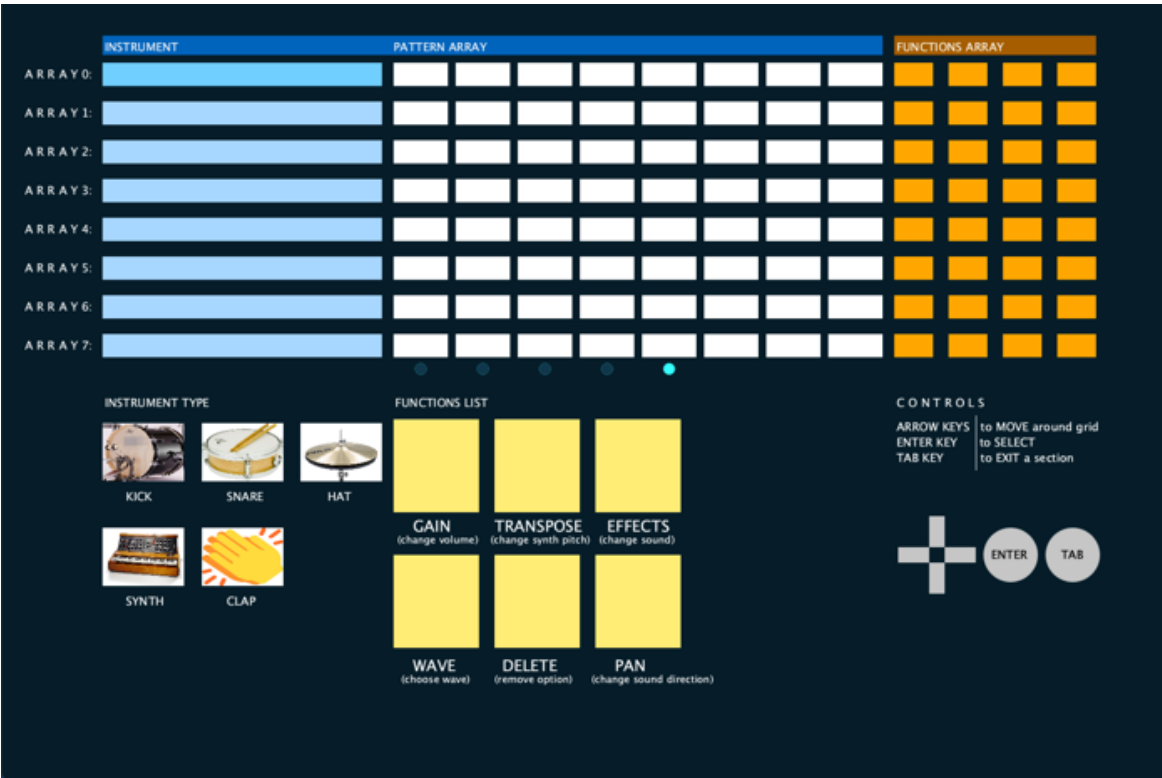


# MIC-TOK: A MUSICAL LIVE CODING ENVIRONMENT FOR CONSTRAINED INPUT



by

Trusting Okechukwu Inekwe

**MIC-TOK: A MUSICAL LIVE CODING ENVIRONMENT FOR CONSTRAINED  
INPUT**

**Prepared by**

Trusting Okechukwu Inekwe

[toinekwe@wpi.edu](mailto:toinekwe@wpi.edu)

Worcester Polytechnic Institute

A thesis submitted in partial fulfillment of the requirements for the Degree of  
Master of Science  
in  
Computer Science

Supervisors

Gillian Smith

Charles Roberts



## **Abstract**

Live coding—a popular, growing field, where performers code on stage—is used in many different contexts, including music, art, and dance. Typical live coding environments require interfaces that use dozens of inputs for a performance and some level of familiarity with basic programming for operability. This makes them potentially inaccessible to some users. Our research attempts to make musical live coding less complex by designing Mic-Tok; a musical live coding environment for a constrained set of inputs. We created an initial version of Mic-Tok using a combination of existing design methods and the Cognitive Dimensions of Notation Framework to serve as a guide. We conducted formative evaluations to inform the design of our tool. With observations from each evaluation, we iteratively modified Mic-Tok. After finalizing our modifications, we conducted summative evaluations to evaluate user perception of the effectiveness of Mic-Tok in making live coding more accessible. Analysis of results from our user study suggest that users perceive Mic-Tok as approachable and expressive.

# Table of Contents

<b>CHAPTER 1. INTRODUCTION</b> .....	<b>5</b>
<b>CHAPTER 2. RELATED WORK</b> .....	<b>1</b>
2.1 LIVE CODING ENVIRONMENTS .....	1
<b>CHAPTER 3. MIC-TOK ENVIRONMENT</b> .....	<b>3</b>
3.1 IMPLEMENTATION .....	3
3.2 MIC-TOK’S ARCHITECTURE .....	2
3.2.1 Notes.....	2
3.2.2 Patterns:.....	3
3.2.3 Functions:.....	3
3.3 PERFORMANCE ENVIRONMENT.....	5
3.3.1 Software GUI.....	5
3.4 FEATURES .....	6
3.4.1 A one-to-one mapping of six words to six buttons for operability.....	7
3.4.2 Tempo.....	7
3.4.3 Rhythm and Timing .....	7
3.4.4 Modulations.....	7
3.4.5 Sequencing .....	8
3.5 A PROGRAMMING SESSION WITH MIC-TOK .....	8
3.5.1 Creating a Pattern.....	8
3.5.2 Adding a function .....	9
3.6 AN ANALYSIS OF THE MIC-TOK ENVIRONMENT USING THE COGNITIVE DIMENSIONS OF NOTATION FRAMEWORK .....	10
<b>CHAPTER 4. RESEARCH METHODOLOGY</b> .....	<b>14</b>
4.1 FORMATIVE EVALUATION.....	14
4.1.1 Setting and Audience.....	14
4.1.2 Formative Study Design/Procedure .....	15
4.2 SUMMATIVE EVALUATION .....	16
4.2.1 Setting and Audience.....	16
4.2.2 Summative Study Design/Procedure .....	16
<b>CHAPTER 5. RESULTS</b> .....	<b>19</b>
5.1 FORMATIVE STUDY RESULTS AND MODIFICATIONS MADE .....	19

5.2 SUMMATIVE STUDY RESULTS .....	21
5.2.1 <i>Demographics</i> .....	21
5.2.2 <i>Cases Studies from Summative evaluation</i> .....	22
5.2.3 <i>Summary of Observations from Case Studies</i> .....	24
5.2.4 <i>User perceptions on approachability of Mic-Tok</i> .....	24
5.2.5 <i>User perception on expressivity of our tool using the Creativity Support Index</i> .....	27
<b>CHAPTER 6. DISCUSSION AND CONCLUSION</b> .....	<b>30</b>
6.1 RESEARCH QUESTION 1: WHAT MUSICAL LIVECODING ENVIRONMENT CAN BE DESIGNED FOR A CONSTRAINED SET OF INPUTS .....	30
6.2 RESEARCH QUESTION 2: HOW DO USERS PERCEIVE MIC-TOK’S USABILITY AND EXPRESSIVE POTENTIAL.....	30
6.3 FUTURE WORK .....	31
<b>CHAPTER 7. REFERENCES</b> .....	<b>32</b>

# Table of Figures

FIGURE 1: MIC-TOK ENVIRONMENT..... 3

FIGURE 2: MICROSOFT ADAPTIVE GAME CONTROLLER..... 4

FIGURE 3: MIC-TOK'S ARCHITECTURE. NOTES ARE USED TO CREATE A PATTERN, FUNCTIONS MODIFY THE PLAY FORMAT OF A PATTERN.  
 PATTERNS ARE OUTPUTTED ..... 2

FIGURE 4: SEQUENCE OF NOTES USED FOR VARIOUS INSTRUMENTS. SYNTH PATTERNS ARE OF NUMERICAL FORMS 1-7  
 (C3-B3) WHILE KICK, SNARE, HAT, CLAP ARE REPRESENTED BY THEIR FIRST LETTER ..... 3

FIGURE 5: SYNTH PATTERN FROM FIGURE 3..... 3

FIGURE 6: CLAP PATTERN FROM FIGURE 3 ..... 3

FIGURE 7: SOME FUNCTIONS APPLIED TO A PATTERN. "TR3" SIGNIFYING TRANSPOSE BY 3, "P1" SIGNIFYING PAN TO  
 THE RIGHT WHILE "P-1" SIGNIFYING PAN TO THE LEFT ..... 4

FIGURE 8: LIST OF FUNCTIONS USED IN MIC-TOK. THE PAN FUNCTION SELECTED (BY RED HIGHLIGHTING), AND THE  
 PAN FUNCTION PARAMETERS ..... 5

FIGURE 9: SOFTWARE GUI ..... 6

FIGURE 10: SCREEN SELECTION (IN RED), I) FIRST CELL OF THE INSTRUMENT SECTION SELECTED, II) KICK IN  
 INSTRUMENT TYPE SELECTED III) A KICK INSTRUMENT SELECTED AND ADDED TO THE FIRST CELL OF THE  
 INSTRUMENT SECTION, IV) SAMPLE KICK PATTERNS CREATED IN THE PA SECTION ..... 9

FIGURE 11: NAVIGATION INVOLVED IN THE APPLICATION OF FUNCTIONS. I) FIRST CELL OF FA SECTION SELECTED II)  
 GAIN CELL OF FUNCTION LIST HIGHLIGHTED III) GAIN FUNCTION SELECTED IV) GAIN FUNCTION AND VALUE  
 ADDED TO THE FIRST CELL OF THE FA..... 10

FIGURE 12: OVERVIEW OF FORMATIVE AND SUMMATIVE EVALUATION OF OUR MUSICAL LIVE CODING ENVIRONMENT  
 ..... 14

FIGURE 13: AN INITIAL VERSION OF MIC-TOK WITH MOVING RECTANGLES IN THE PA (HIGHLIGHTED COLUMN)  
 SIGNIFYING WHAT SECTION IS BEING PLAYED AT THAT INSTANCE..... 16

FIGURE 14(I AND II): DESIGN PATTERNS WE ASKED USERS TO REPLICATE ..... 18

FIGURE 15: I) AN INITIAL VERSION OF OUR TOOL WITH MOVING RECTANGLES IN THE PA (HIGHLIGHTED COLUMN)  
 SIGNIFYING WHAT SECTION IS BEING PLAYED AT THAT INSTANCE II) IMPROVED VERSION OF OUR TOOL WITH  
 MOVING DOTS IN THE PA AND AN INCREASED FONT SIZE OF TEXTS IN CELLS ..... 21

FIGURE 16: A DISTRIBUTION OF HOW STUDY PARTICIPANTS RATED THEIR KNOWLEDGE OF MUSIC ..... 22

FIGURE 17: COMPOSITION FOR A PARTICIPANT A ..... 23

FIGURE 18: COMPOSITION FOR A PARTICIPANT B ..... 23

FIGURE 19: A BOX PLOT SHOWING THE DISTRIBUTION OF USER RESPONSES ON A SCALE OF 1-10. 1 MEANING HIGHLY  
 DISAGREE WHILE HIGHLY AGREE. FROM THE DISTRIBUTION OF USER RESPONSES, WE SEE THAT 75% OF  
 PARTICIPANTS RATED OUR TOOLS SUPPORT OF CREATIVITY ABOVE AVERAGE. .... 29

## **Table of Tables**

TABLE 1: LIST OF FUNCTIONS CURRENTLY USED IN MIC-TOK .....	4
TABLE 2: USER RESPONSES TO THE 6 DIMENSIONS QUESTIONS FROM THE CREATIVITY SUPPORT INDEX .	27

## Chapter 1. Introduction

In the digital arts, *live coding* is a performance or exploratory activity in which the text editor and output is often projected to an audience (1) so the audience can understand how code is being manipulated to generate output. It delivers a shift from the traditional programming cycle of build, compile, run, to providing instant output on code modifications. It is often an audiovisual performance practice (2) that has seen much growth over the last decade and has been applied in various contexts such as poetry (3), and music (4–7). The output media—both audio and visual—help provide instant feedback on code modifications to the audience. This invites the audience to engage conceptually with what the performer is doing. In this research, our primary focus is live coding music. Musical live coding involves the improvisation of music through live edits of source code (8). Some music live coding environments include *Gibber* (9), *Chuck* (7), *Extempore* (10), *ixi lang* (11), *Sonic Pi* (12), *Tidal Cycles* (13), among others. For brevity, throughout this paper we will refer to musical live coding as live coding.

