MicroVerse - Roads

Documentation

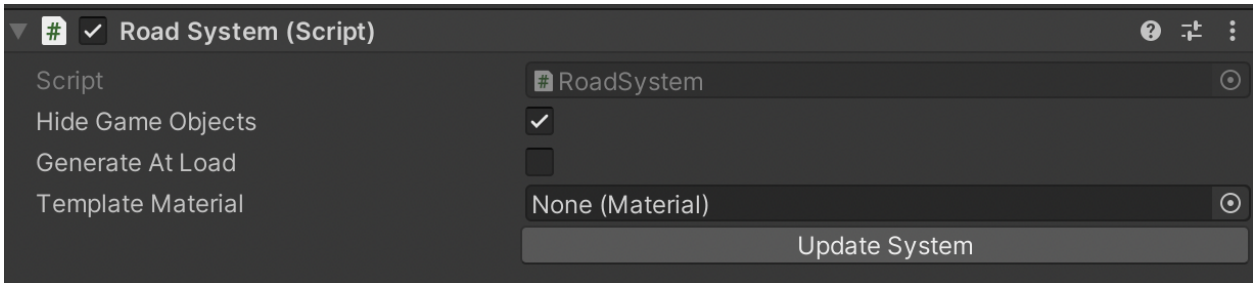
Setup

The Roads module requires Unity's Spline package 2.3 or greater to work. You can install this or upgrade a previous installation by opening the package manager, pressing the + button in the top right corner and selecting "Add Package by Name". Type "com.unity.splines" into the text entry area and press return. Check that version 2.3 or greater is installed.

The MicroVerse and MicroVerse-Splines package also need to be installed.

Once everything is correctly installed, open the MicroVerse content browser from Windows/MicroVerse/ContentBrowser and dock it somewhere useful. You can switch to the road tab and drag an intersection into the scene.

All pieces need to be placed under a GameObject with the RoadSystem component added to it. MicroVerse will automatically create this when you drag an intersection from the content browser into the scene. Note that you can have multiple road systems in a scene if you want to make different kinds of roads from separate collections.



The Road System script controls options and updates for all roads below it. You can choose to hide or show the game objects the road system creates. When Generate At Load is true, the geometry will be created when the scene is loaded. When it's false, the geometry created for roads will be saved into the scene file. Note that this can become a large amount of data, so the tradeoff is between larger scene sizes or slower startup times. For instance, if you are loading your world as the player moves around the scene, it might be better to load larger scene files in the background than to do all the generation at load time, as physics generation can be particularly slow.

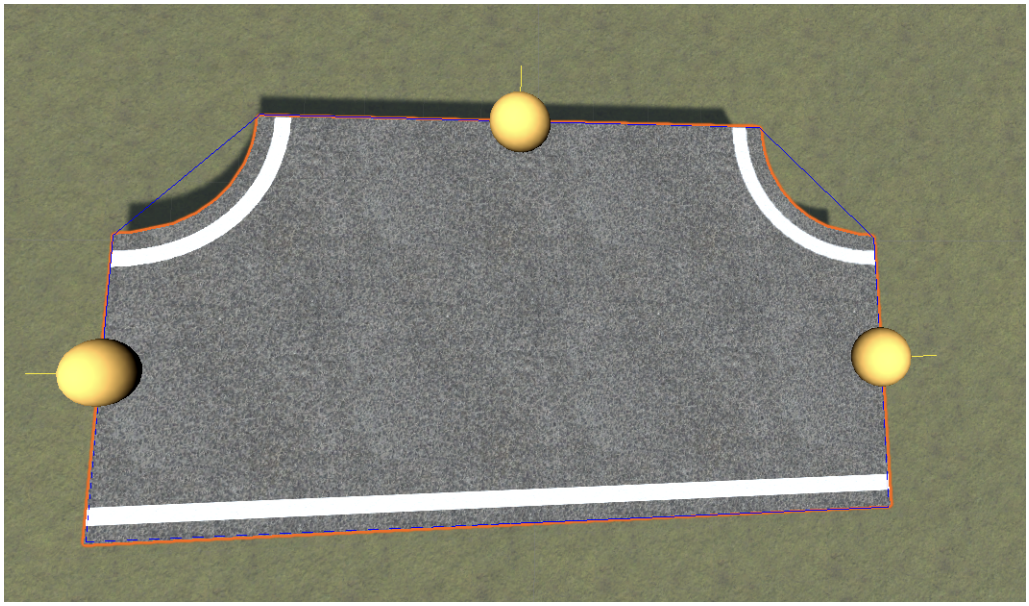
MicroVerse will automatically add a Spline Path component to the road system object. This can be used to control how the road interacts with the terrain. Consult the MicroVerse Spline Path documentation for more.

