STEMKIT 4SCHOOLS

Sound in Scratch

LESSON PLAN 2

Co-funded by the Erasmus+ Programme of the European Union

This project has been funded with support from the European Commission.

Table of Contents

1. Title of Lesson Plan

1.1 General information

1.1.1 Short description

In this lesson plan teacher will present the Scratch application, used to create projects containing media and scripts and programming language design for young people to explore, express themselves, and learn. The activities encourage exploration of key computational thinking concepts and key computational thinking practices.

It involves three key dimensions: (1) computational concepts, (2) computational practices, and (3) computational perspectives.

By studying activity in the Scratch online community and in Scratch workshops, young people will create their own interactive stories, games, and simulations, and share those creations in an online community with other young programmers from around the world.

Creative computing supports the development of personal connections to computing, by drawing upon creativity, imagination, and interests. Students will be more prepared for careers as computer scientists or programmers.

1.1.2 Learning objectives

Scratch is a programming language, created by MIT Media Lab, an open-source development environment that makes it easy to create interactive art, stories, simulations, and games. It is aimed at educating people with little or no programming experience, primarily children between the ages of 8 and 16.

This lesson introduces students to core computer programming concepts and computational thinking skills, exploring aspects of the Scratch programming environment.

It is a great way for kids to introduce programming to those with no previous programming experience. Students will learn to import images and sounds created in Scratch, using interactive art, stories, simulations, and games, a building paint tool and sound recorder as experimental activity.

The main learning objectives of this lesson plan are:

- concept and content understanding of Scratch 2.0. to inspire students to learn computer programming while working on personally meaningful projects such as animated stories and games.
- designing and performing an experiment or scientific investigation with collection of data, analysis and presentation of results, providing tools to solve the technology challenges of tomorrow
- familiarizing with Scratch used by schools in multiple disciplines (math, computer science, language arts, social studies).
- understanding basic structures of programming, using programming language.

1.1.3 Links to curriculum

Scratch targets younger users than the other two systems, focuses on self-directed learning, it includes tools to draw images and record sounds. It builds on the constructionist ideas, to help users make their projects personally engaging, motivating, and meaningful.

Scratch makes it easy to import or create many kinds of media (images, sounds, music); it was designed to invite scripting, provide immediate feedback for script execution, and make execution and data visible.

Students are able to see the progress of their learning visually in their world as a series of different physical projects and constructions.

The domains, subdomains, subjects/topics that this lesson plan can be linked to are:

- Science (Physics/Chemistry/Biology/Geology): scientific method, investigation, experimentation, analysis and interpretation of results
- Computer Science/Informatics: processing unit and peripherals, interfaces, programming language and main structures, coding
- Technology: electronics, open-source hardware and software, sensors, digital signal, circuits, single board computers
- Maths/Statistics: spreadsheets and basic statistics

1.1.4 Materials required

For this lesson plan (and for each student group) besides the STEMKIT console we'll need:

- Hardware and devices for the educator and each student

- PC, laptop, or tablet with an external mouse is recommended (most students find it easier to navigate in the game with a mouse instead of the touchpad)
- Headphones are helpful during gameplay (alternatively, the game audio can be turned down or off)
- Internet access is required for login and multiplayer
- Projector connected to a computer with Scratch open to display which blocks and scripts will be performed, and physical Scratch blocks (optional)

1.1.5 Duration

The duration of this lesson plan is estimated to be about 45-60 mins, i.e., one classroom hour.

1.2 Lesson plan

The activity of the student will concentrate on mastering the new concepts presented. Practicing with Scratch and exploring all the possibilities of the new concepts learned is essential for a robust learning process. The exercises proposed in the lesson are designed to reinforce the learn-by-doing approach.

The lesson helps students to develop and reinforce the knowledge and techniques learned in the tutorials, for the future workplace, building skills like collaboration, communication, critical thinking, and systems thinking.

The open learning environment gives students the freedom to experiment, encouraging creative self-expression and problem-solving.

1.2.1 Introduction to Scratch

The Computer Kit includes everything needed to introduce students to computer science, electronics, and coding. Build your own fully functional computer and explore STEAM.

Many kids have no concept about the components in their devices or how basic things like network connections and file systems work. The ultimate goal of computer building for kids becomes learning how all the physical parts interact and relate to what they see on the screen.

After studying the tutorials, the students will replicate on their own Scratch environment the activities presented during the lesson. They are encouraged to explore, beginning with

the environment presented in the tutorials, all the possibilities open by the newly learned concepts.

Students will be introduced to the computational thinking concepts of loops, events, and parallelism, becoming more familiar with the concepts of sequence, blocks in the Events, Control, Sound, and Looks categories + explore various arts-themed Scratch programs, create an animated music video project.

