Wasif, Nadia

# Creative coding with Scratch. Report of the pilot project in primary school

revised and updated

*2025, 7 S.*

Quellenangabe/ Reference:
Wasif, Nadia: Creative coding with Scratch. Report of the pilot project in primary school. 2025, 7 S. - URN: urn:nbn:de:0111-pedocs-325036 - DOI: 10.25656/01:32503

*https://nbn-resolving.org/urn:nbn:de:0111-pedocs-325036*
*https://doi.org/10.25656/01:32503*

# Creative Coding with Scratch: report of the pilot project in primary school

Nadia Wasif

University College for Teacher Education, Vienna, Austria

**Abstract.** This paper is about a pilot project held in an elementary school in Vienna, Austria. The project tries to combine something old and something new. Old is the wish to teach a programming language, new is the way to do so. The teaching model selected here takes a different path. It puts children at the center, not the learning content. Children's needs can be dealt with individually to experience the development of program as a voluntary and amusing activity. But it is not only for the self-purpose, rather a tool to express themselves. The tasks created by the children lead automatically to the interest in problem solving itself and therefore to computational thinking [1]. Which is experienced as an easy and on the go acquired competence and is getting a self-evident part of life from now.

**Keywords.** Computational Thinking, elementary school, framework conditions, real creativity, implementation

## 1 Introduction

The Austrian Federal Ministry of Education gave the contract for the development of a course, the creation of a corresponding curriculum, the implementation and evaluation of the pilot project "Creative Coding with Scratch" in elementary school as a noncommittal exercise for pupils of elementary level II.

## 2 Analysis of the initial position

In Austria Coding Courses with Scratch for young children are mostly held by pedagogical lay persons as an after-school program, once a week for a period of approximately 3 months, by economically oriented companies in private institutions or event centers.

In secondary school it is carried out mostly as a project in public schools accompanied by qualified teachers. They use the programming language mainly as a tool for the preparation of topic-related presentations. This is mostly carried out in a time-isolated manner in blocks of one unit per day for two weeks in cooperation with class teachers [2]. Since now there are only a handful of projects, which teach Scratch in elementary school.

# 3 Preliminary considerations and framework conditions

Ahead of the project, it is necessary to clarify the conditions for the realisation and to prepare methodological and didactic considerations in advance, which specifically addresses the setting described here and creates a curriculum tailored to this.

## 3.1 Personnel framework

**Course leader**

The basic prerequisite is a great interest in the subject matter; understanding of the logical expression, which is generally common in programming languages; and the willingness to constantly develop and master the programming language "Scratch". In addition, pedagogical, didactic, and methodological knowledge as well as the preparation of a teaching concept and a curriculum for the implementation in the classroom are fundamental.

Equally necessary is the willingness not to teach in the usual sense, but rather as a coach on demand to advise and support, rather than anticipation.
Basic PC knowledge is a great advantage to be able to solve problems that are constantly present in the everyday handling of the children with the computer.

**Students**

The prerequisite for enrolment is the attendance of elementary level II.
Particularly important is the willingness of participants to embark on new learning processes and not only to explore the virtual world as a consumer, but as an active developer.
General prerequisites, such as teamwork, communication skills, creativity and concentration are essential and individual performance progress is based on that also.
A great advantage here would be basal PC experience, handling of the mouse as an input and control unit, as well as the knowledge of the most important command assignments of keys of the keyboard or the knowledge of the naming of these. In fact, none of the participating children had this knowledge.

The knowledge of English according to age is broadened in the course and the English designation for hardware is used as well.

**Technical support**

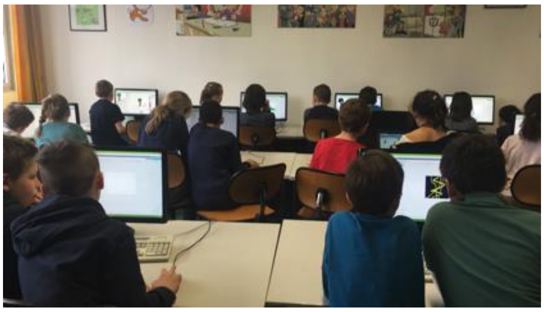
To ensure a smooth run, updates and maintenance of the devices are necessary. An expert is taking care of this. He has access to the premises and equipment and the software of the systems unhindered. He carries out these processes during the teaching-free period, as well as ensuring a trouble-free process by guaranteeing a functioning Internet connection.

