TRAIN RAILS PLUGIN

For Unreal Engine 5.03 and newer.

🚂

TrainRails is an all-in-one solution for adding Trains & Railroads to your game or simulation.

Easily set up custom railroad paths using splines and set trains to drive along.

Bundled Graphics & Sound for a modern diesel train with customizable colors.

TABLE OF CONTENTS

Quick Start

After importing Train Rails into your project, there should be a folder at "Content/TrainRails" with several subfolders.

Navigate to TrainRails/BP and find "BP_TrainRails_Photoreal_RailsOnly"

Drag BP_TrainRails_Photoreal_RailsOnly into your level and place it on the ground.

TrainRails contains a Spline Component. There should be a square coming out on one side.
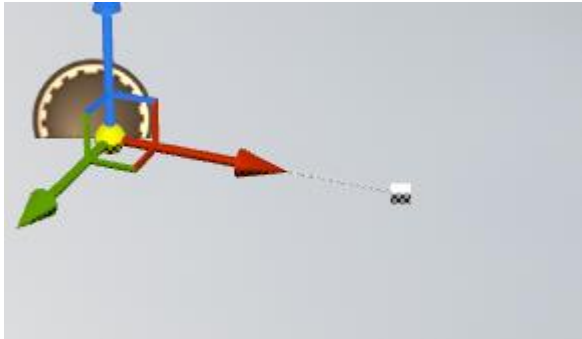
Right click on the square and choose "Duplicate Spline Point." Use the Gizmo to move the new Spline Point to another location far away from the original.

Right click on the spline point and duplicate it again. Move the next spline point to another new location. This spline will become a railroad when the game starts.

WARNING: Be careful not to make sharp turns on the path, as trains will look silly going around sharp turns.

For more help working with Splines: https://docs.unrealengine.com/4.27/en-US/BuildingWorlds/BlueprintSplines/HowTo/EditSplineComponentInEditor/

Locate BP_Train in the TrainRails folder and drag it anywhere into the level.

Select BP_Train and change the following properties:

- [Cars] Car Count = 3
- [Cars] Car Type = BP_Boxcar_Realistic
- [Cars] Engine Type = BP_EngineCar_Realistic
- [Rails] Rails = BP_TrainRails_Photoreal_RailsOnly *(this should match the Train Rails BP you placed in the level earlier)*

The BP_Train you placed should now show the correct realistic Engine & Boxcars

Press "Play" to play the scene in-editor.

The Rails should have spawned in along Rails_BP's spline.

The train should start at the beginning of the rail and begin moving.

🎉 Congratulations, you have a train! 🎉

Now try changing the Train's Desired Speed, Engine Start Position on Rail, Car Count, and the Loop Around Rail settings.

Blueprints Overview: Movement & Logic

TrainRails

Trains move along rails.

Three Rail Types are included in this plugin, and you can make more with your own custom meshes.

- BP_TrainRails_Base is a minimalistic base class with placeholder meshes

- BP_TrainRails_Photoreal_RailsOnly has photorealistic train rail meshes

- BP_TrainRails_Photoreal_GravelBed has the same photorealistic rails, now on top of a bed a gravel.

To Set up a Rail in the Level
1. Drag the desired Rail BP from the Content Drawer into the Level

2. Rotate the actor to the orientation you want the rail to start in

3. Find the spline point that is not located at the actor's root, and drag it to the desired location.

4. Right Click the spline point and choose "Duplicate Spline Point" then move the new spline point to a new location.

   a. For further help on Splines, refer to: https://docs.unrealengine.com/4.26/en-US/BuildingWorlds/BlueprintSplines/HowTo/EditSplineComponentInEditor/

5. Press "Play" and look at your new rail

Important Tips:

- Add more spline points along curves to make them look smoother

To Loop a Rail:

1. Make sure the last Spline Point is relatively near the first

2. Select the Rail BP in the World or Outliner

3. In the Details Panel, select the Spline component

4. Set [Spline] Closed Loop to True

A rail does not have to be looped for the Train to loop along it.

