Flow: A Musical Composition Tool Using Gesture Interactions and Musical Metacreation

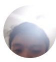
Chapter · June 2018

DOI: 10.1007/978-3-319-92285-0_24

5 authors, including:


Jordan Aiko Deja
De La Salle University


Kevin Gray Chan
De La Salle University


Migo Andres Dancel
De La Salle University


John Patrick Rivera Tobias
De La Salle University

Flow: A Musical Composition Tool using Gesture Interactions and Musical Metacreation

Jordan Aiko Deja, Kevin Gray Chan, Migo Andres Dancel, Allen Vincent Gonzales, and John Patrick Tobias

De La Salle University - Manila, Philippines
Center for Complexity and Emerging Technologies

Abstract. Music composition is a delicate and disciplined art form that is tedious and repetitive. In this preliminary study, we explore the design of an interaction that aims to balance the work of composers with the help of a mobile application. The process of musical composition is not easy for composers. Certain tasks such as figuring out succeeding notes often requires trial-and-error, as well as knowledge of certain theories. Existing technology has employed musical metacreation to assist in this process. This endows machines with the artificial creative capacity to perform musical tasks. In review, the existing technology has not been generally-used in all stages of the musical composition process. By combining several interaction technologies, composers can benefit by being able to do their tasks with significantly less cognitive load and time.

Keywords: Interaction Design, User Interface Design, Interaction Techniques, Gestural Input, Sound and Music Computing, Computational Creativity

1 Introduction

In the process of musical composition, composers tend to follow a strict set of theories and guidelines to maintain the aesthetic quality of the music they create [9, 5, 7]. The process is usually not standardized and can be customized depending on the task of the composer. As such, one composer's approach might not work effectively for another [3, 4]. To address this, we present a tool that will assist in the process rather than force composers to adapt to an unnatural approach. We intend to minimize cognitive load in every stage of the composition process. But generally, composers undergo a series of activities that allow them to draft their musical products, regardless of order or sequence [2, 6]. These are (1) ideation, (2) sketching, and (3) revision. We developed a product that enables composers to undergo these stages, while ensuring a user-centric approach in the process.

2 Framework & Methodology

We present the framework of this study as seen in Figure 1.

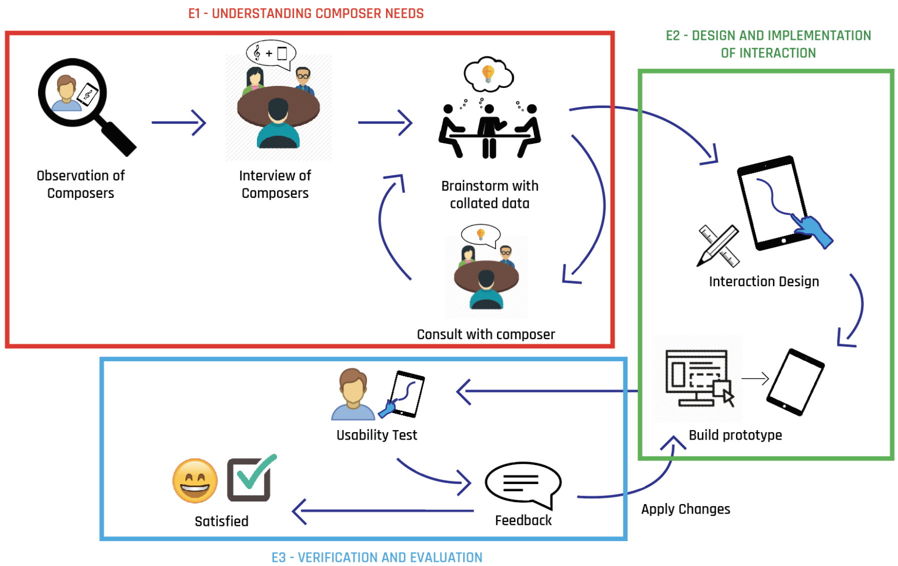


Fig. 1. Flow Research Framework

2.1 Participants

Five (5) subjects of both genders aged 18-40 were recruited through snowball sampling method to take part in the data collection and testing. Note that these subjects were categorized into two (2) user groups based on musical composition experience: amateur, and experienced. These groups were necessary to better understand the different processes that composers perform during composition.

