*Backseat Driver* Roadway Creation Tools

By John Conover

Developer, *Backseat Driver*

1. Summary

This is the documentation for the road design system in *Backseat Driver*. This system utilizes the component-based object creation in Unity in order to create a network of roads. Using this road system simplifies not only road creation in a level, but also for scripting; by maintaining a strict hierarchy in the editor, it becomes extremely intuitive for the programmers to utilize the different roads, road segments and intersections for navigation.

1. Disclaimer:
   1. This system is **not** intended for creating road art. It should only be used to define roads *functionally*, not artistically.
   2. Since the road segments are rectangular, the pathfinding does not give exact road lengths, but rather approximations.
2. Basic Components
   1. Prefabs
      1. **Road Segment**: This is the building block for any **Road**. Each **Road Segment** prefab contains a **NavNode** script attached to it, which allows the GameObject to be used as a vertex in a pathfinding function. Each **Road Segment** has a BoxCollider that is used as a trigger volume. These segments can be linked together to approximate the curvature of the
      2. **Intersection**: A GameObject that is a link between multiple **Roads**. An **Intersection** functions much like a **Road Segment** in the pathfinding function, however it cannot be a child of a **Road**, as they are independent of any single road. Each **Intersection** has both **NavNode** and **Intersection** scripts attached
      3. **LineLink**: A GameObject with a LineRenderer attached do illustrate links between neighboring **Road Segments** and **Intersections**. It is intended to be placed in the **NodeGraph** script.
   2. Scripts:
      1. **NodeGraph**: This script is to be attached the top-level GameObject in the roadway network. This script stores the information of the roadways, with all nodes and connections. The **NodeGraph**’s direct children should be **Intersections** and **Roads**.
      2. **Road**: Defines a road as a system of **Road Segments.** Each road can be given a human-readable name, as well as other information which will be shared between all the **Road Segments** who are children of that road. The **Road Segments** of a single road need not be consecutive (i.e. you can have an **Intersection** that cuts through the middle of a road, with the same road on both sides). See Figure 2.

Figure 1: Recommended hierarchy for a road network. “RoadGraph” has a NodeGraph script attached, each of its direct children have Road scripts attached, and they each have a number of RoadSegments as children.

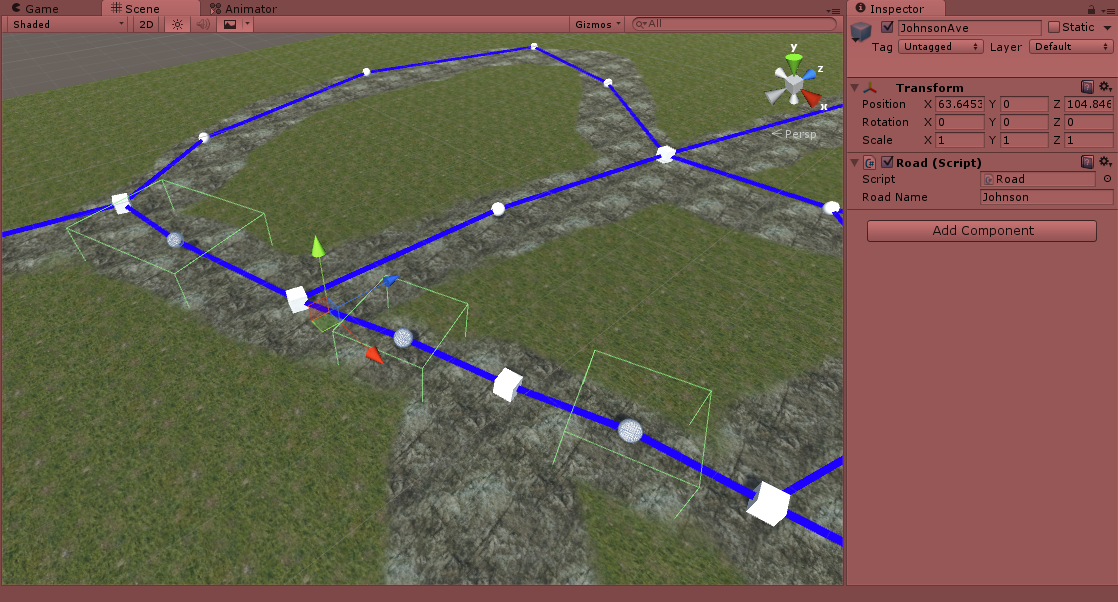
* + 1. **NavNode**: A script attached to every GameObject that can be used as a component of the pathfinding functions. This script is already attached to all **Road Segments** and **Intersections.** Anything possessing this script can be labeled as a **Destination**, which is the endpoint for the pathfinding functions.

Figure 2: “Johnson Ave” is defined by the three RoadSegments whose BoxCollider triggers are visible. Since the Navigator’s “currentRoad” is only redefined when entering a different road, the intersections in the middle are, for all intents and purposes, part of “Johnson Ave”, yet are not its children.

* + 1. **Intersection**: A script already attached to every **Intersection** prefab. Anything with an **Intersection** script also requires a **NavNode** script attached as well.
    2. **Navigator**: A script to be attached to any agent utilizing the pathfinding functions of the roadway system. This script can be referenced by any other script on the GameObject to grab navigation-related information, such as current **Road**, current **NavNode** and the path that is being followed (stored as a stack of **NavNodes**).

1. Rules of Thumb for Creating Roadway Networks
   1. **In order for two Road Segments to be connected, their colliders must overlap.**
   2. **Any connection between two or more roads must have a shared Intersection.**
      1. If two roads overlap without a shared **Intersection**, they will not be registered as connected. Therefore, any two adjacent **Road Segments** from different **Roads** will not connect.
   3. When you wish to modify the trigger volume of a **Road Segment** or an **Intersection**, it is suggested that you resize its BoxCollider instead of the Transform.
      1. If you need to rotate a **Road Segment**, its Transform may be rotated.
2. Navigation
   1. The roadway system is constructed in order to allow a **Navigator** to traverse the network. The **Navigator** is the player-controlled GameObject and the script is responsible for providing messages dependent on how on-course the **Navigator** is.
   2. Upon initialization of the road network, the NavNode.OnRoadsConnected() event is called to initialize path generation. This is a static event, which can be subscribed to by any script and triggered upon formation of a path (e.g. anything that requires a completed road network to function properly).