

Tackling the Technology Gender Gap Together: Scratching the Surface of Musical Coding

Level - Second Level

**Subject area/s – Expressive Arts – Music -
Technologies - Computing Science**



Scratch-ing the Surface of Musical Coding

Level - Second level

Subject area/s – Expressive Arts – Music - Technologies - Computing Science

Experiences and Outcomes –

Computing Science

- I understand the operation of a process and its outcome. I can structure related items of information. **TCH 2-13a**

Music

- I can use my voice, musical instruments and music technology to experiment with sounds, pitch, melody, rhythm, timbre and dynamics. **EXA 2-17a**
- Inspired by a range of stimuli, and working on my own and/or with others, I can express and communicate my ideas, thoughts and feelings through musical activities. **EXA 2-18a**

Benchmarks – (Computing Science)

- Compares activities consisting of a single sequence of steps with those consisting of multiple parallel steps, for example, making tomato sauce and cooking pasta to be served at the same time.
- Identifies algorithms/instructions that include repeated groups of instructions a fixed number of times and/or loops until a condition is met.
- Identifies when a process is not predictable because it has a random element for example, a board game which uses dice.
- Structures related items of information for example, a family tree (MNU 2- 20b).
- Uses a recognised set of instructions/ an algorithm to sort real worlds objects for examples, books in a library or trading cards.

Duration of time – 6 x 1 Hour Lessons

Computing Science Concepts and Approaches:

- Algorithms – creating a set of instructions to plan their music composition.
- Abstraction – taking a musical tune and reducing difficulty to code.
- Debugging – finding and fixing errors in alogorthm and code throughout the tasks.
- Logical thinking – creating the code in the most logical way.
- Decomposition – breaking down a song into parts.
- Perserverance – even when finding coding tricky, to keep going.

Previous Learning:

Computational concepts have been introduced and children have had opportunity to explore basic algorithms.

A knowledge of musical notes could be beneficial but is not necessary.

Overview of learning

This series of lessons will enable learners to code a song/tune using the coding platform 'Scratch' <https://Scratch.mit.edu/>.

Below is a breakdown of the learning. Each lesson can be taught on a weekly basis.

Lessons 1-2

(only necessary if learners need reinforcement of concepts/Scratch):

- Understanding computational concepts/Scratch Program.
- Using unplugged activities to introduce concepts and approaches: algorithms and debugging.
- The coding platform 'Scratch' is introduced, with opportunities for learners to enhance coding skills.

Musical Coding (Lesson 3: 3x1 hour lessons)

- Using Scratch to compose.

Lesson 4

- Testing and Evaluation.

Teaching Notes

- Each lesson is planned below.
- Help sheets and PowerPoints are included.

Pupil Objectives

- Each set of pupil objectives are highlighted in the lesson plans below. Objectives will change as the lessons progress.

Differentiation

Based on learners's ability levels, expectation and support is differentiated. Creating further criteria for the song/tune can be used to challenge. Opportunities for differentiation are highlighted throughout lessons.

Assessment Opportunities

- Photographs,
- Screen shots of code,
- Samples of algorithms,
- Video,
- Peer/self-assessment throughout,
- Final Showcase.

Logins for Scratch are recommended. Learners can save work and it can be stored on the Scratch website. If you are using a teaching account, as shown in the help sheet, you can manage each account and see the progress of each pupil.

Curriculum Links

Literacy: Instructional Writing.

The basis of an algorithm is incredibly similar to instructional writing. Teaching these at the same time could provide a meaningful link to another area of the curriculum.

Resources required

Paper, pens

Laptops/PCs

<http://lego.brickinstructions.com/>

[You Tube Video – Music Key Play](#)

Lessons 1-2 Learning Overview

Learning Intention

- We are learning to use computational thinking.
- We are learning to use Scratch.

Success Criteria:

- I can discuss the meanings of algorithm and debugging.
- I can create an algorithm.
- I can debug an algorithm.
- I can create a simple code using Scratch.

Definitions:

Algorithm

An algorithm is a sequence of instructions or a set of rules to get something done.

Debugging

Finding and fixing errors.

Lesson 1

Success Criteria:

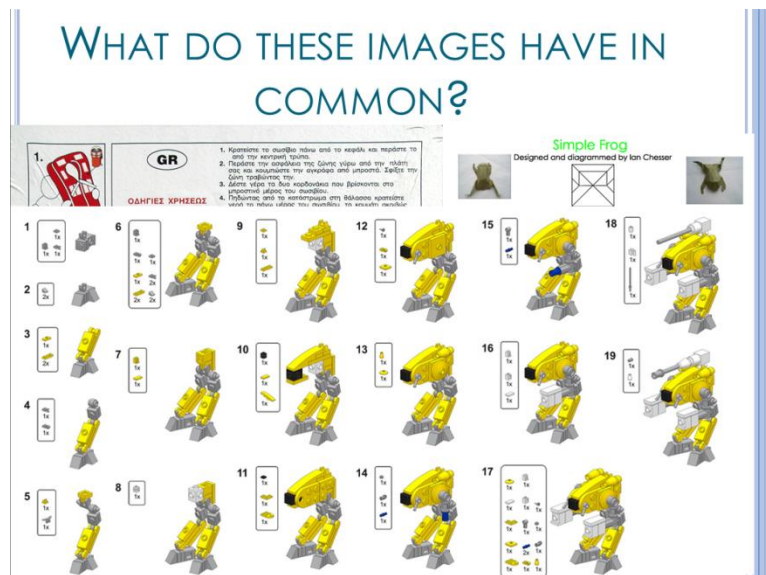
- I can create an algorithm
- I can debug an algorithm

Introduction

During the first week, your aim is to introduce learners to computational thinking and the concept: **algorithm** and approach: **debugging**.

