



## ERGO guidebook for teachers

| Date        | Version | Description of Changes | Author |
|-------------|---------|------------------------|--------|
| May 2026    | 1.0     | First Emission         |        |
| August 2026 | 1.1     | Second Emission        |        |

## **Abstract**

*This guidebook provides actionable guidance for teachers transitioning to or operating within online and hybrid STEAM education environments, with a specific focus on robotics and creative coding. Developed as part of the ERGO project, it addresses the unique challenges of digital teaching, such as hardware limitations, student engagement, and cognitive load. The guide outlines pedagogical principles for effective online instruction, including the 'pedagogy before technology' approach, visual scaffolding, and structured interaction, complemented by practical techniques and teacher personas that illustrate these strategies in action. Furthermore, the document offers a comprehensive review of good practices for empowering girls in STEAM, supported by evidence-based metrics and international initiatives. It concludes with an analysis of essential simulation tools and resource repositories, equipping educators with an operational toolkit to create inclusive, engaging, and purposeful learning experiences for all students.*

# Table of Contents

## **Chapter 1: Good Practices in Online STEAM Education**

- 1.1. Introduction
- 1.2. Core Challenges in Online STEAM Education
- 1.3. Key Principles for Effective Online STEAM Teaching
- 1.4. Practical Techniques for Interactive Online STEAM Teaching
- 1.5. From Theory to Practice: Teacher Personas
- 1.6. Making Online STEAM Teaching More Engaging

## **Chapter 2: Good Practices for Empowering Girls' Participation in STEAM**

- 2.1. Introduction
- 2.2. The Gender Gap in STEAM: European Context
- 2.3. Review of Good Practices
- 2.4. Comparative Overview of Good Practices
- 2.5. Key Lessons and Recommendations
- 2.6. Conclusion
- 2.7. Chapter 2 References

## **Chapter 3: Simulation Tools and Technologies for Online Robotics Education**

- 3.1 Introduction
- 3.2 Simulation tools
- 3.3 Educational Projects and Initiatives Related to the Toolboxes
- 3.3 Resource repositories and other tools

# Chapter 1: Good Practices in Online STEAM Education with a focus on Robotics, Coding & Creative Technology

## 1.1. Introduction

---

The transition from physical to online teaching environments, especially in fields such as robotics, coding, and creative STEAM disciplines, requires far more than simply transferring content to digital platforms. It demands a fundamental rethinking of pedagogy, interaction, and learning design.

Online STEAM teaching must achieve a delicate balance between conceptual understanding, hands-on experimentation (physical or simulated), and sustained student engagement and collaboration. When these three pillars are addressed deliberately, virtual classrooms can become inclusive spaces of experimentation and creation environments where a student in a rural village can build and simulate a robot just as meaningfully as one sitting in a well-equipped laboratory.

This document, produced as part of Activity A2.4, brings together key findings from educational research and classroom practice to provide actionable guidance for teachers working in online or hybrid STEAM contexts. It is structured to move from the challenges teachers face, to the principles that guide effective responses, to practical techniques and concrete examples grounded in the realities of robotics and creative coding classrooms. Teacher personas and case studies are included throughout to illustrate how these ideas play out with real students.

### Who is this document for?

This guide is primarily addressed to:

- Teachers of robotics, coding, and digital making in lower and upper secondary education
- STEAM educators transitioning from face-to-face to online or hybrid delivery
- Teacher trainers and curriculum developers working within EU-funded projects or national initiatives
- School coordinators designing online STEAM programmes

The guidance is applicable to synchronous (live online sessions), asynchronous (self-paced), and blended approaches, though most examples are grounded in synchronous delivery where interaction design is most critical.

## 1.2. Core Challenges in Online STEAM Education

Online STEAM education, and robotics and coding in particular, presents a distinct set of challenges that stem primarily from translating hands-on, experiential learning into digital environments. Understanding these challenges is the necessary first step toward addressing them effectively.

### 1.2.1 Limited Access to Physical Hardware

Robotics kits, sensors, microcontrollers, and physical prototyping materials are central to traditional robotics education. When students learn from home, access to hardware is often absent, inequitable, or logistically complicated. This restricts students' ability to engage in tangible experimentation and iterative prototyping, arguably the heart of engineering learning. In addition, not all the schools are well-equipped with labs or have access to open labs.

#### Opportunity

Simulation environments such as Tinkercad Circuits, Open Roberta Lab, micro:bit MakeCode simulator, and Scratch offer credible alternatives that preserve much of the exploratory value of physical prototyping, particularly when paired with structured inquiry tasks.

### 1.2.2 Reduced Student Engagement

In online settings, passive participation can easily replace active involvement if interaction is not carefully structured. The absence of physical presence removes many of the social cues, body language, peer proximity, shared tools, that naturally sustain attention in a classroom. Students may appear to be present while their minds have wandered, or may disconnect entirely without their teacher noticing.

Research consistently shows that engagement in online learning is strongly linked to interaction frequency and quality (Chen et al., 2018). Without deliberate design, lessons can quickly become one-way broadcasts.

### 1.2.3 High Cognitive Load

Students in online STEAM sessions are simultaneously navigating digital interfaces, following instructions, understanding abstract concepts, writing or running code, and troubleshooting, often without immediate support. This concurrent demand on working memory constitutes a significant cognitive load that can overwhelm learners, especially beginners, and lead to disengagement or surface-level engagement (Chen et al., 2018).

### 1.2.4 Diverse Skill Levels

A typical online STEAM classroom contains students who range from complete beginners to confident coders or makers. This diversity is further amplified in online settings, where individualised support is harder to deliver. Beginners may feel lost and unable to ask for help discreetly, while more advanced students may disengage due to insufficient challenge.

### 1.2.5 Difficulty Providing Real-Time Support

In physical classrooms, teachers move around the room, glance at screens, and intervene at the exact moment a student encounters a bug or a conceptual stumbling block. Online, this ambient monitoring is almost impossible. Delays in feedback can hinder learning progression and increase frustration, particularly in robotics and coding, where errors and debugging are integral to the learning process.

## 1.2.6 Summary Table

| Challenge                 | Impact on Learning                               | Possible Response                                            |
|---------------------------|--------------------------------------------------|--------------------------------------------------------------|
| Limited hardware access   | Reduced experimentation and hands-on learning    | Simulation platforms (Tinkercad, Open Roberta Lab, MakeCode) |
| Reduced engagement        | Passive participation, dropout risk              | Breakout rooms, gamification, frequent checkpoints           |
| High cognitive load       | Confusion, shallow understanding                 | Chunked tasks, visual scaffolding, minimal tool stack        |
| Diverse skill levels      | Frustration for beginners; boredom for advanced  | Tiered tasks, peer mentoring, UDL approaches                 |
| Limited real-time support | Errors go unresolved; frustration builds         | Structured checkpoints, peer debugging, forums               |
| Tool proliferation        | Cognitive fragmentation, time lost on navigation | Single platform + one learning tool principle                |

## 1.3. Key Principles for Effective Online STEAM Teaching

The literature on effective online STEAM education, together with practitioner experience, converges on a set of guiding principles. These are not rigid rules but orientations that should inform planning, tool selection, and moment-to-moment decisions in the virtual classroom. Crucially, research consistently emphasises that the primary barriers to online STEAM education are instructional — not technological. Well-designed lessons can overcome most technical limitations (Chen et al., 2018).

### 1.3.1 Pedagogy Before Technology

The most important principle is the most easily overlooked: digital tools should be selected and used in service of clearly defined learning goals, not the other way around. Teachers who begin with 'what can this tool do?' rather than 'what do I want my students to understand or create?' risk building lessons around features rather than learning.

When teachers prioritise pedagogical clarity what students should understand, create, or achieve — they are better positioned to make purposeful use of technology and create coherent, impactful learning experiences.

#### In practice

Before choosing a platform or app, ask:

- What is the specific learning objective for this session?
- Which interaction does this tool support exploration, discussion, assessment, or creation?
- Could this objective be achieved with a simpler or more familiar tool?

### 1.3.2 Inclusion and Confidence Building

Online STEAM contexts must be actively designed to be inclusive, particularly for beginners and students from underrepresented groups. The initial encounter with abstract concepts and unfamiliar tools in a virtual setting can be profoundly intimidating. Teachers are encouraged to adopt a 'low floor, high ceiling, wide walls' approach (Resnick & Silverman, 2005): ensuring all learners can enter the activity, while creating room for extension and lateral exploration.

Early success experiences are especially important. Small, achievable tasks completed early in a course or session build the self-efficacy that enables students to take on greater challenges later.

### 1.3.3 Minimising and Stabilising the Technological Environment

Rather than introducing multiple platforms that fragment attention, teachers should prioritise the use of one or two reliable, accessible tools that support the core learning objectives. This reduces cognitive friction and allows students to focus on understanding concepts rather than navigating complex digital ecosystems.

Consistency matters: when students know exactly where to find materials, how to join a breakout room, and where to submit work, they can devote their mental energy to the actual learning content.

### 1.3.4 Visualisation First

Many core STEAM concepts, such as control flow, sensor logic, conditional structures, feedback loops, are inherently abstract. The principle of 'visualisation first' holds that these ideas should be made visible through demonstrations, simulations, and step-by-step modelling before students are asked to reproduce or modify them.

In online robotics and coding education, showing how a system behaves before asking students to replicate it significantly enhances comprehension and supports diverse learning styles. Live screen-sharing of code being written and tested, annotated diagrams, and simulation walkthroughs are all effective visualisation strategies.

### **1.3.5 Structured Interaction**

Meaningful engagement does not occur spontaneously in online environments. Teachers must intentionally embed interaction opportunities throughout lessons. These include guided questions, chat-based prompts, collaborative problem-solving in breakout rooms, polls, and peer feedback moments. The goal is to ensure students are active participants at regular intervals, not passive consumers of a broadcast.

Research on online STEM courses shows that structured peer discussions and instructor-student interaction are among the strongest predictors of student success and course completion (Chen et al., 2018).

### **1.3.6 Managing Cognitive Load**

Online STEAM lessons should be carefully paced, focusing on a limited number of key ideas per session and building in time for practice and reflection before introducing new content. Worked examples, partially completed code scaffolds, and visual step-by-step guides all help reduce the extraneous cognitive load that can overwhelm learners.

The principle of 'one tool at a time' applies here too: introducing a new platform and a new concept simultaneously can easily exceed students' working memory capacity.

### **1.3.7 Formative Assessment as a Continuous Thread**

Frequent, low-stakes formative assessment helps teachers monitor understanding in real time and helps students self-regulate their learning. In online settings, this is especially important because the ambient monitoring of a physical classroom is absent. Tools such as Kahoot, Mentimeter, padlets, and in-chat polls can be integrated at multiple points in a lesson, not just at the end, to keep assessment woven into the learning experience rather than separated from it.

### **1.3.8 Universal Design for Learning (UDL)**

UDL is a research-backed framework that promotes inclusive teaching by offering multiple ways for students to engage with, learn, and demonstrate understanding of content (CAST, 2018). In online STEAM contexts, this might mean offering the choice between a physical robot and a virtual simulation, providing video demonstrations alongside written instructions, or allowing students to show their understanding through a screencast rather than a written test.

Research by Chen et al. (2018) highlights that even limited UDL training for instructors can improve engagement and inclusivity, and may increase course completion rates particularly for students with diverse needs.

## 1.4. Practical Techniques for Interactive Online STEAM Teaching

Translating pedagogical principles into practice requires concrete, structured strategies that actively engage students. This section describes key techniques, with reference to robotics and creative coding contexts. Each technique is connected to its underlying pedagogical purpose to help teachers make informed choices about when and how to use it.

### 1.4.1 Breakout Rooms for Collaborative Problem-Solving

Breakout rooms are one of the most powerful tools available to online teachers, but their value depends entirely on how they are structured. Unstructured group time in breakout rooms frequently leads to off-task behaviour, unequal participation, or silence.

Effective breakout room design includes:

- **Clear, time-bound tasks:** Students should know exactly what they are expected to produce or decide within the allocated time.
- **Defined roles:** Assigning roles (e.g., driver/coder, navigator, note-taker, presenter) ensures accountability and gives every student a clear contribution.
- **Structured prompts:** A shared digital document or slide with guiding questions anchors the group's work.
- **Reporting back:** Brief report-outs from each group to the main room reinforce accountability and create shared learning moments.

#### Robotics Example

Groups of three are given a scenario: 'Design a system that automatically waters plants when the soil is dry.' Each group uses a shared Tinkercad workspace to sketch their sensor-actuator logic, with roles of 'engineer', 'tester', and 'presenter'. After 15 minutes, each presenter shares one design decision and one problem they encountered.

### 1.4.2 Gamification and Rapid Engagement Techniques

Gamified approaches are particularly effective in online settings for maintaining attention, generating energy, and providing immediate formative feedback without interrupting lesson flow.

- Kahoot, Quizizz, and Mentimeter can be used as warm-ups, mid-lesson concept checks, or closing reflections. The key is intentional placement not gamification for its own sake.
- 'Live coding challenges' where students race to implement a specific function or debug a broken script within a time limit create focused engagement and authentic problem-solving.
- Leaderboards and badges, used judiciously, can motivate participation but should be designed so that participation itself (not just correctness) is rewarded, to protect less confident students.

#### Creative Coding Example

At the start of a Scratch session on animation, the teacher shares a short broken project and asks students to post in the chat what they think is wrong. The first three correct answers earn a 'Debug Detective' badge displayed in the class padlet. Energy is immediately raised and the diagnostic activity surfaces prior knowledge.

### 1.4.3 Problem-Based Learning (PBL) Missions

Framing lessons around real-world challenges, 'missions', transforms the purpose of coding and robotics from abstract skill-building to meaningful problem-solving. PBL is among the most well-

evidenced approaches in STEM education for developing deeper understanding, transferable skills, and student motivation (Prince, 2004).

Online PBL requires:

- **A compelling scenario:** The problem must feel relevant and authentic i.e. smart cities, environmental monitoring, accessibility technology, social campaign and more
- **Structured phases:** Problem exploration → ideation → prototyping/simulation → testing → reflection and presentation.
- **Scaffolding:** Checkpoint questions, partially completed templates, and teacher check-ins prevent students from getting stuck.
- **Reflection:** Students should articulate what worked, what didn't, and what they would do differently.

### Robotics Example

Students are presented with the 'Smart Crossroads' mission: design a traffic light system that adapts based on pedestrian and vehicle sensor data. Working in pairs over two sessions, they plan their logic using a flowchart whiteboard, code it in the micro:bit simulator, and present a short screencast explaining their solution.

## 1.4.4 Visual Scaffolding and Live Demonstrations

Given the abstract nature of much STEAM content, visual support is not a supplement — it is a core instructional strategy. Effective visual scaffolding in online robotics and coding teaching includes:

- Live screen-sharing of code being built incrementally, including deliberate 'thinking aloud' about decisions and corrections.
- Annotated diagrams and flowcharts on digital whiteboards (Mural, Miro, [...]) that students can see built in real time.
- 'Predict before run' moments: the teacher shows a piece of code and asks 'what will this do?' before executing it, activating predictive reasoning.
- Asynchronous video recordings with annotations that students can rewatch at their own pace.

