Programming music with Sonic Pi promotes positive attitudes for beginners

Petrie, Christopher

2022-04

Petrie , C 2022 , ' Programming music with Sonic Pi promotes positive attitudes for beginners ' , Computers and Education , vol. 179 , 104409 . https://doi.org/10.1016/j.compedu.2021.104409

http://hdl.handle.net/10138/352026
https://doi.org/10.1016/j.compedu.2021.104409

Computers & Education 179 (2022) 104409

Contents lists available at ScienceDirect

Computers & Education

journal homepage: www.elsevier.com/locate/compedu

Programming music with Sonic Pi promotes positive attitudes for beginners

Christopher Petrie

Faculty of Educational Sciences, Helsinki University, Helsinki, Finland

A R T I C L E I N F O

Keywords:
Interdisciplinary projects
Applications in subject areas
Improving classroom teaching

A B S T R A C T

Programming is often misaligned with beginner students' interests and viewed as difficult. However, most students and teachers are not aware that it is possible to utilise domain-specific programming languages that combine programming with other domains like music making. Sonic Pi is one free domain-specific programming platform that enables beginners to code music, which has been designed for and used in schools since its first release in 2012. However, there is a lack of academic research on the Sonic Pi platform about the extent it may affect beginner student attitudes towards programming in a school context. The aim of this study was to investigate the extent Sonic Pi may help to promote positive attitudes towards programming. A mixed-methods case study was developed and trialled in school time with a middle school class, which measured student attitudes with the three subscales of enjoyment, importance, and anxiety. Overall, the results confirmed an alternative hypothesis that all students' subscales for programming attitude increased significantly. While these findings are not generalisable due to its limited scope, they are very positive to inform the design and use of platforms like Sonic Pi in comparison to similar music coding platforms like EarSketch and TunePad.

1. Introduction

Programming is often misaligned with beginner students' interests and viewed as difficult (Anderson, Lankshear, Timms, & Courtney, 2008; Grover, Pea, & Cooper, 2016; Hava & Koyunlu Ünlü, 2021; Margolis & Goode, 2016). Student attitude towards a particular educational endeavour predicts the likelihood that they will persist through difficult problems like those in programming (Fishbein & Ajzen, 1975; Wanzer, McKlin, Freeman, Magerko, & Lee, 2020). Attitude is especially important in middle school where young people are forming strong views on subject areas like computing, which are sometimes available as specialist options later in high school (Günbatar, 2020; Hava & Koyunlu Ünlü, 2021).

A plethora of different programming platform designs aimed at young people have emerged in recent years that attempt to make learning easier and more appealing to beginners (Bell & Bell, 2018; Perera, Tennakoon, Ahangama, Panditharathna, & Chathuranga, 2021). Domain-specific programming languages that combine a subject like music with programming have been developed as part of these efforts, which could be more motivating for students and consequently influence beginner attitudes to be more positive (Aaron & Blackwell, 2013; Aghaee, Blackwell, Stillwell, & Kosinski, 2015; Bell & Bell, 2018; Köppe, 2020). Specifically linking programming with music composition has already shown to potentially promote positive attitudes towards programming at a school level, which may help to alleviate the problem of beginner students viewing it as difficult (Burnard, Lavicza, & Philbin, 2016; Köppe, 2020; Wanzer

E-mail address: christopher.petrie@helsinki.fi.

https://doi.org/10.1016/j.compedu.2021.104409
Received 14 March 2021; Received in revised form 14 December 2021; Accepted 15 December 2021
Available online 17 December 2021

et al., 2020). Studying these designs of educational programming platforms for young people are important because many countries are integrating programming as part of a core subject (Hava & Koyunlu Ünlü, 2021; Lye & Koh, 2014). Additionally, having a diversity of approaches may help to improve beginner student attitudes towards programming (Lusa Krug, Bowman, Barnett, Pollock, & Shepherd, 2021; Wanzer et al., 2020).

Sonic Pi is one such domain-specific programming platform that enables beginners to program their own music. It has been designed to be used at a school level and is considered appropriate for middle schoolers as an introduction to programming (Aaron, 2016; Aaron & Blackwell, 2013). Alternative domain-specific music coding platforms appropriate for beginners at a middle school level include EarSketch and TunePad, which have had academic research conducted on them (McKlin et al., 2021; Wanzer et al., 2020). However, studies on Sonic Pi to date have few specific details as to why students reacted positively and the extent of their effect sizes (Burnard et al., 2016; Cheng, 2018; Köppe, 2020; Lusa Krug et al., 2021). The aim of this research was to investigate this lack of detail by answering the research question: *Can composing music with the Sonic Pi platform help to promote positive attitudes towards programming?* This question is significant to the design of educational software because the use of Sonic Pi for a school context and its characteristics in relation to other music coding platforms are not considered in the literature on how they affect student attitude towards programming.

The research presented in this paper was part of a Master's Education Thesis (Petrie, 2019). It presents findings from a mixed-methods case study of a six-part-lesson unit of work (a group of lessons focused on a topic or theme) designed and conducted by the researcher for a class of middle school students over three weeks in school time. Results advance our understanding of the literature to date about the potential increases in attitude with the Sonic Pi platform for beginner programmers. As a secondary contribution, qualitative themes explain reasons why these increases may have occurred in relation to other major music coding platforms appropriate for middle school students (EarSketch and TunePad).

2. Background

2.1. The importance of attitude

Research across educational contexts universally agree that students' attitudes are a main factor that affect sustained motivation and learning efficacy (Fishbein & Ajzen, 1975; Cheah, 2020; Teo, 2007). More specifically, attitude towards computer usage is a significant contributing factor in the extent students utilise them for learning (Cazan, Cocoradă, & Maican, 2016; Teo, 2006, 2007). Fishbein and Ajzen (1975) defined attitude as positive or negative feelings towards behaviours that could affect aspects contributing to the success of a behaviour's aim. A positive attitude may be especially critical around the ages of 11 and 12 in school because these years have been seen to significantly influence students' later enrolment choices in senior high school and university (Grover et al., 2016; Hava & Koyunlu Ünlü, 2021; Margolis & Fisher, 2002; Wanzer et al., 2020). With efforts to include programming skills into curricula across the world (Grover et al., 2016; Hava & Koyunlu Ünlü, 2021; Lye & Koh, 2014), how to foster positive students' attitude towards it has become important for educators to understand because many studies found it is perceived as a difficult and an unappealing skill to learn (Anderson et al., 2008; Cheah, 2020; Kong, Chiu, & Lai, 2018; Margolis & Fisher, 2002).

2.2. Domain-specific programming platforms that combine music and programming for beginners at a school level

Domain-specific programming platforms have shown to be potentially motivating for beginners and foster positive attitudes because they enable an easier way to combine a domain they are already interested in (like music) with programming (Aghaee et al., 2015; Aaron & Blackwell, 2013; Bell & Bell, 2018; Köppe, 2020; Wanzer et al., 2020). A positive attitude towards an educational endeavour is highly related to developing intrinsic motivation to learn (Fishbein & Ajzen, 1975; Herman, 2012). Specifically linking programming with music composition has already shown to promote positive attitudes towards programming at a school level, which may help to alleviate the problem of beginner students viewing it as difficult (Burnard et al., 2016; Cheng, 2018; McKlin et al., 2018; Köppe, 2020; Wanzer et al., 2020).

The major domain-specific programming languages and platforms that combine music and programming targeting beginners at a school level are: EarSketch (https://earsketch.gatech.edu/landing/#/), TunePad (https://tunepad.live/), and Sonic Pi (https://sonic-pi.net/). These three platforms have been designed as appropriate for an introduction to programming to students at a middle school level (Aaron & Blackwell, 2013; Engelman et al., 2017; Gorson, Patel, Beheshti, Magerko, & Horn, 2017). See Table 1 for the major

Table 1
Domain-specific programming platforms that combine music and programming targeting beginners at a school level.