Learners could bring in their own Lego instructions or bring in Lego (or an alternative) to build as support for following activity. See: <http://lego.brickinstructions.com/>

1. Introducing this concept should be as practical as possible. Start with the following activity (See PowerPoint): What do all of the images have in common?



They are all instructions.

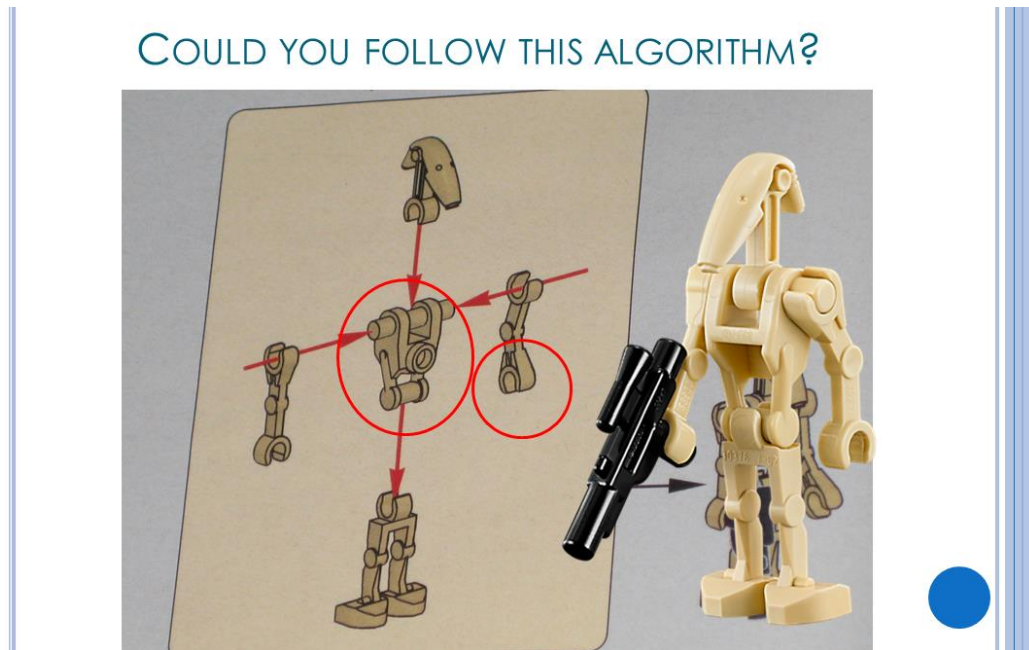
2. Discuss the word 'algorithm'.

Can learners think of algorithms around them? The world is full of them: can they think of the most obscure algorithm they can?

I.e. Maths Calculations, traffic lights, recipes, football!

Main

3. Show slide 6. Discuss algorithm – prompt children to see errors. Highlight term ‘debugging’.



4. Slide 8. Learners practice writing algorithms and testing bugs.

Plenary

5. Test learner’s algorithms as a class. Review success criteria – can the learners tell you what algorithms and debugging is?

Throughout the week, try and use the computational language as much as possible, link it to what you are doing in all areas of the curriculum to embed before next lesson.

Lesson 2

Success Criteria:

- I can create simple code using Scratch.

Please see ‘Scratch – How to guide’ for instructions on how to access Scratch Teacher’s Account.

