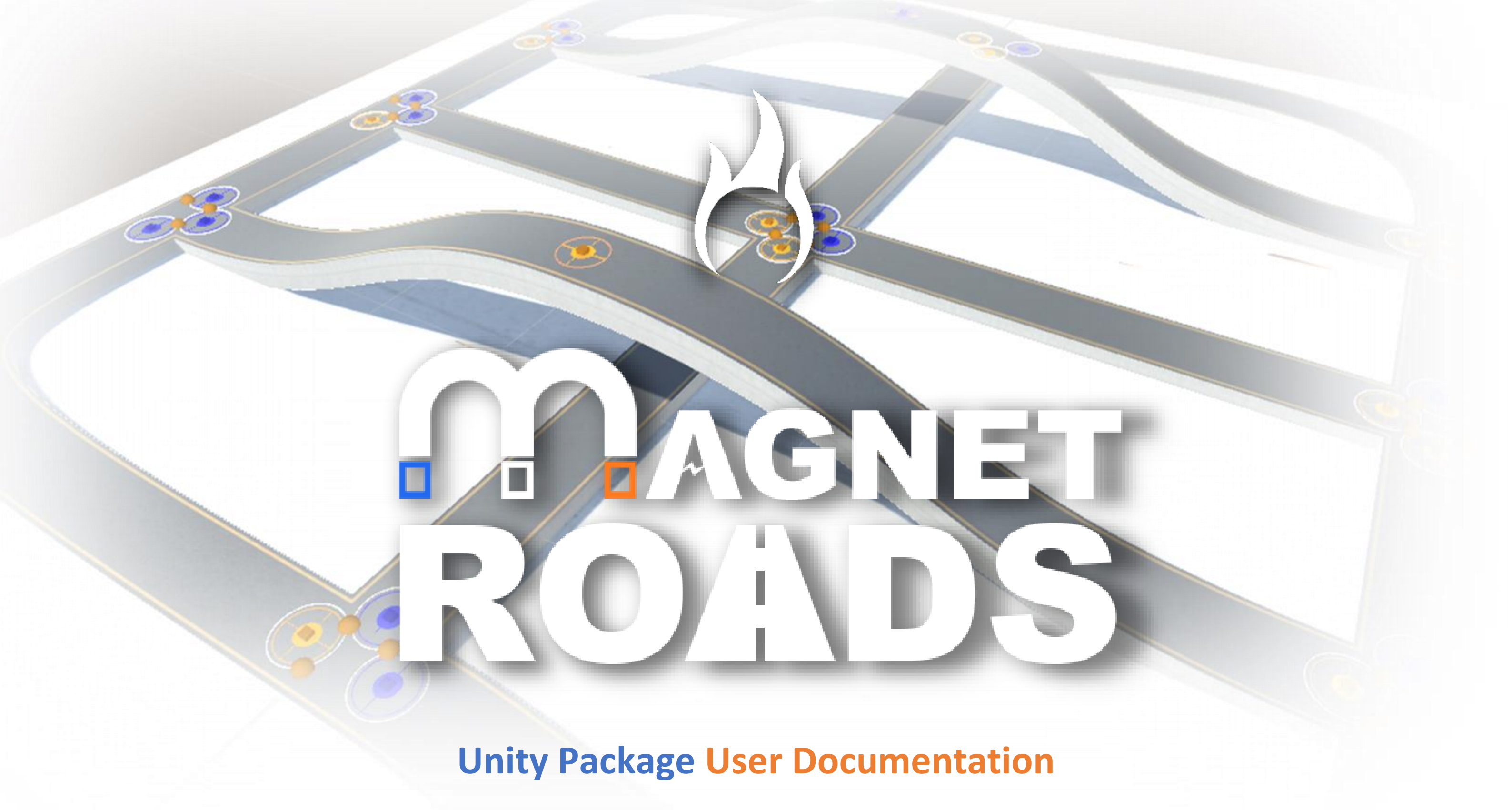


MAGNET ROADS

Unity Package User Documentation

Written by **Jonathan Langley** & **Matthew Oldfield**
contact@tbinteractive.co.uk
www.tbinteractive.co.uk

Contents

Introducing Magnet Roads 1

Magnet Roads 1

- Adding Magnet Roads to Your Scene 1
- Editing Magnet Roads 1
- Scripting With Magnet Roads 2
- Additional Road Objects 3
  - Roadside & Central Fences 4
  - Central Reservation 4
  - Central Objects 4

