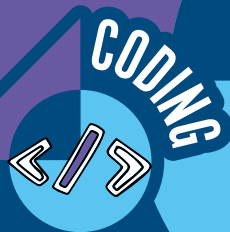
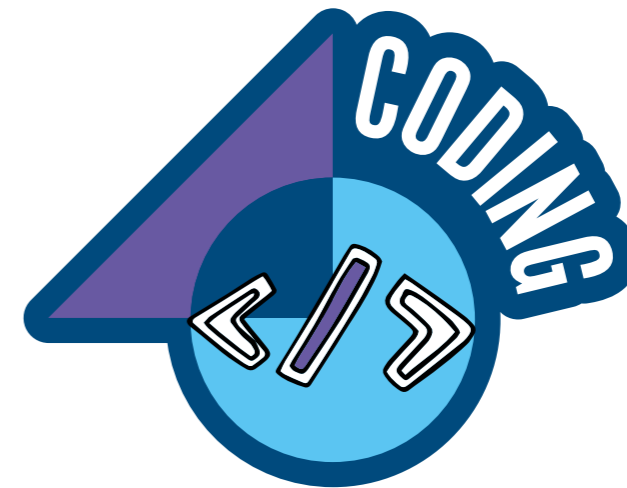


MAKERKOFFER



WWW.BEECREATIVE.LU



EN

Content

- 01 Introduction
 - Tools and materials
 - Levels
 - Skills to be learned
 - Before the game
- 03 Instructions
 - LEVEL 1
 - LEVEL 2
 - LEVEL 3

Learn how to code – with a board game

Introduction

In this board game, robots collect precious coins by picking up waste and disposing of it in the correct way. To move across the board and do this, however, they need to be properly programmed. Each player programmes their own robot with the aim of collecting the most coins and ending up as the winner. As the players gain more experience, the complexity of the game increases.

Why coding/programming?

In our increasingly digital and connected world, programming or coding has become a crucial skill. In order to understand the technologies around us, having a basic understanding of this skill is a must. Only those who have developed a deeper understanding of coding will be able to participate meaningfully in the development of the cars, mobile devices and smart technologies of tomorrow.

Why a board game?

This board game aims to awaken children's and young people's interest in coding in a fun and playful way – without using computers. Having fun is central to the process: The players will be motivated to use coding logic to guide their robot in the best possible way and thus acquire this important skill.

The advantage for you as an educator: You will be able to give children and young people a basic understanding of and an interest in coding, without having any programming knowledge or experience yourself. You do not need a computer, nor do you need to install programming environments or have access to the internet.

Programming is not an end in itself. It is always about a specific application, about achieving concrete goals (in this game: to collect waste and earn coins). Programming means defining individual steps that need to be executed in a certain order, a bit like in a cake recipe.

Another important lesson of the game: Mistakes are helpful – we need them to learn and improve our code – just like real life programmers.

Target group

The game is aimed at children (8 years and older) and young people, who have not yet had any experience with coding or who have no idea what the terms “coding” or “programming” mean. But it can also be a real challenge for adults, especially in the expert version.

Up to four players can take part in one game.

Duration

Printing the templates and assembling the different components of the game takes about 30 minutes before you can start playing (more information in the section on “Tools and materials”).

Each game takes about 20 minutes at the beginner level, 40 minutes at the advanced level and 60 minutes at the expert level. The duration also depends on the experience of the players.

Tools and materials

All the components of the game can be printed out (Print & Play principle). The cutting and gluing instructions for the different parts can be found on the printouts. This task can be carried out by children and adults together. After having assembled the game once, no more materials or tools are needed to play.

You will need:

- Printer
- Paper (A4)
- Scissors
- Adhesive tape

Tip: It is best to use stronger paper to print out the templates or reinforce the printed components with cardboard.

Levels

Players can race their robots across the board in three different difficulty levels. With each level, the complexity of the game mechanics increases; the mission of the players and the size of the board (and therefore of the game world) change.

BEGINNER: Level 1 is played with game board 1. Players acquire a basic understanding of the commands (“Go forward”, “Turn left”, ...) and their effects on the movements of the robots. The players learn the basic principles of robot programming.

The smaller game board allows for short, exciting games. For many young players, this entry level can already be quite a challenge. They have to think several steps ahead and from the robot's point of view (“After two steps, do I have to turn right or left?”).

ADVANCED: Level 2 is played with the larger game board 2. The players now have a mission: They have to clear the game world of waste and collect valuable coins in the process. In addition, they have to check their robot's battery from now on, as its energy is no longer unlimited.

EXPERT: Level 3 is played with game board 3 and adds further aspects of programming. The players can now equip their robots with individual “superpowers” and let them act automatically by programming so-called “functions” and “events”. This level's board is even larger and contains many new obstacles and pitfalls.

Skills to be learned

- Basic understanding of programming
- Thinking from the spatial perspective of the robot on the board
- Set and follow goals
- Plan and carry out a sequence of steps
- Strengthen concentration skills
- Identify errors and find solutions

Before the game

The mission

The phone is ringing. You pick up. It's an important job. It can't wait. Your programming skills are once again in demand. Your robot is ready. The metal is shining. The sensors are reliably delivering data – just like on Day One. Your mission today: Revitalising Polygon L2859. In other words: Collect and dispose of waste which has carelessly been left behind on this planet and could be dangerous for its environment. A piece of cake for you and your robot. Easy money. But not today. It looks like your robot is not the only one with the same mission at the same time in the same place.

Now it's up to you! Programme your robot so that it moves skilfully across the board, collects the waste left behind by people and disposes of it in the correct containers. For every correct disposal, your robot gets a coin.

But beware of the other players' robots. They push, shoot lasers all over the place and try to steal your valuable waste load to get their hands on the shiny coins.

In the end, there can only be one winner. As soon as all the piles of waste have been cleared, the player whose robot earned the most coins wins. If a robot earns three coins before all the waste is cleared, he is declared the winner.

Levels

The game has three levels: beginner, advanced and expert. With each level there are exciting new features, and the difficulty increases.

Level 1:

The beginner level is recommended for all those who play the game for the first time. It is the best and easiest way to get familiar with programming the robots and their movements. The duration of the game is rather short (15-20 minutes) and the board is smaller than the ones for the other levels.

Level 2:

Advanced players get the mission to collect waste and can earn valuable coins by disposing of it correctly. In addition, the robots now have limited energy and need to be charged every now and then.

Level 3:

The expert version is suitable for players who are really into programming and want to build more complex code. The board is really big now and the players can use functions and events to equip their robots with real superpowers.

Preparation

1) Print out all the parts of the game.

2) Cut out the robots along the cutting lines and glue them together using adhesive tape.

3) Cut out the game boards (for Level 2 and 3) along the dotted lines and glue the board pieces together.

4) Cut out all the other components of the game.

Tip: You might want to glue all the components onto 2 mm thick greyboard or cardboard (3-4mm also works) before cutting them out. It's simply more fun when they feel more solid.

Setting up

Choose a level and the corresponding board and place it in the middle.

Each player chooses a robot.

Each player gets a mission control board and places it in front of them, on the edge of the game board.

The programming blocks come in different colours. Each player chooses a colour and places all the blocks of that colour next to their mission control board.

From level 2 on:

The energy bars are distributed. Each player can fully charge their robot's battery with 7 bars.

The piles of waste are distributed on the game board: The waste-cards are placed face-down on the table and shuffled. The waste cards are then drawn at random and placed on the numbered spaces on the board. The first card that is drawn is put on the space with the number 1 and so on.

Instructions LEVEL 1

Goal

The robots are on a colourful flower meadow. The aim of the game is to pick as many flowers as possible and to plant them in your own garden. The game ends as soon as one player has picked three flowers and planted them in his garden.

Duration

About 20 minutes.

To make sure the game does not last too long, the players can agree on a certain number of rounds before starting to play.

Specific material

Small game board for level 1

Mission control boards for level 1 (one for each player).

Rules and programming blocks

Your garden is the starting area of your robot. To pick a flower, the robot must stand exactly on the space with a flower card. Using the command COLLECT, the robot can pick this flower. With the command UNLOAD, he can put it down again.

Programming the robot

There are two important components on the level 1 mission control board:

- The loading areas A, B and C are used to deposit collected items.
- The input mask for the main programme:

The main programme controls the movements and actions of the robot (e.g. collecting objects). In the beginning, the programme is empty. The player creates a programme by filling the empty spaces with programming blocks.

The main programme can contain up to four programming blocks. Each player can use the following blocks for their programme:

4x FORWARD: Each of these blocks moves the robot one space forward, in the robot's direction of view.

3x BACKWARD: Each of these blocks moves the robot one space backwards, opposite to his direction of view.

4x TURN LEFT: The robot turns 90° to the left.

4x TURN RIGHT: The robot turns 90° to the right.

4x WAIT: The robot takes a break. He does not move or perform any action. This command is particularly important, when the programme is to continue afterwards. For the robot, an empty programming space means that the programme has ended. So it is very important to use the WAIT-command, if the robot is to execute more commands after its break.

2x COLLECT: If there is an object on the space that the robot is standing on, he can collect it with this block.

Important: If there is more than one object on the same space, he can only collect the topmost one. The collected object is placed on the first empty loading area of the robot. If all the loading areas are empty, it is put on A. If A is already occupied, it is placed on area B and so on. If all the loading areas are occupied, no more objects can be collected, and they remain on the board.

UNLOAD A/B/C: With this block, the robot places the object from the loading area (as specified on the programming block) on the board. If there is no object on the loading area, nothing happens.

Pushing other robots

If robot A is to move to a space already occupied by robot B, robot A can move to this space and push robot B to the next space in the same direction. Robot B has been pushed from his space! Due to the impact, the pushed robot loses a load and has to leave it behind. Much to the delight of the pusher! After the push, he is now standing on the space with the object and can pick it up with the appropriate COLLECT-block.

Special rules for pushing

Pushing several robots: If several robots stand behind or next to each other, they can all be pushed by an approaching robot. All pushed robots are pushed one space further and lose one load.

