



ANALOG PROGRAMMING

Algorithmic Drawing



Exercise or
Warm-up



ca. 12 min.




6 years+



Prior Knowledge:
None



This exercise makes it clear that instructions have to be very specific to get the same result. The game can be teacher-centred or be played in teams of two. A person first draws a shape and then tries to instruct his / her partner through precise descriptions / instructions so that he / she can draw the same shape.

 This exercise is also great as a methodical training in terms of the processing of tasks, as the children, in order to be able to solve the tasks correctly, must first read all work instructions before they can use the command sequence or implement the algorithm correctly.



Competences

- > Understanding of algorithmic structures
- > Knowing the requirements for successful programming

Preparation for the Algorithmic Drawing

Required material

- > Colored pencils or felt-tip pens
- > Paper
- > Eraser
- > Sharpener
- > Print out of the template sheets
(optional)



1. Game preparation

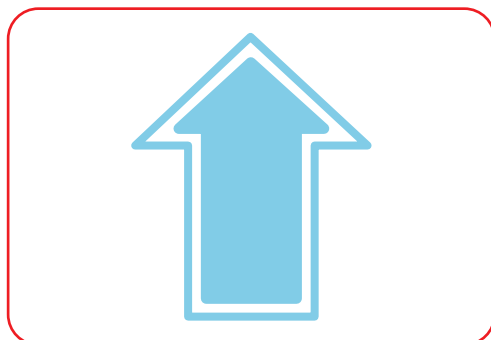
The implementation of the game can be controlled independently in groups of two or by a superordinate game master.

Step 01: Allocation of children

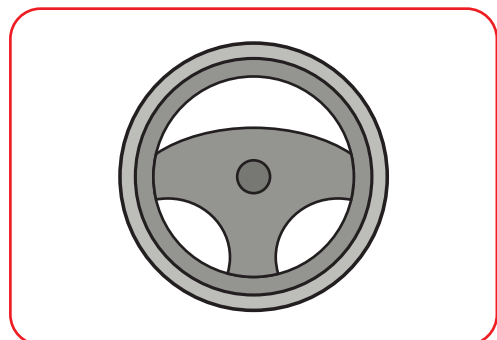
If work in small groups is preferred, the children should first be divided into teams of 2 or 3. More steering wheel symbols may be needed.

Step 02: Role allocation

If no superordinate game master is used, the children must first be introduced to the respective roles of navigator and programmer known from pair programming and their tasks. Further information on pair programming can be found in our OER materials.



- > Role navigator (1 child)
The arrow is your distinguishing feature.
Your job is to keep track and read out the command sequence.



- > Role programmer (min. 1 child)
The steering wheel is your distinguishing feature. You are responsible for the practical implementation and carry out the command sequence.

2. How to play

Purpose of the game: algorithms are sequences of instructions. In this game the children are asked to draw different shapes using a series of commands. Then the respective command sequences and the results are compared and discussed.

Step 01: The navigator thinks about a sequence of commands, first reads it out in its entirety and checks whether every programmer has understood the task. If there are any questions about the instructions, these will be clarified.

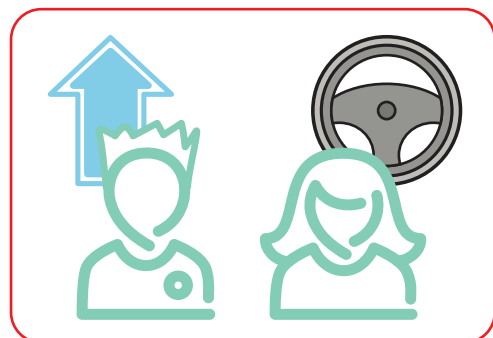
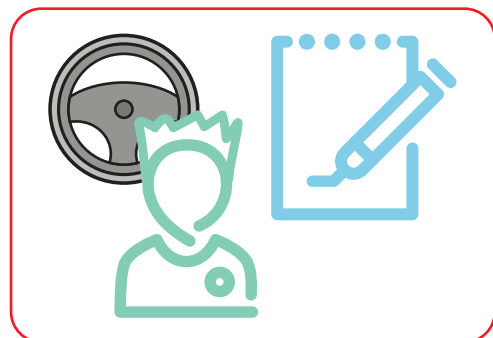
👉 You can also read the instructions aloud one at a time when the others start drawing. In any case, the entire command sequence should already be known.

