

# The Impact of Advancing Technology on Digital Music Production

by

Jessica Pinto

California State Polytechnic University, Pomona

December 2024

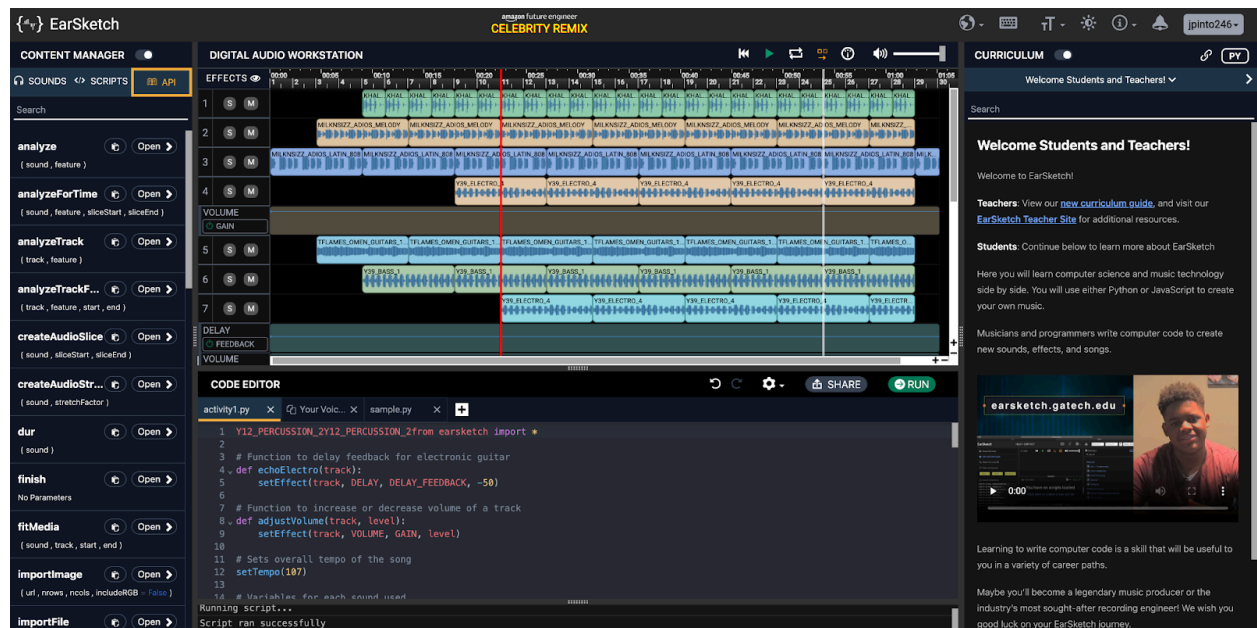
In the early 2000s, revolutionary technological capabilities of digital music production were limited to those that truly needed it and could afford it—mostly record companies and production studios. The same cannot be said in 2024, where anyone with a computer has access to the internet, and in turn, access to anything that the internet has to offer. The advancement of technology has made free digital music production websites increasingly popular worldwide, and websites like Earsketch represent the beginning of a new path in digital music production. However, with such advantages come weaknesses that need to be addressed as the growing possibilities with new technologies can lead to an ambiguous future.

In order to understand the growth of digital music and its production, it is imperative to recognize how this technology has allowed individuals to accomplish musical feats they either could not afford or simply did not have the knowledge to be able to compose. An example of a very simple digital music composition website is an application called Beepbox. Beepbox has a relatively simple and easy to use interface that allows a user to pluck out notes and play it back in repeat to add, remove, or change the piece as much as they like. There are options to choose which instrument the music should sound like, the scale, key, tempo, rhythm, volume, and so much more—all at no cost to the user. For an individual with no musical experience, this would not be a challenging tool to use to compose music. Of course, Beepbox caters more towards entertainment than actual music composition because of the program's simplicity, however it is not impossible to create professional sounding music using its features. With Beepbox one does not have to deal with issues of sampling or copyright because each and every note is placed by the user and they build their music one note at a time. Although it must be noted that apart from its inherent use, it does not require a user to have prior knowledge of the instrument or any other aspect of music. While it does allow for customization of keys, scale, tempo, and rhythm, within

a few minutes of testing out the features, anyone would be able to understand what part of the music these changes would affect. For many that lack the ability to buy instruments, pay for music lessons, or free enough time to practice regularly, musical composition websites like Beepbox are welcome platforms through which they can pursue their artistic interests.