2.2 Study Design

Our research approach was designed to be iterative, to allow for continuous development and improvement of the prototype. A single iteration involves building the prototype, testing it with composers, and using the resulting feedback and data to improve the next version of the prototype. In every iteration, we collected interaction data through the prototyping application: CogTool. This provided us with measurable data on KLM-GOMS, and Fitts Law [8] without having to measure them manually [10, 1]. In each test, the subject was asked to participate in three (3) test setups. The first setup would make the subject use a bare music sheet for composition. The second setup would ask the subjects to use a different mobile musical composition application. The third will make use of our prototype, Flow. The purpose for these setups was to allow us to compare our prototype against existing musical composition approaches. Subjects were asked to perform certain tasks (i.e. adding a note, erasing/deleting a note, etc.) in all the test setups. This helped us see the differences of how the users interacted with the setup for a certain task. For the test setup of Flow, the tasks were also meant to test the features shown in Table 1. After finishing the tasks in all test setups, the subjects were asked to answer a questionnaire. The questions can be seen in Table 2. Note that each of the questions were repeated for each feature (Shown in 1). The questionnaire aims to get quantitative data on the subject's experience while using Flow. Answers are on a scale of 1 - 4, with 1 (Never or Strongly Disagree) being the lowest, and 4 (Frequently or Strongly Agree) being the highest.

Table 1. List of Features

No.	Feature
F1	Add a Note
F2	Edit a Note
F3	Delete a Note
F4	Move Indicator/Cursor
F5	Move Line/Space Selector
F6	Selecting/Highlighting Multiple Notes
F7	Editing Multiple Notes
F8	Deleting Multiple Notes

Table 2. Feature Related Questions

No.	Question
Q1	How much did you use this feature to accomplish tasks?
Q2	Were you comfortable while using this feature?
Q3	Does this feature feel like how you naturally do the task when you're composing?
Q4	Is this feature new but easy to learn given enough time?
Q5	Does this feature make the task at hand way easier than how you do things?

2.3 Data Collection & Analysis

We performed the first round of testing with five (5) respondents. Two (2) of these were experienced while the rest were amateurs. Prior to testing, the subjects were asked to fill out a consent form outlining the tasks that will be performed. These tests were recorded through video, and audio, so that they could be reviewed later. Each test lasted for about 30-45 minutes. At the end of the first iteration's testing phase, we recorded the users' comments and the observed pain points. Through this, we were able to identify the more common problems in the design and used these to improve the prototype for the succeeding iteration.

3 Prototype

The prototype was built using Swift 4.0 for the iOS platform and is optimized for iPads. The interaction was designed to support a seamless method of musical composition using different types of gestures. The three main activities: *ideation*, *sketching* and *revision* were considered in the process of designing these interactions. The first method, *ideation*, is when a composer is in need of assistance in creative thinking. The system assists the composer in this task by providing audio feedback whenever they input a note. The audio feedback was found to be helpful because it let them figure out the flow of their music easier.

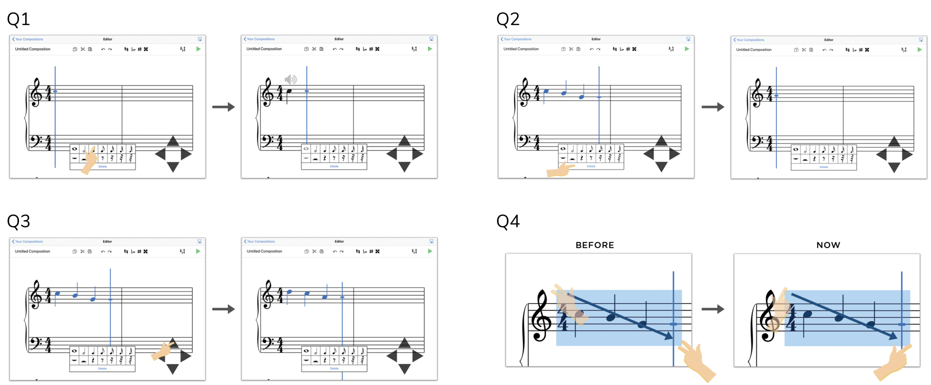


Fig. 2. Flow functions. Q1 demonstrates the add note function which plays a tune when a note is tapped into the the digital sheet. Q2 shows the delete function. Shown in Q3 is the transposition function which increases or decreases the pitch of the selected notes using the up or down arrow keys respectively. Q4 illustrates the highlight interaction before and after changes were made due to the user testing.