There is a template material option as well. The roads included with the package use a custom shader to draw high quality lines and wear on surfaces, and with one exception, all of the roads want to have the same material properties. When the material template is set, all of the road pieces in that system will get the properties used in the Template Material set on them. This allows you to adjust the properties of all the materials via one template material instead of having to update them all individually. Some example templates are available in Content/Roads/MaterialTemplates.

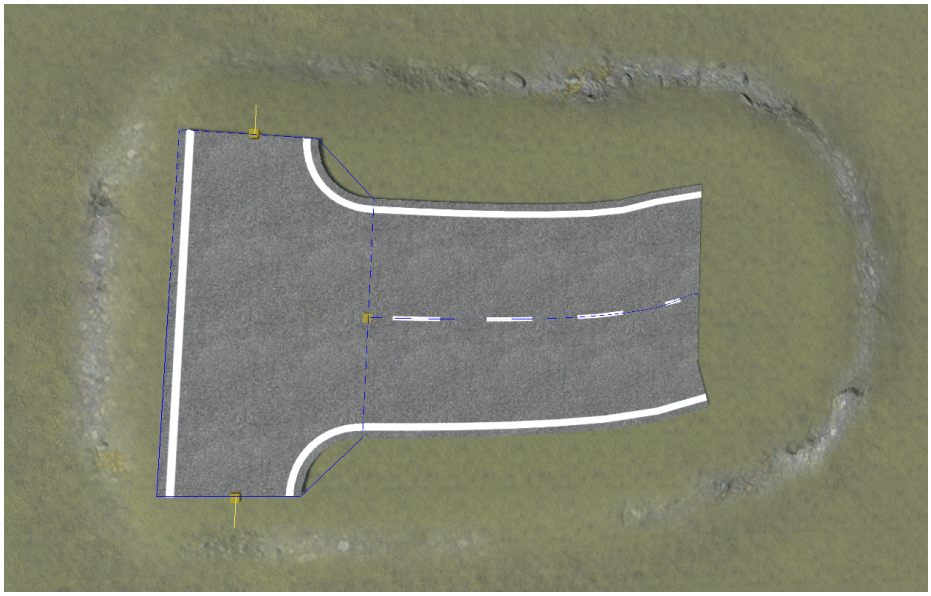
Finally, if you need to manually force a refresh of the entire road system, you can use the Update System button. Under normal operations you won't need to do this, but if you change some of the source data, such as the models, you can use this to force everything to respawn.

Intersections

Intersections allow you to easily create and join roads together.

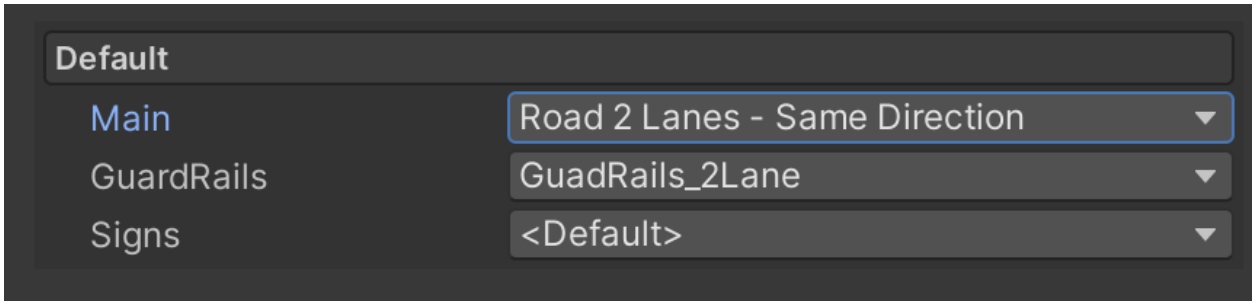


Once an intersection is placed under a road system and selected, you'll see several connection points on the intersection. Double clicking one of these will create a spline based road section.



