**Background**

Sound is a type of energy made by ***vibrations*.** When an object vibrates, it causes movement in surrounding air molecules. These molecules bump into the molecules close to them, causing them to vibrate as well. This makes them bump into more nearby air molecules. This “chain reaction” movement, called *sound waves*, keeps going until the molecules run out of energy. As a result, there is a series of molecular collisions as the sound wave passes through the air, but the air molecules themselves don’t travel with the wave. As it is disturbed, each molecule just moves away from a resting point but then eventually returns to it.

**Pitch and Frequency**  
If your ear is within range of such vibrations, you hear the sound. However, the vibrations need to be at a certain speed in order for us to hear them. For example, we would not be able to hear the slow vibrations that are made by waving our hands in the air. The slowest vibration human ears can hear is 20 vibrations per second. That would be a very low-pitched sound. The fastest vibration we can hear is 20,000 vibrations per second, which would be a very high-pitched sound. Cats can hear even higher pitches than dogs, and porpoises can hear the fastest vibrations of all (up to 150,000 times per second!). The number of vibrations per second is referred to as an object’s ***frequency***, measured in Hertz (Hz).

***Pitch*** is related to frequency, but they are not exactly the same. Frequency is the scientific measure of pitch. That is, while frequency is objective, pitch is completely subjective. Sound waves themselves do not have pitch; their vibrations can be measured to obtain a frequency, but it takes a human brain to map them to that internal quality of pitch.

The pitch of a sound is largely determined by the mass (weight) of the vibrating object. Generally, the greater the mass, the more slowly it vibrates and the lower the pitch. However, the pitch can be altered by changing the tension or rigidity of the object. For example, a heavy E string on an instrument can be made to sound higher than a thin E string by tightening the tuning pegs, so that there is more tension on the string.

Nearly all objects, when hit, struck, plucked, strummed or somehow disturbed, will vibrate. When these objects vibrate, they tend to vibrate at a particular frequency or set of frequencies. This is known as the *natural frequency* of the object. For example, if you ‘ping’ a glass with your finger, the glass will produce a sound at a pitch that is its natural frequency. It will make this same sound every time. This sound can be changed, however, by altering the vibrating mass of the glass. For example, adding water causes the glass to get heavier (increase in mass) and thus harder to move, so it tends to vibrate more slowly and at a lower pitch.

**What is Sound?**  
When we hear something, we are sensing the vibrations in the air. These vibrations enter the outer ear and cause our eardrums to vibrate (or *oscillate*). Attached to the eardrum are three tiny bones that also vibrate: the *hammer*, the *anvil*, and the *stirrup*. These bones make larger vibrations within the inner ear, essentially amplifying the incoming vibrations before they are picked up by the *auditory nerve*.

Diagram

Description automatically generated

The properties of a sound wave change when it travels through different media: gas (e.g. air), liquid (e.g. water) or solid (e.g. bone). When a wave passes through a denser medium, it goes faster than it does through a less-dense medium. This means that sound travels faster through water than through air, and faster through bone than through water.

When molecules in a medium vibrate, they can move back and forth or up and down. Sound energy causes the molecules to move back and forth in the same direction that the sound is travelling. This is known as a *longitudinal wave*. (*Transverse waves* occur when the molecules vibrate up and down, perpendicular to the direction that the wave travels).

Speaking (as well as hearing) involves vibrations. To speak, we move air past our vocal cords, which makes them vibrate. We change the sounds we make by stretching those vocal cords. When the vocal cords are stretched we make high sounds and when they are loose we make lower sounds. This is known as the pitch of the sound.

The sounds we hear every day are actually collections of simpler sounds. A musical sound is called a *tone*. If we strike a tuning fork, it gives off a pure tone, which is the sound of a single frequency. But if we were to sing or play a note on a trumpet or violin, the result is a combination of one main frequency with other tones. This gives each musical instrument its characteristic sound.

Text

Description automatically generated

<https://www.scienceworld.ca/resource/sound/>

# **Digital Audio Fundamentals**

Digital audio brings analog sounds into a form where they can be stored and manipulated on a computer. Audacity is a software application for editing, mixing, and applying effects to digital audio recordings.

## **Digital Sampling**

All sounds we hear with our ears are pressure waves in air. Starting with Thomas Edison's demonstration of the first phonograph in 1877, it has been possible to capture these pressure waves onto a physical medium and then reproduce them later by regenerating the same pressure waves. Audio pressure waves, or [***waveforms***](https://manual.audacityteam.org/man/glossary.html#waveform), look something like this:

Line chart

Description automatically generated

*Analog* recording media such as a phonograph records and cassette tapes represent the shape of the waveform directly, using the depth of the groove for a record or the amount of magnetization for a tape. Analog recording can reproduce an impressive array of sounds, but it also suffers from problems of noise. Notably, each time an analog recording is copied, more noise is introduced, decreasing the fidelity. This noise can be minimized but not completely eliminated.

*Digital* recording works differently: it *samples* the waveform at evenly-spaced timepoints, representing each sample as a precise number. Digital recordings, whether stored on a compact disc (CD), digital audio tape (DAT), or on a personal computer, do not degrade over time and can be copied perfectly without introducing any additional noise. The following image illustrates a sampled audio waveform:

Chart, line chart

Description automatically generated

Digital audio can be edited and mixed without introducing any additional noise. In addition, many digital effects can be applied to digitized audio recordings, for example, to simulate reverberation, enhance certain frequencies, or change the [***pitch***](https://manual.audacityteam.org/man/glossary.html#pitch).

