

Arcade Car Controller (ACC)



Perfect Games

Contents

Introduction	3
Code style	4
Simplified WheelCollider setup	5
Regime Settings	7
How to create car	11
How to debug	13
Car settings	14
AI Control	19
Balance settings	21
Split level loading	22
How to create a track	23
Positioning System	26
Main menu and game menu	28
In race UI	31
Shader: AlphaColorMask	33
Mesh Painter	34
Update history	36
Contacts	39

Introduction

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

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.

I also 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.

Code style

Private and public fields or properties belonging to the class begin with a capital letter:

```
[SerializeField] WheelCollider WheelCollider;
```

```
float StartDistance;
```

```
Vector3 TargetPoint { get { return Target.TransformPoint(Offset); } }
```

In the event that a variable has a property, the variable begins with a small letter, the property with a large.

```
[SerializeField] float steerAngleChangeSpeed = 1;
```

```
public float SteerAngleChangeSpeed { get { return steerAngleChangeSpeed; } }
```

Parameters of the method and variables declared in the body of the method begin with a small letter.

```
/// <summary>
/// Update controls of car, from user control (TODO AI control).
/// </summary>
/// <param name="horizontal">Turn direction</param>
/// <param name="vertical">Acceleration</param>
/// <param name="brake">Brake</param>
ссылка: 1
public void UpdateControls (float horizontal, float vertical, bool brake) {
    float targetSteerAngle = horizontal * MaxSteerAngle;
    CurrentSteerAngle = Mathf.MoveTowards(CurrentSteerAngle, targetSteerAngle, Time.deltaTime * SteerAngleChangeSpeed);

    CurrentAcceleration = vertical;
    CurrentBrake = brake ? MaxBrakeTorque : 0;
}
```

All fields, properties and method names are chosen to maximize the destination.

Simplified WheelCollider setup

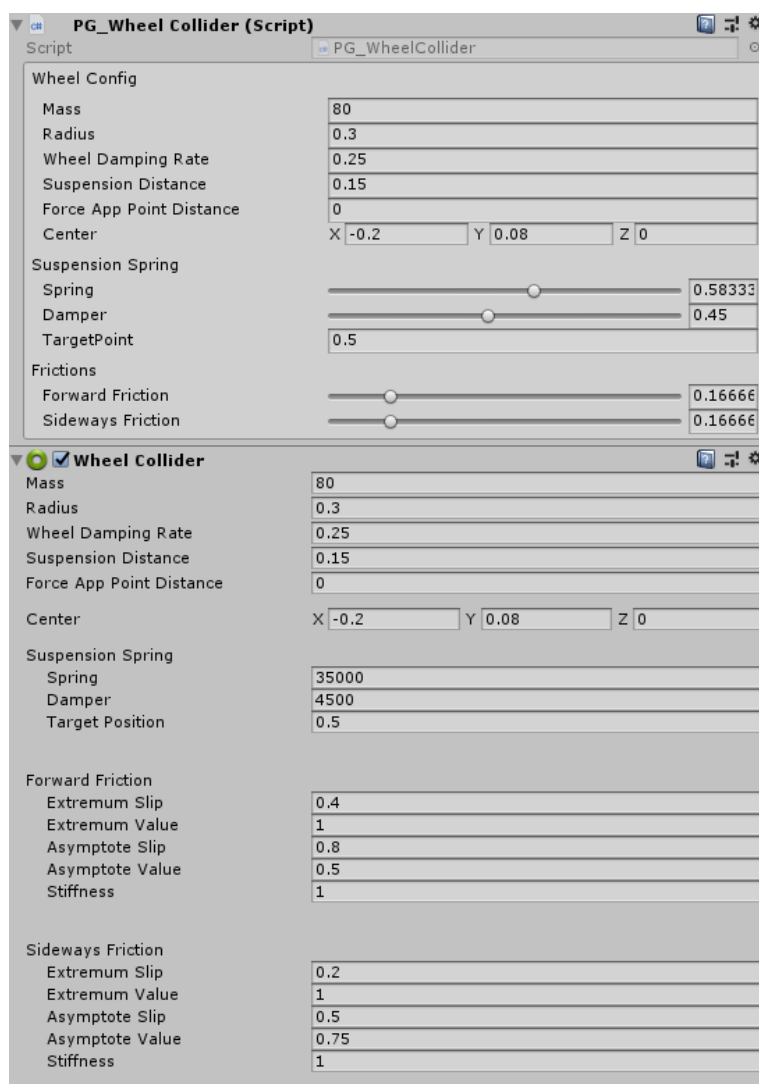
PG_WheelCollider.cs

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. See the section on “Regime Settings” for more details.



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.

Wheel Config

Mass	80		
Radius	0.3		
Wheel Damping Rate	0.25		
Suspension Distance	0.15		
Force App Point Distance	0		
Center	X -0.2	Y 0.08	Z 0
Suspension Spring			
Spring	<div></div>		0.58333
Damper	<div></div>		0.45
TargetPoint	0.5		
Frictions			
Forward Friction	<div></div>		0.16666
Sideways Friction	<div></div>		0.16666

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.

▼ Front Wheels Config

Is Full Config ☐

Frictions

Forward Friction	<div></div>	0.6
Sideways Friction	<div></div>	0.8

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	<div></div>		0
Damper	<div></div>		0
TargetPoint	0		
Frictions			
Forward Friction	<div></div>		0.6
Sideways Friction	<div></div>		0.8

Regime Settings

Now the game has two modes: Standard Racing and Drift. The drift has been greatly changed compared to the previous version of the asset (the machine behaves more dynamically). For those who liked the old version of the drift, I added a config with the settings "DriftRegimeSettings (OldVersion)" you need to select it in the track assets in the "RegimeSettings" field.

Script	TrackPreset
Track Name	Port
Track Icon	Port
Scene Name	PortForDrift
Game Controller	GameController_Port_Drift (GameController)
Laps Count	4
A Is Count	3
Regime Settings	DriftRegimeSettings(OldVersion) (DriftRegimeSettin
Money For First Place	1000

RegimeSettings fields.

The screenshot shows a window titled "RaceRegimeSettings" with a toolbar containing icons for a file, a list, and settings, along with an "Open" button. The main area is divided into several sections:

- Script:** A dropdown menu showing "RegimeSettings".
- Regime Scene Name:** A text field containing "RaceRegimeScene".
- AI Config:** A section header.
- Info:**
 - Regime Caption:** A text field containing "Race".
 - Regime Image:** A dropdown menu showing "RaceRegime".
- Main settings:**
 - Enable Steer Angle Multiplayer:** A checked checkbox.
 - Min Steer Angle Multiplayer:** A text field containing "0.05".
 - Max Steer Angle Multiplayer:** A text field containing "1".
 - Max Speed For Min Angle Multiplayer:** A text field containing "250".
 - Steer Angle Change Speed:** A text field containing "180".
 - Min Speed For Steer Help:** A text field containing "20".
 - Help Steer Power:** A slider control with a value of "0.5".
 - Opposite Velocity Help Power:** A text field containing "0.2".
 - Positive Angular Velocity Help Power:** A text field containing "0".
 - Max Angular Velocity Help Angle:** A text field containing "45".
 - Angular Velocity In Max Angle:** A text field containing "0.3".
 - Angular Velocity In Min Angle:** A text field containing "1.5".
 - Hand Brake Forward Stiffness:** A text field containing "0.5".
 - Hand Brake Sideways Stiffness:** A text field containing "0.2".
- Front Wheels Config:**
 - Is Full Config:** An unchecked checkbox.
 - Frictions:**
 - Forward Friction:** A slider control with a value of "0.6".
 - Sideways Friction:** A slider control with a value of "0.8".
- Rear Wheels Config:**
 - Is Full Config:** An unchecked checkbox.
 - Frictions:**
 - Forward Friction:** A slider control with a value of "1".
 - Sideways Friction:** A slider control with a value of "0.8".

