In 1980, David Cope, a composer and newly appointed professor at the University of California Santa Cruz, was commissioned to write an opera. He was having a horrible case of writer's block and decided to work on another project instead. Cope imagined a program that could process a database full of Bach chorales and output a composition that would imitate Bach's style. Then, he thought, he could simply replace the chorales with his own work and the system would output works that sounded like his own. Seven years later he completed the program. With its assistance, he wrote the opera in two days. He named the program Experiments in Musical Intelligence (EMI) and referred to it affectionately as Emmy. Emmy produced thousands of compositions based on the musical works of Bach, Beethoven, Mozart, Bartok, Rachmaninov, and many others. Emmy produced 11,000 works before Cope he "unplugged her," and five-thousand Emmy-produced Bach Chorales are available for free on Cope's website. In his prolific lifetime, Bach himself was only able to compose around one thousand compositions. Despite this incredible throughput, the Emmy-generated compositions are shockingly similar to the composer's unique style. Professor Winifred Kerner of the University of Oregon constructed a musical version of the Turing Test with Emmy's reproduction of Bach's style: A pianist played a real Bach piece, a Dr. Steve Larson piece written in the style of Bach, and an Emmy piece, and asked an audience to pick which was which. The audience selected Emmy's piece as the actual Bach, and Larson's piece to be the one composed by a computer (Garcia, 2015) - Emmy had passed the musical Turing Test. Machine learning algorithms have come a long way since then, and have become increasingly available in packages easily downloadable onto a laptop computer. Google DeepMind announced in June that it is implementing deep neural networks to music in a project called Magenta, and the source code is mostly public. Computer generated music has immense potential, and is developing very fast.

Cope was not the first to employ algorithms to music composition, and Emmy was not the first computer program to write music. Serial music, chance music, and even the rules of counterpoint that govern tonal music are all applications of algorithms. Due to the mathematical formalism of music, a case can be made that all music is algorithmic. We are at a crux point in music history, because computers can now apply these algorithms. This raises foundational questions about composition and music. If a piece was generated by a computer, then who wrote it? What does it mean to have composed something? If computers can compose well, then is composition a rote task no more glorious or meaningful than chopping wood? Music is often described as the "language of the soul" and to communicate emotion - if so, then how can computers create it when they lack emotions? And, if they can't communicate emotion, then what are these amalgamations of notes that computers can now construct that people listen to and believe are composed by humans? Are the compositions made by computers good? Is David Cope correct that "there is no great music. There is no bad music. All of that is personal taste" (Cope, 2016).

Most of these questions have been around since before 1980 during the mainframe era of computing. There are two main categories of computer music: sound production, and sound composition. Beginning in the mid 1950's, efforts were made to accomplish the kind of synthesis that could be done using analog sine wave generators and tape on digital computers. The MUSICn series of programs lead by Leland Smith at Stanford University, was the most influential and most widely used of this family of programs (Manning, 2013, p. 197). The second category is more interesting - automatic composition, or what Brian Eno and others call generative music. Some efforts pursued both categories, and the composition produced by a computer was also synthesized and played by a computer. For example, *HPSCHD*, a notorious collaborative effort between Lejaren Hiller and John Cage, contains tape which was generated using Hiller's programs and synthesized using the MUSIC IV B program (Manning, 2013, p. 201). Automatic composition is a relative term. A computer can only compose within the limitations set by its hard and software. Computer compositions cannot be truly autonomous because they must be programmed by a human, and initiated with a starting set of parameters. However, they are autonomous in that the composer does not control the process from note to note outside of the initial programming and input parameters. Credit for the first composition composed by a computer is usually given to Lejaren Hiller and Leonard Isaacson for the creation of the *Illiac Suite* in 1957. An in depth look at the *Illiac Suite* reveals both the promise and limitations of these early systems. Composers, like Iannis Xenakis and Gottfried Michael Koenig, were quick to see the value of computers for composing aleatoric and serial pieces respectively. PODn, a series of programs for granular synthesis developed by Koenig, was used by several composers to make innovative and unique compositions like Jean Piche's *Heliograms*, which is a representation of four solar photographs taken from outside the earth's atmosphere. Computers are a portal to new sonic horizons. Computers can process greater amounts of data more quickly than human beings, allowing composers to work at a higher level of abstraction. These systems stimulate composers' imagination. In a 1981 progress report, Hiller wrote "In working with computers, musical ideas come to me that I probably would not otherwise have imagined. This carries over to composing I now do without computers as well" (1981). Comments like these are ubiquitous in automatic composition publications. Early attempts at automatic composition made it clear that both dynamic and complex pieces could be created by computers with minimal human input, but these pieces had an inhuman quality that people found hard to connect with on an emotional level.

