PREPARING FUTURE EDUCATORS: IMPLEMENTING THE CALSCRATCHJR-GR CURRICULUM IN PRESERVICE TEACHER TRAINING

Stamatios Papadakis | University of Crete, Greece


ARCTIC CIRCLE
66°30'
Asia
EUROPEAN RUSSIA
Norwegian Sea
Baltic
Atlantic Ocean
North Sea
EUROPE
English Channel
Bay of Biscay
Greece
Black Sea
Strait of Gibraltor
Tunisia
Mediterranean Sea
Morocco
Algeria
Libya
800 mi
800 km
23°5'N
PRIME MERIDIAN
0°
Africa

CONTEXT, PROBLEM, AND MOTIVATION

WHY THIS MATTERS FOR PRESERVICE TEACHERS

Cognitive and Affective Load

Designing computational thinking tasks for young children is cognitively demanding for preservice teachers still building pedagogical theory.

The Appeal of Automated Generation

AI lesson generation lowers immediate effort but transfers the burden onto evaluation, where novice teachers are least prepared.

Coding Anxiety in Novices

Preservice teachers report heightened anxiety with coding tasks lacking automated feedback, which biases them toward uncritical AI use.

The Pedagogical Stakes

The point is not to make AI the designer, but to train future teachers as critical curators of AI-generated materials.

THE RAPID ENTRY OF GENERATIVE AI INTO TEACHER EDUCATION

Faster Than Programmes Expected

Teacher education programmes began considering GenAI only after preservice teachers were already using LLMs to draft lesson plans.

Use in ScratchJr-Based Settings

LLMs are used to generate ScratchJr activities for children aged 5 to 7, often without consideration of developmental constraints.

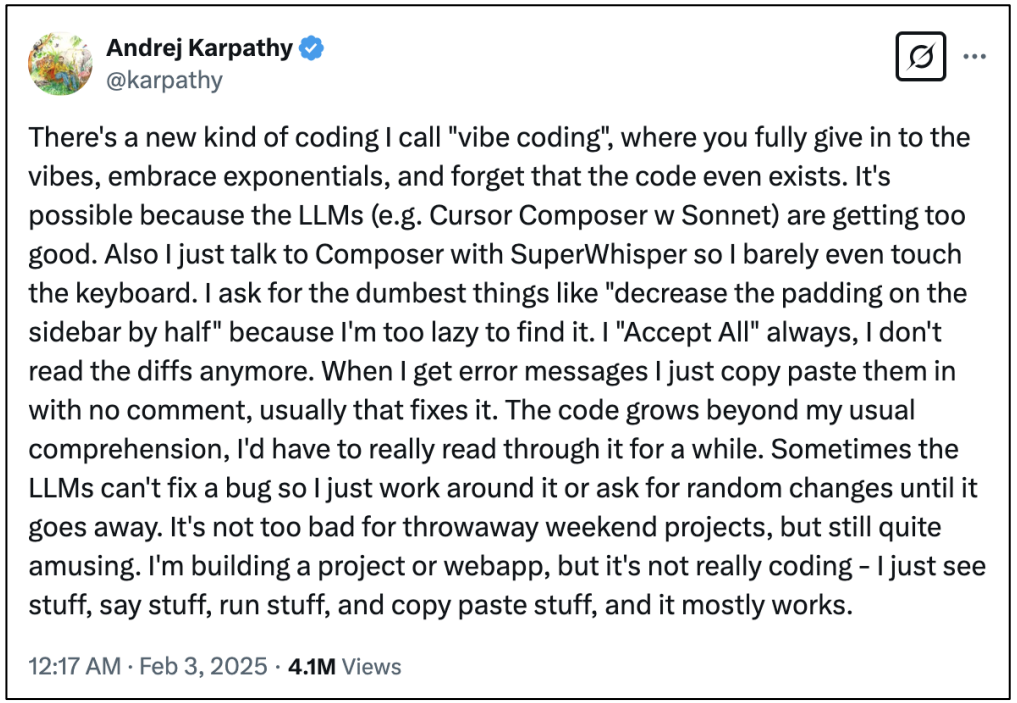
Procedurally Plausible, Not Pedagogically Sound

Outputs include sprite, narrative, and blocks, yet often fail on developmental fit, ScratchJr syntax, or socio-emotional alignment.

