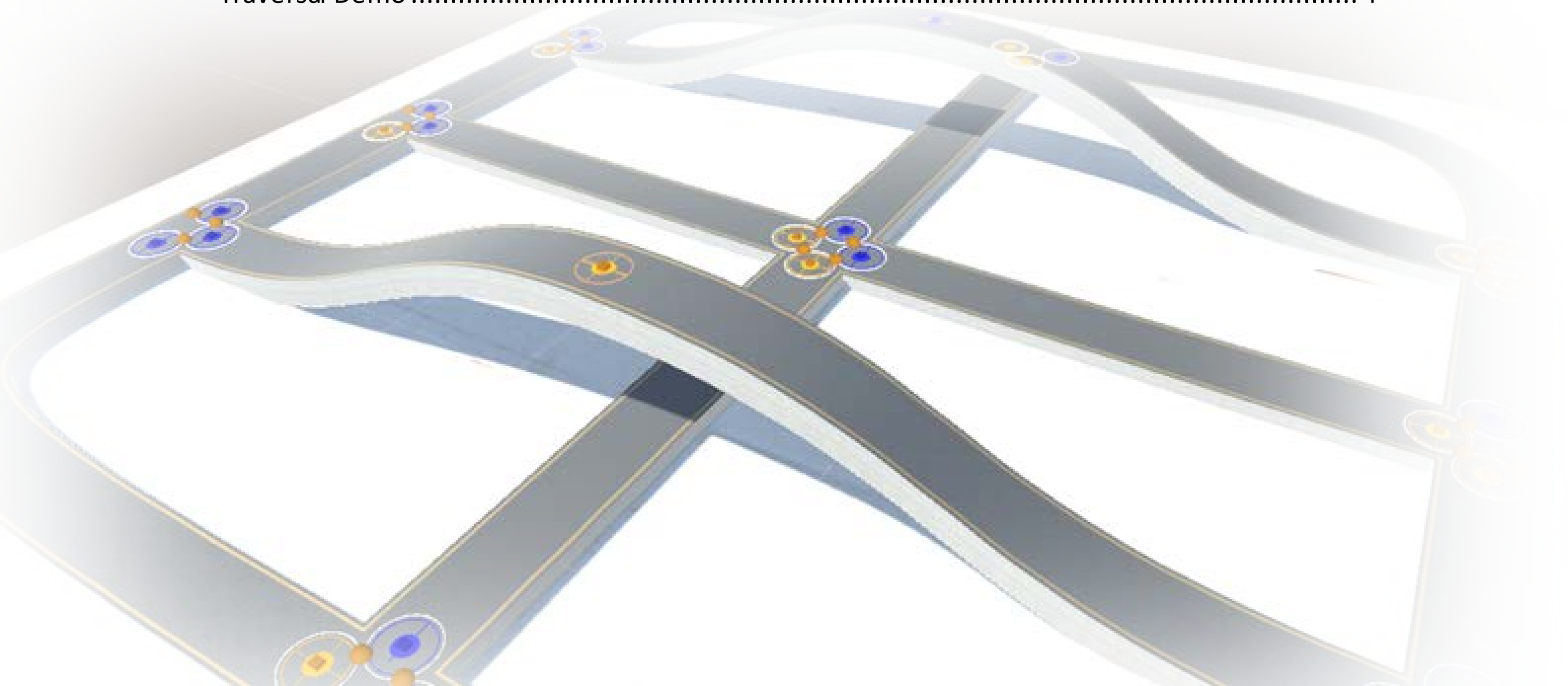
MAGNET ROADS

Unity Package User Documentation

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

Contents



Magnet Roads

User Documentation

Introducing Magnet Roads & Intersections

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 opposite polarity of any other magnet road in the scene (see fig. 1 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. Bipolar snapping points will accept any end of a magnet road regardless of the point's polarity (See fig. 2); bipolar snap points appear as white circles.

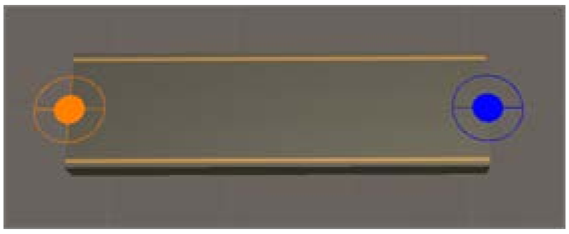


Fig. 1 – Exemplar magnet road w/ polar ends

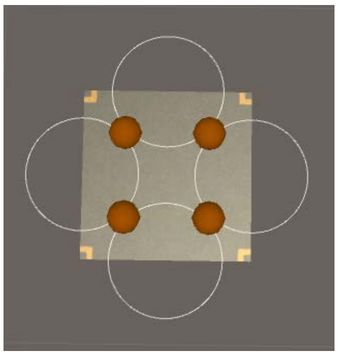


Fig. 2 – Intersection w/ bipolar points

Magnet Roads

Adding Magnet Roads to Your Scene

Once you've added the Magnet Roads package to your project, a '**Torchbearer Interactive**' toolbar menu will be added to your editor window. From this dropdown 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).* Intersections will generate instantly, whereas roads will only appear once selected in the editor. 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***

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.