Audacity's ability to play or record audio directly from your computer depends on your specific computer hardware. Most desktop computers come with a *sound card* with 1/8 inch (3.5mm) jacks for you to plug in a microphone or other source for recording, and speakers or headphones for listening. Many laptop computers have speakers and a microphone built-in. The sound card that comes with most computers is not particularly high quality, in this case you may want to consider using an external USB audio interface. For information on how to set up Audacity for playback and recording, see [Audacity Setup and Configuration](https://manual.audacityteam.org/man/audacity_setup_and_configuration.html).

## **Digital Audio Quality**

The quality of a digital audio recording depends heavily on two factors: the [***sample rate***](https://manual.audacityteam.org/man/glossary.html#sample_rate) and the [***sample format***](https://manual.audacityteam.org/man/glossary.html#sample_format) or bit depth. Increasing the sample rate or the number of [***bits***](https://manual.audacityteam.org/man/glossary.html#bit) in each sample increases the quality of the recording, but also increases the amount of space used by audio files on a computer or disk.

## **Sample rates**

*Sample rates* are measured in hertz ([***Hz***](https://manual.audacityteam.org/man/glossary.html#hz)), or [***cycles***](https://manual.audacityteam.org/man/glossary.html#cycle) per second. This value is the number of samples captured per second in order to represent the waveform. Higher sample rates allow higher audio frequencies to be represented. Provided that the sample rate is more than double the highest audio frequency present, the waveform can be reconstructed exactly from the digital samples. Frequencies that are more than half the sample rate cannot be correctly represented in digital samples, and, if present in the original audio, must be removed before converting to digital. "Half the sample rate" therefore represents an upper limit called the [*Nyquist frequency*](http://en.wikipedia.org/wiki/Nyquist_frequency), and the analog waveform must be entirely below this limit to be correctly represented digitally. Analog frequencies at this limit or above cannot be correctly represented by the digital samples and would cause a kind of distortion called [*aliasing*](http://en.wikipedia.org/wiki/Aliasing).

The human ear is sensitive to sound patterns with frequencies between approximately 20 Hz and 20,000 Hz. Sounds outside that range are inaudible. Therefore a sample rate of 40,000 Hz is the absolute minimum necessary to reproduce the full range of audible sounds. Higher rates (called [*oversampling*](http://en.wikipedia.org/wiki/Oversampling)) are usually used so as to allow adequate filtering to avoid aliasing artifacts around the Nyquist frequency.

The sample rate used by [***audio CDs***](https://manual.audacityteam.org/man/glossary.html#audio_cd) is 44,100 Hz. Human speech is intelligible even if frequencies above 4,000 Hz are eliminated; in fact telephones only transmit frequencies between 200 Hz and 4,000 Hz. Therefore a common sample rate for audio recordings is 8,000 Hz, which is sometimes called *speech quality*. Note that very steep [***filtering***](https://manual.audacityteam.org/man/glossary.html#filter) (called an anti-aliasing filter) is required at the Nyquist frequency in order to prohibit signal above this cutoff point from being folded back into the audible range by the digital converter, and creating the distorting artifacts of aliasing noise.

The most common sample rates measured in Hz are 8,000, 16,000, 22,050, 44,100, 48,000, 96,000 and 192,000. Sample rates can also be referred to in [***kHz***](https://manual.audacityteam.org/man/glossary.html#khz) or units of 1,000 Hz. So in units of kHz the most common rates are expressed as 8 kHz, 16 kHz, 22.05 kHz, 44.1 kHz, 48 kHz, 96 kHz and 192 kHz.

Audacity supports any of these sample rates, however most computer sound cards are limited to no more than 48,000 Hz, 96,000 Hz or sometimes 192000Hz. Again, the most common sample rate *by far* is 44,100 Hz and many cards will thus default to this rate, whatever other rates they support.

In the image below, the left half has a low sample rate, and the right half has a high sample rate (that is, high resolution):

Chart, line chart

Description automatically generated

## **Sample formats**

The other measure of audio quality is the sample format (or *bit depth*), which is usually measured by the number of computer *bits* used to represent each sample. The more bits that are used, the more precise the representation of each sample. Increasing the number of bits also increases the maximum [***dynamic range***](https://manual.audacityteam.org/man/glossary.html#dynamic_range) of the audio recording, in other words the difference in volume between the loudest and softest possible sounds that can be represented.

Dynamic range is measured in decibels ([***dB***](https://manual.audacityteam.org/man/glossary.html#decibel)). The human ear can perceive sounds with a dynamic range of at least 90 dB. However, whenever possible it is a good idea to record digital audio with a dynamic range of far more than 90 dB, in part so that sounds that are too soft can be amplified for maximum fidelity. Note that although signals recorded at generally low levels can be raised (that is, normalized) to take advantage of the available dynamic range, the recording of low level signals will not use all of the available *bit depth*. This loss of resolution cannot be re-captured simply by normalizing the overall level of the digital waveform.

Common sample formats, and their respective dynamic range include:

* 8-bit integer: 48 dB
* 16-bit integer: 96 dB
* 24-bit integer: 145 dB
* 32-bit floating point: near-infinite dB

Note that there are practical limitations on dynamic range due to the capabilities of the hardware and input and output converters. These make the practical limit more like 90 dB for 16-bit.

Other sample formats such as ADPCM approximate 16-bit audio with compressed 4-bit samples. Audacity can import many of these formats, but they are rarely used because of much better newer [***compression***](https://manual.audacityteam.org/man/glossary.html#compressed_format) methods.

