

# CMDrum

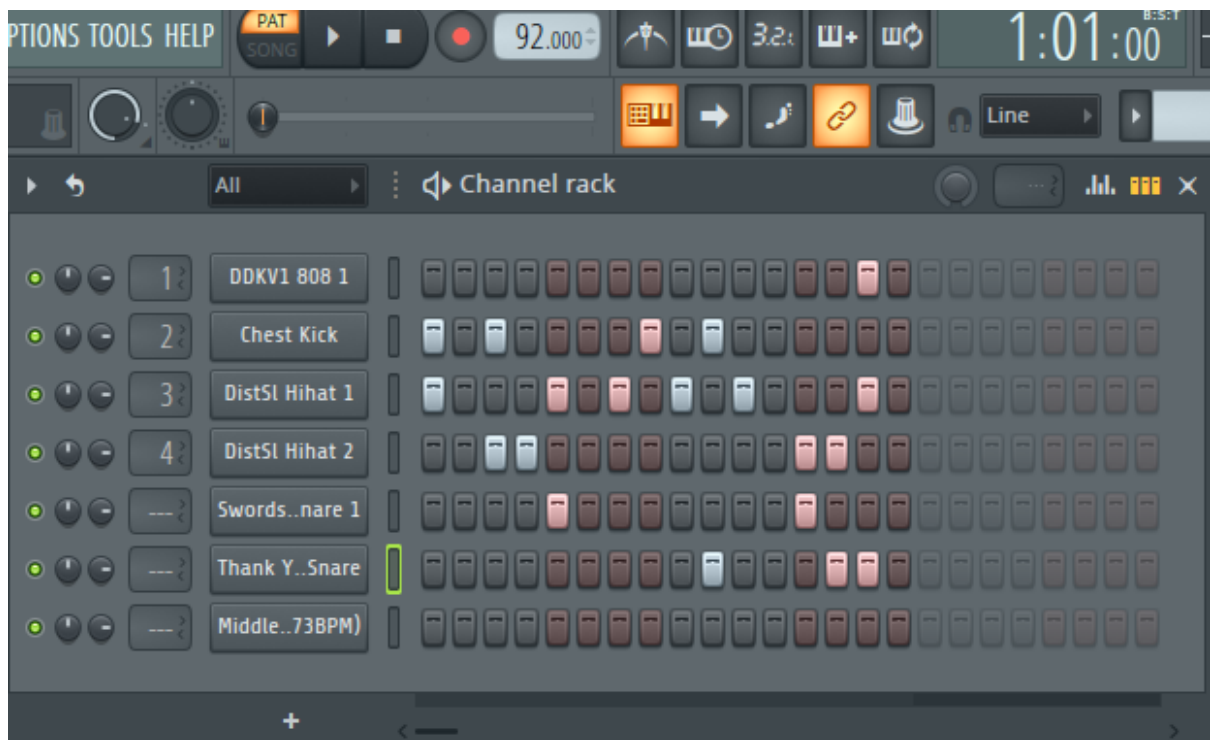
Created By: Alex Haszard | Ben Addison | Dante Vannini | Seb Chkourko

## Team Plan

Our Team 'Percussion Protocol' consists of Alex, Ben, Dante and Seb. We've identified that our team strengths vary with a couple of us being good at programming and the other's being good at management/documentation. We aim to schedule frequent meetings (weekly) to discuss our next week's agenda in more detail than outlined in lectures. This gives us clarity in achieving our milestones for the project. Some tools we are using include Facebook Messenger for remote communication and Trello for team to-do list scheduling.

## What are we going to make?

We are going to make a simple sample-based drum machine in the command line. It will allow users to program simple drum patterns using a step-sequencer GUI designed with terminal characters. The interface will be very reminiscent of the widely used DAW FL Studio's channel rack, which consists of a grid of 'steps' where the users can enter notes and play them back at a user chosen tempo.



*FL Studio's Channel Rack*

Our drum machine will strip down a lot of the functionality and focus on the essence of creating a rhythm. Like the above, would include sounds on the left and the grid on the right. We believe this project will be good for us as our team is knowledgeable in the music production industry and has plenty of experience with similar devices.

## How is it ‘back to basics’:

<https://www.jstor.org/stable/3681976?seq=1> - This article explains how simple digital instruments with different experimental methods of controlling it can allow for “*introducing the practice of experimental electronic music to newcomers while trying to remain attractive to more advanced electronic musicians.*”

We have isolated the “drum machine” as the digital instrument we wish to recreate, and the low-level nature of the command-line interface with keyboard controls can allow for users to interact with our software in the simplest way possible. Considering that, interacting with such software needs to be as straightforward and intuitive as possible. A hands-on device with physical buttons and knobs inspires creativity and is typically much easier to learn than software alone (reference). Our drum machine will focus its drum programming on the computer keyboard to mimic real hardware for programming drum sounds. Meaning our software could be described as “back to basics,” as it allows for people to craft rhythms easily without cluttering the workflow.