What Teacher Education Must Add

Preservice training needs explicit, domain-specific methods to evaluate, revise, and take responsibility for AI-generated materials.

VIBE CODING: THE SPECIFIC RISK FOR EARLY CHILDHOOD



Andrej Karpathy
@karpathy

There's a new kind of coding I call "vibe coding", where you fully give in to the vibes, embrace exponentials, and forget that the code even exists. It's possible because the LLMs (e.g. Cursor Composer w Sonnet) are getting too good. Also I just talk to Composer with SuperWhisper so I barely even touch the keyboard. I ask for the dumbest things like "decrease the padding on the sidebar by half" because I'm too lazy to find it. I "Accept All" always, I don't read the diffs anymore. When I get error messages I just copy paste them in with no comment, usually that fixes it. The code grows beyond my usual comprehension, I'd have to really read through it for a while. Sometimes the LLMs can't fix a bug so I just work around it or ask for random changes until it goes away. It's not too bad for throwaway weekend projects, but still quite amusing. I'm building a project or webapp, but it's not really coding - I just see stuff, say stuff, run stuff, and copy paste stuff, and it mostly works.

12:17 AM · Feb 3, 2025 · **4.1M** Views

Definition (after Karpathy)

Generation of outputs through conversational prompting without critical evaluation of their fitness for purpose.

Three Concrete Failure Modes

Outputs exceed ScratchJr block syntax, misrepresent socio-emotional aims, or bypass deliberate developmental sequencing.