If the Rail is not looped and the Train is set to Loop around it, the Train will teleport back to the beginning after it gets to the end.

Custom Rails

To make a custom rail, make a child class of BP_TrainRails_Base and change the **RailMesh** variable to your desired mesh type.

- It is recommended you make a child class of BP_TrainRails_Base for each rail type

- The rail will look choppy when stretched along the spline if the mesh does not have enough length-wise subdivisions.

- Make sure all materials on the mesh enable "Used with Spline Meshes"
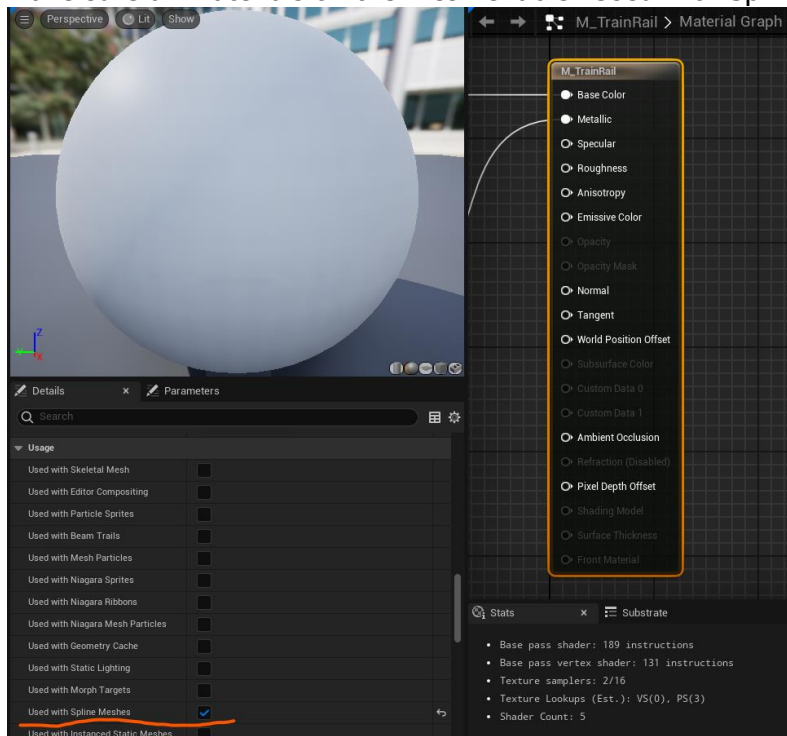
- The origin of your mesh must be its center

On BeginPlay, TrainRails will go over all spline points and connect them with spline meshes

Included variants:

- BP_TrainRails_Base – Has simplistic Placeholder rails. Parent class of other rails, so be careful when changing logic in this class.

- BP_TrainRails_Photoreal_RailsOnly – Child of TrainRails Base with Realistic railroad tracks that line up with the ground.

- BP_TrainRails_Photoreal_GravelBed – In addition to realistic rails, this blueprint also includes a raised gravel bed beneath the rails.

Train

BP_Train is the glue that holds everything else together.

A Train has an Engine and Cars. There will be one instance of the Engine at the front of the train, and as many Cars as you want following it. Both the Engine and Cars must derive from the BP_TrainCar class.

The Construction Script calls a function called GenerateTrain, which spawns the Engine and all the Cars as child actors.

Every tick, UpdateCarPositions is called. This uses the Train's DesiredSpeed and previous Position on the rail to calculate new positions for the Engine and all cars.

To Set Up a Train:

1. Place a Rail in the level if you want the train to be on a rail

2. Place a BP_Train into the level. It can be placed in any location, but it is recommended to place it near the rail you intend to use it with.

3. On the Details Panel, set the Train's Engine Type, Car Type, and Car Count

4. Set [Rails] Rails to the Rail BP you want this train to be on. The train will snap to the Rail

5. Set [Positioning] variables. If you leave the Engine Start Position On Rail at zero, train will start hanging off the rail. It is recommended to hide the start of the rail (e.g. in a tunnel or behind a building) or set the Engine Start Position to a high enough value that all train cars are past the initial point in the rail.