### 3.2 Situative framework conditions

**Computer room**

The exercise takes place in a separate room, which contains three rows PC workstations. There are twelve workstations with 24 seats in the computer room, which gives the maximum number of participants of 24 children.
Unlike the class, special rules apply here. Since the long table rows are very close together and a variety of cables are attached to each unit, you can only move very carefully. For this purpose too, awareness must be created in the classroom and "wandering" of the children must be consistently replaced by "raising hand".

Since many children - in fact about a half of the participants as the previous parental survey showed - have never worked on a personal computer by their own since then, it is necessary to show the children the sensibility of the devices in a playful way and train them to be careful while using them.



**Picture 1:** Computer room with Scratch-kids

**Technical conditions**

Scratch is an opensource software, downloading is not necessary. Online programming is keeping the important, prominent features of the software such as publishing, sharing, and commenting on contributions, as well as the continuing work from any other PC worldwide. In addition, local servers are not accessed as a storage medium, and teachers' care is ensured by the offered and superior educational platform, Scratch.mit.edu.

Scratch requires a Linux operating system, OS X 10.4 or higher, Windows XP, Windows Vista, Windows 7, Windows 8 or Windows 10. Also, an actual Internet browser is required to display the correct plugins.
Already mentioned were monitor, mouse, keyboard and PC as well as an Internet access.

### 3.3 Didactic notes on the implementation

#### Motivation

In principle, the motivation is very high for all children before the beginning of the course. Since the PC is usually only used by adults in the everyday life of the children and therefore it appears particularly attractive. Participation in the course requires a proactive registration, so it is assured, that the students really want to program.

Scratch also has a high demand character through the layout and the child-friendly user interface. To bring in own ideas and thus themselves and to realise this in a completely independently created and programmed program, as well as to present, motivates the pupils very much and makes them proud.

At the beginning of the course, especially in younger children, it was noticeable that the motivation was strongly dependent on the openness of the individual and on the willingness to cooperate with the respective partner, who was completely unknown at this time.

Both very dominant and passive behaviour of the respective partner led to conflicts, due to which team changes had to be carried out. Again and again, the teacher has to check whether both team partners are involved in all processes.

The children are very fond of programming and can hardly wait for the next course date. With such high intrinsic motivation, there is no need for additional extrinsic motivation.

#### Group composition

Registration is voluntary and binding for the school year. In this case, eleven girls and thirty-five boys from up to five different classes participated. They did not know each other. So, it is very important for a good co-operation to create trust and confidence in a playful, age-appropriate way and strengthen the feeling of community. Researchers in computing education draw on both computer science and education - neither field alone is sufficient. [3]

The children work in pairs with the greatest possible diversity. For example, a boy from third grade programs with a girl from fourth grade. Mainly to get to know a different point of view, to see different paths of solutions and to support each other when working on the PC. Because of illness-related failures children, who want to do so for the duration of a unit, work alone or with another partner, to learn new perspectives and experience alternative approaches.

#### Timeframe

The course takes place once a week for the entire school year. In this case the number of participants with 46 children was that high, so it was necessary to divide into two groups, called the Thursday and Friday club. A teaching unit lasts 50 minutes, so larger projects are processed over two or more units.

Surprisingly, access to the PC or the use of Scratch is not permitted for most of the students at home, as shown in the results of the parental survey. Completion of programs can thus not be outsourced. This also had to be considered when selecting the course contents.

# 4 Implementation of the project

## 4.1 Concept

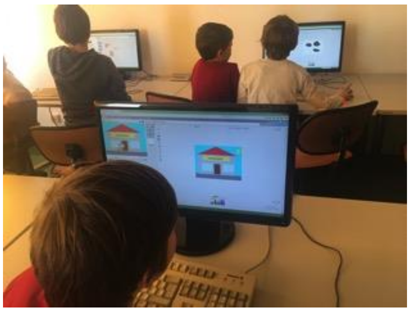
The actual learning outcome shows there are very large differences both regarding the children's experience in dealing with the PC, as well as in the approach to problems. The level of frustration tolerance and the willingness to accept alternative solutions also play an important role for the individual learning progress.

