Author: Enrica Caruana

Title: Crack the Code – Goldilocks Sequence Debugging

Purpose of the Challenge

- To introduce the basics of coding by helping pupils understand that programs are made up of clear, ordered steps (sequences).
- To develop early debugging skills by identifying mistakes in a sequence and correcting them, mirroring how programmers fix errors in code.
- To build logical thinking by encouraging pupils to analyse events, predict outcomes, and justify their choices.
- To strengthen problem-solving skills through trial, discussion, and refinement of ideas in a collaborative setting.
- To make abstract computing concepts concrete by using a familiar story context that pupils already understand
- To promote collaboration and communication as pupils work in groups, explain reasoning, and listen to others' ideas.
- To support inclusion and accessibility in computing by using visual supports, hands-on materials, and differentiated roles.
- To build confidence and positive attitudes towards coding, showing pupils that everyone can be a coder.

Description of the activity

Students debug a mixed-up sequence of picture cards from *Goldilocks and the Three Bears*, reordering events to learn about sequencing, problem-solving, and the basics of coding in a fun, inclusive way.

Target audience

Primary school learners. The story content and picture sequencing may change to accommodate varying ages and abilities.

Experience

Beginner - Accessible to learners with no previous experience in coding or sequencing.

Duration

40 minutes

Recommended tools:

Picture cards illustrating key events from *Goldilocks and the Three Bears*

Printed story sequence cards (laminated for reuse, if possible)

Visual supports if needed (symbols, colour-coding, or numbered cues)

Preferably having at least 1 child who knows the story and can retell it to his/her peers, if not a storybook or digital version of *Goldilocks and the Three Bears*

Table space or floor mats for group work

Differentiated sets of cards (fewer/more steps depending on ability)

Funded by
the European Union

Timer or sand timer (optional, for added challenge)

Instructions

Step 1: Introduce the story (inclusive approach)

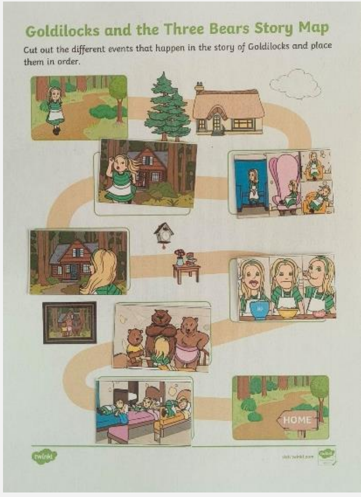
To ensure the activity is fully inclusive, it is suggested not to use any written text, supporting young pupils with literacy difficulties. Instead, a pupil who knows *Goldilocks and the Three Bears* may retell the story to her peers, promoting co-learning, confidence, and shared understanding. If this is not possible, the story may be read aloud or explained by the teacher.

Step 2: Group setup and discovery of errors

Pupils are divided into small groups and seated around story maps with the instructions (picture cards) deliberately jumbled. As they explore the cards, pupils quickly notice on their own that some events are out of order, naturally discovering the concept of errors in sequencing.

Step 3: Introduce coding concepts and present the challenge

Explain that coding is a set of instructions and that debugging is the process of putting them straight. Like programmers, pupils need to find the "bugs" in the story and fix them. Groups are tasked with arranging the picture cards into the correct sequence. The teacher encourages discussion and prediction of the next event. Differentiation is provided with fewer or more cards depending on group ability, and pupils are reminded that making mistakes is normal and an important part of debugging.


Goldilocks and the Three Bears Story Map
Cut out the different events that happen in the story of Goldilocks and place them in order.
HOME

Step 4: Group organisation and role assignment

Within each group, pupils are encouraged to take on roles to support engagement and ensure meaningful participation. For example, one pupil can be the **sorter**, placing cards in the order they think is correct; another can be the **reader**, describing what they see in each card; a **checker** reviews the order and suggests corrections. This fosters cooperation, peer learning, and communication skills.

Step 5: Decode the images and discuss sequencing

Pupils examine each picture card carefully, discussing what is happening in each scene. They predict what might come next and compare ideas within the group to understand the concept of a sequence. This step also supports literacy and language skills, as pupils describe events verbally and are encouraged by peers and the teacher.

Step 6: Sequence the cards collaboratively

Groups place the cards in what they believe is the correct order, negotiating and reaching consensus. The teacher circulates to prompt deeper thinking with questions like, "Why do you think this comes next?" or "Can we check for any mistakes?" Scaffolding, fewer cards, or visual cues are provided as needed to ensure all pupils can participate fully.

Funded by the European Union

Step 7: Debug, refine, and iterate

Once an initial sequence is formed, pupils review it to identify any mistakes. Groups test different orders, make corrections, and discuss why changes are needed. This iterative process mirrors real-life programming, reinforcing resilience, attention to detail, problem-solving, and computational thinking.

Step 8: Share, reflect, and consolidate learning

Each group presents their final sequence to the class, explaining their reasoning and reflecting on any mistakes they corrected. The teacher facilitates a discussion linking sequences, debugging, and problem-solving to everyday tasks and storytelling. Pupils celebrate successes, recognize the value of collaboration, and articulate what they have learned about coding, sequencing, and logical thinking.

Step 9: Make the learning explicit

The session concludes by linking the activity back to key computing concepts such as sequencing, debugging, logical thinking, and collaboration. Pupils are reminded that coding is accessible to everyone and that skills like problem-solving, discussion, and persistence are valuable both in computing and in life.

Funded by
the European Union