### Creative Coding Example 1

Before students create their own Scratch game character, the teacher shares their screen and codes a simple animated sprite from scratch, narrating their decisions: 'I'm choosing a repeat loop here because I want this to happen continuously, not just once, what other ways could we achieve this?' Students take annotated notes on a shared document. The teacher then reviews the answers.

### Creative Coding Example 2

The students are given a specific artwork (i.e. The Nighthawks of Edward Hopper) and they are invited to bring some interactivity in the painting (i.e. moving characters, turning on and off the bar lights, adding music and more) The teacher selects and shows the result (i.e. turning on and off the lights in the bar) and then asks ways according to which this could be achieved? Students take annotated notes on a shared document.

## 1.4.5 Student-as-Teacher and Peer Learning

When students explain concepts to each other, they deepen their own understanding, develop communication skills, and create a more inclusive, peer-supported learning community. In online settings, peer learning requires intentional design.

- 'Micro-teaching' moments: a student explains their solution to a coding problem in 60 seconds before the class moves on.

- Peer code review: students are paired to review and comment on each other's Scratch projects or micro:bit code, using a simple structured rubric.
- Peer mentoring channels: a dedicated Teams thread, or forum where students can ask and answer questions between sessions.

### 1.4.6 Interactive Digital Whiteboards

Digital whiteboards (Miro, Mural, Padlet ) extend visual thinking into collaborative space. In online STEAM teaching, they can serve as:

- Collective brainstorming canvases for the ideation phase of a PBL mission.
- Co-created system diagrams where groups contribute their component to a shared architecture.
- Exit ticket boards where students post one thing they learned and one question they still have.

Importantly, whiteboards create visible artifacts of student thinking that the teacher can review during and after the lesson to identify misconceptions and plan next steps.

//Add notes from BEREADY work here

### 1.4.7 Asynchronous Extension via Forums and Shared Spaces

Not all learning happens in live sessions, and not all students can contribute confidently in real time. Online forums and shared spaces extend interaction beyond synchronous lessons and provide a lower-pressure environment for reflection, question-asking, and peer support.

In robotics and coding contexts, a 'Debugging Wall', a dedicated thread where students post bugs they have encountered and solutions they have found creates a living, community-built knowledge resource.

### 1.4.8 Summary of Techniques

| Technique                       | Pedagogical Purpose            | Example in Robotics/Coding           | Platforms                     |
|---------------------------------|--------------------------------|--------------------------------------|-------------------------------|
| <b>Breakout Rooms</b>           | Collaboration, peer learning   | Design a smart traffic system        | Zoom, Teams                   |
| <b>Kahoot / Quizizz</b>         | Formative assessment, energy   | Identify correct sensor logic        | Kahoot, Quizizz, Mentimeter   |
| <b>Live Demos + Think-Aloud</b> | Modelling, visualisation       | Build a Scratch sprite step by step  | Screen share, OBS Studio      |
| <b>Digital Whiteboards</b>      | Co-creation, brainstorming     | Draw robot system flowchart          | Miro, Mural, Padlet           |
| <b>PBL Missions</b>             | Authentic problem-solving      | Smart crossroads challenge           | Simulation + shared docs      |
| <b>Peer Code Review</b>         | Peer learning, communication   | Review partner's Scratch game        | Google Docs, Teams, Padlet    |
| <b>Discussion Forums</b>        | Inclusion, async reflection    | Debugging Wall thread                | Slack, Teams, Moodle          |
| <b>Student Micro-Teaching</b>   | Deep understanding, confidence | Explain your code solution in 60 sec | Screen share, video recording |

## 1.5. From Theory to Practice: Teacher Personas

The following personas illustrate how different teachers navigate the challenges and apply the principles described in this document. Each persona is composite, drawn from common patterns observed in online STEAM education practice, and is intended to help teachers recognise aspects of their own contexts and draw inspiration for their practice.

### Persona 1: Maria - The Beginner Supporter

|                                            |                                                                                                        |
|--------------------------------------------|--------------------------------------------------------------------------------------------------------|
| <b>Maria</b><br>Lower Secondary<br>Teacher | <b>Age group:</b> Students aged 12–13                                                                  |
|                                            | <b>Key tools:</b> MakeCode Simulator, Scratch, Padlet, breakout rooms                                  |
|                                            | <b>Main challenge:</b> Building confidence with students who have little or no prior coding experience |

#### Profile

Maria teaches a mixed-ability class of 25 students, the majority of whom have never written a line of code. She is acutely aware that the initial online encounter with programming can be daunting and that losing a student in the first two sessions may mean losing them permanently. Her approach is anchored in emotional safety, gradual skill progression, and the belief that mistakes are data, not failures. She designs every session around the question: 'How can I make sure every student feels they can do this?'

#### Practice in Action

Maria never begins a new topic with an open-ended task. She opens with a guided exploration: students run a pre-built MakeCode simulator program, observe what it does, and post one observation in the chat. This 'notice and wonder' opening has zero failure states every student can contribute.

She structures breakout rooms so that each pair includes one slightly more experienced student who acts as the 'driver', while the less experienced student reads the instructions aloud and makes suggestions roles Maria has explained and rehearsed. She checks each room briefly every four minutes, narrating what she sees: 'I love this approach you've found a different way to solve this than the group next door.'

When students get stuck, Maria doesn't immediately provide answers. She asks: 'What have you tried? What happened?' This metacognitive scaffolding builds problem-solving resilience, not just code that works.

#### ERGO Key Takeaway for Teachers

Design for the most anxious learner in the room. If your entry task has a right and wrong answer that is immediately visible to peers, beginners will disengage before they begin.

### Persona 2: Thomas - The Tool Optimizer

|                                              |                                                                                 |
|----------------------------------------------|---------------------------------------------------------------------------------|
| <b>Thomas</b><br>Experienced STEM<br>Teacher | <b>Age group:</b> Students aged 13–15                                           |
|                                              | <b>Key tools:</b> Zoom, Tinkercad, shared Google Slides                         |
|                                              | <b>Main challenge:</b> Reducing cognitive overload caused by too many platforms |

#### Profile

Thomas is a confident, technology-enthusiastic teacher who, in his first year of online teaching, assembled an impressive digital toolkit: one platform for video calls, another for coding, a third for

sharing resources, a fourth for quizzes, and a fifth for discussion. He noticed, however, that students were spending more time asking 'where do I go now?' than thinking about circuits or sensors. He radically simplified.

### Practice in Action

Thomas now runs all sessions on a single video platform, with a consistent lesson structure that students could navigate with their eyes closed: five-minute warm-up quiz in Mentimeter, demonstration with live Tinkercad screen-share, 20-minute build task, pair share in breakout, whole-group reflection via the chat.

He created a one-page 'Digital Classroom Map' a simple visual showing which tool is used when and posts it at the start of every session. New tools are only introduced when they directly serve a learning goal that cannot be achieved with existing tools, and they are introduced one at a time, with a dedicated 10-minute orientation session.

Thomas also produces two-minute 'how-to' video clips for each tool and makes them available in a shared folder. These eliminate the need for him to explain navigation during lesson time.

### ERGO Key Takeaway for Teachers

Every minute spent explaining a tool is a minute not spent on learning. Ruthlessly simplify your digital environment, and invest the time saved in richer interaction.

## Persona 3: Helena - The Visual Explainer

|                                             |                                                                                            |
|---------------------------------------------|--------------------------------------------------------------------------------------------|
| <b>Helena</b><br>Experienced STEAM Educator | <b>Age group:</b> Students aged 13–15                                                      |
|                                             | <b>Key tools:</b> Digital whiteboards, screen share + annotation, simulator (Open Roberta) |
|                                             | <b>Main challenge:</b> Making abstract programming concepts visible and comprehensible     |

### Profile

Helena is a STEAM educator who has spent years building her understanding of how students develop mental models of systems. She has observed that most misconceptions in robotics and coding, infinite loops that students don't intend, sensors that behave unexpectedly, variables that get overwritten, arise not from carelessness but from the fact that the code is invisible: students cannot see the system running inside the machine. Her teaching philosophy is built around making the invisible visible.

### Practice in Action

When introducing conditional logic, Helena doesn't start with code. She opens a blank Miro board and draws a flowchart collaboratively with the class, asking students to contribute decision nodes via sticky notes. Only when the logic is clear on the board does she open the coding environment.

Her live coding demonstrations are structured around the 'predict-run-explain' cycle: she shows a piece of code, asks students to type their prediction of the output into the chat, runs it, and then collectively analyses why predictions were or weren't correct. This approach is particularly effective for surfacing and correcting misconceptions about loops and sensor thresholds.

She regularly records short annotated screencasts after each session, 'lesson replay' videos of two to three minutes that highlight the three key concepts from the class. Students report these are among the most valued resources in the course.

### ERGO Key Takeaway for Teachers

Abstraction is the enemy of understanding in early STEAM learning. Always provide a visual anchor, a diagram, a simulation, a physical analogy, before asking students to work with symbolic or textual representations.

## Persona 4: Alex - The Interaction Designer

|                                            |                                                                                             |
|--------------------------------------------|---------------------------------------------------------------------------------------------|
| <b>Alex</b><br>Secondary School<br>Teacher | <b>Age group:</b> Students aged 13–16                                                       |
|                                            | <b>Key tools:</b> Zoom breakout rooms, Padlet, Kahoot, collaborative coding platforms       |
|                                            | <b>Main challenge:</b> Preventing passive participation and ensuring equitable contribution |

### Profile

Alex came to online teaching from a background in collaborative and project-based learning. He was quickly frustrated by the tendency of online sessions to default to a teacher-talks, students-listen dynamic. He views learning as fundamentally social and has invested significant effort in designing interaction structures that make genuine collaboration happen, not just co-presence.

### Practice in Action

Every lesson Alex teaches is planned with an interaction map: a diagram showing when each type of student interaction is designed to occur (teacher-to-student, student-to-student, student-to-content), and which tool enables it. He aims for no more than seven minutes without a student interaction moment.

His breakout rooms are highly structured. Each group has a shared slide in a collaborative presentation tool. The slide has four sections, one for each role (programmer, debugger, designer, presenter), and students fill in their sections during the activity. When groups reconvene, the slide also serves as the basis for the report-out, reducing the preparation time needed.

Alex uses a 'cold call with safety net' technique for whole-class questioning: he nominates a student to answer, but immediately offers them the option to 'phone a friend' (ask a breakout partner) or 'ask the audience' (post the question to the chat). This maintains a sense of active readiness without creating anxiety.

### ERGO Key Takeaway for Teachers

Interaction in online lessons does not happen by itself. It must be designed, timed, and structured, with explicit role assignments, clear tasks, and regular opportunities to contribute.

## Persona 5: Paul - The Project Designer

|                                                 |                                                                                          |
|-------------------------------------------------|------------------------------------------------------------------------------------------|
| <b>Paul</b><br>Upper Secondary STEM<br>Educator | <b>Age group:</b> Students aged 14–17                                                    |
|                                                 | <b>Key tools:</b> Scratch, micro:bit simulator (MakeCode), Tinkercad, shared Google Docs |
|                                                 | <b>Main challenge:</b> Connecting robotics and coding to meaningful real-world problems  |

### Profile

Paul teaches upper secondary students who often ask 'why are we doing this?' and deserve a good answer. He is motivated by the conviction that robotics and coding should be taught not as sets of isolated technical skills, but as tools for understanding and improving the world. His course is organized not by syntax or sensor type, but by challenge: each unit opens with a real-world problem that students work toward solving.

## Practice in Action

Paul's current unit opens with a five-minute documentary clip about floods in coastal cities and a question: 'How could sensing and automation help communities prepare for or respond to flooding?' Students spend ten minutes in breakout rooms generating ideas without any coding constraint. This problem-framing phase is deliberate, it positions the technical work that follows as a means to an end, not the end itself.

Over the following three sessions, students develop a simulation of a flood sensor and warning system in Tinkercad and micro:bit MakeCode. The final session is a peer showcase, a 'Demo Day', where groups present their simulations via screen-share to the class and receive structured feedback using a peer review rubric focused on functionality, clarity of design, and real-world plausibility.

Paul integrates reflection throughout. At the end of each session, students update a shared 'design journal' documenting their decisions, problems, and next steps. These journals are shared with him and inform his planning for the following session.

## ERGO Key Takeaway for Teachers

Motivation in online learning is sustained by purpose. When students understand why they are learning to program sensors, they are more willing to persist through debugging and difficulty.

## Persona 6: Leila - The Creative Coding Enabler

|                                                       |                                                                                    |
|-------------------------------------------------------|------------------------------------------------------------------------------------|
| <b>Leila</b><br>Arts & STEAM Cross-Curricular Teacher | <b>Age group:</b> Students aged 11–14                                              |
|                                                       | <b>Key tools:</b> Scratch, Padlet                                                  |
|                                                       | <b>Main challenge:</b> Integrating artistic expression with computational thinking |

## Profile

Leila works at the intersection of arts and technology, teaching in a school that emphasises creative and interdisciplinary approaches. She has found that students who resist 'coding lessons' often engage enthusiastically when the same computational concepts are encountered through music, animation, or visual art. She uses creative coding, coding as a medium for artistic expression, as her primary entry point into computational thinking.

## Practice in Action

A typical Leila lesson opens not with a coding environment, but with a stimulus: a short animation, an interactive artwork, or a piece of algorithmically-generated music. Students discuss what they notice, what they imagine the code behind it looks like, and what they would create if they had the same tools.

Using Scratch, Leila introduces variables, loops, and conditional logic through animation projects: students create a 'generative artwork', a pattern that changes based on user input or random values. The computational concepts are identical to those in any robotics lesson, but the creative frame lowers the perceived risk and raises motivation, particularly for students who don't identify as 'technical'.

She uses Padlet as a class gallery: every student's project is posted and anyone can leave a comment using a structured prompt ('I notice... I wonder... What if...'). This creates a peer feedback culture that is appreciative rather than evaluative and extends engagement beyond the live session.

## ERGO Key Takeaway for Teachers

STEAM, not STEM: the Arts dimension is not decoration. Creative coding provides a powerful on-ramp to computational thinking for students who may not respond to technical or engineering framings.

## 1.6. Making Online STEAM Teaching More Engaging: Evidence-Based Strategies

---

Engagement in online learning is not a single thing: it has cognitive, behavioural, and emotional dimensions (Fredricks et al., 2004). Students can be behaviourally compliant (cameras on, task completed) while being cognitively absent. The goal is not compliance but genuine intellectual and social involvement.

The following evidence-informed strategies, drawn from research on online STEM education (Chen et al., 2018; Prince, 2004) and digital pedagogy more broadly, are particularly effective in robotics and coding contexts.

### 1.6.1 Hook and Launch Every Session Intentionally

The first five minutes of an online lesson determine whether students settle into active engagement or drift into passive attendance. Effective hooks include:

- **A broken code challenge:** Students are shown a non-functioning Scratch project or micro:bit program and asked to diagnose the problem. No context needed; the challenge is immediate.
- **A provocative question:** 'Could a robot make a fair referee? Why?'- posted in chat for 90 seconds of silent reflection before a brief discussion.
- **A short 'predict the output' demo:** A piece of code shown on screen; students type their prediction into the chat before it runs.

### 1.6.2 Keep the Cognitive Engine Running: The 7-Minute Rule

In online settings, student attention drifts significantly after sustained passive periods. A practical guideline is to ensure that no more than seven minutes pass without a student action: a chat response, a vote, a short task, a breakout moment, a prediction. This rhythm keeps the cognitive engine engaged and provides the teacher with continuous, low-cost formative data.

