Arcade Car Controller: (Lite version)

Perfect Games

Hello! Thanks for downloading the asset, I hope you find it useful.

If you liked the asset, you can rate it on the asset store page. This helps a lot in promoting assets.

You can also take a look at my other assets:

An asset with excellent car physics:
Universal Car Controller

ACC series, well optimized for mobile projects:
Arcade Car Controller (Lite version)
Arcade Car Controller (Standard version)
Arcade Car Controller (Multiplayer version)

Third-party assets that I myself use and recommend to everyone:

EasyRoads3D Pro v3 - An asset with which it is easy to create roads for games of any genre.

Gaia 2 - Terrain & Scene Generator - An asset with which it is easy to create and generate terrains of any complexity.

Contents

Introduction

First of all, I want to thank you for download this asset, you are helping me to improve. And sorry for my bad English, i'm learning it.

Full version: http://u3d.as/1upP

I have a small request for you: If you like an asset, please write a review on the AssetStore page, it helps a lot in promoting and motivating me to improve and add content.

The primary objective of this project is an easy arcade control, while not losing the feeling of a controlled machine.

The project is being developed for weak devices, so I try to optimize the code, use as few materials as possible, use more compressed textures, use polygon saving.

If you find a bug or have any suggestions, please let me know, the contacts are listed at the end of this document.

Simplified WheelCollider setup

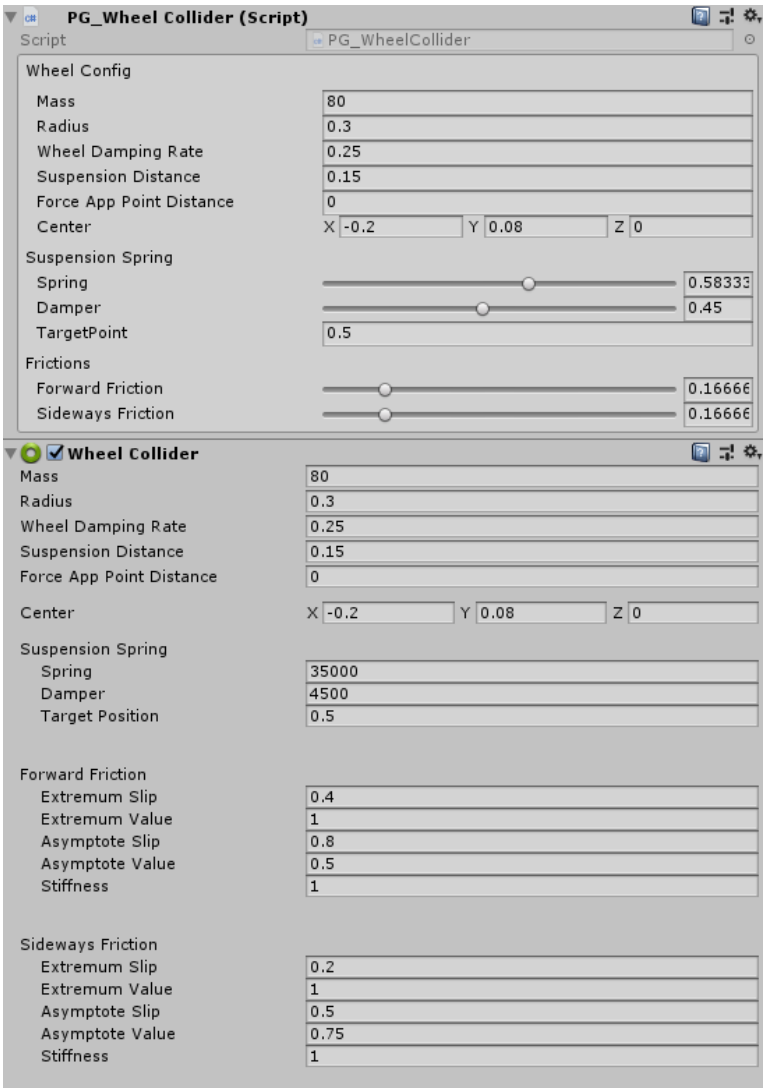
PG_WheelCollider.cs

By popular demand, I created this class. Now the collider slide can be adjusted from 0 (Low Friction) to 1 (High Friction).

All values are duplicated from the standard wheel collider.

For correct editing, you need to edit the PG_WheelCollider.

Also, using this class, it is easy to change the parameters of the collider at runtime.