For more information about Unity's Spline system, consult Unity's spline package documentation. Once you have created the spline, you can move and extend its control points to create a road. If you place another intersection piece under the road system, you can snap the spline's last control point to a connection on that intersection piece to create more complex road systems.

Options

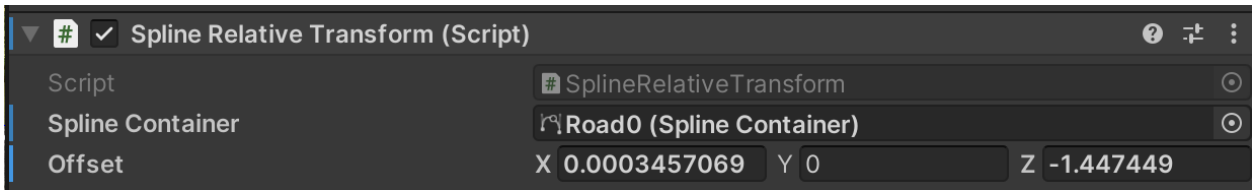


Roads and intersections may have various options available on the component. These will show up on the component if they are configured. In the example above, the road has several options. The main layer controls the core prefab which is used along the road spline - in this case, the difference is the markings used on the road. There's also options for guard rails along the sides of the road, or signs.

You can also change options along the spline, overriding the defaults specified here. To do this, select the Road Spline and make sure you're in spline editing mode. A "D" button should be at the bottom of the toolbar and when selected puts the spline into Data mode. You can double click the spline to add a new control point, which will allow you to change these options at any point along the spline. Note that options do not change immediately at the control point, but rather at the end of each section of road along the spline.

Spline Relative Transform

The SplineRelativeTransform component makes an object relative to a spline in the scene. For instance, you might place a stop sign on a road, and when the road is edited, the sign will move with it. This is a powerful way to add custom objects to your road and keep them in position through road edits.



Note that any object in the MicroVerse content browser with a SplineRelativeTransform component on its root transform will be automatically set to the nearest spline within 30 meters of where it is placed.

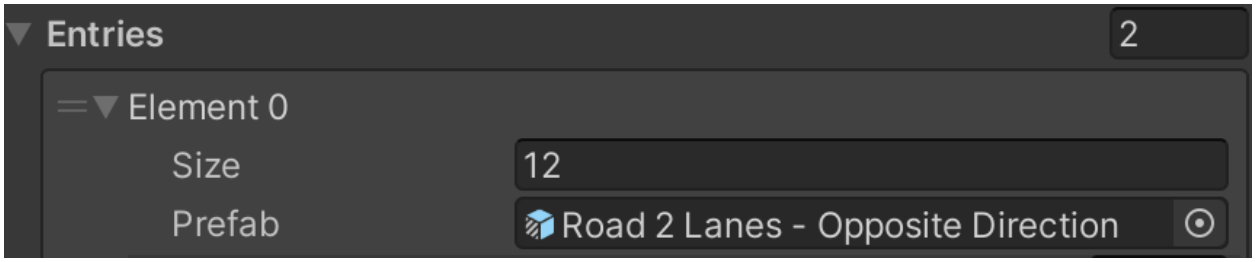
Creating Custom Roads

While a complete system of roads is provided, you can create your own intersection pieces, road segments, and set them up with options, and use the custom shader provided to texture them. The rest of this document is about exactly that.

