

DOCUMENTATION

SLING DRIFT



Asset Overview

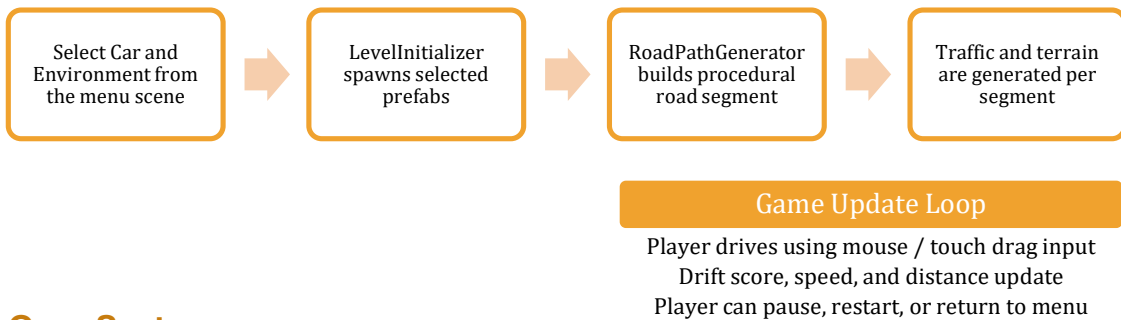
Drift Sling is a procedural endless driving and drifting game system built in Unity. The core gameplay revolves around mouse / touch based sling style control, procedural road generation, dynamic traffic, terrain generation, and real time drift scoring.

This document explains how the systems work together, how to set them up in Unity, and how developers can extend or customize the asset.

Project Requirements

- Unity 2022+ / Unity 6 recommended
- Uses `Rigidbody.linearVelocity` (replace with `Rigidbody.velocity` for older Unity versions)
- PC / Mobile / WebGL friendly

High-Level Gameplay Flow



Core Systems

Road Generation (`RoadPathGenerator.cs`)

Generates an infinite procedural road using mesh deformation.

Road Segment Prefab: Prefab used as the base mesh for each generated road segment. This object is duplicated, deformed, and chained together to form the infinite road.

Road Width: Multiplier applied to the road mesh width during generation. Lower values make narrower roads and increase difficulty.

New Road Threshold Distance: Distance from the player at which a new road segment is generated ahead. Controls how early the next segment spawns to avoid visible pop-in.

Segment Count: Number of road segments kept alive at any time. Higher values increase view distance but cost more performance.

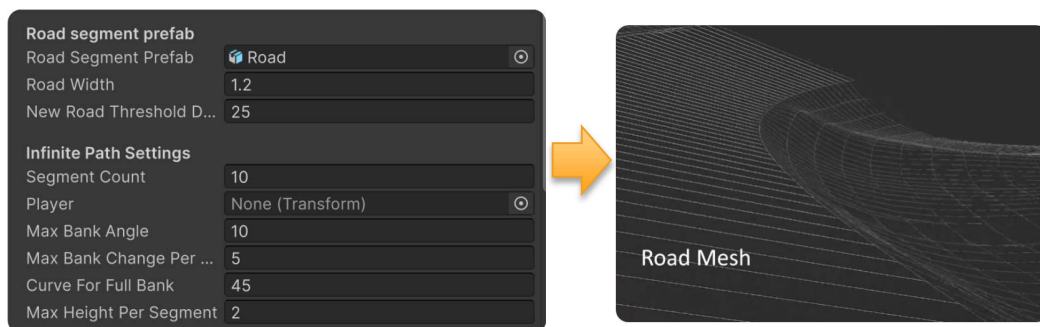
Player: Reference to the player transform used to track position along the road. Required for deciding when to add or remove segments.

Max Bank Angle: Maximum tilt angle applied to road segments during turns. Higher values create more aggressive banking.

Max Bank Change Per Segment: Maximum amount the bank angle is allowed to change between consecutive segments. Prevents sudden, unrealistic banking transitions.

