Realistic Engine Sounds 2 - Lite Edition

By: Skril Studio

Thank you for purchasing Realistic Engine Sounds 2 Lite Edition!

I hope that your project is going to be a huge success!

Please read this documentation carefully. Read at least the first few chapters if you don't like reading documentations.

Let me know if you have some questions that cannot be answered with this documentation. Contact details can be found at the end of this documentation.

If you like this asset please write a review for it in the Unity Asset Store. A review significantly can help the development of this asset and it's visibility in the store. Thanks!

Contents

Package Includes....4
Importing Realistic Engine Sounds 2 into your project....5
How to update from RES1 or from earlier RES2 version....5
Engine Sound Prefabs Quick Explanatory....5
FAQ – Frequently Asked Questions....6
Realistic Engine Sounds 2 Custom Editor....7
Setup for Custom Car Controller (your self made car physics controller)....14
Setup for Edy's Vehicle Physics....15
Setup for Realistic Car Controller....16
Setup for NWH2 Vehicle Physics....17
Setup for Vehicle Physics Pro....18
Setup for Universal Vehicle Controller....19
Setup for FGear Vehicle Physics....20
Setup for IRDS....21
Setup for Realistic Car Controller Pro....22
Setup for Arcade Car Controller....23
Setup for Realistic Traffic Controller....23
Setup for Realistic Car Kit....24
Setup for Sim-Cade Vehicle Physics....24
Setup for Simple Motorcycle Physics Pro....25
Setup for Simple Motocross Physics....26
Setup for Motorbike Physics Tool....27
Setup for Highroad Engine....28
Setup for Arcade Bike Physics Pro....28
Setup for Racing Game Starter Kit....29
Setup for ASH Vehicle Physics....29
Setup for Ushi Arcade Car Physics....30
Setup for Semi-Realistic Car Physics....30
Setup for Arcade Vehicle Physics....30
Setup for Quick And Easy Car Racing....31
Setup for Multiversal Vehicle Controller Pro....31

Muffler Crackle Sound.....32
Shifting Sounds.....33
City Car Gearbox Whine SFX.....34
Tips, Tricks and Optimisations.....35
License.....36
Credits.....36

Package includes:

- 25 .unitypackages for adding compatibility to vehicle physic controllers (containing scripts, prefabs and demo scenes):

  - Custom Car Controller (an example compatibility made for custom car controllers)
  - Edy's Vehicle Physics
  - Realistic Car Controller
  - NWH2 Vehicle Physics
  - Vehicle Physics Pro
  - Universal Vehicle Controller
  - FGear Vehicle Physics
  - IRDS
  - Realistic Car Controller Pro
  - Arcade Car Controller
  - Realistic Traffic Controller
  - Realistic Car Kit
  - Sim-Cade Vehicle Physics
  - Simple Motorcycle Physics Pro
  - Simple Motocross Physics
  - Motorbike Physics Tool
  - Highroad Engine
  - Arcade Bike Physics Pro
  - Racing Game Starter Kit
  - ASH Vehicle Physics
  - Ushi Arcade Car Physics
  - Semi-Realistic Car Physics
  - Arcade Vehicle Physics
  - Quick And Easy Car Racing
  - Multiversal Vehicle Controller Pro

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

  - i4 German 1
  - 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 1
  - V8 Italian 3

- 264 wav files for engine sounds
- 12 wav files for muffler crackle sfx
- 20 wav files for reverse gear sfx (gear whine and beeping warning sounds)
- 4 wav files for reverse gear sfx
- 26 wav files for valves sfx
- 12 wav files for wind sfx

There are total 354 Wav audio files in this package.
There are 58 prefabs in this package.
(prefabs made for vehicle controllers are not counted in)

Importing Realistic Engine Sounds 2 into your project

Import Realistic Engine Sounds 2 into your project and later import the right *.unitypackage for the car controller you're using in your project.

When you're importing this asset for a project that is using a vehicle physics controller that is currently unsuported by Realistic Engine Sounds 2, you need to create your own script that can convert values from your vehicle to RES2 prefab (current RPM, maximum RPM, current speed, maximum speed, gas pedal input value, is currently changing gears).
Feel free to contact me about adding support for an unsuported vehicle physics controller, but you must share a sample project with me, otherwise I can't help to write the compatibility for your vehicle physic controller.

How to update from RES1 or from earlier RES2 version

Before updating to Realistic Engine Sound 2, backup your current project!

Always keep your own customized files and prefabs outside of *RealisticEngineSound* directory!

If you upgrade from RES1, delete *RealisticEngineSound* directory from your project and import RES2 into your project. All prefabs need to be re-added to your vehicles, because all prefabs got new audio files and new values.

If you upgrade from an earlier version of RES2, delete *RealisticEngineSound* directory from your project and import the updated RES2 version into your project.

After these steps import the right *.assetpackage made for adding compatibility to your vehicle physics controller.

That's all, now you have the updated version of Realistic Engine Sounds 2. Read the „Release Notes.txt" to find out whats new in this version.

Engine Sound Prefabs Quick Explanatory

Realistic Engine Sound 2 has three engine sound prefab quality levels:

- **High Quality:** using 18 audio clips for the engine sound prefab (16 audio clips for engine sound + 2 audio clips for aggressiveness sfx, and using pre recorded off load sounds).
- **Medium Quality:** using 10 audio clips for the engine sound prefab (8 audio clips for engine sound + 2 audio clips for aggressiveness sfx, and using pre recorded off load sounds).
- **Low Quality:** using 7 audio clips for the engine sound prefab (5 audio clips for engine sound + 1 audio clip for aggressiveness sfx, and using simulated off load sounds).

Any of these prefabs will work in mobile devices, but in scenes where multiple vehicle can be heard at the same time, multiple High Quality prefabs can reach the „Maximum Real Voices" of your project's audio setting. To avoid this, I recomend to use High Quality prefabs for the player's vehicle and for the opponents vehicle use Medium or Low Quality prefabs. Each prefab is trying to optimise itself by the audio listeners position, stopping its audio clips when they cannot be heard (will only do this if „Optimise Audio Sources" is enabled in the prefab).

You can enable off load simulation for High Quality prefabs which will help to save some space for other sound clips (this will change how the car sound during off load). Doing the same with Medium Quality prefabs will turn them into Low Quality prefabs. Low Quality prefabs are just Medium Quality prefabs with simulated off load sounds.

Don't forget to set the engine sound prefab to „Opponent" mode if it is used for opponent. In open world games switch between „Player" and „Opponent" modes when player enters/exits the vehicle (these steps are needed when Dynamic Audio Mixer is used by the prefab).

FAQ - Frequenlty Asked Questions

- What are the differences between Lite, Plus and Pro Editions?
The difference is how much sound files they have, the engine sound controller script is the same in all editions.

- What are the differences between RES1 and RES2?
Sound controller script has been rewriten from zero, created a new custom editor to make it easier and faster to create new engine sound prefabs, many new features has been added to the sound controller, all wav files has been re-recorded or remastered + 1850 new wav files has been added.

- Are RES1 prefabs compatible with RES2?
Yes, but they may need small tweaking. In Realistic Engine Sounds 2 new features and values has been added which does not exist in the old prefabs, these need to be added by hand for each old versioned prefabs.
If you try to play a scene by using old prefabs with RES2 you probably get errors about missing audio mixers. The easiest way to fix this is to disable „Dynamic Audio Mixer" because this is the feature that is looking for the audio mixers, or the other way is add the missing audio mixers to the old prefabs and keep using the „Dynamic Audio Mixer". („Dynamic Audio Mixer" will change the car's engine sound based on the player camera position, it will sound differently if the player is looking at the front or at the back of the car.) These audio mixers must be the same that came with asset, the same what you can find in original RES2 engine prefabs. „Player Audio Mixer" needs to get „RES2 Player Vehicle" audio mixer and „Opponent Audio Mixer" needs „RES2 Opponent Vehicle" audio mixer. Open an original RES2 engine sound prefabs if you need an example which mixer goes where.

- Will my custom compatibility script made for RES1 will work with RES2?
No, you need to do some changes to your custom script writen for RES1. In most cases you just need to add "SkrillStudio" namespace to your script.

- Do I need to re-add all Realistic Engine Sounds prefabs to my vehicles after upgrading from RES1 to RES2?
Yes, you need to delete the old RES1 prefabs form your vehicles and then add the new RES2 prefabs. This is needed because all RES2 prefabs have new audio clips and new values.

Can't find the answer to your question? Contact me in:

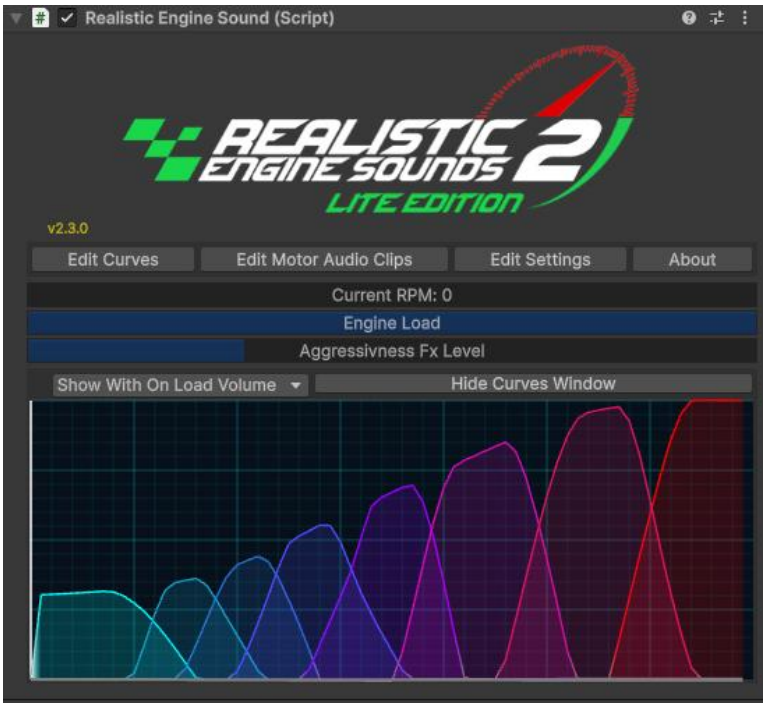
- Email: info@skrilstudio.com

- Facebook message: https://fb.com/yugelmobile

- Contact form in my website: http://skrilstudio.com/contact.html

Realistic Engine Sounds 2 Custom Editor

Realistic Engine Sounds 2 has a custom editor which is not open source and it is designed to help creating and tweaking values for engine sound prefabs. I designed it to be as easy to use as possible, well organised settings to avoid confusion and has a cleaner look than the 1.x.x version of RES. Thanks to the Curves Windows it is way easier to tweak the volume curve values for audio files more precisely. You can edit the curves by clicking on Edit Curves button. This will enable each animation curve that is available for the engine sound audio files.