Practically, not everyone is able to participate in live coding’s creative opportunities due to certain factors relating to its design. The community surrounding live coding provides an opportunity for performers to practice their creative improvisational skills (14). This creativity is observable in organized live coding events such as *Algoraves* (15), which are events where algorithms are used to create live electronic music. Performers at such events either generate novel music or manipulate existing dance music segments through the use of algorithms (15). But as intriguing as live coding is, it is not universally accessible. Live coding environments typically require *complex* input mechanisms. We define a complex device as consisting of dozens of input options; for example, a standard QWERTY keyboard. This requirement prevents people who use a limited set of inputs or non-traditional computing devices from live coding. Live coding is also not available to everyone because most live coding environments requires a performer to have some level of familiarity or expertise with programming languages.



In order to address these two factors and provide the experience of live coding to a wider audience, we designed Mic-Tok; a live coding environment that is programmable via a constrained set of inputs. In Mic-Tok, we made use of syntax replication across different sections to efficiently map these limited set of inputs to a vast set of musical possibilities. Similarly, we used a list of predefined functions for program manipulation. Mic-Tok was created using the Processing environment (Processing) (16); a programming language and environment built for the media arts community. Processing is based on the Java programming language (17) but has also been implemented in other programming language e.g., Python (18). This thesis uses the capitalized word “Processing” to refer to the Java implementation.

A programming session on Mic-Tok involves the creation and play of musical patterns and the modification of these patterns with a limited set of inputs. To make Mic-Tok well suited for live coding, programming manipulations and created content (melody) are looped continuously. Although Mic-Tok is primarily designed for live coding and live coding performance, because it embodies the application of arrays and functions in its design, using it for educational purposes is an area to explore in future research.

This research attempted to answer the following questions:

1. What musical livecoding environment can be created for a constrained set of inputs?
2. How do users perceive the usability and expressive potential of the musical live coding environment designed using a constrained set of inputs?

The findings of this research will help to provide approachable live coding opportunities to diverse groups such as persons with disability, people who use nonconventional forms of inputs, and people who are unfamiliar with coding. Furthermore, it will inform the design of live coding environments to include restricted and unconventional forms of inputs.

In the next chapter we review some existing and similar musical live coding environments and their approaches towards making live coding more approachable. Chapter 3 describes our software which includes our system’s design and design rationale. Chapter 4 describes our research methodology which includes how we conducted our formative and summative user study. Chapter 5 is the Results and Discussion section which describes the findings obtained from our summative study and a detailed analysis our result as well as what we plan to do for future work.

## Chapter 2. Related Work

In this section, we look at current research methods at making live coding less complex. In addition, we explore two livecoding environments that bare similarities to Mic-Tok and provide explanations as to how they differ from Mic-Tok. Finally, we discuss on a 3<sup>rd</sup> party software addition to environments that enable live coding and how it differs from Mic-Tok in terms of number of inputs

### 2.1 Live Coding Environments

A common method to make live coding less complex has been a reduction in syntax. However, this design method only tackles syntax, input still remains unchanged. Many environments have been created, from general purpose to domain specific (19–22), geared towards advancing the live coding practice. Most of these environments as listed by TOPLAP—an organization created to explore and promote live coding (23)—employ input interfaces that provide dozens of inputs for programming, for example a computer keyboard. The use of these kind of input interfaces previously helped to provide a wide range of creative possibilities for a performance. As a result of this, pioneering live coding environments employed complex syntax thereby bringing about the challenge of becoming familiar with this complex syntax to have an effective performance. In addition, managing time effectively during a performance also became a challenge (11). To address this challenge, live coding research fostered the creation of environments that reduce the amount of syntax required for a performance—for example *ixi\_lang* (11). The creation of such kinds of environments is an important milestone as it reduces the time taken while live coding to provide output to an audience, and as a result of this, drives live coding performances towards emulating an actual instrumental music performance. However, these environments only address syntax; complexity still lies in the amount of input required for a live coding performance.

On the other hand, two environments were designed around a constrained set of inputs however, when compared to Mic-Tok, these environments are not designed for direct musical composition. The environments, Betablocker and Al-jazari (24), enable using a gamepad to live code music; Betablocker has also been performed using a stylus. Betablocker is used to write machine code into processors that are fictional and 8-bits in size. These processors work in 256bytes of memory (24). Musical composition on Betablocker requires some familiarity with machine code to operate. For example, to create a musical loop, you need to select these three machine instructions (25) —push a value into a stack [PSH, 123], play it as a note [NOTE] and return it to the beginning [JMP, 0]. Al-jazari on the other hand, is used to program virtual agents moving around a 3D space (24). Musical composition on Al-jazari is an after

effect of selected agent(robot) movements. Similar to Betablocker and Al-jazari, we created an environment that is manipulated by a game controller. However, for Mic-Tok, musical composition is an intended act and not an after effect. In addition, familiarity with a programming language is not a necessity.

Furthermore, there are 3<sup>rd</sup> party software additions to environments that enable live coding and accept non-conventional forms of input. For example, Wagner et al., created a voice driven Java application software called Myna (26) that works in parallel with the Scratch programming environment (27) —an environment which wasn't primarily designed for live coding but has been used for music live coding scenarios while teaching computational thinking (28). Myna provides input through a voice interface, but this doesn't mean input is limited. It makes use of a collection of pre-defined words to serve as input to various activities within Scratch which is not restricted in its number of inputs compared to Mic-Tok software. For our project, we use only six inputs for a performance.

## Chapter 3. MIC-TOK ENVIRONMENT

In this chapter, we attempt to answer our first research question which is

1. What musical livecoding environment can be created for a constrained set of inputs?

We answer this question by describing Mic-Tok's live coding environment below.

### 3.1 IMPLEMENTATION

Mic-Tok (**Figure 1**) was created using the Processing environment. We chose this environment because it is specifically designed for visual, interactive, and artistic displays (29) which are attributes embodied by our system. It also makes use of Processing libraries including the in-built Processing Sound library and other open-source libraries for example for sound, the Minim library by Damien Di Fede and Anderson Mills (30), for widgets and graphics, the controlP5 library by Andreas Shlegel (31).

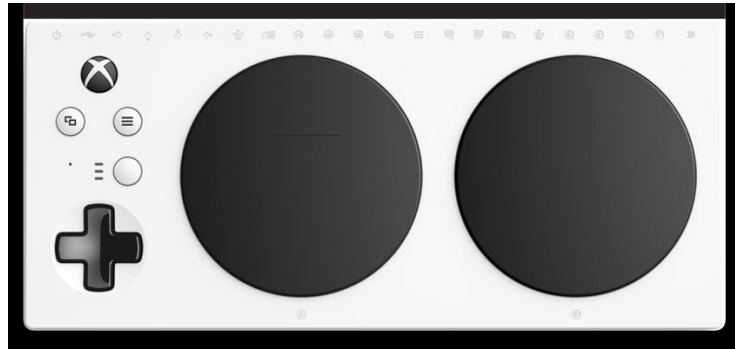


Figure 1: Mic-Tok environment

To provide our limited set of inputs, we originally used the Adaptive Game Controller<sup>1</sup> [55] shown in **Figure 2** but had to change our evaluation strategy to support user studies because we did not expect all

<sup>1</sup> A controller that has six buttons (four directional buttons and two circular buttons), designed by Microsoft for Windows for people with disabilities [22]

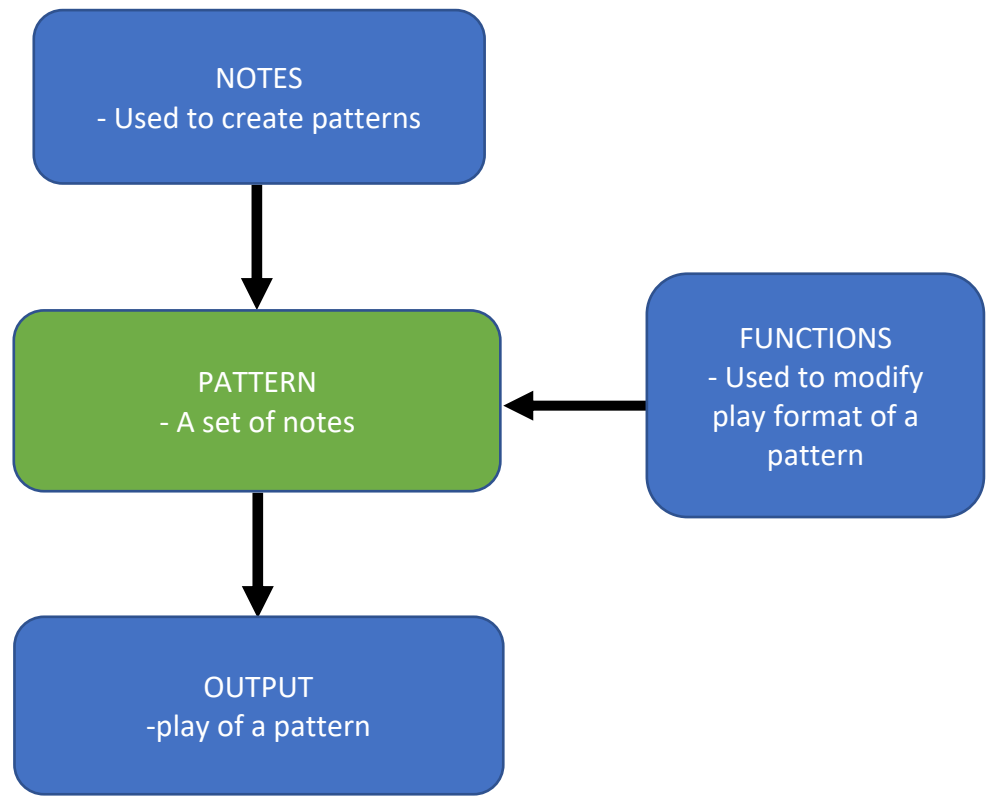
our study participants to have access to an Adaptive controller. Since we could not use the Adaptive Game Controller, to provide input, we used six inputs from a computer keyboard! Reference source not found.. These buttons include ENTER – to select an option, TAB —to deselect an option, UP —to move up the grid, DOWN —to move down the grid, LEFT —to move left of the grid and RIGHT —to move right of the grid.