Curve For Full Bank: Curve angle required to reach the maximum bank angle. Smaller values make banking ramp up faster on gentle curves.

Max Height Per Segment: Maximum vertical height change applied between road segments. Controls how hilly the road can become.



3. Terrain Generation (ProceduralTerrain.cs)

Creates terrain strips alongside the road.

Material: Material applied to the generated terrain mesh. Determines the visual look of the terrain surface.

Length: Length of each terrain strip generated alongside a road segment. Larger values push terrain farther away from the road.

Layers: Number of mesh layers extending outward from the road edge. Higher values create smoother terrain slopes.

Slope Curve: Curve controlling how terrain height falls off from the road edge. Used to shape cliffs, flat shoulders, or gentle slopes.

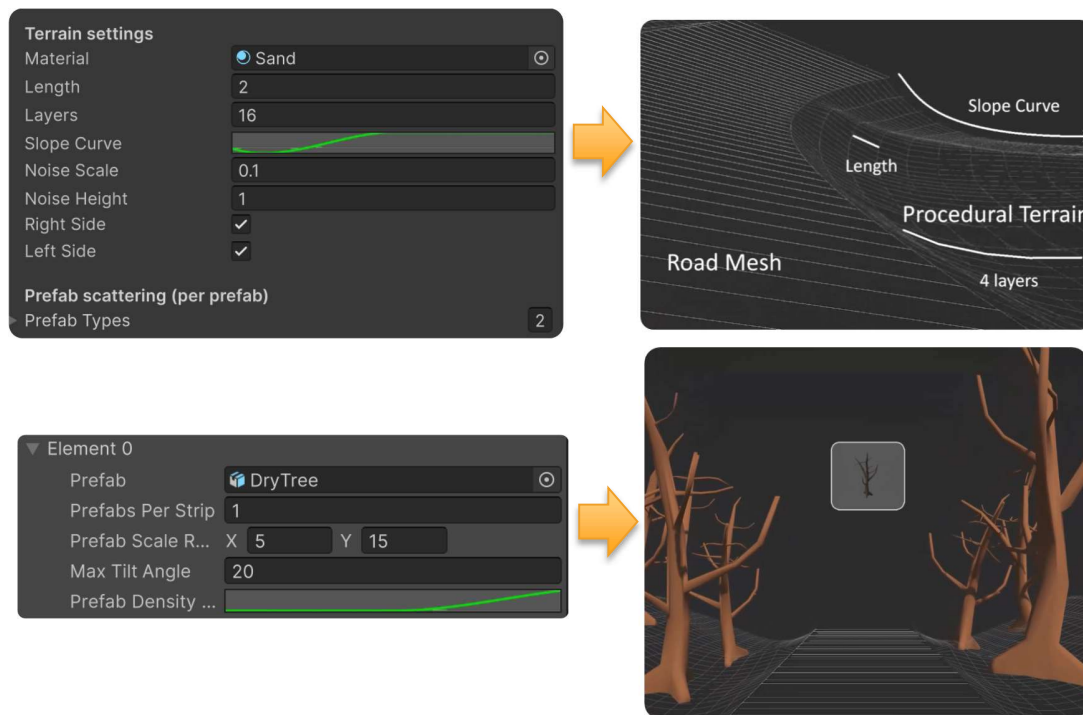
Noise Scale: Scale of Perlin noise applied to the terrain surface. Higher values produce more frequent terrain variation.

Noise Height: Strength of height variation added by noise. Higher values create rougher terrain.

Right Side: Enables terrain generation on the right side of the road. Can be disabled to generate terrain on only one side.

Left Side: Enables terrain generation on the left side of the road. Useful for asymmetric environments.

Prefabs: Array of PrefabTypes class.