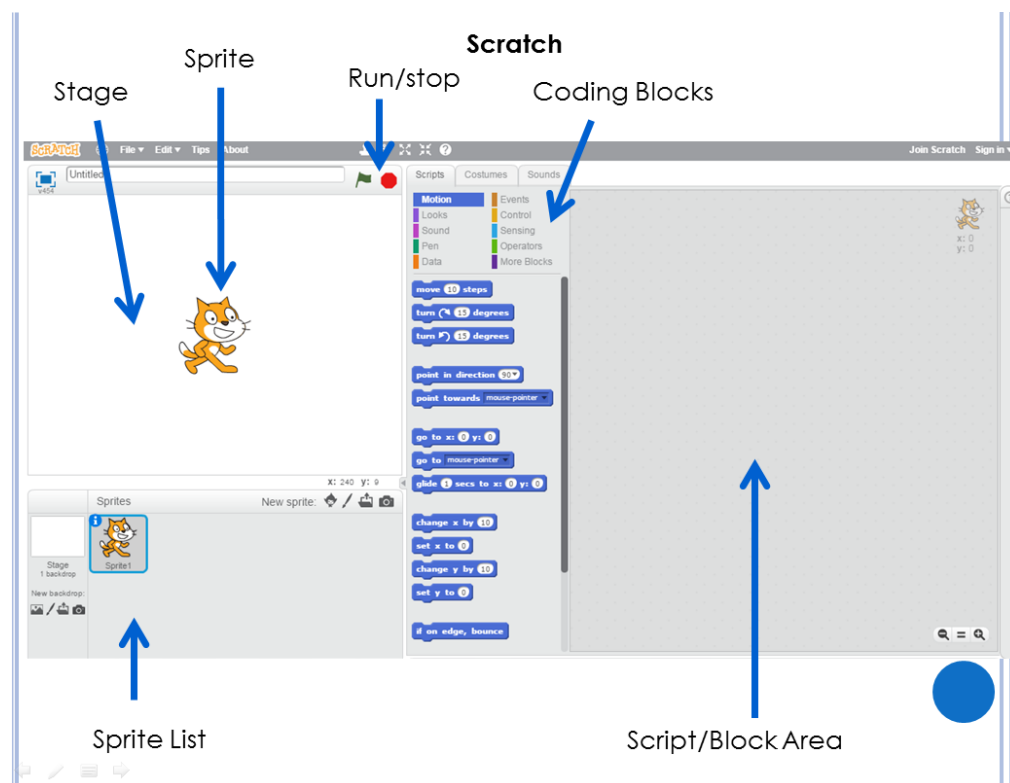
Introduction

This lesson’s aims are to introduce learners to a programming language called ‘Scratch’.

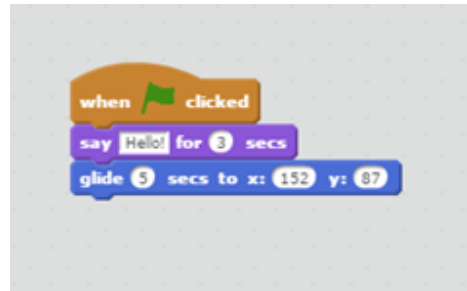
1. Go over concepts from previous lesson. Discuss with learners that we need to make sure algorithms are clear and step by step because a computer only runs commands as they are written.

Main: Scratch/Computer Code

2. When we use algorithms on computers, it is called *scripts* or *code*. This is what computer programmers develop and how computers work.
3. Show slide 10. This is the basic Scratch interface – explain that Scratch is simply a block-based computer language which allows us to program a variety of things.



4. Show how blocks work, basic blocks control sprite movement.



5. Set learners the task on slide 13. In pairs, ask learners to use Scratch to attempt all of the challenges.

Plenary

6. Reinforce concepts and learning over the past two lessons/weeks. What have they learned?

Musical Coding (lesson 3)

Learning Intention

- We are learning to use computational thinking.
- We are learning to read music.
- We are learning to compose a song using coding.

Success Criteria:

- I can create an algorithm.
- I can debug an algorithm.
- I can research using the internet.
- I can understand how music is notated.
- I can recognise notes on a stave.
- I can recognise types of musical notes.
- I can compose a simple tune using code.

Introduction

Begin listening to the simple tune on slide 1 until 17 seconds.

Discuss with learners: What makes it a tune? *Notes*. Every tune is a collection of notes, or chords, played on instruments to create a song.

Second half: Highlight that the notes played are looped and there is a steady drum beat throughout.

The second tune has added an additional layer for effect.

1. Link music to algorithms.

Main

2. Learners must choose a tune that is simple to start with. For example, 'Twinkle Twinkle', or 'Mary had a little lamb'. If this term is close to Christmas, Christmas carols/songs can be straight forward too!

