POLYGON JELLY

Train, Rail and Roller Coaster System

Version: 1.0.4
Doc last updated: 9th Jan 2018

Help / Website / Videos

Overview video for this system: https://youtu.be/HtADtIsoB1w
Forums: https://forums.unrealengine.com/showthread.php?139109-Train-and-Rail-System
Marketplace: https://www.unrealengine.com/marketplace/train-and-rail-system

Polygon Jelly channel: https://www.youtube.com/channel/UCpmlbVl8Hcc_4gvkELgLEFg
Polygon Jelly: http://polygonjelly.com

Update History

Ver 1.0.4 - 9th Jan 2018

- Spline mesh roll is now driven by spline points, allowing for more complex and interesting tracks (see section below)
- Terrain tracks have been updated to use a control spline for better control (see section below)
- Track baking added to convert terrain tracks into standard tracks (see section below)
- New roller coaster example map added
- New cable car example map added
- New speed trigger added
- New models, materials and demo carriage with pawn added for new maps
- Added a map with lots of pre-built sections of roller coaster mesh (just swap the mesh for your mesh / track)

Ver 1.0.3 - 4th Dec 2017

- Updated the mesh spacing for the track and added new options (see notes below *'Mesh Spacing Updates'*)

Ver 1.0.2 - 5th Nov 2017

- Updated to UE 4.18

Ver 1.0.1 - 5th May 2017

- Added a option to allow the starting distance on the track to be set manually.
- Added roll options for tracks
    - Added start, middle and end roll for each track piece
    - Added debug options to visualize the up dir and roll options for each spline point

- Added roll examples map

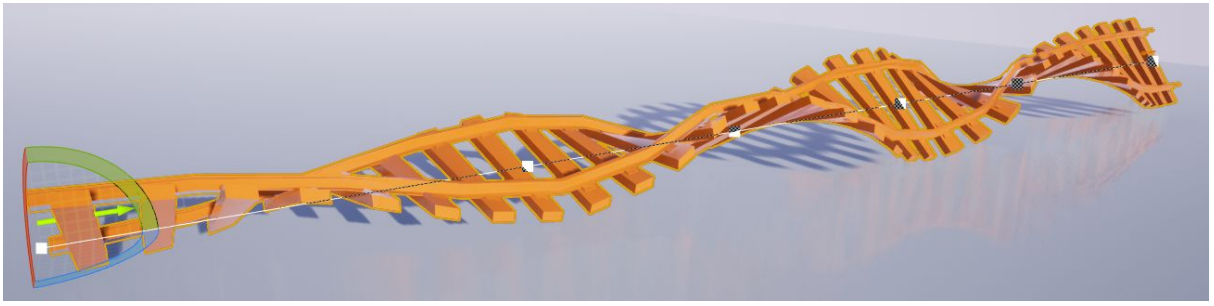
Ver 1.0.0 - 5th April 2017

- Initial release

Mesh Roll Updates (new in Ver 1.0.4)

With this update the old roll options for spline meshes has been removed and the roll is now controlled directly by the spline point.
This allows the user to roll each point individually to create far more interesting and complex tracks.

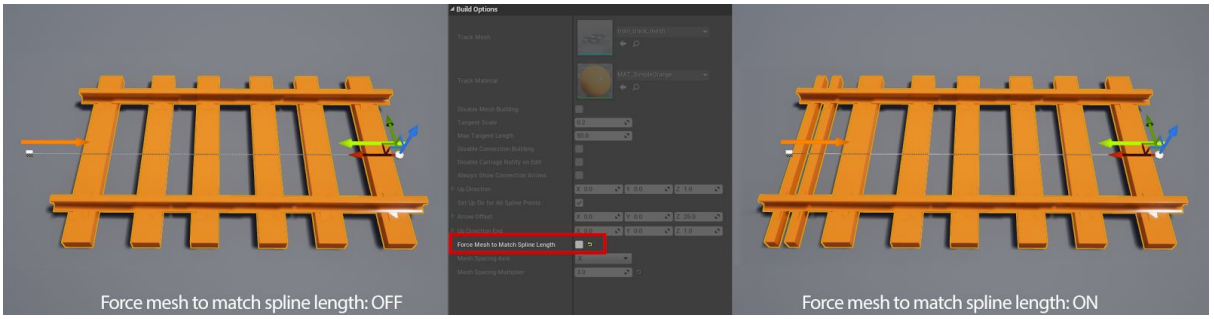
Here's a track with 5 spline points, each with a different extreme roll to demonstrate:

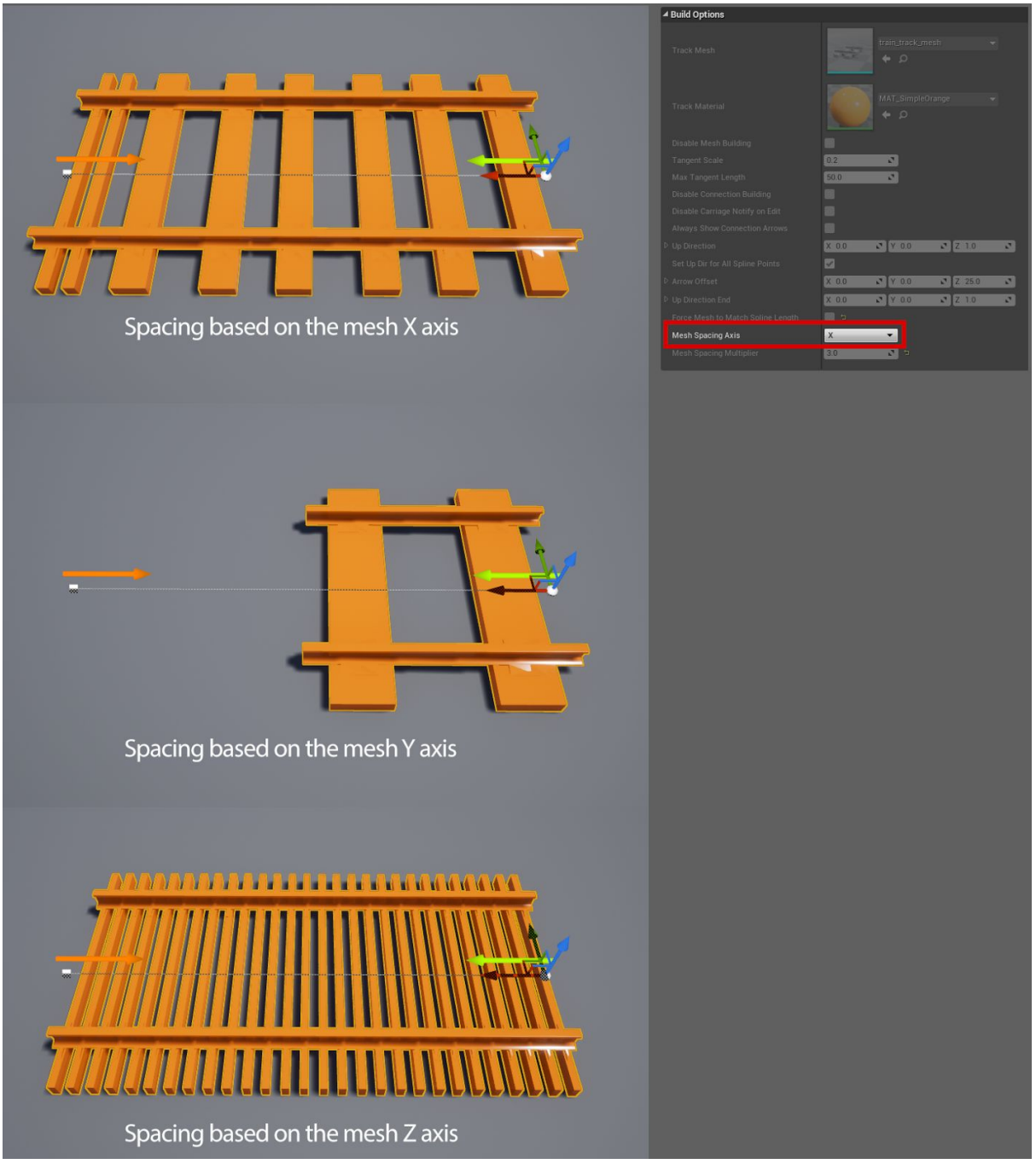


Mesh Spacing Updates (Ver 1.0.3)

Added in 1.0.3 are three options to help you control the mesh spacing and layout better. You can now turn on / off the option to force the track mesh to always match the length of the spline (**ForceMeshToMatchSplineLength**).
You can pick which axis the mesh spacing is based off using (**MeshSpacingAxis**).
Finally you can use a multiplier to control the spacing further should you need it (**MeshSpacingMultiplier**).

All these new options are available in the '*Build Options*' of the track. And you can find detailed descriptions below.





New options:
ForceMeshToMatchSplineLength

True by default, this will make sure that the track mesh is always the same length as the spline. If your track is 50 units in length and the track mesh is 20 then you'll get two normal size sections and one squashed section.

MeshSpacingAxis
Select the axis (X, Y or Z) that you'd like to use when calculating the size of each section of track mesh. E.g. If the mesh is 20 on X, then select x and a piece of mesh will be added every 20 units along the spline.

MeshSpacingMultiplier
Use this to increase the spacing between track mesh sections. 1.0 by default.

General

Terrain Tracks

These are tracks that allow you to quickly build tracks over uneven surfaces without you having to manually place each point.
The terrain tracks have a control spline that should be positioned above the surface, when the track updates it will line trace from each of these points down, where the trace hits is where the track will be placed.

You can copy the tangents from the control spline to the track spline if needed. You can also copy just the first or last, allowing you better control when going from a terrain track to a standard track.

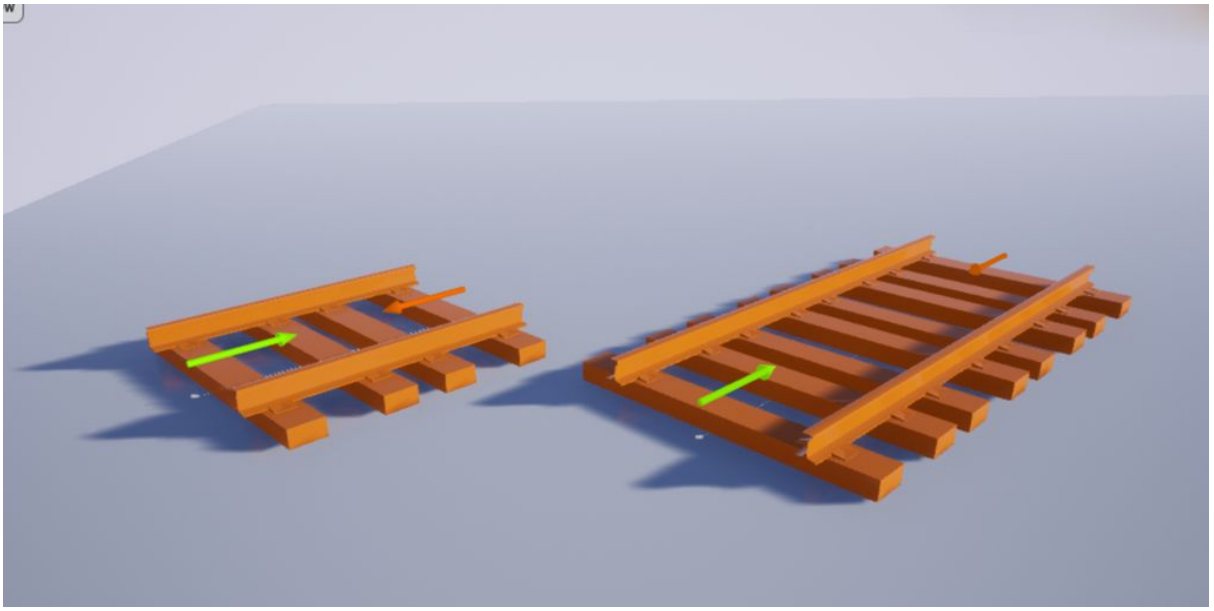
Since terrain tracks use a line trace they are a bit more expensive to create. Once you're happy with the track layout you should bake the terrain track into a standard track. See the section below.

Track Baking

Each track now has an option to select a BP_TrainTrack and to bake from it. Once you select a track the current track spline will have the points cleared and all points from the source track copied into it, at which point the bake track option will be reset.

