

TEACHERS' GUIDE

Table of Contents

Organizing you work with Scottie Go! — 7

Notes for teachers — 9

Module I - Europe 1 — 11

Module Introduction - commands — 12

Quest 1 - controlling the game character, first commands — 13

Quest 2 — 15

Quest 3 - digits, creating numbers, learning addition and subtraction — 16

Quest 4 - making a turn — 18

Quest 5 — 20

Quest 6 — 21

Quest 7 — 23

Quest 8 - walking backwards — 25

Quest 9 — 27

Quest 10 — 29

Module II - Europe 2 — 31

Module introduction - picking up objects from the game-board — 32

Quest 1 — 33

Quest 2 — 35

Quest 3 — 37

Quest 4 — 39

Quest 5 - jumping — 41

Quest 6 — 44

Quest 7 - note — 46

Quest 8 - note — 49

Quest 9 — 52

Quest 10 — 55

Table of Contents

Module III - South America — 57

Module Introduction - the REPEAT loop — 58

Quest 1 - the first loop — 59

Quest 2 — 62

Quest 3 - working inside a loop and outside of a loop — 64

Quest 4 - Scottie GO! EDU as a tool for introducing notions in maths and physics — 67

Quest 5 — 70

Quest 6 — 72

Quest 7 - nested loops — 74

Quest 8 — 77

Module IV - North America — 79

Quest 1 - beaver as a bridge — 80

Quest 2 — 82

Quest 3 — 84

Quest 4 - beaver in motion — 86

Quest 5 — 89

Quest 6 - using REPEAT loop and the SWITCH TO tiles together — 92

Quest 7 - the Beaver makes a turn — 95

Quest 8 — 98

Quest 9 — 100

Quest 10 — 102

Module V - Antarctic — 104

Module Introduction - interaction with Scottie's surroundings - ACTIVATE and PLACE tiles — 105

Quest 1 - the ACTIVATE tile — 107

Quest 2 - the PLACE tile — 110

Quest 3 — 113

...

Table of Contents

Module V - Antarctic ctnd. — 104

...

Quest 4 — 115

Quest 5 — 117

Quest 6 — 120

Quest 7 - SWITCH TO the beavers and move ice floes — 122

Quest 8 — 125

Quest 9 — 127

Quest 10 — 130

Module VI - Asia — 132

Module introduction - conditional loop - REPEAT WHILE — 133

Quest 1 - using the REPEAT WHILE tile — 135

Quest 2 — 138

Quest 3 — 141

Quest 4 - REPEAT WHILE with other conditions — 144

Quest 5 - REPEAT WHILE nested in a loop — 147

Quest 6 — 150

Quest 7 — 153

Quest 8 — 155

Module VII - Australia 1 — 158

Module introduction - VARIABLES — 159

Quest 1 - building aphalt roads — 162

Quest 2 — 165

Quest 3 — 167

Quest 4 - X and Y — 170

Quest 5 — 173

...

Table of Contents

Module VII - Australia 1 ctnd. — 158

...

Quest 6 - variables and loops — 175

Quest 7 — 178

Quest 8 - mathematical operations on variables — 180

Quest 9 — 183

Quest 10 — 185

Module VIII - Australia 2 — 187

Module introduction - conditional statements - IF, ELSE — 188

Quest 1 - action square — 192

Quest 2 — 195

Quest 3 — 197

Quest 4 — 199

Quest 5 — 201

Quest 6 - using multiple conditional statements — 204

Quest 7 — 207

Quest 8 - ELEVATED SQUARE and HERE tiles — 209

Module IX - North America 2 — 212

Module introduction (A) - ELSE IF — 213

Module introduction (B) - the ELSE tile — 216

Module introduction (C) - [optional] - REPEAT (FOREVER) — 219

Quest 1 - ELSE — 222

Quest 2 - ELSE IF and MARKS — 226

Quest 3 — 231

Quest 4 — 233

Quest 5 — 235

...

Table of Contents

Module IX - North America 2 ctnd. — 212

...

Quest 6 — 238

Quest 7 - conveyor belts — 240

Quest 8 - switches — 243

Quest 9 — 246

Module X - Africa — 248

...

Organizing your work with Scottie Go! (1/2)

Make sure the workplace is well-lit

Playing with Scottie Go! requires your pupils to take photos of the code they wrote. Not enough lighting may make it impossible to scan the code properly.

Never combine tiles from different kits

Each tile in a kit has its unique ID number and graphics recognized by the camera of your mobile device. If you mix up tiles from several sets in one code, they may not be recognized properly. We recommend that you make a mark on the back of tiles from each kit with a separate colour to distinguish them from tiles that belong to other kits.

NEVER COMBINE THE HOME VERSION WITH THE EDUCATIONAL VERSION TILES

In the optimal model we recommend to use one kit box per 2 pupils

Before you BEGIN using Scottie Go! EDU with your pupils, make sure they know the rules, e.g. you may want to make a contract with your pupils. If you do so, make sure that one of the rules tells your pupils not to lend each other tiles from their kits. When the class finishes, make sure they pack the tiles back into their respective boxes so as not to mix up the tiles from different kits. If you work with younger children, we recommend that you prepare all the required tiles in advance.

Create a special teacher account that gives you access to all of the quests

You can create a teacher account that will give you access to all quests in all modules. You will be able to access any quest even if you haven't completed the whole Scottie Go! course before. The options accessible through the teacher account allow you to use Scottie Go! while teaching various subjects to cover the programming concepts of your choice, depending on your pupils' age and skill level.

To create the **Teacher** account, choose the following option in the **Scottie Go! Edu** app.

SELECT PLAYER -> NEW PLAYER and then enter this code:

Teacher_13_cps#

DO NOT REVEAL THIS CODE TO PUPILS!

Organizing your work with Scottie Go! (2/2)

The Teacher account allows your pupils to work in teams. In this model you will need only one device to display the Quest description and one mobile device to scan the programs created by your pupils.

The Teacher account allows you to make the best use of Scottie Go! progress cards. To download additional materials, log into your account at:
www.scottiego.com/education

Notes for teachers (1/2)

There is more than one correct solution

This guide provides you with solutions to each quest. Remember that there are numerous ways to solve the same problems with various programs - some of them may even be shorter than the suggested solution.

The game comprises 179 tiles that are active in every module and in every quest

You can solve each quest again and again, using the new programming concepts as well as new tiles introduced in subsequent quests and modules.

Scottie Go! Edu app employs the mechanics of a game, but is a full-fledged educational tool

While using Scottie GO! Edu app, you will be able to both meet your teaching curriculum goals in terms of key coding competencies and tap into the latest world developments in engaging computational thinking in other contexts:

• understanding, analysing and solving problems based on logical and abstract thinking, computational thinking and visual presentation of data,

• programming and problem-solving with the use of computers and other digital devices: assembling and programming algorithms, organising, searching for and sharing information, using computer applications,

• developing social competences, such as: group communication and collaboration, including virtual environments, participation in group projects and project management,

REMEMBER - OUR GAME IS JUST A TOOL THAT ASSISTS YOUR PUPILS IN LEARNING, NOT A GOAL IN ITSELF!

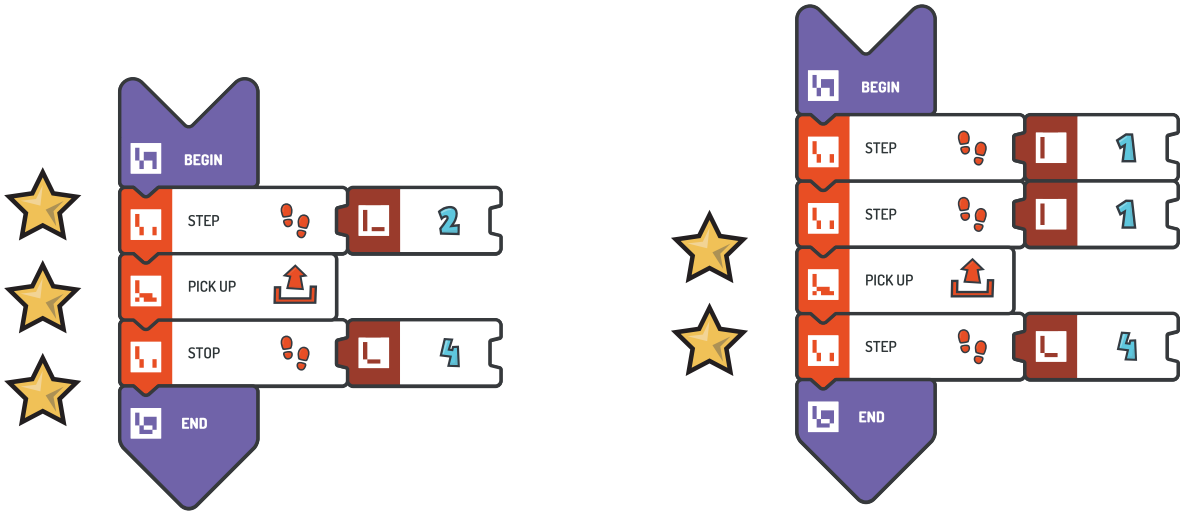
While using Scottie GO! Edu during your lessons, we recommend that you follow the rules related to active methods, lesson objectives, lesson management as well as evaluation. Scottie Go! EDU follows the spiral approach to the newly introduced notions, i.e. in one quest introduces a command which is also used in the subsequent quests in different context. The difficulty level of quests increases with each quest as the game progresses, which allows the teacher and his/her pupils to develop their algorythmic skills. Each time a new command is introduced, it is signalled by an exclamation mark in the top right-hand corner of the game character on your screen.

Notes for teachers (2/2)

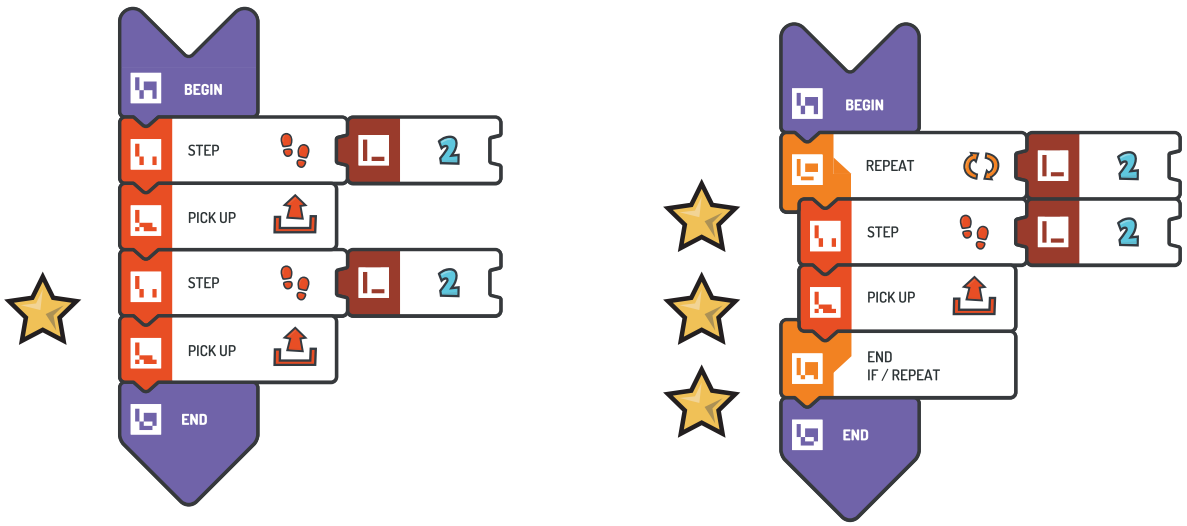
A good coder and an ambitious pupil should always look for an optimal way to solve a problem. The program is "optimal" when it is constructed using as few lines of code as possible and it still achieves its objective.

During their lessons teachers should create the atmosphere that facilitates learning. The **ScottieGo! Edu** app features some elements of formative assessment in an attractive game-based environment. After completing each quest, your pupils will receive from 1 to 3 stars as a reward for the quality and optimisation of their codes. It's a lot of fun without making them compete against each other.

Three stars if the program is the best, two stars if the solution is a bit less efficient. One star means that the quest was solved, but the program was too long.



In some quests, if players want to be granted more than one star, they will need to use specific tiles, e.g. the **REPEAT** tile, in their solution.



Module I
Europe 1



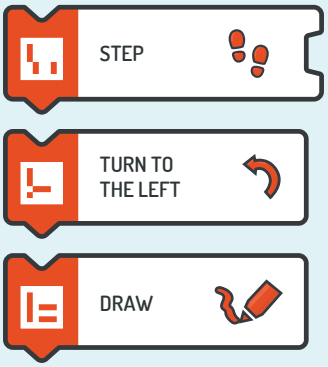
I love scavenger hunt. I wonder how well you will do. Go through all the "X" squares.

Module introduction: commands

In Scottie GO! we distinguish between two types of commands: simple and complex.

Simple commands

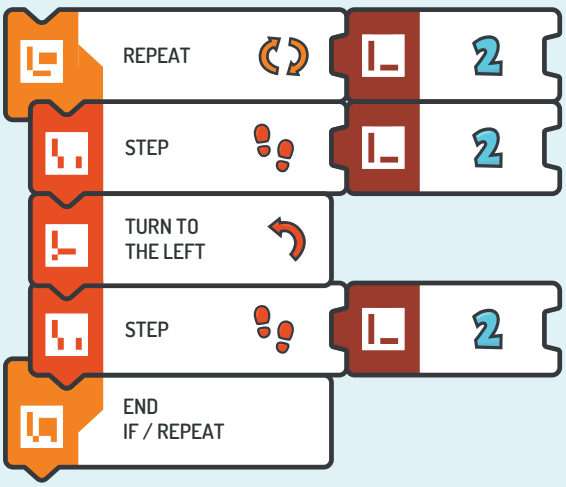
Simple commands are the basic element of any program. Each command contains a single activity or operation to be performed. Examples of simple commands include actions related to how Scottie moves or behaves, e.g.:



Usually one line of code contains one command, just like in Scottie Go! Commands are placed in lines, one under the other, and they are performed in this exact order.

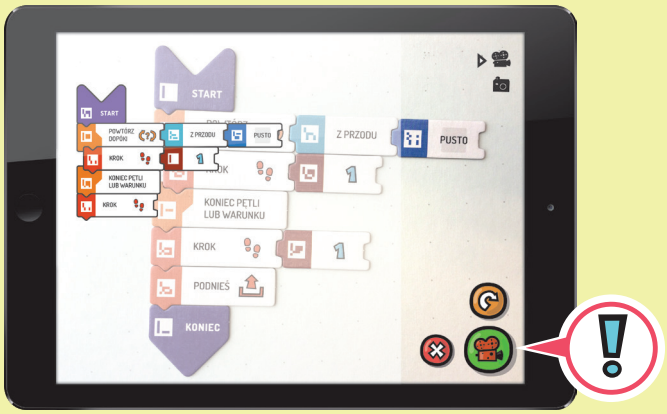
Complex commands

Complex command control how parts or the entire program that we create are interpreted. They usually constitute a sequence of a few simple commands or other complex commands. In general complex commands are placed in brackets. In case of Scottie Go! they are placed between two matching orange blocks. Examples of complex commands include the **REPEAT** loop, **REPEAT WHILE** loop or the conditional command **IF**.



NOTE!
controlling the game character, first commands

The most important aspect of **the first quest** is to teach students how to work with the tutorial included in the **Scottie Go! EDU** app. Students need to learn how to scan programs, which is described on **page 14 of the Scottie Go! User Manual** .

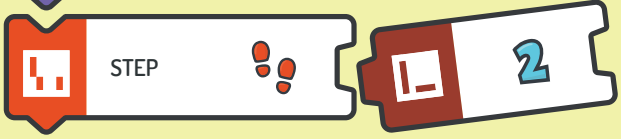


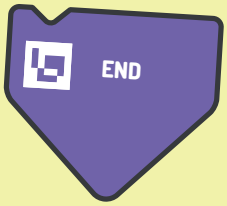
Although usually students have no problems with using the tutorial, they sometimes tend to follow it without focusing on the meaning of the tips. It's important to note that **quests 1-3 in Module I** feature the notion of a **parameter**, which is a key concept throughout the game. A parameter is the value that defines how a command should work. The first command that we learn in the game - **STEP** - requires pupils to define the parameter that controls how many steps Scottie should take.

Each program BEGINs with the **BEGIN** tile.



Between the **BEGIN** and **END** tiles we can place any number of commands that **will be performed in sequence.**





Each program should finish with the **END** tile.

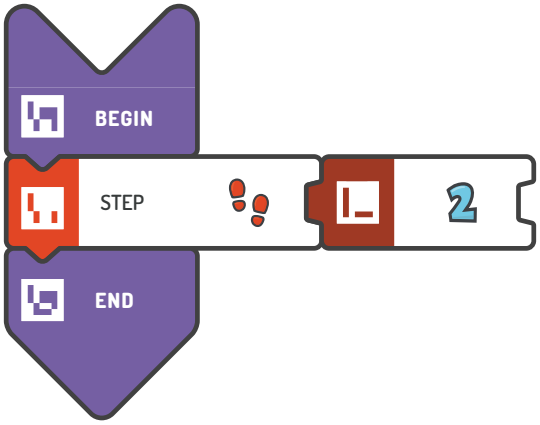
Some commands, e.g. **STEP**, **require a parameter**.

STEP (2) means that Scottie will move 2 squares forward;
STEP (3) means that Scottie will move 3 squares forward, **etc.**

Challenge:



Example solution:



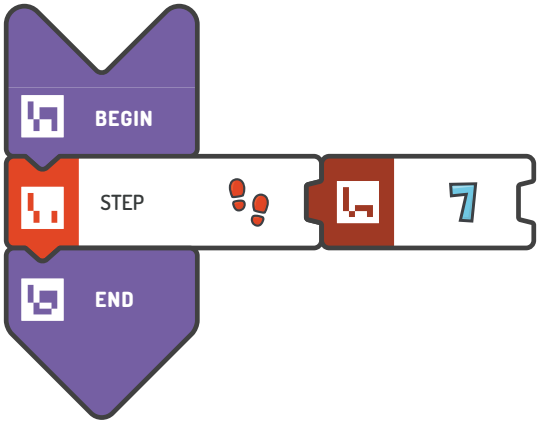
Example solution (in text format)::

```
BEGIN
    STEP (2)
END
```

Challenge:



Example solution:



Example solution (in text format)::

```
BEGIN
    STEP (7)
END
```

Introduction:
digits, creating numbers, learning addition and subtraction

You can create numbers by combining tiles with digits.



