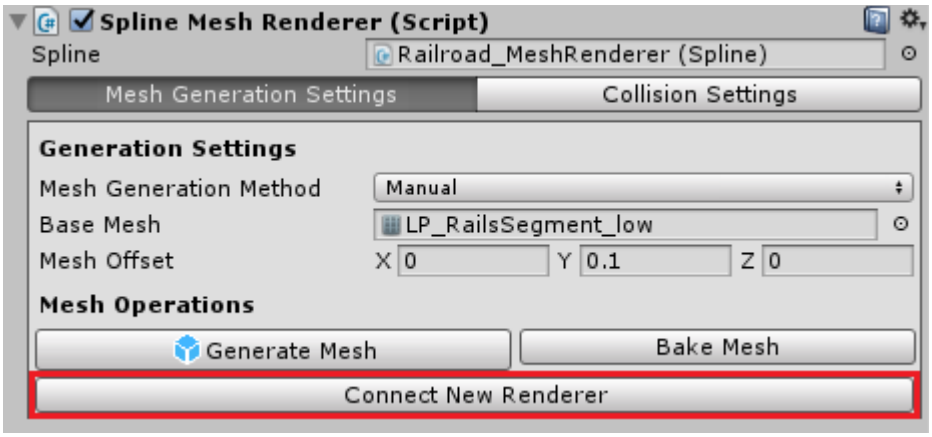

Figure 61 - Connect New Renderer operation

It is very similar to the Append Spline operation, both can be used to produce the same effect.

This feature is very useful if you want to create complex paths, composed by different meshes. For example, you can change the base mesh to create a railroad segment with a different kind of rails.

The new Mesh Renderer will always perfectly connect to the end of the previous one. Like can be seen in the sample image below:

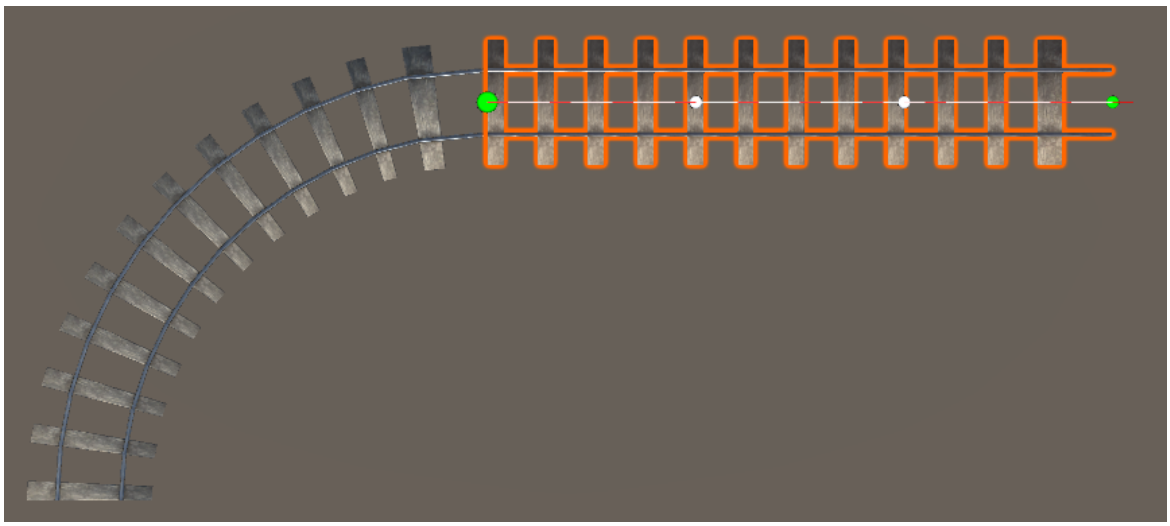

Figure 62 - Connect New Renderer sample

3.4 - Collision Settings

By default, the Spline Mesh Renderer collision is disable. Therefore, no colliders will be generated and the Mesh Collider component attached to the Spline Mesh Renderer will be always disabled.

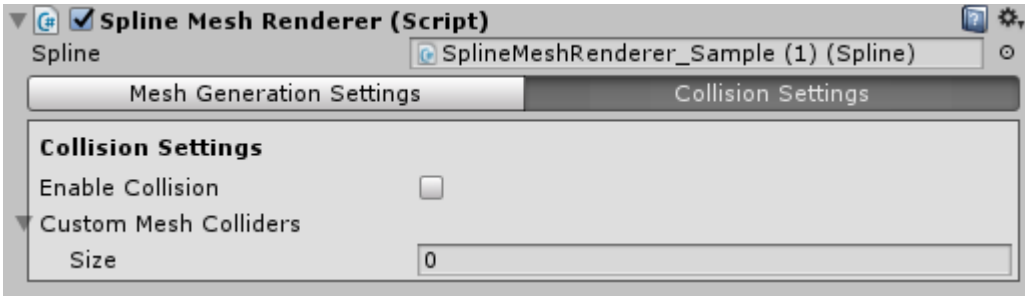

Figure 63 - Spline collision settings

By checking the "Enable Collision" checkbox, by default it will enable the Mesh Collider component and use the generated mesh for detecting collision. However, using the same mesh to render the object and detect collision may not be a great idea sometimes, that's because the Mesh Collider component is the most performance heavy collider in Unity. The ideal situation is to use custom colliders with minimum number of vertices as possible.

The "Custom Mesh Colliders" property allows you to use one or more custom colliders generated from different meshes.

3.4.1 - Creating a Custom Collider

To create a custom collider, Drag & Drop the "CustomColliderRenderer" prefab to your scene.

Figure 64 - Custom Collider Renderer prefab

Select the CustomColliderRenderer object on your scene and set it's reference Spline to the same Spline used by your main Spline Mesh Renderer.

Then, set the Base Mesh property with your custom collider mesh.

