WSM GAME STUDIO

User Manual

v3.4

2021

SUMMARY

1. Intro

Thank you for purchasing “Train Controller (Railroad System)”!

This package contains all that you need to build a simple and functional railroad (models, scripts and SFX).

More models may be included in the future and/or sold separately as extensions (Addons Available).

It’s really simple to use and customize.

1.1. What is new on v3.4?

Up to v3.3, the asset was mainly focused on Physics Based Trains. Starting with v3.4, Spline Based Trains will also be supported.

But why is the inclusion of spline based trains such a big deal?

Well, physics based trains are fun, but are not suitable for all types of projects. For instance, must AAA open world games don’t rely on physics based trains, to name a few recent examples: Assassin’s Creed Syndicate, Red Dead Redemption 2 and the infamous unstoppable GTA V train, all of which were developed using non-physics trains solutions.

But why do AAA games avoid using physics based trains?

The answer is simple, in the current state of technology there is no perfect physics simulation solution for games. Even AAA games are susceptible to random and unpredictable physics glitches caused by the physics engine miscalculations.

Since trains are complex game objects by nature, if the main focus of the game is not accurate train simulation, using a non-physics train solution may be a viable solution to ensure stable and predictable train movement.

By including spline based trains support, the asset will now be useful for a broader range of projects, making it a better train solution for everyone.

1.2. Update Guidelines v3.4

If you haven’t installed any previous versions of this package on your project, or have completely removed any previous versions, then you don’t need to worry about this section.

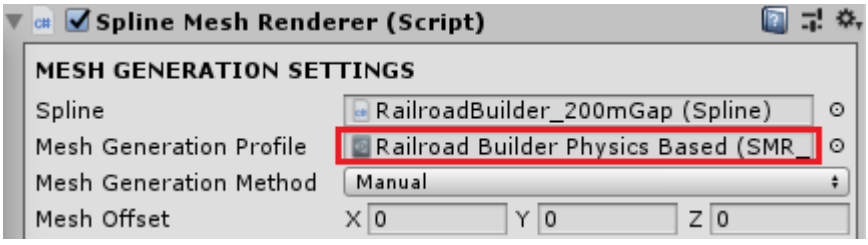
But, if you have already installed any previous versions on your project and downloaded version 3.4 as an **update** inside the Unity Editor. Then, you may need to check your **custom prefabs**, since the Unity Editor will **not** update custom prefabs automatically.

For custom Train prefabs created on previous versions, it’s recommended to replace the locomotive and all wagons to make sure the new changes will apply correctly.

If you have downloaded any Additional Wagons, don't forget to download the corresponding updated version of those wagons before replacing them on your custom train prefabs.

You can also use the new Custom Wagons Creator to rebuild your own custom locomotive and wagons prefabs. This will ensure that the custom wagons and locomotives are set up correctly.

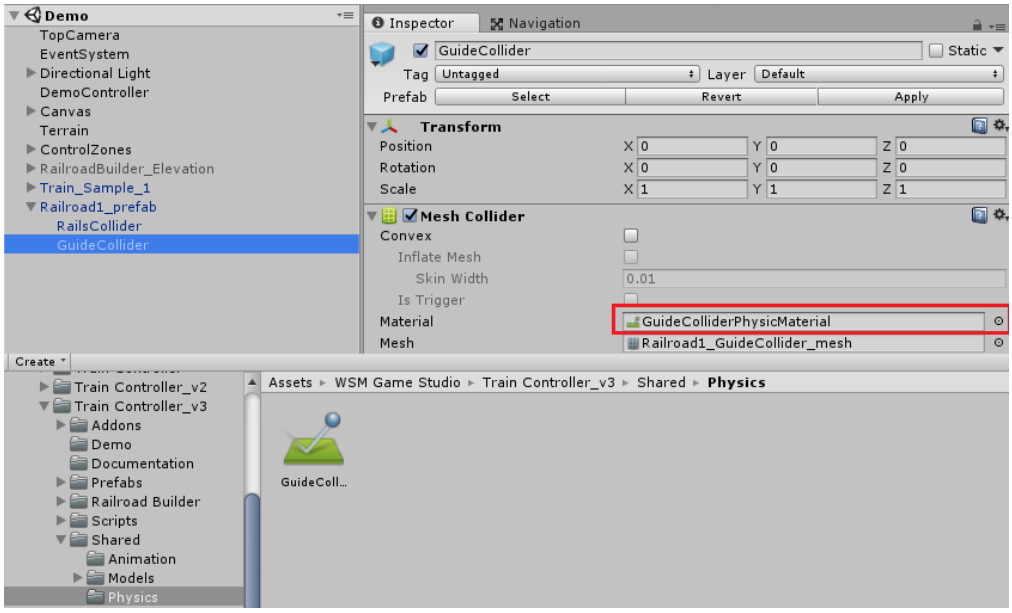
For any railroad builder created before the update, may also need minor adjustments (selecting mesh generation profile and regenerating rails (1 click process)).



Baked rails segments need to be regenerated due to the new features regarding spline backup, railroad builder LOD support and the new BakedSegment component that is used to easily connect baked segments in the editor.

Custom railroad turnouts will also need to be regenerated to support spline based trains.

For custom **baked** railroads and/or rail segments created on versions previous to v3.3, you need to apply the "GuideColliderPhysicMaterial" to your custom rails "GuideCollider" child object manually.



2. How to Use This Asset

In this section you will learn how this asset works and how to use it based on the sample prefabs.

2.1. Quick Start Guide

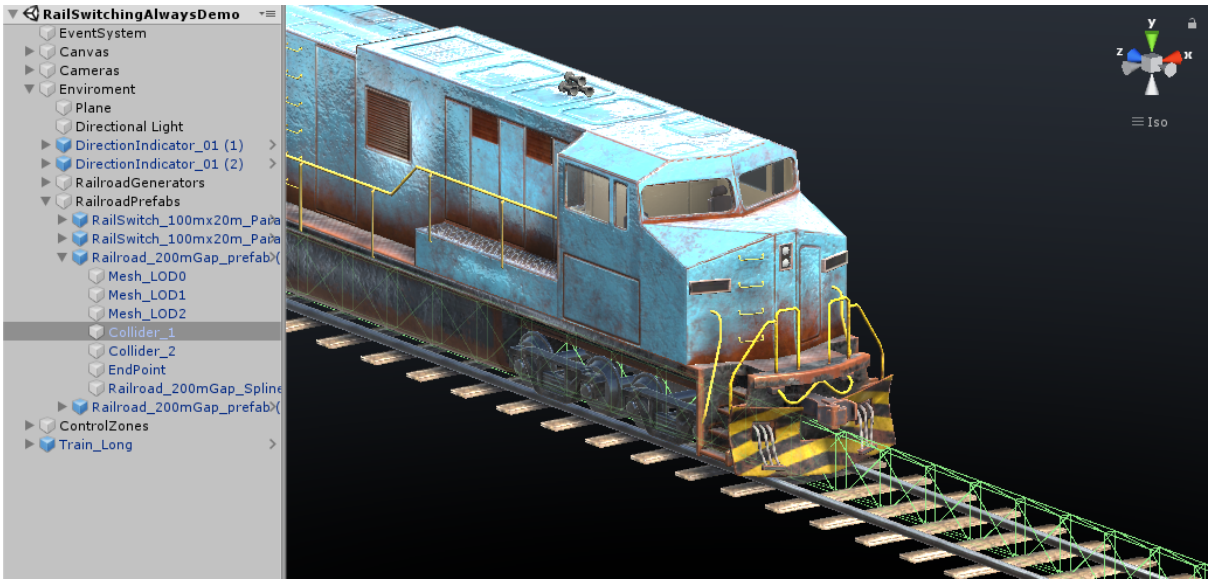
This asset supports both physics based and spline based trains. Regardless of which approach you decide on using on your game, the overall workflow is almost the same.

- Choose between physics based or spline based trains
- Use the Railroad Builder to create railroads on your scene
- Use the Route Manager to define the routes on your scene
- Use the Train Spawner to spawn locomotives, wagons and trains
- Customize

Customization tools are also available to make it easier to customize your project. For more information, take a look at Custom Wagon Creator, Custom Turnout Creator and Rail Generation Profiles sections.

2.2. Physics Based Trains

Up to v3.3, this asset was mainly focused on **physics based train movement**. Physics based trains works by using guide colliders to keep the train on rails, while applying forces to simulate acceleration, brakes and wagon connections.



The rails are composed of base colliders and guide colliders that compose the path that the train will follow.

The TrainController component attached to the locomotive controls acceleration, brakes, SFX, doors, lights and wagons.

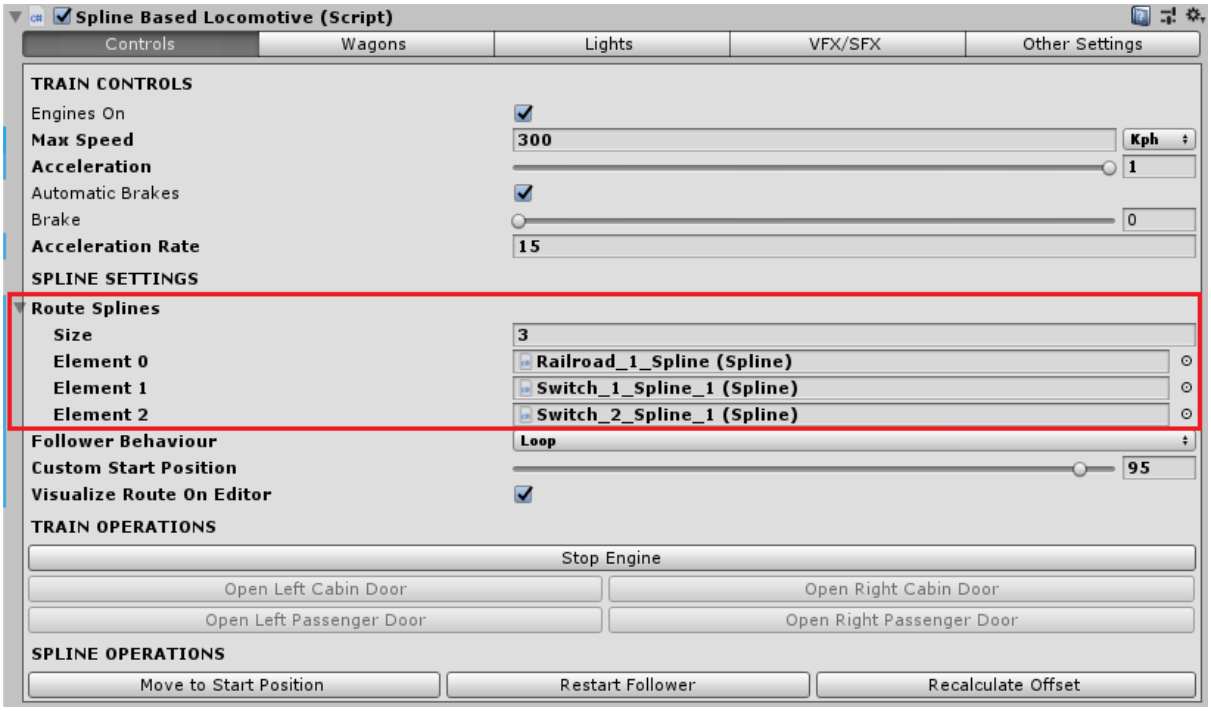
The wagons are connected by Joints and inherit locomotive movement and commands while connected.

2.3. Spline Based Trains

Spline based trains are a non-physics train solution that works by using a spline following algorithm to follow a predetermined route.

> **Note:** Spline based trains reuse the same splines used by the Railroad Builder to generate the rails. However, unlike physics based trains, spline based trains don't need rail colliders to stay on track. For more information about rails generation, please take a look at the Railroad Builder section.

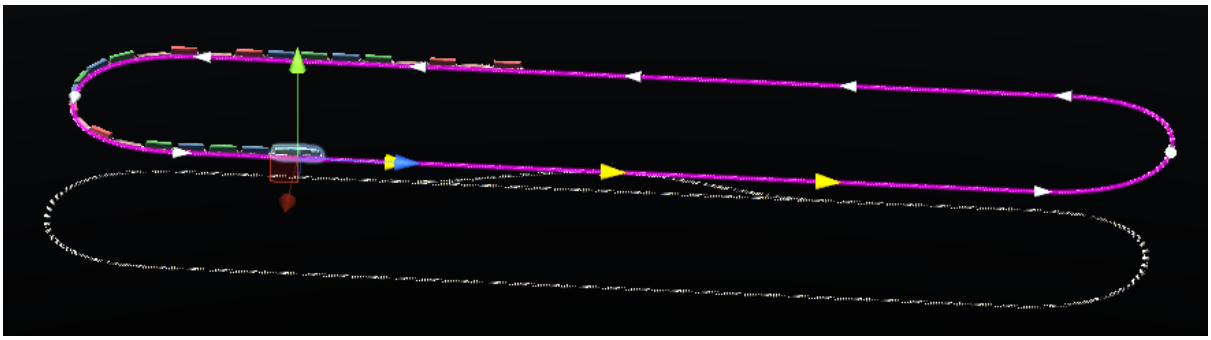
A route is composed of one or more splines assigned to the Spline Based Locomotive component, as shown in the sample image below.



Routes can be manually assigned to any locomotive by dragging and dropping splines on your scene to the Route Splines property of any locomotive, however, it is highly recommended to use Route Manager to create, edit and assign routes to trains.

> **Note:** Spline based trains need splines to move, therefore, if the Route Spline property is not assigned, the train will not be able to move.

When “Visualize Route on Editor” is enabled, it is possible to visualize the current route assigned to the selected locomotive.



The Spline Based Locomotive component attached to the locomotive controls acceleration, brakes, SFX, doors, lights and wagons.

The wagons are connected using a linked spline following algorithm and inherit locomotive movement and commands while connected.

2.4. Physics Based X Spline Based Trains

Physics based and spline based train movement are two distinct approaches used to simulate train movement on games, both of which have its pros and cons:

- **Physics Based Trains Pros**
  - Wheels follows the rails curvature
  - Couplers stay connected to each other on curves
  - Can simulate derailment (if you need to)
  - Simulates train shaking based on speed
  - Turnouts are easier to set up
- **Physics Based Trains Cons**
  - May derail at high speeds on sharp curves
  - Maximum speed limit (105kph)
  - Performance heavy for mobile games
  - Hunting oscillation at higher speeds
  - Susceptible to Unity physics engine glitches
  - Wagons will not follow railroad curves when spawning a train

- **Spline Based Trains Pros**
  - Never derails
  - No speed limit
  - Performance friendly for mobile games
  - Smooth movement at higher speeds (No Hunting Oscillation)
  - Wagons follows railroad curves when spawning a train
- **Spline Based Trains Cons**
  - Wheels doesn't follows the rails curvature
  - Couplers don't stay connected to each other on curves
  - Turnouts relies on Route Manager to change route

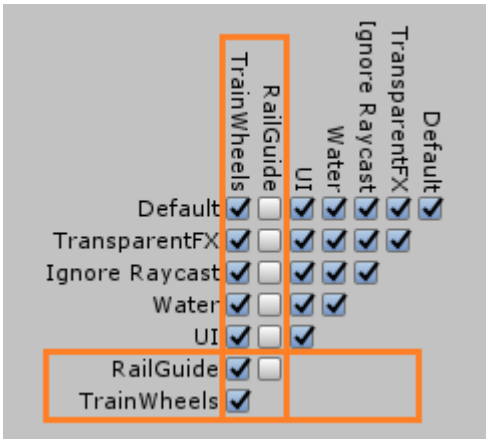
Note: Which approach is ideal for your game, depends on your project's needs. Therefore, before starting your project, analyse the pros and cons and select which one is more suitable for you.

2.5. Recommended Layers for Physics Based Trains (Optional)

The package is ready to use as it is. Although, to avoid unwanted collisions with the rails guide colliders when using physics based trains, it is recommended to customize the Unity Collision Matrix.

Go to "**Edit > Project Settings > Tags and Layers**", add new layers for the rail guides and train wheels (Ex: RailGuide and TrainWheels). Change the layers of the train wheels and rail guides of the prefabs.

Go to "Edit > Project Settings > Physics" and change the collision matrix so the "RailGuide" layer will collide **only** with the 'TrainWheels" layer.

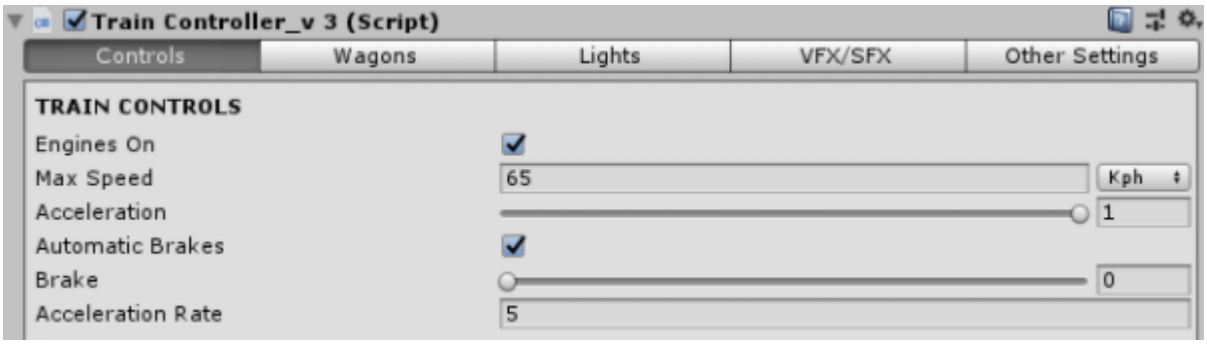


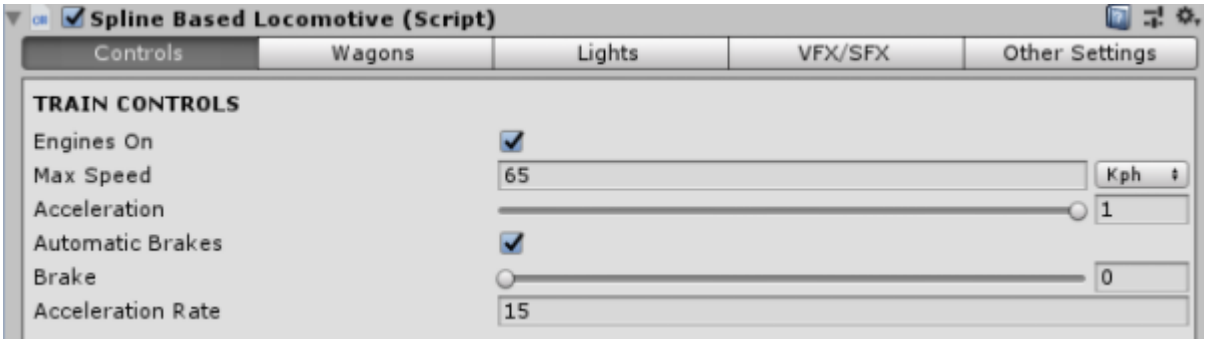
For more info about the collision matrix, check the layer-based collision official documentation.

2.6. Sample Prefabs

This package includes a ready to use railroads and train sample prefabs. These prefabs can be seen in action at the demo scenes included in the project.

If you wish to use these prefabs on your game, all you need to do is drag and drop a railroad and a train to your scene. You can also customize speed and acceleration in the TrainController or Spline Based Locomotive component attached to the locomotive.





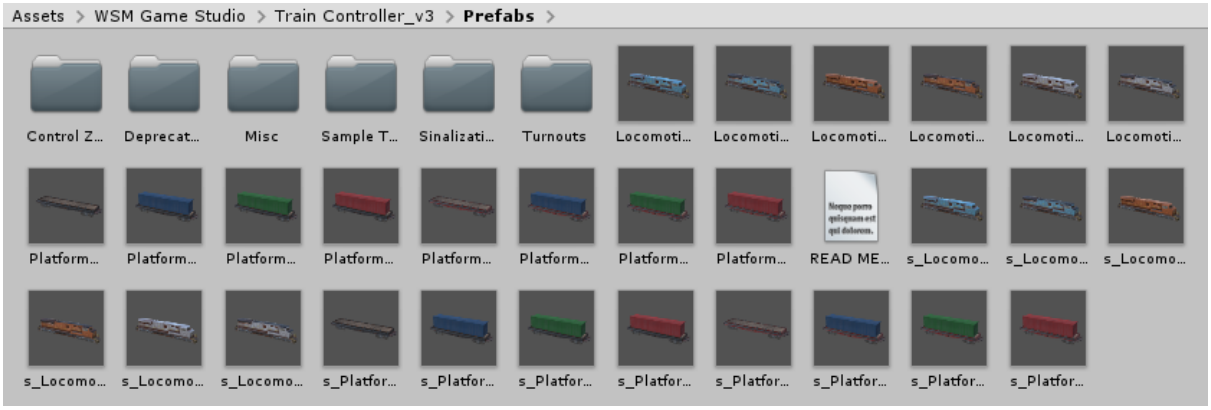
Note: By default, the sample prefabs are configured as automated trains, if you wish to manually control your train, take a look at the player input section.

2.7. Locomotive & Wagons Prefabs

A train is composed of a locomotive and wagons. There are six locomotive variations and a set of wagons prefabs at the “Prefabs” folder.

More locomotives and wagons may be included in this package in the future and/or sold separately as extensions (Addons Available Here)

See the Building a Train section for more info on how to use these prefabs.



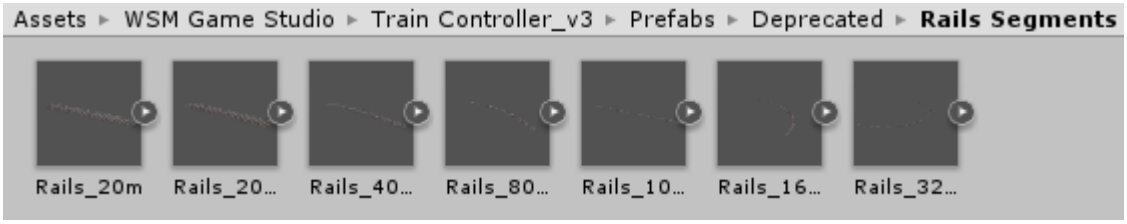
Note: Spline based locomotives, wagons and sample prefabs were named using the ***“s_”*** prefix for easier identification.

2.8. Legacy Prefabs (Deprecated)

Up to version v2.0, the railroads were composed of joined rail segments prefabs.

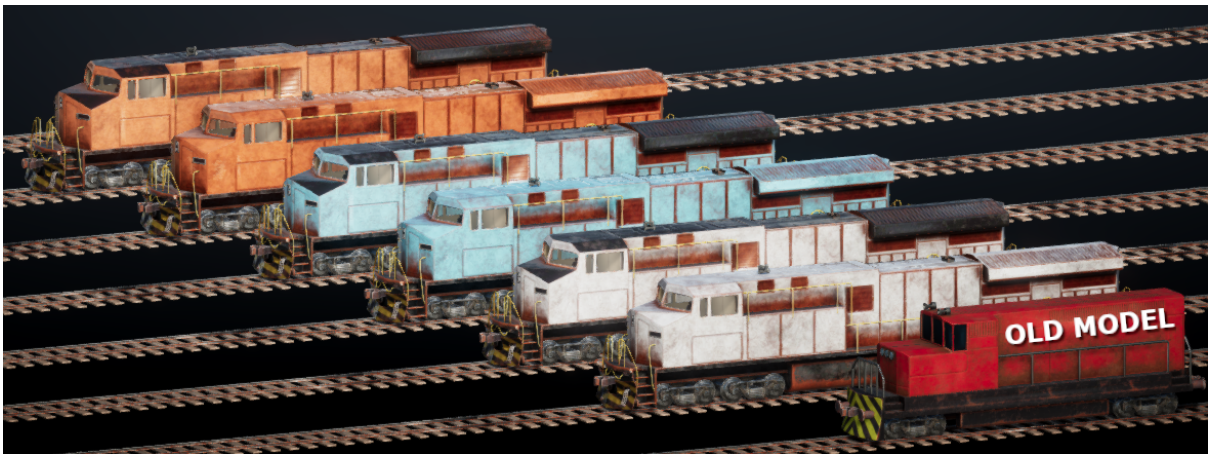
On the v3.0 update, those prefabs were replaced by the Railroad Builder, which is much easier to use and gives much more control over the railroad building process.

However, the rail segments prefabs were kept at the “Deprecated > Rails Segments” folder for those who bought any previous version of the package and wish to keep using them for any reason.

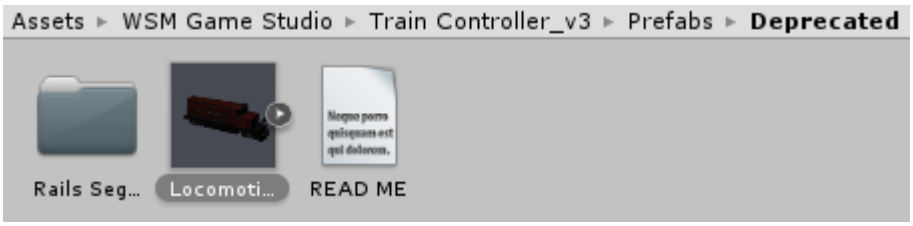


The Railroad Builder is now the official method for building railroads. For more info on how to use it, take a look at the Railroad Builder section.

On the v3.3 update, the default locomotive model was replaced by a more realistic model.



The old locomotive prefab can still be found inside the “Deprecated” folder.



Note: You can still use deprecated prefabs if you wish so, however, keep in mind that deprecated content will not be updated or guaranteed to be compatible with future versions.

3. Train Controls

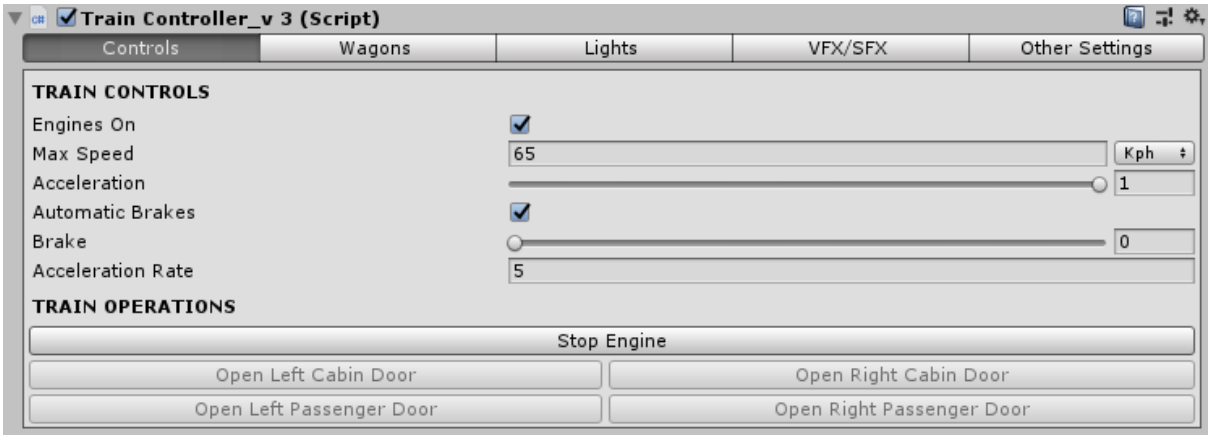
In this section you will learn how to control and monitor your train speed.
You can either have fully automated trains moving around your scene or you can have a player controlled train, either by keyboard input or Unity UI.

3.1. Automated Trains

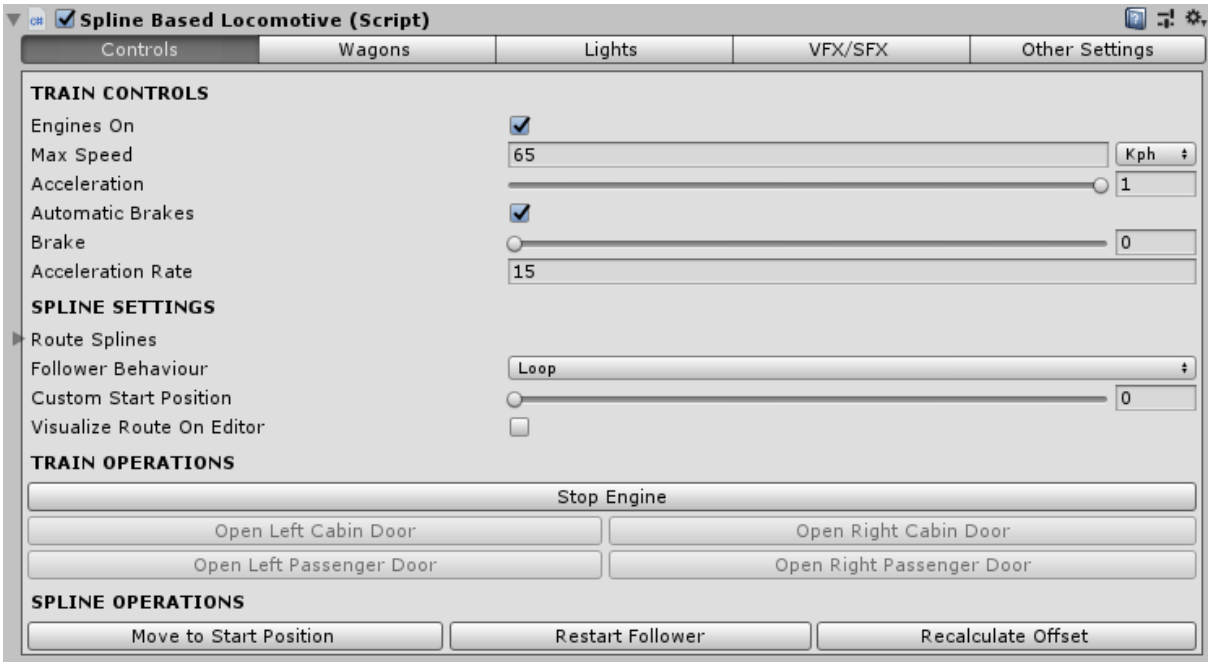
To add an automatic train at your scene, all you need to do is to set up the train behavior by adjusting the locomotive settings at the Unity Editor.

> **Note:** You can also change train settings automatically at runtime by using Control Zones

The **Train Controller** component is responsible for controlling physics based locomotives.



The **Spline Based Locomotive** component is responsible for controlling spline based locomotives.



The **Engines On** property is enabled by default, if you don't want your train to start moving automatically, you can disable it by unchecking it directly or using the button at the **Train Operations** Section.

> **Note:** Some operations are enabled only at runtime. For example, all door related operations.

The train speed is controlled by the **Max Speed Kph** and **Acceleration** properties of the Train Controller script.

The Max speed of physics based trains is a range between 0 kph and 105 kph (based on real world trains average max speed values). Spline based trains max speed is not limited, so if you need your train to move faster than 105kph, spline based trains are the way to go.

The acceleration property, defines train movement direction:

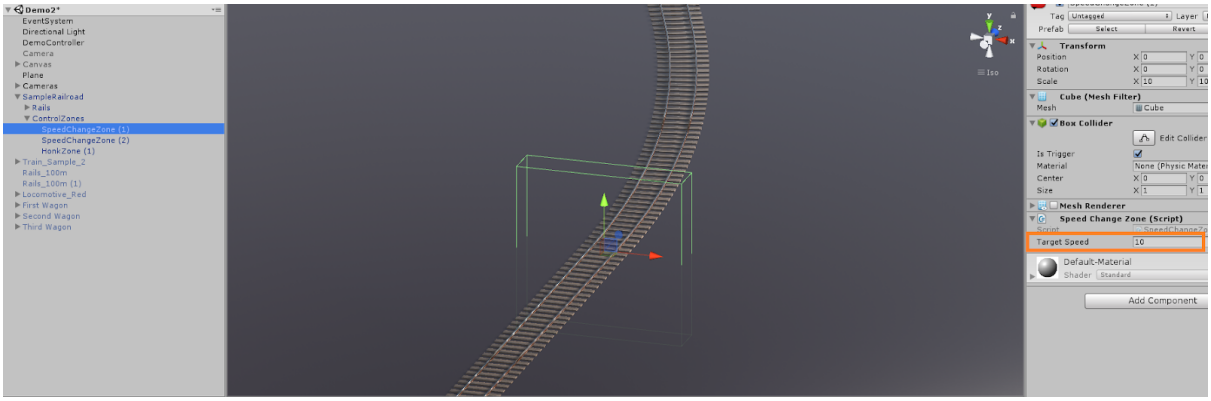
- **Acceleration 1:** Forward
- **Acceleration 0:** Stop
- **Acceleration -1:** Reverse

You can also use the **Brake** property to reduce train speed and stop the train. When the **Automatic Brakes** property is enabled, the brakes are applied automatically when the acceleration property is reduced.

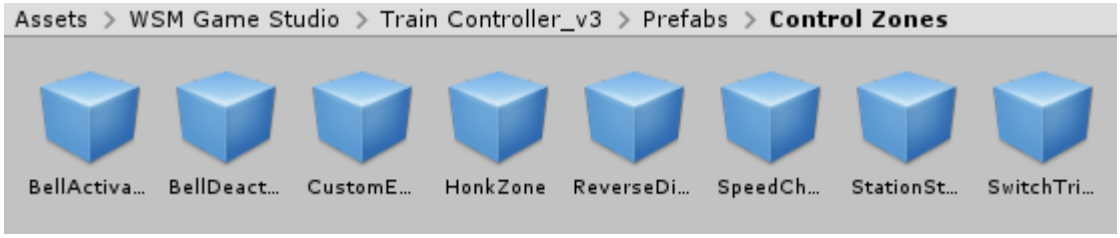
You can switch your train lights on/off in the Unity Editor by using the buttons available at the **Light Operations** section. By default, internal lights are on and external lights are off.

3.1.1. Control Zones

Control Zones are used to control the train behaviour on the railroad at runtime. This is very useful to automate your railroad.



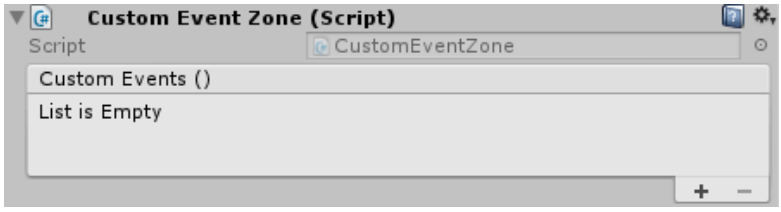
These zones are simple colliders that trigger events once the train passes through them.



- **Bell Activation Zones:** Activates the train bell
- **Bell Deactivation Zone:** Deactivates the train bell
- **Honk Zone:** Activates train horn
- **Reverse Direction Zone:** Reverse train direction
- **Speed Change Zone:** Changes train max speed value
- **Station Stop Zone:** Automatic station stop (see Train Stations section)
- **Switch Trigger Zone:** Switch rails direction (see Turnouts section)
- **Custom Event Zone:** Trigger custom user events

Note: It is recommended to use Speed Change Zones to reduce train speed before railroad curves and increase speed on straight segments. This will add more realism to your scene and increase physics based trains stability on curves.

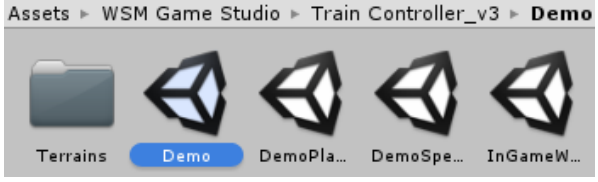
If none of the existing zones fits your project needs, you can use a **Custom Event Zone** to trigger custom events on your scene. This opens infinite possibilities on what you can do in your game.



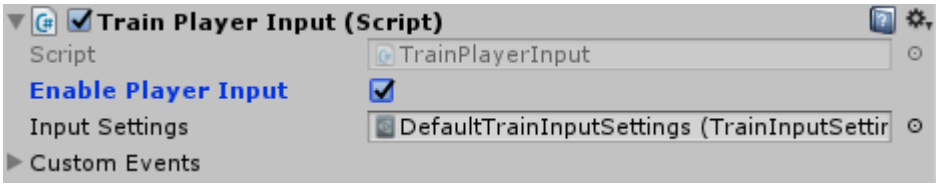
3.2. Player Input

There are two ways of controlling a train with player input: by using the Unity UI or customizable keyboard input.

The Unity UI method basically works by overriding the automatic train settings and calling locomotive public methods directly. Take a look at the first **Demo** scene for a sample on how to control your train using the Unity UI.



To get keyboard input, you need to enable the **Train Player Input** component on your locomotive.

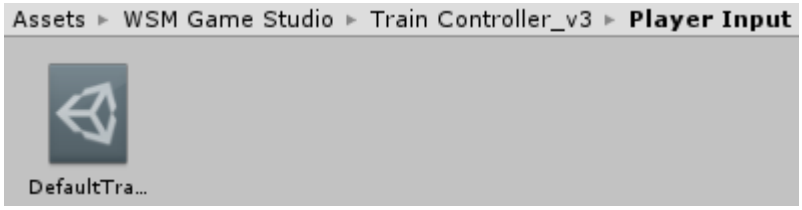


Note: For Unity UI input, make sure the **Enable Player Input** property is disabled. This property is intended to be used for keyboard input only.

This component is responsible to process player input and trigger train default and custom events. The keys configuration is defined by the Input Settings and can be easily customized. You can even have multiple trains, with distinct inputs being controlled at the same time.

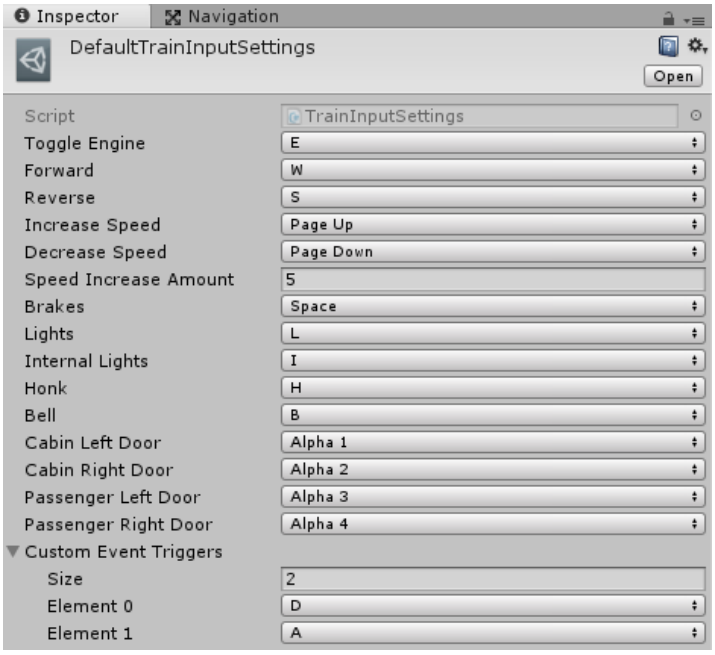
3.2.1. Custom Input Settings

The default input settings are located inside the **Player Input** folder.