You can also create numbers by adding numbers if you use the **PLUS** tile



or by subtracting numbers if you use the **MINUS** tile.

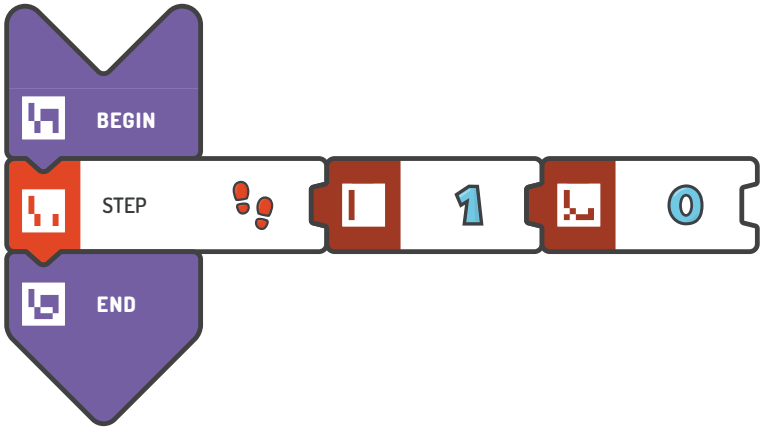


Mathematical operations may include even larger number of tiles (digits/numbers). They may also feature both addition and subtraction.

Challenge:



Example solution:



Example solution (in text format)::

```
BEGIN
    STEP (10)
END
```

Introduction: making a turn

Scottie may turn around. To do so, use the **TURN TO THE LEFT** tile



or **TURN TO THE RIGHT**.

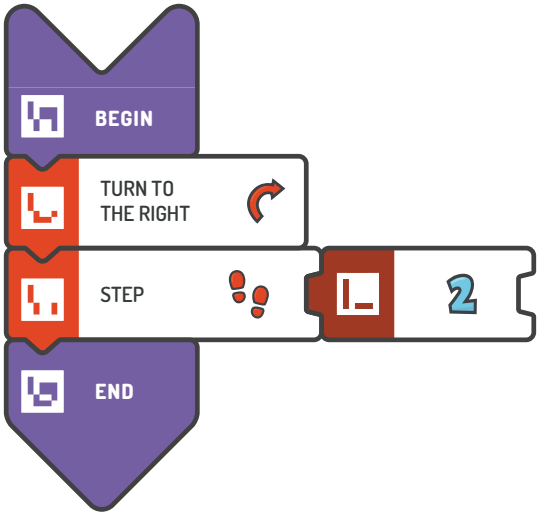


To illustrate how Scottie moves, in this quest your pupils may pretend to be Scottie. It will make it easier for them to understand the way Scottie moves.

Challenge:



Example solution:



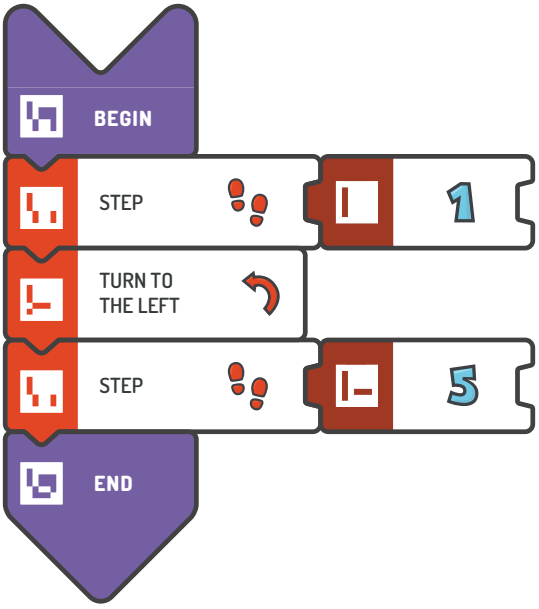
Example solution (in text format)::

```
BEGIN
    TURN TO THE RIGHT
    STEP (2)
END
```

Challenge **(remember that Scottie must go through all the „X" squares)**:



Example solution:



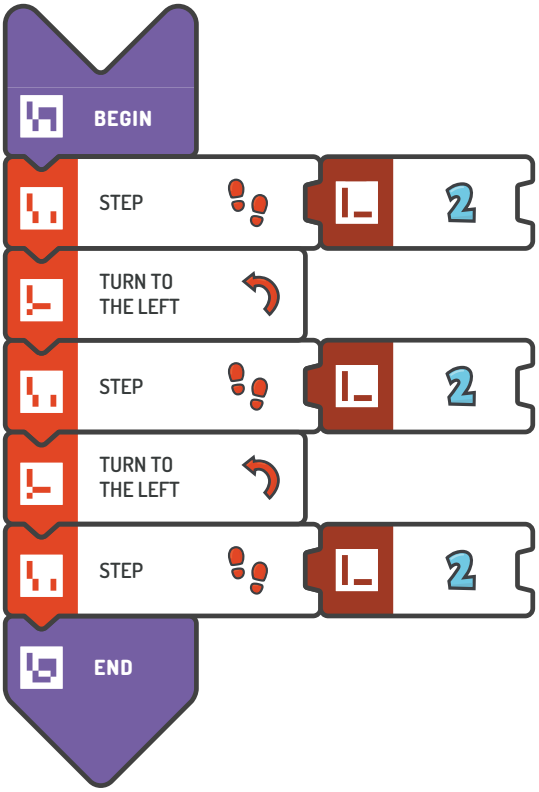
Example solution (in text format)::

BEGIN
STEP (1)
TURN TO THE LEFT
STEP (5)
END

Challenge **(remember that Scottie must go through all the „X" squares)**:



Example solution:



Challenge **(remember that Scottie must go through all the „X" squares)**:



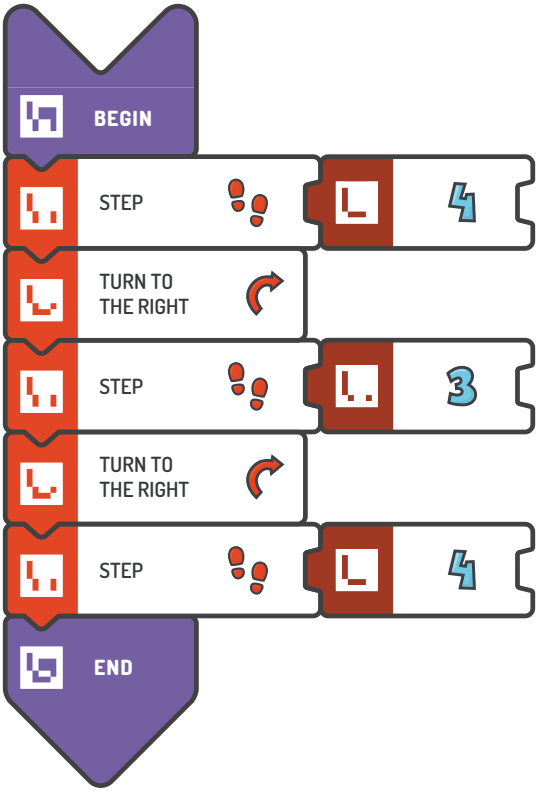
Example solution (in text format)::

```
BEGIN
    STEP (2)
    TURN TO THE LEFT
    STEP (2)
    TURN TO THE LEFT
    STEP (2)
END
```

Challenge:



Example solution:



Challenge:



Example solution (in text format)::

```
BEGIN
    STEP (4)
    TURN TO THE RIGHT
    STEP (3)
    TURN TO THE RIGHT
    STEP (4)
END
```

Introduction: walking backwards

Scottie may also move backwards. To do so, use the **MINUS** tile..



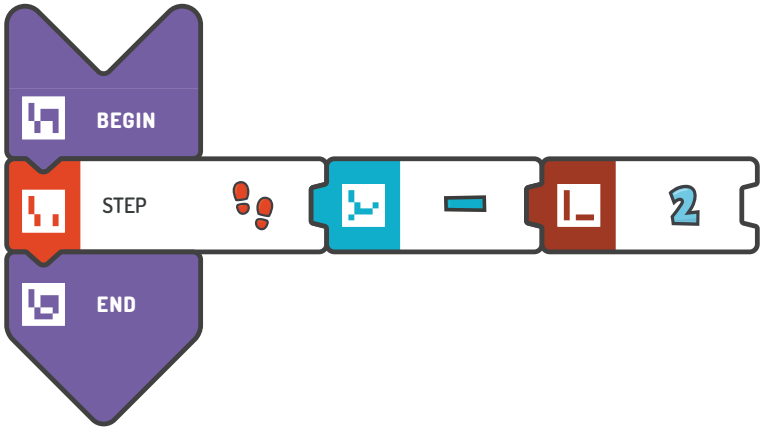
Walking backwards is one of Scottie's new abilities. To perform it in a game, use the **MINUS** tile before the value with the number of steps that Scottie should take.

Pupils who have not covered negative numbers during their maths lessons yet are quick to use the **MINUS** tile as soon as they realize the similarity between the positive and negative temperature that they take for granted and can understand with no difficulty.

Challenge:



Example solution:



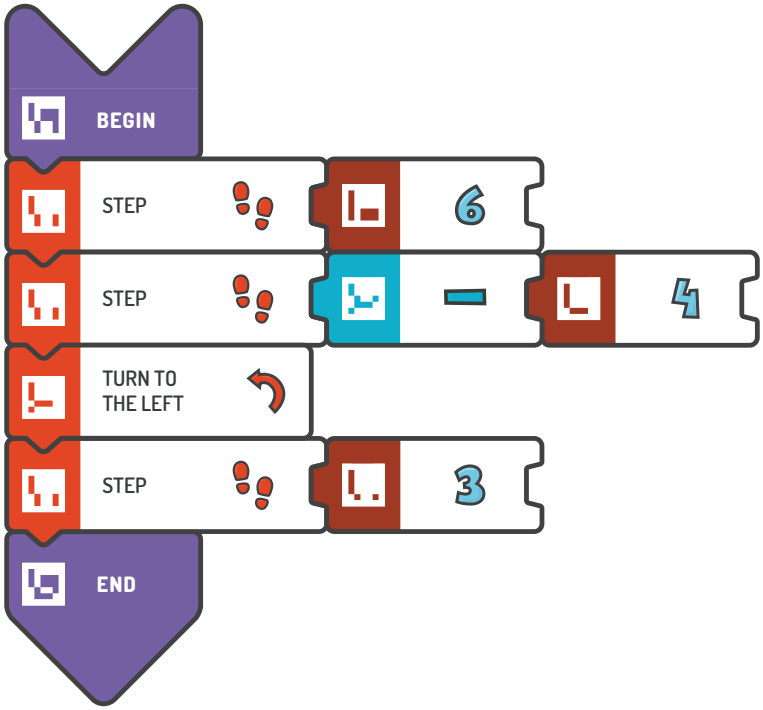
Example solution (in text format)::

```
BEGIN
    STEP (-2)
END
```

Challenge:



Example solution:



Challenge:



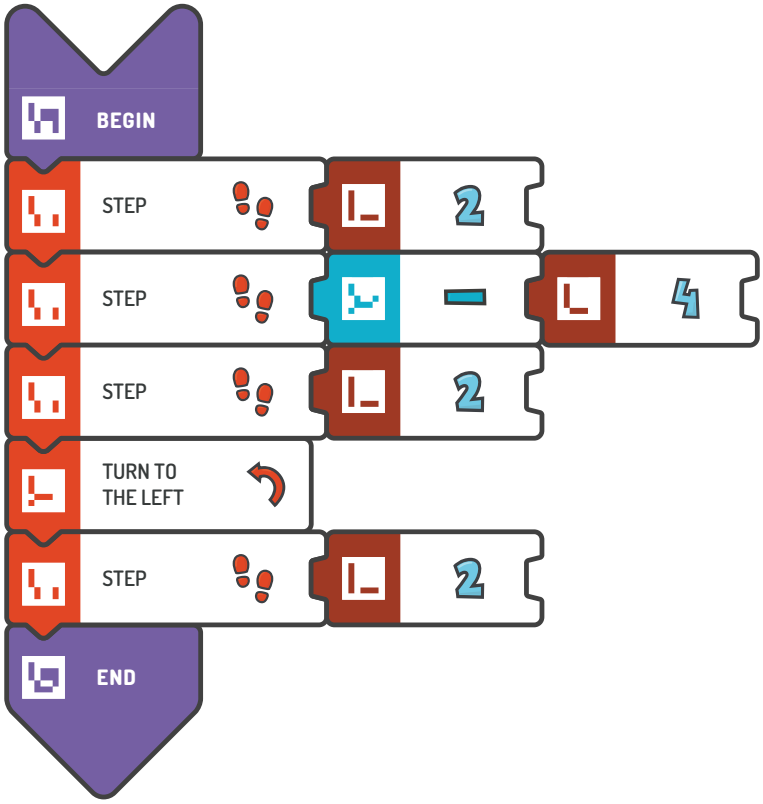
Example solution (in text format)::

```
BEGIN
    STEP (6)
    STEP (-4)
    TURN TO THE LEFT
    STEP (3)
END
```

Challenge:



Example solution:



Challenge:



Example solution (in text format)::

```
BEGIN
    STEP (2)
    STEP (-4)
    STEP (2)
    TURN TO THE LEFT
    STEP (2)
END
```

Module II
Europe 2

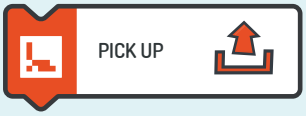


Scottie begins to interact with his surroundings.

Pick all snails up from the ground and take them to the running track.

Module introduction: picking up objects from the game-board

To pick up objects located on the game-board, use the **PICK UP** command.



Use this command in programs, e.g. if you want to pick up snails.

Note!

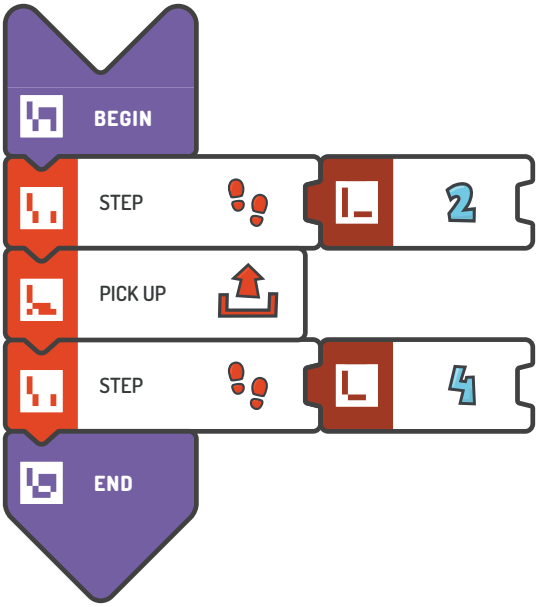
Using the **PICK UP** command on an empty square is treated as an error.

There is no need to use the **PLACE** tile in Module II.

Challenge **(remember that Scottie should finish on the running track)**:



Example solution:



Challenge **(remember that Scottie should finish on the running track)**:



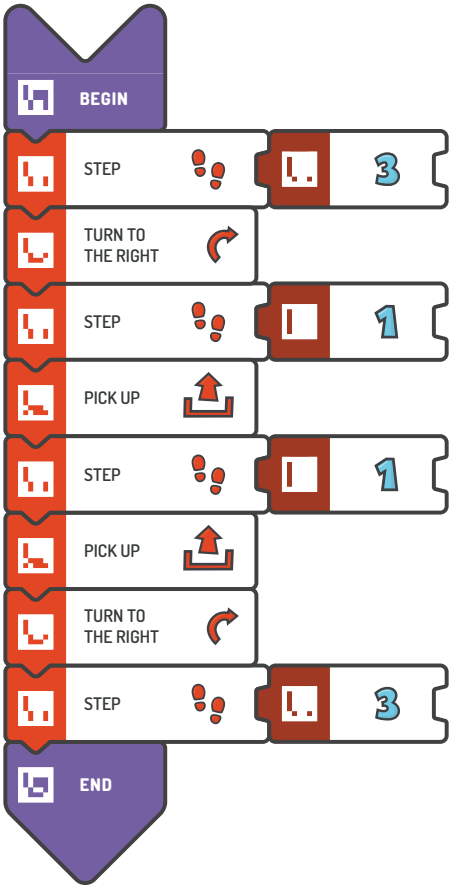
Example solution (in text format)::

```
BEGIN
    STEP (2)
    PICK UP
    STEP (4)
END
```

Challenge **(remember that Scottie should finish on the running track)**:



Example solution:



Challenge **(remember that Scottie should finish on the running track)**:



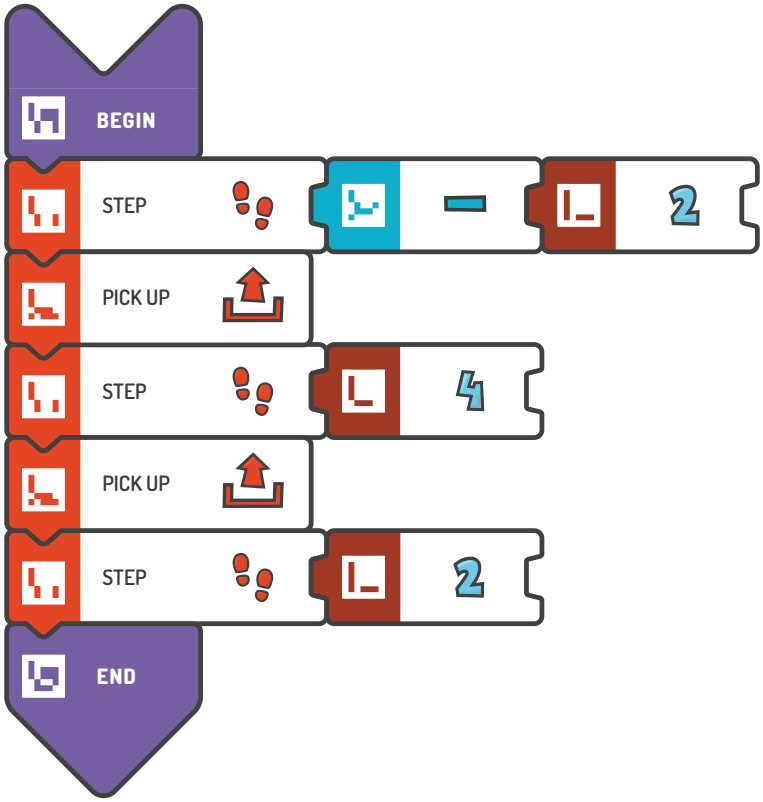
Example solution (in text format)::

```
BEGIN
    STEP (3)
    TURN TO THE RIGHT
    STEP (1)
    PICK UP
    STEP (1)
    PICK UP
    TURN TO THE LEFT
    STEP (3)
END
```

