PRIMO

Coding with Cubetto - Unit 4

Year 1, Ages 5 to 6, UK National Curriculum

Subjects covered:

Maths [time, capacity, money, addition & position]

Materials required:

6x Cubettos

6x Boards

6x Sets of Blocks

6x Standard Maps

Resources provided:

Map template

Capacity worksheet

Cubetto time cards

Building square

Cube net template

Diary outline

Board template

Dinner plate template

Water & Port squares

Compass template

Maze template

Introduction

The Cubetto Playset is a Montessori inspired coding toy that allows children ages 3 to 6 to program a friendly wooden robot without screens and is powered by a programming language you can touch.

New technology can sometimes be overwhelming to understand and adopt. The activities contained in this guide were created by educators for educators.

We want to make it simple for you to integrate the Cubetto Playset and its tangible programming language into your teaching.

Development and learning in other key areas

Beyond coding

The collaborative nature of Cubetto makes it an extremely versatile tool for the classroom. Cubetto fosters learning in key development areas that go beyond programming.

Communication

Children practice listening through a range of stories and narratives in relation to Cubetto, accurately anticipating key events and responding with comments, questions or actions. They also develop their own narratives and explanations.

Dexterity

Children develop coordination in large and small movements around the playset. They negotiate the placement of obstacles around the world map and place blocks on our tangible interface.

Social-Emotional

Children become confident by trying new, open-ended activities that remove “wrong” outcomes, and easily encourage group work. The open nature of the maps allows them to choose the resources they need for their play session.

Mathematics

Children add and subtract blocks to a sequence. They solve problems, including doubling and halving to get Cubetto from A to B. They discuss size, shapes and patterns, distance, position, and time to solve problems.

Logical reasoning

The blocks allow children to create and debug simple programs with their hands. They use technology purposefully to create, organise, store, manipulate and retrieve meaningful sequences.

Introducing the Playset

Introducing Cubetto

Introduce Cubetto as a friendly robot that children can program. Children should be told that Cubetto cannot think for himself, and can only move as programmed by the child, just like any other machine. If in a group setting, sit children in a circle, and allow them to pass Cubetto around to one another, saying hello or acknowledging the presence of the object.

Doing so forms a bond with Cubetto, in the same way they would with a stuffed animal, or a toy, and solving problems through narratives later on is more engaging.

Introducing the Board

Introduce the Board as a remote control that children can use to send instructions to Cubetto.

Without the Board, there is no way of sending Cubetto his instructions.

It is important for children to understand Cubetto is only able to move with a human's command. This is not only empowering, but also key to understanding computing.

Encourage children to also explain what other objects in their homes and lives function within a similar paradigm. A television needs a human to change its channels for example, or a washing machine needs a human to select its settings.

These examples, like Cubetto, are machines that need human programming to do their job.

Introducing the Blocks

Introduce the Instruction Blocks as the directions Cubetto follows when inserted in the Board and sent by pressing the action button.

Different Blocks represent different instructions, and an unambiguous, distinct command. These Blocks are what make up Cubetto's hands on coding language, and are key in the learning of computational thinking.

When each block is inserted in the Board, a child should be encouraged to predict what Cubetto will execute before pressing the "Go" button.

This is key in understanding concepts like program design, and it helps develop abstraction.

Table of contents

Unit 4 Overview

Year 1

By the end of the unit pupils will be able to:

- Understand what a bug is, different types of bug, how to identify and how to debug and test an algorithm.

- Pupils will also be able to create and debug a simple algorithm and use logical reasoning to predict the behaviour of simple programs.

	Lesson 1	Lesson 2	Lesson 3	Lesson 4
NC Computing Objectives	To use logical reasoning to predict behaviour of simple programs	To create and debug a simple program	To create and debug a simple program	To create and debug a simple program
Outcomes	• I can debug a simple algorithm • I can accurately measure in ml	• I can identify bugs in a simple algorithm • I understand that algorithms can develop bugs	• I can predict behaviour and debug a simple algorithm • I can add coins to the value of 50p	• I can debug an algorithm • I can solve problems involving addition and money
Cross-curricular Subject	Maths: Number: Measure (Capacity)	Maths: Measure (Time)	Math: Number: Addition (Money)	Maths: Number: Addition (Money)
Computational Thinking	Algorithms, Collaborating	Algorithms, Debugging	Algorithms, Debugging	Logic, Creating
Main Activities	Cubetto's H2UhOh! 1. Cubetto needs to get water from the mountains to the castle but can't move! Work out why and practise debugging algorithms. 2. Read the water jugs, measure water accurately in ml and put in Cubetto's bucket.	Cubetto's traffic trouble 1. Find out where Cubetto wants to visit and read the clocks in time order. Program Cubetto to visit new buildings on the map, avoiding car traffic. 2. Decorate shops and buildings to add to the map for Cubetto to visit.	Cubetto's treasure hunt 1. Cubetto is on a treasure hunt and needs help working out why its algorithm isn't working. Predict whether Cubetto can collect all the treasure, then debug and test. 2. Add up the treasure collected on the hunt – what if the coins were worth more?	Cubetto's dinner dash 1. Cubetto can't be late for dinner! Predict the function of random block and debug algorithm. Debug and test algorithm to get Cubetto to the restaurant. 2. Add up how much Cubetto spent on dinner then draw a healthy plate.
Challenge	Can you plan Cubetto's journey delivering water to 2 different places?	Where can Cubetto get to with only six blocks?	How could Cubetto get more treasure in less time?	Could we replace the random block with the function to speed it up?
Creative Play	Colour the water and mix together into a "potion"	Make and decorate shops and buildings with 3D templates	Draw a treasure map to show Cubetto's journey	Role play ordering and paying for food in a restaurant.
Resources	Map template, Algorithm to debug, Capacity worksheet, Beakers & jugs	Cubetto time cards, Cube net template, Building & shop squares, Pens & plastic cars	Map template, Algorithm to debug, Plastic coins, Treasure worksheet	Menu & dinner plate, Maps, Restaurant pictures, Algorithms to debug, Number lines
Assessment	Storyboard & grid, Photos, Verbal statements, Observation	Photos, Verbal statements, Observation	Photos, Treasure map, Verbal statements, Observation	Photos, Verbal statements, Observation

	Lesson 5	Lesson 6	Lesson 7	Lesson 8
NC Computing Objectives	To debug a simple program	To use logical reasoning to predict behaviour of simple programs	To create and debug a simple program	To create and debug simple program
Outcomes	• I can use function block to simplify an algorithm • I can solve problems involving direction	• I can predict what a simple algorithm will do • I can create the boundaries of a maze	• I can debug patterns in an algorithm • I can solve problems	• I can use function block to simplify an algorithm • I can solve two-step problems
Cross-curricular Subject	Maths (Position - Compass points)	Maths: Position & Movement (Direction)	Maths: Using & Applying	Math: Problem Solving
Computational Thinking	Logic, Debugging	Logic, Collaborating	Algorithms, Tinkering	Algorithms, Persevering
Main Activities	Cubetto's party planning 1. Use new river squares to make a safe route for Cubetto's boat trip. Use the function block! 2. Make Cubetto into a boat by designing its sides. Make a compass to help direct Cubetto around the map.	Cubetto's counting 1. Create your own maze on the map with an entrance and exit. Write algorithm to get Cubetto to the end of the maze. 2. Solve the maze and write instructions to help someone find the exit. Add a monster as a trap!	Cubetto's presents 1. Cubetto is going to a party but can't dance! Practise your favourite dance moves, then work out how to make Cubetto repeat its dance moves. 2. Work with a partner to make a four step dance routine in the playground on a grid.	Cubetto's party 1. Meet Cubetta – Cubetto's friend! It's their birthday and Cubetto wants to deliver a parcel. Test, debug and fix the algorithm to successfully collect and deliver the parcel. 2. Write a letter to Cubetta about all they have learnt!
Challenge	Without the function block, how far could Cubetto's boat get?	Can two Cubettos use your maze at the same time?	Can you make two Cubettos dance together?	Could you move the parcel and keep the algorithm the same?
Creative Play	Make a boat picture using 2D shapes.	Design a scary monster for your maze	Create your own dance to music and write instructions	Draw Cubetto a picture of your favourite adventure
Resources	Map template, Water squares, Port square, Algorithm to debug	Maze template, Maze images, Blank 4x4 grid, Coloured card	Maze template, Maze images, Blank 4x4 grid, Coloured card	Algorithms to debug, Cubes (parcels), Letter writing resources
Assessment	Photos, Verbal statements, Observation	Map grid, Verbal statements, Observation	Map grid, Verbal statements, Observation	Photos, Verbal statements, Observation

Lesson 1: Cubetto's H20 Uh Oh! (1 of 2)

Cross-curricular Area: Maths: Number: Measure

NC Objectives

To use logical reasoning to predict behaviour of simple programs

Outcomes

- I can debug a simple algorithm
- I can accurately measure in ml

Resources Needed

- Algorithm to debug
- Capacity worksheet
- Beakers, jugs and small buckets.

Prep Needed

- Prepare algorithms with missing block(s), first one starting at mountain.
- Copy map templates and capacity worksheets.
- Over the unit, build a visual working display of 'bugs'.

Resources Provided

- Map template
- Capacity worksheet

Key Vocabulary

- Debug/bug
- Algorithm
- Computer programmer
- Exchange
- Milliliters
- Capacity

Computational thinking concept

Algorithms

Computational thinking approach

Collaborating

Teacher-led Introduction

1. Ask: Does anybody know what H2O is? The scientific name for water.
2. Explain that Cubetto has a problem and needs the children's help: Cubetto is trying to transport mountain water back to the castle but can't move! Tell the children that you're really confused and need their help too.
3. Place Cubetto on the mountain and ask for a volunteer to press the Go button on the prepared bugged algorithm.
4. Ask: Why isn't Cubetto moving? Encourage children to talk in pairs and then collect suggestions.
5. Introduce the word 'debug' and explain that this word was said to have been first used by Grace Hopper who was a computer programmer in 1940s who found a moth in the computer which stopped it working!
6. Explain that bugs are mistakes in algorithms. Bugs give Cubetto wrong instructions, and to debug means to fix the problem.
7. Ask: What is the bug in the algorithm? Model responding: "The bug in the algorithm is the missing block(s)".
8. Ask: How could we solve this problem to get Cubetto from the mountain to the castle?
9. Explain that today pupils will be spotting bugs in algorithms. Explain that people who work in computing (computer programmers) have to find bugs to make sure the algorithms run correctly. Computer programmers often exchange parts or finding missing parts.

Lesson 1: Cubetto the 2D Shape Detective (2 of 2)

Creative Play

Dye the water pots different colours and mix together into a "potion".

Guided Activity

1. Beforehand, explain that they must be extra careful with Cubetto today - they will be using water: computers and water don't mix well!
2. Put children in pairs and ask one child to measure 100ml water into a beaker and place it carefully on Cubetto.
3. Looking at the map together, ask: Where might Cubetto need to take the water? Children to discuss and feedback ideas and reasons.
4. Suggest that Cubetto needs to take fresh mountain water to its friend's boat. Ask children to find the boat on the map.
5. Look at the bugged algorithm together and guess where Cubetto will move. Ask: What is going to happen? How do you know?
6. Ask: Can you find the bug in the algorithm? Talk in partners, circle and feedback.
7. Tell children to use board template to fix the algorithm: add the missing blocks and create, test and debug algorithm on the Board.
8. Repeat for different quantities of water and locations on the map.