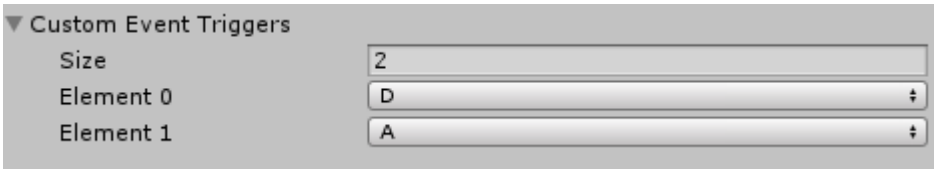
Note: Train Input Settings are stored inside Scriptable Objects, which means, you can create your own custom input settings either by duplicating the file or by using the Unity Editor creation menu:
Right Click > Create > WSM Game Studio > Train Controller > Train Input Settings

In the image below, you can see the default key input settings. You also see this in action at the **DemoPlayerInput** demo scene.

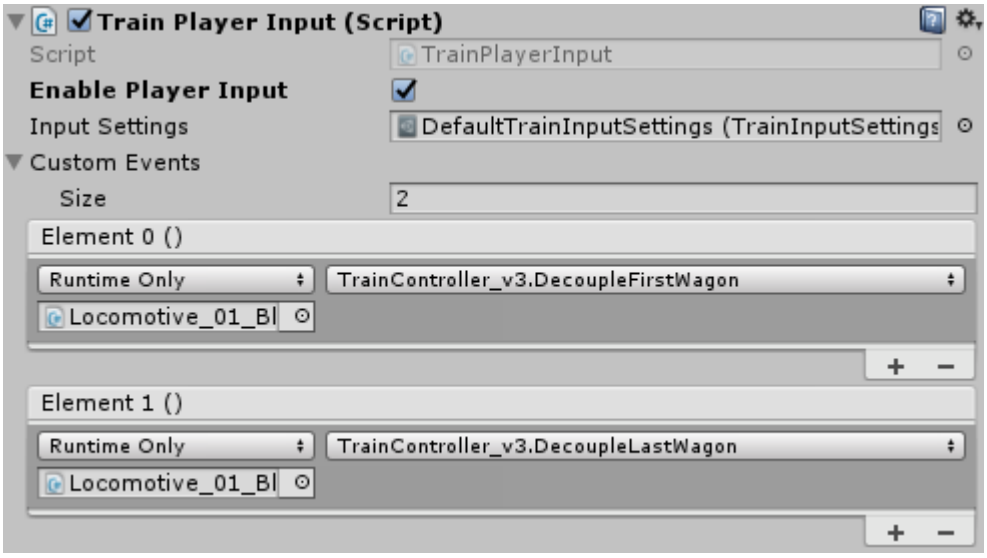


3.2.2. Custom Events

Besides the default train input, you can also create custom events triggered by player input. The key mapping for these events must be added on the **Custom Event Triggers** property of your train input settings file.



The corresponding events must be included at the **Custom Events** property of the Train Player Input component attached to your locomotive.

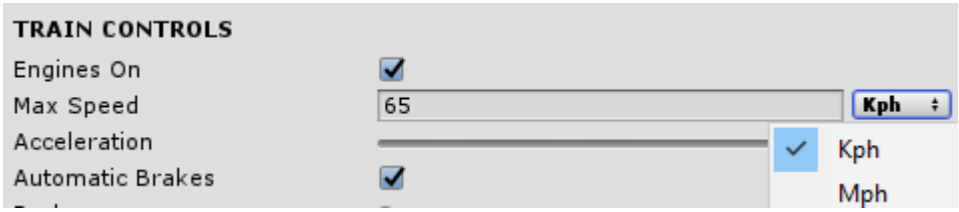


Note: You can have as many custom events as you wish. Just make sure to fill the **Custom Event Triggers** on your input setting file and the **Custom Events** property of the Train Player Input component attached to your locomotive.

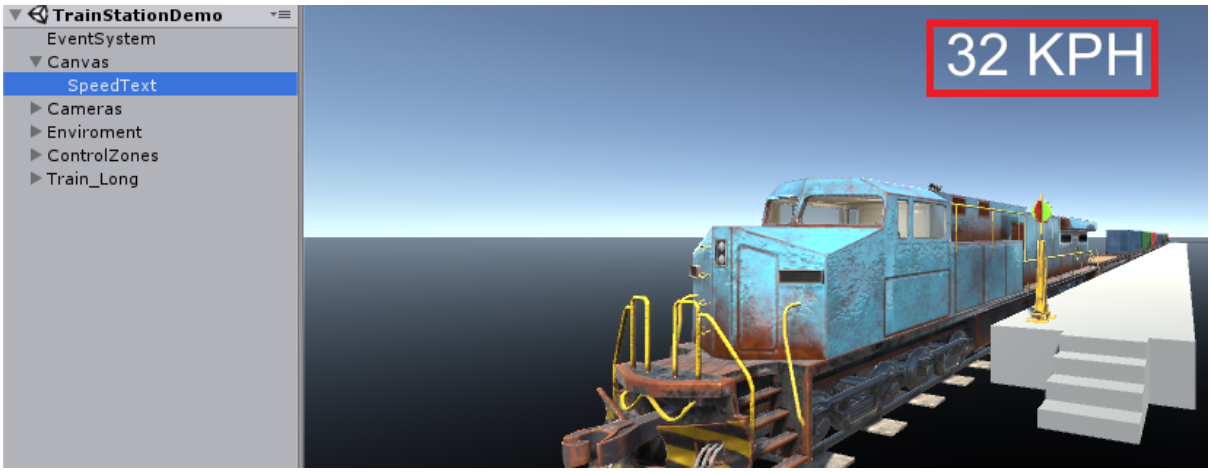
3.3. Train Speed Monitor

You can monitor train current speed, by using the **Train Speed Monitor** attached to the locomotive prefab.

You can print the current speed value to a Text field, just Drag & Drop your text to the “Output Text” property. The speed unit can be selected at the locomotive component, next to the Max Speed property.



In all the Demo scenes included in the project, the speed is printed to the **SpeedText** label, as you can see below.



4. Building a Train

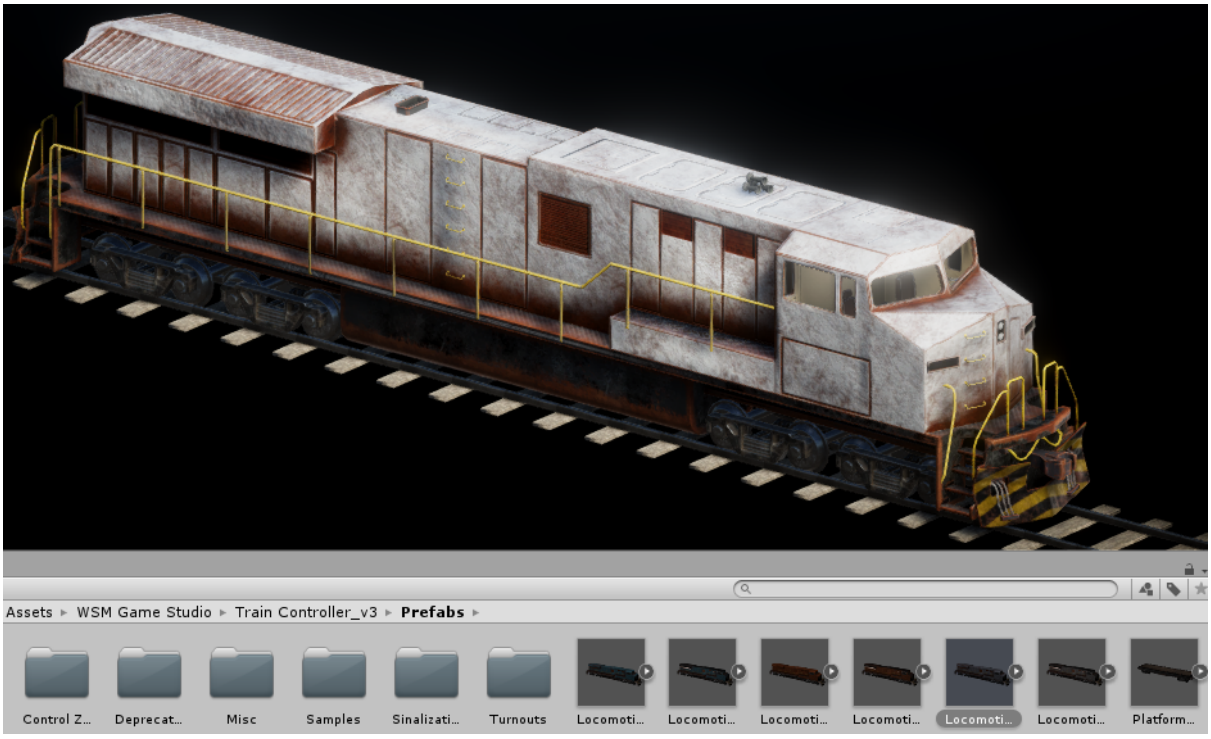
In this section you will learn how to build your custom train using the locomotive and wagons prefabs.

4.1. Connecting Wagons (Editor)

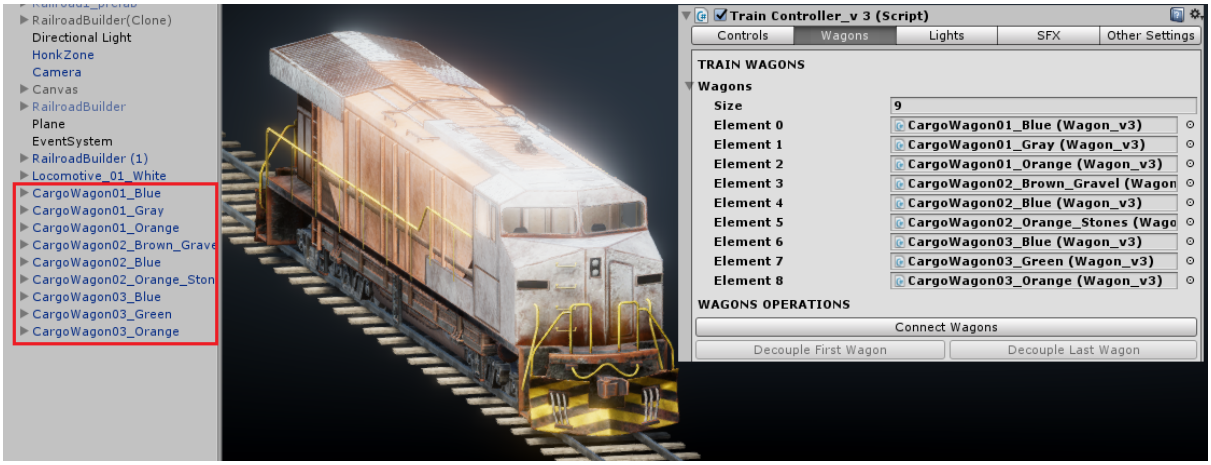
Additional wagons were used in this sample (Available Here)

To start building your train, drag and drop the locomotive prefab on your scene.

> **Note:** Alternatively, you can also use the Spawn Railway Vehicle operation of the Train Spawner to spawn the locomotive on top of the rails.

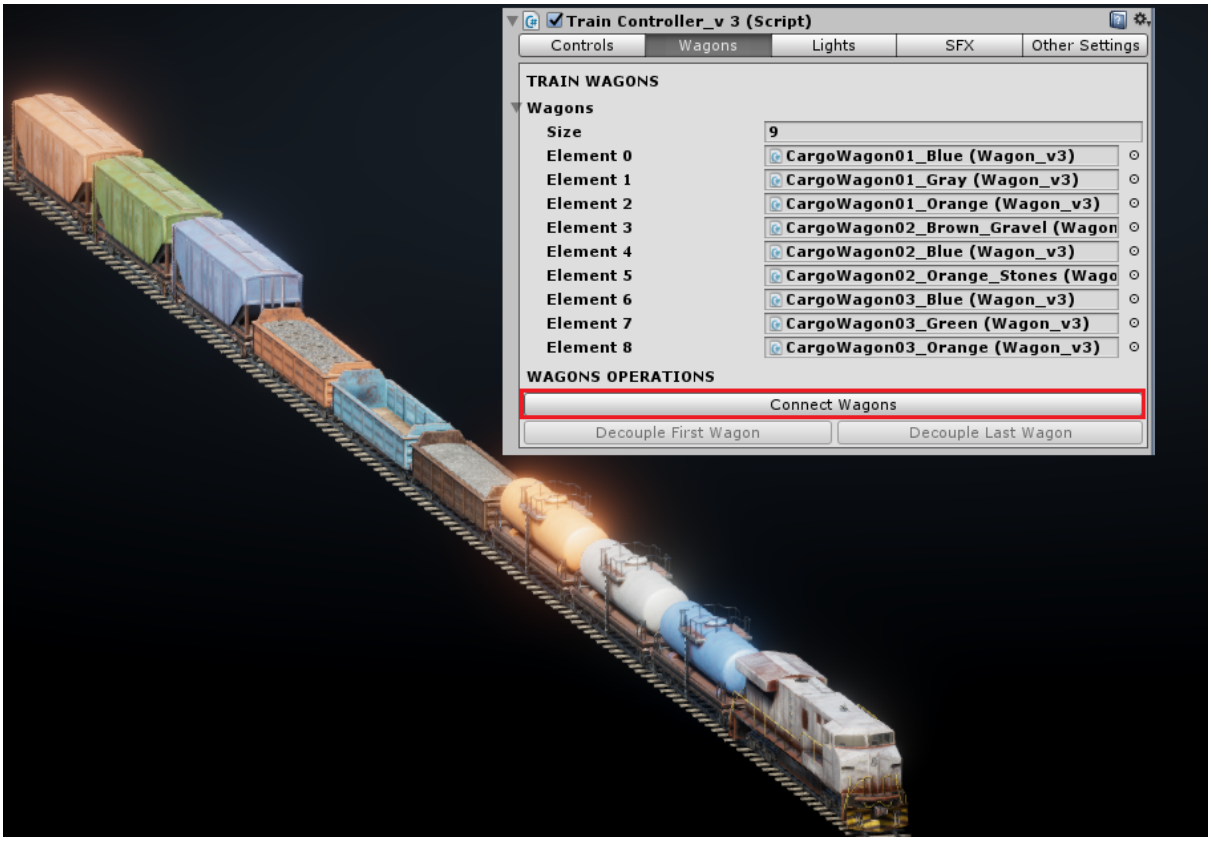


Drag & Drop wagons prefabs on your scene. Then, select the wagons on your scene and drag & drop them on the “Wagons” property of the locomotive in your scene.



Note: For physics based locomotives the wagons property is located under the wagons tab of the TrainController component. For spline based locomotives, the wagons property is located under the Wagons tab of the Spline Based Locomotive component.

For physics based locomotives, click the **Connect Wagons** button at the **Wagon Operations** section and the wagons will be repositioned automatically behind the locomotive, following the wagons list order.

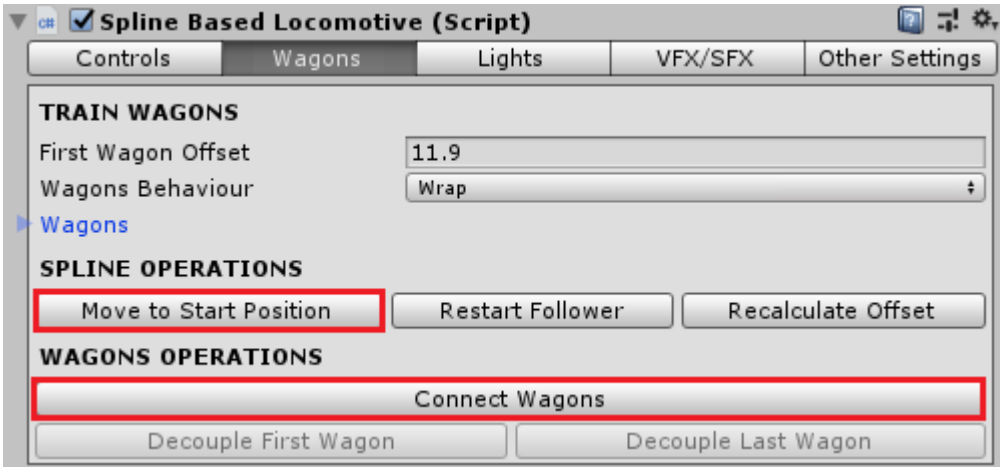


Note: Physics based wagons will connect on a straight line behind the locomotive and will not follow railroad curves. Physics based trains use colliders to follow railroad curves and currently it is not possible to calculate the precise position of the wheel trucks for the wagons on curved railroad sections.

If you're using physics based trains on your project, make sure your locomotive is positioned on a straight rail section before connecting the wagons.

Spline based wagons are not affected by this limitation and will follow railroad curves when connecting.

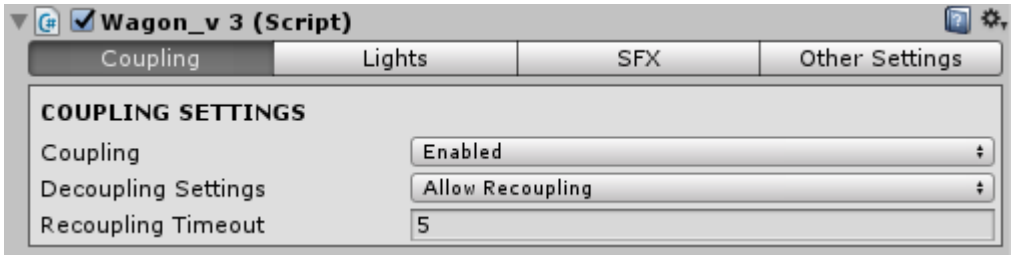
For spline based locomotives, you can click either the **Connect Wagons** or **Move to Start Position** button at the **Spline Operations** section and the wagons will be repositioned automatically behind the locomotive.



Note: the Route Splines property of your spline based locomotive must be assigned in order for the wagons to connect properly. Also, the "Move to Start Position" operation will also move the locomotive to the current selected route, before repositioning the wagons.

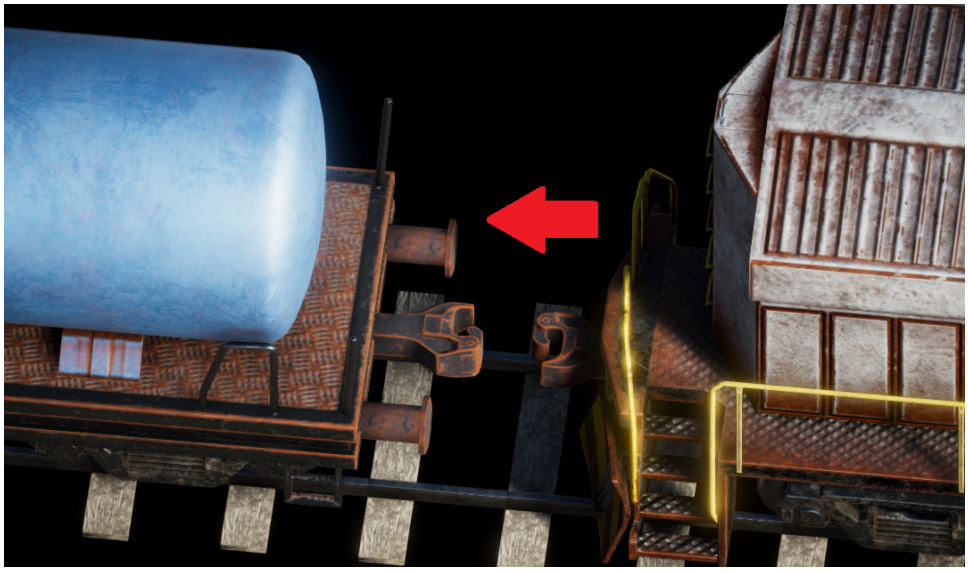
4.2. Connecting Wagons (In-Game)

By default, in-game coupling is enabled for all wagons prefabs. You can check the couple settings under the **Coupling** tab in the Wagon component attached to each wagon prefab.

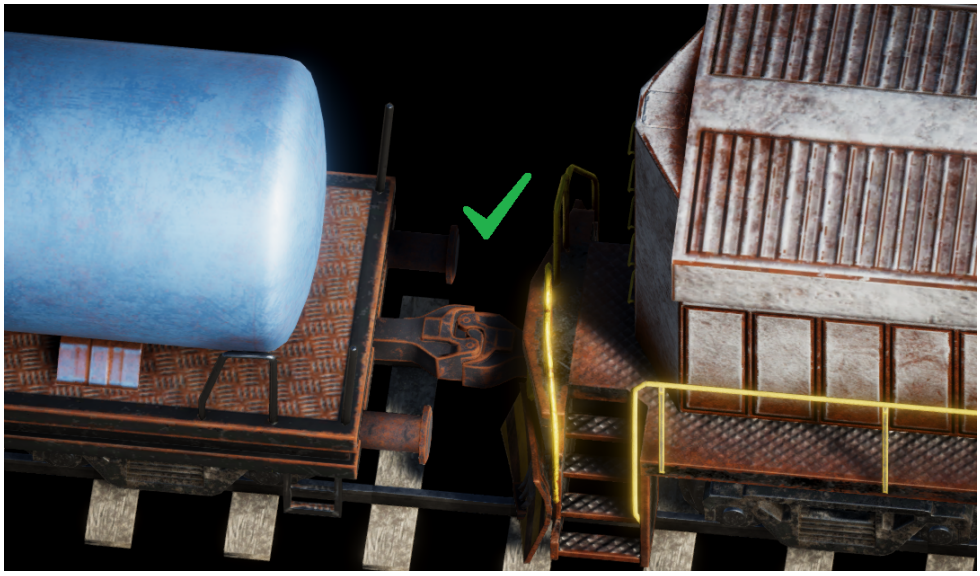


Here, you can enable and disable wagon coupling and configure if this wagon should be allowed to be re-coupled after being decoupled from the train for the first time

To couple a wagon to your train in-game, all you need to do is use the reverse gear and slowly approach the wagon until you hear the connection SFX feedback.



After connecting, the couplers should look like this:

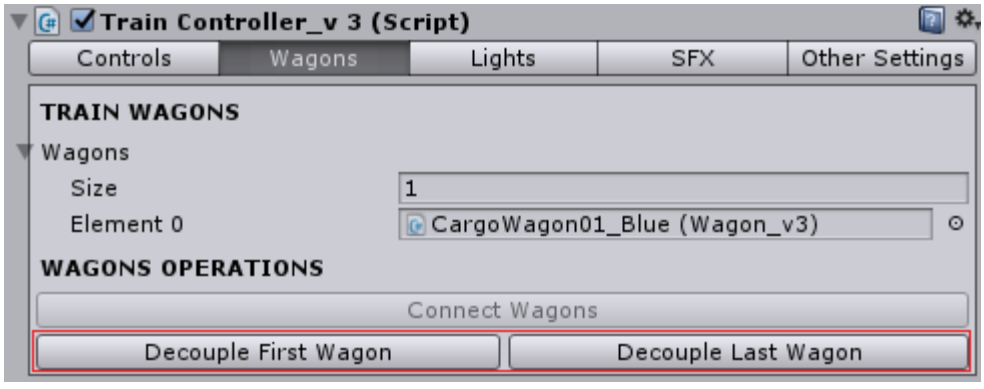


You can connect as many wagons as you wish to your train. Everytime couplers collide, they will connect, provided the coupling is enabled for that wagon.

Note: Physics based trains couplers are universal, so they can be connected from either side. You can also use the locomotive front coupler to connect to a wagon if you wish so.

You can also decouple wagons from your train in-game. You can either decouple the first, the last or any other wagon.

While in the Unity Editor, you can test wagon decoupling using the buttons on the **Wagons Operations** section.



If you have a train controlled by player input, using the default input settings, you can use D and A keys to decouple the first and last wagons respectively.

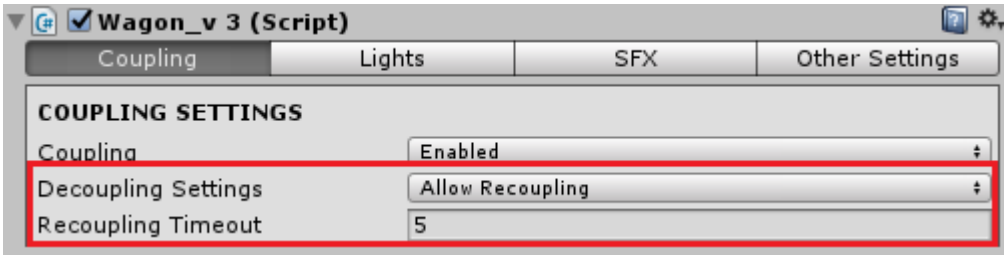
Note: The decoupling methods are configured on the Custom Events at the Train Player Input component. This is a sample on what can be done by using custom events.

You can also call the decoupling methods directly if you wish so. They are public methods located at the locomotive script (TrainController script for physics based trains and SplineBasedLomotive script for spline based trains).

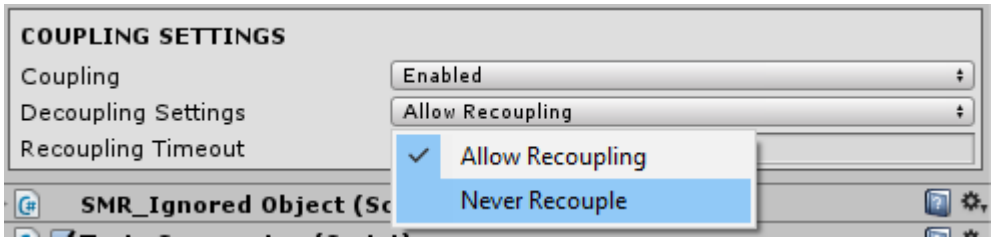
- **DecoupleLastWagon**()
- **DecoupleFirstWagon**()
- **DecoupleWagon**(int index)

> **Note:** When decoupling wagons by **index**, keep in mind that the first wagon is equivalent to index 0, following the .NET collections index logic.

After decoupling a wagon, if recoupling is enabled, you need to move away to separate the couplers before the **recoupling timeout** is elapsed (in seconds), or it will recouple the wagon again automatically.



If you don't want to recouple again for any reasons, you should change the **Decoupling Settings** to **Never Recouple**.



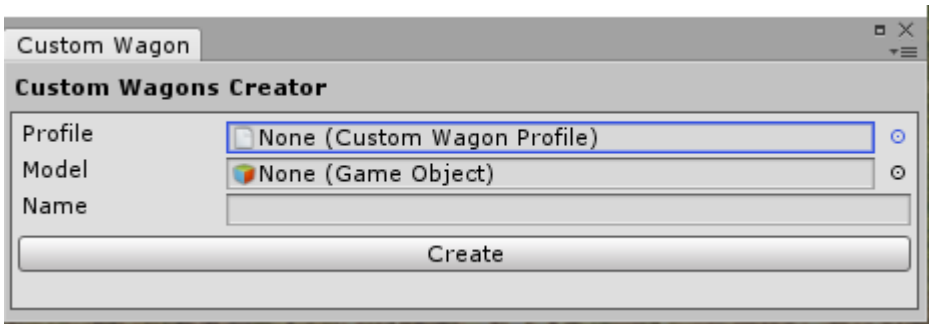
5. Custom Wagon Creator

Wagons and locomotives are complex prefabs, made specifically to be compatible with the rails generated by the Railroad Builder. Due to this complexity, it can be tricky to manually customize prefabs and ensure they will work properly.

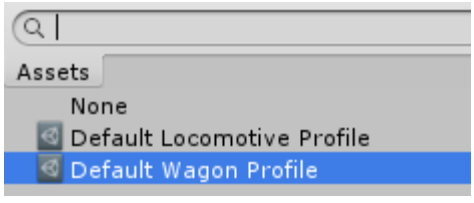
The custom wagon creator was introduced to increase client customization options. It can be used to create multiple custom wagons and locomotives, based on reusable user customizable profiles.

For example, let's assume you have a collection of wagon models you wish to use on your game. By using the custom wagon creator, you can quickly generate multiple customized wagons with your models.

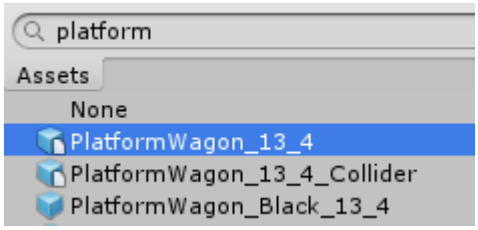
The Custom Wagon Creator is located in the Unity Editor, under **Window > WSM Game Studio > Custom Wagons Creator** menu. Alternatively, it can also be located under **WSM Game Studio > Train Controller > Utilities > Custom Wagons Creator**.



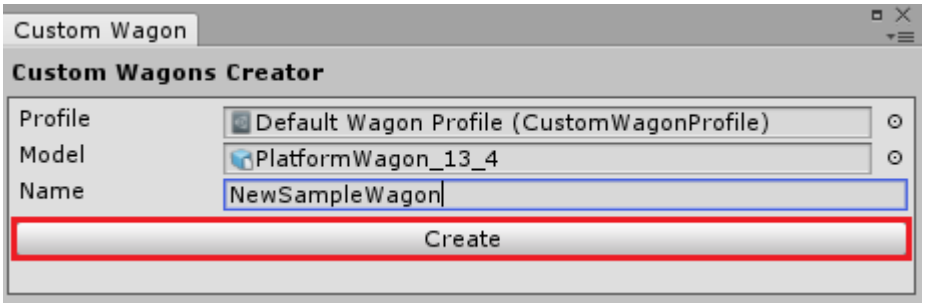
To create you custom wagon, first select the desired profile:



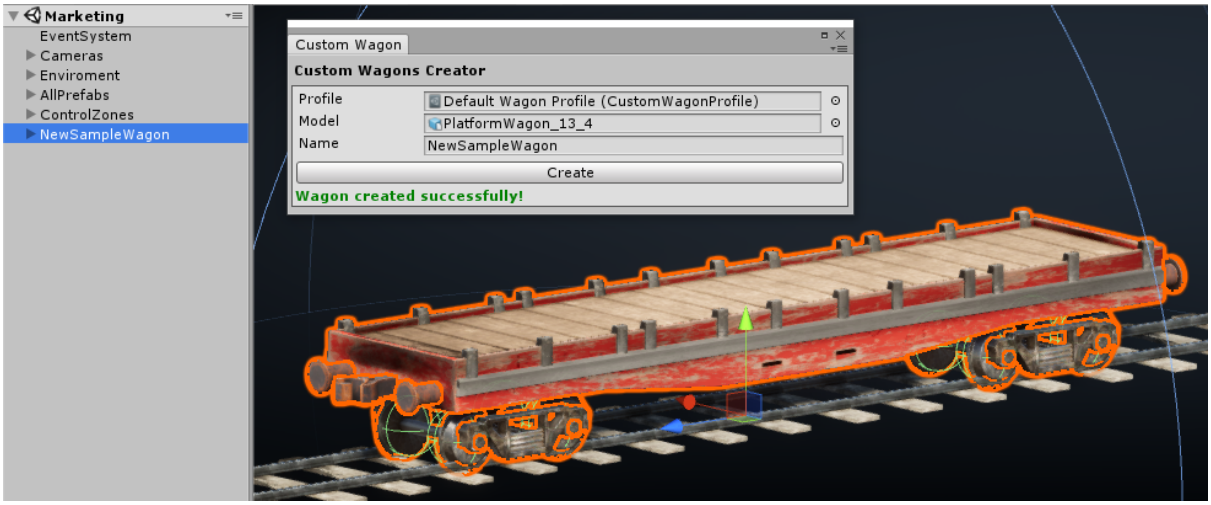
Then, select your custom model:



Select a name for the object and click on “Create”



The custom wagon will be created on your scene.



Verify if the wagon is working properly, then drag & drop it to your preferred folder to save it as a prefab.

Note: You can also create custom locomotives. The type of object being created is defined on the selected profile (See the Custom Profiles section for more details)

5.1. Custom Profiles

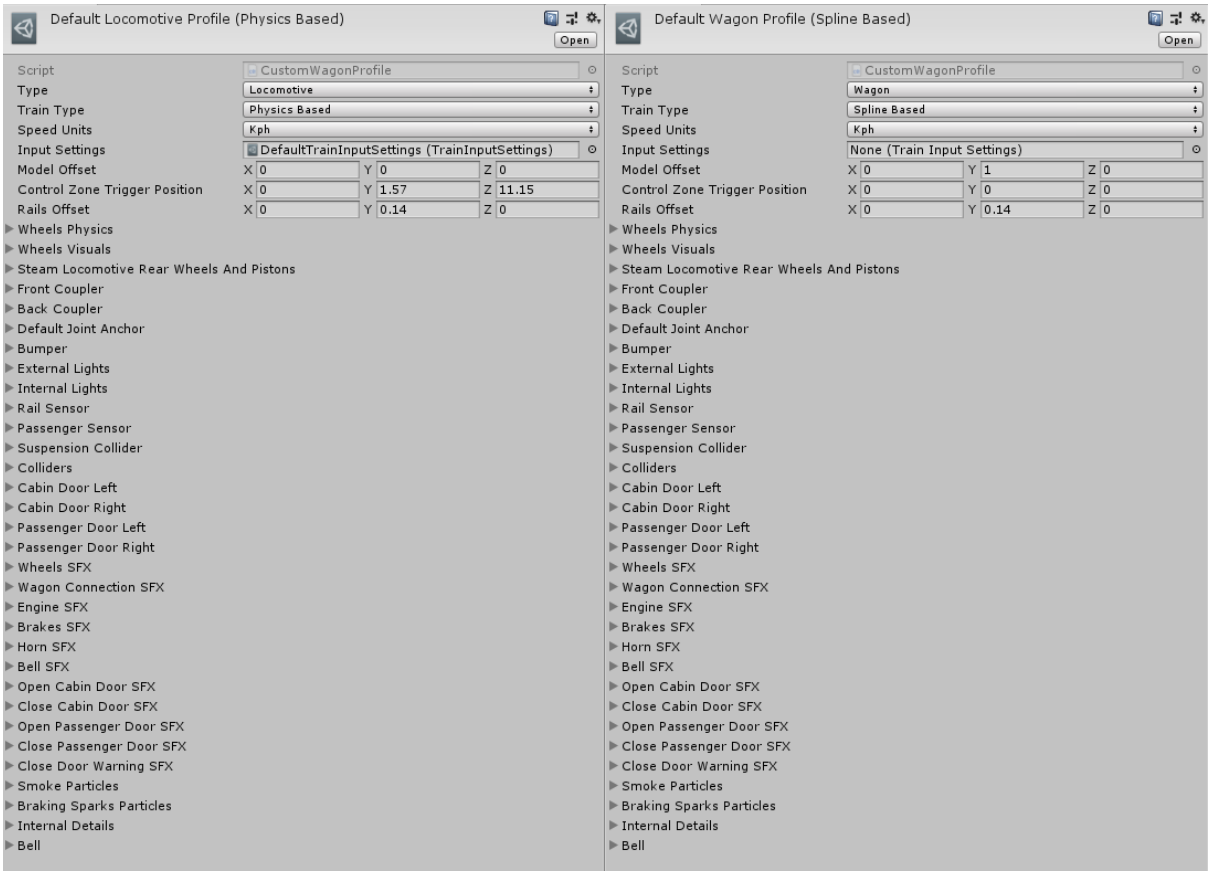
Custom Wagons Profiles defines the type and settings of the created object.

You can think of a profile like a blueprint that defines all the essential components that needs to be included in the prefab to ensure it will work properly.

Profiles can be used for mass creation of custom locomotives or wagons prefabs. For example, if multiple wagons models shares the same configuration, like length, wheel positions, etc... You can use the same custom wagon profile to turn them into ready to use wagon prefabs.

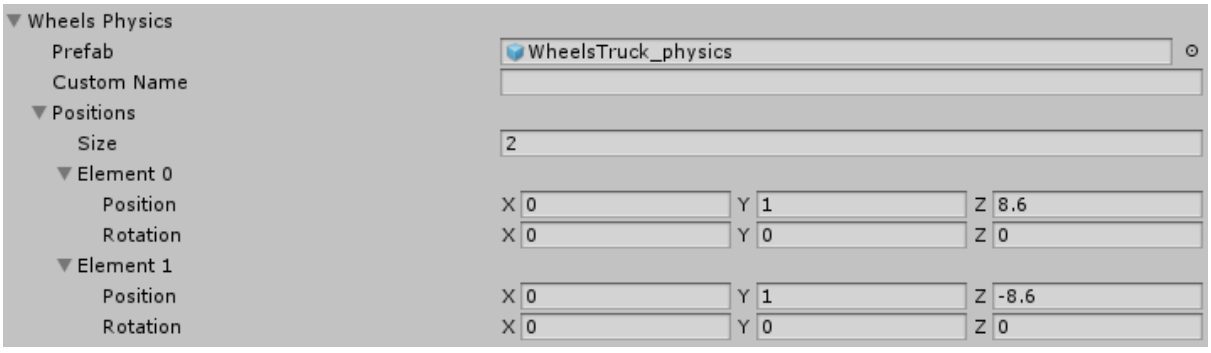
There are four sample profiles included in this package, for creating physics based and spline based wagons and locomotives.

Note: Custom Wagons Profiles are Scriptable Objects, which means, you can create your own custom profiles either by duplicating the existing files or by using the Unity Editor creation menu:
Right Click > Create > WSM Game Studio > Train Controller > Custom Wagon Profile

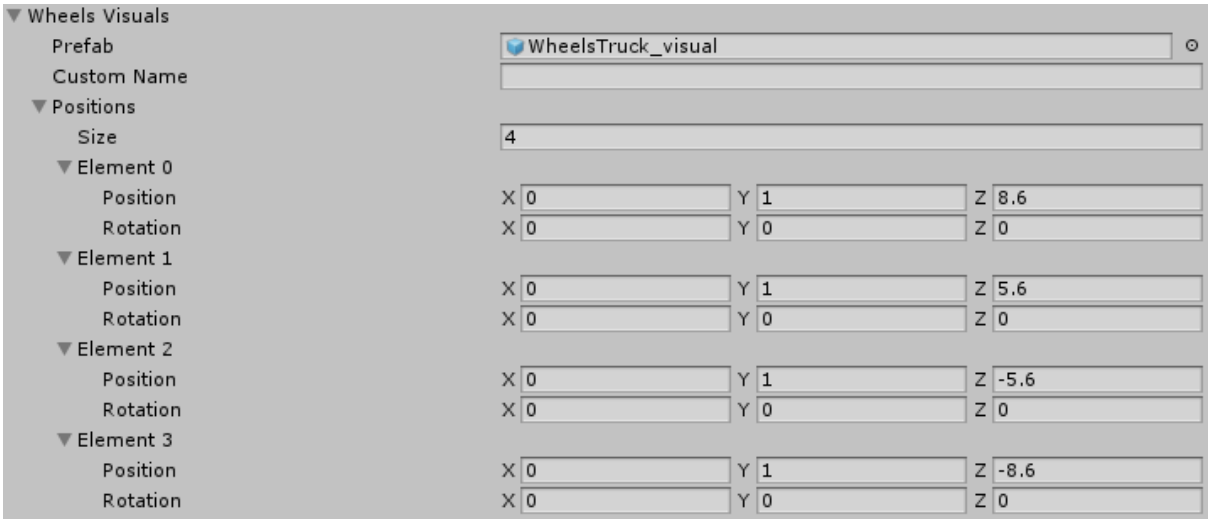


As can be seen in the sample image above, profiles are very complex Scriptable Objects that contain references to all the Parts that will compose your custom wagon. It also defines which type of object will be created (locomotive or wagon).