Some Useful Public Variables:

- Movement

    - Desired Speed – How fast should the train go

- o Loop Around Rail – If the train gets to the end, should it teleport back to the beginning of the current rail?

- Positioning

  - o Space Between Cars – How much space between the back of one car and the front of another?

  - o Engine Start Position – How far along the rail should the Engine start?

- Cars

  - o Car Count – How many cars to spawn AFTER the Engine

  - o Car Type – Class for Cars

  - o Engine Type – Class for Engine

- Rails

  - o Rails – Which instance of BP_TrainRails (or derived class) should the train move along?

Train Cars

All train cars and engines are based on BP_TrainCar.

You can make Child Blueprint Classes of BP_TrainCar to add your own cars & engines.

It is generally not recommended to edit BP_TrainCar, because any change will affect all train cars and engines, including the built-in ones. Consider making child classes instead.

A Realistic Box Car is included in this plugin.

Train Car's position is updated by the BP_Train.

Additionally, there are several moving parts being updated on a Train Car:

- The Front and Back Bogeys (wheel sets) are rotated based on the angle of the track

- As on real train, the car's rotation is set based on the Bogey locations

- Couplers at the front and back pivot towards the next car in line, if applicable.

- Wheels rotate based on train speed

Essential Components, Functions, Variables

- WheelsFront and WheelsBack are the origin for the car's Front and Back set of wheels. In a realistic train, these are used as the roots for each Bogey.

- Front/Back Attachment – Where should the next / previous car attach

- Front/Back Coupler Pivot – Used as the root for coupler rotation. Couplers between train cars rotate to keep the cars connected.

- ApplyColor – Input the Primary and Secondary colors for train car paint. If supported (such as on the Realistic Diesel Engine & Realistic Boxcar), the engine or car's paint will change color to match. The default can be set using Variables.

- ReleaseToPhysics – **Experimental** function to stop moving this car along the track and let physics take over. Great for train crashes or general chaos. Not fully tested, there may be issues! [See notes](#).

- Wheel Speed Multiplier – Wheels of different sizes should spin at different rates. Customize this rate so that the wheels rotate at the right rate. Wheel rotation rate is calculated by DesiredSpeed * Wheel Speed Multiplier

- OffsetAboveRails – How high above the rails this actor's should be. Ensure it is not too low or floating too high, the wheels should be sitting exactly on top of the built-in rails.

- MaxKosherDesiredSpeed – Used for calculating how fast this train is moving as a Percentage. Primarily used for Audio.

Setting Paint Colors

All Train Cars (including Engines) can have their colors set through instance-editable properties.

- ColorPrimary is the main color.
  If the car has only one color, this will be the only color.

- ColorSecondary is the secondary (stripe) color.
  If the car has only one color, this will not be used.

- bTwoColors can be toggled to switch between one and two colors.
  If it is true, cars will have 2 colors (by default: stripes)

In order for this to work properly, any custom car types you create will need to override the following two functions

- Setup Paint Material

- Apply Color

Trains can only use Blueprint Classes for their cars. If you want cars of specific colors on a Train BP,