Before delving into specific examples, it is important to set the stage by looking at music from a computational thinking standpoint. Algorithms have been used as tools to aid music composition for centuries. Since all music is a balance between chaos and order, these algorithms are used to accomplish the composer's target state of organization. Total chaos is unpalatable white noise, while complete organization, like a single sine tone, is equally unsuitable for entertainment. Algorithms are used to achieve balance between these two states. The rules of counterpoint that were followed strictly by composers until Beethoven are so algorithmic that that most accompaniment parts are deterministically derived from them. For example, Mozart only used a fixed number of modulation types. These rules loosened during the Romantic Era when composers like Wagner and Debussy modulated so freely that the sense of tonal center was blurred. The modern trend is to abandon tonality altogether. The serial system invented by Arnold Schoenberg has the unique property of being heavily systematized, but when listened to, is often heard at the macro level as noise. The other side of the spectrum is the chance music of John Cage, who used everything from cards created by ancient Chinese oracles, the I Ching, to the lines in the sheet music he happens to be writing on to determine a composition. Computers allow finite control over the entropy in music allowing composers to target a certain aesthetic.

Lejaren Hiller and Leonard Isaacson are credited with being the first to program these algorithms into a computer, and generated the *Illiac Suite for String Quartet*. Two examples of generative composition predate them. Louis and Bebe Barron produced the score for *Forbidden Planet* using hardwired electronics – especially ring modulators – by overloading them until they exploded. Although the sounds were produced automatically by the electronics, Bebe Barron composed the score by arranging them. Three months before the debut of the *Illiac Suite*, *Push Button Bertha* aired on television – the product of programming by Douglas Bolitho and Martin L. Klein (Ariza, 2012). *Push Button Bertha* was less of a composition and more of a monophonic melody. The *Illiac Suite* was the first real composition by a computer.

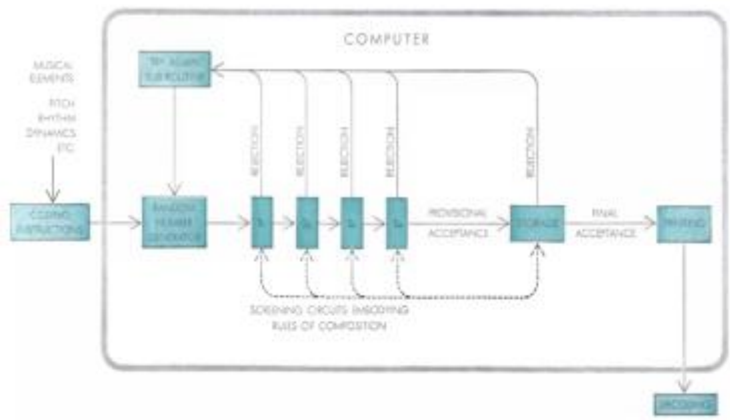


Figure 1 - Figure 1 - Diagram of the program used to produce the Illiac Suite from Hiller's 1959 publication in Scientific American