### 1.6.3 Make Thinking Visible

Students engage more deeply when their thinking is externalised and visible to themselves, their peers, and their teacher. Strategies include:

- Shared collaborative documents where students annotate their reasoning as they code.
- Digital whiteboards where groups map out a system before coding it.
- 'Show your thinking' chat prompts where students explain their next step before executing it.
- Design journals that document decisions, not just outcomes.

### 1.6.4 Connect to Student Identity and Experience

Engagement is sustained when students perceive a connection between the learning and their own lives, interests, or futures. In coding and robotics, this means:

- Offering missions with genuine social relevance: accessibility technology, environmental monitoring, community safety.
- Allowing student choice in project themes: a student who builds a Scratch game about their favourite sport will invest more deeply in understanding variables and loops.
- Explicitly discussing careers and communities where these skills are used including non-obvious ones like music production, healthcare, architecture, and storytelling.

### 1.6.5 Celebrate Errors and Debugging Culture

In robotics and coding, debugging is not a failure mode, it is the primary learning mode. Teachers who explicitly celebrate interesting errors, create 'bug museums' in shared documents, and model their own debugging process in live demonstrations create a culture where mistakes are assets, not embarrassments. This is especially important in online settings where students may be more inhibited about sharing difficulties.

### 1.6.6 Use Storytelling and Narrative

Framing technical challenges as stories or missions dramatically increases engagement. 'A city council needs your help...' or 'A team of marine biologists has a problem...' transforms a coding exercise into a narrative that students want to resolve. This is particularly effective with younger students but has proven value across all age groups in STEAM contexts.

## 1.7. Building Your Own Online Teaching Style

---

Developing an effective online teaching style in STEAM should not be understood as adopting a fixed methodology, but as assembling and constantly refining a personal pedagogical repertoire. No single approach works for every teacher, class, or context. The personas described in Section 5 are not prescriptions, they are invitations to reflection.

Teachers are encouraged to begin with a self-audit, honestly assessing their current practice against the principles outlined in this guide. The following questions can support that reflection:

- Where in my lessons do students shift from receiving to doing? How frequently does this happen?
- What is the simplest technology that would support my key learning objective for each session?
- How do I currently know, in real time, whether students are understanding or confused?
- What visual representations do I make available to support students who struggle with abstraction?
- How do I currently support the least confident student in an online session?

**Iteration is key.** Effective online teachers consistently describe their development as a process of trying one new strategy per session, observing what happens, and adjusting. Small experiments — one structured breakout room, one predict-the-output moment, one Debugging Wall thread — are more sustainable and revealing than wholesale redesign.

**Collaboration with colleagues**, whether within the same school or across the partner network of this project, is also a powerful developmental resource. Peer observation, co-planning sessions, and shared repositories of missions and tasks accelerate growth in ways that individual reflection alone cannot.

## 1.8. Teacher Toolbox: Online STEAM Teaching Toolkit

The following toolkit provides a quick reference for teachers planning online STEAM sessions. Tools are organised by pedagogical function — what they help you achieve — rather than by technology category.

| Strategy / Tool                            | Design Element                  | Pedagogical Purpose                  | Example in Robotics/STEAM                        |
|--------------------------------------------|---------------------------------|--------------------------------------|--------------------------------------------------|
| <b>Breakout Rooms</b>                      | Active / Peer Learning          | Collaboration, problem-solving       | Design a smart irrigation system in small groups |
| <b>Kahoot / Quizizz</b>                    | Formative Assessment            | Check understanding, energy          | Identify correct sensor logic                    |
| <b>Digital Whiteboards</b>                 | Active Learning / Visualisation | Externalise thinking, co-create      | Draw robot system flowchart                      |
| <b>Discussion Forums / Debugging Wall</b>  | Async Engagement                | Inclusion, peer support, reflection  | Post bugs and solutions between sessions         |
| <b>Instructor Videos + Annotated Notes</b> | Instructor Content              | Clarity, re-watchability             | Video: how ultrasonic sensors work               |
| <b>Peer Mentoring / Code Review</b>        | Peer Learning                   | Collaborative knowledge building     | Review partner's Scratch animation               |
| <b>Simulation Platforms</b>                | Hands-on Learning               | Experimentation without hardware     | Tinkercad circuit simulation, Webots robot       |
| <b>Flexible Tasks (UDL)</b>                | Universal Design                | Inclusivity, accessibility           | Choose: physical robot OR virtual simulation     |
| <b>Polls / Mid-lesson Checks</b>           | Formative Assessment            | Maintain attention, monitor progress | 'Which sensor fits this task?' live poll         |
| <b>PBL Missions / Design Challenges</b>    | Project-Based Learning          | Authentic context, motivation        | Smart crossroads; flood warning system           |
| <b>Padlet Gallery / Creative Showcase</b>  | Peer Feedback / UDL             | Appreciation, community, expression  | Share Scratch creative coding projects           |

## 1.9. Chapter 1 References

---

Chen, B., Bastedo, K., & Howard, W. (2018). Exploring design elements for online STEM courses: Active learning, engagement & assessment design. *Online Learning*, 22(2), 59–75.  
<https://doi.org/10.24059/olj.v22i2.1369>

Fredricks, J. A., Blumenfeld, P. C., & Paris, A. H. (2004). School engagement: Potential of the concept, state of the evidence. *Review of Educational Research*, 74(1), 59–109.

Haak, D. C., HilleRisLambers, J., Pitre, E., & Freeman, S. (2011). Increased structure and active learning reduce the achievement gap in introductory biology. *Science*, 332(6034), 1213–1216.

Prince, M. (2004). Does active learning work? A review of the research. *Journal of Engineering Education*, 93(3), 223–231.

Resnick, M., & Silverman, B. (2005). Some reflections on designing construction kits for kids. *Proceedings of the 2005 Conference on Interaction Design and Children*.

Yamal, J. M. (2023). Teaching and designing online STEM courses to support self-directed learning skills. *Post-Secondary Collaborative*. <https://postseccollab.org>

Additional resource: Cambridge ELT Blog (2023). Enhancing engagement: 6 strategies for interactive and effective online learning. <https://www.cambridge.org/elt/blog/2023/11/30/enhancing-engagement-6-strategies-for-interactive-and-effective-online-learning/>

<https://www.cambridge.org/elt/blog/2023/11/30/enhancing-engagement-6-strategies-for-interactive-and-effective-online-learning/>

# Chapter 2: Good Practices for Empowering Girls' Participation in STEAM

## 2.1. Introduction

The purpose of this chapter is to review and document good practices for empowering girls' participation in STEAM-related activities, drawing on a range of European and international initiatives, with reference to prominent frameworks such as EU Code Week among others.

Gender inclusivity is one of the central pillars of the ERGO project. The needs analysis carried out across Poland, Greece, Italy, and Spain during the project's preparation phase revealed that 89.6% of teachers reported their schools lack any structured strategy to encourage girls' participation in STEM, or acknowledge a need for significant improvement.<sup>1</sup> Furthermore, 81.2% of respondents identified gender-inclusive strategies as a priority. These figures underline both the urgency and the opportunity that ERGO addresses.

The practices gathered in this document serve three purposes within ERGO: they inform Chapter 2 of the ERGO Teacher Guidebook ('Promoting Girls' Engagement in Robotics'); they provide models and scenarios for teachers to draw upon during the eCourse and participatory design workshops in WP3 and WP4; and they guide the inclusive design of pilot activities in WP5, including the gender tracking metrics embedded in ERGO's evaluation tools.

## 2.2. The Gender Gap in STEAM: European Context

The underrepresentation of girls and women in STEAM fields is a well-documented and persistent challenge across Europe. According to the *She Figures 2024* report, released by the EU Publications Office on 11 February 2025, only 32.8% of graduates in STEM across the EU were women.<sup>2</sup> Women's representation drops further as they progress through education and professional life: while women outnumber men at bachelor's level across all fields, their presence falls sharply at doctoral level, with women's representation at doctoral level having decreased in half of all narrow STEM fields since 2018.

The EU Digital Decade Progress Report 2024 confirms that women account for just 1 in 5 ICT specialists in Europe.<sup>3</sup> The EU's STEM Education Strategic Plan has set ambitious 2030

---

<sup>1</sup>ERGO Partnership, "ERGO Needs Analysis Survey Report", 2024 (internal project document, annexed to ERGO application KA220-SCH-D0356924).

<sup>2</sup>European Commission, "She Figures 2024: Gender in Research and Innovation", Publications Office of the EU, released 11 February 2025. Available at: <https://op.europa.eu/en/publication-detail/-/publication/2d15c6d4-dae4-11ef-a76b-01aa75ed71a1>

<sup>3</sup>European Commission, "Report on the State of the Digital Decade 2024", September 2024. Available at: <https://digital-strategy.ec.europa.eu/en/library/state-digital-decade-2024>

targets, including that at least 40% of STEM students at tertiary level should be female and that one in three ICT PhD students should be women.<sup>4</sup> Reaching these targets requires systematic, early, and sustained intervention — starting in secondary school, precisely the level at which ERGO operates.

Critically, research consistently shows that girls generally outperform boys in computer and information literacy at a young age,<sup>5</sup> yet self-select out of ICT and STEM pathways as they progress through school. This points to pedagogical and cultural interventions — not aptitude — as the key lever for change, and is where projects like ERGO, and the practices reviewed below, are most valuable.

### **Key European Statistics (2024–2025)**

- Only 32.8% of EU STEM graduates are women (She Figures 2024)
- Women account for just 1 in 5 ICT specialists in Europe (Digital Decade 2024)
- Women represent just 24% of self-employed professionals in science, engineering and ICT
- 89.6% of ERGO survey respondents report no structured strategy for girls' STEM participation
- Girls outperform boys in computer and information literacy at early ages (ICILS 2023)
- EU 2030 target: 40% female STEM graduates at tertiary level

## **2.3. Review of Good Practices**

The following practices have been selected on the basis of their documented impact, their alignment with ERGO's pedagogical approach, their replicability in secondary school settings, and their relevance to online or digitally enabled robotics and STEAM education. They span EU-funded programmes, school-based initiatives, and wider European frameworks.

### **2.3.1 Girls Go Circular / Girls Go STEM (EIT – European Institute of Innovation & Technology)**

Girls Go Circular (GGC), now rebranded as Girls Go STEM, is a flagship EU initiative coordinated by the European Institute of Innovation and Technology, designed to equip students — especially young women — with digital and entrepreneurial skills through free, online education. Its curriculum covers topics including AI, robotics, sustainable mobility, and the circular economy, making it one of the most comprehensive European programmes at the intersection of gender inclusion and STEAM.<sup>6</sup>

By 2024, the programme had reached over 48,000 girls across 35 countries, with 18 specialised learning modules available in 24 languages.<sup>7</sup> In the 2024 edition alone, over 21,000 girls successfully completed courses, and 85 teams participated in the STEM

---

<sup>4</sup>European Commission, "STEM Education Strategic Plan", European Education Area. Available at: <https://education.ec.europa.eu/focus-topics/digital-education/action-plan/Women-participation-in-STEM>

<sup>5</sup>International Association for the Evaluation of Educational Achievement (IEA), "International Computer and Information Literacy Study (ICILS) 2023". Available at: <https://www.iea.nl/studies/iea/icils/2023>

<sup>6</sup>STEAMBrace Project, "EU Initiatives for Gender Equality in STEM Education", October 2025. Available at: <https://steambraceproject.eu/eu-initiatives-for-gender-equality-in-stem/>

<sup>7</sup>JA Europe, "Why Investing in Girls in STEM is Essential for a Thriving Future", February 2025. Available at: <https://jaeurope.org/why-investing-in-girls-in-stem-is-essential-for-a-thriving-future/>

Challenge, culminating in a STEM Forum where 20 teams showcased their projects. The AI for Gender Equality Student Challenge engaged 234 girls from 24 European countries in critical discussions about AI's ethical and societal implications.

### **Key Takeaway**

GGC/Girls Go STEM demonstrates that free, multilingual, online STEAM education can scale to tens of thousands of girls across Europe.

Its integration of real-world challenges (AI, sustainability, robotics) into learning contexts is directly applicable to ERGO's WP4 project design.

The STEM Challenge competition format offers a model for ERGO pilot activities that motivate participation through peer collaboration and showcasing.

## **2.3.2 ESTEAM Fests (EISMEA – European Commission)**

The ESTEAM Fests are a series of events organised under the European Innovation Council and SMEs Executive Agency (EISMEA), bringing together girls, young women, and women professionals to build digital and entrepreneurial competences. They combine hands-on technical workshops with role model presentations, networking, and confidence-building activities.

The 5th Girls ESTEAM Fest took place in Berlin in April 2024, co-organised with the German Federal Ministry for Economy and Climate Action — the first time a ministry co-organised such an event. A total of 120 girls aged 12 to 16 participated, hearing from female entrepreneurs who had started a company at age 15.<sup>8</sup> The 5th Women ESTEAM Fest was held in Milan in March 2024, bringing together approximately 130 women for workshops on data, digital competences, and public speaking. An online edition in September 2024 reached 60 women with workshops on AI, analytics, and leadership. In February 2025, 28 organisations gathered in Brussels for an ESTEAM Fest that included a visit to a local FabLab and a bioeconomy incubator.

### **Key Takeaway**

The ESTEAM model of combining technical skill-building with role model stories, confidence workshops, and networking is highly transferable to ERGO's teacher training and pilot activities.

Co-organisation with a government ministry signals institutional endorsement that amplifies impact and credibility.

FabLab and makerspace visits demonstrate how physical making and digital simulation can be productively combined.

## **2.3.3 BridgeSTEAM – Girls in STEAM Academy**

The Girls in STEAM Academy operates a flagship programme called BridgeSTEAM — a free, two-day intensive coding and digital skills experience for girls aged 14 to 16. The programme deliberately pairs technical coding instruction with discussions on gender equality, global challenges, and personal development, creating a holistic learning experience rather than a purely technical one.

---

<sup>8</sup>European Innovation Council and SMEs Executive Agency (EISMEA), "ESTEAM Fests and Communities for Girls and Women". Available at: [https://eisma.ec.europa.eu/esteam-fests-and-communities-girls-and-women\\_en](https://eisma.ec.europa.eu/esteam-fests-and-communities-girls-and-women_en)

Results are among the strongest documented in any European girls-in-STEAM initiative: over 93% of participants reported increased interest in STEAM careers, and 89% noted improved self-confidence.<sup>9</sup> Over 250 girls have graduated from BridgeSTEAM — including Ukrainian refugee girls and Turkish Cypriot girls — and many continue to benefit through an alumnae network offering job shadowing, public events, and the SheSpeaks podcast, where participants practice public speaking and share their perspectives as young women in STEAM.

### **Key Takeaway**

BridgeSTEAM's outcome data (93% increased STEAM interest, 89% improved confidence) provides concrete benchmarks for ERGO's own evaluation framework.

Its model of pairing technical activities with gender equality discussions directly supports ERGO's inclusive STEAM framework.

The alumnae network model demonstrates the value of sustained community beyond a single activity — relevant to ERGO's sustainability ambitions.