1. Create a Child Blueprint class of the desired car

2. Set the default instance-editable color variables on the Child Blueprint

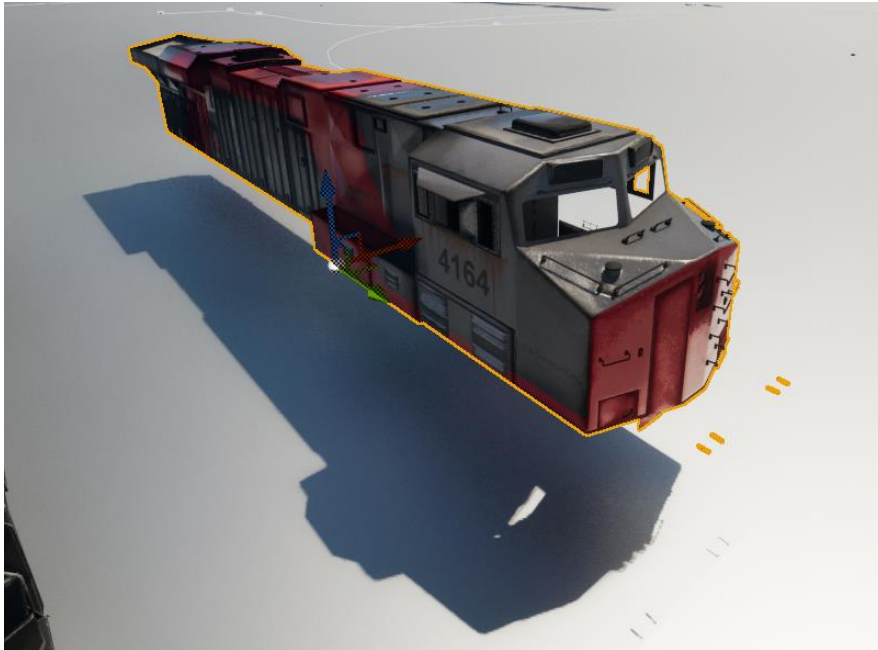3. Set the Train to use that new Child Blueprint class as its car (or engine) type

Engines

Engines are BP_TrainCar s with different audio & visuals.

This plugin includes a minimalist BP_EngineCar_Placeholder, a BP_EngineCar_Realistic, and a few color variations on the Realistic Engine Car.
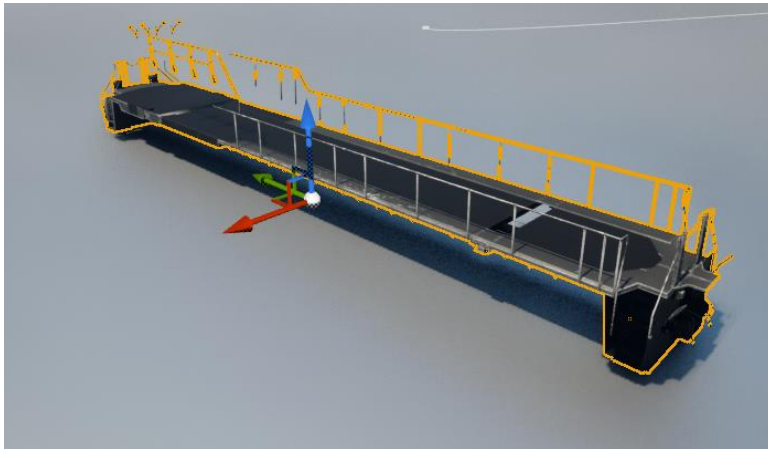
Train Parts Glossary for Realistic Diesel Engine

MainBody

The main body / chassis / outer shell of the engine.

Paint is customized on the material of this mesh, M_MainBody1

Backbone
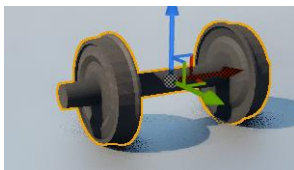

Supports the MainBody.

Bogey
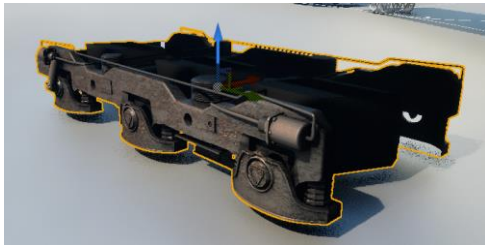
Pivot point must be at the center of the Bogey.

Attach Bogey mesh to the WheelsFront/WheelsBack Component.

Bogey holds wheels in place, and rotates independently of the train.

Wheels

Pivot point for the wheels must be at their center so that the wheels can rotate properly.