Challenge **(remember that Scottie should finish on the running track)**:



Example solution:



Challenge **(remember that Scottie should finish on the running track)**:



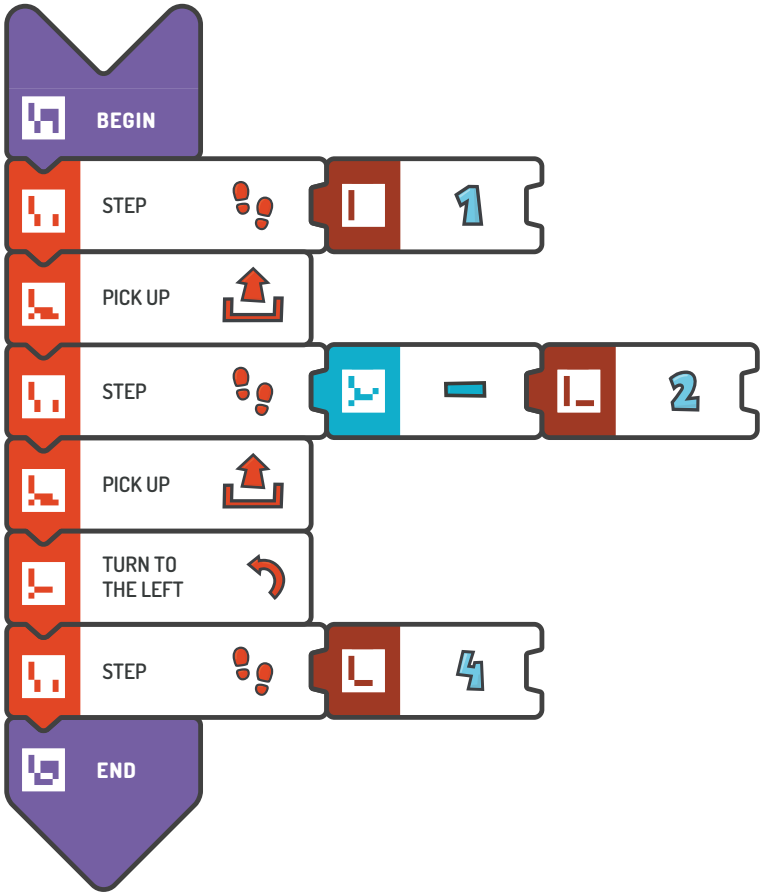
Example solution (in text format)::

```
BEGIN
    STEP (-2)
    PICK UP
    STEP (4)
    PICK UP
    STEP (2)
END
```

Challenge **(remember that Scottie should finish on the running track)**:



Example solution:



Challenge **(remember that Scottie should finish on the running track)**:

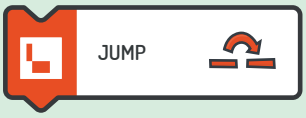


Example solution (in text format)::

```
BEGIN
    STEP (1)
    PICK UP
    STEP (-2)
    PICK UP
    TURN TO THE LEFT
    STEP (4)
END
```

Introduction: jumping

Sometimes Scottie may approach small obstacles on his way. To jump over them, use the red **JUMP** tile. Scottie may only jump forward.

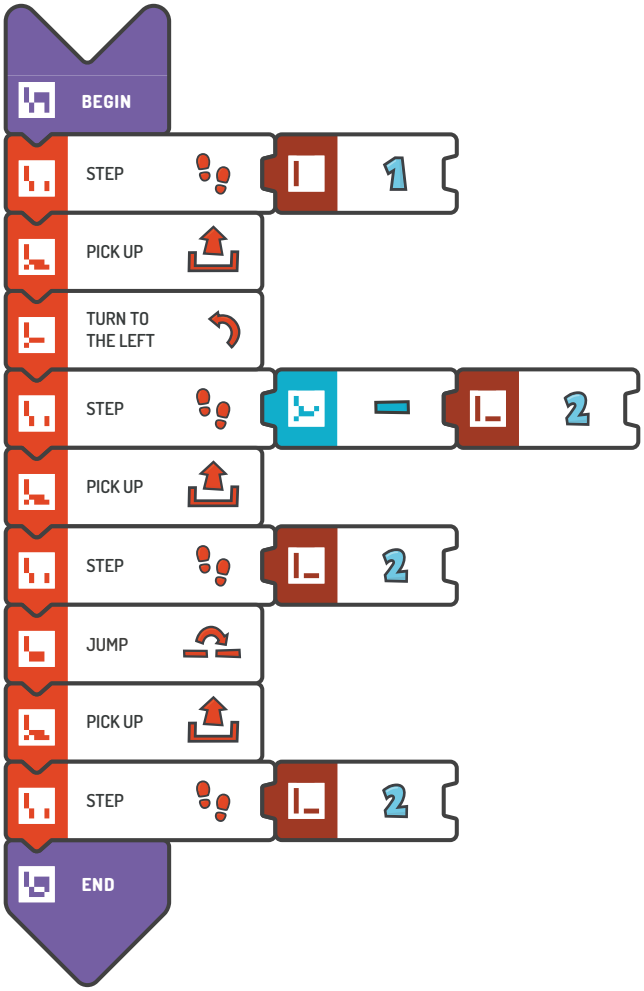


When jumping, Scottie will move from square one to square three, jumping over square two on his way.

Challenge **(remember that Scottie should finish on the running track)**:



Example solution:



Challenge **(remember that Scottie should finish on the running track)**:



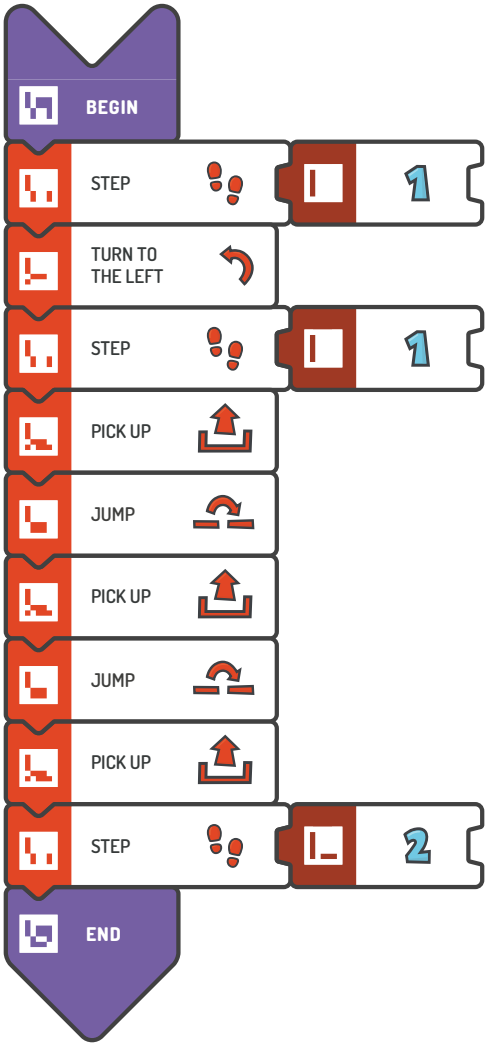
Example solution (in text format)::

BEGIN
STEP (1)
PICK UP
TURN TO THE LEFT
STEP (-2)
PICK UP
STEP (2)
JUMP
PICK UP
STEP (2)
END

Challenge **(remember that Scottie should finish on the running track)**:



Example solution:



Challenge **(remember that Scottie should finish on the running track)**:



Example solution (in text format)::

```
BEGIN
    STEP (1)
    TURN TO THE LEFT
    STEP (1)
    PICK UP
    JUMP
    PICK UP
    JUMP
    PICK UP
    STEP (2)
END
```

NOTE!

Players don't need to use the loop algorithm in this quest, even though it would make the solution much simpler.

During classes pupils often tend do act intuitively and use other tiles, e.g. **REPEAT**, despite the fact that this command is yet to be introduced in the subsequent module.

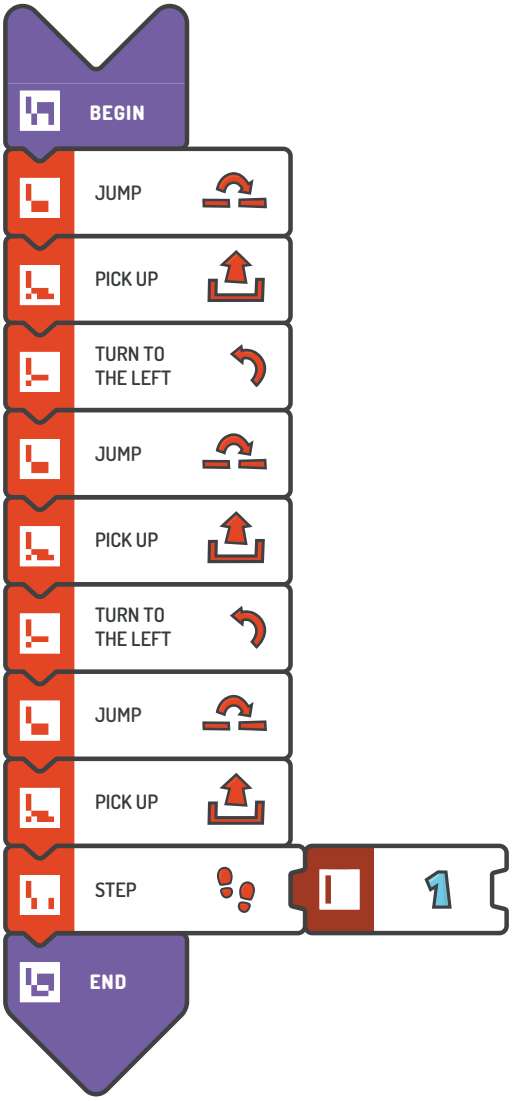


The **REPEAT** as well as the **END IF / REPEAT** tiles may be used even in Module I. The correct way a loop can be constructed is explained in the introduction to Module III.

Challenge **(remember that Scottie should finish on the running track)**:



Example solution:



Challenge **(remember that Scottie should finish on the running track)**:



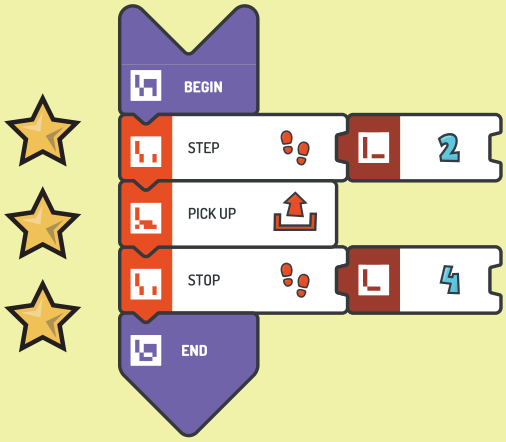
Example solution (in text format)::

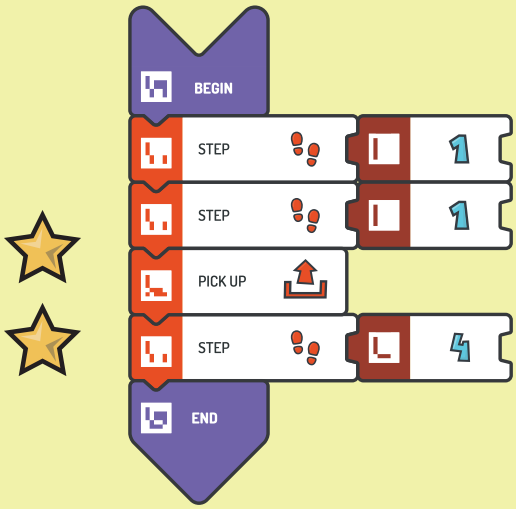
```
BEGIN
    JUMP
    PICK UP
    TURN TO THE LEFT
    JUMP
    PICK UP
    TURN TO THE LEFT
    JUMP
    PICK UP
    STEP (1)
END
```

NOTE!

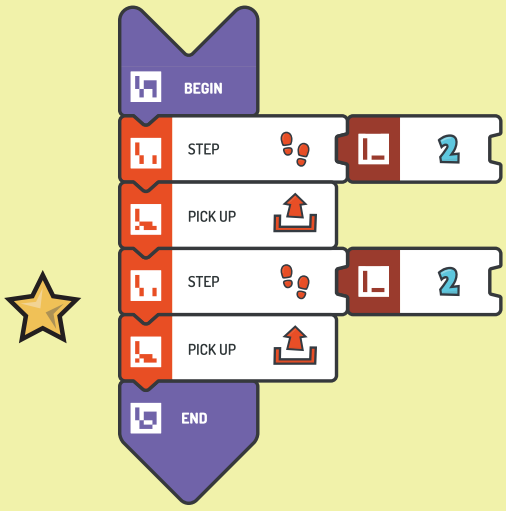
In this quest pupils are likely to struggle to find an optimal route for Scottie that will make the program as short as possible.

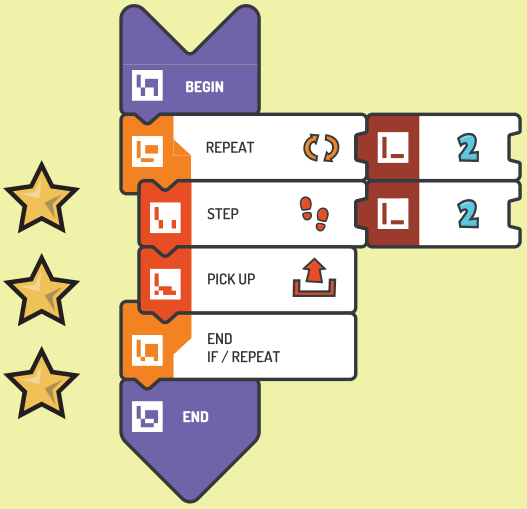
In Scottie Go! EDU players will receive stars a number of stars depending on the quality of the solution - three stars if the program is the best, two stars if the solution is a bit less efficient. One star means that the quest was solved, but the program was too long.





In some quests, if players want to be granted more than one star, they will need to use specific tiles, e.g. the **REPEAT** tile, in their solution.

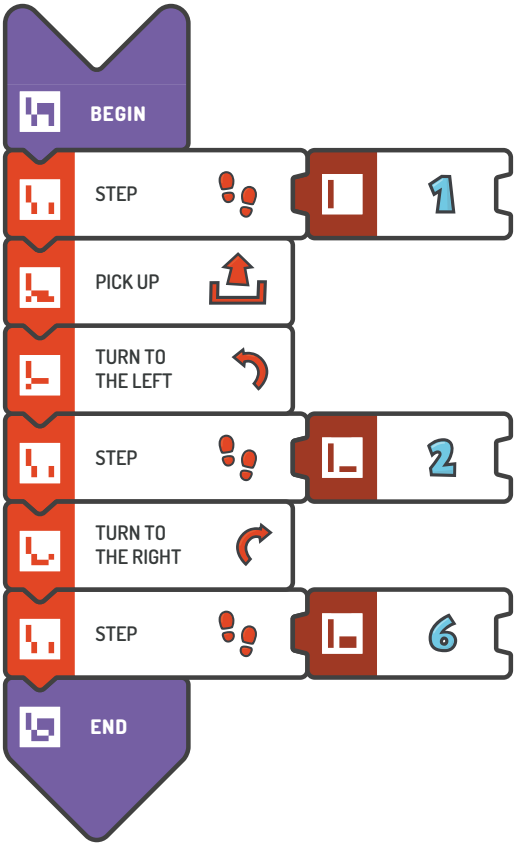




Challenge **(remember that Scottie should finish on the running track)**:



Example solution:



Challenge **(remember that Scottie should finish on the running track)**:



Example solution (in text format)::

```
BEGIN
    STEP (1)
    PICK UP
    TURN TO THE LEFT
    STEP (2)
    TURN TO THE RIGHT
    STEP (6)
END
```

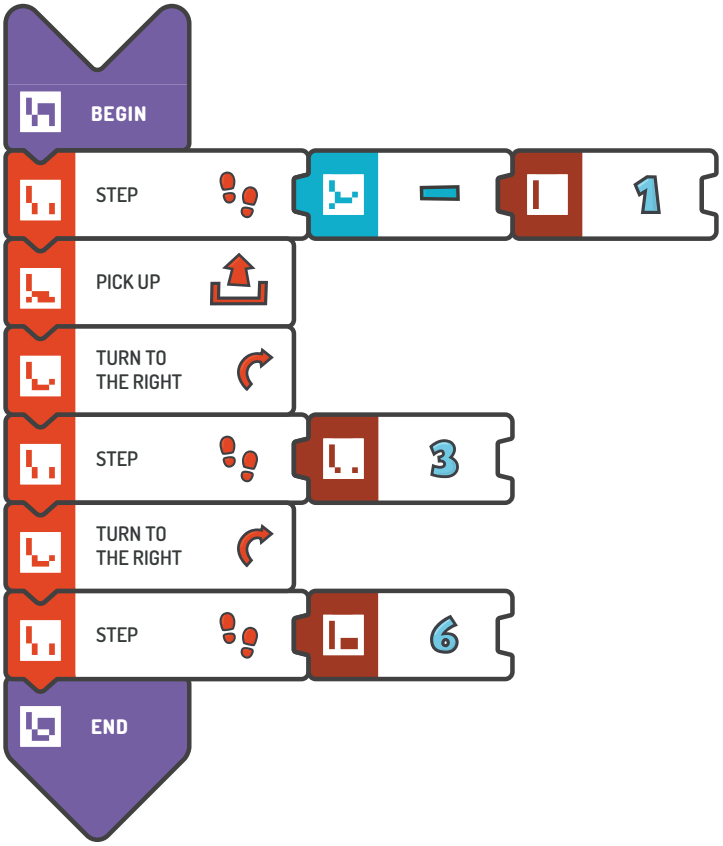
TIP

In this quest we recommend that you advise your pupils to look for an optimal route for Scottie that will make the program as short as possible.