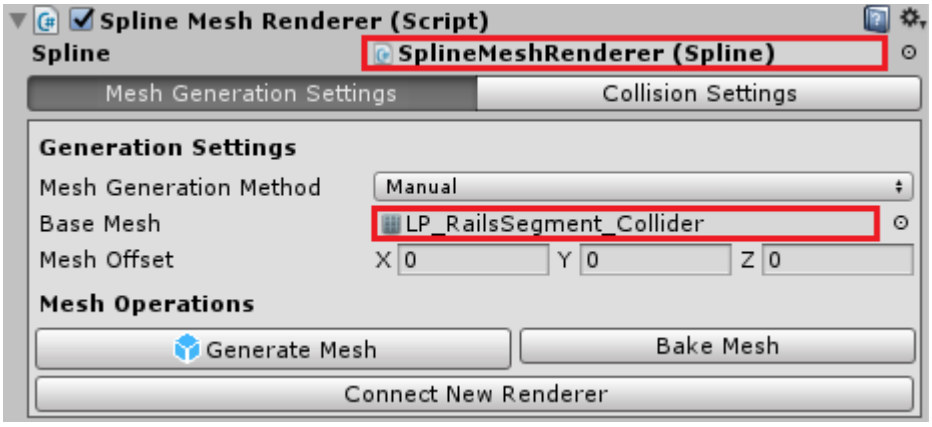

Figure 65 - Custom Collider Renderer setup

Now all you have to do is to add the CustomColliderRenderer object into the “Custom Mesh Colliders” list of your Spline Mesh Renderer object, as shown in the sample imagem below:

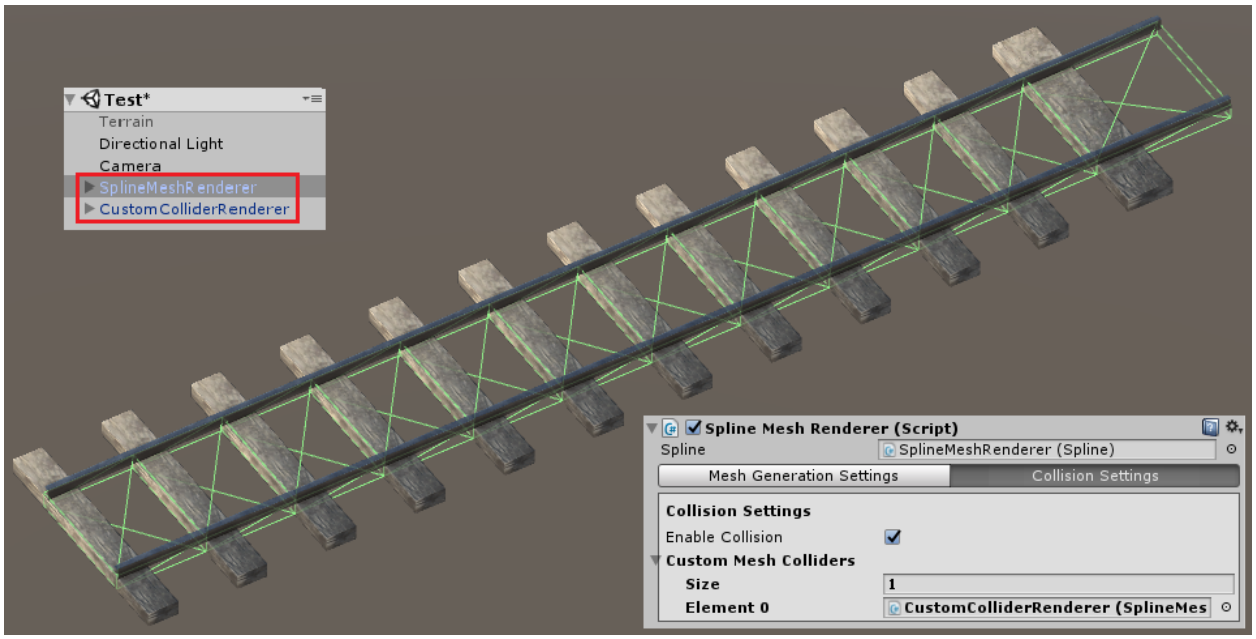

Figure 66 - Custom collider sample

Every time you change your mesh, the custom colliders will also be automatically updated. Also, while using the Bake Mesh feature, custom colliders will be exported and saved as children of the generated prefab.

3.5 - Mesh Vertices Limit

If you intend to build super long meshes for open world games for example, eventually you may see a warning like this:

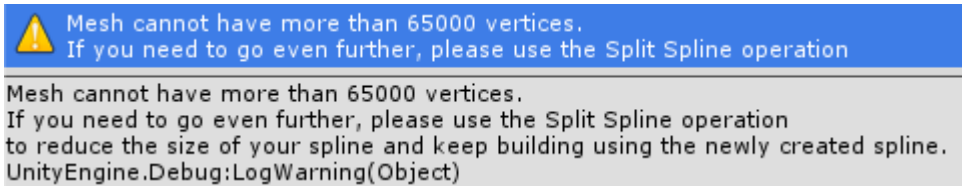

Figure 67 - Vertices limit reached warning

This happens because Unity has a limit of 65000 vertices per mesh. This limit exists to avoid performance issues when rendering large meshes. This is limit is also, very useful to apply Occlusion Culling on your railroad and increase your game performance. If you see this warning, just follow the instructions below to keep building your super long mesh.

With the Spline Mesh Renderer object selected, go to the end of your mesh and check if the mesh ending matches the last control point. It probably don't match, like in the sample image below.

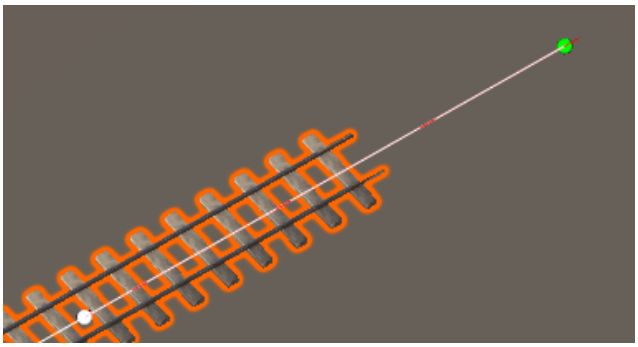

Figure 68 - Vertices limited rendering at the end of spline

To solve this issue and keep building, all you need to do is to select the last control point that was successfully rendered.