Audio CDs and most computer audio file formats use 16-bit integers. Audacity uses 32-bit floating-point samples internally and, if required, converts the sample bit depth when the final mix is exported. Audacity's default sample format during recording can be configured in the [Quality Preferences](https://manual.audacityteam.org/man/quality_preferences.html) or set individually for each track in the [Audio Track Dropdown Menu](https://manual.audacityteam.org/man/audio_track_dropdown_menu.html). During playback, the audio in any tracks that have a different sample format from the project will be [***resampled***](https://manual.audacityteam.org/man/glossary.html#resampling) on the fly using the Real-time Conversion settings in the [Quality Preferences](https://manual.audacityteam.org/man/quality_preferences.html). The High-quality Conversion settings are used when processing, mixing or exporting.

In the image below, the left half has a sample format with few bits, and the right half has a sample format with more bits. If you think of the sample rate as the spacing between vertical gridlines, the sample format is the spacing between horizontal gridlines.

Chart, line chart

Description automatically generated

## **Size of audio files**

Audio files are very large, probably much larger than most files you work with (unless you work with video files). To determine the size of an uncompressed audio file, multiply the sample rate (for example 44100 Hz) by the sample format [***bit rate***](https://manual.audacityteam.org/man/glossary.html#bit_rate) (for example 16-bit) by the number of channels (2 for stereo) by the number of seconds. A completely full 74-minute stereo audio CD takes up over 6 billion bits. Divide this by 8 to get the number of bytes; an audio CD is a little less than 800 megabytes (MB). See [compressed audio](https://manual.audacityteam.org/man/digital_audio.html#compression) below.

## **Clipping**

One limitation of digital audio is that for most purposes it cannot deal with sound pressure waves that exceed the maximum levels it is designed to deal with. When a signal is recorded that exceeds the maximum level of +/-1.0 [***linear***](https://manual.audacityteam.org/man/glossary.html#linear) or 0 [***dB***](https://manual.audacityteam.org/man/glossary.html#decibel), samples outside the range are clipped to the maximum value, like this:

A picture containing boat, different

Description automatically generated

A sound recorded with [***clipping***](https://manual.audacityteam.org/man/glossary.html#clipping) will sound distorted and harsh. While there are some techniques that can eliminate a small amount of noise due to clipping, it is always preferable to avoid clipping while recording. Change the volume on your input source (microphone, cassette player, record player) and set Audacity's input volume control (in [Mixer Toolbar](https://manual.audacityteam.org/man/mixer_toolbar.html)) such that the waveform is as large as possible (for maximum fidelity) without clipping.

Note that at Audacity's default 32-bit float sample format, legitimately captured sample values in excess of the maximum can be *stored* but even if preserved in an [exported](https://manual.audacityteam.org/man/exporting_audio.html) 32-bit float file they will probably still distort on any conventional reproducing equipment. If Audacity encounters legitimate samples above the limit, the [Amplify](https://manual.audacityteam.org/man/amplify.html) effect will show a negative default "Amplification (dB)" value and you may click OK at this setting to reduce the peak amplification to the maximum 0 dB without loss of the original peaks of the waveform.

## **Compressed Audio**

Because digital audio files are so large, reduced sample rates were typically used whenever possible. In 1991, the MP3 (MPEG I, layer 3) standard changed everything. MP3 is a [***lossy***](https://manual.audacityteam.org/man/glossary.html#lossy) compression technique that can dramatically reduce the file size of a digital audio file with surprisingly little effect on the quality. One second of CD-quality audio takes up 1.4 megabits, while a common [***bit rate***](https://manual.audacityteam.org/man/glossary.html#bit_rate) for MP3 files is 128 kbps, which is a compression factor of more than 10x! MP3 works by cleverly "throwing away" details about the audio waveform that humans are not very sensitive to, based on a *psychoacoustic model* of how our ears and brains process sounds. All MP3 files are not created alike; different psychoacoustic models will lead to different amounts of perceived distortion in the audio file.

Audacity as shipped can import and export MP3 files.

With good speakers, most people can hear the difference between a 128 kbps MP3 and an uncompressed audio file from a CD. 256 kbps and 320 kbps MP3 files are more popular among audiophiles who prefer higher quality.

There are many other lossy compressed audio file formats. Audacity fully supports the *[Ogg Vorbis](http://en.wikipedia.org/wiki/Ogg_vorbis)* format, which is similar to MP3 but is a completely open, patent-free standard. Over time the quality of Ogg Vorbis files has come to surpass the quality of MP3, and its format is more extensible so more improvements are possible. Ogg Vorbis is a great choice for your own audio, however the reality is that many more devices such as iPods and other portable audio players support MP3 but not Ogg Vorbis yet.

Other well-known compression methods include ATRAC, used by Sony MiniDisc recorders, Windows Media Audio (WMA), and AAC. Audacity supports more formats by adding the optional [FFmpeg library](https://manual.audacityteam.org/man/faq_installation_and_plug_ins.html" \l "ffdown" \o "FAQ:Installation and Plug-Ins).

## **Lossless Compression**

Lossless compression reduces a file's size with no loss of quality. This seemingly magical method of reducing file sizes can be applied to audio files. While MP3s use lossy compression, newer compression algorithms, such as [***FLAC***](https://manual.audacityteam.org/man/glossary.html#flac) and [***Apple Lossless***](https://manual.audacityteam.org/man/glossary.html#alac) compression, can be used to create lossless compressed audio files.

