Project Proposal – OctoLooper

Wren Hawthorne
wrenhawth@gatech.edu

Abstract—The creation of original pieces of music through songwriting and composition processes has been shown to have significant benefits for students in early music education. However, there are significant practical challenges for students and teachers in incorporating these tasks into music education in primary schools using traditional methods which rely on music theory knowledge and sufficient instrumental proficiency. There is a potential for technology to increase the accessibility of these educational goals, learning from research on Creativity Support Tools (CST) to facilitate and enhance creativity. A tool is proposed, OctoLooper, to address this gap, inspired by research on musical visualization and previous attempts to create musical software for children.

1. INTRODUCTION

1.1. The benefits and challenges of composition in early music education

Music can be a powerful outlet for creative expression for people of all ages, a way to reflect, develop and express ideas, and an important aspect of creative development. Musical composition, the process of creating a new piece of music, can be a rewarding, impactful, and immersive aspect of music education for young students (Johnson-Green, 2020). Along with musical improvisation, composition is a key opportunity for creative exploration, going beyond attempts to perform or reproduce an existing song or musical piece. Music curriculum that incorporates students' own musical voices can also engage students who might be otherwise disengaged in the music education classroom (Norman, 2021).

Despite these benefits, a survey of music teachers found that only a small percentage of these teachers used composition activities often in their classrooms. These teachers cited challenges such as lack of classroom time, logistical concerns over space and noise levels, as well as uncertainty over effectiveness tools and teaching strategies (Strand, 2006). The difficulty in evaluating and carrying out assessments in classroom activities based around students composing musical pieces as an exploratory process presents additional difficulties (Winters, 2012). Composition can also be seen as something that is only achievable by students with a certain level of music theory knowledge, or only accessible to "gifted musicians", which may discourage teachers from including it in music education in primary school settings (Laato et al., 2019).

1.2. Why creative expression matters, and how to encourage it

The creative development of students can have wide-ranging impacts on important learning outcomes, including not only the previously mentioned creative expression, but also problem solving and their ability to generate novel ideas (Hagen et al., 2023). Opportunities for musical expression have important cognitive benefits even from a very young age (Zadnik & Habe, 2017). Expressive activities like songwriting can also have positive effects on social-emotional skills of children, enhancing the social skill benefits already present in typical music education (Dweck, 2023).

One promising option in encouraging and fostering creativity in students is the use of student-directed approaches such as project-based learning. These approaches have been shown to have beneficial impacts on the creativity, engagement, and motivation of students by allowing them to incorporate areas that are personally meaningful to them into their work (Cahyani, 2021; Tobias et al., 2015).

1.3. The promise of "Low Thresholds, High Ceilings, and Wide Walls"

Digital tools can be used to supplement and enhance human creativity, leveraging technology to assist people in expressing their ideas in more innovative and productive ways. The study of Creativity Support Tools (CST) spans multiple disciplines, just as these tools have applications in countless disciplines, including art, music, engineering, science, and countless others (Shneiderman et al., 2006).

From this interdisciplinary research, design recommendations have been proposed, and these recommend a focus on creating tools with "Low Thresholds, High Ceilings, and Wide Walls" (Resnick et al., 2005), which can be summarized as follows:

- **Low Thresholds**: Tools should be easy for novices to get started with, with interfaces that are accessible and non-intimidating
- **High Ceilings**: Tools should support a sufficient complexity for the level of sophistication needed by expert users
- **Wide Walls:** Tools should support a variety of use cases and explorations, enabling a diversity of creative outcomes

2. RELATED WORK

2.1. Traditional composition methods in the classroom

The process of creating a new piece of music can take many forms, and in classrooms a variety of approaches have been taken when incorporating composing tasks into music education. Often these have involved some amount of creating musical notation using either physical sheet music or, in more recent years, the use of music notation software (Laato et al., 2019). When paper-based approaches are used, instruments are often available to allow the students to hear explore possibilities and hear what their piece sounds like (Wong & Lim, 2017), though this is not always the case (Strand, 2006). A certain level of instrumental proficiency is needed to support this form of paper-based classroom activity, making technology-based approaches more approachable and accessible (Laato et al., 2019).