Independent Activity

1. Find a partner to work with today.
2. Look at the capacity sheet and read how much water is in the first jug in milliliters.
3. One person uses the jug to pour the water slowly and the other tells them when it's the right amount.
4. When you have the right amount, write it in the box and pour it into Cubetto's bucket. Repeat for the next amount, swapping roles.
5. How much water does Cubetto have in jug 2, 5 and 6? Add up the amounts.

Challenge

Can you plan Cubetto's journey delivering water to two different places?

Plenary and Assessment

1. Ask volunteers to share how they debugged the algorithms, encouraging use of sentence starters: "The bug in the algorithm was " and "I fixed the bug in the algorithm by ".
2. Ask: What do we measure water in? Show different units for children to choose from (including random ones). Ask: Can we measure other liquids in ml too? What could we measure in ml?
3. Ask: How much did Cubetto have in the bucket in the end? Model adding up the different amounts to find the total.

Lesson 2: Cubetto's Traffic Trouble! (1 of 2)

Cross-curricular Area: Maths - Measure (Time)

NC Objectives

To create and debug a simple program

Outcomes

- I can identify bugs in a simple algorithm
- I understand that algorithms can develop bugs over time

Resources Needed

- Coloured pens
- Plastic cars
- Glue

Prep Needed

- Check batteries
- Cut out building squares and label with different places (e.g zoo, cinema).
- Complete diary outline for class.
- Write times on clock cards to match diary

Resources Provided

- Cubetto time cards
- Building square
- Cube net template
- Diary outline

Key Vocabulary

- Traffic
- Obstacle
- Algorithm
- Bug/ Debug
- Exchange
- Direction
- Obstacle
- Route

Computational thinking concept

Algorithms

Computational thinking approach

Debugging

Teacher-led Introduction

1. Show video of Bangkok traffic jam: https://www.youtube.com/watch?v=CQgqOJdOkDY and ask: What is traffic? What causes traffic?
2. Explain that Cubetto has written a diary of all the places it wants to visit tomorrow and the children need to help Cubetto get there. The problem is that there is so much traffic getting in the way!
3. Introduce Cubetto's Diary and ask: Why might Cubetto be visiting this place? What might Cubetto do here?
4. Recap how to read clocks using an interactive clock: http://www.visnos.com/demos/clock
5. Ask: What does the long hand point to? Minutes. What does the short hand point to?
6. Place the different buildings on the map for children to see.
7. Ask: What happens if Cubetto is moving and comes across an obstacle (something in the way)? Children to talk in small groups and feedback: encourage pupils to use "change direction/ go back/ new route".

Lesson 2:Cubetto's Traffic Trouble! (2 of 2)

Creative Play

Make and decorate shops and buildings using 3D templates

Guided Activity

1. Find a partner to work with today.
2. Show time cards and pupils take it in turns to read the times on the clocks. Ask: Where did Cubetto go first?
3. Ask pupils to put the cards in order and place buildings on map following time order
4. Tell children to create algorithm to get Cubetto to his first destination in the diary (e.g. the zoo).
5. Ask partners to program the Board and test for correct instructions.
6. Add a car to the map that causes traffic for Cubetto and ask: Can Cubetto take the same route? What is the bug in your algorithm now?
7. Explain algorithms don't always start with bugs but that can change over time - they can develop bugs just like coming across traffic.
8. Ask children to write new algorithm for to get Cubetto to its destination and test.
9. Repeat with next destination and allow children to place cars on map to cause traffic for each othera

Independent Activity

1. Find a partner to work with and look at the template. What shape do you think this will make?
2. Cut out the net carefully – don't cut off the tabs!
3. Fold along the lines.
4. Put glue on the tabs and fold together to create the 3D shape.
5. You've made a building! Decorate your building to use on the map. If it's a zoo, draw on animals. If it's a cinema, draw on films you like.

Challenge

Where can Cubetto get to with only 6 blocks?

Plenary and Assessment

1. Show route with traffic (plastic cars) and prepared algorithm. Ask: Will Cubetto reach his destination? What do you need to change/fix it?
2. Ask: Do computers always have bugs? Do all computers have bugs? Emphasise that bugs can develop over time.
3. Ask: How could we use the function button to get Cubetto to a building further away?
4. Recap use of function block and move one building further away on the map. Allow children time to discuss and create algorithm.

Lesson 3: Cubetto's Treasure Hunt! (1 of 2)

Cross-curricular Area: Maths: Number: Addition (Money)

NC Objectives

To create and debug simple programs

Outcomes

- I can predict the behaviour and debug a simple algorithm
- I can add coins to the value of 50p

Resources Needed

- Algorithm to debug
- Treasure Chest worksheet
- Plastic coins

Prep Needed

- Algorithms for collecting treasure for children to debug, starting on compass.
- Create Treasure Chest worksheet of different coin values.

Resources Provided

- Primo Board template
- Map template

Key Vocabulary

- Bugs / debug
- Algorithms
- Incorrect
- Money
- Pence / Pounds p/£
- Predict

Computational thinking concept

Algorithms

Computational thinking approach

Debugging

Teacher-led Introduction

1. Recap what bugs in algorithms mean and how children debugged their algorithms.
2. Show a bugged algorithm (not missing blocks) and ask: What do you think of this algorithm?
3. Explain that you agree that the algorithm looks good and introduce the treasure hunt! Explain that this algorithm should get Cubetto to pick up a piece of treasure on the map.
4. Place Cubetto on compass and treasure on square (that algorithm won't reach). Press the Go button and show your confusion!
5. Ask: What went wrong? Children to talk in small groups & feedback: "I think the bug in the algorithm was "
6. Clarify that so far in the unit the children have found two types of bugs in algorithms: missing blocks and incorrect blocks.
7. Explain that today the pupils will be spotting more bugs in algorithms to help Cubetto go on a treasure hunt!
8. Recap: How many pence in £1? Show various coins and recap their values.
9. Introduce word 'predict' for children to use today: "I predict that this algorithm won't work because "

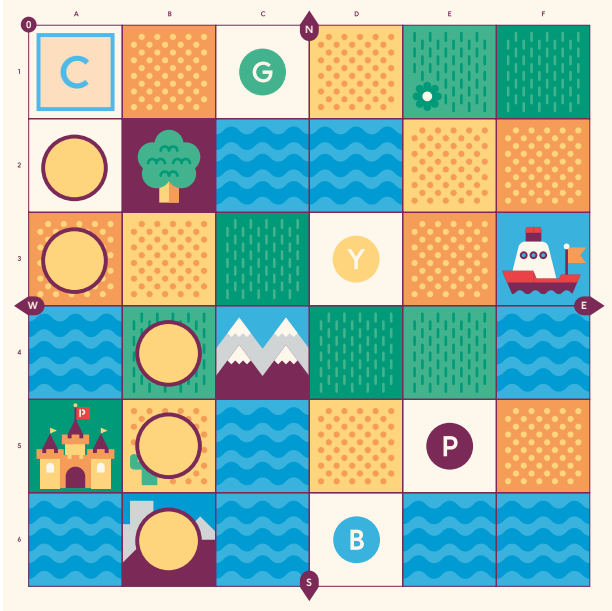
Lesson 3: Cubetto's Treasure Hunt! (2 of 2)

Creative Play

Draw a treasure map to show Cubetto's journey.

Guided Activity

1. Show the coins on the map (see example) and ask: How do you think the treasure got left there?
2. Explain that Cubetto needs to collect all the treasure in one go and that pupils need to find the bug in the algorithm.
3. Look at the algorithm together and ask: Where do you predict Cubetto will move?
4. Show the map template and ask pupils to draw the predicted journey.
5. Ask: Does this algorithm help Cubetto to collect all the treasure? If not, where is the bug? What do we need to change? Discuss.
6. Ask children to use the board template to exchange blocks and create, test and debug the algorithm.
7. Stack the coins on Cubetto as they are collected. When complete, move the coins and repeat.



Independent Activity

1. Add up the treasure Cubetto has collected on the treasure hunt. How many coins are there in total?
2. Look at the first coin value (£1) on the Treasure Chest worksheet: How much money would Cubetto have in total if the coins were £1?
3. Find the £1 coin and draw round it to make the same amount of money.
4. Repeat for other coins in treasure chest.
5. When finished, create a restaurant to use in Lesson 4: Dinner Dash!

Challenge

Can you make Cubetto move faster?

Plenary and Assessment

1. Ask volunteers to share the algorithms they were working on today. Encourage: "The bug in the algorithm was "
2. Ask: What does predict mean? Discuss whether children's predictions were correct.
3. Show treasure hunt algorithm on Board and ask volunteers to place coins on map to show Cubetto's journey.
4. Ask: If each piece of treasure was a £2 coin, how much money would Cubetto have? Repeat for other coin values.

Lesson 4: Cubetto's Dinner Dash (1 of 2)

Cross-curricula Area: Maths: Number: Addition

NC Objectives

To create and debug simple programs

Outcomes

- I can debug an algorithm
- I can solve problems involving addition and money

Resources Needed

- Images of popular restaurants
- Restaurant model (Lesson 3)
- Algorithm to debug
- Number lines/ squares

Prep Needed

- Check batteries.
- Copy dinner plate template and menu.
- Create algorithms to debug, with random block.

Resources Provided

- Restaurant menu
- Dinner plate template
- Maps

Key Vocabulary

- Random
- Predict
- Replace
- Collect
- Route

Computational thinking concept

Logic

Computational thinking approach

Creating

Teacher-led Introduction

1. Ask: What did you have for dinner last night?
2. Ask: Have you ever been to a restaurant? Show images and ask children to identify type of food. Ask: What do you think Cubetto likes to eat?
3. Explain that Cubetto is going out for dinner and can't be late!
4. Show algorithm including random block (black) and ask: What do you think this block does? Children to discuss in groups and present ideas.
5. Demonstrate algorithm and ask: Do you know now what this block does?! Children to discuss in groups and present reasons.
6. Demonstrate block again and show shock that Cubetto did something different this time!
7. Explain this is the random block and ask: What does random mean? We can't predict or guess what will happen.
8. Ask: Why might this cause bugs in an algorithm? We won't know what Cubetto will do!
9. Recap the three types of bugs children have found so far (add to display): missing blocks, incorrect blocks and the random block.

Lesson 4: Cubetto's Dinner Dash (2 of 2)

Creative Play

Role play ordering and paying for food in a restaurant.

Guided Activity

1. Place restaurant on the map and look at the menu together. Ask: What would you eat in this restaurant?
2. Show example algorithm for Cubetto's journey to the restaurant and ask: Will this algorithm get Cubetto to the restaurant? What do you predict?
3. Ask children to use map to draw their prediction and feedback. Answers should vary if they understand the random block.
4. Ask: Why does the random block make writing algorithms difficult? Demonstrate the Board using random block and Cubetto's movement, explaining it is impossible to predict where Cubetto will go.
5. Ask: How do we debug this algorithm? Encourage children to think about which block they can replace the random block with to make it work. Children to discuss in pairs and present ideas.
6. Model testing the new algorithm and choosing an item on the menu, then place a tick by it.
7. Tell children to work in a group to collect all the items on the menu by debugging and testing out new algorithms.

Independent Activity

1. Look at the menu and choose three things that Cubetto would like to eat for dinner.
2. Using your number line/square, add up how much money Cubetto will spend if it orders all three things.
3. On the dinner plate template, draw a healthier meal for Cubetto. What could Cubetto eat instead of burger and fries for a main meal?

Challenge

Could we replace the random block with the function one to speed it up?

