

CodyColor KIT

Teacher trainer resource

Unplugged

Cross-disciplinary

Cross-age

**To foster computational
thinking skills**

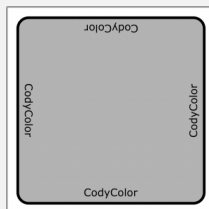
Dear teachers, educators, and families,

The “CodyColor KIT” is a training resource for you all.

*It allows you to explore, learn, and experiment with the **CodyColor** method. CodyColor is an **unplugged coding method**, based on very simple rules, which can be introduced from **early childhood education** and at the same time offers challenging activities for **lower and upper secondary school** students, as well as for **adults**. Its strength lies in the **combination of accessibility and depth**: the same core mechanics can be scaled and adapted to different levels of complexity, making it an effective tool for developing **computational thinking** across ages and contexts.*

*The training resource is being designed in close collaboration with **Alessandro Bogliolo**, the Italian EU Code Week Ambassador. He is the scientific reference person, as well as the original author of the CodyColor method. This ensures coherence with the pedagogical rationale of the method and strong engagement from the existing community of practitioners. Additionally, the new resource provides an opportunity for **international collaboration**.*

Start exploring the kit,



Go straight ahead; it's a grey tile ;)

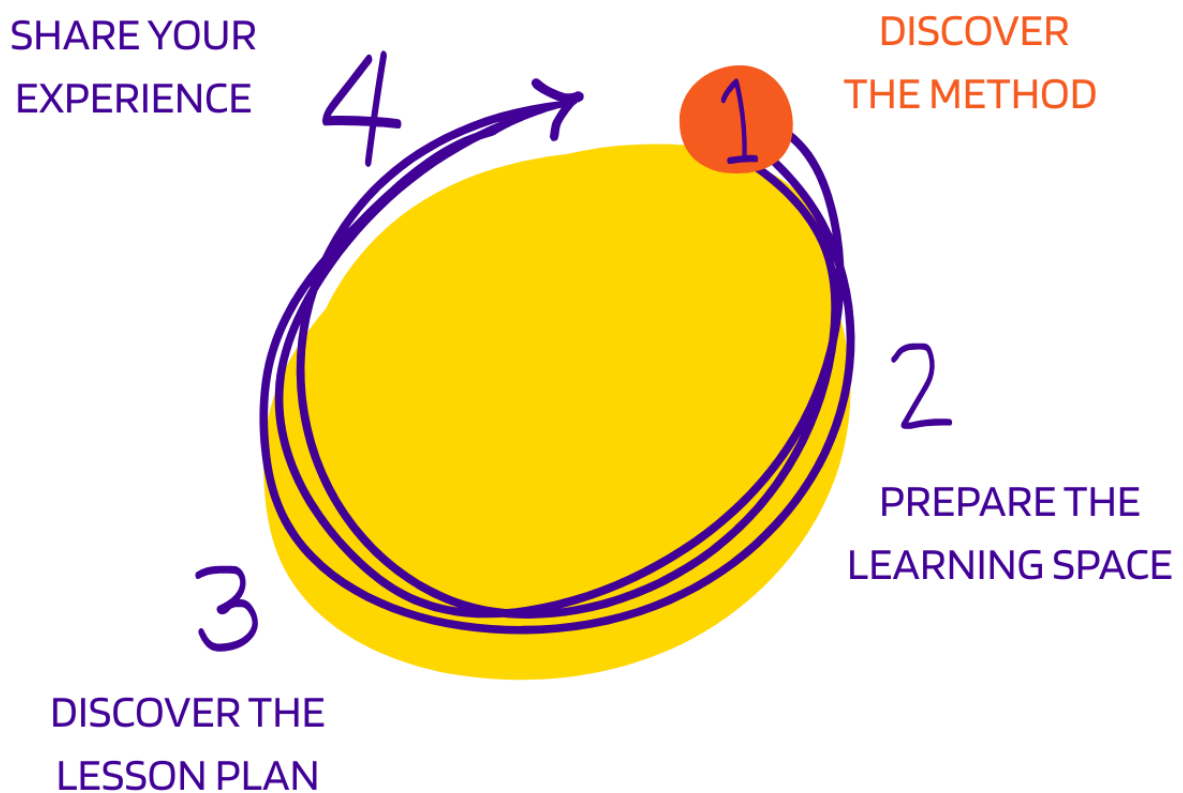
This resource is developed in collaboration with [CodyRoby Academy](https://codyrobby.com/)

How to use the toolkit

In this toolkit you can discover the CodyColor method through sequential and easy-to-follow steps. The objective is to guide you from learning the methods to applying them in the classroom, from learning to teaching with the CodyColor lesson plans as scaffolding materials.