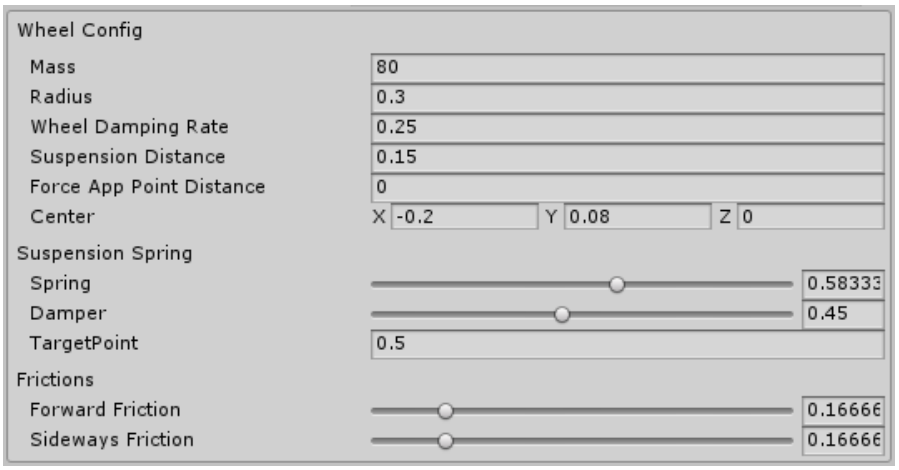


If you are not satisfied with the minimum or maximum values, you can change them in the class PG_WheelCollider: Minimum and Maximum friction constants.

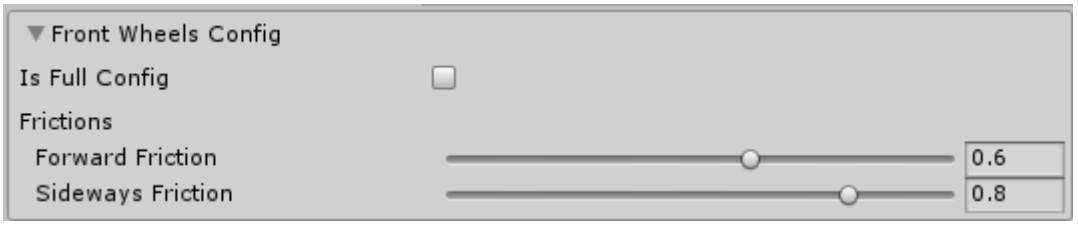
PG_WheelCollider this class uses the PG_WheelColliderConfig structure with attribute [FullField].

```
[SerializeField, FullField] PG_WheelColliderConfig WheelConfig;
```

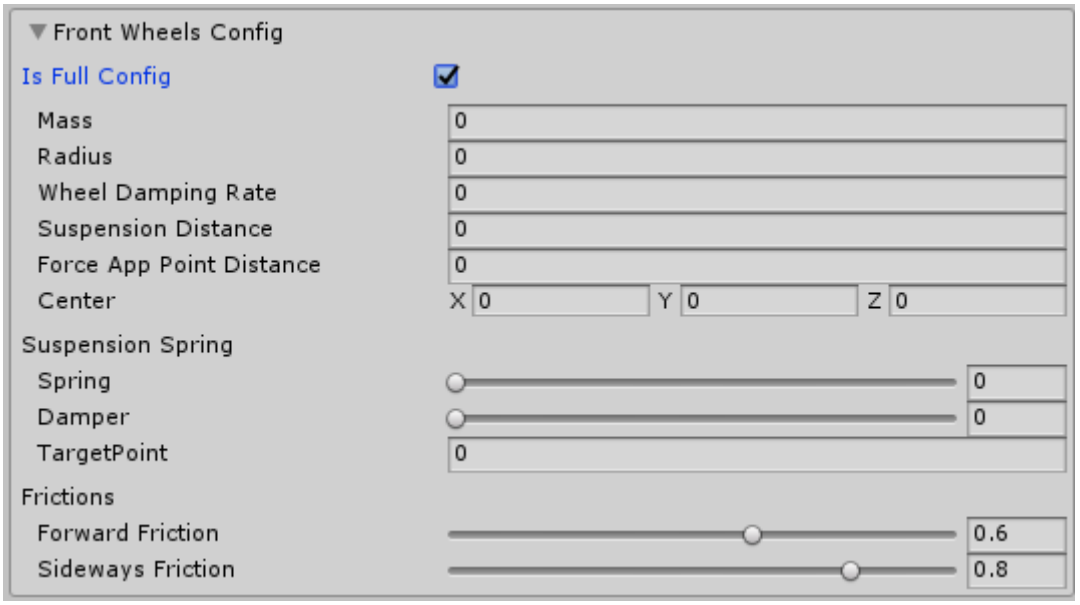
With this attribute, the field is fully displayed.