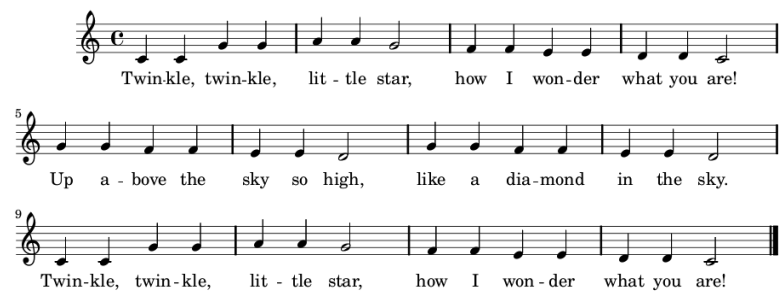
Mary Had a Little Lamb

Sarah Josepha Hale

Lowell Mason



Twinkle Twinkle Little Star



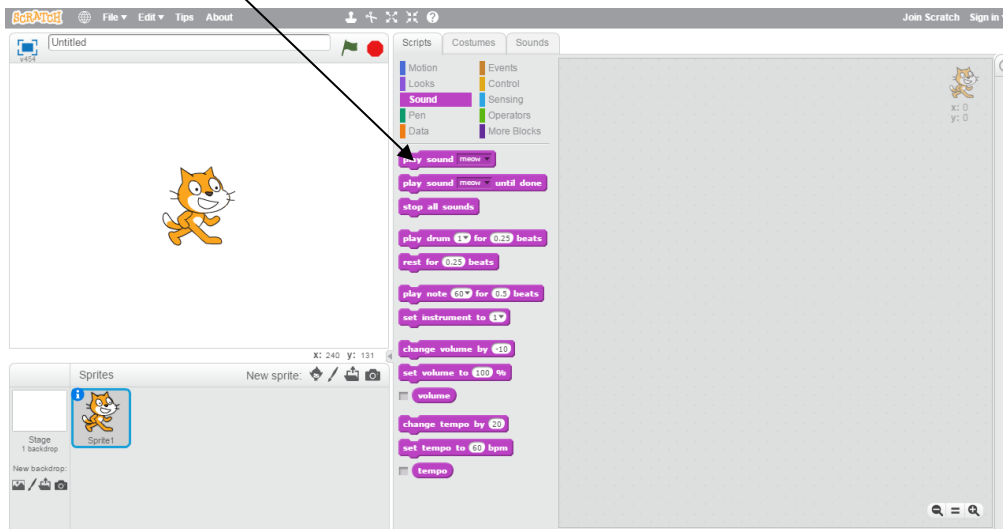
Whether teachers allow learners to choose, ask all learners to code the same song, or choose tunes for learners, is up to teacher's discretion.

3. Slides 5-12 go through basic note structure. This gives learners a basic understanding of how music is composed and the correct terminology, linking with computational thinking throughout. An example of how this is used in Scratch is included.

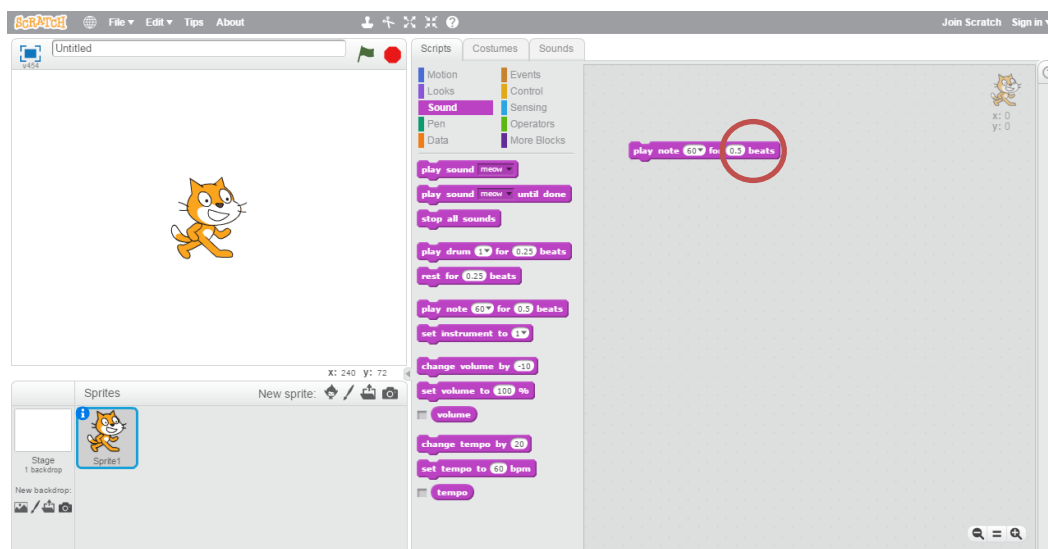
4. Allow learners time for research to explore what tune they would like to start with.
5. Reinforce to learners how to use Scratch to create a tune. Allow opportunity to practice.

Create new Scratch project.

Click on sound scripts.

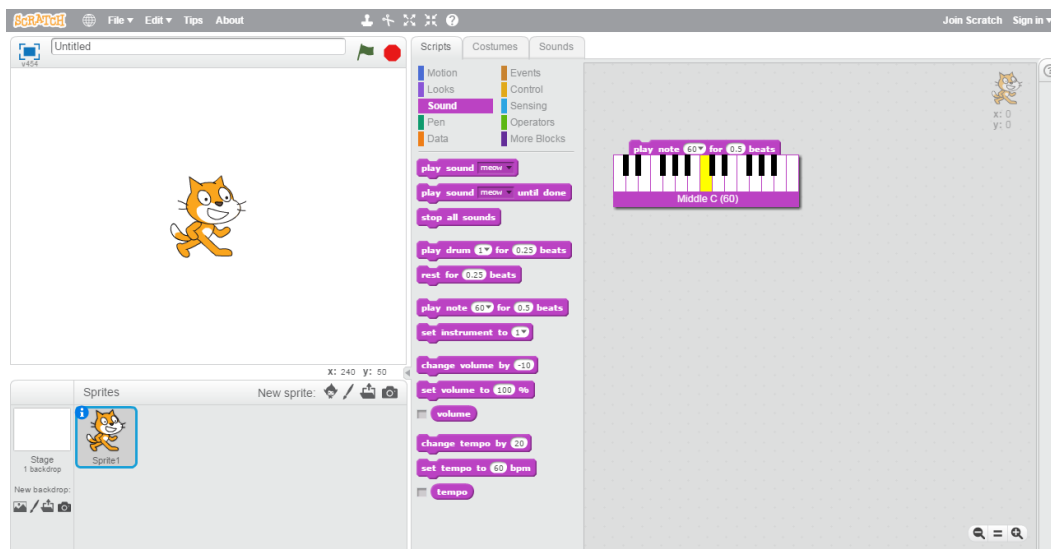


Drag block across to script area. Notice the beat; you can change this to match your tune.