## How are we going to build it?

At the application level, there are multiple components we will need to program to make the drum machine function properly.

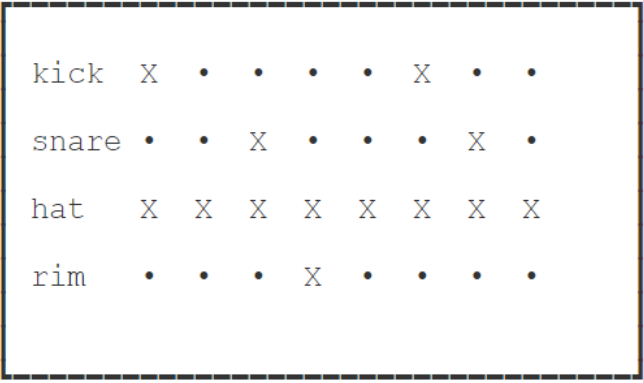
These include:

### Keyboard input

The user can select which track they are editing with the arrow keys and program the sequence using QWERTYUI. Each letter corresponds to 1-8 steps allowing for a rapid laydown and easy editing. One idea is to implement velocity (step volume) which can be done by holding a step and pressing numbers 1-9 but would depend on the capabilities of the audio engine

### Interface

Below are some mockups for interface ideas.



|       |   |   |   |   |   |   |   |   |
|-------|---|---|---|---|---|---|---|---|
| kick  | X | • | • | • | • | X | • | • |
| snare | • | • | X | • | • | • | X | • |
| hat   | X | X | X | X | X | X | X | X |
| rim   | • | • | • | X | • | • | • | • |

*Simple sequencer idea represented using Unicode in terminal*

```

- rand items:
-   patch
-   row

- shift items:
- shift row left
- shift row right

```

[rand]

[shift]

|       |   |   |   |   |   |   |   |   |
|-------|---|---|---|---|---|---|---|---|
| kick  | • | 2 | • | 5 | 1 | • | • | • |
| snare | • | • | 9 | • | • | • | 9 | • |
| hat   | 9 | 4 | 9 | 4 | 9 | 9 | 4 | • |
| rim   | • | • | • | • | • | • | • | • |

simple inspector displaying crucial info about steps

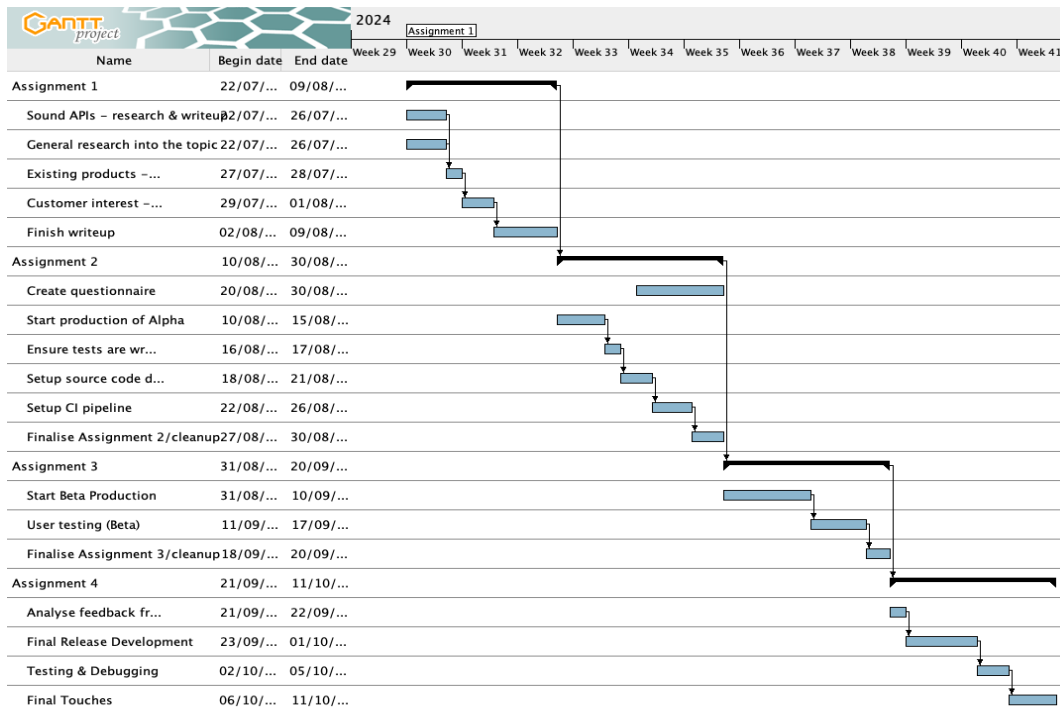
*Additional idea with numbers indicating velocities ('rand' and 'shift' are menus toggled with a shortcut)*

## Audio Engine

Some prototyping has already been underway to assess what our engine should be capable of for a simple sample-based drum machine. Windows provides a universal sound playing function called 'Playsound' through their multimedia functions. While it works for just playing sounds, it's somewhat unstable when playing audio files too quickly, especially when the sound hasn't finished playing. We are implementing a much more flexible audio engine using 'waveOut' which is a low-level API provided in the Windows.h library. This engine thus far is capable of loading .wav files and playing them but cannot play multiple sound events simultaneously. This is a feature that we intend to have but isn't a priority.