	Sonic Pi	EarSketch	TunePad
Programming language	Customized Ruby code	Python, JavaScript, Drag & drop code blocks (Scratch-like)	Python
Design	Both bottom-up & top-down	Top-down (remixing short pre-existing audio loops)	Bottom-up platform (references music theory concepts for trained musicians)
Interface[a]	No digital audio workstation interface (e.g. timeline or sequencer)	Multiple digital audio workstation interface features (e.g. sequencer and timeline)	Multiple digital audio workstation interface features (e.g. sequencer and timeline)

[a] See link in Appendix A for a screenshot of each platform.

similarities and differences between these platforms.

Both EarSketch and TunePad are considered part of an "eco-system of music + coding learning environments" (Introduction to TunePad, n. d.); for example, users can sign in to TunePad with their EarSketch account. The design of TunePad has drawn inspiration from Logo, Scratch, Sonic Pi, Garageband, and EarSketch (Horn et al., 2020, p. 1239), and "was created by the Tangible Interaction Design and Learning (TIDAL) Lab at Northwestern University in collaboration with the EarSketch team at the Georgia Institute of Technology" (TunePad About Us, n. d.). McKlin et al. (2021) have recently argued that the design differences between the two platforms are: TunePad is considered a bottom-up platform because it "references the music theory concepts already in widespread use in K-12 bands, orchestras, and choirs" (p. 929); while EarSketch is considered a top-down platform because it is "akin to remixing" where "students can write code to arrange short, pre-existing audio loops and effects on a multi-track timeline to create a complete song" (p. 929). It is important to note that EarSketch offers users to switch to coding in JavaScript or in a visual drag and drop block mode, whereas TunePad only offers users to code by typing text with Python syntax. McKlin et al. (2021) found that there was higher situational interest by musicians for the TunePad platform because of its more traditional (bottom-up) approach; whereas EarSketch has an easier path to creating full songs for non-musicians (top-down), which was found to result in higher maintained situational interest in TunePad.

Sonic Pi's design differs significantly from both EarSketch and TunePad because the default interface does not offer any visual audio and sequence representations as multi-track timelines of the code that would normally be found in traditional Digital Audio Workstations (DAWs). Sonic Pi also enables the flexibility to remix sequences and samples (like in EarSketch) as well as make music in a traditional way on a note-by-note basis (like in TunePad), which indicates it can be considered both top-down and bottom-up in relation to McKlin et al. (2021) comparison. Moreover, while Sonic Pi uses Ruby as a programming language, its syntax is less like traditional Ruby code to make it easier for beginners to make music (Aaron & Blackwell, 2013); for example, it has a unique syntax to stack the sound of multiple loops at once through the "live_loop" command. In contrast, EarSketch and TunePad's code is similar to traditional Python syntax. Between the three platforms, Sonic Pi perhaps offers the simplest way for beginners to get started making music immediately because these novel commands for coding music have been highly customized.

2.2.1. *Live-coding in Sonic Pi*

Sonic Pi particularly promotes the practice of live-coding where students 'perform' their programming and music skills by projecting code on a screen for an audience to watch (Aaron, 2016), which has been described as a virtuosic activity by Nilson (2007). However, negative reports from beginner programmers were voiced in the Burnard et al. (2016) study when live coding in front of peers, which have been reflected in EarSketch by Freeman and Magerko (2016). As a result, music composition – where students prepare code instead of performing it – is more appropriate for beginners at a school level and the aims of the present study.

2.2.2. *Other programming platforms that combine music and programming*

There are many other domain-specific programming platforms available today that combine music and programming (e.g. JythonMusic, SuperCollider, and more), however, they are often appropriate for advanced students generally at a university level. For example, JythonMusic requires advanced programming knowledge to get started, therefore, it is not considered appropriate as an introduction to programming for beginners at a middle school level (Lusa Krug et al., 2021; Manaris, Stevens, & Brown, 2016).

Additionally, while this study focuses on domain-specific programming languages with music and programming, it is also worth mentioning here that Scratch (https://scratch.mit.edu/) is a popular visual drag and drop based platform for general programming at a school level, which has also been used to code music (Bell & Bell, 2018; Brown & Ruthmann, 2020). However, Scratch has been considered challenging and restrictive to make music on for children (Payne & Ruthmann, 2019).

2.3. *Attitude studies on music and programming*

Prior to 2021, published studies on Sonic Pi attitude changes have largely gathered qualitative post intervention data from short semi-structured interviews with students, which indicated their attitudes toward programming were largely positive with few details why (Burnard et al., 2016; Cheng, 2018; Köppe, 2020). Luca Krug et al. (2021) have recently conducted a study on Sonic Pi with 31 participants ($n = 31$), which indicated statistically significant results for more positive attitudes after exposure to their virtual camp called Code Beats (https://codebeats.weebly.com/). However, there was limited qualitative data reported which only indicated favourable feedback about the Code Beats virtual camp and their intention to consider pursuing computer science classes in the future.

There have been multiple large scale quantitative studies on student attitude changes towards programming with EarSketch, which have indicated statistically significant increases in multiple contexts with medium or large effect sizes (Engelman et al., 2017; Freeman & Magerko, 2016; Freeman et al., 2019; McKlin et al., 2018; Wanzer et al., 2020). Similar quantitative increases and reports have been found in the studies on TunePad by McKlin et al. (2021), Horn et al. (2020), and Gorson, Patel, Beheshti, Magerko, and Horn. (2017). Like literature on Sonic Pi, limited qualitative analysis has been conducted that reveal some understanding of what aspects about coding music and their approach increased different attitude constructs for students. Moreover, except for the studies by Burnard et al. (2016) in Sonic Pi and some EarSketch studies reported in Wanzer et al. (2020), all published research has been conducted for out of school contexts. The participants in these studies are opting to participate in their own time, making them much more personally invested. The articles by Burnard et al. (2016) and Wanzer et al. (2020) also offer a paucity of details on how to implement these platforms in school time. Thus, more studies are needed that are conducted in school time at regular classrooms to understand how domain-specific programming languages can be implemented in traditional education settings.

3. Methods

3.1. Research model and procedure

A mixed-method case study using an experimental unit of work designed by the researcher was used to frame the methodology of this research to answer the question: *Can composing music with the Sonic Pi platform help to promote positive attitudes towards programming?* A key advantage of case studies is they can illuminate a detailed snapshot of reality (Cohen, Manion, & Morrison, 2011). A case study is an appropriate methodology for this research because it closely investigated a unit of work in a school classroom. It is hypothesised that all attitude subscales would increase (as an alternative hypothesis) because all related literature had indicated positive changes to student attitude after being exposed to coding music.

A six-part unit of work (a group of lessons focused on a topic or theme) was developed and designed for a class of 22 beginner programmers at 11 and 12 years of age (see link in Appendix A to view the unit of work plan). With professional experience as a music and computer science teacher, a sole participant researcher conducted this study who also developed the unit of work, delivered it to the participant students, and gathered the data. The unit of work focused on students creating a music composition individually. It was designed to be implemented in class time during a regular school term. The intention of this approach was so that the study could inform the use of Sonic Pi for middle school contexts.

A project brief was given to students where they were asked to create their music compositions of 1–3 min in length. See the unit plan link in Appendix A for details on the programming syntax and music concepts covered in each lesson. Specific learning outcomes in lessons 1–4 aimed to help students build up their music compositions in the following main ways:

- Lesson 1: intended students to create their own melody and get familiar with Sonic Pi
- Lesson 2: asked students to create their own collection of notes stored in a list
- Lesson 3: asked students to create their own simple musical algorithms
- Lesson 4: asked students to create their own electronic synthesiser from basic wave forms (e.g. sine, triangle etc.)
- Lesson 5: was dedicated to bringing these concepts together in a final project for a lesson focused on presenting everyone's work in Lesson 6 (see link in Appendix A to view student exemplars).

