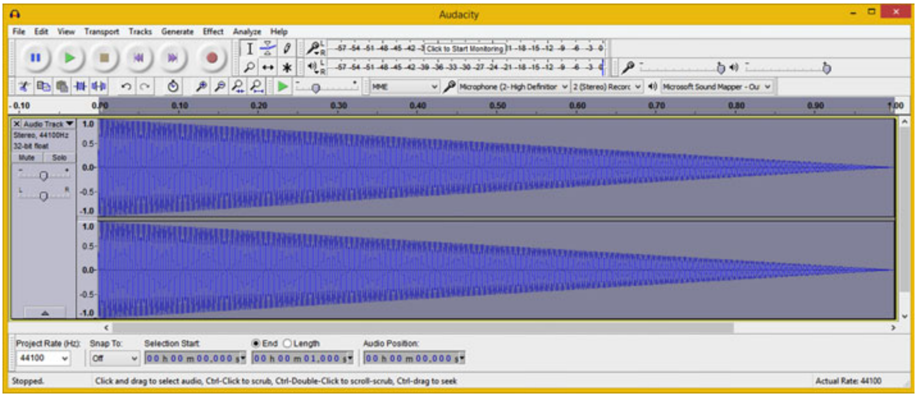

Figure 11-11. *Preview the Stereo Tuning Fork audio data sample*

It’s important to note that you can use a Mono Track for this tuning fork track functionality. You can also delete it once you are finished using it, in order to save system resources such as memory and CPU processing cycles.

Next, let’s look at David R. Sky’s **Fire and Explosion** generator, a virtual special effects shop in Audacity 2.1! You can use this FX tone generator to create special audio effects for your games.

Virtual Carnage: The Fire and Explosion Generator

The Fire and Explosion generator allows you to set the type of audio effect, sound duration, attack time in milliseconds, explosion decay in milliseconds, decay percentage, cut-off frequency, filter quality, Bass boost frequency and decibels, and clipping percentage. I used the defaults, as you can see in the Fire and Explosion dialog shown in Figure 11-12.

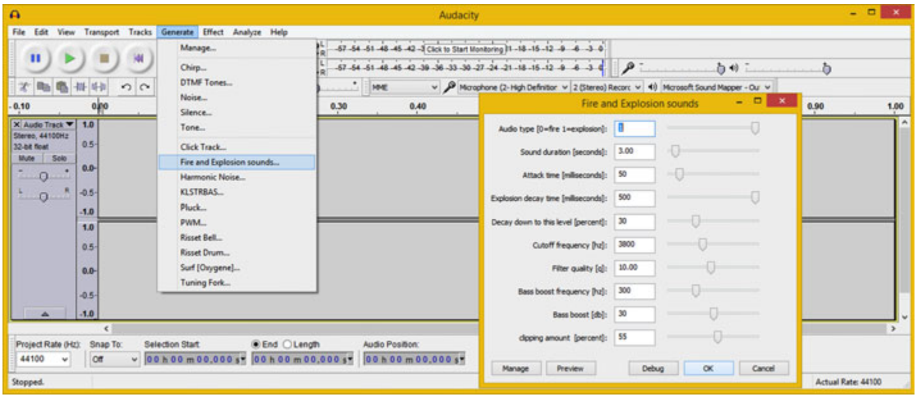

Figure 11-12. *Select your Generate ➤ Fire and Explosion sounds*

I accepted the default setting values and clicked the **OK** button to generate the Explosion effect track (see Figure 11-13). Click the **play** button to preview the explosion effect.

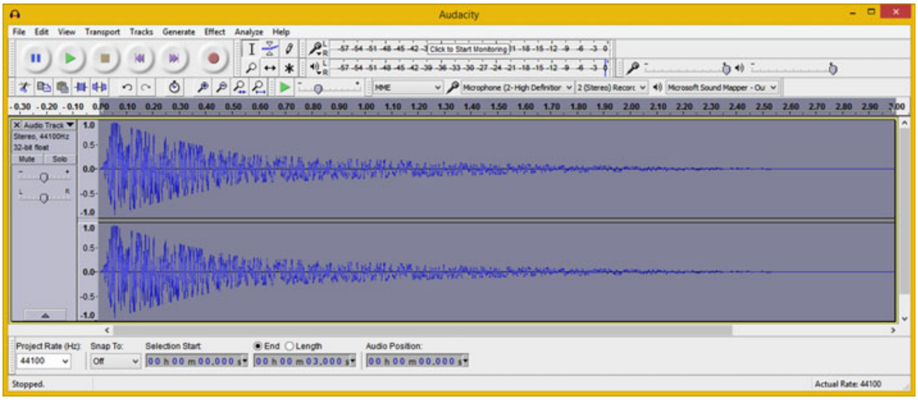

Figure 11-13. *Preview the Stereo explosion audio data sample*

Next, let's look at Steven Jones' **KLSTRBAS** generator, which stands for **cluster bass**.