Such compression basically rewrites the data of the original file in a more efficient way. However, because no quality is lost, the resulting files are typically much larger than image and audio files compressed with lossy compression. For example, a file compressed using lossy compression may be one-tenth the size of the original, while lossless compression is unlikely to produce a file smaller than half of the original size.

Lossless audio formats are most often used for archiving or production purposes, while smaller lossy audio files are typically used on portable players and in other cases where storage space is limited or exact replication of the audio is unnecessary.

<https://manual.audacityteam.org/man/digital_audio.html#:~:text=The%20quality%20of%20a%20digital,on%20a%20computer%20or%20disk>.

Delivery

* To play sound on a digital system, the user needs speakers, or a headset.
* The digital audio file is sent through a digital-to-analog converter (DAC) so that it can be heard.
* Important to test sounds under a variety of different conditions typical for the playback that a user will be using.

Recording

* Microphones translate analog signals into electrical impulses
* An analog-to-digital converter (ADC), converts the electrical impulses to numbers that can be stored, understood, and manipulated by a microprocessor.

Picture2wawes and way more on this site below

<https://www.omnicalculator.com/other/audio-file-size>

some information from html how to handle it

<https://www.w3schools.com/html/html5_audio.asp>

music to unity

# How to Add Background Music into Your Game (No Coding Required).

Today, Sovereign Moon Studios, the makers of your favorite [NoCode game development course](https://www.sovereignmoon.studio/), are excited to bring you a new tutorial that will walk you through the steps of adding background music into your Unity 3D Game Kit game. Adding music into your game’s scenes is easy and my goal for this tutorial is to help you setup background music for your game in about 5 minutes

Let’s jump in!

## **Introduction:**

If you’re new to Unity or 3D Game Kit, at some point you’re going to want to learn how to add music into your game scene. Music can really help you support the emotion you’re trying to convey within your scene. Luckily, within Unity and 3D Game Kit, the process of adding music is really easy.

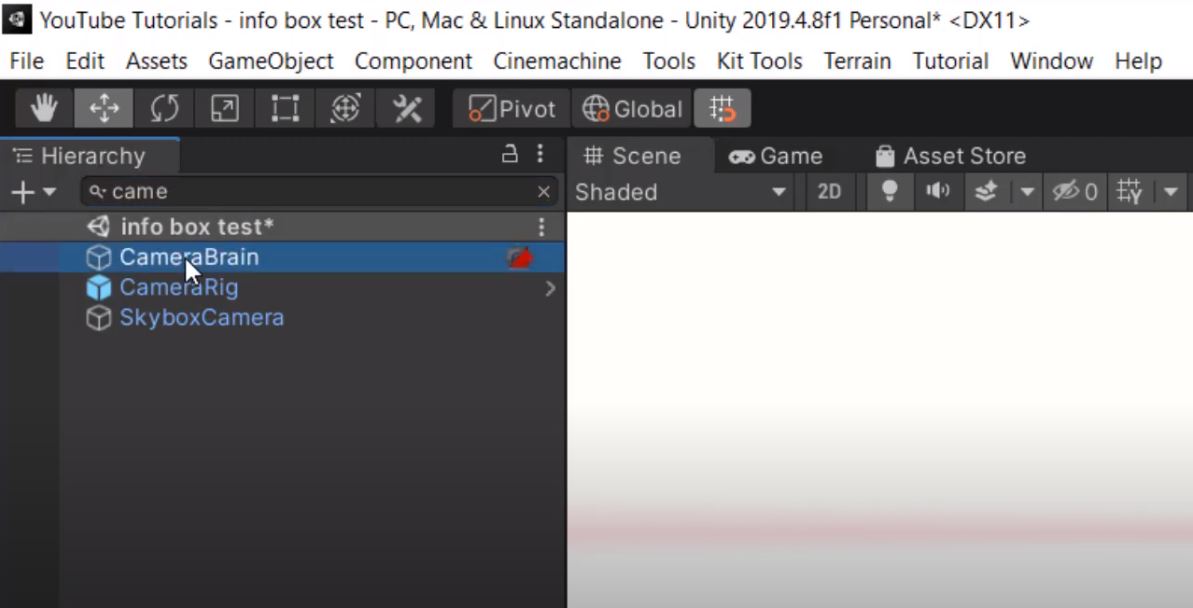
## **How to Add Music**

To add music into your game, you simply need two new elements. You need an **audio listener** and an **audio source**.