Shoving at the edge of the board: A robot at the edge of the board cannot be pushed off the board. It remains on its space and keeps its load.

Shoving into an obstacle: A robot standing near an obstacle (e.g. a tree) cannot be pushed into the obstacle. It remains on its space and keeps its load.

Gameplay

All players place their robots on one of the starting areas. The youngest player chooses the starting area for his or her robot first. The game continues clockwise and the player to the left chooses his or her starting area next (clockwise).

The game is played over several rounds. Each round consists of a programming phase followed by an action phase.

Programming phase: Determining the path and actions of the robot

All the players grab a screen and place it on the mission control board so that their main programme is shielded from the others.

The players decide which path their robot should take on the board and which programme blocks are necessary for the corresponding movements (e.g. go forward) and actions (e.g. collect an object).

They choose the first four blocks from their collection and place them on the four free spaces of the main programme on the mission control board. The order in which the blocks are placed is very important: during the action phase, they will be executed one after the other, from top to bottom.

Action phase: The robots come to life!

The players remove their screens. Now, everybody can see all main programmes. From now on, the main programmes can no longer be changed.

The second youngest player starts. His or her robot can now execute the movement or action placed in the uppermost space of the main programme. However, the player is not allowed to move the robot. He or she announces the action (e.g. "Turn left!") and the robot is then moved by the player to his or her left.

The game continues clockwise. The player who just moved the first robot, announces his or her first programming block (e.g. "Move forward!") and the player to his or her left moves the robot.

As soon as all players have executed their first programming block, they proceed in the same way for the second, third and fourth block.

More rounds: When all programming blocks have been executed, a new round begins. There is a new programming phase followed by a new action phase. All players put up their screens again and build a new programme.

Have fun playing!



Instructions LEVEL 2

Goal

In level 2, the robots have a new mission: collecting waste and disposing of it correctly. The game ends as soon as all waste objects have been collected and disposed of correctly. For each correct disposal, the robot gets a coin as a reward. The robot with the most coins wins the game. If a robot manages to earn three coins, it is declared the winner, even if there are still waste objects left on the board.

Duration

About 40 minutes.

Specific material

Middle-sized board for level 2
Mission control boards for level 2

Rules and programming blocks

The rules and programming blocks from level 1 stay valid. But there are a few important new features!

Battery:

The robots no longer have unlimited energy. Their battery lasts a few rounds, before it needs to be recharged. A long main programme costs a lot of energy, a shorter one only a little.

A main programme with 3 or 4 programming blocks uses up one energy bar of the battery.

If the main programme contains only 1 or 2 blocks, the energy loss is compensated by the robot's solar cells. The robot's energy level stays the same.

If there are no programming blocks in the main programme, the robot can even gain an energy bar. So you can recharge the battery by taking a break and leaving your main programme empty for one round.

Fully charging the battery: The battery can be fully charged at charging stations. If the robot stands on a space with the Charge/Energy symbol after a move, the battery can be fully refilled with energy bars.

New programming block:

The robots can now use a laser gun with a new programming block.

2x FIRE LASER: With this block, the robot shoots a laser beam in its line of sight. The laser is not strong enough to damage other robots, but it startles them and they lose a valuable load. The laser beam reaches to the edge of the board; if there is an obstacle (e.g. a tree or bushes), the beam ends at the obstacle. Firing the laser costs one energy bar. So if you shoot all the time, you will quickly run out of battery.

Waste disposal:

There are different types of waste: some of it belongs in the organic waste bin, the paper waste bin, to the residual waste and some if it can be recycled at a recycling centre. Each type of disposal has its own colour.

The waste cards have a coloured border. It indicates which type of waste it is and where it should be disposed of. Waste with a brown border, for example, belongs to residual waste and has to be unloaded on a brown disposal space.

Collecting waste

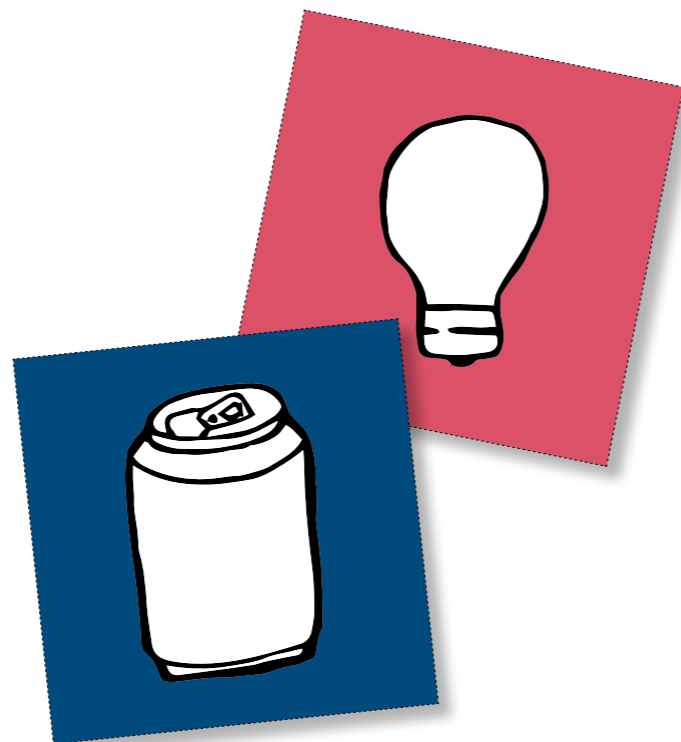
Before it can correctly dispose of a pile of waste, the robot must first pick it up. It can do this with the COLLECT-command while standing on the space with the waste pile. The pile is then loaded on the first available loading area of the robot. If all the loading areas are occupied, the waste pile cannot be collected and remains there.

Unloading waste

After it has been collected, the waste must be taken to the correct disposal area. When the robot stands on the disposal space with the matching colour, he can unload the waste with the command UNLOAD A/B/C.

Earning coins

If a pile of waste is unloaded onto a disposal space, the other players check whether the colours of the waste pile and the disposal space match. If they do, the waste is considered to have been disposed of correctly. The waste card can be put aside. As a reward, the robot gets a coin. The coin is placed on the coin section of the robot's mission control board.



Gameplay

Round after round, the task is to build a clever main programme to be executed step by step by the robot in the subsequent action phase.

Programming phase

As in level 1, the screens are set up and the players build their main programme. Each player must make sure that the battery has enough energy left to move the robot or to fire a laser. Or does the robot perhaps need a little time-out to recharge its battery by one energy bar?



Action phase

All players remove their screens. Before the robots start executing their main programmes, the players perform a battery check. All robots whose main programme contains 3 or 4 programming blocks, must remove an energy bar from their battery before they can start executing their main programme. All robots with 1 or 2 blocks can keep all their energy bars. Robots whose main programme has remained completely empty, can even charge their battery by one energy bar (as long as the battery is not already fully charged).

After the battery check, the main programmes are executed like in level 1.

If the laser is used, the robot loses one energy bar right before the FIRE LASER-command is executed. If the robot has no energy bars left, he cannot fire the laser. Nothing happens.

Let's start!

Instructions LEVEL 3

Goal

Now it's getting really exciting! To master level 3, you should have had some practice with levels 1 and 2. In addition, the robots now have superpowers! With so-called "functions" and "events", they can dash incredibly far in a single move, they can turn around or collect several objects at once. The mission and the goal, however, remain the same as in level 2: To clear the area of waste and to earn three coins before anyone else does. Therefore, it is important to come up with a strategy right from the start and to programme clever functions.

Duration

About 60 minutes.

Specific material

Big game board for level 3

Mission control boards for level 3.

Rules and programming blocks

The rules and programming blocks from levels 1 and 2 are still valid. But there are two new major extensions:

Functions:

With a function, your robot can execute up to three commands at once (in a single turn). It could for example move three steps forward instead of just one. But you're the one who defines exactly what your robot should do. You can choose all programming blocks at your disposal. You can programme a total of two functions. The first one is called "alpha", the second one "beta". Each function can contain one, two or a maximum of three programming blocks. The functions do not run by themselves. They must be started in the main programme with the following new programming blocks:

START ALPHA / START BETA: To start a function, one of these blocks must be executed. If you want to execute "alpha", you have to integrate the block START ALPHA in your main programme. When it's your robot's turn to execute this block, all the commands of the function can be executed one after the other before it's the next player's turn.

Important! Functions are programmed at the beginning of a game and cannot be changed during a game. Each player must think carefully about how to design the functions to his advantage.

Events:

Every robot has sensors. They provide it with important information about what is happening around it. Before each turn, these sensors provide new data about possible events, e.g. whether there is an object to collect under the robot, an obstacle directly in front of it or whether there is another robot in its field of view. Players can decide which event their robot should react to and how. If a robot is standing on a pile of waste for example, he could collect it as a reaction. If a robot is standing right behind another one, he could attack his opponent with a push. However, each robot can only react to one event. Such a short-term reaction can contain a maximum of two blocks.

The sensors can register the following events:

Object under robot: Programming blocks for this event are executed, if there is a pile of waste under the robot at the beginning of the turn (before the main programme is executed).

Obstacle right in front of robot: If the robot is to automatically avoid obstacles, you can use this block.

Opponent right in front of robot: This block can be used to detect other robots as soon as they are right in front of your own robot (not to the left, the right or behind).

Opponent right behind robot: If an opponent approaches your robot from behind, this block can be used to detect him.

Opponent in field of view: All the spaces in the line of sight of the robot constitute its field of view. If there are no obstacles in the way, the field of view can stretch across the whole board. If there is an obstacle in the robot's line of sight, the field of view extends to the obstacle but not beyond it.

Gameplay

The game structure is similar to level 2, but includes the new components and rules.

The functions and events are created once, before the first programming phase begins. They can no longer be modified during the game.

Programming phase

Now it's time again: set up your screens and create your main programme! Don't forget to check if your robot has enough energy for all the actions!

In level 3, the blocks START ALPHA und START BETA can be used to start the functions.

Action phase

As soon as all players have finished their programmes, the screens come down.