The basic structure of the program used to compose the *Illiac* Suite is generalizable to most music generation software, and is thus important to understand. The "Suite" is named after the ILLIAC computer at the University of Illinois that composed the quartet. Lejaren Hiller came up with the project and had a significant amount of prior music experience, while Isaacson participated for the "programming challenge" (Hiller 1959). Hiller's groundbreaking and controversial publication, *Computer Music,* published in Scientific American in 1959, like other publications by geniuses of this era including Alan Turing and Watson and Crick, is imbued with prescience. He discusses his project in terms of information theory, a perspective still taken towards generative composition. Information theory, like music theory, organizes information by disorder. Given a word, for example, what is the likelihood of the next word being a certain other word? In music, this equates to the next chord or note. Naturally, a critical aspect of the data that drives the Illiac is a matrix of probabilities of chords that follow one another (see *Figure 2* in *Appendix*). This is one of several layers of "screening circuits embodying rules of computation" shown in *Figure 1*. "Creativity" is manifest in the system with a random number generator that utilizes the Monte Carlo method (Hiller 1959). A random number that corresponds to a certain note passes through several layers of "screening circuits" that comprise voice leading and probabilistic rules. Hiller also coded layers for serial and purely probabilistic operation. If a note breaks one of the set rules, then the program restarts at the beginning of the loop with a new random number. If it passes, however, it is kept until the rest of the notes of that phrase are generated and checked, then is saved for output. Rhythms, phrase structure, the amount of unison or independent voices, and articulation like *pizz* or *archo* are governed by a byte produced by the random number generator between binary 0000 and 1111, decimal 0 and 15. This limits the computer to 16 combinations of rhythms meaning that the shortest note it can play is an eighth note (see *Figure 3* in *Appendix*). Rhythmic and melodic repetition are induced because the probabilities of a certain note occurring adjust to make some notes and phrases repeat. An inverse operation can also be used for a more modern aesthetic. This guess and check method of composition, which has been imitated by numerous systems, limits the organic nature of the resulting piece.

Analyzing the *Illiac Suite for String Quartet* can assist in understanding the limitations of early computer composition, and can give insight into what makes human composition aesthetically different from computer composition. Hiller chose a traditional ensemble instead of electroacoustic production possibly because he wanted to take on the composing tradition using their usual vehicle of expression and to place the actual sound production in traditional player's hands so that they could add a more of a "human" element. Digital production, a medium he often opted for later, was also not sufficiently developed at that time. The suite is a set of four "experiments." The first two follow strict part writing rules, the third serial, and the fourth is probabilistic, and uses a novel method for choosing notes especially suited for a computer. The first experiment is comprised of three movements, *Presto*, *Andante,* and *Allegro.* The first movement sounds surprisingly similar to some of Aaron Copland's work. Phrases cheerily bounce between instruments like the *Allegro* in the first movement in *Appalachian Spring*. However, unlike the *Appalachian* Spring, the melody does not slowly unfold, and is not traded between instruments. Indeed, there is no connecting melody at all. Upon further inspection, the listener realizes that every phrase ends on the tonic and every cadence is an imperfect authentic cadence, crippling any sense of forward momentum, and making the peace sound surprisingly repetitive, given every component's random origin. This defect becomes even more apparent in the second movement, because there is a pause after each phrase (it was likely set on the open setting – see *Figure 3*). Most phrases seem to be around a measure long, which is uncharacteristic of human music which tends to be comprised of phrases from four to eight bars. These long lines are part of what gives music its expressive quality. Although changes in the number of instruments playing and dynamic level occur, there is never a sense of goal direction, of climax, because of the random source of all elements of the music. Often voices climb to what could be a climax before abruptly cadencing. The missing elements are the narrative components that allow music to be a vehicle for emotional content.

However, music does not have to sound "human;" early generative music excelled in writing serialist and probability music, exemplified by the final two experiments of the *Illiac Suite*. In many ways, computers provide the ultimate fulfilment of the serialists' vision. Computers can quickly compute matrixes and allow absolute deterministic control over sound organization and structure. Milton Babbitt is notorious for this approach, and applied serialist procedures to every aspect of his music by using the Mark II RCA synthesizer. However, it often took Babbitt two years of feeding tape into a massive synthesizer to complete a composition. Even the early generative pieces could be produced in less than a week after the program was written. The aesthetic of many serial works and the third *Illiac* experiment is quite similar, because there is little regard for making the music sound "human" or "express" anything. The fourth experiment used Markoff probability chains to select intervals between successive notes, a purely mathematical technique that has little to do with tonality or serialism (Hiller 1959). Hiller remarked in his article that one of the most profound realizations provided by the second two experiments of the *Suite* was that there was very little aesthetic difference between the two (Hiller 1959). This is because of something Xenakis drew attention to in his article, "The Crisis of Serial Music:" "The linear polyphony is destroyed by its own present complexity. One hears, in reality, only aggregations of notes at various registers. The enormous complexity makes it impossible for the ear to follow the tangled lines, and its macroscopic effect is that of an unreasoned and fortuitous dispersion of sounds through-out the entire frequency-spectrum" (Xenakis, 1966). This presents a paradox. Serialist technique determines all musical parameters, but the end sonic result is unintelligible and, as opposed to the strict order it is composed under, sounds random.

