STEMKIT 4SCHOOLS

# Introduction to Scratch 2.0

## LESSON PLAN 1

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.
- ✓ Scratch 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 be introduced to the concepts of events (one thing causing another thing to happen) and parallelism (things happening at the same time) through performance, be able to explain what events are and how they work in Scratch, be able to explain what parallelism is and how it works in Scratch.

### 1.2.2 Preparation

**Introducing design of Scratch**

The key drivers for change have been the desire to extend the features of the editor and to make it available on a wider range of devices. Version 2.0 is based on Flash technology; most schools and coding clubs have developed their Scratch coding lesson plans based on version 2.0.

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.

There are two versions of Scratch, Scratch 1.4 and Scratch 2.0. Scratch 2.0 offline editor will be used for this course. But first let’s check out Scratch 2.0 online. We would recommend working with FireFox or Chrome when working with Scratch online. Internet Explorer can be problematic during Sign In. Navigate to Scratch 2.0 at scratch.mit.edu.

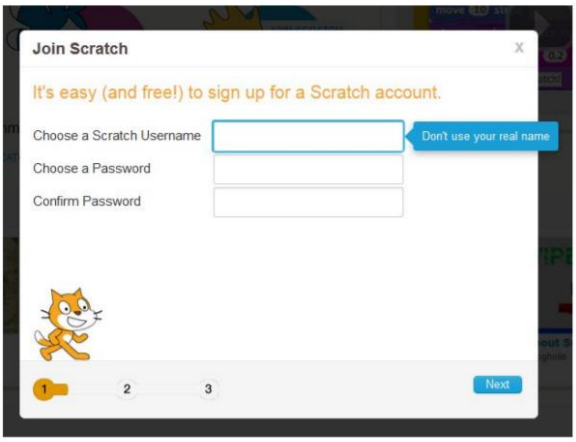


Fig.1. Navigate to Scratch 2.0 at scratch.mit.edu

Click “Create” to navigate to the Scratch programming environment. This is how to access the Scratch 2.0 online programming environment. Take some time to explore the scratch.mit.edu website (Click on Scratch, Create, Explore, Discuss and Help menu options)

The Scratch Project Editor is described under Tips, Getting Started, Map of Project Editor. Click on this to view the details of the Project Editor. Next, start the Step-by-Step, Getting Started with Scratch.

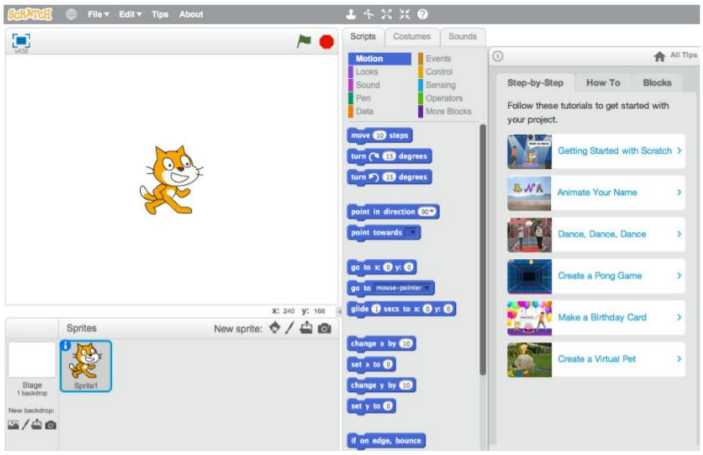


Fig.2. Navigate Step-by-Step introduction

Step through each of the 13 steps in the Step-by-Step introduction. This Step-by-Step guide introduces fundamental programming concepts and allows participants become familiar with the Scratch programming environment.


Fig.3. Click the HELP button at the top to get started. (Source: https://resources.scratch.mit.edu/www/guides/en/EducatorGuidesAll.pdf)

### 1.2.3 Investigation

Scratch Activity: Animate Your Name

This activity is designed to help students explore the computational concepts of loops, events, and parallelism, culminating in the design of personalized music videos. Participants will gain experience with coding as they animate the letters in their name.