Before executing the first move, the players perform a battery check. Depending on the length of the main programme, energy bars are added or removed from the battery, just like in level 2.

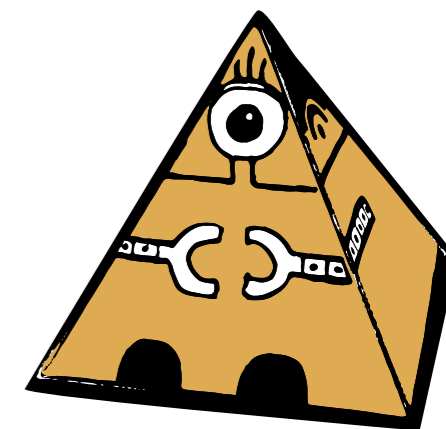
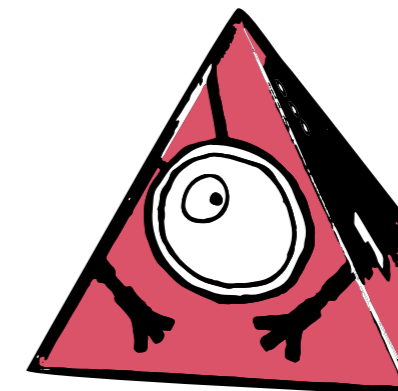
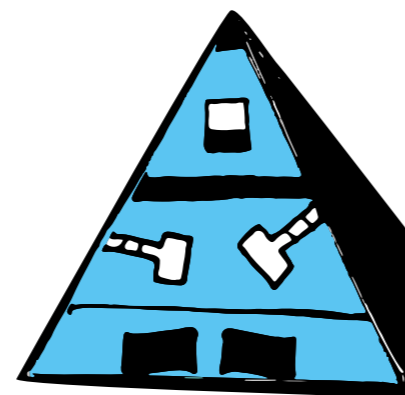
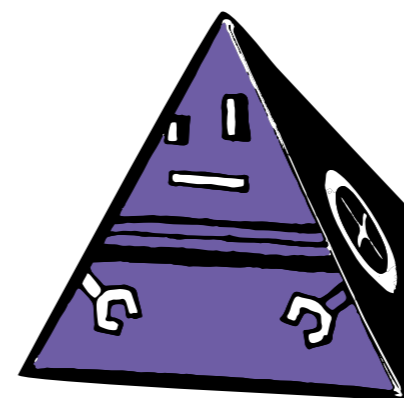
Before taking their turn, each player checks the events. If the programmed event applies (if for example there is a robot right in front of it), the programming blocks of this event are executed immediately (e.g. FORWARD, COLLECT). The event is only checked once per turn. If the player forgets to check for the event, this has no influence on the rest of the game. It cannot be executed later on, the game continues.

Now the first block of the main programme is executed. If a function is started in the main programme, all the blocks of this function are executed in one turn.

Then it's the next player's turn to check for his or her event before he or she too can move on to the main programme.

And now: speed away!

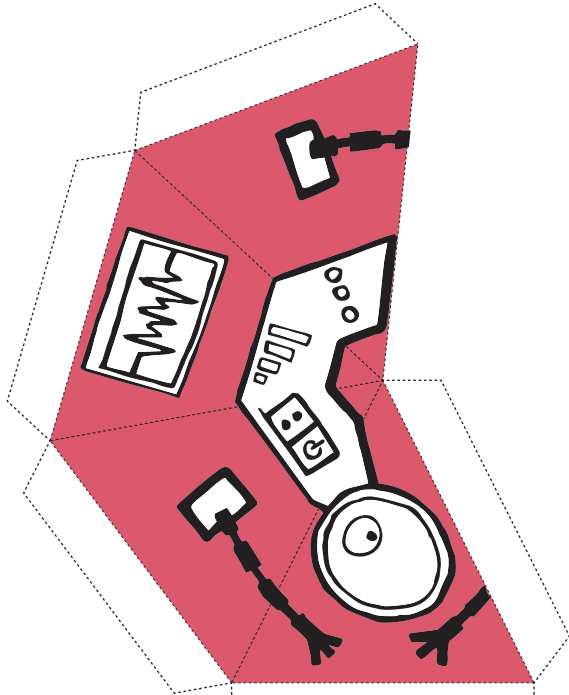
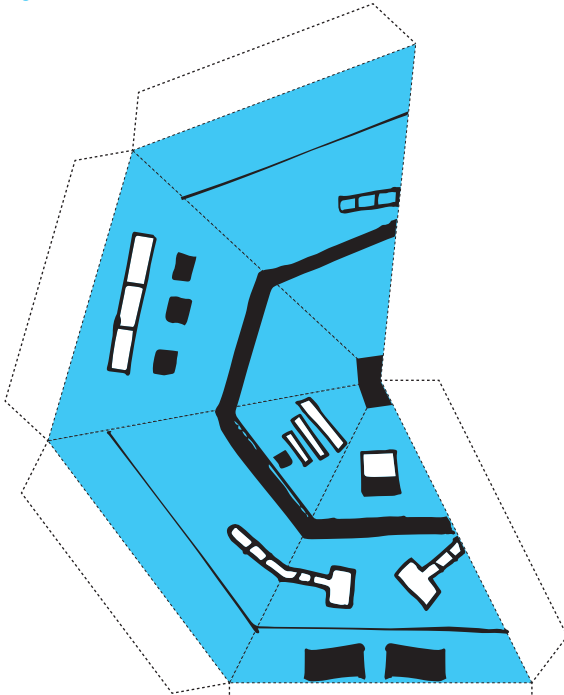
Enjoy!



WWW.BEECREATIVE.LU



ROBTER ROBOTS ROBOTS



OLLI

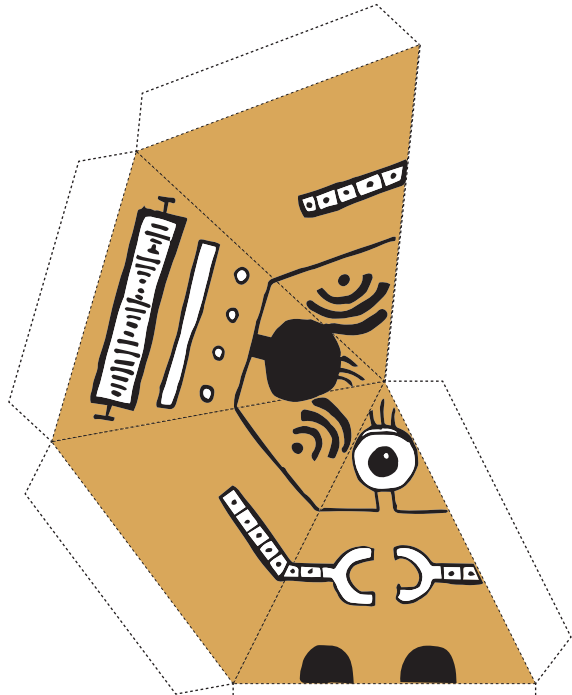
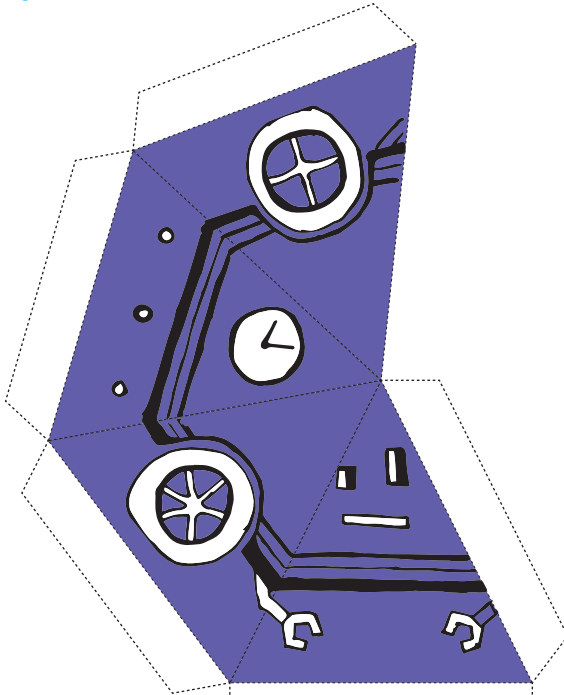