For example, the **Wheels Physics** section, holds a reference to the prefab responsible for simulating train wheels physics, and the **Positions** array defines how many and where the instances should be spawned.



The **Wheels Visuals** section, holds a reference to the prefab responsible for simulating the wheels turning animation, how many and where the instances should be spawned.



Note: As you can see in the sample image below, there are four visual wheels, but only two physics wheels on the physics based locomotive profile. That's because there is no need for more than two physics wheels to keep the locomotive on track, even though the locomotive has four sets of wheels.

5.2. Wagons/locomotive Parts

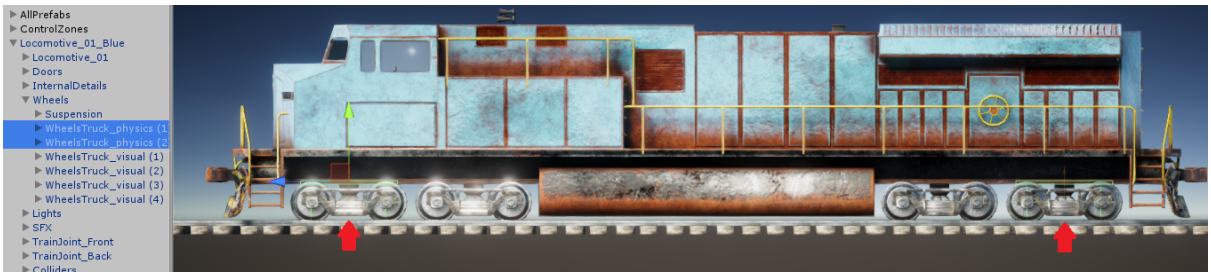
Parts are the essential components needed to create working locomotives and wagons. The default parts prefabs are stored inside the **Wagons Creator > Parts** folder.

There are many different parts, but most of them are self-explanatory. In this section, you will know more about the mandatory parts and how to set them properly on your custom profile.

The mandatory parts are:

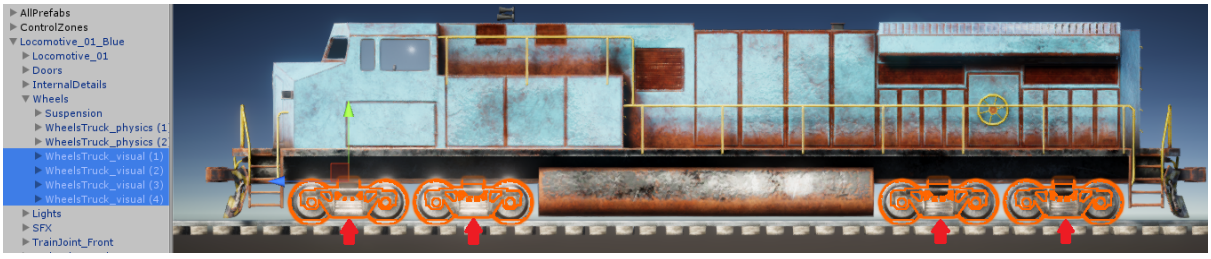
- Physical Wheels **(physics based trains only)**
- Visual Wheels
- Front and Back Couplers
- Default Joint Anchor **(wagons only)** **(physics based trains only)**
- Rail Sensors **(physics based trains only)**
- Suspension Colliders **(physics based trains only)**
- Colliders
- Wheels SFX
- Wagon Connection SFX **(wagons only)**
- Engine SFX **(locomotives only)**
- Brakes SFX **(locomotives only)**
- Horn SFX **(locomotives only)**
- Bell SFX **(locomotives only)**

Physical Wheels are components responsible for physically interacting with the rails and keeping the train on the track. It is recommended to always use only two physical wheels, one at each end (front and rear). Two is the minimum required and by keeping that way you ensure you won't have unnecessary rail friction.



Note: Starting from v3.3, the physical wheels components don't have wheels models being rendered on them anymore. Visual and physics elements were separated for better results.

Visual Wheels are components responsible for rendering the train wheels and the wheels rotation animation. You can have as many visual wheels as you want.

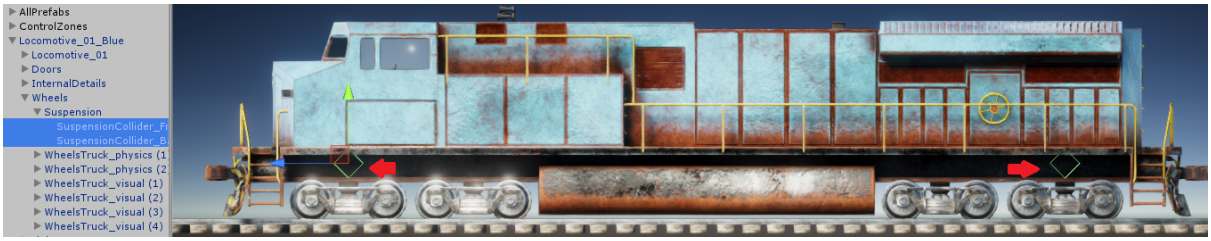


The **Front and Back Couplers** are components responsible for connecting wagons, either on the Unity Editor or in-game (Take a look at the Building a Train section for mode details).

The **Default Joint Anchor** is a component responsible for stabilizing wagons that are not connected to any train. This anchor was created as a workaround to fix a PhysX issue with disconnected joints.

Rail Sensors are components responsible to verify if the train is grounded and on rails. There are two of them (left and right), they must be located on the lower part of the first set of wheels.

Suspension Colliders are auxiliary components used to maintain train stability on terrain elevations. They must be located above the physical wheels components.



Colliders are prefabs containing the default Unity Colliders. If you intend on creating similar wagons that share common colliders, you can save them as prefabs and set them at your custom profile.

The **SFX** components are prefabs that contain Audio Sources with train related SFXs.

> **Note:** Default parts prefabs are intended to be used as a starting point for creating working locomotives and wagons. Therefore, it is **NOT** recommended to edit them directly.
>
> If you wish to take your customization to the next level, by changing wheels models or using different SFXs for example, you can create custom parts, by duplicating the existing parts prefabs and customizing them using the default Unity's prefab editing workflow.

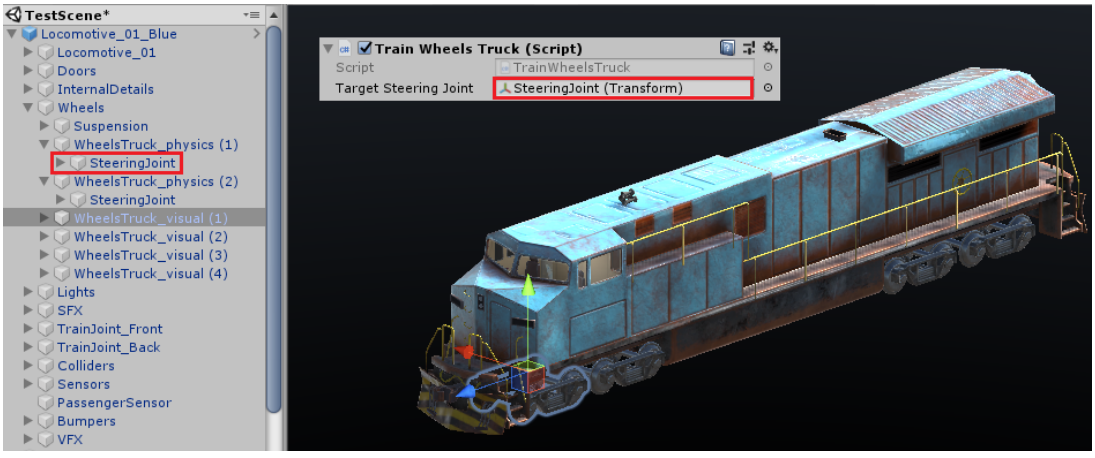
5.3. Manual Adjustments

Although the custom wagon creator already creates ready to use locomotives and wagons, it is recommended to test and validate the generated vehicles and make small manual adjustments if needed.

5.3.1. Train Wheels Truck

When working with physics based trains, it is possible to set up visual wheels trucks to steer accordingly to railroad curves. This feature works by copying the steering rotation of a physics wheel truck to a visual wheels truck. This process is controlled by the **Train Wheels Truck** component, which is also responsible for smoothing the rotation before applying it to the visual wheels trucks to achieve more realistic steering animation.

When creating physics based locomotives and wagons this feature won't be enabled by default. In order to enable it, select the visual wheels truck and manually drag a reference of which steering joint you wish to use as reference for steering.



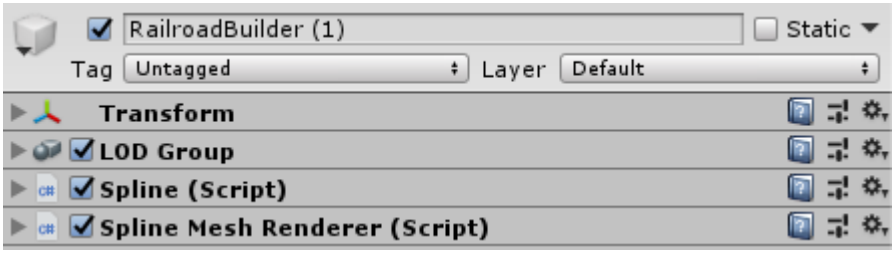
6. Building a Railroad

In this section you will learn how to build your custom railroad, using the Railroad Builder.

6.1. Railroad Builder

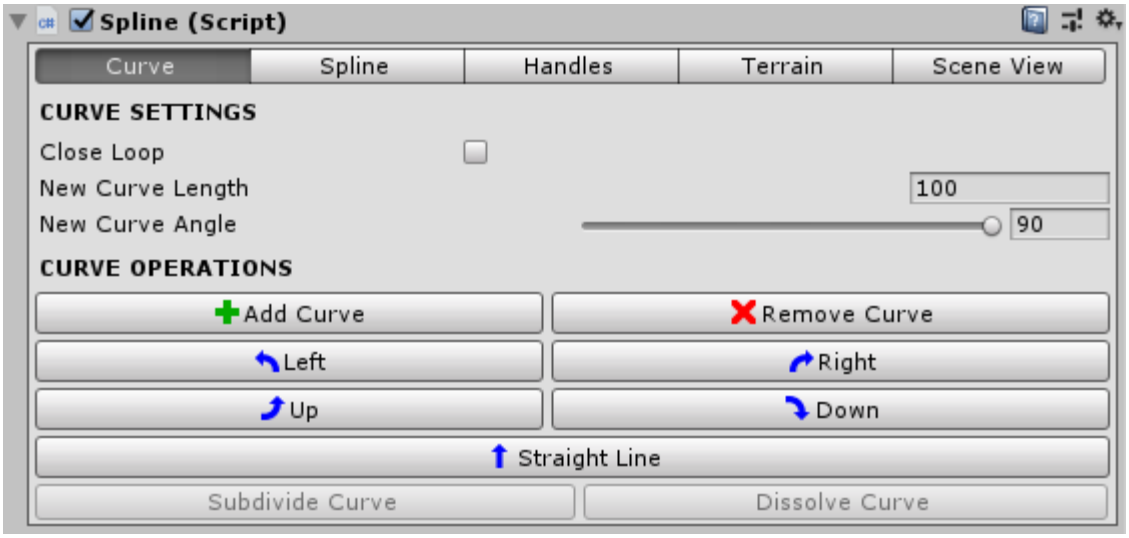
The Railroad Builder is a game object used for creating and editing railroads in the Unity Editor.

A Railroad Builder is composed of a LOD Group, a Spline component and a Spline Mesh Renderer component.



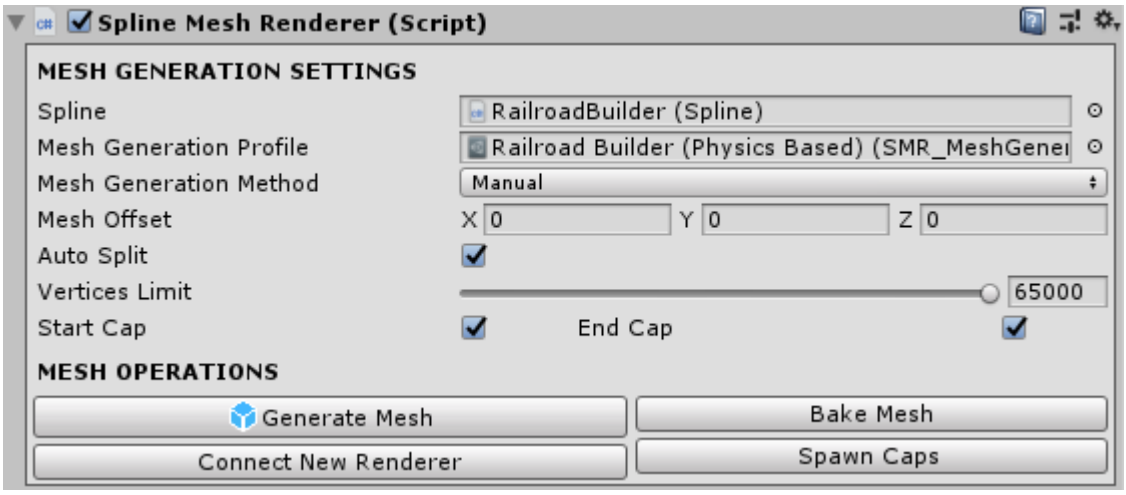
The **Spline** component is an editor extension for creating Bezier Splines on the Unity Editor. A Bezier Spline is a curved path (open or closed) composed by one or more Bezier Curves.

This component is responsible for creating and editing the railroad “path” in the Unity Editor.



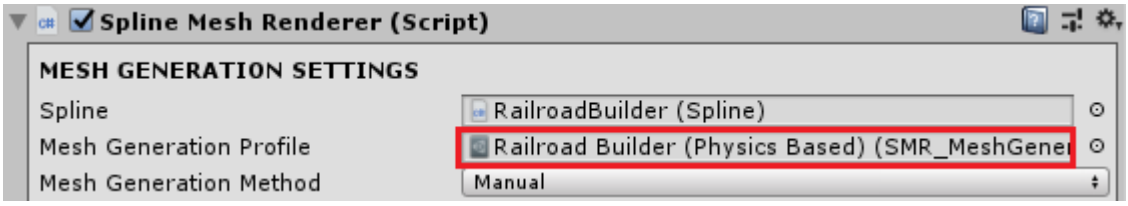
The **Spline Mesh Renderer** component is a level design tool to create dynamically generated meshes. It works by extruding a base mesh segment along a Bezier Spline.

This component is responsible for dynamically generating the rails on your scene.

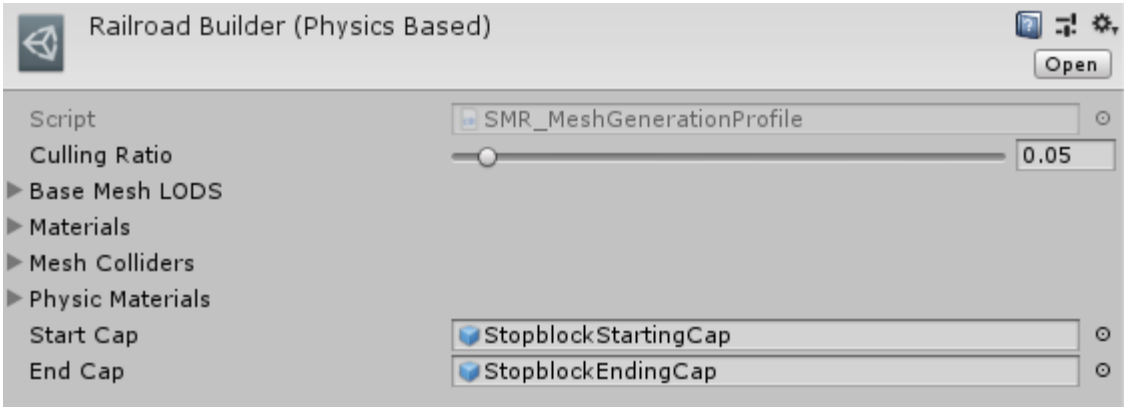


6.1.1. Rail Generation Profiles

The selected rail generation profiles define which type of rails will be generated by the Railroad Builder.



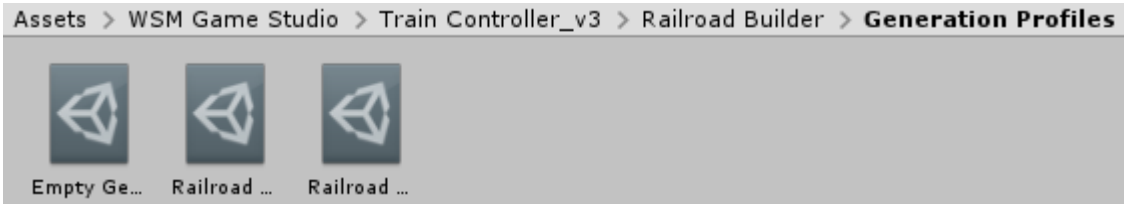
Profiles defines which base mesh segment will be extruded during generation, how many LODs will be used, what materials should be applied, if and what colliders should be used, what physics materials should be applied for each collider and if there is any caps prefabs available to close the start and end of the generated railroad.



Default generation profiles consists of:

- **Railroad Builder (Physics Based)**
  - Generate rails for physics based trains
- **Railroad Builder (Spline Based)**
  - Generate rails for spline based trains
- **Empty Generation Profile**
  - Auxiliar profile used for invisible rails sections

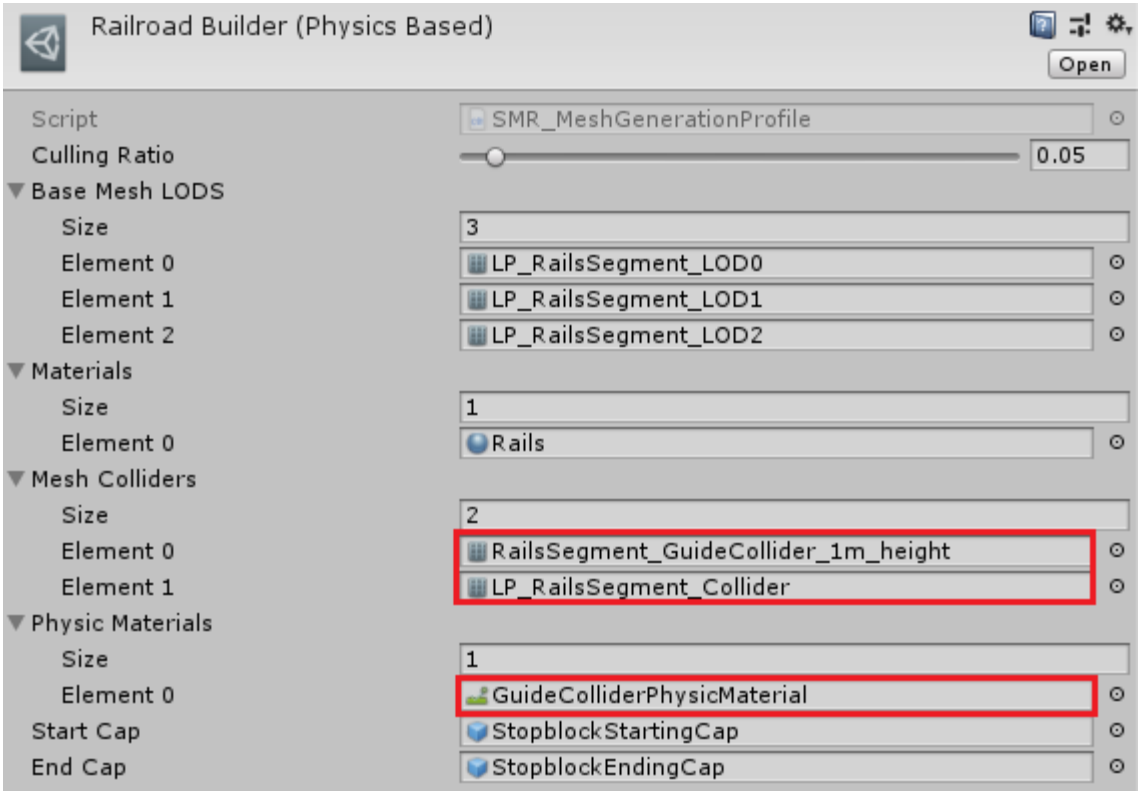
The default generation profiles are located under the “Generation Profiles” folder.



Note: **DO NOT** rename or move default generation profiles to another folder. Otherwise, they will not be assigned automatically when using the unity object creation menu to create a new railroad builder.

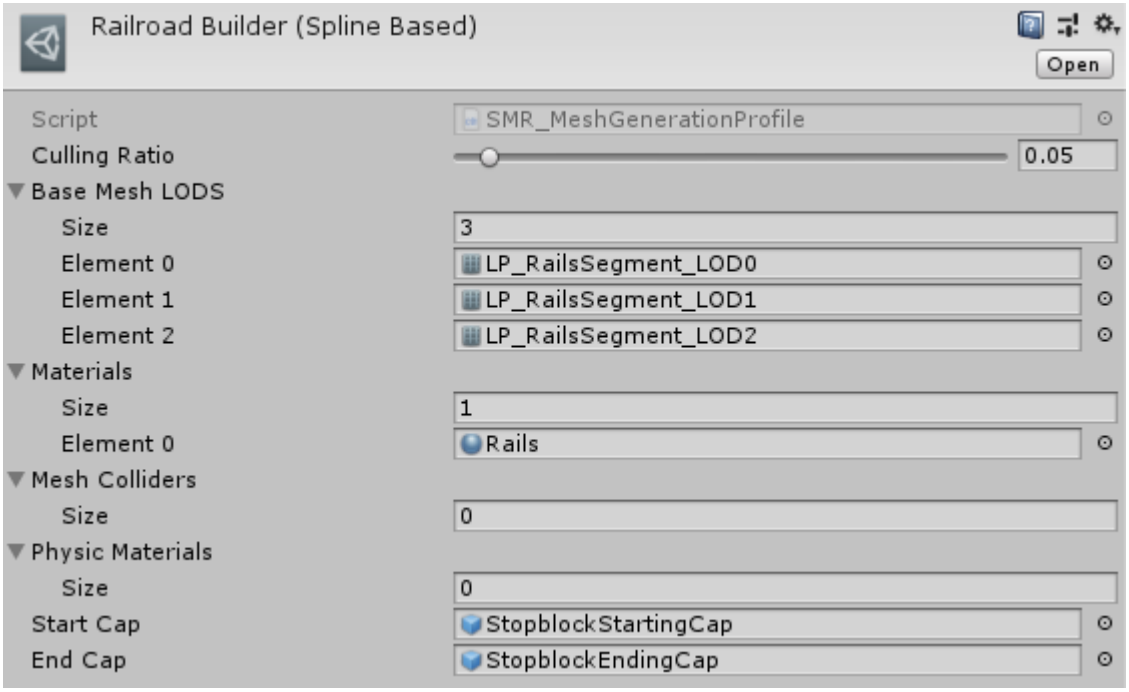
Since physics based trains rely on guide colliders to follow the tracks, physics based generation profiles must always include references to the default base mesh colliders.

A physic material must also be applied to reduce friction of the generated guide collider.



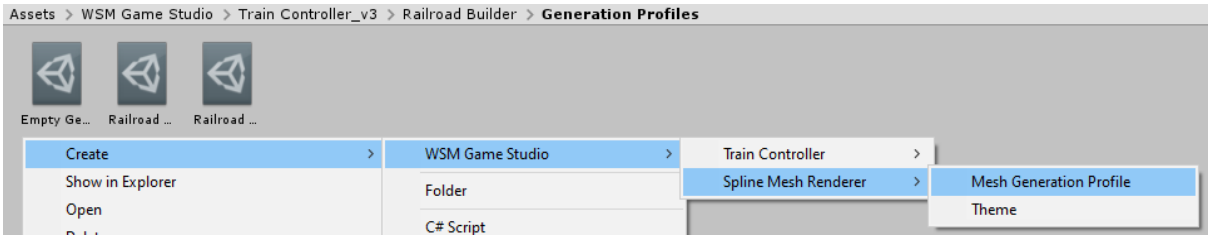
Spline based railroads don't require any colliders, which makes them the most performance friendly solution.

For performance optimization, it is highly recommended to not assign any mesh colliders and physic materials on Spline based generation profiles.



Note: Spline based trains can run on physics based railroads, but physics based trains cannot run on spline based railroads. Nonetheless, for performance sake it is not recommended to use physics based railroads for spline based trains.

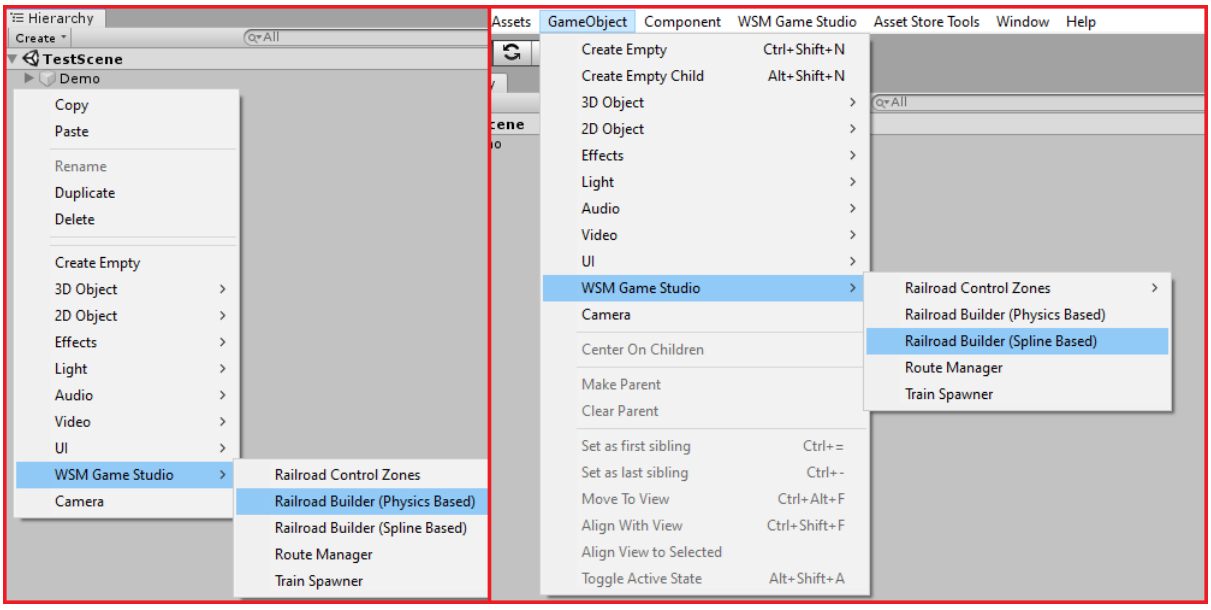
You can also create custom Generation Profiles by using the default file creation menu. Right click on the folder, then navigate to "**Create > WSM Game Studio > Spline Mesh Renderer > Mesh Generation Profile**"



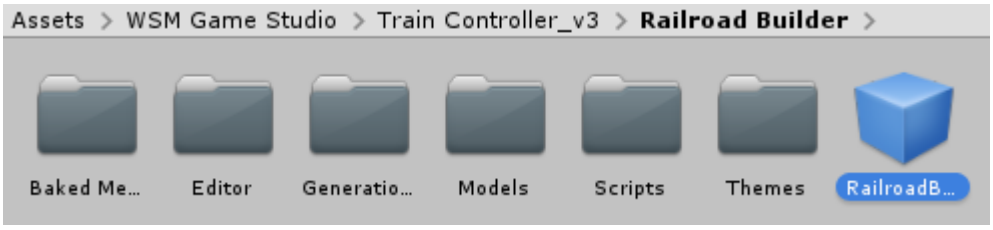
6.1.2. Creating a new railroad

You can create a new railroad on your scene by using the default object creation menu:

- **Hierarchy Window**
  - Right Click on Hierarchy window
  - WSM Game Studio > Railroad Builder
- **GameObject Menu**
  - Click on GameObject menu
  - WSM Game Studio > Railroad Builder



Alternatively, you can also Drag & Drop an instance of the RailroadBuilder prefab into your scene.

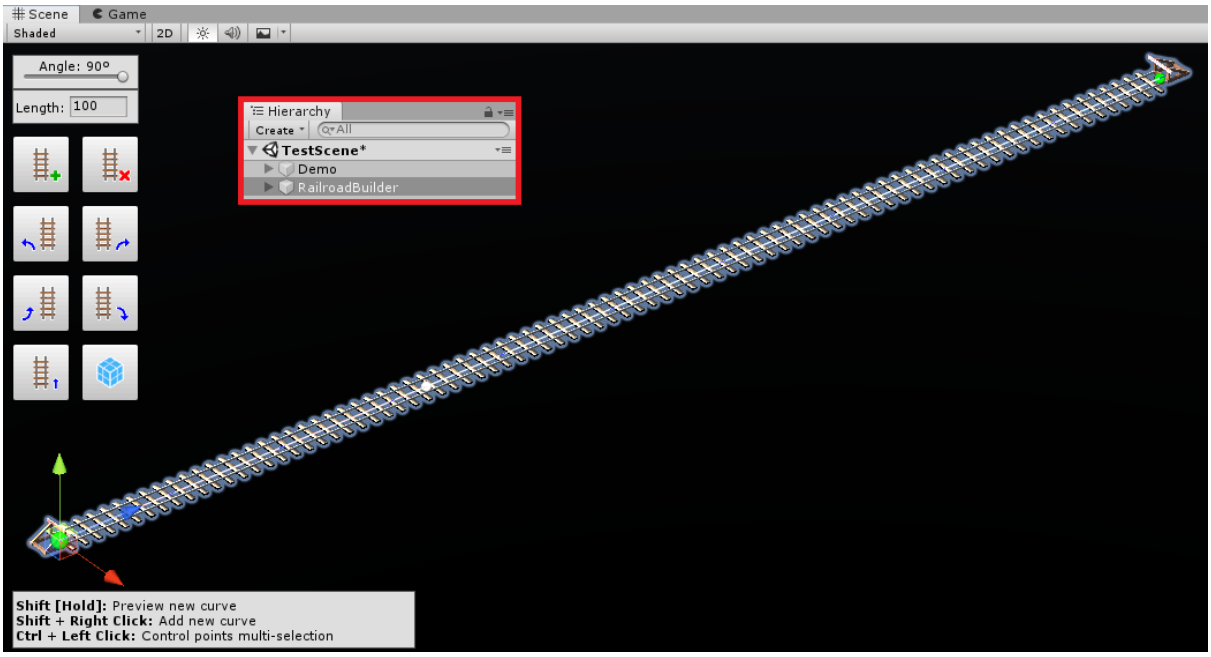


Note: Prior to version 3.4, this prefab was used to create new railroad builders instances. It is not needed for this purpose anymore, however it was kept for the sake of backwards compatibility.

You should never change this prefab directly or save any alterations made to it. It was created with the sole purpose of being used as a starting point.

If you are using a custom rail generation profile for your game, you can either use the object creation menu and change the profile manually, or, you can use the default railroad builder prefab to create a prefab variant that references your custom generation profile.

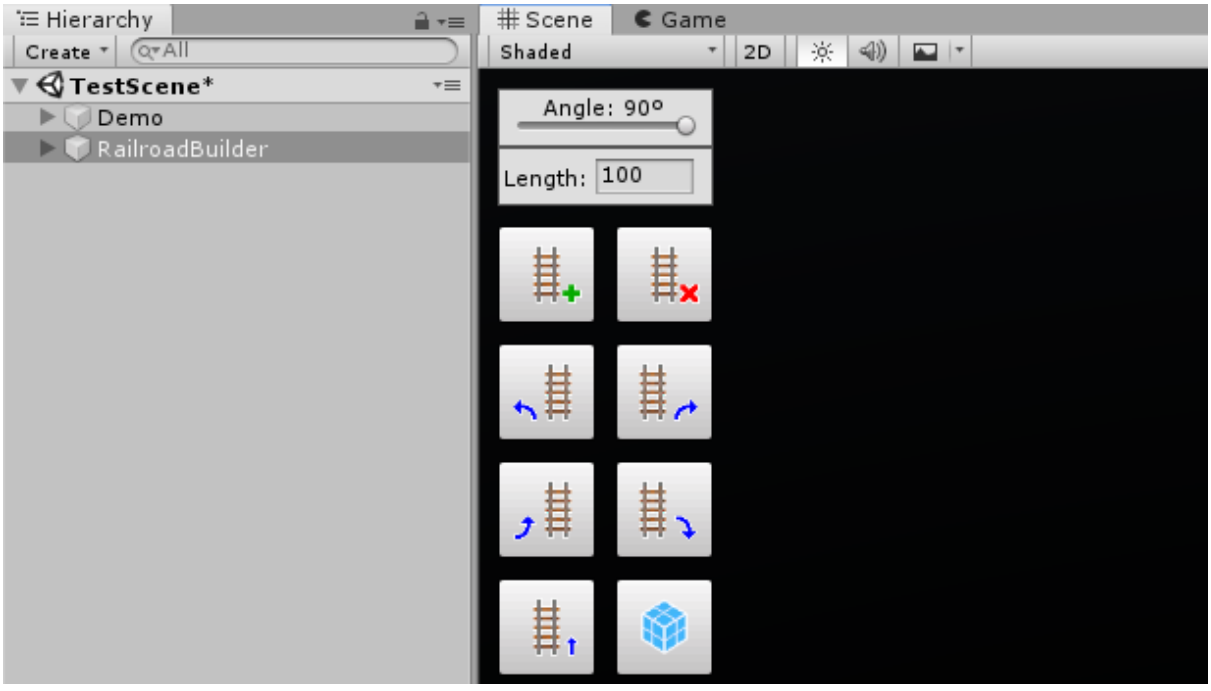
By using the object creation menu, the newly created railroad builder will automatically generate the rails based on the selected profile.



6.1.3. Quick Access Building Menu

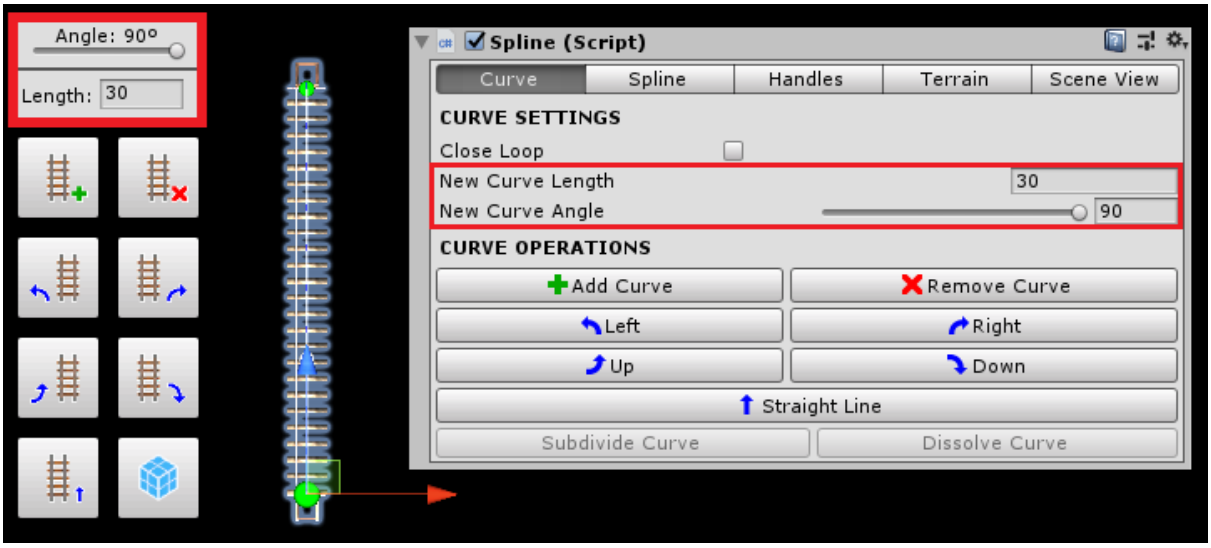
The Quick Access Building Menu was designed to be powerful and user friendly at the same time. It allows the user to quickly create complex railroads with a few button clicks, while maintaining mathematical precision for both curve length and angle.

When a railroad builder instance is selected on the Hierarchy, the quick access building access menu will appear on the top left corner of the Scene View.

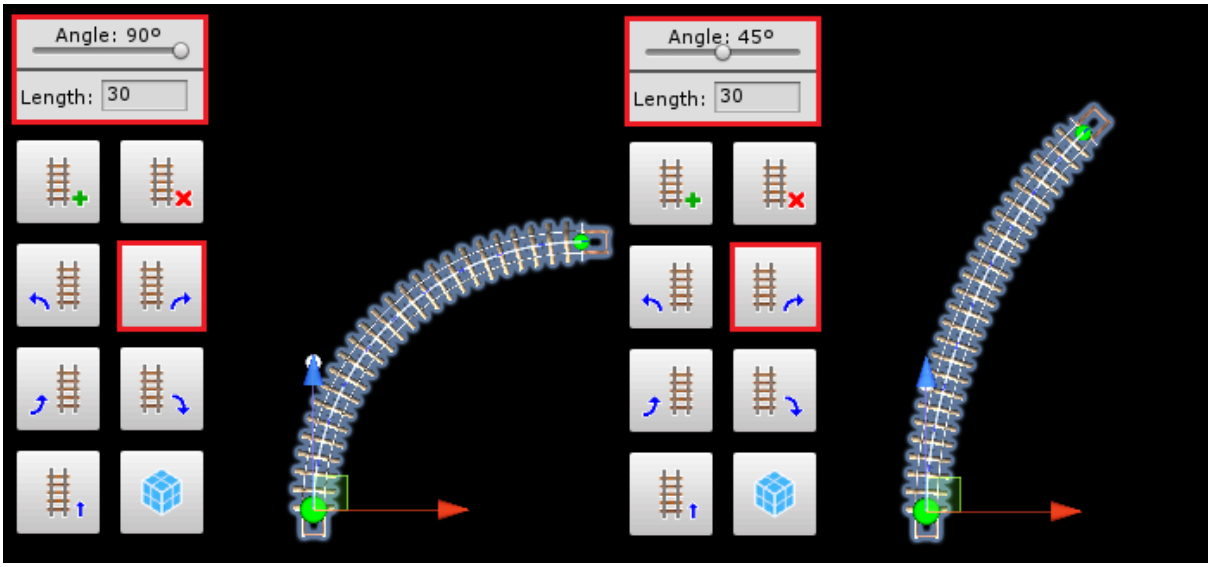


By using this menu, you can quickly create, delete and reshape curves to the desired length and angle.

The new curve length and angle can be adjusted on the quick access menu and/or the unity inspector.