Virtual Cluster Bass: The KLSTRBAS Generator

According to KLSTRBAS tone generator author, Steve Jones: "KLSTRBAS generates dense sounds by combining several waveforms with a fixed frequency ratio between them. Early Roland drum machines created cymbal sound in part by combining multiple square waves with non-integral frequency ratios. The combined signal was then high-pass filtered to produce a very dense cluster of high frequency harmonics. The genesis of KLSTRBAS was a failed attempt to create cymbal sounds using this technique."

As you can see in Figure 11-14, the KLSTRBAS dialog allows you to set your **MIDI key** (note value), **Decay** value, **Fractional Decay** value, **Density** value, **Detune** value, **Flange** value, and **Wave table** indicator value.

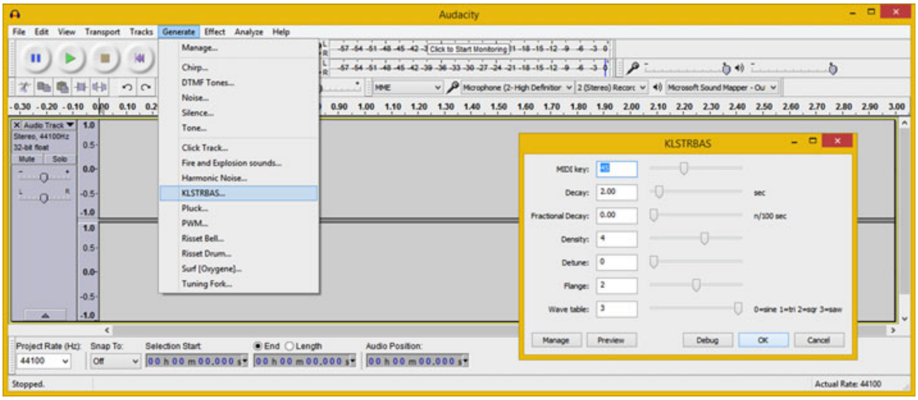

Figure 11-14. *Select the Generate ➤ KLSTRBAS menu sequence*

The wave table indicator can be set to one of four value settings: 0 indicates a sine wave table, 1 indicates a triangle wave table, 2 indicates a square wave table, and 3 indicates a saw wave table.

As evident in the KLSTRBAS dialog, you have a great deal of variation in creating cool tones with this generator. I set the A3 note is the MIDI key, a 2-second decay, a fractional decay value of 0, a density value of 4, a detune value of 0, and a middle flange value of 2. I use the square (also called a pulse) wave table by specifying a value of 3.

I accepted the default setting values and clicked the **OK** button to generate the Cluster Bass data track (see Figure 11-15). Click the **play** button to preview the sample.

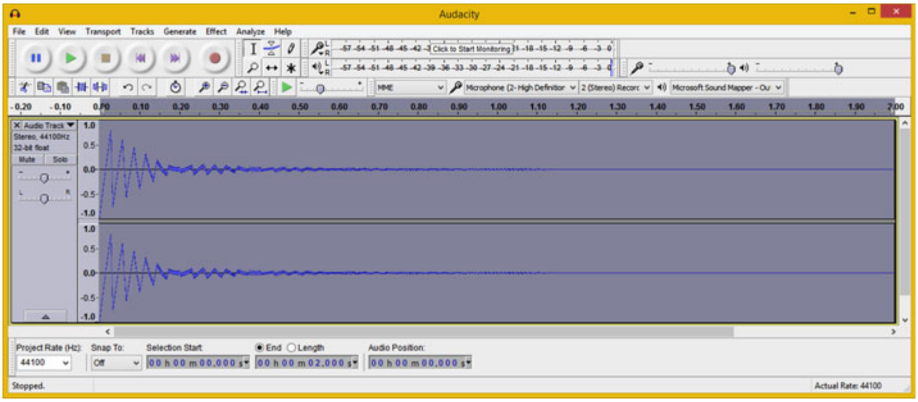

Figure 11-15. *Preview the Stereo Cluster Bass audio data sample*

Next, let's take a look at the harmonic noise generator.

Extraterrestrial Vocals: Harmonic Noise Generator

The harmonic noise generator allows you to set a **MIDI Note List** and the **Number of Harmonics** to generate from this list. There are also **Duration**, **Band Width**, and **Odd Harmonics Only** options in the Harmonic Noise dialog (see Figure 11-16) to generate a very unique noise track.