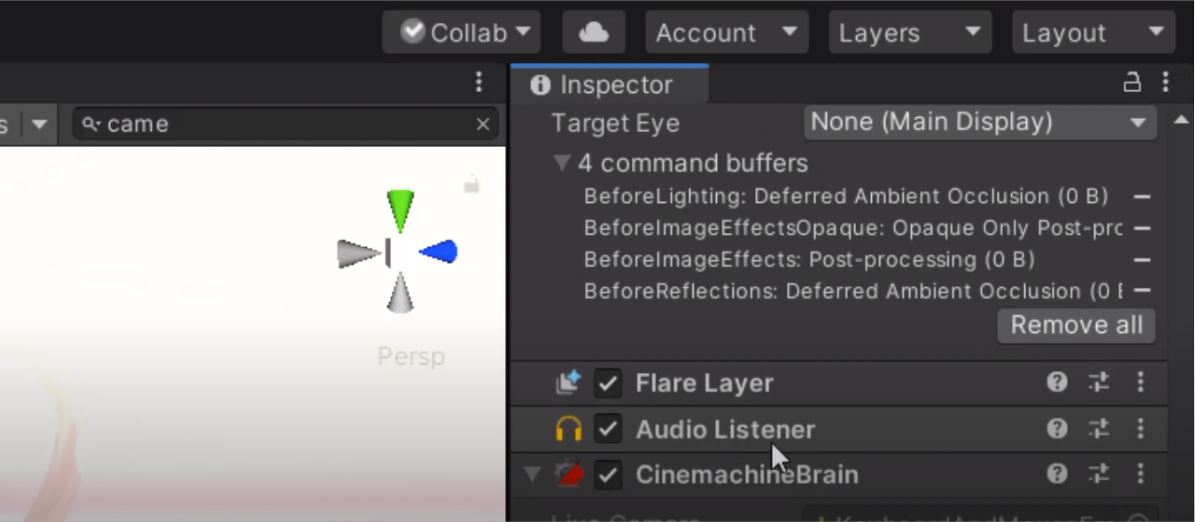
**Audio listener:** Think of an audio listener as the element within your game that hears your music. Essentially, this is the part of your world that captures the sounds that are being produced in your game. For example, if your character walks past a portal, that portal might emit a gentle buzzing or electric sound. Something within your game needs to hear those sounds. In most cases, game developers will add their audio listener to their camera. This is because in most games the camera follows the character around and therefore the sounds that are emitted will be picked up and registered by the camera which shares the perspective of the player. Therefore, all game sounds will appear to surround the character from their proper position within 3D space.

**Audio source:** An audio source is simply object within your game that is producing or emitting sound. That sound could be a sound effect, folly, atmosphere, room tone, dialoge or music.

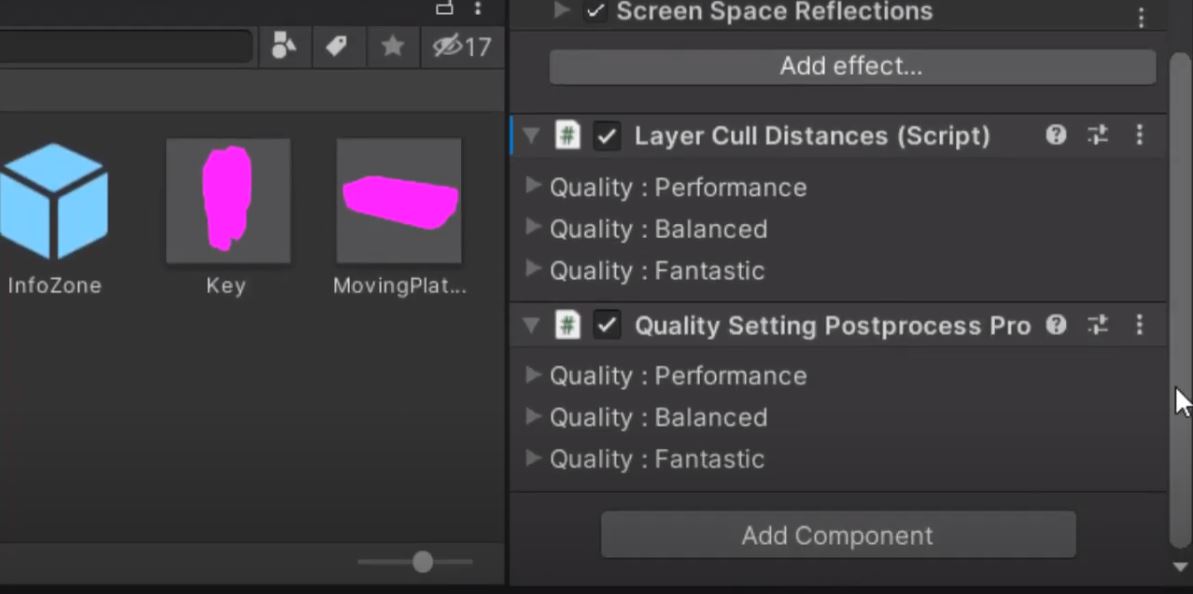
Now, all we need to do is add an audio listener and audio source to our Unity game.



The first thing we need to do is we need to add an audio listener. Within 3D Game Kit, the audio listener is already added to our game camera. If you go into the hierarchy tab and search for “camera” and then select “camera brain” (see image above) you will notice that the camera is already equipped with an audio listener.



If you do not have this component added to your game yet, you can simply scroll to the bottom of the inspector tab and click “add component” and search for “audio listener”.