Plenary and Assessment

1. Ask volunteers to share how they debugged the algorithms today. Ask: What did you replace the random block with?
2. Show a route on the map and ask children to take turns adding a block to the Board to make an algorithm that gets Cubetto to his destination.
3. Ask: What if we replaced one block with the random one? What would happen?
4. Tell the children what you'd like to order on the menu and ask: How much money would I spend?

Lesson 5: Backpack Boat Trip! (1 of 2)

Cross-curricular Area: Maths (Position)

NC Objectives

To debug a simple program

Outcomes

- I can use the function block to simplify an algorithm
- I can solve problems involving direction

Resources Needed

- Large compass to display
- Algorithm to debug
- Card for boat designs
- Push pins

Prep Needed

- Laminate water squares and port.
- Create an algorithm to debug inc. function bloc

Resources Provided

- Water & port squares
- Map template
- Compass template

Key Vocabulary

- Compass
- North/East/South/West
- Function
- Direction
- Bug/Debug
- Route

Teacher-led Introduction

1. Ask children to stand up and show large compass on Board. Ask: Where is this on Cubetto's map? What does a compass help us do?
2. Recap North, South, East and West and give compass directions for children to follow with different movements (e.g. hop South; wiggle North).
3. Explain that North is the same as moving forward using the green block. Ask: What is South the same as? Repeat for other blocks.
4. Recap blocks and ask: What is the function button? Why do we call it a backpack? How is it useful?
5. Ask children to find castle and the boat on the map. Explain that Cubetto is on a boat and wants to take a journey but can only travel on water.
6. Show the water squares and model placing water blocks to make a route for the boat.
7. Ask: Which directions does Cubetto need to move in? Encourage pupils to use compass points.

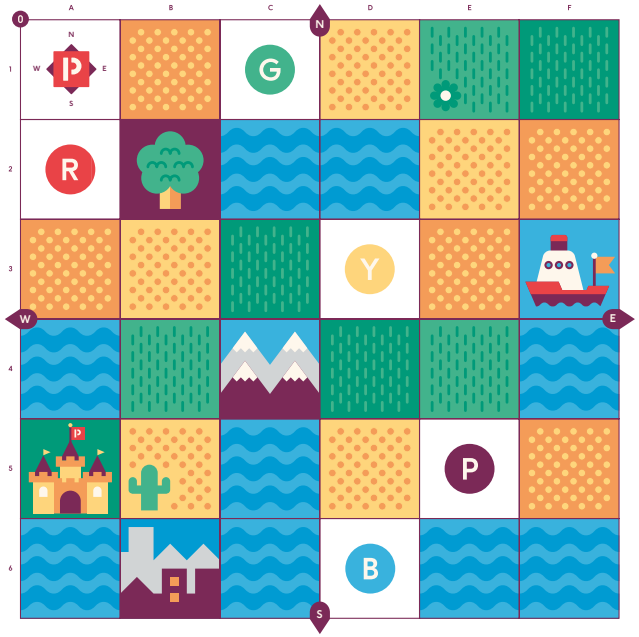
Lesson 5: Backpack Boat Trip! (2 of 2)

Creative Play

Make a boat picture using 2D shapes.

Guided Activity

1. Hand out water squares and ask: Where could we put a square to help Cubetto's boat move further
2. Explore placement of water square on different squares, modelling: "To the South / To the North".
3. Show algorithm for Cubetto's boat journey and ask: Where do you predict Cubetto will end up?
4. Tell children to use the map template to show where they think Cubetto will move.
5. Run algorithm to show problem (e.g. bumps into the mountains because backpack is full of green blocks).
6. Ask: What happened? What did the the function block do?
7. Ask: So how could we debug this algorithm?
8. Place port on map and ask children to decide in pairs where to place water squares, then create and test new algorithm to get Cubetto to the port using the function block.



Independent Activity

1. Draw two sides of a boat and cut these out.
2. Colour in and stick on Cubetto's sides to make into a boat.
3. Ask an adult to help you place the push pin in the compass.
4. Label the compass with the different directions and give a friend instructions on how to move.

Challenge

Without the function block, how far could Cubetto's boat get?

Plenary and Assessment

1. Ask volunteers to share how and when they used the function button.
2. Show the compass on the board for reference. Ask a volunteer to stand at the front, facing the board (the same direction as the class). Tell the volunteer to shout a direction and pupils turn to face the direction given.
3. Show the map and ask: Where would Cubetto end up if it sails: North > West > South from the boat?

Lesson 6: Cubetto's Maze Madness (1 of 2)

Cross-curricular Area: Maths: Position & Movement

NC Objectives

To debug simple programs

Outcomes

- I can predict what a simple algorithm will do
- I can create the boundaries of a maze

Resources Needed

- Pictures of simple mazes from above
- Blank 4x4 grid on board
- Coloured card

Prep Needed

- Check batteries
- Copy maze
- Cut coloured card into strips.
- Prepared algorithm for children to edit.

Resources Provided

- Maze template

Key Vocabulary

- Maze
- Entrance/exit
- Algorithm
- Quarter turn
- Bug/Debug
- Direction
- Route

Computational thinking concept

Logic

Computational thinking approach

Collaborating

Teacher-led Introduction

1. Show video of person walking through a maze: https://www.youtube.com/watch?v=tL3S-kzW_iU and ask: What is a maze? Have you ever been to a maze? Children to talk in groups and feedback.
2. Show pictures of mazes from above and explain that today pupils are going to create a maze for Cubetto!
3. Explain that a maze must have an entrance and exit.
4. Model creating a maze on a 4x4 grid on board using four lines, identify entrance and exit.
5. Ask: What do you think Cubetto is scared of? Draw a monster on the maze for Cubetto to avoid.
6. Model creating maze on real map using coloured card strips.
7. Ask: Can you help me write an algorithm to get to the end? Collect children's ideas and encourage pupils to come to front to create it.
8. Model using language of forward, backwards and ¼ turn left and right to describe Cubetto's route through the maze.

Lesson 6: Cubetto's Maze Madness (2 of 2)

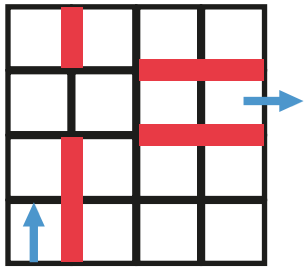
Creative Play

Design a scary monster for your maze.

Challenge

Can two Cubettos use your maze at the same time?

Guided Activity



1. Ask: Can you be sneaky and change my algorithm so Cubetto gets eaten by the monster? Cover your eyes while children change algorithm!
2. Run algorithm and tell children where you think the bug is (what they changed).
3. Recap the bugs they have identified: missing blocks, incorrect blocks, random buttons, incorrect functions and now someone changed the algorithm!
4. Tell children to work in pairs to use the coloured card and create their own maze.
5. When their maze is complete, ask children to create and test algorithm to get Cubetto through the maze.
6. Once tested, ask a partner to 'bug' the algorithm to send Cubetto into a monster then their partner debugs.

Independent Activity

1. Look at the maze and find the start and the end.
2. Use your finger to trace a route through the maze. If you get stuck, go backwards and try another route.
3. When you have found a route, draw a line from start to finish.
4. You could add monsters to your maze!
5. Can you write instructions to tell someone how to complete the maze without looking?

Plenary and Assessment

1. Ask volunteers to share their maze solutions with the class.
2. Ask pupils to come to front and show how they used the card to create the maze on the map.
3. Ask: How did your partner bug your algorithm? Why might someone add a bug?
4. Ask: Where else could you add a trap? What can you do if Cubetto gets stuck?

Lesson 7: Dancefloor Disaster! (1 of 2)

Cross-curricular Area: Maths: Using & Applying

NC Objectives

To debug a simple program

Outcomes

- I can debug patterns in an algorithm
- I can solve problems

Resources Needed

- Music
- Chalk
- Dance algorithms

Prep Needed

- Check batteries.
- Create algorithms that repeat (using function block in function line).
- Create algorithm with opposite block.
- Draw 3x3 grid in playground.

Resources Provided

- N/A

Key Vocabulary

- Algorithm
- Opposite
- Function
- Repeat

Computational thinking concept

Algorithms

Computational thinking approach

Tinkering

Teacher-led Introduction

1. Ask: What different dance moves do you know?
2. Put on different music for children to share and join in with different dance moves!
3. Explain that Cubetto is going to a party but he isn't a very good dancer! Ask: What advice would you give to Cubetto?
4. Explain that pupils are going to help Cubetto learn a dance: the side step!
5. Show algorithm including opposite (cream) block and ask: How do you predict Cubetto will move?
6. Run algorithm, point to the opposite block and ask: What do you think this block does?
7. Ask: What does opposite mean? Can you give an example of opposites?
8. Ask children to work in pairs and do the opposite of each other's movements.

Lesson 7: Dancefloor Disaster! (2 of 2)

Creative Play

Create your own dance to music and write instructions.

Guided Activity

1. Show a prepared algorithm with a function block within the function line and ask: What do you predict Cubetto will do?
2. Ask children to stand up and move as if they were Cubetto following the dance algorithm.
3. Collect ideas and run algorithm on map and watch children's reactions. Ask: What is happening?!
4. Ask children to create the same algorithm on their own Boards and to explore how they think Cubetto is repeating itself.
5. Ask: Have you worked out what the function block does in the function line? Why would you put it inside Cubetto's backpack?
6. Explain that the function block in the function line makes Cubetto repeat forever!
7. Give children time to debug, create and test new algorithms to make Cubetto dance. This is an open ended using and applying activity – the children decide whether there are bugs in their algorithm then re-test until they are happy with Cubetto's movement.
8. When Cubetto moves, ask: Is this how you wanted Cubetto to move? Would you like him to move differently? How could you make that happen?

Independent Activity

1. Work in pairs: one is Cubetto the dancing robot and the other is the Board.
2. Cubetto stands on the grid. The board tells Cubetto how to move to make four dance moves (e.g. hop backwards, jump, criss-cross feet etc).
3. Cubetto carries out the algorithm and then pairs work together to write it down in four simple steps.
4. Can you teach another person your dance?

Challenge

Can you make two Cubettos dance together?

Plenary and Assessment

1. Ask volunteer pair to share their four step dance routine.
2. Ask: What is the algorithm for this dance? Write in order on board.
3. Ask: What did we learn that using the function block in the function line makes Cubetto do? How could we make Cubetto do your dance and repeat it forever? Adapt routine for Cubetto's directions, then write routine in function line followed by function block.
4. Test out the children's dance moves on Cubetto.

Lesson 8: Cubetto's Parcel Problem (1 of 2)

Cross-curricular Area: Maths: Problem Solving

NC Objectives

To create and debug a simple program

Outcomes

- I can recognise numbers and their words
- I can use the random block.

Resources Needed

- Dice (and digital dice on board)
- Number line/square
- 3x3 grids on back of maps with numbers 1-9.

Prep Needed

- Check batteries
- Decorate classroom with balloons!
- Copy bingo cards.
- Turn over the maps and make a 3x3 grid using the same size squares (15cm). Label each with a number 1-9.

Resources Provided

- N/A

Key Vocabulary

- Parcel
- Delivery
- Programming
- Bugs

Computational thinking concept

Algorithm

Computational thinking approach

Persevering

Teacher-led Introduction

1. Surprise the children with a 'delivery' of a parcel at the beginning of the lesson.
2. Ask: What do you think it is? What have you received in the post at home before? Children discuss and feedback experiences.
3. Ask for a volunteer to help open it and reveal the decorated Cubetto. Introduce Cubetto's friend: Cubetta!
4. Explain that pupils are going to help Cubetto and Cubetta with a difficult task, using all the knowledge they have gained about programming.
5. Ask: Which ways can we make Cubetto move? Recap all block functions.
6. Ask: What bugs have we found in algorithms? Refer to display: missing blocks, random buttons, incorrect blocks, function line, people changing blocks!
7. Explain that Cubetto needs to deliver a parcel to Cubetta (a birthday present!) in the castle as quickly as it can but there is a problem with the algorithms!