Intersections 5

- Adding Intersections to Your Scene 5
- Editing Intersections 5
- Getting Usable Information from Intersections 5

Editing Roads & Intersections at Runtime 6

Changelog 7

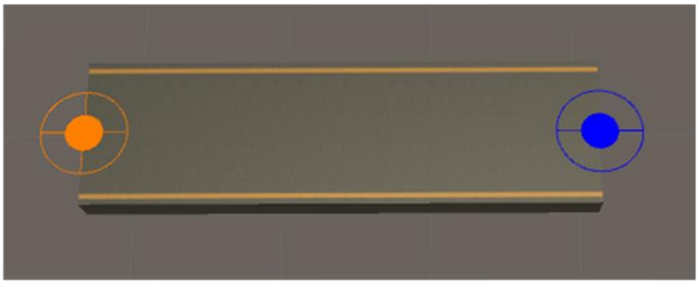
- v2.0.0 7

Magnet Roads v3.0.0

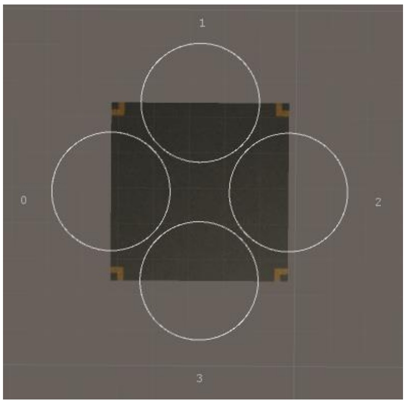
User Documentation

Introducing Magnet Roads

Magnet roads is a simple to use alternative to some of the more complex road creation packages currently available on the Unity Store. Magnet roads allows you to quickly and efficiently create intricately connected road networks or racetracks with a simple to use polarised snapping system.



In short, each road has a '**Positive**' and a '**Negative**' end. These ends can be attached to the any polarity of any other magnet road in the scene (*right* for an example of the polar ends of the magnet roads). In addition to the magnetised road ends, there are also 'Bipolar' intersection points. These unique points only exist on the intersection road pieces.



Magnet Roads

Adding Magnet Roads to Your Scene

Once you've added the Magnet Roads package to your project, a **'Magnet Roads'** toolbar menu will be added to the **'Tools'** dropdown. From here you can spawn new instances of both intersections and magnet roads. Note: these new instances will always spawn at the root of the scene (0.0, 0.0, 0.0). Once selected, the road will be represented by a curve with handles. The two handles at either end will manipulate the source points of the road, the two on the inside handle the curvature of the road. Once you are happy with the position and curvature of the road, simply press the orange **'Generate Road Mesh'** button in the road's inspector window. Note: **To snap**-select an end, drag it into a magnet end and **press 'Generate Road Mesh'**. You may edit and re-generate your magnet roads at any time after initial generation. For additional information on the tools available to edit existing roads, see the **'Editing Magnet Roads'** section.

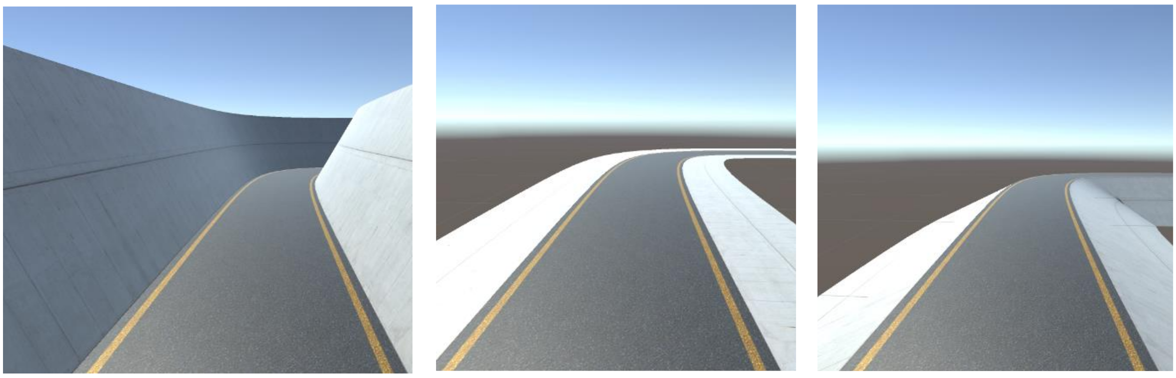
Editing Magnet Roads

You will notice several editable fields in the inspector window when you are manipulating a road's spline. These fields are named and function as such:

FIELD NAME	VALUE TYPE & FUNCTION
`surfaceMaterial`	**Material** – The Material to apply to the road's surface
`sideMaterial`	**Material** – The Material to apply to the road's sides
`sidewalkMaterial`	**Material** – The Material to apply to the sidewalks
`roadWidth`	**Floating Point** – The width of the road to be generated

sidewalkWidth	**Floating Point** – The width of the sidewalk
sidewalkHeight	**Floating Point –** The height of the sidewalk from the road
sideDepth	**Floating Point** – The distance between the road surface and the bottom/top of the road sides
slopeWidth	**Floating Point –** The distance from the edge of the road to the bottom of the sloped edge
stepsPerCurve	**Integer** – The number of points along the curve from which to extrapolate mesh vertex data (higher number = higher poly road)
showRoadOutline	**Boolean** – Toggles whether the outline of the road should be displayed in the editor before generation
roadsideMargin	**Floating Point** – Buffer space at the edge of the road before lanes
totalCarLanes	**Integer –** Total number of navigable lanes on this road
showCarRoutes	**Boolean** – Toggles whether the left and right road lane routes are drawn onto the road
snapRoadToTerrain	**Boolean** – Toggles terrain snapping on this road
terrain	**Terrain** – The terrain source for terrain snapping
distanceFromTerrain	**Floating Point** – Distance from the terrain to snap the road at
editAtRuntime	**Boolean –** Set this road to be editable at runtime
showChildObjects	**Boolean** – Show road's generated child objects in hierarchy

Using these editable values you can achieve numerous road effects. Some examples…



E.g. ***High sided highway style road****, road with sidewalk,* ***downward sloped sides***

Scripting with Magnet Roads

This section will outline the some of the various public methods the user can invoke in their own scripts. *Note: this section will only cover public methods which offer some benefit to the developer, other functionality should be considered only useful to the magnet roads themselves.*

METHOD NAME & PARAMS	RETURN TYPE & FUNCTION
AddCurve (bool atPositive)	**Void –** Extend the spline of the current road at the positive or negative end and add more curve handles.
RemoveCurve (bool atPositive)	**Void –** Remove a spline curve at the positive or negative end of the road; this wont work if there is only one curve left
AttachIntersection (bool atPositive, bool threeLane)	**Void –** Attach a new intersection (either three or four lane) to either the positive or negative end of the road

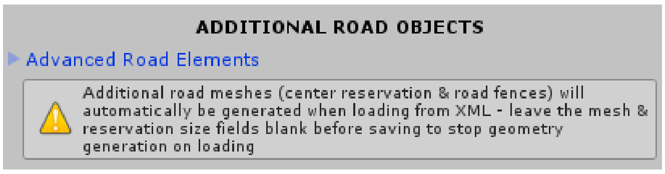
`AttachMagnetRoad (bool atPositive)`	**Void –** Attach a new road to either the positive or negative end of this road
`GetLaneWaypoints (int laneNo)`	**Vector3[]** – Returns a vector array of points along a lane of the selected road based on the total car lanes specified
`GetCentreWaypoints()`	**Vector3[] –** Returns a vector array which runs down the centre of the road from positive to negative
`GetPositiveConnection()`	**GameObject** – Returns the GameObject instance of the road or intersection attached at the positive
`GetPositiveConnection_ MagnetRoad()`	**MagnetRoad** – Returns the MagnetRoad instance of the connection at the positive – returns null if Intersection
`GetPositiveConnection_ Intersection()`	**Intersection** – Returns the Intersection instance of the connection at the positive – returns null if MagnetRoad
`GetNegativeConnection()`	**GameObject** – Returns the GameObject instance of the road or intersection attached at the negative
`GetNegativeConnection_ MagnetRoad()`	**MagnetRoad** – Returns the MagnetRoad instance of the connection at the negative – returns null if Intersection
`GetNegativeConnection_ Intersection()`	**Intersection** – Returns the Intersection instance of the connection at the negative – returns null if MagnetRoad
`SetRoadControlPointPosition (int handleIndex, Vector3 newPosition, bool shouldUpdateMesh)`	**Void -** Set the position of the selected handle index, and if chosen updates the mesh.
`SaveRoadToXML (string path)`	**Void –** Saves the selected road to an XML file at the specified path, will go to project root if no path is provided
`SaveRoadsToXML (string path)`	**Void** – Saves all roads in the scene to an XML file at the specified path, will go to project root if no path is provided
`LoadRoadsFromFile (string path)`	**Void –** Load saved road(s) from an XML file at the defined path, creating and positioning them in the scene

The saving and loading functionality can also be accessed from the **'Tools'** dropdown. In addition to these methods, there is also an accessor for each of the road's snap points. These are: **`SnapNodeRight`** and **`SnapNodeLeft`**; both of which return the **GameObject** of their respective snap point.