As recommended by Lye and Koh (2014), throughout the lessons, the participant researcher endeavoured to adopt a constructionism approach where they aimed to guide students thinking with questions instead of providing direct answers, for example: "Do you know where the bug is?", "When did it last work?", "What is the loudest and softest sound?". However, Ames (2018) and Rubin (2013) argue that constructionism needs to be effectively scaffolded particularly for beginners. Consequently, each new music and programming concept was introduced by the participant researcher at the beginning of the lesson with concrete examples that students could experiment with.

3.2. Instrument used and their validation

3.2.1. Pre and post unit of work questionnaires

Pre and post unit of work questionnaires were designed for all participants and had three main sections.

(1) The first section's purpose was to measure quantitative changes in attitudes towards music composition and programming.
(2) The second section helped to assess changes in perspectives on music and programming with qualitative responses. It had two items that asked all students what their feelings were towards programming and music composition to help record any qualitative changes in their perspectives.
(3) The last section (pre-questionnaire only) asked their name and gathered information about prior experiences in music, programming, and combining the two into one activity. This section intended to help provide comparisons between the participants who were new or had prior experience.

Both questionnaires can be viewed with the links provided in Appendix A.

First section to quantitively measure attitudes.

A survey by Teo (2007) was adapted for the purpose of this study, which was originally developed using a version of the established Computer Attitude Questionnaire (CAQ) by Knezek and Christensen (1996) that had the three subscale constructs of enjoyment, importance, and anxiety. Studies on coding music with EarSketch have also adapted the survey developed by Knezek and Christensen (1996) for their analysis as described by Engelman et al. (2017). These EarSketch studies target students' intentions to persist in computing (rather than attitude towards programming specifically) and did not manage to validate their instrument with statistical significance (Engelman et al., 2017).

The three subscales of enjoyment, importance, and anxiety are defined by Teo (2007) as follows: enjoyment means "enjoyment or liking" (p. 128) or the degree to which the student likes to work on computers; importance is "perceived usefulness" (p. 127) or the perceived need for computers in the present and future; anxiety as "students' confidence" (p. 128) or the apprehension students feel towards activities that involve computers. These three subscales strongly link with Fishbein and Ajzen's (1975) definition of attitude (see Section 2.1) as consisting of three elements: affective (associated with the subscale of enjoyment), cognitive (associated with the subscale of importance), and conative (associated with the subscale of anxiety) (Teo, 2007). Teo's (2007) reliability estimates found these three subscales had internal consistency and acceptable intercorrelations. They have also been successfully used to measure computer attitudes effectively for middle school students ($n = 240$) in South Africa (Bovée, Voogt, & Meelissen, 2007). For these reasons, these attitude subscales were considered suitable as a framework to measure pre and post differences in programming and music for the purposes of this study.

The research question of this study asked to measure attitudes toward programming only. However, attitudes to music composition were also measured in this section to provide a comparison of similarities and differences with programming. This additional data helped to compare the extent of the impact this unit of work had on attitudes.

Questionnaire items to measure attitude.

To ensure the reliability of the original instrument was maintained by Teo (2007), each question was modified minimally from the original questionnaire. For example, "I enjoy doing things on the computer" (Teo, 2007, p. 132) was modified to "I think programming or coding is a lot of fun". Three items were included for each subscale from the original questionnaire to be mindful of the attention span of middle school students to concentrate. It was considered appropriate to merge some questions into one item: for example, "I enjoy computer games very much" and "I enjoy doing things on the computer" (p. 128) was merged to form the item "I think programming (or coding) is a lot of fun". For all music items, the text "programming or coding" was replaced with "music composition" to measure student attitudes toward music composition.

The order format of the questions was not modified from the original questionnaire, where participants are asked about each subscale in the order of enjoyment, importance, and anxiety. As in the original questionnaire for each subscale, there were positively worded items and negative in the middle to reduce acquiescent and extreme response bias (Cohen et al., 2011). However, there is also evidence that reverse items can distort or be misinterpreted by participants (Weijters, Baumgartner, & Schillewaet, 2013), which is a potential limitation of this strategy.

As six-point Likert scales have shown to be more reliable than 5-point scales in psychology tests (Chomeya, 2010), the Likert scale used for this study was also modified from the original Teo (2007) to a six-point Likert scale (from a five-point scale) and given numerical values for statistical analysis as follows: $3 =$ Strongly agree; $2 =$ Moderately agree; $1 =$ Slightly agree; $-1 =$ Slightly disagree; $-2 =$ Moderately disagree; $-3 =$ Strongly disagree. Six-point Likert scales also have the advantage of forcing the participant to decide if their attitude towards a given question is positive or negative (Cohen et al., 2011).

3.2.2. Pre and post interviews (see link in Appendix A for interview guides)

All student participants were invited for a pre and post semi-structured interview to provide detailed descriptions of participant experiences, which was intended to help explain the quantitative questionnaire results. The pre and post interviews specifically asked these students questions on the three attitude subscales of enjoyment, importance, and anxiety. The selection of students to interview were based on those that agreed to participate with parental permission and equal gender representation. These interviewed students were given the pseudonyms: Sarah, Sam, Norman, Laura, and Ben.

3.3. Research context and sample

The sample of participants for this case study were a class of student volunteers with 22 students 11 or 12 years of age (10 female and 12 male). Increasing the number of participants to make the results more significant was not possible without including more researchers (because of a lack of capacity), however, it is intended that this single case study can help to inform subsequent studies that can be larger in scope. The research site was a well-established urban state funded school in New Zealand located in an average socio-economic area. The participants were selected based on their location in relation to the researcher and because of their high interest in participating in the research.

Most students were from a New Zealand European descent (15/22). Other ethnicities included two Indian, two Chinese, one German, and two Samoan. As informed by the school, no student had special needs and significant language difficulties for their age.

Table 2 shows prior experience of students and indicates that many participants were not entirely new to programming, music, or both as a separate activity. It was outside the scope of this study to investigate the nature and rigour of these prior experiences. However, every student was new to combining the two skills into one activity, which still gives perspective on answering the research question posed in this study. The unit of work was not modified or adapted to the students after assessing the prior experience of students in the pre unit of work questionnaire or during its delivery.

Table 2
Prior experience of students in music and programming.

Prior experience	Male	Female	Total
Prior experience in combining programming and music into one activity	0	0	0/22
Prior programming experience	4	2	6/22
Specialist music training (with no prior programming experience	1	3	4/22
Prior experiences in both programming and in music as separate activities	2	2	4/22
Completely new to both programming and music	5	3	8/22
Interviewed students ($n = 5$) who had some programming experience	2	1	3/5
Interviewed students ($n = 5$) who had some specialist music training	1	1	2/5

3.4. Data analysis

The mixed-method approach used in this study provides a high degree of validity and reliability through the merging and triangulation of qualitative and quantitative data (Creswell, 2014). Each measure was first analysed separately then merged as part of a convergent parallel mixed-methods analysis approach (ibid.). This systematic merging of data compares all results with a side-by- side comparison to see if they confirmed or disconfirmed each other. Some sub-groups from the sample of total student participants were analysed with gender and prior experiences (programming, music, and coding music) only to give very cautionary findings because subgroup sample sizes ($n < 12$) are too small (see link in Appendix B).

