Realistic Engine Sounds - Lite

By: Yugel Mobile

Thank you for purchasing Realistic Engine Sounds Lite!

Compatible with Unity Free and Pro version too

+ Mobile Support!

Contents

Package includes:

- 6 .unitypackages for car controllers (containing prefabs and demo scenes):

  - Edy's Vehicle Physics
  - Realistic Car Controller
  - Randomation Vehicle Physics
  - Unity Standard Assets Car
  - NWH Vehicle Physics
  - Motorbike Starter Kit

- 6 ready to use engine sound packages for exterior and interior camera views with engine startup sounds:

- i4 German
- i6 German
- i6 German FREE (available for free in the Asset Store)
- Rotary x8 FREE (available for free in the Asset Store)
- V8 American – Classic
- V8 Italian F355

There are 24 prefabs for (almost) each car controller + 26 clean engine prefabs + 15 universal prefabs for Turbo, Supercharger, Gear changing sound and Muffler crackle noise packs.

- In total there are 360 prefabs (Stock, EVP, RCC, RVP, NWH, MSK and Unity Standard Assets Car prefabs)

- Two Wav audio files for reverse gear (all engine sound prefabs use the same reverse gear Wav audio file)

- 12 controller scripts for prefabs + scripts for demo scenes

- 14 demo scenes:

  - Scene for testing Realistic Engine Sounds in an user friendly UI environment
  - Scene for testing Realistic Engine Sounds Mobile in a user friendly UI environment
  - Unity Standard Assets car with Realistic Engine Sounds
  - Unity Standard Assets car with Realistic Engine Sounds Mobile
  - Realistic Car Controller with Realistic Engine Sounds
  - Realistic Car Controller with Realistic Engine Sounds Mobile
  - Edy's Vehicle Physics with Realistic Engine Sounds
  - Edy's Vehicle Physics with Realistic Engine Sounds Mobile
  - VehicleManager_EVP5_RES
  - Randomation Vehicle Physics with Realistic Engine Sounds
  - Randomation Vehicle Physics with Realistic Engine Sounds Mobile
  - NWH_RES
  - NWH_RES_mob
  - All_engine_sounds (MSK)
  - All_mobile_engine_sounds (MSK)

  Each scene has it's own engine sound prefabs.

There are total 118 Wav audio files in this package.

There are more than *159 prefabs in this package.

*all supported car controller's prefabs are counted in

Importing R.E.S. asset package into your project

- **Edy's Vehicle Physics:** When you're importing my asset for Edy's Vehicle Physics you must import **Edy's Vehicle Physics**, **Realistic Engine Sounds** assets and **EdysVP_RES-Lite.unitypackage** from: ***.. \RealisticEngineSound\Assets_For_EVP5*** folder.

- **Realistic Car Controller:** When you're importing my asset for Realistic Car Controller you must import **Realistic Car Controller**, **Realistic Engine Sounds** assets and **RCCV3_RES-Lite.unitypackage** from: ***.. \RealisticEngineSound\Assets_For_RCC_V3*** folder.

- **Randomation Vehicle Physics:** When you're importing my asset for Randomation Vehicle Physics you must import **Randomation Vehicle Physics, Realistic Engine Sounds** assets and **RVP2_RES-Lite.unitypackage** from: ***.. \RealisticEngineSound\Assets_For_RVP2*** folder.

- **Unity Standard Assets' Car:** When you're importing my asset for Unity Car you must import **Unity Standard Assets**, **Realistic Engine Sounds** assets and **UnityCar_RES-Lite.unitypackage** from: ***.. \RealisticEngineSound\Assets_For_Unity_Car*** folder.

- **NWH Vehicle Physics:** When you're importing my asset for NWH Vehicle Physics you must import **NWH Vehicle Physics, Realistic Engine Sounds** assets and **NWH_RES-Lite.unitypackage** from: ***.. \RealisticEngineSound\Assets_For_NWH*** folder.

- Motorbike Starter Kit: When you're importing my asset for Motorbike Starter Kit you must import **Motorbike Starter Kit, Realistic Engine Sounds** assets and **MSK_RES-Lite.unitypackage** from: ***.. \RealisticEngineSound\Assets_For_MSK*** folder.

- **Other vehicle controller:** When you're importing my asset for a project that is using another vehicle controller, just import „**Realistic Engine Sounds"**. You need to create your own script to set Realistic Engine Sound script's rpm to your vehicle controller's rpm. If you have problems doing this, feel free to contact me, I will try to help you when I have some free time.

How to update from older RES version?

Before you update Realistic Engine Sounds for new version, backup your current project!

If you upgrade from RES-Lite version, delete everything in *RealisticEngineSounds* folder except the *Prefabs* folder and import RES-Lite into your project.

If you upgrade from older RES-Lite, delete *RealisticEngineSounds* from your project, and import the updated RES version into your project.

After this step import the required *.assetpackage for your car controller.

That's all, now you have the updated version of Realistic Engine Sounds - Lite. Read the Release Notes to find out whats new in this version.

Don't forget to write a review for my asset in the Unity Asset Store. I would like to hear some feedback. Thank you!

How is this package working?

Before you start using this package you need to understand how is this package working, especially if you want to use your own Wav audio files.

There are two RES controller scripts: - RealisticEngineSounds.cs and RealisticEngineSounds_mobile.cs

RealisticEngineSounds_mobile.cs don't have deceleration sounds, it may use less audio sources.

For mobile devices I recommend to use RealisticEngineSounds.cs for Player's car and RealisticEngineSounds_mobile.cs for Opponent's car. If you want your player's car to be louder, use an Audio Mixer with RES script.

Realistic Engine Sound script uses 9 Wav audio files:

- Idle: for car's idle rpm
- Low_On: low rpm sound on – used for accelerating
- Low_Off: low rpm sound off – used for decelarating
- Med_On: medium rpm sound on, car is accelerating
- Med_Off: medium rpm sound off, car is decelerating
- High_On: high rpm sound on, car is accelerating
- High_Off: high rpm sound off, car is decelerating
- MaxRPM: car is on maximum rpm and rpm limiting is set to on (if „Use rpm limit" is turned off, audio source is not created for this sound)
- Reversing: whistling sound is played with engine sounds when car is in reverse gear (only if reverse gear is enabled in R.E.S. script)