Challenge **(remember that Scottie should finish on the running track)**:



Example solution:



Challenge **(remember that Scottie should finish on the running track)**:



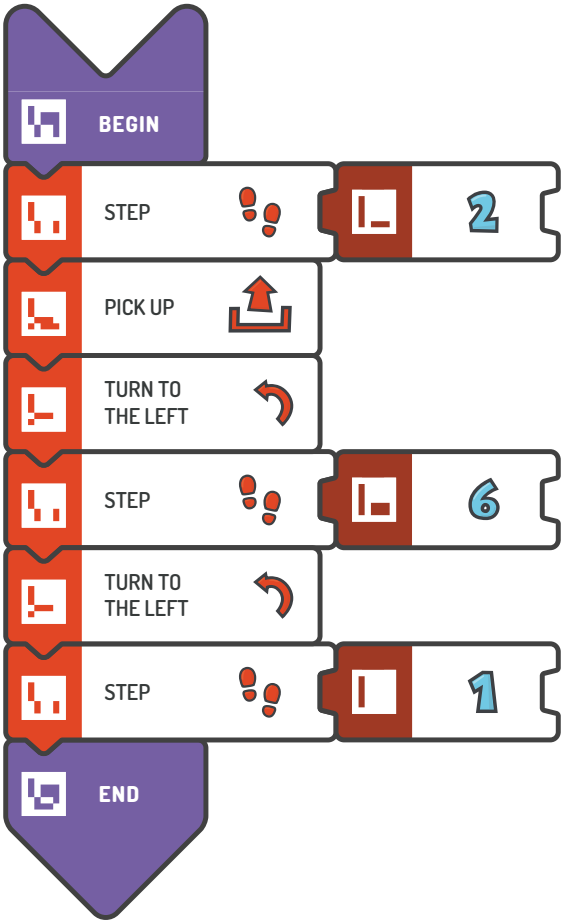
Example solution (in text format)::

```
BEGIN
    STEP (-1)
    PICK UP
    TURN TO THE RIGHT
    STEP (3)
    TURN TO THE RIGHT
    STEP (6)
END
```

Challenge **(remember that Scottie should finish on the running track)**:



Example solution:



Challenge **(remember that Scottie should finish on the running track)**:



Example solution (in text format)::

```
BEGIN
    STEP (2)
    PICK UP
    TURN TO THE LEFT
    STEP (6)
    TURN TO THE LEFT
    STEP (1)
END
```

Module III
South America



Repeating actions.

Pick up all the feathers from the game-board.

MODULE INTRODUCTION: the REPEAT loop

It is probable that after completing a few Module I quests some of your pupils will notice that Scottie has to perform some actions numerous times. Sometimes those operations have to be performed in exactly the same order. It would be convenient to find some way to automatise some of them. It would also teach students to optimise their work by finding the most efficient solutions. If you are looking for a simple way to repeat some operations many times, the **loops are just the tool for you and your pupils**.

The **REPEAT loop** is one of the most frequently used loops. In Scottie Go! you will find its simplest version. If you want to use a loop, you need to specify **which part of the program should be repeated,**



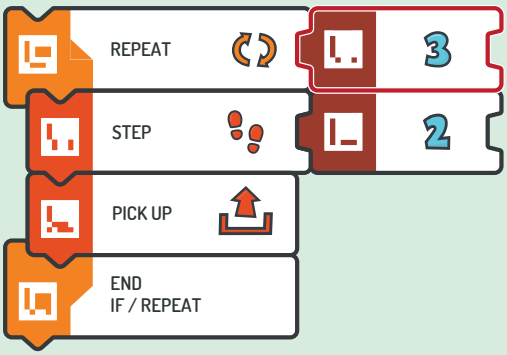
and then specify how many times it should be repeated. The program that specifies the commands to be repeated comprises two tiles: **REPEAT** and **END LOOP OR CONDITION**. Those tiles are equivalent to brackets that are used in popular programming languages..



INTRODUCTION: the first loop

A loop is one of the most basic and most frequently used notions of programming. Mastering loops is also a key algorithmic competence. If not for the loops, numerous digital devices couldn't function effectively and the programs would be too long which would make it much less effective. A loop allows you to repeat a sequence of commands until the desired goal is achieved.

In this quest your pupils need to remember to use the **REPEAT** loop. A loop must always be completed with an **END IF / REPEAT** tile. The tile highlighted in red on the image below should specify the number of repetitions for the sequence of commands in the loop:



The loop defined with the tiles above may be transcribed in text format in the following way:

```
REPEAT (3) {
        STEP (2)
        PICK UP
}
```

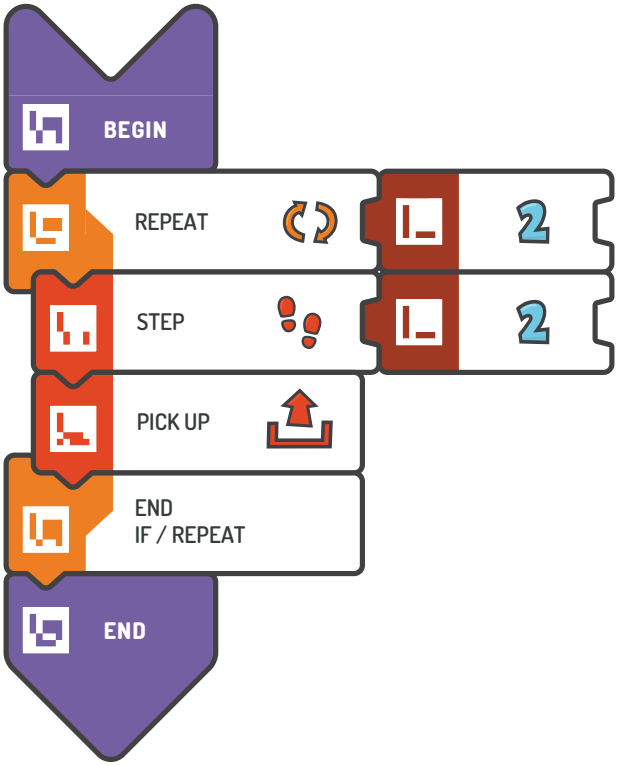
- where "**REPEAT** (number of repetitions)**{**" marks the place where the **REPEAT** tile with the number of repetitions was used.

- where "**}**" marks the place where the **END IF / REPEAT** tile was used.

Challenge:



Example solution:



Challenge:



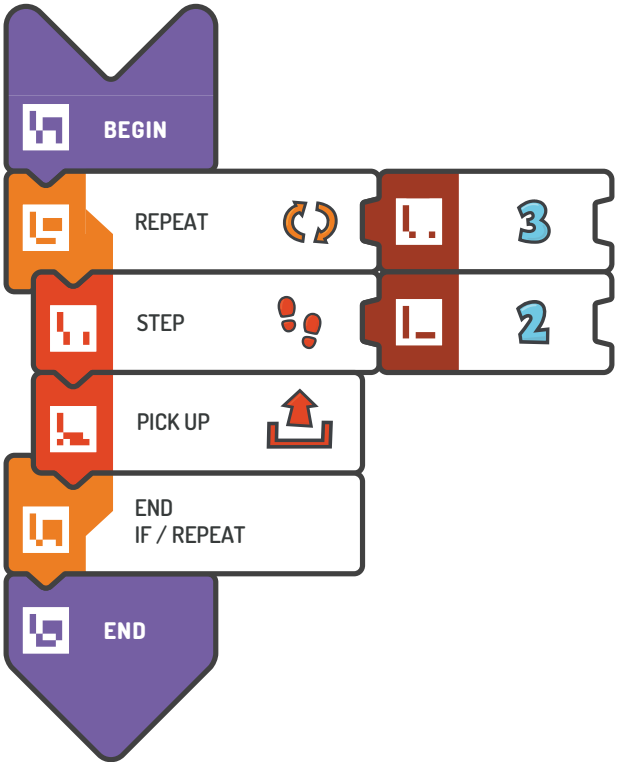
Example solution (in text format)::

```
BEGIN
    REPEAT (2) {
        STEP (2)
        PICK UP
    }
END
```

Challenge:



Example solution:



Challenge:



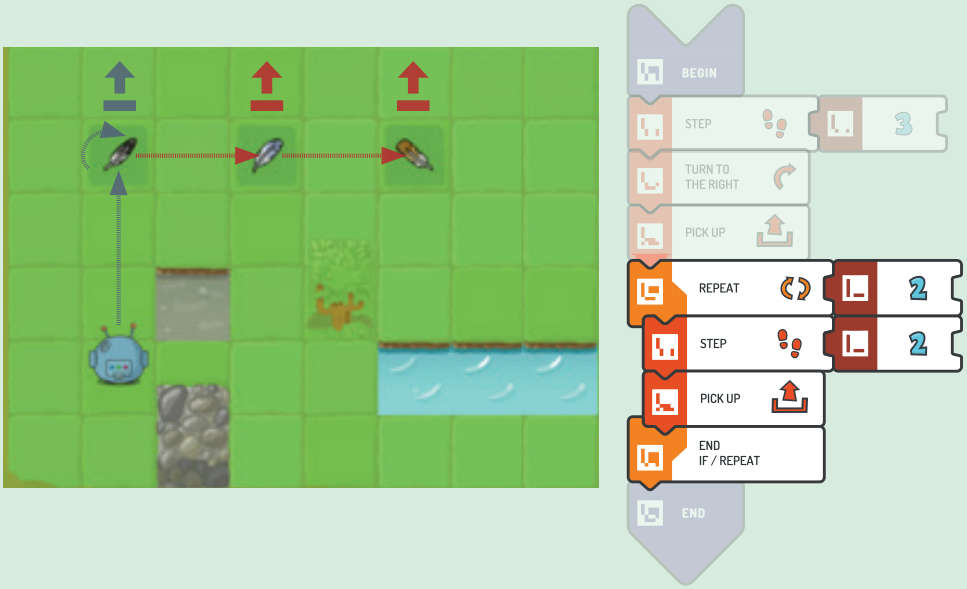
Example solution (in text format)::

```
BEGIN
    REPEAT (3) {
        STEP (2)
        PICK UP
    }
END
```

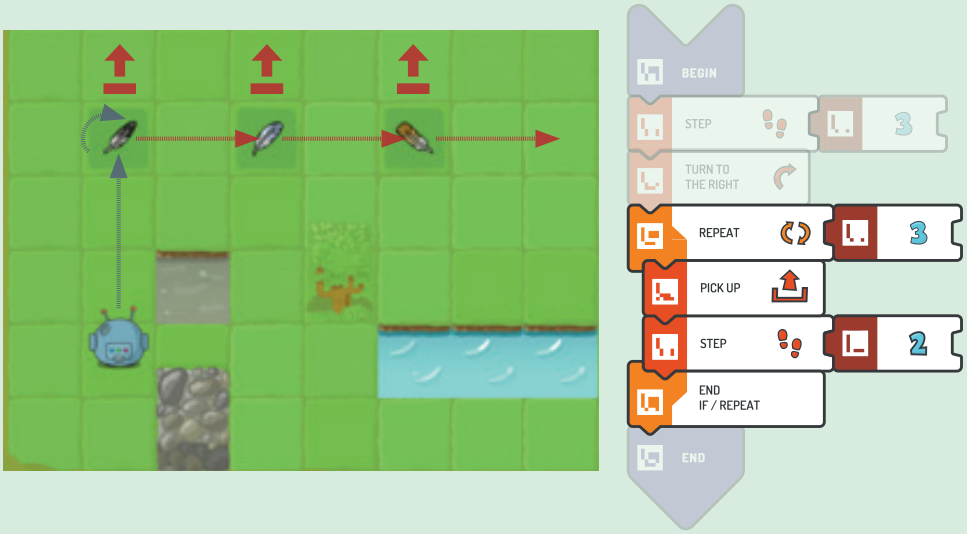
INTRODUCTION:
working inside a loop and outside of a loop

In some cases certain commands in a program will need to be repeated, while others will not! Tell your pupils to place the commands they don't want to be repeated outside of a loop! What is extremely important in loops is the order of events that they need to imagine before they begin writing this part of the program. The order of operations will influence Scottie's behaviour.

In the example below, if you pick up the first feather outside of a loop, the loop will require two repetitions.

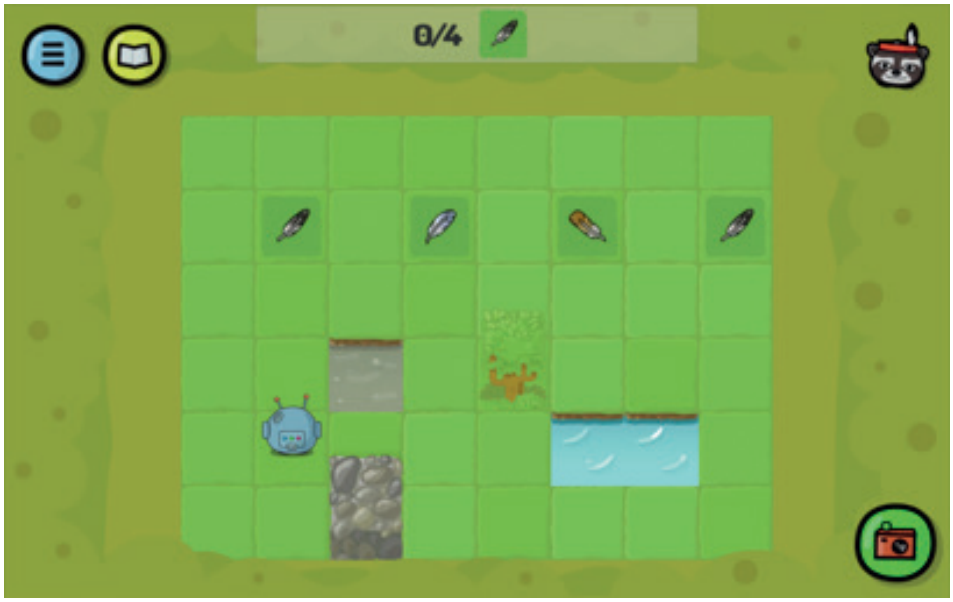


If you pick up the first feather inside a loop, the loop will have to be repeated 3 times.

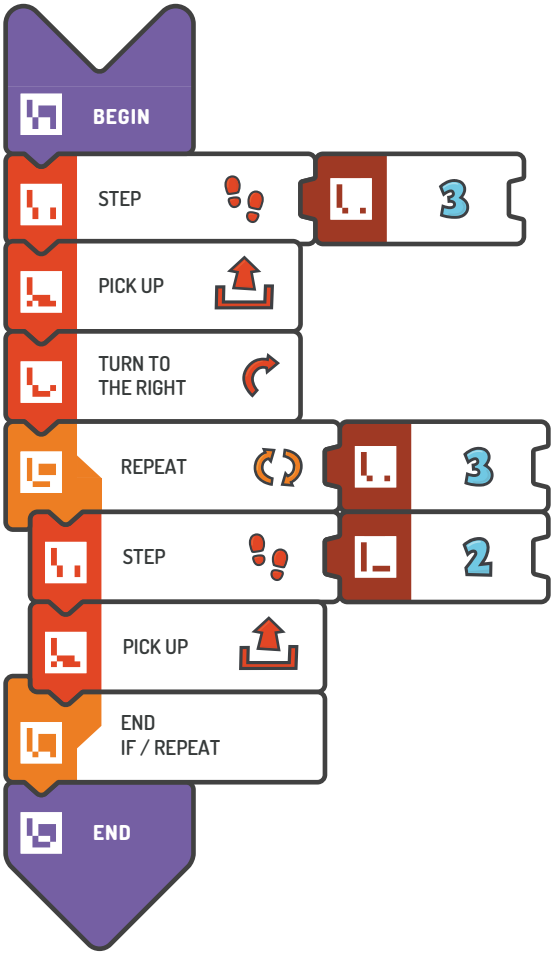


Both loops include two commands. Notice which of the two solutions has fewer lines of code?

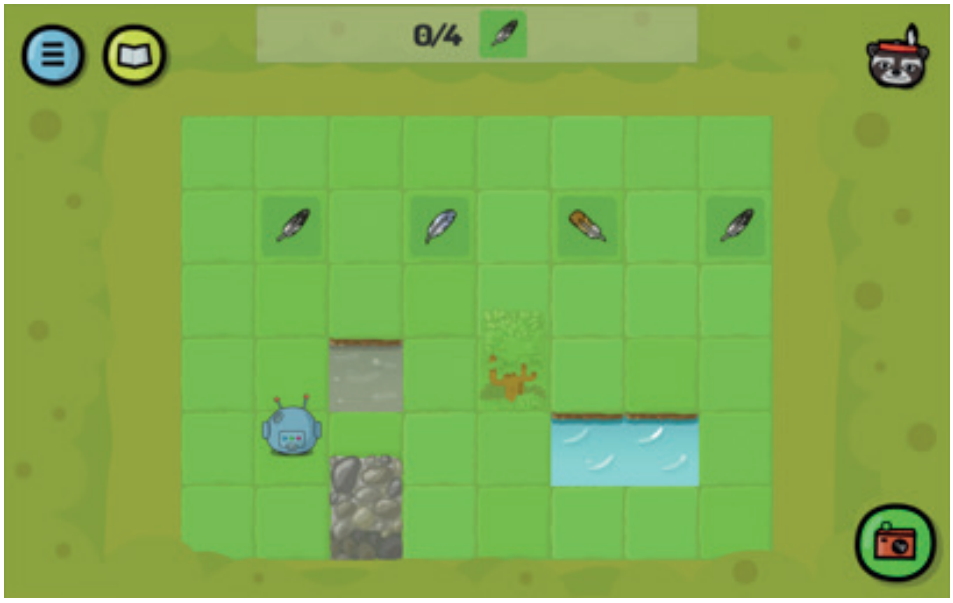
Challenge:



Example solution:



Challenge:

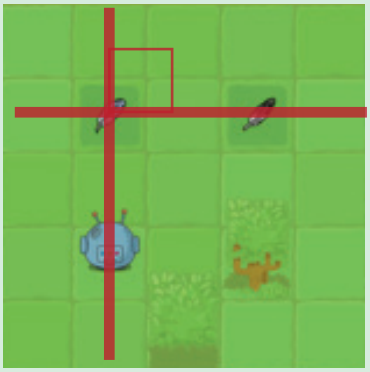


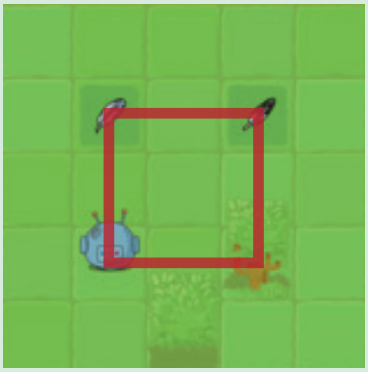
Example solution (in text format)::

```
BEGIN
    STEP (3)
    PICK UP
    TURN TO THE RIGHT
    REPEAT (3) {
        STEP (2)
        PICK UP
    }
END
```

INTRODUCTION:
Scottie GO! EDU as a tool for introducing notions in maths and physics

In this quest the route that Scottie needs to negotiate while picking up the feathers is part of a square. Scottie moves along the corners of this square, in which the feathers are located. It's a great chance to BEGIN talking to your pupils about right angles, parallel and perpendicular lines.