*Figure 2: Microsoft Adaptive Game Controller*

## 3.2 Mic-Tok's Architecture

Mic-Tok architecture is divided into three major structures namely, Notes, Patterns and Functions. Notes are used to create Patterns. Patterns are used to create musical composition while Functions are used to manipulate these created patterns or. **Figure 3** gives a pictorial view of the structure of Mic-Tok. We describe these structures in detail below:



*Figure 3: Mic-Tok's Architecture. Notes are used to create a Pattern, Functions modify the play format of a pattern. Patterns are outputted*

### 3.2.1 Notes

Notes are used to create patterns for the five musical instruments we use which include, a KICK, SYNTH, SNARE, CLAP and HAT. Notes for a Synth are in the form of natural numbers ranging from 1 to 7 to represent the third octave notes, C3 - B3 respectively of a piano keyboard while notes for the rest of the instruments consist of cell selections of which a selected cell will be replaced with the first letter of that instrument. For example for the KICK a letter ("K"), a SNARE ("S"), a HAT ("H"), and a CLAP ("C"), as shown in **Figure 4**.

INSTRUMENT	PATTERN ARRAY							
SYNTH	1	2		3		2		
KICK	K	K	K		K			
SNARE		S	S	S				
HAT	H		H		H			
CLAP	C	C	C		C	C		

Figure 4: Sequence of notes used for various instruments. Synth patterns are of numerical forms 1-7 (C3-B3) while KICK, SNARE, HAT, CLAP are represented by their first letter

### 3.2.2 Patterns:

Patterns created describe the order in which notes of an instrument would be played. Patterns have been used in live coding environments for example (9,11). In Mic-Tok, these notes in a pattern are played in a timely manner starting from left to right. From example, the pattern for the Synth in **Figure 5 & Figure 6** starting from left plays “1” (represented by the C3 note of the piano keyboard) played first followed immediately by “2” (D3), then a skip, then “3” (E3), a skip again, then “2” (D3), two skips and play returns to the start and plays in the same order. Notes for other instruments are played in similar fashion. To create a pattern, a user must first choose one of these instruments and afterwards create a pattern of their choice. Pattern creation is by the selection of cells (we will describe this in detail in sections below)

SYNTH	1	2		3		2		
-------	---	---	--	---	--	---	--	--

Figure 5: Synth pattern from figure 3

CLAP	C	C	C		C	C		
------	---	---	---	--	---	---	--	--

Figure 6: Clap pattern from figure 3

### 3.2.3 Functions:

Functions are used to modify the play of a created pattern. Functions have been used in live coding environments to induce dynamics to a created pattern for example as seen in (9,11). In Mic-Tok, we use functions to provide a high-level system so users can easily modify the play of a patterns. Each function selected is applied to a created pattern’s FA (**Figure 7**) and modifies play of that pattern.

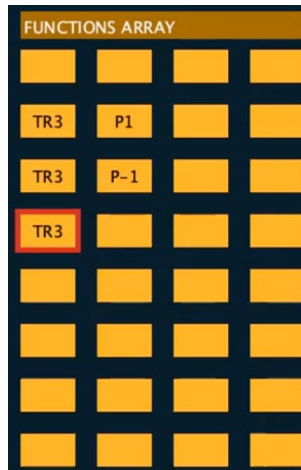


Figure 7: Some functions applied to a pattern. "TR3" signifying Transpose by 3, "P1" signifying pan to the right while "P-1" signifying pan to the left

We make use of six functions:

Table 1: List of functions currently used in Mic-Tok

Function name	Action
Transpose	to transpose a pattern a full octave up to a specified integer value between the range of -4 to +4, 0 being the default middle C3 octave
Pan	to set the pan of a pattern between the range of -1 to +1, 0 being the default octave
Wave	to choose the type of wave (sine, square or saw) for an instrument/pattern. These include a SQUARE, SAW and SINE (default) wave
Effects	to set a Bitcrush effect on an instrument/pattern
Gain	to set the gain of an instrument/pattern. from DEFAULT to HIGH
Delete	to delete a function or pattern

For each of these functions, we also provided a list of selectable options to serve as arguments. They are represented by radio buttons and are contextual to a function. They are also only visible after selecting a function, and their aim is for users to pass arguments to a selected function. Choosing an option manipulates the play of the pattern it is applied to. We provided a list of options as a constrain measure because we needed to be able to provide variability despite our constrained input. An example can be seen in **Figure 8**, Selecting the Pan function will provide three hardcoded options to choose from namely,



LEFT (-1), RIGHT (1), DEFAULT (0). After selecting an option, it is applied to the FA to modify audio to shift audio to the right Snare.

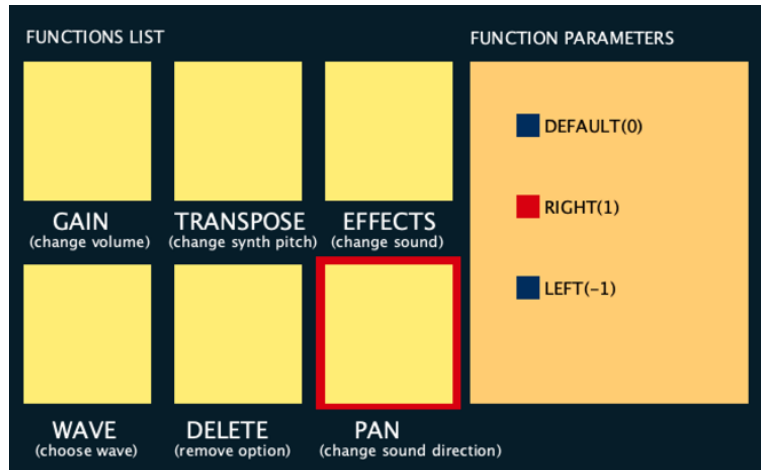


Figure 8: List of functions used in Mic-Tok. The Pan function selected (by red highlighting), and the Pan function parameters

### 3.3 Performance environment

#### 3.3.1 Software GUI

Mic-Tok GUI, as shown in **Figure 9** is divided into three sections: Pattern, Functions Array (FA) and Menu. These sections are described in detail below.

**1. Pattern Section:** The Pattern section is located at the upper left section of our screen. It consists of two subsections namely, an instrument section and a pattern array (PA) section as shown in **Figure 9**. The instrument section is used to store and signify what instrument's pattern will be created while the PA section is used to create patterns by cell selections. The PA consists of 8 rows (arrays) of size 8 each to represent a musical measure and is used to store patterns from one of the musical instruments

**2. Functions Array (FA) Section:** The FA section is located at the upper right section of our screen. It consists of 8 rows (arrays) and 4 columns as shown in **Figure 9** and is used to store selected functions from the Menu section (explained below) for each created pattern in the Pattern section. These functions modify the audio of a created pattern. Selected functions for each array in the Pattern section are displayed in the FA.

**3. Menu:** The Menu section is everything below the Pattern and FA. It is used to select programming options —functions and instrument type. Selecting an instrument type will place it in the instrument part of the Pattern section while selecting a function will open a dialog box in the selected function area.

In addition, highlighting is used across Mic-Tok so users can identify what part of the screen they are currently working on. For example, the red highlighting as shown for the Pan function in **Figure 8**.

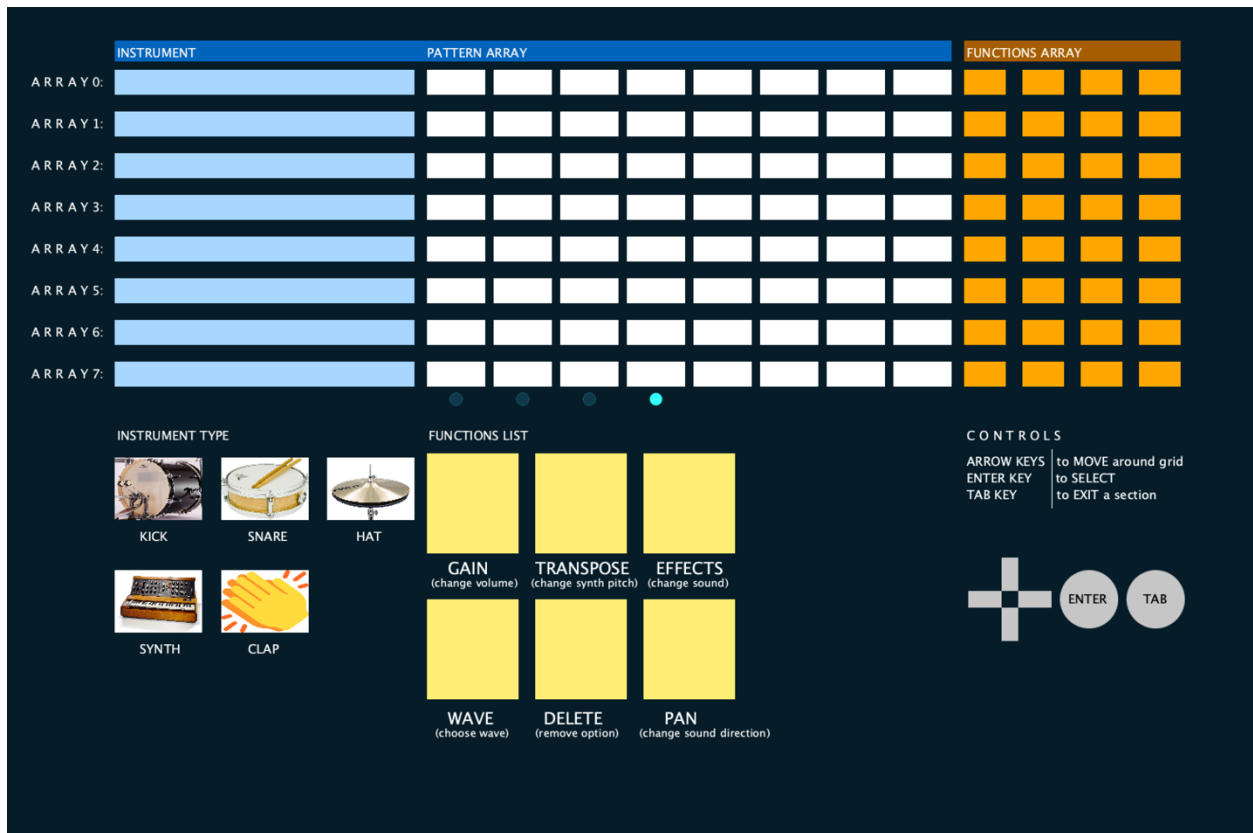


Figure 9: Software GUI

### 3.4 Features

Mic-Tok has five features: syntax, tempo, rhythm and timing, modulation and sequencing. However, because our goal is to reduce complexity in livecoding, we designed some of these features around constraints. We describe these features in detail below.