Lesson 8: Cubetto's Parcel Problem (2 of 2)

Creative Play

Draw Cubetto a picture of your favourite adventure.

Challenge

Could you move the parcel and keep the algorithm the same?

Guided Activity

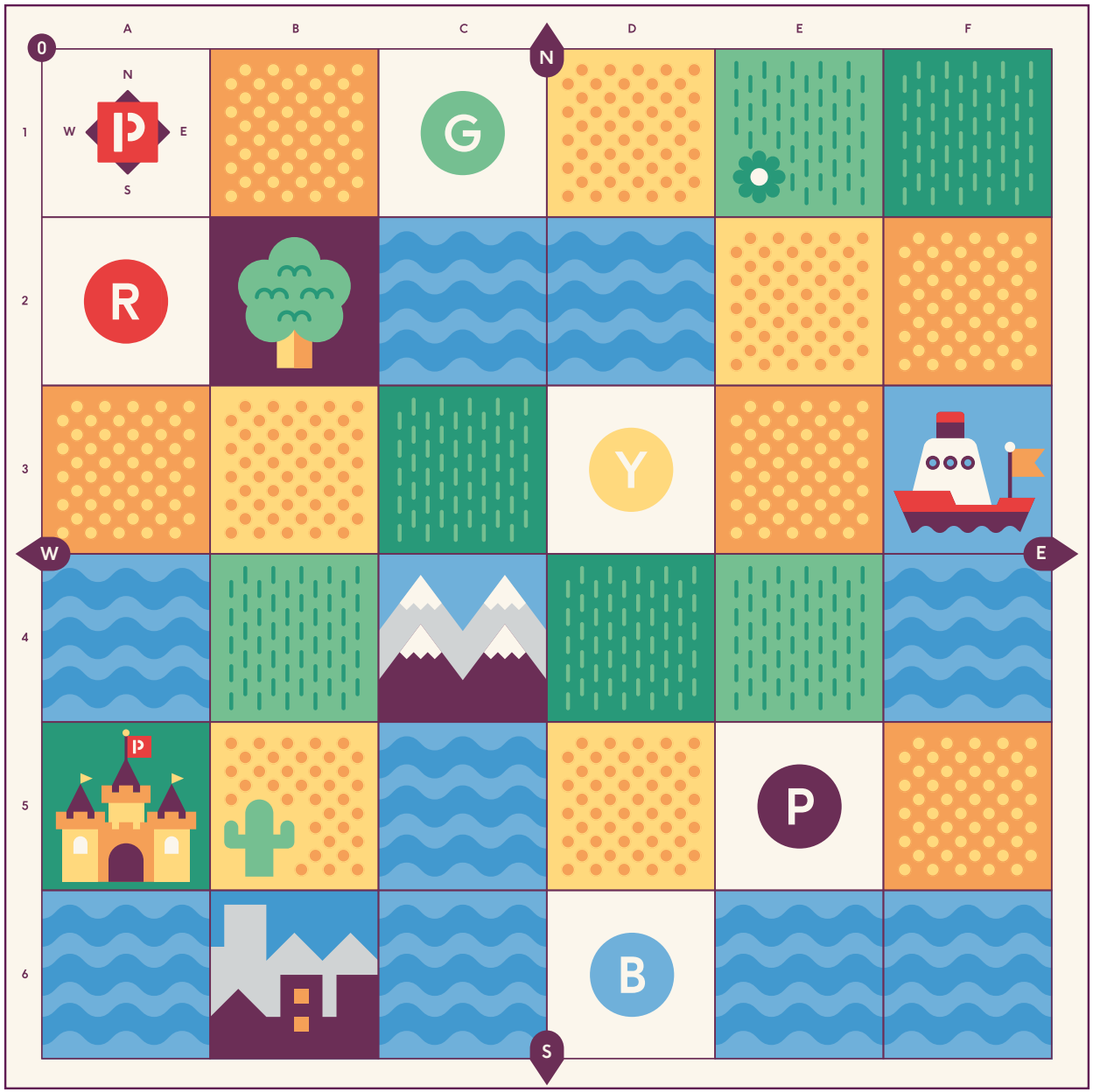
1. Place the parcel on the map and explain that Cubetto always starts on the compass, collects the parcel and ends at the castle.
2. Show algorithm on board template to include: incorrect first turn, function line moving forward 3, random block, and one missing.
3. Ask: What do you see? Give children time to describe all they understand about bugs through their explanation.
4. Ask: Where do you predict Cubetto will go? Discuss.
5. Run algorithm several of times and discuss children's predictions. Ask: Are there any bugs that need fixing? How would you fix it? Encourage children to sequence their order of debugging this algorithm: there are many stages to fixing the algorithm.
6. Give children time to debug, test and run the algorithm.
7. Tell children to swap their algorithm with another person and to test them out. Ask: Could the algorithm be improved? How?

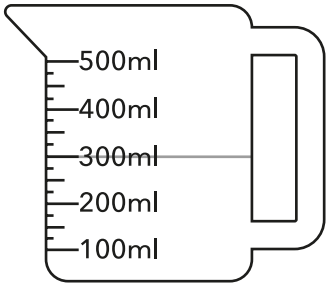
Independent Activity

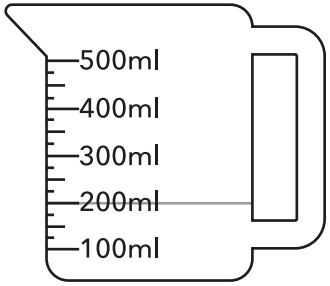
1. Think about all the things you have learnt about Cubetto and what he can do. Where has Cubetto travelled to? What has he done?
2. Write a letter to Cubetta telling them all about the adventures and problems they have helped Cubetto to solve.
3. Can you add tips for Cubetta to avoid getting intro trouble (getting bugs) in the future? Look at the display to remind you.

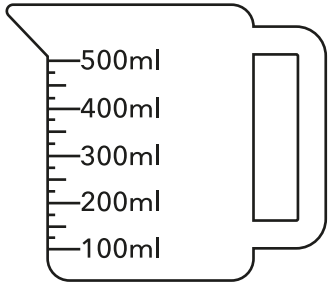
Plenary and Assessment

1. Ask volunteers to share their solutions to debugging the algorithms. Ask: What steps did you take to fix it? What did you change first? Did you help improve a friend's algorithm? How?
2. Ask volunteers to share their letters to Cubetto telling them all about what they have done with Cubetto.
3. Look together at the Bug Display and recap what the children have learnt about the different bugs and how to solve them.
4. Tell the children that they are now expert computer programmers, just like Grace Hopper!