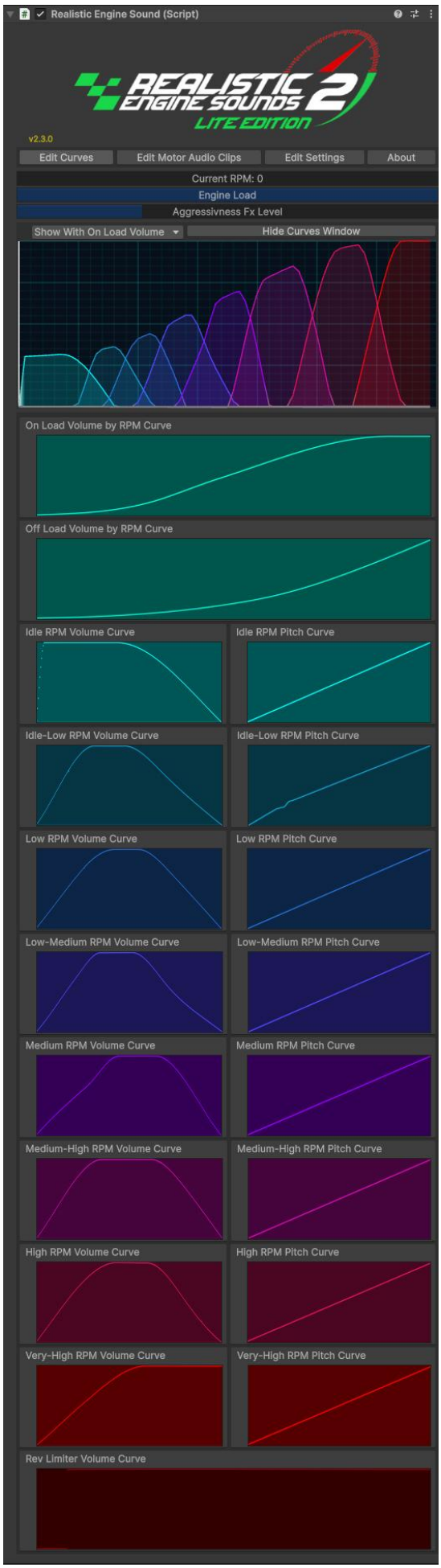
- **Edit Curves:** this option will enable the page where you can edit audio files curve value. These values will determine audio filess volume and pitch.
- **Edit Motor Audio Clips:** here you can change those audio files that are related to engine sounds.
- **Edit Settings:** you can change settings or enable/disable features.
- **About:** links to contact support, documentation, add-on sound packs and other Skril Studio assets.

Each curve represents a different RPM-level audio file, ordered from left to right:

White Line = Current RPM
Idle RPM → Idle-low RPM → Low RPM → Low-Medium RPM → Medium RPM → Medium-High RPM → High RPM → Very High RPM

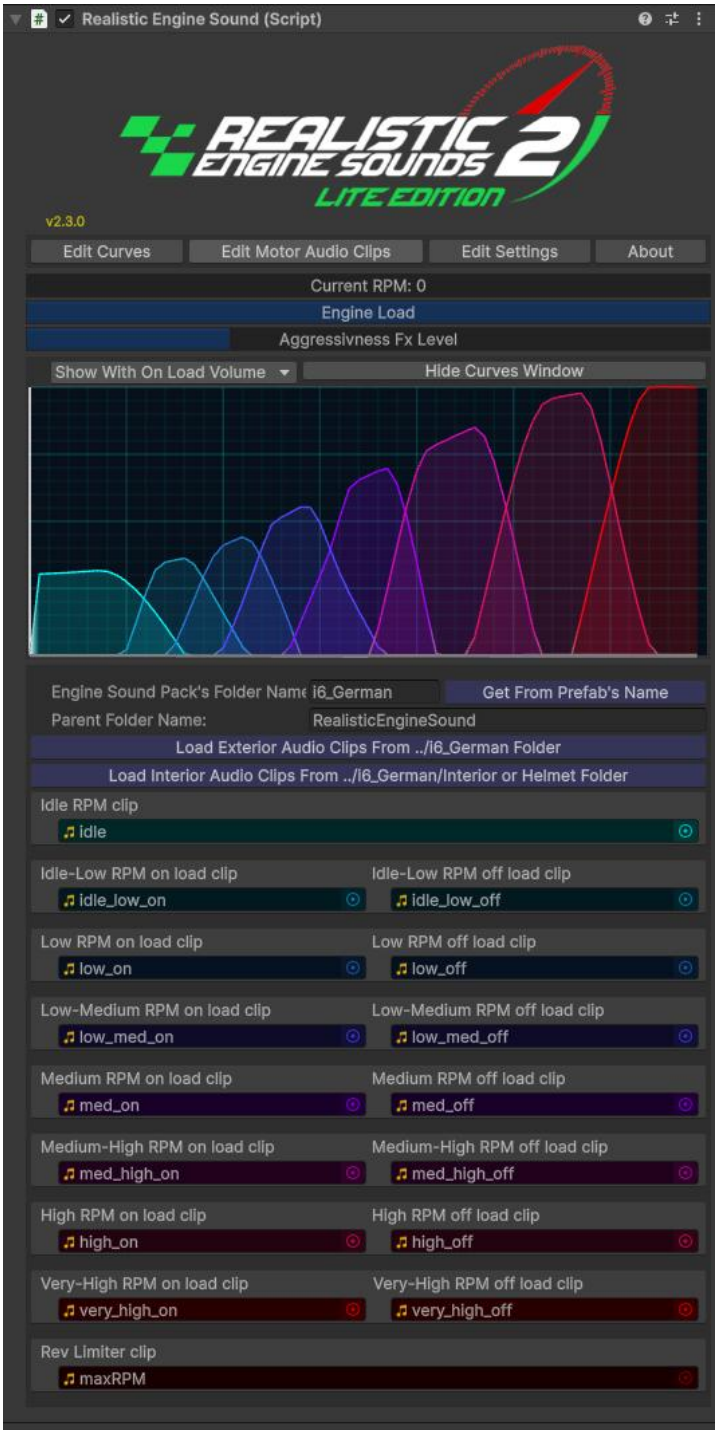
The Rev Limiter audio level is not visible in the window.

Curve Window can show „Raw Curves" which are the real values of each animation curve, or can show „On Load Curves" wich show the real value of the prefab that can be heard when the engine load level is at maximum, or it can also show „Off Load Curves" wich is the real value of the prefab when the engine load level is at minimum. By default it shows the „On Load Values".
Curves Window can be hiden by „Hide Curves Window' button.



Edit Curves:

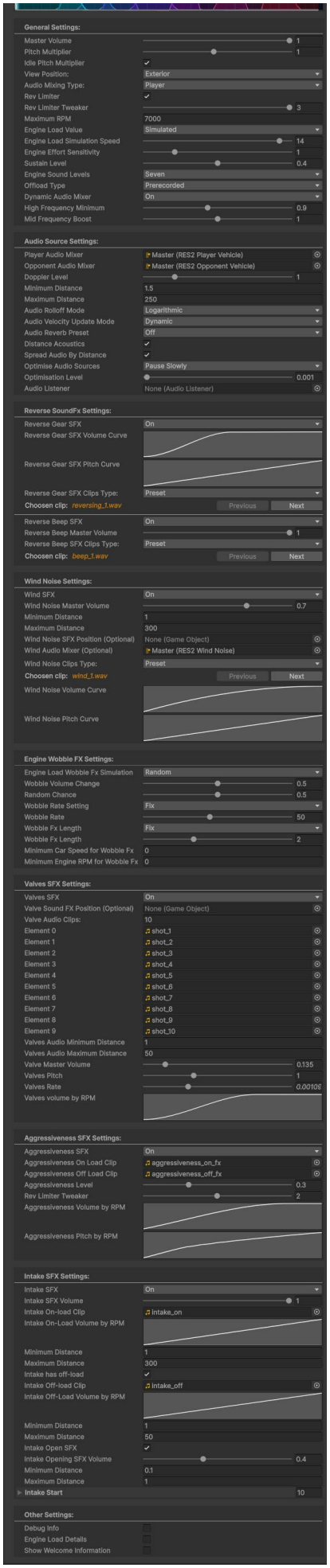
- **On Load Volume by RPM Curve:** edit the value of on load volume. All engine sound volume curves volume will not goes above On Load Volume curve's value when engine load is currently at maximum level (is accelerating, etc).
- **Off Load Volume By RPM Curve:** edit the value of off load volume. All engine sound volume curves volume will not goes above Off Load Volume curve's value when engine load is currently at minimum level (is decelerating, or gas pedal not pressed, etc).
- **Idle RPM Curves:** edit volume and pitch values for idle rpm audio.
- **Idle-Low RPM Curves:** edit volume and pitch values for idle-low rpm audio files (both on-load and off-load audio files are using these animation curves).
- **Low RPM Curves:** edit volume and pitch values for low rpm audio files (both on-load and off-load audio files are using these animation curves).
- **Low-Medium Curves:** edit volume and pitch values for low-medium rpm audio files (both on-load and off-load audio files are using these animation curves).
- **Medium Curves:** edit volume and pitch values for medium rpm audio files (both on-load and off-load audio files are using these animation curves).
- **Medium-High Curves:** edit volume and pitch values for medium-high rpm audio files (both on-load and off-load audio files are using these animation curves).
- **High Curves:** edit volume and pitch values for high rpm audio files (both on-load and off-load audio files are using these animation curves).
- **Very-High Curves:** edit volume and pitch values for very-high rpm audio files (both on-load and off-load audio files are using these animation curves).
- **Rev Limiter Volume Curve:** this can change how the rev limiter start playing after very-high rpm audio. I'm not recomending changing this animation curve too much, because the current value is the one that works the best with the asset.