RegimeSceneName: The name of the scene for this regime.

AIConfig: AI configuration. For both regimes, one AI behavior with different parameters is used.

RegimeCaption, RegimeIcon: The name and icon of the regime to display in and in the UI.

The following settings are also in CarController, for use separately from the regimes.

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).

HandBrakeForwardStiffness: To change the friction of the rear wheels with a hand brake.

HandBrakeSidewaysStiffness: To change the friction of the rear wheels with a hand brake.

FrontWheelsConfig and RearWheelsConfig: The friction settings for the front and rear wheels are unique for each regime.

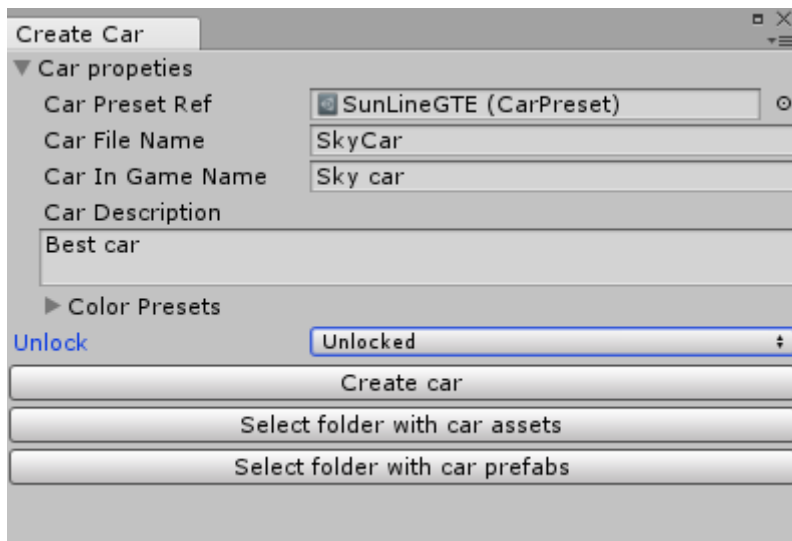
DriftRegimeSettings inherited from RegimeSettings has additional settings "Drift settings" they are needed to balance the calculation of drift score.

Drift settings	
Wait Drift Time	0.3
Wait End Drift Time	2
Min Angle	20
Max Angle	90
Min Speed	40
Score Per Meter	20
Max Multiplier	9
Min Score For Inc Multiplier	1000
Money For Drift Multiplier	0.004

How to create car

Auto add method ([Video instruction](#)):

The car can be easily added using the window "Window/Arcade Car Controller (ACC)/Create car"



CarPresetRef: Asset of the car for copying prefab. It's easier to change an existing prefab than create a new one.

CarFileName: All car files will be created with this name.

CarInGameName: The name of the car that will be displayed in the game (Menu).

Unlock: A way to unlock a car.

After setting all the parameters, you need to click the "Create car" button, after which all the necessary files will be created. After creating the car, you can edit the prefab (Change the view, change the configuration).

"Select folder ..." buttons are needed to search for folders in which files are created.

All default settings and file saving paths are in the CreateHelperSettings asset.

Manual way to add ([Video instruction](#)):

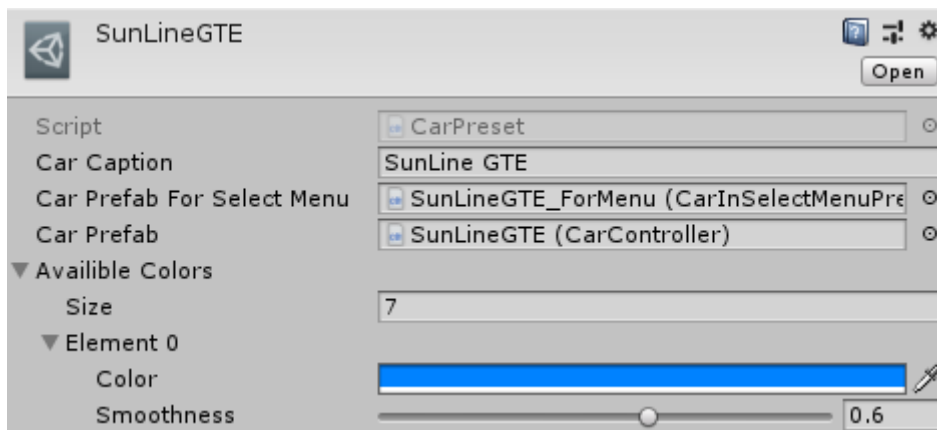
In order for the car to be selected in the selection menu you need to create:

The prefab of the car for the game (the process is specified in the CarSettings section).

For menu you can use the main prefab with the car, or you can create a separate prefab (More detailed) separately for the menu with the CarInSelectMenuPrefab component (To select a color).

Asset CarPreset with links to prefabs, a description of the car and the available colors.

CarPreset have unlock logic, field "Lock Type".



!!!Attention!!! Color selection is only available for cars with a AlphaColorMask shader.

For convenience, you can duplicate the existing prefabs and assets.

The link to the created asset should be placed in the list GameController-CarsForThisRegime.

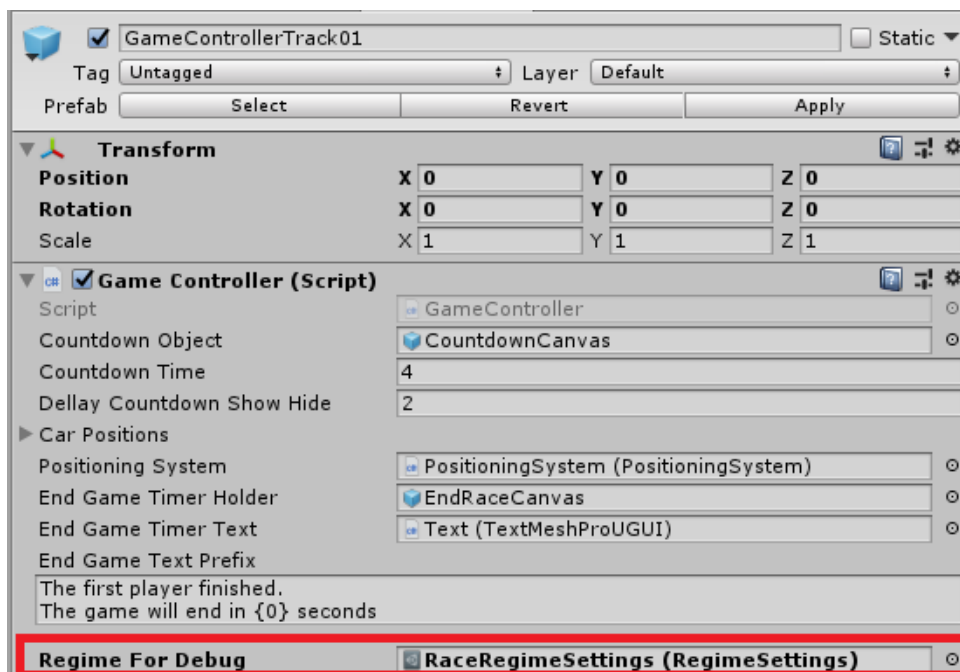
To use cars with CarController separately from everything, you need to add the prefab of the car to the scene, add the UserControl component to this prefab, add the AudioListiner to the car to listen to the sounds, add a camera to the scene and specify a link to the managed car in CameraController.OverrideCar.