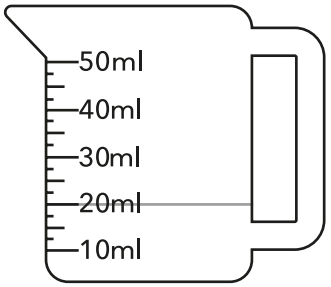


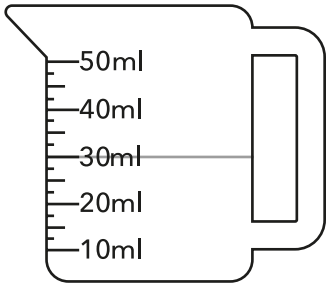


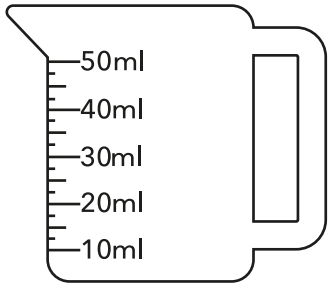




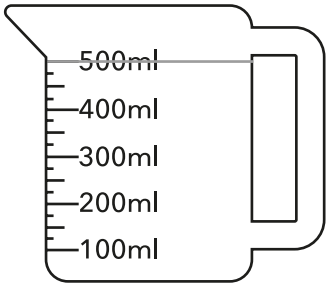
400ml

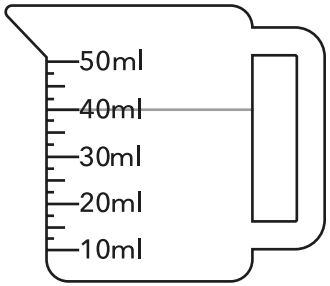


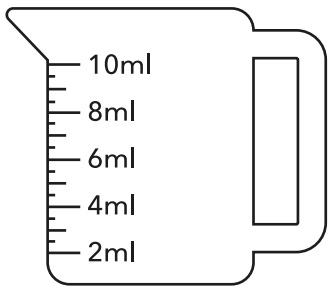