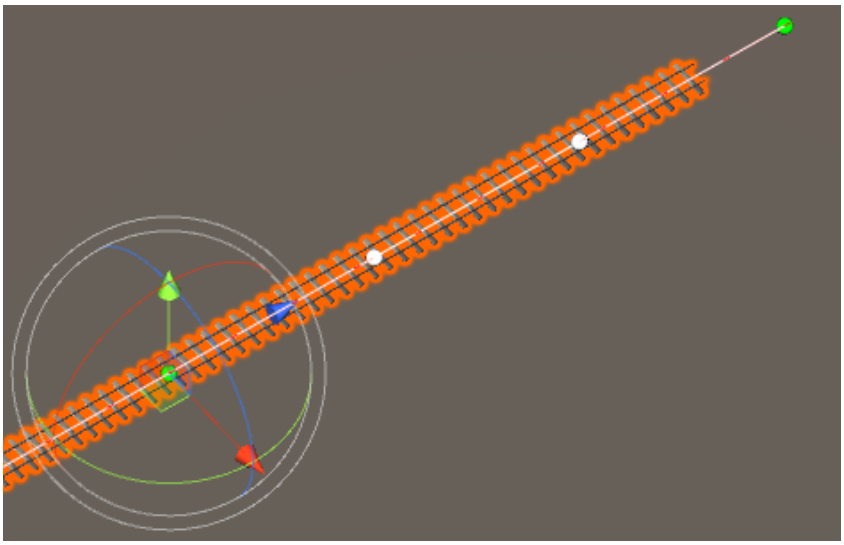

Figure 69 - Last successfully rendered control point selected

And then use the Spline Spline operation to start a new section for your mesh.

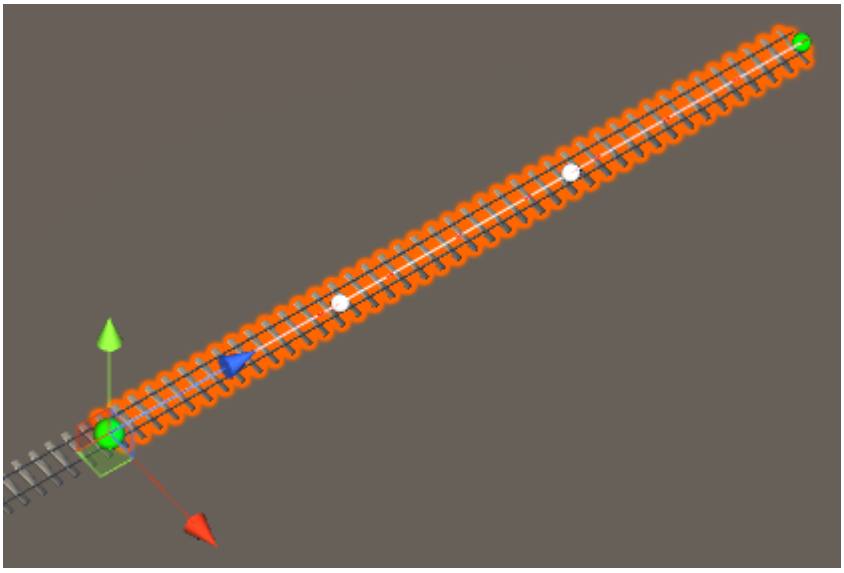

Figure 70 - New section created using the Spline Spline operation

NOTE: Real time mesh generation is disabled automatically when max number of vertices is greater the 65000. This is the default behaviour to increase the editor performance while building your mesh, since processing large meshes data in realtime is costly.

You can change back to the real time mesh generation manually after reducing the rails, but, for very long meshes it's recommended to use the manual mesh generation button to update the mesh after adjusting your splines curves.

The max length of each segment, depends on the base mesh vertices count. For example, the base rail mesh included in the package, has 56 vertices in total. By using it, it is possible to create railroad segments up to 1.35 Km each. To create railroads longer that 1.35 Km, use the technique described above to create multiple railroad segments connected to each other.

4 - Spline Prefab Spawner

The "Spline Prefab Spawner" component is a level design tool that allows you to easily spawn prefabs along a spline path to populate your scene.

It can be used to easily populate your scene with objects that repeats along a path, such as light poles, bridge pillars, check points, etc.

4.1 - Creating a Spline Spawner

To create a Spline Spawner, Drag & Drop the “SplinePrefabSpawner” prefab to your scene

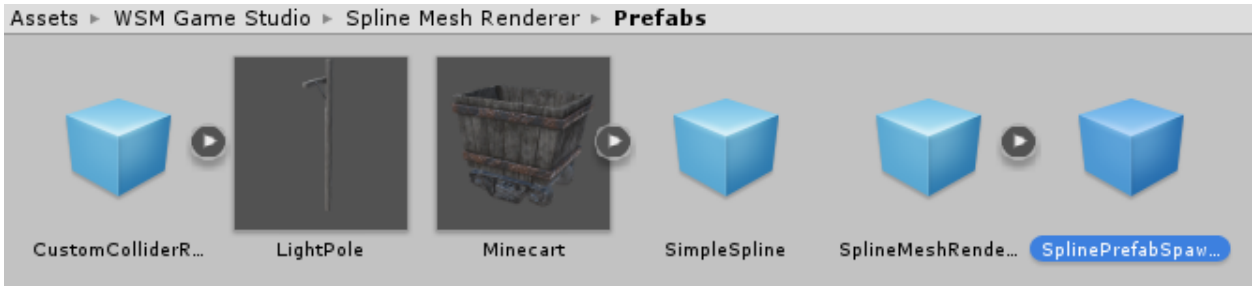

Figure 71 - SplinePrefabSpawner prefab

Note that the “SplinePrefabSpawner” prefab already has a Spline component attached to it. By default, the Spline Prefab Spawner component will use the attached spline component as reference. But, if you wish to use another spline on your scene as a reference, you can replace it on the Spline property.