How to debug

To quickly launch a track scene (in debug mode) with a player's car you need Place the car prefab with the "CarController" component on the scene in the right place for you, then start the scene.

To check the AI, you need to add the "DriftAIControl" component to the car on the scene.

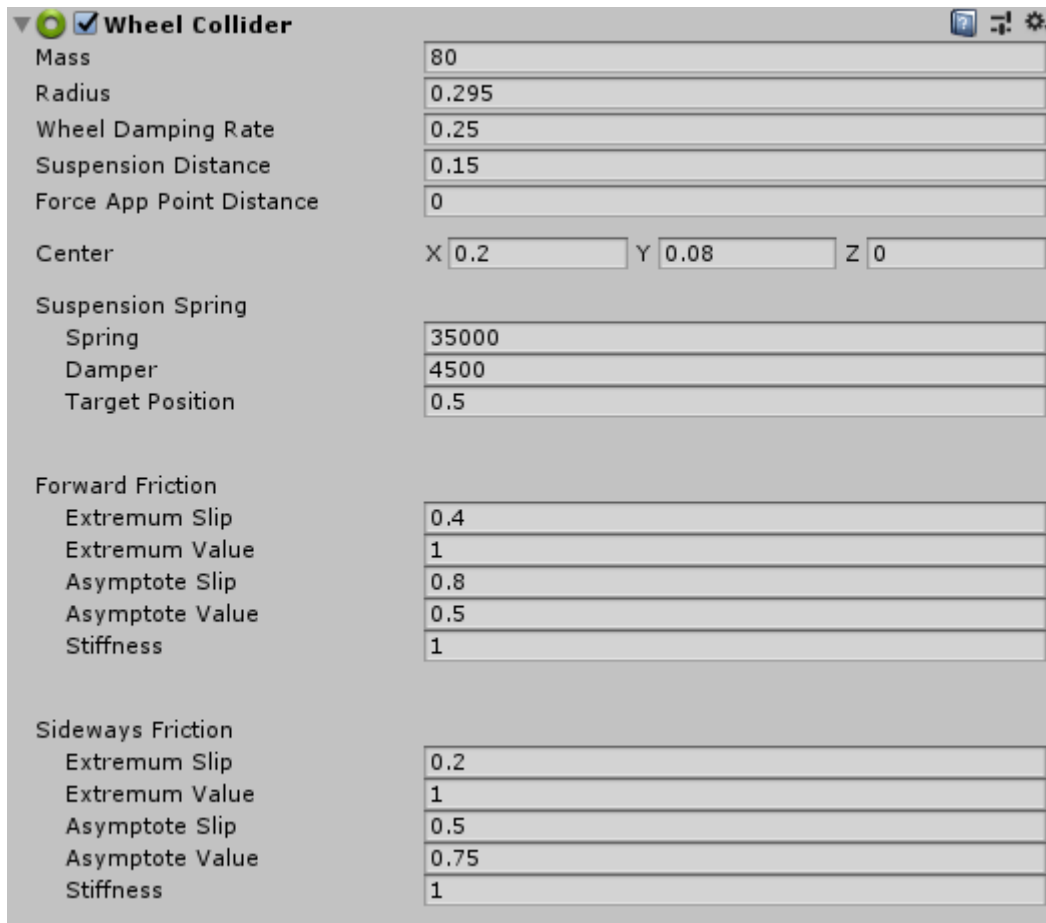
The regime for debug is selected in GameController / RegimeForDebug.



Car settings

The physics of the car is based on standard WheelColliders.

WheelColliders config:



The screenshot shows the 'Wheel Collider' configuration window. It has a left sidebar with a tree view containing 'Wheel Collider' and a main area with various settings. The settings are organized into sections: Mass, Radius, Wheel Damping Rate, Suspension Distance, Force App Point Distance, Center (X, Y, Z), Suspension Spring (Spring, Damper, Target Position), Forward Friction (Extremum Slip, Extremum Value, Asymptote Slip, Asymptote Value, Stiffness), and Sideways Friction (Extremum Slip, Extremum Value, Asymptote Slip, Asymptote Value, Stiffness). Each setting has a corresponding input field with a numerical value.

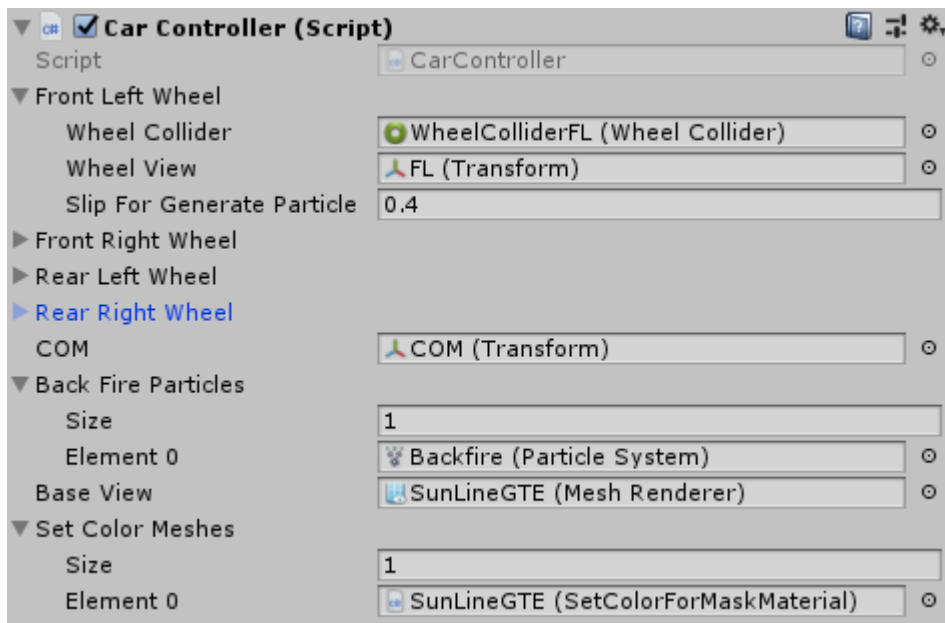
Property	Value
Mass	80
Radius	0.295
Wheel Damping Rate	0.25
Suspension Distance	0.15
Force App Point Distance	0
Center X	0.2
Center Y	0.08
Center Z	0
Suspension Spring	
Spring	35000
Damper	4500
Target Position	0.5
Forward Friction	
Extremum Slip	0.4
Extremum Value	1
Asymptote Slip	0.8
Asymptote Value	0.5
Stiffness	1
Sideways Friction	
Extremum Slip	0.2
Extremum Value	1
Asymptote Slip	0.5
Asymptote Value	0.75
Stiffness	1

Wheel collider settings can be duplicated from existing cars, friction settings are loaded at the start of the level from `WorldLoading.RegimeSettings.FrontWheelsConfig` and `RearWheelsConfig`.

`CarController.cs`: Main gameplay class.

All car settings can be used in any genre (Not only drift).

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: Center of mass.

BackFire: BackFire emitters.

SetColorMeshes: All painted parts in selected color (Only for AlphaColorMask shader).