### **3.4.1 A one-to-one mapping of six words to six buttons for operability**

Mic-Tok uses six words for full operability. A reason for only six words was to provide a one-to-one mapping of inputs to actions and as a consequence simplify its use. They are, select, deselect, move up, move down, move right and move left, and they are mapped to the following keyboard buttons respectively ENTER, TAB, Direction buttons (up, down, right and left).

Each of these codes are applicable regardless of what section or context the user is, in the environment. We designed in such manner because we needed to find a means to provide multiple possibilities despite our input constraints. For example, the ENTER key which is mapped to the code “select” can be used for selection across every section of the platform to select cells, functions, function arguments, and whole sections. The same applies to the direction buttons which are used to navigate in any section of our screen. Designing in such a consistent manner, provides more musical opportunities with a limited input. In addition, it also helps to simplify Mic-Tok by redirecting the user’s mind from thinking about syntax to more emphasis on creating content.

### **3.4.2 Tempo**

Mic-Tok uses a fixed tempo of 126bpm. One of the constrain measures we implemented in Mic-Tok was restricting tempo manipulations. To provide this fixed tempo, we used the timely execution of the draw function (29) in Processing. We set the timely execution of the draw function to 4fps (frames per second).

### **3.4.3 Rhythm and Timing**

There are 8 cell blocks in the PA to represent a measure. Timing for notes are fixed to quarter notes. A user can only manipulate when they want quarter notes to be played by the way in which they make their cell selections. For example, to play a note for every quarter, a user selects all the cells. This produces a sound for every beat. To create a timing of 1/4, a user selects the first and fifth cell in a pattern row. Limiting time manipulations to this form is another constrain measure we implemented.

### **3.4.4 Modulations**

Modulations are managed by the Minim Library available in Processing. In Mic-Tok, parameters of functions can be modulated. Functions are located in the Function List section as shown in **Figure 8**. To

modulate a parameter, a user first navigates to the function list section and selects a function. Selecting a function opens up a dialog box where these parameters can be selected (**Figure 8**).

### **3.4.5 Sequencing**

Patterns are played in a sequential order from left to right. To create the continuous timely execution of pattern sequences we used the timely execution of the draw function (29) Processing provides. We set the timely execution to 4fps (frames per second). When creating a pattern, as soon as a cell is selected, and the highlighted columns (playing from Left to Right) reaches the column where the selected cell is, the sound of the instrument in its instrument section is triggered.

## **3.5 A programming session with Mic-Tok**

A programming session with Mic-Tok involves a user creating a pattern in the Pattern section and applying functions to modify it. All selections are made by pressing the ENTER key and all deselections are made by pressing the TAB key. To move around, the D-pad buttons are used.

### **3.5.1 Creating a Pattern**

At the very start of Mic-Tok, the Pattern section continuously blinks. This is an indication of the current location of the cursor. To create a pattern, a user must first select an instrument. To do this, the user presses the ENTER key which highlights the first cell of the instrument section as shown in **Figure 10 (i)**. A further press of the ENTER key will direct the user to the instrument type section as shown in **Figure 10 (ii)**. Further pressing of the ENTER selects an instrument and places selected instrument in the previously chosen cell of the instrument section as shown in **Figure 10 (iii)**. After selecting an instrument, the user is directed back to the instrument section where they can begin creating by pressing the RIGHT key to move to the PA section and selecting cells of their choice. For example, as seen in **Figure 10 (iv)**



Figure 10: Screen Selection (in red), i) first cell of the instrument section selected, ii) Kick in Instrument type selected iii) A Kick instrument selected and added to the first cell of the instrument section, iv) sample kick patterns created in the PA section

### 3.5.2 Adding a function

To add a function, after creating a pattern a user presses the TAB key to shift the cursor back to the Pattern section header. Then they press the RIGHT key to shift the cursor to the FA section. Thereafter, they press the ENTER key to shift the cursor to the first cell (highlighted in red) in the FA as shown in **Figure 11 (i)**. Further pressing of the ENTER key shifts navigation to the Functions area as shown by the highlighted GAIN function in **Figure 11 (ii)**. After navigating to a choice function, Pressing the ENTER key again opens up the “SELECTED FUNCTION” section as shown in **Figure 11 (iii)**. Finally, pressing the ENTER key places the function in the highlighted cell in the FA as shown in **Figure 11 (iv)**.

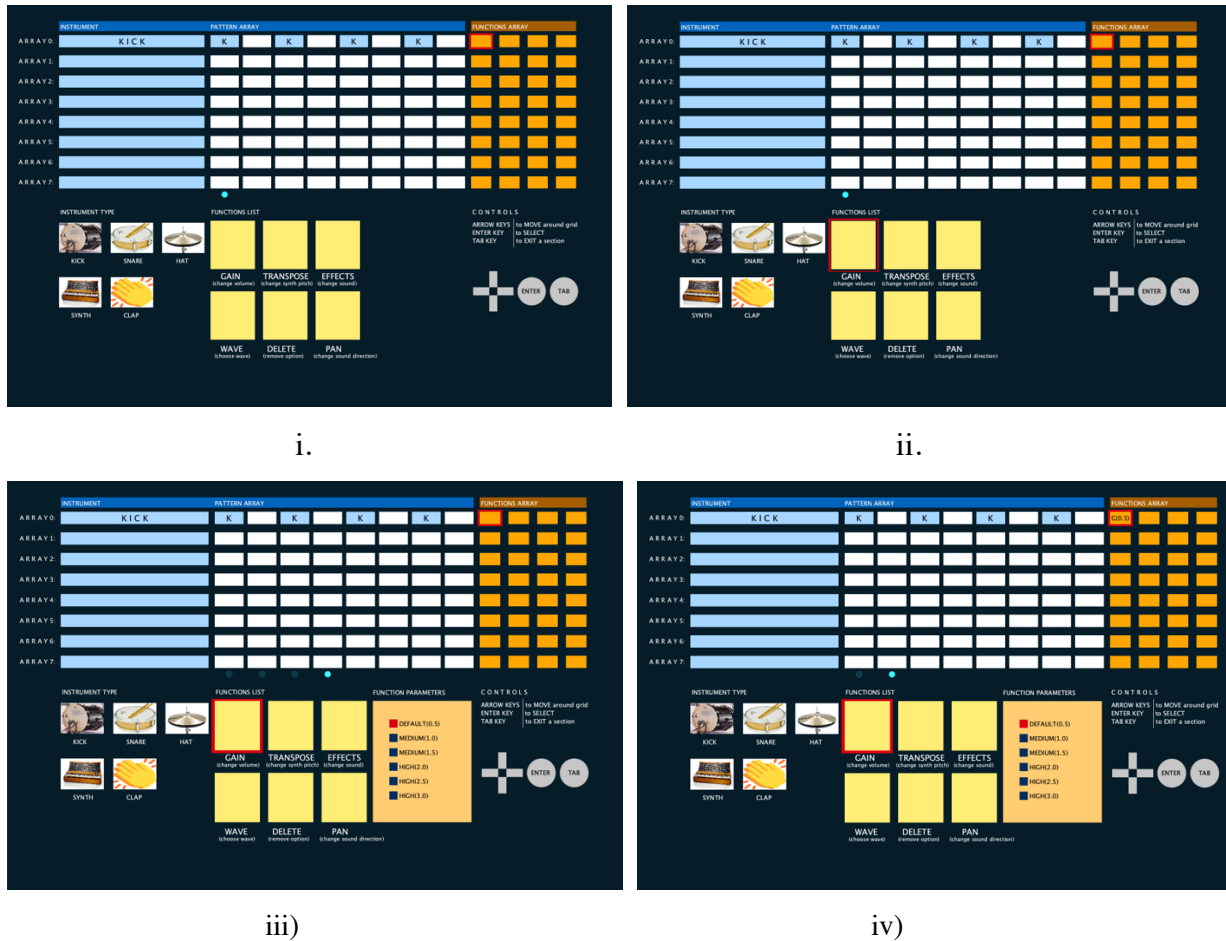


Figure 11: Navigation involved in the application of functions. i) First cell of FA section selected ii) Gain cell of Function List highlighted iii) Gain function selected iv) Gain function and value added to the first cell of the FA

### 3.6 An analysis of the Mic-Tok environment using the Cognitive Dimensions of Notation Framework

In this section, we provide an analysis of Mic-Tok using the Cognitive Dimensions of Notation Framework (CDN). We use the CDN for the following reasons: it is a framework used to evaluate the usability of notational systems or programming environments for example (32,33). We especially use it because it has been used to analyze musical live coding systems for example as seen in (6,34). Also, the CDN has been created to aid creators of notational systems assess the effectiveness of their design on users (35) by informing them on some of the design trades-offs they can make when creating their tool. CDN comprises of 14 design principles/dimensions, however, for this research we used 10 of these

principles because they can be applied to Mic-Tok. We describe these dimensions and what design decisions we took in our design based on what these principles imply.

1. **Hard mental operations:** This refers how mentally taxing notations in an environment are (36).

**Design decision:** We tried to simplify our notation so users will not have a hard time while using the system (36). Some measures we took include the following: (1) restricting the number of inputs to only six buttons. (2) limiting the number of possible activities to only two namely, creating a pattern and applying a function.

2. **Progressive Evaluation:** This refers to the ability to view and implement incomplete code. This helps users understand the effects their incremental decisions have on output.

**Design decision:** Similar to live coding environments, in Mic-Tok, evaluation is progressive. This is evident in the instant feedback on code modifications. Since we created a live coding environment, this design principle had to be a part and parcel of our environment and it is discernable in the form of audio output. For Mic-Tok, users can create and modify patterns while it is playing, and in a timely manner hear the effect of these modifications while audio is playing.

3. **Error-Proneness:** This can be described as to what extent the notations used in an environment influence the likelihood of a user making an error (36).

**Design decision:** One way in which we minimized the possibility for error is by making Mic-Tok a fully visual programming environment (37). Visual programming languages are less prone to errors when compared to textual languages where a user has to type (36). However, in the design of Mic-Tok, there are certain areas where the notation used might cause an error. For example, the cells in the PA are small and closely packed to each other and as such, errors might occur in the form of selecting wrong cells, sections, notes or functions. In addition, to further reduce the possibility of error, we made use of a blinking cursor at every selection so users can easily tell what section of the system they are and to guide them carefully

4. **Viscosity:** This can be described as the resistance to change. How much does it take to make a change (36).

**Design decision:** Not much thought was put into designing Mic-Tok to address viscosity. However, one of the methods in which we tried to reduce viscosity is by reducing the amount of button presses

required to make a change in a section in Mic-Tok. For example, when in the PA, if a user wants to make a change to a created pattern for any of the instruments besides a synth instrument, all they had to do was go to the location where that highlighted cell is and select it again and it is deselected. There is not much resistance to change. On the other hand, movements that could bring about higher viscosity are trying to make a change between sections of the interface. For example, navigating from the function parameter area to the PA (and vice versa) to make a change.

5. **Diffuseness/Terseness:** This refers to how much surface area notations require to reach a result (36).

**Design decision:** In Mic-Tok, notations are fixed. A reason for this is so we don't create a cognitively tasking system for the user (36). Most of the notations we used were fixed and so there were no extra space requirements to show anything else. The only section where there is sometimes a need for an extra space is in the "FUNCTION PARAMETERS" or "SELECT NOTE" options where after selecting a function or Synth instrument (respectively), opens up a dialog box. However, these sections are well placed and do not affect other sections of the screen.