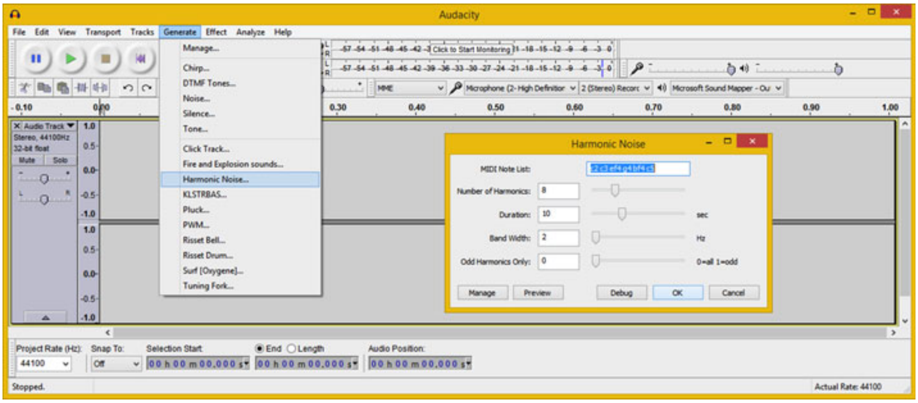

Figure 11-16. *Select a Generate ➤ Harmonic Noise menu sequence*

I accepted the default setting values and clicked the **OK** button, generating the Harmonic Noise sample data track seen in Figure 11-17. Click the **play** button to preview the Harmonic Noise stereo sample, which you'll see (and hear) is out-of-this-world extraterrestrial in nature. The waveforms even look like alien centipedes—and they sound even stranger than they look!

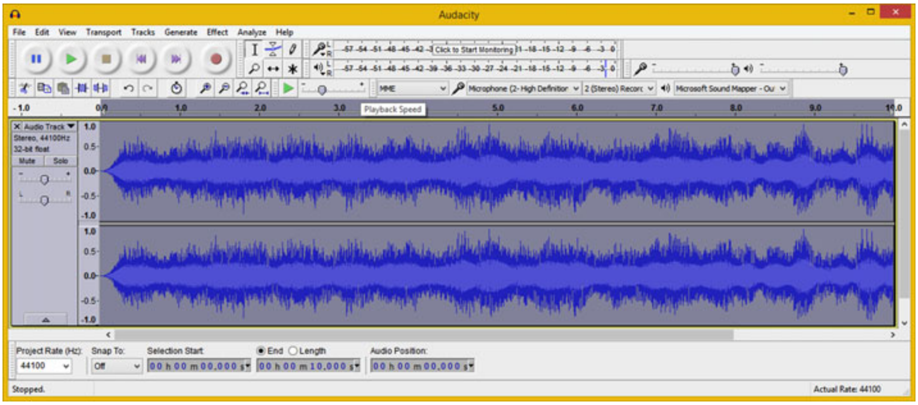

Figure 11-17. *Preview the Stereo Harmonic Noise audio sample*

Next, let's return to an audio tone generator that comes preinstalled in Audacity 2.1—the **Pluck** tone generator. No, not the Cluster Pluck generator; let's keep it clean here. It's just a simple pluck-string instrument generator that can be used to simulate bass guitars, guitars, or even banjos, using the proper digital audio effects processing.

Virtual String Instrument: The Pluck Generator

The Pluck tone generator is much simpler than the other audio synthesis algorithms you've seen in this chapter. It allows you to set **Pluck MIDI Pitch**, which defaults to middle C, as well as **Fade-out type**, which I set to gradual, because it's more like what a plucked string does in real life. There is also Duration in seconds, which I set to 1.00 second. You can see these options in the Pluck dialog shown in Figure 11-18.

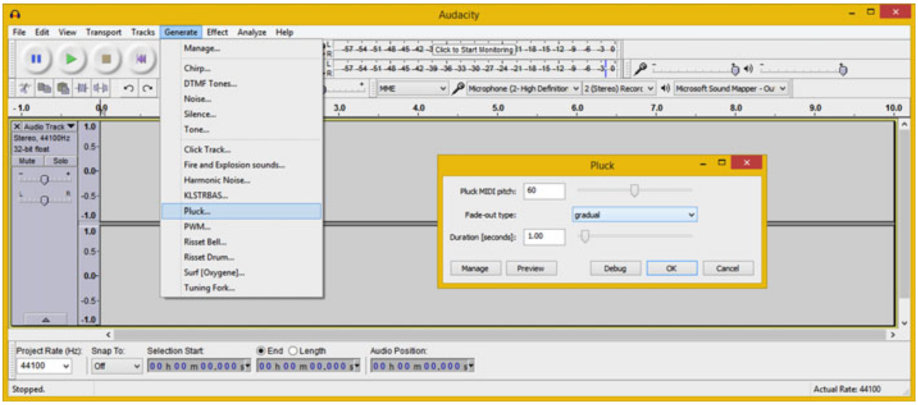

Figure 11-18. *Select the Generate ➤ Pluck menu sequence*

I accepted the default setting values and clicked the **OK** button, generating the Pluck sample data track shown in Figure 11-19. Click the **play** button to preview the Pluck stereo sample. I noticed a high level at the beginning of the sample, so I took a screenshot of the playback to show this.