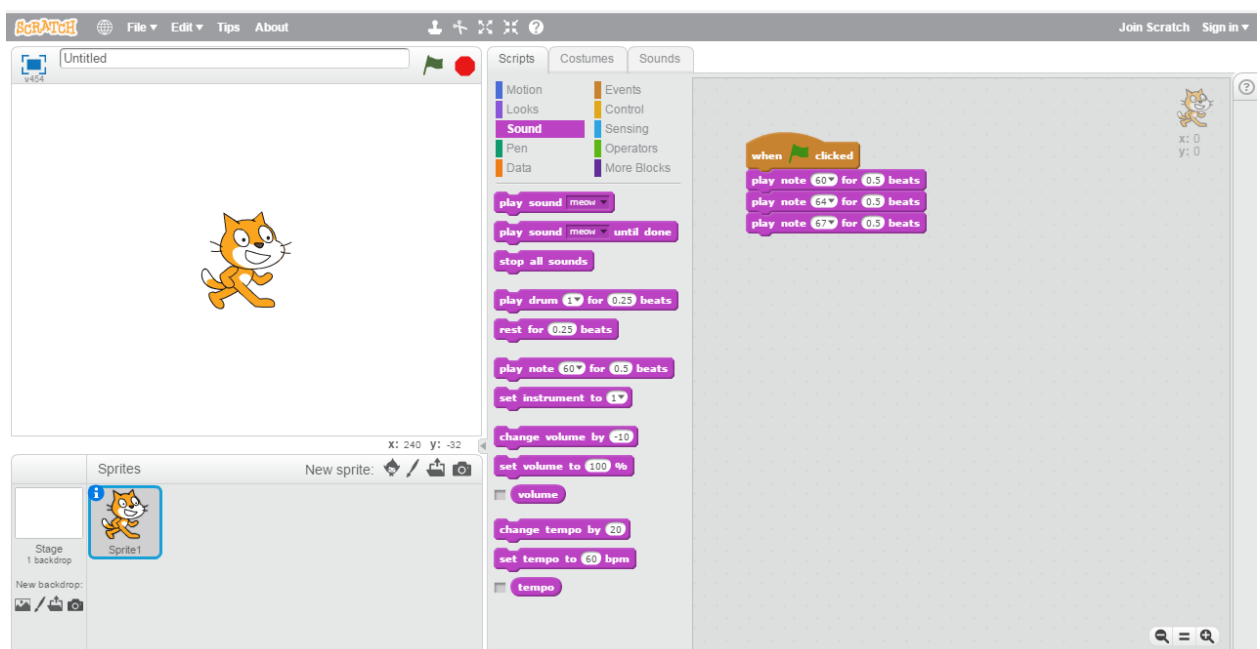
Playing a different note:

Select drop down arrow beside number. The keyboard shows the notes, clicking a different key will change the note.



Stacking blocks and changing notes will develop your tune.

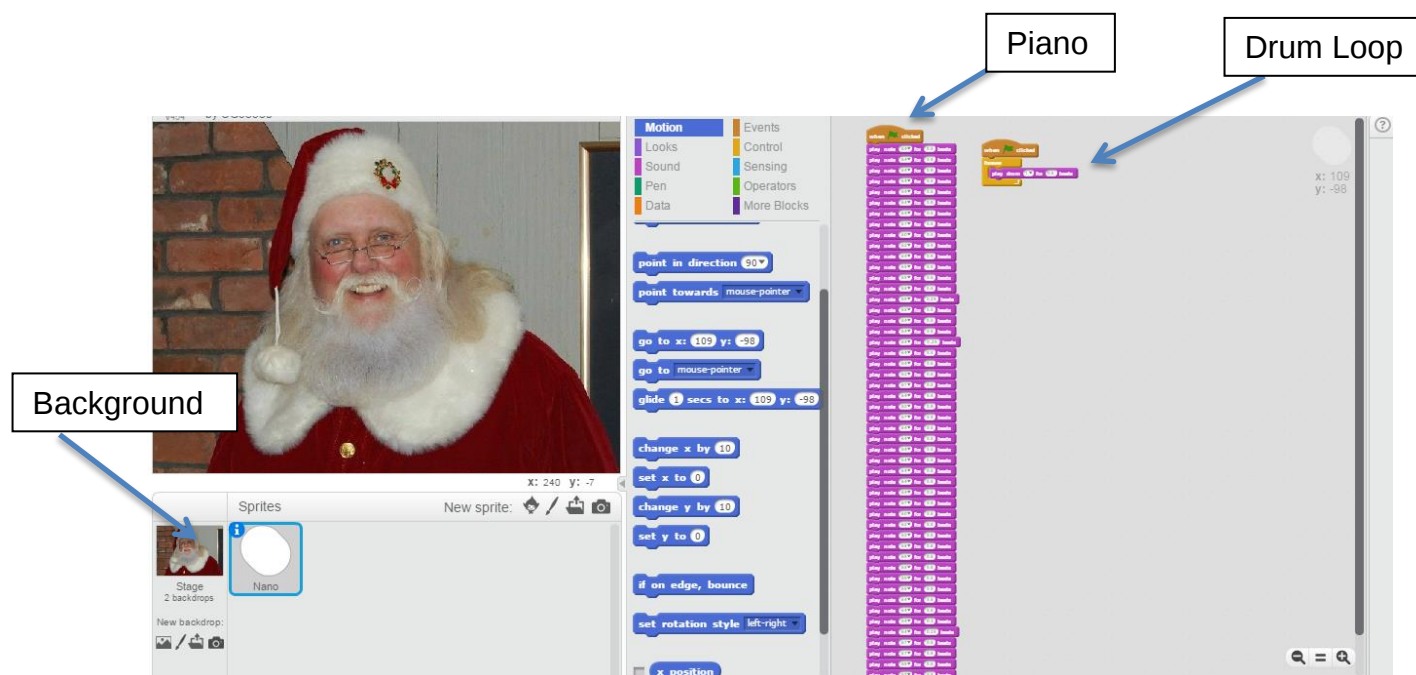
There are rest blocks, changing instrument blocks and drum blocks to add to your tune.