2.2. Digital Audio Workstation (DAW) software use in classrooms

Digital Audio Workstation (DAW) software is a category of applications designed to support the creation of various types of digital audio. For this proposal, I will specifically be referring to the subset of DAW software aimed at music creation. An example of a DAW, GarageBand, can be seen in ***Figure 1***.

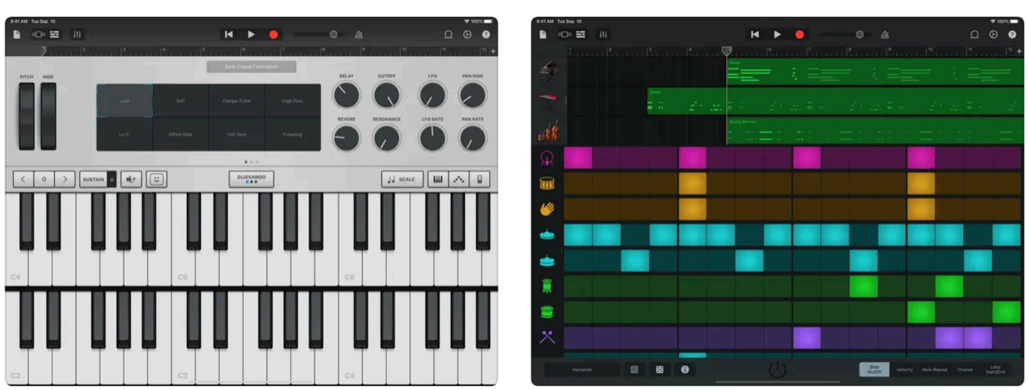

Figure 1—Screenshots from the GarageBand app on iPad, a free and popular DAW application. Source: Apple App Store

These applications often support a host of features to make the process of music creation easier, such as notation editing, recording, editing, mixing, and exporting musical pieces. In addition to recording through microphones, these applications will include a suite of virtual instruments, including synthesizers, sampled traditional instruments, and various combinations of sounds.

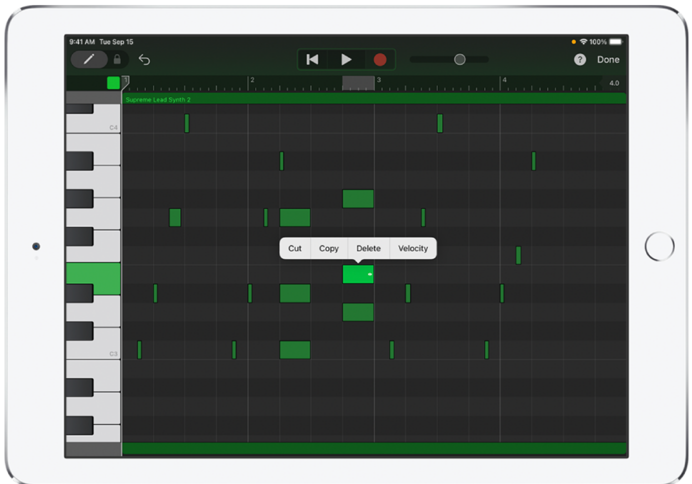

Figure 2—An example of a Piano Roll input method. Source: GarageBand User Guide

For these virtual instruments, DAWs support a variety of input methods, such as playing them through a touch interface or external MIDI keyboard, or through adding notes directly to a "Piano Roll" representation of the song, as shown in ***Figure 2***. In a Piano Roll interface, the pitch of notes is represented by its vertical position, and its timing is represented by its horizontal position, similar to traditional western musical notation.

When technology is used to enable composing in classroom settings, tablet applications like GarageBand for iPad are often used (Huovinen & Rautanen, 2020; Norman, 2021) due to its availability and relatively simple interface. Using tablet software for composing tasks addresses some of the practical difficulties with traditional composition methods in large classroom settings, as headphones can help make noise levels more manageable, and a variety of input levels can help support students with various amounts of music theory knowledge.

This ease of access, combined with the input and visualization techniques used (such as the Piano Roll interface discussed above) may help students develop a deeper understanding of music theory through the process of music composition (Laato et al., 2019). However, there are also drawbacks to their use in classroom settings, as DAW interfaces may interrupt creative flow and require more pre-planning in their use, particularly in collaborative environments (Huovinen & Rautanen, 2020).