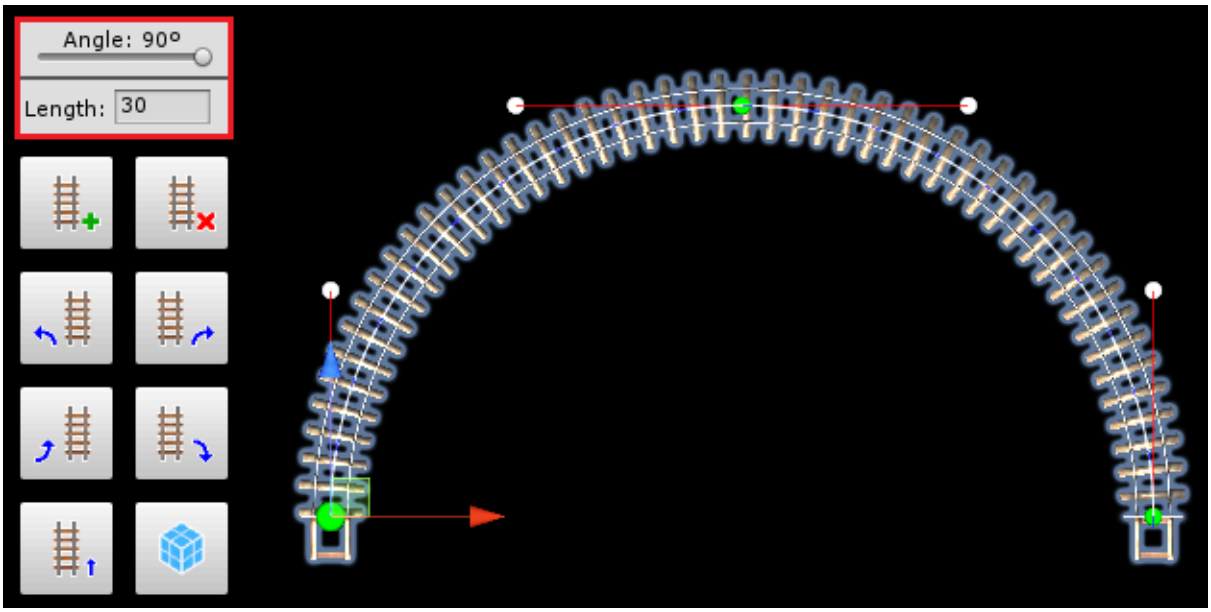
The curve length is measured in meters and affects both the Add Curve and Curve Shaping operations. The angle property is measured in degrees and affects only the Curve Shaping operations. By using both, it is possible to create curves of any length and curvature.



In the sample image above, these properties were used to create a 30m long 90º curve (left) and a 30m long 45º curve (right). In order to reproduce this example, all you need to do is to adjust the length and angle values and click on the button with the “right arrow” icon in order to create a curve to the right.

The angle property is limited to 90º in order to make it easier to create perfect circular shapes, in which the bezier handles are located in the optimal position (according to bezier splines best practices). However, curvatures greater than 90º can be achieved by using multiple curves and adding the curve angles values.

For example, if you wish to create a 60m long 180º curve, you can either use two 30m long 90º curves, four 15m long 45º curves or even three 20m long 60º curves.



6.1.3.1. Add Curve

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

6.1.3.2. Delete Curve

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

6.1.3.3. Curve Shaping Operations

The “Left” button reshapes the last curve of the railroad into a perfect curve pointing to the left. The curvature angle is defined by the "New Curve Angle" property and can be adjusted at the Quick Access Building Menu.

The “Right” button reshapes the last curve of the railroad into a perfect curve pointing to the right. The curvature angle is defined by the "New Curve Angle" property and can be adjusted at the Quick Access Building Menu.

The “Up” button reshapes the last curve of the railroad into a perfect curve pointing upwards. The curvature angle is defined by the "New Curve Angle" property and can be adjusted at the Quick Access Building Menu.

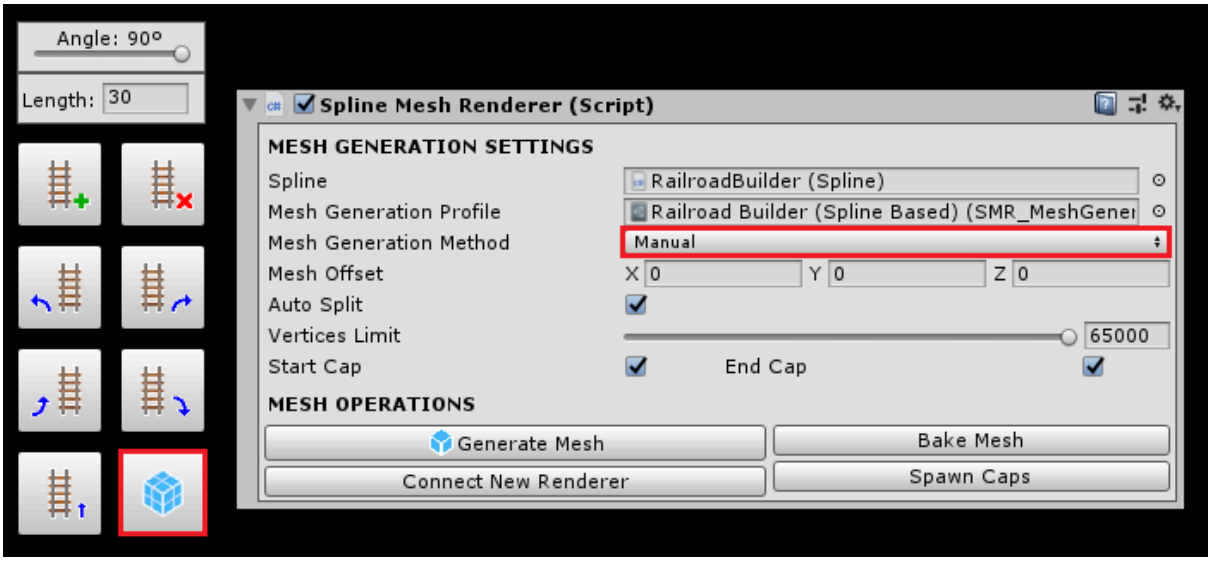
The “Down” button reshapes the last curve of the railroad into a perfect curve pointing to down. The curvature angle is defined by the "New Curve Angle" property and can be adjusted at the Quick Access Building Menu.

The “Straight Line” operation reshapes the last curve of the railroad 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 in any direction.

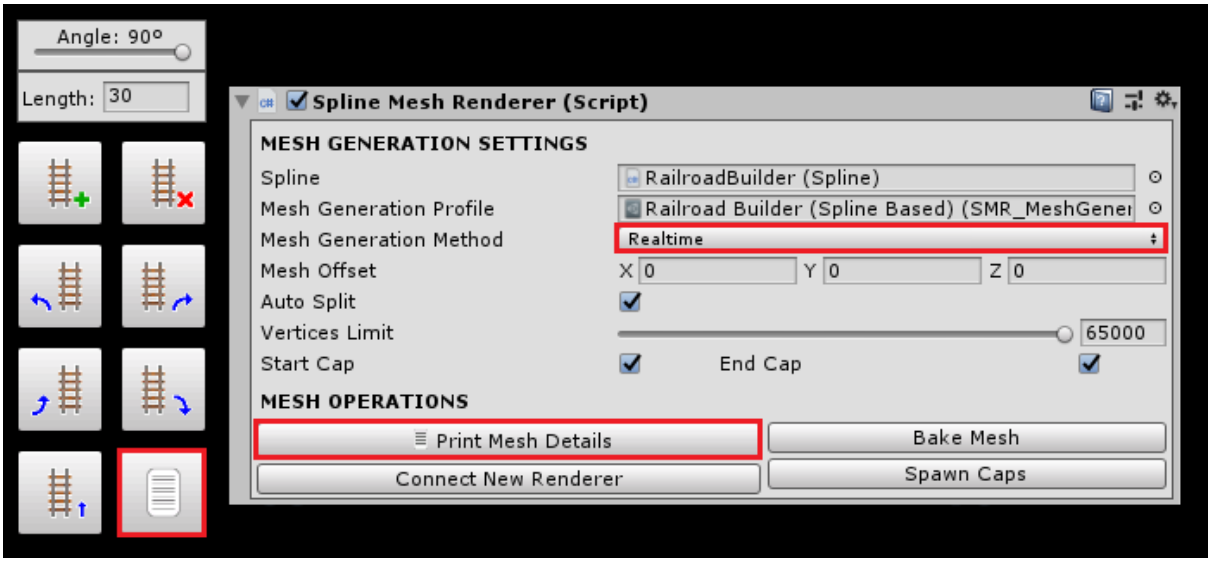
6.1.3.4. Generate Mesh

The “Generate Mesh” button is used to regenerate the railroad mesh, after making any changes.

When the manual mesh generation method is selected, the Generated Mesh button must be clicked to apply any changes made to the railroad.



When the realtime mesh generation method is selected, changes will be applied automatically. In this case, the Generate Mesh button is replaced by the Print Mesh Details button.



Note: Manual mesh generation is selected by default, since it allows a performance friendly workflow on any computer.

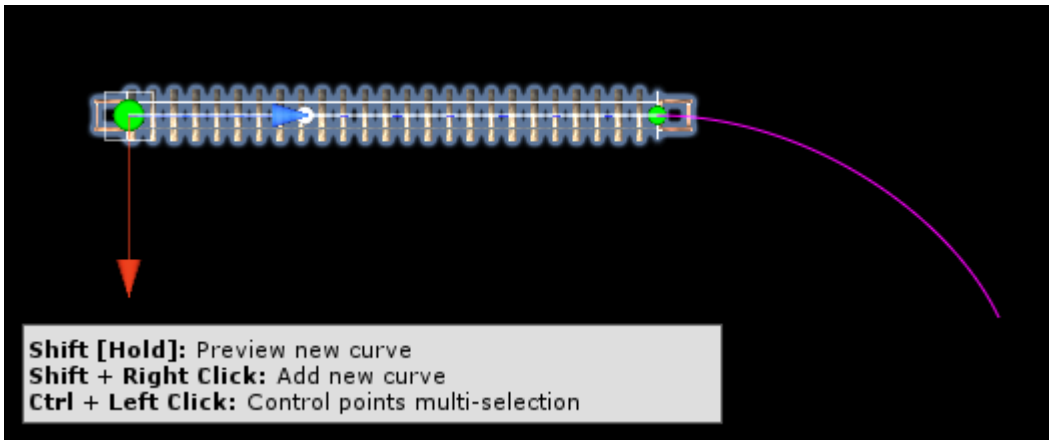
6.1.4. New Curve Projection (Shift+Click)

The Quick Access Building Menu is a very powerful and precise tool, however, it is restricted by the New Curve Length and New Curve Angle properties. Of course this is very useful in

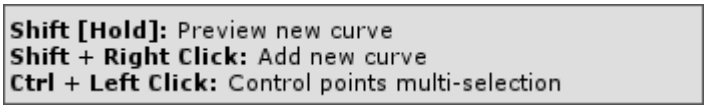
any situation in which curve precision is required, but it can be a bit of a pain when you need to create irregular shapes quickly.

For example, if you're using the railroad builder to create a railroad on a terrain, curve irregularity would make it look much more natural then perfect curves.

By using the new curve projection feature, you can manually create curves of any length and/or curvature in your scene very quickly. All you need to do is hold shift and move the mouse around to preview the curve (magenta line) and right click to confirm its creation.



When a railroad builder instance is selected, the following shortcut menu will be visible at the lower left corner of the scene view in order to remind you of this feature.

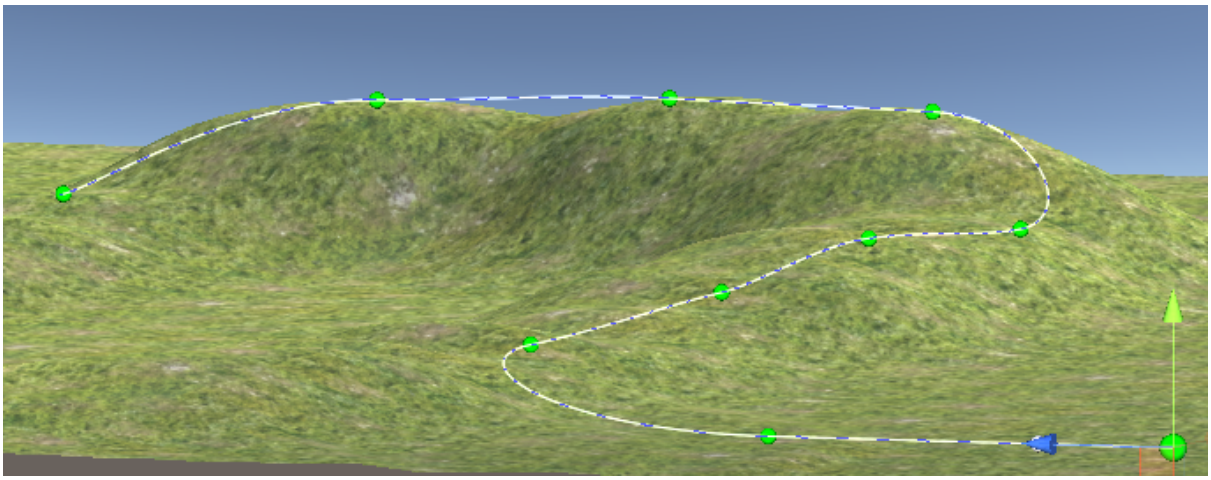


6.1.4.1. Terrain Projection

When using this technique to add new curves on terrains, the spline control points will always intersect with the terrain height and a smooth curvature will be created in between the control points.

This can be very useful to create smooth railroads uphill or downhill.

NOTE: If you wish your terrain elevations to follow your spline or vice versa, please take a look at the Terraforming and Follow Terrain sections.

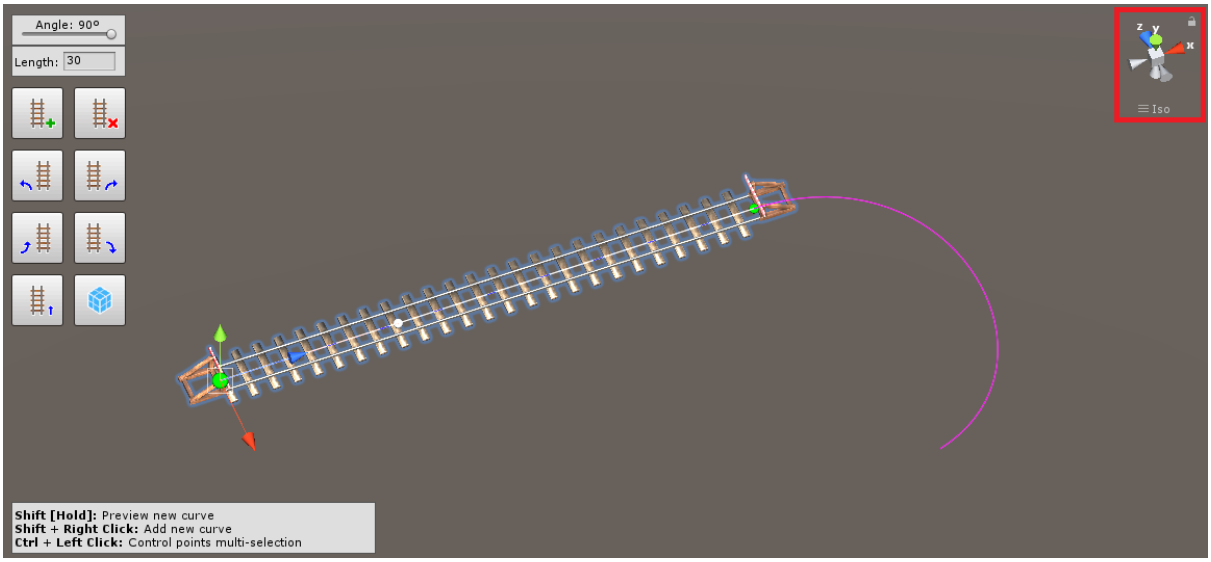


When projecting into terrains, by default the spline handles will be aligned horizontally for better results.

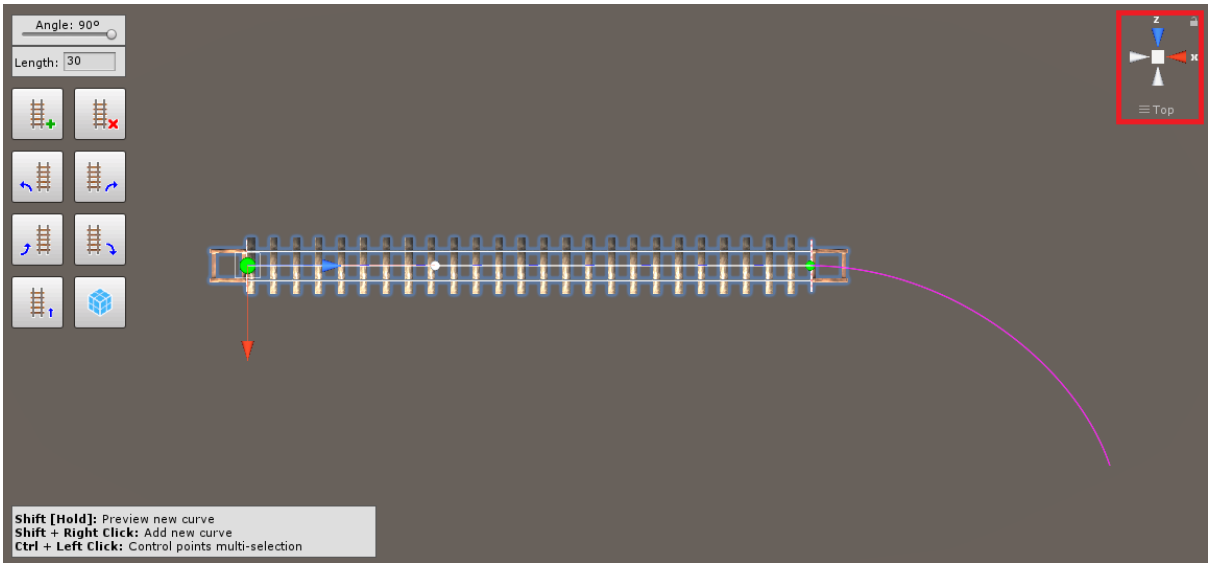
NOTE: This feature is not restricted only to terrains, you can also project on top of planes and/or any other colliders in your scene.

6.1.4.2. 2D Projection

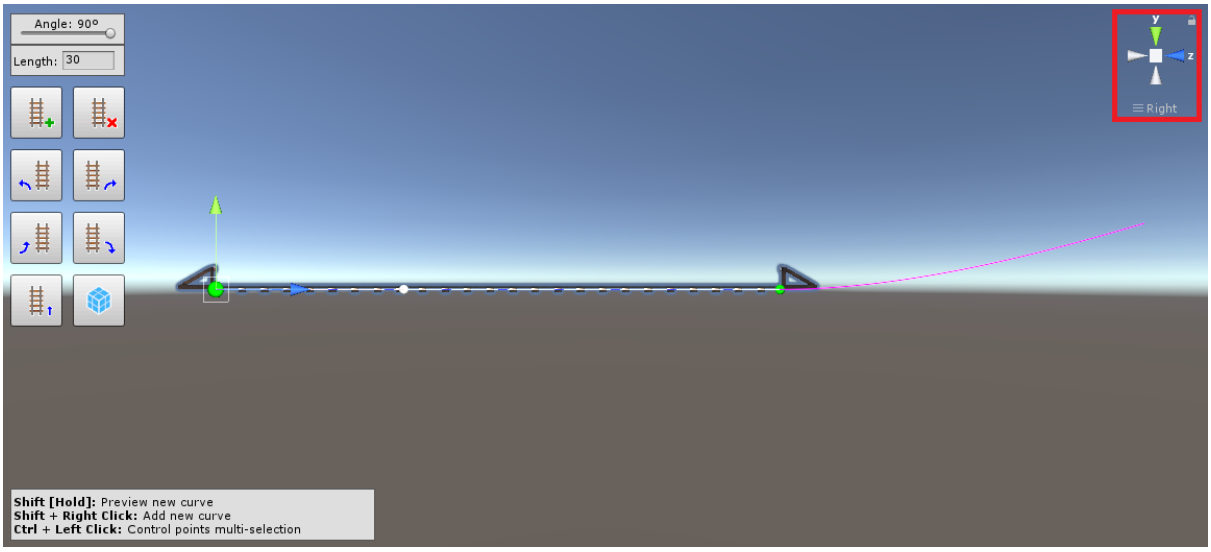
If you don't have a terrain or object to project on, you can also project on "air", but in this case the projection will be made into an imaginary **horizontal** or **vertical** plane.
By default, if you're looking at your scene at a random orientation, the new curve will be projected into a **horizontal** imaginary plane.



However, for better results it is recommended to use the **Top Down** view when you wish to project horizontally.



You can also make vertical projections, in that case you need to use the **Right View** in order to do that. Also, when working with vertical projection, make sure your spline is oriented along the Z or Y axis.



NOTE: If you are working on a 3D scene and need a vertical spline in your scene that is not oriented in the Z or Y axis, you can use the **Right View** orientation in order to create your projection, then rotate your spline transform to the desired direction once it is done.

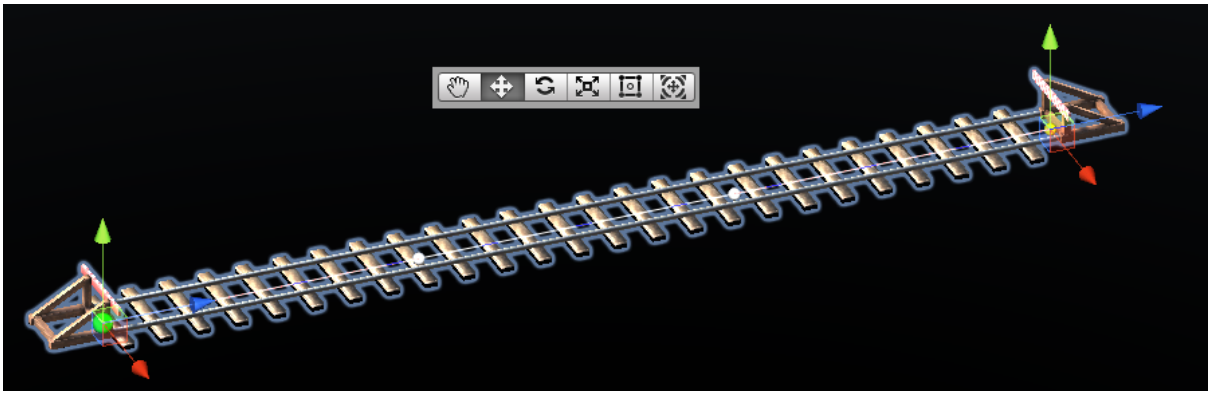
6.1.5. Manually editing the railroad

The shape of the railroad curves are defined by the position of the control points and handles. Besides, using the Quick Access Building Menu and the New Curve Projection features to shape your railroad, you can also manually adjust the position and rotation of the control points.

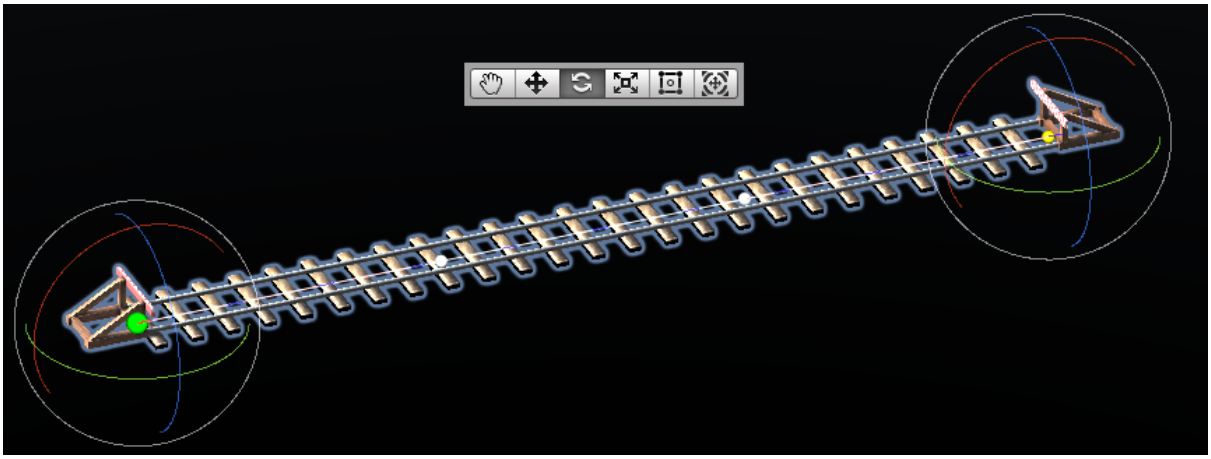
6.1.5.1. Single Point Editing

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

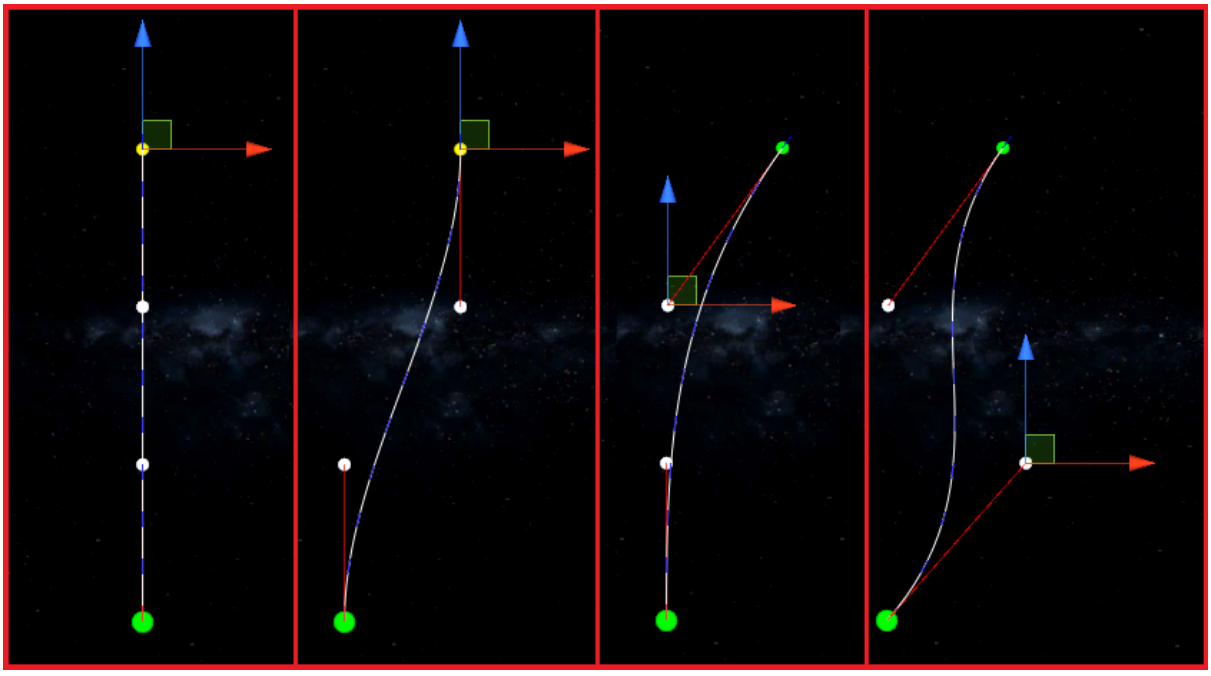
When a point is selected, its position or rotation handle will appear on the Scene View, depending on what tool is selected in the Unity Editor.



In order to use the rotation handle, you need to select the Rotate Tool in the Unity Editor.



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.

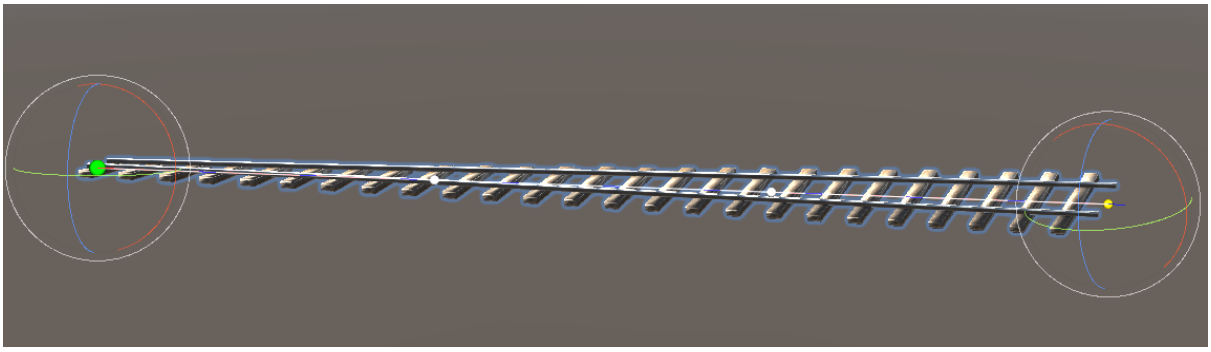


The selected point position and normal values can be seen under the Handle tab of the Spline Component.

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.

The Normal property identifies the local upwards direction for that point, it is used as reference to automatically calculate the rotation along the spline.

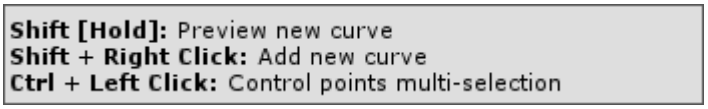
By using the rotation handle to adjust a control point normal it is possible to bend the rails.



6.1.5.2. Multi-selection

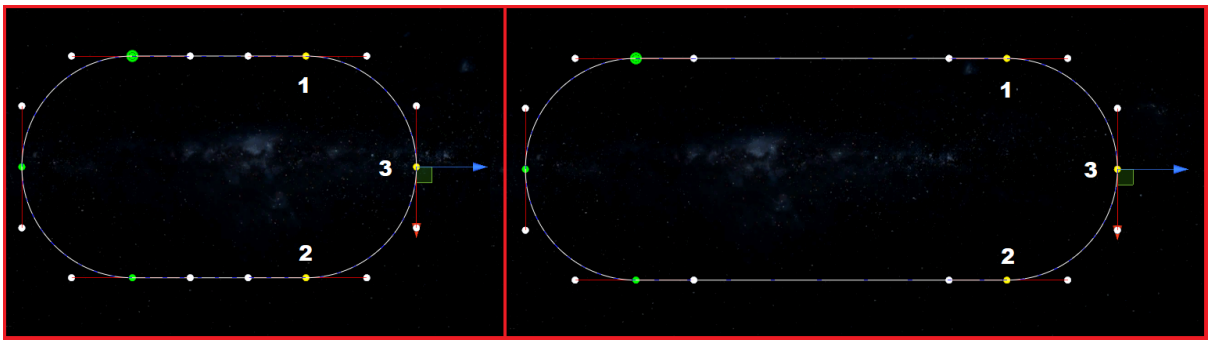
You can now edit multiple control points at once. In order to do that, hold Ctrl and left click to select/deselect control points.

When a spline is selected on the hierarchy, the following shortcut menu will be visible at the lower left corner of the scene view in order to remind you of this feature.



- **Selection Color Code**
  - **Selected**: Yellow
  - **Unselected:** Green

In the sample image below, the multi-selection was used to move points 1, 2 and 3 at the same time.



Note: Within the context of a bezier spline there is no practical use for handles multi-selection. For this reason, handles cannot be multi-selected, only control points can (green points).

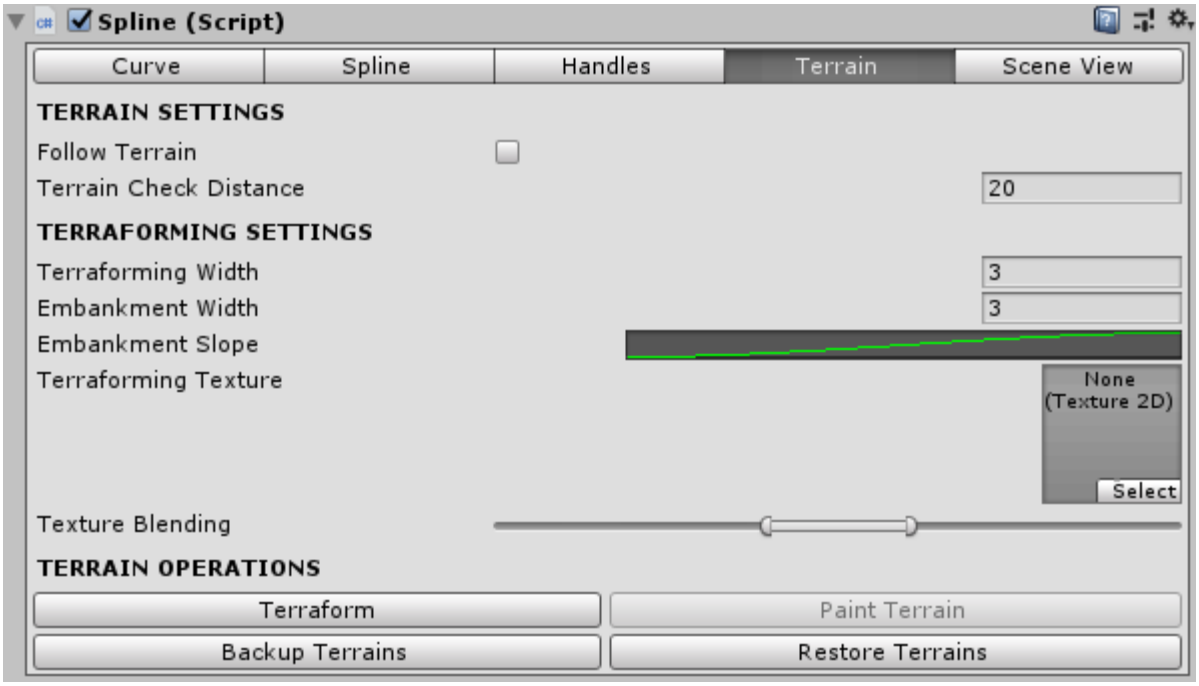
However, handles will inherit changes made to the corresponding parent control point for the sake of consistency.

6.1.6. Terrain Features

Regarding the relationship between railroads and terrains, there are only two possible outcomes: you can either adjust the railroad to the terrain **or** adjust the terrain to the railroad. Luckily for you, this asset supports both.

You can use the Follow Terrain feature to project the spline to your terrain. Perfectly, adjusting the spline to the terrain elevations.

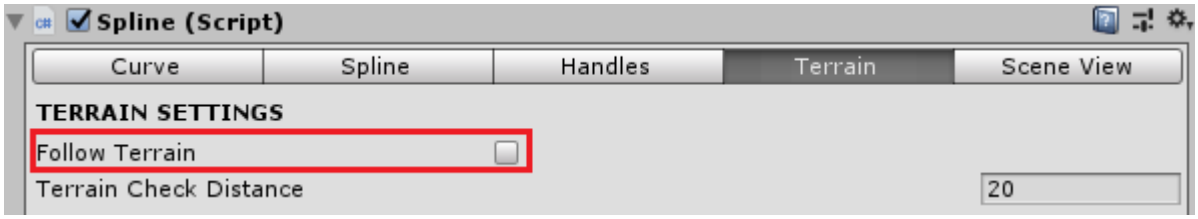
Alternatively, you can use the Terraforming feature to adjust the terrain to your spline.



6.1.6.1. Follow Terrain

The “Follow Terrain” property is used to add terrain elevation support **without** the need of manually adjusting your railroad to fit your terrain. It works by creating a virtual projection of

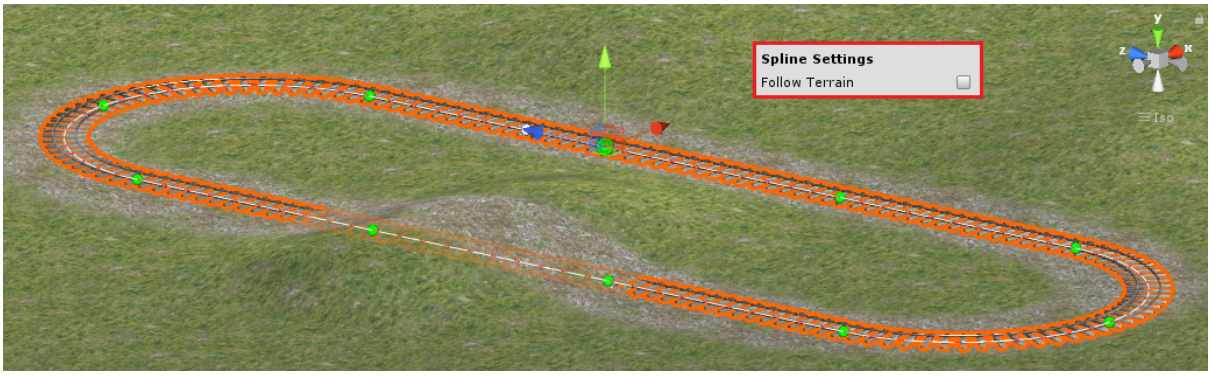
the railroad spline on your terrain. The spline shape is preserved as it is, and the virtual projection can be used as a reference by the railroad builder to generate the rails.



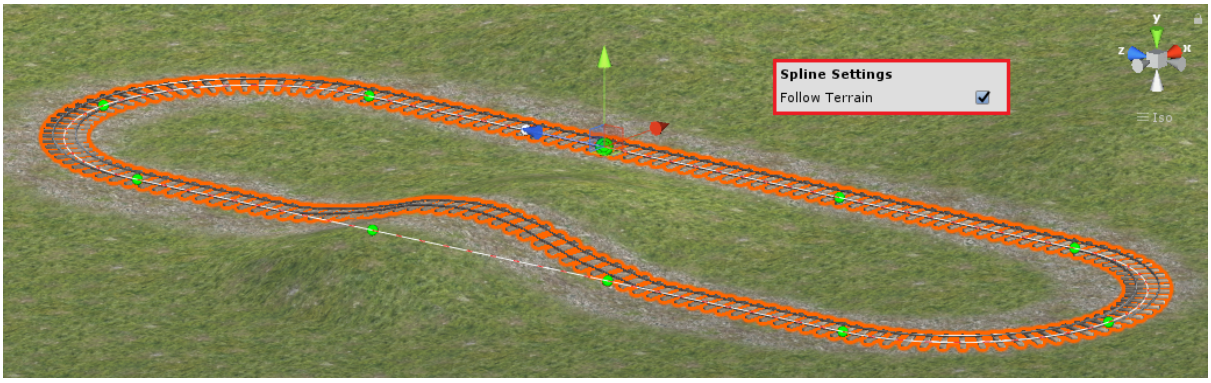
This technique allows you to use simple flat splines to create complex railroads that will automatically adjust to any type of terrain elevation.

For example, let's assume you wish to build a circular railroad.

First, you build a simple flat spline on the desired shape, but you realize that a segment of the railroad is passing through the terrain elevations, as shown in the sample image below.



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.



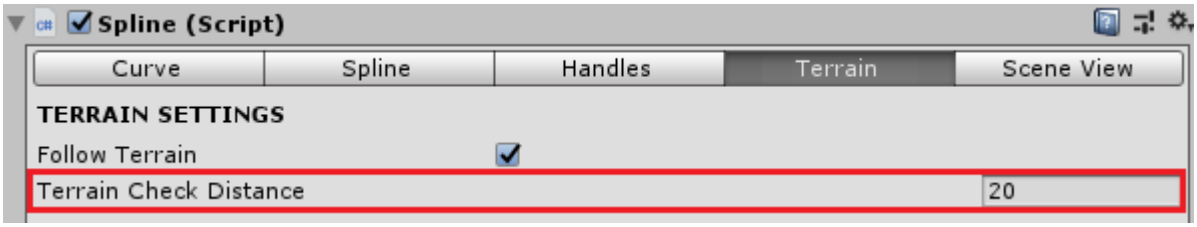
NOTE: The railroad spline shape remains unchanged, even though the rails mesh is now being projected on the terrain.