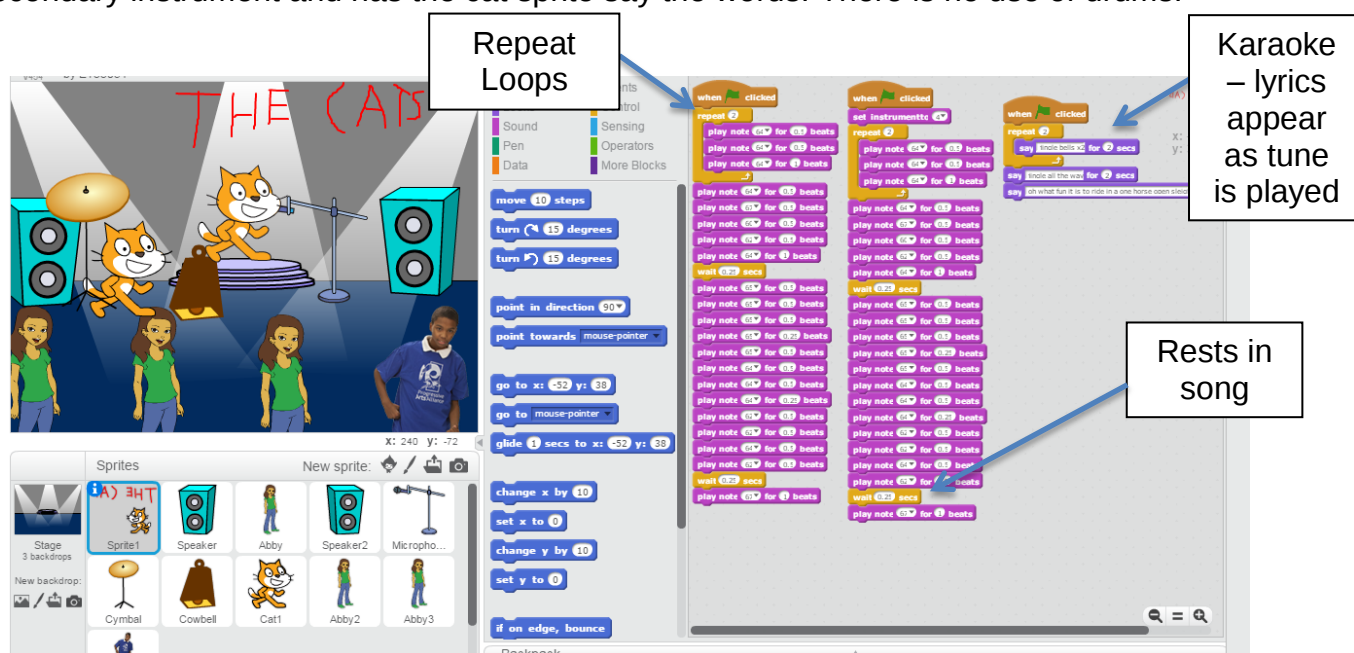
6. Set learners task of composing a tune. Stress the importance of keeping it simple. Give learners time to compose; this could be a number of lessons. Encourage perseverance and collaboration.