Make sure to properly assign wheels by overriding the SetupWheels function so they can be spun properly.

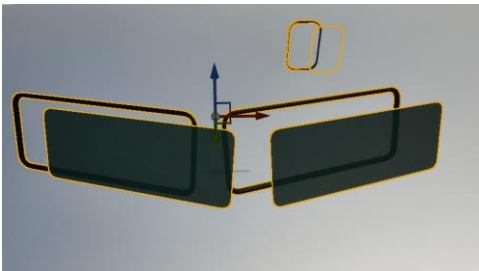To adjust their rate of spin, adjust Wheel Speed Multiplier.
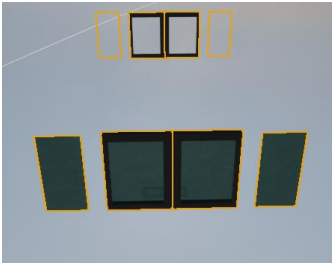
Couplers

Ensure their pivots are placed where they would be in real life so that they can pivot properly as the train turns.

Attach it to the Front CouplerPivot & BackCouplerPivot component.
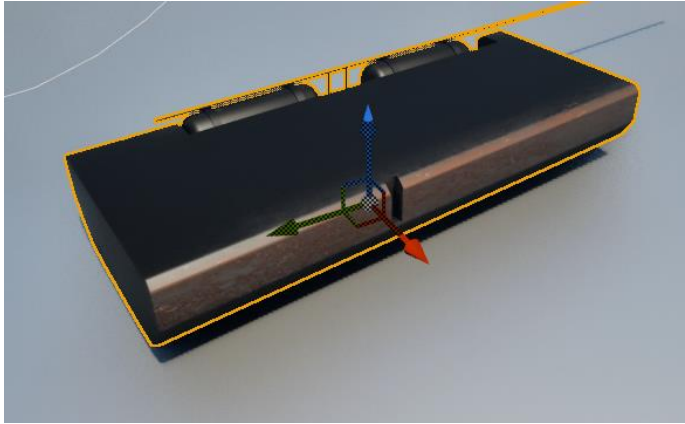
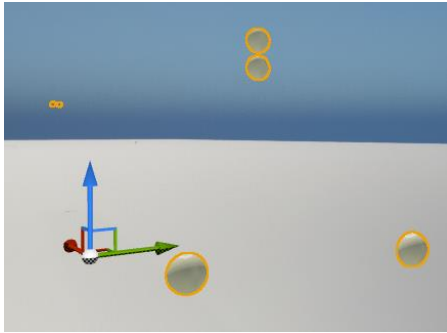*FrontWindow & FrontWindowGlass*

Window, Window-Side[2,3,4]
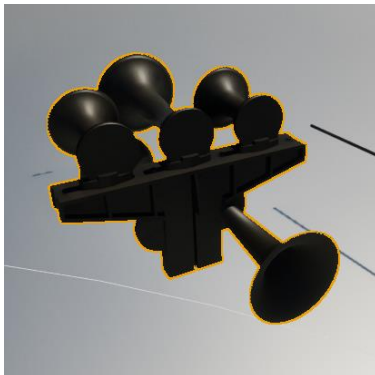

Door, Door-Window, Door-Window-Glass

Fuel-Tank
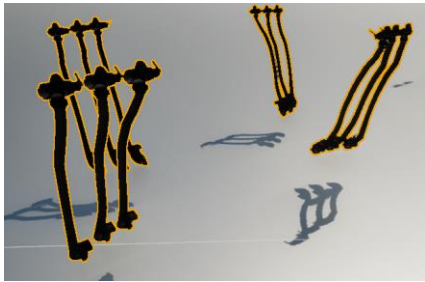

Headlight

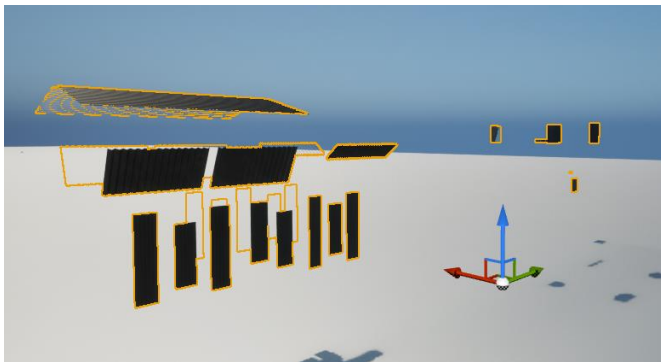Also includes a rear light.