▼ Car Config

Steer Settings

Max Steer Angle

40

Engine and power settings

Drive Type

RWD

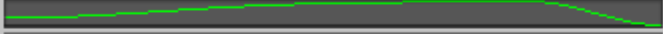
Automatic Gear Box

☒

Max Motor Torque

400

Motor Torque From Rpm Curve



Max RPM

7000

Min RPM

700

Cut Off RPM

6800

Cut Off Offset RPM

500

Rpm To Next Gear

6500

Rpm To Prev Gear

3500

Max Forward Slip To Block Change

0.2

Rpm Engine To Rpm Wheels Lerp

20

▼ Gears Ratio

Size

5

Element 0

3.59

Element 1

2.02

Element 2

1.38

Element 3

1

Element 4

0.87

Main Ratio

4.3

Revers Gear Ratio

4

Braking settings

Max Brake Torque

1000

Target Speed If Braking Ground

20

Braking Speed One Wheel Time

1

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.

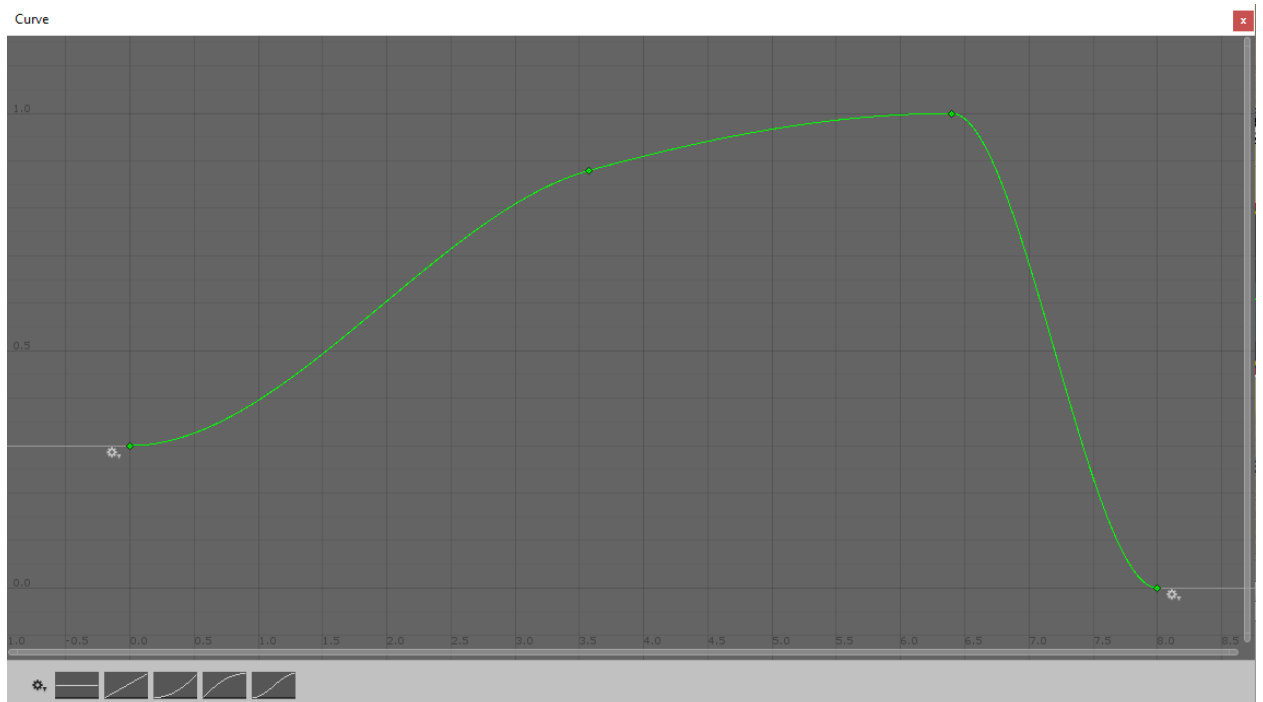
AutomaticGearBox: TODO It is necessary to implement a manual gear shift, now only the automatic gearbox works.

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.

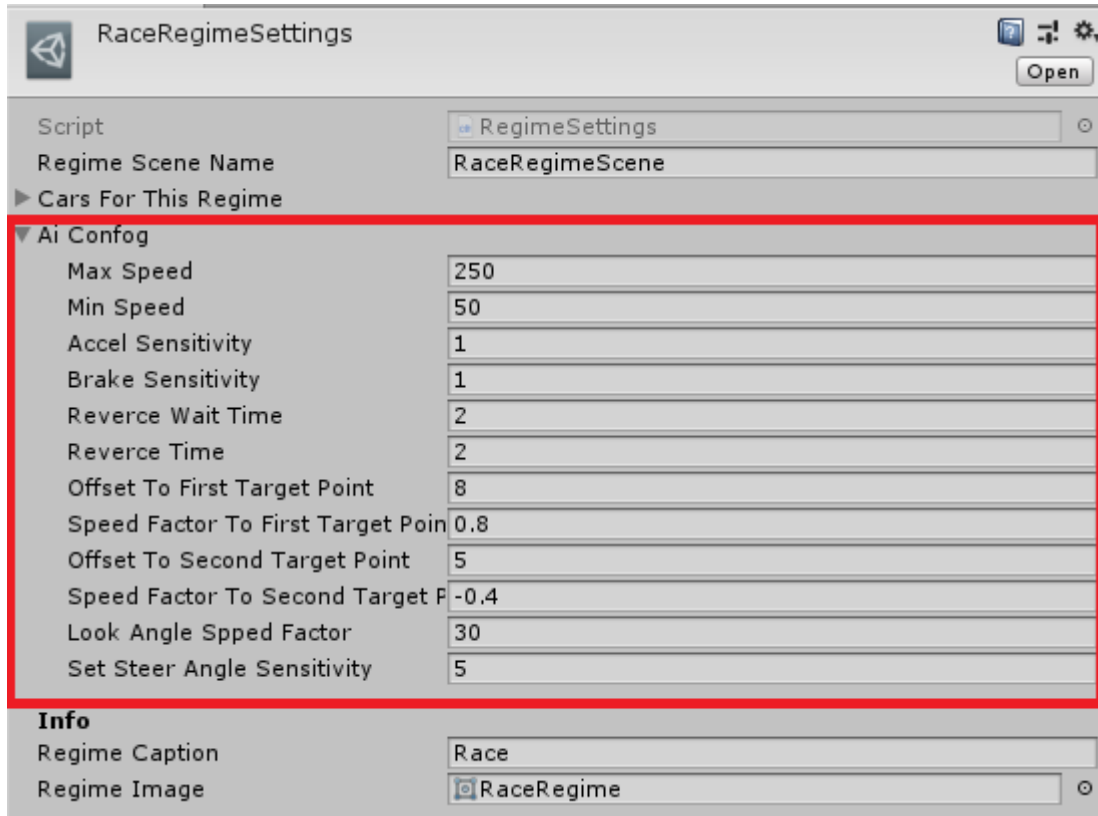
TargetSpeedIfBrakingGround: The car stops to this speed when driving on the braking ground.

BrakingSpeedOneWheelTime: Braking speed with one wheel on the braking ground.

To use cars with CarController separately from everything, you need to add the prefab of the car to the scene, add the UserControl component to this prefab, add the AudioListener to the car to listen to the sounds, add a camera to the scene and specify a link to the managed car in CameraController.OverrideCar.