7. Once confident, learners can then add percussion and improve the stage for their song.

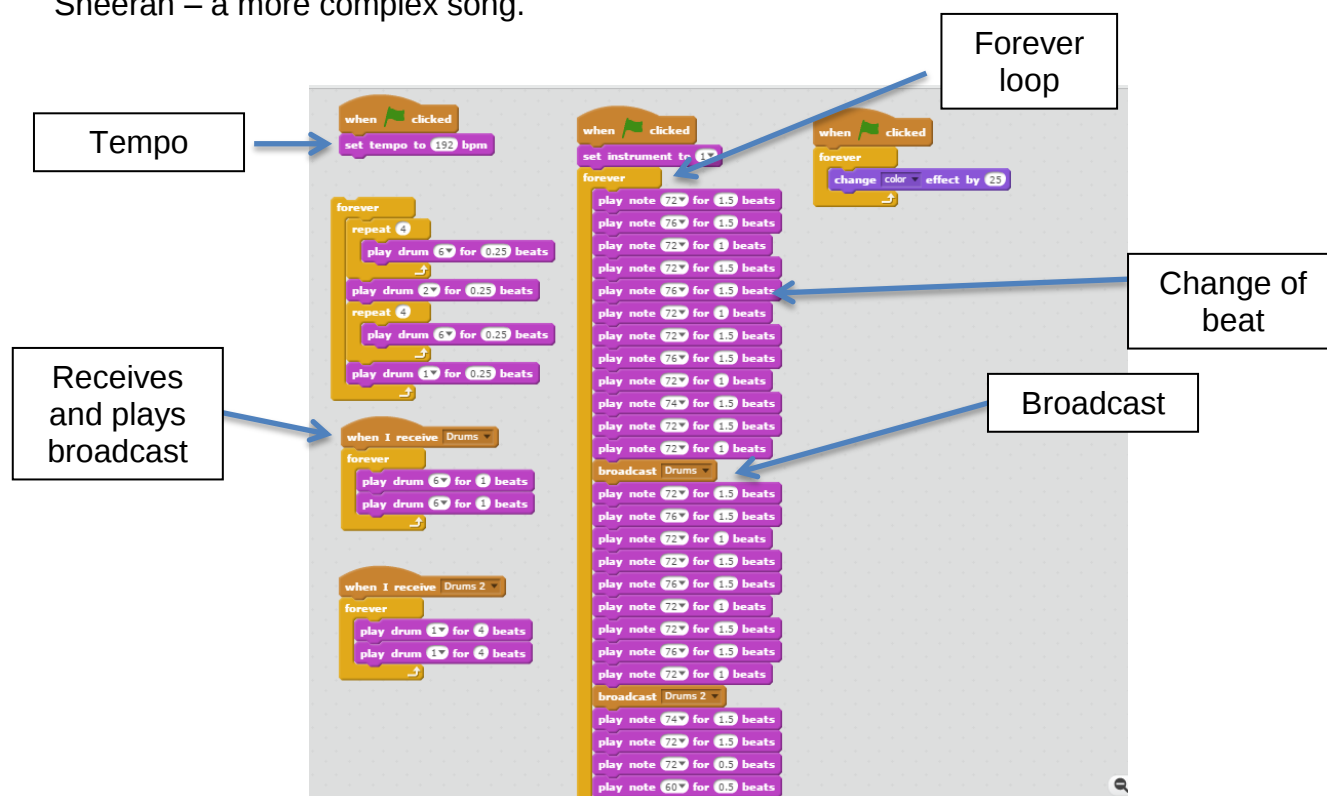
This code plays jingle bells, however; no loops are used in simplifying piano code. There is use of basic drums. The background is stationary, this could be developed.



This code also plays jingle bells; the learner has used repeat loops, rests, has added a secondary instrument and has the cat sprite say the words. There is no use of drums.