6. **Consistency:** This refers to how much can be inferred by a user after learning some of an environment.

**Design decision:** There are two main activities in our system. One is to create a pattern, and another is to apply a function. The process involved in creating a pattern is the same regardless of what type of pattern a user wants to create and for play modifications, the same regardless of what modifications a user desires to make. In addition, button manipulations are similar across every section of the system. With this design method, we created a consistent our design

7. **Visibility:** This refers to the ability to view notations used in a system easily.

**Design decision:** We created a mostly visible system to reduce mental costs (36). For Mic-Tok, every notation is visible with the exception of two which are only visible after selecting them. They are, functions and synth notes and they. As a result, Mic-Tok is a mostly visible.

8. **Role-expressiveness:** This refers to how easy notations can be understood.

**Design decision:** We tried to create an environment where notations can be easily understood. One of the areas where we suspected its notation might be unclear is the functions section. However, we tried



to enhance its understandability by the use of secondary notations in the form of a brief description for each function (36). Similarly, despite these descriptions, a user unfamiliar to some music terminologies might still be unclear as to what these functions mean to apply them.

9. **Premature Commitment:** This refers to constraints imposed on the manner to which notations of a system can be used (36).

**Design decision:** For premature commitment, one of the constraints we implemented in Mic-Tok is the restricting the creation of patterns to only after selecting an instrument. A reason for this was because the creation of a pattern was different between a Synth and the rest of the instruments we used. We wanted the user to first specify what instrument they wanted to use so the type of pattern can be known by the system before hand

10. **Closeness to mapping:** this refers to how close notations used are to their results

**Design decision:** In terms of closeness to mapping, some of the notations we used bore closeness to their result. For example, selecting a KICK instrument produced a drum kick sound or selecting a Hat produced a Hat sound. Another closeness to mapping were some of the input buttons we used. For example, the direction buttons will be known as what is needed to move around, the enter key will be known to select

## Chapter 4. Research Methodology

This research involved two sets of usability studies. First, a formative evaluation on an initial version of Mic-Tok. The goal of this study was to find out usability problems with our initial version and optimize it. Our second study was a summative evaluation on our modified software to find out user perception of our tool. We describe these studies in detail below. **Figure 12** provides a simplified pictorial view of our methodology.

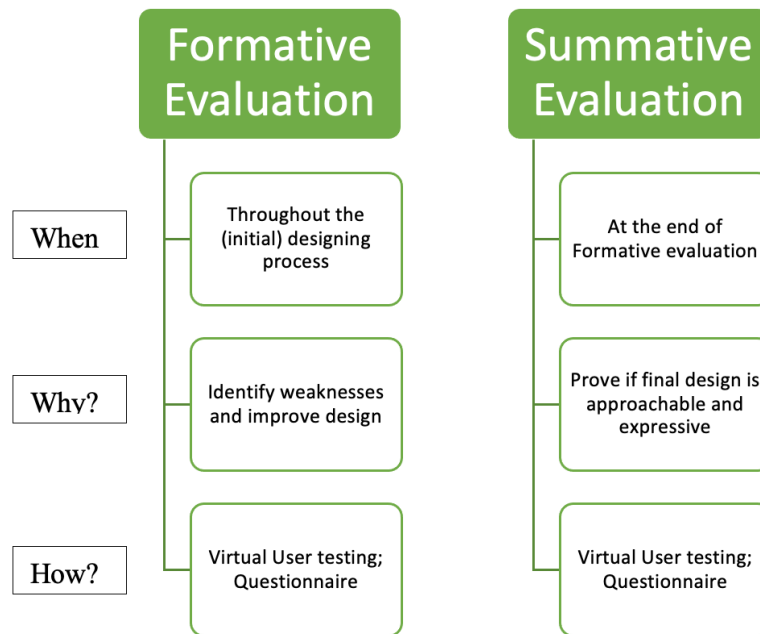


Figure 12: Overview of Formative and Summative Evaluation of our Musical Live coding Environment

### 4.1 Formative Evaluation

Formative evaluations help to improve a design prototype towards a higher quality and sharpen the underlying temporary design principles towards a decisive set of design principles (38). To optimize our tool and finalize design methods we used in Mic-Tok, we iteratively conducted formative evaluations on an initial version of Mic-Tok (**Figure 13**). Results and observations from this study are presented in our results section in Chapter 5 however, study procedures are discussed in the subsections below.

#### 4.1.1 Setting and Audience

We experimented online via Zoom with 10 WPI students who were above 18 years of age. We used the convenience sample method (39) in recruiting participants for the study. The study was for a duration of two weeks and we modified Mic-Tok over the course of the study. Our study was conducted online due

to restrictions posed by the COVID-19 pandemic. All recordings were through Zoom video recorder. We uploaded our software to google drive and asked users to download the software via a provided link. This study was for 30 minutes and divided into four stages.

#### **4.1.2 Formative Study Design/Procedure**

Our formative evaluation was divided into 4 stages. We describe these stages below.

##### ***Stage 1 – Demographics + Tutorial***

To begin our study, users were asked to fill a presurvey where they provided demographic information which included, their age and knowledge of music. We asked for knowledge of music to find out how that affected our results. After completing the presurvey, participants were given a tutorial about how to operate our system using keyboard controls. This was for 5-minutes

##### ***Stage 2 – Activity***

In this stage, participants performed basic activities that we provided to enable them them familiar with operating the tool. This took 5 minutes and included:

1. Creating a pattern with an instrument
2. Applying functions to a created pattern

##### ***Stage 3 – Improvisation***

In this stage, we asked users to improvise i.e. to create patterns and apply functions of their choice for 10-minutes while thinking out loud. The stage lasted for 10 minutes.

##### ***Stage 4 – Exit questions***

In this stage, we asked users again 10 exit questions aimed at capturing how they felt about our tool. These questions were on user experience question. For more details on these questions, (see **Appendix A**).

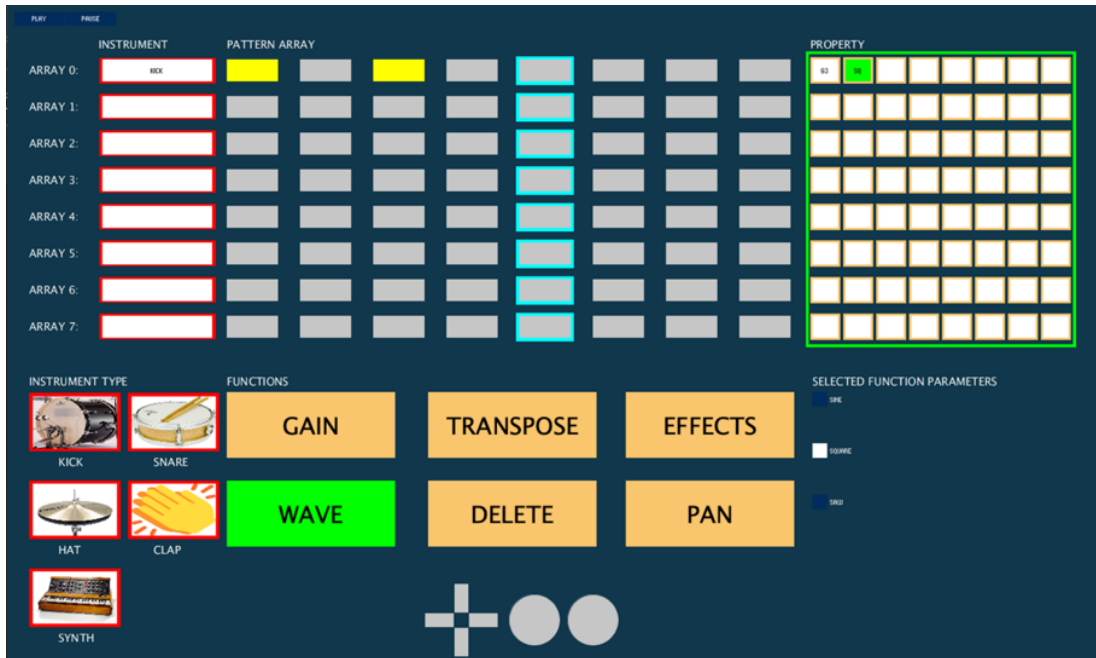


Figure 13: an initial version of Mic-Tok with Moving rectangles in the PA (Highlighted column) signifying what section is being played at that instance

## 4.2 Summative Evaluation

Summative evaluations help to verify the actual effectiveness to establish whether goals of a study have been achieved (38). In this evaluation study, our goal was to determine how users perceived our tool with regards to its approachability and expressiveness. Results and observations from this study are presented in our results section in Chapter 5 however, study procedures are discussed in the following subsections.

### 4.2.1 Setting and Audience

We conducted a summative evaluation of a finalized version of our software. To recruit participants for the study, we again adopted the convenience sample method with 14 WPI students. These students were different from those who participated in our formative study. The study lasted for a week and was conducted virtually via zoom. We ran our software on our system, shared our screen via zoom and asked users to request remote access to our system to operate our software while we recorded their play time. Each study lasted for 35 minutes and was in four stages.

### 4.2.2 Summative Study Design/Procedure

Our summative study was divided into four stages. A detailed description of what these stages entailed is described below.

### ***Stage 1 – Demography + Tutorial***

The study, commenced with participants completing a pre-survey questionnaire where they provided demographic information including, their age and knowledge about music. We asked the participants how they rated their knowledge of music for us to have an idea about how that would influence survey results. After completing the presurvey questionnaire, participants were given a tutorial on how to operate our system using keyboard controls.

### ***Stage 2 – Activity + Improvisation***

In this stage, participants performed basic activities that we provided to enable them to be familiar with operating the tool. This took 5 minutes and included:

1. Creating a pattern with an instrument
2. Applying functions to a created pattern

In turn, we asked users to improvise i.e. to create patterns and apply properties of their choice by themselves for 10-minutes while thinking out loud.

### ***Stage 3 – Replicating pre-designs***

In this stage, we presented two designs for our participants to replicate using our tool (see **Figure 14**). These design patterns were very similar to each other and our goals were to evaluate the time taken to complete a task and to find out if users made any improvements within the time taken to complete a second task. These activities cut across the major functionalities that our system embodies, namely adding an instrument, creating a pattern, and adding properties to created patterns. The stage lasted for 10 minutes.

### ***Stage 4 – Exit questions***

For stage 4, we asked users 10 exit questions to capture their impressions about our tool. These questions ranged from user experience questions to questions on creativity. Creativity questions were adopted from the Creative Support Index method (CSI) (40). We discuss more on the CSI in subsequent chapter, however, for more details on these questions, (see **Appendix A**).



## Chapter 5. Results

In this chapter, we attempt to answer our second research question which is

2. How do users perceive Mic-Tok's usability and expressive potential?

We answer this question using observations and user responses from our summative study. To answer on user perception on the usability, we use an established set of questions derived from the CDN which evaluates how users perceive the usability a system (41), while to answer the user perception on expressivity, we use a part of Creativity Support Index (CSI) (40). But first, to begin, we provide a summary of observations made from our formative study on an initial version of Mic-Tok and modifications we made to produce our final version.