Quantitative analysis. Quantitative data was gathered from the pre and post questionnaire items (18 total) (see Section 3.3). Each questionnaire item in this section was assigned a numerical value from negative 3 to positive 3 for each option on the Likert scale. For each subscale, the three associated questionnaire items were added together and divided by three to create an average score for each student. Thus, each student had a mean attitude score for each subscale (enjoyment, importance, and anxiety) for both pre and post questionnaires in music and programming. A dependent samples *t*-test and Cohen's *d* estimates were calculated on each subscale (enjoyment, importance, and anxiety) for programming and music.

Qualitative analysis. As recommended by Hsieh and Shannon (2005), a directed content analysis approach was employed for all qualitative data. Using this approach, all data was categorized according to the predetermined coding schemes of the framework employed in this study for attitude by Teo (2007) (enjoyment, importance, and anxiety). Once all raw qualitative data had been categorized, thematic analysis was used to identify and describe common patterns across the data under each code as themes (Cohen et al., 2011). During this process, constant comparison with the raw data was employed so that the original context of each measure was not lost. The qualitative themes drawn using this approach were then compared with the quantitative data in accordance with the convergent parallel mixed-method design (Creswell, 2014).

Ethics. The Educational Research Human Ethics Committee's principles and guidelines at the University of Canterbury with adolescents was granted before this study commenced. All participants (students, parents, and school leadership) gave informed and voluntary consent before participating. There were no participants who opted not to participate in the original class. Pseudonyms were used for all participants and the participating school so their privacy and confidentiality can be adhered to.

4. Results

4.1. Pre and post questionnaire attitude items

Pre (T1) and post (T2) questionnaires were taken by all student participants to help measure attitude changes in programming and music composition. Based on Cohen's (1992) guidelines (strong >0.65; moderate between 0.35 and 0.65; and weak between 0.2 and 0.35), Table 3 indicate *t*-test and Cohen's *d* values for programming and music have strong effect sizes with minimal *p*-values of $< .001$. All programming attitude subscales indicate higher *t*-values than music, suggesting the unit of work had a more significant difference in all attitude subscales for programming in comparison to all attitude subscales for music. The greatest positive increase are the attitude programming subscales for importance ($t = 12.28$, Cohen's $d = 1.72$, and $p < .001$) and anxiety ($t = 8.71$, Cohen's $d = 1.54$, and $p < .001$). For the anxiety subscale, note that higher numbers indicate an anxiety *decrease*.

Notably, the smallest Cohen's *d* effect size from T1 and T2 is the music anxiety attitude subscale ($d = 0.70$). Students may have had lower levels of anxiety pre unit of work in music because they're likely exposed to more music activities than programming in their daily lives. This explanation is confirmed with positive T1 responses for music (M = 0.39) in comparison to programming (M = −0.98) for the anxiety attitude subscale (1.37 difference).

Strong correlations are indicated when comparing all T2 subscales between programming and music (enjoyment r = 0.82, importance r = .68, anxiety r = 0.83). According to Dowdy, Wearden, and Chilko (2004), a risk of multicollinearity can occur when there is a linear relationship among two or more independent variables with a variance inflation factor (VIF) above 5. Tests indicated a low level of multicollinearity for all attitude subscale correlations (enjoyment VIF = 3.28, importance VIF = 1.72, and anxiety VIF = 1.89). As a result, these strong correlations suggest students' attitudes may have increased because of exposure to music composition, exposure to programming, or the Sonic Pi platform.

Subgroup analysis: Quantitative subgroup findings can be seen with the link Appendix B. However, these small sample sizes ($n < 12$) means these sub-group findings need to be treated with caution.

In summary, males ($n = 12$) had larger differences between means from T1 and T2 when compared with females ($n = 10$) on all three subscales. For all subscales, males had a higher mean score in T2. Except for the anxiety subscale, males started with higher

Table 3
T-test, p-value, and Cohen's d calculations for programming and music attitude subscales (T1 and T2 questionnaire; $n = 22$).

	Programming enjoyment	Music composition enjoyment	Programming importance	Music composition importance	Programming anxiety	Music composition anxiety
t-value	8.39	6.07	12.28	6.49	8.71	7.21
Cohen's *d*	1.31	1.39	1.72	1.48	1.54	.70
p-value	$p < .001$	$p < .001$	$p < .001$	$p < .001$	$p < .001$	$p < .001$

means in the enjoyment and importance subscales when compared with females. However, all qualitative data did not reveal any obvious or explicit differences between males and females on all subscales.

The students with prior experience only in programming ($n = 6$) and both programming and music ($n = 4$) had the highest averages for all attitude subscales in T1 and T2. Those students without prior experiences in either music or programming ($n = 8$) had the lowest averages for both subjects. These findings reflect that prior programming experience positively affects attitudes more than those without. Apprehension about learning programming or trying something new may be the reason why students without prior experience in programming had the lowest averages. Nevertheless, all subgroups saw a positive mean increase from T1 to T2 on all subscales. Some qualitative differences were detected between those with prior experience in programming, which are reported under the following sections.

4.2. *Enjoyment*

Fig. 1 provides a box and whisker chart as a visual representation for the enjoyment subscale in programming. This chart indicates the interquartile and upper whisker increased to above zero in T2, which suggests positive attitudes for 75% of students. The mean also confirmed an increase from T1 ($M = -0.45$) to T2 ($M = 1.08$) by 1.53. Standard deviations remained comparatively similar T1 ($SD = 1.35$) to T2 ($SD = 1.21$). With the high t-test value and Cohen's d effect size ($t = 8.39$, Cohen's $d = 1.31$, and $p < .001$), these results strongly suggest the unit of work increased the enjoyment subscale for most students in programming significantly.

Qualitative data. Pre and post questionnaire data revealed students' enjoyment of programming in response to the short answer question "Write a few short sentences about your feelings towards learning more programming or coding". These responses were evaluated as either positive or negative: 14 of 22 T1 responses were evaluated as positive (63.64%), which increased to 20/22 positive responses in T2 (90.91%). Three notable contrasting changes from T1 to T2 in this short answer question were found from Jacob, Charlotte, and William as indicated in Table 4.

These three responses in Table 4 reinforce the quantitative results that significant positive changes occurred in students' perception of programming before and after the unit of work. There was a lack of negative qualitative pre-questionnaire data from those with prior experience in programming, which reinforces the higher quantitative averages reported in Appendix B. However, there were only two negative responses from T2 by students with no prior experience in programming: "it did get a bit hard" (Sophia, T2) and Ava "confusing" (Ava, T2), suggesting the difficulty was the most significant barrier to enjoyment to programming for Ava and Sophia. Nevertheless, the majority of data confirms that most students enjoyed programming by the end. A summary of the responses can be seen in Table 5.

These responses helped to identify the dominant theme that emerged from the qualitative data: *novelty of making music with code*. This theme helps explain why students enjoyed the unit of work, which was a new experience for all students.

4.3. *Importance*

Fig. 2 box and whisker chart indicate the interquartile and whisker increased to above zero in T2 for the programming importance subscale. However, three outlier cases (Grace, Sophia, and Ava) indicated below zero results in T2 who had no prior experience in programming or music. Comparing these individual three outlier cases with T1 results indicate a positive or negative average change within a range of 1, suggesting these students had slightly more negative attitudes in their perceived importance of programming after the unit of work. Nevertheless, mean calculations for all students indicated an increase of 1.88 (from T1 $M = -0.65$ to T2 $M = 1.23$). Standard deviation results were similar but became narrower from T1 ($SD = 1.23$) to T2 ($SD = 0.95$). With supportive evidence from the high t-test value and Cohen's d effect size ($t = 12.28$, Cohen's $d = 1.72$, and $p < .001$), this quantitative analysis collectively suggests the unit of work increased the importance subscale for programming significantly for most students.