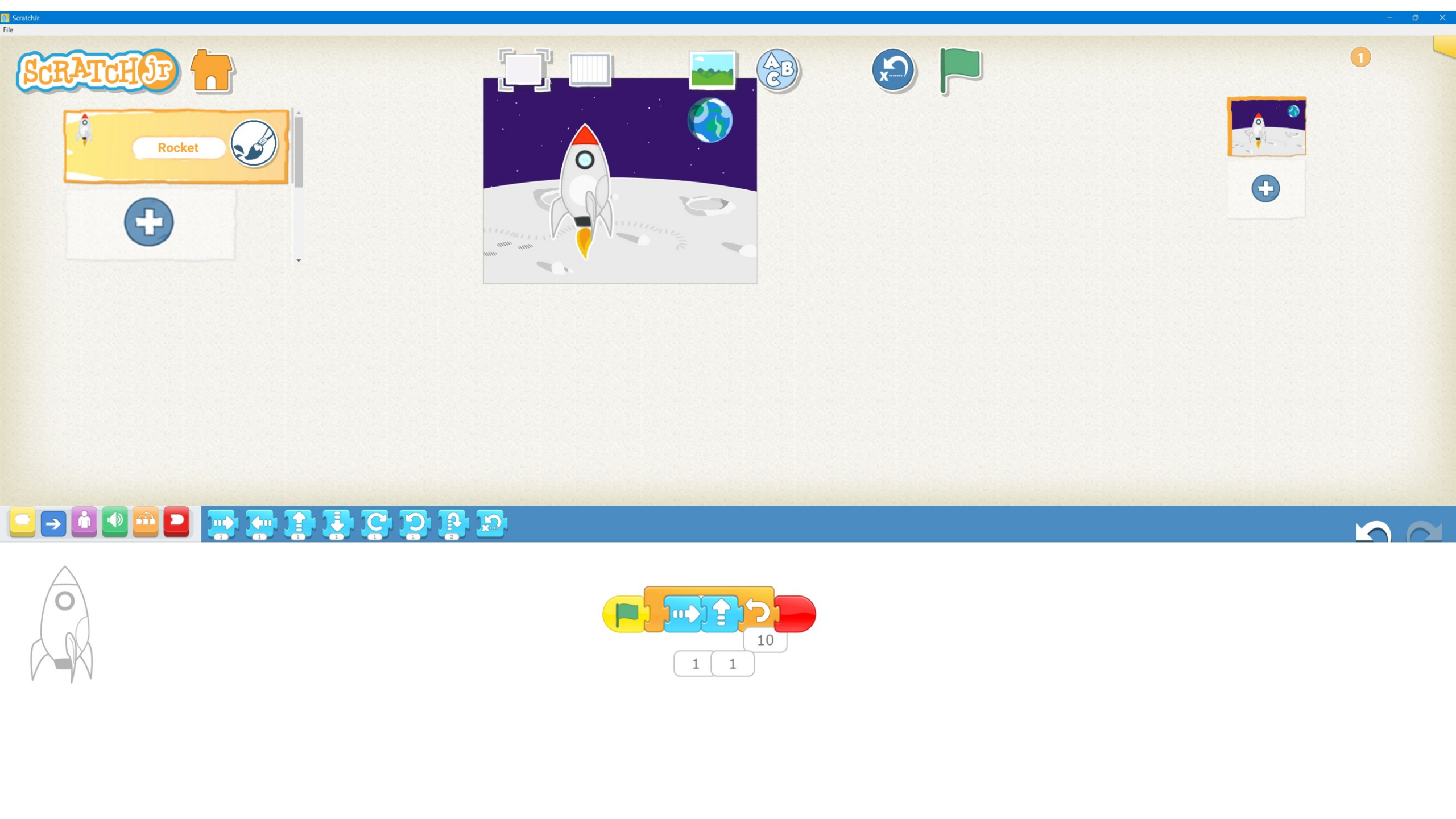
Why Novices Are Vulnerable

Preservice teachers facing cognitive overload and coding anxiety will default to the path of least effort: unconstrained prompting.

What the Framework Does

The Critical Curation Protocol builds deliberate pauses into design, requiring explicit pedagogical justification at each step.

THEORETICAL FOUNDATIONS


ScratchJr
File
SCRATCH JR
Rocket
1
10
1
1

Coding as Another Language - ScratchJr (CAL-ScratchJr)

Kindergarten

▸ Additional Languages

The Coding As Another Language for ScratchJr (CAL - ScratchJr) curriculum introduces powerful ideas from computer science, in conversation with literacy in a playful, structured, and developmentally appropriate way. The Coding as Another Language (CAL) approach, developed by Prof. Marina Umaschi Bers and members of her DevTech Research Group, understands the learning of computer science as a new literacy for the 21st century that supports young learners in developing new ways of thinking about themselves and the world.

The CAL curriculum for Kindergarten has 24 lessons, designed for a total of 18 hours, but can be adapted to any learning setting. Students will learn computer science , develop problem solving and computational thinking in the context of creating their own projects. In addition, to strengthen the literacy connection, the curriculum is integrated with two books, "A Computer Called Katherine" and "Knuffle Bunny."

Download the entire unit document, or edit by making a copy to adapt to your teaching context.

View a video of Professor Marina Umaschi Bers describing the CAL ScratchJr curriculum and its pedagogical approach.

View | Download | Visual Overview

Download Full Vocabulary List

Download Full Design Journal

- ▸ Pedagogy
- ▸ Powerful Ideas
- ▸ ScratchJr Skills
- ▸ Integrated Curriculum Design
- ▸ Pacing
- ▸ Materials
- ▸ Classroom Management
- ▸ Group Sizes
- ▸ Alignment of Academic Framework
- ▸ Assessments
- ▸ Copyright

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

View | Download | Visual Overview

Powerful Ideas of Computer Science
Design Process

Powerful Ideas of Literacy
Writing Process, Sequencing

Children will be able to:

- Identify steps of the Design Process.
- Apply the steps of the Design Process to an engineering challenge.

Vocabulary covered:

- Design Process
  - Ask
  - Imagine
  - Plan
  - Create
  - Test
  - Improve
  - Share
- Cycle

Necessary Material: Anchor Chart for Design Process

Warm Up

Design Process Song *(Suggested Time: 5 minutes)*

- Display the *Design Process Anchor Chart* on the screen
- Teach and sing the Design Process song

(to the tune of "Twinkle, Twinkle")

Ask and imagine, plan and create,

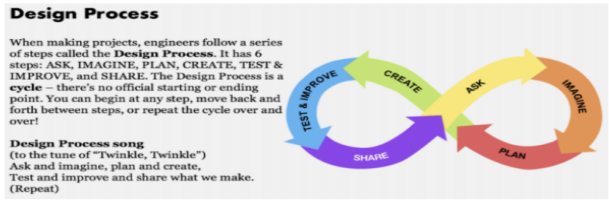
Test and improve and share what we make.