Or in a shorter version: the track you select will be copied into this track.

Why would you want to use this?
It's designed for copying terrain tracks into regular tracks so they don't have to use line traces every time the track is constructed.



Connection Building

Tracks make connections between each other. The connections are made at the end points of the track, these are the start or end points of the track.
The start and end of a track are called the '**connection points**'.
The start point of a track is marked with a green arrow and the end point with an orange arrow.

To make a connection between one track and another drag the start or end point within distance of the start or end of another track, or the current track (if creating a self looping track).
Once you make the connection the arrows will disappear.
You can adjust the acceptable radius in the track settings.

Currently (UE 4.15) there is no way for a Blueprint actor to inform other actors in the world if it's about to be deleted in the editor. As such, if you delete tracks and don't manually make a new connection into them, they will have stale data. To remove this stale data, just edit the track or use the connection tool.

The **connection tool** is very simple, it rebuilds all connection in the game world. During development if you're moving tracks around a lot, it's good practice to keep it in the level. Once you're done, you can delete it.
It will rebuild connections in the constructor and once on being play.
There is no need to have the connection tool in a shipping level.

3D Models

The models provided were built by us are for example only and as such are pretty simple. Please feel free to use them for your projects, this is just here to let you know that they might need some sprucing up if you intend to use them in production.

Splines and Direction Of Travel

The term spline and track spline are used interchangeably throughout the docks, but they both refer to the same thing; the spline that a track is built on and that a train carriage will travel along.

Splines have a start and end, Splines have a length. As you travel along a spline you are at a distance on that spline.
Spline distance starts at 0, the start of the spline and end at x, where x is the length of the spline.

The tracks use splines to build the track mesh and to allow bogies to travel along them.

A bogie / carriage is considered to be moving forward as it travels in a positive distance along the spline, E.g. Moving from position 0 to position 10, to position 20, …
A bogie is considered to be moving backwards as it travels down the spline, E.g. 20, 10, 0.

The distance on the spline is used to generate a transform (position, rotation and scale).
As the bogie moves along the spline we use this transform to position it, the carriage and do a bunch of other work to render the train correctly.

Inverted Direction Of Travel

If you've been digging around in the code and you're wondering what the 'InvertDirOfTravel' travel is or the references made to it in the Blueprints, then you're in the right place.

As stated above the distance along the spline goes from 0 at the start to X at the end.
As the bogie / carriage moves from one track to another it's not always going to be going from the end of one spline to the start of another.
For example it's possible to have tracks connected from start point to start point, or end point to end point.

When a bogie is traveling on a track spline in a any way other than from start to end it has the 'InvertDirOfTravel' set to true.

We then use this flag in various locations to adjust how the system behaves. The best example is moving the bogie, when false we add to the current position to move the bogie forward, but when it's true, we subtract.

If this still isn't clear, fire us a message and we can produce more detailed docs or create a video example.

Trigger Bogies

As a carriage moves down the track we check for region overlaps with triggers.
To cut down on the number of checks made each tick we only check certain bogies.
We store the bogies that will be used for checks in the **'TriggerBogies'** array in the train carriage.

By default the sensor, front and rear bogie are added to this array. If you add custom bogies or want to change the setup you should update this array as needed.

Collision Bogies

The collision example map gives a detailed breakdown of the various collision options.
By default every carriage adds two collision bogies, one at the front and one at the back of the carriage. These are used to detect impacts and react accordingly, E.g. Derail if we hit something big and hard enough.

You might want to expand this for a custom setup, the collision bogie code can be used as a full example of how to setup additional collision detection.
Also included in the examples is a collision filtering example so you can ignore things like the player.

Parent and Child Carriages

Settings:
By default parent settings are passed down to child carriages. Obviously we don't want to sync every settings, so included are some of the obvious choices for syncing. If you need to add some settings to this sync or remove some, they can all be found in the function: **'UpdateSettingsFromParent'**.

Functions:
The train setup is simple; if a carriage has a parent, it will ask it to make critical decisions such as how to handle an obstacle or how to handle the end of the track.

Distance Checks:

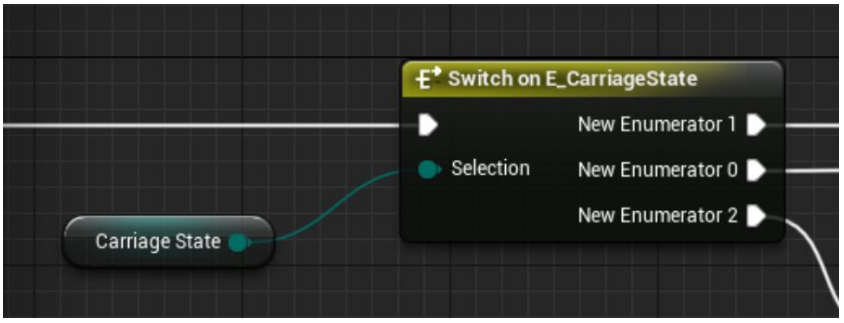
If distance checks are enabled, it' the child that checks the distance to the parent. This allows the parent to now worry about the child.

Speed Calculations:
The child will always get it's speed and velocity from the parent. There is a lerp for how close the child will match,

Setup:
The current setup is to have the child carriages behind the parent carriage. It is of course possible to change this if you need them to be out front. If you do change the default setup you will need to slightly adjust the 'TickCarriageMovement' function, it's marked in green.

Currently we tick the parent first if it's moving forward and the children after it. If going in reverse the children are ticked first and the parent last. This is done so that we always get the first carriage depending on direction to report back to the others about the end of the track or other serious events the whole train should be aware of.

Enum Nodes Displaying 'New Enumerator'



This is a known bug in UE, if this happens, just right click on the node and select 'refresh node' to fix it.

Changing Blueprints

The goal with this system is to give you something you can use for your project or to give you something to build on to get what you need, so changing the Blueprints is encouraged. We have found that if you're making changes to any blueprint that has a lot of instances in a level it's always to do that in a blank level so that the recompile doesn't invalidate the current instances. Of course, this advice doesn't matter if you're making massive changes to the base class.

