## Singling: Developing a Program to Sonify Texts

We developed Singling as an interactive interface for the transposition of textual data into MIDI code which can then be sounded according to a user-generated assemblages of sonic attributes. Singling converts the properties and relations of the input data via systematic transformations, where the output is completely deterministic with no randomness in the sound that is produced from the same input text, operationalizing Singling as a sonification tool (Hermann; Walker and Nees). This section discusses the creation and affordances of Singling. First, we describe our process of creating linguistic parameters and framework that would condition the input data; then we detail the transformations of text to sound that occur within Singling; and, finally, we discuss our use of MIDI as output and new transformation of the sonified data.

### The sound of letters and words

We are working with frequency as the basic unit as it is a value that can be easily manipulated to output a note programmatically (Wolfe). From there, we use jFugue to convert frequency (wavelength) to MIDI notes and microtones, calculated using a chosen alphabetic string or higher level lexicogrammatical features as determiners of the note played, normalized to fit in the range of a target octave defined by the user.

Our initial work with TextSound (Kohll) was based on letter frequency and the sonification of individual alphabetic characters of a text mapped to the overtone series (see Figure 1). Letter frequency is a feature of text often studied by cryptographers and is useful in modern data-compression techniques such as Huffman coding. Letter frequency data can only be gleaned by analysis of large amounts of representative texts, a task well suited to modern computing and has been shown to be a kind of dna test of the writer and subject of the text. “Different authors have habits which can be reflected in their use of letters. Hemingway’s writing style, for example, is visibly different from Faulkner’s” (World Heritage Encyclopedia).