AI Control

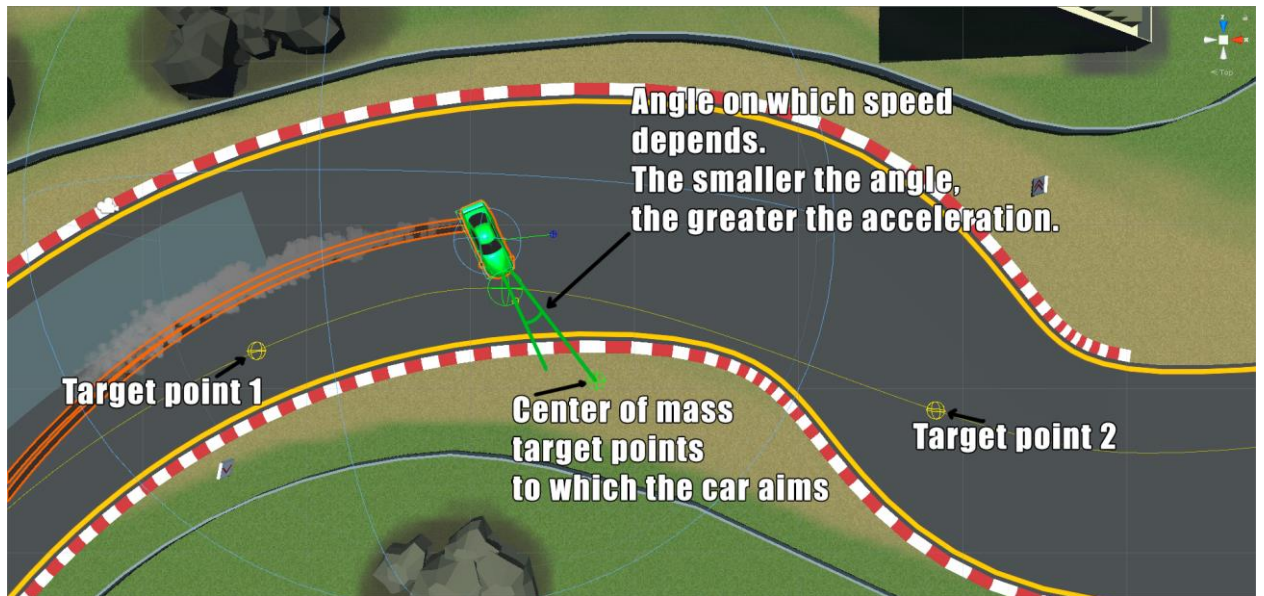
RegimeSettings



The screenshot shows a software window titled "RaceRegimeSettings". It contains several configuration fields. The "Script" field is set to "RegimeSettings". The "Regime Scene Name" field is set to "RaceRegimeScene". Below these, there is a section for "Cars For This Regime" which is expanded to show "AI Confog" settings. These settings include "Max Speed" (250), "Min Speed" (50), "Accel Sensitivity" (1), "Brake Sensitivity" (1), "Reverce Wait Time" (2), "Reverce Time" (2), "Offset To First Target Point" (8), "Speed Factor To First Target Poin" (0.8), "Offset To Second Target Point" (5), "Speed Factor To Second Target P" (-0.4), "Look Angle Spped Factor" (30), and "Set Steer Angle Sensitivity" (5). At the bottom, there is an "Info" section with "Regime Caption" set to "Race" and "Regime Image" set to "RaceRegime".

Field	Value
Script	RegimeSettings
Regime Scene Name	RaceRegimeScene
Cars For This Regime	
AI Confog	
Max Speed	250
Min Speed	50
Accel Sensitivity	1
Brake Sensitivity	1
Reverce Wait Time	2
Reverce Time	2
Offset To First Target Point	8
Speed Factor To First Target Poin	0.8
Offset To Second Target Point	5
Speed Factor To Second Target P	-0.4
Look Angle Spped Factor	30
Set Steer Angle Sensitivity	5
Info	
Regime Caption	Race
Regime Image	RaceRegime

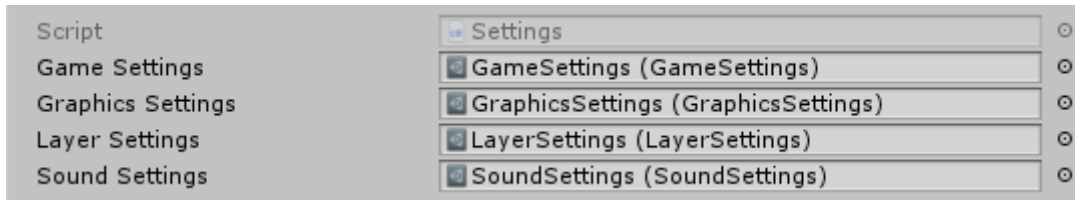
The bots are configured in the "AI Config". For all cars one config is suitable. For each regime, you can configure unique AI behavior.



Bot always aim to the center of mass. Acceleration is calculated based on the angle to the center of mass.

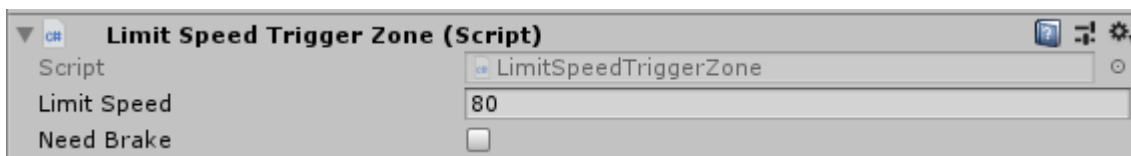
Balance settings

The project uses a convenient balance system. The desired value can be accessed from any part of the code.



Settings: This asset is loaded from resources. It contains links to other assets of settings. For what the fields in assets are commented out in the code. The hierarchy of assets has already been met earlier, in the section "Project flow chart".

It is also possible to limit the speed for AI, using the LimitSpeedTriggerZone component, when hit on a trigger with this component, the AI will begin to slow down to the speed specified in the setting. Restriction zones are in the PortForRace scene object SpeedLimitsForAI.

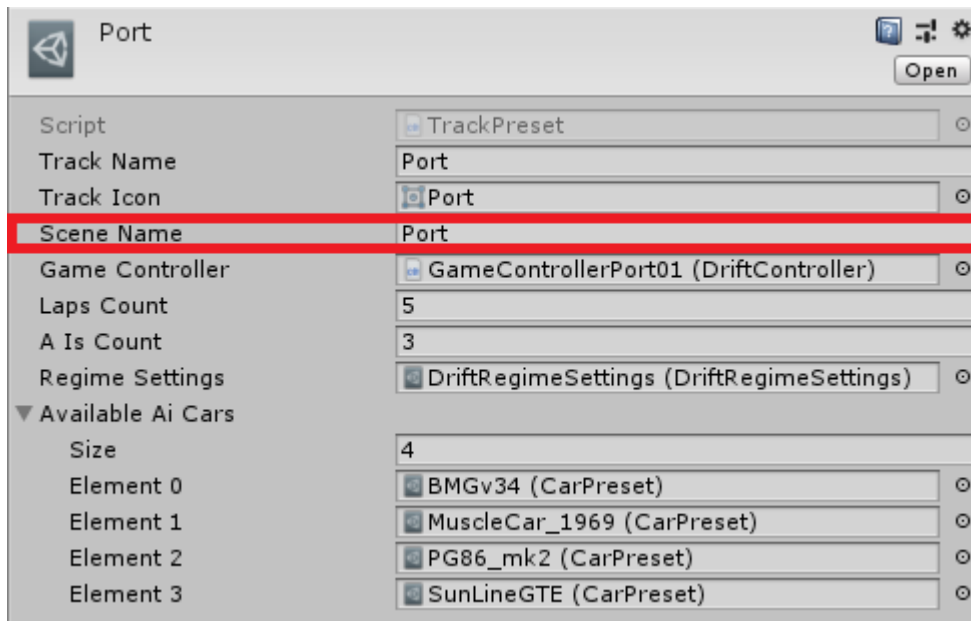


NeedBrake: When the flag is set, AI will slow down with braking in this zone to the desired speed.