Horn

Choo choo.

Hoses

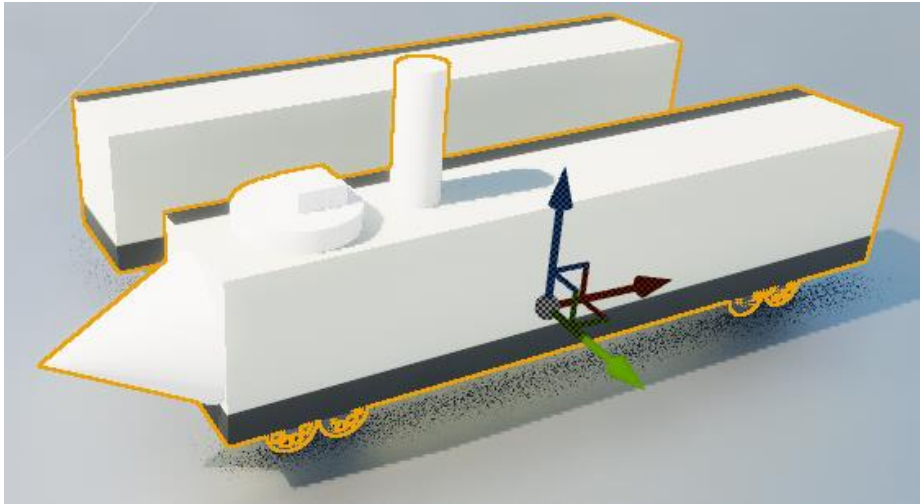Attached to Front and Back of Engine

Vent

All Vents on the engine.

Modern Diesel Engine Assembled (without bogeys & wheels)

Placeholder Engine & TrainCar

These works of modern art exist as simplistic placeholders.

They can be used as base classes for custom car & engine types.

Adding Your Own Train Car/Engine

To add another realistic train car or engine, bring in all desires meshes.

Your new car must be a child of BP_TrainCar

For more complex cars, make a duplicate of BP_EngineCar_Realistic (6 wheels) or BP_Boxcar_Realistic (4 wheels and a door)

For simpler cars, consider duplicating BP_EngineCar_Placeholder since it has fewer visuals to swap out.

In addition to setting all models correctly,

- Ensure that the Pivot Point of Wheels and Bogeys is their center, as they will rotate from this point.

- Ensure the positions of WheelsFront/Back, Front/Back Attachment, Front/Back Coupler Pivots (if you have couplers) are correct.
  (Wheels scene component should remain in between WheelsFront and WheelsBack)

- Override the "Setup Wheels" function to add all wheels to the Movement Wheels Right / Left arrays (or just use Right if your wheels and axles are combined in a single mesh)

- If you want to recolor your train, override function SetupPaintMaterial to create Material Instance Dynamics of your train's body, and override function Applycolor to set the appropriate parameters on those material instances.

- In the event Graph, Event BeginPlay and Event Tick might be disabled. Connect those nodes to any new node to enable them. You can delete the new node you added after they are enabled.

- Adjust the Wheel Speed Multiplier property and test until your wheels spin at the correct rate (this is multiplied by Desired Train Speed)

- Adjust Offset Above Rails if necessary until your train sits properly on any rails it is assigned to

- Set Color and Sound variables as appropriate

- If your car has a door that can open, override the Set Door Open function to open/close it.

Additional Goodies

Some useful blueprints are in the BP/Test folder. You may use these or write your own similar to them.