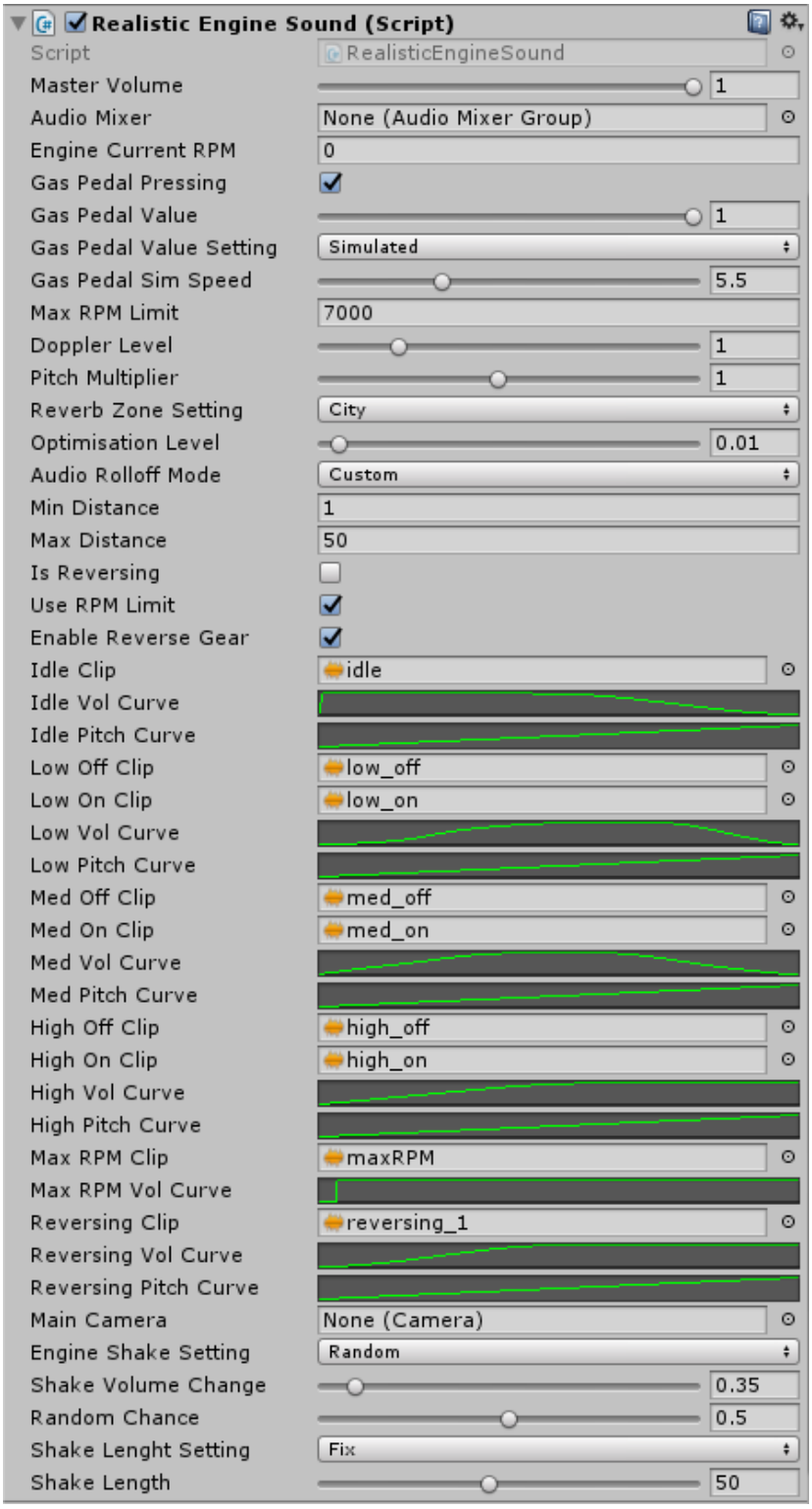
RES script can work with only one or two wav files. You don't need to use all wav file listed above. For example try using only Idle and High wav files and set other audio clips to „None".

You do not need to add Audio sources to your Game Object, the script will do it for you.

Each audio has two settings:

- Volume Curve
- Pitch Curve

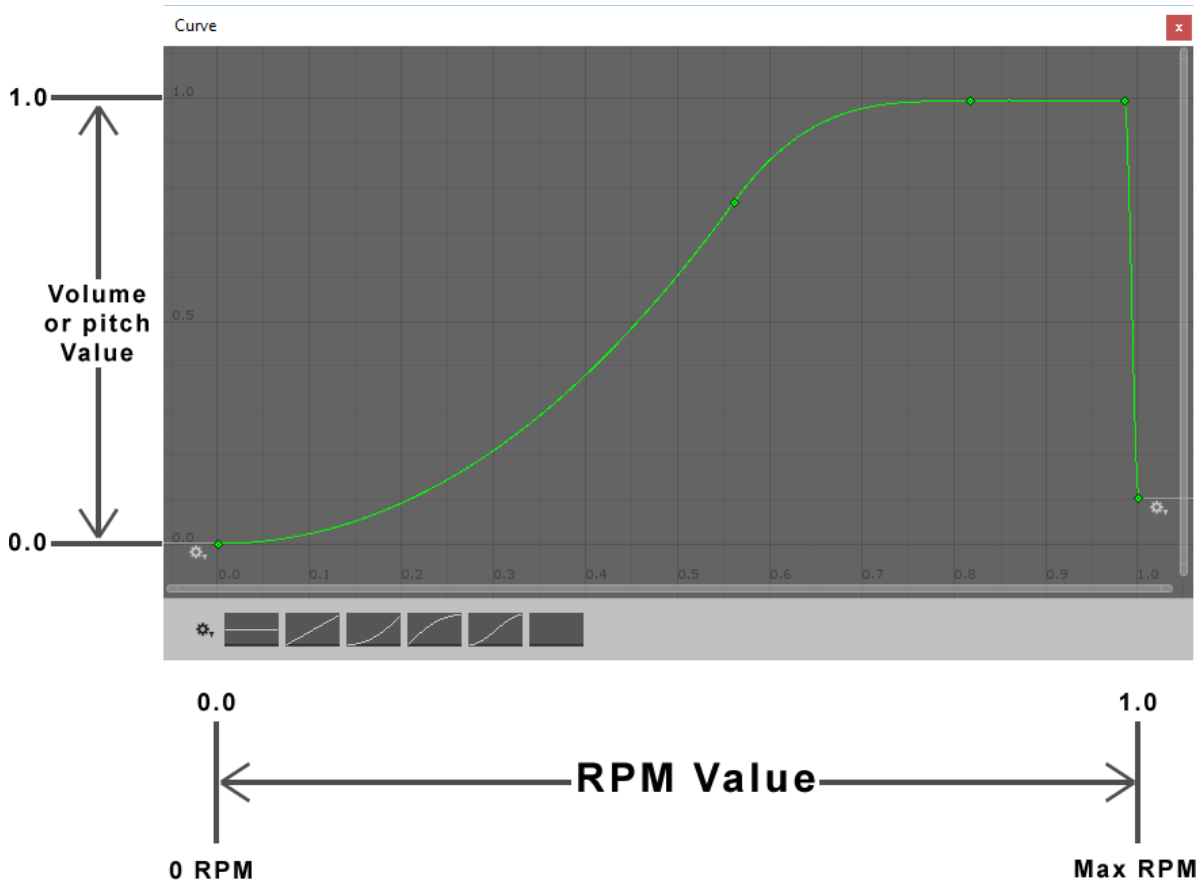
(Low_On and Low_Off use the same setting. This is valid to Med_on-Med_off and High_On-High_off too!)