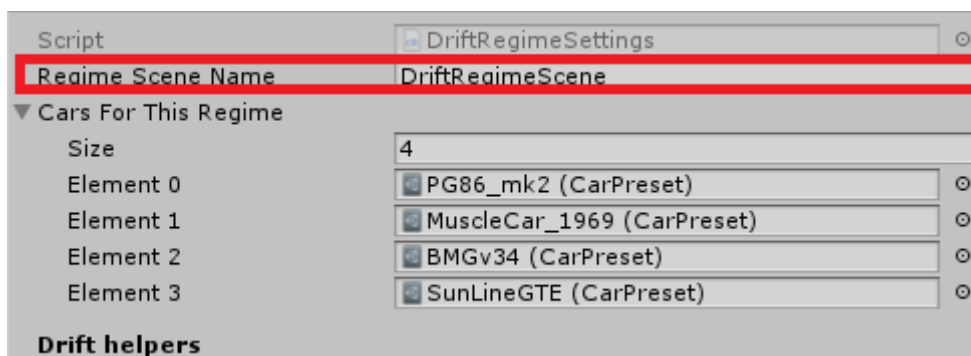
Split level loading

The level is loaded from 2 scenes. The 1st scene of the level, it contains objects of level geometry, a game controller, and an effect controller that are unique for the level. The 2nd scene for the regime, on it is the UI.

The name of the scene level is indicated in the level asset:



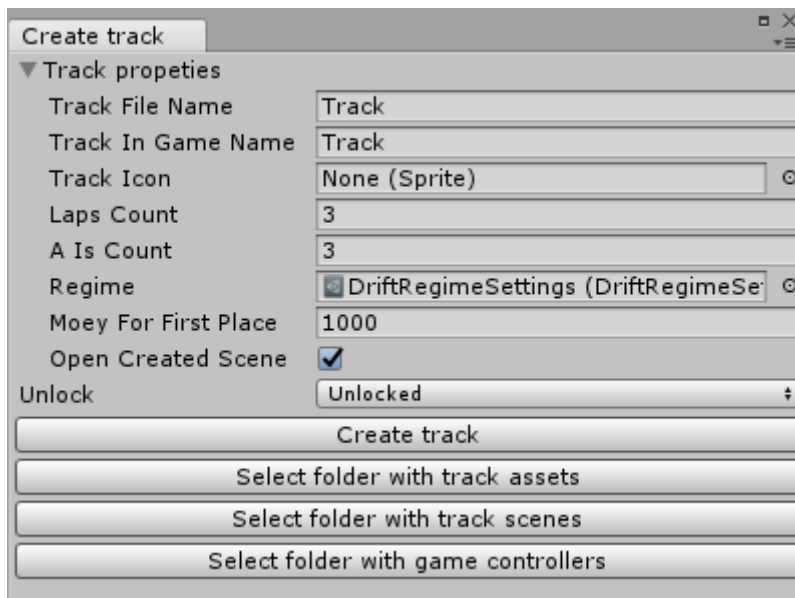
The name of the scene regime is indicated in the RegimeSettings asset.



How to create a track

Auto add method ([Video instruction](#)):

The track can be easily added using the window "Window/Arcade Car Controller (ACC)/Create track"



TrackFileName: All track files will be created with this name.

TrackInGameName: The name of the track that will be displayed in the game (Menu etc.).

Unlock: A way to unlock a track.

After setting all the parameters, you need to click the "Create track" button, after which all the necessary files will be created. After creating the track, you can edit the created scene.

"Select folder ..." buttons are needed to search for folders in which files are created.

All default settings and file saving paths are in the CreateHelperSettings asset.

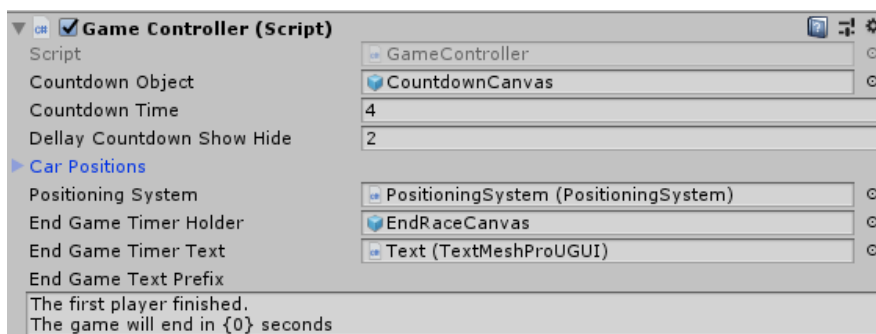
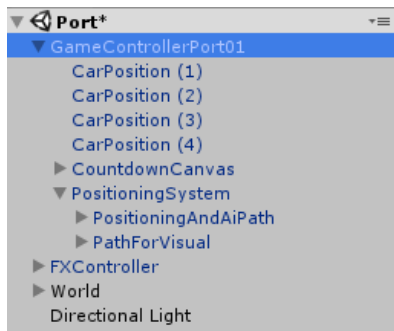
Manual way to add:

Level scene.

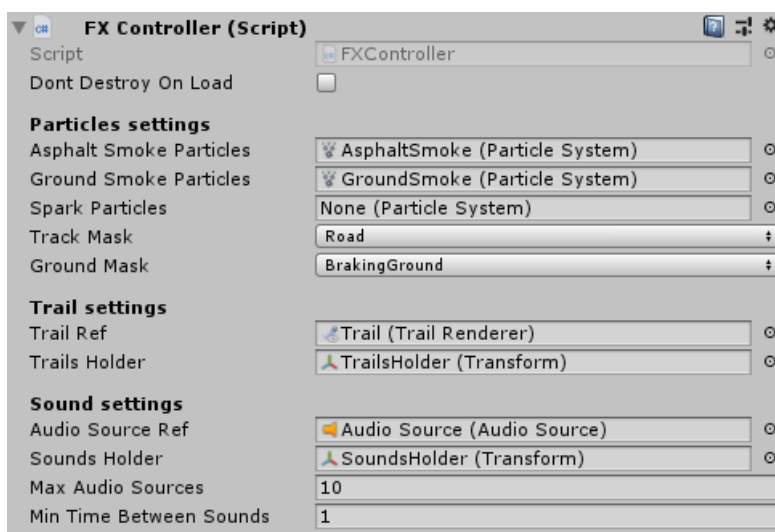
On the scene level must be present:

Level geometry with colliders.

GameController, with start car positions, positioning system, countdown and EndGameTimer (To complete the multiplayer game).