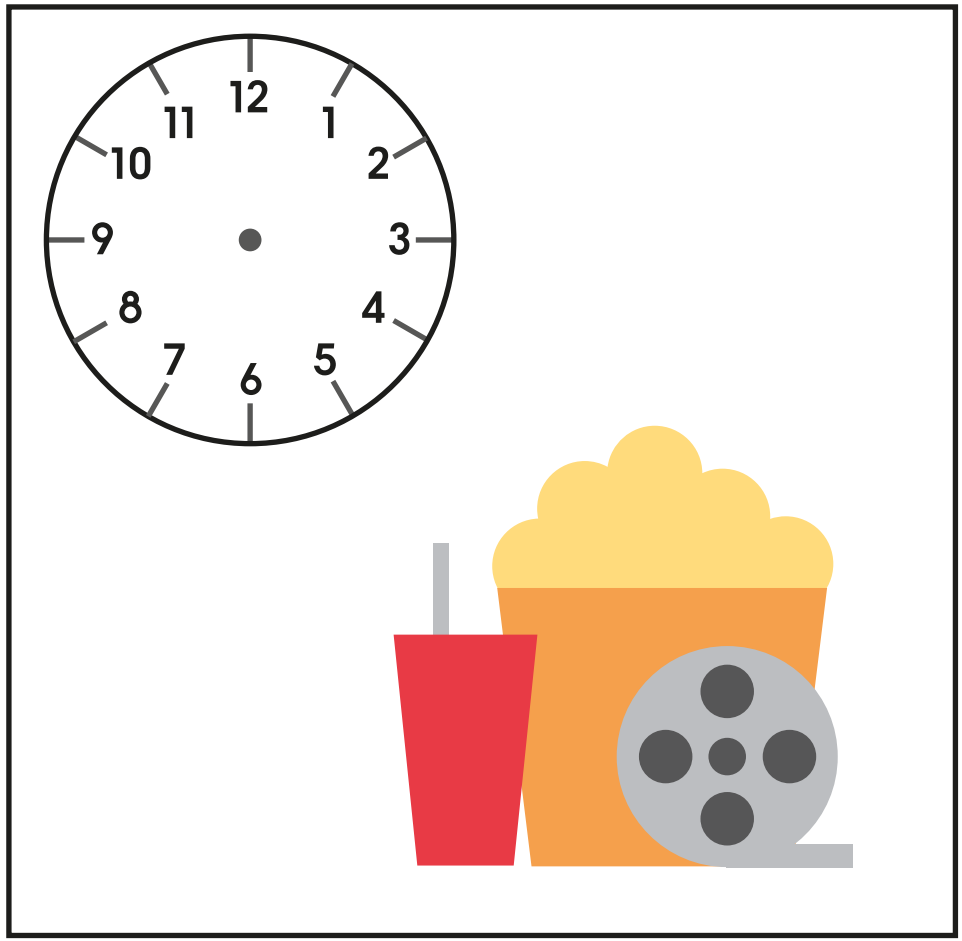
10ml

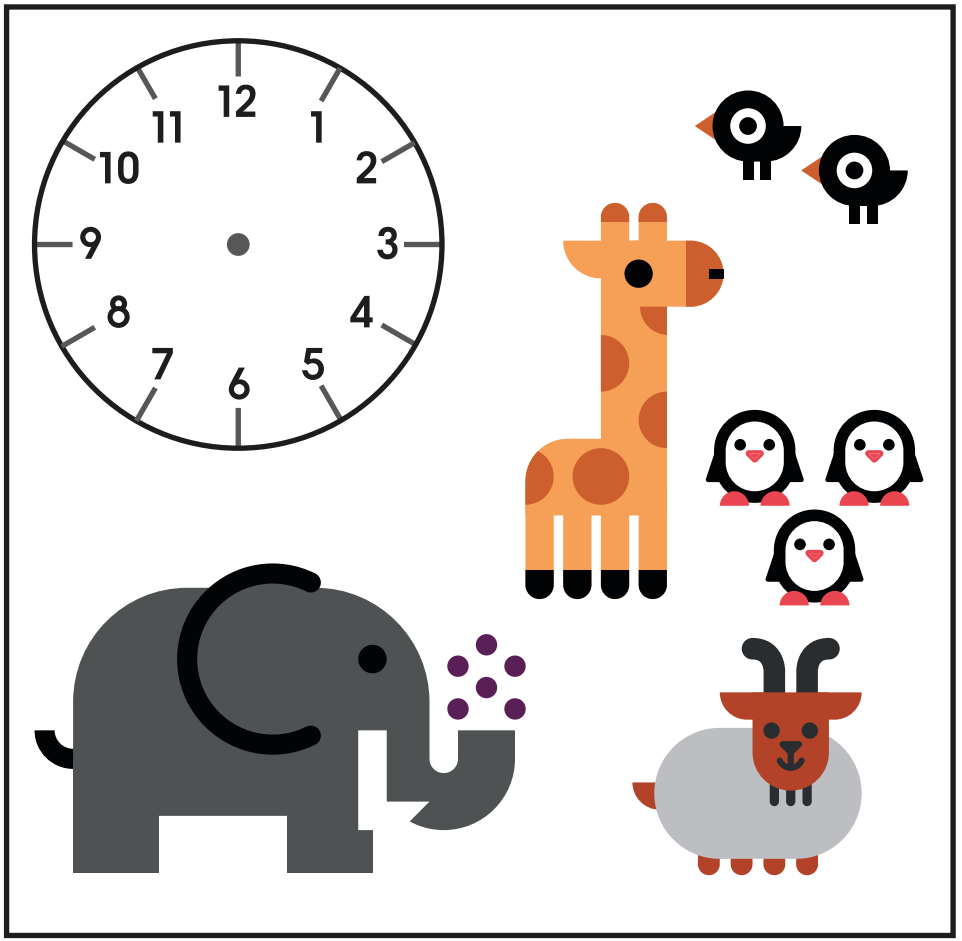


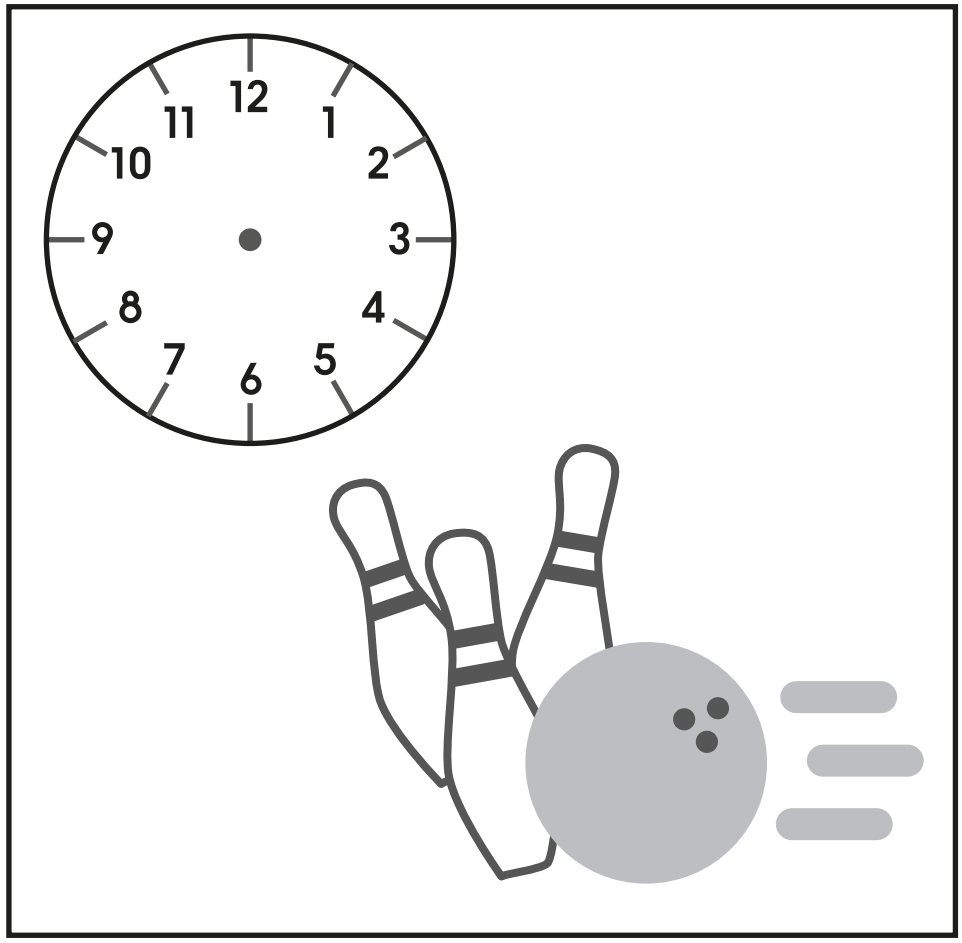


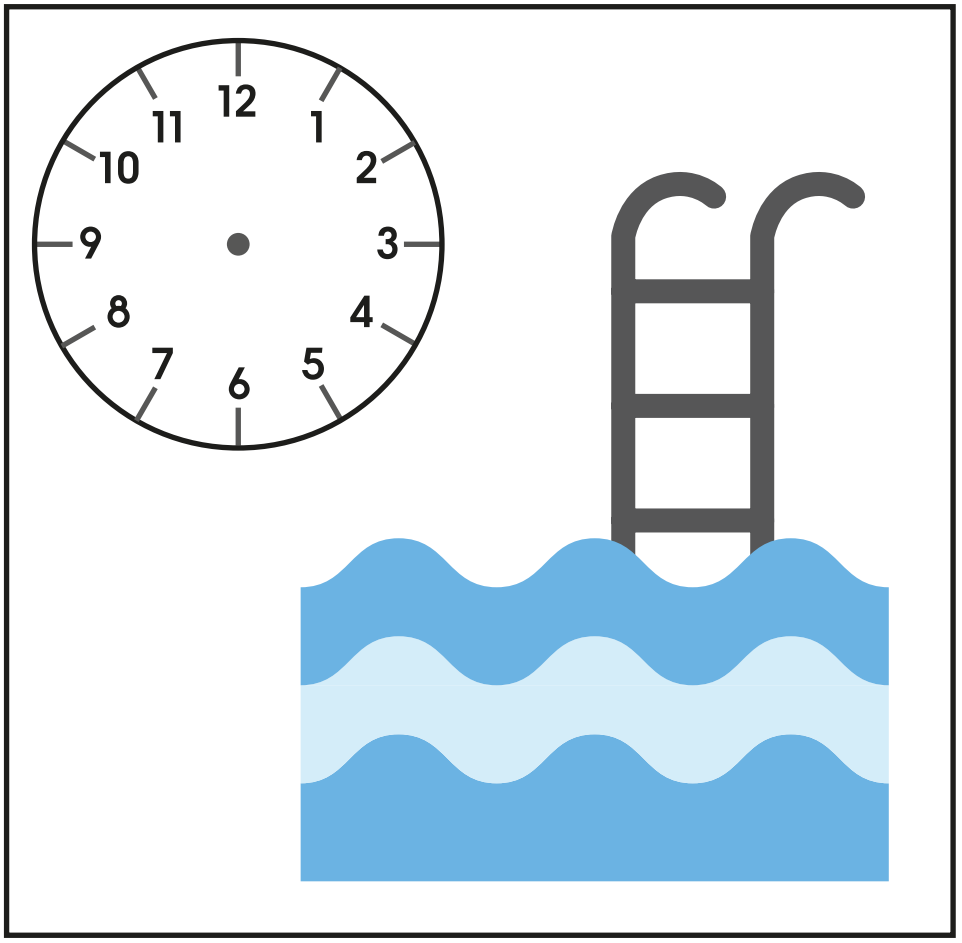


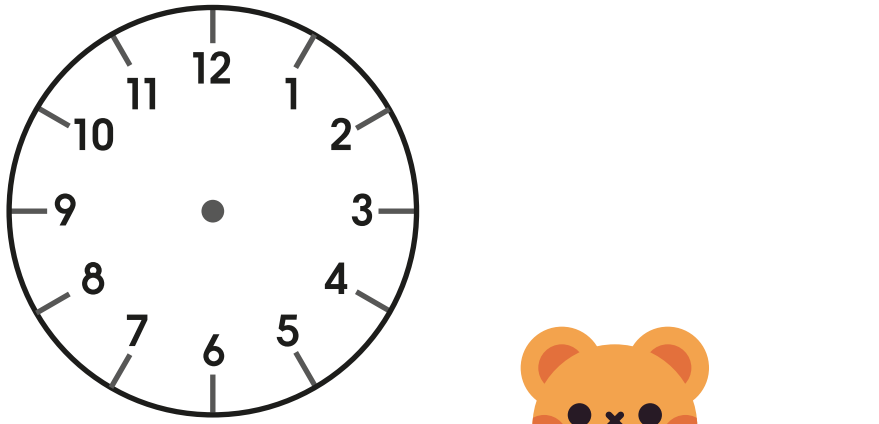
8ml

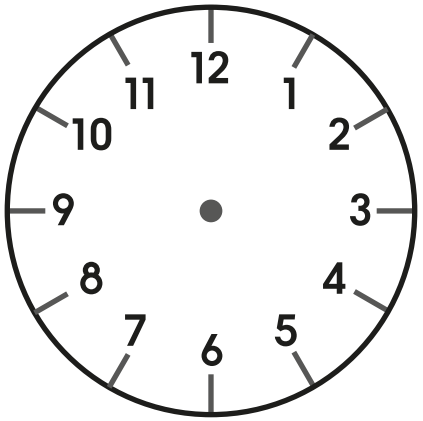


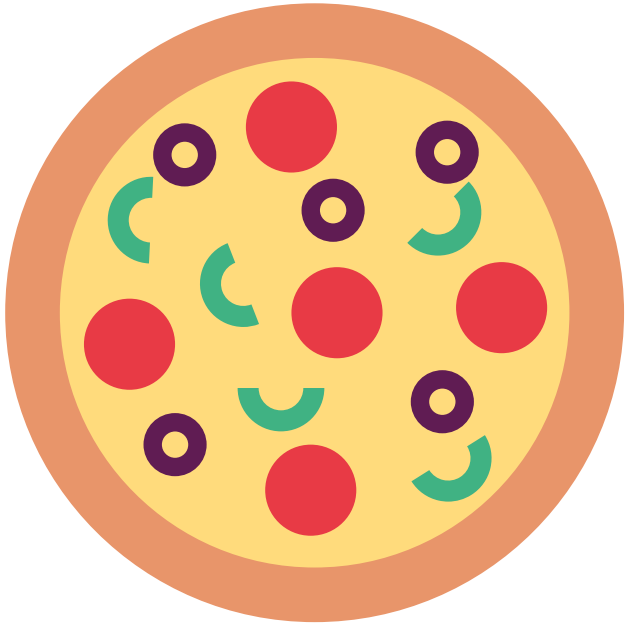