Car Controller (MouseDriftCarController.cs)

(This also works with touch input for mobile devices)

This script handles player car control using mouse drag input. Vertical drag controls acceleration, horizontal drag controls steering, and the script applies arcade-style physics, drifting behavior, wheel visuals, braking, and stability adjustments.

Wheel Colliders: WheelColliders are responsible for steering and suspension on the wheels.

Wheel Meshes (visual): Visual transform that follows the rotation and steering of the wheel meshes. Does not affect physics, only visuals.

Max Motor Torque: Maximum torque applied to the rear wheels. Higher values increase acceleration strength.

Arcade Acceleration: Additional forward force applied directly to the Rigidbody. Controls how arcade-like and punchy the acceleration feels.

Top Speed: Maximum forward speed the car is allowed to reach. Caps velocity to keep gameplay controllable.

Max Steer Angle: Maximum steering angle applied to the front wheels. Higher values allow tighter turns.

Brake Torque: Braking force applied when throttle input is released. Controls how quickly the car slows down. Important for drifts.

Rear Steer Factor: This is what gives the vehicle a feeling of drifting. The amount of steering applied to the rear wheels which is corrected by the visuals. Higher values increase drifting angle and oversteer.

Drift Cut Off Speed: Minimum speed required for drift behavior to activate. Below this speed, drifting is reduced or disabled.

Linear Damping: Extra damping applied to the Rigidbody's velocity. Helps stabilize the car and prevent excessive sliding.

Center Of Mass Offset: Offsets the Rigidbody's center of mass. Lower Y values increase stability and reduce rollovers.

Max Vertical Drag: Maximum vertical mouse drag distance used for acceleration input. Limits how much throttle can be generated from dragging.

Max Horizontal Drag: Maximum horizontal mouse drag distance used for steering input. Limits how much steering can be generated from dragging.

Vertical Dead Zone: Minimum vertical drag required before acceleration is applied. Prevents small accidental inputs.

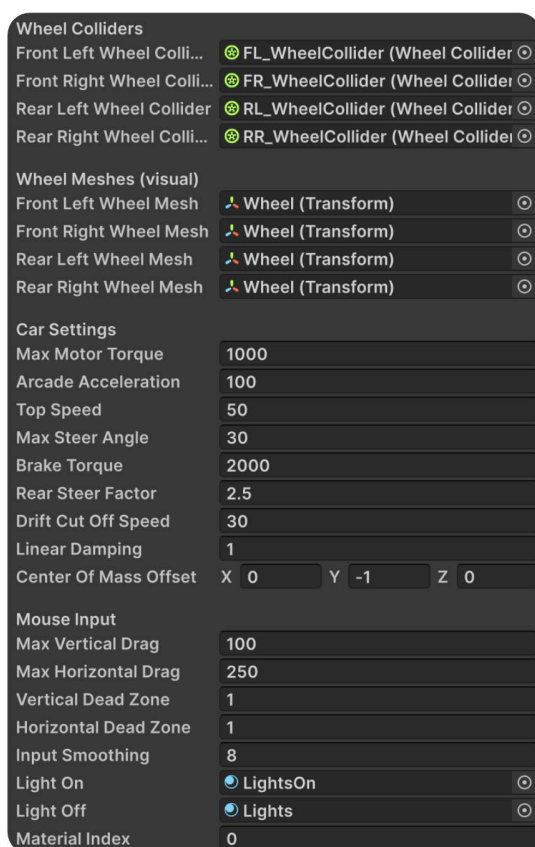
Horizontal Dead Zone: Minimum horizontal drag required before steering is applied. Prevents small accidental steering.

Input Smoothing: Controls how quickly steering and acceleration inputs interpolate. Higher values feel smoother and heavier, lower values feel snappier.

Light Off: Material applied when the car is actively accelerating. Used for visual feedback such as headlights or glow.

Light On: Material applied when the car is not accelerating. Represents idle or coasting state. If the user does not press anything, the car automatically applies brakes and you can see the light turn on.