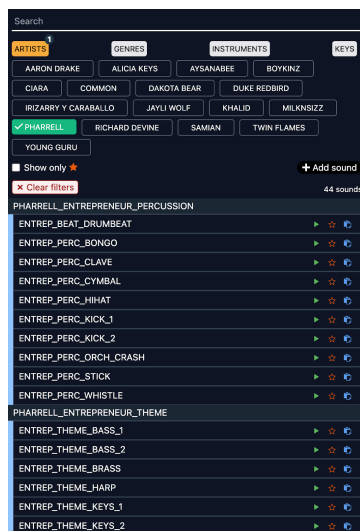
Created with a similar goal in mind, Earsketch is an application that allows users to code their compositions using popular programming languages like Python and Javascript. In an era where computer science is a growing field and many individuals are interested in both areas of study—computer science and music—it is an excitingly refreshing application. Unlike Beepbox, which uses the click of a button to make sound and create compositions, Earsketch relies on the use of popular programming languages like Python and JavaScript to reproduce music. Of course, here we see a higher level of knowledge as a prerequisite for use, though the multitude of resources provided by Earsketch detailing its features certainly reduces this gap. Despite its intimidating structure at first glance, Earksetch opens up a connection between individuals of two vastly different domains and allows them to connect on a digital middle ground.

Interestingly enough, I was introduced to Earsketch in my Computer Science class, and as a Computer Science major and avid pianist, I was intrigued by the new tools at my disposal. With my prior knowledge of Python, I was able to create a variety of digital music compositions and even utilize my understanding of music theory to help with this process. While the general layout of Earsketch may seem complicated at first glance, those with even a little understanding of programming would be able to grasp the concept of the application. This is because “EarSketch’s focus on immediate opportunity to act, a high level of abstraction, and a connection to multitrack audio editing paradigms leads to a feature set that fully supports an introductory computer science curriculum”, the result of which is displayed below (Moore, 81).



For those familiar with programming platforms like Visual Studio Code, Google Colab, or Eclipse, the layout is very familiar and encourages exploration of the website’s resources. For those with no experience with any of the aforementioned platforms, there are guided tours and in-depth tutorials with working code to help ease the learning experience. These tutorials are certified educational materials as “the EarSketch curriculum is aligned with the programming standards of the College Board’s Advanced Placement (AP) Computer Science Principles (CSP) course”, ensuring that while students learn how to compose music, they are also learning the appropriate coding methods based on national standards (Moore 82). In the screenshot above, the code in the editor is executable code that I have written to create a song, and while it is an example of a more complicated script, the same song can be easily implemented using very limited knowledge of the language, inviting users of all skill levels to code digital music. Under the “sounds” menu, a user can find a large variety of sampled music from artists like Alicia Keys, Khalid, Pharrell Williams, and many others. Under each of these artists, one can find anywhere from fifty to two thousand sounds that match the discography of the artist. The amount

of tracks can be initially overwhelming, but "sounds are grouped into collections that contain loops in the same style and key and are designed to fit well together", making it easy for those with limited music knowledge to find samples that compliment each other (Moore, 82). If the user is more musically proficient, there are filters that allow them to search for samples given a specific key. Not only does this make it easier for them to find tracks that are unique and sound good together, but it also combines samples that are not necessarily from the same artist or



belong to the same song. An exhaustive list of Earsketch's available tracks would span the remainder of this essay, however the image on the left is a representation of just a few tracks from Pharrell Williams—keep in mind he is one of twenty featured artists, each of whom provide a similar or greater amount of samples, free of charge. As can be seen, even if a user does not know what these instruments look like, they are able to make music that sounds almost professional by incorporating such

samples.

While having samples at our disposal is one thing, being able to actually manipulate the music is another. That is where the code comes in. In Earsketch, every single track needs to be imported from the library, and each variation is implemented using the provided API (Application Programming Interface). APIs "are mechanisms that enable two software components to communicate with each other using a set of definitions and protocols" (Amazon Web Services). These definitions and protocols are what connect the music and code together, so that the instructions used from the API allow the user to alter the music by issuing an easily

understandable command. An example of this can be seen in the Figure 1.1, where the code written is:

```
setEffect(track, DELAY, DELAY_FEEDBACK, -50)
```

The **setEffect** function is an instruction that the API reads and uses to change the track, delay its feedback, and specifically delay it by -50. Delay is not the only effect a user can change. The API provides a myriad of options when it comes to their effects, including distortion, pan, pitch shift, volume, and compressor to name a few. Each of the highlighted portions above can be changed to produce a different timbre. Briefly, distortion “adds a dirty, fuzzy, and gritty effect to a sound by clipping the sound wave and adding overtones”, pan “affects the mix between the left and right audio channels”, pitchshift “raises or lowers the pitch of a sound”, and compressor “reduces the volume of the loudest sections of a sound and amplifies the quietest sections” (Earsketch). The function **setEffect** is one of thirty two functions provided by the API that can be used to change the music. Some other function examples are **fitMedia** (used to add the music to the code), **makeBeat** (creates a rhythmic pattern using english characters), **setTempo** (sets the tempo based on a number using the same scale as a metronome), and **shuffleList** (randomly rearranges items in a list—items that can be made up of samples). Of course, in addition to API functions, there are other Python libraries that allow a user to loop the music, create customized lists, add beats, and use conditional statements like **if-else** to have more control over the music than ever possible with a human player. Now, some may imagine that this type of control would possess reverse salients similar to that of the player piano and other self playing instruments that lacked the emotional depth and expressiveness of human players. However, this weakness is a significantly smaller issue with Earsketch because of the

use of samples. Many of these samples are taken from real human voices and acoustic instruments played by human musicians, ensuring that the same depth and emotion in the music is always carried forward.