## **2.3.4 STREAM IT – Erasmus+ Project on Structural Barriers in STEAM**

STREAM IT is an Erasmus+ project specifically focused on identifying and addressing the institutional and structural challenges that limit girls' participation in STEAM education and careers, with a target age group of 14 to 18. Its research work involved an extensive review of existing European STEAM education initiatives, making it a valuable reference for ERGO's own good practices review.

As part of its hands-on activities strand, STREAM IT organised a dedicated International Girls in ICT Day event at the Institute Mihajlo Pupin, where 20 girls aged 13 to 14 explored advanced simulators, 3D printers, and a coding workshop tackling real-life programming challenges, accompanied by a panel of five female engineers and one researcher providing career insights.<sup>10</sup> The initiative also created workshop series, hands-on activities at science centres, mentoring programmes, and collaborative networks at national and international level.

### **Key Takeaway**

STREAM IT's focus on structural and institutional barriers — not individual attitudes — frames gender inclusivity as a systemic challenge, which is the level at which ERGO's teacher training is most impactful.

Its combination of technical simulation activities with female professional panels provides a direct model for ERGO webinars in WP3.

The project's multilevel approach (workshops, mentoring, networks) maps naturally onto ERGO's own cascading structure.

## **2.3.5 Girls in ICT Talks – Malta (2024–2025)**

---

<sup>9</sup>EU Digital Skills and Jobs Platform, "Girls in STEAM Academy – Good Practice". Available at: <https://digital-skills-jobs.europa.eu/en/inspiration/good-practices/girls-steam-academy>

<sup>10</sup>STREAM IT Project, "STEAM for All: Best European Practices in Gender-Inclusive Education", May 2025. Available at: <https://streamit-project.eu/2025/05/01/steam-for-all-best-european-practices-in-gender-inclusive-education/>

The Girls in ICT Talks initiative in Malta offers a school-embedded model of gender-inclusive STEAM promotion, delivered through collaboration between the Malta Digital Innovation Authority (MDIA), the national career guidance service, and the Church Secretariat's digital literacy department.

In the 2024–2025 edition, nine schools participated — both state and church institutions — with sessions led by MDIA Women Ambassadors who shared their personal journeys in ICT.<sup>11</sup> The sessions were designed specifically for Year 7 girls, using relatable examples and accessible language to make ICT feel exciting and attainable. Key messages included that curiosity and creativity — not perfect grades — are what matter most in technology. Feedback from schools was overwhelmingly positive, with educators highlighting how the sessions complemented classroom learning and introduced students to curriculum topics in real-world contexts.

### **Key Takeaway**

The partnership between a digital authority, schools, and career guidance services provides a scalable institutional model that ERGO school partners can replicate locally.

Focusing on relatable storytelling and accessible language — rather than technical jargon — is especially effective with girls who do not yet self-identify as 'tech people'.

Embedding sessions within normal school hours, rather than as extra-curricular activities, maximises reach.

### **2.3.6 EU Code Week – Girls in Digital and GCIB Sprints**

EU Code Week, the European Commission's annual initiative to promote digital skills and coding across Europe, has developed a dedicated gender programme called Girls in Digital (GiD), operating through Girls Code it Better (GCIB) Sprints.<sup>12</sup> These are short, 3 to 4-hour hands-on learning experiences where girls collaborate to tackle real-world challenges using digital tools, led by female tech professionals who serve as role models. Sprints can be run fully online as well as in-person, making them compatible with ERGO's emphasis on digital-first learning.

EU Code Week is directly referenced in the ERGO proposal as a channel through which results will be disseminated, and EDUMOTIVA (P2) has established connections to the Code Week ambassador network. Over 1.34 million participants have been reported on the EU Code Week platform, with female participation averaging 40% across reported figures.<sup>13</sup> The GCIB Replication Kit — a publicly available, step-by-step guide for organising Sprints — makes the model immediately accessible to ERGO teachers who complete the WP3 training. The 2024 Launch Event featured a dedicated 'Girls and Women in Digital' session with inspiring stories from female tech professionals.<sup>14</sup>

---

<sup>11</sup>EU Digital Skills and Jobs Platform, "Empowering Future Innovators: 2024–2025 Edition of Girls in ICT Talks Marks Another Successful Year", May 2025. Available at: <https://digital-skills-jobs.europa.eu/en/latest/news/empowering-future-innovators-2024-2025-edition-girls-ict-talks-marks-another-successful>

<sup>12</sup>EU Code Week, "Girls in Digital – Girls Code it Better Initiative". Available at: <https://codeweek.eu/girls-in-digital-week>

<sup>13</sup>European Health and Digital Executive Agency (HaDEA), "Inspiring the Next Generation: Girls in ICT Day 2025", April 2025. Available at: [https://hadea.ec.europa.eu/news/inspiring-next-generation-girls-ict-day-2025-2025-04-23\\_en](https://hadea.ec.europa.eu/news/inspiring-next-generation-girls-ict-day-2025-2025-04-23_en)

<sup>14</sup>EU Digital Skills and Jobs Platform, "Highlights from the EU Code Week 2024 Launch Event", October 2024. Available at: <https://digital-skills-jobs.europa.eu/en/latest/news/highlights-eu-code-week-2024-launch-event>

### **Key Takeaway**

GCIB Sprints require no prior experience from participants and no expensive equipment — directly mirroring ERGO's own low-barrier, simulation-based approach.

The Replication Kit means ERGO-trained teachers can independently organise GCIB activities post-project, strengthening sustainability.

Girls in Digital Week (March 2026) falls within the ERGO teacher training period, offering school partners a ready-made European event to align their activities with.

### **2.3.7 Programming, Robotics and Rights for Girls (EU Delegation, Geneva)**

This event, organised at the EU Delegation to the United Nations in Geneva as part of EU Code Week, represents one of the most documented examples of combining robotics education with a gender rights framework. EU Ambassador Walter Stevens opened a one-day workshop for 150 young participants, specifically framed around encouraging girls and young women to invest in digital technology expertise.

Girls aged 6 to 17 coded an electronic light card (hepiaLight) and programmed the collaborative robot PoppyJr, while engaging with discussions about their right to education.<sup>15</sup> The dual framing — technical skill-building alongside civic awareness of gender equality as a fundamental right — is a powerful pedagogical combination. Connecting robotics activities to wider social and rights contexts gives girls a reason to care about technology beyond its technical dimensions, and has been shown to resonate strongly with female learners drawn to purposeful, socially embedded work.

### **2.3.8 STEAMBrace (Horizon Europe)**

STEAMBrace is a Horizon Europe 36-month project that aims to address the existing gender gap in STEM fields by leveraging the potential of STEAM education and establishing a European-wide coordination network: the STEAM Alliance for Europe.<sup>16</sup> The project takes a specifically arts-integrated approach, arguing that when creative and artistic methodologies are embedded in STEM learning, girls are more likely to participate, express themselves, and remain engaged. STEAMBrace collaborates with complementary projects such as STREAM IT and Road-STEAMer, exchanging strategies to enhance creative pedagogy and boost diversity in education.

In pilot activities across Europe, over 120 students and 9 teachers engaged in STEAMBrace workshops that combined science with design, AI, and creative thinking. The project also established the first coordination network for gender-responsive STEAM education at European level, and organises the STEAM Week for future women innovators as an annual showcase event.

---

<sup>15</sup>European External Action Service (EEAS), "EU Code Week – Programming, Robotics and Rights for Girls", EU Delegation to the UN Geneva. Available at: [https://www.eeas.europa.eu/delegations/un-geneva/eu-code-week-programming-robotics-and-rights-girls\\_en](https://www.eeas.europa.eu/delegations/un-geneva/eu-code-week-programming-robotics-and-rights-girls_en)

<sup>16</sup>STEAMBrace Project, "The Rise of STEAM Education: Why Arts Matter in STEM", 2025. Available at: <https://steambraceproject.eu/arts-in-stem-steam-education/>

### **Key Takeaway**

STEAMBrace's arts-integrated approach directly reinforces ERGO's own emphasis on the 'A' in STEAM — the integration of artistic and creative thinking into robotics and coding activities.

Its European coordination network (STEAM Alliance for Europe) is a potential dissemination channel for ERGO results beyond the consortium.

The STEAM Week format provides a model for ERGO's promotional events in WP5.

### **2.3.9 Girls Who Game (Dell Technologies, Microsoft & Intel)**

Girls Who Game (GWG) is an extracurricular programme created by Dell Technologies in partnership with Microsoft and Intel, providing students across the world with an opportunity to explore STEM through gaming.<sup>17</sup> Using Minecraft: Education Edition as the primary platform, participants develop computational thinking, critical thinking, communication, collaboration, creativity, and problem-solving skills in a learner-driven environment. The programme targets girls primarily but is open to all, and operates through school clubs with virtual mentorship from STEM professionals.

Since its launch in 2019, GWG has grown to over 75 clubs per cohort across the United States, Canada, and beyond. The programme deliberately uses gaming as an entry point — recognising that the word 'code' can be off-putting to girls who would otherwise engage enthusiastically with technology. Mentors share non-linear career paths to demonstrate that there is no single route to a STEM career.

### **Key Takeaway**

GWG demonstrates the power of using familiar, engaging digital environments (gaming) as a gateway to computational thinking — a principle directly applicable to ERGO's simulation-based approach.

The mentorship model, where diverse STEM professionals share personal career stories, provides inspiration for the role model content in ERGO's WP3 eCourse.

The club-based, after-school format shows how schools can sustain engagement beyond one-off activities.

### **2.3.10 Girls Code Too (UK)**

Girls Code Too is a UK-based initiative passionate about introducing girls to rewarding careers in technology through coding and design workshops.<sup>18</sup> Its freely available micro:bit lesson resources offer a practical, no-equipment-barrier entry point into physical computing and coding — students use the micro:bit microcontroller to build beginner projects including name tags, simple sensors, and interactive games. The lessons are structured as downloadable workshop guides, making them directly usable by classroom teachers without specialist training.

The micro:bit platform itself has been used across Europe — including in refugee camp settings in Athens, British Council summer schools in Spain, and digital literacy programmes

---

<sup>17</sup>Dell Technologies, "Girls Who Game – Empowering the Next Generation in STEM". Available at: <https://www.dell.com/en-us/lp/dt/girls-who-game>

<sup>18</sup>Girls Code Too UK, "Micro:bit Lessons". Available at: <https://girlscodetoo.co.uk/lessons/microbit/>

across the Balkans — as a vehicle for coding education. Its combination of physical and digital elements makes it particularly engaging for students who might not respond to screen-only coding environments.

### **Key Takeaway**

The freely downloadable workshop format of Girls Code Too is directly replicable by ERGO school partners, requiring no specialist resources beyond the micro:bit hardware.

The micro:bit platform bridges physical making and digital coding — mirroring ERGO's interest in combining simulation tools with tangible learning outcomes.

The UK focus is complementary to ERGO's European scope: materials can inspire locally adapted versions in partner countries.

## **2.3.11 greenlight for girls (g4g)**

Greenlight for girls (g4g) is an international non-profit organisation based in Brussels, dedicated to inspiring girls of all ages and backgrounds to pursue STEM subjects by introducing them to science in fun and exciting ways. Its flagship activity is the g4g Day — a one-day, girl-focused event featuring hands-on workshops run by volunteer STEM professionals, open to the local community and designed to reach girls in less advantaged communities.

Since its inception, g4g has reached over 60,000 girls and young women through a network of more than 13,000 STEM professionals and mentors, with over 500 face-to-face events in 39 countries.<sup>19</sup> In 2024, g4g partnered with Microsoft Portugal and Pavilhão do Conhecimento to organise an event in Lisbon specifically aimed at inspiring 200 middle-school girls to pursue STEM careers. G4g also collaborates with the Erasmus+ project EITIC-EU, creating open resources for teachers and parents to support girls in STEAM. The organisation operates across Europe with a particular emphasis on reaching schools in Brussels, Denmark, Germany, Portugal, and the UK.

### **Key Takeaway**

G4g's model of 100+ hands-on workshops run by role model volunteers in a single day event is highly replicable for ERGO multiplier events in WP5.

Its explicit focus on reaching girls from less advantaged communities aligns directly with ERGO's equity objectives and the profile of partner schools such as 1 EPAL Korydallos (P5).

The EITIC-EU collaboration shows how g4g resources can be integrated into Erasmus+ projects — a direct precedent for ERGO.

## **2.3.12 EQUALS-EU – Europe's Regional Partnership for Gender Equality in the Digital Age**

EQUALS-EU is a Horizon 2020-funded coordination and support action that promotes gender equity in social innovation and digital inclusion across 29 EU Member States and Associated Countries.<sup>20</sup> The project builds capacity through multilateral partnerships, creates inclusive innovation ecosystems, and conducts a combination of online and face-to-face mentoring,

---

<sup>19</sup>greenlight for girls, "g4g and Microsoft / Ciência Viva Collaborate to Bridge Gender Gap in STEM", February 2024. Available at: <https://www.greenlightforgirls.org/2024-hot-off-the-press/2024/2/15/g4g-microsoft-and-cincia-viva-collaborate-to-bridge-gender-gap-in-stem-gf54g>

training, and lifelong learning programmes focused on empowering women and girls in STEM and ICT. Its activities include interdisciplinary hackathons, innovation camps, high-level seminars alongside UN and EU flagship events, and a dedicated professional network for women and girls in social innovation and entrepreneurship.

The project's scope and approach are particularly relevant for understanding the systemic and policy-level conditions for gender equity in digital education. EQUALS-EU has produced open Gender Equity Tools for digital inclusion that can inform the design of inclusive policies and practices, and its activities have reached women and girls across Europe and beyond through its consortium of 19 organisations from 15 countries.

### Key Takeaway

EQUALS-EU's systemic approach — covering access, skills, and leadership simultaneously — provides a policy framework against which ERGO's own gender-inclusive activities can be positioned.

Its hackathon and innovation camp formats are directly adaptable for ERGO's participatory design workshops in WP4.

The project's open Gender Equity Tools are a ready-made resource that ERGO teacher trainers can reference when designing inclusive activities.