MILLI





ROBOTER ROBOTS ROBOTS



ROLLI



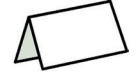
LILLI

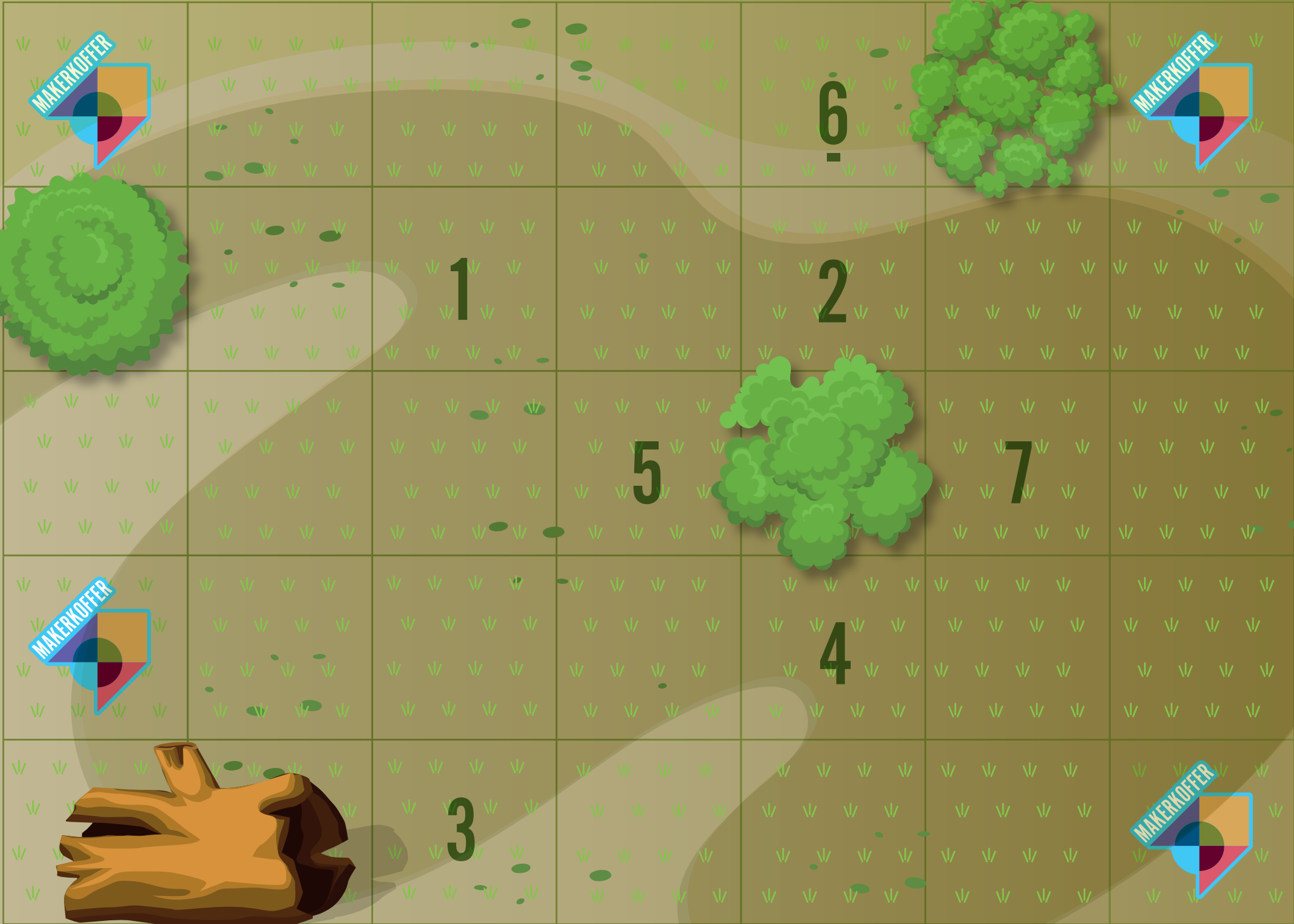


SICHTBLENDE ÉCRAN SCREEN



SICHTBLENDE ÉCRAN SCREEN





MAKERKOFFER

MAKERKOFFER

MAKERKOFFER

MAKERKOFFER

1

6

2

5

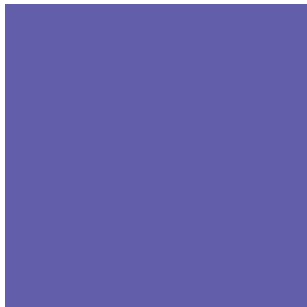
7


4


3




Ladeflächen
Espaces de chargements
Loading areas


A 


B 


C 

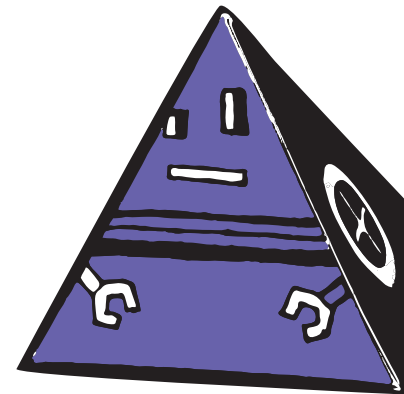
Eingabemaske für das Hauptprogramm
Masque de saisie pour le programme principal
Input mask for the main programme

1 

2 

3 

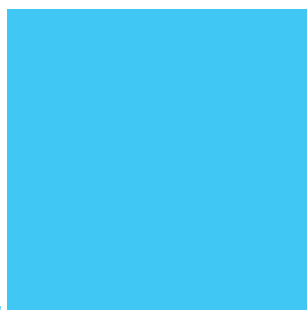
4 



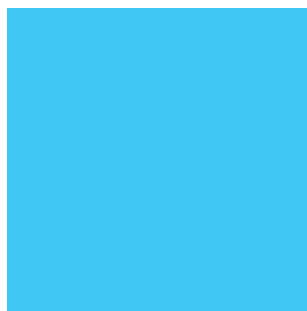


Ladeflächen
Espaces de chargements
Loading areas

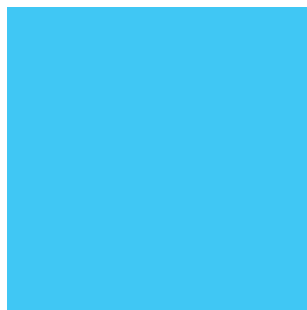
A



B



C



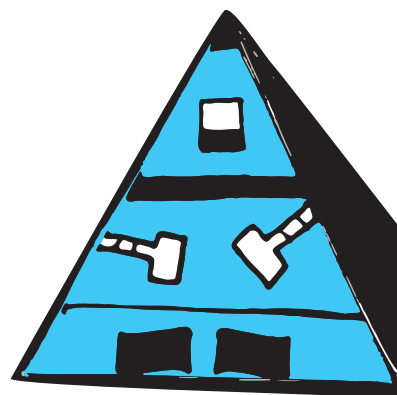
Eingabemaske für das Hauptprogramm
Masque de saisie pour le programme principal
Input mask for the main programme

1

2

3

4





Ladeflächen
Espaces de chargements
Loading areas

Eingabemaske für das Hauptprogramm
Masque de saisie pour le programme principal
Input mask for the main programme

A



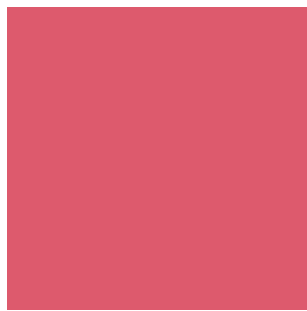
1

2

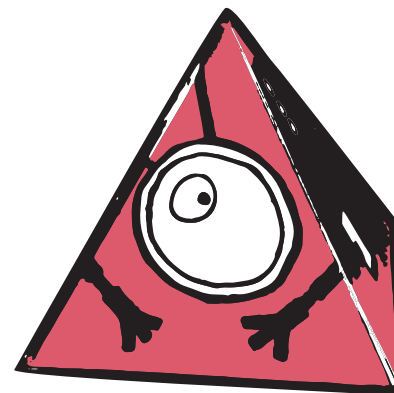
3

4

B



C





Ladeflächen
Espaces de chargements
Loading areas

A

B

C

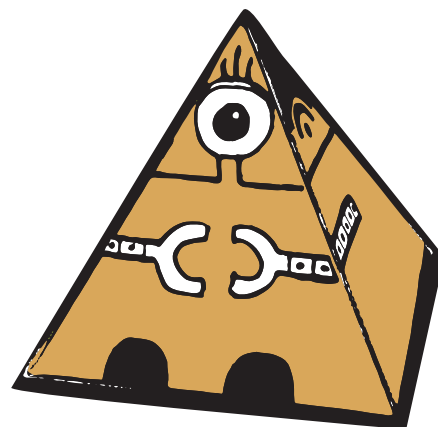
Eingabemaske für das Hauptprogramm
Masque de saisie pour le programme principal
Input mask for the main programme

1

2

3

4









































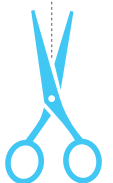


PROGRAMMIERBLÖCKE LEVEL 1
 BLOCS DE PROGRAMMATION LEVEL 1
 PROGRAMMING BLOCKS LEVEL 1



ROLLI

 <p>VORWÄRTS AVANCER FORWARD</p> 	 <p>LINKS DREHEN TOURNER A GAUCHE TURN LEFT</p> 	 <p>RECHTS DREHEN TOURNER A DROITE TURN RIGHT</p> 
 <p>VORWÄRTS AVANCER FORWARD</p> 	 <p>LINKS DREHEN TOURNER A GAUCHE TURN LEFT</p> 	 <p>RECHTS DREHEN TOURNER A DROITE TURN RIGHT</p> 
 <p>VORWÄRTS AVANCER FORWARD</p> 	 <p>LINKS DREHEN TOURNER A GAUCHE TURN LEFT</p> 	 <p>RECHTS DREHEN TOURNER A DROITE TURN RIGHT</p> 
 <p>VORWÄRTS AVANCER FORWARD</p> 	 <p>LINKS DREHEN TOURNER A GAUCHE TURN LEFT</p> 	 <p>RECHTS DREHEN TOURNER A DROITE TURN RIGHT</p> 
 <p>RÜCKWÄRTS ARRIÈRE BACKWARD</p> 	 <p>RÜCKWÄRTS ARRIÈRE BACKWARD</p> 	 <p>RÜCKWÄRTS ARRIÈRE BACKWARD</p> 
 <p>WARTE ATTENDRE WAIT</p> 	 <p>WARTE ATTENDRE WAIT</p> 	 <p>WARTE ATTENDRE WAIT</p> 
 <p>WARTE ATTENDRE WAIT</p> 	<p>EINSAMMELN COLLECTER COLLECT</p>	<p>EINSAMMELN COLLECTER COLLECT</p>
<p>ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C</p>	<p>ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C</p>	<p>ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C</p>









































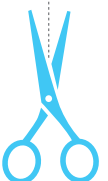


PROGRAMMIERBLÖCKE LEVEL 1
 BLOCS DE PROGRAMMATION LEVEL 1
 PROGRAMMING BLOCKS LEVEL 1



LILLI

 VORWÄRTS AVANCER FORWARD 	 LINKS DREHEN TOURNER A GAUCHE TURN LEFT 	 RECHTS DREHEN TOURNER A DROITE TURN RIGHT 
 VORWÄRTS AVANCER FORWARD 	 LINKS DREHEN TOURNER A GAUCHE TURN LEFT 	 RECHTS DREHEN TOURNER A DROITE TURN RIGHT 
 VORWÄRTS AVANCER FORWARD 	 LINKS DREHEN TOURNER A GAUCHE TURN LEFT 	 RECHTS DREHEN TOURNER A DROITE TURN RIGHT 
 VORWÄRTS AVANCER FORWARD 	 LINKS DREHEN TOURNER A GAUCHE TURN LEFT 	 RECHTS DREHEN TOURNER A DROITE TURN RIGHT 
 RÜCKWÄRTS ARRIÈRE BACKWARD 	 RÜCKWÄRTS ARRIÈRE BACKWARD 	 RÜCKWÄRTS ARRIÈRE BACKWARD 
 WARTE ATTENDRE WAIT 	 WARTE ATTENDRE WAIT 	 WARTE ATTENDRE WAIT 
 WARTE ATTENDRE WAIT 	EINSAMMELN COLLECTER COLLECT	EINSAMMELN COLLECTER COLLECT
ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C	ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C	ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C











