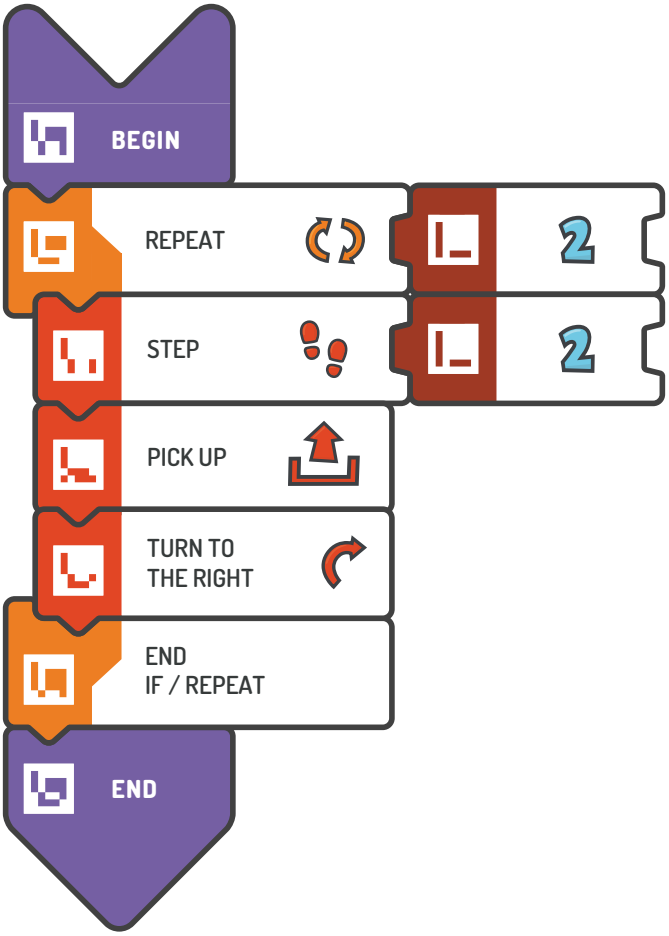
Scottie Go! EDU is played on a rectangular grid that constitutes a natural tool for explaining the notions of vectors, Cartesian coordinate system and resultant forces.



Challenge:



Example solution:



Challenge:



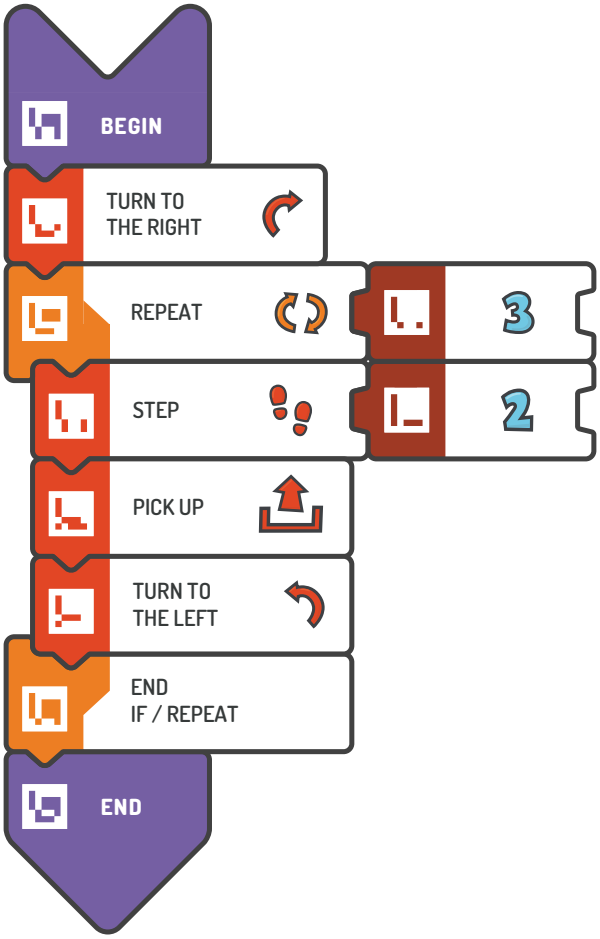
Example solution (in text format)::

```
BEGIN
    REPEAT (2) {
        STEP (2)
        PICK UP
        TURN TO THE RIGHT
    }
END
```

Challenge:



Example solution:



Challenge:



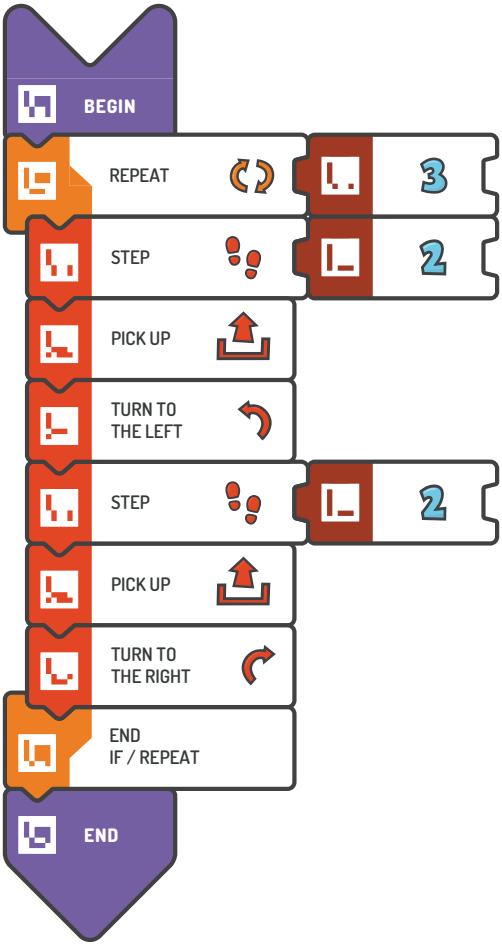
Example solution (in text format)::

```
BEGIN
    TURN TO THE RIGHT
    REPEAT (3) {
        STEP (2)
        PICK UP
        TURN TO THE LEFT
    }
END
```

Challenge:



Example solution:



Challenge:

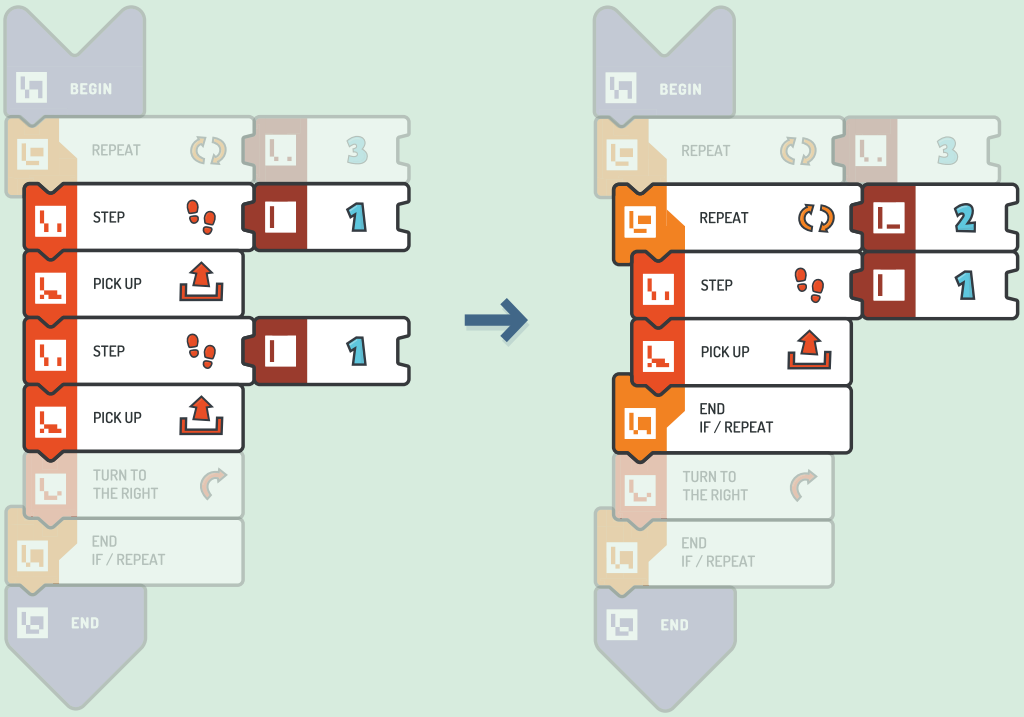


Example solution (in text format)::

```
BEGIN
    REPEAT (3) {
        STEP (2)
        PICK UP
        TURN TO THE LEFT
        STEP (2)
        PICK UP
        TURN TO THE RIGHT
    }
END
```

INTRODUCTION:
nested loops

To decrease the number of lines in a code, you can insert another, additional loop in an existing **REPEAT** loop. Such loops are called nested loops. Such algorithmic thinking may lead to further optimalization of code and influences the level of algorithm complexity. If your pupils can master nesting loops, they will develop not only their algorithmic intuition but also understand the need for improving the code to make it better and better.



Below you will find this in the text form:

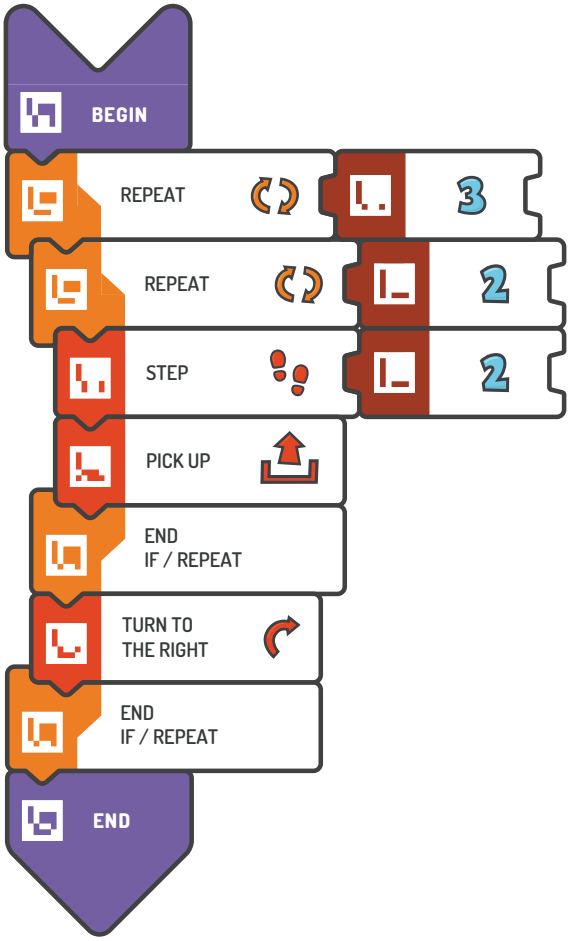
```
BEGIN
    REPEAT (3) {
        PICK UP
        STEP (1)
        PICK UP
        STEP (1)
        TURN TO THE RIGHT
    }
END
```

```
BEGIN
    REPEAT (3) {
        REPEAT (2) {
            PICK UP
            STEP (1)
        }
        TURN TO THE RIGHT
    }
END
```

Challenge:



Example solution:



Challenge:



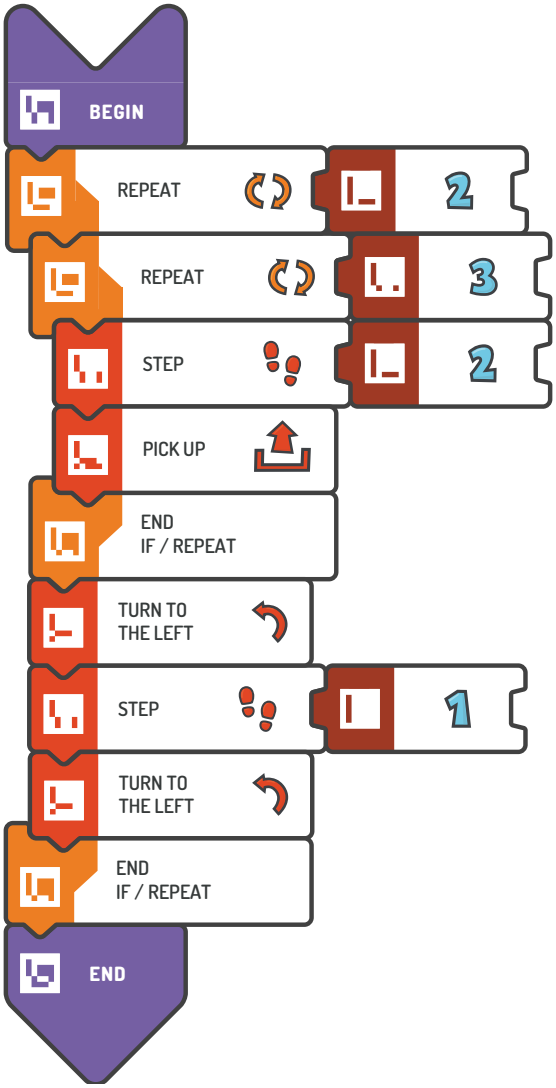
Example solution (in text format)::

```
BEGIN
    REPEAT (3) {
        REPEAT (2) {
            STEP (2)
            PICK UP
        }
        TURN TO THE RIGHT
    }
END
```

Challenge:



Example solution:



Challenge:



Example solution (in text format)::

```
BEGIN
    REPEAT (2) {
        REPEAT (3) {
            STEP (2)
            PICK UP
        }
        TURN TO THE LEFT
        STEP (1)
        TURN TO THE LEFT
    }
END
```

Module IV
North America



You will find a lot of gold in the nearby rocks.
Help Scottie collect the gold with the assistance of his animal friends - beavers.

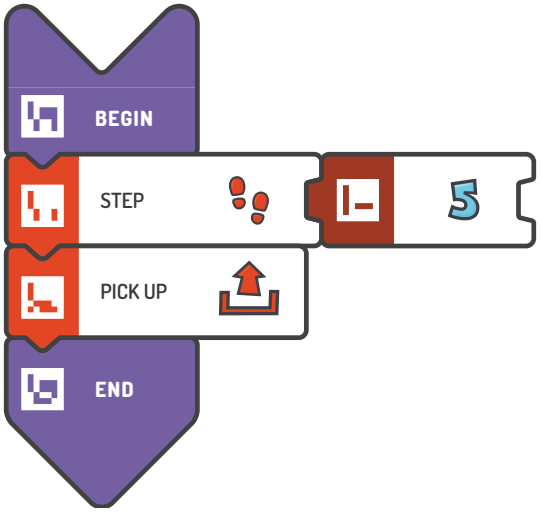
INTRODUCTION:
beaver as a bridge

To cross the river, Scottie needs to step on the Beaver's head. Your pupils will have to use the tiles they have already learned to achieve a new result.

Challenge:



Example solution:



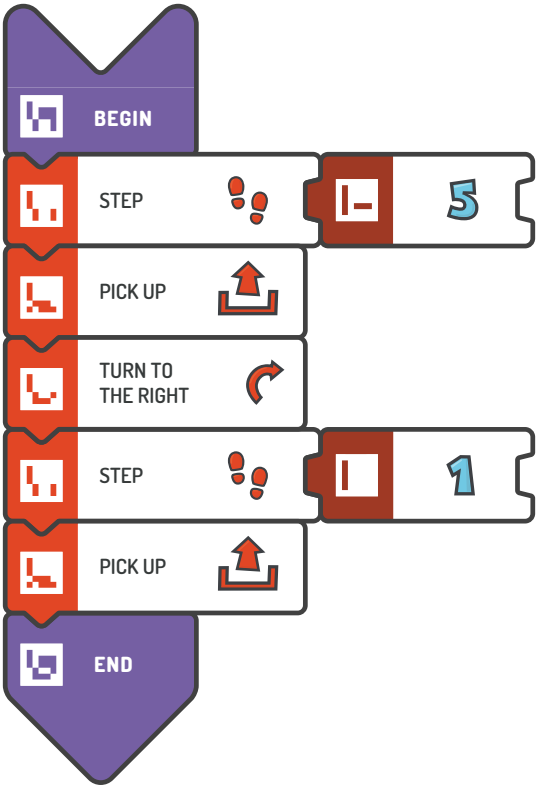
Example solution (in text format)::

```
BEGIN
    STEP (5)
    PICK UP
END
```

Challenge:



Example solution:



Challenge:



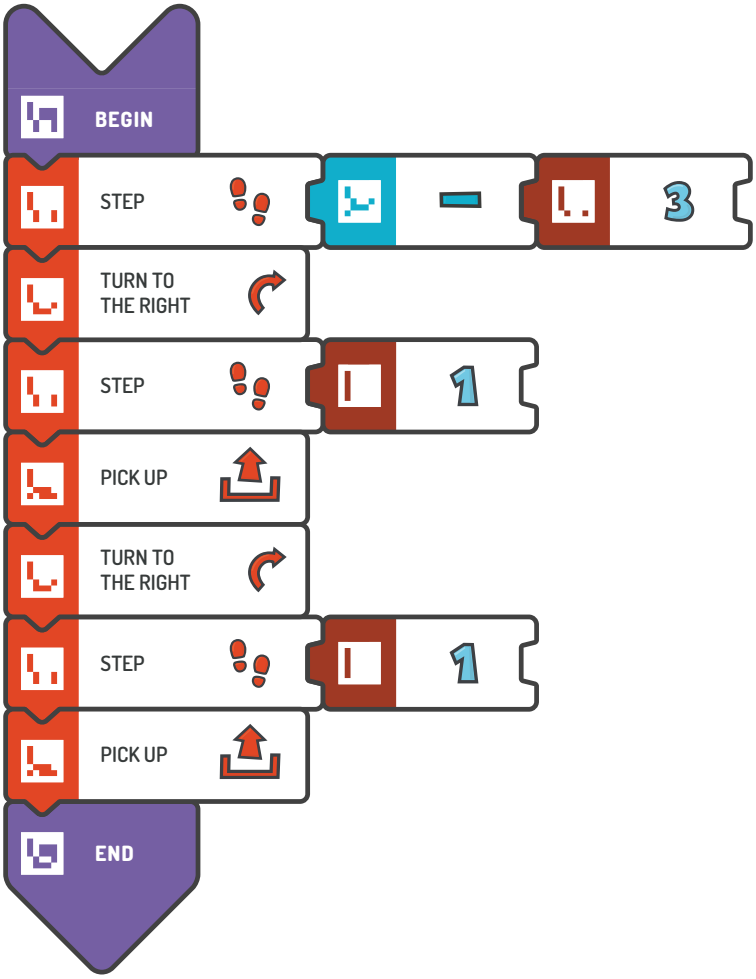
Example solution (in text format)::

```
BEGIN
    STEP (5)
    PICK UP
    TURN TO THE RIGHT
    STEP (1)
    PICK UP
END
```

