

3.5 CODING IN NATURE: PROGRAM A HUMAN ROBOT

This 90-minute STEAM project introduces 1st graders to the basics of coding through a movement-based activity in a natural setting. The lesson is structured into 2 clear stages, starting with an outdoor session where students act as "programmers" and "robots." By giving step-by-step verbal commands to guide their partners around stones and branches, students learn to "debug" their instructions in real-time when the "robot" take a wrong turn. The second stage takes place in the classroom, where students translate their physical experience into visual "route maps" using symbols and arrows. It is an interdisciplinary approach that connects math, physical activities and digital literacy, showing children how to solve complex problems through simple, logical steps. Also it teaches kids that mistakes are just puzzles waiting to be solved.

Basic Information Sheet

Age group	Student number	Duration	Number of stages	Subjects connection and relevance
1 st grade students	Whole class (max. 20-25 students)	min. 2 x 45 minutes	2 stages	Mathematics, Environmental Studies, Physical Education, Visual Arts, Digital Literacy

STEAM Event Summary

In this activity, teachers introduce first-grade students to the basics of algorithmic thinking through a playful outdoor game. Students take turns acting as a "programmer" and a "human robot," using simple verbal commands to guide movement in a natural environment. The activity is designed to be highly interactive, movement-based, and age-appropriate, supporting both cognitive and physical development.

SPIRIT Skill Focus

- Flexibility
- Critical thinking
- Problem-solving

National curriculum Link and Objectives

- Mathematics: Counting steps, understanding directions, spatial reasoning
- Physical Education: Motor coordination, obstacle course navigation
- Visual Arts: Drawing “route maps”
- Environmental Studies: Recognizing natural obstacles, observing surroundings

Real-world STEAM Connection and Problem definition

Have you ever tried to guide someone who cannot see exactly where to step? In this activity, students face the challenge of guiding a “human robot” through a natural environment full of obstacles such as stones, branches, or bushes. The goal is to give clear, short, step-by-step instructions so the robot can safely reach a target. When instructions do not work, students try again and improve them just like programmers fixing mistakes in code. At the same time, children learn to move carefully in nature and become more aware of their surroundings.

Materials at a Glance

Outdoor Materials:

- Natural obstacles such as stones, sticks, branches, bushes, or tree roots (found on site)
- Markers to define routes and destinations (e.g., pebbles, ribbons, cones, or sticks)
- Optional stopwatch for timed challenges

Classroom Materials:

- Paper or drawing sheets for route maps and algorithm planning
- Command sheets or instruction templates with arrows and symbols
- Pencils, coloured pencils, or crayons
- Notebooks (optional)

Visual and Support Materials (optional but recommended):

- Printed command cards or icons (e.g., move forward, turn left/right, jump, stop)
- Example “program” or route drawing shown by the teacher
- Visual arrows or symbols for students who need additional support

Indoor/Outdoor and Classroom layout



- Phase 1 – Outdoor (schoolyard or park): Human-robot game in a natural environment, focusing on movement, listening, and giving clear instructions.
- Phase 2 – Classroom: Drawing the route and translating actions into simple, visual “programs,” followed by reflection and improvement (debugging).

Notes on Research-based approach and integration

This activity is based on discovery learning and problem-based learning. Students learn by doing: they test ideas, notice mistakes, and improve their solutions through experience. The interdisciplinary STEAM approach supports active engagement, cooperation, and deeper understanding in a playful way. The STEAM elements are integrated as follows:

- Science: Observing natural obstacles and understanding the environment.
- Technology: Using step-by-step instructions as algorithms.
- Engineering: Planning efficient routes and overcoming challenges.
- Art: Creatively designing paths and visualizing solutions.
- Math: Sequencing steps, patterns, and logical reasoning

Lesson Flow Overview (linked to details below)

Stage	Title	Brief Description	SPIRIT Skills	Duration	Link to Details
1st	Human-Robot Game Outdoors	Students work in pairs. One is the “robot,” the other the “programmer.” The programmer gives commands (e.g., “take two steps forward, turn left, jump”), which the robot performs. The obstacle course is made of natural elements (sticks, stones, bushes). The teacher demonstrates the game with a volunteer acting as a robot.	• Problem-solving • Critical thinking • Flexibility • Creativity	45 minutes	STAGE 1 – Human-Robot Game Outdoors

2nd	Paper-Based Algorithm Creation and Debugging	Students return to the classroom and individually or in pairs draw the route they followed. They add commands (arrows, symbols, numbers). The teacher introduces simple algorithm ideas such as steps in order (sequence), repeating steps (repetition), and fixing mistakes (debugging) using drawings and symbols. Students test their “programs” by acting them out again. This stage helps students connect their physical experience from the outdoor game with visual and symbolic representations.	• Creativity • Problem-solving	45 minutes	STAGE 2 – Paper-Based Algorithm Creation and Debugging
-----	---	---	---	------------	--

Tips and Tricks for dealing with challenges (for the whole STEAM program)

- Use visual aids (arrows) if students get stuck
- Encourage shy children through role changes or team programming
- Teach that mistakes are part of programming (debugging = finding and fixing errors)
- Use timed challenges to maintain momentum.