Let's start!

1. DISCOVER THE METHOD
2. PREPARE THE LEARNING SPACE
3. DISCOVER THE LESSON PLAN
4. SHARE YOUR EXPERIENCE



| Table of contents

How to use the toolkit.....	2
1. Discover the method.....	4
The chessboard.....	5
The robot.....	6
The CodyColor instruction set.....	7
Start of execution.....	8
Execution dynamics.....	8
Programming.....	9
2. Prepare the learning space.....	11
How to build the floor chessboard.....	11
How to obtain the CodyColor tiles.....	12
3. Discover the lesson plan: Pas-de-deux.....	13
How to play in two separate rooms.....	16
4. Showcase and share your experience on the EU Code Week platform.....	18
APPENDIX.....	20



| 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

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.

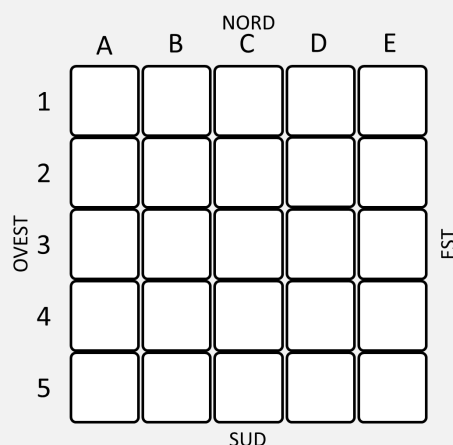
Generally, programming languages, including unplugged ones, require that instructions be composed in **sequence** to form the **program**, which determines the order in which the executor must carry them out.

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

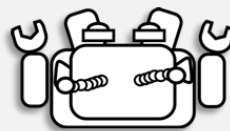
The 5×5 chessboard can be tabletop or floor-based. The tabletop version can simply be drawn on a sheet of paper. Its construction is so straightforward that it needs no explanation; however, anyone wishing to use the chessboard from the CodyRoby kit and doesn't already have it can print the last page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



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).

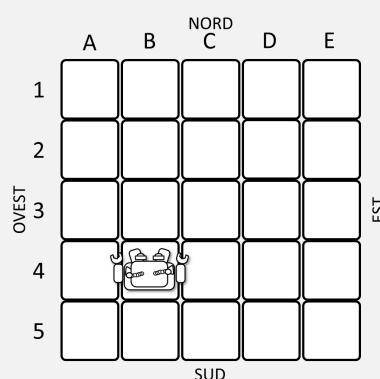
The robot

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.



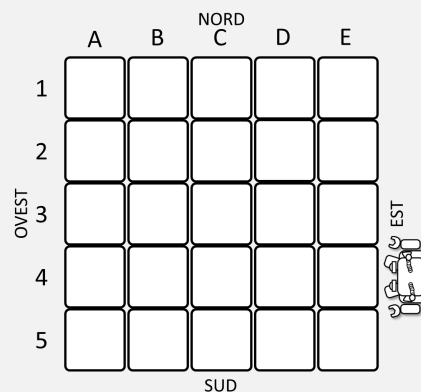
For **floor** activities, the robot is a **person** who directly plays the role of executor.

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.

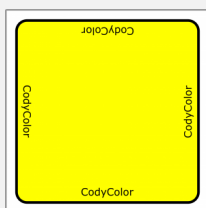


The CodyColor instruction set

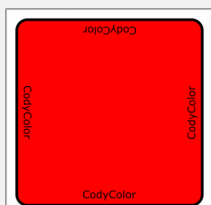
Two features make CodyColor's rules particularly simple:

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).

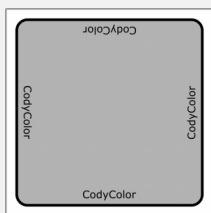
Yellow tile: turn left. The yellow tile represents a 90° in-place rotation counterclockwise and is read "turn left."



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."



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).

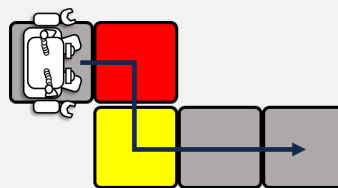
If, after executing an instruction, the robot is facing outside the chessboard, **execution ends** and the robot exits the chessboard.

Programming

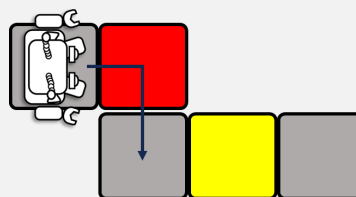
The programmer writes the code by placing the colored tiles on the chessboard. In this way, they lay out the instructions directly along the path that the robot will follow while executing them.