BP_HornBlower – Blow Train Horn
Blows a train engine's horn after a set delay. Optional loop.

- Delay Until Horn – After the game starts, how long until the horn is blown?

- Horn Length – How long should the horn sound for?

- Loop – If true, after the Horn is finished, the Delay will start counting down again, and then the horn will sound again. Process repeats.

- Train To Control – Which train should the function be called on?

Warning: If the Engine of the train does not have a Horn sound assigned, no horn sound will be played. **TrainHorn_Cue** is recommended since it can play for any amount of time and has a Doppler Effect.

RailSpeedOscillator
Allows a train to gradually speed up and slow down between two set speeds.

- High Speed – The train's max speed before slowing down.

- Low Speed – The lowest speed the train slows down to before speeding up again.

- TimeToChangeSpeed – How long should it take train to go from Low speed to High speed? Or from High Speed to Low Speed

- TrainToControl – Which train should we control the speed on?

Warning: Setting two Speed Oscillators to control the same train may have unpredictable effects.

BP_EngineCar_Realistic_RainbowPaintTest
Has some logic to cycle through the rainbow, changing paint color on tick. Not recommended for use.

Purpose is to test a variety of paint colors on the engine.

Additional Notes

Lumen
This product supports Lumen for Unreal Engine 5.0+

For more information on Lumen, see the Documentation:

https://docs.unrealengine.com/5.0/en-US/lumen-global-illumination-and-reflections-in-unreal-engine/

ReleaseToPhyics Function

ReleaseToPhysics functions are included in BP_Train and BP_TrainCar. They should be considered **experimental** and **not officially supported**.

The function can be called on the train to Derail it, or during a Train Crash.

There are bugs when the function is used with Realistic Engine and Car types.

They may work properly with BP_TrainCar, but not with its children.

I made use of them in the game Dead Line, but those trains only had one static mesh per car.

Dead Line Game Footage: https://youtu.be/dffDQC2D4hU

Some fun with turning off gravity: https://youtu.be/YMs4SXDgKVY

You may override them in child classes. If the child class has multiple meshes, you may need to turn off collisions & Physics on all but the Cube mesh. You can try doing this in the ReleaseToPhysics function.

Alternatively, you may try using Physics Constraint components to prevent physics bugs even if multiple components have physics and/or collisions:
https://docs.unrealengine.com/4.27/en-US/unreal
phy/Physics/Constraints/ConstraintsUserGuide/

Bonus

Here's a fun bug I had while working on this plugin - https://youtu.be/_IJYVAEVigY
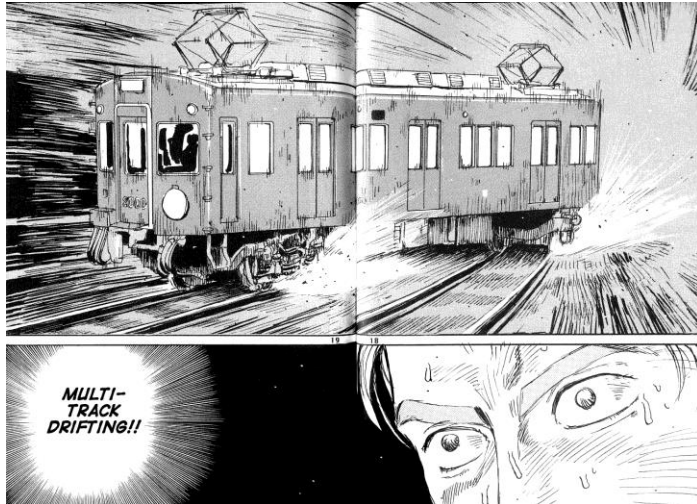
Multi-Track Drifting is not officially supported and should be considered utterly absurd.

Credits

Plugin developed by Timm Johnson

http://sonictimm.com/

Realistic Diesel Engine & Boxcar modeled and textured by Marwan Fathi

https://www.fiverr.com/mohamed_bw