### 5.1 Formative study results and modifications made

After finalizing our formative evaluation and analyzing our results, a range of issues were identified from our initial design. Most of these issues centered around graphics and musical terminologies used in our software. We provide a summary of problems identified and modifications made for each of these problems.

**Issue:** Navigation

**Observation:** Users had challenges in understanding the workflow and in identifying where they were at a given time. For example, one user said, *“the fact that I did not know where I was on the interface. for example, after exiting with the tab key, I often forgot on which section of the interface I was. And within each section as well, it was hard to know on which line/row I was until I move to right or left from the current position.”*

**Modification made:** We made use of a red blinking cursor to always notify users the section they are currently at and to help the user get acquainted with the workflow.

**Issue:** Color choices

**Observation:** Users expressed displeasure with the color combinations used. The colors were either not contrasting enough with each other for a user to understand where they were, what was happening, or they were too bright. Users also, had issues with the fonts sizes of texts written in cells.

**Modification made:** We made use of highly contrasting colors to enable users easily identify the various sections and what they are working on. For the fonts we made sure we increased the font sizes for texts in the cells (**Figure 15.ii**)

**Issue:** Moving rectangles: a way of notifying the user of what section is currently playing (**Figure 15.i**)

**Observation:** To show a user what column is currently playing at an instance in the Pattern Array, we made use of moving rectangles. We found out that this became distracting to our users while they used the tool. (**Figure 15.i**)

**Modification made:** To show a user what column is currently playing at an instance in the Pattern Array, we made use of a blinking circle just below the Pattern Array section (blue dot in the 8<sup>th</sup> column of **Figure 15.ii**)

**Issue:** Terminologies

**Observation:** Some participants did not understand some of the musical terminologies for functions and function parameters used in our system. For example, a user said, *“the tool needs knowledge of music, so maybe learning about music terminologies will help”*.

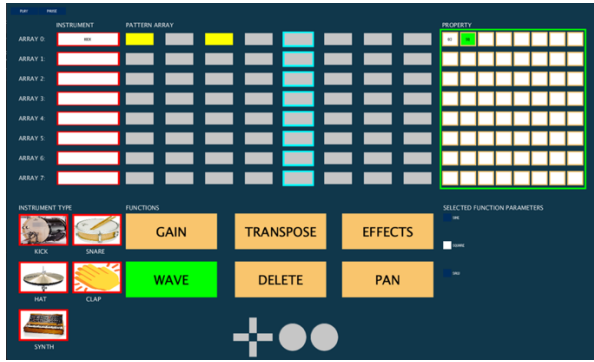
**Modification made:** We provided explanations for functions and utilized descriptive names for function arguments. For example, for Pan which we described as “change sound direction”, we utilized function parameters like Default (0), LEFT (-1) and RIGHT (1) as Pan arguments.

**Issue:** Section headers and number of cells used

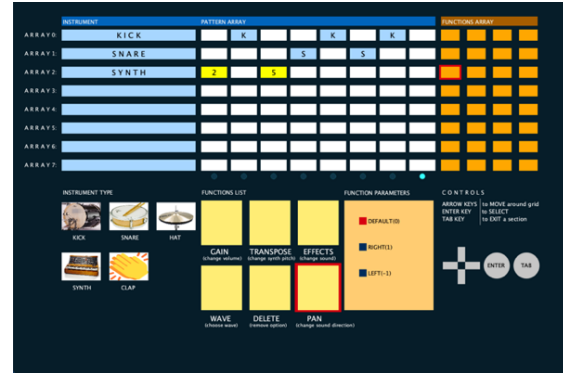
**Observation:** A user while thinking out loud described the 8 by 8 cell blocks in the Property section of **Figure 15(i)** as *“too many and can be distracting”*. Another confusion a user had was understanding the relationship between sections of the interface,

**Modification made:** We modified the 8 by 8 cell block in the property section to a 4 by 8 cell (**Figure 15(ii)**).





i.



ii.

Figure 15: i) An initial version of our tool with moving rectangles in the PA (Highlighted column) signifying what section is being played at that instance ii) Improved version of our tool with moving dots in the PA and an increased font size of texts in cells

## 5.2 Summative study results

In this section, we begin answering our second research question on user perception of Mic-Tok in regard to its complexity and expressivity. To begin, we provide a demographics of how our study participants rated their knowledge of music, next we describe what an interaction with Mic-Tok entailed from three use cases and finally, we provide responses in regard to complexity using the Cognitive Dimensions of Notations and in regard to expressiveness using the Creative Support Index.

### 5.2.1 Demographics

We asked users to rate their knowledge of music and out of 14 participants, 64.29% (n = 9) identified themselves as beginners, 7.14% (1) identified as an expert, 7.14% (1) identified as intermediate, and 21.43% (3) identified as no knowledge.

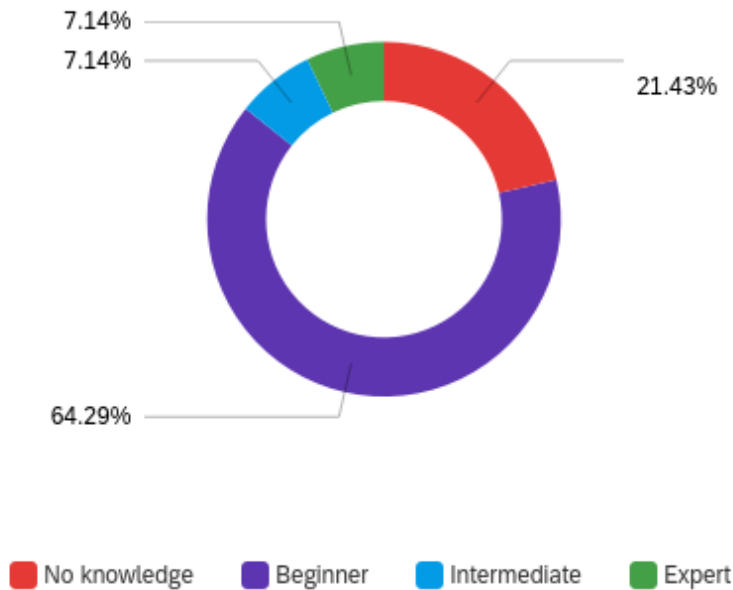


Figure 16: A distribution of how study participants rated their knowledge of music

### 5.2.2 Cases Studies from Summative evaluation

To illustrate how users use Mic-Tok to compose music, here we describe case studies from two participants during their improvisation session. We chose these two particular participants because they self-identified differently in musical expertise. Participant A self-identified as an expert while Participant B self-identified as Beginner. We describe their interactions during the improvisational session in detail below.

#### Participant A

Participant A rated their knowledge of music as “expert”. Participant A constructed patterns for all the musical instruments. A’s composition method consisted of selecting an instrument and creating its pattern at the same time. While composing, A asked questions like, “*what if I wanted to play eight notes instead of quarter notes*”. However, despite rating themselves as expert, Participant A still wasn’t familiar with one of the functions —the WAVE function. Another observation we made was Participant A seemed like they had a goal they wanted to achieve with their composition. After creating patterns for the first four instruments namely, CLAP, SNARE, SYNTH, and HAT, and feeling excited about their creation, A got

discouraged after they could not successfully create a nice tune using the KICK. **Figure 17** shows what participant A created.

INSTRUMENT	PATTERN ARRAY								FUNCTIONS ARRAY				
CLAP	C			C									
SNARE		S					S						
SYNTH						5			TR2	SIN			
HAT								H					
KICK	K		K		K		K		G(2)				

Figure 17: Composition for a Participant A

### Participant B

Participant B rated their knowledge of music as “beginner”. B enjoyed the sound but was not so keen to try out all the functions. B constructed patterns for the KICK, SYNTH and CLAP only. B’s composition method was similar to participant A which included selecting an instrument and creating a pattern for that instrument at the same time. As a response to our exit question, B made comments like, “*As someone seeing this program for the first time, I struggled a bit with figuring out which functions to place from letters alone (e.g. that SQ was for a square wave, as I couldn’t tell it was a function in the wave category until I was told so)*”. Another observation showed that unlike participant A, there was no intention/goal to Participant B’s composition. **Figure 18** shows what participant B created.

INSTRUMENT	PATTERN ARRAY								FUNCTIONS ARRAY				
KICK	K		K		K								
SYNTH	5		4			6			TR-1	B_ON	SIN		
CLAP						C							

Figure 18: Composition for a Participant B

### 5.2.3 Summary of Observations from Case Studies

An interesting difference we can observe from participant A and B is the fact that because A has experience with music, they had a goal to their composition. On the contrary B's lack of musical background affected how they approached their composition. A was more enthusiastic to try out options while B was more reluctant thereby indicating that interaction with Mic-Tok varies according to level of musical expertise. Also, in the design of similar environments in future, there needs to be a medium where users can save and share their constructed designs to inspire participants like B who said, “*the most challenging part was thinking of my own patterns because there was a lot of freedom with what to choose*”.

### 5.2.4 User perceptions on usability of Mic-Tok

#### *The CDN questionnaire*

Although the Cognitive Dimensions of Notation (CDN) was originally and is predominantly used as a guide to designers to evaluate the usability of their system (35), however it has also been used to evaluate user perception on the usability of a system via an established questionnaire that touch on the various CD (41,42). Though we did not directly apply these questions in our summative study, but, because answers and observations obtained from our user study can answer these CD questions, we analyze our user responses on Mic-Tok’s usability using some questions from the CD questionnaire.

For our analysis we use questions from ten dimensions of the CD questionnaire. We use ten because these dimensions are applicable to Mic-Tok. For each of these dimensions, we provide responses and observations of user sentiment on notations that fall into that dimension. We describe these sentiments as “Positive” meaning users had no problem with notations used in that dimension while “Negative” meaning users had a problem with notations used in that dimension. We begin by dimensions which users had a negative sentiment towards —Role-expressiveness and Viscosity—, and then we describe the rest of the dimensions.

**Role-expressiveness** (Negative): when reading a notation can you tell what it means?(41)

Out of all the notations used in our tool, 4 participants found understanding the function notations used challenging. We asked users what activities they found challenging, a user said, “*Adding a function. Perhaps because I am unfamiliar with naming and the potential effects on the sound*”. Although we tried to address this after our formative evaluation, by providing brief descriptions on what each of these functions did, observations indicate users still found it hard to understand. A reason for this might have

been because functions were not explained well enough. Another reason for this might have been because out of our 14 participants, most (9) of our study participant identified as beginners when it came to their knowledge of music

**Viscosity** (Negative): when you need to make changes to something previously created, how easy is it to make the change? (41).

To describe viscosity, we describe it in the context of an important observation we made —users needing a mouse. Some users complained about their inability to use a mouse (8 participants). One user said “*I don't like that I can't use the mouse! The interface seems like it'd work really well if you could drag and drop parts*”. In Mic-Tok, after applying a function to a created pattern, if a user wants to navigate back to the PA to modify an existing pattern, they had to press the TAB button to go back in equivalent number of times as they pressed the enter button to go forward. For example, if they pressed the enter key to go to the function parameters section, 4 times i.e., select the Function Array Header → Function Array → Function List → Function Parameters, to go back to the Pattern section, they had to press the Tab button backwards in the form, deselecting the Function Parameters → Function List → Function Array → Function Array Header.