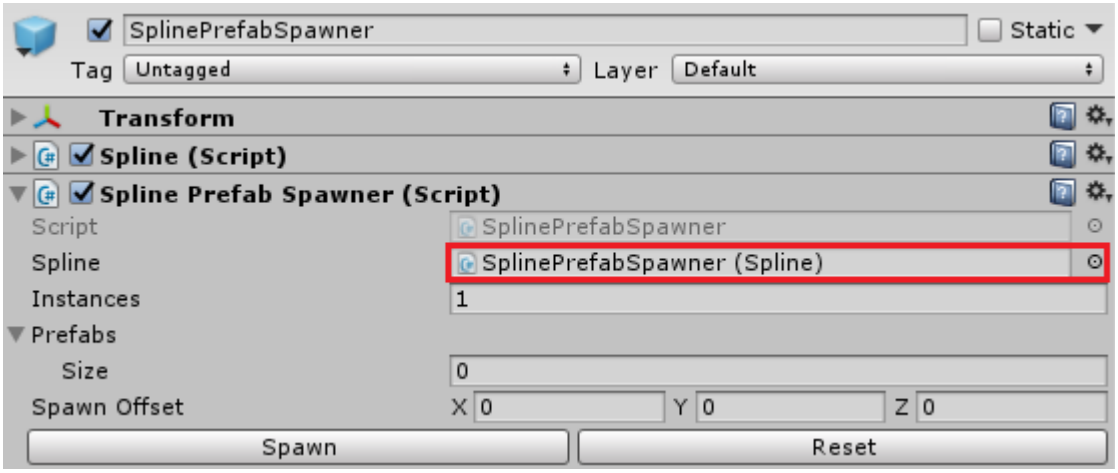

Figure 72 - Reference Spline

The “Instances” property defines how many instances of each prefab will be spawned along the spline.

The “Prefabs” property is a list of all the prefabs you want to spawn.

By default, the instances will be spawned on top of the spline, but you can use the “Spawn Offset” property to change the position of the spawned object if you wish so.

4.2 - Spawning Prefabs

Spawning prefabs is really simple, just fill the Prefabs and Instances property on the Inspector and click on the “Spawn” button.

If the Follow Terrain property of the reference Spline is enabled, the spawned objects will follow the terrain elevations as well.

In the sample image below, it was used to spawn 14 light poles along a spline.

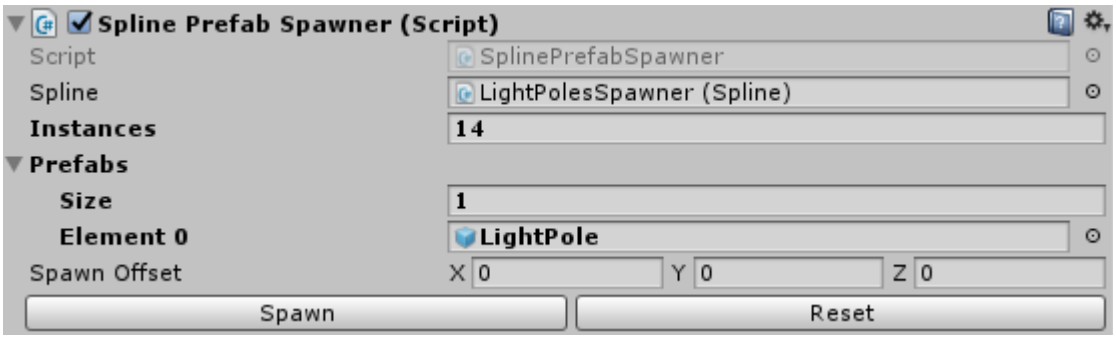

Figure 73 - Spline Prefab Spawner sample

The result can be seen on the image below. Note how the light poles were instanced based on the terrain elevation.

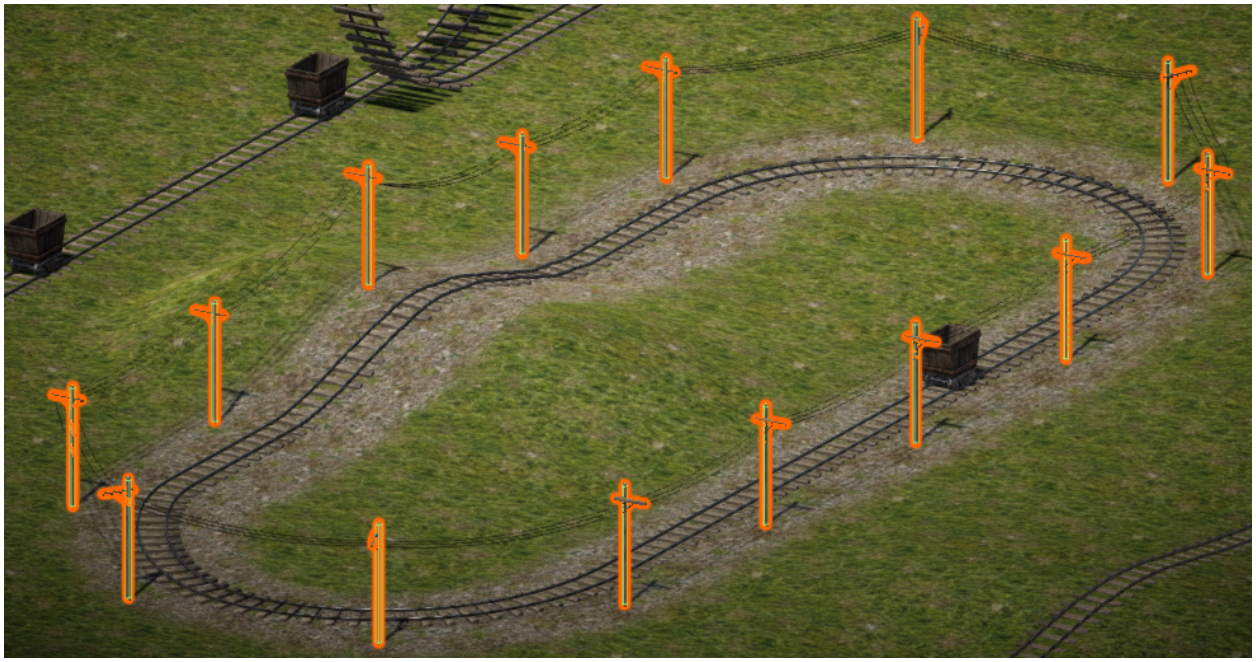

Figure 74 - Spawned light poles

If you decide to change the number of instances after spawning, there is no problem, just change it and click the Spawn button again and it will automatically respawn the new number of instances.

NOTE: All spawned instances are set as children of the Spline Prefab Spawner object. This is the default behaviour, to keep the objects Hierarchy organized and to easily identify what objects were spawned using the Spline Prefab Spawner component.

4.3 - Deleting Prefabs

Deleting spawned prefabs is also pretty simple. All you need to do is click the “Reset” button and all spawned instances will be deleted.