Edit Motor Audio Clips:

Here you can change the engine sound audio clips of the currently selected prefab. You can change the audio clips by hand one-by-one or you can get the audio files automatically with a few button clicks (audio files must be named the same as in the screenshot, prefab name must be the same as the folder where the audio files are placed and this folder needs to be inside the assets *Sounds* folder, audio clips made for interior view needs to be inside a folder with name *Interior* – the folder structure as for any other sound packs that comes with the asset).

- Get From Prefab's Name: this button will get the folder's name from the prefab name with a single click.
- Load Exterior Audio Clips From../*folder name* Folder: this button will load all audio clips from the soundpacks folder with a single button click – files names needs to be the same as in the screenshot and folder structure needs to be correct as I mentioned above.
- Load Interior Audio Clips From ../*folder name*/Interior Folder: this button will load those audio files that are made for Interior view (if they are exist) and also change some settings for the prefab inside „Edit Settings" pannel: master volume, prefab type, and the sound clips for sound effects like wind noise, reverse gear, etc.



Edit Settings:

General Settings:

- **Master Volume:** Sets the maximum volume an audio source can get.
- **Pitch Multiplier:** Multiplies audio sources pitch value by this value.
- **Idle Pitch Multiplier:** Enable or disable Idle audio sources's pitch multiplication with 'Pitch multiplier' value.
- **View Position:** Choose in which camera view you will use this prefab. This will switch between the Reversing SFX and Wind Noise SFX clips exterior / interior view sounds. Changing this value does not change the motor sound clips.
- **Audio Mixing Type:** Different audio mixer is used for player and for opponents. Change this value when the vehicle is controller by the player or by an opponent. Do not use player type prefabs for opponents because it will conflict the sound mixing. This works like this because in Unity prefabs can't instatiate their own audio mixer. In an open word game you should change this value everytime when player enter/exit a vehicle. You can even use your own custom audio mixer by choosing „Custom" mode, but in this case Dynamic Audio Mixer feature will not work.
- **Use Rev Limiter:** Enable or disable the pre-recorded rev limiter sound.
- **Rev Limiter Tweaker:** Sets the volume of very_highOn audio source while rev limiter is played. The higher the value the more it will reduce very_highOn audio volume.
- **Maximum RPM:** the maximum RPM your vehicle can reach.
- **Engine Load Value:** if set to simulated, engine load value will be simulated based on the gas input value. If it is set to not simulated then you need to control this value with your vehicle's engine load value. Engine Load determines which load audio clip is louder, on load or off load (example: high_on or high_off). Both will be played with 50% volume when engine load is at 50%.
- **Engine Load Simulation Speed:** if Engine Load Value is set to simulated then the simulated engine load „acceleration" can be edited here. (how fast to switch between on load and off load audio clips)
- **Engine Effort Sensitivity:** Engine Effort detection sensitivity.
- **Sustain Level:** Adjusts the Sustain Level.
- **Engine Sound Levels:** determine how much audio clips will be used for this prefab. It works like this: idle clip + sound levels + rev limiter + other optional sfx (like wind noise, reverse gear sfx, etc).
If set to „Seven" then prefab can use up to 16 audio clips + other sfx.
If it is set to „Three" then up to 8 audio clips will be used (the number of used audio clips is also depnding on „Off Load Type" value).
If set to „One" then only one sound clip will be used.
„Level seven" has the following levels: idle-low, low, low-med, med, med-high, high and very-high.
„Level three" has the following levels: low, med and very high.
„Level one" has only one level which is the idle sound – any soundclip can be used here, ideal for classic arcade games.
- **Off Load Type:** If set to pre-recorded then it will use the pre-recorded audio clips for engine off load, with this setting the prefab will use more audio sources. If set to simulated then it will mimic engine off load sounds with On Load audio clips by using audio mixing techiques, this will also reduce the used audio sources to almost half what it will be with pre-recorded off load sounds. These audio mixing tehniques will only work if the prefab usage is set to „Player" and the default audio mixer is used in the prefab. If set to „Opponent" then Off Load sound will be mimicked with the volume of On Load audio clips.

- **Dynamic Audio Mixer:** This feature requires your scene's Audio Listener and the AudiMixer that is included with the asset. This feature is dynamically EQ mixing the prefab's audio based on the player camera's current view. The car will sound differently when the player is looking at the front or at the back of the vehicle. This is a feature that is not available in every AAA game!
- **High Frequency Minimum:** Available with Dynamic Sounds Mixing. This sets the minimum value for high frequencies when the camera is looking at the back of the player's car. This feature requires the AudiMixer that is included with the asset.
- **Mid Frequency Boost:** Available with Dynamic Sounds Mixing. Original value is: 1.0f. It will 'boost' the prefab sound's mid frequency, this can be also used to 'de-boost' the prefab's mid frequencies. This feature requires the AudiMixer that is included with the asset. Also good for simulating stock / tuned car sounds in a racing game.

Audio Source Settings:

- **Player Audio Mixer:** Used for dynamic audio mixing (in this case it is required to use that audimixer that is shipped with the asset) or you can use your own audio mixer if you are disabled dynamic audio mixing. This mixer is designed to work only with one vehicle! If you will use it for muiltiple vehicles – like player and opponents at the same time, then you will get weird sounds from the vehicles. (Dynamic sound mixing conflict).
- **Opponent Audio Mixer:** Different audio mixer is used for opponents because those prefabs does not use dynamic audio mixing.
- **Custom Audio Mixer:** audio mixing type need to be set to „Custom" mode to make this part appear. With custom audio mixer you can use your own audio mixer, but in this case Dynamic Audio Mixer feature will not work.
- **Doppler Level:** Determines how much doppler effect will be applied to the audio sources (if is set to 0, then no effect is applied).
- **Minimum Distance:** Within the MinDistance, the sound will stay at loudest possible. Outside MinDistance it will begin to attenuate. Increase the MinDistance of a sound to make it 'louder' in a 3d world, and decrease it to make it 'quieter' in a 3d world.
- **Maximum Distance:** The distance where the sound stops attenuating at. Beyond this point it will stay at the volume it would be at MaxDistance units from the listener and will not attenuate any more.
- **Audio Rolloff Mode:** How fast the sound fades. The higher the value, the closer the Listener has to be before hearing the sound. (This is determined by a Graph). For vehicle the most realistic is the „Logarithmic" mode.
- **Audio Velocity Update Mode:** Describes when an AudioSource or AudioListener is updated. If you experience audio stuttering, try changing this value to Fixed mode or to Dynamic mode.
- **Audio Reverb Preset:** this will add a reverb zone to the prefab that will constatly make your vehicle's sound with a reverb effect.
- **Distance Acoustics:** Dynamically adjusts audio effects based on the distance between the vehicle and the listener to create a more realistic sense of space and environmental acoustics.
- **Spread Audio By Distance:** Dynamically adjusts the stereo spread of vehicle audio sources based on the distance from the listener. In interior view, the spread value remains fixed instead of changing dynamically.
- **Optimise Audio Sources:** Pause or destroy audio sources that are currently not needed or can't be heard because of their farther distance, or continue playing them even when they are not needed.
  If „**Pause Slowly**" chosed, unused audio sources will be paused slowly as they reach their sound clips end. Rarely it can cause weird pitch noises, it is also depending on the chosed Audio Velocity Update mode.
  If „**Pause Fastly**" is chosed then unused audio sources will be paused as soon as they are not needed – it may cause click/pop noises.
  If „**Destroy**" is choosed unused audio sources will be deleted. These deleted audio sources later will be re-created when they needed. Warning! 'Destroy' feature may generate some GC allocation, but it will help reducing the number of currently used Audio Sources in your scene.
- **Optimisation Level:** Audio sources whose volume is currently less that this value will be stop playing or will be deleted. (depending on the above 'Optimisation' setting). Using too much value may produce audio 'popping', 'clipping' or 'laggy' like sounds.
- **Audio Listener:** Audio Listener is required for optimising audio. It helps to detect which audio sources can't be heard because of their farther distance. Usually the prefab will get the Audio Listener automatically but you can set it up by hand if it is needed. Audio Listeners are usually attached to main camera.

Reverse SoundFX Settings:

- **Use Reverse Gear SFX:** Enable reverse gear's „whining" soundfx and it's settings.
- **Reverse gear SFX Volume Curve:** the curve where you can change the volume of reverse gear which is affected by the vehicles current speed. It will reach it's maximum value around at 55km/h.
- **Reverse Gear SFX Pitch Curve:** the curve where you can change the pitch of reverse gear which is affected by the vehicles current speed. It will reach it's maximum value around at 55km/h.
- **Reverse gear SFX Clips Type:** „Preset" sound clips are hard coded and can be choosen easly here only in editor mode (will not work in a built play mode) or set it to „Custom" type to use your own sound clip.
- **Use Reverse Beep SFX:** Reverse gear's beep warning sound which will be played in reverse gear.
- **Reverse Beep Master Volume:** The master volume of reverse gear's beeping warning soundfx.
- **Reverse Beep SFX Clips Type:** „Preset" sound clips are hard coded and can be choosen easly here only in editor mode (will not work in a built play mode) or set it to „Custom" type to use your own sound clip.