Accessed None error in Bogie

This can sometimes happen in the editor only if you change the track a lot. Since the train holds a reference to the track it is on, and then uses the spline inside the track this can sometimes get lost of the track is modified enough to cause the editor to reconstruct.
If it happens, just move the train in the editor or change a setting and it will re-attach to the track and the new reference will be used.

Actor Merging

Once you've laid out your tracks and you're happy with how they look you should consider merging the actors.
The tracks use spline mesh components to create the track mesh, so a long length of track can have hundreds of these depending on your mesh size.
To improve performance, you can merge all these individual meshes into one single mesh.

To do this, select the track and then go Window -> Developer Tools -> Merge Actors.
You'll be presented with a dialog with merge options, change the options if you need to and save the merged mesh.
On the track you selected disable mesh building with the option "Disable mesh building" and the track mesh will be turned off for this track. Add a static mesh component to the track and then set the mesh from the merge as that mesh.

Spawning Tracks At Runtime / Streaming Maps

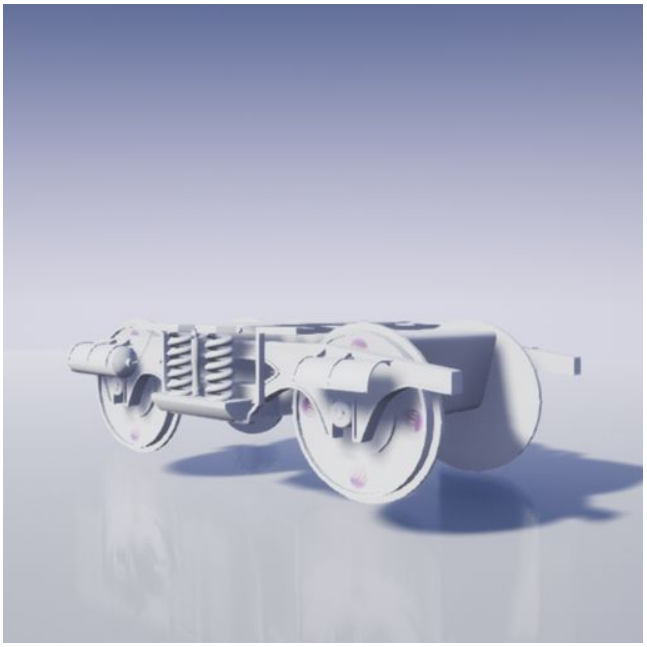
The system was designed around creating the tracks in the editor. That being said it is possible to spawn tracks at run time.

Once you've spawned the track in the level you'll need to call 'BuildTrackConnections' on the spawned tracks to make connections with surrounding tracks.

Remember to take care when spawning / streaming levels at run time, if you remove a trigger that a track needs, be sure to update the trigger and vice versa.

Blueprints

Here's a list of all the blueprints, listing their functions, macros, events and components in order with details on each.



AC_Bogie

Parent: Actor component
Purpose: Represents a train car bogie

Summary: Bogies represent a position on a track. A train carriage uses one or more bogies to travel along a track.

Note: Traveling against the spline

Functions

ChangeTrack
Changes the track the bogie is currently on.
TickMovement
Moves the bogie along the current track spline.
If the bogie is moving from spline start to end, the speed is added to the distance. If the direction of travel is inverted, then it's subtracted. See the note in <TODO: ADD NOT LINK HERE>
GetTransform
Gets the world transform for the current distance on the spline.

Note: If the direction of travel is inverted, we rotate the rotator by 180. We do this so the carriage or anything using the transform don't rotate as it changes tracks. <TODO: Add note here>

GetDistanceToEndOfTrack
Get the distance to the end of the track.

Note: If the bogie is traveling from the start of the spline to the end of the spline, then this should be end distance - current distance. But we must adjust this if the bogie is traveling against the spline. TODO: Add note link here

GetConnectionPoint
Returns the connection point for the current track spline this bogie is on.
UpdateDistanceForEndOfCloseLoop
Updates the bogie's current distance on the spline when it's traveling on a closed loop.

Macros

None

Events

None

Components

SplineComponent
Reference to the spline this bogie is currently traveling on

Variables

SplineDistance
Distance on the current spline
SplineLength
Length of the current spline
SplineIndex
Index of the current spline in the current track's splines
Track
Track this bogie is currently on
DistanceToEndOfTrack
Distance to the end of this track / spline
NeedsAlignment
Unused

DebugName
Debug name used when printing info about this bogie
IgnoreEndOfTrack
If set to true, this bogie will not fire events for the parent carriage when it reaches the end of the track
IsSensorOrHelperBogie
Used by triggers and end of track notifications. If set to true, this bogie will be handled as a helper bogie
InvertDirOfTravel
Used to indicate of this bogie is traveling against the spline. See docs for full details on this.
DistanceFromCarriage
How far this bogie should be from the carriage. This is used for alignment and other distance / placement calculations

BFL_TrainAndRail

Parent: Blueprint Class
Purpose: Function library for the train and rail system

Summary: Holds some shared functionality

Note: None

Functions

GetTrackIgnoreTag
Returns the ignore tag used by components. E.g. Tracks consider any spline in the actor a track spline, splines with this tag will not be used.
GetUsableSplinesFromTrack
Get an array of spline components in the track actor to use as track splines.
GetStartAndEndLocationOfSpline
Return the world location of the first and last point on a spline.
GetDistanceBetweenLocations
Return a float which is the distance between two vectors.
IsValidTrackSplineIndex
Checks the supplied index to see if it's valid.
HasTrackSplines
Returns a bool indicating if this track has any track splines.
GetClosestSplinePoint
Returns the closest spline point to a supplied location.
GetClosestDistanceOnSpline
Returns the closest distance on the spline to the supplied location.
GetClosestDistanceBetweenSplinePoints
Returns the closest distance between two spline points
FindLocationOnTracks
Given a requested distance and a starting track / distance this will find the final location on the tracks that is the requested distance from the starting location. It will also return details on all tracks passed through on the way to the final location.

Note: While this looks like an imposing funcion, it's really simple. You start at a distance on a track spline and you move forwards or backwards on that track until you reach your destination.
As the function walks the tracks it keeps track of all the tracks it had to traverse, this information is passed back with the final location and track.
This function is used in a lot of places, the best example is in triggers; we have our start point on the track, where the trigger is attached and we want to find the distance from this point that the trigger will activate based on the trigger radius.