- Master Volume: sets the prefabs maximum volume.
- Audio Mixer: add an audio mixer here to control all engine sounds volume at one place and add audio effects like Distortion. (can be leaved at None)
- Engine Current RPM: your car's current rpm (this needs to be set from a custom script, or use one of my controller scripts)
- Gas Pedal Pressing: if it's true it will play accelerating sounds, if it's false it will play decelerating sounds
- Gas Pedal Value: how much % is the gas pedal pressed (1 = 100%, 0,5 = 50%, 0 = 0%)

- Gas Pedal Value Setting (Simulated or Not Simulated): in *Simulated* setting sets the *Gas Pedal Value* and it will be simulated by *Gas Pedal Sim Speed* value when the car is started accelerating (this gives a smooth change between decelerating and accelerating sounds). In *Not Simulated* setting the *Gas Pedal Value* is need to be set by manually - this is good for joystick controlled games
- Gas Pedal Sim Speed: sets the simulation speed for *Gas Pedal Value* when *Gas Pedal Value Setting* is set to *Simulated.*
- Max RPM Limit: your car's max RPM (for RCC this is set automatically from RCC settings)
- Doppler Level: Determines how much doppler effect will be applied to this audio source (if is set to 0, then no effect is applied).
- Pitch Multiplier: sets the prefab's maximum pitch. With this setting you can make unique a sound with just one click!
- Reverb Zone Setting: adds a reverb zone to the game object and set it's preset.
- Optimisation Value: Audio Source with volume value below this value will be destroyed (at bigger value may be played less audio sources at a time, but engine sound may not change smoothly).
- Audio Rolloff Mode: How fast the sound fades. There are three Rolloff modes: Logarithmic, Linear and Custom Rolloff. Default is set to Custom.
- Min Distance: Within the Min distance the engine sound will cease to grow louder in volume.
- Max Distance: Max Distance is the distance a sound stops attenuating at. RES script will only create audio sources if the script is closer to Main Camera than Max Disrance. To save some performance in scenes with multiple cars, set Max Distance to a lower value.
- Is Reversing: if it's true, it will play reversing sound with engine sounds too when *Enable Reverse Gear* is enabled.
- Use RPM Limit: if it's true it will play max rpm clip if *Engine Current RPM* value = *Max RPM Limit* value.
- Enable Reverse Gear: enable reversing sound effect.
- Main Camera: this is set by automatically, but if it gives you NULL errors, you need to add your Main Camera here. Main Camera is needed for Max Distance value. If the script is farer from the Main Camera than Max Distance, it will destroy all of its audio sources, because of the distance this audio sources already can't be heard.
- Engine Shake Setting: this gives a „shake" effect when the player hits Gas. Settings: Off, Random, Allways On.
- Shake Volume Change: how much volume have the shaking effect.
- Random Chance: if Engine Shake is set to Random, this value can be set to how much chance will have the shake effect to appear.
- Shake Lenght Setting: how long is the shake effect. Settings: Random, Fix.
- Shake Lenght: if Shake Lenght Setting is set to Fix, Shake Lenght value can be edited. Bigger value gives longer shake effect.

Volume Curve and Pitch Curve settings are modified in a Curve Editor window. (see the image bellow)



From Left to right you see the car engine's rpm value.

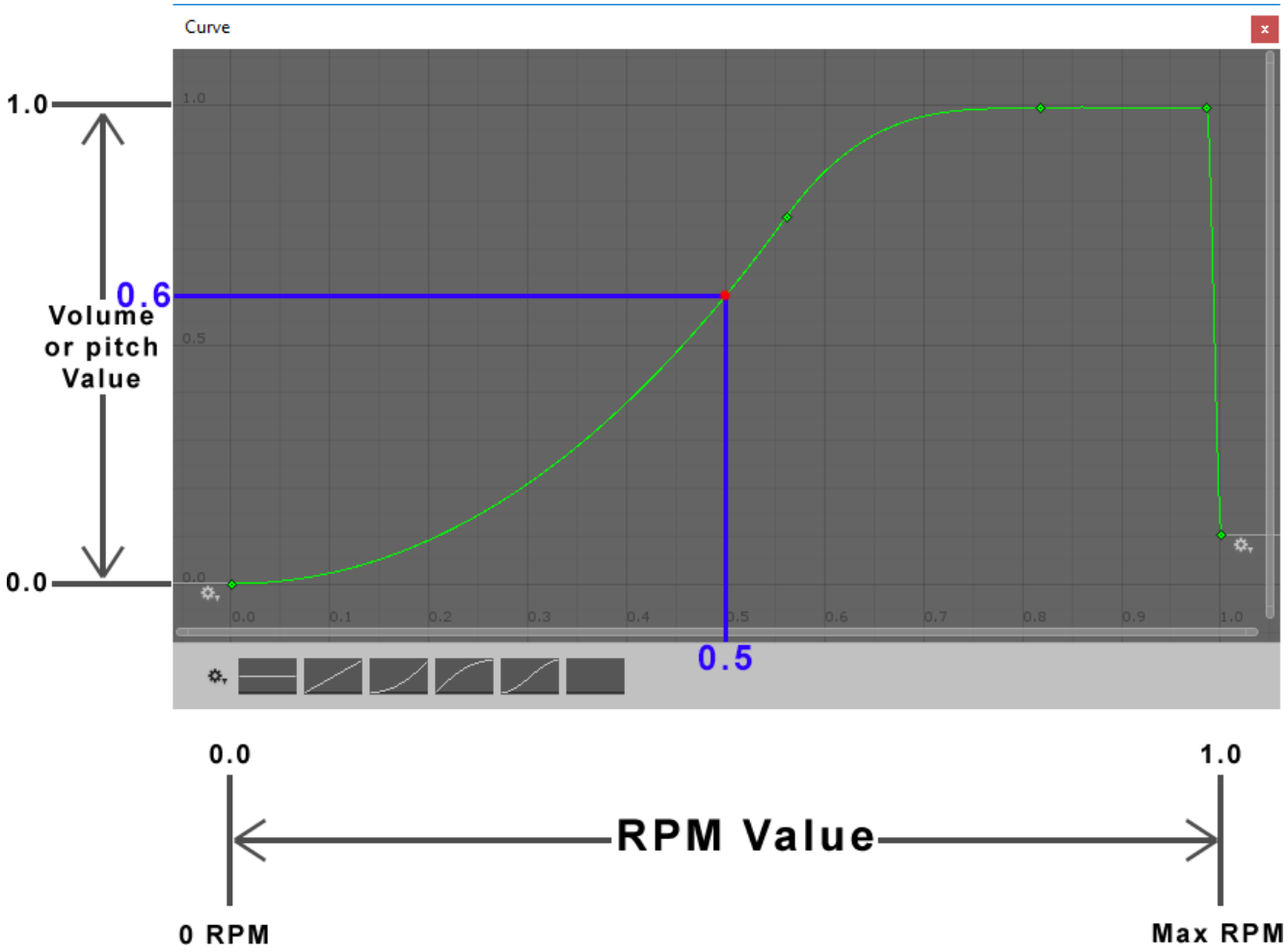
From up to down you see the volume's or pitch's value (depending on which one's setting you opened).

In this image you see the High rpm's volume setting for i4_german prefab.

These settings are set between 0 and 1 values by RPM Value.

0 value is = 0 rpm, 1 value is = Max RPM Limit, RPM value is calculated by Engine Current rpm.

For example: If your car's *Engine Current rpm* is 3500 and *Max rpm Limit* is 7000, your RPM value will be 0.5 (See the image bellow).



In this image you see High_On audio file's volume settings.

If rpm value = 0.5 volume value will be ~ 0.6

Setup for Unity Car

Youtube video url for this tutorial: https://www.youtube.com/watch?v=0eDsfkSUU54

Import Unity Standard Assets and Realistic Engine Sounds.
Now import **UnityCar_RES-Lite.unitypackage** from the following folder:
.. \RealisticEngineSound\Assets_For_Unity_Car\..

