

A quick overview of the sound engine

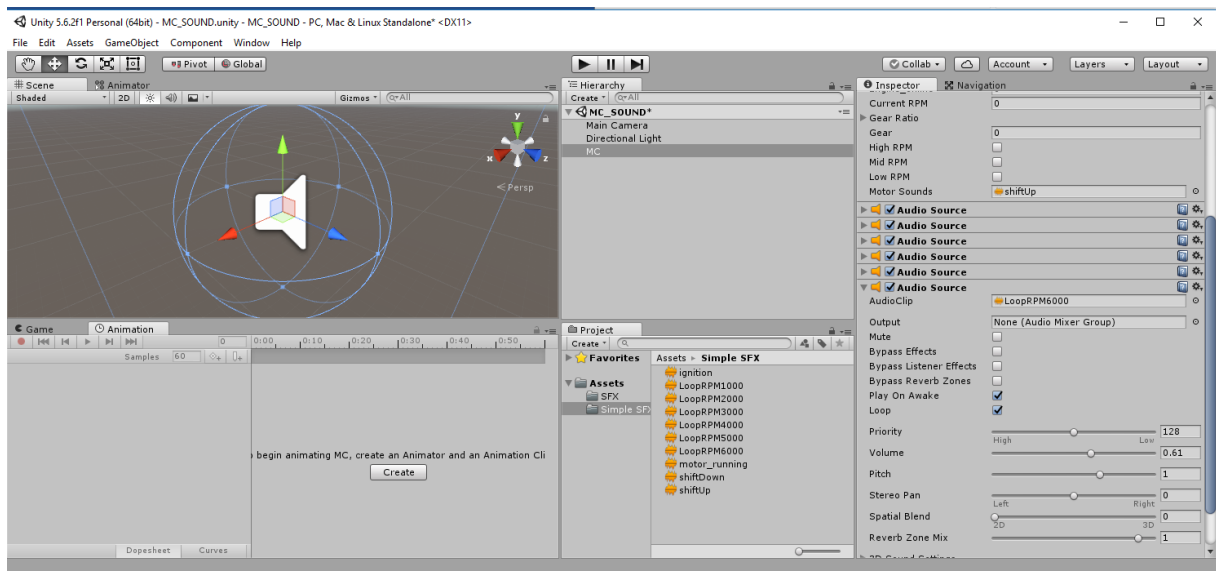
By Seid Saleh

Sound in Unity

If the sound is chosen to play as a 3D sound (see the Spatial Blend bar) it will have a varying volume on the right or left ear depending on the Audio Listener which acts as the players head. If one wants to avoid the trouble of positioning the listener the spatial sound can be set to 2D, volume will now be constant.

For sounds in Unity to play and AudioSource component needs to be attached to the game object. The Audio Source component can then play the AudioClip it's attached to, which is done by dragging and dropping and audio file from the assets folder to the Audio Source. Unity allows Audio Clips to be changed in game and for the motor this means that the pitch has been increased depending on RPM and gear ratio.

For a sound to be played and looped check the "Loop" checkbox and play the sound once with `AudioSource.Play()`. To end the loop use `AudioSource.Stop()`.

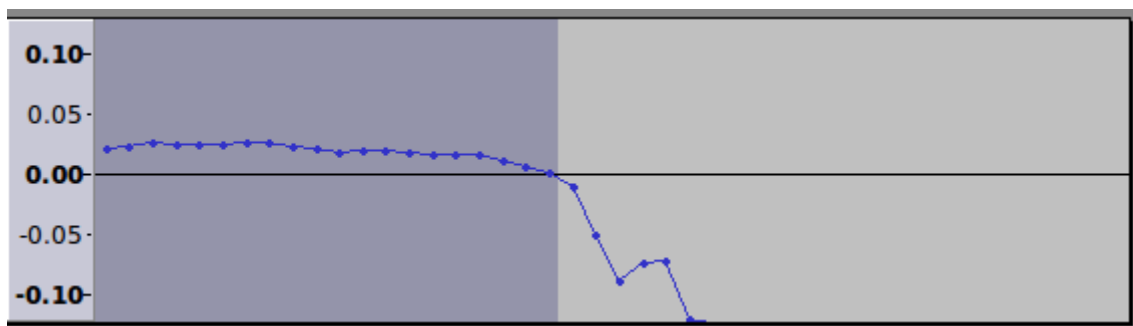


Spatial Blend, Volume and pitch on the bottom right for the 6000 RPM loop. Changes can be made in code for runtime manipulation

The loops

For the sound loops to work in well Unity use the wav files (uncompressed) with the highest possible resolution (32 bit). While the files are larger the quality change is noticeable. For some reason sounds also loop better in void Fixed Update() and can be harmed by the Unity importation.

The motor has 3 RPM loops that are manipulated to provide sounds for low RPM, medium RPM and high RPM. More loops for a better experience can be added and all the loops has been manipulated to start at a zero-crossing point (silence or 0db) or very close to a zero crossing point.



Marked zero-crossing point in the sound editor Audacity

This means they loop almost seamlessly. The points are sometimes “almost” at the zero-crossing point as stereo sound has been chosen and two crossing point than has to be made instead of one (the loop also has to end at a zero-crossing point with the same direction). While audio clips can be forced to be made Mono at have true zero crossings the Stereo loop was of a high enough quality to use with the added benefit of aural perspective (sound localization for Unity to use 3D sounds properly).

It is recommended to stay within and RPM range of maximum 1000 for the loops. Meaning and RPM at 3000 should be acceptable at 2000 and 4000 with pitch manipulation. This range can however be shortened to 500 at each 1000 RPM from 0 – 6000 as the sounds are available.

Possible improvements

- Add more RPM loops besides the existing 3
- Add dynamic Volume changes, right now there's only 3 different volumes that are static for low, medium and high RPM and changes in between can sound rough
- Test RPM sounds and tune the pitch changes if needed
- Change gear change to a flat pitch increase

Currently the pitch is chosen by this formula:

```
gearRatio = new int[] { 300, 800, 1500, 2500, 3800, 5000, 6000 }; // 6 gears + neutral
```

```
LoopRPM3000.pitch = (currentRPM) / (maxRPM - gearRatio[gear]) + 1;
```

where `gearRatio[gear]` uses the current gear to find a suitable constant pitch change depending on maximum RPM and the current RPM. Changing the gear change to be a flat pitch change is possible but the sound has so far worked okay.

An alternative approach

Instead of recording RPM loops for each new motor cycle one can use procedural audio generation. By simulating the exhaust dynamics any the sound of any motor cycle with any number of pipes could be generated for any RPM. This was considered outside the time scope as it could be made in time but would not sound as good as the RPM loop method and be extremely computationally heavy.