First, gather as a group to introduce the theme and spark ideas. Ask each participant to say their name, and then have everyone in the group act out the shape of the first letter.

Preview the tutorial: scratch.mit.edu/name or vimeo.com/llk/name.

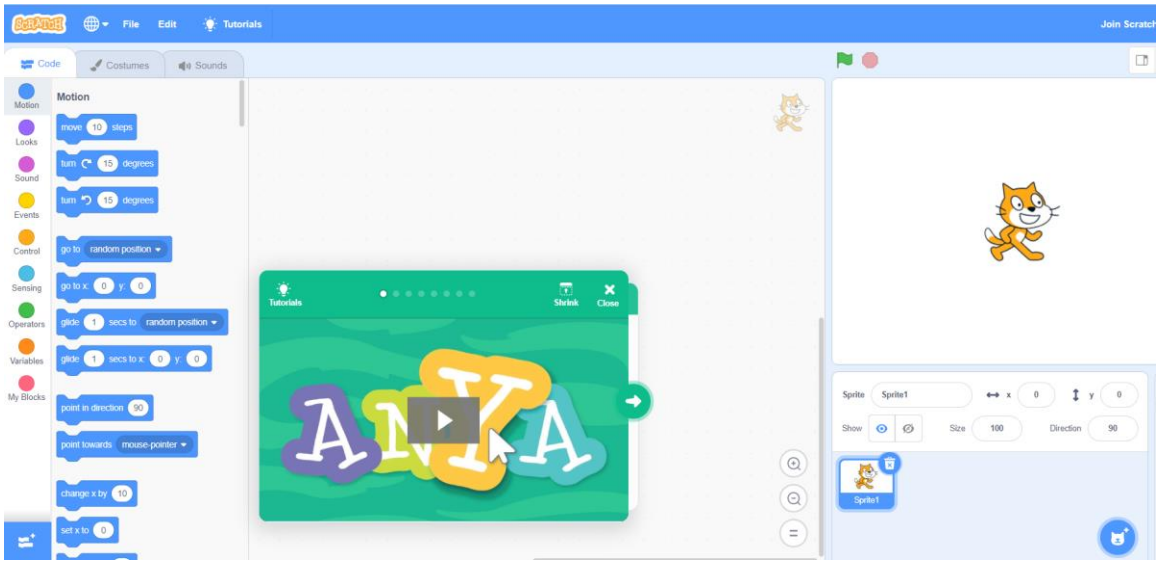


Fig. 4. Tutorial: scratch.mit.edu/name

Teacher presents the introductory video for the Animate Your Name tutorial. The video 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.

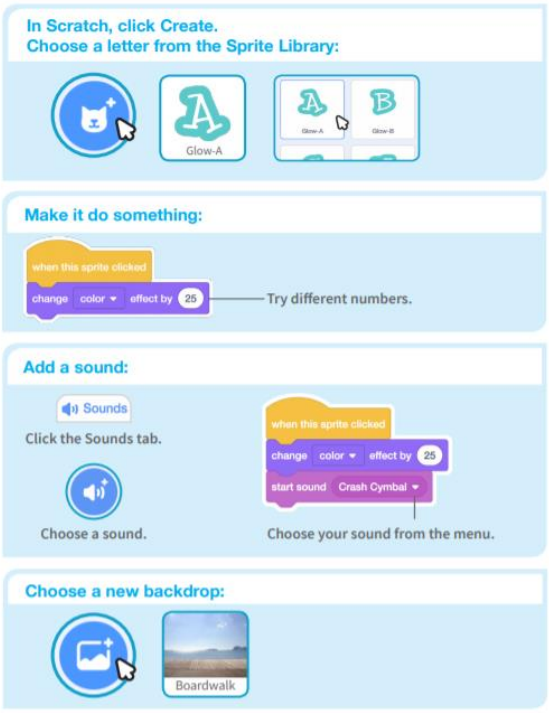

Fig. 5. Steps of the tutorial: scratch.mit.edu/name (Source: https://resources.scratch.mit.edu/www/guides/en/EducatorGuidesAll.pdf)

Teacher will print the Activity Cards - sets of Animate Your Name cards to have available for participants during the workshop. (scratch.mit.edu/ideas)

Teachers can use a projector to show examples and demonstrate how to get started.