Many other experiments were done during the mainframe era of computing, but few attempted tonality in the same way as Hiller did in the *Illiac Suite* – instead taking advantage of the strengths of computers in serial and stochastic music. No discussion of generative music is complete without briefly discussing the pioneering work of Iannis Xenakis and Gottfried Michael Koenig. Xenakis wrote ST/10 in FORTRAN to automate some of his stochastic compositional procedures first implemented in *Achorripsis (*Manning, 2013, p. 203*)*. The resulting compositions made by ST/10 and its successor ST/48, include *080262*, *Morsima-Amorsima*, and *Atrees*. These works are complex, but maintain a distinctly Xenakis sound. At WDR, Gottfried Michael Koenig wrote several composing programs with a wide variety of applications. PROJECT 1 was modeled on traditional principles of serial composition. PROJECT 2 was an improvement on PROJECT 1 and allowed more interactive specification as well as stochastic abilities (Manning, 2013, p. 204). The PODn series confirmed the diverse abilities of granular synthesis, and was used by many composers including Jean Pastiche.

As Stravinsky enjoined, we should cease "tormenting [the composer] with the why instead of seeking for itself the how and thus establish the reasons for his failure or success." Not all music has to have a huge governing idea, and there are areas where computer writing can excel. The early work on music generation during the mainframe era of computing laid the foundation for what is today entire field of research. While it is unlikely computers will be able to compose heavily programmatic piece like Berioz's *Symphony Fantastique* for some time, there is plenty other areas they can excel. Reproductions of classical style, like those created by Emmy, give us a chance to listen to new material inspired by the canonical great composers. Computers have immediate application in serial and stochastic music. Ambient and background music are genres that could benefit from a huge amount of material, as shown by Brian Eno, and would not put any composers out of business. Really, the composer can never be replaced. The greatest music expresses human ideas and emotions. Besides, there must be someone to code and run the programs. However, computer aided composition has unprecedented possibilities. They can inspire us. They process way more data than we ever could. They can be used as a tool explore new concepts and ways of organizing sound, and give us a unique perspective on existing forms. Composing computers make us take a step back to understand how composition works and what makes music a form of expression. Indubitably, this field will continue to grow and evolve. The next Mozart could be a computer.

References

Ariza, Christopher (2012). *Two Pioneering Projects from the Early History of Computer-Aided Algorithmic Composition.* Massachusetts Institute of Technology, MA.

Cope, David (2016, January 26). *Computer Synthesis of Musical Creativity*. California, UCSC. Retrieved from https://www.youtube.com/watch?v=f_ZSR9E_8x0

Garcia, Chris (2015, April 29). *Algorithmic Music – David Cope and EMI*. Retrieved from http://www.computerhistory.org/atchm/algorithmic-music-david-cope-and-emi/

Manning, Peter (2013). *Electronic and Computer Music.* New York, NY: Oxford University Press.

Hiller, Lejaren (1981). *Composing with Computers: A Progress Report.* Buffalo:,NY.