The structure is used in the mode settings without the [FullField] attribute and the IsFullConfig switch turned off. To transfer only friction parameters to the wheels.



If you turn on the IsFullConfig switch, then all the WheelCollider parameters will be transfer.


Front Wheels Config
Is Full Config
Mass
0
Radius
0
Wheel Damping Rate
0
Suspension Distance
0
Force App Point Distance
0
Center
X 0
Y 0
Z 0
Suspension Spring
Spring
0
Damper
0
TargetPoint
0
Frictions
Forward Friction
0.6
Sideways Friction
0.8

Car settings

The physics of the machine is based on standard WheelColliders. A very important place is occupied by the balance of all settings of car.

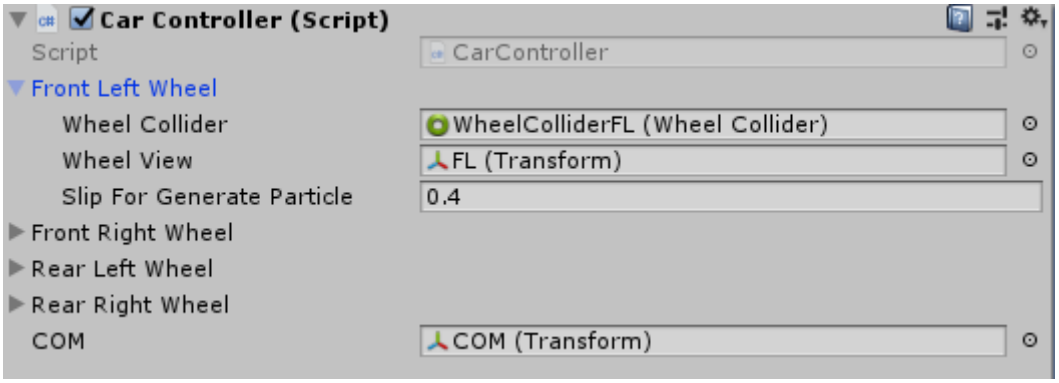
Driving behavior is determined by the wheel colliders settings and helper settings.

CarController.cs: Main gameplay class.

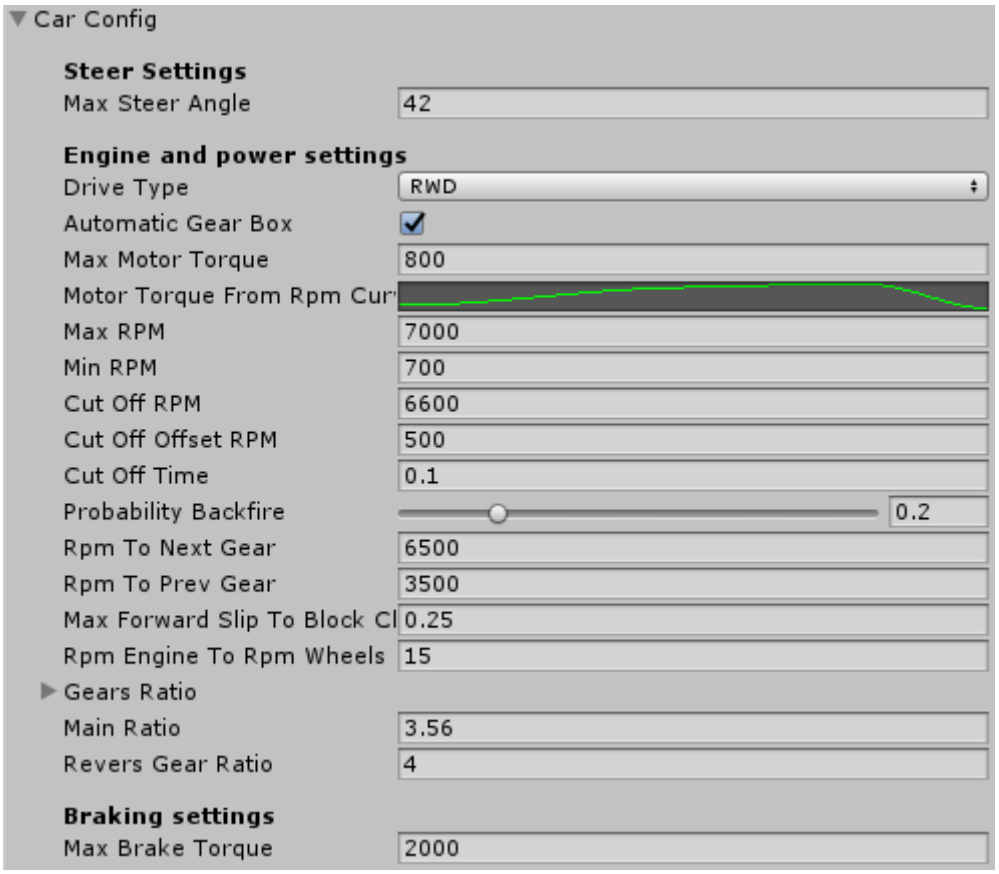
All car settings can be used in any genre.