Material Index: Index of the material slot to swap on the renderer. Allows changing a specific material without affecting others.



Traffic System

TrafficSpawner places AI traffic cars on new road segments. TrafficCarAnimator moves and aligns traffic cars to the road and railings. It uses raycasts to accomplish this.

Car Prefabs: List of traffic car prefabs that can be spawned on the road. One is randomly selected per spawn.

Spawn Probability: Chance that a road segment will spawn traffic. Higher values result in denser traffic.

Opposite Direction Probability: Chance that spawned traffic will drive against the player's direction. Used to introduce oncoming vehicles.

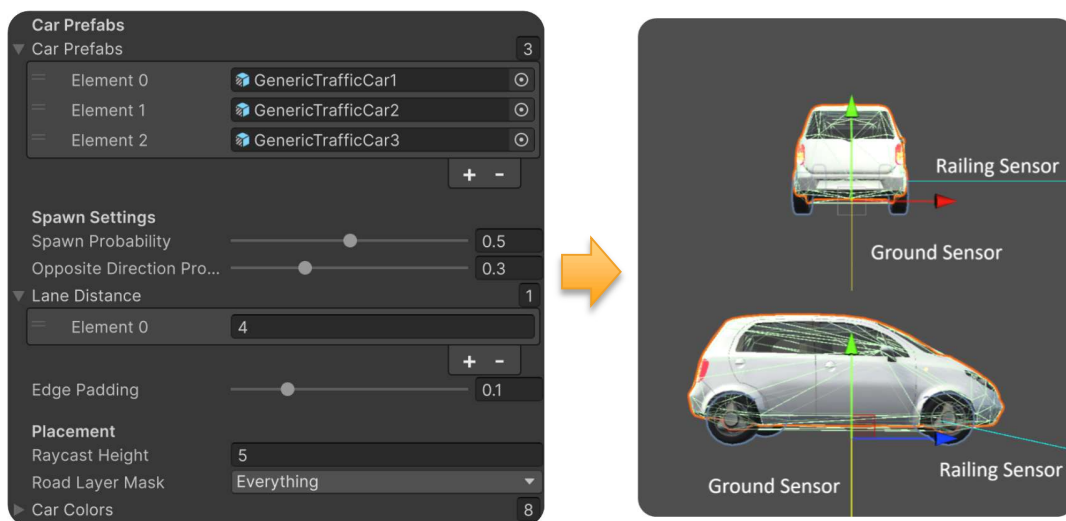
Lane Distance: Distance from the railing that traffic cars try to maintain. Controls how far into the road lane traffic drives.

Edge Padding: Normalized padding that prevents traffic from spawning too close to segment edges. Reduces clipping and unsafe spawns.

Raycast Height: Height from which traffic placement raycasts downward onto the road. Must be high enough to always hit the road surface.

Road Layer Mask: Layer mask used to detect the road during placement raycasts. Ensures traffic snaps only to valid road geometry.

Car Colors: List of colors randomly applied to traffic cars at spawn time. Used to add visual variety to traffic.



7. Menu & Settings

MainMenuController manages car/environment selection and settings. LevelSettings stores difficulty and quality presets.

Settings Include:

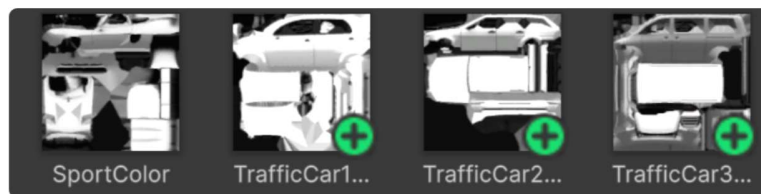
- **Traffic density:** Spawn probability of vehicles per segment
- **Road width:** 1 – 1.2 is ideal

- **View distance:** 8 – 15. Refers to how many segments are generated initially.
- **Traffic speed:** 6-12 is ideal.
- **Shadow quality:** Choose from high, medium and low.



