

## CATCHING TWO BIRDS WITH ONE STONE - THE CASE OF MUSIC AND PROGRAMMING FUNDAMENTALS

**Ilana Lavy**

*Information Systems Dep., Yezreel Valley College (Israel)*

### Abstract

This study describes a unique learning experience of elementary school students that learn programming fundamentals via engagement in music programming of known melodies using the Scratch 3.0 environment. Eleven six-grade students participated in weekly activities each lasting 90 minutes along one semester. The learning units were planned in a way that the engagement with the programming of well-known melodies stimulates the curiosity to get mastery over the above programming concepts and structures. The research aim is to explore the effects of teaching fundamental programming structures and concepts via music. Since it is a pilot study, the research method used is of action research in which each lesson is redesigned according to conclusions and insights gained from previous lessons. The research data include transcripts of the class discussions, the students' feedback after each lesson and the researcher's reflective journal. From the data analyzed so far, four categories emerged: enjoyment, interest, gaining of programming knowledge and experiencing feelings of success. Instead of the teacher's initiation of teaching new programming concepts, the need for these concepts emerged from the students during their engagement in music programming. The students were enthusiastic during the learning lessons, and they were curious to learn and were eager to get mastery over advanced concepts right from the first lesson. The music programming provided them with practical meaning for the learned programming structures and concepts.

**Keywords:** *Music, programming fundamentals, Scratch 3.0, spiral learning, constructivism.*

---

### 1. Introduction

Accelerated technological development creates opportunities to revolutionize the pedagogy world. Incorporating technology into education is a complex and costly challenge. Technological trends identified are available and inexpensive computing infrastructures, cloud computing, the Internet of Things, and so forth (OECD, 2019). In "Mindstorms: Children, Computers, and Powerful Ideas", Seymour Papert (1980) paved the way to the idea to teach programming to young students. Papert believed that the learning of programming should be done through playful exploration in which young students take control of their own learning.

In the field of education, various teaching methods have evolved over the years. Among the effective ways to learn the spiral method. Bruner (1960) proposed the spiral curriculum, a teaching approach in which each topic or skill area is revisited at intervals, at a more complex level each time. This teaching method enables the establishment and solidification of the learning topic in a profound manner. Spiral learning provides a gradual learning process from simplistic ideas to complicated ones (Johnston, 2012). Considering what is written above, I found the spiral learning method suitable for teaching programming principles to young students. To enable the students to find the practical meaning of programming principles I chose to teach programming via music using a simple programming language (Scratch). To avoid unnecessary syntax mistakes, Scratch commands are designed as blocks that can be dragged and executed. The Scratch environment is taught in some elementary schools in Israel using the Hebrew language. Despite the difficulty of programming in a language that is not one's native language, I believe it is important to teach programming in English because it is the language of all the programming languages and students have to get used to it right from the beginning. To overcome the language barrier, I added the translation of the new English words in each of the learning units.

Via the composing of familiar melodies, the learner will be acquainted with programming concepts such as variables, repetitive structures, conditional commands, functions and more. Moreover, the teaching/learning process will follow the constructivist theory in a way that the students will be engaged in personal exploration tasks. In the last fifteen minutes of each session, a class discussion will take place in which the new concepts will be discussed, and the students will share the insights they gained during their individual explorations.

## **2. Brief theoretical background**

### **2.1. Music and programming**

Music adds joy, color and pleasure to our lives, and those who are blessed with musical talent even create it for us. Due to the significant role of music in our lives, there are those who even attribute therapeutic capabilities to it (Clair & Bernstein, 1990). Recognizing its merits, music has become an integral part of many multimedia environments developed over the years such as MicroWorlds Project Builder, Scratch, ScratchJr, Alice, etc. These multimedia environments were developed to make programming learning accessible while also developing reading, writing, and analytical skills for young learners. The ScratchJr environment provides a platform for developing problem-solving skills in a playful and tailored manner to the development of the skills mentioned above in early childhood education (Papadakis et al., 2016). The Scratch software, used in this study, is a blockchain visual programming language that is most suitable for young learners. One of the common attributes the above-mentioned environments share is music which provides an added value to the projects students can develop.

In the present study, music is used as a tool for learning. That is, through the programming of simple and familiar melodies, students become familiar with basic programming concepts and principles such as variables, parameters, functions, repetitive and control commands, parallel programming and more.