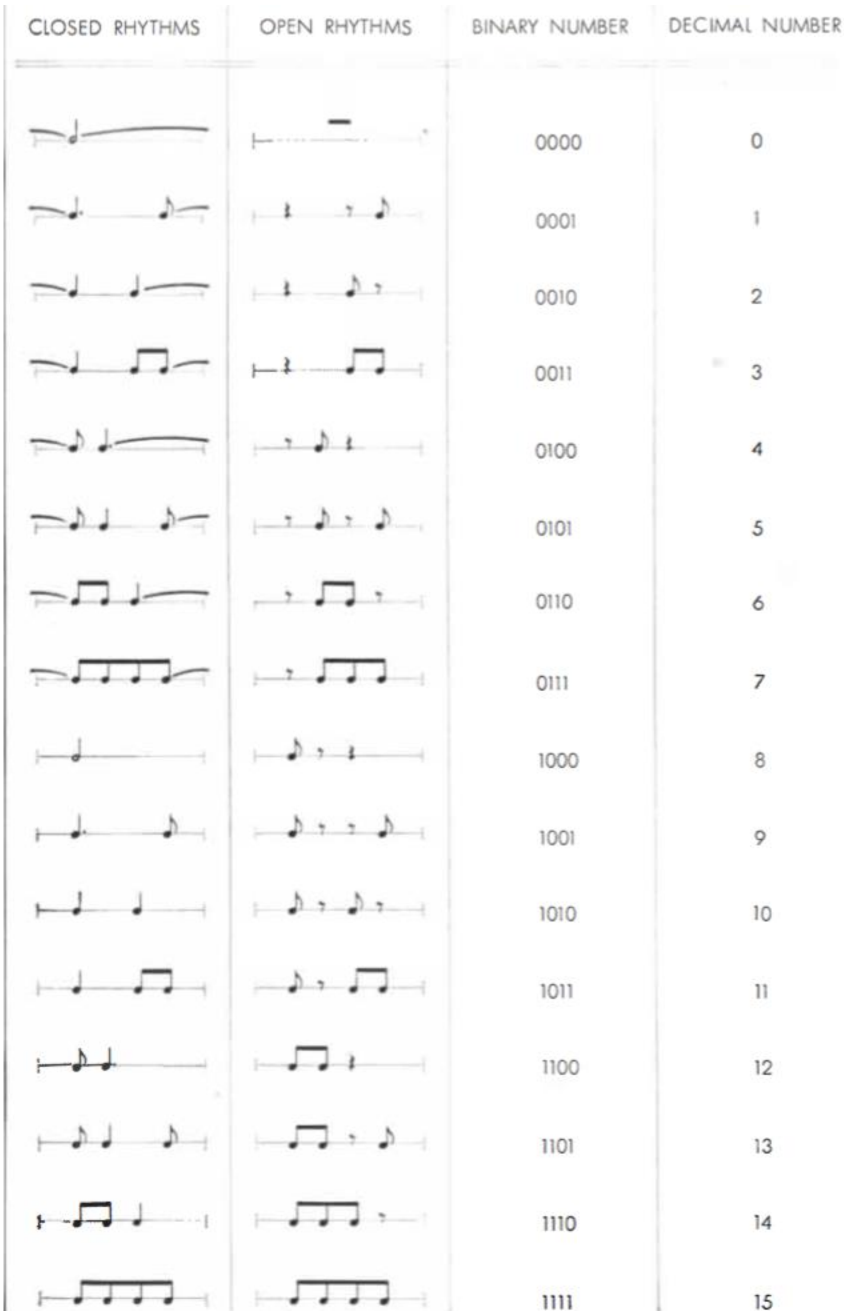
Xenakis, Iannis (1966). *The Origins of Stochastic Music*. Cambridge University Press. Retrieved from jstor.org/stable/942502

Appendix

	a		*b*		*c*		*d*		*e*		*f*		*g*		*h*		*i*	
INTERVALS	W	P	W	P	W	P	W	P	W	P	W	P	W	P	W	P	W	P
UNISON	1	1.00	2	.67	3	.50	4	.40	5	.33	6	.29	7	.25	8	.22	9	.20
OCTAVE	0	.00	1	.33	2	.33	3	.30	4	.27	5	.24	6	.21	7	.19	8	.18
FIFTH	0	.00	0	.00	1	.17	2	.20	3	.20	4	.19	5	.18	6	.17	7	.16
FOURTH	0	.00	0	.00	0	.00	1	.10	2	.13	3	.14	4	.14	5	.14	6	.13
MAJOR 3RD	0	.00	0	.00	0	.00	0	.00	1	.07	2	.09	3	.11	4	.11	5	.11
MINOR 6TH	0	.00	0	.00	0	.00	0	.00	0	.00	1	.05	2	.07	3	.08	4	.09
MINOR 3RD	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	1	.04	2	.06	3	.07
MAJOR 6TH	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	1	.03	2	.04
MAJOR 2ND	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	1	.02
MINOR 7TH	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
MINOR 2ND	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
MAJOR 7TH	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
TRITONE	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00

SHIFTING PROBABILITIES governing one species of Markoff chain music are shown in part here; colored letters key the columns of the table to sections of the music shown on page 118. The probabilities (P) of the intervals are defined by their "weights" (W), which shift according to a simple arithmetical formula. Markoff music has been composed by several similar schemes.

Figure 2



RHYTHMIC CODING for the computer is based on 4/8 meter (chosen arbitrarily) with the eighth-note as the smallest rhythmic unit. All rhythms possible under these restrictions can be expressed as binary numbers equivalent to the decimal numbers from 0 to 15. Random choice determines the order of rhythmic patterns and selects "closed" or "open" form.

Figure 3