**Hard mental operations** (Positive): What kind of notations require the most mental requirement with this notation (41)?

There were no observed notations that caused our users hard metal requirements, rather, users described the tool as intuitive (8 participants), nice (4 participants), fun (4 participants), nothing challenging (4 participants). All participants completed provided tasks.

**Error-Proneness** (Positive): Do some kind of mistakes seem particularly common? (41)

Out of our 14 study participants, 2 participants made errors while creating a pattern in the PA. A reason for this can be attributed to how closely packed cell notations are in the PA as compared other sections. However, for the rest of the 12 participants, there were no other observed errors

**Consistency** (Positive): are there areas that ought to be similar, but the notations used make them different? (41)

Out of our 14 participants, there were no observed challenge with consistency of notations used in Mic-Tok. Creating a pattern is similar across all rows and adding a function is similar across all systems. In addition, button manipulation is similar across every section of the system. One user commented “*Its ease of use; a "back" button and "select" button are a good way non-mouse to navigate the grid, and the*

*flashing is a good indicator of what is selected. I enjoyed playing with sounds and seeing what little melodies I could make.”*

**Closeness to mapping** (Positive): which parts seem to be a particularly strange way of doing or describing something? (41)

There were no observed user comments on how notations used in the system were not closely related to their results. Notations worked as they were supposed to. For example, selecting a KICK instrument produced a drum kick sound or selecting a Hat produced a Hat sound. All participants instantly saw the effects of notations they manipulated, adding a note in the PA provided audio when play sequence reached that note. A user said, *“Everything seemed easy, because you hear the results of what u do”*. Pertaining to closeness to mapping in terms of inputs, 1 participant expressed satisfaction with how similar input notations used in Mic-Tok were to the other systems, further emphasizing how closely mapped input in Mic-Tok is to its expected result. They said *“it was intuitive to use, and it’s sounds and visuals were both reasonably polished. The controls were simple would be friendly to most input schemes”*

**Visibility & juxtaposition** (Positive): how easy was the visibility of the notations in the system? If you need to compare two sections of notation, can you see them at the same time? (41)

From observing our 14 participants, there were no observed problems with the visibility (or the juxtaposition), of the notations in Mic-Tok. One user commented, *“It was intuitive to use, and its sounds and visuals were both reasonably polished. The controls were simple would be friendly to most input schemes.”*

**Diffuseness/Terseness** (Neutral): what sort of things take longer to describe? (41)

There were no responses provided from our 14 participants relating to how they perceived the notations used in this dimension.

**Progressive Evaluation** (Positive): how easy can the notations used in the system provide feedback before completing a task?

There were no observed problems in regard to progressive evaluation from our 14 participants, there were no observed problems with evaluating their progress at any point while using the system. One user said, *“Everything seemed easy, because you hear the results of what u do”*.

**Premature Commitment** (Neutral): when working with the notations of the system, can you use them in any order? what problems can this cause a user’s work?

Asides one user asking how to change the timing of notes from our study, none of our 14 participants provided any response regarding the constrains in design.

To conclude, out of the 14 cognitive dimensions we used 10 which are applicable to Mic-Tok to evaluate its usability. Out of the 10, users had problems with 2 dimensions namely, role-expressiveness and viscosity. For viscosity, 8 participants complained about their inability to use a mouse while for role expressiveness, 4 participants complained about how some of the functions used in Mic-Tok were not understood. On the other hand, for two of the dimensions namely, diffuseness and premature commitment, there were no observed responses from our users that relate to these dimensions

### 5.2.5 User perception on expressivity of our tool using the Creativity Support Index

To capture user perception on the creativity/expressivity of our tool using user perception, we adopted a part of the Creativity Support Index (CSI) (40). The CSI as a whole is a set of questions that touch on six dimensions of creativity support. It has been applied to quantify the creativity support of systems, for example (43). It comprises of six dimensions which include Exploration, Expressiveness, Immersion, Enjoyment, Results Worth Effort, and Collaboration and they are used to evaluate the ability of a digital tool to support the creativity of its users. As part of our study, we asked users to rate our tool by their agreement to these questions from these dimensions on a scale of “Highly Disagree” (1) to “Highly Agree” (10) (see question 7 in **Appendix A**). For this research, we used only a part of the CSI method because we are only interested in how users think of our tool rather than quantify the our tool’s creativity support. Quantifying our tool will be something to explore in further research. **Table 2** below provides user ratings to these questions.

*Table 2: User responses to the 6 dimensions questions from the Creativity Support Index*

#	Dimension	Question	Minimum	Maximum	Mean
1	Enjoyment	I enjoyed using this tool.	4.00	10.00	8.14
2	Enjoyment	I would be happy to use this tool on a regular basis.	3.00	10.00	7.29

3	Exploration	It was easy for me to explore many different ideas, options, designs, outcomes using this tool.	2.00	9.00	6.93
4	Expressiveness(person)	I was able to be very creative while doing the activity inside this system or tool.	5.00	10.00	7.29
5	Expressiveness(tool)	The system or tool allowed me to be very expressive.	3.00	10.00	7.00
6	Immersion	My attention was fully tuned to the activity, and I forgot about the system or tool I was using.	2.00	10.00	7.50
7	Results worth effort	I was satisfied with what I got out of the system or tool.	4.00	9.00	8.14

From **Table 2** out of the 14 study participants, we observe that participants collectively rated the tools support for creativity above a median of 5. Enjoyment had an aggregated mean of 8.14. Most users said they had a nice experience using the tool. Exploration—which was the lowest— had a mean of 6.29 of Expressiveness was a 7.29 in terms of users themselves being able to express themselves and 7.0 in terms of the tool’s support for expressivity. Immersion was a 7.50. Results worth effort was a mean of 8.14 indicating the satisfaction users felt from their effort. On the other hand, we observe that immersion and exploration had the smallest ratings, meaning, some of our users felt they were not immersed in the activity, and similarly some also felt they were restricted while using Mic-Tok.

To view the distribution of user responses from our 14 participants, **Figure 14** below shows a box plot of the distribution of user responses to these factors



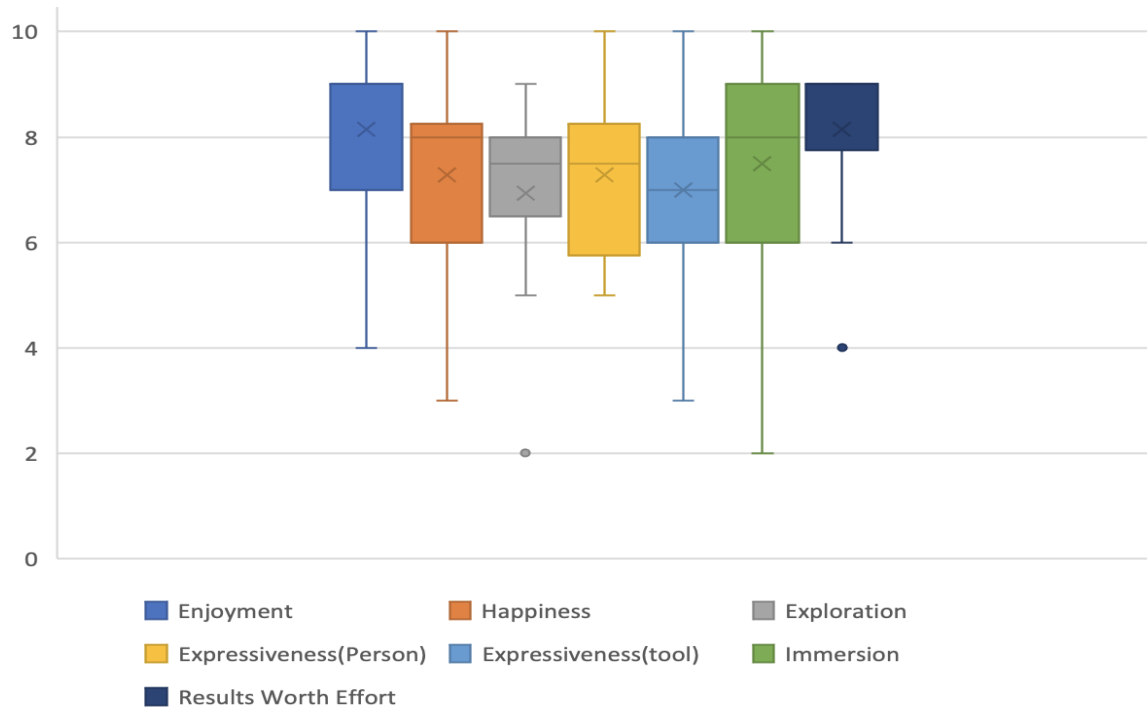


Figure 19: A box plot showing the distribution of user responses on a scale of 1-10. 1 meaning highly disagree while highly agree. From the distribution of user responses, we see that 75% of participants rated our tools support of creativity above average.

Out of 14 participants and from the distribution of answers, the box plot in **Figure 13** shows that 75% of users rated our tools support for creativity above the average of 5. An Indication that users perceive Mic-Tok as a tool that supports for creativity.

## Chapter 6. Discussion and Conclusion

The goal of this study was to make livecoding more accessible to users. To achieve this goal, we created a musical livecoding environment called Mic-Tok. With this study, we tried to answer two questions. (1) what methods can be potentially used in designing a musical livecoding environment and (2) how do users perceive the usability and expressive potential of the environment

### 6.1 Research Question 1: What musical livecoding environment can be designed for a constrained set of inputs

To answer our first research question, we created Mic-Tok—a musical livecoding environment for constrained inputs. The processes involved in designing Mic-Tok involved designing an initial version of Mic-Tok using a set of pre-existing designs and the Cognitive Dimensions of Notations as a guide. To improve Mic-Tok from its original paper prototype, we conducted formative studies and iteratively modified Mic-Tok following observations made from each study.

### 6.2 Research Question 2: How do users perceive Mic-Tok’s usability and expressive potential

#### *User perception on usability*

To answer our second research question, after our summative studies on users, we analyzed user responses on Mic-Tok using questions from ten dimensions of the CDN questionnaire. Out of these ten dimensions, users did not identify any major problem areas or concerns relating to six, they identified problems in two and did not provide any response relating to two. However, because six out of these dimensions were positive, we conclude that it is suggestive of the fact that users perceive Mic-Tok as approachable and expressive. In regard to problems users identified—Role-expressiveness and Viscosity, for role-expressiveness, users identified problems with the functions used. In the design of similar environments in future, greater emphasis needs to be placed on the method to which functions are chosen and constructed in such environments. For Viscosity,—in terms of users needing a mouse functionality, we put forth that the complaint may need further verification. Our hypothesis is that some of the users who wanted the mouse functionality might have said so because our study was conducted online, making the naturalistic disconnect and latency issues factor in. We assume the experience might be different if we were able to conduct our study in a more naturalistic setting.