Challenge:



Example solution:



Challenge:

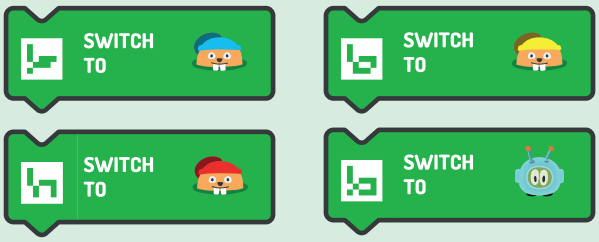


Example solution (in text format)::

```
BEGIN
    STEP (-3)
    TURN TO THE RIGHT
    STEP (1)
    PICK UP
    TURN TO THE RIGHT
    STEP (1)
    PICK UP
END
```

INTRODUCTION:
beaver in motion

You can also use the Beaver as a bridge or as a raft. You can program how the Beaver swims by using new tiles:



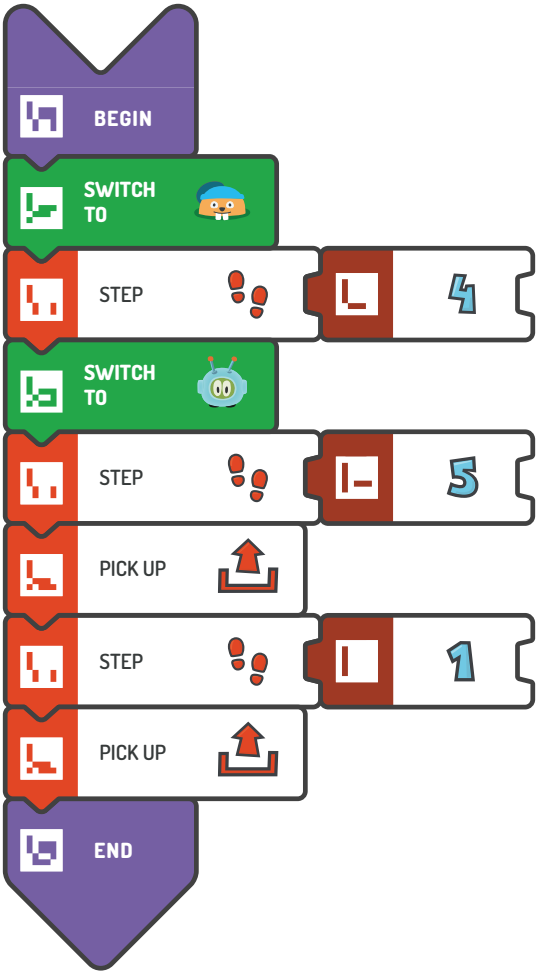
NOTE!

Each Beaver has a different hat (**yellow, blue or red**), so it is important to pick an appropriate tile from the box..

Challenge:



Example solution:



Challenge:



Example solution (in text format)::

```
BEGIN
    SWITCH TO  - BLUE BEAVER:
    STEP (4)
    SWITCH TO - SCOTTIE:
    STEP (5)
    PICK UP
    STEP (1)
    PICK UP
END
```

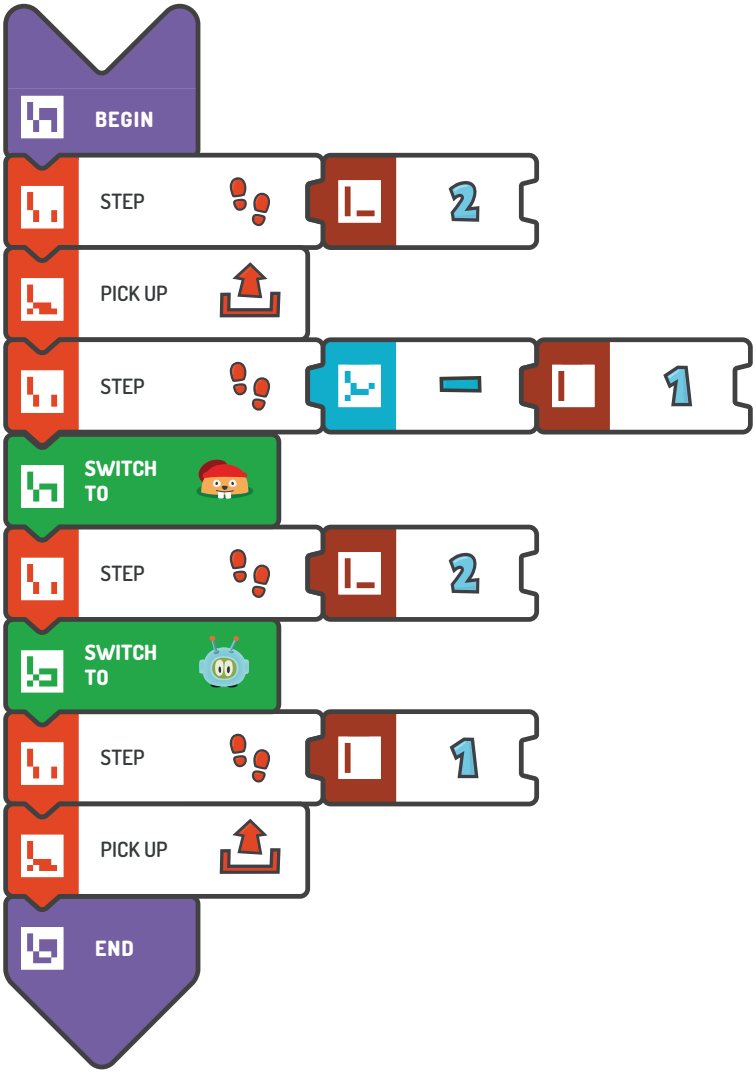
TIP

If you want Scottie to move to his destination quicker, get him to swim on a beaver. It is a great chance to look for an optimal route for Scottie so as to make the program as short as possible.

Challenge:



Example solution:



Challenge:



Example solution (in text format)::

```
BEGIN
    STEP (2)
    PICK UP
    STEP (-1)
    SWITCH TO - RED BEAVER:
    STEP (2)
    SWITCH TO - SCOTTIE:
    STEP (1)
    PICK UP
END
```

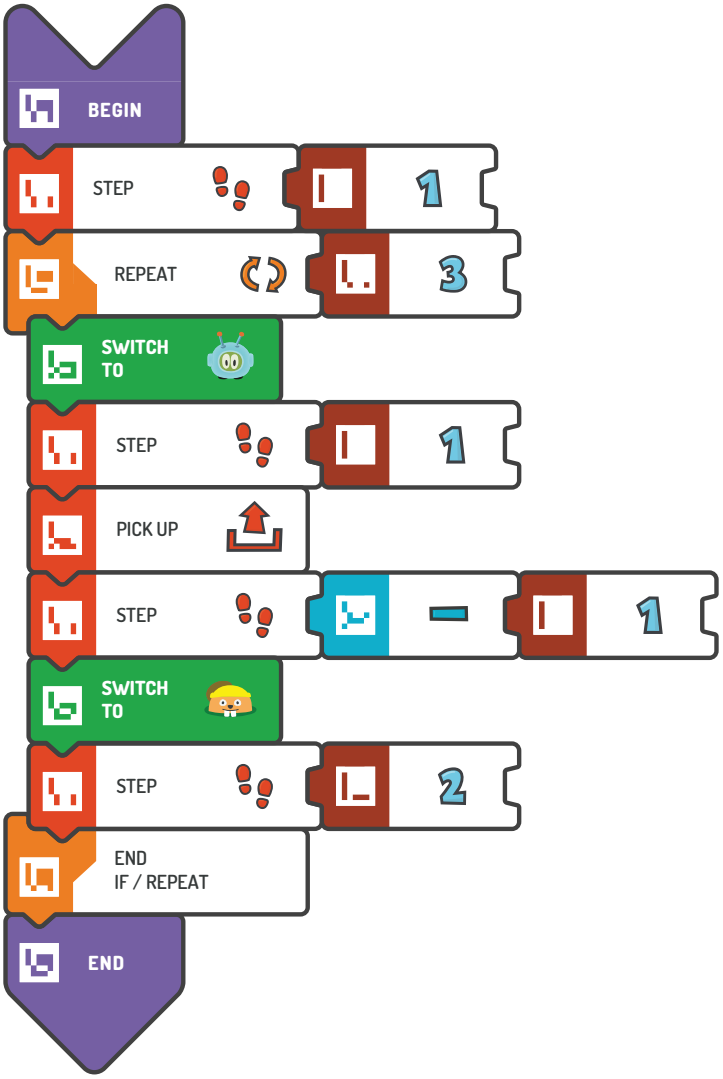
INTRODUCTION:
using REPEAT loop and the SWITCH TO tiles together

In this quest your pupils should use the **REPEAT** loop and the **SWITCH TO** tiles together to make in-game characters act in turn. It will help them consolidate the already known notions and develop their skills in combining indivifual bits of code.

Challenge:



Example solution:



Challenge:

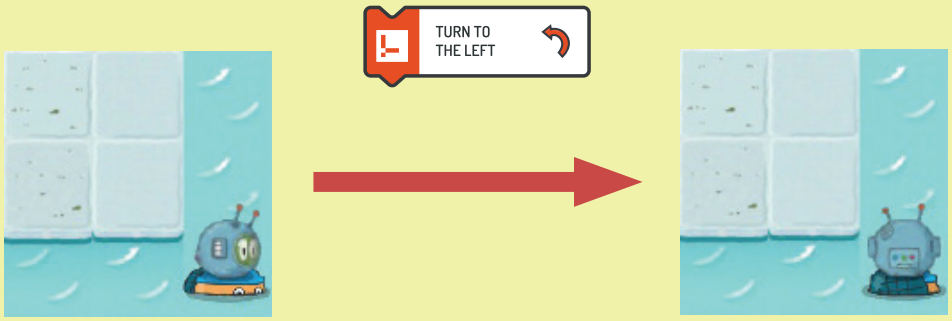


Example solution (in text format)::

```
BEGIN
    STEP (1)
    REPEAT (3) {
        SWITCH TO - SCOTTIE:
        STEP (1)
        PICK UP
        STEP (-1)
        SWITCH TO - YELLOW BEAVER:
        STEP (2)
    }
END
```

NOTE!
the Beaver makes a turn

The Beaver also can turn. If Scottie is swimming on the Beaver's head when the Beaver makes the turn, he will turn together with the Beaver.

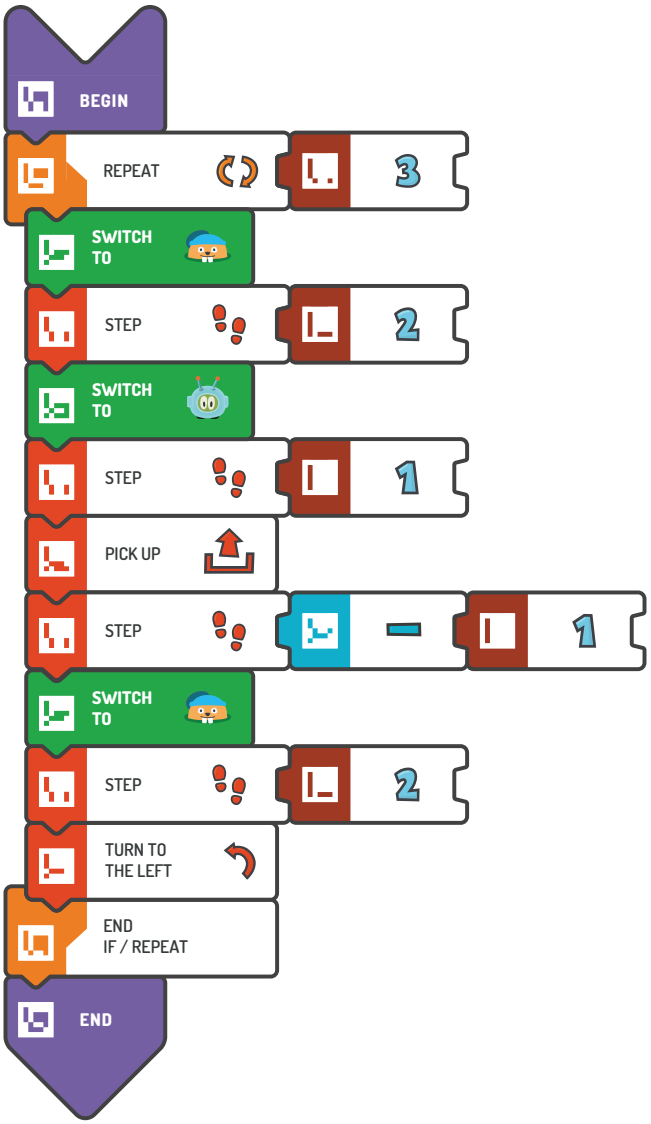


Now you can use the Beaver's assistance. Complete the subsequent quests using the **REPEAT** loop and make the Beaver turn.

Challenge:



Example solution:



Challenge:



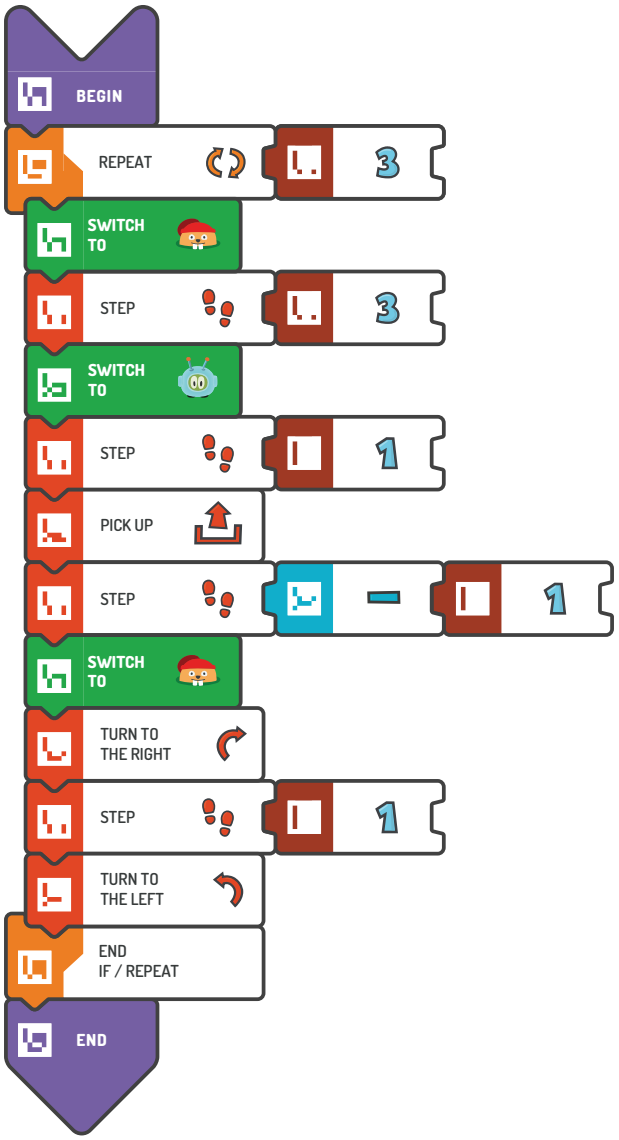
Example solution (in text format)::

```
BEGIN
    REPEAT (3) {
        SWITCH TO - BLUE BEAVER:
        STEP (2)
        SWITCH TO - SCOTTIE:
        STEP (1)
        PICK UP
        STEP (-1)
        SWITCH TO - BLUE BEAVER:
        STEP (2)
        TURN TO THE LEFT
    }
END
```

Challenge:



Example solution:



Challenge:



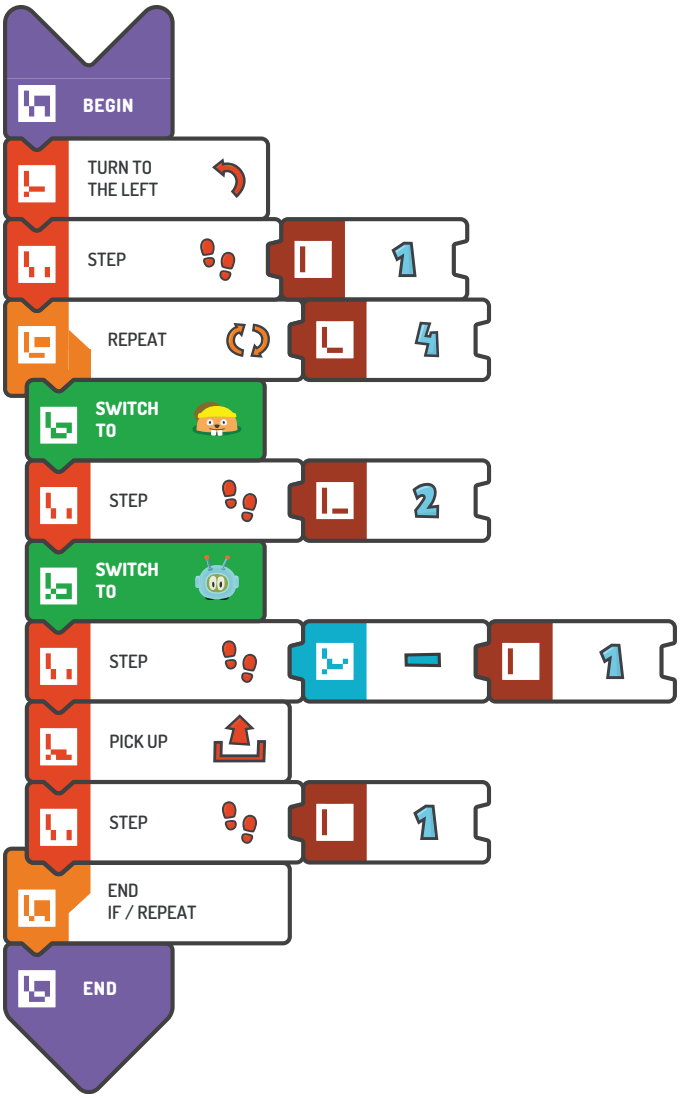
Example solution (in text format)::

```
BEGIN
    REPEAT (3) {
        SWITCH TO - RED BEAVER:
        STEP (3)
        SWITCH TO - SCOTTIE:
        STEP (1)
        PICK UP
        STEP (-1)
        SWITCH TO - RED BEAVER:
        TURN TO THE RIGHT
        STEP (1)
        TURN TO THE LEFT
    }
END
```