GetLowestFloat
Returns the lowest and highest of two supplied floats.
GetInternalCompTag
Returns the internal component tag. Used by auto added components, E.g. Arrows added to the start and end of a track spline.

Macros

None

Events

None

Components

None

Variables

None

BP_ConnectionTool

Parent: Blueprint Class
Purpose: Rebuild and update connections for any track in the game world. To be used during development only.

Summary: Tracks make connections to one another. Each track is an actor. Currently the Unreal Engine doesn't have an in editor event for when an actor is deleted. So if you delete a lot of tracks, we need to rebuild the connections.

This tool does that at construction and on EventBeginPlay, so if you're doing a lot of editing it can be handy to keep it in the level.

This is only needed during development, so once you're happy with the track layouts or your done building, you can delete this.

Functions

RebuildAllConnections

Macros

None

Events

None

Components

EditorIconBB
Icon billboard for this blueprint

Variables

DisableLevelWarningMessage
If true the warning message this Blueprint prints when the level begins will not be displayed.

BP_TR_DebugTool

Parent: Blueprint Class
Purpose: Debug functions and helpers stored in a single tool instead of bloating the track or carriage classes.

Summary: Handy functions if you're modifying the train or carriage classes. E.g. Render all bogies in real time by selecting the carriages.
Handy if you want to visualize where the sensor or collision bogies are.

Warning: This is a debug tool, so it doesn't do a lot of error checking for you.

Functions

Conn_RebuildAllConnections
Rebuild all connections for all train tracks in the game world.
Conn_PrintConnectionInfoForAllTracks
Print connection info for all tracks in the game world.
PrintConnectionInfoForTrack
Print connection info for track passed into the function.
Conn_PrintConnInfoForSelectedTrack
Print connection info for selected tracks
PrintConnectionPointInfo
Print connection point info about the connection point passed in
SelCar_PrintAllBogieNames
Print the bogie names in 3D at their location for the selected carriages.
Track_PrintTriggerRegions
This will print out info on each trigger region on the selected tracks and also add a simple box collision as a rough marker for the region. Since it's a simple box, it won't be too accurate on bends.
Track_DeleteAllTriggerRegions
Deletes all regions from the selected tracks.
SelCar_ShowDebugHitch
Render debug sphere at the hitch location.
SelCar_DisplaySensorBogie
Renders the sensor bogie for the selected carriages.
Demo_RefreshInfoBoxes
Refreshes the test on all demo info boxes.

Macros

None

Events

None

Components

EditorIconBB
Billboard editor icon

Variables

RebuildAllConnections
PrintConnInfoForAllTracks
PrintConnInfoForSelectedTrack
SelectedTracks
SelectedCarriages
SelCar_PrintBogieNames
SelCar_ShowDebugHitches
SelCar_ShowSensorBogie
PrintTriggerRegions
DeleteAllTriggerRegions
RefreshDemoInfoBoxes
ClickToRerun

BP_TerrainTrainTrack

Parent: BP_TrainTrack

Purpose: Extension of standard train track that atuo builds the spline and uses line trace to lace those points on uneven ground.

Summary: Try saying the title three times fast.

This should be used for quick building of tracks across landscapes or other uneven surfaces.

Designed to be used for single sections of train track that is places on uneven terrain. The user sets the start and end point of the track and the required number of spline points. The spline is auto created and each point is conformed to the terrain underneath it.

Note: Construction script deletes the default spline, calls the terrain functions to create the new spline and then hands back to the parent class to set everything up with the new spline.

Rotation should be applied to the whole actor since it needs to move in a straight line.

Functions

InitalizeTerrainTrack
Add the new spline component, assigns to the terrain spline variable and then calls the AddSplinePoints function.
AddSplinePoints
Wipes all current spline points, adds in each new spline point. Each spline point that's added gets a line trace from it's location down, if it hits anything with this trace, then that will be the location of this spline point.

Macros

None

Events

None

Components

Auto added spline component, a reference to the added spline is set in the TerrainSpline variable described below.

Variables

StartPoint
Start point for the track spline, this connects into any other start or end location of a train track.

This is a vector exposed as a 3D Gizmo.

EndPoint

End point for the track spline, this connects into any other start or end location of a train track.

This is a vector exposed as a 3D Gizmo.

TerrainSpline

Reference to the spline that the new points will be added to. This is automatically updated and should not be edited manually.

PointCount

Number of points required on this spline.

Note: Min number of points is 3.

TraceOffsetAbove

When adding a spline point, this offset is added to the start point of the line trace.

TraceDistance

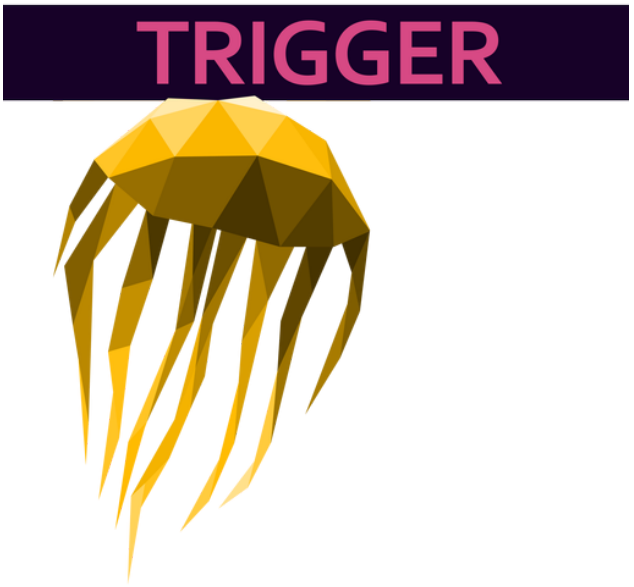
How deep this trace should go, if you're using a large landscape, you might want to make this bigger.

UseTraceOnStartPoint

Set this to true if you want the trace to happen on the start point, otherwise it will match the start location perfectly.

UseTraceOnEndPoint

Set this to true if you want the trace to happen on the end point, otherwise it will match the end location perfectly.



BP_TrackTrigger

Parent: Blueprint Class
Purpose: Used to generate events in the game when a train carriage enters the trigger region.