6.1.6.1.1. 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 railroad 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 railroad the “Follow Terrain” feature should look for the terrain intersection.



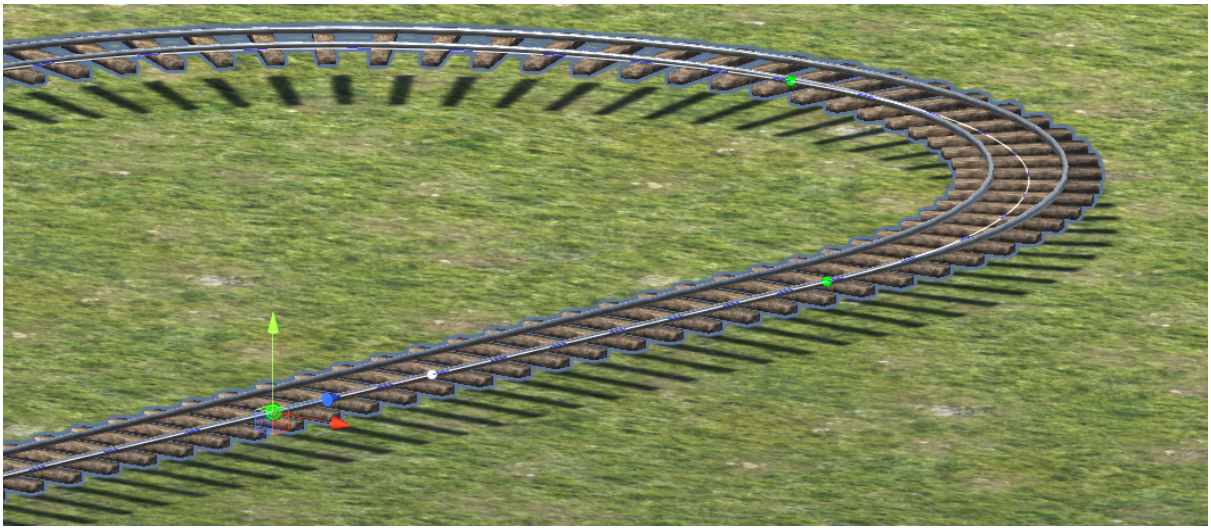
NOTE: By default the “Terrain Check Distance” is set to 20 meters, which means it will look for the terrain 20 meters downwards and 20 meters upwards, covering a vertical range of 40 meters.

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.

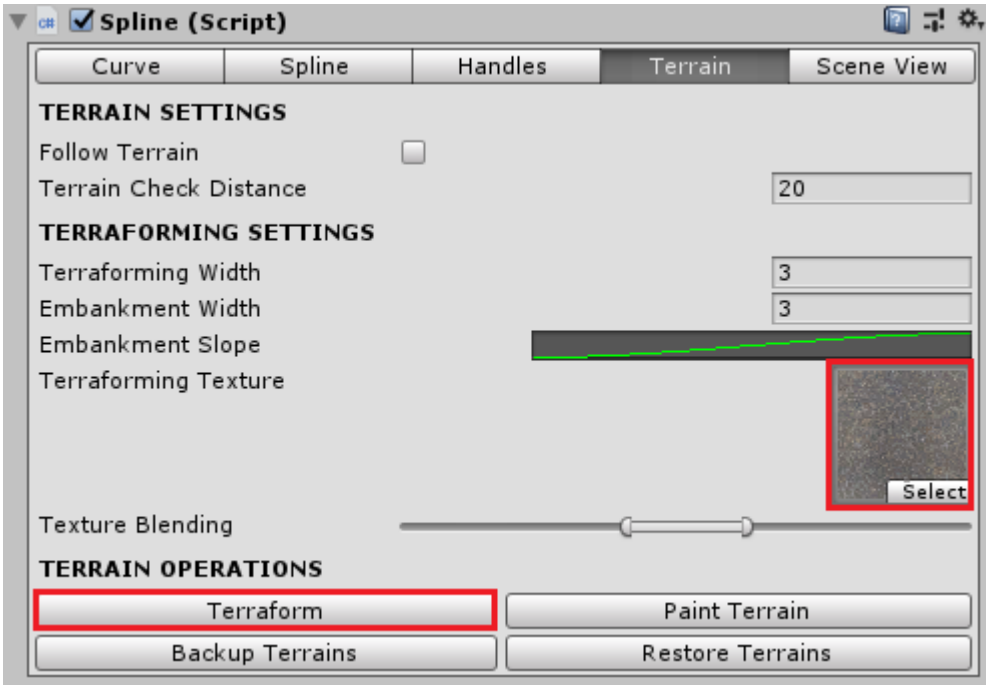
Alternatively, you can also roughly adjust the spline to bring it closer to your terrain until it reaches the check range. This alternative is performance friendly and can be done quite quickly.

6.1.6.2. Terraforming

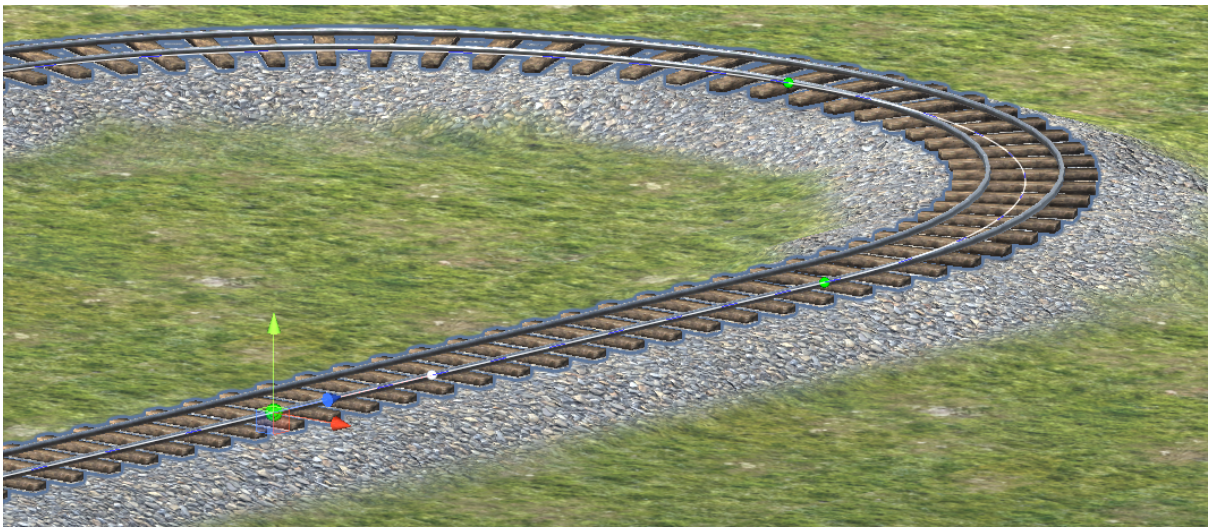
In some situations, you may wish to adjust the terrain heights to match your railroad. For example, let’s assume you are building a railroad for physics based trains and for the sake of train stability, you wish your railroad to be as flat and smooth as possible. On top of that, you also desire to create Track Ballast along your railroad for detailing purposes.



Let’s assume you already built your railroad by using a railroad builder that is located 1 meter above the terrain. So, all you have to do now is to adjust the terrain height along the railroad and apply a rock texture to simulate the track ballast rocks. However, manually adjusting the terrain heights and textures would be very time consuming. That’s why the terraforming operation was created to make this process as simple as a button click.



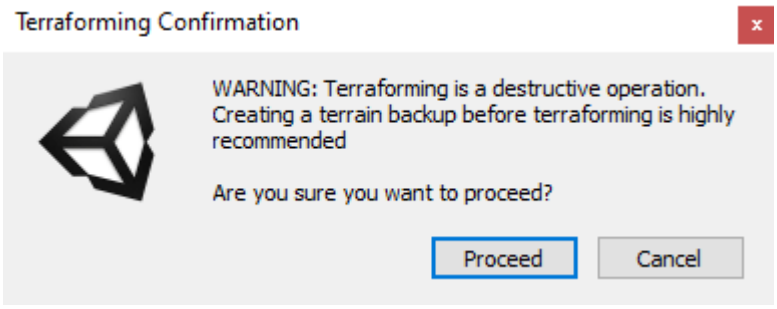
In order to create the track ballast, just select a rock terrain texture and click on the “Terraform” button.



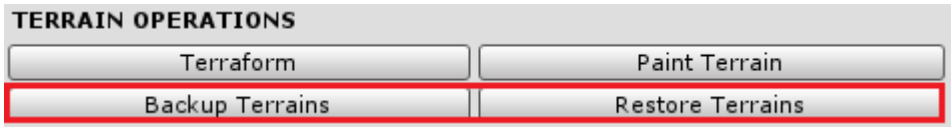
Note: The Terraform feature will do the best it can to create a smooth result based on the selected Terraforming Settings. However, terrains with lower heightmap resolution may present some terraforming artifacts (non-smooth areas). In these cases, it is recommended to use the Unity default Smooth Height brush to make minor adjustments where needed.

6.1.6.2.1. Terrain Backup

Since terraforming terrain painting is a destructive operation (which means, ctrl+z will not undo terraforming), it is highly recommended to back up your terrains before proceeding with it. In order to remind you of the importance of backups, this confirmation dialog will pop up whenever you click on the “Terraform” button.



You can create and restore terrain backups by using the following buttons.



A terrain backup is just a copy of its terrain data file, saved alongside the original terrain data. To keep things simple, just think of the backup file as a screenshot of your terrain settings at the moment of the backup creation.

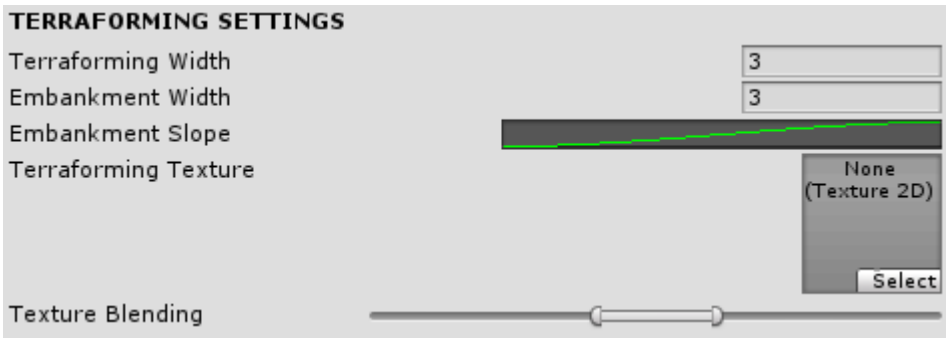
Note: Keep in mind that only 1 backup file per terrain is supported at any moment in time. Therefore, make sure to only update your terrain backups when you're happy with the modifications you've made.

In order for the "Restore Terrains" feature to work properly, the backup and the current terrain size, heightmap and splatmap resolutions **MUST BE THE SAME**. If you change any of your terrain settings in your scene or rename your terrains, it is recommended to update your terrain backups before using the Terraforming or Paint Terrain features.

Note: As a rule of thumb, it is recommended to always update your backups before using the Terraforming or Paint Terrain operations. That way, you can always use the Restore Terrains feature to undo the operation in case you wish to change the terraforming settings and try again.

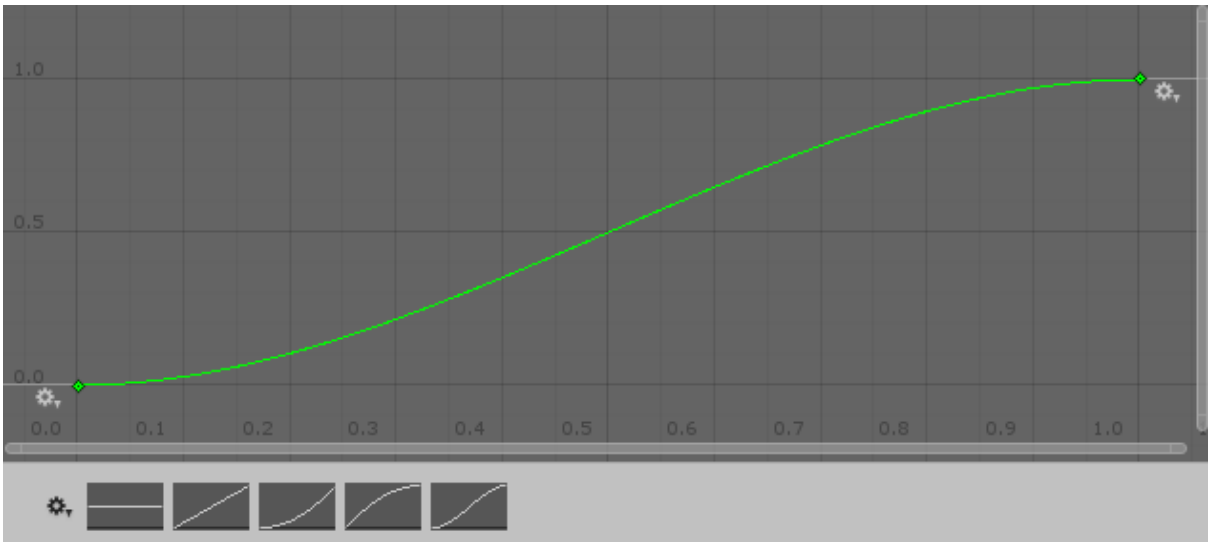
6.1.6.2.2. Terraforming Settings

The terraforming settings affects the "Terraform" and "Paint Terrains" operations.



The “Terraforming Width” property defines how much of the terrain will be “flatten” along your railroad when using the Terraform operation.

The “Embankment Width” defines how far the terrain smoothing effect will be applied, based on a smoothing curve defined by the “Embankment Slope” feature.



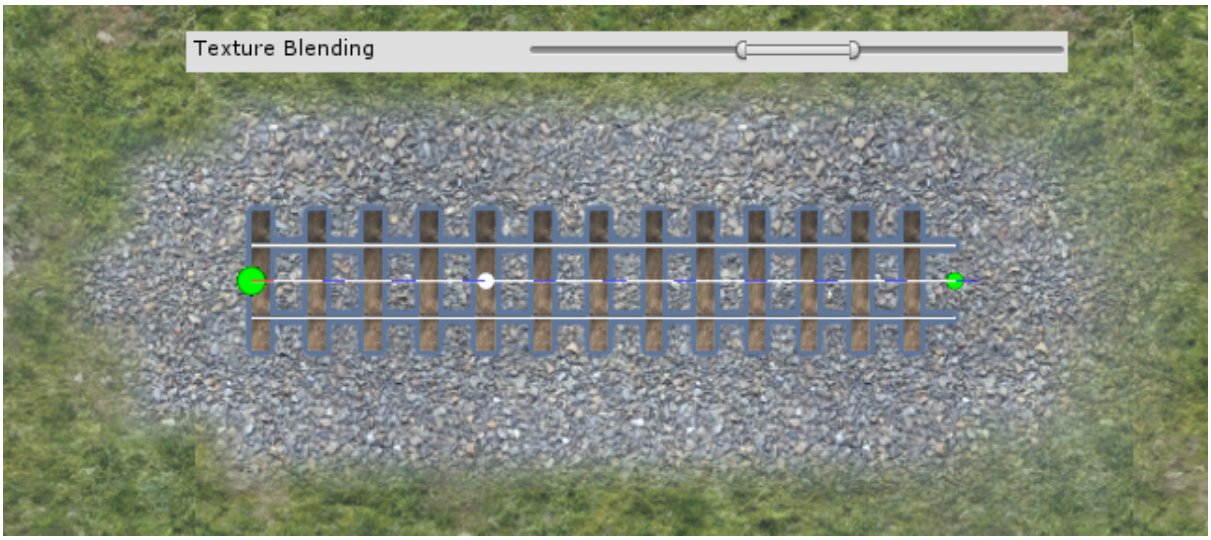
If you click on the “Embankment Slope” property in the inspector, the Unity default curve editing dialog will be shown. You can choose the available presets or manually adjust the curve. However, keep in mind that the far left value must always be 0 and the far right value must always be 1.

Note: Both Terraforming and Embankment Width properties are measured in pixels. Therefore, the affected area may change based on your terrain’s heightmap and splatmap resolutions.

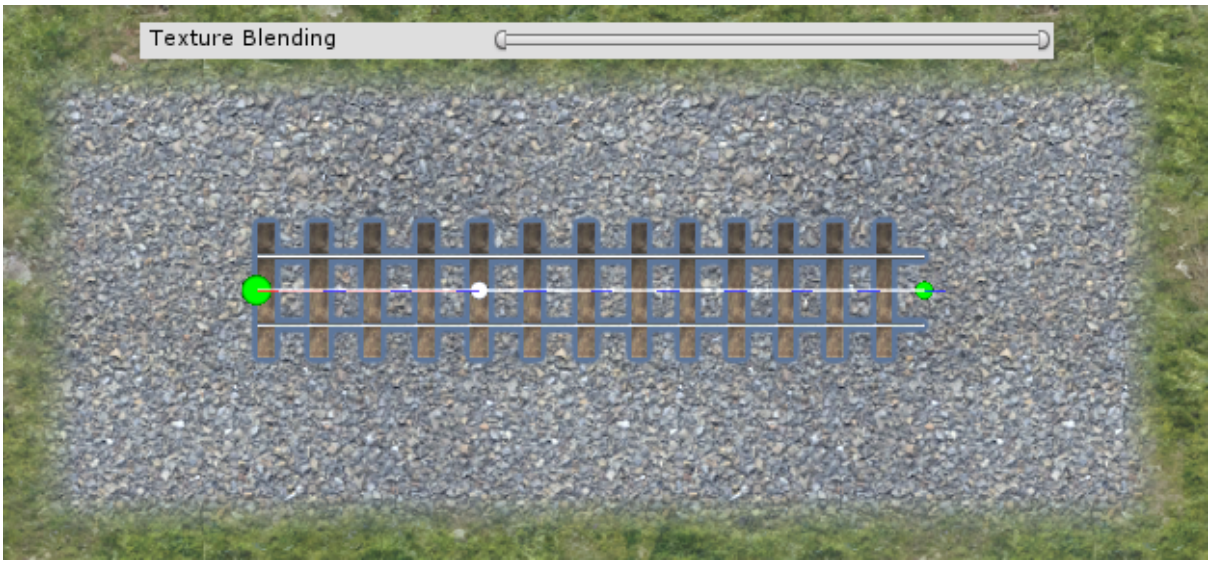
The “Terraforming Texture” is used to repaint the area affected by the “Terraform” and/or “Paint Terrains” features.

Note: The “Terraforming Texture” property is used to identify which terrain layer should be applied on the affected region. In order for the terrain painting to be successful, your terrain must have a corresponding terrain layer in which the Diffuse texture is the same as the selected terraforming texture.

You can also adjust how much “Texture Blending” will be applied while repainting. The blending is more visible at the railroad starting and ending points. By using the default blending, it will produce a rounded texture blending effect on both ends. This is useful while painting more natural paths.



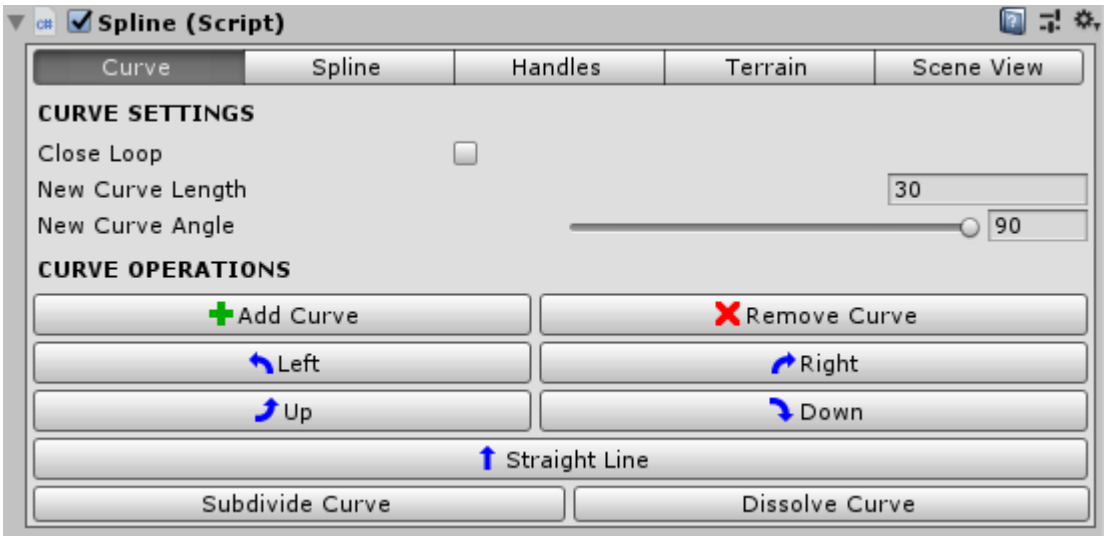
However, if you wish your path to be squared on both ends, you can drag the blending range all the way to the end in order to produce this effect. This can be useful for creating “human made” paths.

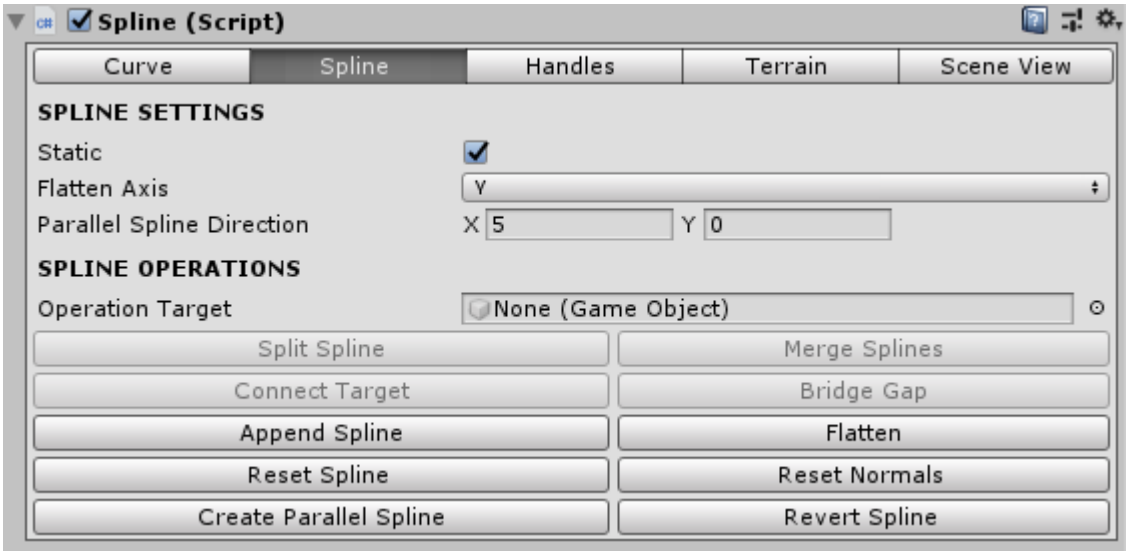


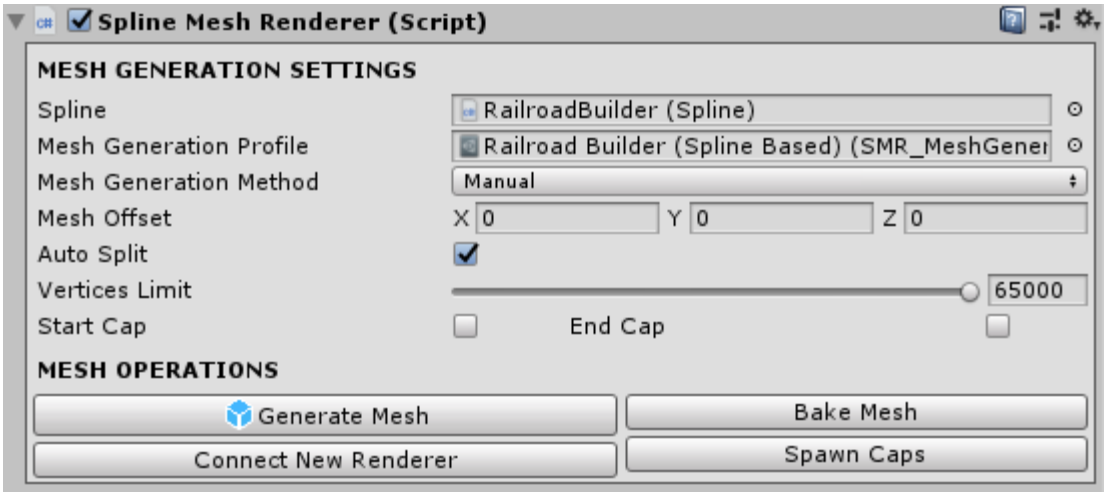
6.2. Railroad Operations & Settings

In this section you will find information about useful Railroad Builder operations and settings that can be used to improve your railroad building workflow.

These operations and settings are distributed among the Spline and Spline Mesh Renderer components of the Railroad Builder.

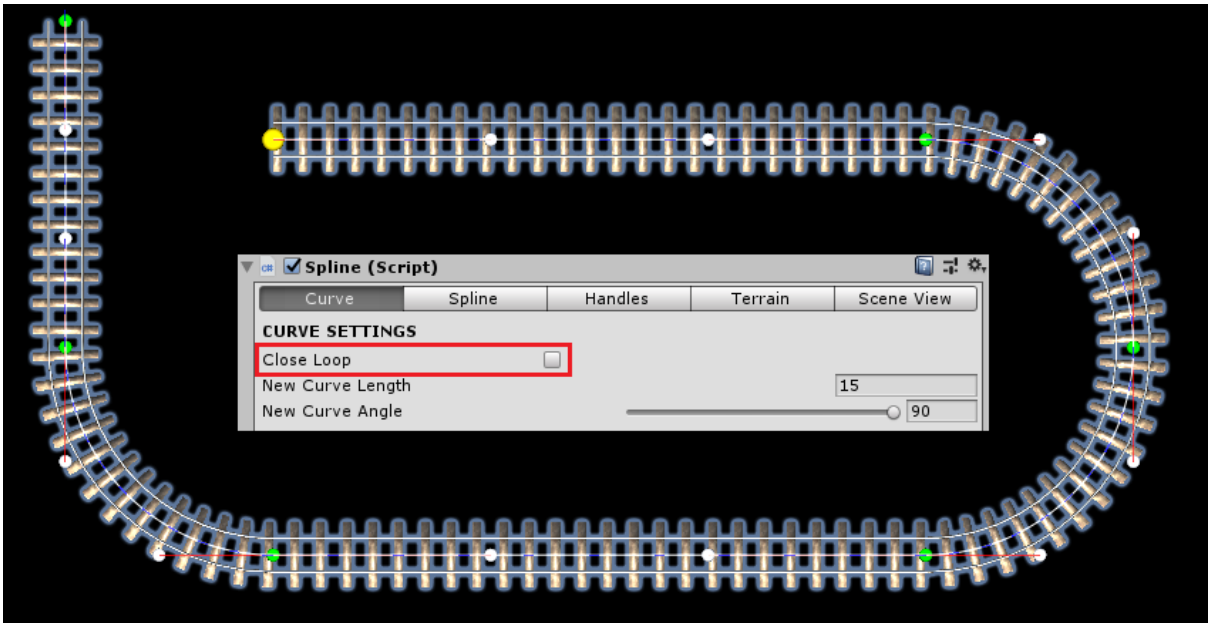




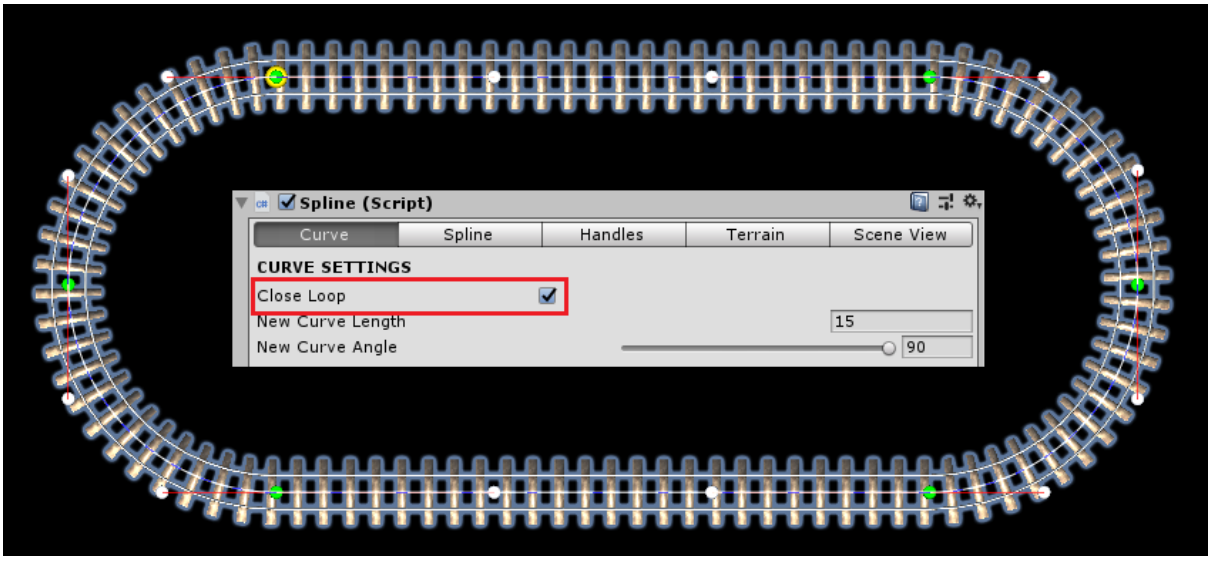


6.2.1. Close Railroad Loop

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



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



As you can see, the last curve was adjusted to create a closed railroad.

Note: When the Close Loop property is disabled, the last curve will be disconnected and reseted into a straight line.

6.2.2. Subdivide Curve

The "Subdivide Curve" operation divides the selected curve in half, creating two distinct curves. This operation is useful to create new curves on the middle of the spline.

Note: To select a curve, just select the first control point of the curve.

6.2.3. Dissolve Curve

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

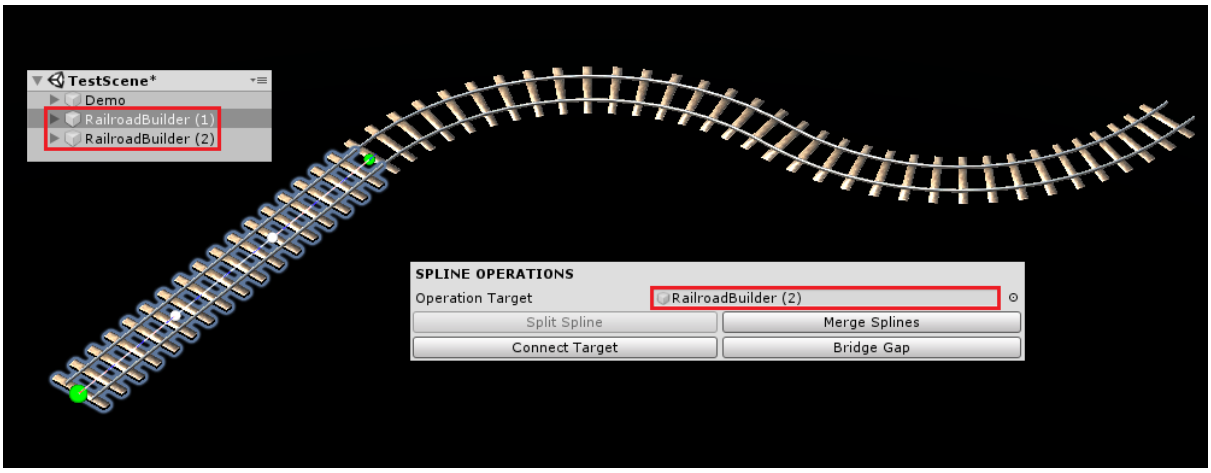
6.2.4. Splitting a Railroad

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

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

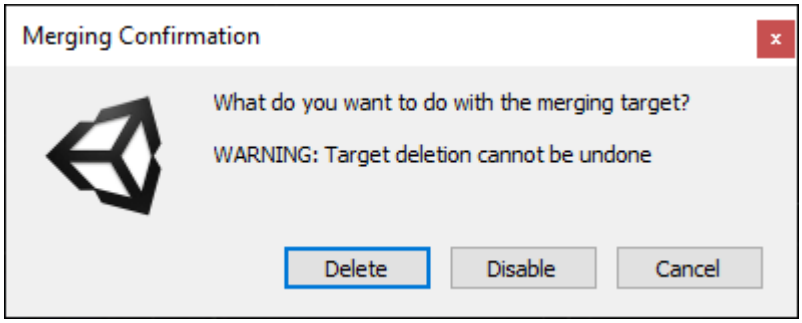
6.2.5. Merging Railroads

The “Merge Splines” operation is the opposite of the Split Spline operation. By using this operation you can merge two railroads into one.



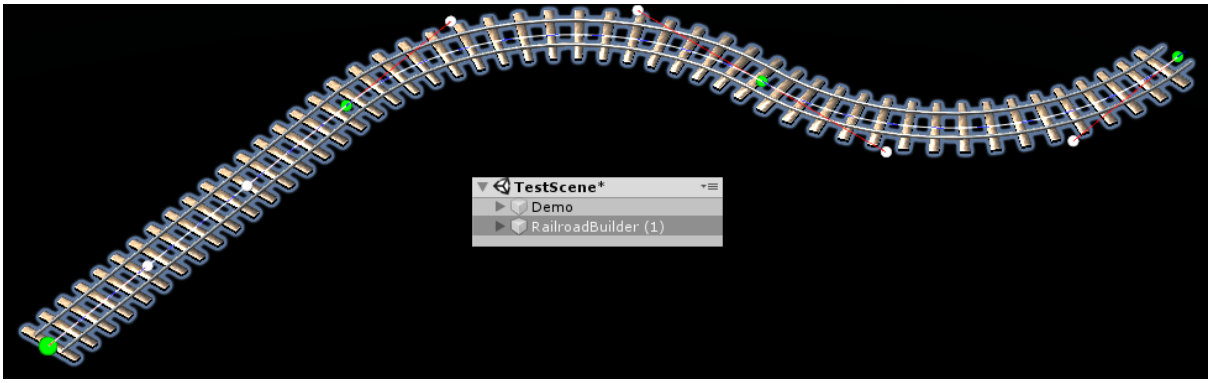
In the sample image above, the merging operation is being used to merge two railroad segments into a single railroad segment.

In order to do that, the “RailroadBuilder (2)” was set as the operation target at the Spline component of the “RailroadBuilder (1)”.

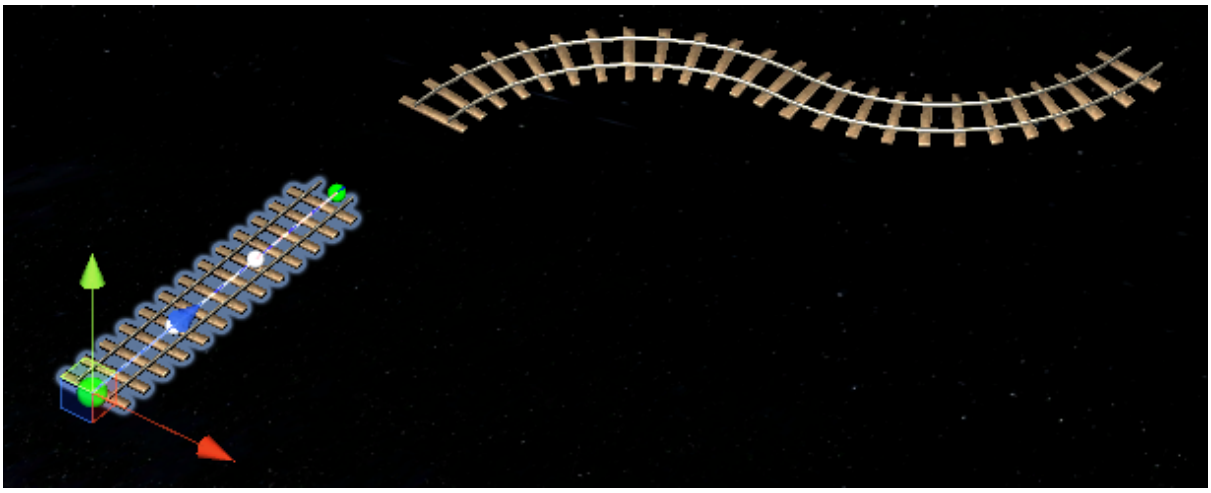


Once you click at the “Merge Splines” button, the confirmation dialog above will be displayed in the Unity Editor. You can either choose to delete or disable the target object. In this

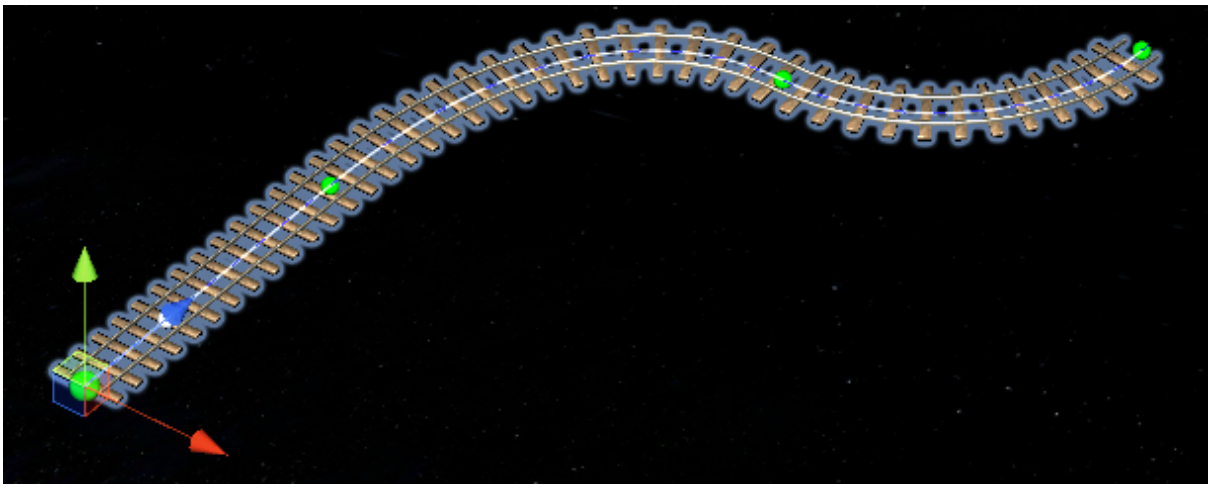
example, the delete option was selected, therefore only the “RailroadBuilder (1)” object remains at the scene after merging as can be seen at the sample image below.



In the sample above, the “RailroadBuilder (2)” was perfectly positioned at the end of “RailroadBuilder (1)”. But, this may not always be the case. Eventually, you may wish to merge railroads that are not connected to each other.



When merging disconnected railroads, the gap between the railroads will be closed automatically.