(Repeat)

Opening Tech Circle

Introduce the Design Process *(Suggested Time: 15 minutes)*

- Explain the **Design Process** using the Design Process Anchor Chart.



- Walk through the Design Process from the Teddy Bear project last class.
  - **Ask:** How can we program this dance on ScratchJr?
  - **Imagine:** Brainstormed movements.
  - **Plan:** Decided on the parameters we want to use and labeled our character.
  - **Create:** Make the character and the program.
  - **Test and Improve:** Were the parameters right? Do we need to make our character bigger?
  - **Share:** We shared at the end of class!

Unplugged Time

Your Own Design Process *(Suggested Time: 30 minutes)*

- Explain that the Design Process isn't just for ScratchJr, it's for making anything! Today they are going to get to choose what they design using the Design Process!
- **Ask:** Ask the children a question to serve as a prompt for their creation. For example, how strong can we make a tower out of building blocks, or what kind of butterfly house would be good for our playground? These creations can be done with recycled materials, craft supplies, building blocks, or other materials of your choosing, so select a prompt that aligns with the building materials you choose.
- **Imagine**: Break into small groups to brainstorm ideas for their projects.
- **Plan**: Ask the children at the end of the small group time to share their plans.
- **Create**: Give children time to use craft and recycled materials to make their creation!
- **Test and Improve**: Pause and ask children if their creations are following their plan? Are there any changes they want to make now that they've started actually making their project?
- **Share**: Go around and share creations as a class.

Closing Tech Circle

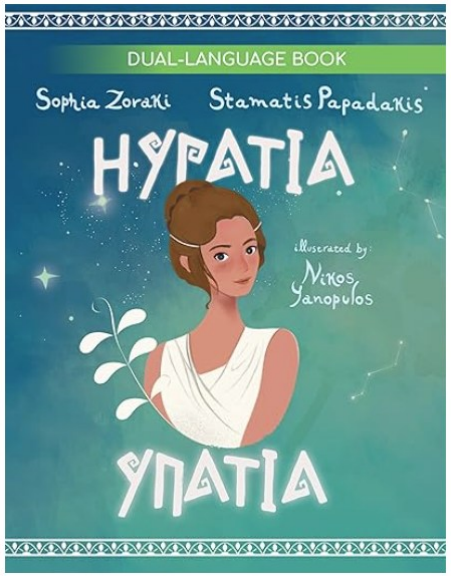
It's a Cycle! *(Suggested time: 5 minutes)*

- Ask the children if viewing their classmates' projects gave them any new ideas for their own projects! For example: "When I saw that [student name] added a rocket ship to their project, it inspired me to add an astronaut to my project!"
- Remind the children that the Design Process is a **cycle**. Sharing just leads to more asking and more creating and then more sharing!
- What could you include in your project from another person's project? Why?

Opportunities for Differentiation

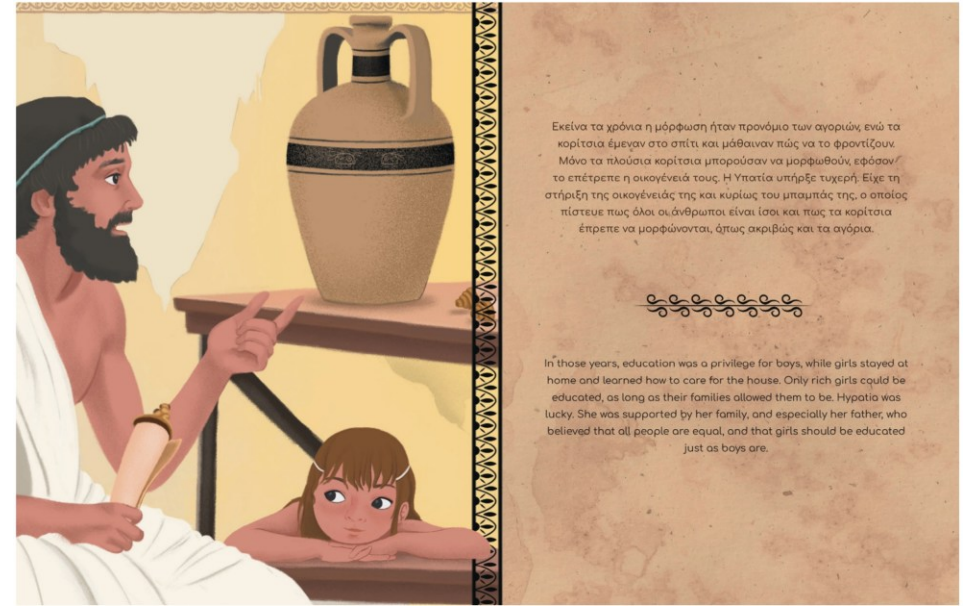
Virtual Learning

- Notify parents ahead of time that children will need craft supplies for this lesson.