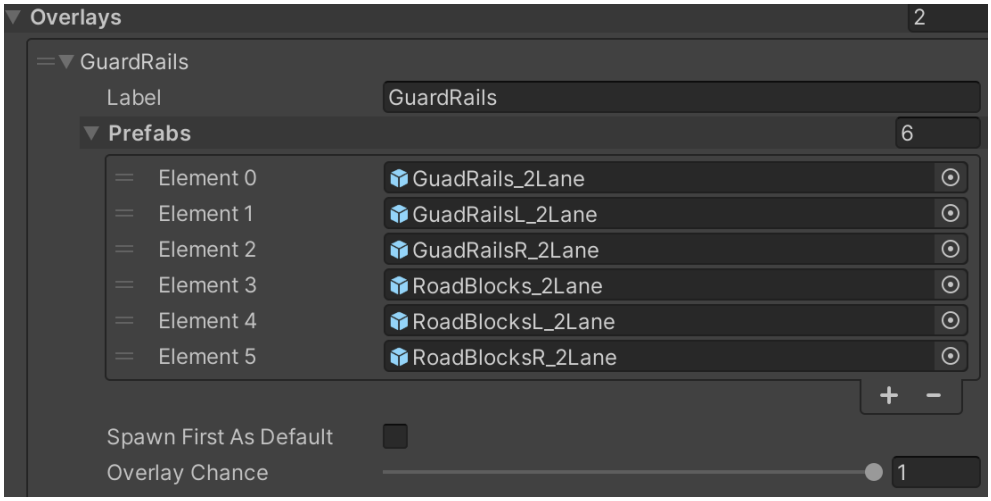
Connections

Connections are defined via a Connector component on a game object. These are placed on intersections, and describe what kind of road will be created when the user double clicks on them by specifying a Road Config.

Road Config



The Road Config specifies which prefab will be instantiated and bent along the spline. It can have multiple elements, which show up as the “Main” option on the component. For instance, this example is a 2 lane road which has marking options for going both directions or a single direction. **Note that the Size option is extremely important!** This is how often the prefab is repeated along the spline, so in this case one of these will be instantiated every 12 meters.

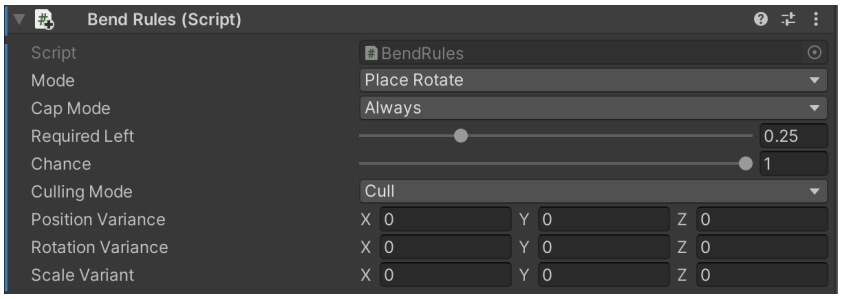


Each Entry has an array of “Overlays”. These are prefab’d scenes that get instantiated along the spline. In this example, the user will see an option for “GuardRails” and be able to select from the prefabs in the list. You can choose to spawn the first option in the list as a default case, or nothing will be spawned by default if not chosen, or specify a chance for how often things are spawned.

You can have as many overlay types as you want, allowing you to layer options for things like guardrails, signage, etc.

BendRules

When a prefab is used in the system, the default is to bend the object along the spline. However, you can add the BendRules component to your object to override this behavior.



The mode option can be used to place objects instead of bend them. Options are:

- Bend : Bend the object along the spline
- Place : Position the object along the spline
- Place Rotate : Position and rotate the object along the spline
- Place Rotate No Slope : Position and rotate, but keep the object upright on hills

Cap Mode allows you to specify when the object is visible. This can be used to place objects at the begin or end of a spline, but not at the begin or end of every instance of the spline. For instance, with pieces that begin or end a fence.

Chance allows you to randomly not spawn objects.