Additional Road Objects

Magnet Road's now supports the creation of additional road geometry. To access these features you must expand the **'Advanced Road Elements'** item inside of the **'Additional Road Objects'** subsection within the inspector.

Filling-out these fields will mean that this data is saved to the XML file – once this data is saved the additional geometry will also be generated when using the loading functionality. To stop this happening, make sure that the mesh references for the fences/objects are set to **'None'** and that the reservation size vector is set to zero.



ADDITIONAL ROAD OBJECTS

▸ Advanced Road Elements

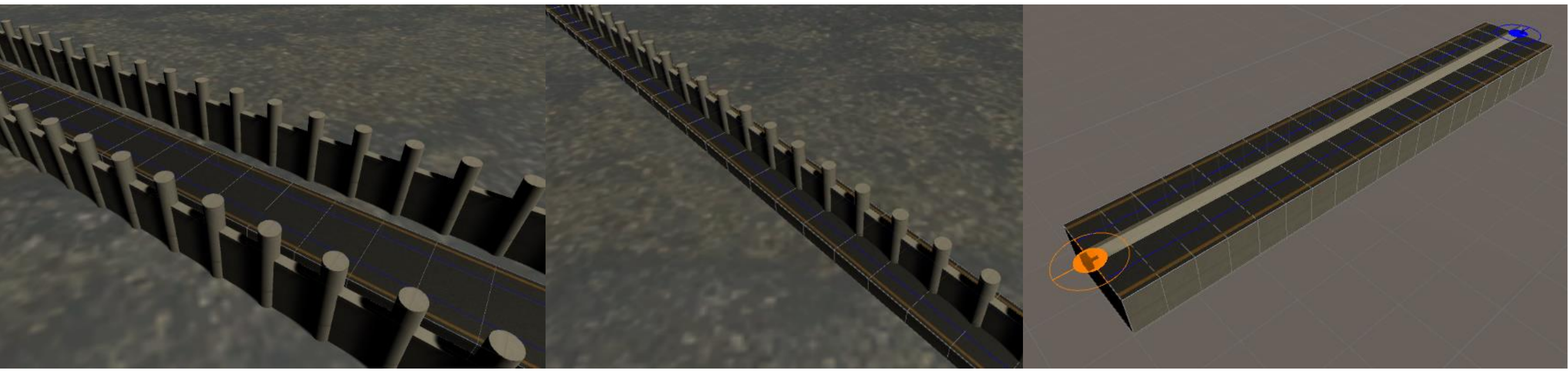
⚠ Additional road meshes (center reservation & road fences) will automatically be generated when loading from XML - leave the mesh & reservation size fields blank before saving to stop geometry generation on loading

Roadside & Central Fences

Firstly, let's look at generating roadside & central fences. There is some common data we need to provide here, so let's explain that first. **Note:** *fence panel meshes are expected to be aligned lengthways along the Z axis.*

FIELD NAME	TYPE & FUNCTION
panelMesh	**Mesh –** Determines the mesh to use for the fence's panel
panelScale	**Vector2 –** Define the X & Y scales for the panel piece, Z is automatically set
panelRotation	**Vector3 –** Define a custom rotation offset for the fence panel pieces
panelMaterial	**Material –** The material to apply to the fence panel
postMesh	**Mesh –** Determines the mesh to use for the fence's post
postScale	**Vector3 –** Define the scales for the post piece
postRotation	**Vector3 –** Define a custom rotation offset for the fence post pieces
postMaterial	**Material –** The material to apply to the fence post

Exemplar fences (roadside, central) and a central reservation



Central Reservation

Central reservations are a special kind of additional geometry that creates one long divider down the centre of the road. They require the following data to be generated:

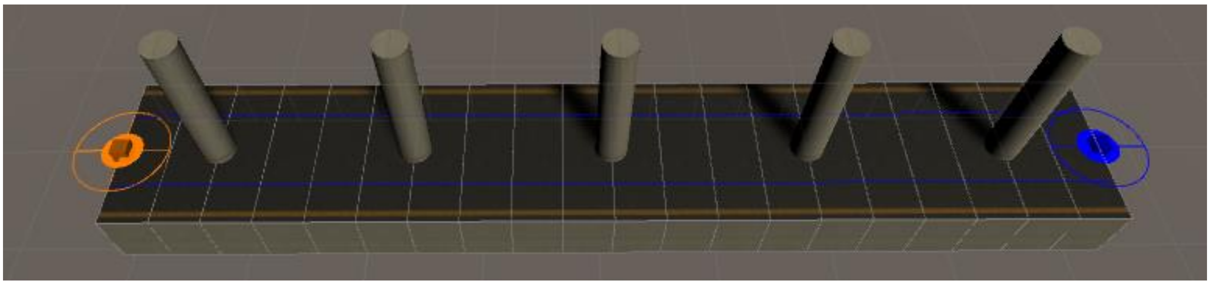
FIELD NAME	TYPE & FUNCTION
reservationSize	**Vector2 –** Define the height and width of the reservation
reservationSlope	**Floating Point –** Define the sides' slope distance from the reservation
topMaterial	**Material –** The material to apply to the top surface of the reservation
sideMaterial	**Material –** The material to apply to the side surface of the reservation

Central Objects

You can also populate the centre of your roads at intervals with custom objects. To create these objects you must provide the following data:

FIELD NAME	TYPE & FUNCTION
objectMesh	**Mesh –** Define object mesh to be spawning
objectScale	**Vector3 –** Define the object to spawn's scale
objectRotation	**Vector3 –** Define the object to spawn's rotation
objectMaterial	**Material –** The material to apply to the spawned objects
totalObjectsToSpawn	**Integer** – The total number of objects to spawn

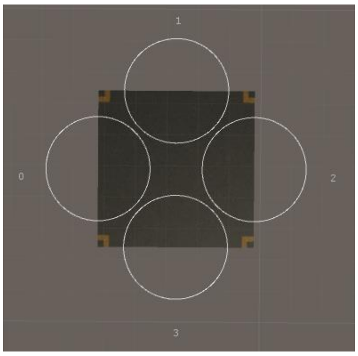
Exemplar generated central objects…



Intersections (Legacy)

Adding Intersections to Your Scene

Like the roads before, to spawn a new intersection you simply click the ‘**Tools’** toolbar menu, and go to **Magnet Roads -> New Intersection**. Here you will have two options; a three and a four lane intersection. In terms of functionality, these intersections function the same as one another; the major difference being the number of points at which Magnet Roads can connect.



Editing Intersections

Again, like the roads, intersections have some values in the inspector which you can manipulate – only, much more basic. These are as follows:

FIELD NAME	VALUE TYPE - FUNCTION
`surfaceMaterial`	**Material –** The Material to apply to the intersection’s surface
`sideMaterial`	**Material** – The Material to apply to the intersection’s sides
`roadWidth`	**Floating Point** – The width intersection to be generated (preferably identical to that of any connecting roads)
`sideDepth`	**Floating Point –** The size of the mesh generated on the sides of the intersection.
`slopeWidth`	**Floating Point –** The distance between the road’s edge and the sides

Getting Usable Information from Intersections

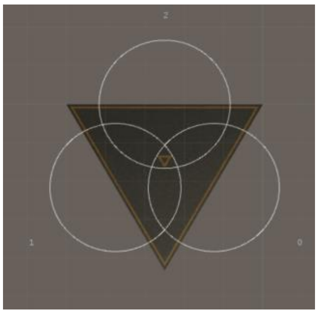
Unlike the roads, intersections do not have a great many methods to call to extrapolate information. Instead, roads simply hold references to the **SnapPoints** it possesses. These are acquired through the accessor: **`SnapNode`**.

Pressing the orange ‘**Regenerate Intersection Mesh**’ will re-create any deleted start or snap points as well as update the intersection with any new **`roadWidth`** or **`sideDepth`** values.

Dynamic Intersections

Adding Intersections to Your Scene

Like the roads before, to spawn a new intersection you simply click the ‘**Tools’** toolbar menu, and go to **Magnet Roads -> New Dynamic**



Intersection. This will create a new dynamic intersection with 3 possible connection point at the root of your scene.

Editing Intersections

Like the intersection before, dynamic intersections are very similar in the values that can be manipulated with a few extra values, for the extra features.