### **2.2. The spiral learning method**

The term 'spiral curriculum' was coined by Bruner (1960). This term encompasses the following ideas: (a) complex topics, concepts/ideas are revisited several times during school studies; (b) The level of complexity of these topics concepts/ideas increases with each repetition; and (c) Repetition of these topics, concepts/ideas is always associated with their prior learning. Spiral learning enables, among other things, to assimilate the new concepts/ideas and their understanding in each revisit. Spiral learning also provides a process of gradual learning from simple ideas to complex (Harden, 1999; Johnston, 2012).

Schneiderman (1977) used the spiral learning approach to teach beginner programmers, arguing that this approach makes programming education more natural for students, reduces the anxiety associated with learning of a new language and promotes the development of computer literacy. He also argued that the spiral approach enables the parallel acquisition of syntactic and semantic knowledge continuously, which increases student motivation by using meaningful examples, building on prior knowledge, and all in tune with the student's cognitive skills. Also, learning in this approach enables the assimilation of new material, and develops confidence in performing increasingly difficult tasks.

## **3. The study**

### **3.1. Population**

This study is a preliminary study conducted in an action research format (Stringer, 2013). Eleven (eight male and three female) students in grade six (one from grade five) from an elementary regional school in the north of the country participated in the study. Three students (two males and one female) have basic music knowledge. They attend the one-week meeting of 1.5 hours during school hours.

### **3.2. Research aim and questions**

The aim of the study is to explore the effect of learning programming via music on the students' attitudes and perceptions as regards programming studies. The research questions derived from this aim are:

1. Does learning through music affect: (a) Students' desire to learn to program, (b) their motivation to invest in learning, (c) their understanding and internalization of the concepts learned and in what ways.
2. Is there any added value to learning programming through music and if so, what is that value?

3. Has learning via music influenced students' self-concept regarding their ability to handle programming learning? If so, in what way?

Due to space limitations, in this paper, I will refer only to the first research question.

### 3.3. Research tools

Students were provided with study units developed by the researcher. The idea behind these units is that during the process of music programming, the need for relevant programming structures will emerge. The units were developed in the spirit of the constructivist approach to learning (Cooperstein & Kocevar-Weidinger, 2004). The learning activities were built according to following principles: (a) the learning is in English and not in the students' home language; (b) the learning method is designed to match the spiral learning approach, and (c) to raise the learning effectivity, the learning units were built in the spirit of the constructivism. Although the Scratch environment was translated into many languages, it is important to educate students to program in English right from the beginning since all programming languages are in English. The study units were designed according to the spiral learning method, in which the learning concepts are revisited several times while their level of complexity rises. The learning units are also planned for self-learning and include activities through which the students become acquainted with basic concepts of programming such as methods, variables, repetition and control commands, parallel processes and so forth. By the end of each session, a class discussion is initiated in which emerging ideas and insights are discussed and summarized.

The present research is action research (Stringer, 2013), which in this case its purpose is to learn from the feedback given by the students and from the researcher's insights in order to improve successive learning units (Denscombe, 2014). Action research includes the following stages: (1) identify a change; (2) observe the present situation; (3) plan different possible interventions; (4) carry out the intervention; (5) observe the effects of the intervention; (6) evaluate against original goals; and (7) share findings.

In the current study, the understanding that traditional methods used to teach programming deter students from choosing programming studies has led to the development of a learning method that integrates enjoyable activity (music) with programming (addressing stages 1–3). Within the weekly lessons, the developed units were taught to the study participants (stage 4). During the lessons, the researcher wrote a reflective journal in which she documented her insights as regards the students' expressions, their behavior and the kind of questions asked. The writing in the reflective journal was also influenced by the students' feedback given at the end of each lesson (stages 5 and 6). The insights gained from the evaluation process were implemented in the successive learning unit (stage 7).

After each meeting, the data that include the students' outcomes of the inquiry tasks, the reflective notes taken by the researcher and the feedback given by the students at the end of each lesson were analyzed using content analysis method (Neuendorf, 2002), which revealed four main categories: experiencing pleasure during learning, interest stemming from needs, acquisition of programming knowledge, and Acquire experiences of success.

## 4. Results and discussion

In what follows, the first research question is addressed.