The Animate Your Name tutorial shows participants how to create their own projects. Teacher will support participants as they create interactive name projects.

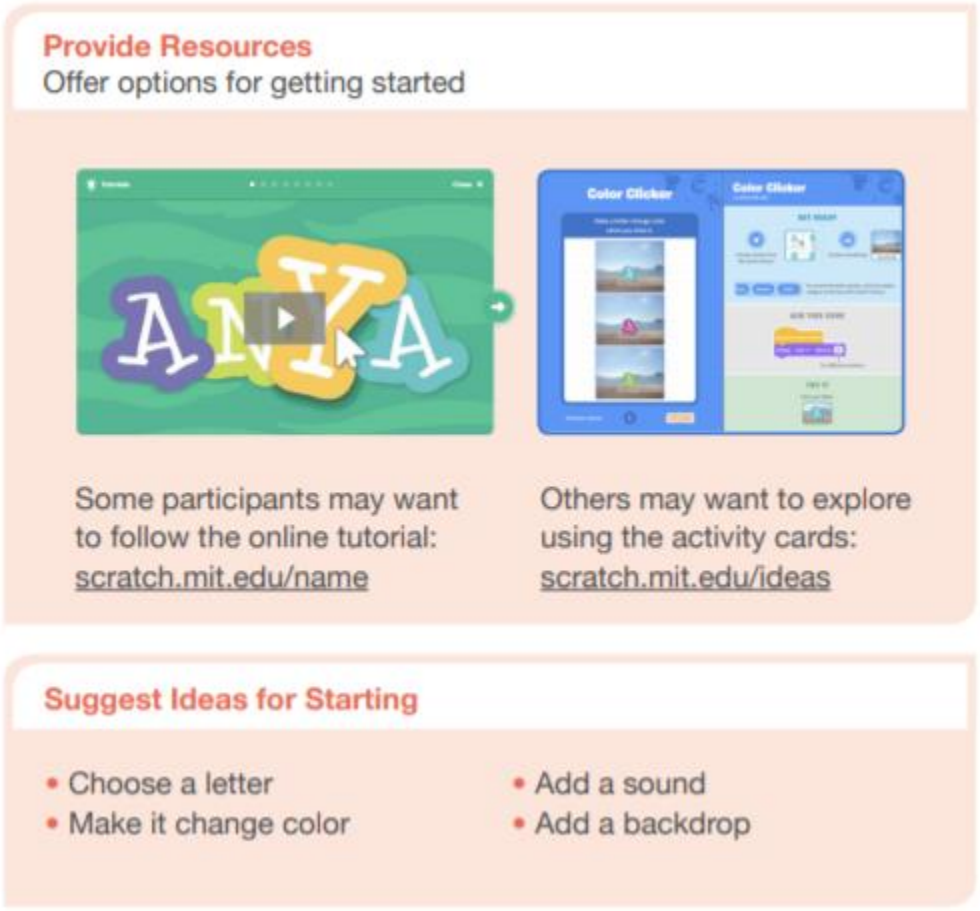


Fig. 6. Create interactive name projects (Source: https://resources.scratch.mit.edu/www/guides/en/EducatorGuidesAll.pdf)

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

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.