Wind Noise Settings:

- **Enable Wind SFX:** Enable wind noise and it's audio settings. Enabling wind noise will add a wind noise sound effect that is affected by your vehicle's current speed. It will add a little detail to your vehicle's sound.
- **Wind Noise Master Volume:** The master volume of the wind noise within this prefab.
- **Minimum Distance:** Within the Minimum distance the wind noise will cease to grow louder in volume.
- **Maximum Distance:** The distance where the sound stops attenuating at. Beyond this point it will stay at the volume it would be at MaxDistance units from the listener and will not attenuate any more.
- **Wind Noise SFX Position (Optional):** Wind noise sfx will be instantiated in this gameobject. Position this gameobject somewhere in your vehicle if you want to change the wind noise's position. If you leave it empty, Wind Noise will be instantiated at engine sound audios position.
- **Wind Audio Mixer (Optional):** You can use a separate audio mixer for wind noises if you want to. Otherwise it will use the engine sound's audio mixer if any is used.
- **Wind Noise Clips Type:** „Preset" sound clips are hard coded and can be choosen easly here only in editor mode (will not work in a built play mode) or set it to „Custom" type to use your own sound clip.
- **Wind Noise Volume Curve:** Set wind noise sound effect's volume affected by the vehicles current speed.
- **Wind Noise Pitch Curve:** Set wind noise sound effect's pitch affected by the vehicles current speed.

Engine Wobble FX Settings:

- **Engine Load Wobble FX Simulation:** This can be set to run randomly or to run always. It will 'shake' the engine load value when gas pedal is just got pressed to simulate racecar like rpm reaction. It is also good for simulating trucks with bigger engines.
- **Wobble Volume Change:** Sets the volume change amount when simulating engine load shake fx.
- **Random Chance:** Increase or decrease the random chance of playing shake fx.
- **Wobble Rate Setting:** The frequency of the shake fx. This can be set to change this value randomly.
- **Wobble FX Lenght:** The longetivity of shake fx. This can be set to change this value randomly.
- **Minimum Car Speed for Wobble FX:** the minimum speed your vehicle needs to reach to start playing shake FX.
- **Minimum Engine RPM for Wobble FX:** the minimum RPM your vehicle needs to reach to start playing shake FX.

Valves SFX Settings:

- **Enable Valve SFX:** Valve sound fx will be played as frequently as much rpm has the prefab currently. This sfx is good for a game where you can open the engine hood / bonet, this can simulate those white noises that a running engine has inside the enginebay. Valve SFX pitch does not change with engine RPM. This SFX may generate some GC!
- **Valve Sound FX Position (Optional):** Valve sound fx will be instantiated in this gameobject. Position this gameobject into your vehicles engine bay for more realistic effect. If you leave it empty, Valve FX will be instantiated at engine sound audios position.
- **Valve Audio Clips:** the short audio clips that will be played randomly based on engine rpm and other settings.
- **Valves Audio Minimum Distance:** Within the Minimum distance valve SFX will cease to grow louder in volume.

- **Valves Audio Maximum Distance:** The distance where the sound stops attenuating at. Beyond this point it will stay at the volume it would be at MaxDistance units from the listener and will not attenuate any more.
- **Valve Master Volume:** Sets the maximum volume of all valve sound fx shots.
- **Valves Pitch:** Sets the pitch of all valve sound fx shots.
- **Valves Rate:** Sets the how frequently will be played one shot of a valve sound fx based on engine RPM.
- **Valves Volume by RPM:** Set the valve sound effects volume by the vehicles engine RPM.

Aggressiveness SFX Settings:

- **Enable Aggressiveness SFX:** Aggressiveness sound effect will add an aggressive noise to the engine sound. Good for simulating tuned cars, race cars, open or straight pipe exhaust systems, it also good for giving a stock car sound feeling when aggressiveness SFX is disabled or level reduced to zero. Each engine sound pack has it's own aggressiveness audioclips.

- **Aggressiveness On Load Clip:** On load clip for aggressiveness sfx.
- **Aggressiveness Off Load Clip:** Off load clip for aggressiveness sfx.
- **Aggressiveness Level:** Sets the level / master-volume of aggressiveness sound effect.
- **Rev Limiter Tweaker:** Sets the volume of aggressiveness audio source while rev limiter is played. The higher the value the more it will reduce aggressiveness audio volume.
- **Aggressiveness Volume by RPM:** Set aggressiveness sound effect's volume by the vehicle's engine RPM.
- **Aggressiveness Pitch by RPM:** Set Aggressiveness sound effect's pitch by the vehicle's engine RPM.

Intake SFX Settings:

- **Enable Intake SFX:** Intake sound effect will add extra details to the engine sound when the camera is looking at the front of the car (or at engine bay) and ads details to the interior engine sound too.
- **Intake SFX Volume:** Sets the volume of intake sound effect.
- **Intake On-Load Clip:** On load clip for intake sfx.
- **Intake On-Load Volume Curve:** Sets intake sound effect's on-load volume by the vehicle's engine RPM.
- **Minimum Distance:** Within the Min distance the intake on-load sfx will cease to grow louder in volume.
- **Maximum Distance:** (Logarithmic rolloff) MaxDistance is the distance a sound stops attenuating at.
- **Intake has off-load:** Enable off-load sounds for intake SFX.
- **Intake Off-Load Clip:** On load clip for intake sfx.
- **Intake Off-Load Volume Curve:** Sets intake sound effect's off-load volume by the vehicle's engine RPM.
- **Minimum Distance:** Within the Min distance the intake off-load sfx will cease to grow louder in volume.
- **Maximum Distance:** (Logarithmic rolloff) MaxDistance is the distance a sound stops attenuating at.
- **Intake Open SFX:** Enable intake start SFX which plays an 'intake opening' sound.
- **Intake Opening SFX Volume:** Sets the volume of intake opening sound effect.
- **Minimum Distance:** Within the Min distance the intake opening sfx will cease to grow louder in volume.
- **Maximum Distance:** (Logarithmic rolloff) MaxDistance is the distance a sound stops attenuating at.
- **Intake Start:** Intake opening SFX, which is played once when the gas pedal just got pressed down.

Other Settings:

- **Debug Info:** Show detailed debug values in the inspector. Usefull when tweaking values in new prefabs.
- **Enable Load Details:** Show detailed Engine Load values in the inspector.
- **Show Welcome Information:** Will show the welcome information in the inspector. This can be turned off and it is saved globally for all Unity projects.

Setup for Custom Car Controller

This integration is made for custom car physics controllers, that you made by yourself.
The unitypackage can be found here:
\RealisticEngineSound\Assets_For_Vehicle_Controllers\CustomCarController
CustomCarController_RES2-Lite.unitypackage

It contain a well commented example script for integration (*CustomCarController_RES2.cs*) which is a good quick start to connect your custom car controller with Realistic Engine Sounds 2. You need to modify this script in a way to make it work with your custom car controller. This script will convert values from your car controller to RES2.
You can find this script here after importing *CustomCarController_RES2-Lite.unitypackage*:
\RealisticEngineSound\Assets\Scripts\Demo Scene Scripts\CustomCarController

You need to connect the following values from your car controller to RES2:

car's current engine RPM -> connects to res2.engineCurrentRPM,
maximum engine RPM -> connects to res2.maxRPMLimit,
car's current speed in km/h or mph -> connects to res2.carCurrentSpeed,
car's maximum speed in km/h or mph -> connects to res2.carMaxSpeed,
the current gear of the car's transmission -> connects to currentGear (it is assumed that reverse gear = -1, neutral gear = 0, ... etc),
engine load is managed by the car's gas pedal value, which is usually controlled by *Input.GetAxis("Vertical")*, if you controll it differently then edit the script according to that. 0 = gas pedal is released, more than 0 means gas pedal is pressed down.

You also need to rename CustomCarController to your car controller script's name.

The CustomCarController_RES2.cs script shows exactly how these values should be connected to RES2 values. If everything is done right, RES2 will became compatible with your custom car controller.

The unitypackage also contain demo scenes, where you can test all engine sounds by switching between them with UI buttons. You just need to drag and drop the pre-made gameobject named as "*RES2 High Quality Prefabs - drop this into your car*" into your car, make sure this gameobject is positioned inside your car. If you modified the above mentioned script correctly, your car will have new engine sounds! :)

Setup for Edy's Vehicle Physics

Youtube video url for this tutorial: https://youtu.be/9tGYwWWnooc

Import Edy's Vehicle Physics and Realistic Engine Sounds 2 into your project.
Now import **EVP_RES2-Lite.unitypackage** from the following folder:
../RealisticEngineSound/Assets_For_Vehicle_Controllers/EVP/..

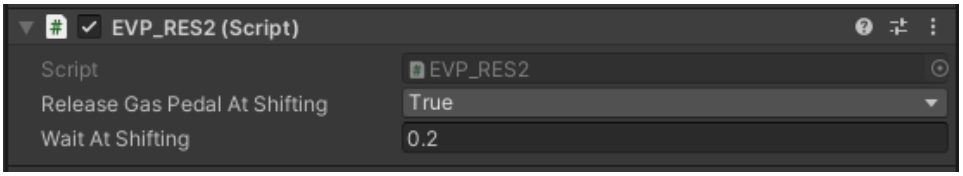
This unitypackage contain a script, a prefab and demo scenes made for Edy's Vehicle Physics compatibility.

Drag and drop "EVP to RES2" prefab into an EVP vehicle then drag and drop engine sound prefab into this prefab. You can add one prefab or two prefabs (exterior and interior prefabs).

Increase exterior engine sound prefab's „Minimum Distance" value if you want louder engine sounds, it should be equal to the vehicle's size (watch the video tutorial). Modify the prefabs position as seen in the video tutorial, move the exterior prefab to a position where the vehicle will be at the center of „Minimum Distance"'s sphere, and move the interior prefab where the vehicle's engine is positioned in reality.

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 load sounds for a short time (for „Wait At Shifting" time).
- **Wait At Shifting:** if Release Gas Pedal At Shifting is set to true, it will play off load engine sounds for X seconds. In this example off engine sounds will be played for 0.2 seconds.

Youtube video url for this tutorial: https://youtu.be/YxhmxsdUWmI

Import Realistic Car Controller and Realistic Engine Sounds 2 into your project.
Now import **RCC_RES2-Lite.unitypackage** from the following folder:
../RealisticEngineSound/Assets_For_Vehicle_Controllers/RCC/..
This unitypackage contain a script, a prefab and demo scenes made for Realistic Car Controller compatibility.