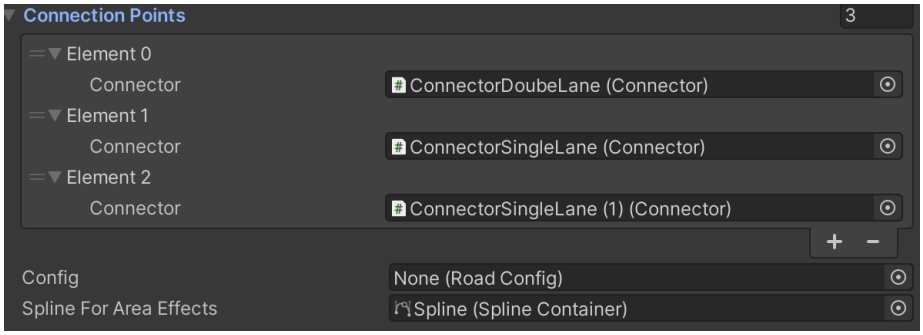
As the scene is repeated along the spline, it may not fit exactly, meaning only a percent of the spline is left. You can use the “Required Left” property to prevent objects from spawning when only a small percent of the spline is used.

Culling mode decides what to do with objects when they reach the end of a partial spline. When set to cull, if the object center is past the end of the spline the object will be removed. If set to clamp, it will be placed by clamped to the end of the spline.

Finally, you can have position, rotation, and scale variance applied to the objects.

Intersections

The intersection component needs to be added to any Intersection in the system.



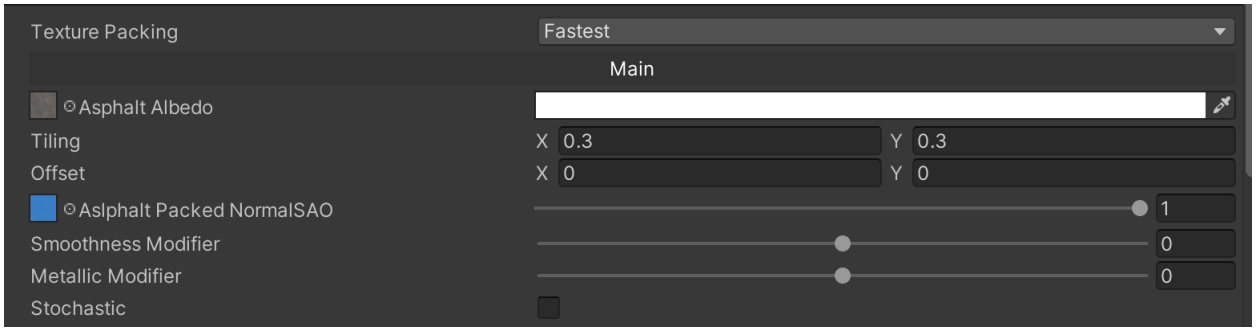
To setup the intersection, add any Connectors to it and place them where the connections should happen. The spline road will be created along the Z (blue) axis, so that should be pointed in the direction you want the spline created. Once the connection game objects are placed, add them to the Connection Point list.

You can specify a Road Config for an intersection as well. This config will be used to provide the user with overlay options for the intersection, allowing you to have prefab options and overlay options (pre-placed fences, etc) for an intersection. Finally, you can use a spline to define an area for how the intersection affects the terrain when using MicroVerse. You can create any number of closed splines, and these will be used to adjust the terrain.

Shading

Included with the package is a special shader to solve some issues with texturing Roads. If you wish to modify this shader, you will need the latest version of Better Shaders and the Better Lit Shader installed. Better Shaders is used to compile the shader for all render pipelines, and the shader includes special effects from the Better Lit shader for things like wetness, puddles, snow, and Trax integration.

The custom shader solves the problem of how to have crisp markings on the road, which don't alias in the distance or blur up close, while also providing for custom effects like road wear, and using a minimal amount of data.