By completing this activity, students will create and play projects on tablet as well as their laptop and desktop computer. It will also be available on smartphones – although their small screen size could be a challenge. They will start to explore this creative diversity with a deep dive into animation, art, and music.

1.2.2 Preparation

Students will use the versions of Scratch 2.0, working with FireFox or Chrome with Scratch online. Navigate to Scratch 2.0 at scratch.mit.edu.

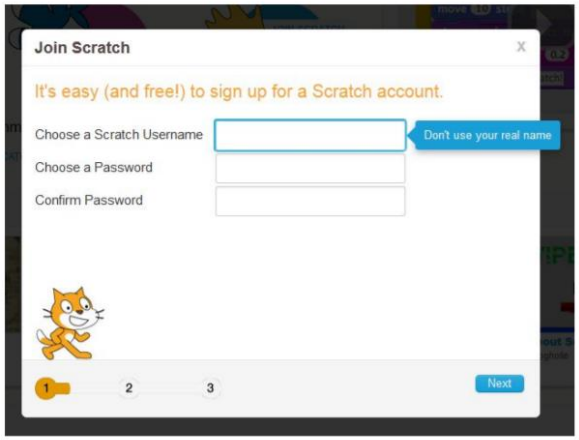

Fig.1. Navigate to Scratch 2.0 at scratch.mit.edu

Click “Create” to navigate to the Scratch programming environment. Students will utilize Scratch to create sounds, instruments, bands, or styles of music. They will build their own music-inspired Scratch project by pairing sprites with sounds to design interactive instruments.

1.2.3 Investigation

Scratch Activity: **Build-A-Band**

This activity is designed to help students to create a program that combines interactive sprites with interesting sounds, and develop greater fluency with sequence, loops, events, and parallelism.

First, gather as a group to introduce the theme and spark ideas. Ask each participant to say their favourite band. Teacher will show example projects from the Build-a-Band studio to guide students. He will give students time to create interactive instruments by pairing sprites with sounds.

Preview the tutorial: on YouTube, https://www.youtube.com/watch?v=XFf-cYUvvzU

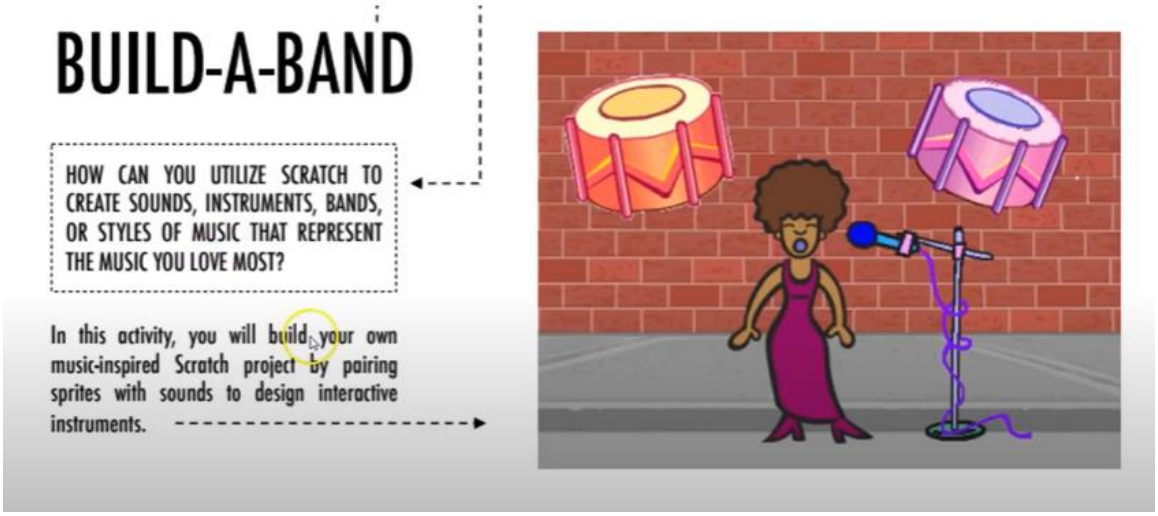

Fig. 2. Tutorial: https://www.youtube.com/watch?v=XFf-cYUvvzU

Teacher shows a variety of projects for ideas and inspiration.

Teacher will demonstrate the first few steps of the tutorial and students will see how to get started. He will encourage students to experiment with different ways to express sounds in Scratch by exploring other blocks in the Music category or using the editing tools within the Sounds tab.