PROGRAMMIERBLÖCKE LEVEL 1
 BLOCS DE PROGRAMMATION LEVEL 1
 PROGRAMMING BLOCKS LEVEL 1



MILLI

 VORWÄRTS AVANCER FORWARD 	 LINKS DREHEN TOURNER A GAUCHE TURN LEFT 	 RECHTS DREHEN TOURNER A DROITE TURN RIGHT 
 VORWÄRTS AVANCER FORWARD 	 LINKS DREHEN TOURNER A GAUCHE TURN LEFT 	 RECHTS DREHEN TOURNER A DROITE TURN RIGHT 
 VORWÄRTS AVANCER FORWARD 	 LINKS DREHEN TOURNER A GAUCHE TURN LEFT 	 RECHTS DREHEN TOURNER A DROITE TURN RIGHT 
 VORWÄRTS AVANCER FORWARD 	 LINKS DREHEN TOURNER A GAUCHE TURN LEFT 	 RECHTS DREHEN TOURNER A DROITE TURN RIGHT 
 RÜCKWÄRTS ARRIÈRE BACKWARD 	 RÜCKWÄRTS ARRIÈRE BACKWARD 	 RÜCKWÄRTS ARRIÈRE BACKWARD 
 WARTE ATTENDRE WAIT 	 WARTE ATTENDRE WAIT 	 WARTE ATTENDRE WAIT 
 WARTE ATTENDRE WAIT 	EINSAMMELN COLLECTER COLLECT	EINSAMMELN COLLECTER COLLECT
ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C	ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C	ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C











































PROGRAMMIERBLÖCKE LEVEL 1
 BLOCS DE PROGRAMMATION LEVEL 1
 PROGRAMMING BLOCKS LEVEL 1



OLLI

 VORWÄRTS AVANCER FORWARD 	 LINKS DREHEN TOURNER A GAUCHE TURN LEFT 	 RECHTS DREHEN TOURNER A DROITE TURN RIGHT 
 VORWÄRTS AVANCER FORWARD 	 LINKS DREHEN TOURNER A GAUCHE TURN LEFT 	 RECHTS DREHEN TOURNER A DROITE TURN RIGHT 
 VORWÄRTS AVANCER FORWARD 	 LINKS DREHEN TOURNER A GAUCHE TURN LEFT 	 RECHTS DREHEN TOURNER A DROITE TURN RIGHT 
 VORWÄRTS AVANCER FORWARD 	 LINKS DREHEN TOURNER A GAUCHE TURN LEFT 	 RECHTS DREHEN TOURNER A DROITE TURN RIGHT 
 RÜCKWÄRTS ARRIÈRE BACKWARD 	 RÜCKWÄRTS ARRIÈRE BACKWARD 	 RÜCKWÄRTS ARRIÈRE BACKWARD 
 WARTE ATTENDRE WAIT 	 WARTE ATTENDRE WAIT 	 WARTE ATTENDRE WAIT 
 WARTE ATTENDRE WAIT 	EINSAMMELN COLLECTER COLLECT	EINSAMMELN COLLECTER COLLECT
ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C	ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C	ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C





BLUMEN KARTEN LEVEL 1
CARTES FLEURS LEVEL 1
FLOWER CARDS LEVEL 1





1

4

3

2

MAKERKOFFER

MAKERKOFFER

MAKERKOFFER

MAKERKOFFER

HIER ZUSAMMENLEBEN COLLER ENSEMBLE ICI STICK TOGETHER HERE

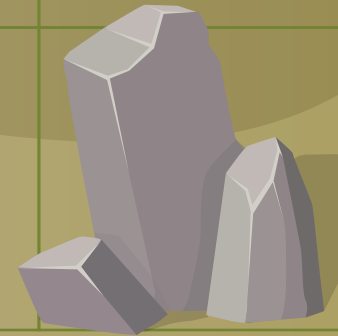
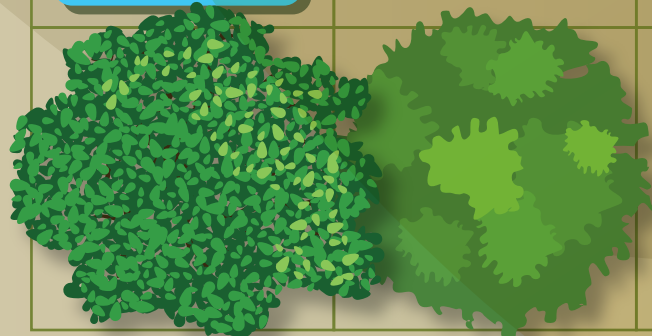
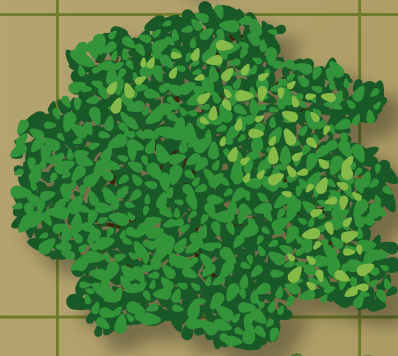


7

5

6

8





Ladeflächen
Espaces de chargements
Loading areas

A

B

C

Eingabemaske für das Hauptprogramm
Masque de saisie pour le programme principal
Input mask for the main programme

1

2

3

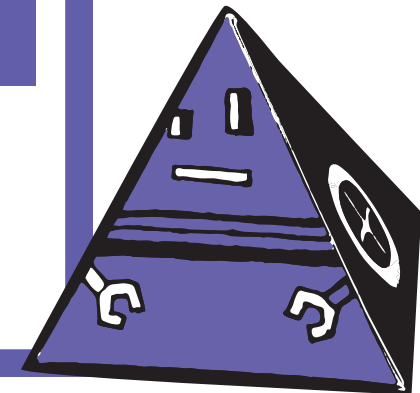
4

Batterie Batterie Battery

--	--	--	--	--	--	--

Coins Coins Coins

--	--	--	--	--





Ladeflächen
Espaces de chargements
Loading areas

A

B

C

Eingabemaske für das Hauptprogramm
Masque de saisie pour le programme principal
Input mask for the main programme

1

2

3

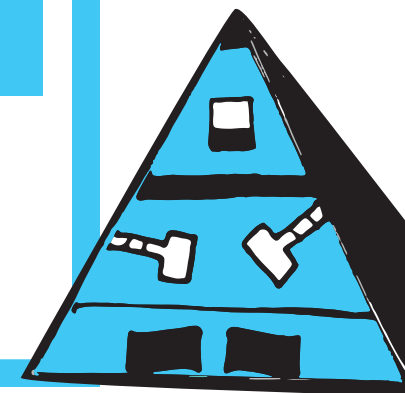
4

Batterie Batterie Battery

Seven blue rectangular slots representing a battery.

Coins Coins Coins

Five blue rectangular slots representing coins.





Ladeflächen
Espaces de chargements
Loading areas

A

B

C

Eingabemaske für das Hauptprogramm
Masque de saisie pour le programme principal
Input mask for the main programme

1

2

3

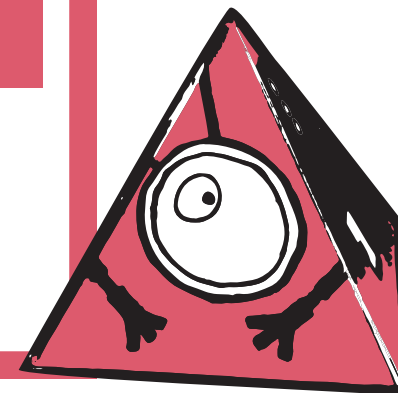
4

Batterie Batterie Battery

7 battery icons

Coins Coins Coins

5 coin icons





Ladeflächen
Espaces de chargements
Loading areas

A

B

C

Eingabemaske für das Hauptprogramm
Masque de saisie pour le programme principal
Input mask for the main programme

1

2

3

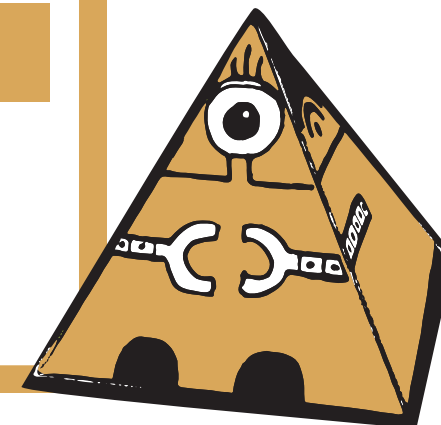
4

Batterie Batterie Battery

Seven battery icons

Coins Coins Coins

Five coin icons





PROGRAMMIERBLÖCKE LEVEL 2
 BLOCS DE PROGRAMMATION LEVEL 2
 PROGRAMMING BLOCKS LEVEL 2



ROLLI

VORWÄRTS AVANCER FORWARD	LINKS DREHEN TOURNER A GAUCHE TURN LEFT	RECHTS DREHEN TOURNER A DROITE TURN RIGHT
VORWÄRTS AVANCER FORWARD	LINKS DREHEN TOURNER A GAUCHE TURN LEFT	RECHTS DREHEN TOURNER A DROITE TURN RIGHT
VORWÄRTS AVANCER FORWARD	LINKS DREHEN TOURNER A GAUCHE TURN LEFT	RECHTS DREHEN TOURNER A DROITE TURN RIGHT
VORWÄRTS AVANCER FORWARD	LINKS DREHEN TOURNER A GAUCHE TURN LEFT	RECHTS DREHEN TOURNER A DROITE TURN RIGHT
RÜCKWÄRTS ARRIÈRE BACKWARD	RÜCKWÄRTS ARRIÈRE BACKWARD	RÜCKWÄRTS ARRIÈRE BACKWARD
WARTE ATTENDRE WAIT	WARTE ATTENDRE WAIT	WARTE ATTENDRE WAIT
WARTE ATTENDRE WAIT	EINSAMMELN COLLECTER COLLECT	EINSAMMELN COLLECTER COLLECT
ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C	ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C	ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C
LASER ABFEUERN LANCER LASER FIRE LASER	LASER ABFEUERN LANCER LASER FIRE LASER	



































