MusicType: ECE243 Final Project

Ketevan Gallagher and Mujtaniba Islam

1. Overview and User Guide

Overview: Write and Play Mode

MusicType is a text-based music creator, inspired by the programming language Strudel. Users can input text in Write Mode, and then when they switch to Play Mode, the symbols and commands they have written will be converted to music. A graphic that shows the notes and sounds being played is also displayed on the bottom half of the screen. Notes will light up as they are played in this graphics display.

The program automatically starts in Write Mode. To enter Play Mode, the user must press the tab key. To go back to Write Mode, the user must press the escape key. Text that is written in Write Mode is saved during Play Mode, and the cursor placement does not change during Play Mode.

Figure 1. An example of MusicType in Write Mode

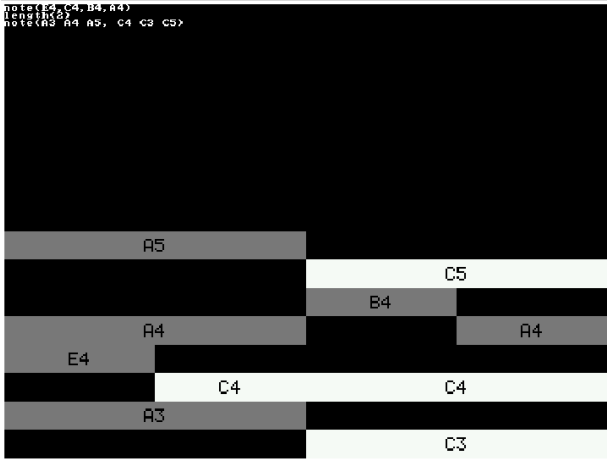


Figure 2. An example of MusicType in Play Mode

Notes

Users can play notes using the note() command. Notes separated by commas are played in series while those separated by spaces are played in parallel. Multiple note() commands are played in parallel. Users can type in notes A through G and specify the octave of the note with a number after. For example, A4 specifies the note A in the fourth octave. Users can also create drum sounds. The drum commands are:

- kd = kick
- sd = snare
- rim = rimshot
- hh = hihat
- oh = openhat
- cp = clap

Length

A note() line is two seconds by default. However, users can use the length() command which takes in an integer to change the length of a note line. All note lines after the length() command are the length specified by the length command.

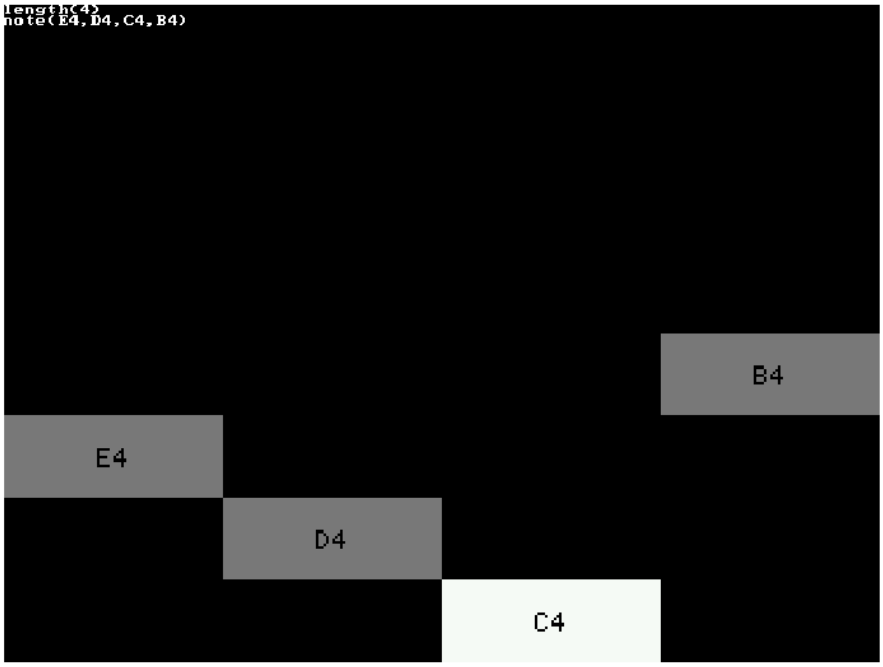


Figure 3. An example of the length command being used to change the length of the notes.