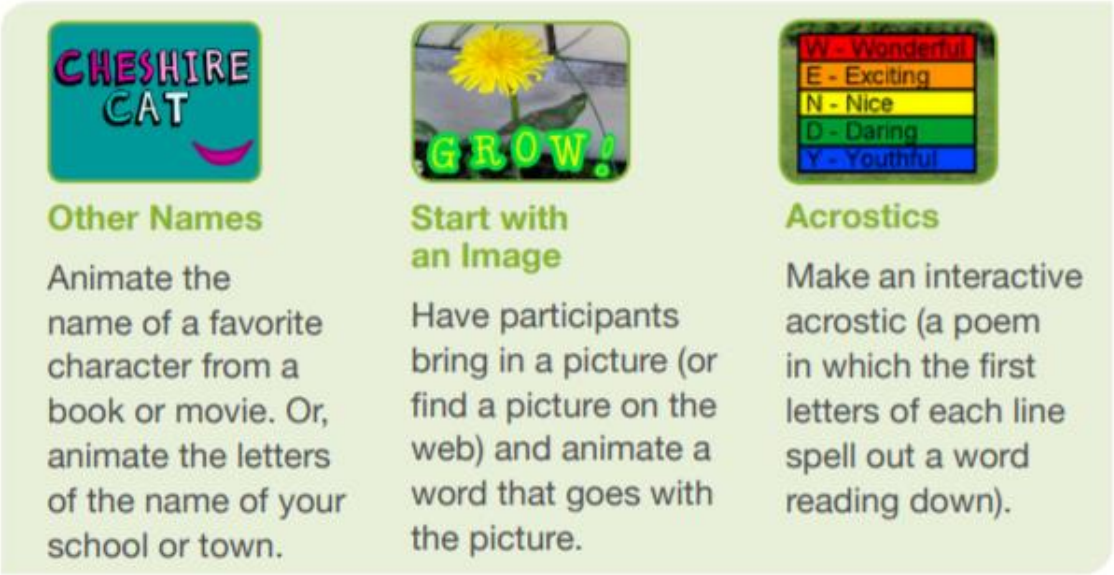


Fig. 7. Animated name project (Source: https://resources.scratch.mit.edu/www/guides/en/EducatorGuidesAll.pdf)

### 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.

The activity is designed to support familiarity and increasing fluency with computational creativity and computational thinking. Participants are now ready to complete some project work using resources on scratch.

Students should save their own copy to an appropriate area on the school network with an appropriate file name, e.g. adding their initials to the file name. Once a project is shared, another user can add comments.

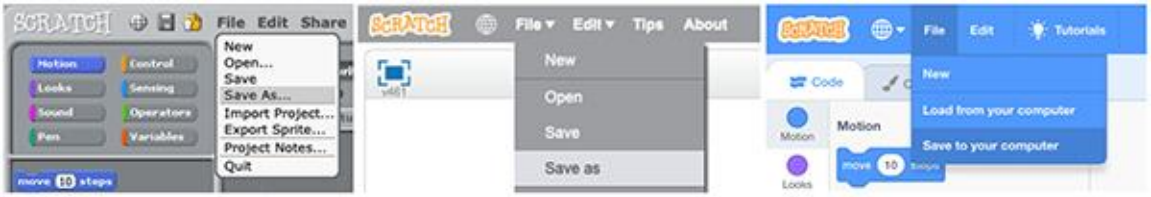


Fig. 8. Saving a copy of the Scratch file in Scratch 1.4, 2.0 and 3.0 respectively (Source: https://resources.scratch.mit.edu/www/guides/en/EducatorGuidesAll.pdf)

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

**Wrap-up:** What I learned

Three standard reflection questions to assess students' learning and surface their challenges and questions through student journaling and class discussion during the wrap-up of each lesson.

- What did you learn today that you didn't know before?
- What was most challenging?
- What questions do you have?

### 1.2.5 Follow-up exercise (optional)

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

*Scratch Activity: Reproduce the actions in the tutorial. Experiment with the size and direction of the steps and turns.*

## 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
- 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


Glossary of key terms

- Algorithm: A set of steps to accomplish a task to solve a problem.
- Code or computer program: A set of instructions that a computer can follow. For example, an app or a game, like Minecraft, is a computer program. The terms can be used interchangeably.
- Sequence: identifying a series of steps for a task
- Loops: running the same sequence multiple times
- Parallelism: making things happen at the same time
- Events: one thing causing another thing to happen
- Conditionals: making decisions based on conditions
- Operators: support for mathematical and logical expressions
- Data: storing, retrieving, and updating values