To address the individual needs of the young age group and on the other hand to achieve the real, fundamental intention of programming, the children work heuristically.
The access is original. Therefore, they do not program with maps, workbooks, tutorials, remixing or copying or completing existing programs.

All these resources specify a sequence of work steps to be performed, deliver building blocks, or even the entire script, thereby giving a strict path to be followed. In this way, these materials replace the problem-solving - which in the actual sense represents programming - and in addition they unify the codes. The child does not experience himself as a creative creator, but merely as an operator, fulfilling a clearly defined task either correctly or not. It is also a problem, that the written task, even if only used at a later stage for help or "control", has just one "right" solution. This does neither correspond to the lived nor the programmed reality.

In this way, the children remain unknown users, coding mutates to copying, and the "Creative" in "Creative Coding" is not considered.

It is characteristic for each programmer that the individual thoughts and approaches of each person in creating a code are as unique as a personal fingerprint and he develops over duration of its programming activity. This is an organic process that takes time, patience, and a fine-tuned set of tasks to help the children develop their skills. The ability of the participants increases the amount, as well as the difficulty of the presented building blocks and this leads to more complex codes.



**Picture 2:** Children create a background

## 4.2 Teaching and individual support

The lesson starts with a short, common information input, which introduces a new component or a group of blocks that has not been used yet. Together the children discuss the functionality and possible applications. They act simultaneously in the program and test the function immediately. First questions and insights are shared.
Afterwards, the partners discuss the idea they want to work on and develop an initial concept.

If teams do not have their own concept, the teacher advises in an individual discussion to concretise existing ideas, but deliberately does not present any topic. Many children insist on the uniqueness of their idea and react very sensitively if a neighbour- team wants to implement a similar one. Here too, the teacher must intervene and support the development teams.

Afterwards, each individual team works optimally, completely individually at the appropriate pace at the respective level.
From this point onwards the teacher can, if necessary, support or advise specific teams, but in any case, he can permanently return positive feedback.

Once a positive implementation has been presented, the following decisions are made:

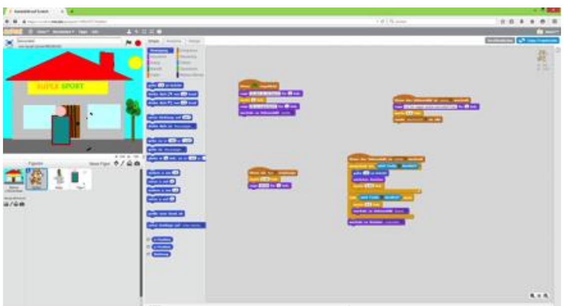
Are there further wishes for the implementation?
Is a modification such as the optimisation of the processes desirable?
Would the team like to deal with the created program even more intensively (additional tools, prolonged action, etc.)?
Would the team like to edit an analogous project?
Would the team, with the knowledge acquired in this unit, want to modify old projects?

At the end of each unit, a "reflection tour" takes place, in which the participants can view, test, and evaluate the work of others in real or virtual space. The enormous variety of implementations is always overwhelming, and the teams present and explain their ideas and products very proudly.

Discussions about occurring obstacles and the individual way to the solution are also taking places in the reflection. The children could bring up unsolved problems and solutions as well if they want to share outside their teams.

## 5 First conclusion

The project has been running and being observed, documented, and analysed for the duration of a school year, to develop a curriculum based on the data and results of the testing. On children's behalf the goal was to find a fun way to communicate skills directly through the medium of computer. Skills, that pick up every single child right where it is and on one hand thereby gives him the necessary tools and on the other hand the desired freedom to realise their own ideas in programmed Projects. The way learning objectives per unit have been fulfilled so far, not only in a basic, but also in a far-reaching way. Each of the twenty-three teams has programmed at least twenty different programs.



**Picture 3:** Screenshot

## References

1. Papert, S.: Mindstorms: Children, computers, and powerful ideas. Basic Books Inc., New York (1980)
2. Cusack, S. and others: Children Teaching Teachers: The New Grassroots for Advancing Coding in the Classroom, Lesley University Community of Scholars, Boston (2014)
3. Guzdial, M.: Education: Paving the Way for Computational Thinking, ACM, New York (2008)

Published 2017, revised and updated 2025