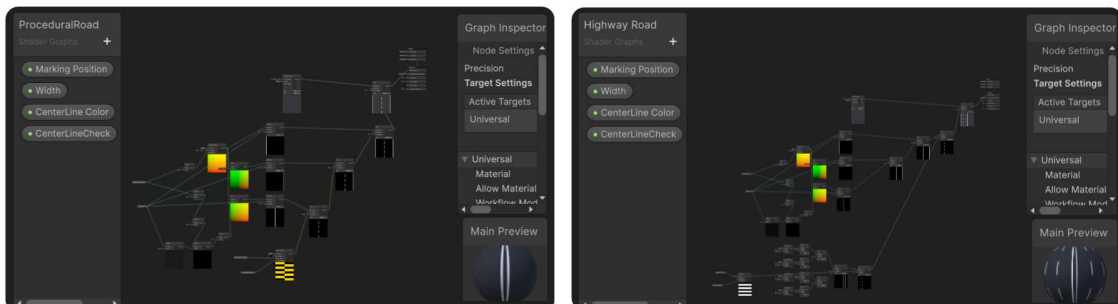
Textures (Procedural Road Shader, Car Textures)

The textures in this asset are made to be customized. The car textures can have a tint added to them to color them. By default the body texture of the car is white.



Road Shader graph:

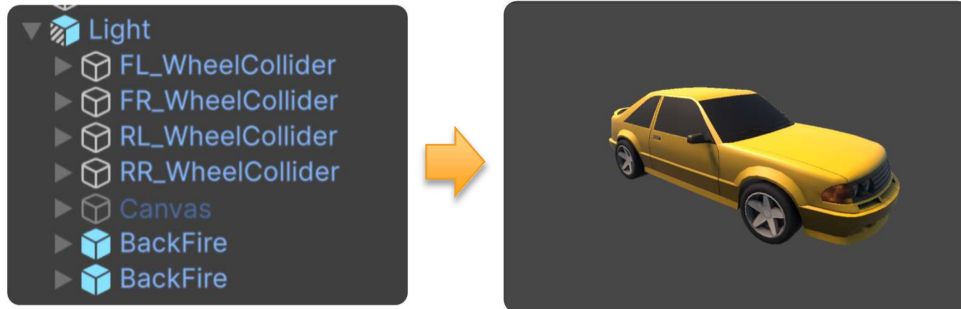
HighwayRoad.shadergraph and ProceduralRoad.shadergraph combine rectangle nodes and checkerboard nodes to make the markings on the road. The user can access the parameters in the shader directly by clicking on the generated road object and going over to the material.