This unitypackage gives you prefabs and demo scenes for Standard Assets car.

Open Unity Standard Assets_Car Sample Scene from the following folder:

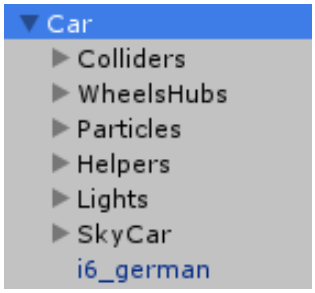
../SampleScenes/Scenes/..

Drag and drop one of my prefab for Unity Car on your car's game object. You can find prefabs for Unity Standard Assets car in the following folder:

../RealisticEngineSound/Assets/Prefabs/ Unity_Car_prefabs

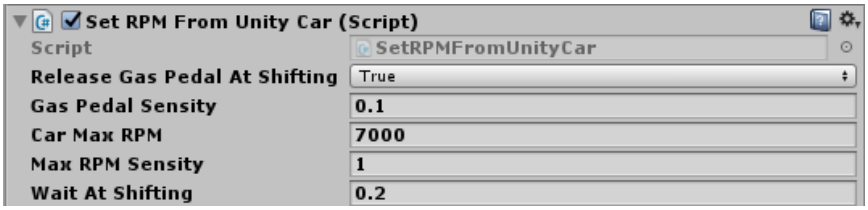
In this example I used i6_german prefab on my car game object.

Standard Assets car gameobject should look like this after you dropped one of my prefab on it :



That's all, now your Unity Standard Assets car have a cool engine sound, it's time to test it! Press Play button! ☺

Unity Car to RES controller script settings:



- **Release Gas Pedal At Shifting(True or False):** if set to true it will release gas pedal at shifting, and it will play off sounds for more realistinc engine sounds for a short time (Wait At Shifting value).
- **Gas pedal Sensity:** sensity of detecting gas pedal pressing
- **Car Max RPM:** car's maximum RPM
- **Max RPM Sensity:** sensity of playing rpm limit sound. (if CarMaxRPM is set to 7000, and Max RPM Sensity is set to 1, RPM limiting sound will start playing at 6999 RPM. If Max RPM Sensity is set to 100, RPM limiting will start playing at 6900 RPM.)
- **Wait At Shifting:** if Release Gas Pedal At Shifting is set to true, it will play off engine sounds for X seconds. In this example off engine sounds will be played for 0.2 seconds.

Setup for Edy's Vehicle Physics

Youtube video url for this tutorial: https://www.youtube.com/watch?v=D4ZUnQ_vdZc

Import Edy's Vehicle Physics and Realistic Engine Sounds.
Now import **EdysVP_RES-Lite.unitypackage** from the following folder:
.. \RealisticEngineSound\ Assets_For_EVP5\..

This unitypackage gives you prefabs and demo scenes for Edy's Vehicle Physics.

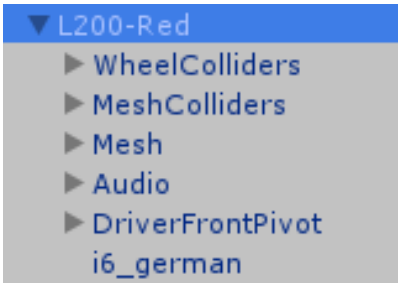
Open any Edy's Vehicle Physics Sample Scene.

Drag and drop one of my prefab for Edy's Vehicle Physics on your car's game object. You can find prefabs for E.V.P. car in the following folder:

../RealisticEngineSound/Assets/Prefabs/EVP5_Prefabs

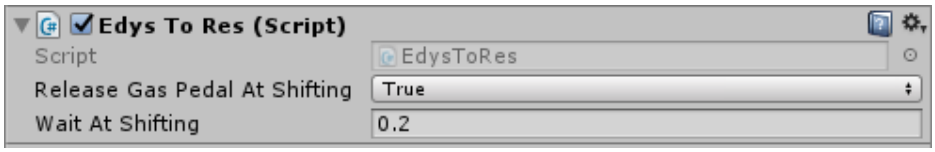
In this example I used i6_german prefab on my car game object.

EVP car gameobject should look like this after you dropped one of my prefab on it:



That's all, now your Edy's Vehicle Physics car have a cool engine sound, it's time to test it! Press Play button! ☺

Edy's to RES controller script settings:



- **Release Gas Pedal At Shifting (True or False):** if set to true it will release gas pedal at shifting, and it will play off sounds for a short time (Wait At Shifting value).
- **Wait At Shifting:** if Release Gas Pedal At Shifting is set to true, it will play off engine sounds for X seconds. In this example off engine sounds will be played for 0.2 seconds.

Camera controller for Edy's Vehicle Physics

Youtube video url for this tutorial: https://www.youtube.com/watch?v=D4ZUnQ_vdZc

You can control interior and exterior sounds by changing camera view with the stock EVP camera controls. All you need to do is just add a gameobject with my camera controller script to your car.

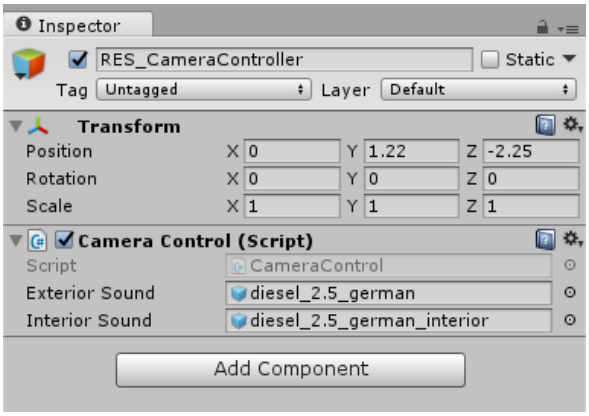
Add an engine pack with exterior and interior sounds to you EVP car. In this example I used the diesel prefab. Add a new gameobject to your car and name it to whatever you want. In this example I named it to "RES_CameraController".

Now your car should look like this:

In your newly created gameobject attach "CameraControl.cs" script. (you can find it in here: ..\RealisticEngineSound\Assets\Scripts\Demo Scene Scripts\EdysDemoScripts)

Into this script attach your exterior and interior engine sound prefabs from your car.

It should look like this:



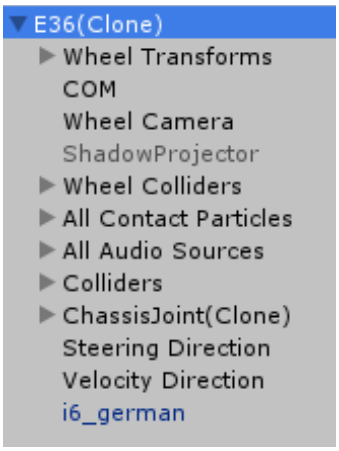
Positioning this gameobjects can make your car sound louder or less louder.

Now time to test it, press play and start changing camera views with the default change camera key. You can edit the default change camera key in "Camera Controller" gameobject's "VehicleCameraController.cs" script.