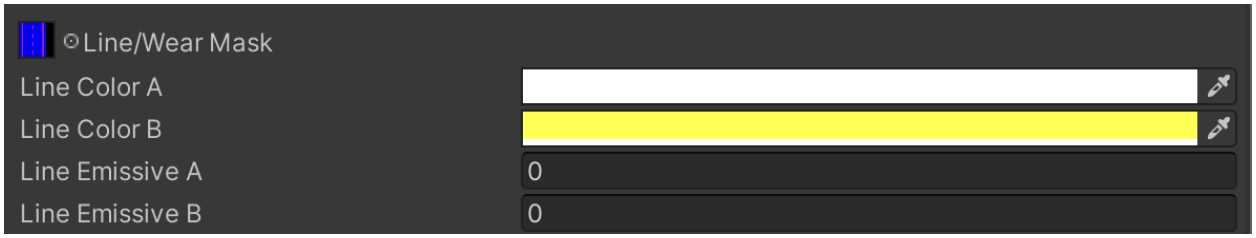
At the top of the shader is the packing mode used for normal, smoothness, metallic, and ambient occlusion data. In fastest mode, the normal, smoothness, and ambient occlusion is packed into a single texture to save sampling cost, and only a metallic value is provided. The channel layout for this texture is:

Red: Smoothness
Green: Normal Y
Blue: Ambient Occlusion
Alpha : Normal X (from the red channel in a normal map)

When in Unity packing mode, two textures are used - a standard normal map, and an HDRP style mask map with the blue channel being unused.

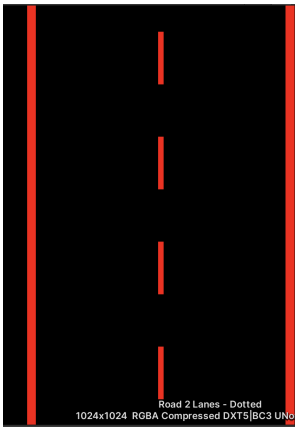
Note that the Albedo map should have a heightmap in the alpha channel. This is primarily used for effects, such as snow build up and puddle placement.

The Asphalt is textured in world space using a single texture sample triplanar technique. You can control its scale, and use stochastic sampling on it, which while more expensive, prevents any tiling from being visible in the texture.

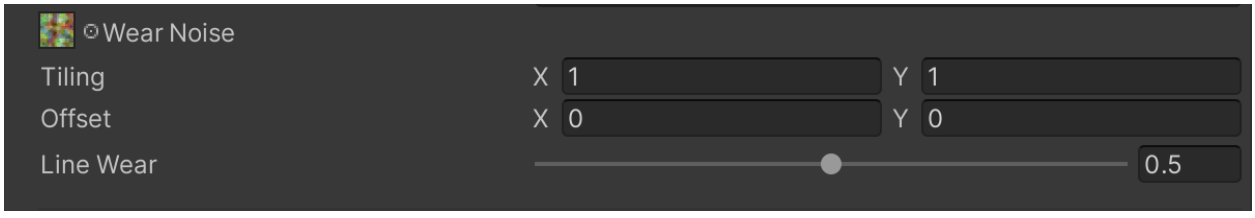


The **Line/Wear Mask** is a special texture which determines where the two line types go and where wear appears on the roads. This is mapped in UV space, and LineA is packed in the red channel with Line B in the green channel. Wear A is in the blue channel, and Wear B is in the alpha channel.

You can specify the colors of the two lines and an emissive amount if you want them to be emissive.

	Here's an example of the red channel of a mask texture. This will create crisp white lines on the road, which do not alias or break down when close. For yellow lines (assuming the above line color properties), you would paint into the green channel.

Wear Noise



Next you can specify a texture to use for noise in the system. Note that it uses the RGB channels each for a different noise. This texture is in world space.

The line Wear value controls how much the noise in the Blue channel affects the lines, fading them where that noise value is high.

There are two wear systems available. These can be used to place cracks, paving issues, and other types of wear along the roads. The wear map is a normal map sampled in world space, but masked by the texture in the Line/Wear mask texture, which is sampled in UV space. This allows you to control how much wear appears in different areas of the road or intersection, but have the cracks which appear not to be the same each time (because they're mapped in world space).

You can adjust the overall weight of the effect, provide a tint value, and adjust the smoothness, occlusion, and normal strength of the effect. The effect is also modified by the noise in red and green of the noise texture sample, and you can control the contrast of that noise to have wear fade in gradually or quickly.

Effects

Wetness, Puddles, Snow and Trax are available from the Better Lit shader for weathering effects. To see more about these effects and their parameters, as well as information about integration with weathering systems like Weather Maker 2 and Enviro, see the Better Lit Shader documentation.