Summary: Triggers attach to tracks and have a radius, this radius is used from the attach point to generate trigger regions. These trigger regions are sections of the track, that when a carriage enters them cause an event to be triggered.

These events can be overridden to setup custom triggers. All carriage and bogie information is passed into them.
The trigger region overlaps are the final part of the carriage update to happen, this allows you to add offsets or make other changes to the carriage relative to the location on the track.

Note: Base class is in the system folder, but example child classes can be found in the Triggers folder, and on the triggers example map.

Functions

InitalizeTrackTrigger
Run by construction script to setup the track when first created
AttachToTrackAndBuildRegions
Snaps trigger to closest point on the track and builds the trigger regions on that track
DisableTrigger
Disable the trigger from firing
TickEnableTrigger

Updates the countdown to re-enable the trigger
EnableAndResetTrigger
Makes the trigger active and will fire the PostTriggerResetEvent
TriggerFireCheck
Used to evaluate if the trigger can fire or not

Macros

ResetEnableTimer
Resets the enable timer.

Events

InternalTriggerEvent
Override this event in your trigger class and add custom code here to happen when the trigger is fired.
PostTriggerResetEvent
Override this event in your trigger if you want to perform any logic when the trigger is reset.

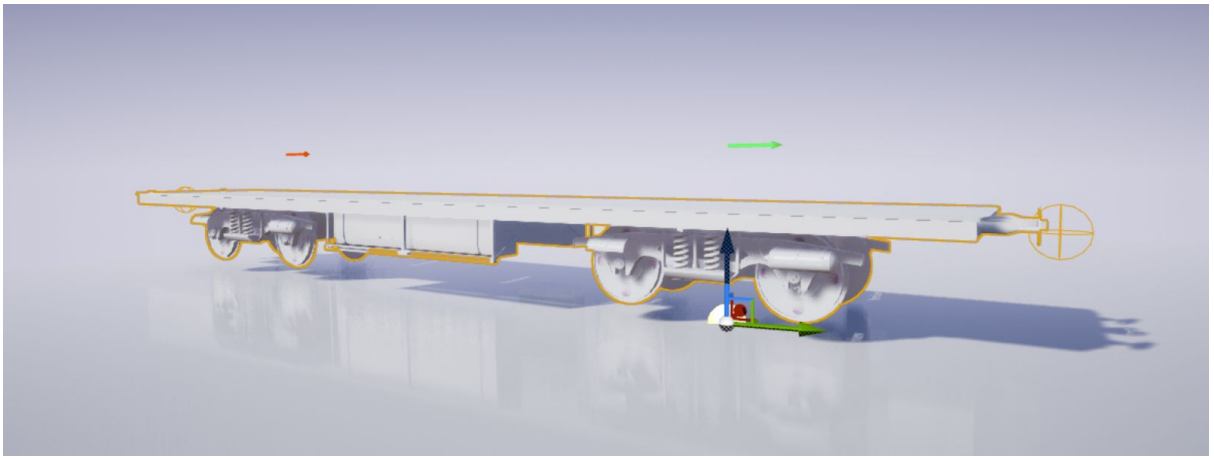
Components

RadiusSphere
Visual aid for the location of the trigger

Variables

Track
Track this trigger is attached to.
TrackSplineIndex
Track spline index this trigger should attach to.
TriggerRadius
Distance on track from the attach point in both directions for this trigger.
DistanceSampleCount
Sample count to use when attaching to the track
AutoSet_DistanceOnSpline
Distance on the attached spline, this is automatically set, do not edit manually.
TracksWithTriggers
Array of tracks this trigger is attached to.
ApplyToAllTrackSplines
Bool indicating if this trigger should apply to all track splines.
Active
Bool indicating if this trigger is active or not.
IgnoreHelperBogies

If true, the trigger will not fire if the triggering bogie is a helper or sensor bogie.
TriggerOnce
Trigger only once.
UseEnableTimer
If true, the once triggered the trigger will become active again once the number of seconds set in ResetDelay has passed.
ResetDelay
Time in seconds between resets after the trigger has been fired. UseEnableTimer must be set to true for this to be used.
TimeRemainingForReset
Internal countdown used for reset. This is automatically set, do not edit manually.
TriggerOncePerCarriage
If set to true, the trigger keeps track of which carriages have fired the trigger. When the trigger is reset, this will be reset.
TriggeredCarriages
Internal array used to track which carriages have fired the trigger. This is set automatically, do not edit manually.



BP_TrainCarriage

Parent: Blueprint Class
Purpose: Apples

Summary: Apples

Note:
Engine vs. Carriage
Recompile time / effect

Functions

InitalizeTrainCarriage
AttachToTrack
UpdateCarriageMeshLocations
TickCarriageMovement
TickBogieMovement
BogieChangeTracks
SetupBogies
HandleEndOfTrack
ToggleCarriageDirection
StopCarriage
InitalizeMesh
UpdateBogieDistances
TickCollision
SensorPulse
PutSensorBogieOutFront
HandleActiveObstacle
SetupCollision
DerailCarriage
ApplyOrReleaseBrakes
SetCarriageState
SetCarriageDirection
LeanCarriage
PerformRollCheck
UpdateStartingDistanceRelativeToParent
RefreshPositionOnTrack
CheckDistanceToParent
DisconnectFromParent
IsActorOnIgnoreList
ManuallySetupCarriage
DisconnectChild
ConnectToParent
CalculateSpeed
IsDirectionOfTravelInverted
NotifyAboutTrackUpdate
GetWheelSpeed
SetupAnimations
UpdateAnimations
SetParentCarriage
SetChildCarriage
UpdateSettingsFromParent
WarpCarriageToStartingPoint
ResetCarriage
AddActorToCollisionIgnoreList

Macros

GetBogies
GetSensorBogieDistance
GetFrontBogieBasedOnDirOfTravel
SetCarriageHasActiveObstacle
AddCollisionBogie
IsStopped
IsMoving
HasParent
HasChild
CreateRandomDerailValues
HasTrack
GetHitchToBogieXDistance
AddLocationOffsetToTransform
BrakingClamps
BrakingDeccel
InvertCarriageAcceleration
SetTransformScale
CalculateDerailForce

Events

PostMoveUpdate

Components

RearHitch
FrontHitch
RearBogieArrow
FrontBogieArrow
SensorBogie
RearBogie
RearBogieMesh
FrontBogieMesh
CarriageBodyMesh
FrontBogie

Variables

Visual Options

BodyMesh
Static mesh to use for the train body.
BogieMesh

Skeletal mesh to use for the bogie.