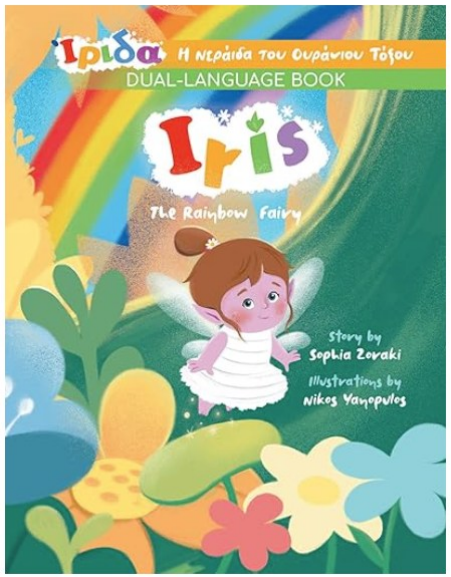
Ο Θέωνας ήταν Έλληνας μαθηματικός και μάθαινε τους ανθρώπους να σκέφτονται σωστά και να υπολογίζουν γρήγορα χρησιμοποιώντας τη λογική και την εξυπνάδα τους. Ήταν επίσης αστρονόμος αλλά και διευθυντής της μεγάλης βιβλιοθήκης της Αλεξάνδρειας!
Του άρεσε πολύ να παίρνει μαζί του στη βιβλιοθήκη τα παιδιά του και να τα φέρνει σε επαφή με ανθρώπους που αγαπούσαν το διάβασμα.
Η Υπατία, κάθε φορά που βρισκόταν εκεί, δεν σταματούσε να διαβάζει! Άνοιγε τους παπύρους, έβρισκε νέες πληροφορίες και συνομιλούσε με πολύ μορφωμένους ανθρώπους.

Theon was a Greek mathematician, and he taught people how to think properly and how to calculate quickly using logic and intelligence. He was also an astronomer and director of the Great Library of Alexandria!
He really liked to take his children to the library and bring them into contact with people who loved reading. Whenever she was there, Hypatia did not stop reading! She would open the scrolls, find out new things and talk to well-educated people.



Εκείνα τα χρόνια η μόρφωση ήταν προνόμιο των αγοριών, ενώ τα κορίτσια έμεναν στο σπίτι και μάθαιναν πώς να το φροντίζουν. Μόνο τα πλούσια κορίτσια μπορούσαν να μορφωθούν, εφόσον το επέτρεπε η οικογένειά τους. Η Υπατία υπήρξε τυχερή. Είχε τη στήριξη της οικογένειάς της και κυρίως του μπαμπά της, ο οποίος πίστευε πως όλοι οι άνθρωποι είναι ίσοι και πως τα κορίτσια έπρεπε να μορφώνονται, όπως ακριβώς και τα αγόρια.

In those years, education was a privilege for boys, while girls stayed at home and learned how to care for the house. Only rich girls could be educated, as long as their families allowed them to be. Hypatia was lucky. She was supported by her family, and especially her father, who believed that all people are equal, and that girls should be educated just as boys are.



Iris is a little fairy who loves colors, but her dress is as white as Trickle's snow. It was made by Mrs. Spider, the forest dressmaker, who can make all kinds of dresses, but only in white. But Mrs. Spider once told Iris a secret. She told her that if she could find a way to paint the seven stripes of her dress, she would become the last Weather Fairy!

Oh, how much Iris wanted that! She tried crawling on the grass to paint her dress green, but she just got it dirty. She tried diving in the lake's water to paint it blue, but instead, she got soaking wet from top to bottom. Then Iris flew close to the sun, hoping her dress would capture just a little yellow from the sunshine, but she only dried it.