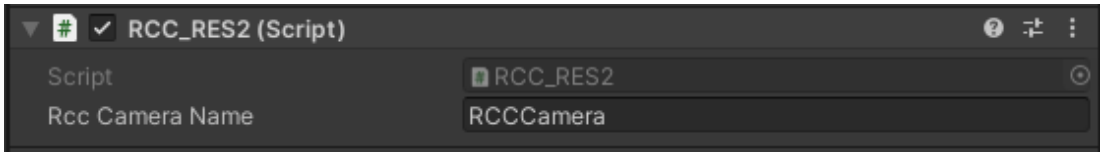
Drag and drop "RCC to RES2" prefab into an RCC vehicle then drag and drop engine sound prefab into this prefab. You can add one prefab or two prefabs (exterior and interior prefabs).

Sometimes you need to set RCC car's stock engine sound to „Off" in advance before entering play mode.

Increase exterior engine sound prefab's „Minimum Distance" value if you want louder engine sounds, it should be equal to the vehicle's size (watch the video tutorial). Modify the prefabs position as seen in the video tutorial, move the exterior prefab to a position where the vehicle will be at the center of „Minimum Distance"'s sphere, and move the interior prefab where the vehicle's engine is positioned in reality.

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

Edy's to RES controller script settings:



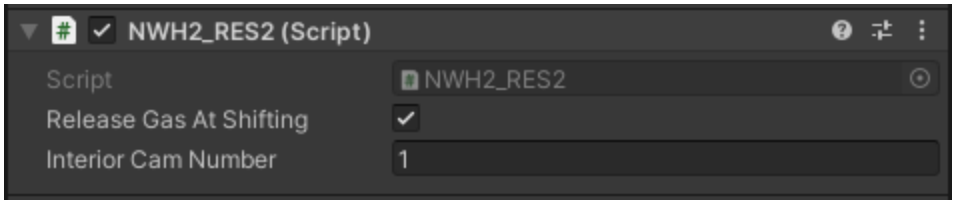
- **RCC Camera Name:** your RCC camera's name in your scene.

Youtube video url for this tutorial: https://youtu.be/yNZoYj975rk

Import NWH2 and Realistic Engine Sounds 2 into your project.
Now import **NWH2_RES2-Lite.unitypackage** from the following folder:
../RealisticEngineSound/Assets_For_Vehicle_Controllers/NWH2/..
This unitypackage contain a script, a prefab and demo scenes made for NWH2 compatibility.

Drag and drop "NWH2 to RES2" prefab into an NWH2 vehicle then drag and drop engine sound prefab into this prefab. You can add one prefab or two prefabs (exterior and interior prefabs). Change „Interior Cam Number" to your vehicle's interior camera's order number (starting from 0).

Increase exterior engine sound prefab's „Minimum Distance" value if you want louder engine sounds, it should be equal to the vehicle's size (watch the video tutorial). Modify the prefabs position as seen in the video tutorial, move the exterior prefab to a position where the vehicle will be at the center of „Minimum Distance"'s sphere, and move the interior prefab where the vehicle's engine is positioned in reality.



- **Release Gas At Shifting:** if set to true it will release gas pedal at shifting, and it will play off load sounds while the car is changing gears.
- **Interior Cam Number:** your vehicle's interior camera order number (starting from 0).

Youtube video url for this tutorial: https://youtu.be/uKWra_PpT_4

Import Vehicle Physics Pro and Realistic Engine Sounds 2 into your project.
Now import **VPP_RES2-Lite.unitypackage** from the following folder:
../RealisticEngineSound/Assets_For_Vehicle_Controllers/VPP/..

This unitypackage contain a script, a prefab and demo scenes made for Vehicle Physics Pro compatibility.

Drag and drop "VPP to RES2" prefab into a VPP vehicle then drag and drop engine sound prefab into this prefab. You can add one prefab or two prefabs (exterior and interior prefabs).

Increase exterior engine sound prefab's „Minimum Distance" value if you want louder engine sounds, it should be equal to the vehicle's size (watch the video tutorial). Modify the prefabs position as seen in the video tutorial, move the exterior prefab to a position where the vehicle will be at the center of „Minimum Distance"'s sphere, and move the interior prefab where the vehicle's engine is positioned in reality.

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

- **Max RPM:** your vehicle's maximum RPM, rev limiter.
- **Max Speed:** your vehicle's maximum speed.
- **Camera Gameobject Name:** the name of the camera gameobject.

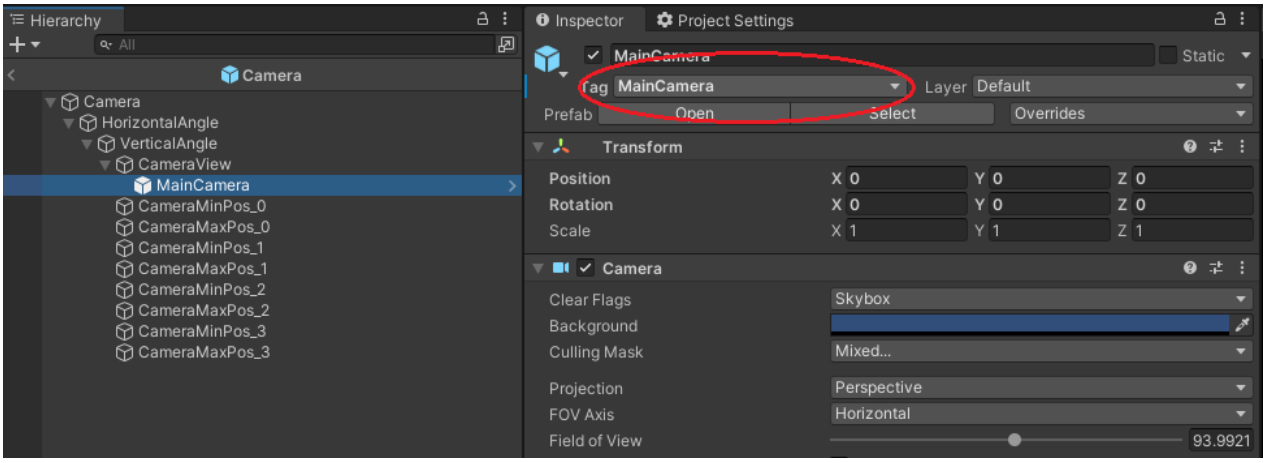
Setup for Universal Vehicle Controller

Youtube video url for this tutorial: https://youtu.be/xGcaf-CBPVY

Import Universal Vehicle Controller and Realistic Engine Sounds 2 into your project.
Now import **UVC_RES2-Lite.unitypackage** from the following folder:
../RealisticEngineSound/Assets_For_Vehicle_Controllers/UVC/..
(or import **UVC_RES2-Lite_NewInputSystem.unitypackage** for the new input system).

This unitypackage contain a script, a prefab and demo scenes made for Universal Vehicle Controller compatibility.

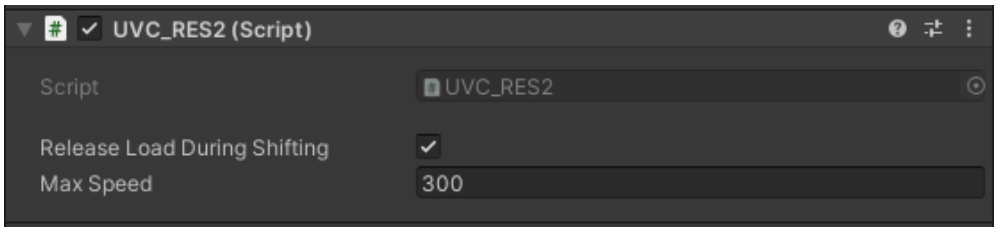
Make sure that the camera gameobject inside UVC's Camera prefab is tagged as „MainCamera". You don't need to tag the whole prefab, just the camera gameobject which is named as "MainCamera".



Drag and drop "UVC to RES2" prefab into an UVP vehicle then drag and drop engine sound prefab into this prefab. You can add one prefab or two prefabs (exterior and interior prefabs).

Increase exterior engine sound prefab's „Minimum Distance" value if you want louder engine sounds, it should be equal to the vehicle's size (watch the video tutorial). Modify the prefabs position as seen in the video tutorial, move the exterior prefab to a position where the vehicle will be at the center of „Minimum Distance"'s sphere, and move the interior prefab where the vehicle's engine is positioned in reality.

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



- **Release Load During Shifting:** if set to true it will release gas pedal at shifting, and it will play off load sounds while the car is changing gears.
- **Max Speed:** the maximum speed your vehicle can reach.

Setup for FGear Vehicle Physics

Youtube video url for this tutorial: https://youtu.be/cBMP8u8LP1c

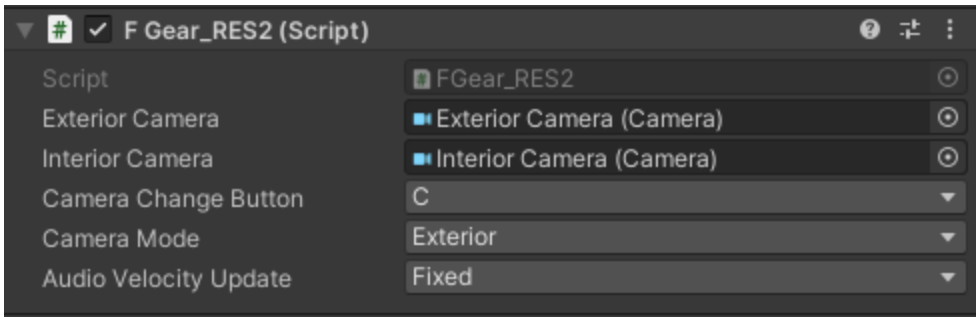
Import FGear Vehicle Physics and Realistic Engine Sounds 2 into your project. Now **FGear_RES2-Lite.unitypackage** from the following folder:
../RealisticEngineSound/Assets_For_Vehicle_Controllers/FGear/..