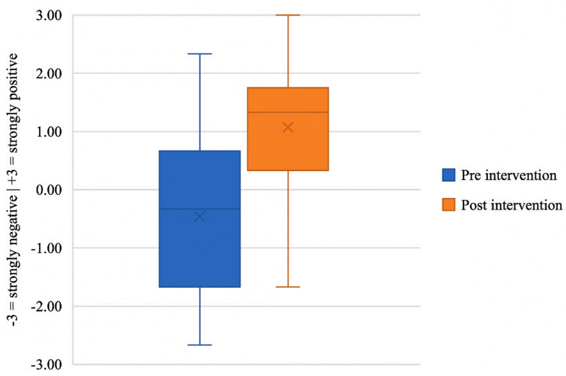

Fig. 1. Box and whisker plot chart for the enjoyment subscale in programming.

Table 4
Notable contrasting changes from T1 to T2 in this short answer question "Write a few short sentences about your feelings towards learning more programming or coding".

	T1	T2
Jacob	"boring"	"coding is really fun and interesting, I liked working with my friend on cool sounds"
Charlotte	"not sure sorry"	"I feel quite good, it was quite fun thanks!"
William	"OK"	"coding can be really cool, I feel really good"

Table 5
Summary of the responses to "Write a few short sentences about your feelings towards learning more programming or coding".

Theme	Number of students	Example quote
Novelty of coding music	8	"I really like making my own music and it was cool to learn to make stuff with Sonic Pi" (Laura, T2 interview)
Enjoyed coding music	5	"i was surprised how fun it was. It is re [al]ly cool coding music!!!!! i re [al]ly liked learning about it" (Ben, T2)
Liked the creative aspect of programming	4	"I liked working on making and creating stuff' (Charlotte, T2)
Incomprehensible responses	4	–

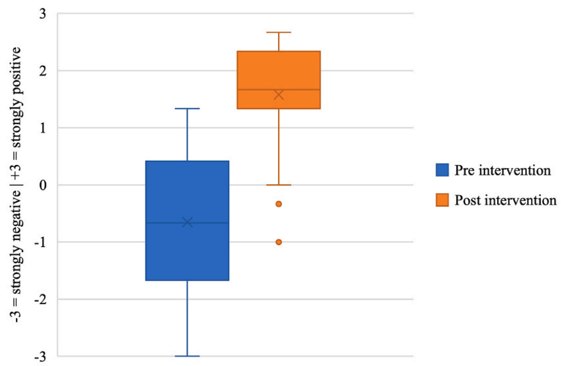

Fig. 2. Box and whisker plot chart for the importance attitude subscale for programming.

4.3.1. Qualitative data

Programming as an essential literacy: The idea that programming is seen as an essential literacy emerged as the only common theme in students' reflections and post unit of work student interviews for this subscale. For example, responses from three students with prior programming experience:

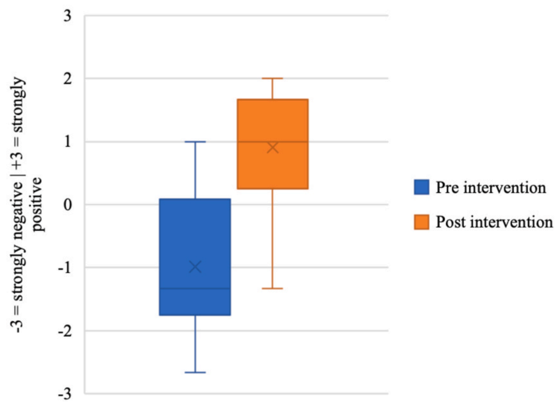

Fig. 3. Box and whisker chart for the anxiety subscale of programming.

- Norman thought he would study programming in high school and or university because "you could get a job that pays lots of money" (Norman, T2 interview).
- Laura perceived that programming will be an important skill to learn in the future because "we all use technology a lot" (Laura, T2 interview);
- Ben thought programming as valuable because "I think maybe everything is going to get taken over by robots, so I'll want to program robots to do things for us." (Ben, T2 interview).

Six students who had no prior programming experience also confirmed this theme; for example, Ava said "I think we will need this skill in the future" (Ava, T2, questionnaire). Thus, programming was commonly reported as a valuable skill to learn by many participants regardless of their prior experience in this skill. It was unclear whether this theme emerged because of students' exposure to Sonic Pi specifically or the unit of work.

4.4. Anxiety

Fig. 3 box and whisker chart provides a visual representation that the interquartile and upper whisker had increased to above zero in the T2 questionnaire for the anxiety programming subscale (strongly positive equates to low levels of anxiety and strongly negative equates to high levels of anxiety). A mean increase of 1.89 was indicated between the T1 mean ($M = -0.98$) and T2 mean ($M = 0.91$). Standard deviation remained comparatively similar between T1 ($SD = 1.09$) and T2 ($SD = 0.89$). With the high t-test value and Cohen's d effect size for this subscale ($t = 8.71$, Cohen's $d = 1.54$, and $p < .001$), all quantitative evidence strongly suggests the unit of work decreased anxiety in programming significantly for the student participants.

Qualitative data. The ten students reporting prior programming experience (including those with both music and programming) reinforced a qualitative relationship to lower anxiety levels as indicated the quantitative data reported in Appendix B. For example, Ben wrote, "I am excited to learn more about coding because I do a lot of it.", and Norman "I've done a little bit of it before so I think I know that I might enjoy it". Conversely, those students who had no prior programming experience also reinforced the quantitative data with higher anxiety levels than those with prior programming experience by describing their unfamiliarity towards programming in T1. For example, Michael wrote: "I've never done it so dunno about it".

However, the post unit of work reflections indicated most students had become more confident programmers. For example, Sarah (with no prior programming experience) said, "I think because I feel better about it like I know more now" (Sarah, T2 interview). Thus, while students who were inexperienced in programming first indicated a relatively high T1 anxiety, the majority indicated a decrease in this subscale by the end of the unit of work. Thus, these findings indicate that students with prior programming experience had much less anxiety than those without initially, however, all students were able to significantly reduce their anxiety with exposure to Sonic Pi by the end of the unit of work.

Programming perceived as difficult. A notable qualitative theme that emerged on students' anxiety was the perceived difficulty of programming, which was still the case after post unit of work:

- "Coding with Sonic Pi was quite fun, but it did get a bit hard" (Emma, T2)
- "feeling good, it was easy but some parts were quite hard" (Sophia, T2)
- "I liked it, but it was hard tho" (Daniel, T2).

Quantitative data reflected this theme from all experience subgroups. For example, the idea of programming as a difficult skill to learn is also reinforced in post-unit of work interviews: Norman with prior programming experience thought learning how to program in high school or university is more difficult than other subjects because "you need to be really good at computers, like know so much stuff about them" (Norman, T2 interview) and Laura also with prior programming experience thought high school or university programming classes are "really hard [because] I think it seems really hard. You have to be good at typing a lot and solving things that are hard." (Laura, T2 interview). While the quantitative results indicated a decrease in anxiety, these reflections suggest students' perception of programming as a difficult skill to learn may have contributed to their sense of anxiety. Notably, no instances were recorded of students who thought programming is easy or comparatively difficult to other subjects.

5. Discussion

In answer the research question: '*Can composing music with the Sonic Pi platform help to promote positive attitudes towards programming?*', results confirm an alternative hypothesis in all attitude subscales significantly (enjoyment, importance, and anxiety). Thus, Sonic Pi may be an effective way to introduce programming to students around 11 or 12 years of age. While these findings are not generalisable to other contexts due to its limited scope, the study's aim was achieved to provide academic research in a detailed mixed-method case-study in a school context on Sonic Pi, which was previously lacking. It also reinforces findings in prior studies on Sonic Pi (Brunard et al., 2016; Cheng, 2018; Köppe, 2020; Lusa Krug et al., 2021) EarSketch (Siva, Im, McKlin, & Freeman, 2018; Wanzer et al., 2020) and TunePad (Horn et al., 2020; McKlin et al., 2021) that teaching domain-specific programming languages which combine music and programming can foster positive attitudes.