2.3. Musical composition tools for children

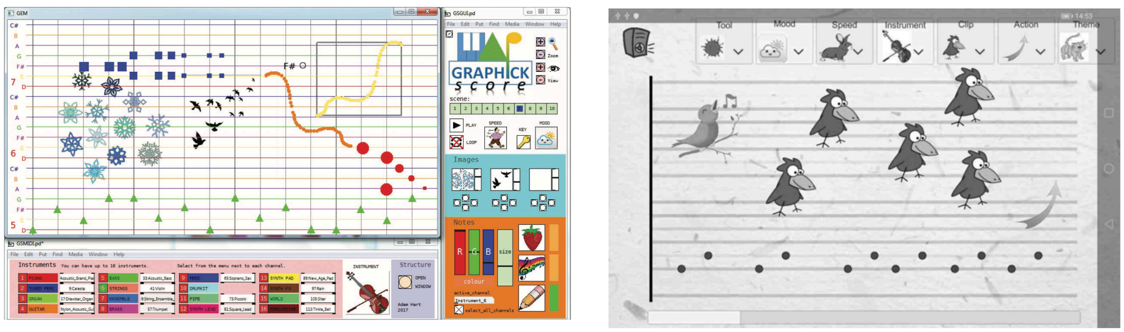

Figure 3—GraphickScore (left) and Paynter (right), two examples of graphical composition tools designed for children. Source: (*Hart, 2017*; *Hart & Williams, 2021*)

Multiple attempts have been made to create music compositional tools for use by children, often using graphical interfaces to enable young children without knowledge of music theory to create novel pieces of music. In some applications, such as GraphickScore and Paynter (pictured in ***Figure 3***), the inclusion of cartoon illustrations as options for this musical drawing was found to inspire children in the process of music making. In other words, the students would use the stickers to "draw" a story that would then be reflected in the musical piece that they created (Hart, 2017; Hart & Williams, 2021).

Figure 4—Screenshots of the graphical score (left), arpeggio (middle), and drum sequencer (right) functions of the Synth4Kids application. Source: (Mygdanis, 2022)

Some musical applications aimed at children are created with a surprisingly rich feature set, such as Synth4Kids. In addition to containing adjustable synthesizer sounds, filters, and effects, this app allows users to draw a graphical score, customize arpeggios (musical chords broken up into individual notes) and program drum patterns (as shown in ***Figure 4***). Elements of its design are inspired by various pedagogical techniques from music education, including its use of the pentatonic scale for easier improvisation and its pairing of pitches with colors (Mygdanis, 2022).

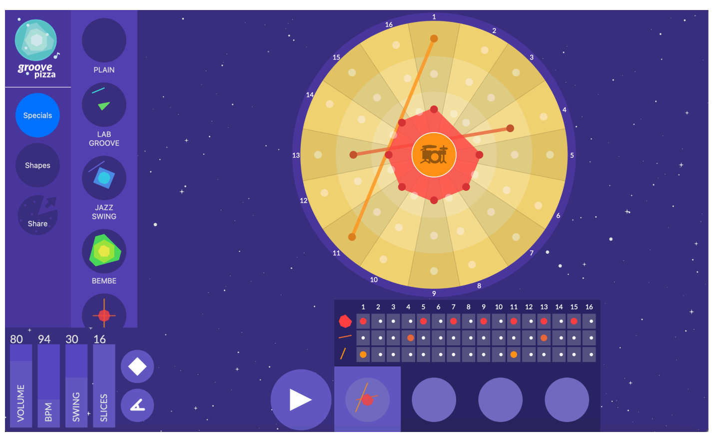

Figure 5—The interface of Groove Pizza, showing how the timing of notes in the "chameleon" preset contains certain geometrical relationships when arranged in a circle. Source: ***Groove Pizza by NYU Music Experience Design Lab***

Another example of the use of visualization to develop understanding of musical concepts can be seen in the "Groove Pizza" application (***Figure 5***), where the steps of a repeated rhythm are illustrated on a circular grid. By connecting the notes of each instrument in that circle, geometric lines, angles, and shapes are formed that reflect complex musical information about their timing and patterns. This design direction is continued by incorporating it into the interface input methods, as users can add either individual notes to the existing pattern or add a "shape" of notes all at once. Users can also rotate patterns of notes to experiment with rhythmic displacement, showing how novel interfaces can make advanced musical concepts more accessible (Hein & Srinivasan, 2019).