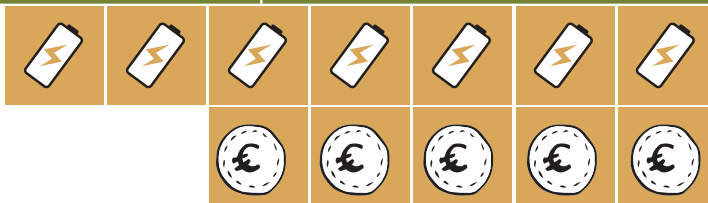


PROGRAMMIERBLÖCKE LEVEL 2
 BLOCS DE PROGRAMMATION LEVEL 2
 PROGRAMMING BLOCKS LEVEL 2



LILLI

 VORWÄRTS AVANCER FORWARD 	 LINKS DREHEN TOURNER A GAUCHE TURN LEFT 	 RECHTS DREHEN TOURNER A DROITE TURN RIGHT 
 VORWÄRTS AVANCER FORWARD 	 LINKS DREHEN TOURNER A GAUCHE TURN LEFT 	 RECHTS DREHEN TOURNER A DROITE TURN RIGHT 
 VORWÄRTS AVANCER FORWARD 	 LINKS DREHEN TOURNER A GAUCHE TURN LEFT 	 RECHTS DREHEN TOURNER A DROITE TURN RIGHT 
 VORWÄRTS AVANCER FORWARD 	 LINKS DREHEN TOURNER A GAUCHE TURN LEFT 	 RECHTS DREHEN TOURNER A DROITE TURN RIGHT 
 RÜCKWÄRTS ARRIÈRE BACKWARD 	 RÜCKWÄRTS ARRIÈRE BACKWARD 	 RÜCKWÄRTS ARRIÈRE BACKWARD 
 WARTE ATTENDRE WAIT 	 WARTE ATTENDRE WAIT 	 WARTE ATTENDRE WAIT 
 WARTE ATTENDRE WAIT 	EINSAMMELN COLLECTER COLLECT	EINSAMMELN COLLECTER COLLECT
ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C	ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C	ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C
LASER ABFEUERN LANCER LASER FIRE LASER	LASER ABFEUERN LANCER LASER FIRE LASER	









































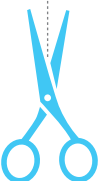
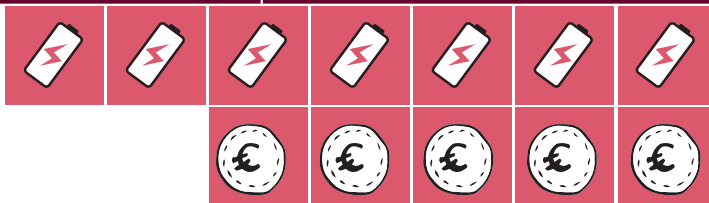


PROGRAMMIERBLÖCKE LEVEL 2
 BLOCS DE PROGRAMMATION LEVEL 2
 PROGRAMMING BLOCKS LEVEL 2



MILLI

 VORWÄRTS AVANCER FORWARD 	 LINKS DREHEN TOURNER A GAUCHE TURN LEFT 	 RECHTS DREHEN TOURNER A DROITE TURN RIGHT 
 VORWÄRTS AVANCER FORWARD 	 LINKS DREHEN TOURNER A GAUCHE TURN LEFT 	 RECHTS DREHEN TOURNER A DROITE TURN RIGHT 
 VORWÄRTS AVANCER FORWARD 	 LINKS DREHEN TOURNER A GAUCHE TURN LEFT 	 RECHTS DREHEN TOURNER A DROITE TURN RIGHT 
 VORWÄRTS AVANCER FORWARD 	 LINKS DREHEN TOURNER A GAUCHE TURN LEFT 	 RECHTS DREHEN TOURNER A DROITE TURN RIGHT 
 RÜCKWÄRTS ARRIÈRE BACKWARD 	 RÜCKWÄRTS ARRIÈRE BACKWARD 	 RÜCKWÄRTS ARRIÈRE BACKWARD 
 WARTE ATTENDRE WAIT 	 WARTE ATTENDRE WAIT 	 WARTE ATTENDRE WAIT 
 WARTE ATTENDRE WAIT 	EINSAMMELN COLLECTER COLLECT	EINSAMMELN COLLECTER COLLECT
ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C	ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C	ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C
LASER ABFEUERN LANCER LASER FIRE LASER	LASER ABFEUERN LANCER LASER FIRE LASER	









































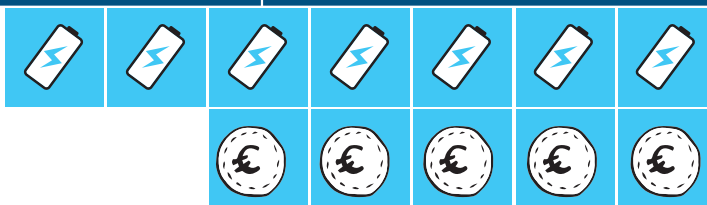


PROGRAMMIERBLÖCKE LEVEL 2
 BLOCS DE PROGRAMMATION LEVEL 2
 PROGRAMMING BLOCKS LEVEL 2



OLLI

 VORWÄRTS AVANCER FORWARD 	 LINKS DREHEN TOURNER A GAUCHE TURN LEFT 	 RECHTS DREHEN TOURNER A DROITE TURN RIGHT 
 VORWÄRTS AVANCER FORWARD 	 LINKS DREHEN TOURNER A GAUCHE TURN LEFT 	 RECHTS DREHEN TOURNER A DROITE TURN RIGHT 
 VORWÄRTS AVANCER FORWARD 	 LINKS DREHEN TOURNER A GAUCHE TURN LEFT 	 RECHTS DREHEN TOURNER A DROITE TURN RIGHT 
 VORWÄRTS AVANCER FORWARD 	 LINKS DREHEN TOURNER A GAUCHE TURN LEFT 	 RECHTS DREHEN TOURNER A DROITE TURN RIGHT 
 RÜCKWÄRTS ARRIÈRE BACKWARD 	 RÜCKWÄRTS ARRIÈRE BACKWARD 	 RÜCKWÄRTS ARRIÈRE BACKWARD 
 WARTE ATTENDRE WAIT 	 WARTE ATTENDRE WAIT 	 WARTE ATTENDRE WAIT 
 WARTE ATTENDRE WAIT 	EINSAMMELN COLLECTER COLLECT	
ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C		
LASER ABFEUERN LANCER LASER FIRE LASER		