Step 02: The programmer executes the command sequence and paints the correct picture according to their understanding.

Step 03: Once all team members have drawn a result, the navigation role is redistributed.

👉 Make sure that each child in the group was allowed to navigate once at the end of the game.

Step 04: Reflect what the game has to do with programming and algorithms.



This is what a command sequence and the resulting drawings could look like. Somehow a picture doesn't fit. Can you find out which one it is?

Step 01: Draw a square.

Step 02: Draw lines that don't touch.

Step 03: Use one color for the lines.



3. Evaluation/Reflection



As already described at the beginning, the reflection phase should be given a little time afterwards as an important element for processing what has happened.

Besides questions like:

- > „How did it go in your group?“
- > „Why did it work well / badly?“,

which mainly deal with the topic of the functionality and methods of the groups, one should above all pursue the following question:

„What does the game have to do with programming?“

Well a lot, even if it might not look like it at first. But in order to be able to give an answer to this question, we should perhaps first clarify a few basic things.

What is an algorithm and what is it made of?

As already described above, the term defines a sequence of instructions with which a specific problem can be solved. Pretty much every recipe or assembly instruction corresponds to this definition, if you like. In our case, the problem is that part of the group is supposed to paint a picture. But how should the picture be painted?

Work instructions / command sequences must be formulated so that this problem can be solved. If you now look at the results and discuss them with the children, two basic insights should be able to be discovered.

1. There is more than one solution.

2. The more clearly the instructions or the command sequences are formulated, the more uniform the images become. One speaks here of “well-defined” algorithms.

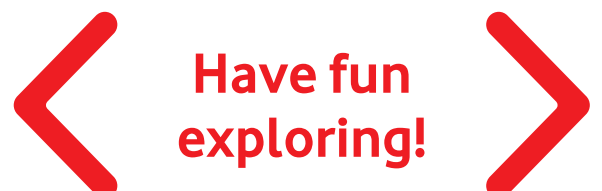
If the children have played the color code or decoding game beforehand, they should now be able to understand the basic principles of communication between humans and computers / machines.

1. Computers / machines need clear commands in order to be able to act.

2. Each command has a specific, clearly defined meaning.

3. A program consists of commands linked together. These command sequences are called algorithms.


As a supplement to the topic of algorithms, our toothbrushing game is ideal. There are plenty of other examples of algorithms in our everyday life.



Template

Example of a painting algorithm

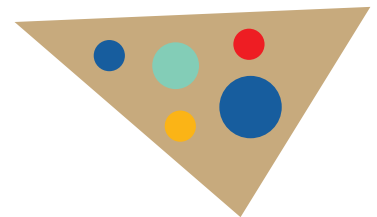
You can use these examples to help you get started with the game. The following four command sequences resulted in four forms. Which sequence of instructions fits which form?

 **Task:** Read through the four command sequences and combine them with the corresponding images.

Step 01: Draw a square.
Step 02: Draw lines that don't touch.
Step 03: Use one color for the lines.



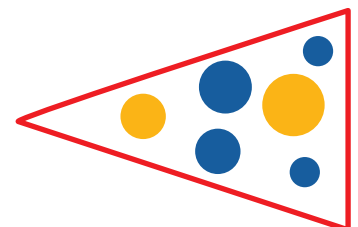
Step 01: Draw a triangle.
Step 02: Draw circles in three sizes.
Step 03: Use two colors for the circles.




Step 01: Draw a circle.
Step 02: Draw individual lines in it.
Step 03: Use multiple colors for the lines.



Step 01: Draw a triangle.
Step 02: Draw circles in it.
Step 03: Use four colors for the circles.



 **Task:** Make up your own sequences of commands and read them to your group. It can be more than 3 steps.

Step 01: ...
Step 02: ...
Step 03: ...

Template Pair Programming Symbols



Programmier - Symbol

The child, who is currently the driver, i.e. the programmer, receives as an identifier.

Navigator - Symbol

The child who is currently the navigator, i.e. the one who knows the direction, receives it as an identifier.