In the fields of the wheel type is configured: links to, WheelCollider, wheel view, and the value of the slip at which particles are emitted.

COM: Car center of mass



Max steer angle: the max angle at which the front wheels turn.



Power settings: Settings for the engine and gearbox simulation. I wanted that all settings were close to reality.

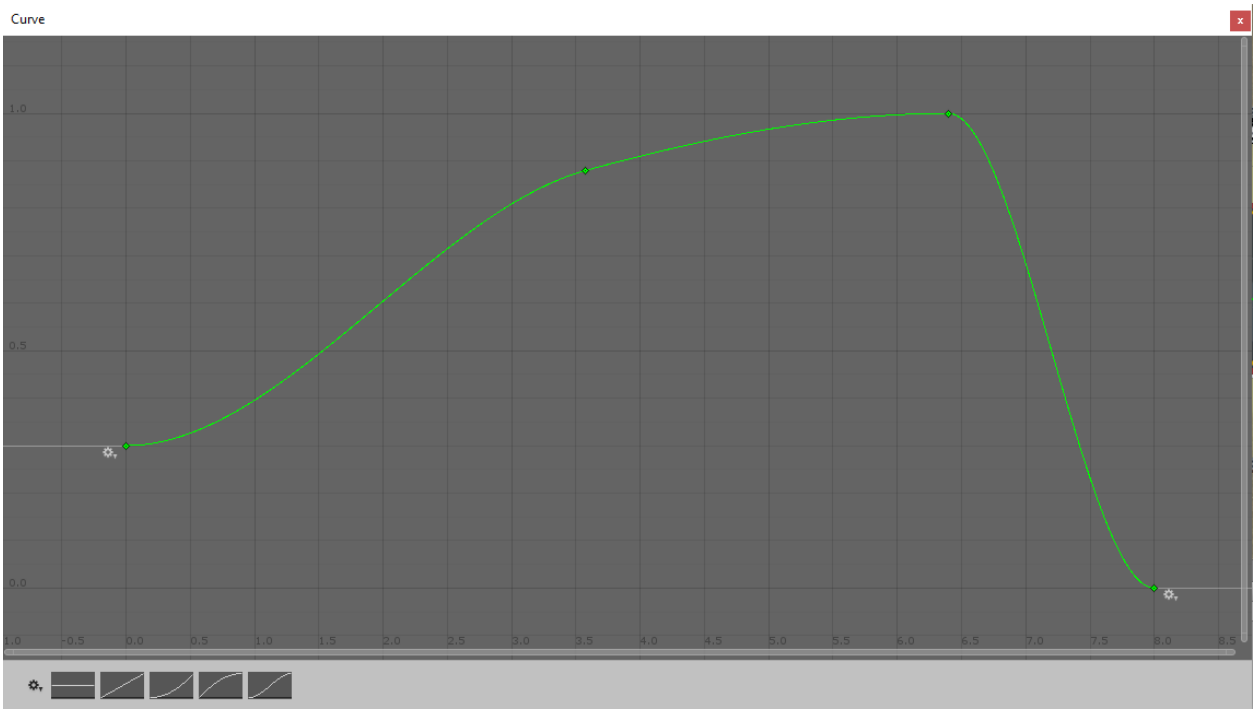
AutomaticGearBox: TODO It is necessary to implement a manual gear shift, now only the automatic gearbox works.

MaxMotorTorque: Torque issued by the engine in max point, without multiplier gears ratio.

MotorTorqueFromRpmCurve :

By horizontal axis the current engine rpm x1000.