Interesting differences were found between subgroups, which all need to be treated with caution because of the small sample size. While both males and females saw increases in attitude from T1 to T2, males generally began with more positive attitudes in T1. Additionally, the unit of work had a greater positive affect on males in T2 for programming. These gender differences are commonly reflected in the literature on attitude towards programming across platforms and languages (Anderson et al., 2008; Günbatar, 2020; LaBouliere, Pelloth, Lu, & Ng, 2015; Vekiri, 2010), which implies coding music is no different. Those with prior experiences in programming had the highest average increase from T1 to T2 in all subscales (discussed more under subscale themes in the next paragraph). Nevertheless, all subgroup results in this study are encouraging because they all indicated quantitative increases from T1 to T2. More targeted research with a higher sample size is needed to investigate if these differences between genders and prior experience persist.

Qualitative subscale themes. The qualitative theme of *novelty of making music with code* for the enjoyment subscale has been similarly reinforced by prior studies on EarSketch (Wanzer et al., 2020) and TunePad (McKlin et al., 2021). Under the importance subscale, the only theme that emerged from qualitative data yielded that some students perceive programming skills important because it was viewed as an essential literacy. However, as students were not exposed to general information about how technology is used in society in the unit of work, this theme does not explain why the importance subscale increased so significantly. The major qualitative theme for the anxiety subscale revealed that many still perceived programming as a difficult skill to learn after the unit of work. This theme is consistent across the reviewed literature on attitudes toward programming in general (Anderson et al., 2008; Başer, 2013; LaBouliere et al., 2015; Margolis & Fisher, 2002; Mason & Rich, 2020). Thus, the qualitative data failed to explain if and how the large quantitative differences in the importance and anxiety subscales specifically attribute to coding music, the unit of work, or Sonic Pi. As all participants were new to combining music and programming into one activity, Sonic Pi helped students see new ways on how it can be utilised creatively in different domains as indicated by the theme from the enjoyment subscale *novelty of making music with Sonic Pi*. Thus, it is possible that this theme could help explain the particularly large effect sizes in the importance and anxiety subscales by influencing students' perception on the role programming might play in their future (importance) and how anxious they feel about programming (anxiety) through the enjoyment of coding music.

However, it is important to be critical without more specific themes on coding music because many studies outside of programming education have discovered increased exposure to general computer use can also promote positive attitudes (Ames, 2018; Barron, Walter, Martin, & Schatz, 2010; Bovée et al., 2007; Cazan et al., 2016; Teo, 2008). Moreover, the reviewed literature with case studies using a variety of other programming platforms and tasks also unanimously support the notion that students react positively when exposed to programming (Allsop, 2018; Brennan & Resnick, 2012; Davies et al., 2013; Günbatar, 2020; Kong et al., 2018; Vekiri, 2010). Therefore, it is difficult to specifically attribute the large increases in attitudes found in this study with the activity of music composition using Sonic Pi and the unit of work's design from the qualitative themes gathered. Unfortunately, all published case studies on Sonic Pi (Brunard et al., 2016; Cheng, 2018; Lusa Krug et al., 2021; Köppe, 2020), EarSketch (Wanzer et al., 2020), and TunePad (Horn et al., 2020; McKlin et al., 2021) report few qualitative explanations about reasons why increases occurred for comparisons to be made, which as mentioned, is a weakness in the literature to understand these significant attitude changes. Thus, the qualitative methods to inform why these increases can occur need to be strengthened because it is not clear if they are because of: (a) domain-specific programming languages in general; (b) the specific combination of music and programming; (c) Sonic Pi; or (d) exposure to programming in general.

Aghaee et al. (2015) stated there are three major factors that affect motivation when learning programming: bricoleurism (ability to learn by improvising and a kind of informal education), technophilia (curiosity about new technologies), and artistry (ability to display creativity and originality). These three characteristics could help explain why programming attitudes were more positive in the present study because they are embedded into the design and intended use of Sonic Pi, EarSketch, and TunePad as a natural consequence of combining music and programming. A positive attitude towards an educational endeavour contributes to motivation (Fishbein & Ajzen, 1975; Herman, 2012). Thus, these three factors could provide a guide for future studies in the gathering of more successful qualitative data.

Design of domain-specific music programming languages at a school level. As mentioned in Section 2.2, McKlin et al. (2021) considers that TunePad is a bottom-up, note-based design suiting students who have prior experience in music because it is "mimicking familiar music paradigms" (p. 933). EarSketch in contrast, is considered a "top-down, audio-based remixing approach" by McKlin et al. (2021, p. 933) because this approach makes creating full songs easier for beginners with pre-made sequences. Because the design of Sonic Pi offers almost no familiar visualizations like those in EarSketch and TunePad, Sonic Pi is perhaps even more 'bottom-up' than TunePad, with its interface more focused on coding in its default view. The findings in the present study reinforce this comparison because students who indicated prior experience in programming had more positive attitudes in all subscales when compared to those students without. However, this explanation is countered with no significant differences found between the EarSketch and TunePad platforms in the comparative study by McKlin et al. (2021). One possible advantage of Sonic Pi's relative simplicity is its interface means students are forced to make music algorithmically without visualisations of what the code is representing. This design approach may be a reason why Sonic Pi could suit students with more prior experience and interest in programming. McKlin et al. (2021) also suggested that designers of educational software should carefully consider prior experiences of students and tailor learning with this in mind. However, similar to the limitations in the present study, their findings are very fragile with a low sample size ($n = 26$) and sub-groups. Future research needs to consider the nature and rigour of these experiences as – except for the study McKlin et al. (2021) – are not reported in the reviewed literature on Sonic Pi, EarSketch, and TunePad.

Use of domain-specific music programming languages at a school level. It was found in Section 2.3 that only two case studies on EarSketch and Sonic Pi have been conducted in school contexts, which report a paucity of details of how they were implemented. As students with negative attitudes are not likely to volunteer to such programs, attitude data in these studies is likely skewed positively because their participants are sacrificing their own time (making them much more personally invested). The present study was implemented with a regular class of middle school students in school time. Thus, while the students still volunteered to participate, the alternative was that they would then complete regular schoolwork, which may or may not have been exciting to them. It is common in primary and middle school for one teacher to deliver most (if not all) the subjects that need to be taught to students. Thus, it would be ideal for the regular classroom teacher to plan a unit of work on Sonic Pi that links both with the music and computing curriculums. The main challenge with this is most middle school teachers will need professional development to deliver a unit of work in Sonic Pi. However, the novel characteristics of Sonic Pi's design, which uses traditional text-based programming with simple Ruby commands to easily start coding music, contributes to its accessibility for beginners (Aaron & Blackwell, 2013). Thus, it is suggested that the amount of professional development required perhaps would not need to be extensive as with EarSketch and TunePad. The benefits could mean they could not only foster positive attitudes towards programming easily, but to deliver parts of the computing and music related curriculums simultaneously, which could save time.

5.1. Limitations

The major limitation of this research that it was a relatively small single case-study, which is not large enough to make the findings generalisable to other contexts (Cohen et al., 2011). Similarly, results from the differences found between sub-groups (e.g. gender and prior experiences in music and programming) can only be cautious due to the small sample size. Additionally, the prior experiences of the participant students and the gathering of interview data may have skewed the results to be more positive. Nevertheless, all students in this study were new to combining music and programming into one activity.

