CodyColor KIT

Teacher trainer resource

Unplugged

Cross-disciplinary

Cross-age

To foster computational thinking skills

January 2026 | By Alessandro Bogliolo and the Italian EU Code Week Hub

Funded by the European Union

1. Discover the method

CodyColor is an **unplugged coding method** (without electronic devices) based on very simple rules. It can be introduced as early as preschool, yet supports challenging activities for lower and upper secondary school and adults.

As in many coding methods, the activity takes place on a grid or **chessboard**. The standard grid size is 5×5 squares.

Every coding activity requires a clear separation of roles between **programmer** and **ideal executor**. The programmer writes the program and the executor faithfully executes it. Both adhere to the rules of a rigorous programming language.

The basic element of any programming language is the **set of elementary instructions**, which defines both how they must be represented (in textual languages each instruction is represented

Funded by
the European Union

by a conventional word) and how they must be executed (i.e., which elementary action they correspond to).

In chessboard-based coding, the size of the squares determines the step length, while orthogonality determines the in-place rotation angle: 90° to the right (clockwise) or to the left (counterclockwise). This makes it possible to be rigorously precise in an intuitive way (once clarified) without having to specify movement lengths or rotation angles.

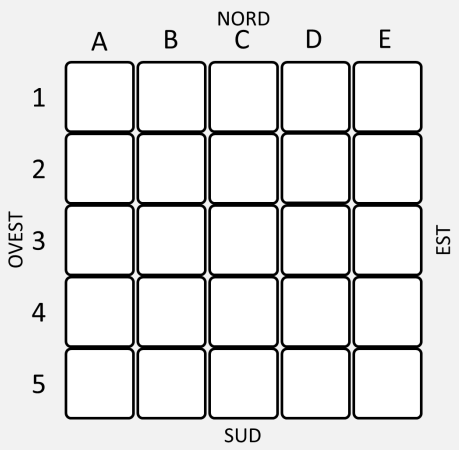
Generally, programming languages, including unplugged ones, require that instructions be composed in **sequence** to form the **program**, which determines the order in which the executor must carry them out.

To execute an instruction, the executor must first **read it**—i.e., become aware of it (if the programmer prevents the robot from seeing the instruction, it cannot execute it)—and **interpret it**—i.e., recognize it as one of the conventional instructions associated with a specific action (if the programmer used instructions outside the repertoire, the robot would not know what to do).

Since ideal executors do nothing without a program, it is as if they instinctively read the instructions and, again instinctively, after executing one instruction, move on to the next. I call it instinct only to indicate they are designed to behave this way; they do not need the programmer to tell them to read the instructions, otherwise we would fall into a paradox.

The chessboard

The 5×5 chessboard can be tabletop or floor-based. The tabletop version can simply be drawn on a sheet of paper. Its construction is so straightforward that it needs no explanation; however, anyone wishing to use the chessboard from the CodyRoby kit and doesn't already have it can print the last page of this guide.

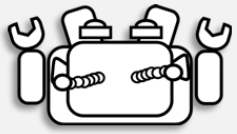


Funded by
the European Union

It is useful to label the chessboard's rows and columns, and assign to the 4 sides the names of the cardinal points. To do this, choose a point of view, assign the cardinal points as if looking at a map, and label columns (letters) and rows (numbers) as shown, starting from the top-left corner (NORTH-WEST).

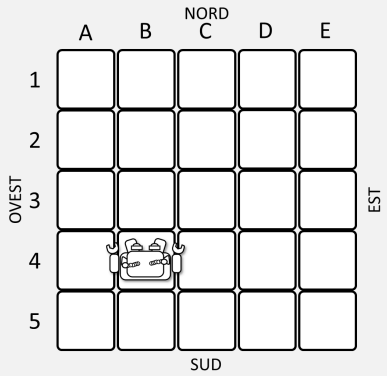
The robot

We call the executor the **robot**. For **tabletop** activities, the robot placed on the chessboard is a **pawn**, which can be made in any way, provided it clearly shows the direction it is facing. A player acting as the executor interprets and executes the instructions by moving the pawn. Those who wish to use **Roby** as the pawn will find it to print and cut out on the last page of this guide.



For **floor** activities, the robot is a **person** who directly plays the role of executor.

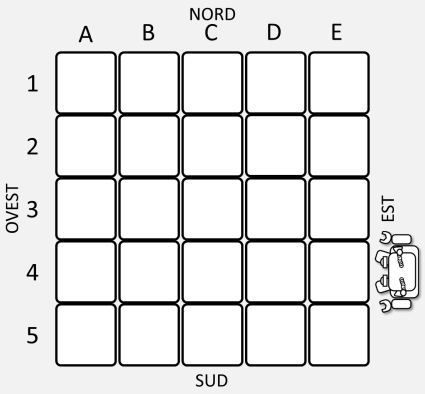
In all cases, the robot must take on a well-defined position within the chessboard. This means it must be clearly placed on a single square (not straddling two) and oriented along the chessboard's axes, facing a precise direction. This allows its position to be described using the column and row coordinates and the cardinal point it faces. For example, **B4 North**, as shown.



