

# Car Controller Script

This script is a C# script designed for controlling a car in a game. It contains variables and methods that are used to control the car's movement, visuals, and audio.

## Public Variables

### Suspension

- SuspensionDistance (float): This determines how much the wheels can move up and down.
- suspensionForce (float): The force with which the suspension pushes the car upwards.
- suspensionDamper (float): The damping force of the suspension.
- groundCheck (Transform): A point from which a ray will be shot in downward direction and it will check whether the vehicle is grounded or not. So this point should be above the wheels. But, it is also the point where the accelerating and braking force will apply. So if you place it slightly below the center of the body, you will get accelerating and braking motion.
- fricAt (Transform): The point where the sideways friction force is applied. if you place this slightly below the center of body, the vehicle will lean into the turns. But don't place it too much below the center. As it can cause the vehicle to roll, or flip while turning.
- CentreOfMass (Transform): The center of mass of the car when it's in air. So put it just below the wheel's.

### Car Stats

- accelerationForce (float): The force with which the car accelerates.
- turnTorque (float): The torque applied to turn the car.
- brakeForce (float): The force with which the brakes are applied.
- frictionForce (float): The sideways force on the vehicle body to oppose sliding.
- dragAmount (float): The drag force that the car experiences.
- TurnAngle (float): The angle at which the wheels turn.
- maxRayLength (float): The maximum length of the ground check ray.
- sleepTime (float): The time taken for the wheels to turn.

### Visuals

- TireMeshes (Transform[]): The wheel mesh Transforms that rotates (for all wheels).
- TurnTires (Transform[]): The wheel mesh Transforms that turn locally when the vehicle is turning(for front wheels).

## Curves

- `frictionCurve` (`AnimationCurve`): The curve that determines the friction force based on the car's sideways velocity.
- `accelerationCurve` (`AnimationCurve`): The curve that determines the acceleration force amount based on the car's forward velocity.
- `separateReverseCurve` (`bool`): A `bool` that determines whether a separate curve is used for reversing (reverse acceleration).
- `ReverseCurve` (`AnimationCurve`): The curve that determines the acceleration force amount based on the car's backward velocity when reversing.
- `turnCurve` (`AnimationCurve`): The curve that determines the torque applied to turn the car based on the car's forward velocity.
- `driftCurve` (`AnimationCurve`): The curve that determines the angular drag force experienced during drifting or you can say sideways slipping.
- `engineCurve` (`AnimationCurve`): The curve that determines the pitch of the engine sound based on forward velocity of the car.

## Other Settings

- `engineSounds` (`AudioSource[]`): The sound sources for the car's engine. These are present on gameobject named "body" under the parent transform of body. The first 2 audios are engine sounds. And the last one is the skid, or drift sound.
- `airDrag` (`bool`): A `bool` that determines whether air drag is applied to the car.
- `SkidEnable` (`float`): The sideways speed at which the car can skid(skidmarks will appear).
- `skidWidth` (`float`): The width of the skid marks.

## Private Variables

- `carControls` (`CarInputs`): An instance of the `CarInputs` class.
- `rb` (`Rigidbody`): The `Rigidbody` component of the car.
- `carVelocity` (`Vector3`): The velocity of the car.
- `hit` (`RaycastHit`): The result of the ground check.
- `grounded` (`bool`): A flag that determines whether the car is on the ground.

## Public Methods

### `audioControl()`

This method is responsible for controlling the car's audio. It sets the pitch of the engine sound according to the current speed of the car, and it also handles the skid sound effect when the car is drifting. It has no parameters and no return value.

### **tireVisuals()**

This method is responsible for rotating the tire meshes and rotating the turn tires(front) based on the current steering angle of the car. It has no parameters and no return value.

### **accelerationLogic()**

This method calculates and applies the acceleration of the car based on the input from the player and the current speed of the car. It has no parameters and no return value.

### **brakeLogic()**

This method calculates and applies the braking force to the car. It has no parameters and no return value.

### **frictionLogic()**

This method applies the sideways friction force to the car when it is turning or drifting. It has no parameters and no return value.

### **turningLogic()**

This method calculates and applies a torque to the car for turning the car body based on the current speed of the car. It has no parameters and no return value.