By vertical axis the current motor torque multiplier (1 is max, 0 is min).



MaxRPM: Max rpm for engine.

MinRPM: Min rpm for engine.

CutOffRPM: RPM on which the cut-off works.

CutOffOffsetRPM: The count of RPM lost during the cut-off.

RpmToNextGear: RPM at which the next gear is change, if the wheels do not forward slip.

RpmToPrevGear: RPM at which the previous gear is change, if the wheels do not forward slip.

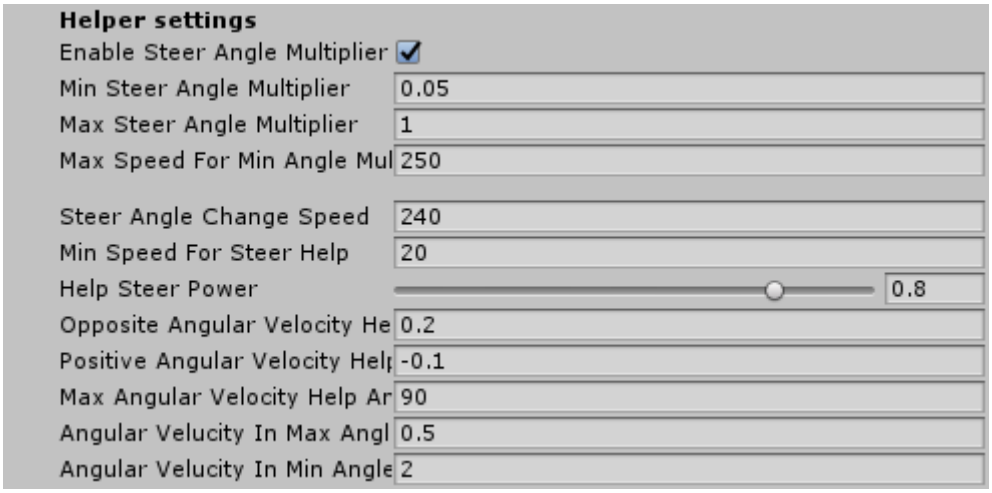
MaxForwardSlipToBlockChangeGear: Max forward slip at which gear shifts are possible.

RpmEngineToRpmWheelsSpeed: The speed of change of engine speed.

GearsRatio, MainRatio, ReversGearRatio: Gear ratios of the gearbox. All data is taken from the documentation for the car's gearbox.

MaxBrakeTorque: Max brake torque.

CarConfig also sets up helper variables. In the main and multiplayer versions, these settings are in the regime settings.



EnableSteerAngleMultiplayer: Restricting steering at high speed.

MinSteerAngleMultiplayer: Minimum steering wheel multiplier. Achieved at maximum speed (MaxSpeedForMinAngleMultiplayer).

MinSteerAngleMultiplayer: Maximum steering wheel multiplier. Achieved at minimum speed (0).

MaxSpeedForMinAngleMultiplayer: The speed at which the maximum restriction of steering rotation is included.

SteerAngleChangeSpeed: The rate of change of the angle of rotation of the front wheels.

MinSpeedForSteerHelp: The speed at which management assistance is enabled.

HelpSteerPower: The power of help with side drift.

OppositeVelocityHelpPower: The force of additional rotation of the rigid body is used to stop the current drift.

PositioveAngularVelocityHelpPower: The force of the additional rotation of the solid is used to enhance the current drift.

AngularVelucityInMaxAngle: Max angular velocity, reached at max drift angles (For limit angular velocity).

AngularVelucityInMinAngle: Max angular velocity, reached at min drift angles (For limit angular velocity).

There are 3 cars in the asset with different control settings.

Shader: AlphaColorMask

Fields:

Color - Mixing color.

Albedo (RGBA) – Map with base color with alpha. In a pixel with transparency 0, color is taken only from the texture, where alpha 1 color of the texture is mixed with the color of the material.

Metalic (RGBA) – Map metallic and Smoothness. The Smoothness is taken from the map where the alpha of the main color is 0. Where the alpha of the main color is 1 the Smoothness is taken from the Smoothness field.

Preparing texture is easiest with Adobe Photoshop. Simply using a mask hides those places that would not like to paint with the help of color from the material.

Contacts

On all questions you can contact by e-mail:

Discord :

#perfect-games

e-mail:

Bugrov1990@gmail.com

Bugrov1990@mail.ru

war.90@mail.ru