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).

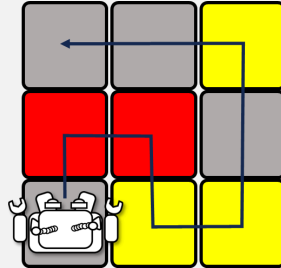
This is an **example of a simple, easily recognizable, correctly programmed path**.



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.



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





| 2. Prepare the learning space

How to build the floor chessboard

The floor chessboard, recommended for offering the activity in **preschool** and **primary school**, or for doing it in a group in a more engaging way, must be sized so that even an adult can stand inside a square (it is recommended to use a space of **2 m by 2 m**, with squares **40 cm per side**). To draw the floor chessboard you can use adhesive tape to mark the lines that form the grid: **6 horizontal lines** of 2 m, 40 cm apart from each other, to delimit the 5 rows; **6 vertical lines of 2 m**, 40 cm apart from each other, intersecting the previous ones to delimit the 5 columns.

Naturally, you can use for this purpose specific materials already available (such as EVA mats) or learning environments already equipped with floor chessboards of suitable dimensions.

How to obtain the CodyColor tiles

Print the yellow, red, and grey tiles found in the Appendix, one per A4 sheet. If you think you will use them often, laminate them: they will last longer and you will be able to reuse them in multiple classes and in different editions of Code Week.



| 3. Discover the lesson plan: Pas-de-deux

The challenge in brief:

Pas-de-deux is a competitive challenge between two teams (A and B) on a 5x5 chessboard. The teams place the pawns (robots) at opposite corners of the board (A1 and E5) and, taking turns, place CodyColor tiles one by one to build the longest possible paths.

The team that composes the longest path for its robot before it leaves the board wins.

Target:

From primary school onwards

Learning objectives:

- Develop computational thinking
- Strengthen spatial orientation and rotations
- Experiment with strategies, compare hypotheses, and make group decisions

- Train collaboration, communication, and respect for rules.

Initial setup:

The game begins with the 5×5 board empty, with only two grey tiles at A1 and E5. Team A places its robot at A1 facing South, and Team B places its robot at E5 facing North. Thus, at the start, both robots face empty squares and wait for their teams to choose which colored tile to place there; they then move onto it and execute the corresponding instruction according to CodyColor's rules.

- Teams take turns alternately: A, then B, then A again, and so on.

Execution:

On their turn, the team chooses a colored tile (yellow/red/grey) to place on the square in front of their robot.

After placing the tile, the robot automatically moves forward, reads and interprets the color, and, if necessary, performs the rotation (in case of a yellow or red tile).

Special cases:

1. If, after executing the instruction just placed by its team, the robot has in front of it a square that already contains a colored tile (placed either by its own team or by the opposing team in previous turns), it must also move onto that tile and execute it according to CodyColor's rules. The turn ends only when the robot stops in front of an empty square.
2. If, at the start of a turn, the robot of the team to move has in front of it a colored tile placed by the opposing team in the previous turn, it proceeds by moving onto that tile and any others it encounters along the way, respecting CodyColor's rules. Only when the robot arrives in front of an empty square does it stop, waiting for its team to make its move by placing a tile.
3. In cases 1 and 2, the robot travels across multiple squares in the same turn. Note that cases 1 and 2 can also occur together if the board is already very full; this typically happens in the final phases of the game.

When, after executing the instruction associated with the color of the square it is on, the robot is facing outside the board, it exits without waiting for the next turn and ends its team's game.

End of the challenge and scoring

The challenge ends when the robots of both teams leave the board. This means the game continues even when only one of the two teams is still in play, because the victory is decided

