In the Car folder you'll find a few ready-made Car prefabs, which are used in the Car sample scenes. These prefabs demonstrate how you can use the Car scripts we've included, including some AI-controlled variants which are able to drive directly towards a specified target, or follow a waypoint-based route.

The scripts included have been designed to be modular in nature, so that the extra features of a particular car can be set up (such as brake lights, engine sound, and skid trails).

The simplest way to get started setting up your own car is to use our "CarPrefabEmpty", and to follow these steps:

1) Start with a suitable scene. There ought to be enough flat ground to drive around on.

2) Place the "CarPrefabEmpty" in the scene.

3) Place your 3d model into the scene, as a child of the "CarPrefabEmpty" GameObject. Notice that your car model should be at the same level as the "WheelHub" objects that were already present.

4) Make sure your car's position in the inspector is set to 0,0,0. This will ensure your car's position is aligned with the Empty Prefab's position.

5) Expand the hierarchy of your car model in the hierarchy so that the wheel models of your car are visible.

6) Select each "WheelHub" in turn, and drag the corresponding wheel model from your car into the "Wheel Model" slot of the Wheel component on the WheelHub GameObject. Then click the "Align to assigned wheel model". If your car has four wheels, you'll need to do this step four times, once for each wheel.

7) Hit play, and try driving your car!

You'll probably want to add a camera rig so that the camera follows the car as it drives off. See the Camera Rig guidelines for instructions on how to do that.

The Car script exposes a number of properties which determine the power, maximum speed, and handling of the car, however ultimately the script is designed to provide a fun arcade game-like feel rather than attempting to be a realistic simulation of car handling. The gears and revs values are purely cosmetic, and are calculated retrospectively - they can be used by the engine sound component or GUI to display revs and gear.

For more detail about each setting, see the comments in the script.

The CarUserControl script takes input from the "CrossPlatformInput" class included in the sample assets, however if you're not targeting mobile or prefer to use a different system to read input, you can simply use unity's built in Input class in place of CrossPlatformInput. For more information, see the Cross Platform Input guidelines.

The sample scenes provided work on standalone builds and also include cross platform touch and tilt controls which are set up and ready to publish to mobile.

The CarAIControl component can be added instead of the CarUserControl component, allowing the car to be AI controlled. You can set it up to either head towards a specified target, or (by using the WaypointProgressTracker) follow a waypoint-defined route. See the WaypointCircuit guidelines for instructions.