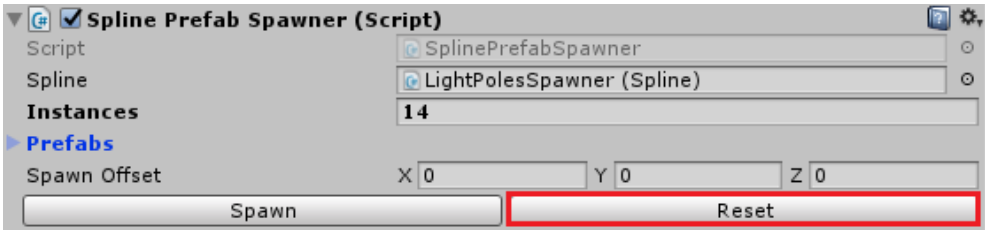

Figure 75 - Reset operation

NOTE: Spline Prefab Spawner component identifies what objects were spawned by it by looking at his children on the Hierarchy. **DO NOT** set other objects in your scene as children of an object that has a Spline Prefab Spawner component attached to it to avoid deleting objects unintentionally.

5 - Spline Follower

The “Spline Prefab Spawner” component allows any object on your scene to follow a path defined by spline. It can be used to easily create dynamic moving props for visual or gameplay purposes, such as complex platforms movement, orbiting behaviour, etc.

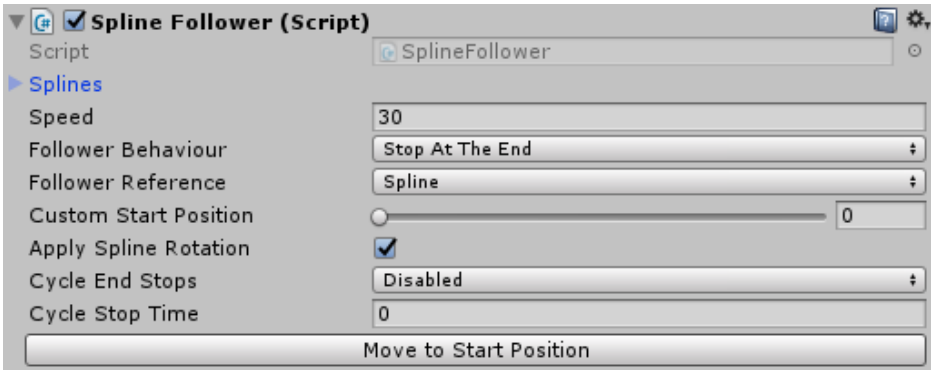

Figure 76 - Spline Follower component

5.1 - Creating a Spline Follower

To add the spline following behaviour to any object all you need to do is to add a “Spline Follower” component to it.

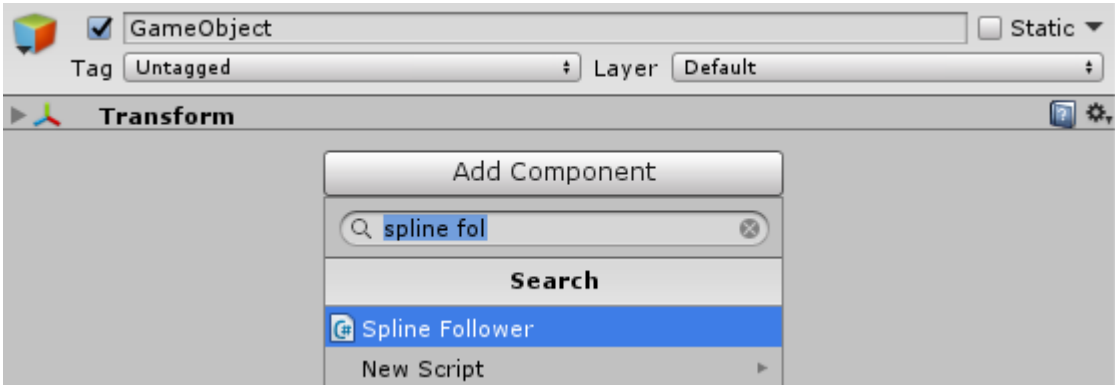

Figure 77 - Adding the Spline Follower component

5.2 - Following Single x Multiple Splines

It is possible to follow one or more splines. The “Splines” Property defines what splines will be followed. For multiple splines, it will always consider the order of the splines on the list.

The “Speed” Property defines how fast the object will move regardless of how many splines it will follow.

5.3 - Following Behaviour

The “Follower Behaviour” Property defines how the object should behave once it reaches the end of the spline list.

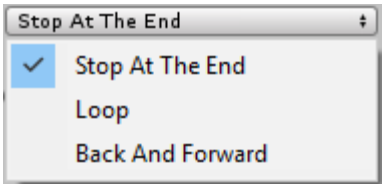

Figure 78 - Follower Behaviour options

The “Follower Reference” Property defines if the object should follow the spline absolute shape or it’s terrain projection.

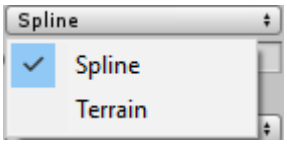

Figure 79 - Follower Reference options

> **NOTE:** The Follow Terrain property of the reference spline must be enabled for the Terrain follower reference option take effect.

The “Apply Spline Rotation” Property defines if the object should rotate while moving along the spline. This is property is enabled by default, if you wish to create moving platforms for example you need to disabled this option to keep the platforms from rotating.

5.4 - Start Position

The “Custom Start Position” Property defines the percentage of the first spline the object will use as a start point position. For example, if you wish the object to start at middle of the spline, you can set this property to 50%.