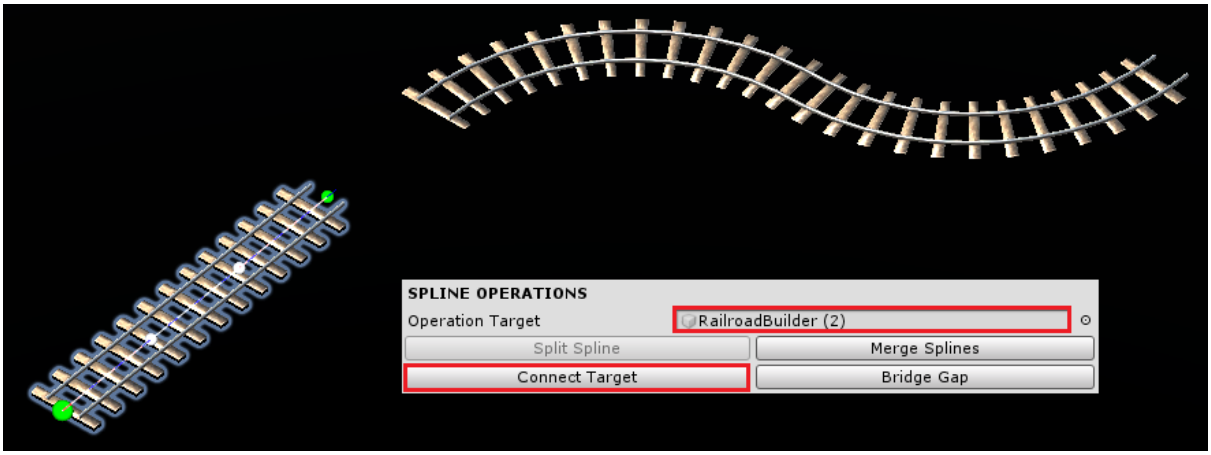


NOTE: The merging feature will do the best it can to preserve the curve shape of disconnected railroads, if you don’t want any distortion on the resulting railroad you can

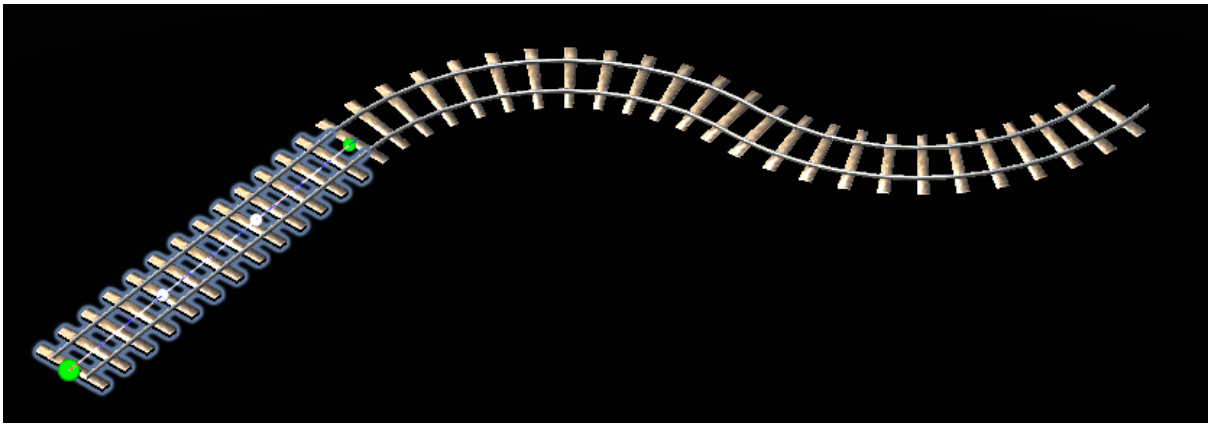
either move the target railroad position by using the Connect Target operation or close the gap prior to merging using the Bridge Gap operation.

6.2.6. Connect Target Object

In some situations, you may need to reposition an object on your scene to the end of a railroad. You can use the “Connect Target” operation in order to do that.



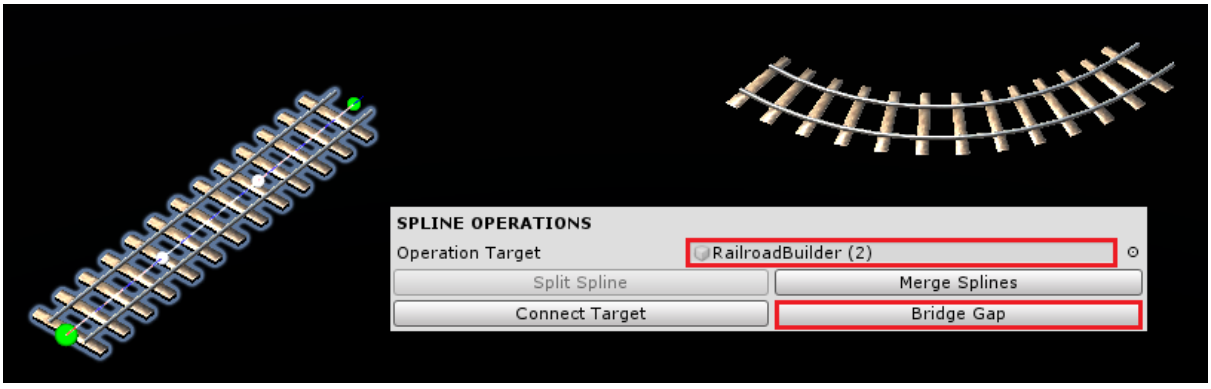
In the following example, the connect operation was used to move the “RailroadBuilder (2)” to the end of the selected railroad, in order to align both railroad segments.



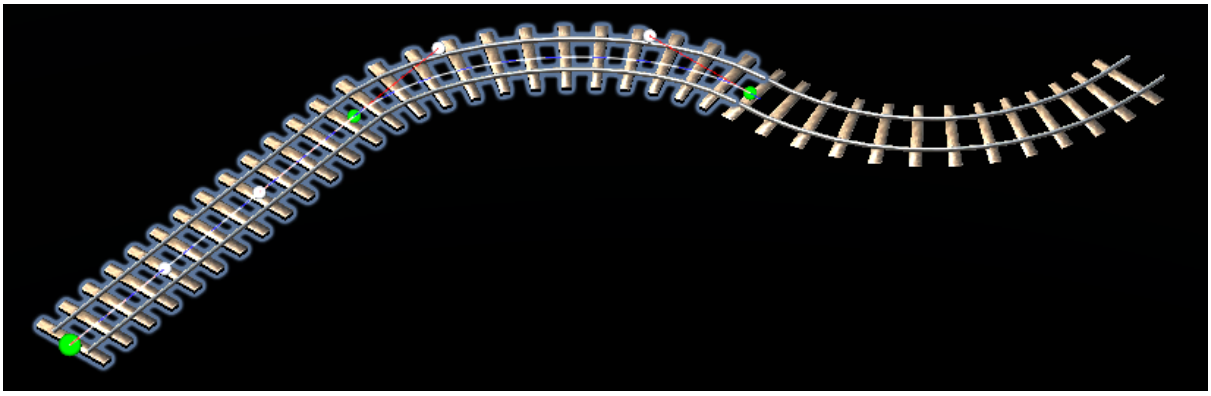
NOTE: This operation is very useful for level design, as it allows you to connect not only other railroads, but any object to the end of your railroad.

6.2.7. Bridge Railroad Gap

Consider the following situation: you have a gap between two railroads in your scene and you want to close it using a new curve.

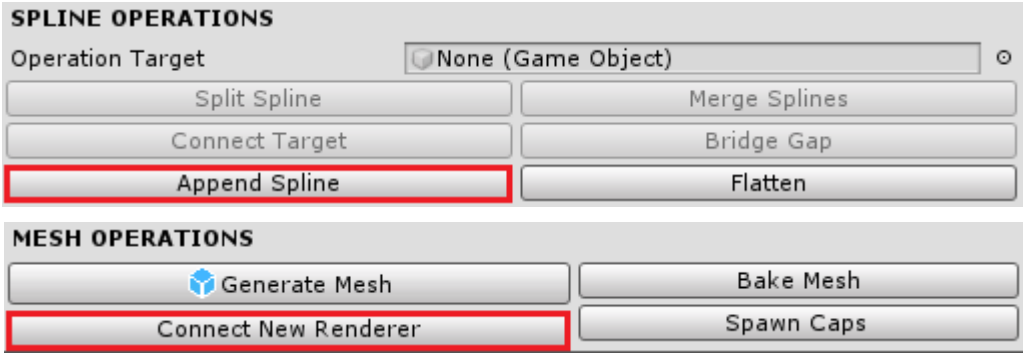


In order to do this, you can use the “Bridge Gap” operation. It will create a new curve and automatically adjust the last point position to close the gap between the objects.



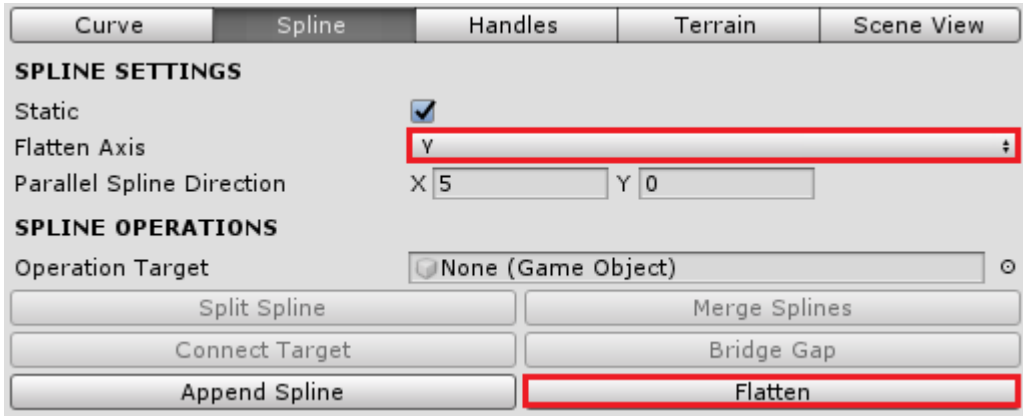
6.2.8. Append Railroad

Both “Append Spline” and “Connect New Renderer” operations can be used to create a new railroad at the end of the current selected one. The newly created railroad inherits all the characteristics and components from the original object. However, the new railroad is generated with only one straight curve segment.



6.2.9. Flatten Railroad

The “Flatten” operation sets all control points and handles at the same level of the first control point of the railroad spline, transforming a 3D into a “2D spline”. The spline dimension that will be “removed” when using this operation is defined by the Flatten Axis property.



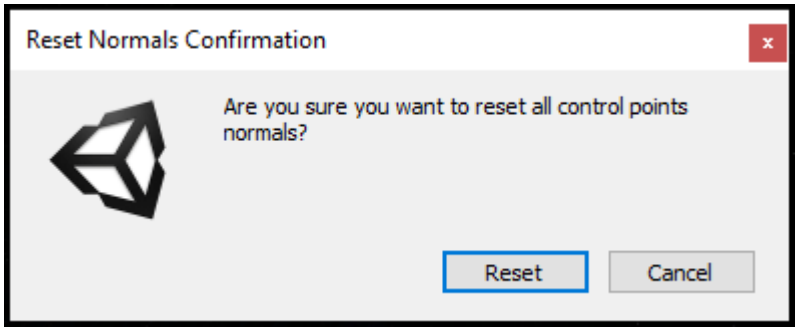
Note: The Flatten spline feature is a legacy feature from v2.0 that was originally designed as a quick way to undo the old version of the Follow Terrain feature that was deprecated.

On version 3.0, 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.

In version 3.4, the Flatten Axis property was introduced in order to extend this feature's usefulness to converting 3D splines into 2D splines.

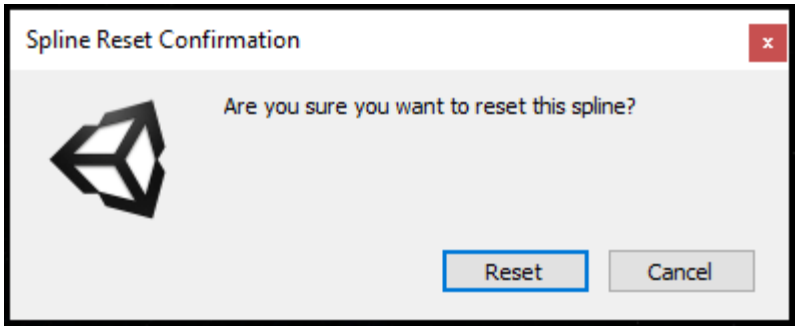
6.2.10. Reset Railroad Normals

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



6.2.11. Reset Railroad Spline

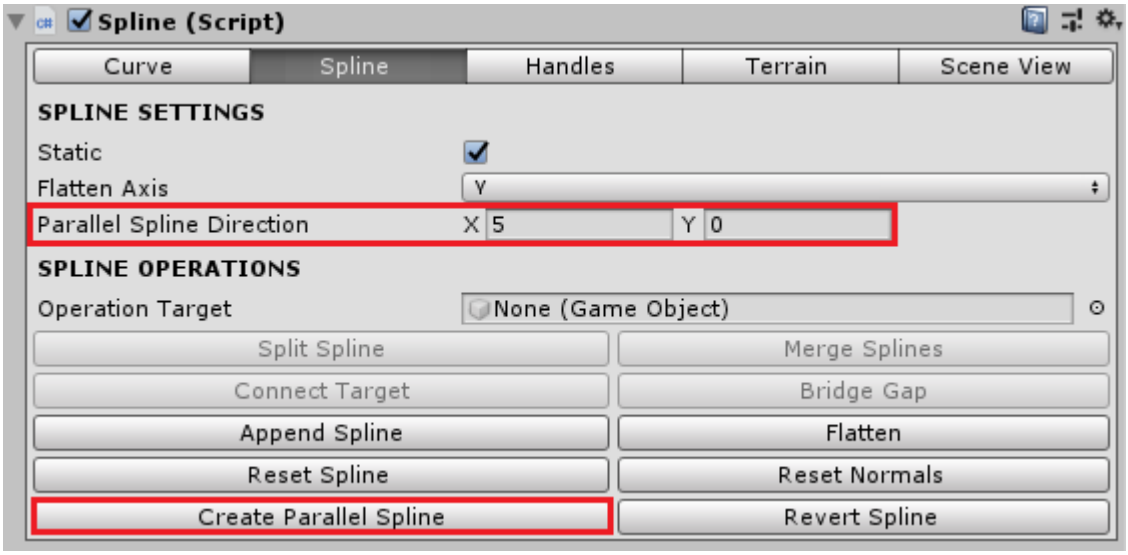
The “Reset Spline” operation completely resets the railroad spline to its initial state: only one straight curve segment. This operation is useful if you want to recreate your railroad from scratch.



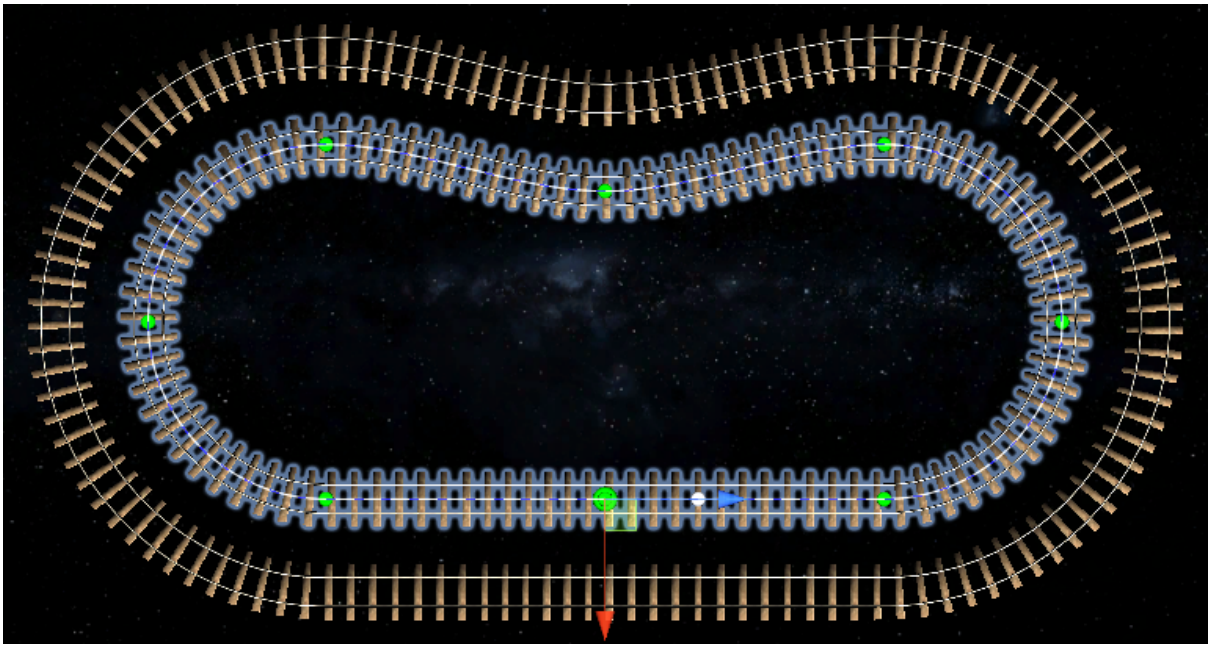
6.2.12. Create Parallel Railroad

If you need to create parallel railroads on your scene, you can use the “Create Parallel Spline” operation instead of manually building another railroad.

All you need to do is to feed the desired direction and distance on the “Parallel Spline Direction” property and click the “Create Parallel Spline” button.



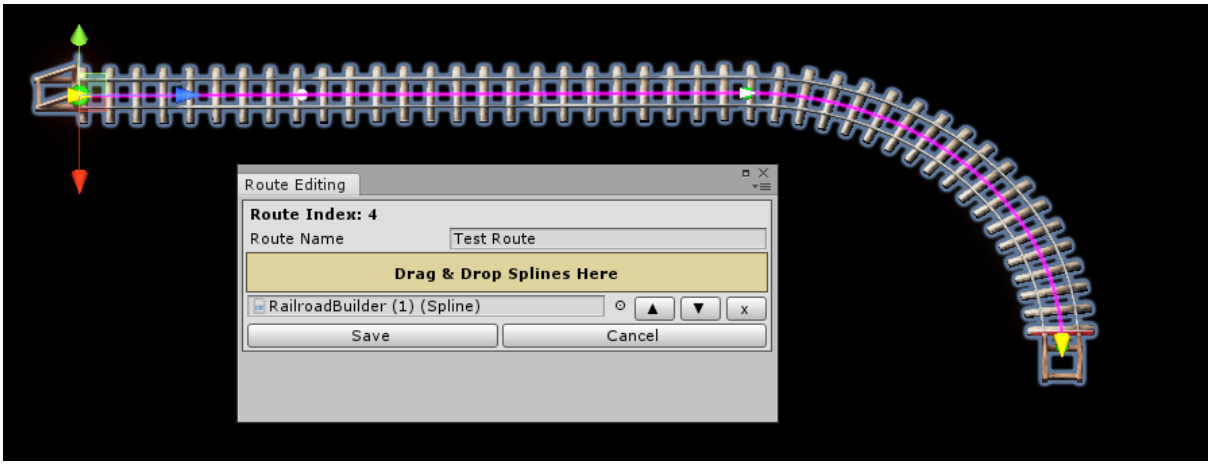
In the sample below, this operation was used to create a parallel railroad 5 meters to the right of the original spline.



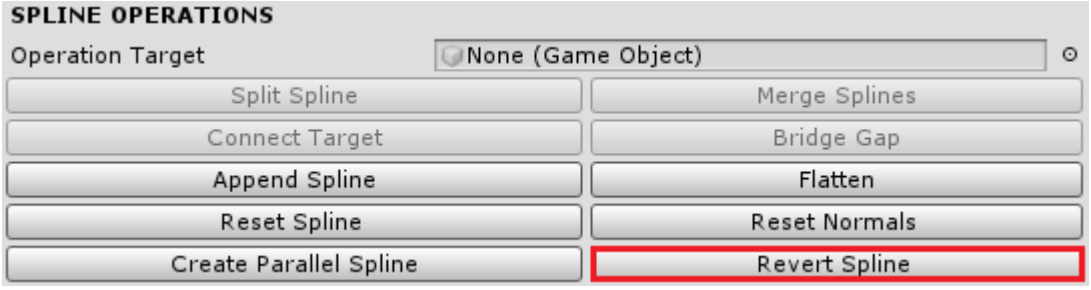
Note: In order to identify the local horizontal axis to create the parallel spline, this feature makes use of the control points normals. If you have any distortion while creating a parallel spline, adjust the control points normals and try again.

6.2.13. Revert Railroad Direction

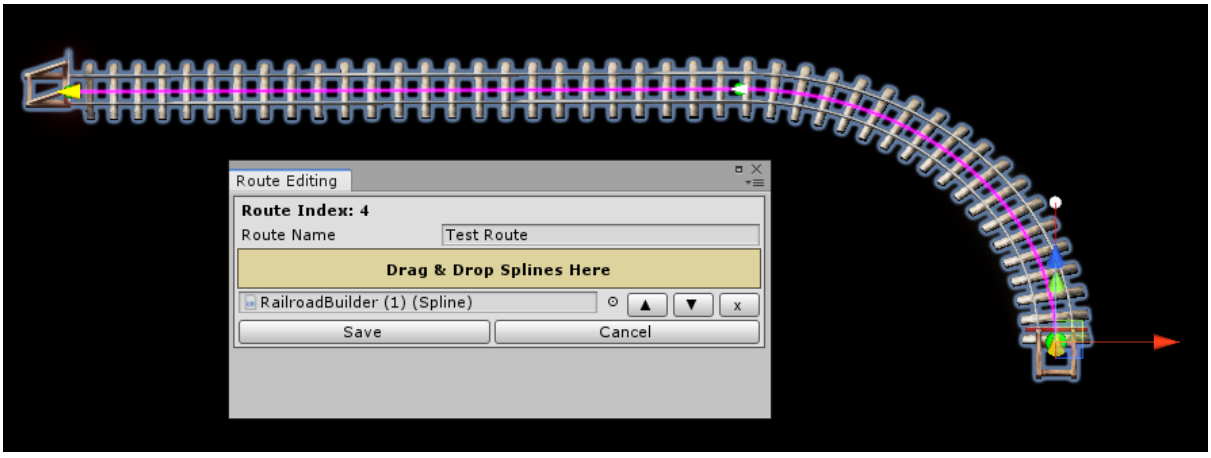
Spline based railroads are directional, by default the start (first control point) is always at the origin of the railroad builder object.



The “**Revert Spline**” operation can be used to revert the railroad direction if needed.



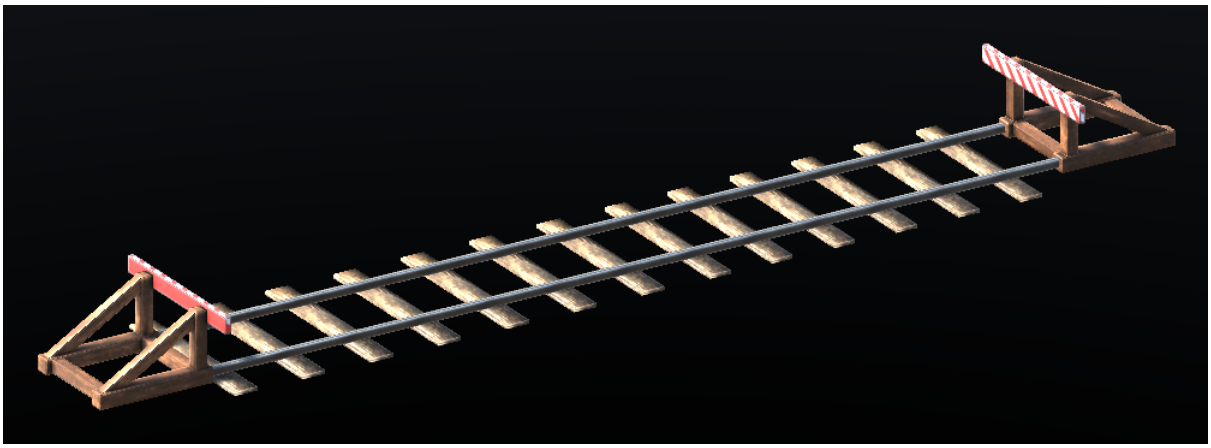
Reverting will move the railroad builder to the end of the spline and revert all control points.



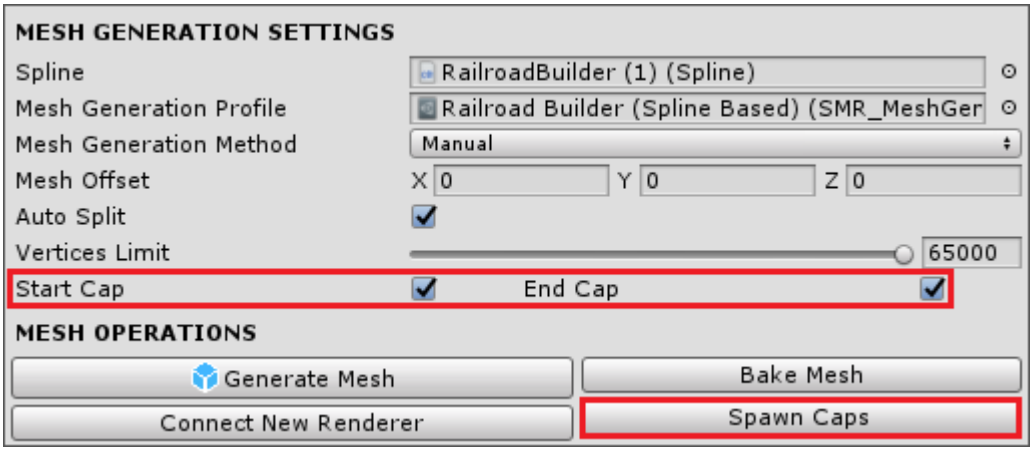
Note: Spline based trains can move back and forwards on a railroad. However, the train direction is tied to the railroad direction.

6.2.14. Spawning Stopblocks

The default rails generation profiles are pre-configured to add stopblocks at the start and end of railroads. They will spawn automatically whenever a railroad is generated, if the “Start Cap” and/or “End Cap” properties are enabled.



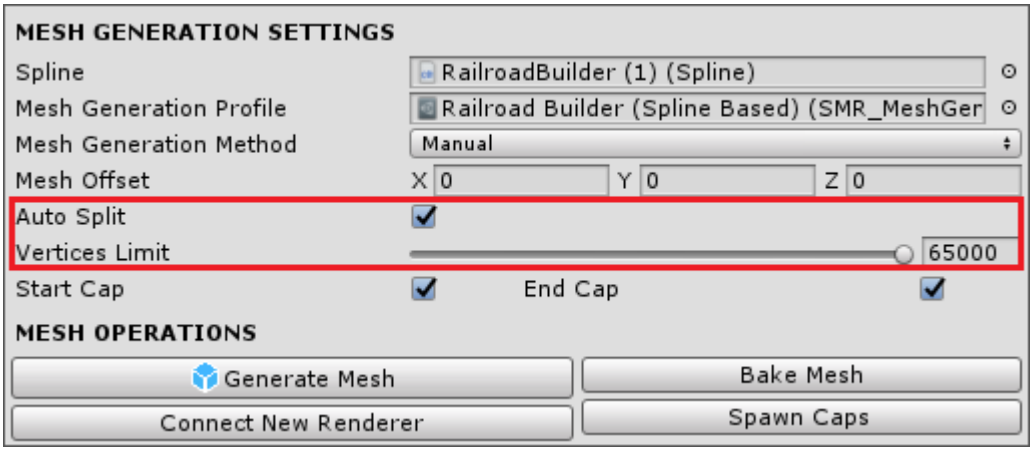
Whenever you need to update the railroad caps without regenerating the mesh, you can use the “Spawn Caps” operation in order to do that. This can be very useful, if you’ve enabled or disables the start or end caps, but haven’t made any changes to the railroad.



6.2.15. Auto split

Since Unity has a limit of 65000 vertices per rendered mesh, if you are building a very long railroad, eventually you may need to split your railroad in order to avoid reaching the limit.

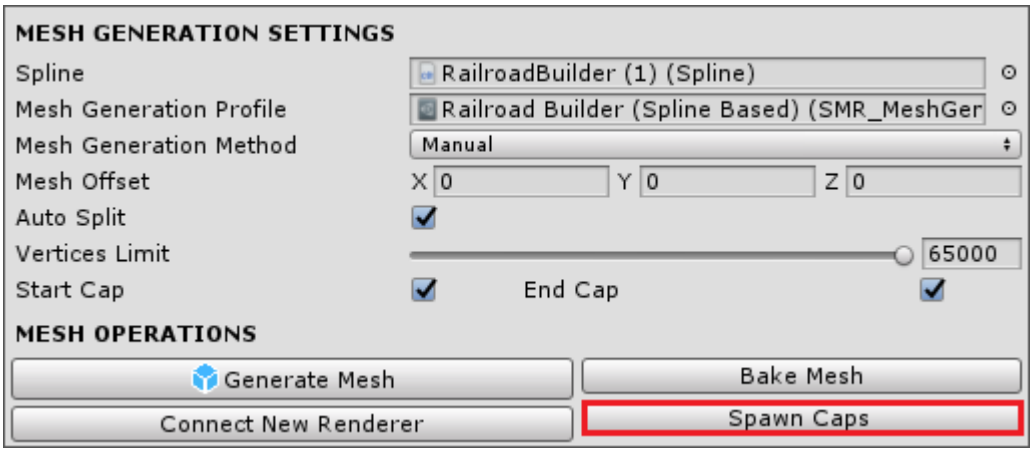
You can manually split by using the Split Spline feature. However, in order to make things easier, you can also use the Auto Split feature and control vertices limit for each generated segment.



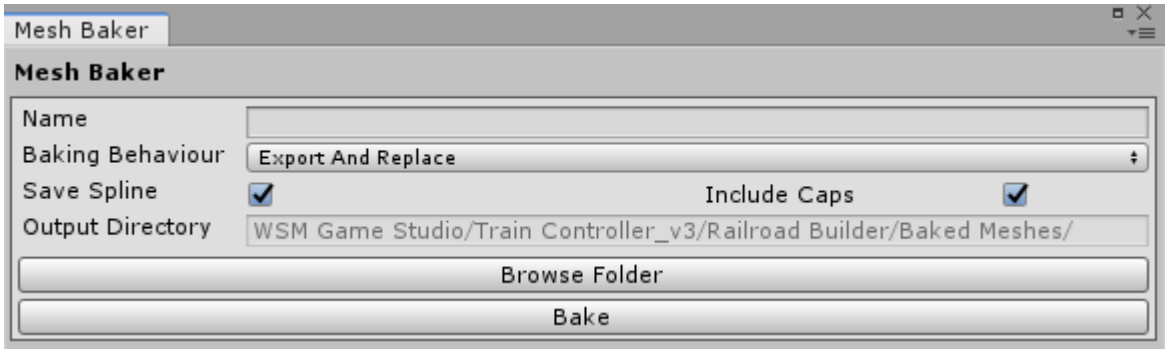
Whenever you create a new railroad builder, the “Auto Split” feature is enabled by default and the “Vertices Limit” is set to the Unity default limit (65000).

6.3. Railroad Prefab Export (Mesh Baker)

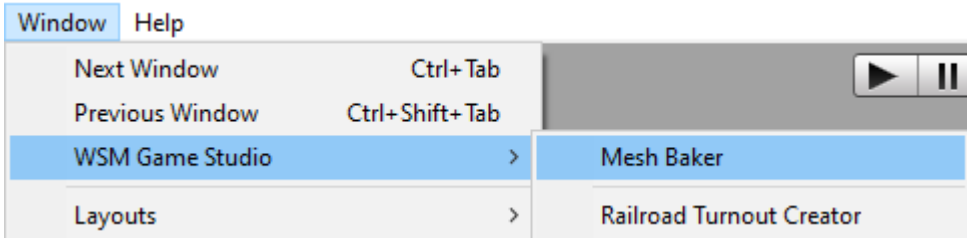
When you finish editing your railroad, you can bake a static version of it to improve the performance of your game. This feature saves the generated mesh and creates a ready to use prefab that can be used to replace the Railroad Builder on your scene.



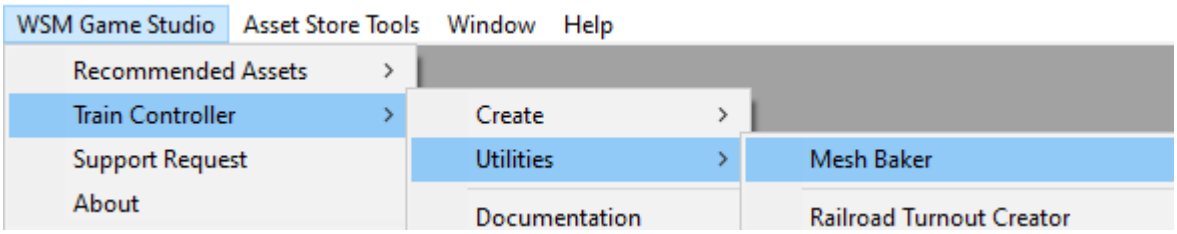
By clicking on the “Bake Mesh” button the Mesh Baker window will appear.



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

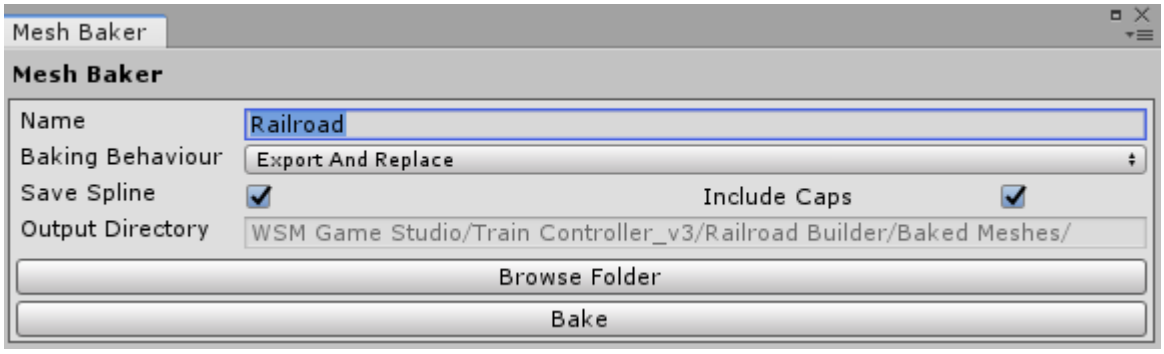


Alternatively, it can also be found under “**WSM Game Studio > Train Controller > Utilities > Mesh Baker**”

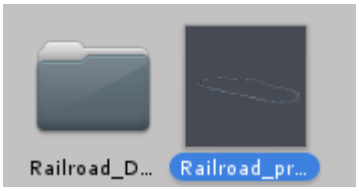


By default, the prefabs will be baked into the “Baked Meshes” folder, but you can change the Output Directory by using the “Browse Folder” button.

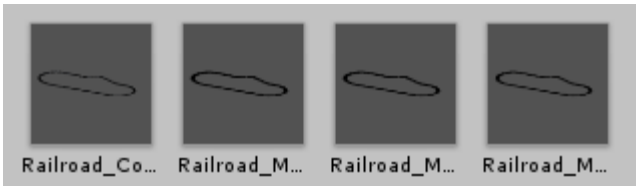
To start the baking process, all you need to do is select a name for your prefab, decide if you want to save the spline and/or include the caps or not and click on the “Bake” button.



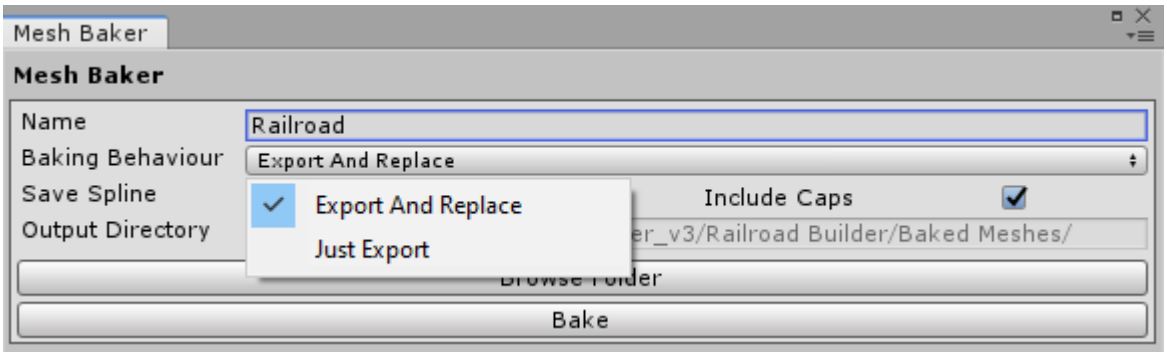
The “Bake” feature will automatically save the meshes and the prefab on the selected Output Directory.



The meshes are save on a “data” folder alongside the prefab

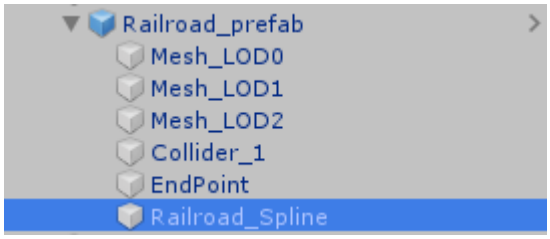


The generated prefab can be used to replace the Railroad Builder on your scene. This will reduce the scene loading time and increase performance. If you’ve let the default “Export And Replace” baking behaviour, the railroad builder will be disabled and an instance of the baked prefab will be created on your scene to replace it automatically. If you don’t want the automatic replacement to happen, select the “Just Export” option instead.



Note: If you want to change your railroad in the future you can keep the Railroad Builder as a disabled object on your scene instead of deleting it. That way you can always make adjustments and bake the generated mesh again.

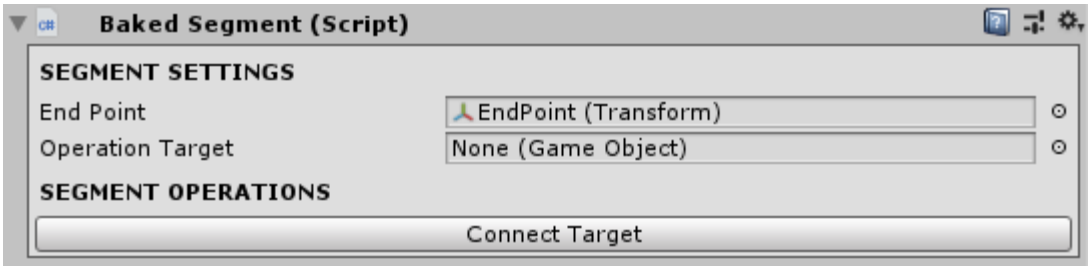
If you keep the “Save Spline” option enabled, a copy of the spline used to create this object will be saved as a child of the baked prefab.



Note: The saved spline is supposed to be a backup of the original spline used to bake the prefab, but it can also be used normally by any spline related feature.

6.3.1. Baked Rail Segments

Starting with v3.4, all prefabs created by the Mesh Baker will have a Baked Segment component attached to them by default.



This is an auxiliary component that works just like the railroad builder Connect Target operation. It was included to make it easier to connect baked railroad segments to each other, or to any railroad builder instances on your scene.