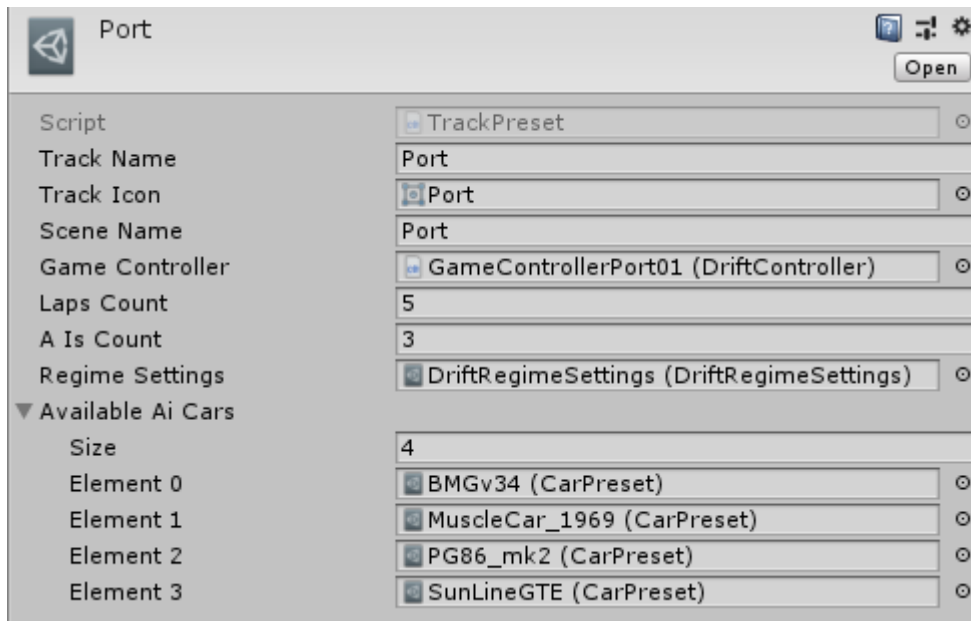


FXController, is now used for all levels one prefab, if necessary, it can be changed.



Track preset.

This asset contains all the necessary configuration for loading the level.



Track Name: The name of the track displayed in the track selection menu.

Track Icon: The icon of the track displayed in the track selection menu.

Scene Name: Loaded Scene name.

GameController: Link to prefab GameController track.

Regime Settings: Link to the regime to load an extra scene.

Available Ai Cars: Cars available for AI of this track.

TrackPreset have unlock logic, field "Lock Type".

Assets are created from the context menu "Create / GameBalance / Game / TrackPreset". Or you can duplicate the existing one.

After creating a link to an asset you need to put in the list GameSettings / Tracks.

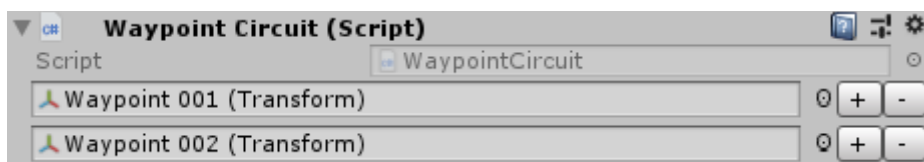
Positioning System

Positioning system To cancel scoring in the wrong direction and to control circles. PathForVisual is needed to display the track map in the track selection menu, later it will still be used for the mini map.

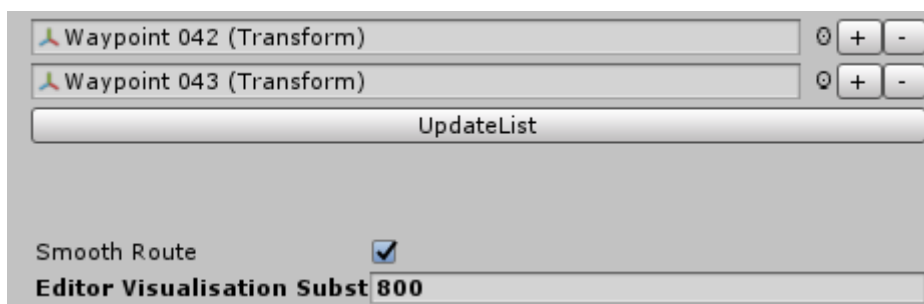
Positioning system settings.



Waypoints



....



WaypointCircuit is the way to go. To add items, click the "+" button or add an empty child object and click the button "UpdateList".

WaypointCircuit.cs taken and modified from "Standard Assets". Now works only for the circle. After a few updates will be the ability to select a segment with a start and finish.

At the first level premiere, the path is yellow. The path must lie on the optimal trajectory.



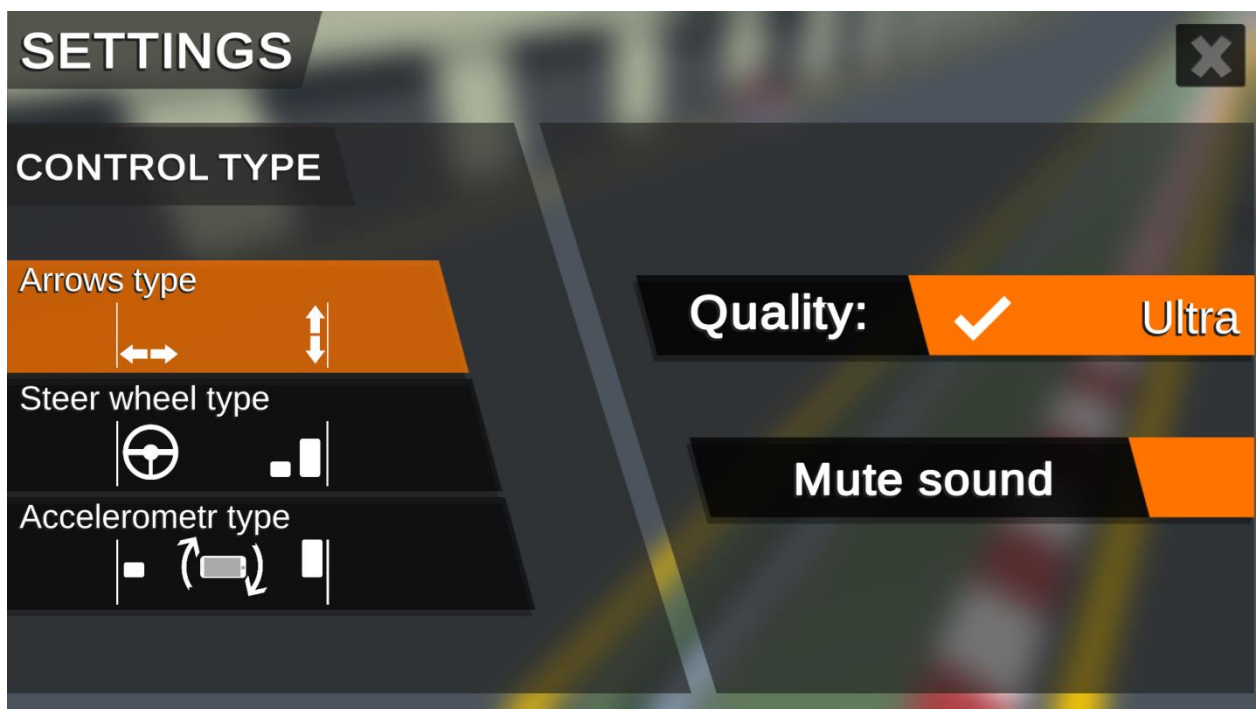
!!!Attention!!! The first and last points must have the same coordinates at the finish point in order to correctly count the laps.

Main menu and game menu.

The size and functionality of all elements is designed for a mobile platform.

All windows in the game have animations and have a standard animator.

Settings window. In main menu and in game menu.



In the left part of the settings, you can select the type of control. The selected type is not available for re-selection. Arrows is selected by default. The right side is settings graphics quality and mute the sound.

Select car window:



At the bottom of the parameters of the car. The maximum value is selected among all available cars.