Challenge:



Example solution:



Challenge:



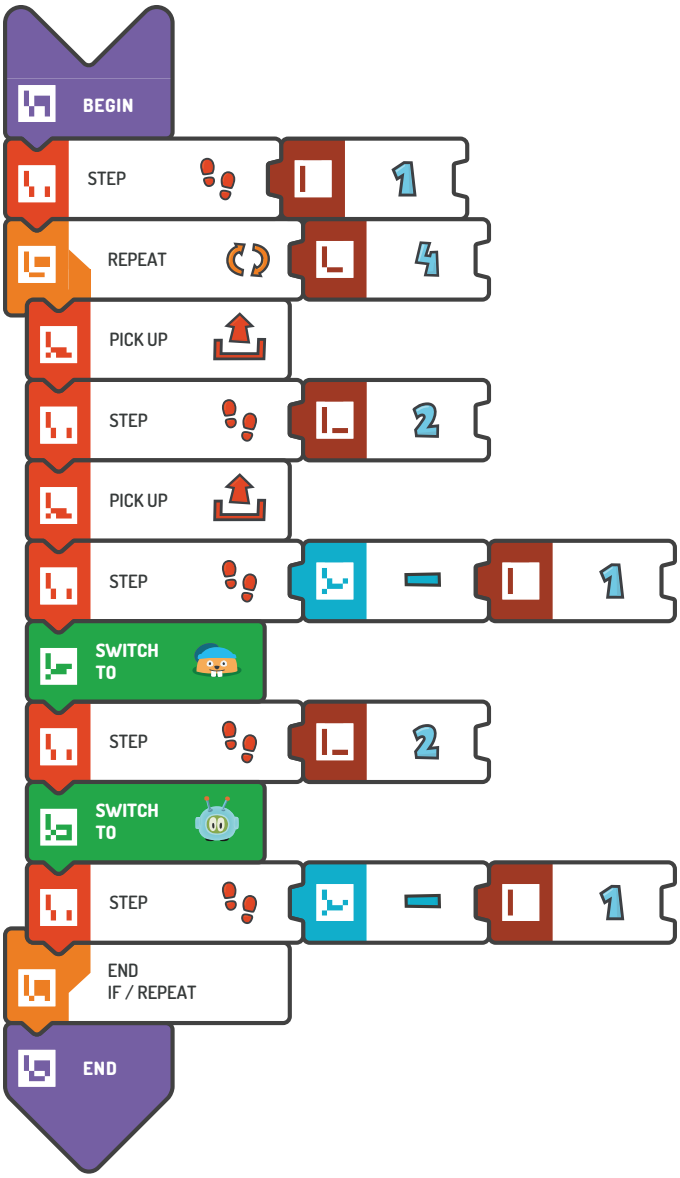
Example solution (in text format)::

```
BEGIN
    TURN TO THE LEFT
        STEP (1)
        REPEAT (4) {
            SWITCH TO - YELLOW BEAVER:
            STEP (2)
            SWITCH TO - SCOTTIE:
            STEP (-1)
            PICK UP
            STEP (1)
        }
END
```

Challenge:



Example solution:



Challenge:



Example solution (in text format)::

```
BEGIN
    STEP (1)
    REPEAT (4) {
        PICK UP
        STEP (2)
        PICK UP
        STEP (-1)
        SWITCH TO - BLUE BEAVER:
        STEP (2)
        SWITCH TO - SCOTTIE:
        STEP (-1)
    }
END
```

Module V
Antarctic



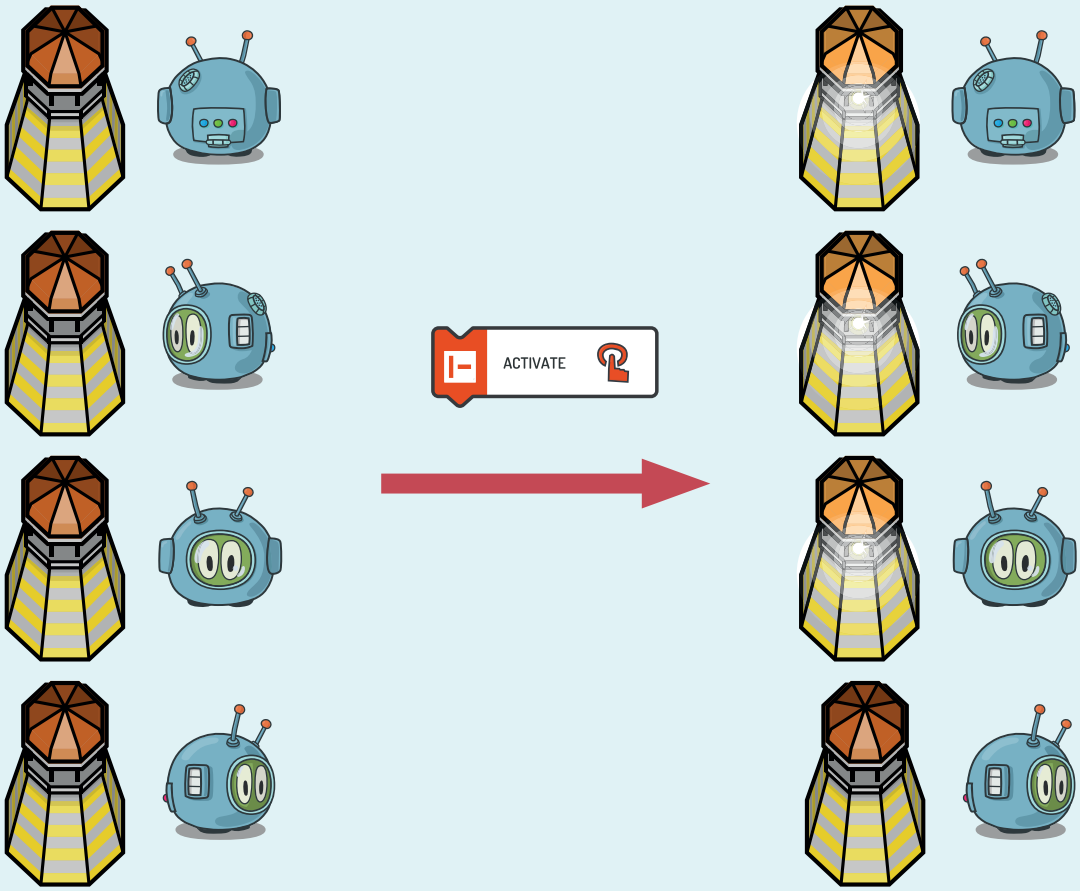
Scottie helps sailors. In this quest switch on the light in all the lighthouses on the peninsula to mark the harbour for the ships that are transporting spaceship parts.

INTRODUCTION TO MODULE 1/2: interaction with Scottie's surroundings - ACTIVATE and PLACE tiles

In this module Scottie continues his interactions with objects and he **begins to change his surroundings**. The **ACTIVATE** and **PLACE** tiles are introduced.



You can place the **ACTIVATE** tile in the program to make Scottie switch on the lighthouse next to which he is standing, provided he is not facing away from it.

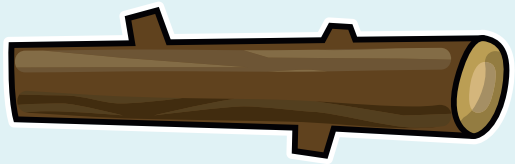


INTRODUCTION TO MODULE 2/2: interaction with Scottie's surroundings – ACTIVATE and PLACE tiles

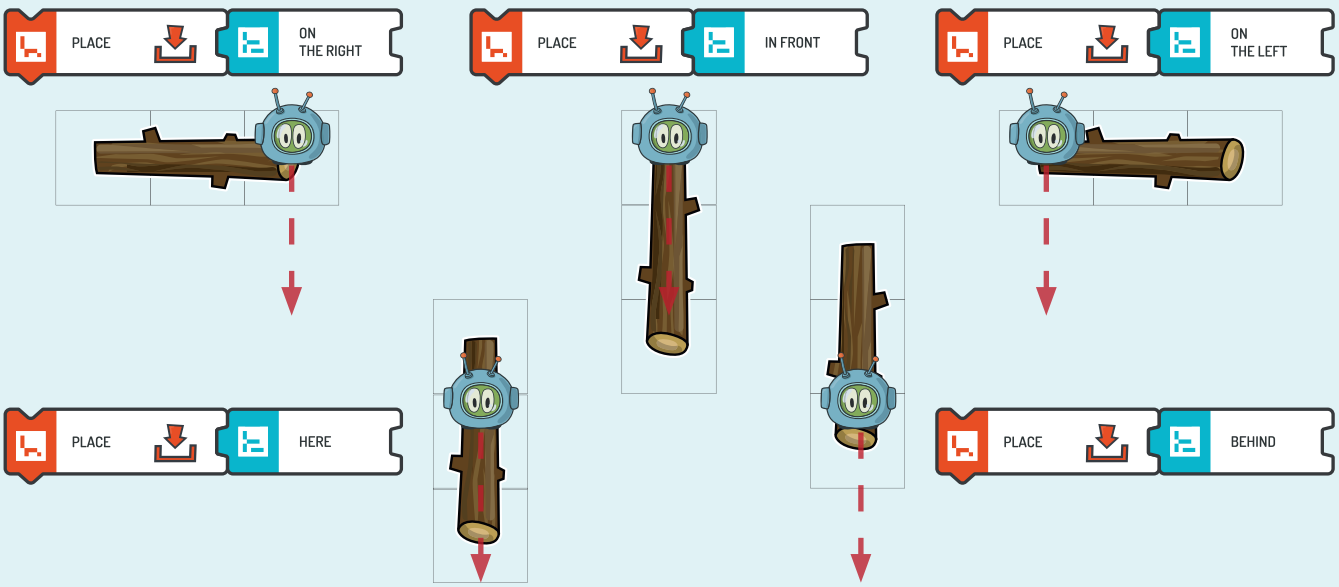
In this module Scottie continues his interactions with objects and he **begins to change his surroundings**. The **ACTIVATE** and **PLACE** tiles are introduced.



This module introduces also tree logs that Scottie can pick up (using the **PICK UP** tile).



Once he picked up a tree log, Scottie can place it on the game-board in five various positions, using the **HERE, IN FRONT, ON THE RIGHT, ON THE LEFT, BEHIND** tiles as parameters for the **PLACE** tile.

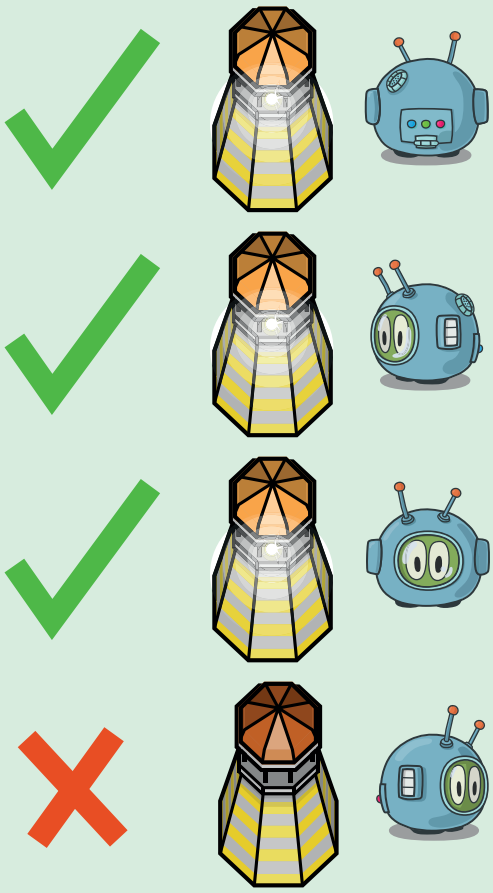


INTRODUCTION: the ACTIVATE tile

Switch on the lighthouses with the new **ACTIVATE** tile.



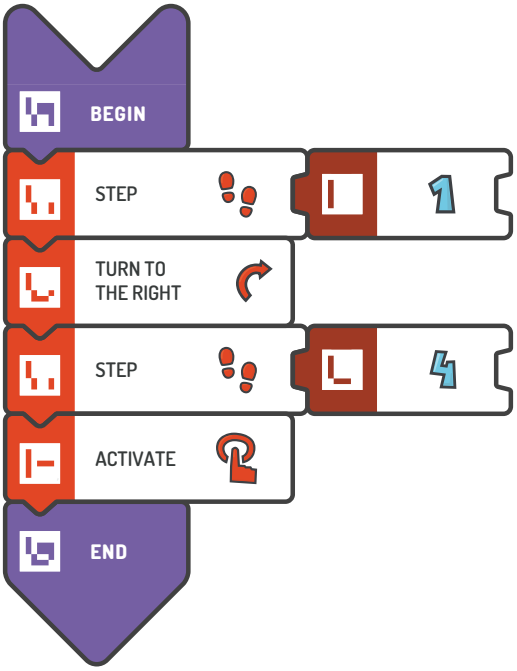
To switch on a lighthouse, Scottie must face it or stand next to it. He must not be facing away from a lighthouse, as the **ACTIVATE** tile will not work.



Challenge:



Example solution:



Challenge:

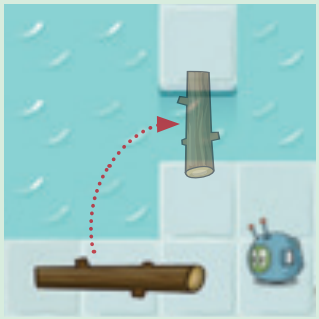


Example solution (in text format)::

```
BEGIN
    STEP (1)
    TURN TO THE RIGHT
    STEP (4)
    ACTIVATE
END
```

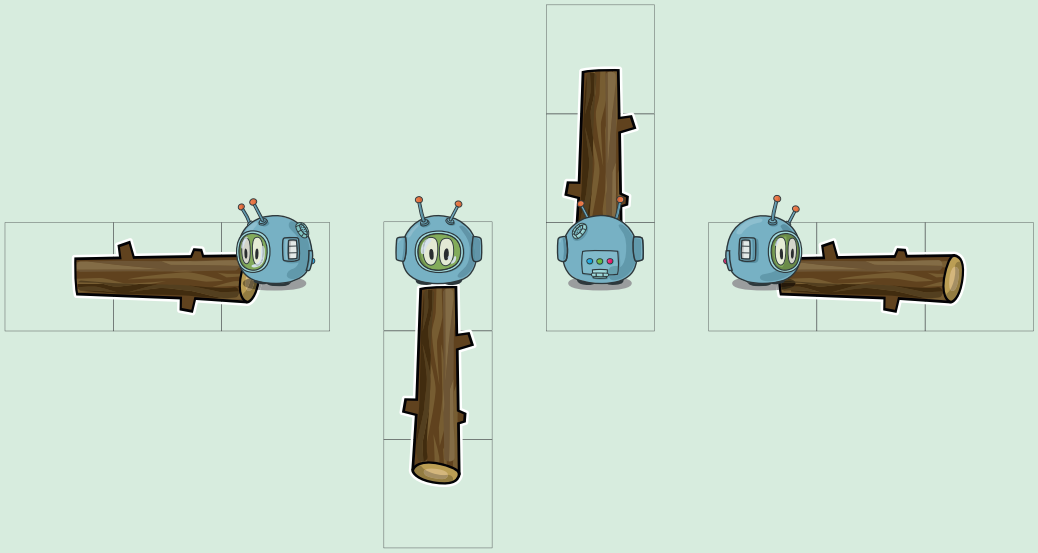
INTRODUCTION: the PLACE tile

To get Scottie across a ravine, pick up a tree log and drop them in place in correct position using the new **PLACE** tile.



To place a tree log over a ravine, Scottie must stand one square in front of the ravine. You must also select the appropriate parameter for the **PLACE** tile, e.g. **PLACE(IN FRONT)** if Scottie is facing the ravine.



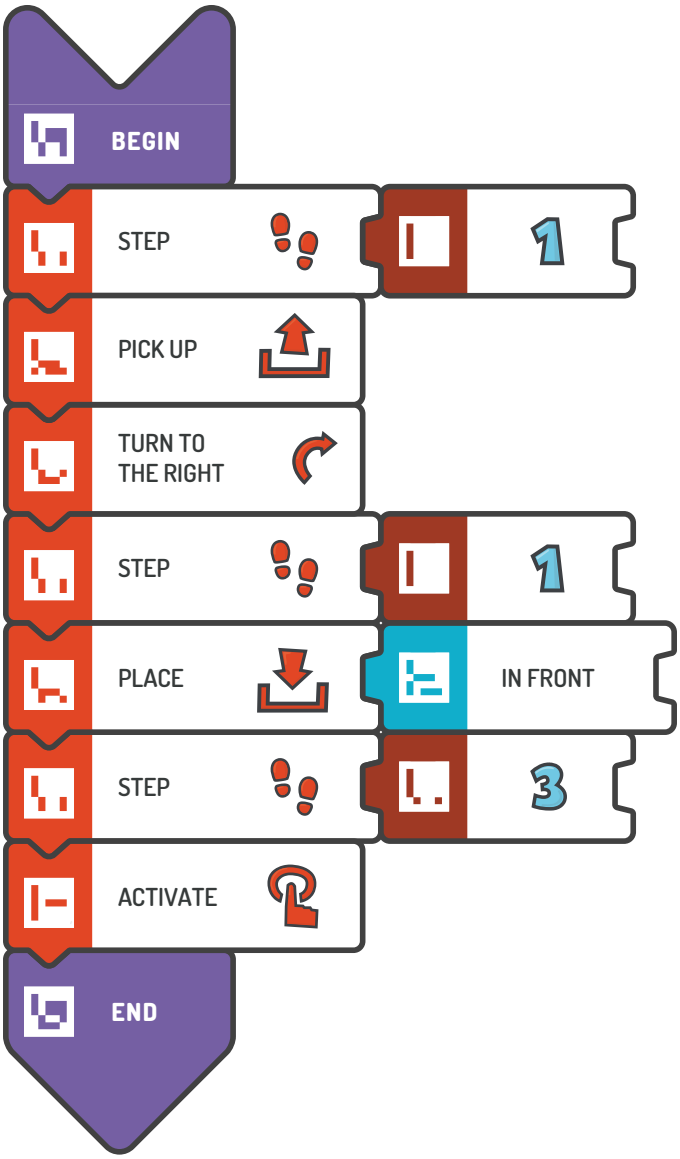


To pick up a tree log, Scottie must be standing on **any** of the three squares that the tree log is spanning.