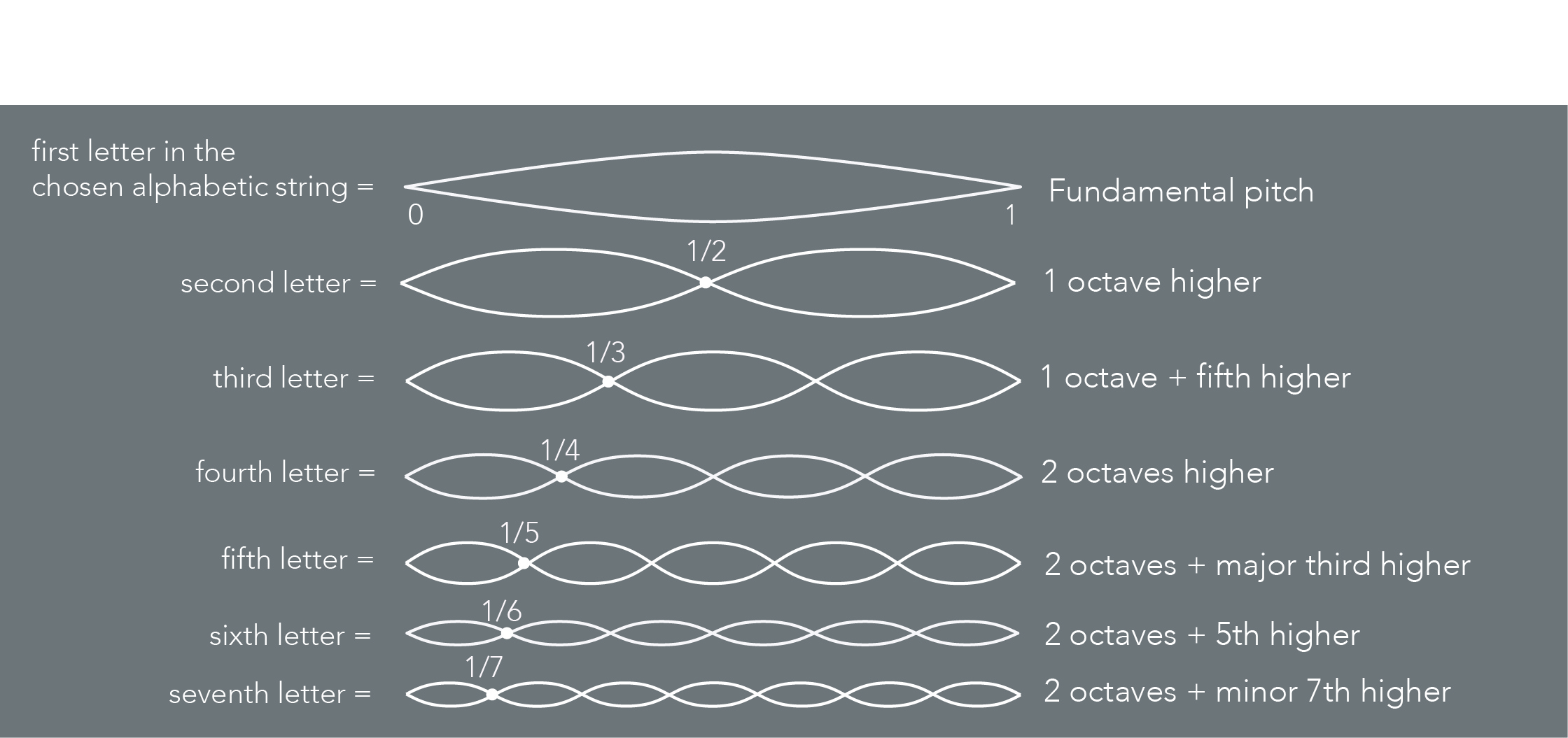


Figure 1. Alphabetic string as it corresponds to the overtone series represented as wavelength

We have included three different alphabetic frequencies derived from three different methodologies. In Mayzner’s widely cited publication *Tables of Single-letter and Digram Frequency Counts for Various Word-length and Letter-position Combinations,* he culled a corpus of 20,000 English words for his letter and word frequency tables. In 2012, the 85 year old Maynzer reached out to Norvig at Google to use “the computing power that is now available to significantly expand and produce such tables as I constructed some 50 years ago” (Norvig). This frequency listing is “37 million times more than” Mayzner's 20,000-mention collection”: ETAOINSRHLDCUMFPGWYBVKXJQZ. We also include the widely referred to standard order frequency used by typesetters (“Letter frequency”): ETAOINSHRDLUCMFWYPVBGKQJXZ. As well as the letter frequency based upon the Oxford dictionary (“Oxford Languages: The Home of Language Data”): EARIOTNSLCUDPMHGBFYWKVXZJQ. Users will be able to program Singling to use any word frequency analysis they choose: however, these are the three we currently have included.

There are many use cases for examining and experiencing language at the level of the letter, but we also desired a broader, more bird’s eye view of the textual field. When we discovered the lexical ontology of WordNet, the scope of our textual territory radically shifted. Underneath the hood, we have expanded TextSound’s function to convert words in addition to characters. To achieve this we relied on WordNet, a linguistic database created by Princeton University’s Cognitive Science Laboratory (Princeton University). WordNet is a semantic network that interlinks words employing lexical and conceptual relations. It is composed of synonym sets (called *synsets*) that groups synonymous words, and where all the words belonging to a single synset express the same concept (Fellbaum), but not every word in a synset is exchangeable with the other words in all contexts and a single word might appear in different synsets in the database. For example, the synset 01013367 groups the words Inventory, Inventorying, Stocktaking and Stock-taking; but the word Inventory is also in the synset 04328672 along with the word Stock.

WordNet 3.1, the most recent version of the database that was used for Singling, is organized in four components: nouns, verbs, adjectives, and adverbs. Each component is further organized into lexicographer files (called *LexNames*) that are based on syntactic categories and logical groupings. For example, nouns are organized into 26 LexNames (that are identified with the ID’s from 03 to 26) such as body nouns, animal nouns or person nouns. Like other research projects,“the availability of the WordNet database was an important starting point… [because] WordNet covers the entire English lexicon and provides an extraordinarily large amount of conceptual distinctions. As well, it is particularly useful from a computational point of view because it was developed for easy access and navigation through its hierarchies” (Strapparava and Valitutti 1083).

WordNet databases (which are available online or as downloadable files) contain information such as synset ID, LexName ID, type (whether it is a noun, adjective, etc.), words included within the synset and relations to other synsets (such as hyponymy and meronymy). To make the databases adjust to our needs, we reorganized the information available in their database. We discarded the more granular and complex meaning-mapping work of synonym sets, which comprises the structural field of the WordNet ontology, but which was problematic for our purposes. Word meaning is an ongoing problem for computational lexicographers because “[w]ord meaning is in principle infinitely variable and context sensitive. It does not divide up easily into distinct or discrete sub-meanings. Lexicographers frequently discover in corpora loose and overlapping word meanings, and standard or conventional meanings extended, modulated, and exploited in a bewildering variety of ways” (Edmonds and Eneko). While we have retained WordNet’s LexName identifiers, we are using them in a very different way, and for this reason have renamed them Lexicogrammatical Classes (LGC).