Next, all we need to do is add our music into the game

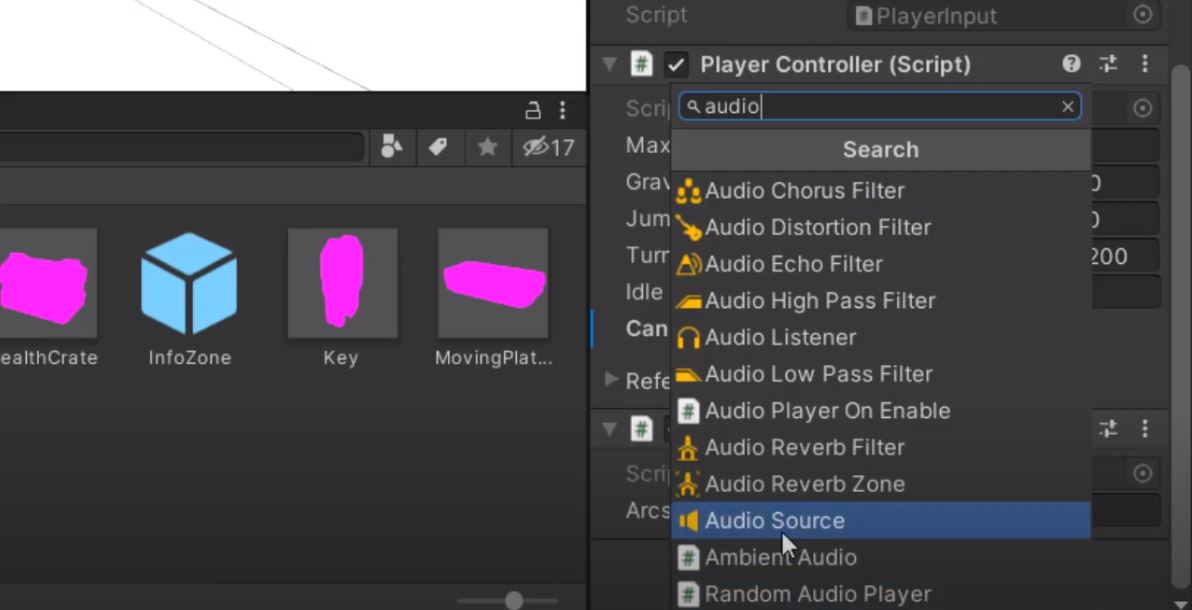
## **Adding the Source of the Sound**

The source of the game music can be anything. Obviously, if you had a music emitting object like a radio, television or sound system, you would want to associate your game’s music with those object. Remember, because our audio listener is connected to our camera, the audio will change depending on our character’s position in relation to those objects. For example, in 3D space, the sound will appear more distant if our character is futher away from those objects, and louder if the character is closer.

However, in todays tutorial we want to focus on adding background scene music into the game that remains consistent throughout the entire game. In order to do this, we need to set our character as the sound source. Essentially, our character will be emitting the sound of the music and our camera will be listening to the sound.

Within Unity’s 3d Game Kit, we need to go to our heierachy tab and search for our character. In our case we need to search for “Ellen” since this is our main character’s name. However, if you’re not using 3D Game Kit you’ll need to search for your character’s name.

Once Ellen is selected, we can go over into the inspector tab. Currently there is no audio source added to her character so we need to do that now. To add an audio source, simply scroll to the bottom of the inspector tab and click on “add component”. Search for and select “audio source”.



Once you have added an audio source to the player we can start setting up our background music.

## **Adding Background Music into Unity**

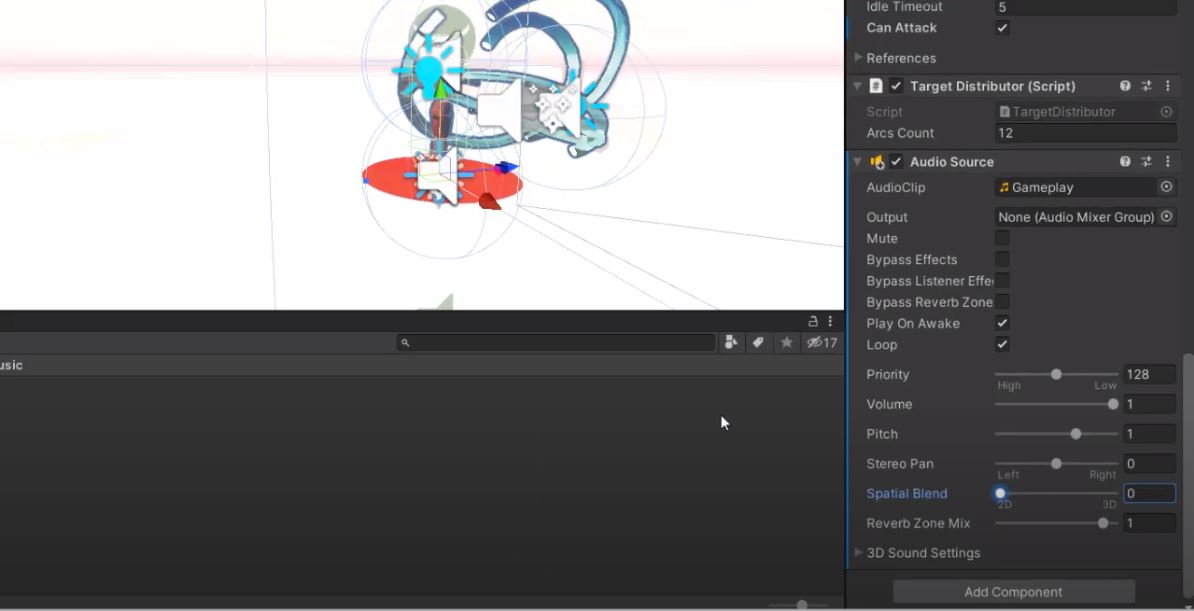
All we have left to do is modify a few settings in the audio source component section.

The first thing we need to do is add our clip. Unity supports many audio file formats including mp3, WAV, MPEG, OGG Vorbis, AIFF, MOD, IT, S3M and XM.

You’ll also want to ensure that “loop” is selected so the music continues to loop in the background while your character is in the game world.

You’ll also want to select “play on awake” so that the music begins as soon as your character spawns.

Lastly, you’ll want to adjust your volume and set your music to play in 2D or 3D space. For this tutorial, we’re not adding any directionality to our source so we’ll set it to 2D space. Your audio source section should look something like this.



Once we’ve completed all of these steps we can play our game and the music will be added to our scene.