6. Conclusions

This study sought to discover if Sonic Pi can foster positive attitudes towards programming. Results strongly confirm an alternative hypothesis that it can with *t*-test and Cohen's *d* values for both programming and music having strong effect sizes with minimal *p*-values of $< .001$. Qualitative themes under each attitude subscale were discovered indicating why changes may have occurred as follows: *novelty of making music with code* (enjoyment), *programming is perceived as an essential literacy* (importance), and *programming is perceived as difficult* (anxiety). However, like the reviewed literature to date on Sonic Pi, EarSketch, and TunePad, findings do not provide sufficient data to confirm what specific aspects of the increases may be because of domain-specific programming languages, the specific combination of music and programming, Sonic Pi, or exposure to programming in general. Thus, future studies need to give special attention to the gathering of more specific qualitative data, perhaps with the help of the framework by Aghaee et al. (2015) to guide questioning on the design and use of music coding platforms. Additionally, with focused methodological planning, it would be particularly beneficial to conduct more in-depth research on the extent gender and different prior experiences affect attitude. Along with the limitations considered in Section 5.1, future research could use a larger sample of beginner programmers using Sonic Pi, EarSketch, and TunePad, with control groups using general programming languages (like Scratch and Python) to understand attitude differences more clearly between coding music and general programming.

CRediT author statement

Christopher Petrie: Conceptualization, Methodology, Software, Formal analysis, Data curation, Writing- Original draft preparation, Visualization, Investigation, Software, Validation, Reviewing and Editing.

APPENDIX A

Links

Pre questionnaire for participant students:
https://docs.google.com/document/d/1ddLwljJspZKaIcZWMoYiJwn6FeSSviTKwSp3w9v5j1A/edit?usp=sharing.
Post questionnaire for participant students:
https://docs.google.com/document/d/1RXeU5EISr2v2Nn_odKuu-25LQb1hMXK7PUVWbBaNlEA/edit?usp=sharing.
Pre and post semi-structured interview guide for participant students:
https://docs.google.com/document/d/129C0EHJdX0-ik4aYveXIvd8Q4y2WZy–_8G9BvphTpU/edit?usp=sharing.
Unit-plan:
https://drive.google.com/file/d/1QQjuCgfGPvuQkB4XrK4Kavq-ImPcBHdE/view?usp=sharing.
Student exemplars:
https://drive.google.com/file/d/1XTj0kW2lhxbega8B0HMJ8Js2CQM065Yv/view?usp=sharing.
TunePad, EarSketch, & Sonic Pi interfaces:
https://drive.google.com/file/d/15susReW_oXKaer_IMDU3YZmD7BaS3SLL/view?usp=sharing.

APPENDIX B

Subgroup analysis of gender and prior experiences:
https://docs.google.com/document/d/143pZxeoYkyOzYJajvw5M3rC34pP_7g64nM4f1Ugs7wc/edit?usp=sharing.

References

Aaron, S. (2016). Sonic Pi – performance in education, technology and art. *International Journal of Performance Arts and Digital Media, 12*(2), 171–178. https://doi.org/10.1080/14794713.2016.1227593

Aaron, S., & Blackwell, A. F. (2013). From sonic Pi to overtone: Creative musical experiences with domain-specific and functional languages. *Proceedings of the First ACM SIGPLAN Workshop on Functional Art, Music, Modeling & Design - FARM, '13*, 35. https://doi.org/10.1145/2505341.2505346

Aghaee, S., Blackwell, A. F., Stillwell, D., & Kosinski, M. (2015). Personality and intrinsic motivational factors in end-user programming. In *2015 IEEE Symposium on visual languages and Human-Centric computing* (pp. 29–36). VL/HCC). https://doi.org/10.1109/VLHCC.2015.7357195.

Allsop, Y. (2018). Assessing computational thinking process using a multiple evaluation approach. *International Journal of Child-Computer Interaction, 19*, 30–55. https://doi.org/10.1016/j.ijcci.2018.10.004

Ames, M. (2018). Hackers, computers, and cooperation: A critical history of logo and constructionist learning. *Proceedings of the ACM on Human-Computer Interaction, 2* (CSCW), 1–19. https://doi.org/10.1145/3274287

Anderson, N., Lankshear, C., Timms, C., & Courtney, L. (2008). 'Because it's boring, irrelevant and I don't like computers': Why high school girls avoid professionally-oriented ICT subjects. *Computers & Education, 50*(4), 1304–1318. https://doi.org/10.1016/j.compedu.2006.12.003

Barron, B., Walter, S. E., Martin, C. K., & Schatz, C. (2010). Predictors of creative computing participation and profiles of experience in two Silicon Valley middle schools. *Computers & Education, 54*(1), 178–189.

Başer, M. (2013). Attitude, gender and achievement in computer programming. *Middle-East Journal of Scientific Research, 14*(2), 248–255. https://doi.org/10.5829/idosi.mejsr.2013.14.2.2007

Bell, J., & Bell, T. (2018). Integrating computational thinking with a music education context. *Informatics in Education, 17*(2), 151–166. https://doi.org/10.15388/infedu.2018.09

Bovée, C., Voogt, J., & Meelissen, M. (2007). Computer attitudes of primary and secondary students in South Africa. *Computers in Human Behavior, 23*(4), 1762–1776. https://doi.org/10.1016/j.chb.2005.10.004

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. Vancouver, Canada. In *Proceedings of the 2012 annual meeting of the American educational research association* (Vol. 1, p. 25). Retrieved from http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf.

Brown, A. R., & Ruthmann, S. A. (2020). *Scratch music projects*. Oxford University Press.

Burnard, P., Lavicza, Z., & Philbin, C. A. (2016). Strictly coding: Connecting mathematics and music through digital making. In *Proceedings of Bridges 2016: Mathematics, music, Art, Architecture, education, Culture* (pp. 345–350). Phoenix, Arizona: Tessellations Publishing.

Cazan, A.-M., Cocoradă, E., & Maican, C. I. (2016). Computer anxiety and attitudes towards the computer and the internet with Romanian high-school and university students. *Computers in Human Behavior, 55*, 258–267. https://doi.org/10.1016/j.chb.2015.09.001

Cheah, C. S. (2020). Factors contributing to the difficulties in teaching and learning of computer programming: A literature review. *Contemporary Educational Technology, 12*(2), Article ep272. https://doi.org/10.30935/cedtech/8247

Cheng, L. (2018). Teaching live coding of electronic dance music: A case study. *Dancecult, 10*(1). https://doi.org/10.12801/1947-5403.2018.10.01.10

Chomeya, R. (2010). Quality of psychology test between Likert scale 5 and 6 points. *The Journal of Social Sciences, 6*(3), 399–403. https://doi.org/10.3844/jssp.2010.399.403

Cohen, L., Manion, L., & Morrison, K. (2011). *Research methods in education* (7th ed.). London, New York: Routledge.

Creswell, J. W. (2014). *Research design: Qualitative, quantitative, and mixed-methods approaches (Fourth, international student*. Los Angeles, Calif: SAGE.

Davies, D., Jindal-Snape, D., Collier, C., Digby, R., Hay, P., & Howe, A. (2013). Creative learning environments in education—a systematic literature review. *Thinking Skills and Creativity, 8*, 80–91. https://doi.org/10.1016/j.tsc.2012.07.004

Dowdy, S., Wearden, S., & Chilko, D. (2004). *Statistics for research*. New Jersey, USA: John Wiley & Sons.

Engelman, S., Magerko, B., McKlin, T., Miller, M., Edwards, D., & Freeman, J. (2017). Creativity in authentic STEAM education with EarSketch. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on computer science education* (pp. 183–188). https://doi.org/10.1145/3017680.3017763

Fishbein, M., & Ajzen, I. (1975). *Belief, attitude, intention, and behavior: An introduction to theory and research*. MA: Addison-Wesley.