Consider the aspects of the word *map* as shown in figure 2:



Figure 2. Possible transformation parameters for the word Map in Singling

Within this rich polysemous ontology of meaning we were able to decipher the broad categorical features of the textual landscape we wished to sonify in our platform: word category, 45 LGC as established by the WordNet creators, and now a distinctive feature of their widely applied ontology, as well as the breadth of polysemy of individual words (the quantity of lexicographic categories a word belongs to).

After the reorganization of the WordNet databases to best fit our work on Singling, we were interested in exploring how sentiment analysis could be integrated into the program. Because of this, we added SentiWordNet (Baccianella, Esuli and Sebastian), which is compatible with WordNet and contains an automatic annotation of positive, objective and negative dimensions of words in relation to the synsets. The annotations in this database, assigned with a value between 0 and 1 to each dimension, are not limited to a single value per word, but assess how the ﻿same term might have different opinion-related properties. For example, the word Common is considered to be somewhat positive (with a 0.25 value) and somewhat negative (with a 0.25 value too) at the same time, showing how it can consider different contextual and semantic features of the words.

In order to further explore the semantic and affective dimensions of the textual data into Singling, we added other linguistic components to the database. In the first place, we added Modals to our database by following the same structure that was previously discussed. Following James (“Toward a Theory of Choice through Visuals”), we also included sentiment analysis data to the modals by following a simple pattern where 1 as a value determine its necessity (for example, *Must* has 1 in positivity and *Must Not* has 1 in negativity), and values closer to 0 represent possibility (for example, Would and Would Not). Modals were assigned 45 as a LGC.

The following step was adding symbols that would allow adding sound features to the database when encountered with one of those. These symbols were assigned the LGC 46. Furthermore, and hoping to add features that would allow working with social media data in the future, emojis were included in the database and assigned 47 as a LGC. For this we used Novak and colleagues’ database which rates emojis based on how frequently they were being used. This database also included how many times each emoji was used along with a positive, neutral or negative text, allowing us to integrate sentiment analysis into it. For this, the number of occurrences in each of the three instances (positive, neutral and negative) was transformed into a value from 0 to 1 —following the same structure of SentiWordNet.

Finally, we added a database with prepositions. To achieve this, we used the publicly available list of prepositions in Wiktionary (“English prepositions”) that contains 650 words and phrases. Out of this process, we eliminated all phrases since they are incompatible with Singling. Then, looking to add complexity to our database, we open-coded the prepositions to add LGCs into the database, resulting in three new LGCs: 48) agential prepositions that denote a position in relation to the self; 49) relation prepositions that talk about the relation between the self and another subject or object; 50) time prepositions that refer to relations of the self over time. Table 1 shows the final list of all the linguistic components and LGC used in Singling.

Table 1

LGCs in the database of Singling, modified from “LexNames”. Fields highlighted in blue means that these LGCs were modified by the authors, and fields highlighted in red were added.

|  |  |  |
| --- | --- | --- |
| **LGCs** | **Category** | **Details** |
| 00, 01, 44 | Adjectives | [00] Adjective all, [01] pertainmyms, [44] participles |
| 02 | Adverbs | [02] Adverbs all |
| 03 - 28 | Nouns | [03] Noun tops, [04] acts, [05] animals, [06] artifacts, [07] attribute, [08] body, [09] cognition, [10] communication, [11] events, [12] feeling, [13] food, [14] group, [15] location, [16] motive, [17] object, [18] person, [19] phenomenon, [20] plant, [21] possession, [22] process, [23] quantity, [24] relation, [25] shape, [26] state, [27] substance, [28] time |
| 29 - 43 | Verbs | [29] Verbs of body, [30] change, [31] cognition, [32] communication, [33] competition, [34] consumption, [35] contact, [36] creation, [37] emotion, [38] motion, [39] perception, [40] possession, [41] social, [42] stative, [43] weather |
| 45 | Modals | [45]Modals all |
| 46 | Symbols | [46] Symbols all |
| 47 | Emojis | [47] Emojis all |
| 48 - 50 | Prepositions | [48] Agential prepositions, [49] relational, [50] time |

The specific features or categories of language that we have identified for as data for consideration has been an evolving discussion that moves us between the formally granular, and the broader conceptual and grammatical category. Ours is a lexigrammatical approach to language that seeks to bring fresh and unexpected insights to our understanding of linguistic knowing and expressing. As Borges writes, “the impossibility of penetrating the divine scheme of the universe cannot dissuade us from outlining human schemes, even though we are aware that they are provisional” (104).