## **Conclusion**

So  that’s it. I hope you’ve enjoyed this tutorial.

Remember, if you’re trying to take your n0-code game development skills to the next level, consider enrolling in our n0-code game development course where you’ll learn how to make breathtaking video games without having to know how to code or how to draw.

<https://www.sovereignmoon.studio/how-to-add-music-to-unity-3d-game-kit/>

## **unity**

## Creating Audio Sources

Audio Sources don’t do anything without an assigned **Audio Clip**. The Clip is the actual sound file that will be played back. The Source is like a controller for starting and stopping playback of that clip, and modifying other audio properties.

To create a new Audio Source:

1. Import your audio files into your Unity Project. These are now Audio Clips.
2. Go to **GameObject->Create Empty** from the menubar.
3. With the new GameObject selected, select **Component->Audio->Audio Source**.
4. In the **Inspector**  
   , find the **Audio Clip** property on the Audio Source Component and assign a clip, either by dragging one from the Project Window or by clicking the small circle icon to the right of the **Inspector** property, then selecting a clip from the list.

**Note:** If you want to create an **Audio Source** just for one **Audio Clip** that you have in the Assets folder then you can just drag that clip to the **scene view**  
 - a **GameObject**  
 with an **Audio Source** component will be created automatically for it. Dragging a clip onto on existing **GameObject** will attach the clip along with a new **Audio Source** if there isn’t one already there. If the object does already have an **Audio Source** then the newly dragged clip will replace the one that the source currently uses.

<https://docs.unity3d.com/2019.3/Documentation/Manual/class-AudioSource.html>

# Audio File Formats: How to Choose the Right File Type

Digital audio files are the raw material of [music production](https://blog.landr.com/music-production/).

From [streaming platforms](https://blog.landr.com/how-to-put-music-on-spotify/) to [sample packs](https://samples.landr.com/?utm_source=blog&utm_medium=organic&utm_campaign=Engagement_Samples_EN_Core_Blog&utm_term=audio_file_formats&utm_content=SamplesMicrosite), all the audio you work with has to be stored somewhere in a file.

Audio file formats are digital standards for storing audio information.

The raw data in a stream of audio from the analog-to-digital converter in your [audio interface](https://blog.landr.com/best-audio-interfaces/) is encoded using a technique called PCM or [pulse code modulation](https://en.wikipedia.org/wiki/Pulse-code_modulation).

PCM audio needs to be organized into a file so you can work with it, or play it back in a system.

Different audio file formats use different containers and varying methods of data compression to organize the PCM stream.

Depending on which you choose, each format represents the same information in different storage sizes or quality levels.

In addition to that, some audio file formats carry [metadata](https://blog.landr.com/music-metadata/) that supplies information about the file or its content.

## **Lossless vs. lossy audio files**

There are two main types of audio file—lossless file formats, and lossy file formats.

The difference between the two has to do with data compression.

Data compression means making the files take up less space on a hard drive. It’s not the same as the [dynamic range compression](https://blog.landr.com/how-to-use-a-compressor/) used in music production.

Some methods of data compression make the file smaller but still preserve 100% of the information in the raw audio stream. These are known as ***lossless compressed formats***.

Other compression types work by eliminating data in the audio that doesn’t make a big impact on the sound. Some information is thrown away using this method, so these are known as ***lossy compressed formats***.

## Uncompressed audio formats

There are other audio file formats where no data compression is used. These are called ***uncompressed audio formats***.

These file types act as a container for raw audio data without reducing its size or quality in any way.

These are the largest files to work with, but they provide the highest level of detail in the audio information.

Uncompressed audio files are the type most often used for recording and [mixing music](https://www.landr.com/how-to-mix) in a DAW.

Even so, uncompressed audio files also come in different quality levels. These are based on the accuracy and precision with which the analog audio signal was converted to digital.

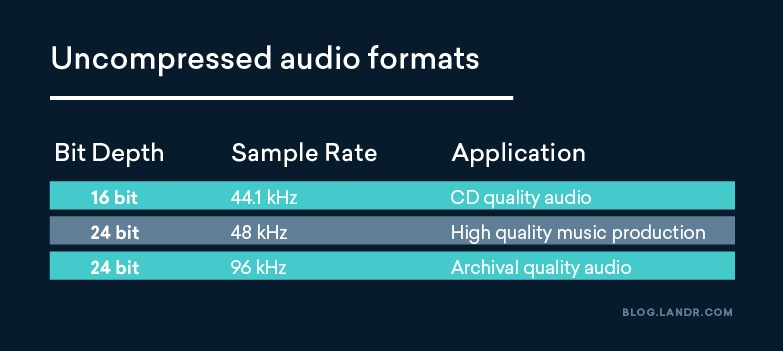
The higher the sample rate and bit depth used, the more information is captured in the conversion process.

Bit depth represents the precision of the AD/DA converter for measuring amplitude, or the volume level of the signal.

**Sample rate** means the number of times the measurement is taken in a second. Higher sample rate means more individual measurements made.

Uncompressed audio files are the type most often used for recording and mixing music in a DAW.

Here is a list of common quality levels for uncompressed audio:



## **Audio bitrate**

Lossy compressed audio files can be encoded at different quality levels.

The quality of this file format is determined by the bitrate, or the amount of data encoded per second.

At lower bitrate settings, the compressed files will be much smaller, but may sound worse.

A high quality standard for MP3 compression is 320 kbps. At these settings it’s very difficult to distinguish compressed audio from uncompressed in [casual listening tests](https://www.npr.org/sections/therecord/2015/06/02/411473508/how-well-can-you-hear-audio-quality).