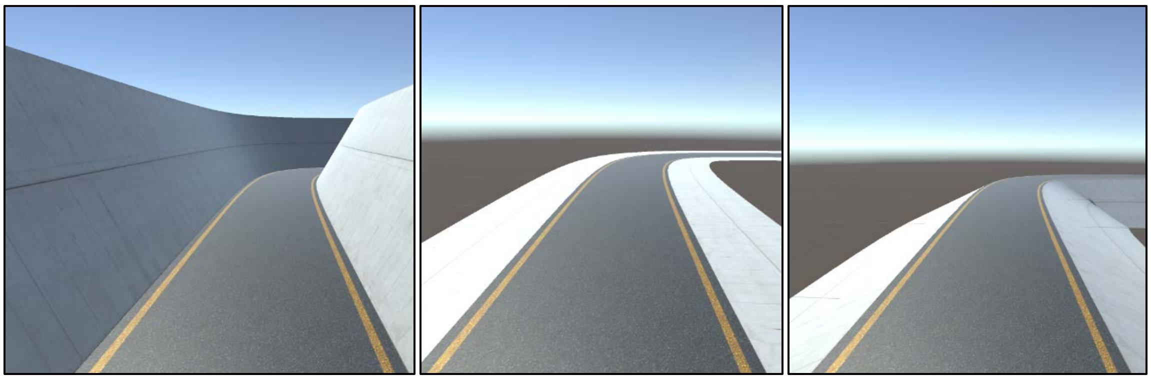
0113 314 1332
contact@tbinteractive.co.uk
www.tbinteractive.co.uk

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
`roadWidth`	**Floating Point** – The width of the road to be generated
`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 or not the outline of the road should be displayed in the editor before generation
`showCarRoutes`	**Boolean** – Toggles whether or not the left and right road lane routes are drawn onto the road
`showRoadSides`	**Boolean** – Toggles whether the outline of the road sides are drawn in editor before generation
`sideDepth`	**Boolean** – The distance between the road surface and the bottom/top of the road sides

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***

In addition to the road's editable values, there are also some test buttons embedded into the inspector window. These extrapolate vertex data from the spline to produce a **Vector3** array; using this array the test method then sends a '**Road Follower**' to track this route in-editor – even allowing the user to select their own unique test vehicle GameObject (see fig. 3). For more information on the methods available to the user see the section called '**Getting Useable Information from Roads**'.

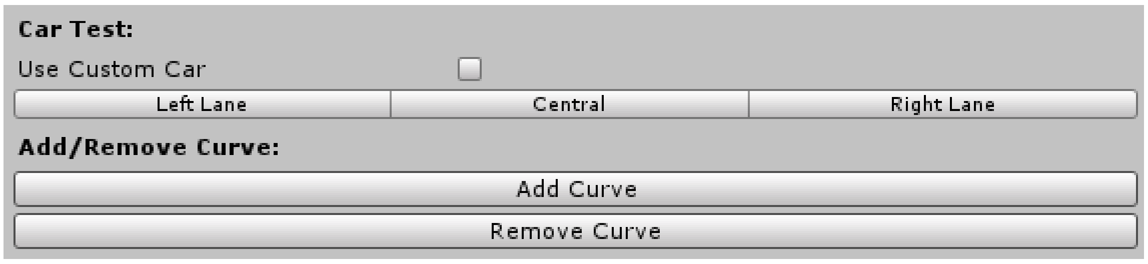


Fig. 3 – Road inspector's testing buttons, indicating the different lanes available to test (+add/remove curve buttons)

Getting Usable Information from Roads

This section will outline the various public methods the user can invoke in their own scripts to pull some of the pre-generated data out of the roads. *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	RETURN TYPE – FUNCTION
`GetMiddleCarPath()`	**Vector3[]** – Returns a vector array of points along the road's curve based on the number of **stepsPerCurve** moving from the positive to the negative end of the road.
`GetLeftCarPath()`	**Vector3[] –** Similar to the previous method, however this one offsets the vector array to the left side of the road based on the **roadWidth**. Again, works from positive to negative.
`GetRightCarPath()`	**Vector3[]** – Essentially the inverse of the previous method: offset to the right, moves from negative to positive.
`GetClosestSnapPointFromVector()`	**SnapPoint** – This returns the closest snap point on the road to a user defined world space vector.

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 **Transform** of their respective snap point.

Intersections

Adding Intersections to Your Scene

Like the roads before, to spawn a new intersection you simply click the '**Torchbearer Interactive**' 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 exactly the same as one another; the major difference being the number of points at which Magnet Roads can connect.

Intersection Start Points

A unique feature of the intersections is that they also generate a set of objects called **StartPoints.** These points indicate entrances to certain lanes on the road and are of potential use to developers looking to create traffic system for their roads (*see the example scene: **TraversalDemo** to look at StartPoints in-use, also see Fig. 4).*

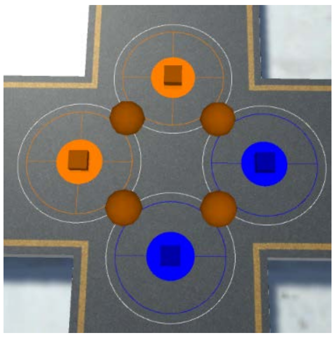


Fig. 4 – Start Points visible as spheres

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.

Getting Usable Information from Intersections

Unlike the roads, intersections do not have a great many methods to call in order to extrapolate information. Instead, roads simply hold references to the **SnapPoints** and **StartPoints** it possesses. These are acquired through the accessors: `SnapNodes` and `StartNodes` respectively.

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.

Included Examples

Track Demo

The track demo contains a small circuit created with Magnet Roads, download a suitable vehicle from the Asset Store and give it a try…

Traversal Demo

This demo relies on two scripts, separate from the Magnet Road tool. These are: **TravelNode.cs** and **RouteManager.cs.** The TravelNode script simply stores information about the roads connecting intersections and the relevant start points to those routes; the RouteManager script makes use of all these TravelNodes to divine viable routes between the intersections.