This unitypackage contain demo scenes, a script and a prefab made for FGear compatibility.

Drag and drop „FGear to RES2" prefab into an FGear vehicle, then drop RES2 engine sound prefab(s) into this prefab. When adding both engine sound prefabs made for exterior and interior views the first one is always the exterior prefab. For this you also need to add cameras for exterior and interior views. The script will automatically switch between camera views with the choosen key – by default it is „C" key. You can even add only just one engine sound prefab if you need it like that, in this case you do not need to add an interior camera to the prefab. You also need to **disable stock FGear engine sounds**. To disable stock FGear engine sounds, in FGear car's "Effects" component set "Engine High Clip" and "Engine Low Clip" to "None".

Increase exterior engine sound prefab's „Minimum Distance" value if you want louder engine sounds, it should be equal to the vehicle's size (watch the video tutorial). Modify the prefabs position as seen in the video tutorial, move the exterior prefab to a position where the vehicle will be at the center of „Minimum Distance"'s sphere, and move the interior prefab where the vehicle's engine is positioned in reality.

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



- **Exterior Camera:** FGear's stock exterior camera.
- **Interior Camera:** interior camera, need to be added by yourself for your vehicle.
- **Camera Change Button:** camera can be changed by this key.
- **Camera Mode:** currently used camera, changing this value on runtime will change the camera too. If you change this to „Interior" mode in advance before playing your scene, enable your interior camera, and disable in FGear's camera gameobject the „Camera" and „Audio Listener" components.
- **Audio Velocity Update:** changing this can solve audio stuttering problems with engine sounds, different FGear vehicle types may need different Audio Velocity Update mode.

Setup for IRDS

Youtube video url for this tutorial: https://youtu.be/dNjE0jndD8Q

Import IRDS and Realistic Engine Sounds 2 into your project.
Now import **IRDS_RES2-Lite.unitypackage** for the source edition of IRDS,
or import **IRDS_dll_RES2-Lite.unitypackage** for IRDS closed source edition from the following folder:
../RealisticEngineSound/Assets_For_Vehicle_Controllers/IRDS/..

This unitypackage contain a script, a prefab and demo scenes made for IRDS compatibility.

Drag and drop "IRDS to RES2" prefab into an IRDS vehicle then drag and drop engine sound prefab into this prefab. You can add one prefab or both prefabs (exterior and interior prefabs of the same engine sound pack). When adding both engine sound prefabs made for exterior and interior views the first one is always the exterior prefab.

Increase exterior engine sound prefab's „Minimum Distance" value if you want louder engine sounds, it should be equal to the vehicle's size. Move the exterior prefab to the center of the car, and move the interior prefab where the vehicle's engine is positioned in reality.

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

Youtube video url for this tutorial: https://youtu.be/jyxlAzhVfB8

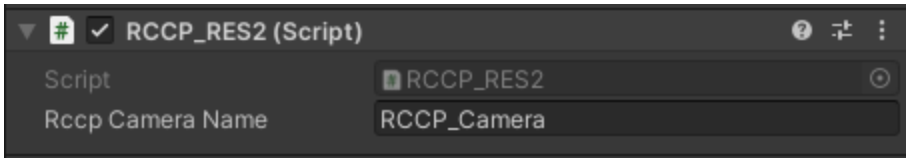
Import Realistic Car Controller Pro and Realistic Engine Sounds 2 into your project.
Now import **RCCP_RES2-Lite.unitypackage** from the following folder:
../RealisticEngineSound/Assets_For_Vehicle_Controllers/RCCP/..
This unitypackage contain a script, a prefab and demo scenes made for Realistic Car Controller Pro compatibility.

Drag and drop "RCCP to RES2" prefab into an RCCP vehicle then drag and drop engine sound prefab into this prefab. You can add one prefab or both prefabs (exterior and interior prefabs of the same engine sound pack). When adding both engine sound prefabs made for exterior and interior views, the first one is always the exterior prefab.

For louder engine sound, increase exterior engine sound prefab's „Minimum Distance" value if you want louder engine sounds, it should be equal to the vehicle's size. Move the exterior engine prefab to a position where the vehicle will be at the center of „Minimum Distance"'s sphere, and move the interior prefab where the vehicle's engine is positioned in reality.

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

RCCP to RES2 controller script settings:



- **RCCP Camera Name:** your RCCP camera's name in your scene.

Setup for Arcade Car Controller

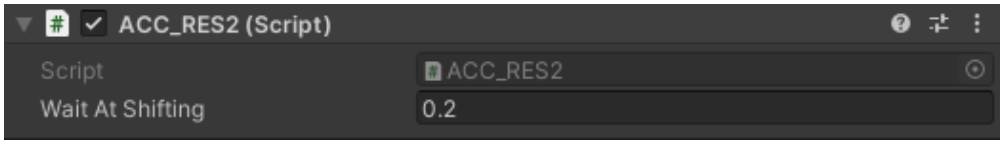
Youtube video url for this tutorial: https://youtu.be/4WuvH5wtuF4

Import Arcade Car Controller and Realistic Engine Sounds 2 into your project.
Now import **ACC_RES2-Lite.unitypackage** from the following folder:
../RealisticEngineSound/Assets_For_Vehicle_Controllers/ACC/..
This unitypackage contain a script, a prefab and demo scenes made for Arcade Car Controller compatibility.

Drag and drop "ACC to RES2" prefab into an ACC vehicle then drag and drop engine sound prefab into this prefab.

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

ACC to RES2 controller script settings:



- **Wait At Shifting:** during gear changes it will be waited for this long time in secconds before the engine sound starts raising it's engine load value.

Setup for Realistic Traffic Controller

Youtube video url for this tutorial: https://youtu.be/gZA57Jza8I8

Import Realistic Traffic Controller and Realistic Engine Sounds 2 into your project.
Now import **RTC_RES2-Lite.unitypackage** from the following folder:
../RealisticEngineSound/Assets_For_Vehicle_Controllers/RTC/..
This unitypackage contain a script, a prefab and demo scenes made for Realistic Traffic Controller compatibility.

Drag and drop "RTC to RES2" prefab into RTC vehicle prefab then drag and drop one RES2 engine sound prefab into this prefab. Repeate this step for all RTC vehicle prefabs.

That's all, now your Realistic Traffic Controller vehicles have cool engine sounds! Enjoy it! ☺

Setup for Realistic Car Kit

Youtube video url for this tutorial: https://youtu.be/MgqeOtmpHBg

Import Realistic Car Kit and Realistic Engine Sounds 2 into your project.
Now import **RCK_RES2-Lite.unitypackage** from the following folder:
../RealisticEngineSound/Assets_For_Vehicle_Controllers/RCK/..
This unitypackage contain a script, a prefab and demo scenes made for Realistic Car Kit compatibility.

Drag and drop "RCK to RES2" prefab into an RCK vehicle then drag and drop an engine sound prefab into this prefab. You can add one prefab or both prefabs (exterior and interior prefabs of the same engine sound pack). When adding both engine sound prefabs made for exterior and interior views, the first one is always the exterior prefab.

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

Setup for Sim-Cade Vehicle Physics

Youtube video url for this tutorial: https://youtu.be/JgnO_TkJrrw

Import Sim-Cade Vehicle Physics and Realistic Engine Sounds 2 into your project.
Now import **SCVP_RES2-Lite.unitypackage** from the following folder:
../RealisticEngineSound/Assets_For_Vehicle_Controllers/SCVP/..
This unitypackage contain a script, a prefab and demo scenes made for Sim-Cade Vehicle Physics compatibility.

Drag and drop "SCVP to RES2" prefab into an SCVP vehicle then drag and drop an engine sound prefab into this prefab.

That's all, now your Sim-Cade Vehicle Physics vehicle have a cool engine sound! Enjoy it! ☺

Setup for Simple Motorcycle Physics Pro

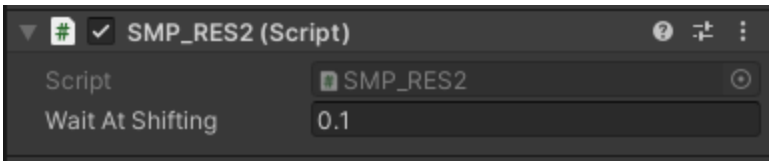
Youtube video url for this tutorial: https://www.youtube.com/watch?v=tjFs3ewi1H4

Import Simple Motorcycle Physics Pro and Realistic Engine Sounds 2 into your project.
Now import **SMP_RES2-Lite.unitypackage** from the following folder:
../RealisticEngineSound/Assets_For_Vehicle_Controllers/SMP/..
This unitypackage contain a script, a prefab and demo scenes made for Simple Motorcycle Physics Pro compatibility.

Drag and drop "SMP to RES2" prefab into an SMP motorbike, then drag and drop an engine sound prefab into this prefab.

That's all, now your Simple Motorcycle Physics Pro motorbike have a cool engine sound! Enjoy it! ☺

SMP to RES2 controller script settings:



- **Wait At Shifting:** during gear changes it will be waited for this long time in secconds before the engine sound starts raising it's engine load value.

Note: currently Simple Motorcycle Physics Pro's own engine RPM simulation is not great, but I tried to improve it with the compatibility script.

Now it's much better than SMP asset's stock RPM simulation, but it's still not perfect.

I will update the compatibility for SMP when the engine RPM simulation gets improved in SMP.

Setup for Simple Motocross Physics

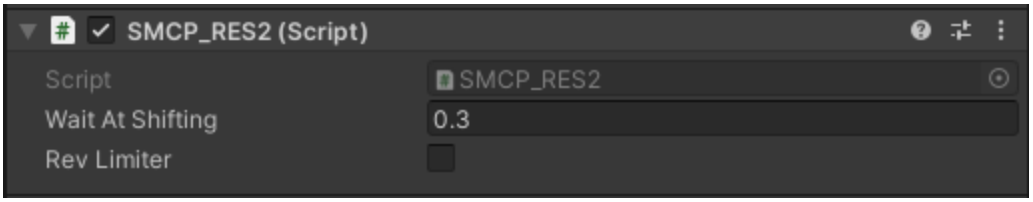
Youtube video url for this tutorial: https://www.youtube.com/watch?v=u8H49MROWbo

Import Simple Motocross Physics and Realistic Engine Sounds 2 into your project.
Now import **SMCP_RES2-Lite.unitypackage** from the following folder:
../RealisticEngineSound/Assets_For_Vehicle_Controllers/SMCP/..
This unitypackage contain a script, a prefab and demo scenes made for Simple Motocross Physics compatibility.