### 4.1. Experiencing pleasure during learning

Jenkins (2002) argued that learning programming can be very boring. Lectures focusing on syntax details will never be inspiring. Exercises that include simple mathematical manipulations on data such as student scores, different business inventory management, or bank account information will never increase learners' interest. However, looking at many programming lessons shows examples of the above. In referring to the enjoyment and interest in learning, he emphasized that programming at its best can be a fun and creative activity, and many students can take great pleasure in being exposed to inspiring tasks and regret learning. This is still rare, learning music programming is an example of fun and creative learning.

During the lessons, the students were very curious and enthusiastic and expressed a feeling of joy. These feelings were highlighted in verbal phrases like: "Wow! [Proudly] Look what my animation looks like!" Or "I like the combination of music and programming. It revives the 'dry' commands"; "listen to the music I created; I have no mistakes at all! Imagine that [proudly] I have just learned how to read music notes". These phrases and more were repeated in every lesson. Students' enjoyment was noticeable also by their body language. Happy voices were heard during lessons and there was a relaxed atmosphere in class. They arrived at the lessons before the start time and stayed even after the lessons ended.

In the second lesson, I decided to teach them basic music knowledge, so that they will be able to read music notes. At the end of the lesson, all of them could write notes of a song independently. Later, when I offered them readily done notes of a song, they refused to utter they wanted to write the notes by

themselves. During class discussions, most of them said in similar words that the music made a difference. They also mentioned that if they had to study the programming commands in the traditional way, they probably would have stopped attending these classes.

#### **4.2. Interest stemming from needs**

From the first class, the students showed great interest. Their interest was reflected in asking questions that had to be addressed in successive lessons. Questions regarding the use of advanced programming commands that have not been learned, how to play the music in higher/lower octaves, how to play the music with two hands in concert and more.

Although each student worked independently on the inquiry tasks, there was a lot of activity in the classroom. During the lessons, many questions were asked by the students, some of whom were referred to the researcher and some to other classmates. Students made sure to check on the progress and achievements of their classmates, and once they encountered unknown feature or programming structure that their peers used, they were eager to gain control of them as well. They were constantly asked questions about advanced functionalities that were not included in the current learning unit. As a result, an atmosphere of the positive competition was induced which resulted in a creative and amazing project. The fact that there was an immediate result of the programming commands they used (they could hear the melody written up to that point) motivated them to continue and finish successfully the music they programmed. Moreover, when one of the students discovered a feature or capability in the software, she/she shared it with other classmates at once.

The significant insight from lessons was that the need to learn a certain functionality stemmed from the students rather than the teacher's decision to teach it. For example, after encoding notes of a well-known tune ('Little Jonathan'), the students noticed that the octave in which the music was playing was quite low. The researcher asked them what could be done to play it at a higher octave. They said it didn't make sense to rewrite all notes by changing the characters' values and there must exist a programming functionality that will provide a more elegant solution. They said that this functionality will enable us to change the values in a way that the same melody could be played in different octaves without making too many changes. At that point, they were ready to learn the concept of a variable, which in fact the need for it came from their behalf. After demonstrating how to define a variable and its associated commands (initializing a variable, changing its value), they developed their idea by saying that we should use two variables, one for the value of the played character and one for the time the character should be played. Then they went over all the tunes previously programmed and changed the commands to include variables without being asked to do so.

In another case, they observed that for some melodies, there are two lines of notes, one for the melody (the right hand) and one for the accompaniment (left hand). They asked how we could translate this situation into programming. One of them replied that we had to write two different programs, one for right-hand characters and one for left-hand characters, and both programs had to be played simultaneously. Then another question was raised: How to play two programs at the same time. Although this was an initial phase of the course, they were given a basic explanation of parallel processes, stating that we will return to discuss this topic in more detail later.

To get students to internalize the meaning of nesting loops, students were asked to compose accompaniment of various types of drums for their programmed melodies. After a simple demonstration, the students were so enthusiastic and offered to build a drum accompaniment for each of the melodies they had programmed so far and competed as regards the level of accompaniment complexity.