Challenge:



Example solution:



Challenge:



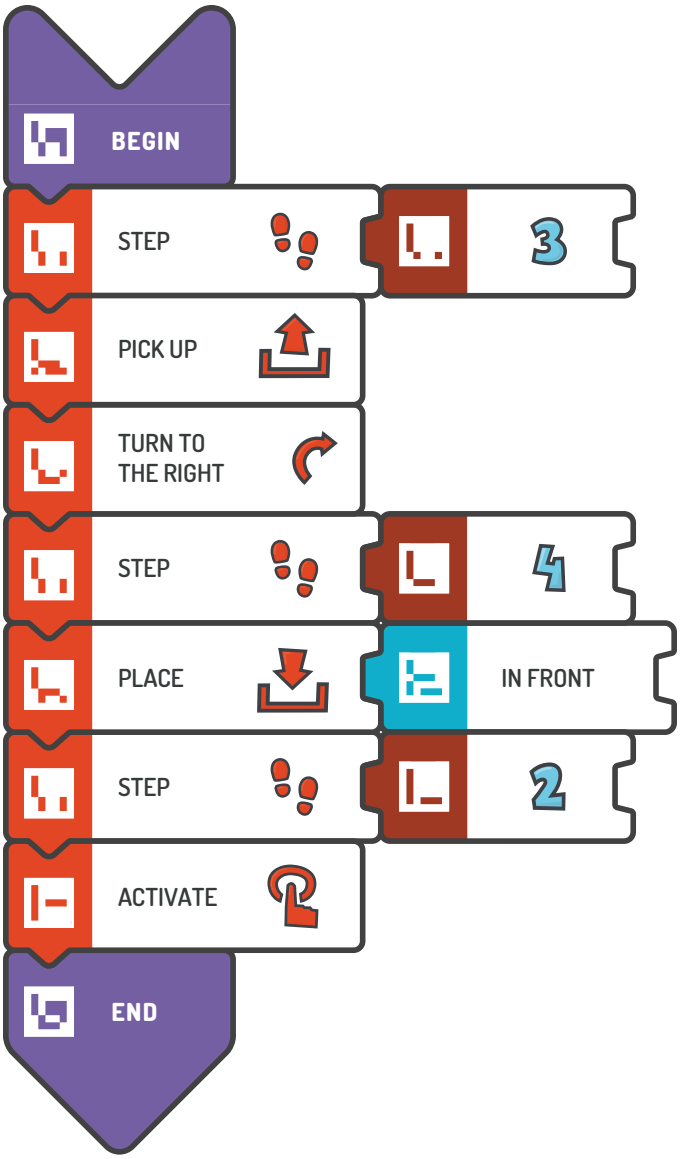
Example solution (in text format)::

```
BEGIN
    STEP (1)
    PICK UP
    TURN TO THE RIGHT
    STEP (1)
    PLACE (IN FRONT)
    STEP (3)
    ACTIVATE
END
```

Challenge:



Example solution:



Challenge:



Example solution (in text format)::

```
BEGIN
    STEP (3)
    PICK UP
    TURN TO THE RIGHT
    STEP (4)
    PLACE (IN FRONT)
    STEP (2)
    ACTIVATE
END
```

Challenge:



Example solution:



Challenge:



Example solution (in text format)::

```
BEGIN
    STEP (3)
    PICK UP
    TURN TO THE LEFT
    PLACE (IN FRONT)
    STEP (3)
    ACTIVATE
END
```

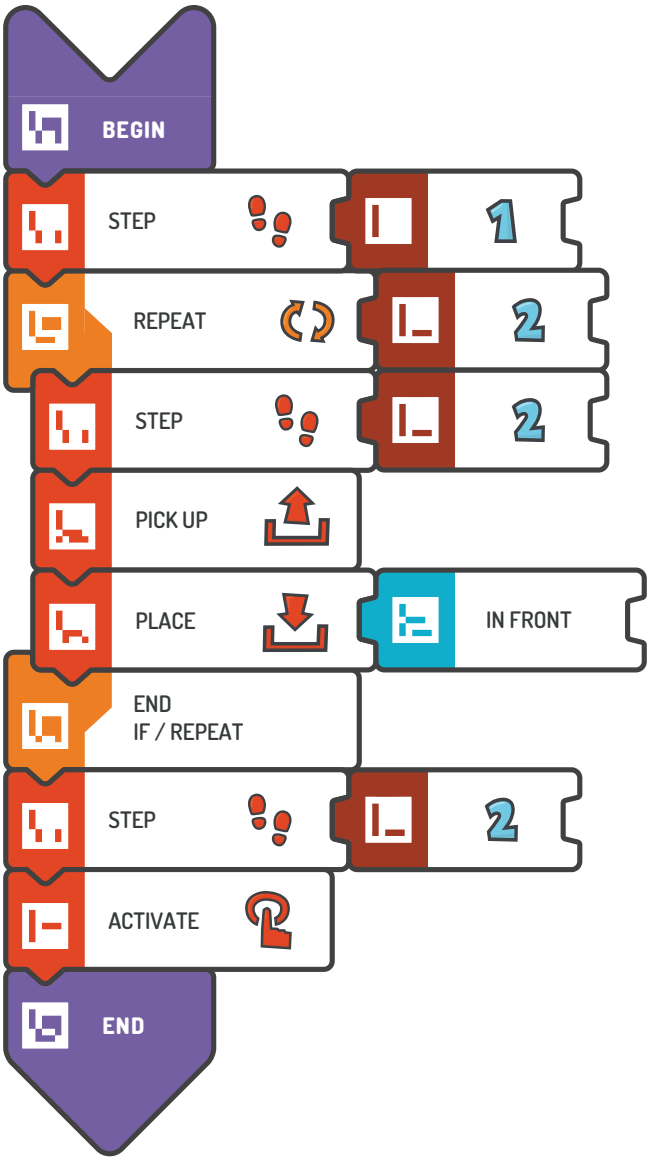
NOTE

In subsequent quests make sure your pupils use the **REPEAT** tile, create new loops and continue to optimise their programs.

Challenge:



Example solution:



Challenge:



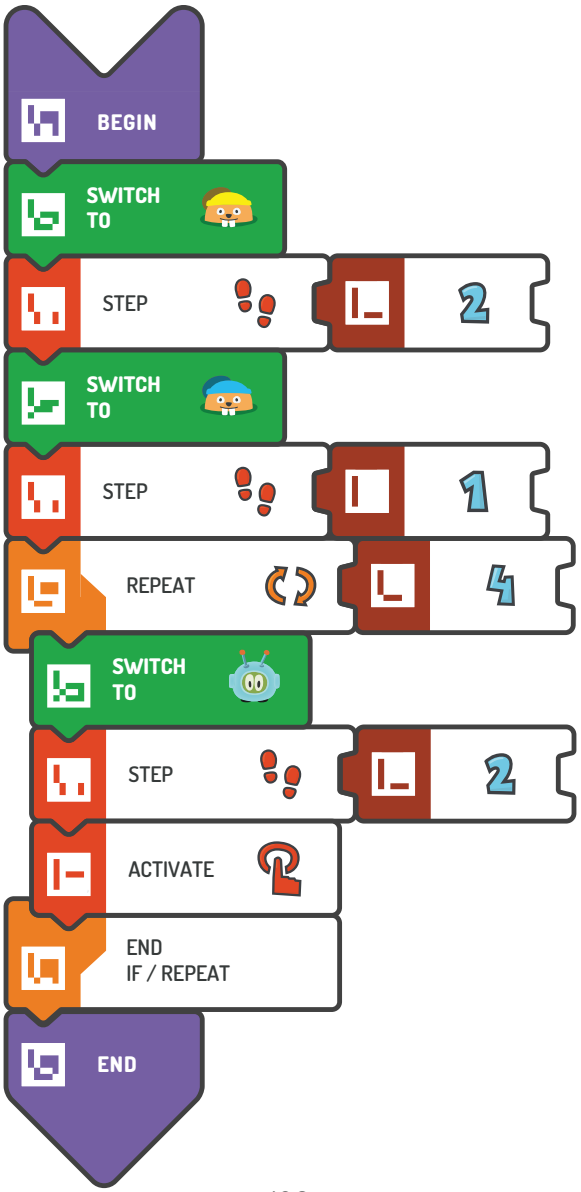
Example solution (in text format)::

```
BEGIN
    STEP (1)
    REPEAT (2) {
        STEP (2)
        PICK UP
        PLACE (IN FRONT)
    }
    STEP (2)
    ACTIVATE
END
```

Challenge:



Example solution:



Challenge:



Example solution (in text format)::

```
BEGIN
    SWITCH TO - YELLOW BEAVER:
    STEP (2)
    SWITCH TO - BLUE BEAVER:
    STEP (1)
    REPEAT (4) {
        SWITCH TO - SCOTTIE:
        STEP (2)
        ACTIVATE
    }
END
```

INTRODUCTION:
SWITCH TO the beavers and move ice floes

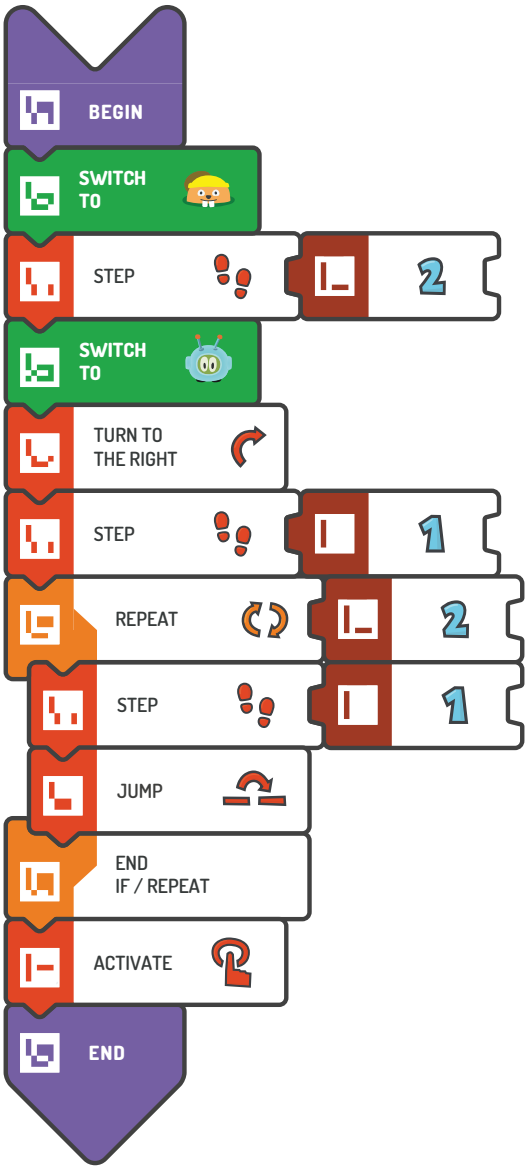
When controlling the beavers, you can move small ice floes and create new routes for Scottie. Use \nality in this quest.

Make sure your pupils use the **REPEAT** loop.

Challenge:



Example solution:



Challenge:



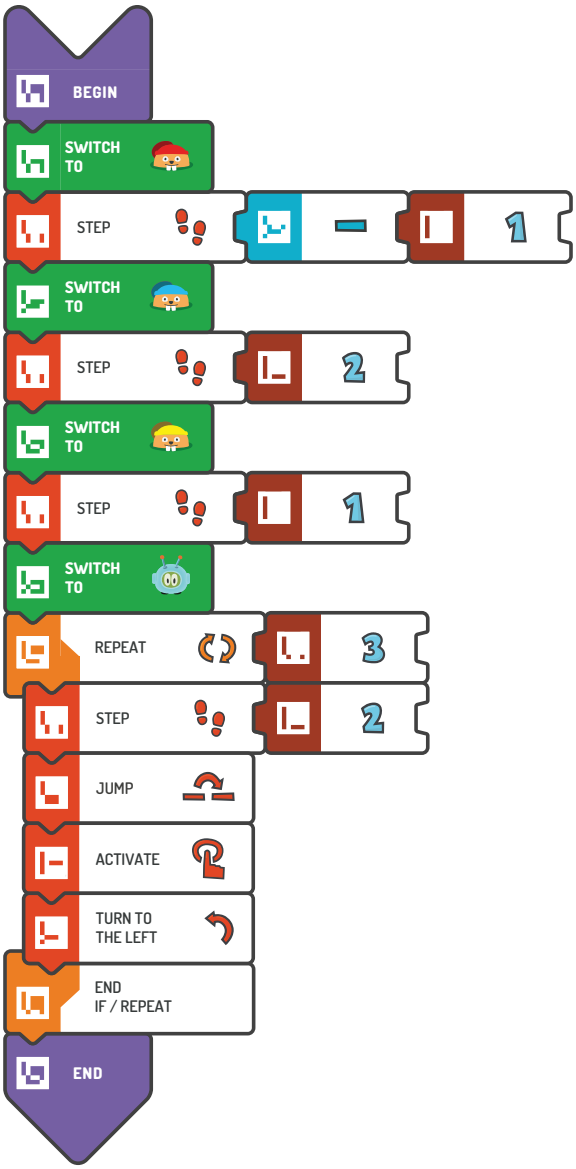
Example solution (in text format)::

```
BEGIN
    SWITCH TO - YELLOW BEAVER:
    STEP (2)
    SWITCH TO - SCOTTIE:
    TURN TO THE RIGHT
    STEP (1)
    REPEAT (2) {
        STEP (1)
        JUMP
    }
    ACTIVATE
END
```

Challenge:



Example solution:



Challenge:



Example solution (in text format)::

```
BEGIN
    SWITCH TO - RED BEAVER:
    STEP (-1)
    SWITCH TO - BLUE BEAVER:
    STEP (2)
    SWITCH TO - YELLOW BEAVER:
    STEP (1)
    SWITCH TO - SCOTTIE:
    REPEAT (3) {
        STEP (2)
        JUMP
        ACTIVATE
        TURN TO THE LEFT
    }
END
```

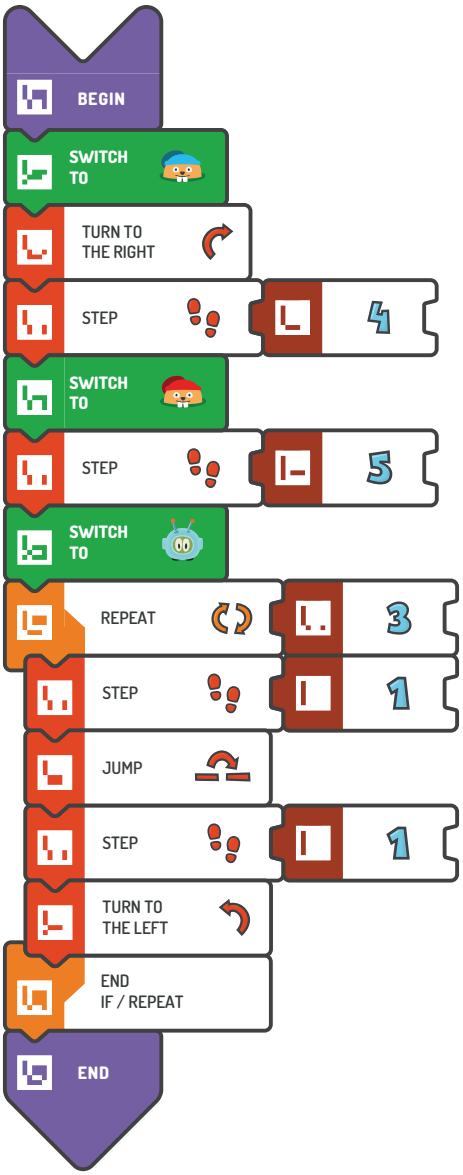
HINT:
selecting a character

Select a character to program to optimise the code (make it as short as possible).

Challenge:



Example solution:



Challenge:



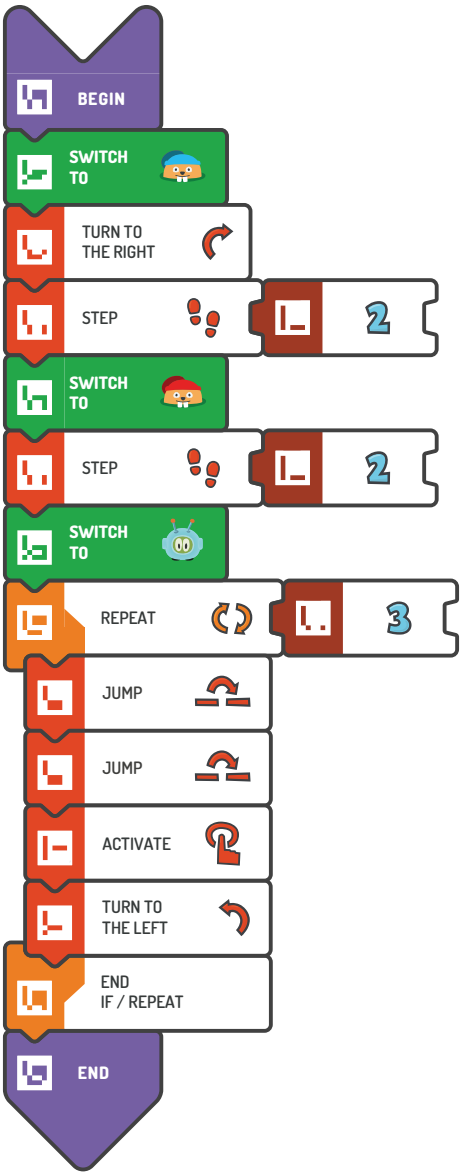
Example solution (in text format)::

```
BEGIN
    SWITCH TO - BLUE BEAVER:
    TURN TO THE RIGHT
    STEP (4)
    SWITCH TO - RED BEAVER:
    STEP (5)
    SWITCH TO - SCOTTIE:
    REPEAT (3) {
        STEP (1)
        JUMP
        STEP (1)
        JUMP
        ACTIVATE
        TURN TO THE LEFT
    }
END
```

Challenge:



Example solution:



Challenge:



Example solution (in text format)::

```
BEGIN
    SWITCH TO - BLUE BEAVER:
    TURN TO THE RIGHT
    STEP (2)
    SWITCH TO - RED BEAVER:
    STEP (2)
    SWITCH TO - SCOTTIE:
    REPEAT (3) {
        JUMP
        JUMP
        ACTIVATE
        TURN TO THE LEFT
    }
END
```