Drag and drop "SMCP to RES2" prefab into an SMCP motorbike, then drag and drop an engine sound prefab into this prefab.

That's all, now your Simple Motocross Physics motorbike have a cool engine sound! Enjoy it! ☺

SMCP to RES2 controller script settings:



- **Wait At Shifting:** during gear changes it will be waited for this long time in secconds before the engine sound starts raising it's engine load value.
- **Rev Limiter:** disables the rev limiter SFX inside the engine sound prefab

Note: currently Simple Motocross Physics own engine RPM simulation is not perfect, but I tried to improve it within the compatibility script.

I will update the compatibility for SMCP when the engine RPM simulation gets improved in SMCP.

Setup for Motorbike Physics Tool

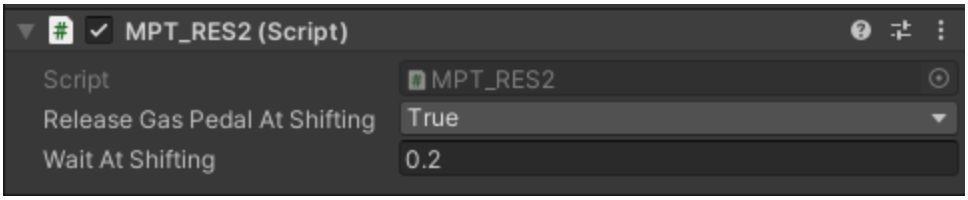
Youtube video url for this tutorial: https://youtu.be/ep_2lortpNg

Import Motorbike Physics Tool and Realistic Engine Sounds 2 into your project.
Now import **MPT_RES2-Lite.unitypackage** from the following folder:
../RealisticEngineSound/Assets_For_Vehicle_Controllers/MPT/..
This unitypackage contain a script, a prefab and demo scenes made for Motorbike Physics Tool compatibility.

Drag and drop "MPT to RES2" prefab into an MPT vehicle then drag and drop an engine sound prefab into this prefab.

That's all, now your Motorbike Physics Tool vehicle have a cool engine sound! Enjoy it! ☺

MPT to RES2 controller script settings:



- **Release Gas Pedal At Shifting:** if set to true it will release gas pedal at shifting, and it will play off load sounds while the car is changing gears.
- **Wait At Shifting:** during gear changes it will be waited for this long time in secconds before the engine sound starts raising it's engine load value.

Setup for Highroad Engine

Youtube video url for this tutorial: https://www.youtube.com/watch?v=fNO7J_j8VzU

Import Highroad Engine and Realistic Engine Sounds 2 into your project.
Now import **HE_RES2-Lite.unitypackage** from the following folder:
../RealisticEngineSound/Assets_For_Vehicle_Controllers/HE/..
This unitypackage contain a script, a prefab and demo scenes made for Highroad Engine compatibility.

Drag and drop "HE to RES2" prefab into HE vehicle then drag and drop engine sound prefab into this prefab.

If the cars are too loud for you, edit the *minDistance* value in the *HE_RES2.cs* script at line 40 from 100 to 50 or 25 value.

That's all, now your Highroad Engine vehicle have a cool engine sound! Enjoy it! ☺

Setup for Arcade Bike Physics Pro

Youtube video url for this tutorial: https://youtu.be/v50ScuiKu9k

Import Arcade Bike Physics Pro and Realistic Engine Sounds 2 into your project.
Now import **ABPP_RES2-Lite.unitypackage** from the following folder:
../RealisticEngineSound/Assets_For_Vehicle_Controllers/ABPP/..
This unitypackage contain a script, a prefab and demo scenes made for Arcade Bike Physics Pro compatibility.

Drag and drop "ABPP To RES2" prefab into ABPP bike then drag and drop an engine sound prefab into this prefab.

That's all, now your Arcade Bike Physics Pro bike have a cool engine sound! Enjoy it! ☺

Setup for Racing Game Starter Kit

Youtube video url for this tutorial: https://youtu.be/czPZEK7mmPM

Import Racing Game Starter Kit and Realistic Engine Sounds 2 into your project.
Now import **RGSK_RES2-Lite.unitypackage** from the following folder:
../RealisticEngineSound/Assets_For_Vehicle_Controllers/RGSK/..
This unitypackage contain a script, a prefab and demo scenes made for Racing Game Starter Kit compatibility.

Drag and drop "RGSK To RES2" prefab into RGSK vehicle then drag and drop an engine sound prefab into this prefab. Stock RGSK engine sound needs to be disabled by hand in car prefab, look for 'VehicleController' script in your car prefab, then open **Sound** tab and set **"Engine Sound"** to **"None (Audio Clip)"**.

That's all, now your Racing Game Starter Kit vehicle have a cool engine sound! Enjoy it! ☺

Setup for ASH Vehicle Physics

Youtube video url for this tutorial: https://www.youtube.com/watch?v=8FpCjfcdOIw

Import ASH Vehicle Physics and Realistic Engine Sounds 2 into your project.
Now import **ASHVP_RES2-Lite.unitypackage** from the following folder:
../RealisticEngineSound/Assets_For_Vehicle_Controllers/ASHVP/..
This unitypackage contain a script, a prefab and demo scenes made for ASH Vehicle Physics compatibility.

Drag and drop "ASHVP To RES2" prefab into ASHVP car then drag and drop an engine sound prefab into this prefab.

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

Setup for Ushi Arcade Car Physics

Youtube video url for this tutorial: https://youtu.be/CObxU79htFU

Import Ushi Arcade Car Physics and Realistic Engine Sounds 2 into your project.
Now import **UACP_RES2-Lite.unitypackage** from the following folder:
../RealisticEngineSound/Assets_For_Vehicle_Controllers/UACP/..
This unitypackage contain a script, a prefab and demo scenes made for Ushi Arcade Car Physics compatibility.

Drag and drop "UACP To RES2" prefab into UACP car then drag and drop an engine sound prefab into this prefab.

That's all, now your Ushi Arcade Car Physics car have a cool engine sound! Enjoy it! ☺

Setup for Semi-Realistic Car Physics

Youtube video url for this tutorial: https://youtu.be/FDcLiYvo_X8

Import Semi-Realistic Car Physics and Realistic Engine Sounds 2 into your project.
Now import **SRCP_RES2-Lite.unitypackage** from the following folder:
../RealisticEngineSound/Assets_For_Vehicle_Controllers/SRCP/..
This unitypackage contain a script, a prefab and demo scenes made for Semi-Realistic Car Physics compatibility.

Drag and drop "SRCP To RES2" prefab into a SRCP car then drag and drop an engine sound prefab into this prefab.

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

Setup for Arcade Vehicle Physics

Youtube video url for this tutorial: https://youtu.be/L3U1O0usJOE

Import Arcade Vehicle Physics and Realistic Engine Sounds 2 into your project.
Now import **AVP_RES2-Lite.unitypackage** from the following folder:
../RealisticEngineSound/Assets_For_Vehicle_Controllers/AVP/..
This unitypackage contain a script, a prefab and demo scenes made for Arcade Vehicle Physics compatibility.

Drag and drop "AVP To RES2" prefab into an AVP car then drag and drop an engine sound prefab into this prefab.

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

Setup for Quick And Easy Car Racing

Youtube video url for this tutorial: https://youtu.be/uCEkbiokYeQ

Import Quick And Easy Car Racing and Realistic Engine Sounds 2 into your project.
Now import **QECR_RES2-Lite.unitypackage** from the following folder:
../RealisticEngineSound/Assets_For_Vehicle_Controllers/QECR/..
This unitypackage contain a script, a prefab and demo scenes made for Quick And Easy Car Racing compatibility.

Drag and drop "QECR To RES2" prefab into a QECR car prefabs "Vehicle_sc" gameobject, then drag and drop an engine sound prefab into this prefab. Repeat these steps for all vehicle prefabs.

That's all, now your Quick And Easy Car Racing car have a cool engine sound! Enjoy it! ☺

Setup for Multiversal Vehicle Controller Pro

Youtube video url for this tutorial: https://youtu.be/9IfhpC6yUQo

Import Multiversal Vehicle Controller Pro and Realistic Engine Sounds 2 into your project.
Now import **MVC_RES2-Lite.unitypackage** from the following folder:
../RealisticEngineSound/Assets_For_Vehicle_Controllers/MVC/..
This unitypackage contain a script, a prefab and demo scenes made for Multiversal Vehicle Controller Pro compatibility.

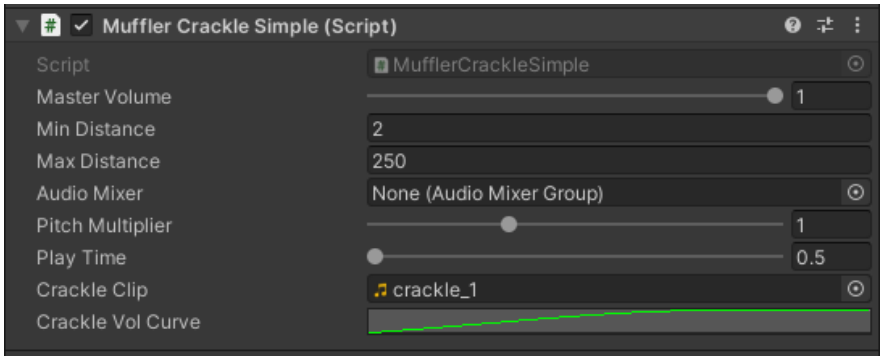
Drag and drop "MVC To RES2" prefab into a MVC car then drag and drop an engine sound prefab into this prefab. You may also need to mute the stock engine sound's audio mixer for the best experience (inside Unity Engine's Audio Mixer window).