Only by knowing the executor's starting position can the programmer predict the effect of the program being written.

In activities where the robot's starting point is outside the chessboard, the robot will still assume a starting position consistent with the chessboard—namely, facing the chessboard and aligned as if it were on a square that extends the chessboard's grid.

Funded by
the European Union

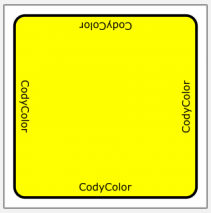


The CodyColor instruction set

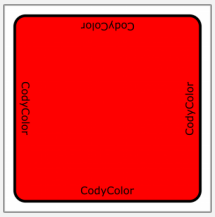
Two features make CodyColor's rules particularly simple:

1. **The elementary instructions are only three and are represented by colors that are very easy to distinguish, with no words written on them** (so reading ability is not required, there are no language barriers, and the instruction is recognized at a glance).
2. **The instructions are placed directly on the squares where they must be executed** (so there is no need to memorize them or read them elsewhere).

Yellow tile: turn left. The yellow tile represents a 90° in-place rotation counterclockwise and is read "turn left."

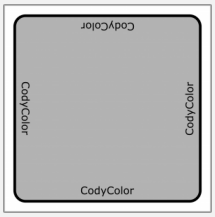


Red tile: turn right. The red tile represents a 90° in-place rotation clockwise and is read "turn right."



Grey tile: go straight. The grey tile represents the absence of in-place rotation and is read "go straight."

Funded by
the European Union



The tiles have dimensions consistent with the squares of the chessboard, since they are placed inside them. They can therefore be the same size—in which case it is as if the entire chessboard were colored—or smaller, in which case they are placed at the center of the square to which they are assigned.

The tiles can be made from colored paper sheets, cutting out squares of the desired size, but anyone who wants to use the **CodyColor kit** directly will also find printable, cut-out tiles on the last page of this guide (See Appendix).

Start of execution

When the robot is started, it automatically moves to the square in front of it to look for an instruction. It does this whether it is already on a square with a colored tile, on an empty square, or even outside the chessboard. This automatic advancement corresponds to what we previously called the robot's "instinct" to proceed in reading the program. If the robot is already on a colored tile at the start of execution, it simply advances to the square in front of it, without performing the rotation that might be indicated by the color it is on. To avoid misunderstandings, **it is best that the robot's starting point be free of tiles or contain a grey tile** that does not require any rotation.

Execution dynamics

The robot **reads the instruction** when it arrives on it. At that point it **interprets it** and **executes it** by rotating in place to the left (if the tile is yellow) or to the right (if the tile is red), or by not rotating (if the tile is grey).

After executing the instruction it is on, the robot checks that there is a colored tile on the square in front and moves onto it to read the next instruction. This action is implicit, i.e., not encoded by any instruction, and we call it **"passage to the next square."** If the square in front is empty, the robot stops and waits for a colored tile to be placed there (if foreseen by the activity dynamics).

Funded by
the European Union

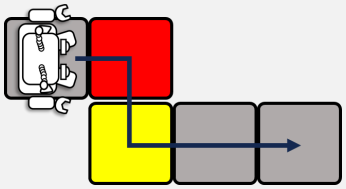
If, after executing an instruction, the robot is facing outside the chessboard, **execution ends** and the robot exits the chessboard.

Programming

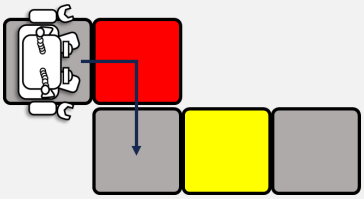
The programmer writes the code by placing the colored tiles on the chessboard. In this way, they lay out the instructions directly along the path that the robot will follow while executing them.

Put like this it may sound simple, but it isn't—neither for the programmer (who must account for the colors when constructing the path to ensure consistency), nor for the executor (because the path may not be obvious when the chessboard is full of colors).

This is an **example of a simple, easily recognizable, correctly programmed path.**

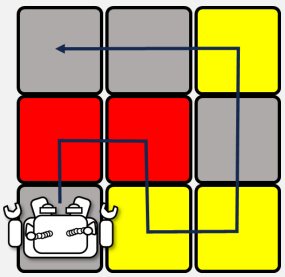


This is an **example of a programming mistake on a seemingly recognizable path**. If you try to follow it while respecting CodyColor's rules, you will realize you end up off-path.



Funded by
the European Union

This is a **correctly programmed path, but not recognizable at a glance**, because it forms a compact area of all colored tiles.



Funded by
the European Union


CodeWeek
Build your digital future
Funded by
the European Union