## Development Process - Agile

- GitHub actions for Continuous integration
  - o Cmake visual studio
- Weekly meetings
- Trello for task assignment
- Charts
  - o [Pert chart](#)
  - o Gantt Chart



## Existing Products

We researched a few similar products to gain inspiration.



This is the drum sequencer from FL Studio. You drag and drop audio files into each rack to add new sounds, and each step represents  $\frac{1}{4}$  of a beat. The panel in the top left titled “All” allows you to select different groups, which causes only sounds from the selected group to be shown. You can click anywhere in the grid to enter a step.



This is the Behringer RD-6 and is a physical sequencer, with the 16 steps on the bottom being used to sequence multiple different sounds. This is done by first selecting (play->write) mode, and then in the instrument section select the instrument you wish to sequence. This is then sequenced by selecting the grey step buttons.



This is Tahti studio, an online drum machine/sequencer. Each of the top tabs (numbered 1 through 8) is the step sequence for a different audio sample. You sequence each sample by selecting its tab, then filling the slots in the sequencer by left clicking. Tempo can be changed in the top left with a by left clicking and dragging the mouse up and down. You can also add effects to each sample in the effects rack.

All these products share a similar sequencer design which is the main inspiration for our terminal-adapted interface. We are planning to use a similar multi line display to FL studio, where we can see the sequence of multiple different sounds displayed together. However, we may use a similar method for writing each sequence as the Behringer rd-6; by selecting which channel you wish to sequence while in some sort of “write mode”, setting up the sequence, and then saving and playing back from play mode. We may also try to add some effects, as in Tahti studio, although this would be more of an extra feature if we find ourselves with time at the end of the project and is not a core requirement.

## Customer Interest

Referencing the ‘back to basics’ section, musical instruments that balance capability with ease-of-use are widely searched for. While some instruments are highly capable, they are often paired with an interface that can be difficult to navigate, as some examples from a reddit thread mention.

A lot of people look for easy ways to get started with music and are often drawn away by advanced tools, so our simplistic direction may allow for more new musicians. It could also be used in an education setting, as it is so simple it allows for users who know almost nothing about music production to make simple beats.

Our tool can be used for not only music production, but also other areas such as live sets or quickly jotting down ideas.

During the user testing, we should test with non-experienced users to make sure that our program can be picked up easily. If we find users are struggling, we could provide a tutorial or manual which would be useful for beginners.

Some evidence of customer interest through online discussion:

[https://www.reddit.com/r/WeAreTheMusicMakers/comments/2xdi8j/is\\_there\\_a\\_free\\_basic\\_and\\_simple\\_drum\\_machine/](https://www.reddit.com/r/WeAreTheMusicMakers/comments/2xdi8j/is_there_a_free_basic_and_simple_drum_machine/)

Most responses involve either a whole DAW or downloading your own sounds, etc. We think our software would be the ideal solution for this kind of user.

[https://www.reddit.com/r/synthesizers/comments/17q7s5j/cheapest\\_and\\_easiest\\_drum\\_machine\\_for\\_a\\_live\\_band/](https://www.reddit.com/r/synthesizers/comments/17q7s5j/cheapest_and_easiest_drum_machine_for_a_live_band/)

This one also has the same issue, might point us more towards our product being used for live sets which may involve arrangement of sequences, and including some “live” sounding drums.

[https://www.reddit.com/r/WeAreTheMusicMakers/comments/pe10a/tips\\_for\\_writing\\_drum\\_riffs\\_im\\_a\\_guitarist\\_not\\_a/](https://www.reddit.com/r/WeAreTheMusicMakers/comments/pe10a/tips_for_writing_drum_riffs_im_a_guitarist_not_a/)

This one is a guitarist who wants to quickly jot down ideas for drum riffs. The top response is about simplicity, so we believe our idea would be great for this kind of use-case.

<https://www.sinesquares.net/musicgear/hardware-vs-software>

This article discussed differences between hardware and software music production. While it is broad and stems outside the nature of our product. Some valuable points were made such as software being prone to feature bloat. Software technically has no limit to the amount of information you fit into a single window which leads to a lot of music production software full of every feature somebody might need. It also makes points on how professional producers use both hardware and software for specific purposes in the studio. We’ve adapted this key idea into our user input method (using the keyboard as a sequencer).