3. PROPOSED WORK

3.1. OctoLooper, an app to help children create their own music

My proposal centers around the creation of a web application to enable children (target demographic would be children between the ages of 5-10) to compose and create their own songs, using a fun and accessible loop-based workflow.

A tentative name for this proposal is OctoLooper, as I intend to have the interface center around a cartoon octopus to make the interface more fun and approachable for children. An octopus was chosen as the number 8 has several relevances to common musical concepts in popular music and western musical traditions:

- Major and minor scales have 8 notes
- When creating repeating rhythms in a step sequencer or looper tool, 8 is a common length for a short pattern
- In a musical piece with 4/4 time signature, each bar has space for 8 eight notes

Ideally, the legs of the octopus could be integrated into the various visualizations of the musical piece that is being created by the child.

3.2. Key Features

3.2.1. Step-sequencer based workflow

To reduce the need for technical proficiency, the app would resolve around the concept of a musical sequencer, where musical notes (or chords) are placed on a series of *steps* (equal time intervals) through either touch or mouse input. The musical sequence can be replayed on a loop as it is being constructed, so that users can hear the loop as it is being constructed.

I hope to combine the best of both step-sequencer workflows and loop-based workflows by enabling users of the app to create "loop blocks" through the step-sequencer interface, and then combine those loops to create a larger musical piece.

3.2.2. Visualizations that assist musical understanding and engage users

Novel visualizations can provide enhanced understanding of musical concepts, by bringing to light information that is not otherwise conveyed by traditional music notation. Traditional western music notation is largely a linear representation, it starts from the top-left of the score and proceeds (with certain elements of repetition) to the bottom-right, being "read" line-by-line like a book.

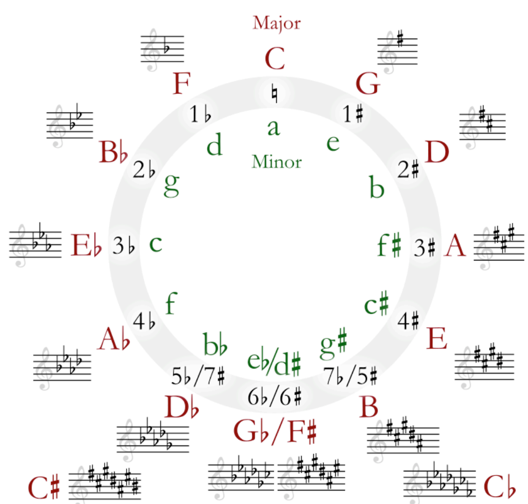

Figure 6—The circle of fifths. Source: Wikipedia

Prior tools like Groove Pizza have shown how emphasizing the circular nature of musical rhythms can illuminate rhythmic concepts and relationships and enable new ways of musical thinking for users (Hein & Srinivasan, 2019). With OctoLooper, I hope to expand those concepts into new areas by exploring visualization and input concepts for melodies and chord progressions. An example of existing musical theory to pull from is the "Circle of Fifths" (***Figure 6***), which organizes pitches in a way that is helpful for understanding the harmonic relationships between pitches, which is in turn useful for creating chord progressions.

To make the application engaging enough for young users to experiment with these musical concepts, I hope to make these visualizations fun and interesting as well. Potential examples to explore would be tentacles of the titular cartoon octopus representing the musical "path" of a melody, or different colored tentacles representing different types of chords when creating backing chord progressions for a song.

3.2.3. Easy-to-use input methods

OctoLooper will be divided into three main sections for users to create songs (or sections of songs) by creating a backing rhythm, the chord progression that carries the "feel" of a song, and the melody that makes up the leading voice of a song or piece of music. Additional parts of the interface will allow users to specify things such as the tempo of a song or the key of a song in an approachable way.

The following section titles do not reflect how these application parts will be referred to in the application itself, as they will instead be renamed into concepts that are more recognizable by children who may not have been exposed to these musical concepts.