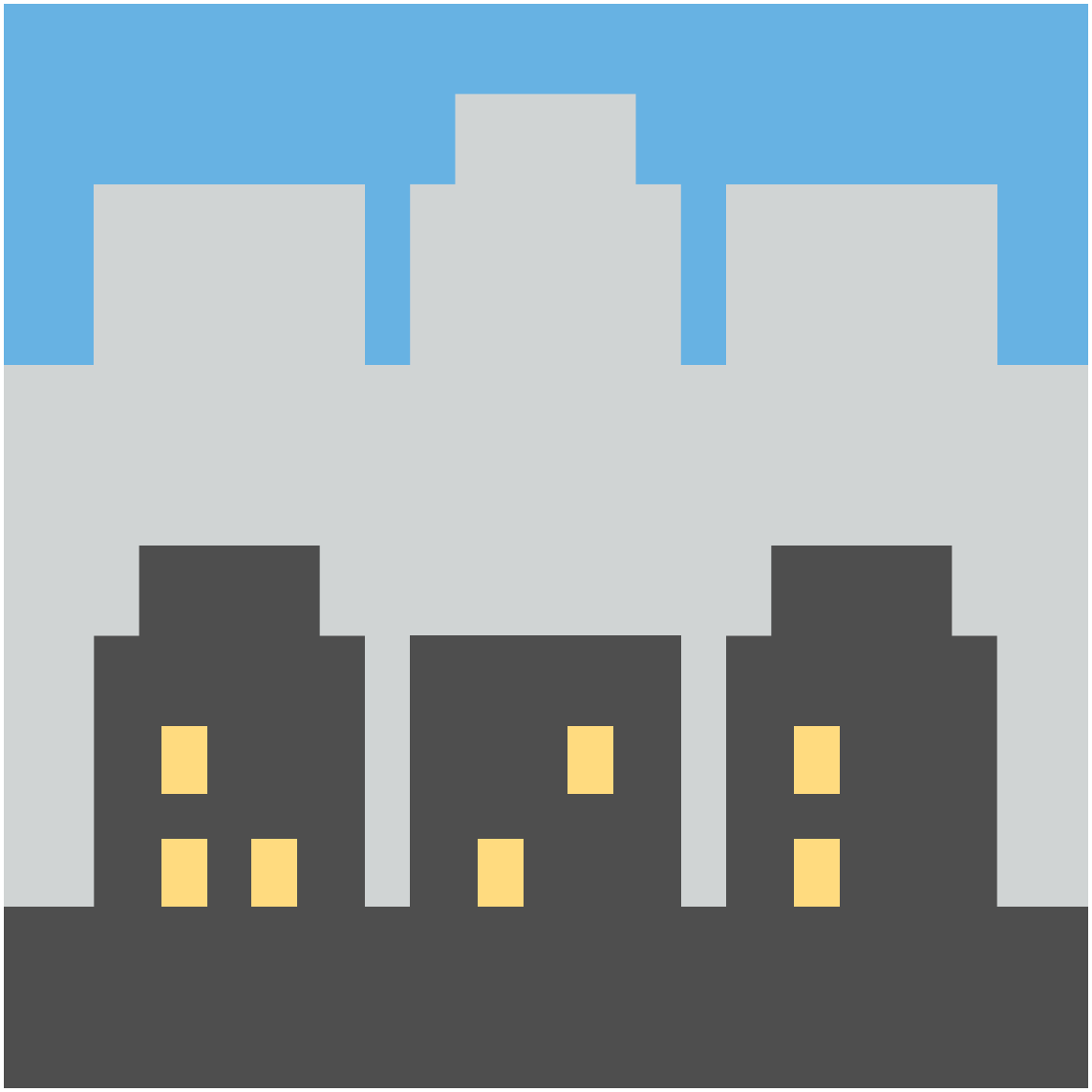


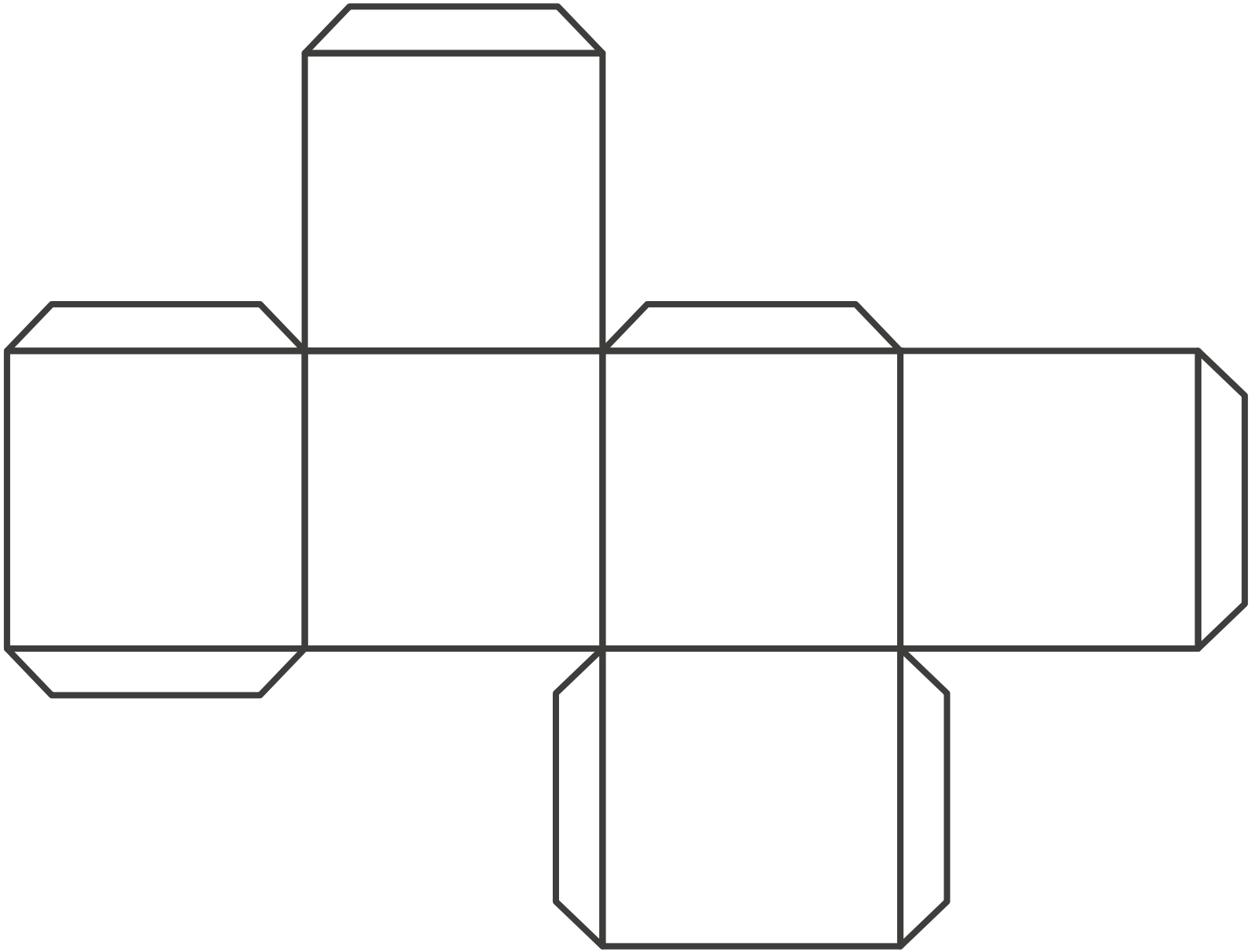


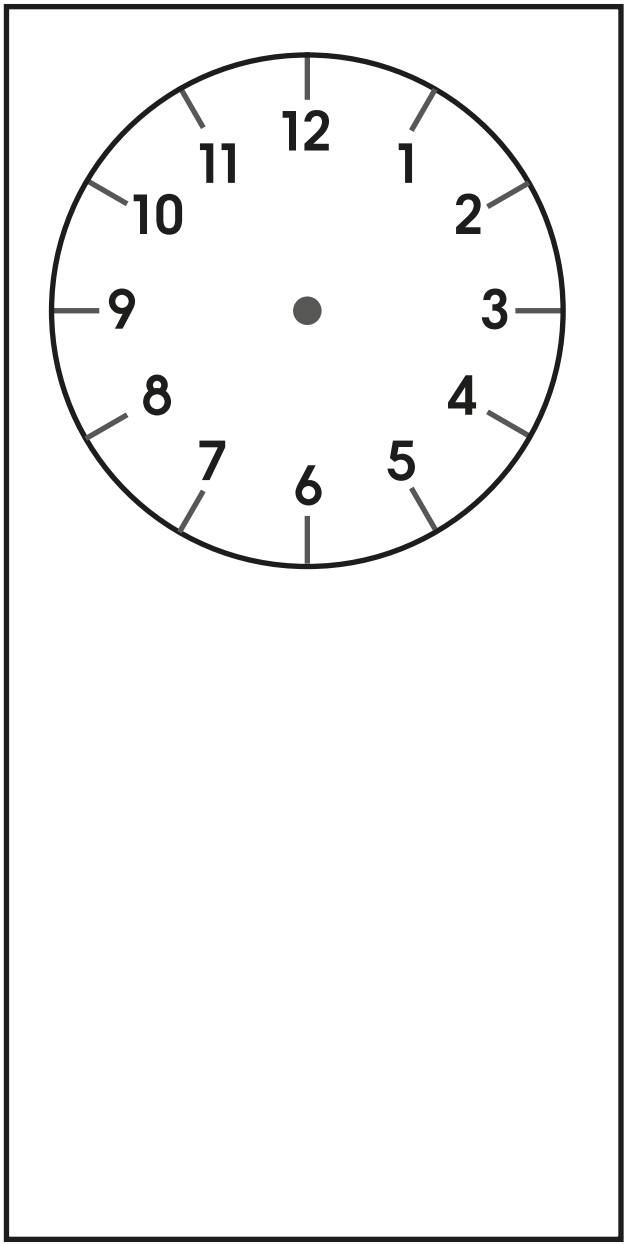


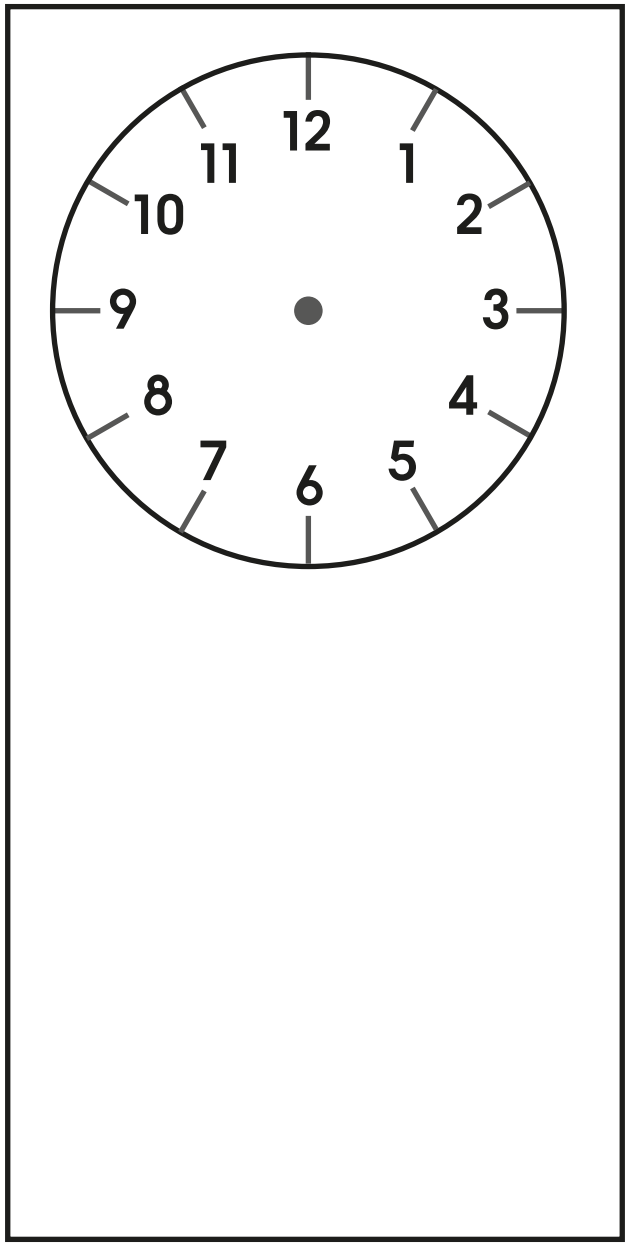


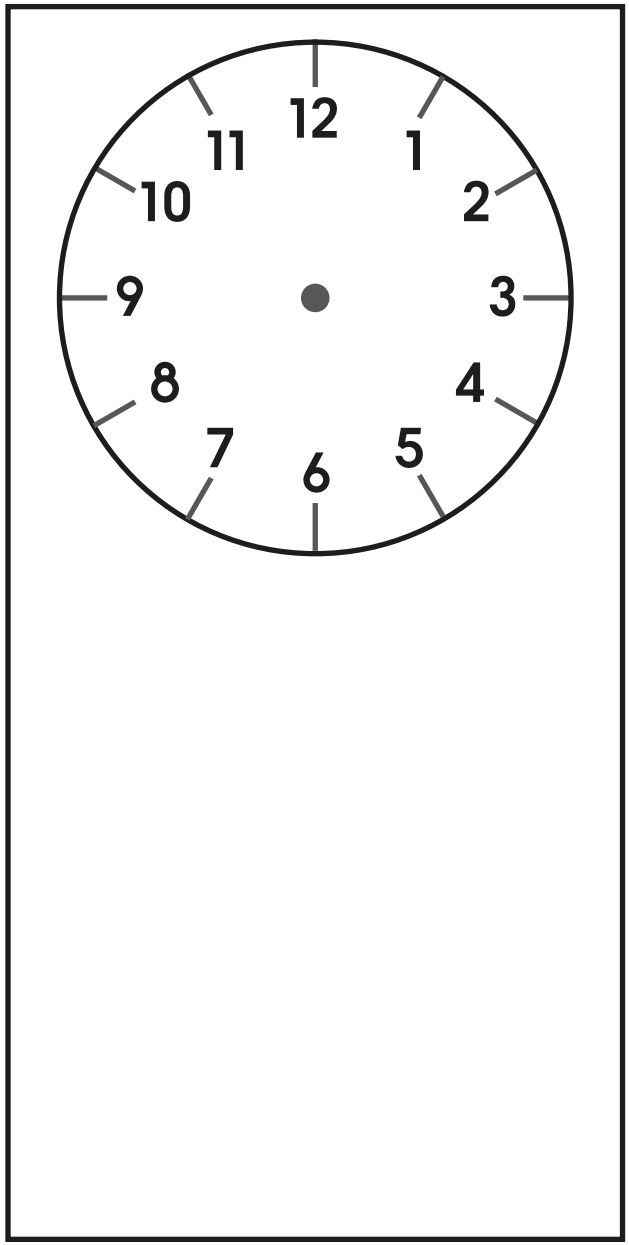


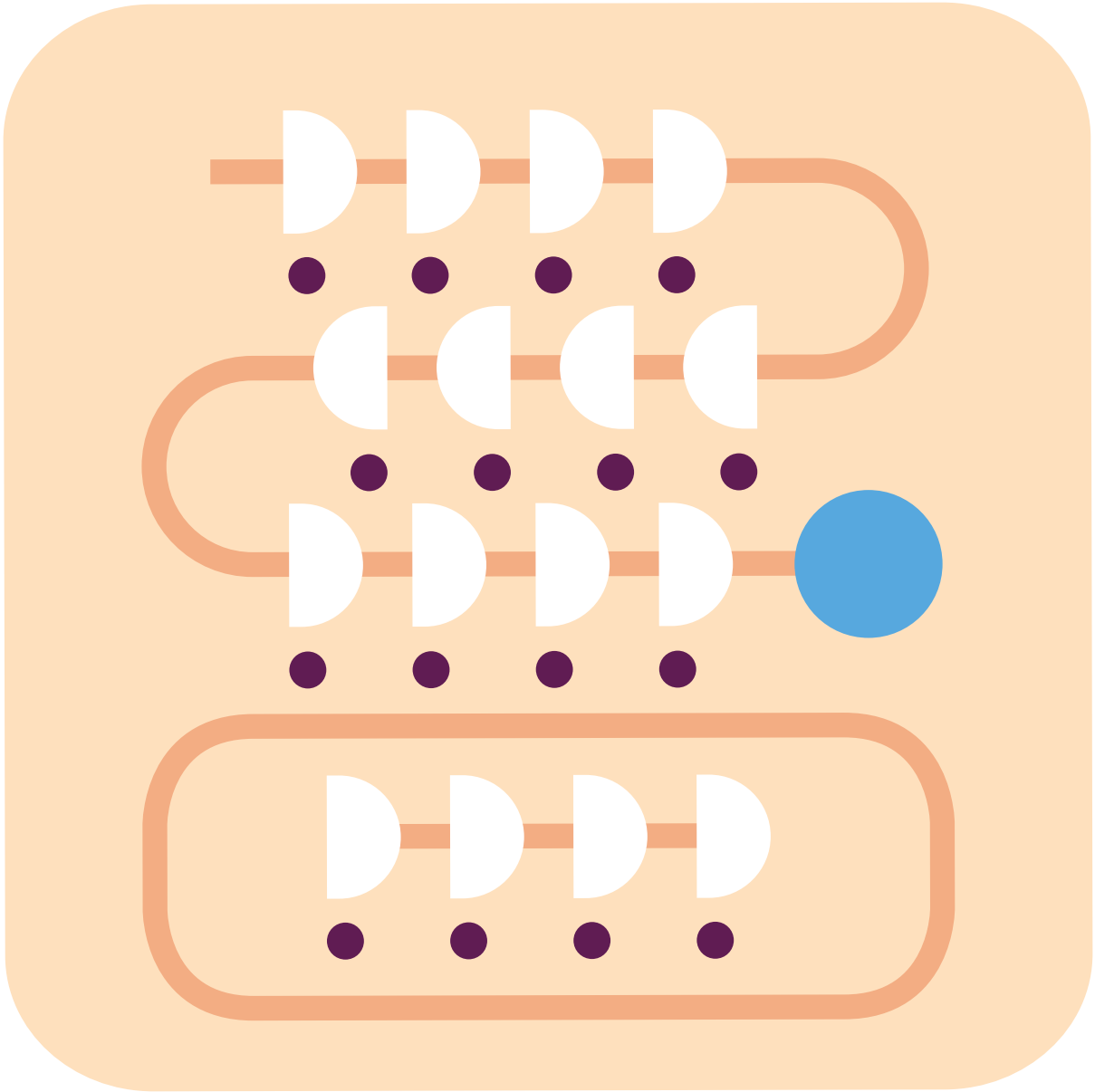