## 2.4. Comparative Overview of Good Practices

The table below summarises the practices reviewed, their scope, and their key relevance to ERGO activities.

| Practice / Initiative                   | Scope                       | Key Features                                                                               | Relevance to ERGO                                                                          | Website                                                                                                                                                                              |
|-----------------------------------------|-----------------------------|--------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Girls Go Circular / Girls Go STEM (EIT) | Europe-wide (35+ countries) | Free online modules; AI, robotics, sustainability; 48,000+ girls; 24 languages             | Model for ERGO's online, challenge-based robotics projects and WP4 design criteria         | <a href="http://eit-girlsgocircular.eu">eit-girlsgocircular.eu</a>                                                                                                                   |
| ESTEAM Fests (EISMEA)                   | Europe-wide events          | Technical workshops + role models + confidence-building; FabLab visits; ministerial co-org | Model for ERGO multiplier events in WP5; combines making, digital skills and inspiration   | <a href="http://eismea.ec.europa.eu/esteam-fests-and-communities-girls-and-women_en">eismea.ec.europa.eu/esteam-fests-and-communities-girls-and-women_en</a>                         |
| BridgeSTEAM – Girls in STEAM Academy    | Cyprus, expanding EU        | Free 2-day coding programme; 93% increased STEAM interest; 89% improved confidence         | Provides outcome benchmarks and pedagogical model for ERGO's girl-focused pilot evaluation | <a href="http://digital-skills-jobs.europa.eu/en/inspiration/good-practices/girls-steam-academy">digital-skills-jobs.europa.eu/en/inspiration/good-practices/girls-steam-academy</a> |
| STREAM IT (Erasmus+)                    | Multi-country               | Structural barriers research; simulation                                                   | Frames gender inclusion systemically;                                                      | <a href="http://streamit-project.eu">streamit-project.eu</a>                                                                                                                         |

<sup>20</sup>EQUALS-EU, "Europe's Regional Partnership for Gender Equality in the Digital Age". Available at: <https://equals-eu.org/>

|                                           |                                        |                                                                                               |                                                                                                  |                                                                                                                                                                                                             |
|-------------------------------------------|----------------------------------------|-----------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                           | (EU)                                   | + female panels; mentoring networks                                                           | informs ERGO's teacher training and evaluation design                                            |                                                                                                                                                                                                             |
| Girls in ICT Talks (Malta)                | Malta (school-based)                   | Women Ambassadors in schools; Year 7 target; storytelling + relatable ICT examples            | School-embedded model replicable by ERGO school partners; emphasises accessibility               | <a href="https://mdia.gov.mt/services/women-in-ict">mdia.gov.mt/services/women-in-ict</a>                                                                                                                   |
| EU Code Week – Girls in Digital / GCIB    | Europe-wide                            | 3–4h online/in-person sprints; female role models; Replication Kit; no experience needed      | Direct synergy with ERGO — same networks, same partner (EDUMOTIVA), replicable model             | <a href="https://codeweek.eu/girls-in-digital-week">codeweek.eu/girls-in-digital-week</a>                                                                                                                   |
| Programming, Robotics & Rights for Girls  | Geneva (EU Delegation)                 | Collaborative robots + gender rights framing; 150 girls aged 6–17; civic + technical blend    | Model for linking ERGO robotics activities to gender equality narrative in teacher materials     | <a href="https://eeas.europa.eu/delegations/un-geneva/eu-code-week-programming-robotics-and-rights-girls_en">eeas.europa.eu/delegations/un-geneva/eu-code-week-programming-robotics-and-rights-girls_en</a> |
| STEAMBrace (Horizon Europe)               | Europe-wide                            | Arts-integrated STEAM; STEAM Alliance for Europe; STEAM Week; VR/AR; policy network           | Arts + STEAM integration reinforces ERGO's 'A' emphasis; EU-level dissemination channel          | <a href="https://steambraceproject.eu">steambraceproject.eu</a>                                                                                                                                             |
| Girls Who Game (Dell / Microsoft / Intel) | USA, Canada, expanding globally        | Gaming-based STEM learning; Minecraft; 75+ clubs; mentorship; computational thinking          | Gaming as coding entry point mirrors ERGO's simulation approach; mentorship model for WP3        | <a href="https://dell.com/en-us/lp/dt/girls-who-game">dell.com/en-us/lp/dt/girls-who-game</a>                                                                                                               |
| Girls Code Too (UK)                       | UK (resources freely available)        | Free micro:bit workshop guides; physical computing; no specialist training needed             | Replicable workshop format; micro:bit bridges physical making and digital coding for ERGO pilots | <a href="https://girlscodetoo.co.uk">girlscodetoo.co.uk</a>                                                                                                                                                 |
| greenlight for girls (g4g)                | International (Europe, Americas, Asia) | One-day STEM events; 60,000+ girls reached; volunteer STEM role models; free community events | Multiplier event model for ERGO WP5; equity focus matches ERGO partner school profiles           | <a href="https://greenlightforgirls.org">greenlightforgirls.org</a>                                                                                                                                         |
| EQUALS-EU (Horizon 2020)                  | 29 EU countries + Global South         | Gender equity in digital innovation; hackathons; mentoring; open Gender Equity Tools          | Policy framework and open tools complement ERGO's inclusive design approach in WP4               | <a href="https://equals-eu.org">equals-eu.org</a>                                                                                                                                                           |

## 2.5. Key Lessons and Recommendations

Across the practices reviewed, a consistent set of enabling principles emerges.

### 2.5.1 Pedagogical Design Principles

- Ground activities in real-world, socially meaningful challenges: sustainability, health, equality, community problems. Girls are more strongly motivated by purpose-driven technology than by technology for its own sake.
- Use creative and culturally accessible entry points (games, art, storytelling, social challenges) rather than abstract coding exercises to lower self-exclusion barriers.
- Integrate the Arts into STEM activities deliberately: arts-based approaches have been shown to increase girls' engagement, self-expression, and sense of belonging in STEAM contexts (STEAMBrace; greenlight for girls).
- Ensure activities are low-floor — accessible with no prior experience — while still offering depth for those who want to explore further.
- Combine technical skill-building with discussion of gender equality, career possibilities, and girls' right to participate in science and technology.
- Make female role models visible throughout: invite female professionals, feature women-led projects, highlight female researchers and engineers.

### 2.5.2 Structural and Organisational Conditions

- Train teachers first: cascading models that empower educators to deliver inclusive activities to their own students are the most sustainable and scalable approach at European level.
- Ensure all activities are free of charge and require no expensive equipment, directly addressing the socio-economic barriers that disproportionately affect girls in under-resourced schools.
- Embed activities within regular school hours and curriculum wherever possible, to maximise participation and avoid creating an 'opt-in' dynamic that self-selects only the most motivated girls.
- Connect school activities to wider European networks (EU Code Week, eTwinning, Scientix, Girls Go STEM, EQUALS-EU, greenlight for girls) to amplify reach, sustainability, and visibility.
- Seek institutional co-endorsement — from education authorities, universities, or ministries — to signal legitimacy and encourage broader school participation.

### 2.5.3 Evaluation Metrics to Adopt

- Percentage of girls reporting increased interest in STEAM careers after ERGO pilot activities (benchmark: 93% in BridgeSTEAM; see note 9).
- Percentage of girls reporting improved self-confidence in using digital and robotics tools (benchmark: 89% in BridgeSTEAM; see note 9).
- Drop-out rate among girl participants during pilot activities (ERGO target: no more than 1 in 10).
- Percentage of teachers who feel confident implementing inclusive robotics activities after ERGO training (target: 70% positive responses, per ERGO proposal).

- Qualitative reflections from girls on their sense of belonging and engagement in STEAM contexts.

## 2.6. Conclusion

The practices reviewed in this chapter collectively demonstrate that empowering girls in STEAM is not a matter of a single intervention or formula, but rather a consistent set of pedagogical principles applied across a range of formats, settings, and partnerships. Whether through intensive coding sprints, school ambassador visits, project-based EU-funded curricula, or robotics workshops framed around rights and social justice, the most effective initiatives share a common DNA: they are hands-on, purposeful, free of heavy resource requirements, embedded in supportive adult relationships, and connected to broader communities and networks.

## 2.7. Chapter 2 References

All sources cited in this document are listed below in numerical order, corresponding to the footnote numbers used in the text.

- [1] European Commission, "She Figures 2024: Gender in Research and Innovation", Publications Office of the EU, released 11 February 2025. Available at: <https://op.europa.eu/en/publication-detail/-/publication/2d15c6d4-dae4-11ef-a76b-01aa75ed71a1>
- [2] European Commission, "Report on the State of the Digital Decade 2024", September 2024. Available at: <https://digital-strategy.ec.europa.eu/en/library/state-digital-decade-2024>
- [3] European Commission, "STEM Education Strategic Plan", European Education Area. Available at: <https://education.ec.europa.eu/focus-topics/digital-education/action-plan/Women-participation-in-STEM>
- [4] International Association for the Evaluation of Educational Achievement (IEA), "International Computer and Information Literacy Study (ICILS) 2023". Available at: <https://www.iea.nl/studies/iea/icils/2023>
- [5] ERGO Partnership, "ERGO Needs Analysis Survey Report", 2024 (internal project document, annexed to ERGO application KA220-SCH-D0356924).
- [6] JA Europe, "Why Investing in Girls in STEM is Essential for a Thriving Future", February 2025. Available at: <https://jaeurope.org/why-investing-in-girls-in-stem-is-essential-for-a-thriving-future/>
- [7] STEAMBrace Project, "EU Initiatives for Gender Equality in STEM Education", October 2025. Available at: <https://steambraceproject.eu/eu-initiatives-for-gender-equality-in-stem/>
- [8] European Innovation Council and SMEs Executive Agency (EISMEA), "ESTeAM Fests and Communities for Girls and Women". Available at: [https://eisma.ec.europa.eu/esteam-fests-and-communities-girls-and-women\\_en](https://eisma.ec.europa.eu/esteam-fests-and-communities-girls-and-women_en)
- [9] EU Digital Skills and Jobs Platform, "Girls in STEAM Academy – Good Practice". Available at: <https://digital-skills-jobs.europa.eu/en/inspiration/good-practices/girls-steam-academy>
- [10] STREAM IT Project, "STEAM for All: Best European Practices in Gender-Inclusive Education", May 2025. Available at: <https://streamit-project.eu/2025/05/01/steam-for-all-best-european-practices-in-gender-inclusive-education/>
- [11] EU Digital Skills and Jobs Platform, "Empowering Future Innovators: 2024–2025 Edition of Girls in ICT Talks Marks Another Successful Year", May 2025. Available at: <https://digital-skills->

jobs.europa.eu/en/latest/news/empowering-future-innovators-2024-2025-edition-girls-ict-talks-marks-another-successful

[12] EU Code Week, "Girls in Digital – Girls Code it Better Initiative". Available at: <https://codeweek.eu/girls-in-digital-week>

[13] EU Digital Skills and Jobs Platform, "Highlights from the EU Code Week 2024 Launch Event", October 2024. Available at: <https://digital-skills-jobs.europa.eu/en/latest/news/highlights-eu-code-week-2024-launch-event>

[14] European External Action Service (EEAS), "EU Code Week – Programming, Robotics and Rights for Girls", EU Delegation to the UN Geneva. Available at: [https://www.eeas.europa.eu/delegations/un-geneva/eu-code-week-programming-robotics-and-rights-girls\\_en](https://www.eeas.europa.eu/delegations/un-geneva/eu-code-week-programming-robotics-and-rights-girls_en)

[15] European Health and Digital Executive Agency (HaDEA), "Inspiring the Next Generation: Girls in ICT Day 2025", April 2025. Available at: [https://hadea.ec.europa.eu/news/inspiring-next-generation-girls-ict-day-2025-2025-04-23\\_en](https://hadea.ec.europa.eu/news/inspiring-next-generation-girls-ict-day-2025-2025-04-23_en)

[16] EU STEM Coalition / Horizon Magazine, "The Art of Inclusion: EU Moves to Empower Girls in Science", February 2025. Available at: <https://www.stemcoalition.eu/publications/the-art-of-inclusion-eu-moves-to-empower-girls-in-science/>

[17] STEAMBrace Project, "The Rise of STEAM Education: Why Arts Matter in STEM", 2025. Available at: <https://steambraceproject.eu/arts-in-stem-steam-education/>

[18] Dell Technologies, "Girls Who Game – Empowering the Next Generation in STEM". Available at: <https://www.dell.com/en-us/lp/dt/girls-who-game>

[19] Girls Code Too UK, "Micro:bit Lessons". Available at: <https://girlscodetoo.co.uk/lessons/microbit/>

[20] Scientix, "STEM Job Profiles". Available at: <https://www.scientix.eu/resources/knowledge/stem-job-profiles>

[21] greenlight for girls, "g4g and Microsoft / Ciência Viva Collaborate to Bridge Gender Gap in STEM", February 2024. Available at: <https://www.greenlightforgirls.org/2024-hot-off-the-press/2024/2/15/g4g-microsoft-and-cincia-viva-collaborate-to-bridge-gender-gap-in-stem-gf54g>

[22] EQUALS-EU, "Europe's Regional Partnership for Gender Equality in the Digital Age". Available at: <https://equals-eu.org/>

# Chapter 3: Simulation Tools and Technologies for Online Robotics Education

## 3.1 Introduction

In the contemporary landscape of STEM (Science, Technology, Engineering, and Mathematics) education, visual programming environments have emerged as important resources for fostering computational thinking among students. By bridging the gap between abstract logic and concrete execution, these environments provide an intuitive entry point for young learners to engage with the complex principles of robotics and software design.

This chapter explores a selection of the main simulation tools and digital toolboxes currently available for robotics education in schools. These platforms were chosen not only for their technological relevance but also for their distinct pedagogical methodologies—ranging from block-based programming to interactive, game-like simulation environments. Each resource presented here offers an approach to overcoming technical barriers, such as hardware limitations, while supporting educators in creating inclusive, accessible, and stimulating learning pathways.

The following sections provide a detailed analysis of these tools, their educational applications, and their potential to transform both in-person and distance learning environments.

The following Toolboxes have been selected for the following analysis:

- Open Roberta Lab
- Ozobot
- Scratch
- MakeCode
- Tinkercad

The following resources have been selected, too:

- Cardboard Robots
- GrabCAD
- Robot Guide
- The Poppy Project
- WRO Education
- LEGO Education

Besides its main description, each tool is analysed considering the following elements:

- **Accessibility:** Evaluates the inclusivity of the platform, specifically regarding technical barriers (e.g., installation requirements), linguistic support, and the presence of features that accommodate diverse learning needs and varying hardware capabilities.
- **Programming Environment:** Describes the interface utilized (e.g., block-based vs. text-based), the presence of simulation capabilities, and the mechanism for deploying code to hardware, highlighting the transition from abstract logic to physical execution.
- **Age Group:** Identifies the primary developmental target for the tool, assessing the scalability of the activities from early primary education through to secondary and vocational levels.
- **Cost:** Analyzes the financial feasibility for institutions, distinguishing between free, open-source web platforms and hardware-dependent ecosystems that require initial capital investment.
- **Learning Objectives:** Outlines the core pedagogical goals fostered by the tool, ranging from computational thinking and problem-solving to mechanical engineering and collaborative design.
- **Classroom Usability:** Assesses the practical integration of the tool within a lesson plan, considering the availability of support materials, curriculum alignment, and ease of management for educators.

For a comprehensive and updated overview of the collection of identified resources, please refer to the following repository:

[ERGO-WP2-ResourceCollection.xlsx](#)

Besides software simulation tools and online repositories, the collection includes more resource types:

- Youtube thematic channels
- Lecture series
- Massive Open Online Courses (MOOCS)

## 3.2 Simulation tools

### 3.2.1 Open Roberta Lab

Link: <https://lab.open-roberta.org/>



#### Description

Open Roberta Lab is a comprehensive, free, cloud-based programming environment designed to facilitate robotics education for children and young people. Established in 2014 in Sankt Augustin, it serves as a technological extension of the broader "Roberta" project—an educational initiative launched by the Fraunhofer IAIS in Germany in 2002. Developed by a team including Beate Jost, Reinhard Budde, Thorsten Leimbach, and Achim Kapusta, the platform enables users to program various robotic systems, including Lego Mindstorms, Arduino, BBC Micro-Bit, Calliope mini, NAO, and NXT.<sup>1</sup>

The platform's programming language, NEPO (an acronym for "New Easy Programming Online," or "OPEN" spelled backward), is open-source and rooted in Google's Blockly library, drawing inspiration from MIT's Scratch. In this environment, programming is intuitive: users snap together visual blocks representing specific robot functions. These blocks are color-coded by data type to ensure logical connections: light blue for logical values, dark blue for numbers, green for strings, yellow for colors, purple for lists, and pink for connections.<sup>1</sup>

A key feature of Open Roberta Lab is its 2D simulation environment. This allows users to test their created programs immediately, providing instant feedback on robot behavior—such as obstacle avoidance—without requiring physical hardware. In certain versions, the simulation environment supports two-wheeled robots equipped with ultrasonic, touch, and color sensors, as well as LED displays.<sup>1</sup>

Designed for accessibility, the platform functions entirely within a web browser, requiring no installation or registration. Initially funded by Google.org, it is now utilized in 120 countries and is available in multiple languages, including Italian, English, German, French, and Spanish. The initiative emphasizes collaborative learning, engaging teachers, universities, and students, with a specific guiding principle of supporting female students in STEM. Since 2016, the Scuola di Robotica has acted as the Italian partner, producing resources and tutorials to promote the platform within Italy.