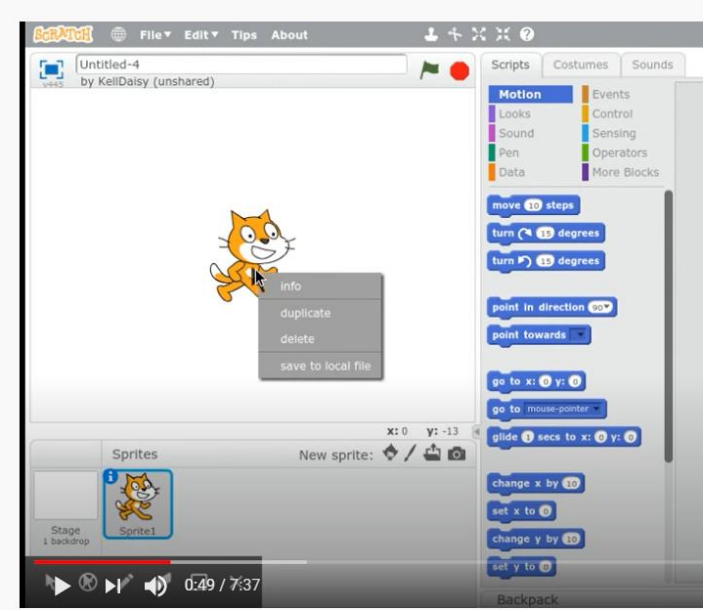

Fig. 3. Steps of the tutorial: scratch.mit.edu/name (Source: https://resources.scratch.mit.edu/www/guides/en/EducatorGuidesAll.pdf)

Teacher will allow students to demonstrate their bands to one another or let students walk around to interact with classmates' instruments.

Teachers can use a projector to show examples and demonstrate how to get started. He will ask students to think back on the design process by responding to the reflection prompts in their design journals or in a group discussion.

We recommend a gallery walk: have students put their projects in presentation mode and then invite them to walk around and explore each other's projects. Optionally, have students add their projects to the Build-a-Band studio or a class studio.

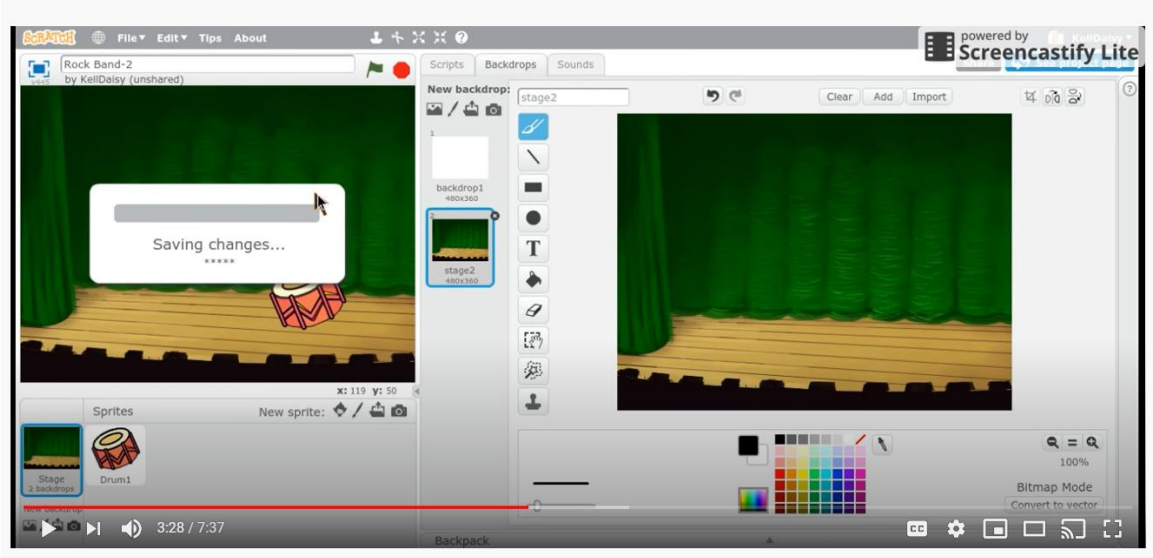

Fig. 4. Create interactive projects (Source: https://resources.scratch.mit.edu/www/guides/en/EducatorGuidesAll.pdf)

When someone gets stuck, teacher will connect them to another participant who can help.

Follow the online tutorial: *scratch.mit.edu/music*

Teacher will help participants feel comfortable trying different combinations of blocks and seeing what happens. Students can use the ideas and concepts from this workshop to create a wide variety of projects. They will choose a drum or other instrument, press a key to play a sound, create a rhythm and try changing the rhythm.