"Good Morning, Witch Tricolor. I am Iris, a little fairy from the forest, and I love colors. But look... My dress is all white. Will you help me make it colorful so that I can become a Weather Fairy?"

"Welcome, Iris, with your dress of white
Come, let's see the colors that delight.
Enter my hut, dear fairy guest
We'll put colors on your dress."

In the hut, everything was colorful. Witch Tricolor took three little bottles out of a cabinet. One with red, one with blue and one with yellow paint. "Hold still. I will use these colors to paint three stripes on your lovely white dress," she said to Iris.

CODING AS ANOTHER LANGUAGE (CAL)

Coding as Literacy, Not Skill

Bers reconceptualises programming as a symbolic system through which children express meaning, not merely a technical skill.

Aligned with Early Childhood Pedagogy

Coding is integrated with drawing, storytelling, and play; the activity, not the code, is the unit of pedagogical meaning.

Why ScratchJr Fits CAL

Icon-based blocks remove syntactic barriers, allowing pre-readers to construct interactive narratives without typed code.

What Counts as Success

Computational tasks are vehicles for developing the values and competencies the activity targets, never ends in themselves.

POSITIVE TECHNOLOGICAL DEVELOPMENT (PTD)

Six Positive Behaviours

Communication, collaboration, community building, content creation, creativity, and choices of conduct.

From Task Completion to Behaviour

PTD shifts the assessment focus from whether code runs to whether the activity authentically operationalises a positive behaviour.

PTD as Evaluative Lens for AI Output

Each AI-generated activity is evaluated against the six PTD behaviours before it is considered deployment-ready.

Alignment with Greek Mandates

The Greek preschool Teacher's Guide requires that technology integration support the child's holistic socio-emotional development.

SCRATCHJR: AFFORDANCES AND THE ASSESSMENT GAP

Designed for Pre-Readers

Block-based, icon-driven environment for ages 5 to 7; no text input, no variables, no nested loops, no conditional statements.

No Automated Assessment

Unlike older block environments (e.g. Dr. Scratch for Scratch 3), ScratchJr has no embedded code-analysis or CT-scoring tool.

Where LLMs Drift

LLMs often hallucinate features ScratchJr does not have: variables, score tracking, if/then logic, nested loops.

Implication for Preservice Training

Preservice teachers must audit each AI output against ScratchJr's actual block palette before classroom use.

THE THREE-PHASE CRITICAL CURATION PROTOCOL

RATIONALE AND DESIGN OF THE PROTOCOL

AI as Draft, Teacher as Decider

GenAI supports educational decision-making but does not substitute for it; outputs are drafts requiring human evaluation.

Designed-In Friction

Unconstrained prompting is too easy; the protocol builds in mandatory pauses where each design choice must be justified.

The Three Phases

Phase 1: AI-Assisted Storyboarding. Phase 2: Critical Curation. Phase 3: Formative Assessment.

Anchored in Policy and Pedagogy

Aligned with UNESCO AI Competency Framework for Teachers and the JRC CompuThink II call for preservice-stage intervention.

PHASE 1: AI-ASSISTED STORYBOARDING

Constrained Prompting

Preservice teachers use PTD activity planning sheets to define exact parameters the LLM must operate within.

Role-Prompting the LLM

The LLM is instructed to act as an early childhood instructional designer constrained by ScratchJr's visual syntax.

Block-Palette Restriction

Prompts restrict the AI to ScratchJr's native block categories: triggering, motion, looks, and control.

The Zero-PII Mandate

No identifying information about children, classrooms, or schools appears in prompts or AI responses.

WORKED EXAMPLE: “COMMUNITY GARDEN”

Target PTD Value

Generosity, operationalised through turn-taking between two sprites in a two-scene ScratchJr project.

Constrained Block Palette

Prompt restricts AI to simple event triggers (Start on Tap, Start on Bump) and motion sequences only.

Computational Concept Targeted

Events and simple parallelism (Brennan & Resnick); the “Send Message” block carries the social meaning of turn-taking.

Why This Is Authentic Operationalisation

The virtue is not in the narrative label; it is in the code logic. Each sprite’s movement is triggered by the other’s completion.

PHASE 2: CRITICAL CURATION

AI Output as First Draft

Generated materials are treated as preliminary drafts subject to structured evaluation before classroom use.