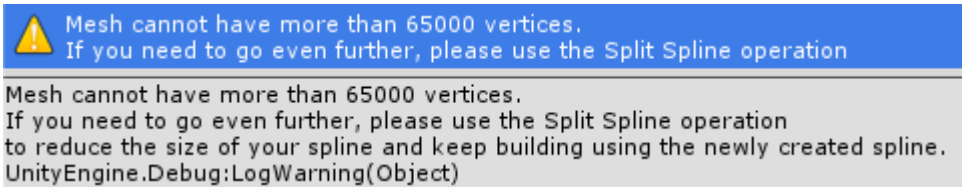
Note: The EndPoint property holds a reference to the transform located at the end of the baked segment. It is used as reference by the Connect Target operation. **DO NOT** change the endpoint property reference or move the EndPoint child transform manually.

6.4. Connecting Baked Rails, Railroad Builders and Turnouts

Whenever you need to connect railroad builders, baked rails or turnouts to each other, use the “Connect Target” operation (railroad builder connect target, baked segment connect target)

6.5. Building Long Railroads

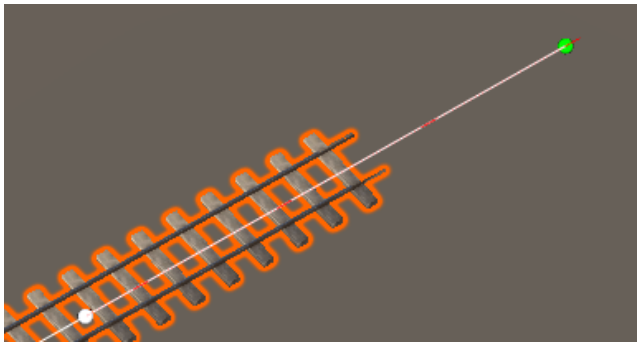
If you intend to build super long railroads for open world games for example, eventually you may see a warning like this:



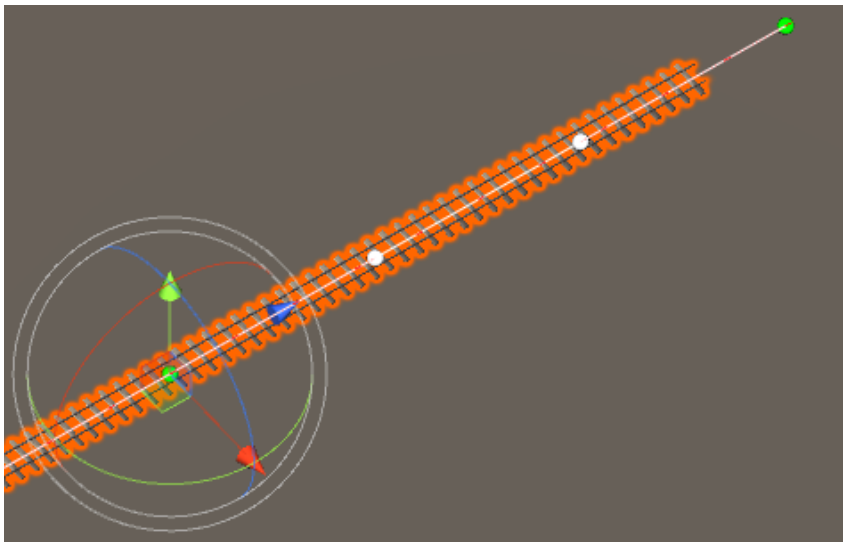
This happens because Unity has a limit of 65000 vertices per mesh. This limit exists to avoid performance issues when rendering large meshes. This limit is also very useful to apply Occlusion Culling on your railroad and increase your game performance. If you see this warning, just follow the instructions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

Note: The process described below is only necessary if the Auto Split feature is disabled, as it is a step-by-step guide on how to manually split long meshes manually to avoid reaching the Unity vertices limit.

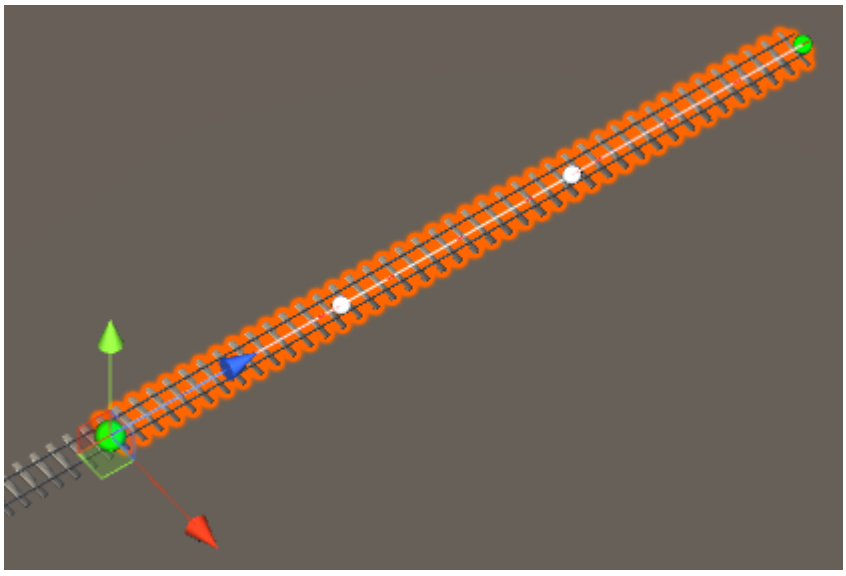
With the Railroad Builder object selected, go to the end of your mesh and check if the mesh ending matches the last control point. It probably won't match, like in the sample image below.



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



And then use the Split Spline operation to start a new railroad section.



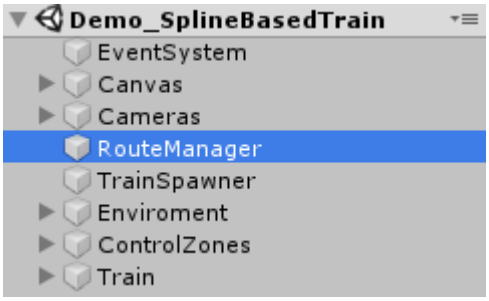
Note: Real time mesh generation is disabled automatically when the max number of vertices is greater than 65000. This is the default behaviour to increase the editor performance while building your mesh, since processing large meshes data in realtime is costly.

You can change back to the real time mesh generation manually after reducing the rails, but, for very long railroads it's recommended to use the manual mesh generation button to update the mesh after adjusting your railroad 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 (LOD0). By using it, it is possible to create railroad segments up to 1.35 Km each. To create railroads longer than 1.35 Km, use the technique described above to create multiple railroad segments connected to each other.

7. Route Manager

The Route Manager is an auxiliary component that can be used to define and reuse railroad routes on your scene. It can be used to assign routes to spline based trains, spawn trains on top of the rails using the Train Spawner or to automatically change spline based trains routes in-game using railroad switches.

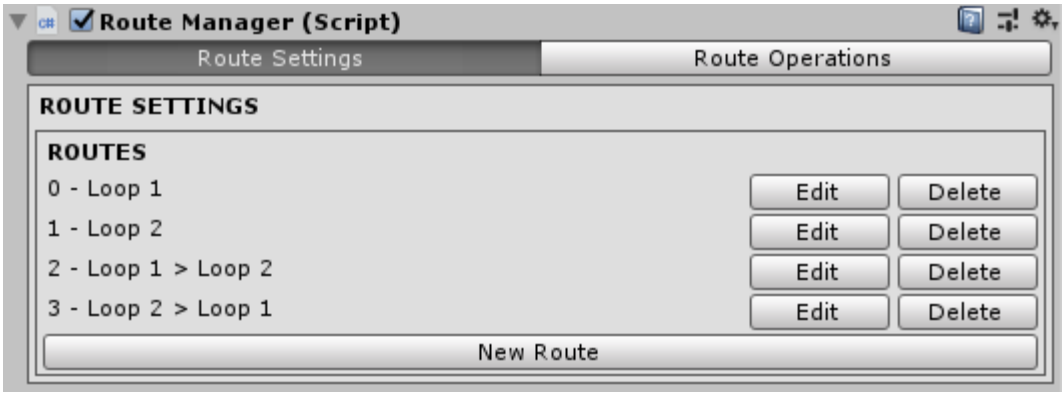


You can add a new Route Manager on your scene by using the default Unity object creation workflow: "**GameObject > WSM Game Studio > Route Manager**" or "**Right Click on Hierarchy > GameObject > WSM Game Studio > Route Manager**"

Note: The Route Manager is designed as a Singleton. Only 1 Route Manager is necessary per scene.

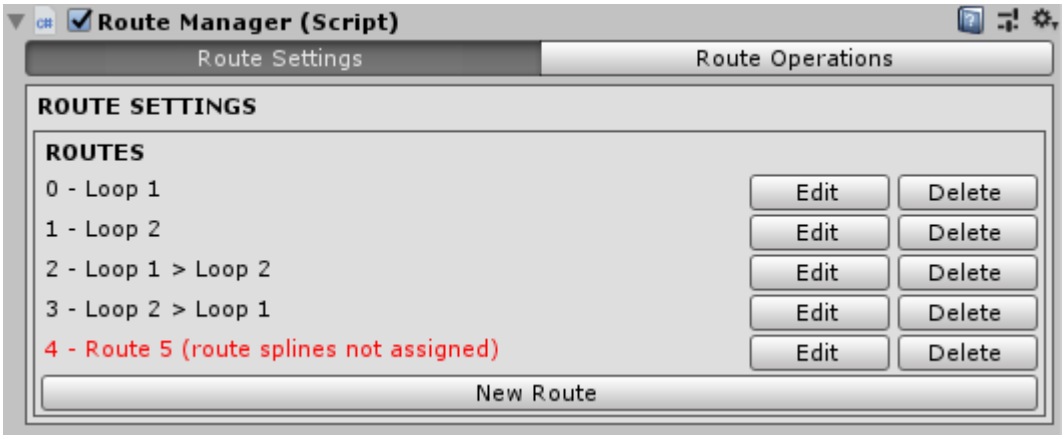
7.1. Creating a New Route

By using the route manager, you can create, edit and delete routes on your scene.



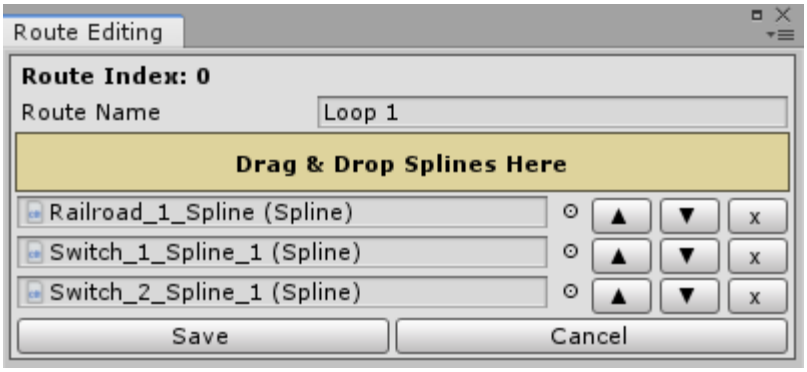
Note: A route is composed of one or more splines. The splines referenced on a route, can be part of a Railroad Builder, Baked Rail Segment and/or Railroad Switch.

To create a new route, click at the "**New Route**" button. Newly created routes, will be shown in red to inform that no splines were assigned yet.

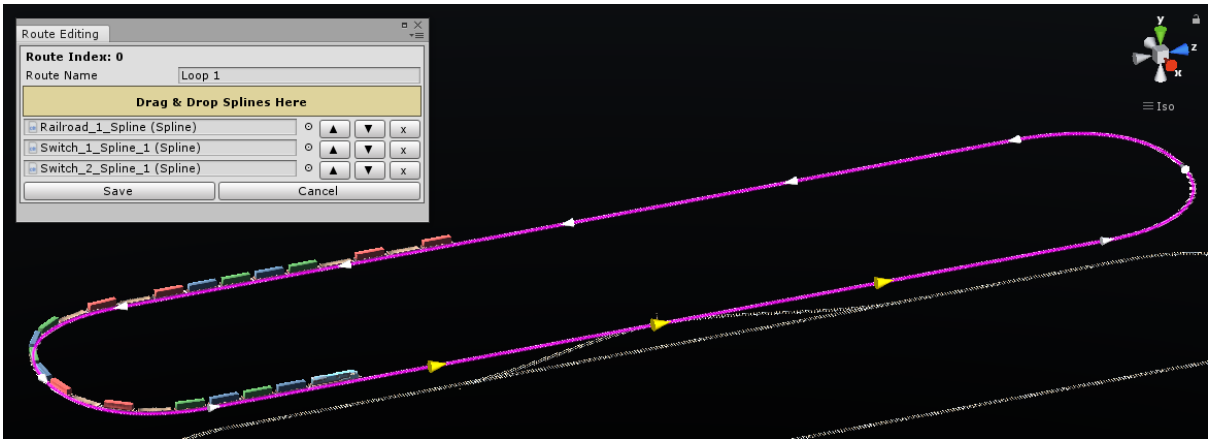


In order to edit the route name and assign splines to it, click the "**Edit**" button to open the **Route Editing** window.

Drag & Drop splines from your scene into the yellow box to add them to the route. You can use the up and down arrows to change the order of the splines assigned to the route. To remove a spline, click the x button.

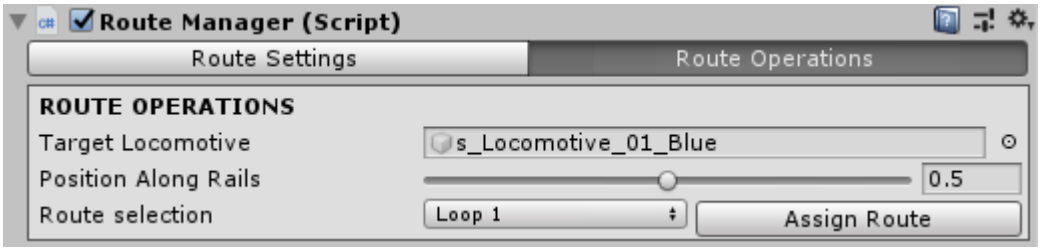


When the route editing window is open, the selected route will be visible in the Scene View. Cones are used to identify route direction, yellow cones are used to mark the start and end of each spline.



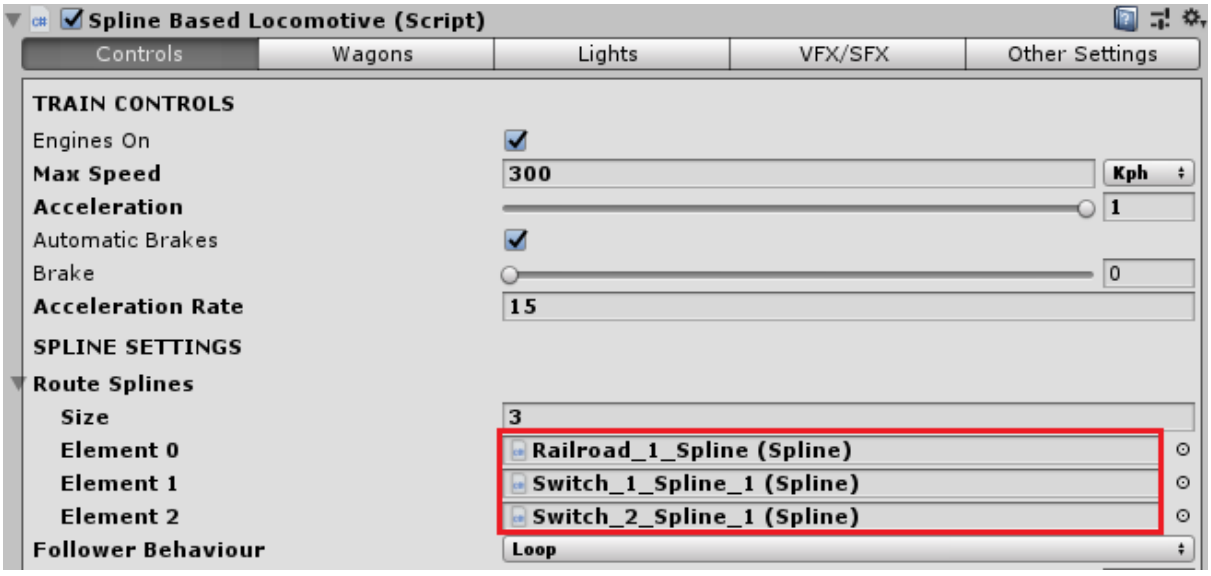
7.2. Assigning a Route to a Train

To assign a route to a train, select the target train locomotive, then select a route and click the “Assign Route” button.



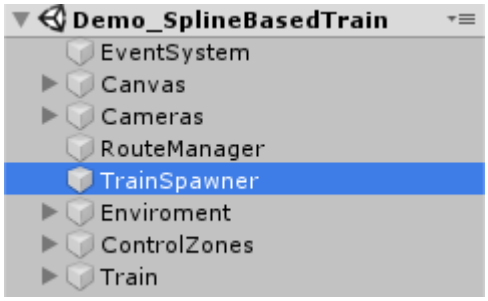
Note: The Route selection drop down will contain all valid routes available on your scene.

To verify if the route was applied successfully on a spline based train, you can check the target locomotive “Route Splines” property.



8. Train Spawner

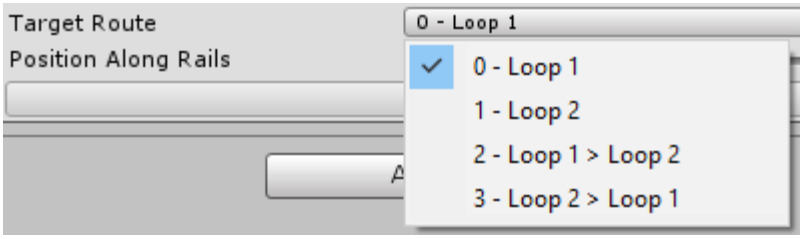
The Train Spawner is an auxiliary component that can be used to spawn locomotives, wagons and/or complete trains on your scene.



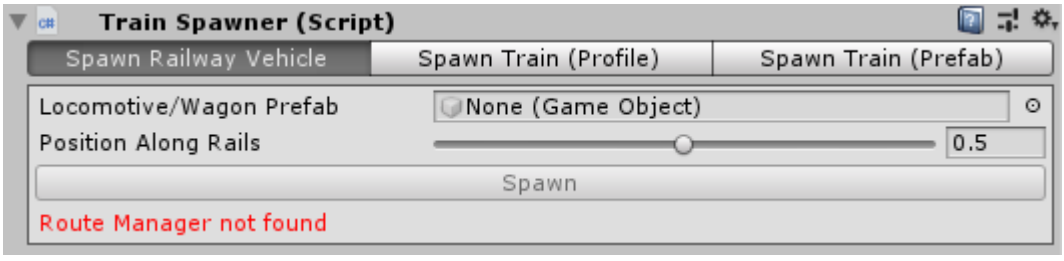
You can add a new Train Spawner on your scene by using the default Unity object creation workflow: "**GameObject > WSM Game Studio > Train Spawner**" or "**Right Click on Hierarchy > GameObject > WSM Game Studio > Train Spawner**"

> **Note:** The Train Spawner is designed as a Singleton. Only 1 Train Spawner is necessary per scene.

The Train Spawner was designed to make it easier to spawn trains on top of railroads. Therefore, in order to use it, you need to have at least one railroad already built and assigned to a route. For more detail about routes, please take a look at the Route Manager section.



The Target Route drop down will load automatically all routes available in your scene as shown in the sample image above. If a route manager is not found, a warning message will be shown instead.



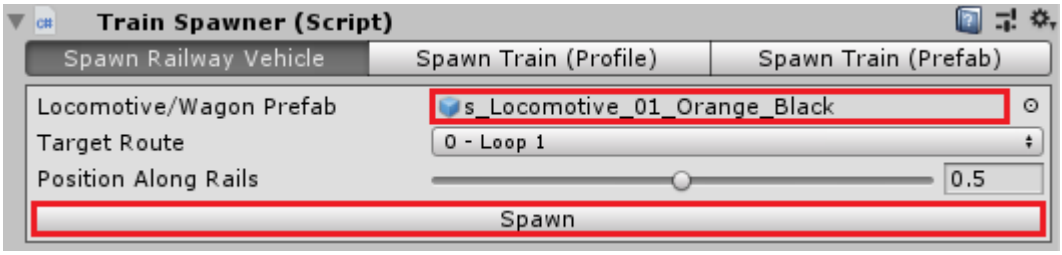
The "Position Along Rails" property is a range between 0 and 1 that defines the percentual position along the selected route. By default it is set to 50% (0.5).

8.1. Spawning Railway Vehicles

You can spawn single locomotives and/or wagons on your scene. This is useful to distribute wagons along your railroad for the in-game wagon connection feature.

To spawn a new railway vehicle instance on your scene:

- Select the locomotive or wagon prefab
- Select target route
- Adjust position along rails (if needed)
- Click on Spawn



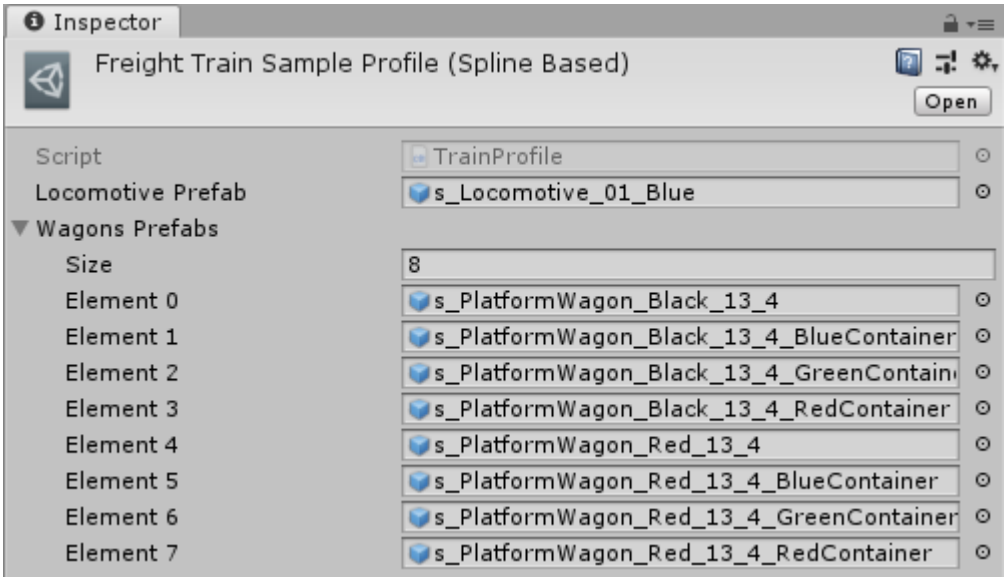
8.2. Spawning Trains

You can spawn complete trains on your scene either by using a train profile or a ready to use train prefab.

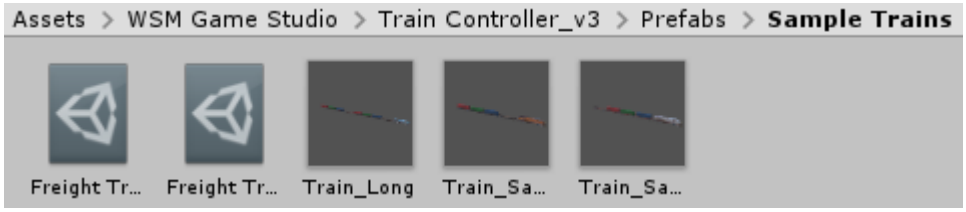
Note: Spline based wagons will automatically follow railroad curves on spawn. However, physics based wagons are spawned on a straight line behind the locomotive, therefore, physics based trains should be spawned on straight rails segments.

8.2.1. Spawn Train Profile

A train profile defines which locomotive and wagons prefabs will be instantiated to spawn the train.



Sample train profiles are included alongside the sample train prefabs.



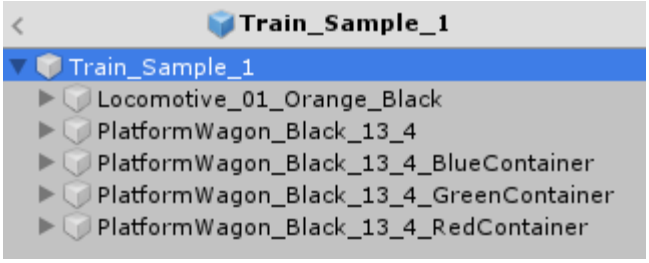
Note: Train profiles are Scriptable Objects, which means, you can create your own custom train profiles either by duplicating an existing profile file or by using the Unity Editor creation menu:
Right Click > Create > WSM Game Studio > Train Controller > Train Profile

To spawn a new train instance on your scene:

- Select a train profile
- Select target route
- Adjust position along rails (if needed)
- Click on Spawn

8.2.2. Spawn Train Prefab

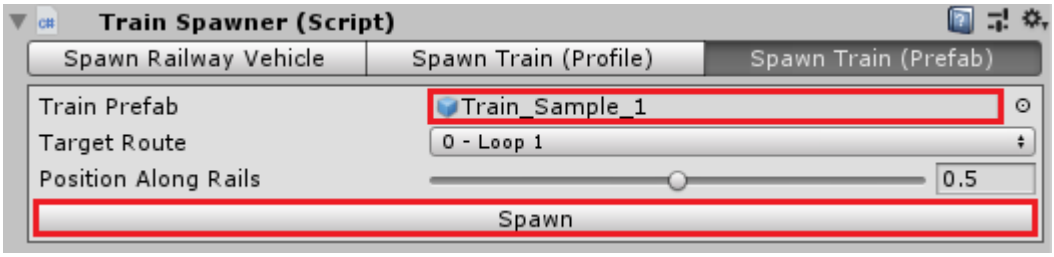
A train prefab is composed of a locomotive and wagons (connected to each other) set as children of an empty game object as saved as a prefab. For more information on how to create train prefabs, please take a look at the Connecting Wagons (Editor) section.



Note: The wagons of a train prefab must be referenced at the corresponding locomotive wagons list.

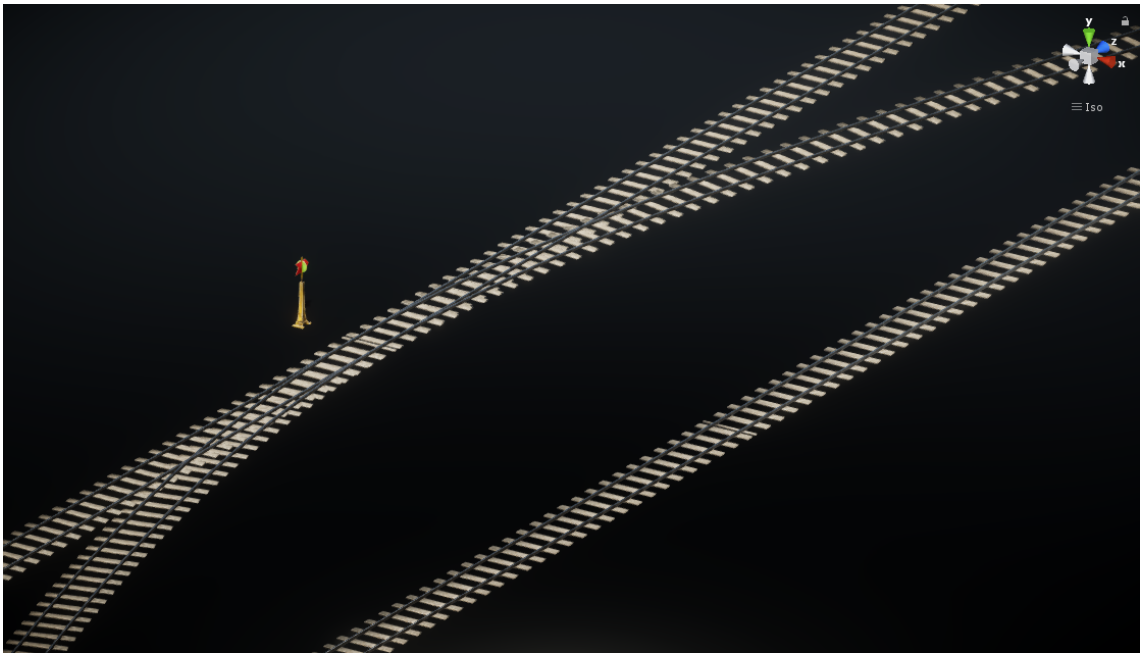
To spawn a new train instance on your scene:

- Select a train prefab
- Select target route
- Adjust position along rails (if needed)
- Click on Spawn


Train Spawner (Script)
Spawn Railway Vehicle
Spawn Train (Profile)
Spawn Train (Prefab)
Train Prefab
Train_Sample_1
Target Route
0 - Loop 1
Position Along Rails
0.5
Spawn

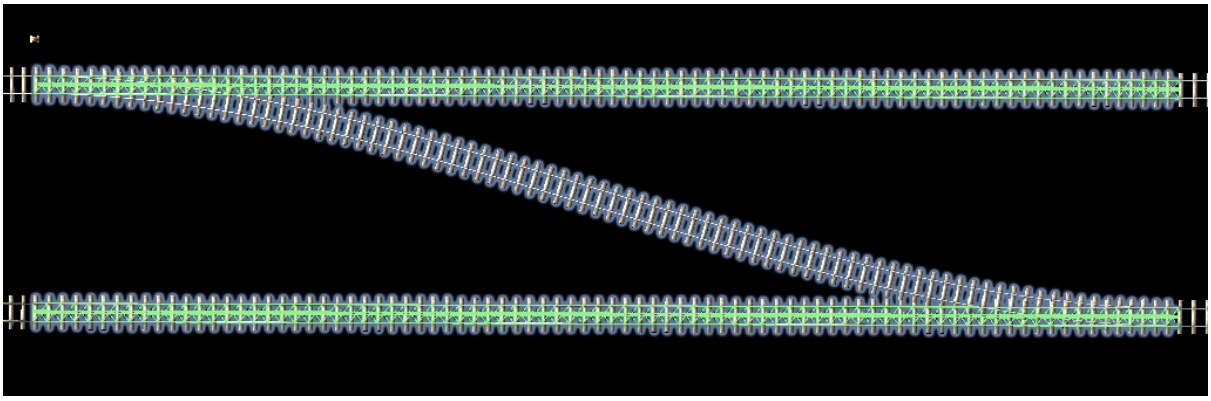
9. Railroad Switches (Turnouts)

In this section you will learn how to use, signalize and create custom Railroad Switches using the “Railroad Turnout Creator”.

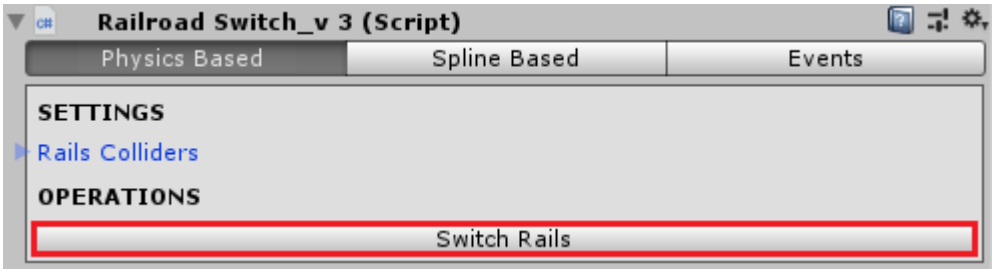


9.1. Physics based switches

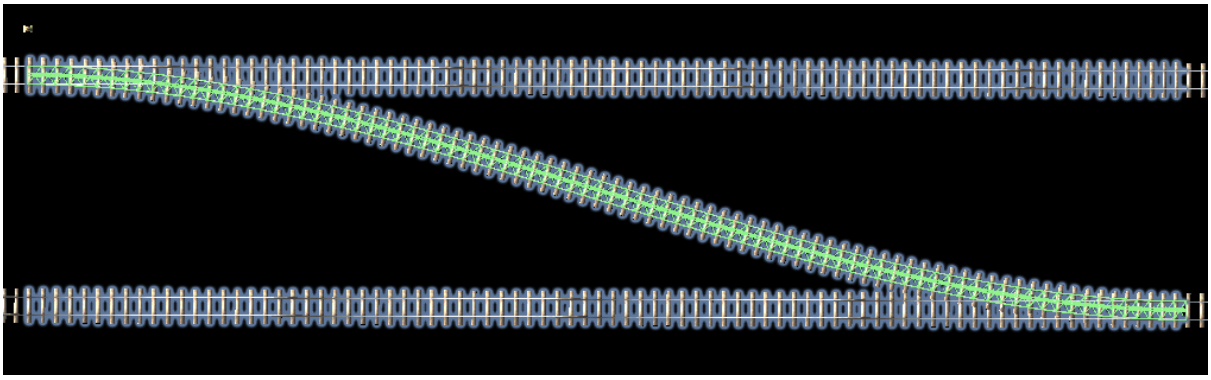
Physics based railroad switches works by enabling/disabling the rails “Guide Colliders” to redirect the trains in the desired direction. This behaviour is controlled by the “RailroadSwitch_v3” component.



You can change the initial direction of a physics based switch in your scene by using the “Switch Rails” button on the inspector.

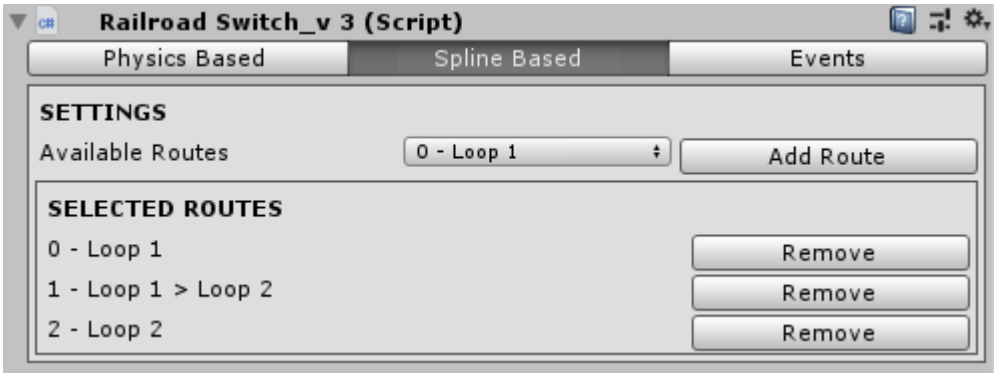


For example, this is the same railroad switch, after switching the active rails.



9.2. Spline Based switches

Spline based railroad switches works by changing spline based trains current route at runtime.



The routes affected by the railroad switch needs to be selected under the “Spline Based” tab of the RailroadSwitch component.

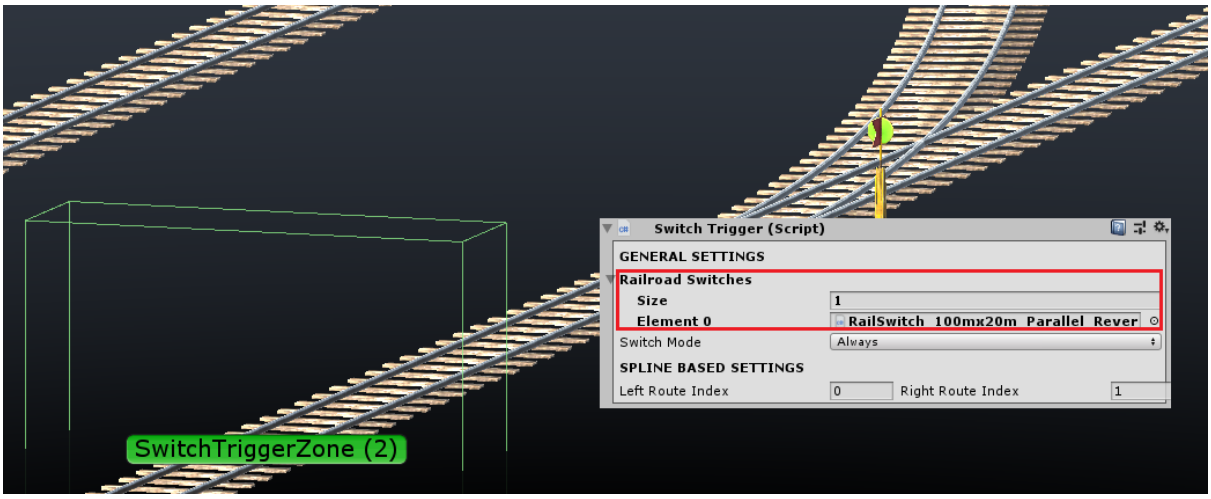
Note: Spline based switches depend on pre-existing railroad routes on your scene. For more detail about routes, please take a look at the Route Manager section.

9.3. Automated switches

Switches can be activated automatically when a train enters a Switch Trigger Zone.

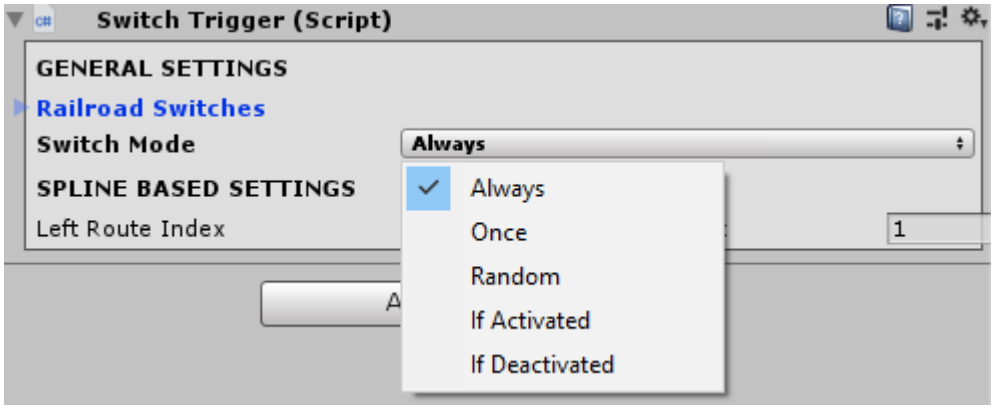
You can add a new Switch Trigger Zone on your scene by using the default Unity object creation workflow: “**GameObject > WSM Game Studio > Railroad Control Zones > Switch Trigger Zone**” or “**Right Click on Hierarchy > GameObject > WSM Game Studio > Railroad Control Zones > Switch Trigger Zone**”

Switch Trigger Zones can activate one or more Railroad Switches at the same time. You just need to drag & drop the railroad switches you want to control to the “Railroad Switches” property of the Trigger in your scene.



Note: Railroad switches can also be activated manually by calling the "SwitchRails" method of the "RailroadSwitch_v3" component.

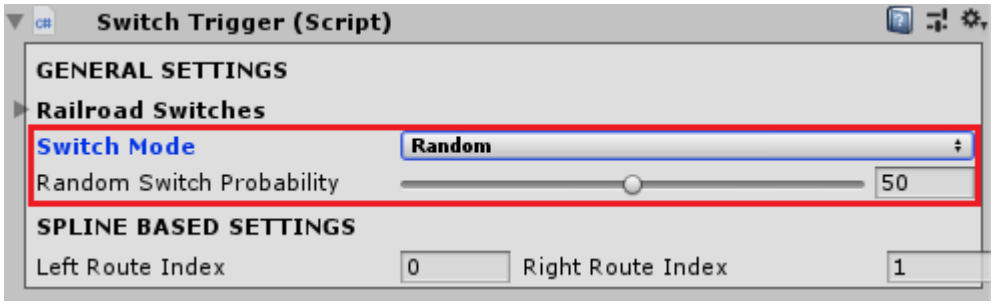
Switch Trigger Zones are a powerful way to automate your railroad. It allows you to create custom complex railroad traffic control within minutes, just by selecting the desired switching mode.