## Accessibility

Open Roberta Lab lowers the barriers to entry for robotics education through a multifaceted approach:

- **Technical Accessibility:** The platform functions entirely within a web browser and requires no installation or registration. This "zero-install" design makes it compatible with a wide array of devices, including low-specification hardware often found in schools, such as Chromebooks, iPads, and older PCs.
- **Hardware Independence:** A standout feature is its integrated 2D simulation environment. By allowing users to test programs virtually, the platform ensures that students can learn robotics and control logic even if their school lacks physical hardware, significantly reducing the cost and logistical barriers to entry.
- **Cognitive Accessibility:** The use of NEPO—a block-based programming language rooted in Google's Blockly—makes coding intuitive. By snapping together color-coded blocks, students avoid the syntax errors common in text-based programming, allowing beginners to focus entirely on logic and problem-solving.
- **Linguistic and Geographic Inclusivity:** The platform is available in multiple languages (including Italian, English, German, French, and Spanish) and is currently utilized in over 120 countries. Its free, open-source nature ensures that cost does not limit its implementation in diverse educational settings, from under-resourced schools to distance learning programs.
- **Inclusive Pedagogy:** The initiative explicitly emphasizes collaborative learning and supporting female students in STEM, demonstrating a commitment to accessibility not just in technical terms, but in fostering inclusive classroom participation.

**Programming Environment:** Open Roberta Lab is a cloud-based, web-native platform that requires no software installation. It utilizes NEPO, an intuitive block-based programming language rooted in Google's Blockly. This interface allows users to construct programs by dragging and dropping visual blocks, which eliminates syntax errors and helps beginners focus on computational logic. The platform supports a robust 2D simulation environment, allowing students to test robot behaviors virtually before deploying code to physical hardware.

**Age Group:** Primarily targeted at children and young people, the platform is suitable for students across the K-12 spectrum, ranging from primary through secondary school.

**Cost:** The platform is entirely free and open-source, ensuring it is accessible to all schools and organizations regardless of budget constraints.

**Learning Objectives:** The primary objectives include developing computational thinking, mastering fundamental robotics and control logic, understanding the interaction between hardware (sensors and actuators) and software, and fostering collaborative problem-solving skills through iterative design.

**Classroom Usability:** The platform offers high usability. Its browser-based nature ensures compatibility with low-spec hardware like Chromebooks and tablets. The integrated simulator allows for learning even without physical robot kits, making it highly adaptable for both in-

person and distance learning environments. It is further supported by extensive tutorials and community resources that reduce preparation time for educators.

### 3.2.2 OZObot

<https://ozobot.com/>



#### **Description**

OZObot was founded in 2012 by Nader Hamda with an ambitious goal: to create innovative ways for young people to play, learn, and interact in an increasingly digital world. The creative spark became concrete when Hamda's 10-year-old daughter suggested the name "OZObot" for these small robots. After two years of testing and prototyping, the first model, Bit, was presented to the public. Ozobot's great innovation was twofold: on one hand, the ability to program the robots without a screen using simple color codes; on the other hand, access to a more advanced block programming environment called OzoBlockly. Over time, the ecosystem has expanded: after Bit came Evo, a more evolved robot, connected via Bluetooth, updatable, and capable of interacting with a dedicated app, becoming a "creative companion that grows with the student." Today, Ozobot's mission is to give all children the opportunity to create with technology.

The Ozobot toolbox is not a single software, but a constantly evolving ecosystem of tools designed to offer coding and computer science challenges at home and school, "with or without a robot."

Here are its main functions:

#### ***For Home***

This area focuses on the Evo app and offers different modes of play and learning.

- Games on the app:
  - "OzoLaunch": a challenge that transforms the robot launch into a target game, combining physical movement and programming;
  - creating personalized tricks, "Custom Tricks": users can program customized sequences of lights, sounds, and movements, stimulating creativity;
  - other game modes such as star collection and chat with friends via "Ozoji," which are emojis that Evo robots execute.
- OzoBlockly programming on tablet: a fundamental function that allows creating and uploading complex programs to the robot directly from the app.
- Direct Control (Drive): a mode that transforms the smartphone into a remote control to manage Evo's speed, sounds, and lights.

#### ***For School***

This toolbox is designed for teachers and offers increasingly rich and structured resources:

- Ready-to-Run Lessons: a free digital curriculum with ready-to-download and use lessons covering various STEM topics;
- Ozobot Challenge Mats: interactive and immersive tools designed for middle and high school. Each “mat” can host up to 8 students (in pairs) and 4 Evo robots programmed simultaneously. Each set includes the mat, challenge cards, and accessories, and comes with lessons. Four themes are available, transforming the classroom into an adventurous environment:
  - Mission to Mars: students simulate a space mission on Mars, programming Evo to locate supplies and overcome obstacles like canyons and volcanoes;
  - Ocean Explorer: students explore the deep sea, from the intertidal zone to the midnight zone, learning the basics of block programming;
  - Soccer and Basketball: combine a passion for sport with coding. Students program Evo to dribble, pass, shoot at the basket or goal, evading defenders.
- Creative challenges for the first day of school: 8 creative challenges for students starting from the first day, many of which do not even require the physical robot. Some examples are:
  - Program Your Partner: in pairs, one acts as the “programmer” and the other as the “robot,” who must follow precise instructions to complete tasks.
  - Emoji Emotions Pathway: students code their emotions into a path, gentle curves for calm and rapid changes of direction for euphoria.
  - Debug the Teacher’s “Mistakes”: the teacher presents a deliberately “bugged” code, and students, in groups, must find and correct the errors.
- Adaptable lessons for virtual learning: resources designed for distance learning, many of which do not require possessing the physical robot. Examples include:
  - Pop Star: teaches the concept of “function” by programming Evo to become a pop star who sings and dances.
  - Mad Dash: students use measurements and angles to program Evo in a grocery race during a pandemic.

The Ozobot toolbox is, therefore, much more than a list of functions; it is a testament to an evolution: from a family idea to a mature, flexible, and research-based educational ecosystem. It offers paths ranging from free play to structured teaching, from analog to digital programming, and from in-person to virtual learning.

### Accessibility

- **Versatile Modes:** Offers both screen-free (color coding) and block-based programming, accommodating different developmental stages.
- **"With or Without a Robot":** Many resources, including virtual lessons and digital challenges, do not require the physical robot, making it accessible for home learning or budget-constrained environments.
- **Scalable Ecosystem:** The "Evo" robot is designed to grow with the student, transitioning from simple physical color-code interaction to complex block-based programming.
- **Supportive Materials:** Provides structured curricula, teacher resources, and ready-to-run lesson plans that lower the entry barrier for educators.

**Programming Environment:** Ozobot features a dual-mode environment. It offers an analog, screen-free method using sequences of color codes to control the robot's actions, and a digital, visual block-based environment called OzoBlockly (rooted in Blockly), which allows for complex program creation and deployment.

**Age Group:** Suitable for a wide range of students, from early childhood (using simple color codes) through middle and high school (using OzoBlockly).

**Cost:** Requires investment in physical hardware (Ozobot robots), though many curricular resources, lesson plans, and remote learning activities are provided for free.

**Learning Objectives:** The primary objectives include mastering fundamental coding logic (sequence, loops, variables), bridging the gap between visual and digital abstraction, developing STEAM skills, and fostering creative problem-solving through project-based challenges.

**Classroom Usability:** High. The platform is highly adaptable for both in-person and distance learning. It is supported by an extensive library of ready-to-use lessons, challenge mats, and cross-curricular activities that integrate coding with subjects like music and mathematics.

### 3.2.3 Scratch

<https://scratch.mit.edu/>



#### Description

Developed by the Lifelong Kindergarten Group at the MIT Media Lab, Scratch stands as a foundational pillar in contemporary educational technology. It is a visual programming environment specifically designed to empower young people to create their own interactive stories, games, and animations. Unlike traditional programming languages that require memorizing complex syntax, Scratch utilizes a "low floor, high ceiling" design philosophy. This means that beginners can achieve immediate success with simple scripts, while experienced users can leverage advanced features to build complex, sophisticated projects.

The heart of the Scratch experience is its event-driven, block-based programming language. Users assemble programs by snapping together color-coded, puzzle-piece blocks, each representing a command or action. These scripts control "Sprites"—interactive characters or objects—within "Scenes" (or stages) defined by customizable backdrops. This tactile, visual approach removes the barrier of syntax errors, allowing students to focus entirely on logical reasoning and computational thinking.

Accessibility is a core tenet of the platform. Being entirely browser-based, Scratch requires no software installation, running smoothly on a wide array of hardware, from modern desktop computers to school-provided tablets. This web-native nature, combined with its availability in numerous languages, ensures that it can reach students across diverse socioeconomic and geographic backgrounds. Furthermore, the platform's "remix" culture is revolutionary: it allows users to view, study, and modify the code of others' projects, fostering a global, collaborative learning community where knowledge is shared and built upon collectively.

In the classroom, Scratch is highly versatile. Its pedagogical value extends far beyond computer science, acting as a bridge between disciplines. As evidenced by projects like "Educational Games with Scratch," which combined literature with programming, it can integrate humanities—such as history or art—with scientific principles, enabling students to demonstrate their understanding of complex topics through creative, playful products. This cross-curricular adaptability, supported by a vast ecosystem of tutorials and ready-made lessons, makes Scratch an essential tool for educators aiming to foster not only coding skills but also creative problem-solving, collaboration, and narrative expression in the 21st-century classroom.

### **Accessibility**

Scratch promotes accessibility through an inclusive design focused on creative learning:

- **Low Floor:** The visual block interface eliminates the syntax barriers of traditional code, making programming accessible to beginners and those with learning difficulties.
- **Design 'Low Floor, High Ceiling':** The platform allows for simple, immediate projects while offering advanced tools for complex creations, adapting to diverse skill levels.
- **Linguistic and Social Inclusivity:** It supports numerous languages and fosters a global collaborative environment. The 'remix' philosophy allows students to learn by modifying others' projects, creating a supportive peer-to-peer system.
- **Web-Native Technology:** Being entirely browser-based, it requires no complex installation and runs on any internet-connected device, making it ideal for schools with varied technology.

**Programming Environment:** Scratch offers a visual block-based programming environment using 'puzzle-piece' logic. It is an event-driven system where users define the behavior of 'Sprites' (characters/objects) within 'Scenes' (backgrounds) by assembling color-coded command blocks into scripts.

**Age Group:** Primarily designed for young people aged 8 to 16, but flexible for use in primary school contexts (with ScratchJr) or for complex projects in secondary education.

**Cost:** The platform is entirely free and open-source, ensuring universal access for schools and students.

**Learning Objectives:** Primary objectives include developing computational thinking, logical design capabilities, creative expression, problem-solving, and social collaboration through sharing and remixing projects.

**Classroom Usability:** High. Its browser-based nature and vast online community provide teachers with endless tutorials, ready-made lessons, and inspiration, making it a flexible tool for integrating coding across subjects from sciences to humanities.

### 3.2.4 Makecode

<https://makecode.microbit.org/>



#### Description

MakeCode, developed by Microsoft, represents a sophisticated yet accessible ecosystem designed to introduce students to the world of physical computing and software engineering. Since its inception, MakeCode has evolved into a premier tool for modern STEM classrooms, distinguished by its unique ability to bridge the gap between novice block-based programming and professional text-based languages like JavaScript and Python. This "low floor, high ceiling" approach is fundamental to its design, ensuring that it remains approachable for elementary students creating their first animations while offering the depth required for advanced learners to manipulate hardware at a lower level.

The platform supports a vast array of hardware, most notably the BBC micro:bit, Adafruit Circuit Playground, and various robotics kits, acting as a crucial interface between student-written code and physical electronic components. Its core functionality revolves around a robust, browser-based environment that facilitates a seamless transition from visual logic to syntax-based coding. Users can toggle between these modes at any time, allowing students to visualize how blocks translate into code, which serves as a powerful scaffolding technique for computer science instruction.

A defining strength of MakeCode is its integrated, high-fidelity simulator. This tool provides an immediate feedback loop by allowing students to execute and observe their code's behavior in a virtual environment. This is particularly valuable in settings where physical hardware might be limited or unavailable, as it permits iterative testing and debugging without the need for constant hardware flashing. When physical deployment is possible, the process is streamlined to an efficient drag-and-drop workflow that compiles programs directly into machine-ready formats.

Beyond its standard features, MakeCode is highly extensible. Through a community-driven repository system, educators and developers can create and import custom extensions to control specific sensors, actuators, and unique hardware peripherals. This extensibility ensures the platform remains relevant as new technologies emerge, enabling personalized learning pathways that cater to specific classroom projects. By fostering this blend of structured, guided learning—via built-in "Skillmaps"—and open-ended creative exploration, MakeCode empowers

students not just to learn coding, but to design, prototype, and build interactive technology that interacts with the physical world.

### **Accessibility**

MakeCode lowers barriers to entry through a platform designed for diverse learning environments:

- **Zero-Install Web-Based Environment:** It runs entirely within modern web browsers, eliminating the technical barriers often associated with software installation. This makes it an ideal solution for low-specification hardware, such as school Chromebooks, iPads, or standard PCs.
- **Guided Learning:** Built-in, step-by-step interactive project paths (Skillmaps) support self-paced learning, allowing students to progress independently.
- **Streamlined Deployment:** The platform simplifies code transfer via a simple drag-and-drop mechanism (compiling to .HEX files), reducing technical troubleshooting.
- **Extensibility:** Support for community-made plugins via GitHub allows educators and advanced users to integrate custom code blocks for specific sensors or hardware.

**Programming Environment:** MakeCode utilizes a block-based programming interface rooted in visual logic. It is specifically designed for hardware interaction, allowing users to compile code directly into native machine code, providing a seamless bridge between visual programming and physical robotic or electronic components.

**Age Group:** Primarily targeted at K-12 education, with specific emphasis on middle and secondary school robotics and electronics projects.

**Cost:** The platform is free, ensuring accessibility for schools and individual students.

**Learning Objectives:** Primary objectives include developing computational thinking, understanding the interaction between hardware and software, mastering logical design, and promoting iterative problem-solving through hands-on robotics and electronics projects.

**Classroom Usability:** High. Its browser-based architecture and seamless hardware integration make it highly adaptable for both in-person and remote settings. Extensive tutorials and project paths significantly reduce the preparatory burden for educators.