Setup for Realistic Car Controller

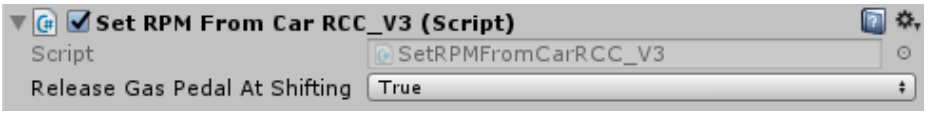
Youtube video url for this tutorial: https://www.youtube.com/watch?v=VcvwsZiF7H4

Import Realistic Car Controller and Realistic Engine Sounds.
Now import **RCC_RES-Lite.unitypackage** from the following folder:
.. \RealisticEngineSound\ Assets_For_RCC_V3\..
This unitypackage gives you prefabs and demo scenes for Realistic Car Controller.
Open any Realistic Car Controller Scene, click on *Option Menu* and spawn a car. In this example I spawned E36 car.
Drag and drop one of my prefab for Realistic Car Controller on your car's game object. You can find prefabs for R.C.C. car in the following folder:
../RealisticEngineSound/Assets/Prefabs/RCC_V3_Prefabs
In this example I used i6_german prefab on my car game object.
RCC car gameobject should look like this after you dropped one of my prefab on it:



That's all, now your Realistic Car Controller car have a cool engine sound! Enjoy it! ☺

RCC V3 to RES controller script settings:



- **Release Gas Pedal At Shifting (True or False):** if set to true off engine sounds will be played at shifting.

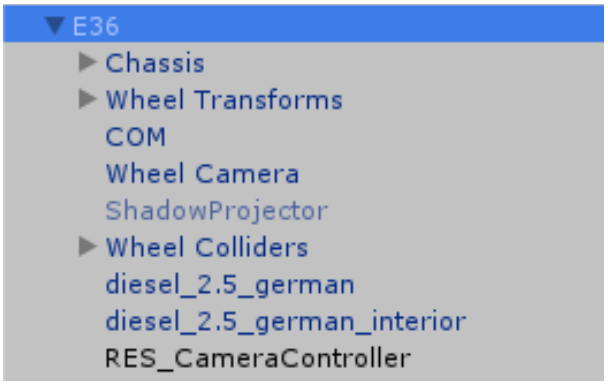
- Camera controller for Realistic Car Controller

Youtube video url for this tutorial: https://youtu.be/VcvwsZiF7H4?t=86

You can control interior and exterior sounds by changing camera view with the default RCC camera controls. All you need to do is just add a gameobject with my camera controller script to your car.

Add an engine pack with exterior and interior sounds to you RCC car. In this example I used the diesel prefab. Add a new gameobject to your car and name it to what you want. In this example I named it to "RES_CameraController".

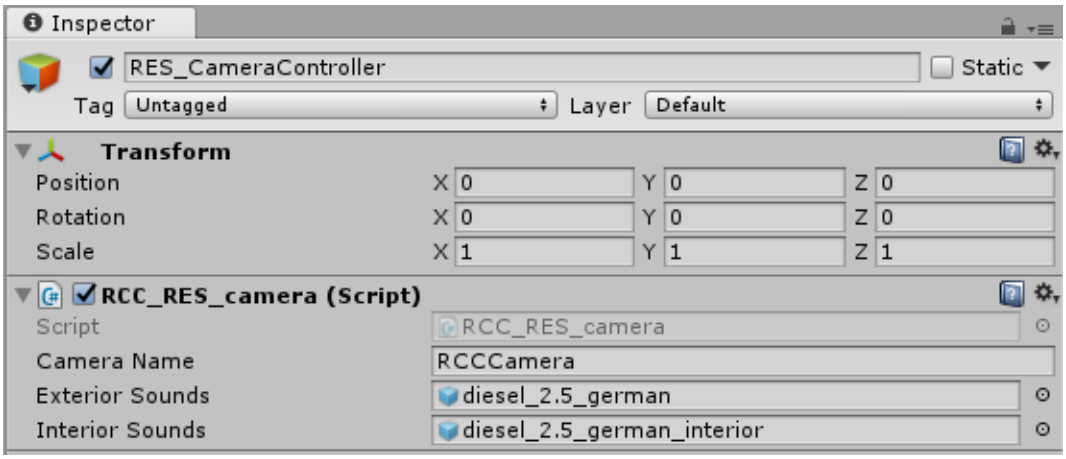
Now your car should look like this:



In your newly created gameobject attach "RCC_RES_camera.cs" script. (you can find it in here: ..\RealisticEngineSound\Assets\Scripts\Demo Scene Scripts\RCCDemoScripts)

Into this script attach your exterior and interior engine sound prefabs from your car.

It should look like this:



- **Camera Name:** RCC's camera controller name

Positioning this gameobjects can make your car sound louder or less louder.

Now time to test it, press play and start changing camera views with the default change camera key.

Setup for Randomation Vehicle Physics

Youtube video url for this tutorial: https://www.youtube.com/watch?v=02JMdDAPsnE

RVP simulates rpms very badly, it can get a bit better by tweaking the transmission's setting in your RVP car, but it never going to be as good as other car controller assets.

Import Randomation Vehicle Physics and Realistic Engine Sounds.

Now import **RPV_RES-Lite.unitypackage** from the following folder:

.. \RealisticEngineSound\ Assets_For_RVP2\..

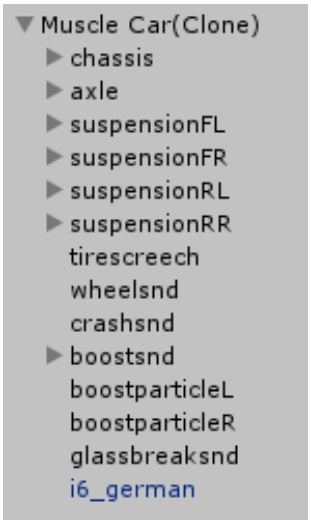
This unitypackage gives you prefabs and demo scenes for Randomation Vehicle Physics.

Open any Randomation Vehicle Physics Sample Scene.

Spawn a car, drag and drop one of my prefab for Randomation Vehicle Physics on your car's game object. You can find prefabs for R.V.P. car in the following folder: ../RealisticEngineSound/Assets/Prefabs/ RVP2_Prefabs

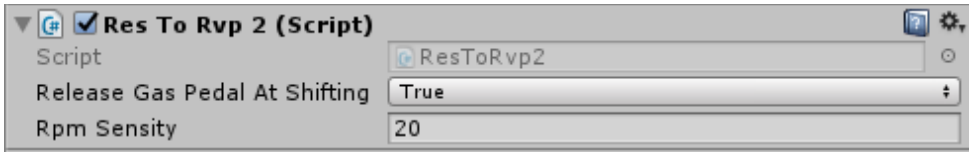
In this example I spawned *Muscle Car* and used i6_german prefab on my car game object.

RVP car gameobject should look like this after you dropped one of my prefab on it:



That's all, now your Randomation Vehicle Physics car have a cool engine sound, it's time to test it!☺

RVP to RES controller script settings:



- **Release Gas Pedal At Shifting (True or False):** if set to true off engine sounds will be played at shifting.