### Sonified texts: transformation processes in Singling

Along with the work on integrating new linguistic features into Singling, we explored different sonifications processes to transform the textual data into sounds. Singling’s sound transformation module aims to provide users precise and customisable control over the musical outputs that the program creates. Transformations can either be configured globally or triggered by text modifiers such as word type (nouns, verbs, adjectives, etc.), LGC, word length and symbols/characters. The list of text modifiers and their applicable sound transformations are listed in the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Text modifier types** | **Comparison operators** | **Text modifier values** | **Sound modifiers** |
| WORD\_CATEGORY | EQUAL\_TO | noun (n), verb (v), adj (a), adv (r), modal (m), symbol (s), prepositions (p) | TEMPO, NOTE\_DURATION, OCTAVE, INSTRUMENT, VOLUME, MIDI\_NOTE, ATTACK, DECAY |
| WORD\_LENGTH | EQUAL\_TO, GREATER\_THAN, LESS\_THAN | any integer | TEMPO, NOTE\_DURATION, OCTAVE, INSTRUMENT, VOLUME, MIDI\_NOTE, ATTACK, DECAY |
| LEXICOGRAMMATICAL\_CLASS (LGC). | EQUAL\_TO | 1-50 | TEMPO, NOTE\_DURATION, OCTAVE, INSTRUMENT, VOLUME, MIDI\_NOTE, ATTACK, DECAY |
| SYMBOLS | EQUAL\_TO | list of symbols from database | TEMPO, NOTE\_DURATION, OCTAVE, INSTRUMENT, VOLUME, MIDI\_NOTE, ATTACK, DECAY |
| CHARACTER | EQUAL\_TO | consonants and vowels | TEMPO, NOTE\_DURATION, OCTAVE, INSTRUMENT, VOLUME, MIDI\_NOTE, ATTACK, DECAY |

Table 2. Transformational modifiers and their parameters in Singling

Transformations in Singling are cumulative (meaning, you could add different sound transformations within the same sonification), however inputs with the same parameters cannot be entered more than once. For example, if nouns are set to decrease tempo by x, then nouns cannot also be set to also increase tempo by y. We hope to introduce many more types of sound transformations in future versions of the program, such as modulation of frequency using pitch bend.

Another important feature of Singling is the option for users to choose the default behaviour of each note. In this case, users can choose from three defaults behaviour: 1) Overtones: Default mode of operation where notes are determined by the ordering of alphabetic frequencies with relations to the overtone series; 2) Static: Every note will sound the same, determined by base settings, unless otherwise instructed by sound transformation(s); 3) Mute: Notes are muted, unless otherwise instructed by sound transformation(s). These predetermined features would enable users to interact with the textual data in different ways, adjusting the process to different contexts of uses of Singling. The application can also use jFugue’s RealTimePlayer function to convert text to sound notes in real-time instead of constructing and sending a full string of musical notes to be converted to MIDI for playback.

There are instances where words will fall into multiple syntactic categories or LGCs. In such cases, additional channels or *voices* (as referred to by JFugue) are used to sonify each of the LGCs in harmony. One constraint with this method is that the MIDI specification has a limit of 16 voices, one of each is reserved for the use of percussion instruments specifically which we are not covering in this version of Singling. As a result, we are limited to sonifying a maximum of 15 separate notes related to a single word in the database; although in practicality, it is difficult for the listeners to distinguish that many individual notes at the same time, so we are working with this limitation for the current iteration of the program.

Finally, one of the most immediately obvious changes in Singling in comparison to other programs (including TextSound) is the graphical user interface (GUI) to drive the program. This GUI, as shown in Figure 4, allows the user to adjust some basic sound default parameters for the sonification (at the left of the figure), input the data in a textual box (in the centre of the figure) and add or remove sound transformations (with the scrolling panel visible at the right of the figure). This GUI allows users to interact easily with Singling, adding cumulative sounds transformations to the textual input, rather than relying on command line arguments as a way to make this more user-friendly.