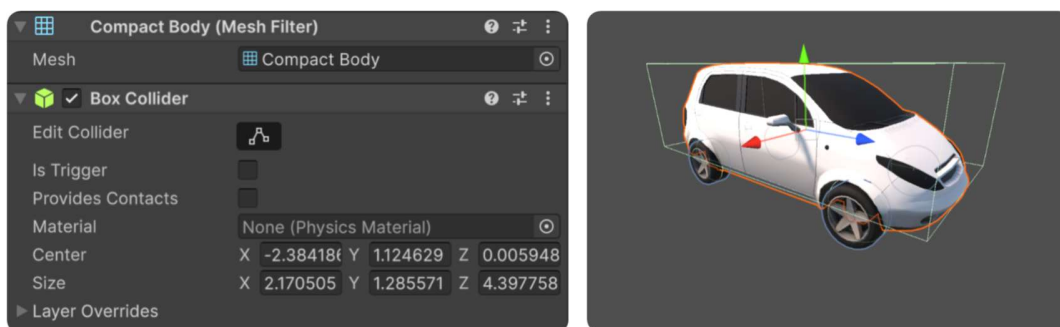
Extending the Asset

Adding Cars

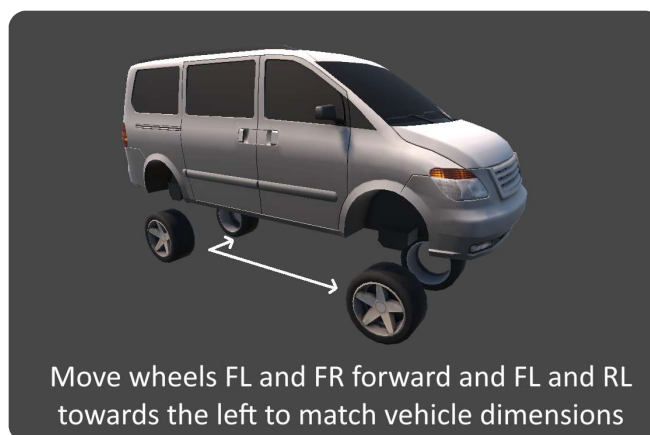
- Duplicate an existing car prefab



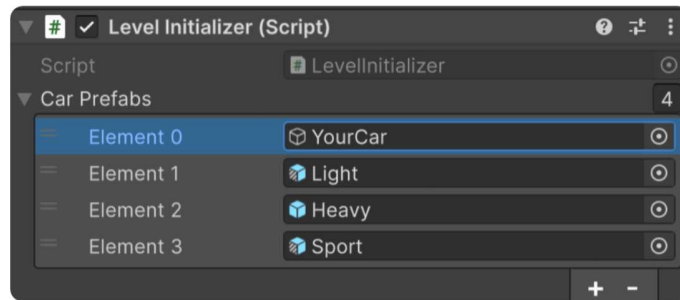
- Replace the mesh with your model's mesh or add a new GameObject with your model and place it under the car controller parent object.



Adjusting the WheelBase:

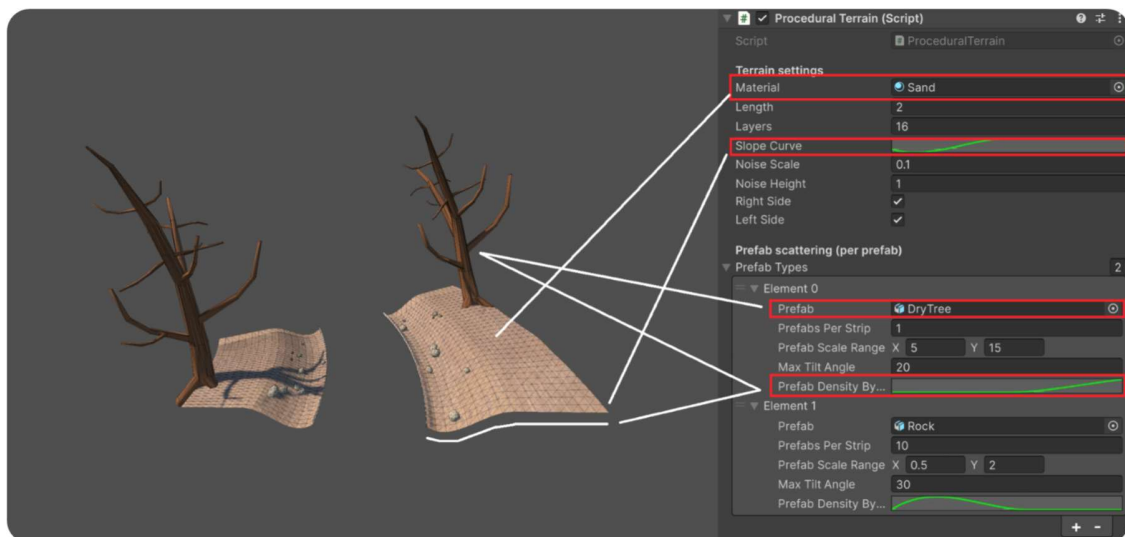


- Adjust the colliders as needed and add to Menu and LevelInitializer arrays



Adding Environments:

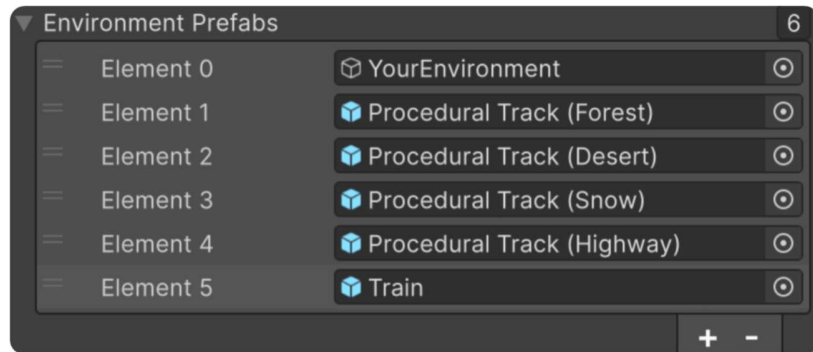
- Duplicate the Procedural Terrain Prefab.



Replace the material field with the material you have made for your custom environment. The slope refers to how you'd like the terrain to be shaped.

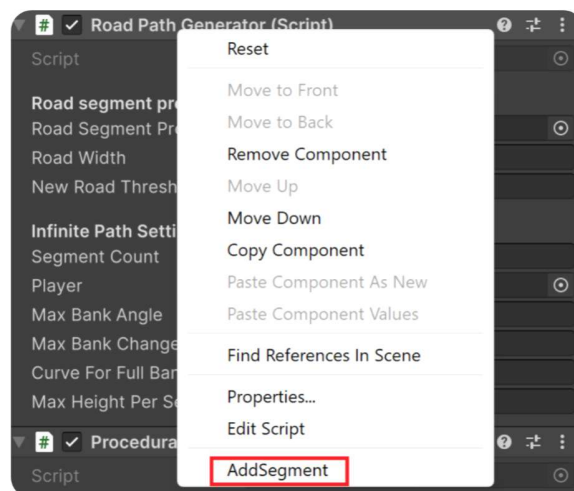
For prefabs, you can add your own prefabs and arrange them on the terrain. Prefab density per layer refers to the probability distribution of the placement of prefabs per layer. 0,0 refers to the inside of the terrain close to the road with a probability of 0 and 1,1 refers to the outside of the terrain with a probability of 1.

- Register prefab in menu and initializer



Adding Gameplay Elements:

- Best hook: `RoadPathGenerator.AddSegment()`. You can test your added element by right clicking on the Road Path Generator script and click on Add Segment. Do not forget to add your script's reference (`yourScript.GetComponent<YourScript>().Function`) in the start as well as `AddSegment()` function.



Tuning Guide

Drift feel (for more oversteer): Increase Rear Steer Factor and Max Steer Angle to make the rear slide wider and recover slower. If the car becomes uncontrollable, increase Linear Damping slightly and lower Center Of Mass Offset Y further (more negative).

Drift feel (more grip / easier control): For regular car movement, increase the threshold to 200 (or a really high number). Reduce Rear Steer Factor and increase Brake Torque to shorten slides and make the car settle faster. If steering feels too weak, increase Max Steer Angle without increasing Rear Steer Factor.

Drift activation timing: Drift Cut Off Speed decides when drifting starts to matter at speed. Lower it to allow drifting at slower speeds, raise it to require more speed before the car breaks loose. Alternatively, activate this with the press of a button.

Acceleration punch: Max Motor Torque controls how strongly the rear wheels push under throttle. Arcade Acceleration adds extra forward force to make acceleration feel more “boosty” without changing wheel behavior.

Top speed and pacing: Top Speed caps how fast the car can go so the road stays readable. If you increase Top Speed, also increase Brake Torque or Linear Damping so the player can recover from mistakes.