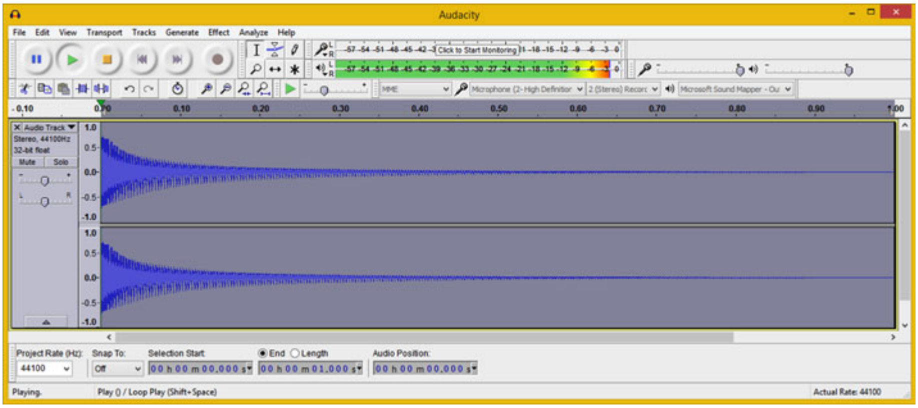

Figure 11-19. *Preview a Stereo Pluck digital audio data sample*

This is most likely because a plucked string generates a high decibel level right after the string is released, generating all of its audio "power," or its amplitude, at that initial point in time.

These Generate menu digital audio synthesizers, or waveform generation algorithms, bring a lot of power to your digital audio editing and compositing work process.

If you think "outside of the box" and utilize Audacity's vast Effect menu in conjunction with the Analyze menu, you will be able to create any digital audio assets that you need for your multimedia projects.

Summary

In this chapter, you looked at digital audio synthesis, known as **tone generation** in Audacity, using the Generator menu. First, you doubled the amount of tone generation algorithms in Audacity 2.1.1, and then you learned how and where to get even more of these Nyquist plug-ins.

After that, you looked at seven of the most useful tone generator algorithms available to Audacity. You explored their algorithm configuration dialogs and the resulting synthesized digital audio waveforms.

In the next chapter, you learn about **digital audio data footprint optimization** concepts, terms, and principles.

CHAPTER 12

The Data Footprint of Digital Audio: Compression

You now know the fundamental concepts, terminology, principles, and work processes surrounding digital audio editing, compositing, and synthesis, which allows you to create digital audio assets for your new media content. Now it is time to get into digital audio **data footprint optimization** and the way to go about getting the smallest possible digital audio file size for your audio assets, while at the same time getting the pristine audio playback quality results that you need for your interactive digital application development.

You'll look at the primary considerations for optimizing your digital audio asset, including **device compatibility** (a baseline digital audio asset to gauge compression against each codec that you learned about in Chapter 4), optimizing digital audio usage (system memory and in network bandwidth considerations), and similar advanced topics on how digital audio is used.

Audio Optimization: Device Compatibility

Optimizing your digital audio assets for playback across the widest range of platforms, operating systems, and hardware devices is going to be easier than optimizing digital video or digital image assets. This is because there's a wide range of display screen resolutions and aspect ratios compared to the more limited range of digital audio playback support (except for the latest Android hardware featuring 24-bit HD Audio playback compatibility).

People's ears can't perceive the same quality differences in digital audio that their eyes can perceive in digital images. There are two important "sweet spots" for digital audio support across hardware devices that you should target. A lower-quality audio (narration track or sound effects) use an 8 kHz or an 11.25 kHz sampling rate, with 8-bit, 12-bit, or 16-bit sample resolution.

Medium-quality audio (game music loops, longer sound effects, ambient background audio) use a 22.05 kHz or a 32 kHz sampling rate, with 12-bit or 16-bit sample resolution.

Electronic supplementary material The online version of this chapter (doi:10.1007/978-1-4842-1648-4_12) contains supplementary material, which is available to authorized users.

W. Jackson, *Digital Audio Editing Fundamentals*, DOI 10.1007/978-1-4842-1648-4_12