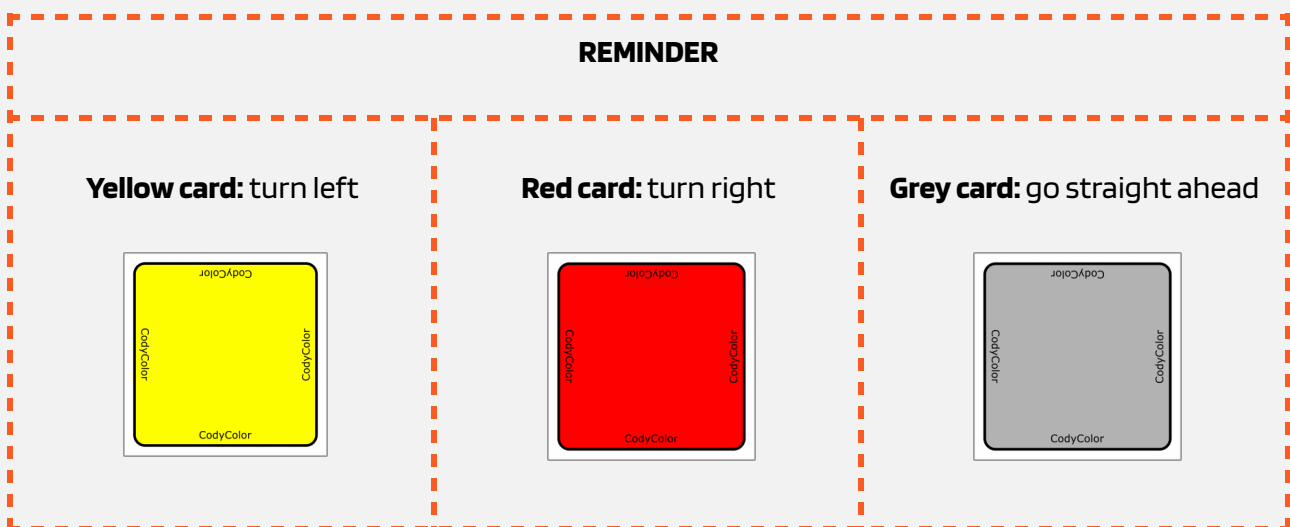
only at the end by checking which team has built the longest path (not necessarily the team that stayed in play for more turns).

At the end of the challenge, both teams reposition their robots at the starting positions, without removing the colored tiles from the board, to compare the length of the programmed paths. The comparison is carried out by moving the two robots simultaneously, one step at a time, to the rhythm set by everyone clapping.

On the clap, both robots move to the square in front of them, look at the color of the tile they are on, execute the corresponding instruction, and wait for the next clap. The two teams count the steps out loud as long as their robots are in play. Each team stops counting only when its robot leaves the board.

Each team earns as many points as the steps taken by its robot before leaving the board.

The challenge consists of three rounds whose scores are summed. In the event of a tie, additional tie-breaker rounds are played.



How to play in two separate rooms

The “Pas de deux” challenge can also be played remotely, for example, between two classes in different schools (even in different cities or countries) and communicating via an instant messaging chat (e.g., WhatsApp).

In both classrooms, the 5×5 grid must be set up by strictly following the arrangement of row and column labels, as well as the notation of the cardinal points used as directions. This is essential so that positions can be communicated unambiguously by specifying their coordinates.

In each of the two classrooms, both the team’s own robot (played by a student) and the opponent’s robot, called the “avatar,” will move on the grid. It may be useful to have students wear distinctive vests or other markers to differentiate their own robot from the opponent’s avatar and to make the experience more engaging and intuitive.