BogieDistance

Distance rear bogie should be from the front bogie.

BodyOffset

Offset that will be applied to the body mesh when positioning.

BogieOffset

Offset that will be applied to the bogie mesh when positioning.

UseBogieRollOnBody

If true the roll from the bogie rotation will be applied to the body mesh.

WheelAnimSpeedScale

Current speed of the carriage will be multiplied by this value and used as the playback speed for the bogie wheel animations.

BogieAnimClass

Animation class to be used on the bogie animations.

AutoFilled_ActorScale

Used for correctly scaling the mesh. Auto generated, so not edit manually.

Track Settings

AttachedTrack

Track this carriage is attached to.

AttachedTrackSplineIndex

Index of the spline on the track this carriage is attached to.

StartingDistanceOnSpline

Starting distance on the spline. This is auto set, do not edit manually unless you have 'ManuallySetDistanceOnTrack' set to True

ManuallySetDistanceOnTrack

If true then you must manually enter a value for 'StartingDistanceOnTrack'. If False, the train carriage will pick the closest point on the spline.

EndOfTrackOption

What this carriage should do when it hits the end of the track.

AttachDistanceSampleCount

Number of distance samples to check when attaching this carriage to the track.

GracefulStopForEOT

Flag used internally to indicate of the carriage came to a graceful stop.

SensorTriggeredEOT

Flag used internally to indicate if a sensor or helper bogie triggered the end of track response.

StartInverted

If true, the carriage will start inverted on the attached track.

SnapToTrack

If true, the carriage will snap to the selected track.

Engine Settings

Velocity
Current velocity of this carriage. Set automatically, do not manually edit.
Friction
Friction used when calculating the final speed of this carriage.
Gravity
Gravity used when calculating the final speed of this carriage.
MaxSpeed
Max speed this carriage can reach.
Mass
Mass of this carriage used when calculating the final speed of this carriage.
MinSpeed
Min speed this carriage should be at.
CurrentSpeed
Internal tracking of the current speed. Set automatically, do not manually edit.
Acceleration
Acceleration to use when calculating speed. This is only applied if the carriage type is Engine.
Deceleration
Deceleration to use when calculating speed. This is only used when braking.
StartingDirectionOfTravel
Initial travel direction of this carriage.
UseConstantSpeeds
If true then the engine will always be at max speed.
IsBraking
Internal flag to indicate if this carriage is currently braking. Set automatically, do not manually edit.
InternalAcceleration
Acceleration value used internally. Based on the public variable, this value is set as positive if moving forward or negative if moving backwards. Set automatically, do not manually edit.

Carriage Settings

CarriageState
Current state of the carriage.
DisableRearBogie
If true the rear bogie will be removed
CarriageType
Type of carriage, currently Engine and Carriage. Engine has acceleration, does not and needs to be hooked to an engine to move.

Collision Settings

ImpactCollisionEnabled
If true the auto added collision bogies will knock items back and respond to impacts.
DisableSensorBogie
If true the sensor bogie will be disabled. This must be enabled for obstacles to be detected.
SensorBogieDistance
Distance the sensor bogie should be from the carriage.
SensorOffset
Sensor bogie offset from the train track
SensorBoxSize
Size of the box to use for box traces for objects at the sensor location on the tracks.
MinObstacleSize
Anything below this size will be ignored by the sensor. The bounds radius of the obstacle is checked against this value.
ObstacleResponse
How the carriage should respond to an obstacle on the tracks.
HasActiveObstacle
Internal flag used to indicate if this carriage has an obstacle.
TimeBetweenObstaclePulses
Time between obstacle pulses, this is only used when the carriage has an active obstacle.
TimeToNextPulse
Internal countdown to next pulse. This is set automatically, do not edit manually.
FrontColBogieOffset
Offset applied to the front bogie position.
RearColBogieOffset
Offset applied to the rear bogie position.
CollisionBogies
Bogies used for collision detection. If you add any custom collision bogies to the carriage make sure to add them to this array so they can be used for collision checks.
CollisionQueryTypes
Query types to use when doing any collision or sensor related traces.
PlowImpactStrength
Strength of the plow when knocking items back and away from the train carriage.
PlowImpactThreshold
Amount of force that an impact must have in order to slow down this carriage.
NeverDerail
If true this carriage will never derail from an impact.
DerailForce
Amount of force applied to the carriage when it derails. This value is multiplied by the current speed to create the final force.
TimedIngoreList
Internal array of items that are to be ignored during collision checks.
ChildDerailDampen

When the carriage derails it will also derail its children. The derail force that's passed down to the child is multiplied by this value to create the final derail force for the child.

Train Carriages

ParentCarriage
Parent carriage for this carriage.
ChildCarriage
Child carriage for this carriage. This is auto updated if the other carriage sets a parent.
DistanceFromParent
Distance this carriage should start at from the parent carriage.
MaxDistanceFromParent
Max distance this carriage can be from the parent before disconnect.
ChildPlacedBehindCarriage
If true the child carriage will be placed behind the parent.
UseDistanceLimitChecks
If true distance between parent and child will be checked against the MaxDistanceFromParent and if it is exceeded, the child will disconnect from the parent.
TimeBetweenDistanceChecks
Time between distance checks.
TimeToNextDistanceCheck
Internal countdown until next distance check. Automatically set, do not edit manually.
ConnectedParentHitch
Internal reference to the parent hitch.
ConnectedChildHitch
Internal reference to the child hitch.
ParentSpeedLerp
Lerp value used when matching the parent speed.
DisconnectedFromParent
Internal flag to indicate if this carriage has been disconnected from the parent.
InheritParentSettings
If true, this carriage will grab settings from the parent carriage.

Lean Effect

UseLeanEffect
If true the lean effect will be used by this carriage.
LeanMaxYawDelta
Max delta between yaw values that must be reached for the lean effect to kick in.
CurrentLeanAmount
Internal value used to track the current lean amount. Auto set, do not edit manually.
TargetLeanAmount
Internal value used to track our current lean target. Auto set, do not edit manually.
LeanIgnoreDelta
If the lean delta is above this value it will be ignored.