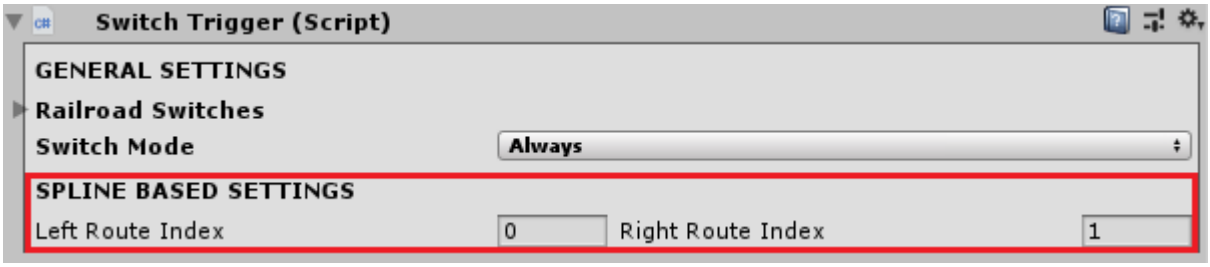
You can choose to switch the direction of the rails every time a train hits the trigger (always), only once, randomly or only if the turnout is activated or deactivated.

"If Activated" and "If Deactivated" are useful to avoid derailing by creating safety check triggers before each railroad turnout.

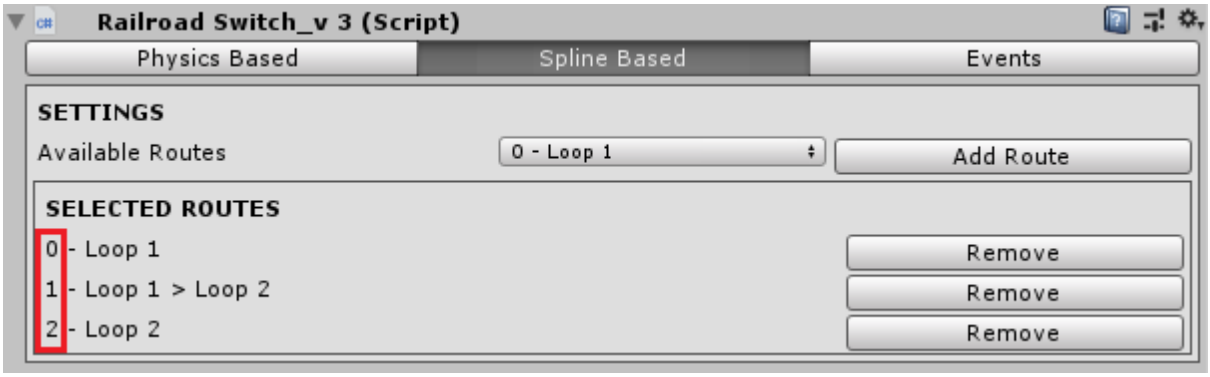
Random activation is controlled by the "Random Switch Probability" property, so you can fine tune the switch to activate more or less often. This property is only visible when the random switch mode is selected.



The Left Route Index and Right Route Index properties are used to identify which routes are affected by the switch trigger when using spline based trains.



The indexes are numbers corresponding to the Selected Route indexes in the Railroad Switch component. By default the indexes are set to 0 and 1, which corresponds to the first two selected routes.



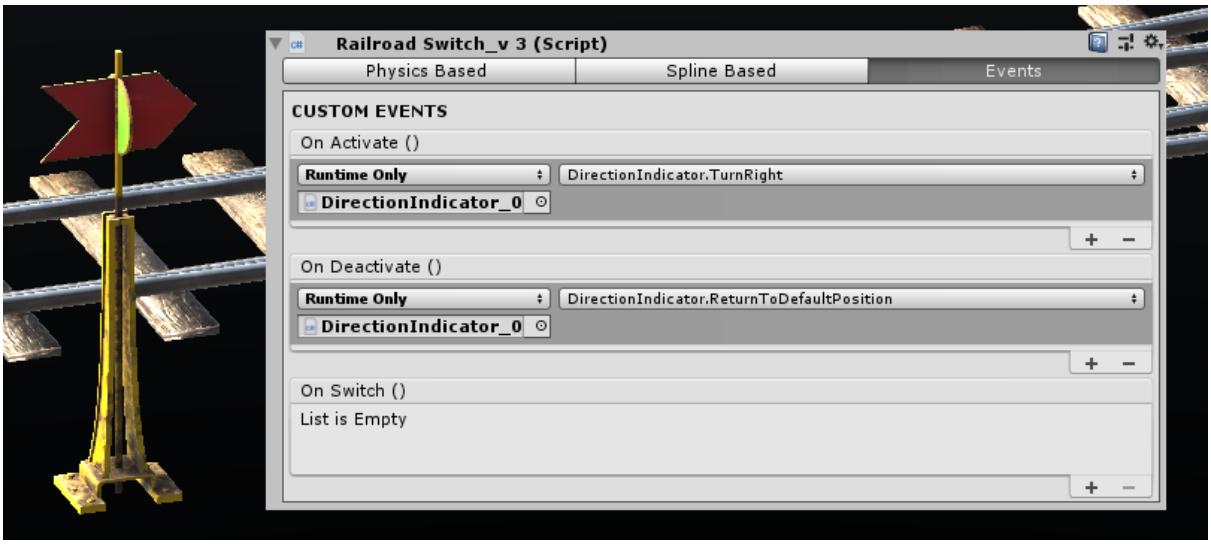
The “**OnActivate**”, “**OnDeactivate**” and “**OnSwitch**” event stacks can also be used to trigger custom events on your game as you wish. For example, they can be used for sinalization (see Switch Sinalization section for more details)

9.4. Switch Signalization

Under the “Prefabs/Sinalization” folder you will find the “DirectionIndicator_01” prefab.

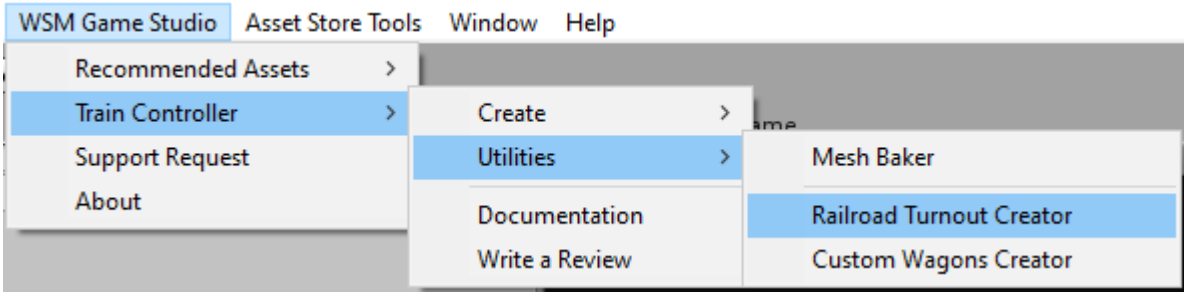
It is a ready to use animated direction indicator based on old real world direction indicators. It can point right, left or forward (default position)

Drag & Drop it to your scene and place it next to the railroad switch you want to sinalize. Select the railroad switch object on your scene and use the “OnActivate” and “OnDeactivate” event stacks to call the “TurnRight”, “TurnLeft” or “ReturnToDefaultPosition” methods like in the sample image below.

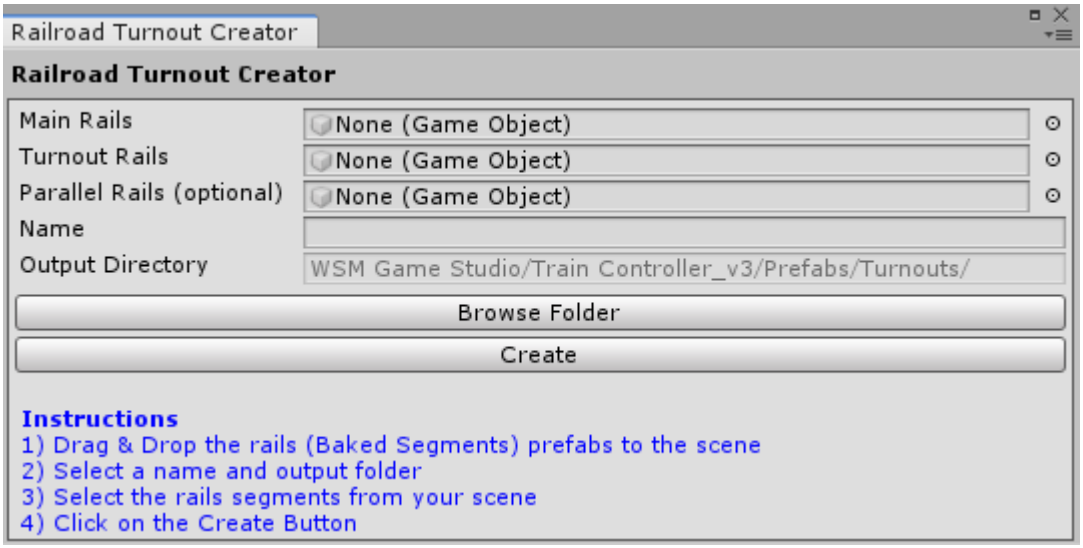


9.5. Custom Turnouts Creator

You can create custom railroad switch prefabs by using the built-in Railroad Turnout Creator.



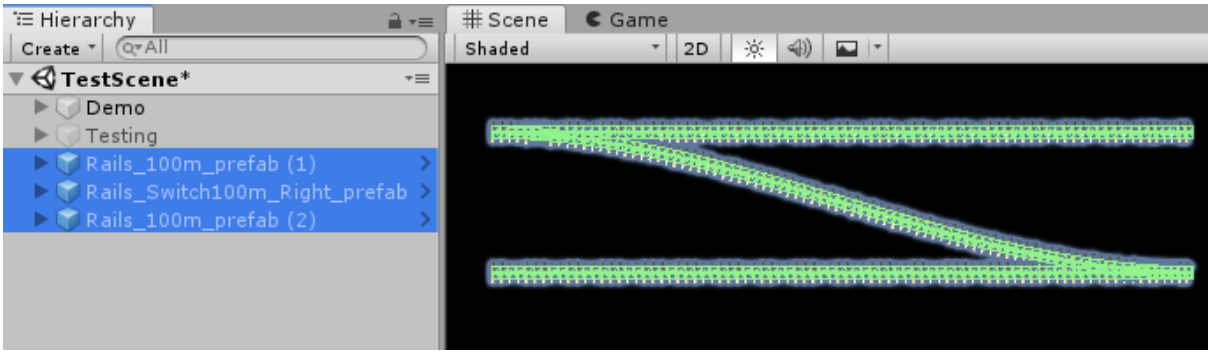
It can be found under “**WSM Game Studio > Train Controller > Utilities > Railroad Turnout Creator**”



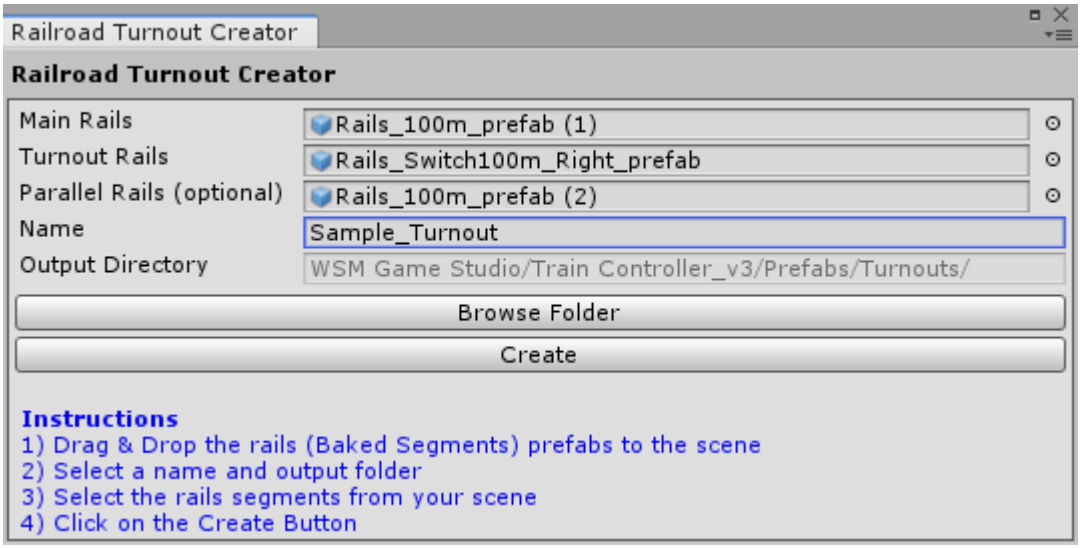
Custom railroad turnouts are built using baked rail segments. Custom rail segments with any length or curvature can be created by using the Railroad Builder and Mesh Baker (See Building a Railroad section for more details)

Note: When building custom turnouts for spline based trains, make sure to **save the spline** when exporting baked rails segments using the mesh baker.

Drag & drop the rail segments on your scene and move them into the desired position. If needed, the connect target operation can be used to move the rail segments.

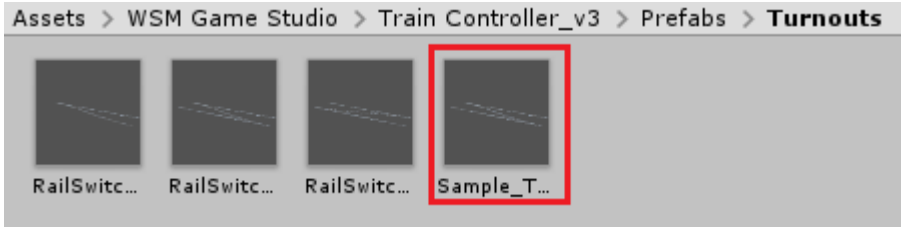


Open the Turnout creator window, drag & drop the rail segments on your scene into the corresponding rails field, select a name for your custom turnout and click on “Create”.

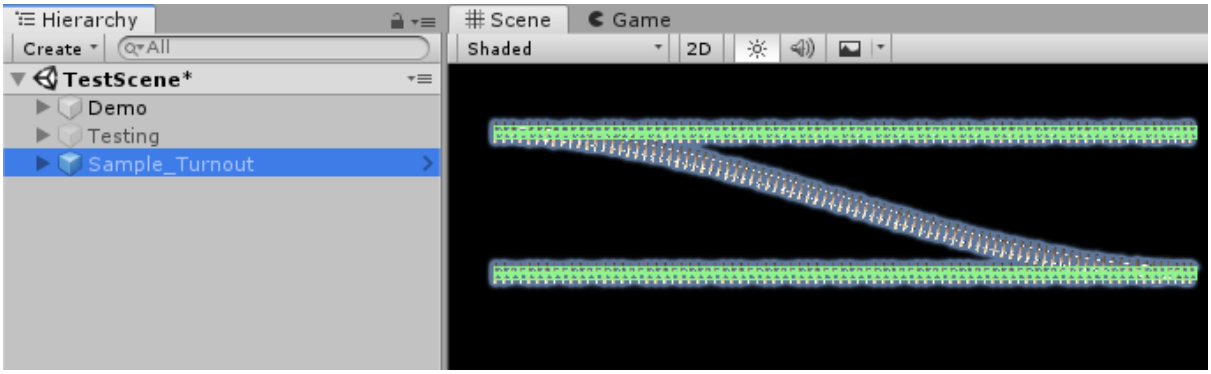


Note: "Main rails" and "Turnout Rails" segments are mandatory. "Parallel Rails" are optional.

If done correctly, the prefab will be saved on the selected output folder.



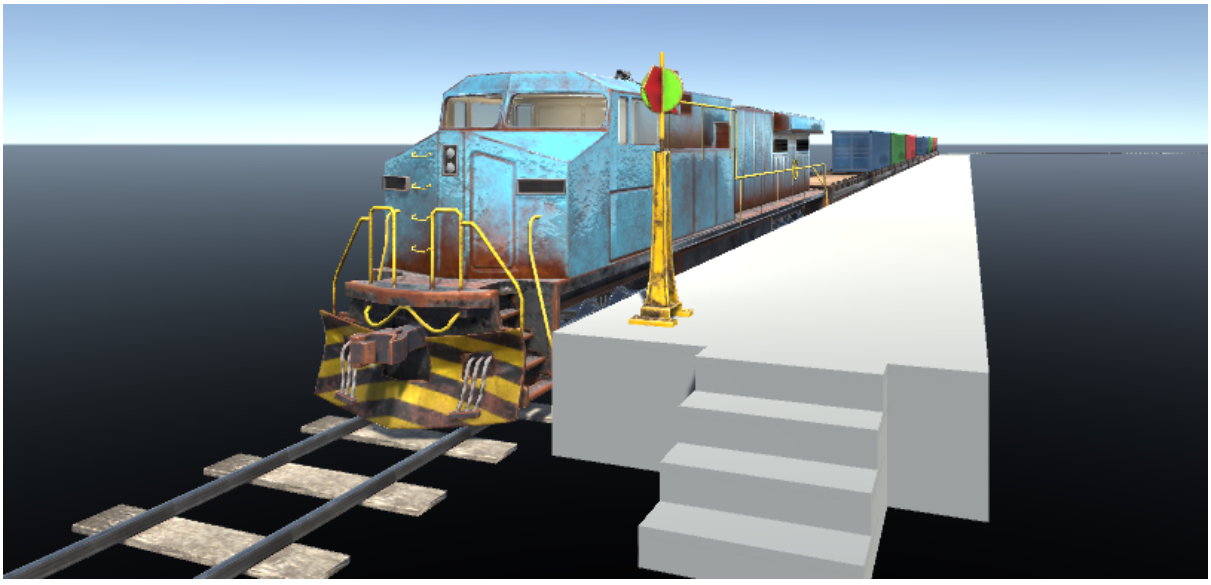
The rail segments used to build the turnout will also be replaced by an instance of the newly created turnout on your scene.



Note: The rail segment that was selected as "Turnout Rails" during the creation process will be disabled by default.

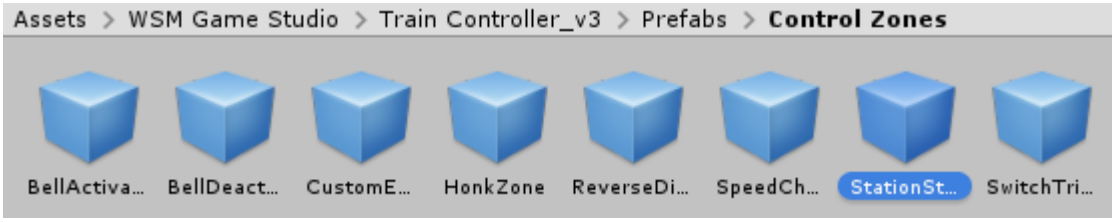
10. Train Stations

In this section you will learn how to simulate train stations

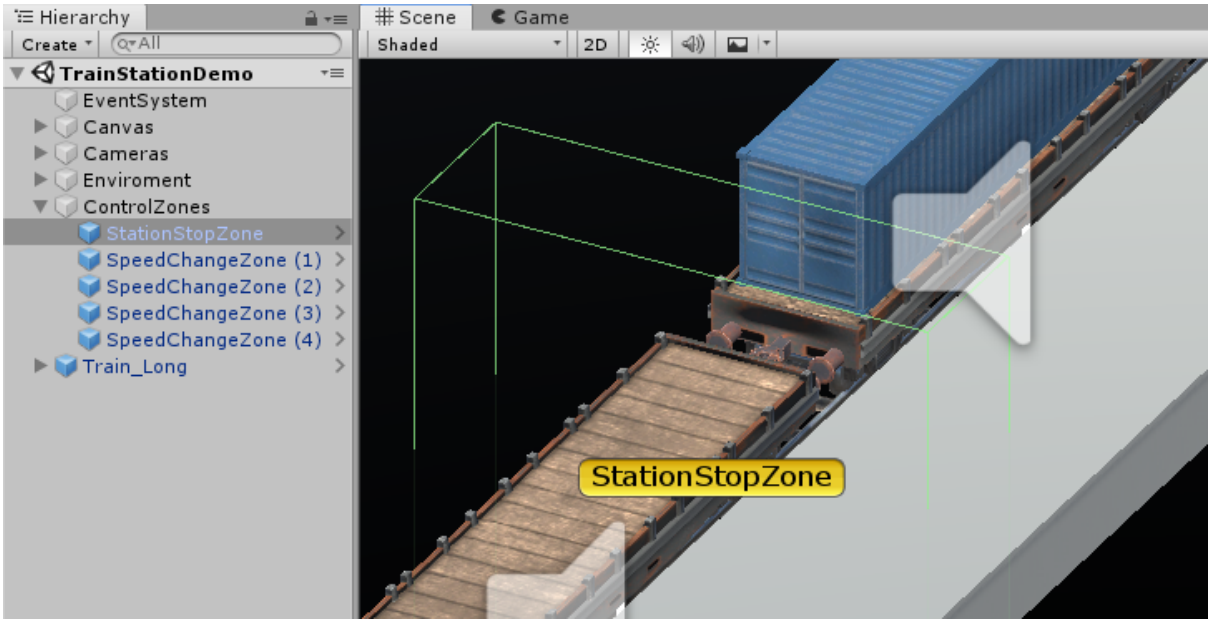


10.1. Stopping at a Station

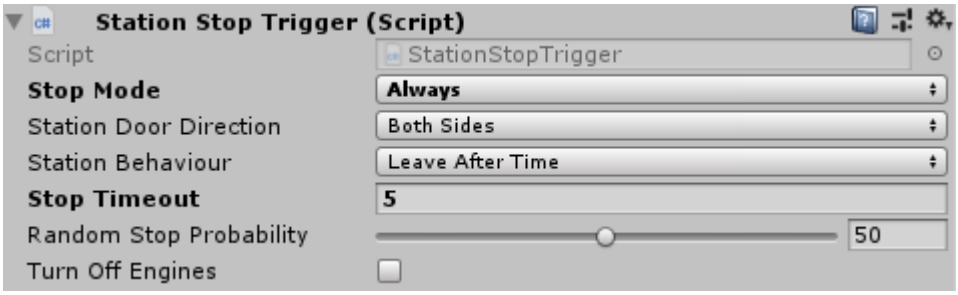
In the “Prefabs/Control Zones” folder you will find the “StationStopZone” prefab.



It can be used to simulate train station stops. Once a train hits a “StationStopZone” trigger, it will start braking until it stops.

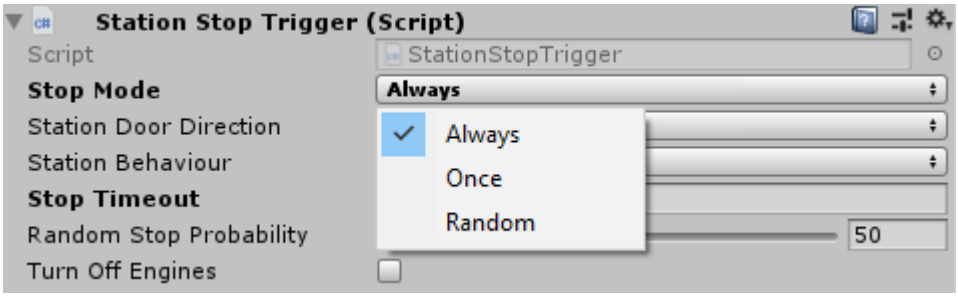


You can configure when the train should stop and how it behaves at the station after stopping.

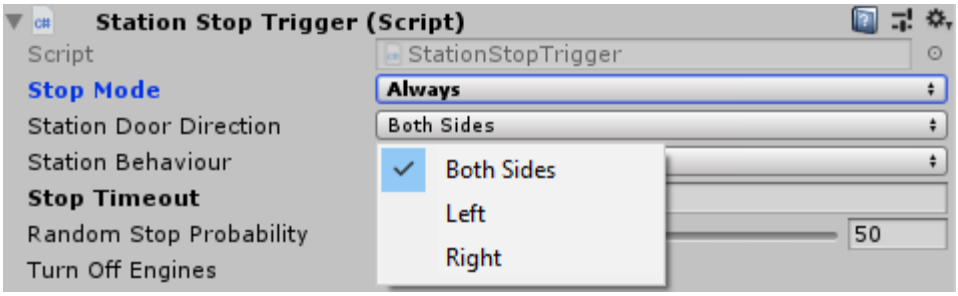


You can choose to always stop, stop only once or randomly stop at the station.

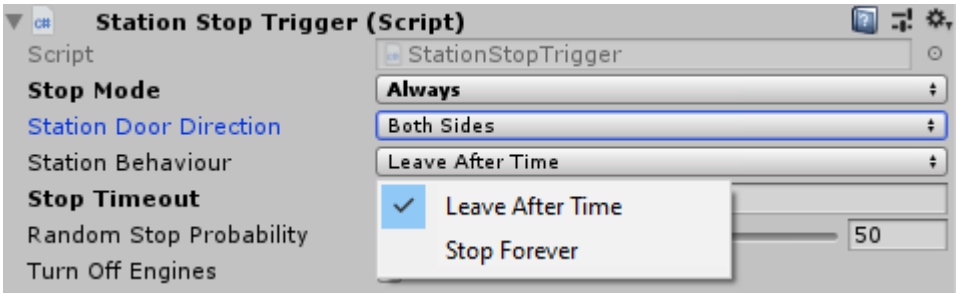
Random activation is controlled by the “Random Stop Probability” property, so you can fine tune if the train will stop more or less often.



The “Station Door Direction” property is used to open train doors automatically after stopping. It defines which side of the train the doors should open for that station.



You can also choose to leave the station after some time or stay forever. The “Stop Timeout” property defines how many seconds the train will remain on station before leaving.



The “Turn Off Engines” can also be used to stop the train on a final destination if needed.

10.2. Station Custom Events

You can call a custom event stack whenever a train is arriving or leaving a station. These events are:

- **Arriving At Station Events**
  - OnBrakesActivation()
  - OnStop()
  - OnAfterStopping()
- **Leaving Station Events**
  - OnLeaveWarning()
  - OnBeforeLeaving()
  - OnLeave()

Custom events should be set at the **Train Station Controller** component attached to each locomotive.

10.3. Train Doors

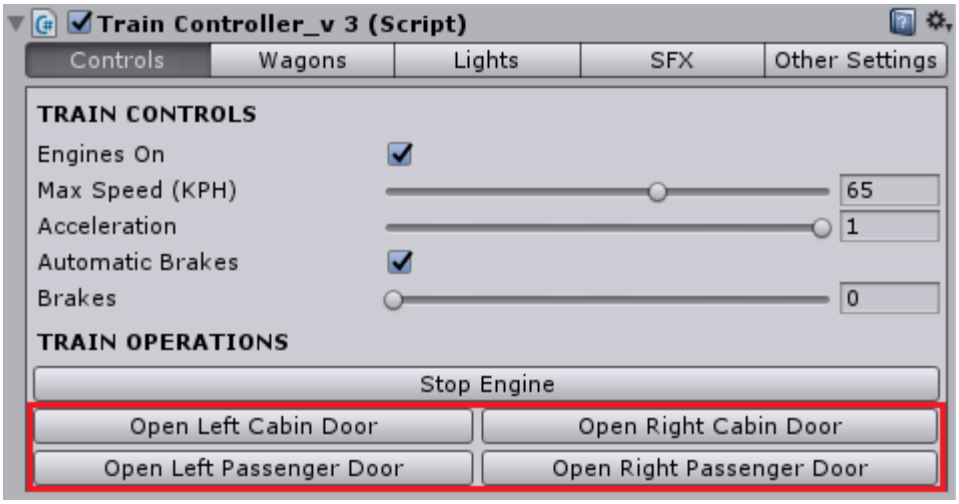
This package includes train doors open and close animations made with Mecanim, which means you will be able to customize the animations in the Unity Editor if you wish so.

By default, when the train reaches a station it will automatically open the passenger doors, if any (See the Train Stations section for more details).

But, if you wish to open/close the doors at any other time, you can call the open/close methods directly from a button click event or from a custom script for example. All you need to do is to call the methods from the **TrainDoorsController** component attached to your locomotive.

- **Open Door Methods**
  - OpenCabinDoorLeft()
  - OpenCabinDoorRight()
  - OpenPassengerDoors()
- **Close Door Methods**
  - CloseCabinDoorLeft()
  - CloseCabinDoorRight()
  - ClosePassengerDoors()

You can also call these methods in the Unity Editor by using the buttons available on the **Train Operations** section of the **Train Controller** script. This is useful to test your custom train doors.



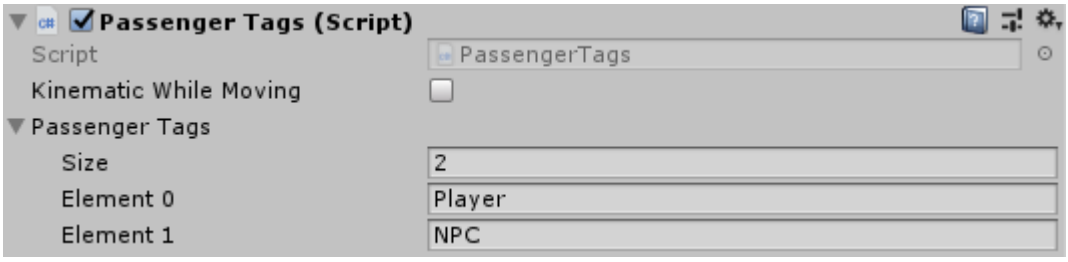
Note: Door related operations are only enabled on **Play Mode**.

10.4. Attaching Passengers

This package includes a simple method for attaching passengers and objects to trains and carrying them as passengers or cargo.

By default, all attached objects are set as children and inherit the wagon's movement. It is the same technique used to attach characters to moving platforms.

It works by attaching automatically objects tagged as any of the configured passengers tags to the wagons, if the object is positioned inside a wagon. These tags are set at the “Passenger Tags” component attached to locomotives.



Note: If the ‘Kinematic While Moving“ property is enabled, any physics based object will be set to kinematic while the train is moving.

Since there are several character controllers available on the market, and even custom character controllers developed for singular projects, it is not possible to guarantee all character controllers will behave properly while attached to a wagon.

It all depends on how your character controller handles movement. For example, character controllers that use animation root motion, may not inherit parent objects movement, since the animation root motion can override the character position.

In general, if your character controller works properly with simple moving platforms and properly inherits position and rotation when set as children of a moving game object, it should work fine.

Note: Keep in mind that physics based trains will shake at higher speeds, just like real world trains. This could cause minor involuntary movement (sliding) on physics based character controllers, caused by the physics engine handling collisions and inertia. It may also induce motion sickness on First Person Character controllers.

If you intend to have passengers or characters riding the trains **it is highly recommended to use** spline based trains on your project.

11. Scripting Reference

This section contains information about useful public properties and methods that can be safely accessed by custom scripts.

11.1. Locomotives and Wagons

- **ILocomotive Properties**
  - bool EnginesOn { get; set; }
  - float Acceleration { get; set; }
  - bool AutomaticBrakes { get; set; }
  - float Brake { get; set; }
  - float MaxSpeed { get; set; }
  - SpeedUnits SpeedUnit { get; set; }
  - bool BellOn { get; }
  - float Speed_MPS { get; }
  - float Speed_KPH { get; }
  - float Speed_MPH { get; }
  - ITrainDoorsController DoorsController { get; }
  - GameObject GetGameObject { get; }
  - List<GameObject> ConnectedWagons { get; }
- **ILocomotive Methods**
  - void ToggleLights();
  - void ToggleInternalLights();
  - void Honk();
  - void ToogleBell();
  - void ToggleEngine();
  - void UpdateDoorController();
  - void AddWagons(List<GameObject> newWagons);
  - void RemoveAllWagons();
  - void CalculateWagonsPositions();
  - void CalculateWagonsPositions(List<Spline> targetRails);
  - void AssignRoute(Route newRoute, float t);
- **IRailwayVehicle Methods**
  - void ToggleLights();
  - void ToggleInternalLights();
- **ITrainDoorsController Properties**
  - StationDoorDirection StationDoorDirection { get; set; }
  - bool CabinLeftDoorOpen { get; }
  - bool CabinRightDoorOpen { get; }
  - bool PassengerLeftDoorOpen { get; }
  - bool PassengerRightDoorOpen { get; }
- **ITrainDoorsController Methods**
  - void OpenCabinDoorLeft();
  - void OpenCabinDoorRight();
  - void CloseCabinDoorLeft();
  - void CloseCabinDoorRight();
  - void OpenPassengersDoors();
  - void OpenPassengersDoors(StationDoorDirection doorsDiretion);
  - void ClosePassengersDoors();

- void ClosePassengersLeftDoors();
- void ClosePassengersRightDoors();
- void UpdateWagonsDoorsControllers();

Note: Both TrainController_v3 (physics based locomotives) and SplineBasedLocomotive scripts inherit from ILocomotive and IRailwayVehicle interfaces.

Wagon_v3 (physics based wagons) and SplineBasedWagon scripts inherit only from the IRailwayVehicle interface.

11.2. Railroad Builder & Baked Rails Segments

- **Spline Properties**
  - bool Loop { get; set; }
  - float NewCurveLength { get; set; }
  - float NewCurveAngle { get; set; }
  - bool StaticSpline { get; set; }
  - Vector3Axis FlattenAxis { get; set; }
  - Vector2 ParallelSplineDirection { get; set; }
  - HandlesVisibility HandlesVisibility { get; set; }
  - float AutoHandleSpacing { get; set; }
  - bool FollowTerrain { get; set; }
  - float TerrainCheckDistance { get; set; }
  - int TerraformingWidth { get; set; }
  - int EmbankmentWidth { get; set; }
  - AnimationCurve EmbankmentSlope { get; set; }
  - Texture2D TerraformingTexture { get; set; }
  - float MinTextureBlending { get; set; }
  - float MaxTextureBlending { get; set; }
- **Spline Methods**
  - void AddCurve()
  - void AddCurve(Vector3 point, Vector3 upwardsDirection)
  - void RemoveCurve()
  - void ShapeCurve(Vector3 direction)
  - void ResetLastCurve()
  - void SubdivideCurve(int selectedIndex)
  - void DissolveCurve(int selectedIndex)
  - void SplitSpline(int selectedIndex)
  - void Merge(GameObject target, bool deleteTarget, bool disableTargetIfNotDeleted)
  - void ConnectTarget(GameObject target)
  - void BridgeGap(GameObject target)
  - GameObject AppendSpline()
  - void Flatten()
  - void Reset()
  - void ResetNormals()
  - GameObject CreateParallelSpline()
  - void Terraform(bool setHeights, bool paintTextures)
  - void BackupTerrains()
  - void RestoreTerrains()
- **SplinMeshRenderer Properties**

    - Spline Spline
    - SMR_MeshGenerationProfile MeshGenerationProfile
    - MeshGenerationMethod MeshGenerationMethod
    - Vector3 MeshOffset
    - bool AutoSplit
    - int VerticesLimit
    - bool StartCap
    - bool EndCap
- **SplinMeshRenderer Methods**
    - void ExtrudeMesh()
    - void PrintMeshDetails()
    - GameObject ConnectNewRenderer()
    - void SpawnCaps()
- **BakedSegment Properties**
    - Transform EndPoint { get; set; }
    - GameObject OperationTarget { get; set; }
- **BakedSegment Methods**
    - void ConnectTarget()
    - void ConnectTarget(GameObject target)

Note: Although, the EndPoind property of a BakedSegment as a “setter” it should not be changed. This property holds a reference to a Transform located at the end of the baked segment, that is used as reference by the Connect Target operations.

11.3. Route Manager

- **RouteManager Properties**
    - List<Route> Routes { get; set; }
    - float PositionAlongRails { get; set; }
- **RouteManager Methods**
    - void CreateRoute()
    - void DeleteRoute(int routeIndex)
    - bool ApplyRoute(SplineBasedLocomotive locomotive, int routeIndex, bool applyCustomPositionAlongRails)

11.4. Train Spawner

- **TrainSpawner Properties**
    - float PositionAlongRails { get; set; }
- **TrainSpawner Methods**
    - GameObject SpawnTrain(TrainProfile trainProfile, List<Spline> targetRails, float t, string trainName)
    - GameObject SpawnTrain(GameObject trainPrefab, List<Spline> targetRails, float t, string trainName)
    - GameObject SpawnTrain(GameObject locomotivePrefab, List<GameObject> wagonsPrefabs, List<Spline> targetRails, float t, string trainName)
    - GameObject SpawnRailwayVehicle(GameObject railwayVehiclePrefab, Vector3 position, Quaternion rotation)
    - GameObject SpawnRailwayVehicle(GameObject railwayVehiclePrefab, Vector3 position, Vector3 lookTarget)

- GameObject SpawnRailwayVehicle(GameObject railwayVehiclePrefab, List<Spline> targetRails, float t)

11.5. Railroad Switches

- **RailroadSwitch_v3 Properties**
  - bool Activated { get; }
  - List<GameObject> RailsColliders { get; set; }
  - List<Spline> RailsSplines { get; set; }
  - List<int> AffectedRoutes { get; set; }
  - UnityEvent OnActivate { get; set; }
  - UnityEvent OnDeactivate { get; set; }
  - UnityEvent OnSwitch { get; set; }
- **RailroadSwitch_v3 Methods**
  - void SwitchRails()
  - void UpdateActivationStatus()
  - void SwitchRailsByIndex(int[] indexes, bool activation)
  - void SplineBasedSwitchRails(ILocomotive locomotive, int leftRouteIndex, int rightRouteIndex)

11.6. Control Zones

- **CustomEventZone Properties**
  - UnityEvent CustomEvents { get; set; }
- **BellZone Properties**
  - ZoneTriggerType TriggerType { get; set; }
- **ReverseDirectionZone Properties**
  - ReverseDirectionMode ReverseDirectionMode { get; set; }
- **SpeedChangeZone Properties**
  - float TargetSpeed { get; set; }
- **StationStopZone Properties**
  - StopMode StopMode { get; set; }
  - StationDoorDirection StationDoorDirection { get; set; }
  - StationBehaviour StationBehaviour { get; set; }
  - float StopTimeout { get; set; }
  - int RandomStopProbability { get; set; }
  - bool TurnOffEngines { get; set; }
- **SwitchTriggerZone Properties**
  - SwitchMode SwitchMode { get; set; }
  - int RandomSwitchProbability { get; set; }
  - List<RailroadSwitch_v3> RailroadSwitches { get; set; }
  - int LeftRouteIndex { get; set; }
  - int RightRouteIndex { get; set; }

12. 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).

For more information about licensing, please refer to the Asset Store EULA and EULA FAQ.

13. Contact Info & Support

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

Support requests are now being handled exclusively by email at wsmgamestudio@gmail.com

Your request must contain the following information:

- Asset Store Invoice Number
- Unity editor version
- Detailed description of the issue (including screenshots if possible)

The best way to reach me is by email at **wsmgamestudio@gmail.com**

Asset Store	Facebook
Discord	Twitter
Sketchfab	Youtube Channel
Instagram	Artstation