The second method, *sketching*, is when the composer makes the draft of the initial melody of a composition. The application allows the composer to test out different rhythms while composing with just a tap. Users can simply tap a note from the menu which will then appear on the digital music sheet (see Figure 2 Q1). The composer can then repeat this process to complete the composition. When the composer wants to revise a portion of his existing work then he can proceed to *Revising*.

Revising is supported by the system through different touch gestures as shown in Figure 2 Q4 and Q2. In the initial prototype, to select a series of notes, the user performs the highlight by placing two fingers on the screen and dragging them in a diagonal direction until the rectangular highlight region covers the notes the they want to select. However, from the results of the testing and user feedback, we redesigned this interaction to require only one finger to highlight. We believe that this interaction not only uses less fingers, but is also less prone to failed attempts. After highlighting, the user can choose to either edit, delete, cut, copy, or paste certain notes by choosing the rights options from the top menu. Similarly, the user can tap on the up or down arrow keys to instantly transpose the selected notes to a higher or lower pitch respectively (see Figure 2 Q3).

4 Results

4.1 Preliminary Results from CogTool

CogTool was used to collect preliminary cognitive interaction data for the prototype. Table 3 lists attributes that provide estimates on how efficient a person would be when using the application.

Table 3. CogTool Results Based on Module

Attribute	Attribute Description	Value
	Motor Module	
PECK Fitts Coefficient	b coefficient in Fitts's equation for PECK movements.	0.075
Default Target Width	Effective width, in degrees visual angle, of targets with undefined widths.	1.000
Minimum Fitts Time	Minimum movement time for an aimed [Fitts's] movement.	0.100
Motor Burst Time	Minimum time for any movement.	0.050
Motor Initiation Time	Time to initiate a motor movement.	0.050
Motor Feature Prep Time	Time to prepare a movement feature.	0.001
	Imaginal Module	
Imaginal Delay	Time in seconds to respond to an imaginal request	0.200
	Temporal Module	
Time Noise	Temporal noise	0.015
Time Master Start Incement	Temporal start interval	0.011
Time Multiplier	Temporal multiplier	1.100

The attributes mentioned in Table 3 represent cognitive operations from the Keystroke-Level Model (KLM) [11] generated by CogTool. In the Motor Module, it is best to minimize the time it takes to perform tasks. Flow does this well according to its Minimum Fitts Time, which indicates the time it takes to move to an area based on the distance from the target and the width of the target [8]. This means that it takes approximately less than a second on average to locate buttons or targets in Flow's interface. Similarly, the Imaginal Delay notes the time it takes to respond to requests from the system like in confirmation buttons or pop ups. A value of 0.200 suggests that there's a very short processing time in between requests.

4.2 Testing Results

Table 4 lists an overview of the results from the questionnaire given after performing the test setups in Iteration 1 and Table 5 lists the overview for Iteration 2. As mentioned before, answers are on a scale of 1 - 4, 1 (Never or Strongly Disagree) being the lowest, and 4 (Frequently or Strongly Agree) being the highest for Iteration 1. There is an added option for choosing 0 (N/A) for features that were unused during the testing for Iteration 2. Each of the questions are repeated for each feature.