Freeman, J., & Magerko, B. (2016). Iterative composition, coding and pedagogy: A case study in live coding with EarSketch. *Journal of Music, Technology & Education, 9*(1), 57–74. https://doi.org/10.1386/jmte.9.1.57_1

Freeman, J., Magerko, B., Edwards, D., Mcklin, T., Lee, T., & Moore, R. (2019). EarSketch: Engaging broad populations in computing through music. *Communications of the ACM, 62*(9), 78–85. https://doi.org/10.1145/3333613

Gorson, J., Patel, N., Beheshti, E., Magerko, B., & Horn, M. (2017). TunePad: Computational thinking through sound composition. In *Proceedings of the 2017 Conference on Interaction design and children* (pp. 484–489). https://doi.org/10.1145/3078072.3084313

Grover, S., Pea, R., & Cooper, S. (2016). Factors influencing computer science learning in middle school. *Proceedings of the 47th ACM Technical Symposium on Computing Science Education - SIGCSE, '16*, 552–557. https://doi.org/10.1145/2839509.2844564

Günbatar, M. S. (2020). Computational thinking skills, programming self-efficacies and programming attitudes of the students. *International Journal of Computer Sciences and Engineering Systems, 4*(2), 24–35. https://doi.org/10.21585/ijcses.v4i2.96

Hava, K., & Koyunlu Ünlü, Z. (2021). Investigation of the relationship between middle school students' computational thinking skills and their STEM career interest and attitudes toward inquiry. *Journal of Science Education and Technology*. https://doi.org/10.1007/s10956-020-09892-y

Herman, G. L. (2012). Designing contributing student pedagogies to promote students' intrinsic motivation to learn. *Computer Science Education, 22*(4), 369–388. https://doi.org/10.1080/08993408.2012.727711

Horn, M., Banerjee, A., West, M., Pinkard, N., Pratt, A., Freeman, J., et al. (2020). TunePad: Engaging learners at the intersection of music and code. In *Interdisciplinarity in the learning sciences: Proceedings of the 14th International Conference of the learning sciences, ICLS 2020, [Nashville, Tennessee, USA], online Conference, June 19-23, 2020*. https://repository.isls.org/handle/1/6320.

Hsieh, H.-F., & Shannon, S. E. (2005). Three approaches to qualitative content analysis. *Qualitative Health Research, 15*(9), 1277–1288. https://doi.org/10.1177/1049732305276687

Introduction to TunePad. (n.d.). TunePad. Retrieved August 26, 2021, from https://tunepad.live/app/dropbook/15398.

Knezek, G., & Christensen, R. (1996). *Validating the computer attitude questionnaire (CAQ)*. Retrieved from https://files.eric.ed.gov/fulltext/ED398243.pdf.

Kong, S.-C., Chiu, M. M., & Lai, M. (2018). A study of primary school students' interest, collaboration attitude, and programming empowerment in computational thinking education. *Computers & Education, 127*, 178–189. https://doi.org/10.1016/j.compedu.2018.08.026

Köppe, C. (2020). Program a hit – using music as motivator for introducing programming concepts. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in computer science education* (pp. 266–272). https://doi.org/10.1145/3341525.3387377

LaBouliere, J. J., Pelloth, A., Lu, C.-L., & Ng, J. (2015). An exploration of the attitudes of young girls towards the field of computer science. In *2015 IEEE Frontiers in education Conference (FIE)* (pp. 1–6). https://doi.org/10.1109/FIE.2015.7344265

Lusa Krug, D., Bowman, E., Barnett, T., Pollock, L., & Shepherd, D. (2021). Code Beats: A virtual camp for middle schoolers coding hip hop. *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, 397–403. https://doi.org/10.1145/3408877.3432424

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior, 41*, 51–61. https://doi.org/10.1016/j.chb.2014.09.012

Manaris, B., Stevens, B., & Brown, A. R. (2016). JythonMusic: An environment for teaching algorithmic music composition, dynamic coding and musical performativity. *Journal of Music, Technology & Education, 9*(1), 33–56. https://doi.org/10.1386/jmte.9.1.33_1

Margolis, J., & Fisher, A. (2002). *Unlocking the clubhouse: Women in computing*. Cambridge, Massachusetts, USA: MIT Press.

Margolis, J., & Goode, J. (2016). Ten lessons for computer science for all. *ACM Inroads, 7*(4), 52–56. https://doi.org/10.1145/2988236

Mason, R., & Rich, P. (2020). Development and analysis of the elementary student coding attitudes survey. *Computers in Education, 153*, 103898. https://doi.org/10.1016/j.compedu.2020.103898

McKlin, T., Magerko, B., Lee, T., Wanzer, D., Edwards, D., & Freeman, J. (2018). Authenticity and personal creativity: How EarSketch affects student persistence. *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 987–992. https://doi.org/10.1145/3159450.3159523

McKlin, T., McCall, L., Lee, T., Magerko, B., Horn, M., & Freeman, J. (2021). Leveraging prior computing and music experience for situational interest formation. *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, 928–933. https://doi.org/10.1145/3408877.3432431

Nilson, C. (2007). Live coding practice. In *Proceedings of the International Conference on new interfaces for musical Expression*. New York, NY: NIME. http://www.nime.org/proceedings/2007/nime2007_112.pdf.

Payne, W., & Ruthmann, S. A. (2019). Music making in Scratch: High floors, low ceilings, and narrow walls? *The Journal of Interactive Technology and Pedagogy*. https://jitp.commons.gc.cuny.edu/music-making-in-scratch-high-floors-low-ceilings-and-narrow-walls/.

Perera, P., Tennakoon, G., Ahangama, S., Panditharathna, R., & Chathuranga, B. (2021). A systematic mapping of introductory programming languages for novice learners. *IEEE Access, 9*, 88121–88136. https://doi.org/10.1109/ACCESS.2021.3089560

Petrie, C. (2019). *Interdisciplinary Computational Thinking with Music and Programming: A Case Study on Algorithmic Music Composition with Sonic Pi*. Master's thesis submitted to the University of Canterbury https://doi.org/10.26021/9534.

Rubin, M. J. (2013). The effectiveness of live-coding to teach introductory programming. *Proceeding of the 44th ACM Technical Symposium on Computer Science Education - SIGCSE, '13*, 651–656. https://doi.org/10.1145/2445196.2445388

Siva, S., Im, T., McKlin, T., Freeman, J., & Magerko, B. (2018). Using music to engage students in an introductory undergraduate programming course for non-majors. *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 975–980. https://doi.org/10.1145/3159450.3159468

Teo, T. (2007). Perceived importance, enjoyment, and anxiety as correlates of computer attitudes. *Psychological Reports, 100*(1), 127–135. https://doi.org/10.2466/pr0.100.1.127-135

Teo, T. (2008). Assessing the computer attitudes of students: An Asian perspective. *Computers in Human Behavior, 24*(4), 1634–1642. https://doi.org/10.1016/j.chb.2007.06.004

TunePad About Us. (n.d.). Retrieved August 26, 2021, from https://tunepad.live/about.

Teo, T. (2006). Attitudes toward computers: A study of post- secondary students in Singapore. *Interactive Learning Environments, 14*(1), 17–24. https://doi.org/10.1080/10494820600616406

Vekiri, I. (2010). Boys' and girls' ICT beliefs: Do teachers matter? *Computers & Education, 55*(1), 16–23. https://doi.org/10.1016/j.compedu.2009.11.013

Wanzer, D. L., McKlin, T., Freeman, J., Magerko, B., & Lee, T. (2020). Promoting intentions to persist in computing: An examination of six years of the EarSketch program. *Computer Science Education, 30*(4), 394–419. https://doi.org/10.1080/08993408.2020.1714313

Weijters, B., Baumgartner, H., & Schillewaert, N. (2013). Reversed item bias: An integrative model. *Psychological Methods, 18*(3), 320–334. https://doi.org/10.1037/a0032121