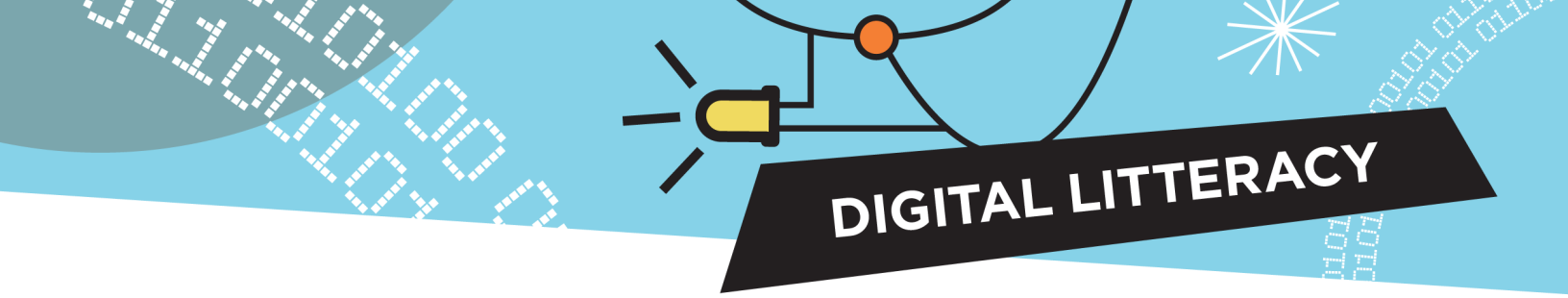


Computational Thinking
Exploring Algorithm Sequencing

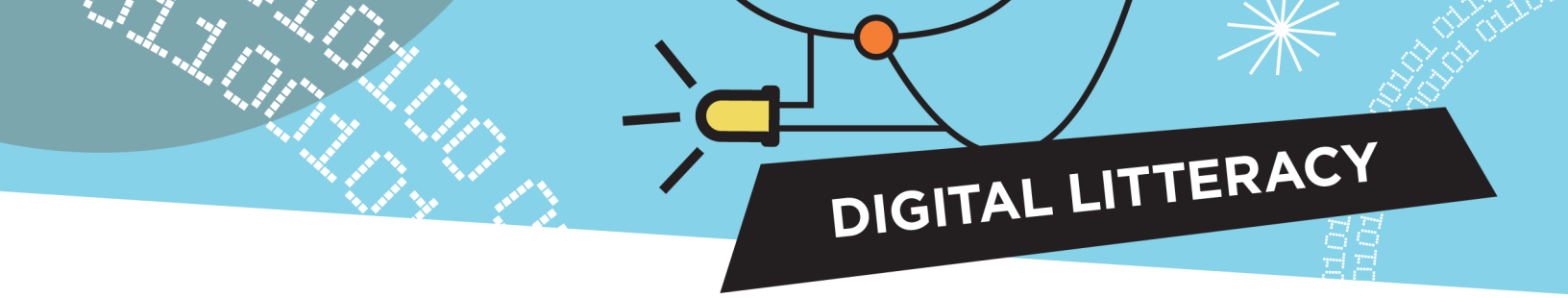
These unplugged activity ideas are meant to show examples of each key concept of computational thinking, in order to demonstrate its importance. You may find overlaps in the concepts presented in the activities; this is to be expected, since they are a part of a whole process that happens simultaneously.

Everyday life scenarios

1. Instead of working toward a solution, start with the ending and ask the participants to reverse engineer all the steps that took place for this specific thing to happen. You can use the events in the Exploring Decomposition document as example, or use a more tactile approach by asking participants the steps to constructing a LEGO™ structure from a picture.
    - A variation would be to ask one participant to describe a LEGO™ structure in a decomposition manner, and have a second participant use an algorithm approach.
    - They could also explain to another person who is standing back-to-back how to build it, using their own pieces.
2. Include the beginning and the end of a text, and ask participants to fill in the blanks. Examples include presenting the first and last line of a fairy tale, or describing how a video game level begins, then jumping to how the level's "boss" (the computer-controlled enemy) was defeated.

Discussion

- How important was it to choose the right words in your algorithm sequencing?
- Were all the sequences the same, or were there some differences in their order? If so, did it work anyway?
- Is sequencing easier, harder, or the same when you know the anticipated end goal?



45-60 *minutes* | *Grade 5 and up* | *Solo or Team* | *Advanced*

Story Maze: Algorithm Sequencing in Computer Programming

Sequencing is extremely important in computer programming, as it provides the order in which code has to be executed. Without the sequence, codes could come into conflict, be applied too early, too late, or not at all. This would result in an output that does not reflect what the programmer wanted to achieve. A computer doesn't know the difference between right and wrong sequencing; it will perform the actions it was programmed to, regardless of whether they make sense or not. It's the programmer job to decide and implement the appropriate algorithmic sequence.