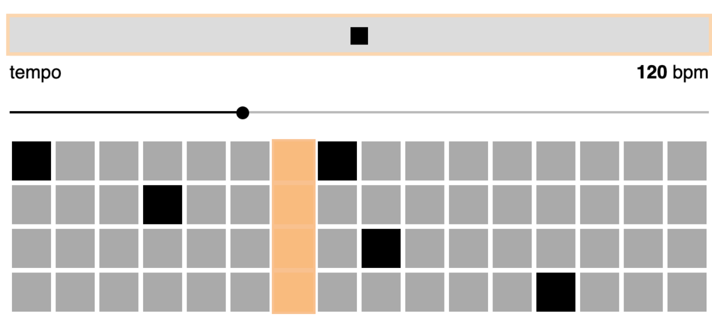

Figure 7—An example of a simple step sequencer built using Tone.js. Source: ***Tone.js documentation***

Rhythms—For rhythmic input, the OctoLooper prototype will initially offer a selection of built-in rhythms from a variety of styles. Time-permitting, a step sequencer (a basic example is shown in ***Figure 7***, as well as more complex examples in ***Figure 4*** and ***Figure 5***) will be added to allow users to customize these existing rhythms or create a new pattern from scratch. Similar to step sequencers in DAWs and other music applications, this will consist of a series of horizontal rows, each representing a different percussion sound, and allow users to specify which musical steps those sounds are triggered on.

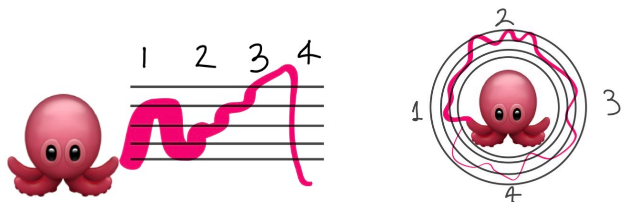

Figure 8—Rough sketches of design ideation for melodic input, showing a linear (left) and more circular (right) visualization of the melodic path (shown in pink)

Melodies—For creating melodies, an interface will be created to allow the creation of a "melodic path", which will then be translated into a melody that is in key with the rest of the musical piece. This may take the form of a graphical interface, where users will draw a line to represent the melody (explorations of this concept are pictured in ***Figure 8***) or it may take the form of a drag-and-drop interface, where notes are dragged from another section of the app onto the musical grid located next to or around the central octopus character.

Chords— Chords are groups of notes that are played together to establish harmony, either simultaneously or individually in rhythmic patterns called arpeggios. Chords are often played in repeated patterns, called chord progressions, which form the basis of harmony in western popular music across multiple genres. Their interactions with tonality and resolution make them a valuable tool for understanding western classical music as well as music from other cultural traditions.

However, thinking about music in terms of chords or chord progressions may be difficult for users with limited amounts of music theory training. In other music software, adding chords often requires playing or inputting the multiple notes that make up a chord. While some software features such as GarageBand touch instruments add features to allow users to play a chord with one touch, this doesn't allow placing instances of a chord in a song through the use of a sequencer interface, instead requiring users to play a chord "live" to be recorded.

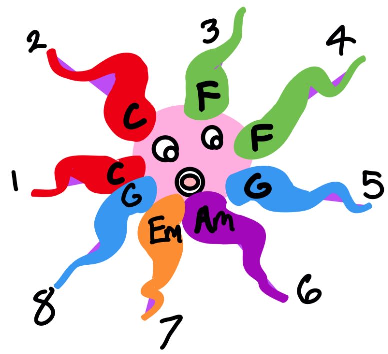

Figure 9—Exploration of using color pairing to represent a circular chord progression.

With OctoLooper, I want to explore how interface design and visualizations can make constructing a sequence of chords to establish the "feel" of a song more accessible to young musicians. An example of this exploration is shown in ***Figure 9***, where a circular interface and paired colors is used to make selecting a series of chords more accessible and fun for users. In this proposed example, users would touch or click on each tentacle to toggle between a selection of chords that were in-key with the current song. Additional interface design will be considered to better illustrate and communicate the difference between different types of chords such as the "happy-sounding" major triads and the "sad-sounding" minor triads.

3.2.4. Prompting for creativity scaffolding

As creativity scaffolding can be an effective tool for encouraging creativity (Hagen et al., 2023), OctoLooper will offer a feature to help spark ideas and inspire users to create new songs. This "What should I make?" feature will use open-ended prompts to give users a direction to start creating, without including narrow requirements around things like sounds used and genre.