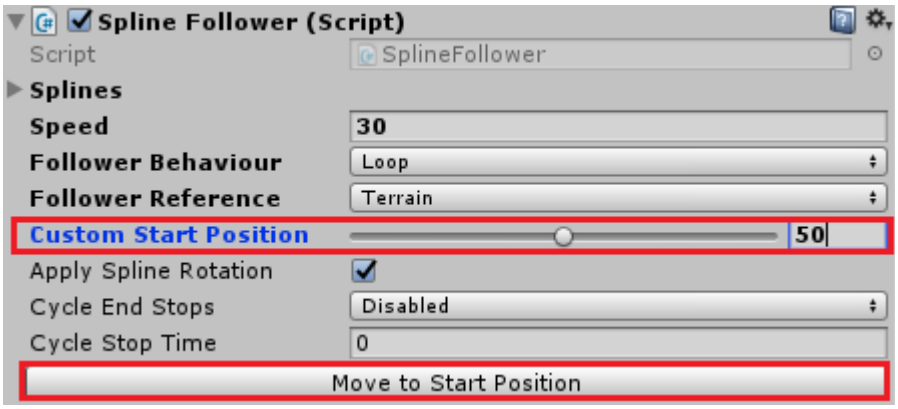

Figure 80 - Start position properties

After adjusting the position click the “Move to Start Position” button to preview where the object will start.

5.5 - Stops

The “Cycle End Stops” Property defines if and when the object should make temporary stops along the way.

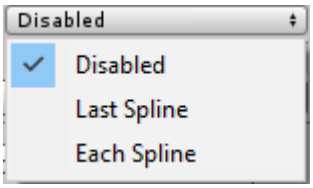

Figure 81 - Cycle End Stops options

The “Cycle Stop Time” Property defines how long temporary stops will last (in seconds).

6 - Practical Use Samples

This section contains some practical uses samples created with the Spline Mesh Renderer, Spline Follower and Spline Prefab Spawner.

6.1 - Minecarts Demo Scene

At the first demo scene, the Spline Follower and Spline Mesh Renderer were used to create a complex railroad path for minecarts.

Figure 82 - Minecarts sample

This demo demonstrates the power of the spline components when used together to create complex scenes. Note how the minecarts correctly follows the loop, jump and terrain elevations.

This complex multi-spline railroad was first created as a single Closed Loop spline, then it was divided into smaller segments using the Split Spline operation.

Each segment was then configured accordingly to the desired behaviour. The Custom Upwards Direction of the top segment of the vertical loop was set to Down,

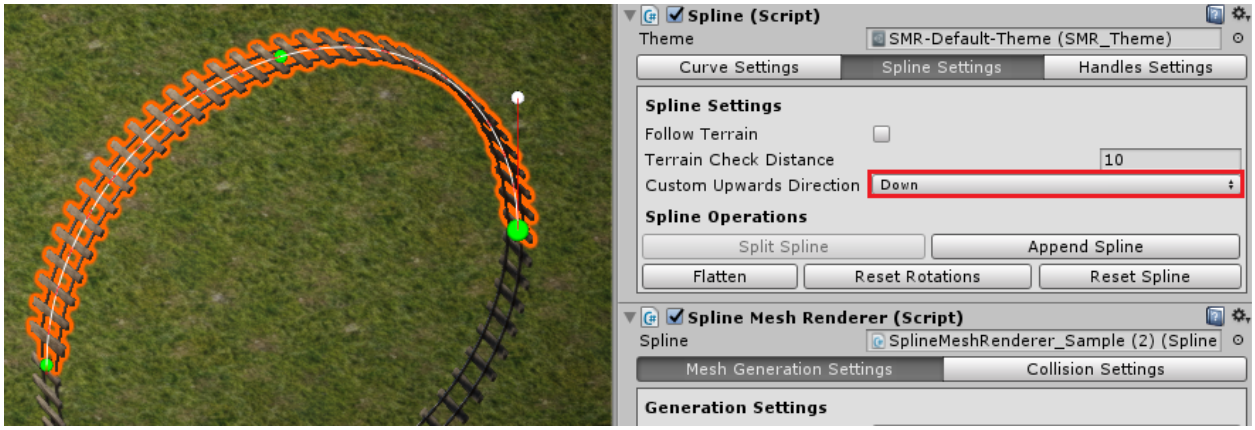

Figure 83 - Vertical loop settings

The Mesh Renderer component of one of the segments was disabled to simulate the "jump" behaviour.

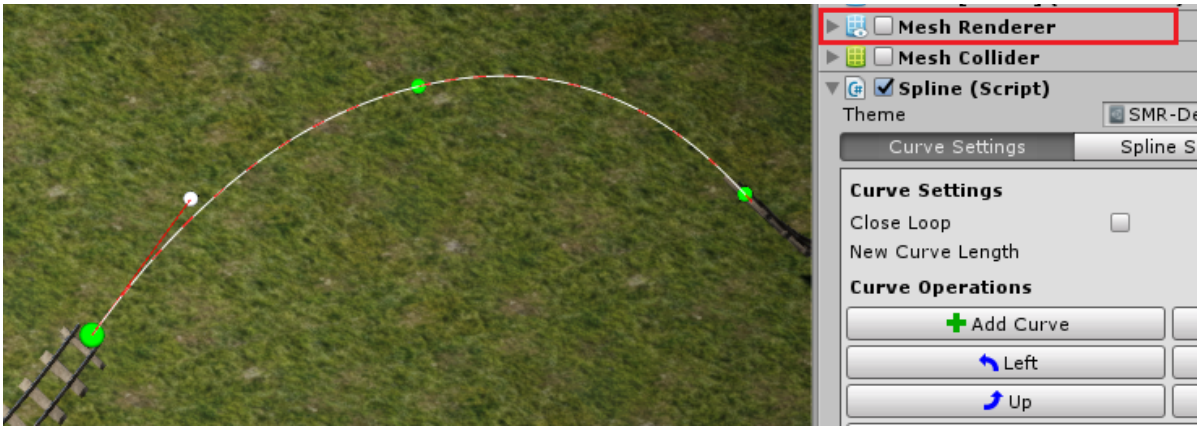

Figure 84 - Jump settings

And finally the Follow Terrain feature was enabled on the segments that went through terrain elevations.