LeanLerpAmount
Lerp value used when moving between the current lean and the target lean.
ApplyLeanToBogies
If true the lean effect will be applied to the bogies as well. If false, then it will just be applied to the body mesh.
LeanReturnToRestMultiplier
Used to return the lean to the original location when the lean is at 0.
LeanMinSpeedPrecentage
Min precentage of the max speed that the carriage must be at before the lean effect will kick in.
LeanAmount
Lean value that will be used when leaning to mesh to the side.

Roll Settings

UseRollCheck
If true, the roll check will be used.
TimeBetweenRollChecks
Time to wait between roll checks.
TimeToNextRollCheck
Internal countdown to next roll check. This auto updated, do not edit manually.
LastRollingSample
Last rolling sample used.
RollDeltaThresholdRPY
Roll, Pitch and Yaw thresholds to use during the roll check.
RollDerailForce
Derail force to use if a roll threshold is broken. This value is multipled by the current speed to create the final value.
RollThresholdEval
Which type of threshold evaluation to use.

Carriage Internal

FrontBogieTransformLastTick
We store the transform from the last tick because it’s used in a few places.
TriggerBogies
Array of bogies that are used when checking for trigger overlaps. Bogies not on this list won’t be checked. We do this to cut down on the number of checks we do each tick. Add your custom boies here if you need to.
FrontBogieTransform
Transform from the current tick for the front bogie. We store this because it’s used in so many places and it’s faster than each of those places requesting it from the spline each time.
Bogies

Array of bogies in this carriage. Stored in this array because it's faster than getting all components of a class each time the bogies are needed. It's also easier than constantly checking if one is disabled or not, we just process this list.

Debug Options

PrintDebugStrings
If true any functions that need to print a message at run time will be allowed to do so. E.g. Sensor bogie will print what's currently blocking it from moving forward on the tracks.

BP_TrainTrack

Parent: Blueprint Class
Purpose: Represents a train track

Summary: Apples

Functions

InitalizeTrainTrack
InitalizeTrackSplines
BuildTrackConnections
SetupArrows
AddConnectionArrow
GetSplineByIndex

GetSplineIndex
PickConnectionToUse
RegisterTrackTrigger
ValidateTriggers
AddTriggerRegion
ClearTriggerRegions
CheckForRegionOverlap
GetTrackSplineCount
NotifyCarriagesAboutTrackUpdate
BuildTrackMesh
RebuildTrackConnections

Macros

WithinConnectionDistance

Events

CarriageEnteredTrack
CarriageExitedTrack

Components

DefaultTrackSpline

Variables

Build Options

TrackMesh
Mesh to use for the tracks
TrackMaterial
Material to use for the tracks
DisableMeshBuilding
Turn on or off mesh building. Use this option to speed things up if you're placing big tracks.
TangentScale
If the length of the spline tangent are greater than MaxTangentLengththen it will be multiplied by this value to reduce it.
MaxTangentLength
When adding spline mesh the length of the tangent will be compared to this value, if it's greater than this value, the tangent will be scaled back.
DisableConnectionBuilding
Turn connection building on or off, turn this off to speed things up when building large tracks.
DisableCarriageNotifyOnEdit
Turn on or off sending notifications to carriages in the world about the track being edited.
AlwaysShowConnectionArrows

If turned on, arrows that the track connection points will always be visible.

UpDirection

Set the up direction for the track spline. Default is 0, 0, 1

SetUpDirForAllSplinePoints

If true then the UpDirection will be manually set at each spline point.

ArrowOffset

Offset applied to the connection arrows.

ForceMeshToMatchSplineLength

True by default, this will make sure that the track mesh is always the same length as the spline. If your track is 50 units in length and the track mesh is 20 then you'll get two normal size sections and one squashed section.

MeshSpacingAxis

Select the axis (X, Y or Z) that you'd like to use when calculating the size of each section of track mesh. E.g. If the mesh is 20 on X, then select x and a piece of mesh will be added every 20 units along the spline.

MeshSpacingMultiplier

Use this to increase the spacing between track mesh sections. 1.0 by default.

Connection Options

MaxConnectionDistance

If the distance between two track connection points is greater than this value, then the connection won’t be made.

TracksToIgnore

Any track on this list will be ignored when building track connections.

Internal

TrackSplines

Internal array of track splines. This is created and updated automatically, do not edit manually.

Triggers

Triggers

Array of references to triggers that have a region on this track. This is updated automatically, do not edit manually.

DisableAllTriggers

If true, then any trigger on this track will be ignored.

TriggerRegions

Array of trigger regions on this track, this updated automatically, do not edit manually.

Active Connection Options

End_UseRandomConnection
Should the end connection point use a random connection
End_UseActiveConnection
Should the end connection point use the active connection
End_ActiveConnection
Track the end point should use as the active connection. End_UseActiveConnection must be set to true for this to be used.
Start_UseRandomConnection
Should the start connection point use a random connection
Start_UseActiveConnection
Should the start connection point use the active connection
Start_ActiveConnection
Track the start point should use as the active connection. Start_UseActiveConnection must be set to true for this to be used.

Roll Options

StartRoll
Visual roll that will be applied to the mesh at the start of the track. See docs example map and docs for more roll info.
MiddleRoll
Visual roll that will be applied to the mesh at the middle of the track. See docs example map and docs for more roll info.
EndRoll
Visual roll that will be applied to the mesh at the end of the track. See docs example map and docs for more roll info.
MiddleRollSoften
Amount the roll at the middle of the track is reduced, this is used to prevent sharp bends in the track. See docs example map and docs for more roll info.

Debug Options

ShowUpDir
Draws a debug arrow indicating the up direction for the track, this will be a single arrow if the up direction is being set for all points, otherwise an arrow will be added to every spline point.
ShowUpDirForAllPoints
Forces an arrow to be rendered at every spline point showing the up direction
ShowRollInfo
Shows a text representation of the roll at the start, middle and end of the track. See docs example map and docs for more roll info.
RollInfoOffset
Offset applied to the text render.

RollInfoSize
Size of the text that is rendered.
RollInfoStartAndEndOffset
Offset that is applied to the start and end roll text render only, so overlaps with other tracks are prevented.