MÜLL KARTEN LEVEL 2 & 3
CARTES DÉCHETS LEVEL 2 & 3
WASTE CARDS LEVEL 2 & 3



SPIELBRETT LEVEL 3 PLATEAU DE JEU LEVEL 3 GAME BOARD LEVEL 3



2

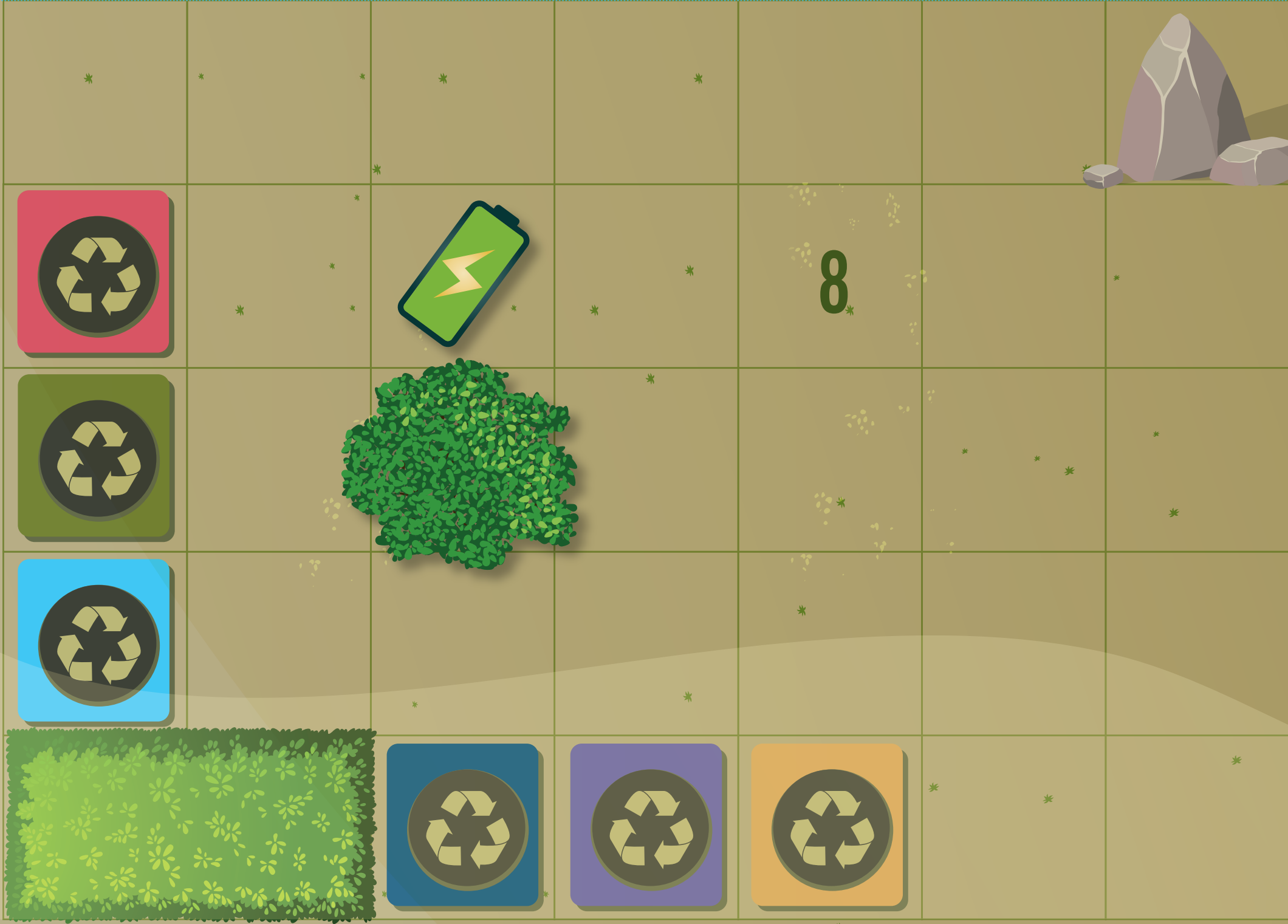


7

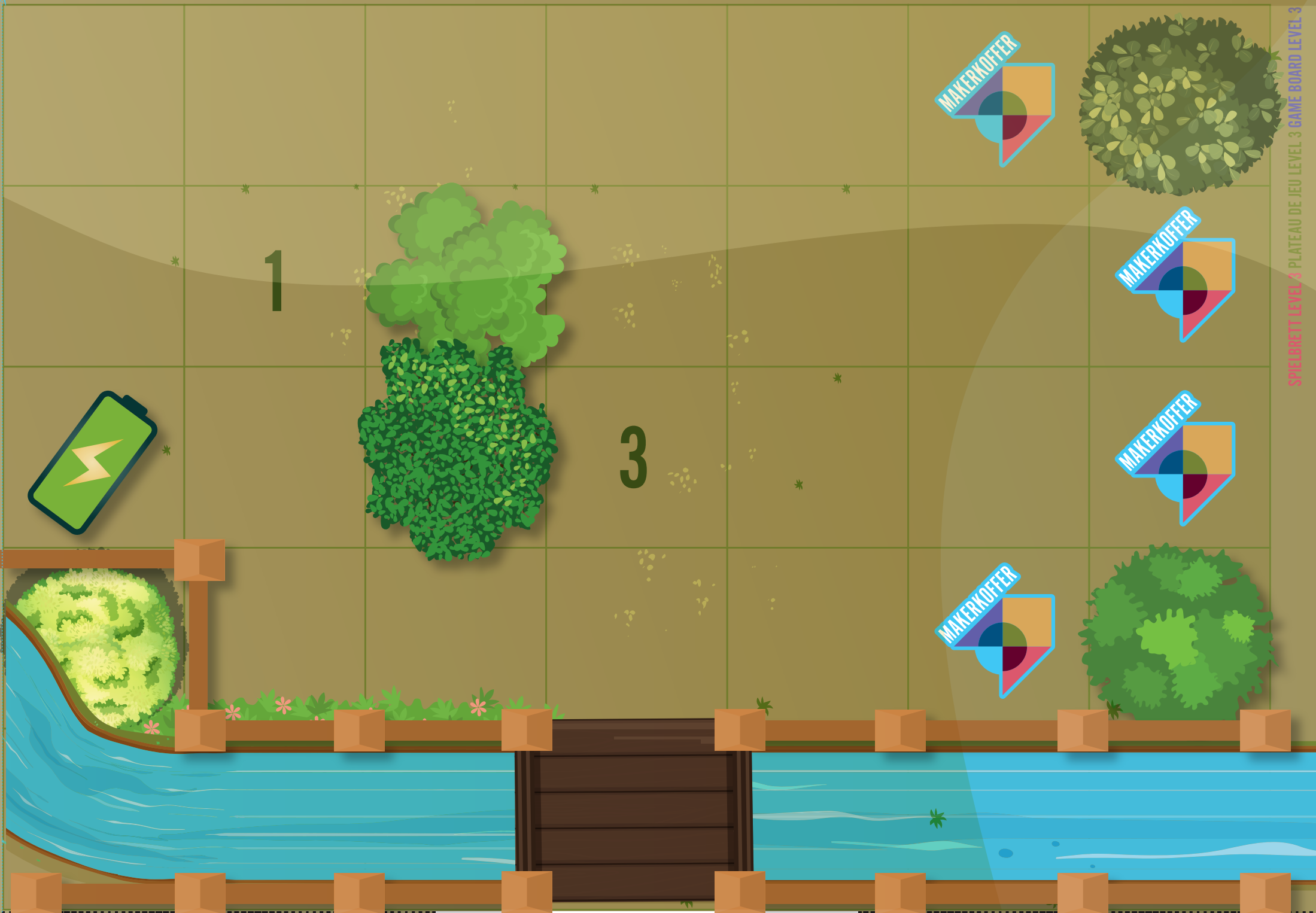
5

HIER ZUSAMMENKLEBEN COLLER ENSEMBLE ICI STICK TOGETHER HERE

HIER ZUSAMMENKLEBEN COLLER ENSEMBLE ICI STICK TOGETHER HERE



HIER ZUSAMMENKLEBEN COLLER ENSEMBLE ICI STICK TOGETHER HERE



1

3



HIER ZUSAMMENLEBEN COLLER ENSEMBLE ICI STICK TOGETHER HERE

SPELRETT LEVEL 4 PLATEAU DE JEU LEVEL 3 GAME BOARD LEVEL 3



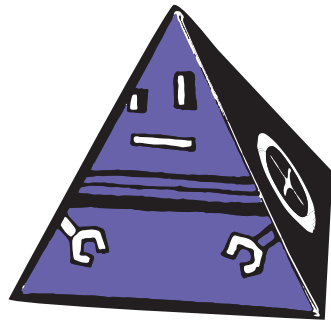
6

4

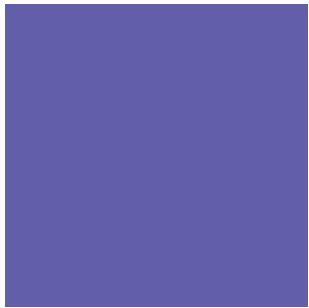




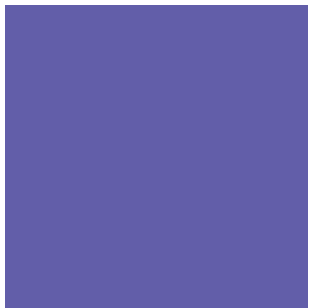
Ladeflächen
 Espaces de chargements
 Loading areas



A



B



C



Batterie Batterie Battery



Coins Coins Coins



Event
 Event
 Event



falls si if



dann alors then



Eingabemaske für das Hauptprogramm
 Masque de saisie pour le programme principal
 Input mask for the main programme



1



2



3



4

Funktion "alpha"
 Fonction "alpha"
 Fonction "alpha"



Funktion "beta"
 Fonction "beta"
 Fonction "beta"

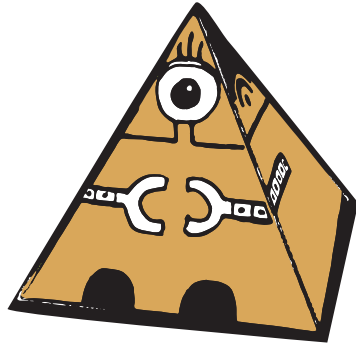




MISSION CONTROL BRETT LEVEL 3
 TABLEAU DE CONTRÔLE DE MISSION LEVEL 3
 MISSION CONTROL LEVEL 3



Ladeflächen
 Espaces de chargements
 Loading areas



A



B



C



Batterie Batterie Battery



Coins Coins Coins



Event
 Event
 Event



falls si if



dann alors then



Eingabemaske für das Hauptprogramm
 Masque de saisie pour le programme principal
 Input mask for the main programme



1



2



3



4

Funktion "alpha"
 Fonction "alpha"
 Fonction "alpha"

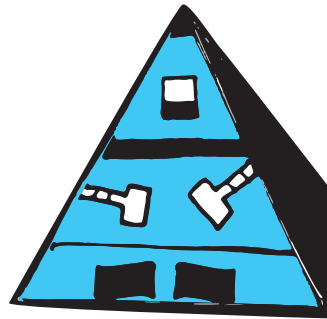


Funktion "beta"
 Fonction "beta"
 Fonction "beta"

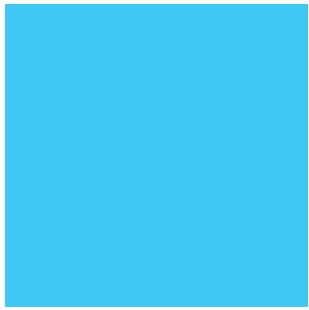




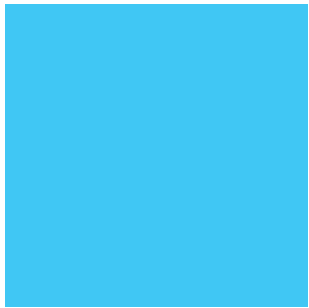
Ladeflächen
 Espaces de chargements
 Loading areas



A



B



C



Batterie Batterie Battery



Coins Coins Coins



Event
 Event
 Event

falls si if



dann alors then



Eingabemaske für das Hauptprogramm
 Masque de saisie pour le programme principal
 Input mask for the main programme

1



2



3



4



Funktion "alpha"
 Fonction "alpha"
 Fonction "alpha"

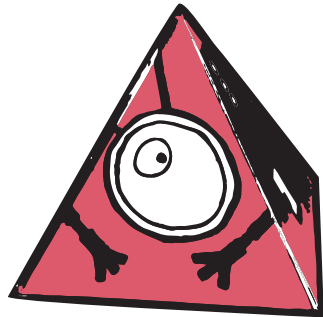


Funktion "beta"
 Fonction "beta"
 Fonction "beta"