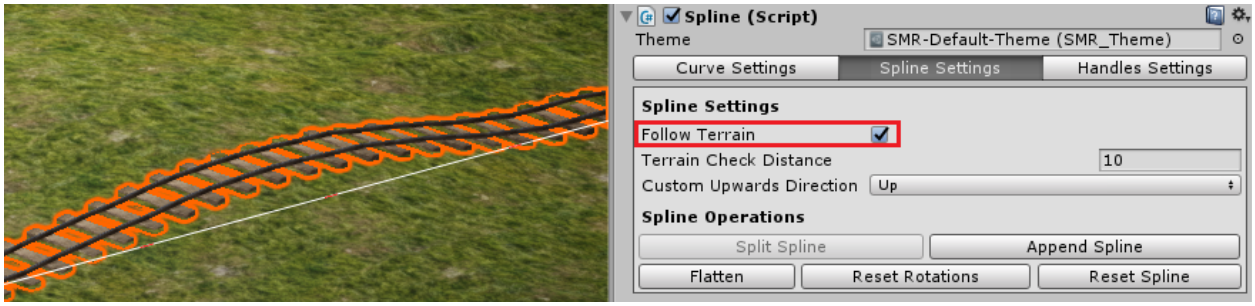

Figure 85 - Follow terrain

The Spline Prefab Spawner was used to instantiate some light poles.

Figure 86 - Light poles

6.2 - Solar System Demo Scene

At the “Follower_SolarSystem” demo scene the Spline Follower component was used to create planetary orbit behaviour.

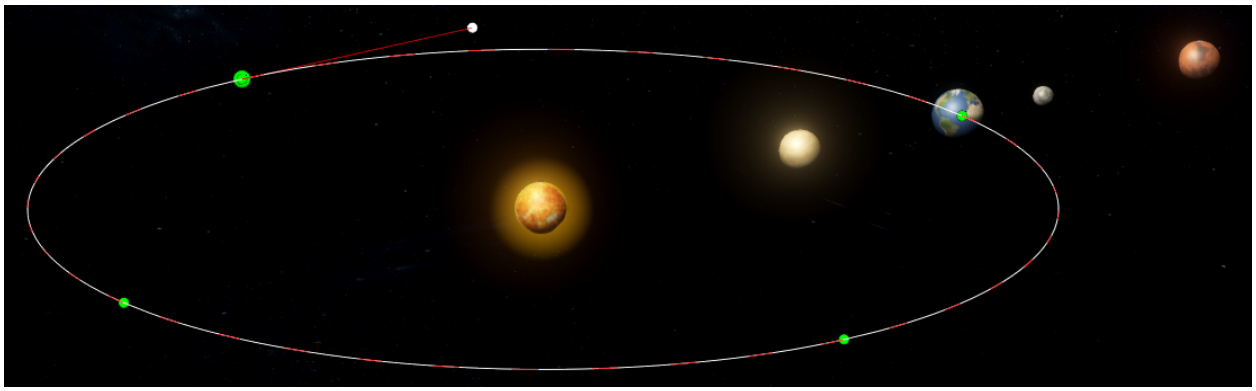

Figure 87 - Solar system sample

The Curve Shaping Operations were used to create perfect circular splines for the orbits. Since each predefined curve creates a perfect 45º curve, the orbit circle is composed by four 45º curves.

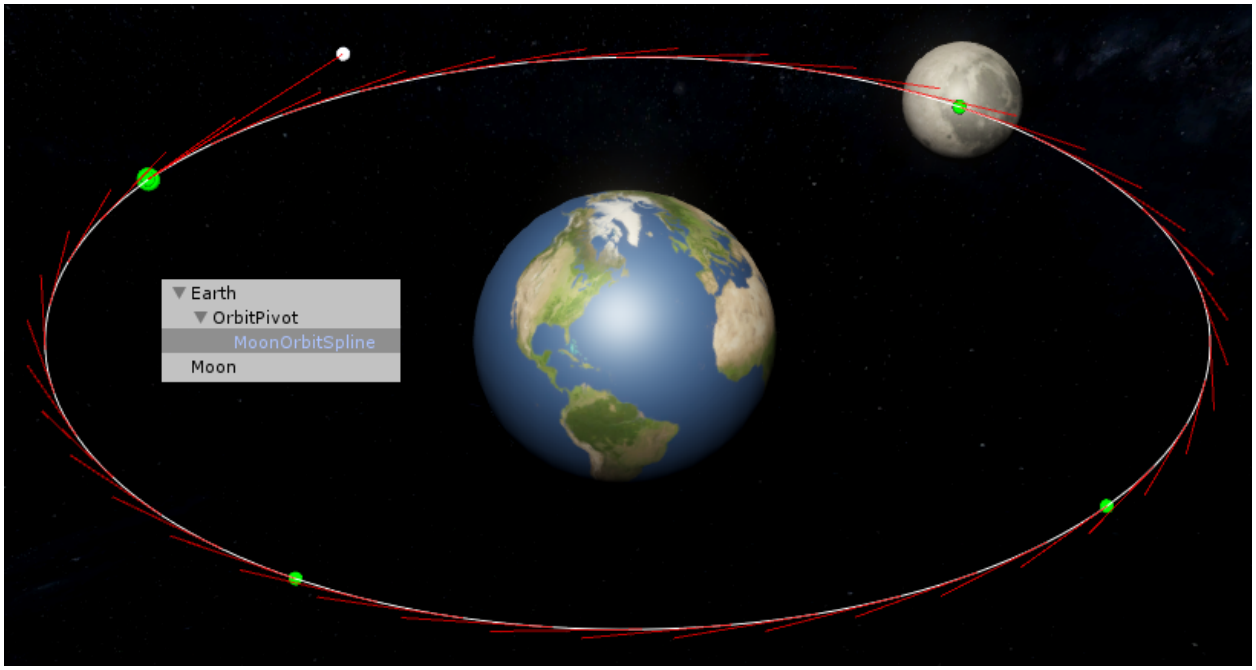

Figure 88 - Moon orbit

As you can see in the sample image above, the "Moon Orbit Spline" object is set as children of the Earth object. As any other game object, splines also inherits their parents movement, by using this in our advantage the moon orbits correctly around the Earth and inherits the Earth's orbit movement around the Sun.

6.3 - Moving Platforms Demo Scene

At the "Follower_MovingPlatform" demo scene the Spline Follower component was used to create complex moving platform behaviour.