Lesson goal

The goal is to develop an understanding of how to translate a story into a specific sequence of actions (algorithms) that a computer would understand.

Preparation

- Print or share the document *Exploring Algorithm: Story Maze Activity.*
- As a class, come up with a list of action words.
    - What would these words look like if you asked the students to act them out?
- Have students think about a video or computer game they play.
    - What are the components of the game?
    - Does it have a story?
    - Characters?
    - Challenges?

Tasks

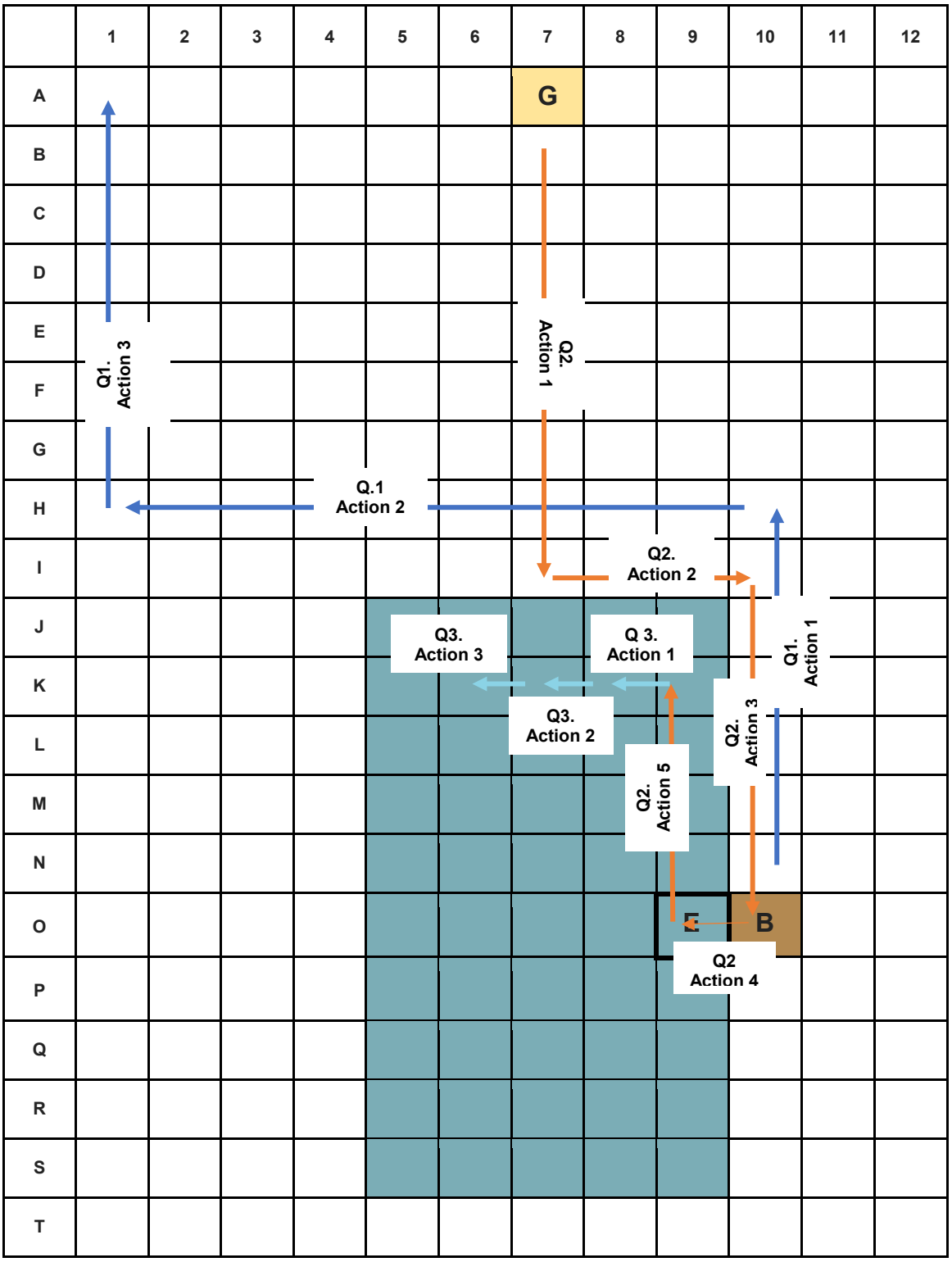
- Take a story, either made up or one you already know, and write it out as if you were making a video game out of it.
- Use a grid (in the example sheet found in the appendix) to plot out your characters' actions and how they will move throughout the game.
- You can also create a grid on the floor using tape, and have students take on the role of the characters and the "programmer" who will tell the characters what to do.

Go further

Have your students code their story in Scratch JR.

Solutions to Story Maze activity

Grid Game: Goldilock and the Three Bears



B = Bear starting point

G = Goldilocks starting point

= House delimitation

E = House entrance