In addition, time-permitting, OctoLooper will include features such as song templates and randomization to give users jumping-off points for the creation of new songs.

3.2.5. *A variety of fun and musical sounds*

The usability and engagement of a musical application is significantly affected by the instruments and sounds available when using it. This would especially be true for this project, as a simple and accessible musical interface for children needs to provide sufficient sounds for musical exploration.

With this in mind, OctoLooper will contain a variety of sounds and instruments to support a variety of genres users may wish to create music in, such as electronic, pop, hip-hop, rock, and jazz. OctoLooper will also contain a number of sound effects to facilitate "musical playing" and be potentially more engaging and fun for younger users.

3.3. Proposed Architecture

3.3.1. *Web application: React web-app with Shoelace components*

To better reach a variety of desktop and mobile devices, OctoLooper will be created as a web application. This responsive web app built with react will respond to both mouse input on desktop and touch input on mobile.

The shoelace component library will be used to speed up the prototyping process by using opinionated, ready-made components for the general interface of the application.

3.3.2. *Timing and sound generation: Tone.js*

To synthesize and produce a variety of sounds while coordinating the timing of interleaved musical aspects, the Tone.js library will be used. This library provides important tools for scheduling Web Audio events, as well as a series of synthesizers, filters, and digital effects that can be used for music creation.

3.3.3. *Saving and persisting songs: Web Storage API*

While sharing created songs is likely outside the scope of this proposal, the Web Storage API will be used to to enable persisting of created songs to user's individual computers. The use of localStorage will allow users to store multiple songs in their browser's memory and "load" them into the app to continue working on them.

4. DELIVERABLES

4.1. Intermediate Milestone 1

The first intermediate milestone for OctoLooper will center around the visualizations and input methods for rhythms and chord progressions, as this will be an opportunity to gather feedback on if the tool makes these concepts more accessible to those without musical training. Ideally, this will take the form of not only a video presentation demoing this portion of the OctoLooper app, but will also involve sharing access to the app to a wider audience to gather feedback.

4.2. Intermediate Milestone 2

While the first milestone only included public access of OctoLooper as a stretch goal, it will be integral for this second milestone. The video component of the demo will showcase the aspects of the interface I am most interested in feedback on, namely the chord progression and melodic inputs. The public beta and request for feedback will be used to guide my refinements and updates in the final weeks of the project.

4.3. Final Project

The Final Paper and Final Presentation will focus on my research on CSTs and the role of music composition in early music education, and how that informed and inspired the design and development of OctoLooper.

In addition, the final project archive will include the code repository for the OctoLooper application, along with documentation on running it locally. As it is a web application, a link will also be included to a hosted version of the tool.

As it is developed, I plan to create a record of design iterations and explorations for the visualizations and musical input methods that exist at the core of OctoLooper, and those records will also be included in the submitted archive.

5. TASK LIST

Week #	Task #	Task Description	Estimated Time (hours)
6	1	Create basic web app layout, deployment, and structure	2
6	2	Source sample rhythms and instrument sounds (includes Tone.js learning)	7
6	3	Explore options for chord progression input	8
7	4	Create looping mechanism (includes Tone.js learning)	6
7	5	Integrate chord progression input with sound generation (includes Tone.js learning)	7
7	6	Write script for Intermediate Milestone 1 video	2
7	7	Record demo of basic looping and chord input for Intermediate Milestone 1	2
INTERMEDIATE MILESTONE 1 DUE			
8	8	Create basic melodic input	8
8	9	Enhance visualization of different musical patterns	5
8	10	Integrate melodic input with sound generation	5
9	11	Performance profiling for latency and sound timing, optimize if necessary	8
9	12	Allow users to persist songs into storage (and potentially share through links or files)	4
9	12	Deploy application in a publicly accessible way for beta testing	3
9	13	Write script for Intermediate Milestone 2 Video	2
9	14	Record demo of interface additions and publicize beta testing for Intermediate Milestone 2	2
INTERMEDIATE MILESTONE 2 DUE			
10	15	Synthesize and incorporate relevant feedback (including bug fixes) from public beta	4
10	16	Expand rhythmic section to include step sequencer	7
10	17	Write script for Final Presentation	4
11	18	Write Final Paper	10
11	19	Record Final Presentation	2
11	20	Compile and submit Final Project	2
FINAL PROJECT DUE			