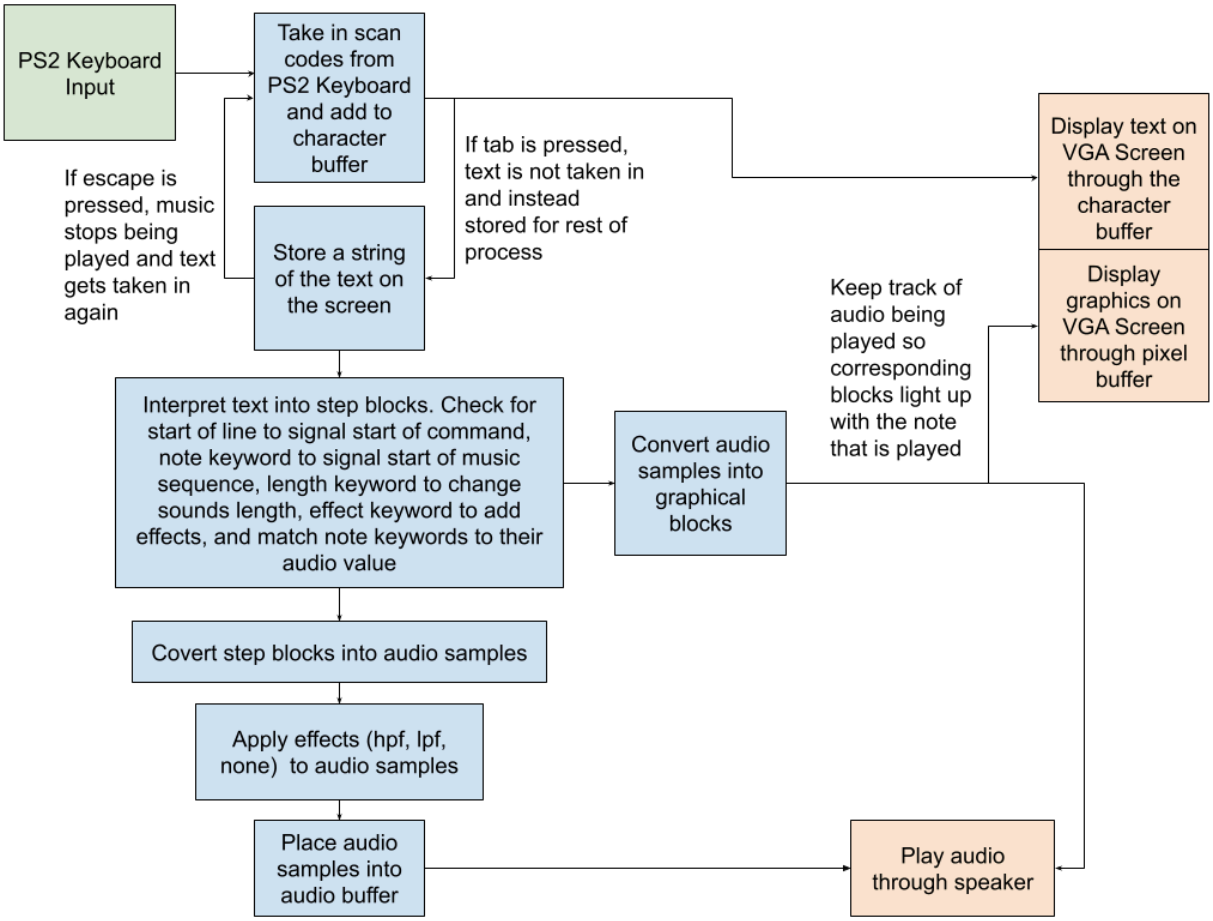
Effects

Effects can also be applied to note lines. The effect() command takes a three character string and integer separated by a comma. The three types of effects and their three character symbol are:

- lpf = low pass filter
- hpf = high pass filter
- non = none

The integer should be between 0 and 100. The closer to 0, the more the effect is applied. The closer to 100, the less the effect is applied. For example, effect(lpf,30) would produce a more muffled sound than effect(lpf,70). The effect command is applied to all note lines after it. The formula for the low pass filter and the high pass filter were both taken from Wikipedia.

2. Block Diagram



3. Next Steps

We'd like to implement more features that are in the Strudel programming language, such as brackets and multiplication so users can add longer sequences more easily. We'd also like to implement other effects in addition to a simple low pass and high pass filter, such as reverb or echo. Finally, on the visual side, we'd like to have the notes scroll across the screen. This would also make it easier to see longer sequences, as currently when a user inputs many notes they can become quite narrow.

4. Attribution Table

Ketevan Gallagher	Mujtaniba Islam
Main function to organize when the program is in write and play mode, and when to process text	Audio data structures (stepBlocks, stepSequences, etc)
Taking in text from the PS/2 keyboard and converting scan code to characters	Drum sample and note data
Interpreting text: ● Converting note() into StepSequences ● Converting length() into a sound length for each note() line ● Updating effect type and filter parameters for effect()	Audio system: ● Sample-based audio generation with per-sequence step-timing ● Support polyphonic notes, drum sample playback, and real-time mixing across sequences
Displaying text on the screen as the user types it	Visualizer system: ● Grid-based display with horizontal step mapping and vertical pitch/drum separation ● Dynamically map notes to rows and render labeled note/drum blocks ● Highlighting
Clearing text and the keyboard buffer	Audio-Visual sync: ● Visuals driven directly by audio state ● Highlight aligned with currently playing sound
Switching between write and play mode	Clearing audio FIFO
Apply effects such as low pass filter and high pass filters to audio samples	