Unfortunately, just as these samples resolve a century-old issue relating to lack of expression, it also brings up the question of who is the true owner of this sample centered music that anyone can use without spending a dime. Earsketch informs their users that they “are not creating sounds from scratch, but instead combining and using them in new ways” (Earsketch). This means that users have to be careful with how they label their compositions if they primarily use Earsketch to compose music and attempt to publish their creations, even though “Copyright owners tend not to sue those who trade software, video, or music files over the Internet” (Reese 1346). The legal side of digital music production is a complicated and unpredictable domain, where every case has a different outcome.

As we look at the current issues that may arise with sampling, it is also vital to consider the growth of artificial intelligence in the 21st century and how this might be significant for a website like Earsketch. As of 2024, Earsketch is a relatively new website, with a small database and the possibility to do so much more when it comes to programming. This means that in the future, developers would be looking to expand their sample database, and solely showcasing a growing number of popular celebrities may not be financially realistic. Moreover, most Copyright “suits rely on theories of secondary liability, focusing on those who provide services or write software that can be used in an act of infringement” (Reese 1346). By providing a larger amount of sampled music from renowned artists, Earsketch is placing a target on themselves, which may make AI an attractive option. At this very moment there are AI music generators that can produce entire songs with a single prompt. If Earsketch were to include such artificially

generated music, it would certainly allow for a more customizable experience for the user, but the dilemma lies in who would become the true owner of this music. AI models work by coming to conclusions using logic processes and utilizing the data it has been provided. This unveils an even deeper layer of the problem where the music generated from such models may just be smaller and smaller samples of other songs in their databases. Furthermore, artists and composers today constantly remake and republish older songs in the same manner, and with digital composition platforms making the process more accessible, it is difficult to conclude who should be held accountable when it comes to Copyright infringement. As a niche application, Earsketch is doing a great job providing rich educational material and creative outlets for their users, but as the application widens its scope, it might be worth taking a closer look at their sources of music to allow users more freedom when it comes to publishing their creations from Earsketch. Unfortunately, even new technologies have their weaknesses, and stepping into new territory could mean there is even more ambiguity following each advancement.

Heading towards a new era of technology means change is imminent within the field of music, and while free digital composition websites provide tools to make music more accessible around the globe, we must also understand where their limits lie, especially when it comes to music in the shared digital space. Access to free musical composition tools like Beepbox and Earsketch provide inquisitive individuals a way to learn basic music composition and open an avenue for them to express themselves through music. Despite this, it must also be made clear the limits to which such tools can be used. As a concept, Earsketch is one of the most innovative websites I have used, but with its current database and possible distribution issues concerning the sampled music it is based on, users might not be able to truly experience coding their music from start to finish or find joy in publishing *their* creations.



# Bibliography

Nesky, John. “Beepbox.” *BeepBox*, 20 Feb. 2012,

[www.beepbox.co/#9n31s0k0l00e03t2ma7g0fj07r1i0o432T1v1ud4f10r8q011d23A5F3BfQ0031Pfa35E362263276T1v1u25f0q0w10v3d08A5F4B0Q000dPc696E2b8638T3v1uaef0q0x10p71d23Sp99f9c9Vppbaa9gE1b9T2v1u15f10w4qw02d03w0E0b4h400000000h4g000000014h000000004h400000000p1gBWqfwLEFeP000000](http://www.beepbox.co/#9n31s0k0l00e03t2ma7g0fj07r1i0o432T1v1ud4f10r8q011d23A5F3BfQ0031Pfa35E362263276T1v1u25f0q0w10v3d08A5F4B0Q000dPc696E2b8638T3v1uaef0q0x10p71d23Sp99f9c9Vppbaa9gE1b9T2v1u15f10w4qw02d03w0E0b4h400000000h4g000000014h000000004h400000000p1gBWqfwLEFeP000000).

Lemley, Mark A., and R. Anthony Reese. “Reducing Digital Copyright Infringement without Restricting Innovation.” *Stanford Law Review*, vol. 56, no. 6, 2004, pp. 1345–434.

*JSTOR*, <http://www.jstor.org/stable/40040194>. Accessed 14 Dec. 2024.

Earsketch.gatech.edu. “Earsketch Copyright.” *Copyright*,

[earsketch.gatech.edu/curriculum/en/v1/copyright.html](http://earsketch.gatech.edu/curriculum/en/v1/copyright.html). Accessed 13 Dec. 2024.

“What Is an API? - Application Programming Interface Explained - AWS.” *Amazon Web*

*Services*, [aws.amazon.com/what-is/api/](https://aws.amazon.com/what-is/api/). Accessed 14 Dec. 2024.

Freeman, Jason, et al. “Earsketch: Engaging Broad Populations in Computing through Music:

Communications of the ACM: Vol 62, No 9.” *Communications of the ACM*, 21 Aug.

2019, [dl.acm.org/doi/10.1145/3333613](https://dl.acm.org/doi/10.1145/3333613).