## 3.2.5 Tinkercad

<https://www.tinkercad.com/>



### **Description**

Tinkercad, developed by Autodesk, is a versatile, free, browser-based platform that constitutes an essential entry point into the realms of 3D design, electronics, and programming. Designed with an emphasis on intuitiveness, it serves as a fundamental tool for K-12 education and introductory engineering courses, allowing users to move from conceptual design to virtual

testing without the need for sophisticated hardware or advanced technical skills. The platform is structured into three integrated environments: 3D Design, which uses constructive solid geometry (CSG) to facilitate rapid prototyping; Circuits, which acts as a virtual electronics laboratory for circuit assembly and testing; and Code, which unifies programming with hardware manipulation. By allowing users to simulate electronic behavior — from LEDs and resistors to Arduino microcontrollers — in a risk-free virtual environment, Tinkercad drastically lowers the barrier to entry for engineering education.

Its pedagogical value lies in its ability to support project-based learning across multiple disciplines. Students can design and 3D print prototypes, explore fundamental electronic principles, and learn to program simultaneously. Although not a professional-grade CAD suite or a high-fidelity robotics simulator, it excels as an accessible classroom tool that encourages experimentation and promotes the engineering design process. By allowing virtual prototyping of electronic subsystems, it prepares students for the challenges of physical robotics, helping them understand the critical interactions between software logic and hardware implementation before engaging in physical fabrication.

### **Accessibility**

- **Browser-based/No installation:** Runs entirely within modern web browsers, eliminating technical overhead and ensuring compatibility with various school hardware.
- **Intuitive interface:** The drag-and-drop design simplifies complex tasks, making it accessible to students with little or no prior engineering experience.
- **Safe experimentation:** Provides a virtual environment for testing circuits and logic without the risk of damaging physical components.
- **Collaborative cloud ecosystem:** Offers simple sharing and collaboration tools, supporting both in-person and distance learning contexts.
- **Rich resource library:** Offers a vast array of ready-to-use tutorials and projects, significantly reducing preparation time for educators.

**Programming Environment:** Tinkercad features an integrated and multimodal programming environment. The "Code" section supports both visual block-based programming (suitable for beginners) and text-based coding (using C/C++ for Arduino), allowing for a smooth transition from basic computational logic to standard syntax-based programming. This environment is deeply linked to the Circuits workspace, allowing real-time simulation of how code interacts with sensors, actuators, and microcontrollers.

**Age Group:** Primarily aimed at K-12 education, but widely used in introductory university courses for STEM and engineering fundamentals.

**Cost:** The platform is free, ensuring universal access for schools, educators, and individual students.

**Learning Objectives:** Primary objectives include developing fundamental skills in 3D design and digital fabrication, understanding electronic circuits and component behavior, mastering basic programming and computational thinking, and promoting the iterative engineering design process.

**Classroom Usability:** High. Its browser-based architecture makes it exceptionally adaptable for both physical and virtual classrooms. The integration of 3D modeling, electronics, and coding

into a single hub, supported by a vast ecosystem of tutorials, makes it a highly efficient tool for interdisciplinary STEM education.

### 3.3 Educational Projects and Initiatives Related to the Toolboxes

The dissemination of the above-described toolboxes analyzed is not limited to their technical availability but is realized in a multitude of school projects and training initiatives at local, national, and international levels.

An example is the European project “Coding Can Be Fun (CCBF),” promoted by the European School Education platform, in which students from Portugal, Bulgaria, and North Macedonia “will learn to break down complex tasks into smaller phases by programming robots to complete specific challenges.” Thanks to this project, students will be able to enhance logical thinking and strategic planning, and through the use of sensors, motors, and microcontrollers, they will understand the interaction between hardware and software. Furthermore, the work will also be carried out in groups, which will allow students to develop interpersonal skills such as task delegation and communication, while learning programming principles in Open Roberta's block environments. The project aims to encourage students to “design unique solutions to real-world problems using robotics, promoting creative thinking and iterative design.”

Remaining in the European context, during Code Week 2025-2026, a European initiative aimed at promoting coding and computational thinking, the Portuguese school EBS Tomás de Borba proposed the activity “Desenhar com Open Roberta Lab,” which we can literally translate as “Drawing with Open Roberta Lab.” This activity is presented as a challenge: “fourth-grade students are called upon to reproduce the images created by the lines left on the simulator screen by the virtual robot, by creating an appropriate code to achieve the desired effect.”

Another initiative is the project “Lessons with Open Roberta,” promoted by the Scuola di Robotica, “a non-profit association founded in 2000 on the initiative of a group of roboticists and scholars of human sciences with the aim of promoting the conscious engagement of robotics and new technologies.” This initiative was made available as a free downloadable resource, based on the use of the free Open Roberta Lab programming environment, which allows block programming of various robotic kits, such as Lego Mindstorms EV3, Wedo, and Arduino; it offers a simulation function that makes the activity accessible even in the absence of the physical robot, thus facilitating adoption in contexts with limited resources or in distance learning mode. Starting from this tool, the Scuola di Robotica created a training course for secondary school teachers called Open Roberta Reboot, aimed at providing interdisciplinary operational methodologies for coding and robotics, with particular attention to managing the alternation between in-person and distance learning, the role of play in learning, the development of soft skills, and participatory design. At the conclusion of the course, participants developed a “lesson plan,” or a didactic learning unit, subsequently collected in a digital publication that forms the core of the resource. Each lesson plan provides activities, tools, duration, school level, and a digital link to the presentations. The project offers teachers an operational and interdisciplinary repertoire for educational coding and robotics pathways.

A significant example of the pedagogical application of the Ozobot toolbox in Italy is represented by the European MUMABOT project (acronym for *MU*sic, *MA*thematics, and *roBOT*ics) launched

by the University of Milan Bicocca in February 2025. The project, which falls under the Erasmus Plus KA2 program within the international INTEC initiative (*Developing an innovative In-service Training model for ECEC Staff*), involved kindergarten and primary school teachers from the Comprehensive Institutes of Gorgonzola (IC Rita Levi Montalcini and IC Molino Vecchio). The initiative brought the small programmable Ozobot robot into classrooms with the goal of exploring the potential of intersections between different expressive languages, particularly music, mathematics, and educational robotics. The action-research hypothesis underlying the project is that the combined use of two or more different languages facilitates children's access and expression of knowledge, thereby promoting learning. The training course, which began in February 2025 with the participation of seven teachers from three school complexes: Collodi and Rodari kindergartens and Molino Vecchio primary school, subsequently extended to about thirty teachers on May 26, 2025, in a play-training meeting that united the two comprehensive institutes. The initiative testifies to a precise intention to create a network of contamination and exchange of good practices among professionals operating in the same territory, with a view to collaboration and mutual enrichment.

A further example of the pedagogical application of the Ozobot toolbox in the Italian school context is represented by the activity "Vowels with Ozobot," carried out as part of Code Week 2023 by the Collegno Marconi Comprehensive Institute – L. Moglia Primary School, class 1A, in Collegno (Turin). The activity, held in October 2023, involved students in programming a path that the robot would follow to write the vowels, combining the learning of coding fundamentals with writing practice. The didactic procedure involved several phases: preparing a sheet of paper or a whiteboard with a space dedicated to writing the vowels; using Ozobot's color programming language to create the necessary instructions; positioning the robot at the start of the path and initiating the programming; finally, observing the robot which, autonomously following the color path, represented the writing of the vowels, and then stopped upon completion of the letter. During the activity, the students were able to observe the robot as it moved along the path, visually demonstrating the tangible effect of the created programming instructions. This approach promoted the understanding of fundamental coding concepts, such as the sequence of instructions and the control of the robot's movement, allowing children to learn while having fun and to see the concrete result of their work. The initiative represents an effective example of how educational robotics, through tools like Ozobot, can be integrated into primary school to develop computational thinking in a transversal and playful way.

A significant example of the use of Scratch Junior in primary school is represented by the activity "A scary night!", carried out by the fourth grade of "La miniera del sapere" Primary School of the Spoleto 1 Comprehensive Institute, during Eu Code Week 2024. The activity, conducted in person, offered students the opportunity to explore visual block programming in a fun and engaging way thanks to the Scratch Junior environment. Children were invited to design their own Halloween-themed characters using the integrated graphics editor, giving free rein to their creativity. Subsequently, they programmed the characters to animate them, make them move, and interact with each other through dialogues and actions, thus learning the basics of programming in a playful and stimulating context. This initiative demonstrates how Scratch Junior, in its simplified version accessible even to the youngest children, can be effectively integrated into transversal didactic paths, combining the learning of computational thinking with expressive creativity and the festive theme of Halloween.

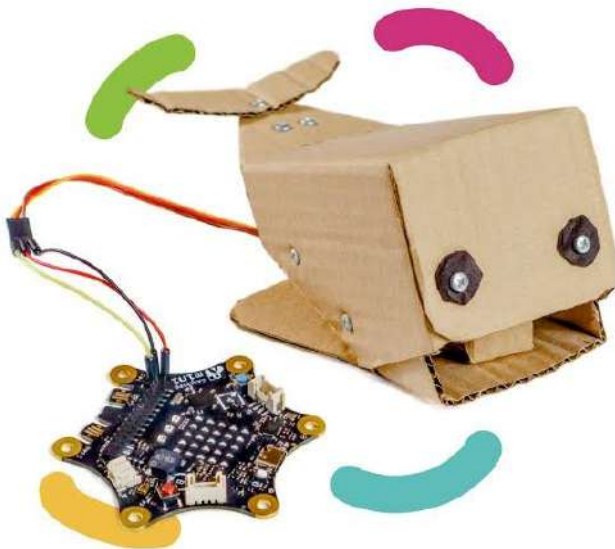
Another interesting application of Scratch in a school context comes from the "Felicia e Peppino Impastato" State High School, where the 3D class of the scientific high school carried out a

project under DM 65 entitled “Educational Games with Scratch: Learning Programming by Creating.” The initiative combined the study of Dante’s *Inferno* with visual programming, guiding students in the creation of an interactive game inspired by the Dantesque cantos. Through this process, the students learned fundamental coding concepts such as loops, conditionals, and variables, applying them directly to the creation of a playful-didactic product. The project represents a virtuous example of integration between humanities disciplines, such as literature, and scientific ones, such as programming, demonstrating how Scratch can act as a bridge between seemingly distant knowledge domains, fostering active, creative, and contextualized learning.

## 3.3 Resource repositories and other tools

### 3.3.1 Cardboards robot

<https://cardboard.lofirobot.com/>



Cardboard Robots, curated by Maciej Wojnicki, represents a unique pedagogical approach that merges the "Maker" movement with educational robotics. Unlike commercial kits that arrive in pre-fabricated plastic boxes, this project encourages students to construct their robotic systems from the ground up using widely available recycled materials—specifically cardboard—and basic, low-cost electronic components. This methodology transforms the learning process from a passive consumer experience into an active, creative, and engineering-focused endeavor. By requiring students to design the physical structure, chassis, and aesthetics of their robot, the platform cultivates a deep understanding of mechanical stability, structural integrity, and the physical constraints of engineering. It validates the philosophy that high-quality, effective learning tools do not require expensive proprietary hardware, but rather creativity, resourcefulness, and iterative design. The platform provides a suite of robot plans, apps, and building techniques designed for global replication, making it a powerful resource for educators working in diverse socio-economic contexts. It seamlessly integrates art, design, and engineering, allowing students to express their personality through their builds while

simultaneously mastering the logical foundations of control systems. This approach not only teaches robotics but also instills principles of sustainability and waste reduction, as students learn to repurpose everyday objects into functioning technological devices.

### Accessibility

- **Low-Cost Materiality:** By utilizing recycled materials like cardboard, the project eliminates the cost barrier of physical robotic kits, making it accessible even in under-resourced schools.
- **DIY Philosophy:** The plans are designed for easy replication worldwide, democratizing access to robotics education regardless of a school's budget or location.
- **Structural Understanding:** The process of building from scratch helps students with different learning styles visualize mechanical concepts, providing a tactile alternative to abstract software-only simulations.
- **Universal Compatibility:** As the construction is hardware-agnostic, the platform can be adapted to work with various microcontrollers (such as Arduino, micro:bit, or others), allowing educators to integrate the tech they already possess.

**Programming Environment:** Cardboard Robots is hardware-agnostic and does not force a single software. Instead, it relies on the electronics the educator chooses to attach. This allows for total flexibility: students can use block-based environments (like Scratch or MakeCode) if using micro:bit/Arduino, or transition to text-based coding (Python/C++), making the platform compatible with virtually any robotics programming interface.

**Age Group:** Highly flexible; suitable for primary school students (building simple mechanical structures) through to secondary and vocational school students (designing complex, autonomous robotic systems).

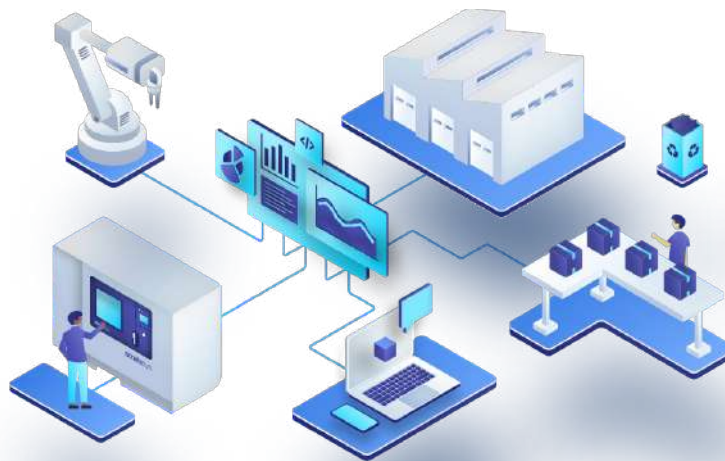
**Cost:** Extremely low. The primary costs are basic electronic components (motors, sensors, controllers), while the structural materials are essentially free, derived from recycled sources.

**Learning Objectives:** Primary objectives include developing mechanical engineering skills, understanding the relationship between physical structures and software control, promoting sustainable engineering practices, fostering creative design, and mastering basic electronics.

**Classroom Usability:** High, particularly for environments that embrace "Maker-space" pedagogy. It requires more preparation time than pre-made kits (due to the physical construction phase), but it offers a much higher pedagogical payoff by giving students total agency over their designs, making it an excellent tool for interdisciplinary projects combining art, technology, and science.

### 3.3.2 GrabCAD

<https://grabcad.com/>



## Description

GrabCAD, a Stratasys solution, serves as a significant online community and software platform for engineers, designers, and students. With a library containing over 6 million free CAD files and tutorials, it acts as a comprehensive repository for mechanical design and 3D modeling. It provides essential tools for additive manufacturing, including GrabCAD Print for 3D printing preparation, workflow management tools, and SDKs for enterprise connectivity. Pedagogically, it is a powerful resource for students to study professional-grade 3D designs, reverse-engineer mechanical assemblies, and prepare digital models for fabrication, bridging the gap between theoretical design and real-world manufacturing. While not a simulation tool for robotics code, it is an invaluable toolbox component for students learning to design the physical structures and components that robots require.

## Accessibility

- **Cloud-based Access:** The library is accessible via web browsers, allowing students to browse and download resources from any location, supporting both in-person and distance learning.
- **Extensive Resource Library:** Provides a vast collection of free CAD files and tutorials, offering students real-world engineering examples to study and adapt.
- **Global Community:** Encourages an open sharing culture where students can view, analyze, and learn from professional-grade designs.
- **Professional Integration:** Exposes students to industry-standard workflows and additive manufacturing tools, preparing them for future engineering environments.