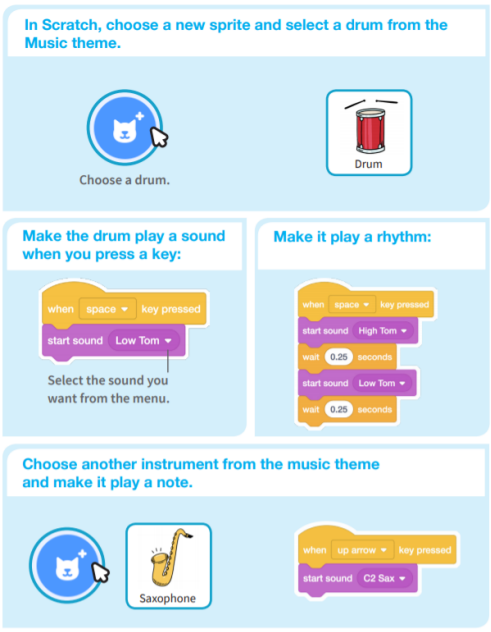

Fig. 5. Create interactive projects (Source: https://resources.scratch.mit.edu/www/guides/en/EducatorGuidesAll.pdf)

Teacher will provide resources, offering options for getting started; he walks around the room to see and listen to each other's' musical projects on their computers or laptops.

For example, Drum Beat: drag out the "play drum 48 for 0.2 seconds" block, use the drop-down arrow to select different percussion instruments, duplicate this block and change the length of the beats to create a rhythm.

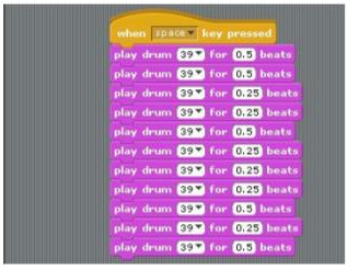

Fig. 6. Animated project (Source: ScratchManualTermTime20152016.pdf)

Teacher will support participants as they create musical projects, on their own or in pairs; he will suggest adding instructions so others know how to play the project, such as which keys to press, Add musical instruments, Play with musical patterns, Play a random note or sound, Add animation, Use beatbox sounds in a loop, Record short sounds to play.

Teacher will add instructions and credits to a project, (click the button: “See project page”). Students will give their project a title, add instructions and credits, and then click Share.

1.2.4 Conclusion

Scratch is one of the most widely used coding tools in schools. Scratch impact almost everything we do at school, for fun, in our personal and work lives.

The Scratch programming environment and language work together to create a system that is exceptionally quick to learn. Educators are integrating Scratch across many different subject areas and age groups.

In final phase we recapitulate what we did and how, which were the main steps, discuss any difficulties experienced.

Discussion Time

Students will discuss the day with their colleagues and reflect on what they have learned:

- ✓ Introduction to Scratch
- ✓ Getting Started with Scratch
- ✓ Sound with Scratch

Discuss the cross curricular use of Scratch in particular for music. Discuss aspects of Scratch that are relevant to numeracy.

Reflect as a group:

- ✓ What do you like best about the project you made or heard?
- ✓ What else might you like to add?

1.2.5 Follow-up exercise (optional)

As a follow-up to this lesson plan, we may proceed to the following exercise:

Scratch Activity:

1. Can you create a song using Scratch?

Students can search the internet for notes for different songs, when they are finished upload their song to the studio will share it with the group.

2. Battle of the Bands - with a partner or in a group.

1.3 References or Resources

List of useful references and additional resources.

Here are some useful references and additional resources related to this lesson plan.

- KAY, A. 2010. Squeak etoys, children, and learning. http://www.squeakland.org/resources/articles
- Resnick, M., Maloney, j., Monroy-Hernandez, 2009. Scratch: Programming for all. Comm. ACM 52, 11, 60–67.
- ComputerProgrammingInTheEnglishClassroom.pdf
- Maloney, J., Resnick, M., Rusk, N., Silverman, B., and Eastmond, E. 2010. *The scratch programming language and environment*. ACM Trans. Comput. Educ. 10, 4, Article 16 (November 2010), 15 pages. DOI = 10.1145/1868358.1868363. http://doi.acm.org/10.1145/1868358.1868363
- https://education.abc.net.au/home#!/media/1214681/intro-to-scratch-20
- https://scratch.mit.edu/studios/475523/
- http://web.media.mit.edu/~jmaloney/papers/ScratchLangAndEnvironment.pdf
- https://www.thomasbuxton.towerhamlets.sch.uk/blogs/year3/2017/11/17/year-3-computing-scratch-projects/
- http://scratched.gse.harvard.edu/guide/
- scratch.mit.edu/name
- ScratchManualTermTime20152016.pdf
- https://resources.scratch.mit.edu/www/guides/en/EducatorGuidesAll.pdf
- https://www.stem.org.uk/resources/elibrary/resource/35832/scratch-beginners