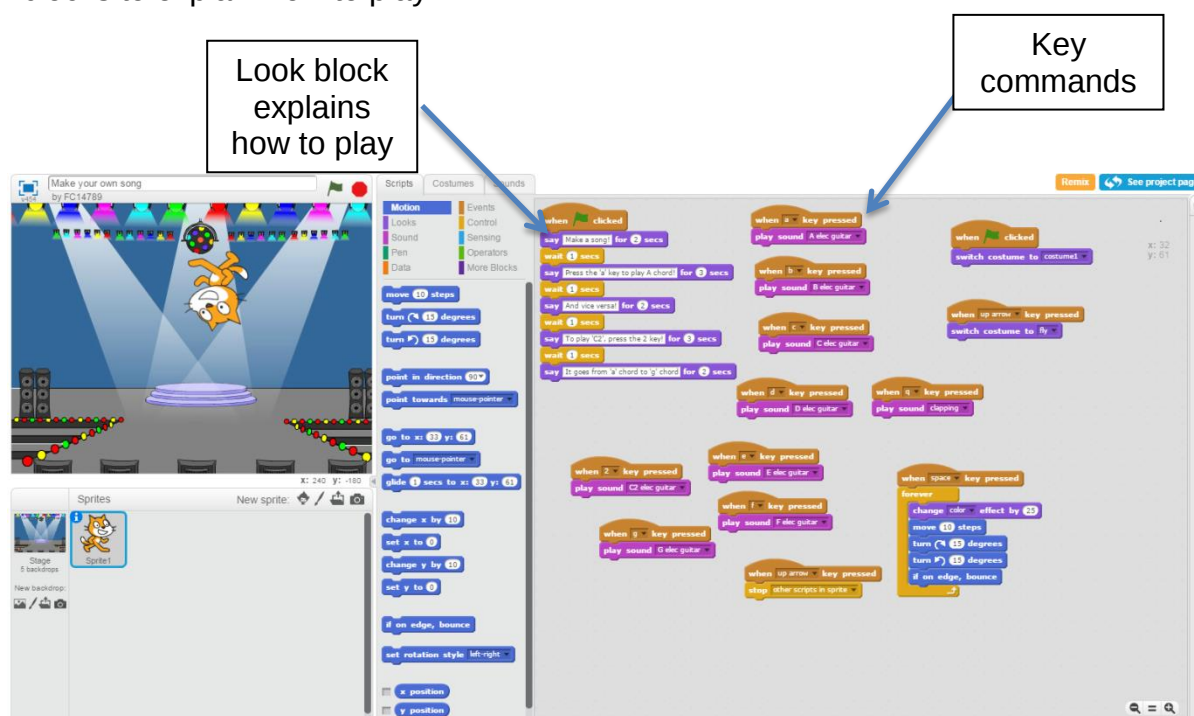


This code has used repeat loops and set the tempo. It uses broadcast blocks to add drum beats at certain parts of the song. It is coded to play 'Shape of You' by Ed Sheeran – a more complex song.



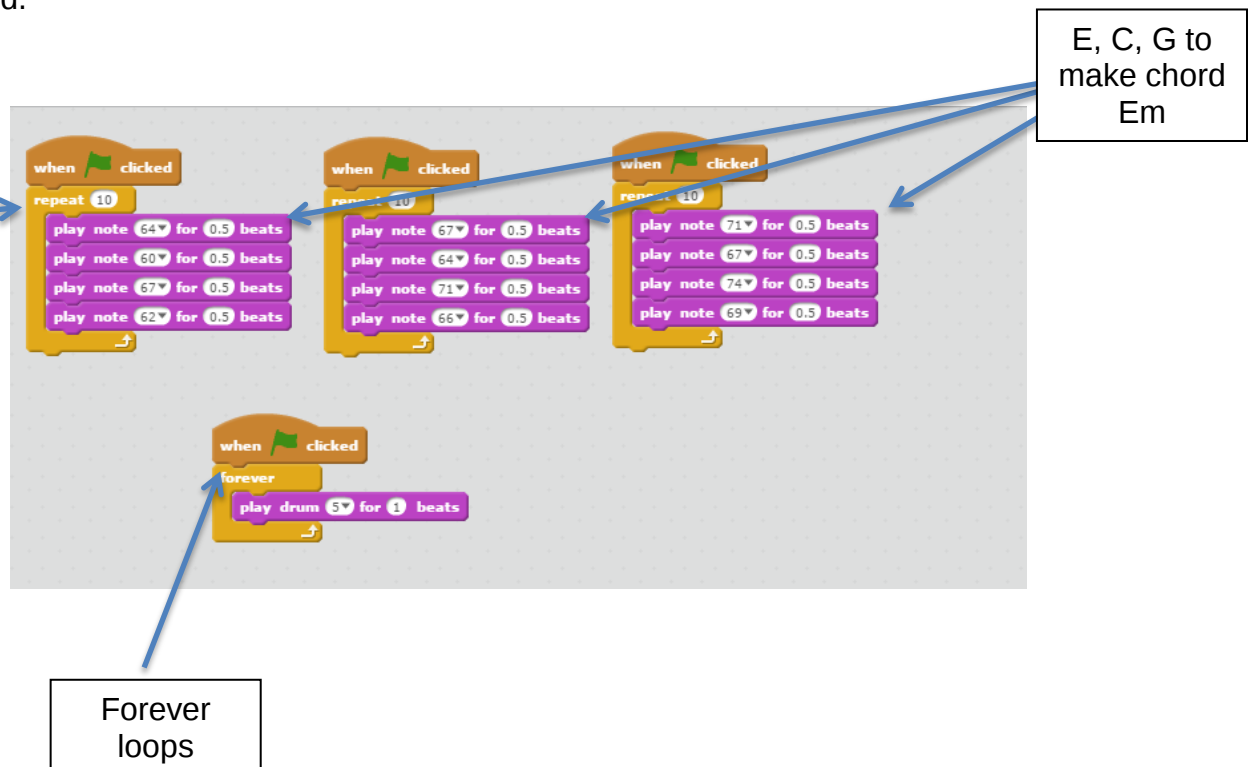
- After using auto-playing mode in Scratch. Learners should experiment with pressing the correct keys to play notes, turning Scratch into an instrument. This allows learners to compose their own tune.

This code has used key commands to play a note using the keyboard. It has used the look blocks to explain how to play.



9. Learners use what they have developed to further their project by composing their own song. The four notes/chords: *Em*, *C*, *G* and *D* are the most commonly used to compose popular music.

This code plays the notes simultaneously to simulate a chord. The first notes are E, G, B – which is the Em chord. It continues, playing C, G and D chords. A looped drum beat is also used.



Lesson 4

Learning Intention

- We are learning to use computational thinking.
- We are learning to compose a song using coding.

Success Criteria:

- I can compose a simple tune using code.

This is the opportunity for learners to showcase their musical Scratch compositions.

1. The format for the lesson is at the teacher's discretion:

For example:

- Peer-assessment
- Class Showcase
- Self-assessment
- Parent Exhibition
- Assembly

2. Learners can reflect on their experience – have they met the criteria? What would they improve? Would they do anything differently? What are the next steps?