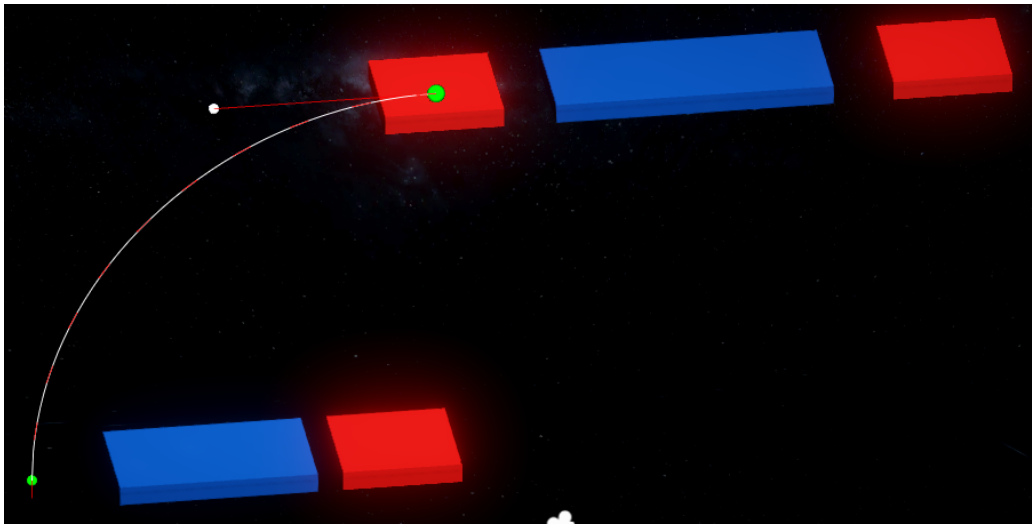

Figure 89 - moving platforms sample

In most games moving platforms don't rotate around themselves, so they “Apply Spline Rotation” property was disabled to ensure the platforms would still face the right direction all the time.

Cycle End Stops were also used to simulate stops at interest points, this can be used to let the player know that the platform has reached its destination.

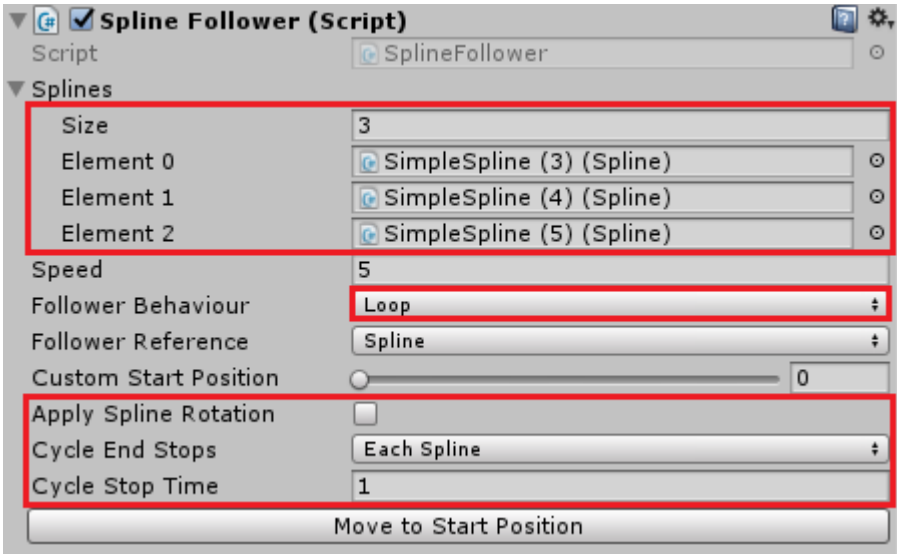

Figure 90 - Platform settings sample

In the sample image above, the platform was setup to follow multiple splines and stop for 1 second at the end of each one. When it reaches the end of the splines list, it starts all over again (Loop following behaviour).

7 - Performance Guidelines

Procedural mesh generation is powerful however it also is a costly operation. This section contains performance guidelines to avoid performance issues on the Unity Editor and improve the loading time and performance of your game.

7.1 - Manual x Realtime Generation

Always prefer using the manual Mesh Generation Method workflow. Although the real time generation is useful while working with short meshes it can be performance heavy for the Unity Editor while working with long meshes.

7.2 - Terrain Following

Avoid using the Follow Terrain feature and the real time mesh generation at the same time while working with long meshes.

Also, always prefer working with low Terrain Check Distance values, increasing its value is the easiest way to guarantee the spline will correctly identify the terrain all over your map, however, if any segment of your spline does is not correctly projected in your terrain you can also roughly drag the spline closer to the terrain instead of raising the check distance property.

7.3 - Mesh Length

By using the Spline Mesh Renderer component is possible to create long curved meshes. For example, it is possible to create railroad segments up to 1.35 Km long.

However, for better performance always prefer creating short mesh segments.

If you need to create long meshes, you can create multiple segments connected to each other by using the Connect New Renderer or the Split Spline Features.

Also, by using short mesh segments you can take advantage of the Occlusion Culling feature to improve the performance of your game.

7.4 - Base Mesh Vertices Count

When creating custom Base Mesh models for the Spline Mesh Renderer, keep the polycount low.

By using low poly models you reduce the Unity renderer workload. Unity has a limit of 65000 vertices per rendered mesh, however it doesn't mean it is performance wise to have high poly meshes all around your scene.

Low poly base meshes, besides being more performance friendly, also allows you to create longer mesh segments on the Unity Editor.

7.5 - Prefab Baking

When preparing the final build of your game, always bake all meshes generated by the Spline Mesh Renderer as prefabs using the Bake Mesh feature.

This will drastically reduce the scene loading time. This happens because the Spline Mesh Renderer procedural mesh generation is executed on the start of the scene to ensure all meshes are correctly generated.

In the final build of your game, always replace the Spline Mesh Renderer components by baked prefabs.

If you are using the Spline component attached to the SplineMeshRenderer object on your scene and need it to still be active in play mode for gameplay purposes, you can keep it by removing the Spline Mesh Renderer and Mesh Renderer components manually.

If you don't need the Spline component for gameplay purposes, then you can disable or delete the entire object on your hierarchy after replacing it by the baked prefab.

8 - License

By purchasing this asset you are allowed to use it for unlimited games and/or 3D projects (like animations, simulation softwares, etc). Both personal and commercial use.

You are NOT allowed to resell or distribute the assets components individually or as part of another asset package (including, models, scripts, etc).

9 - Contact Info & Support

If you have some questions, need support or have some business inquiries, feel free to get in touch.

Asset Store
Sketchfab
Instagram
Facebook
Twitter
Youtube Channel