That's all, now your Multiversal Vehicle Controller Pro car have a cool engine sound! Enjoy it! ☺

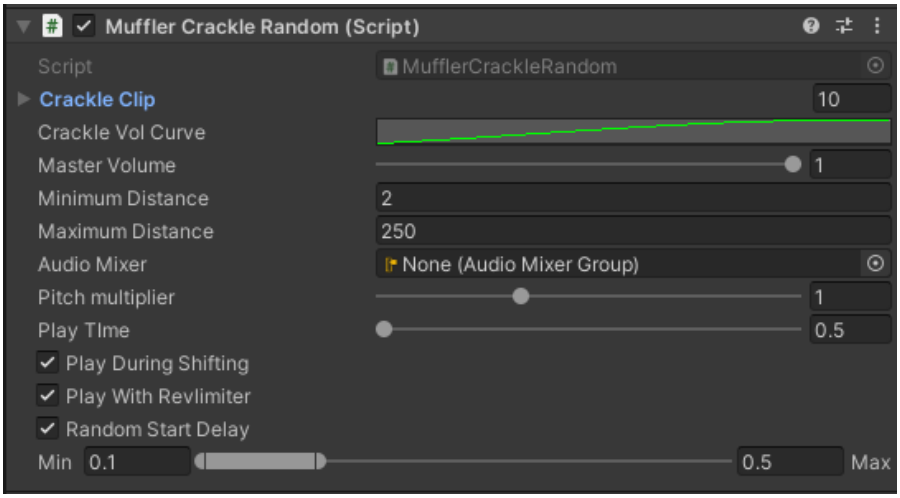
Muffler Crackle Sound

Muffler crackle prefabs will add a muffler rumble effect to your car sound which will be played when the gas pedal just got released.

Non random script:



Random sound clips script:



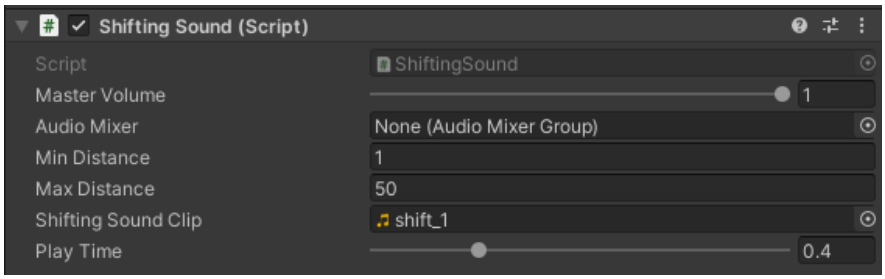
- **Master Volume**: Sets the gameobject's maximum volume.
- **Min Distance**: Within the MinDistance, the sound will stay at loudest possible. Outside MinDistance it will begin to attenuate. Increase the MinDistance of a sound to make it 'louder' in a 3d world, and decrease it to make it 'quieter' in a 3d world.
- **Max Distance**: The distance where the sound stops attenuating at.
- **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).
- **Crackle Clips**: Sound clip played for „Play Time" secconds when gas pedal is released.
- **Crackle Vol Curve**: Volume settings for muffler off clips.
- **Pitch Multiplier**: Multipli the audio source's pitch value.
- **Play Time:** The time for sound clips are played in secconds.
- **Play During Shifting**: Disabling this will play crackle sounds only when gas pedal is released.
- **Play with rev limiter:** Crackle soudn will be played when the car's rpm reached the rev limiter.
- **Random Start Delay:** Will wait for a random time based ont he values before playing the sound clips.

Drag and drop this prefab on your RES2 engine prefab.

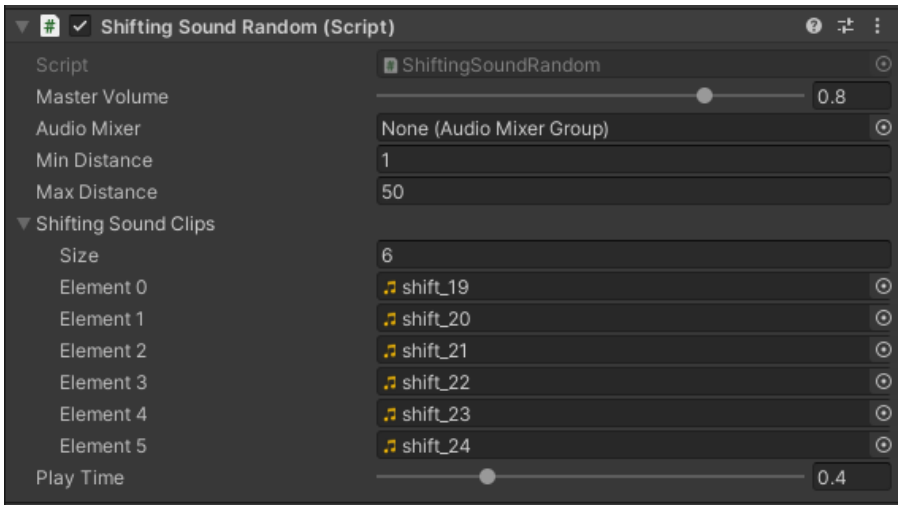
Shifting Sounds

Shifting add a gear changing sound that can be heard inside the vehicle.

Non random script:



Random sound clips script:

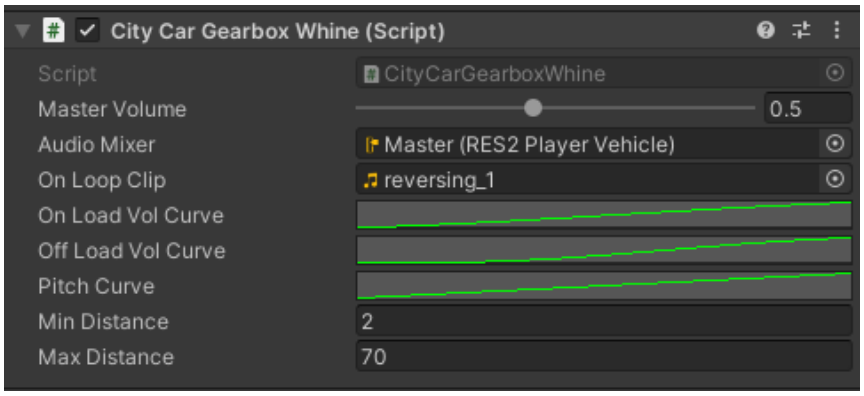


- **Master Volume**: sets the gameobject's maximum volume.
- **Audio Mixer**: add an audio mixer here to mix the sound of gear changin sounds separatelly. If set to none, it will use the engine prefabs's audio mixer (if the engine prefab have an audio mixer).
- **Min Distance:** Within the Min distance the sound will cease to grow louder in volume.
- **Max Distance:** MaxDistance is the distance a sound stops attenuating at.
- **Shifting sound clips:** the sound clip which is played when car is changing gears.
- **Play Time:** for how long the sound will be played.

Drag and drop this prefab on your RES2 engine prefab.

City Car Gearbox Whine SFX

Used to simulate regular city cars gearbox whine, which can be heard at higher speeds - mostly heard in older cars but some affordable newer cars produce this noise too.



- **Master Volume**: Sets the AudioSource's maximum volume.
- **Audio Mixer**: Add an audio mixer here to mix the gearbox whine sounds separately. If set to None, it will use the engine prefab's audio mixer (if the engine prefab has one).
- **On Loop Clip**: Looping gearbox sound clip.
- **On Vol Curve:** Sets the on load volume for gearbox sound clip.
- **Off Vol Curve:** Sets the off load volume for gearbox sound clip.
- **Pitch Curve:** Sets the pitch value gearbox sound clip.
- **Min Distance:** Within the Min distance the sound will cease to grow louder in volume.
- **Max Distance:** MaxDistance is the distance a sound stops attenuating at.

Drag and drop this prefab on your RES2 engine prefab.

Tips, Tricks and Optimisations

- For player vehicle use „High Quality" prefabs and use „Low Quality" prefabs for opponents.
- In scenes with more vehicles try reducing „Max Distance" in RES prefabs to reduce the number of currently playing 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 and may cause click like noises.
- Increase „Aggressiveness" SFX level to mimic engine tuning, open or straight pipe exhaust systems or to mimic a fallen off exhaust after a big car crash.
- Increase pitch value (for examlpe) to 1.1f to mimic a tuned engine or an engine upgrade.
- In an open world game switch betveen „Player" and „Opponent" inside the engine prefab when player enter or exit a vehicle when you are using the „Dynamic Audio Mixer".
- Delete those engine sound packages that you may never use in your game to reduce your game's 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).

License

Refer to the license from the store where you purchased this asset.

Credits

I would like to say a big ***Thank You*** to my friends and customers who allowed me to record their vehicles, or who sent me their own recordings of vehicle engine sounds and other sounds.

Submit your own recordings to me, and I will transform them into complete game ready engine sound packs. Please ensure that your recordings are of good quality, if possible with minimal wind and other unwanted background noises (such as crowds, people talking, etc.). The newly created sound packs might be featured in upcoming add-on sound packs. As a token of appreciation, you will receive a free copy of the final product if your donated recordings are utilized in the creation of new sound pack(s).

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

Youtube playlist with tutorials and sample videos:
https://youtube.com/playlist?list=PLYFtdNoo8S3hnyyBGM07cVBAQlo-1hJrg

Please write a review for this asset in the Unity Asset Store. Thank you!

Add-on engine sound packs for Realistic Engine Sounds 2: https://skrilstudio.com/s/addons

Realistic Car Shaders - URP: https://skrilstudio.com/s/rcsurp

UI SFX Mega Pack (2682 wav files): https://bit.ly/40oKeCd

Check out my other assets: https://skrilstudio.com/s/more

Own the Lite Edition? Upgrade to Plus or Pro at a discount: https://skrilstudio.com/s/res2all

Made by: **Skril Studio**
Contact: *info@skrilstudio.com*
Website: *https://skrilstudio.com*
Facebook Page: *https://www.facebook.com/skrilstudio*

May not be available to respond to messages during weekends and holidays.