- **RPM Sensity:** sensity of detecting car's RPM

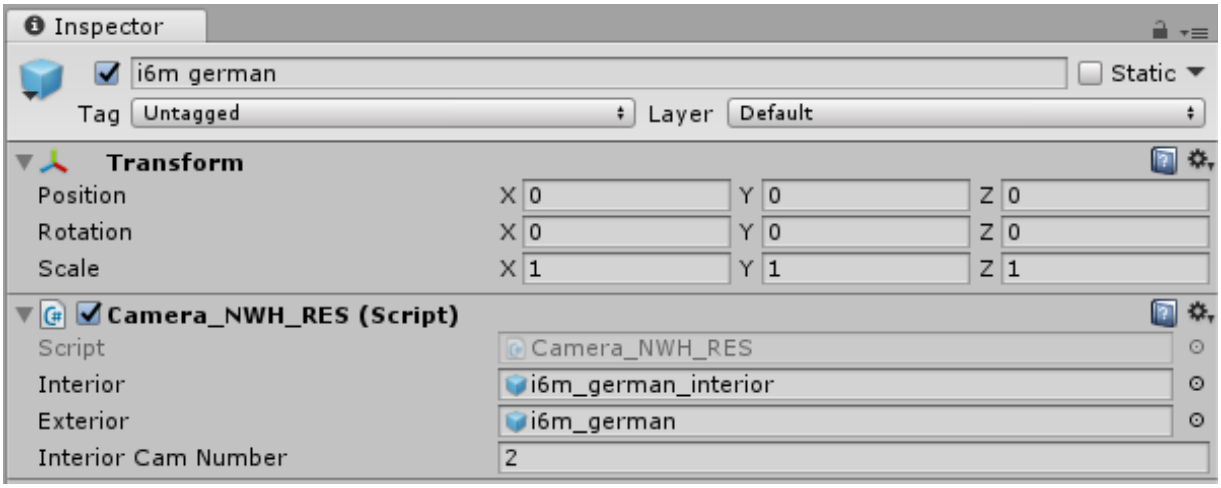
Setup for NWH Vehicle Physics

Youtube video url for this tutorial: https://www.youtube.com/watch?v=LiigkmlRQWs

Import NWH Vehicle Physics, set up to make it work and import Realistic Engine Sounds.
Now import **NWH_RES-Lite.unitypackage** from the following folder:
.. \RealisticEngineSound\ Assets_For_NWH\..
This unitypackage gives you prefabs and demo scenes for NWH Vehicle Physics.
Open any NWH Vehicle Physics scene, drag and drop one of my prefab for NWH Vehicle Physics on your car's game object. You can find prefabs for NWH car in the following folder:
../RealisticEngineSound/Assets/Prefabs/ NWH_Prefabs

NWH_RES controller script:

Camera_NWH_RES controller script:



This prefabs are working with NWH's camera system, automatically switch between interior and exterior RES sounds.

That's all, now your NWH Vehicle Physics car have a cool engine sound! Enjoy it! ☺

Short Youtube video url for this tutorial. In this tutorial I'm using "Motorbike Engine Sounds" RES addon pack, but the usage of prefabs are the same without this addon pack: https://www.youtube.com/watch?v=Xddos137BBk

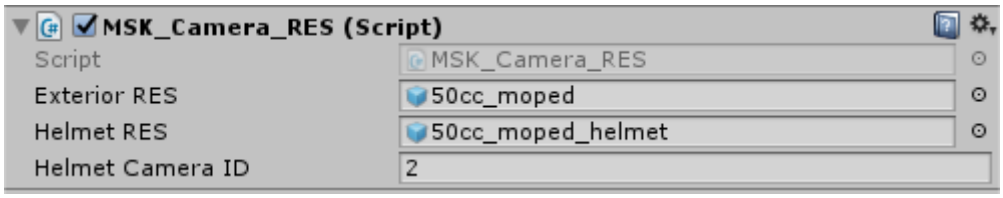
Get Motorbike Engine Sounds - RES Addon Pack here: https://goo.gl/LPkoZy

Firstly import Realistic Engine Sounds and Motorbike Starter Kit assets and later import

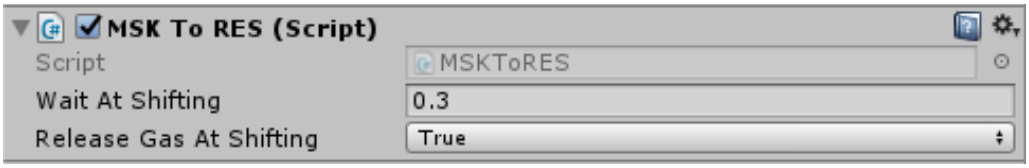
MSK_RES-Lite.unitypackage from ../RealisticEngineSound/Assets_For_MSK folder. This import prefabs, scripts and demo scenes for MSK and RES.

Look for MSk-RES bike prefabs in ../RealisticEngineSound/Assets/Prefabs/MSK_Prefabs/Bikes folder.

Drag and drop one MSK-RES prefab on your MSK bike. This prefabs contain a controller script that swich between helmet and exterior camera view prefabs.



HelmetCameraID: is the number of helmet camera in your MSK gameobject.



WaitAtShifting: sets the waiting time in seconds for shifting.

ReleaseGasAtShifting: if set to True, off engine sounds will be played for "WaitAtShifting" seconds.

Set Up Your Own Engine Sound

Youtube video url for this tutorial:

https://www.youtube.com/watch?v=xsLy18rokSk

Create a new scene, add a new game object and attach Realistic Engine Sound script to your new game object.

Set audio clips, if you don't have your own, use audio clips from:

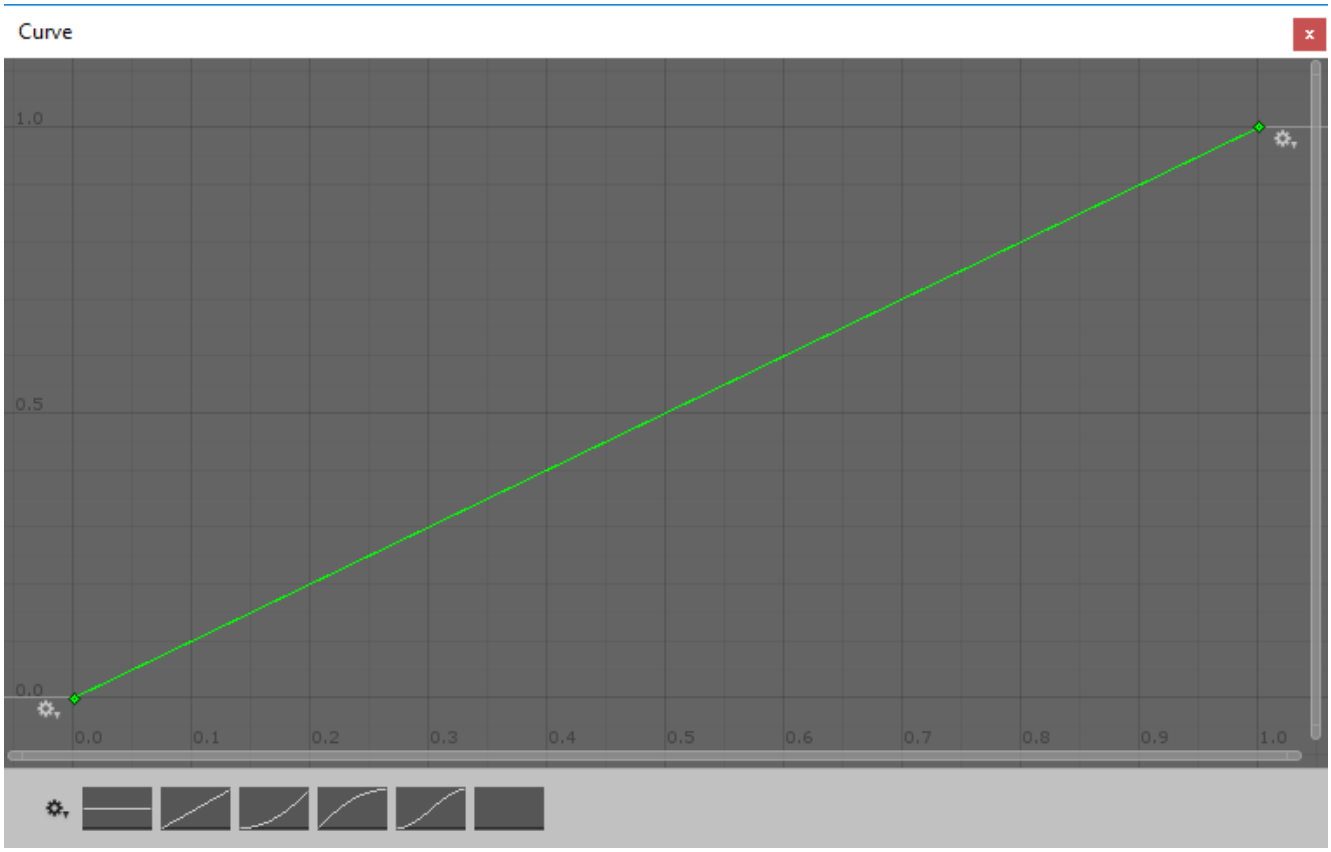
../RealisticEngineSound/Assets/Sounds/... and choose any folder that has engine names.