Table 4. Feature Related Questions

	Q1	**Q2**	**Q3**	**Q4**	**Q5**	**AveF**
F1	4.0	3.2	2.6	3.4	2.0	3.04
F2	2.2	1.8	2.0	3.0	1.8	**2.16**
F3	2.0	1.8	1.8	3.2	2.4	2.24
F4	4.0	4.0	3.4	3.6	3.4	**3.68**
F5	3.8	3.2	3.6	3.4	3.0	3.40
F6	2.4	2.0	1.8	3.2	2.8	2.44
F7	2.4	1.8	2.2	2.4	2.4	2.24
F8	2.2	1.8	2.2	2.4	2.2	**2.16**
AveQ	2.88	2.45	2.45	3.08	2.50	

Iteration 1 From the data gathered and analyzed, a clear difference in respondent sentiment was observed from features that needed multiple note selection (see F2 and F8) versus those that did not. This is most notable in the edit and delete features which had the lowest scores on average. Both of these features required the selection feature. Most of the respondents were not able to figure out and correctly execute the gesture for selection which was a two-finger drag. Although it is common for mobile applications to use one-finger drag to scroll, the results suggest otherwise. It was found that the gesture for highlighting (two-finger drag) would have felt more natural if it was switched with the scroll gesture (one-finger drag). Other than the note selection feature, respondents have expressed that most of the features worked well and felt comfortable to use. Although it was not like how they would regularly write music (i.e. pen and paper), Flow's method of composing was easy to learn and get used to. Majority liked the ease of using the cursor/indicator because they can simply tap on the location they want or use the arrow keys when they want to be accurate.

Table 5. Feature Related Questions

	Q1	**Q2**	**Q3**	**Q4**	**Q5**	**AveF**
F1	4.0	3.2	2.6	3.4	2.0	3.04
F2	2.2	1.8	2.0	3.0	1.8	**2.16**
F3	2.0	1.8	1.8	3.2	2.4	2.24
F4	4.0	4.0	3.4	3.6	3.4	**3.68**
F5	3.8	3.2	3.6	3.4	3.0	3.40
F6	2.4	2.0	1.8	3.2	2.8	2.44
F7	2.4	1.8	2.2	2.4	2.4	2.24
F8	2.2	1.8	2.2	2.4	2.2	**2.16**
AveQ	2.88	2.45	2.45	3.08	2.50	

Iteration 2

5 Conclusion & Future Work

This study provides a framework for designing mobile musical composition applications. To achieve this, we first had to perform user research which included performing interviews with composers and observing their creative processes. The results of the initial tests led to the design and development of a usable mobile musical composition tool that aided composers through musical metacreation. Given that only an initial prototype was used during the testing, some features were not yet implemented or fully working. The current prototype only included the functions necessary for a composer to create a complete composition. Similarly, the musical metacreation feature was not yet present in the prototype. However, the interviews and tests done during the early prototyping stage suggest that musical metacreation, or being given suggestions on possible notes to write next, would be a valuable tool for composers during their *ideation* activities. We have yet to incorporate the results from the second and third iteration and their corresponding feedback. We performed analysis using CogTool and these results are yet to be correlated with the results of the interaction testing. Future work would also include the integration of a machine learning inference engine that might possibly help composers in the events of a ”creative block”.

References

1. Bellamy, R., John, B., Kogan, S.: Deploying cogtool: integrating quantitative usability assessment into real-world software development. In: 2011 33rd International Conference on Software Engineering (ICSE). pp. 691–700 (May 2011)
2. Bennett, S.: The process of musical creation: Interviews with eight composers. Journal of research in music education 24(1), 3–13 (1976)

3. Cheng, C.: On approaching a performance of paul hindemith's der schwanendreher (2016)
4. Collins, D.: The Act of Musical Composition: Studies in the creative process. Routledge (2016)
5. Collins, D.: A synthesis process model of creative thinking in music composition. Psychology of music 33(2), 193–216 (2005)
6. Graf, M.: From Beethoven To Shostakovich-The Psychology Of The Composing Process. Read Books Ltd (2013)
7. Kikuchi, J., Yanagi, H., Mima, Y.: Music composition with recommendation. In: Proceedings of the 29th Annual Symposium on User Interface Software and Technology. pp. 137–138. ACM (2016)
8. MacKenzie, I.S.: Fitts' law as a research and design tool in human-computer interaction. Human-computer interaction 7(1), 91–139 (1992)
9. Rothgeb, J.: Strict counterpoint and tonal theory. Journal of Music Theory 19(2), 260–284 (1975)
10. Sauro, J.: Estimating productivity: composite operators for keystroke level modeling. Human-Computer Interaction. New Trends pp. 352–361 (2009)
11. Sauro, J.: Measuring task times without users (2011), https://measuringu.com/predicted-times/