FIELD NAME	VALUE TYPE - FUNCTION
surfaceMaterial	**Material –** The Material to apply to the intersection's surface
sideMaterial	**Material** – The Material to apply to the intersection's sides
roadWidth	**Floating Point** – The width intersection to be generated (preferably identical to that of any connecting roads)
sideDepth	**Floating Point –** The size of the mesh generated on the sides of the intersection.
slopeWidth	**Floating Point –** The distance between the road's edge and the sides
connectionAmount	**Integer -** The amount of connections that are created (Min Val = 3)
isEditableAtRuntime	**Boolean -** Sets if any changes made at runtime should be made instantly
drawOutlines	**Boolean -** Used to disable/enable a outline of current values in scene view

Getting Usable Information from Dynamic Intersections

Like the intersections, Dynamic intersections do not have a great many methods to call to extrapolate information. Instead, roads simply hold references to the **SnapPoints** it possesses. These are acquired through the accessor: **`SnapNode`**.

There are some interfaces that allow developers to interact and generate the dynamic intersections.

METHOD NAME & PARAMS	RETURN TYPE & FUNCTION
`SetUp()`	**Void –** This method removes the current road and generates a new one with the current field values.
`AttachMagnetRoad (int entranceNo)`	**Void -** This method creates a new magnet road to at the entrance provided. This requires the road to be generated.

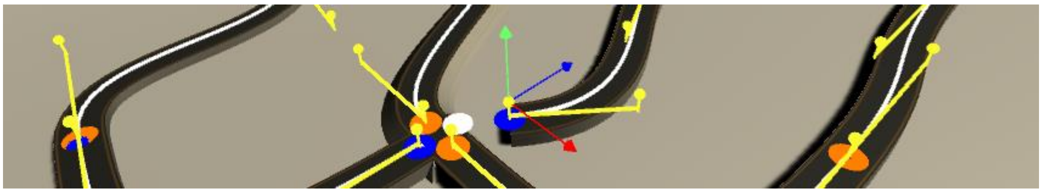
Editing Roads & Intersections at Runtime

As of version 1.1.0, real-time road editing has been implemented into Magnet Roads as a feature in of itself. This functionality provides users with access to some brand new methods.

METHOD NAME	VALUE TYPE - FUNCTION
CreateNewSplineRoad()	This method creates a new Magnet Road at the centre of the world (0.0, 0.0, 0.0).
CreateNewThreeLane()	This method creates a new three way intersection at the root of the world.
CreateNewFourLane()	This method creates a new four way intersection at the root of the world.
CreateNewDynamicIntersection()	This method creates a new dynamic intersection at the root of the world.
EnableRuntimeEditing()	This allows the user to manipulate the road at runtime.
DisableRuntimeEditing()	This prevents the user from being able to manipulate the road at runtime.

In addition to these methods, there is also an accessor used to return the roads editable state (Boolean) called **`IsEditableAtRuntime`.**

How runtime editing looks in-game…



Changelog

v3.0.0

+ Added Dynamic intersections with changeable connection amount
+ Added Decal system for Magnet roads
+ Added Snap to collider feature to Magnet roads
+ Can now Snap points of the same type to each other
+ Added SetRoadControlPointPosition method to Magnet road
+ Added two way intersections
+ Added sidewalks to intersections
~ Updated to 2018.2.0f2
~ Updated handle colours to make them more distinguishable
~ Updated TBUnityLib to v1.6.0 - addition of plane generation for dynamic intersections
~ Updated Gizmos handles - Removed unused functionality
~ Fixed bug where clear mesh would sometimes throw a null reference
~ Fixed bug where gizmos where not clickable

v2.0.0

+ Terrain snapping added for magnet roads
+ Changed roads to allow for custom number of lanes
+ Can now spawn intersections directly at road ends
+ Can now add roadside fences with custom post & panel meshes
+ Can now add central fences with custom post & panel meshes
+ Can now add central reservation geometry to roads, i.e. street lights
+ Can now generate sidewalks along roads
+ Added 'Advanced Road Elements' section to inspector UI for new generation tools
+ Updated to Unity 2017
+ Shiny new logo
+ Roads now store connections internally for easier access when programming
+ Road connection serialization using Guid identifiers for persistence
+ Updated TBUnityLib to v1.5.1 - only minor fixes/reformatting
~ Fixed issues with intersection rotation when loading from xml
~ Fixed collision bug on roads with side's higher than the road itself
~ Improvements to editor inspectors
~ Edited existing code files to remove useless comments
~ Readability/major optimisation pass on all existing code files
- Removed old example scenes
- Removed road follower testing from roads
- Removed StartPoints from Intersection and removed code file

Credits

Torchbearer Interactive Ltd
Jonathan Langley
Edward Andrew
Peter Waugh
Matthew Oldfield

External Contributors
Diego Alpini