You can find Reversing Clips here:

../RealisticEngineSound/Assets/Sounds/Reversing

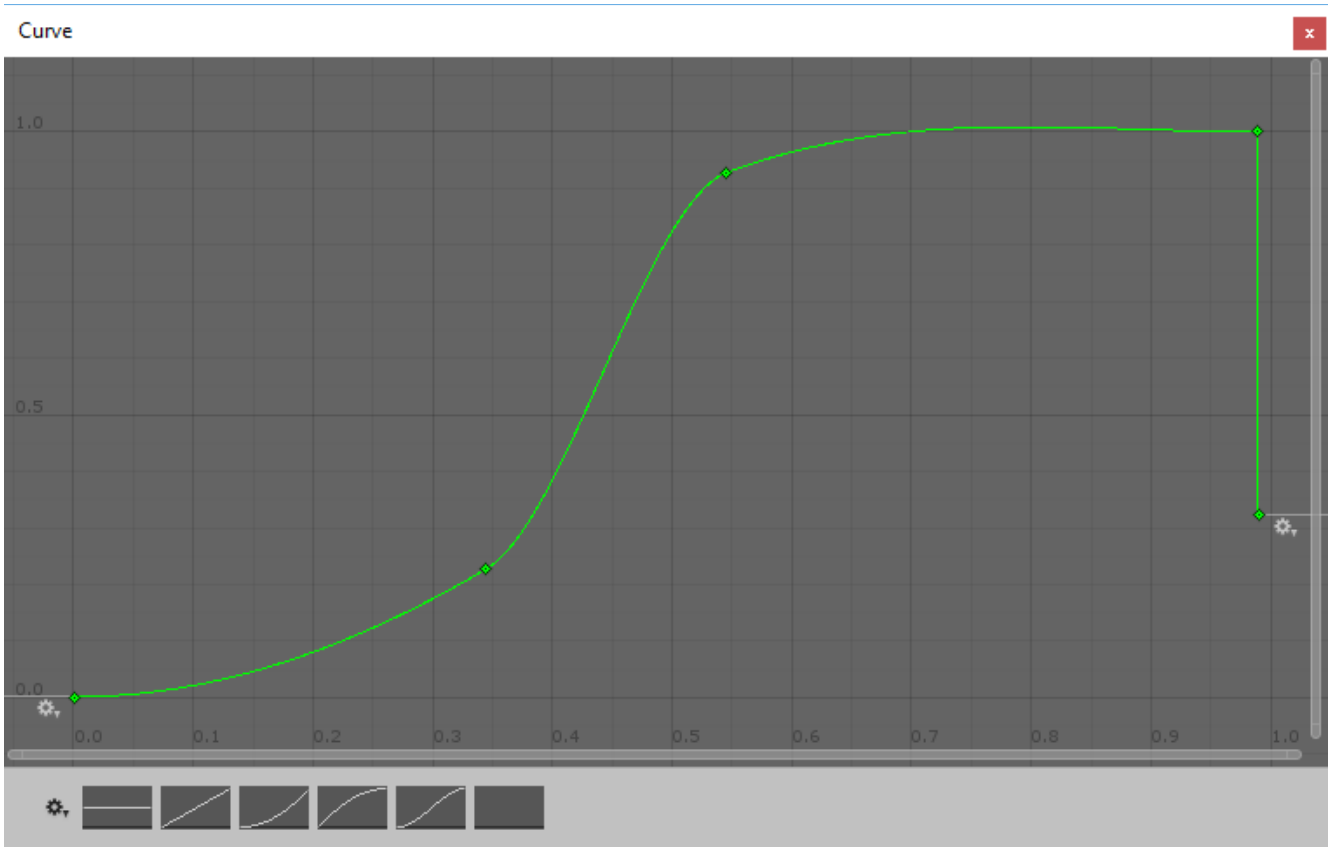
Set each audio clips Volume Curve and Pitch Curve settings.

For pitch Curve settings it's better to set it to look like this:



*Values from left to right do not set it to more than 1.0 value. Values more than 1.0 will never take effect.

For volume curve settings you need to try out a few settings that fits better for your audio clip. Bellow you can see volume curve's setting for high rpm sounds:



*Values from left to right do not set it to more than 1.0 value. Values more than 1.0 will never take effect.

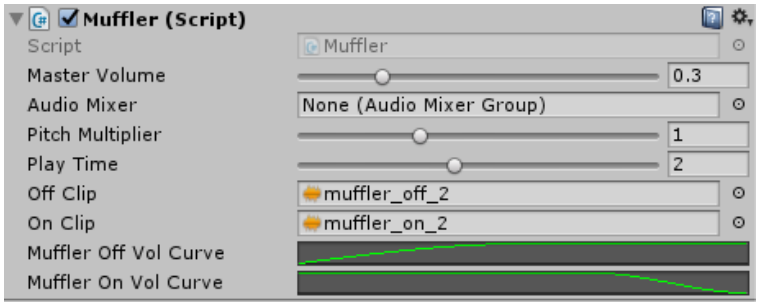
If you have struggles finding the best settings for your audio clips, look inside my prefab's settings, or change one of my prefab's audio clips to your audio clips – this is the fastest way to test your own audio clips.

If you have set up everything, press the play button and start increasing "Engine Current RPM" in Realistic Engine Sound script to test your engine sound.