Difficulty level tailoring notes

The activity allows flexible adaptation to group needs:

- **Beginner learners:** Pre-made command sheets with visual aids (arrows, icons); robot follows only simple commands
- **Advanced learners:** Students formulate and explain their own commands
- **Expert learners:** Students independently design paths, debug errors, and suggest alternative solutions

STEAM Program Debriefing and Reflection Questions

- What was the most interesting part of the activity?
- Which command was hardest to follow/give and why?
- What did we do when we made a mistake? How did we fix it?
- What did you learn about clear communication and teamwork?
- If you could do the task again, what would you do differently?
- How could we make this game even more fun?
- Can you think of similar commands you could try at home or on the playground?

3.5.2. Stage details and Activities Flow

STAGE 1 – HUMAN ROBOT GAME OUTDOORS

How does this develop particular SPIRIT skills

- **Problem-Solving:** Children plan and refine step-by-step commands to guide the “human robot” through natural obstacles, practicing practical solutions.
- **Critical Thinking:** They analyse the environment, anticipate challenges, and sequence commands logically to achieve the goal.
- **Flexibility:** Students adjust strategies and instructions when obstacles or mistakes occur, learning to adapt.
- **Creativity:** Designing routes and visualizing solutions encourages imaginative thinking and innovative approaches
- **Curiosity, Sense of Wonder and Openness:** Exploring the natural environment fosters observation, inquiry, and engagement with nature.
- **Valuing People and Nature:** Collaboration with partners and awareness of natural elements promote respect, empathy, and responsibility
- **Creativity and Curiosity:** designing paths and exploring natural elements encourages imaginative thinking and engagement with the environment.

What do we want to achieve regarding SPIRIT skill development (student understanding and/or behaviour)?

Through this activity, students are expected to develop the following SPIRIT skills:

- **Problem-solving and Critical thinking:** students plan routes and give precise directional commands, then analyse and correct mistakes (“debugging”).
- **Flexibility and Resilience:** they adapt strategies when obstacles or errors occur, learning to persevere.
- **Valuing People and Nature:** confidence and verbal expression improve through role-switching and partner collaboration, fostering respect and teamwork

Materials and tools needed for implementation

- **Math:** Directions, counting steps
- **Environmental Studies:** Identifying natural obstacles
- **PE:** Motor coordination, obstacle navigation
- **Digital Culture:** Basics of algorithmic thinking

Preparation notes

- Teacher should pre-arrange the obstacle course
- Ensure every pair tries both roles.

Guided Questions

- How can you plan your steps to overcome obstacles?
- What can you do when your first plan doesn't work?
- Can you think of new ways to guide the robot through the environment?
- How can you support your partner and work together to reach the goal?

Stage Debriefing Questions (Optional)

- Which command was hardest to follow?
- What was funny or surprising about the game?

Tips and Tricks for dealing with stage challenges

- Encourage shy students to start as the robot
- Introduce a timed challenge: who can guide their robot fastest?
- Use visual arrows if verbal guidance is difficult.

STAGE 2 – Paper Based Algorithm Creation and Debugging

How does this develop particular SPIRIT skills

- Creativity: designing personal route plans and “programs”
- Problem-Solving: identifying and fixing mistakes.

What do we want to achieve regarding SPIRIT skill development (student understanding and/or behaviour)?

Through this activity, students will develop problem-solving skills by:

- Recording simple algorithms: Translating actions into clear, step-by-step instructions.
- Debugging and redesigning steps: Identifying mistakes and revising their plans to overcome obstacles.
- Applying creative solutions: Thinking flexibly and inventively to solve unexpected challenges during the activity

This progression ensures that students follow through from planning to testing to improving, reinforcing structured problem-solving strategies.

Academic/Curriculum Objective connection

- Math: Sequences, counting steps
- Visual Arts: Drawing paths, using symbols
- Digital Culture: Recognizing algorithmic structures

Materials and tools needed for implementation

- Paper, coloured pencils, direction arrows
- Instruction templates
- Visual aids for inspiration

Preparation notes

- Show an example of a simple drawn “program”
- Assist with using symbols (e.g., → = move forward)

Guided Questions

- How would you draw your program?
- What happens if the robot gets lost?
- Which command would you change and why?

Stage Debriefing Questions (Optional)

- What did you like about your own program?
- How did you solve it when something didn’t work?

Tips and Tricks for dealing with stage challenges

- Don’t aim for perfection: encourage learning from mistakes
- Provide templates to help students who need more support.

Annexes

Pilot implementation (detailed example with lesson/activity plan), attachments

The activity was piloted with a group of first graders (ages 6–7) in an outdoor schoolyard setting. The following materials were tested:

1. Command Sheet Template:

Icons used: → Step forward ↶ Turn left ↷ Turn right ↑ Jump ■ Stop
Students can draw or cut and arrange these icons in sequence.

2. Robot Path Drawing Sheet:

Blank template for students to draw the route (The teacher can pre-draw or leave it to students' creativity)

3. Teacher Guide:

- Example commands (e.g., take two steps, turn right)
- Typical errors and fixes
- Differentiation strategies (e.g., visual aids)

4. Student Evaluation Sheet (Simple feedback questions with pictograms):

- What did you like most? 😊
- What was difficult? 😞
- What was funny/surprising? 😲

5. Word and Picture Cards:

Printable command cards for vocabulary and command sequence activities →, ↶, ↷, ↑, ■

6. Obstacle Course Layout Plan:

- Made from natural elements (stones, branches, bushes)
- Three difficulty levels (short-straight / winding / obstacle-avoiding)
- Teacher instructions included