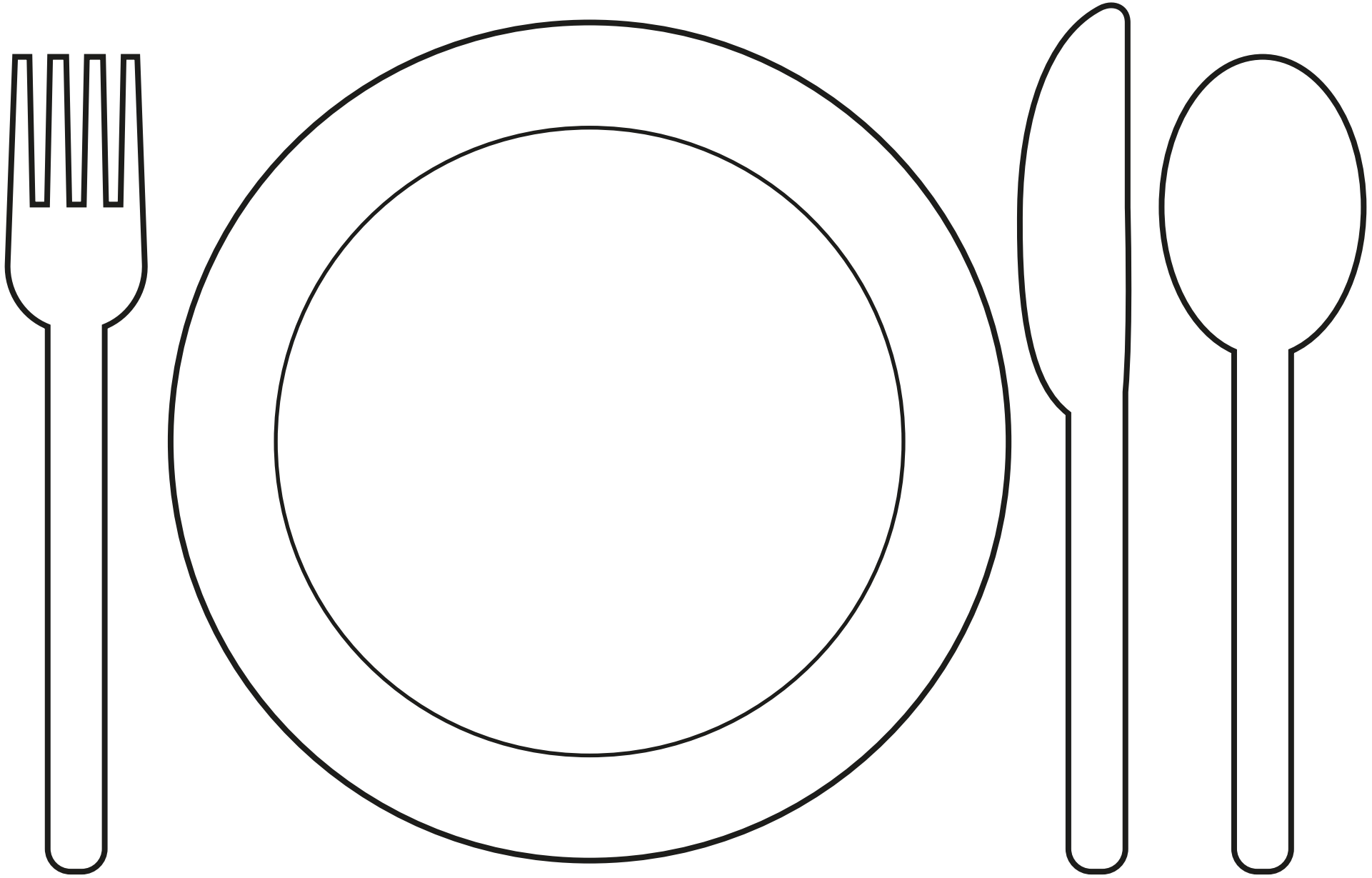


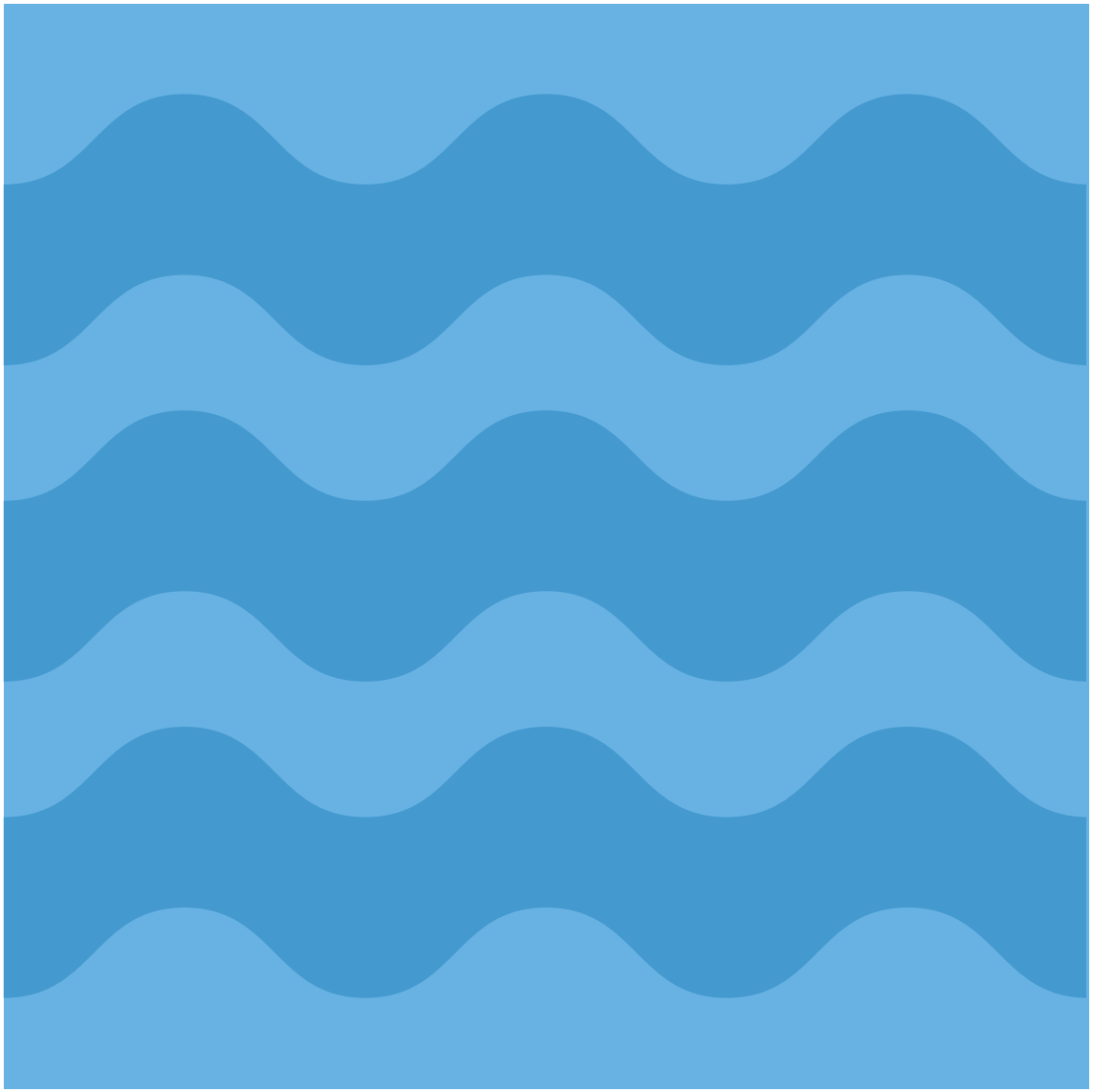


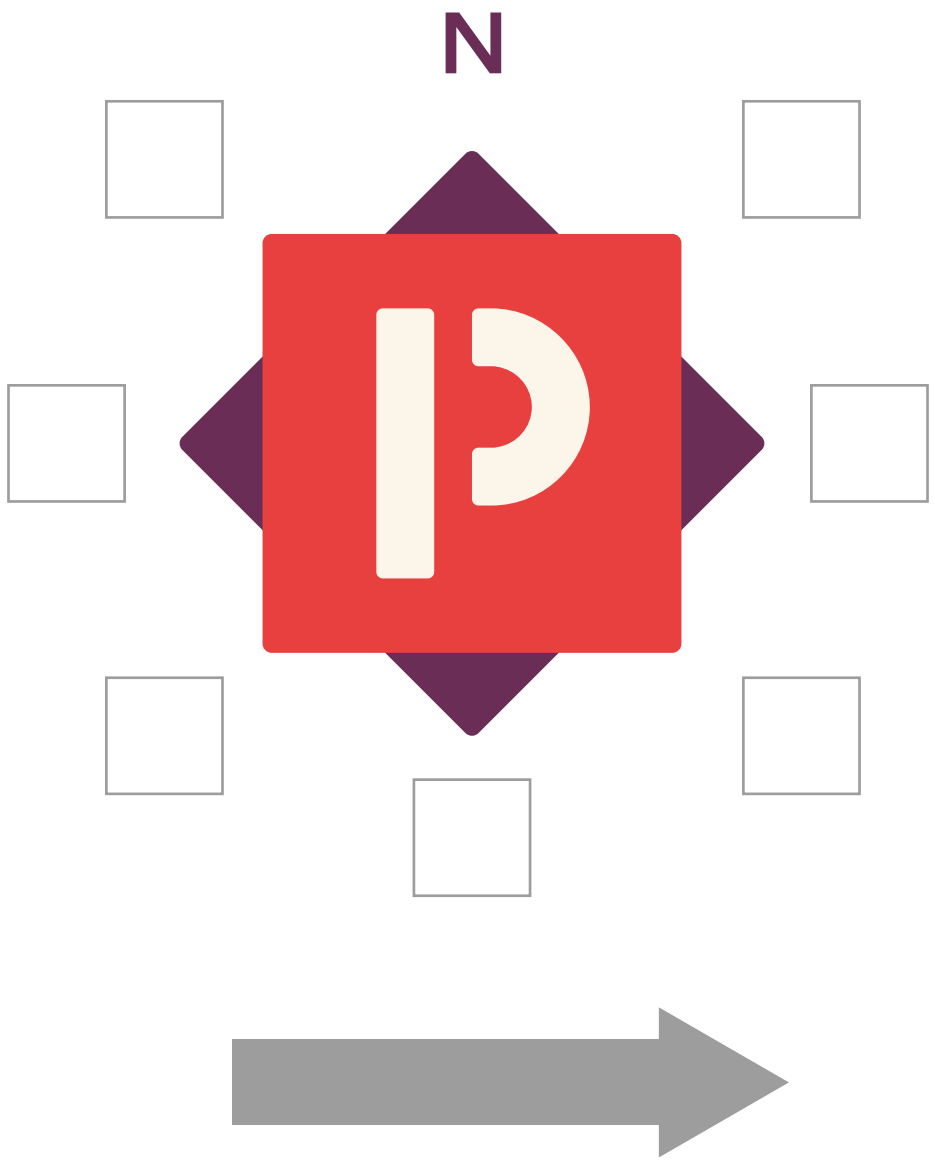




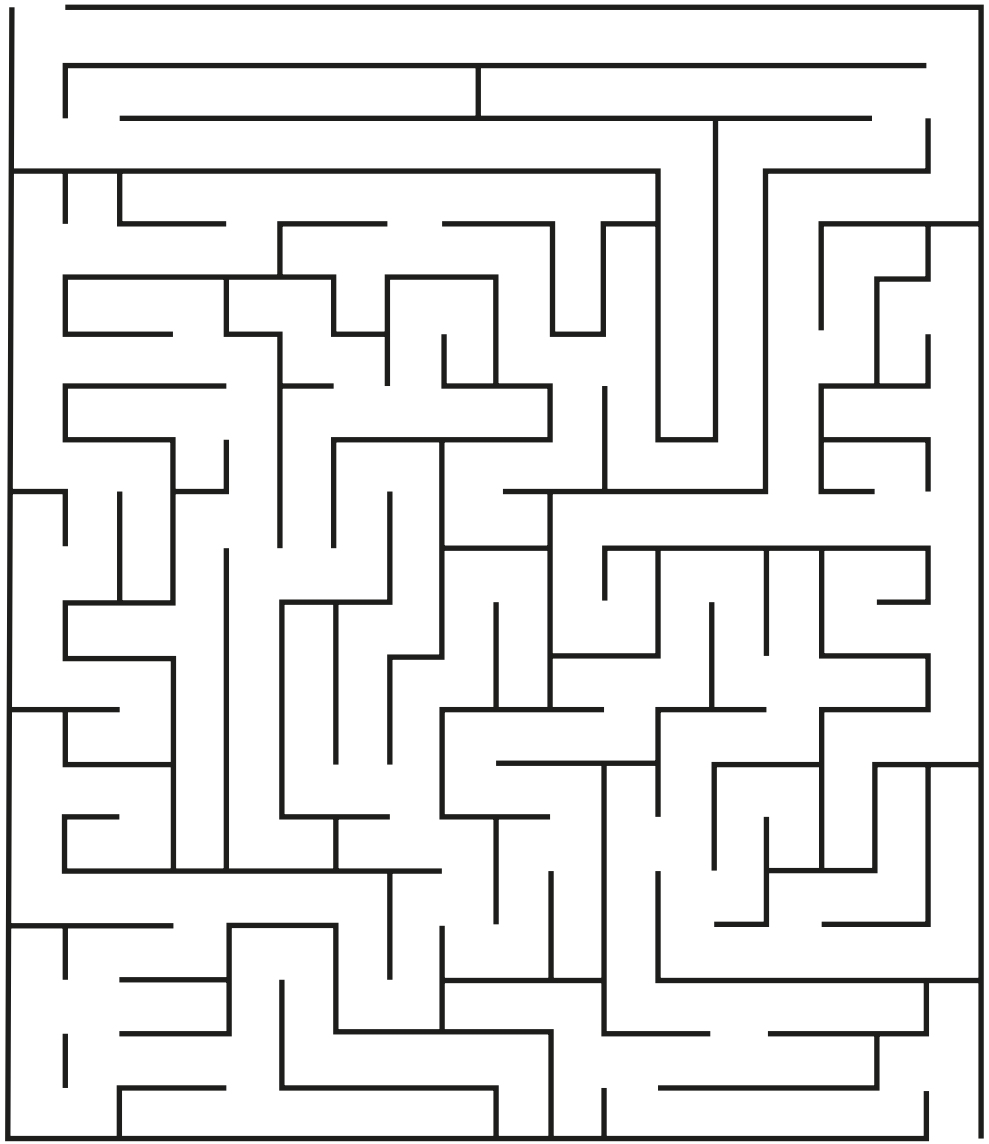








START



FINISH

PRIMO

Visit our website:

www.primotoys.com

Get in touch:

edu@primotoys.com