#### **4.3. Acquisition of programming knowledge**

During the students' engagement in the inquiry tasks, they became acquainted with basic programming structures and concepts. To facilitate the understanding of some of the concepts, the learning units have been designed according to the spiral learning method (Bruner,1960). That is, complex programming structures and concepts, were revisited several times in a raised level of complexity. For example, dealing with repetition structures first referred to definite loops, then presented with indefinite loops and finally with nested loops. The need to use loops, as well as other learned concepts, stemmed during music programming. That is, when a group of notes had to be repeated, they used a definite loop and when they had to write an accompaniment of drums to music, they used nesting of loops. Variables have been associated with playing melodies at various octaves. These associations have a dual role. First, they enabled the connection of programming concepts to concrete needs, and second, they may develop students' understanding and appreciation of these programming structures, hoping that when they meet these structures and concepts in the future, they can easily assimilate them.

When the students were asked to animate their melodies, a competition atmosphere was created in relation to the sophistication of the animations they provided. Some of the students created

sophisticated games, using advanced features of the Scratch environment which resulted in expanding their programming knowledge. the acquisition of programming knowledge was done in an enjoyable manner. The engagement in music and the aim to reach correct and rich music motivated them to get mastery over the programming structures and concepts that will help them reach their goal.

#### 4.4. Acquire experiences of success

The students' behavior during lessons reflected on their self-perception as regards to their ability to cope with the inquiry tasks. Practicing programming using music brought humor and color to the lessons. Curiosity and eagerness to learn advanced functions to improve the students' products were observed. The review of student products revealed that a great deal of thought was invested in them. The students were motivated to create relevant animation to the melody they programmed, to accompany drum accompaniment to fit the correct rhythm of the melody using advanced programming commands that included control structures. During the time between two successive lessons, they sent via email to the researcher improved versions of their programmed animations, which can imply on their level of satisfaction from their achievements. These results are consistent with Bandura (1986), who stated that a person with a high sense of self-aptitude tends to respond to challenges and persist in efforts to achieve his/her goals.

After the sixth meeting, one of the students said: "I have to say I had doubts at the beginning of the course if I could learn to program. What broke the ice was the music practice; it raised my motivation to create a sophisticated animation program that would fit into the melody." Another student said: "Music has made programming something concrete and efficient, which increased my desire to learn it. Not just a theoretical study of commands, but learning for concrete purposes, such as creating drums accompaniment or playing in various octaves." The student said: "I don't believe I am saying it, but despite all my fears at first, I understand the programming commands thanks to music. I learn because I wanted to and not because I had to."

From the above, it can be concluded that music played an important role in influencing their self-perception in relation to their ability to handle programming tasks. This way of learning increased their motivation to succeed in providing solutions to concrete problems. During the activities, students experienced feelings of success both instantaneously (accurately of music) and over time (projects that were appreciated by peers and the researcher). These feelings helped the improvement of their self-perception in their ability to cope with programming learning. Furthermore, during the learning of programming structures and concepts, they also enriched their musical knowledge.

#### References

- Bandura, A. (1986). *Social foundations of thought and action: A social cognitive theory*. Englewood Cliffs, NJ: Prentice-Hall.
- Bruner, J.S. (1960). *The Process of Education*. Cambridge MA, Harvard University Press.
- Clair, A. A., & Bernstein, B. (1990). A preliminary study of music therapy programming for severely regressed persons with Alzheimer's-type dementia. *Journal of Applied Gerontology*, 9(3), 299-311.
- Cooperstein, S. E., & Kocevar-Weidinger, E. (2004). Beyond active learning: a constructivist approach to learning. *Reference services review*, 32(2), 141-148.
- Denscombe, M. (2014). *The good research guide: for small-scale social research projects*. McGraw-Hill Education (UK).
- Harden, R. M. (1999). What is a spiral curriculum? *Medical teacher*, 21(2), 141-143.
- Jenkins, T. (2002, August). On the difficulty of learning to program. In Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences (Vol. 4, No. 2002, pp. 53-58).
- Johnston, H. (2012). *The Spiral Curriculum. Research into Practice*. Education Partnerships, Inc
- Neuendorf, K. A. (2002). *The Content Analysis Guidebook*, Thousand Oaks, CA: Sage Publications.
- Papadakis, S., Kalogiannakis, M., & Zaranis, N. (2016). Developing fundamental programming concepts and computational thinking with ScratchJr in preschool education: a case study. *International Journal of Mobile Learning and Organization*, 10(3), 187-202.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.
- Shneiderman, B. (1977). Teaching programming: A spiral approach to syntax and semantics. *Computers & Education*, 1(4), 193-197.
- Stringer, E. T. (2013). *Action research*. Sage publications.