End game window for Drift regime.



End game window for Race regime.

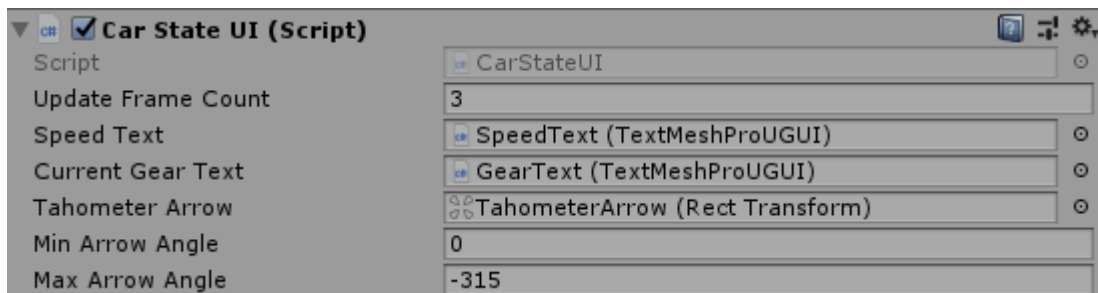


The left side indicates the player's results and the cash prize for the race. On the right side is a table of players with points or race time. The window for ending the game for each regime is its own, located on the scene specified in the regime settings.

In race UI



Speedometer, tachometer, current gear. Used for all modes, located in prefab CarStateCanvas, CarStateUI component.



Drift indicator. The left side displays the scored points and the bar start / gain points. The current multiplier and the progress bar to the next multiplier are displayed on the right side (Maximum multiplier is 9, configured in DriftRegimeSettings). The indicator is reflected only in drift regime.



Race regime statistics:



A screenshot of a game interface showing race statistics. The background is a 3D-rendered road with a yellow center line and a red-and-white dashed edge line, curving to the right. A green tree is on the left side of the road. The statistics are displayed in a semi-transparent dark grey box with white text.

Time:	01:08:73	1	Mason(b)
Current lap:	00:16:20	2	Player 82382
Best lap:	00:52:50	3	Emily(b)
Lap:	2/3	4	Michael(b)

Drift regime statistics



A screenshot of a game interface showing drift statistics. The background is a 3D-rendered green field with a single green tree. The statistics are displayed in a semi-transparent dark green box with white text.

Time:	01:59:28	1:Michael(b)	33704
Total:	32573	2:Player 82382	32573
Best:	15228	3:William(b)	0
Lap:	2/3	4:Sofia(b)	0

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.

Shader: AlphaColorMaskSmoothnessFromMetalic

It has all the same fields, except for the field of Smoothness, Smoothness is taken from the map Metalic (RGBA).

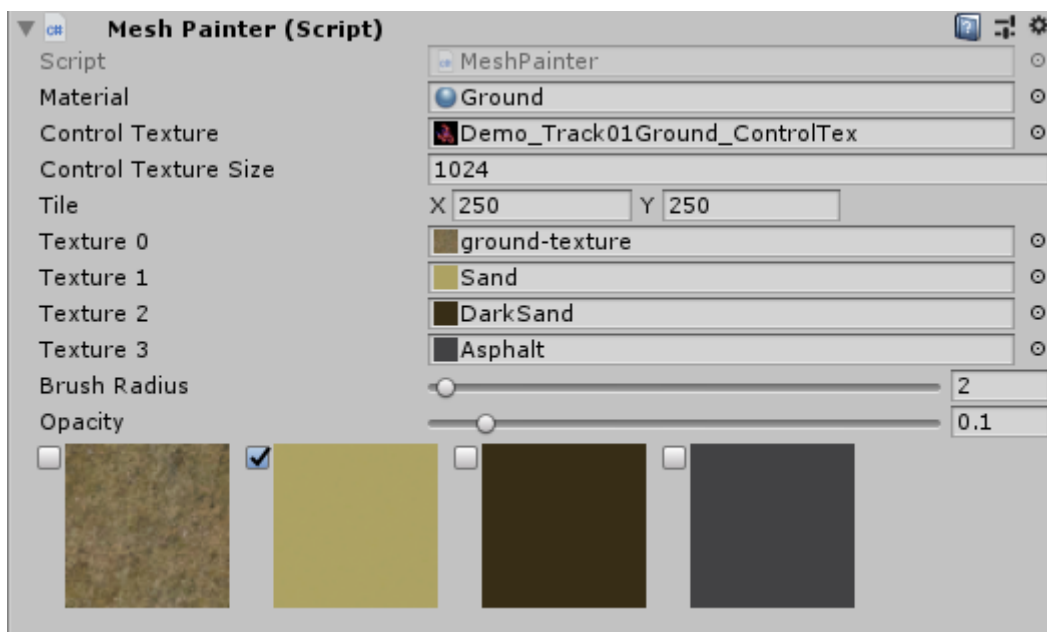
SetInstancingMaterialFields.cs:

Sets the parameters of color and Smoothness at start. Without affecting the base material.

Mesh Painter

Additional script .

In this project, I needed the functionality of the terrain, but its use is too bad for the performance on weak devices. I decided to make a version with the ability to draw on the mesh with the desired texture. The MeshPainter script is used only in the editor to change the control texture, only the shader is used in the game.



Material: For each object of type MeshPainter need your own material of type Custom/MultiTextures.

Control Texture: It is created automatically if there is no reference to an existing texture in the field. It has 4 channels (A, R, G, B). Each channel is responsible for its texture from the list below.

Tile: Tile all textures from the list below.

Texture0,1,2,3: The textures needed for drawing. By default, the entire mesh is colored with the first texture.

Brush Radius and Opacity: Settings of brush.

Toggles with textures: To select the current texture as a drawing. When you select a texture, the drawing mode in the scene is turned on, now you can draw only on the selected object, other objects from the scene view

cannot be selected, only in the hierarchy. When changing the texture or selecting another object, the texture is preserved.

Update history.

Update 1.0.1

Changed the main car to a more detailed model.

Fixed bugs when importing asset into higher versions of unity.

Update 1.1.0

Added main menu and game menu.

Added 3 new cars.

Changed WheelColliders balance. Now the gameplay has become much more dynamic.

Slightly changed the structure of the project, preparation for the introduction of AI.

Added save/load settings and player results.

Improved imitation of gearbox (now it is not twitching and does not jump through the gear).

Fixed minor bugs.

Update 1.2.0

Added AI control for drift regime.

Added the ability to select colors for cars.

Redesigned user interface.

Fully redone positioning system.

Small fixes in the style of the code, for more convenient reading.

Minor bugs fixed.

Update 1.2.1

Improved control (tilt control) for mobile devices.

Update 1.2.2

Added backfire effect.

Update 1.2.3

Improved control (Steer wheel control) for mobile devices.

Update 1.2.5

Physics and graphics optimization of for mobile devices.

Update 1.3.0

Added new track (Port).

Added select menu for tracks.

Slightly improved imitation gearbox.

Changed loading logic level.

Fixed small bugs.

Update 1.3.1

Added "Enable AI" option in the game settings.

Update 1.3.2

Added in-game store and content unlock logic.

Minor bugs fixed.

Update 1.5.0

Added a new regime "Standard Racing".

A convenient setting for the wheel collider has been added.

Added support for the Xbox controller.

Change in drift regime: Now the cars have more dynamic behavior (The old behavior settings are in the asset, the documentation describes how to use them).

Improved AI behavior. One AI is used for all modes with different settings.

Improved positioning system.

Bugs fixed.

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