The Two-Rejection Rule

Each AI output must be rejected and revised at least twice, with documented rationale for each rejection.

What Counts as a Valid Rejection

Only rejections citing (a) platform syntax violation, (b) PTD values misalignment, or (c) developmental coherence failure count.

Additional Audits

Age-Appropriate Complexity Audit; CAL/PTD Values-Alignment Audit; Societal Bias and Ethical Computing Audit.

PHASE 3: FORMATIVE ASSESSMENT

The Assessment Workload

Manual observational assessment increases workload and is a documented source of teacher anxiety in early STEM.

Hybrid CT + PTD Rubric

Intersects Brennan & Resnick's CT dimensions (Concepts, Practices, Perspectives) with PTD behavioural indicators.

AI Drafts, Teacher Finalises

The LLM drafts scenario-specific anchor text; the preservice teacher rewrites for direct classroom observability.

Three-Level Scaling

Behavioural indicators scaled Absent / Emerging / Present, observed during or immediately after the child's ScratchJr session.

IMPLEMENTATION AND INITIAL FINDINGS

THE GREEK CALSCRATCHJR-GR CONTEXT

The National Curriculum Mandate

Greek preschool curriculum requires digital tools to be embedded cross-thematically within play centres, not as isolated tech instruction.

CT Mandated, Capacity Lagging

JRC CompuThink II: Greece has compulsory CT in primary and lower secondary, but reports lack of adequately trained teachers.

The OECD Diagnosis

OECD review of Greek education: digital practice remains at the digitisation stage; only 63% of students see teachers as willing to use digital resources.

Why Preservice, Not In-Service

OECD and JRC both identify initial teacher education, not post-qualification workshops, as the primary structural lever for change.

COURSE-EMBEDDED MICRO-PILOT

Setup

20 preservice preschool educators, paired work, one shared design brief: a community-garden ScratchJr activity for ages 5 to 7.

What Was Collected

Initial and revised lesson drafts, rejection logs with documented rationale, instructor rubric scores, brief exit slip.

What We Found: Feasibility

Students did not treat first outputs as final; they revised against platform, age, value alignment, and documented their reasoning.

IMPLICATIONS, LIMITATIONS, AND CONCLUSIONS

IMPLICATIONS FOR PRESERVICE TEACHER EDUCATION

Beyond AI Tool Use

The skill to develop is not how to use AI tools, but how to evaluate, reject, and revise their outputs against pedagogical criteria.

Procedural Scaffolds Over Affect

Evidence from CAL-ScratchJr PD: self-efficacy does not predict coding competency; procedural scaffolds carry more weight.

Domain-Specific, Not Generic

Existing AI literacy frameworks target older learners or text-based environments; this protocol is specific to ScratchJr and PTD.

Institutional Implication

Academic integrity policies must distinguish AI-assisted plagiarism from documented critical co-design.

LIMITATIONS AND FUTURE WORK

What the Pilot Does Not Show

The pilot establishes feasibility only; it does not yet show that the protocol reduces uncritical AI reliance over time.

The Language Issue

LLM outputs and support materials work more reliably in English than in Greek; mitigated here by bilingual prompt support.

Required Next Steps

Longitudinal cohorts, peer-observation cycles, practicum integration; the four Kong-Lai-Sun conditions for durable PD gains.

Risk of Procedural Overhead

For anxious novices, structured rejection requirements may act as a stressor; future designs should stratify by prior competence.

TAKE-HOME MESSAGE

"I wish it need not have happened in my time," said Frodo.
"So do I," said Gandalf, "and so do all who live to see such times. But that is not for them to decide. All we have to decide is what to do with the time that is given us."

Book I, Chapter 2, "The Shadow of the Past", in The Fellowship of the Ring.
J. R. R. Tolkien (John Ronald Reuel Tolkien)

AI Has a Place

GenAI can reduce preparation time and seed ideas, but only inside a pedagogical framework that defines what counts as good.

The Cost of Vibe Coding

Without structured curation, AI produces ScratchJr activities that resemble coding lessons but fail on the dimensions that matter.

What the Protocol Embeds

An argument: AI supports design; the teacher retains agency over pedagogical logic. Drafts go in; judgement decides what comes out.

The Shift

From AI-generated lesson plans to teacher-curated, child-centred coding activities. Thank you.