### *User perception on expressivity*

To answer on the expressivity part for our second research question, after our summative studies on users, we analyzed responses from this study using the Creative Support Index. Analysis from our 14 participants, point towards users perceiving Mic-Tok as a tool that supports creativity

## **6.3 Future work**

There is work to be done in improving the function notations used in Mic-Tok. For example, adopting a much more effective way to describe functions so users easily understand what they do and their role. Also, we would like to conduct another user study in a more physical setting to investigate how users respond to Mic-Tok, especially in terms of the viscosity of the system. From this user study, we would like to move beyond just user perception to actually quantify Mic-Tok's potential for creativity support to have an idea how Mic-Tok performs in terms of creativity and what possible steps need to be taken to improve it. Another future work would include implementing Mic-Tok in other programming languages. A reason for this is so we explore more possibilities we couldn't find in Processing. Lastly, because Mic-Tok uses terminologies and processes common to computational processes, improving it so it can be used as a medium to educate users on computational thinking will be an area we would explore.

## Chapter 7. References

1. Collins N. Live coding and teaching SuperCollider. *Journal of Music, Technology & Education*. 2016 May 1;9(1):5–16.
2. Swift B, Sorensen A, Gardner H, Hosking J. Visual code annotations for cyberphysical programming. In: 2013 1st International Workshop on Live Programming (LIVE). 2013. p. 27–30.
3. Machuca MH, Cardoso JAL, Cerro JAT, Bravo FJL. Live coding for all: three creative approaches to live coding for non-programmers. *International Journal of Performing Arts and Digital Media*. 2016 Jul 2;12(2):187–94.
4. Ogborn D. Live Coding in a Scalable, Participatory Laptop Orchestra. *Computer Music Journal*. 2014 Mar;38(1):17–30.
5. Nilson C. Live coding practice. In: Proceedings of the 7th international conference on New interfaces for musical expression [Internet]. New York, NY, USA: Association for Computing Machinery; 2007 [cited 2021 May 6]. p. 112–7. (NIME '07). Available from: <https://doi.org/10.1145/1279740.1279760>
6. Blackwell A, Collins N. The Programming Language as a Musical Instrument. In Proceedings of the Psychology of Programming Interest Group. 2005;11.
7. Wang G, Cook PR. ChuckK: A Concurrent, On-the-fly, Audio Programming Language. Proceedings of International Computer Music Conference. 2003;8.
8. McLean A, Wiggins G. Live Coding Towards Computational Creativity. In: International Conference on Computational Creativity:5.
9. Roberts C, Kuchera-Morin J. GIBBER: LIVE CODING AUDIO IN THE BROWSER. Proceedings of the International Computer Music Conference. Vol 11, p.6. 2012:6.
10. Sorensen AC. Extempore: The design, implementation and application of cyber-physical programming language. The Australian National University. 2018 :202.
11. Magnusson T. The IXI Lang: A SuperCollider Parasite for Live Coding. Proceedings of the International Music Conference. 2011 Jan 1;
12. Aaron S, Blackwell AF, Burnard P. The development of Sonic Pi and its use in educational partnerships: Co-creating pedagogies for learning computer programming. *Journal of Music, Technology & Education*. 2016 May 1;9(1):75–94.
13. McLean A, Wiggins G. TIDAL -- PATTERN LANGUAGE FOR LIVE CODING OF MUSIC. In Proceedings of the 7th sound and music computing conference. :7.
14. Foster A. The Impact Of Live Coding Within An Educational and Performance Setting. Sr Proj Spring 2020 [Internet]. 2020 Jan 1; Available from: [https://digitalcommons.bard.edu/senproj\\_s2020/227](https://digitalcommons.bard.edu/senproj_s2020/227)

15. Collins N, McLean A. Algorave: Live Performance of Algorithmic Electronic Dance Music. In Proceedings of the International Conference on New Interfaces for Musical Expression :4.
16. Reas C, Fry B. Processing: programming for the media arts. *AI & Society*. 2006 Sep 1;20(4):526–38.
17. Gosling J. The Java Language Specification. Addison-Wesley Professional 2000 :604.
18. Sanner MF. PYTHON: A PROGRAMMING LANGUAGE FOR SOFTWARE INTEGRATION AND DEVELOPMENT. *J Mol Graph Model*, 17(1):7.
19. Roberts C, Wakefield G. Tensions & Techniques in Live Coding Performance. In: The Oxford Handbook of Algorithmic Music. 2018.
20. Rohrhuber J, de Campo A, Wieser R. NOTES ON LANGUAGE DESIGN FOR JUST IN TIME PROGRAMMING. The SuperCollider Book. MIT Press, Cambridge, Massachusetts. :4.
21. Impromptu : an interactive programming environment for composition and performance | In Australasian Computer Music Conference 2009: Improvise[cited 2021 May 6]. Available from: <https://eprints.qut.edu.au/31056/>
22. Wang G, Cook PR. 2004: On-the-Fly Programming: Using Code as an Expressive Musical Instrument. In: Jensenius AR, Lyons MJ, editors. A NIME Reader [Internet]. Cham: Springer International Publishing; 2017 [cited 2021 May 6]. p. 193–210. (Current Research in Systematic Musicology; vol. 3). Available from: [http://link.springer.com/10.1007/978-3-319-47214-0\\_13](http://link.springer.com/10.1007/978-3-319-47214-0_13)
23. About [Internet]. TOPLAP. 2012 [cited 2021 May 6]. Available from: <https://toplap.org/about/>
24. Mclean AGR, Griffiths D, Collins N, Wiggins G. Visualisation of live code. Electronic Visualisation and the Arts (EVA 2010) In 2010 [cited 2021 May 6]. Available from: <https://scienceopen.com/document?vid=93004c0a-a346-4926-9df6-1345696c36ad>
25. Bovermann T, Griffiths D. Computation as Material in Live Coding. *Comput Music J*. 2014 Mar 1;38(1):40–53.
26. Wagner A, Rudraraju R, Datta S, Banerjee A, Sudame M, Gray J. Programming by voice: a hands-free approach for motorically challenged children. In: CHI '12 Extended Abstracts on Human Factors in Computing Systems [Internet]. New York, NY, USA: Association for Computing Machinery; 2012 [cited 2021 May 6]. p. 2087–92. (CHI EA '12). Available from: <https://doi.org/10.1145/2212776.2223757>
27. Resnick M, Maloney J, Monroy-Hernández A, Rusk N, Eastmond E, Brennan K, et al. Scratch: programming for all. *Communications of the ACM*. 2009 Nov;52(11):60–7.
28. Ruthmann A, Heines JM, Greher GR, Laidler P, Saulters C. Teaching computational thinking through musical live coding in scratch. In: Proceedings of the 41st ACM technical symposium on Computer science education [Internet]. New York, NY, USA: Association for Computing Machinery; 2010 [cited 2021 May 6]. p. 351–5. (SIGCSE '10). Available from: <https://doi.org/10.1145/1734263.1734384>
29. Reas C, Fry B. Processing: a programming handbook for visual designers and artists. Cambridge, Mass: MIT Press; 2007. 710 p.

30. Mills, J.A., Di Fede, and Nicholas Brix, N. (2010). Music Programming in Minim. In *New Interfaces for Musical Expression 2010*.
31. sojamo/controlp5: A gui library for processing.org [Internet]. [cited 2021 May 6]. Available from: <https://github.com/sojamo/controlp5>
32. Bellingham M, Holland S, Mulholland P. A cognitive dimensions analysis of interaction design for algorithmic composition software. In: du Boulay B, Good J, editors. Brighton: University of Sussex; 2014 [cited 2021 May 6]. p. 135–40. Available from: <http://users.sussex.ac.uk/~bend/ppig2014/PPIGproceedings.pdf>
33. Kauhanen M, Biddle R. Cognitive dimensions of a game scripting tool. In: *Proceedings of the 2007 conference on Future Play* [Internet]. New York, NY, USA: Association for Computing Machinery; 2007 [cited 2021 May 6]. p. 97–104. (Future Play '07). Available from: <https://doi.org/10.1145/1328202.1328220>
34. Aaron S, Blackwell A, Hoadley R, Regan T. A principled approach to developing new languages for live coding. In *New Interfaces for Musical Expression*. 2011 Jan 1;
35. Blackwell AF, Britton C, Cox A, Green TRG, Gurr C, Kadoda G, et al. Cognitive Dimensions of Notations: Design Tools for Cognitive Technology. In: Beynon M, Nehaniv CL, Dautenhahn K, editors. *Cognitive Technology: Instruments of Mind*. Berlin, Heidelberg: Springer; 2001. p. 325–41. (Lecture Notes in Computer Science).
36. Green TRG, Petre M. Usability Analysis of Visual Programming Environments: A ‘Cognitive Dimensions’ Framework. *Journal of Visual Languages & Computing*. 1996 Jun 1;7(2):131–74.
37. Ray PP. A Survey on Visual Programming Languages in Internet of Things. *Scientific Programm*. 2017 Mar 8; 2017:e1231430.
38. Bannan B. The Integrative Learning Design Framework: 114 - 133 An Illustrated Example from the Domain of Instructional Technology. *An introduction to educational design research*, In 2013. p. 114–33.
39. Acharya A, Prakash A, Saxena P, Nigam A. Sampling: Why and How of it? Anita S Acharya, Anupam Prakash, Pikee Saxena, Aruna Nigam. *Indian Journal of Medical Specialties*. 2013 Jan 7;
40. Cherry E, Latulipe C. Quantifying the Creativity Support of Digital Tools through the Creativity Support Index. *ACM Transactions on Computer-Human Interaction (TOCHI)*. 2014 Jun 1;21(4):21:1-21:25.
41. Blackwell AF, Green TRG. A Cognitive Dimensions Questionnaire Optimised for Users. In *Proceedings of the Psychology of Programming Interest*. 2000.
42. Bobkowska AE. Abstract Cognitive Dimensions Questionnaire Applied to Visual Modelling Language Evaluation- a Case Study. In *Proceedings of the Fifteenth Annual Workshop of the Psychology of Programming Interest*. 2003.
43. Andolina S, Klouche K, Cabral D, Ruotsalo T, Jacucci G. InspirationWall: Supporting Idea Generation Through Automatic Information Exploration. In: *Proceedings of the 2015 ACM SIGCHI Conference on Creativity and Cognition* [Internet]. New York, NY, USA: Association for Computing

Machinery; 2015 [cited 2021 May 6]. p. 103–6. (C&C '15). Available from:  
<https://doi.org/10.1145/2757226.2757252>

# Appendix A

**Q1**  
How would you rate your knowledge on music theory?

No knowledge  
 Beginner  
 Intermediate  
 Expert

---

**Q2** iQ  
What do you like about the software?

---

**Q3** iQ  
What don't you like about the software?

---

**Q4** iQ  
What activities did you find the most challenging? And Why?

---

**Q5** iQ  
What activities did you find easy to do? And Why?

---

**Q6** iQ  
What did you learn from using the software?

**Q7**  
On a scale of "Highly Disagree" (1) to "Highly Agree" (10), choose a number that best describes your experience for each of the following sentences:

	0	1	2	3	4	5	6	7	8	9	10
I enjoyed using the system or tool.											
I would be happy to use this system or tool on a regular basis.											
It was easy for me to explore many different ideas, options, designs, or outcomes, using this system or tool.											
I was able to be very creative while doing the activity inside this system or tool.											
The system or tool allowed me to be very expressive.											



My attention was fully tuned to the activity, and I forgot about the system or tool that I was using.										
I was satisfied with what I got out of the system or tool.										