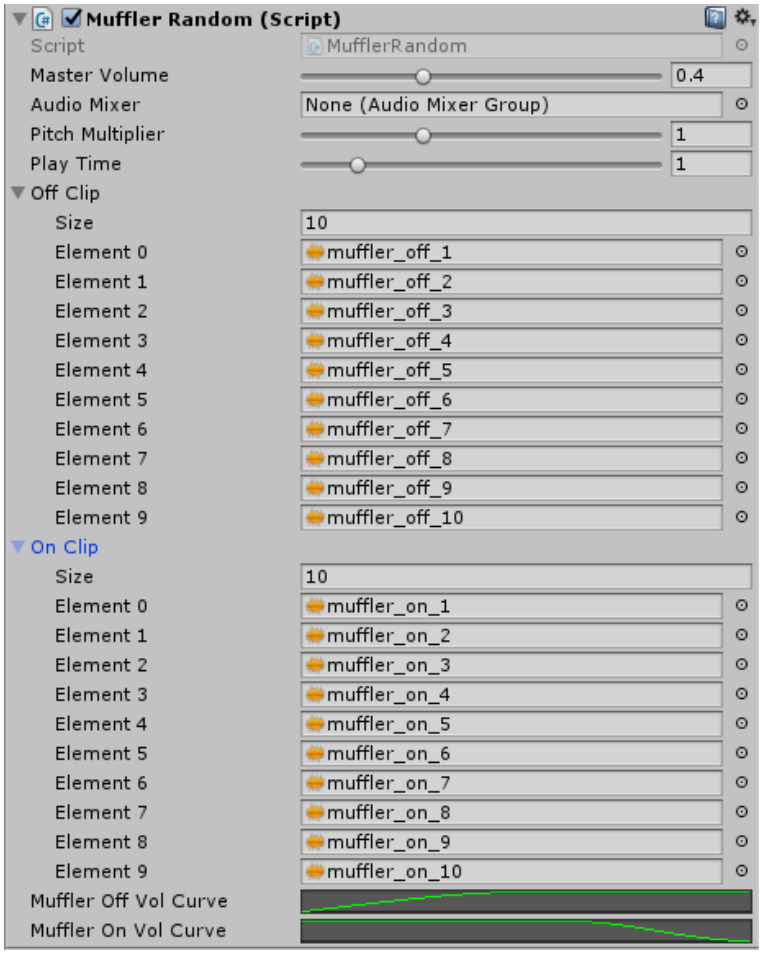
Don't forget to enable "Gas Pedal Pressing" to simulate accelerating sounds. ☺

Muffler Crackle Sound

Non random script:



Random sound clips script:



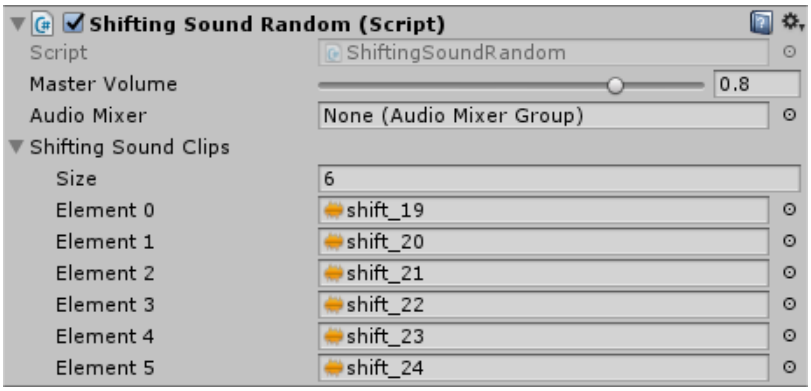
- **Master Volume**: sets the gameobject's maximum volume.
- **Audio Mixer:** add an audio mixer here to control all sound's volume at once. If set to none, it will use the engine prefabs's audio mixer (if the engine prefab have an audio mixer).
- **Pitch Multiplier**: multipli the prefab's maximum pitch value.
- **Play Time:** the time for sound clips is played in secconds
- **Off Clip**: sound clip played for „Play Time" secconds when gas pedal is released.
- **On Clip:** sound clip played for „Play Time" secconds when gas pedal is pressed.
- **Muffler Off Vol Curve**: volume settings for muffler off clips.
- **Muffler On Vol Curve:** volume settings for muffler on clips.

Drag and drop this prefab on your RES prefab.

Shifting Sounds

Non random script:

Random sound clips script:



- **Master Volume**: sets the gameobject's maximum volume.
- **Audio Mixer**: add an audio mixer here to control all sound's volume at once. If set to none, it will use the engine prefabs's audio mixer (if the engine prefab have an audio mixer).
- **Shifting sound clip:** the sound clip which is played when car is changing gears.

Drag and drop this prefab on your RES prefab.

Tips, Tricks and Optimisations

- To get more aggresive engine sounds, add an audio mixer to the RES prefab and add Disortion effect to the Master chanel. (this may eat some performance on mobile) Here is a video how to do this and what results it give: https://www.youtube.com/watch?v=oiq6tQQuiJ0

- On mobile devices you can use regular prefabs instead of mobile prefabs. Try using regular prefabs for player's car and mobile prefabs for opponent's car. Try out to see what FPS you get. If you do not get any performance loss, use regular prefabs for opponent's vehicle too, to get the most realistic engine sounds in your mobile game.

- In scenes with more vehicles, to save performance try reduncing „Max Distance" in RES prefabss to reduce active Audio Sources in your scene.

- Increase the „Optimisation Level" in RES prefab to reduce the used Audio Sources. This also reduce the quality of the RES prefab.

- Delete those engine sound packages that you may never use in your game to reduce your game's file size. Do the same for shifting, turbo, supercharger, etc sounds. This will reduce your compiled game's size with a few 10 or more MegaBytes. (good for mobile games).

- **[For experienced users]** Before doing this, backup your project. If you would like to optimise your game, edit your car controller scripts to remove the usage of it's stock engine sound's audio source. If needed remove the Audio Sources too by hand. If you don't know what lines you need to delete from your car controller script, delete the stock engine sound's Audio Sources in runtime and wait for the Errors. This errors will show you wich code lines are using the Audio Sources. But be carefull, do not remove lines that calculate the car's RPM. The less Audio Sources you have in your scene, the better performance you get on mobile. This modifications can give you errors inside RES_To... car controllers.cs script, because it will look for the stock Audio Sources to disable it. Remove this lines and done. Another thing is if you ever update your car controller to a newer version, this modifications may need to be done again. Now you completly removed your car controller's stock engine sounds and saved some performance.

License

You can use this asset for unlimited games.

One license per seat for personal and commercial use.

You can't resell or redistribute the package or any single file from the package on any store!

Credits

All of the sound files are recorded, created and mastered by me.

All of the scripts are written by me.

I would like to say a *Thank You* for the vehicle owners who allowed me to record their car's and bike's engine sound in car and bike meetings and other public events. Some sound files are recorded from *dyno* videos found around the web.

Feel free to contact me if you have any questions or suggestions for this asset.

Youtube playlist with tutorials and sample videos:

https://www.youtube.com/watch?v=ByGSyMrpacw&index=1&list=PLYFtdNoo8S3gv1XVv193fI6AhFmgwMMG0

Official Forum: *https://forum.unity.com/threads/released-realistic-engine-sounds.479120/*

Don't forget to write a review for my asset in the Unity Asset Store. Thank you!

Get engine packs for RES Lite:
https://assetstore.unity.com/publishers/29907?aid=1101l3r5B

Upgrade to the Plus version here: https://goo.gl/t1hm7Z

Upgrade to the Pro version here: https://goo.gl/DCTgsb

Made by: Szlacki Attila

Contact: slaczky@gmail.com