**Programming Environment:** GrabCAD is not a programming environment. It is a repository and workflow platform for CAD design and 3D printing. It does not provide coding blocks or simulation interfaces but integrates with engineering software (CAD) and 3D printing ecosystems to support the design phase of robotics.

**Age Group:** Primarily targeted at secondary education and university-level engineering and design students.

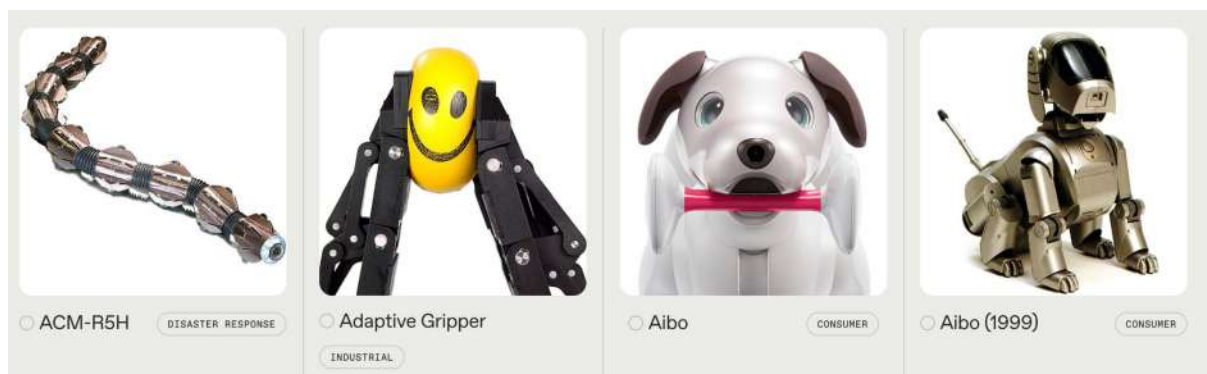
**Cost:** The library and platform tools are free for community use, ensuring broad access for educational institutions.

**Learning Objectives:** Primary objectives include developing mechanical design and CAD proficiency, understanding additive manufacturing workflows, studying real-world engineering assemblies, and fostering design iteration through the analysis of existing models.

**Classroom Usability:** High for engineering and design-focused curricula. It serves as an excellent resource for students to find reference models for their 3D printing or mechanical design projects. While it lacks the integrated "virtual classroom" or code-simulation features found in other tools, its role as a design hub makes it a necessary companion for Maker-space and robotics-building activities.

### 3.3.3 Explore robots

<https://robotsguide.com/robots>



Explore Robots is a comprehensive, open-access online encyclopedia and directory dedicated to the field of robotics. It serves as a centralized knowledge hub for students, educators, and robotics enthusiasts, providing a meticulously organized database of various robotic systems—ranging from commercial consumer robots to advanced industrial and research-grade machines. Each entry typically includes detailed technical specifications, manufacturers, intended applications, and media, offering a clear snapshot of the current robotics landscape. Pedagogically, it acts as an essential research tool. Unlike simulation environments that focus on coding and behavior, Robot Guide facilitates theoretical inquiry and technical literacy. It allows students to conduct comparative analyses, explore the evolution of robotic design, and identify appropriate hardware for specific engineering challenges, making it an invaluable companion for research-driven robotics projects.

#### Accessibility

- **Browser-based Reference:** The platform is accessible via any web browser, ensuring that students can conduct research from any device, whether at home or in the classroom.
- **Intuitive Categorization:** The directory is organized by robot type, application, and manufacturer, making it easy for students with varying levels of experience to find relevant information.

- **Educational Transparency:** By providing clear, standardized technical data, it demystifies complex hardware and helps students build a foundational understanding of robotic components and capabilities.
- **Global Resource:** As a free, public-facing database, it democratizes access to technical specifications that might otherwise be locked within manufacturer-specific documentation or expensive academic journals.

**Programming Environment:** Robot Guide is not a programming environment. It is a research and reference repository. It does not provide coding blocks, simulation interfaces, or code deployment tools; instead, it provides the technical background and hardware information necessary to inform programming and design decisions.

**Age Group:** Primarily targeted at secondary education and university-level students, though it is accessible to any learner interested in researching robotic systems.

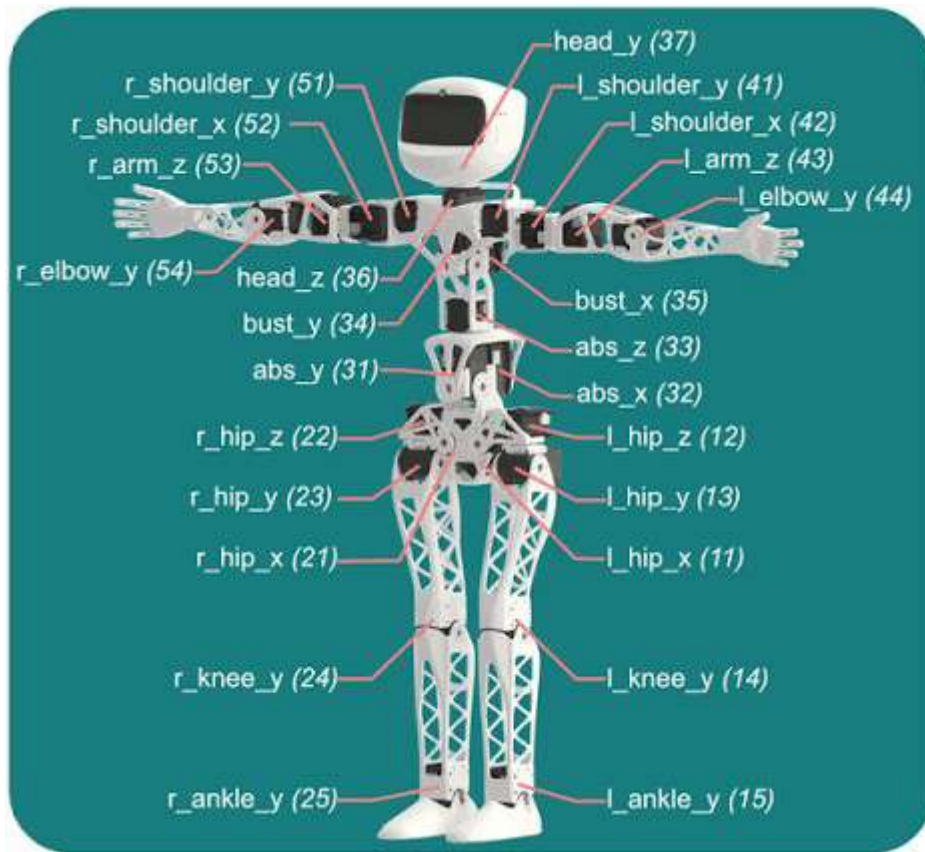
**Cost:** The platform is free to access, ensuring no financial barrier for educational institutions or individual researchers.

**Learning Objectives:** Primary objectives include developing research and information literacy skills, understanding the breadth and diversity of robotic applications, learning to interpret technical specifications, and supporting informed decision-making in hardware selection for engineering projects.

**Classroom Usability:** High for research-oriented curricula. It serves as a vital resource for students engaged in project planning, design research, or theoretical assignments. While it lacks interactive tools for coding or simulation, its role as a knowledge base makes it a necessary theoretical companion to the practical toolboxes discussed in earlier sections.

### 3.3.4 The Poppy project

<https://www.poppy-project.org/en/>



## Description

The Poppy Project is an open-source, interdisciplinary ecosystem that brings together a vibrant community of beginners, experts, scientists, educators, developers, and artists. United by the vision that robots are powerful tools for learning and creative expression, the project focuses on the creation of humanoid and quadruped robots using 3D printing and accessible electronics. Unlike pre-packaged educational kits that arrive as "black box" solutions, Poppy emphasizes a "Do-It-Yourself" and "Maker" approach, providing the hardware blueprints, software, and community support necessary to build and customize sophisticated robotic platforms. Pedagogically, it serves as a unique hub for exploration where users can advance from simple motor control to complex behavioral programming and biomechanical studies, acting as a crucial bridge between hobbyist electronics and professional-grade robotics research.

## Accessibility

- **Open-Source Platform:** All design files, blueprints, and code are shared freely, removing proprietary barriers and allowing schools to replicate or modify platforms according to their needs.
- **Modular Hardware:** The robot designs are modular and 3D-printable, allowing for easy repair, customization, and scalability, which fosters a deeper understanding of mechanical structure.
- **Collaborative Ecosystem:** The project is supported by a large, active community, providing extensive documentation, shared tutorials, and peer support that democratizes access to complex robotics.
- **Versatility:** The platform is adaptable, allowing it to function as a simple teaching tool for basic interaction or a sophisticated research platform for advanced automation.

**Programming Environment:** Poppy utilizes a flexible, multi-layered environment. It primarily uses the visual, block-based programming language *Snap!* (an extension of Scratch) for initial interactions, making it accessible to students with limited coding experience. For advanced users, researchers, and university-level applications, it leverages the *Pypot* library, a Python-based framework that enables the creation of complex behaviors and integrates seamlessly with the Robot Operating System (ROS).

**Age Group:** Primarily targeted at secondary education, vocational training, and university-level engineering, though its block-based interface allows for engagement at younger levels in structured laboratory settings.

**Cost:** Moderate. While the software and design files are free and open-source, the physical implementation requires an investment in 3D-printed parts and actuators (servomotors), making it more resource-intensive than browser-based simulators but significantly more affordable than proprietary humanoid robots.

**Learning Objectives:** Primary objectives include deepening the understanding of biomechanics and human-robot interaction, mastering the integration between mechanical hardware and software control, developing advanced programming skills (Python/ROS), and fostering an interdisciplinary approach that combines art, engineering, and computer science.

**Classroom Usability:** High for Maker-space and laboratory-based robotics curricula. It requires a higher degree of technical skill and preparation time compared to browser-based toolboxes due to the physical assembly and maintenance components, but it offers unparalleled pedagogical depth for students exploring robotics at an advanced, project-oriented level.

### 3.3.4 WRO Learn

[https://wro-learn.org/en\\_us/welcome](https://wro-learn.org/en_us/welcome)

#### Description

WRO Education, primarily known through its platform WRO Learn, is an online learning ecosystem developed by the World Robot Olympiad (WRO) Association to make high-quality robotics education accessible to students, coaches, and teachers globally. Serving as a centralized knowledge hub, the platform provides structured educational content that supports learners throughout their robotics journey—from absolute beginners taking their first steps in programming to advanced, competitive teams preparing for complex international robotics challenges. By offering essential resources that build technical knowledge, collaborative skills, and engineering confidence, WRO Learn acts as a critical bridge between foundational robotics education and real-world employability. The platform is designed to democratize access to expertise, providing a curated pathway of resources that helps users navigate the technical and strategic requirements of competitive robotics, regardless of their starting point, with a core mission of empowering young people through inclusive STEM opportunities.

## Accessibility

- **Free Online Access:** As a free, web-based platform, it eliminates financial barriers to accessing structured robotics curricula and training materials.
- **Global Reach:** It is designed to be accessible to a worldwide audience, supporting educators and robotics teams across diverse geographical and socioeconomic backgrounds.
- **Structured Progression:** By catering to both novices and experienced competitors, the platform provides scaffolded resources that accommodate different levels of prior knowledge and experience.
- **Educational Empowerment:** It provides necessary documentation and support that helps coaches and teachers, even those with limited prior experience, to successfully facilitate robotics instruction.

**Programming Environment:** WRO Education is primarily a learning and resource repository rather than a standalone programming compiler or simulator. It hosts educational guides, tutorials, and materials that support various industry-standard programming environments (such as EV3-G, Spike Prime, or Python, depending on the specific competition category), providing the technical background and methodology necessary to build robotic programs.

**Age Group:** Targeted at a wide spectrum of learners, including primary and secondary school students participating in robotics teams, as well as the coaches and educators who facilitate their learning.

**Cost:** Free. The platform is available at no cost to ensure broad, equitable access for the global robotics community.

**Learning Objectives:** Primary objectives include developing foundational and advanced robotics skills, strategic competition planning, mastering systematic problem-solving, fostering teamwork and collaborative communication, and building professional confidence within STEM fields.

**Classroom Usability:** High. As an online repository of curated content, it is highly useful for teachers and coaches to structure their robotics training programs. It allows educators to easily select resources that align with their students' skill levels, making it a valuable complement to the hands-on robotics kits and hardware used within the WRO ecosystem.

### 3.3.5 Lego education

<https://education.lego.com/en-gb/>



## Description

LEGO Education is a globally recognized pedagogical ecosystem that integrates the iconic LEGO brick with structured, standards-aligned curriculum to foster hands-on learning. Designed specifically for classroom environments, it utilizes a "learning through play" methodology that makes abstract STEM concepts—such as engineering, coding, and physical science—tangible and intuitive. The ecosystem encompasses a progressive suite of tools, ranging from early learning solutions for primary students to sophisticated robotic systems like SPIKE Prime and MINDSTORMS. Pedagogically, it is structured to support a continuous learning pathway, allowing students to advance from simple physical building and basic logic to complex automation and programming. By providing teacher-led resources, curriculum integration, and physical hardware, LEGO Education acts as a comprehensive bridge between creative exploration and formal academic achievement, emphasizing collaborative problem-solving and the engineering design process.

## Accessibility

- **Tactile Learning:** The physical nature of the building bricks provides a concrete, multi-sensory entry point into robotics, making it particularly accessible for students who benefit from kinesthetic and hands-on experiences.
- **Scalable Difficulty:** The ecosystem is designed to grow with the student, offering simplified, visual-based environments for beginners and advanced, text-based programming for experienced learners.
- **Structured Support:** Offers extensive teacher guides, curriculum-aligned lesson plans, and classroom management resources that significantly lower the barrier for educators to implement high-quality robotics instruction.
- **Inclusive Design:** The modularity of the system allows for diverse building techniques and creative interpretations, accommodating different learning styles and abilities within the same classroom setting.

**Programming Environment:** LEGO Education provides a highly versatile programming environment that spans the spectrum of skill levels. It primarily uses an intuitive, visual block-based language (derived from Scratch/Blockly) which allows for immediate interaction with

hardware. For advanced users, it offers full support for text-based programming languages, most notably Python, allowing a seamless transition from basic logic to professional syntax-based coding.

**Age Group:** Extremely flexible; suitable for early childhood education (pre-school) through to primary, middle, and secondary school levels.

**Cost:** High. Unlike browser-based simulators, LEGO Education requires a significant initial investment in physical hardware kits (hubs, sensors, motors, and bricks). While the software is generally accessible, the classroom implementation depends on the acquisition of the physical robot sets.

**Learning Objectives:** Primary objectives include developing mechanical engineering and structural design skills, mastering physical computing and control logic, promoting computational thinking, and fostering core 21st-century skills such as collaboration, critical thinking, and iterative problem-solving.

**Classroom Usability:** High. It is specifically built for classroom use, with hardware and software designed to integrate into standard lesson times. Its structured curriculum and reliability make it a premium but highly effective tool for formal education settings that prioritize hands-on experimentation and robust, interdisciplinary learning.