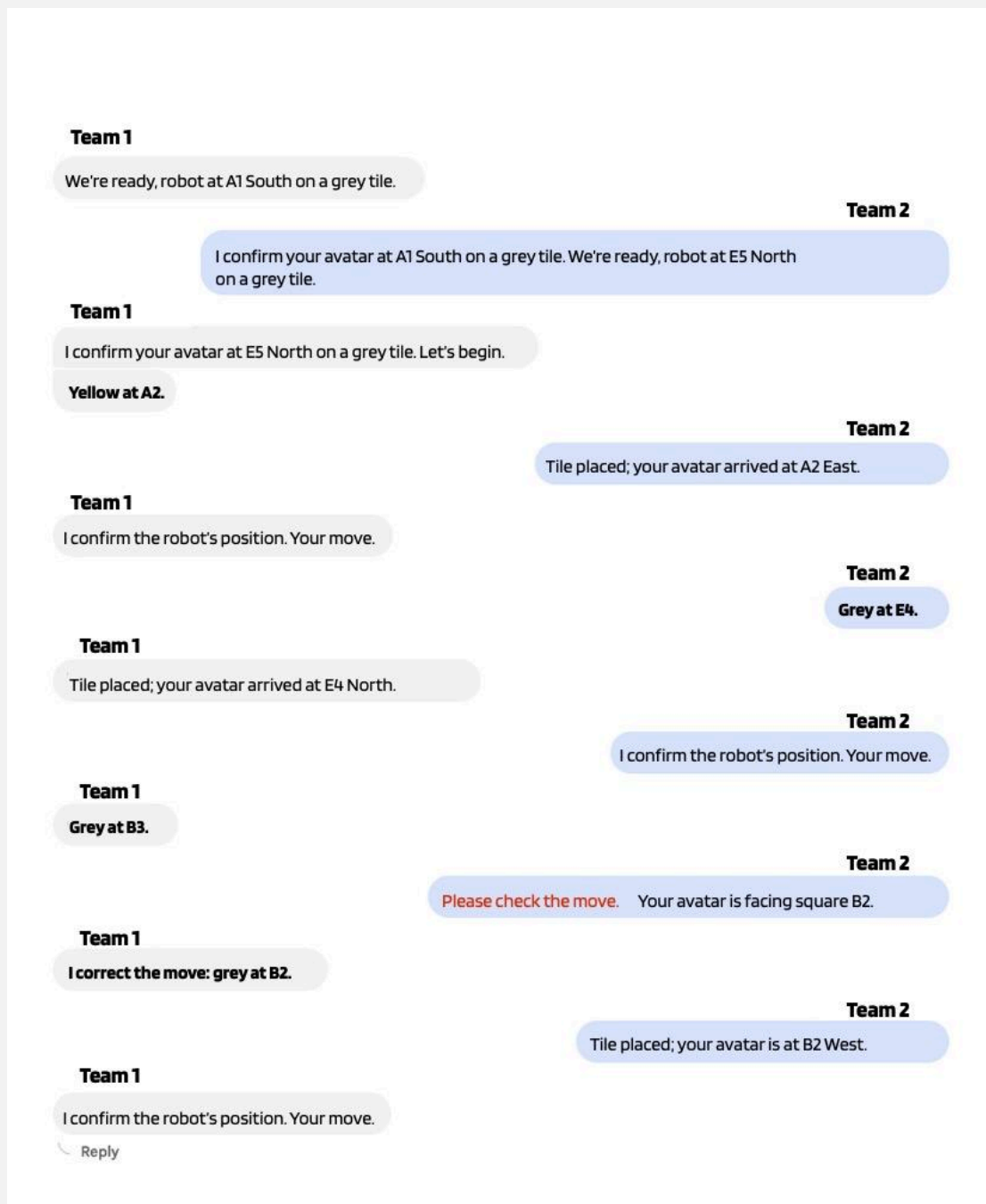
Initially, according to the rules of *Passo a Due*, both grids will be empty, with grey tiles placed in A1 and E5. During the game, the grids in the two classrooms will always be kept consistently updated, both in terms of the positions of the two robots and the colored tiles placed on the floor.

Game dynamics:

1. Before starting, the teachers agree on who starts from square A1 and who from E5, and on which team makes the first move.
2. The teacher of the team whose turn it is communicates their move via chat to the teacher of the other team with a text message specifying the tile color and the position (e.g. *yellow A2*). In the meantime, the local team places the tile and the robot moves according to the *Pas-de-deux* rules.
3. The team that receives the message via chat announces the instruction chosen by the opposing team, places the communicated tile on their own grid, and moves accordingly to the person acting as the opponent’s robot avatar. At this point, the teacher of the team not taking the turn sends a text message specifying the final position reached by the avatar (for example, *A2 EAST*).

The teacher of the team whose turn it is checks that the avatar position communicated by the opposing team matches the position of their own robot and confirms this to the other teacher. Only at this point does the turn pass to the other team.

For greater clarity, please refer to the figure illustrating the exchange between the two teachers. Let us imagine that the two teachers have already decided which team starts first (Team 1) and the respective starting positions: Team 1 in A1 facing South, Team 2 in E5 facing North. This is a possible example of a chat exchange.





4. Showcase and share your experience on the EU Code Week platform

Every time you bring a CodyColor challenge to your classroom—during a school event, a team-building activity, or a professional development course—[register the activity on the European Code Week map](#). Each organizer will receive a certificate of participation for their commitment and will contribute to a campaign raising awareness of the importance of computational thinking skills.

If you would like to connect with an international group of enthusiastic teachers, join the Facebook group for EU Code Week teachers! To take an extra step and collaborate with other schools in your country or across borders, join the Code Week 4 All challenge.

TIPS

You can use and customize the following text for registering your activity [on the European Code Week map](#) and showcase your experience

Title: "CodyColor challenge in class ___name you class___"

Description:

*"A floor grid, colored tiles, and plenty of strategy: today the class worked in teams with CodyColor, an unplugged coding activity that trains computational thinking, collaboration, and forecasting skills. As the 'robots' moved square by square, the boys and girls learned to discuss choices, correct errors, and build the best strategy together." **#CodyColor #CodeWeekEU***

| APPENDIX

Your cut-out tiles ready to print

