Program Notes: *As The Numbers Say* (Short Version)

This work was inspired by a YouTube video, From 0 to 100000, which transcribed the numbers 0 through 100,000 in order onto a MIDI score and played it. The piece was chaotic, yet rhythmic and ordered, with recurring patterns overlapping each other depending on the digits involved. *As The Numbers Say* attempts to take this design philosophy in a different direction. The piece starts at 3, and counts up to 800 at a rate of 217 beats per minute. Each beat, 10 different instruments check a set of conditions that tell them whether or not to play, based on the properties of the current number – and if they play, different properties of that number are used to *how* the instrument plays, including pitch and duration. These heuristics include whether a number is prime, a happy number, or a palindrome in some base, the highest or lowest digits in a particular base, the number's factors or divisors, the number's binary representation, and more. Each instrument was created by hand in CSound format, and the entire piece is constructed and executed using CSound/MUSIC-N as an agent. As the piece continues, the instruments overlap more and more, building almost to cacophony and back.

Biographical Statement

Louis Jacobowitz is a student at North Carolina State University, studying Computer Science and Psychology. From a childhood learning classical-style piano and clarinet, and from a household of classical musicians, Louis likes consistent, deterministic music with an emphasis on rhythm, melody, and harmony. This comes through in his composition, which is oriented more towards structure than randomness.

Program Notes: *As the Numbers Say* (Long Version)

This work was inspired by a YouTube video, From 0 to 100000, which transcribed the numbers 0 through 100,000 in order onto a MIDI score and played it. The piece was chaotic, yet rhythmic and ordered, with recurring patterns overlapping each other depending on the digits involved. *As The Numbers Say* attempts to take this design philosophy in a different direction. This piece starts at 3, and counts up to 800 at a rate of 217 beats per minute. Each beat, 10 different instruments, each created by hand in CSound (together, taking a tour through the various techniques CSound is capable of), check a set of conditions that tell them whether or not to play, based on the properties of the current number. As the piece continues, the instruments overlap more and more, building almost to cacophony and back. A set of python scripts was written to construct and execute the piece:

Name	**Construction**	**Heuristics**	**Activity**
bass	Additive Synthesis via Karplus-Strong on different overtones	Rises chromatically until beat is a prime number; then, drops to the original pitch.	Plays continuously throughout
quick	Additive synthesis – pulse wave fundamental with sine wave overtones	Plays double staccato notes, with pitch depending on the "digital root of (beat minus the sum of its divisors)". When the beat is in the Fibonnaci sequence, pauses for a length determined by the beat's digital root	Comes in early and is active continuously until near the end
long	Additive synthesis, with frequency modulation as each note progresses	Plays long tones, with length twice the number of bases (16 or less) in which the beat is or is not a palindrome, whichever is higher, and pitch corresponding to the lower of the same. Pauses between notes based on number of '2's in the beat in base-3	Comes in early and is active continuously until near the end
high	Additive synthesis of a sine+sawtooth+square wave on the fundamental, and pulse waves for vibrato	Plays chords, with the size of the chord depending on number of ones in the beat's binary representation, then rests for a number of beats determined by the beat's digital sum.	Comes in at around 0:45, remains active until near the end

arpeggio	Basic Additive synthesis of sine waves, with the fourth overtone emphasized	Picks a beat and plays an arpeggio over the following beats depending on its prime factors. Once done, picks the next non-prime beat.	Comes in around a minute in, and remains active until near the end
buzzy	Uses "buzz" with sawtooth+square waves, but changes the harmonics throughout duration.	Pitch is based on the digital root of the beat's cube. When beat is divisible by its digital root but not by 3, pauses depending on the difference between the beat's highest and lowest digit	Comes in three separate times: around 0:30, 1:45, and 2:30
erratic	A "drumbell" sound, adapted from ACCCI 03, mostly additive synthesis	Plays the tonic, fourth, or fifth depending on whether the beat is divisible by its last digit and whether it's even	Comes in three separate times: around 1:10, 1:55, and 2:40
churchbell	Uses csound's "vibes" command, adding one strike with vibrato and one without	Plays only when the number of '1's in the beat's binary representation is also a factor of the beat, with pitch determined by lowest digit.	Is continuously active from about 1:15 to 3:00
happy	Additive synthesis of some csound waveguides: "wbow" sourced with a sawtooth wave, and "wclar" with a pulse wave	Toggles on and off whenever the beat is a "happy number", with pitch determined by how many bases in which the beat is happy.	Comes in three separate times: around 1:20, 2:00, and 2:45
ascent	Uses csound's "buzz" command given both a sine and square wave, and pulse waves for vibrato	Pitch depends on the beat's highest digit in base 12. Plays a new note whenever that changes, and holds until then.	Comes in three separate times: around 1:30, 2:20, and 3:45