## **The 6 most common audio file formats**

There are many different file formats out there, but not all of them are widely used.

In fact, there are only a handful that you’ll commonly see in the wild. Here are the main ones to know:

### MP3

MP3s are the most common file type for general listening.

The use of MP3s exploded during the file sharing revolution of the early 2000s. The reason why has to do with the sound quality they were able to achieve in such a small package.

MP3s were the first audio file format that made music easy to send back and forth across the internet in listenable audio quality.

They were also easy to encode from tracks on a CD, which led to the proliferation of illegal music downloads.

MP3 files are still very common today and some digital download stores like Bandcamp sell them as their primary format.

### AAC

AAC is a lossy compressed format designed by a group of digital technology companies including Dolby, Microsoft and Bell . It was intended as a more efficient successor to MP3. AAC is known for being the standard format for Apple devices like the iPod.

### Ogg Vorbis

Ogg Vorbis is an open source alternative to lossy compressed formats like MP3. It’s notable for being the file type used for audio material on Wikipedia.

Despite its widespread use, MP3 is a proprietary format. In response, the open source community created Ogg Vorbis as an alternative that’s free and editable.

### FLAC

FLAC is an open source lossless compressed file format. It was one of the first lossless compressed formats to gain popularity.

FLAC files make listening to lossless audio possible on devices with limited storage. The benefits of lossless audio as a listening medium are often debated by audiophiles, but see for yourself if you prefer them!

### WAV/AIFF

These are the most commonly used file types for working with lossless uncompressed audio.

Since there is no change to the amount of information included, both file types have similar performance.

WAV was created for use on PC, while AIFF was developed by Apple for the Mackintosh. Both formats are compatible on either platform.

For music production, use uncompressed audio with high sample rate and bit depth (24 bit / 48 kHz WAV or AIFF)

## **How to choose an audio file format**

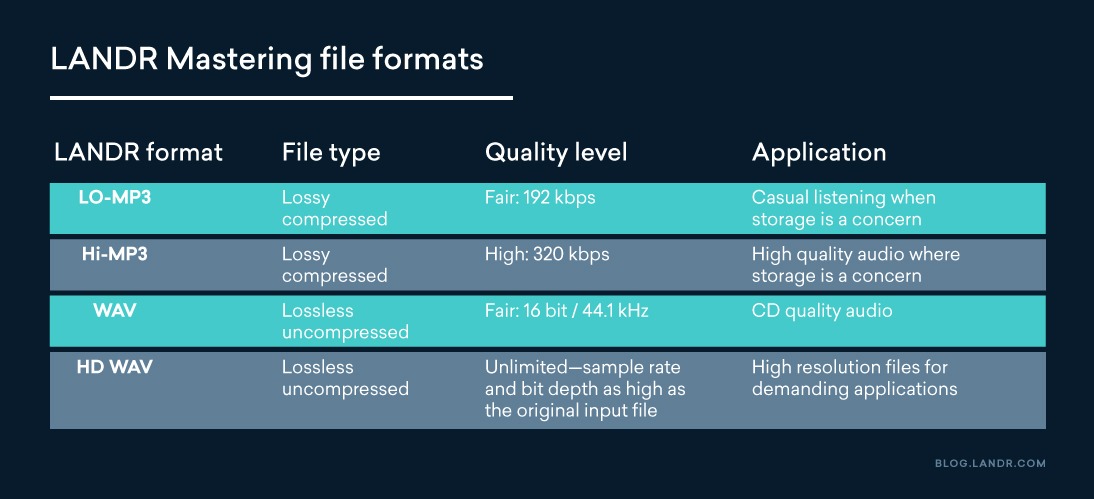
With the background info out of the way, here are the simple guidelines for choosing an audio file format:

* For music production, use uncompressed audio with high sample rate and bit depth (24 bit / 48 kHz WAV or AIFF)
* For general listening, choose a high bitrate compressed format (320 kbps MP3, AAC or similar)
* For critical listening, choose a lossless compressed format (FLAC)

## LANDR Mastering file formats

[LANDR Mastering](https://www.landr.com/en/online-audio-mastering?utm_source=blog&utm_medium=organic&utm_campaign=Engagement_Mastering_EN_Core_Blog&utm_term=audio_file_formats&utm_content=MasteringGuestsite) offers downloads in four different quality tiers—LO-MP3, Hi-MP3, WAV and HD WAV.

Here’s a breakdown of each one:



***Hot tip:*** No process in music production can add audio quality back if it didn’t exist already. To get the highest quality file from LANDR Mastering, you need to start with a high resolution file. We recommend setting your DAW sessions to at least 24 bit / 44.1 kHz, or higher if possible.

## **File system**

Audio file formats are a technical detail in digital audio that may not seem important.

But choosing the right one can make a difference to your final product.

If you’ve made it through this article you’ll have a great starting for understanding audio file formats.

https://blog.landr.com/audio-file-formats/

# **codec**

(**coder-decoder**or **compression-decompression**), A standard for compressing and decompressing digital media, especially audio and video. Codecs are used to save files to disk and transfer media over a computer network. Rapid compression and decompression of this data reduces bandwidth requirements and increases the amount of interactive and multimedia content accessed and transmitted over the network.

Different types of video are better encoded in different formats, just as some audio codecs better encode human voice and others better encode musical instrument music. In general, the most efficient codecs also require considerable processing power. Multimedia distribution always requires a balance between computing power and bandwidth.

https://www.britannica.com/technology/codec