Input responsiveness (steering and throttle): Input Smoothing controls how quickly the car reacts to mouse drag changes. Higher values feel heavier and more stable, lower values feel instant but twitchy.

Input reach (how far you must drag): Max Vertical Drag and Max Horizontal Drag set how much movement maps to full throttle and full steer. Increase these if the player hits max input too easily, decrease them if the player struggles to reach full control.

Input noise filtering: Vertical Dead Zone and Horizontal Dead Zone prevent accidental micro-inputs. Raise them if the car jitters when clicking, lower them if the car feels unresponsive.

Road difficulty (space to drift): Road Width directly changes how much room the player has to correct drifts. Narrow roads require earlier braking and smaller steering swings.

Road readability (banks and hills): Max Bank Angle and Max Height Per Segment define how extreme turns and hills can become. If players lose control on random sections, reduce Max Bank Change Per Segment and Max Height Per Segment.

View distance and performance: Segment Count controls how much road exists ahead and behind the player. Increase it for smoother visual continuity, decrease it if CPU usage rises due to terrain, railings, and traffic.

Traffic difficulty (density): Spawn Probability controls how often a segment gets traffic. Increase it for more cars, decrease it to keep the road clear for learning drift control.

Traffic difficulty (oncoming risk): Opposite Direction Probability adds head-on cars that reduce reaction time. Keep it low for casual play and raise it only when Road Width is generous.

Terrain cost vs detail: Layers and prefab scattering are the biggest terrain cost multipliers. Reduce Layers and prefab counts first if frame time spikes during generation. Alternatively you can also introduce fog to counteract the issue of visible generations.

Common Issues & Solutions

Car does not move: This usually happens when WheelColliders are not assigned or the player is not dragging downward enough to generate throttle. Assign all four colliders, reduce Vertical Dead Zone, and increase Max Motor Torque or Arcade Acceleration if acceleration is still weak.

Car moves but cannot steer: This is typically caused by missing front wheel collider references or Max Steer Angle being too low to notice. Assign the front colliders and raise Max Steer Angle or reduce Max Horizontal Drag so steering reaches full input sooner.

Car instantly spins out on small drags: Steering input is too strong or drift is too aggressive for the current speed. Reduce Rear Steer Factor, increase Input Smoothing, and decrease the max steer angle.

Car slides forever after releasing input: Braking and damping are too low, so the rigidbody keeps carrying momentum. Increase Brake Torque and Linear Damping until the car reliably settles without killing all drift.

Speed / distance / drift display reads wrong or stays at zero: This can happen if the project Unity version does not support Rigidbody.linearVelocity. Replace all linearVelocity usage with Rigidbody.velocity in the car and UI scripts. This can also occur if the canvas element (text box) is not assigned to the CarMetricsUI.cs script.

Traffic cars spawn too close to segment edges: Spawn position is too near the segment start/end or the padding is too small. Increase Edge Padding to keep spawns away from seam areas.

Traffic jitters left or right while driving: Railing correction and yaw response are based on Physics.Raycast. If there is something that interrupted the car during this correction, the car can go in a different direction and mess up.

Terrain appears on only one side: Side toggles are not enabled for both sides in ProceduralTerrain. Enable Left Side and Right Side, and confirm there are terrain generators present for the environment instance.

Attribution and licensing

Font Attribution: This asset uses the Bungee Spice font. Bungee Spice is a Google Font and is licensed under the SIL Open Font License (OFL), which permits use, modification, and redistribution.

3D Model Attribution: Vehicle models are based on the Generic Passenger Car Pack by Comrade1280. Source: <https://sketchfab.com/3d-models/generic-passenger-car-pack-20f9af9b8a404d5cb022ac6fe87f21f5>

In case you have any questions, please contact us at info@aikodex.com.

Happy Drifting!

AiKodex