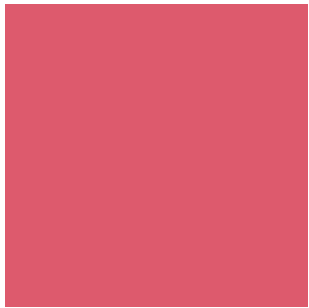




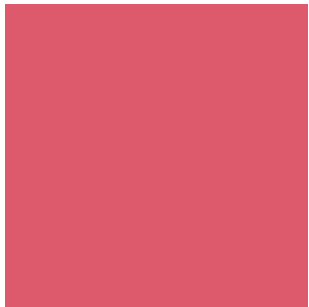
Ladeflächen
 Espaces de chargements
 Loading areas



A



B



C



Batterie Batterie Battery



Coins Coins Coins



Event
 Event
 Event

falls si if



dann alors then



Eingabemaske für das Hauptprogramm
 Masque de saisie pour le programme principal
 Input mask for the main programme

1



2



3



4



Funktion "alpha"
 Fonction "alpha"
 Fonction "alpha"



Funktion "beta"
 Fonction "beta"
 Fonction "beta"





PROGRAMMIERBLÖCKE LEVEL 3
 BLOCS DE PROGRAMMATION LEVEL 3
 PROGRAMMING BLOCKS LEVEL 3



ROLLI

VORWÄRTS AVANCER FORWARD	LINKS DREHEN TOURNER A GAUCHE TURN LEFT	RECHTS DREHEN TOURNER A DROITE TURN RIGHT
VORWÄRTS AVANCER FORWARD	LINKS DREHEN TOURNER A GAUCHE TURN LEFT	RECHTS DREHEN TOURNER A DROITE TURN RIGHT
VORWÄRTS AVANCER FORWARD	LINKS DREHEN TOURNER A GAUCHE TURN LEFT	RECHTS DREHEN TOURNER A DROITE TURN RIGHT
VORWÄRTS AVANCER FORWARD	LINKS DREHEN TOURNER A GAUCHE TURN LEFT	RECHTS DREHEN TOURNER A DROITE TURN RIGHT
RÜCKWÄRTS ARRIÈRE BACKWARD	RÜCKWÄRTS ARRIÈRE BACKWARD	RÜCKWÄRTS ARRIÈRE BACKWARD
WARTE ATTENDRE WAIT	WARTE ATTENDRE WAIT	WARTE ATTENDRE WAIT
WARTE ATTENDRE WAIT	EINSAMMELN COLLECTER COLLECT	EINSAMMELN COLLECTER COLLECT
ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C	ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C	ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C
LASER ABFEUERN LANCER LASER FIRE LASER	LASER ABFEUERN LANCER LASER FIRE LASER	STARTE "ALPHA" LANCER "ALPHA" START "ALPHA"

Gegner im Sichtfeld
Adversaire dans le champ visuel
Opponent in field of view

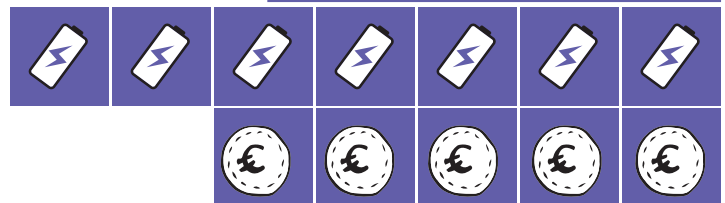
Objekt unter Roboter
Objet sous robot
Object under robot

Hindernis direkt vor Roboter
Obstacle devant robot
Obstacle in front of robot

Gegner direkt vor Roboter
Adversaire devant robot
Opponent in front of robot

Gegner direkt hinter Roboter
Adversaire derrière robot
Opponent behind robot

STARTE "BETA"
LANCER "BETA"
START "BETA"





PROGRAMMIERBLÖCKE LEVEL 3
 BLOCS DE PROGRAMMATION LEVEL 3
 PROGRAMMING BLOCKS LEVEL 3



LILLI

VORWÄRTS AVANCER FORWARD	LINKS DREHEN TOURNER A GAUCHE TURN LEFT	RECHTS DREHEN TOURNER A DROITE TURN RIGHT
VORWÄRTS AVANCER FORWARD	LINKS DREHEN TOURNER A GAUCHE TURN LEFT	RECHTS DREHEN TOURNER A DROITE TURN RIGHT
VORWÄRTS AVANCER FORWARD	LINKS DREHEN TOURNER A GAUCHE TURN LEFT	RECHTS DREHEN TOURNER A DROITE TURN RIGHT
VORWÄRTS AVANCER FORWARD	LINKS DREHEN TOURNER A GAUCHE TURN LEFT	RECHTS DREHEN TOURNER A DROITE TURN RIGHT
RÜCKWÄRTS ARRIÈRE BACKWARD	RÜCKWÄRTS ARRIÈRE BACKWARD	RÜCKWÄRTS ARRIÈRE BACKWARD
WARTE ATTENDRE WAIT	WARTE ATTENDRE WAIT	WARTE ATTENDRE WAIT
WARTE ATTENDRE WAIT	EINSAMMELN COLLECTER COLLECT	EINSAMMELN COLLECTER COLLECT
ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C	ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C	ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C
LASER ABFEUERN LANCER LASER FIRE LASER	LASER ABFEUERN LANCER LASER FIRE LASER	STARTE "ALPHA" LANCER "ALPHA" START "ALPHA"

Gegner im Sichtfeld
 Adversaire dans le champ visuel
 Opponent in field of view

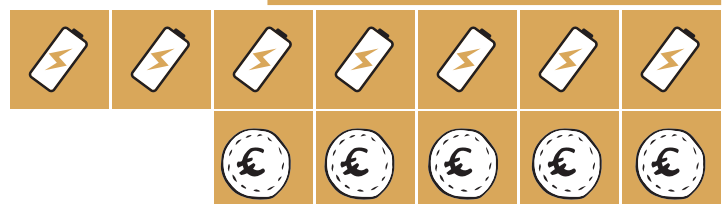
Objekt unter Roboter
 Objet sous robot
 Object under robot

Hindernis direkt vor Roboter
 Obstacle devant robot
 Obstacle in front of robot

Gegner direkt vor Roboter
 Adversaire devant robot
 Opponent in front of robot

Gegner direkt hinter Roboter
 Adversaire derrière robot
 Opponent behind robot

STARTE "BETA"
 LANCER "BETA"
 START "BETA"





PROGRAMMIERBLÖCKE LEVEL 3
 BLOCS DE PROGRAMMATION LEVEL 3
 PROGRAMMING BLOCKS LEVEL 3



MILLI

VORWÄRTS AVANCER FORWARD	LINKS DREHEN TOURNER A GAUCHE TURN LEFT	RECHTS DREHEN TOURNER A DROITE TURN RIGHT
VORWÄRTS AVANCER FORWARD	LINKS DREHEN TOURNER A GAUCHE TURN LEFT	RECHTS DREHEN TOURNER A DROITE TURN RIGHT
VORWÄRTS AVANCER FORWARD	LINKS DREHEN TOURNER A GAUCHE TURN LEFT	RECHTS DREHEN TOURNER A DROITE TURN RIGHT
VORWÄRTS AVANCER FORWARD	LINKS DREHEN TOURNER A GAUCHE TURN LEFT	RECHTS DREHEN TOURNER A DROITE TURN RIGHT
RÜCKWÄRTS ARRIÈRE BACKWARD	RÜCKWÄRTS ARRIÈRE BACKWARD	RÜCKWÄRTS ARRIÈRE BACKWARD
WARTE ATTENDRE WAIT	WARTE ATTENDRE WAIT	WARTE ATTENDRE WAIT
WARTE ATTENDRE WAIT	EINSAMMELN COLLECTER COLLECT	EINSAMMELN COLLECTER COLLECT
ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C	ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C	ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C
LASER ABFEUERN LANCER LASER FIRE LASER	LASER ABFEUERN LANCER LASER FIRE LASER	STARTE "ALPHA" LANCER "ALPHA" START "ALPHA"

Gegner im Sichtfeld
Adversaire dans le champ visuel
Opponent in field of view

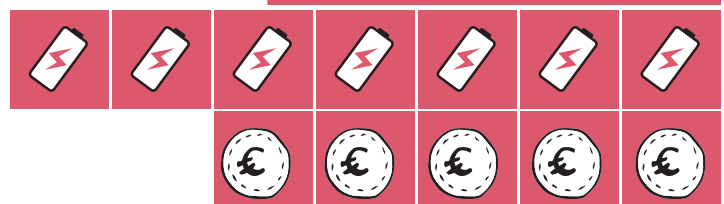
Objekt unter Roboter
Objet sous robot
Object under robot

Hindernis direkt vor Roboter
Obstacle devant robot
Obstacle in front of robot

Gegner direkt vor Roboter
Adversaire devant robot
Opponent in front of robot

Gegner direkt hinter Roboter
Adversaire derrière robot
Opponent behind robot

STARTE "BETA"
LANCER "BETA"
START "BETA"





PROGRAMMIERBLÖCKE LEVEL 3
 BLOCS DE PROGRAMMATION LEVEL 3
 PROGRAMMING BLOCKS LEVEL 3



OLLI

VORWÄRTS AVANCER FORWARD	LINKS DREHEN TOURNER A GAUCHE TURN LEFT	RECHTS DREHEN TOURNER A DROITE TURN RIGHT
VORWÄRTS AVANCER FORWARD	LINKS DREHEN TOURNER A GAUCHE TURN LEFT	RECHTS DREHEN TOURNER A DROITE TURN RIGHT
VORWÄRTS AVANCER FORWARD	LINKS DREHEN TOURNER A GAUCHE TURN LEFT	RECHTS DREHEN TOURNER A DROITE TURN RIGHT
VORWÄRTS AVANCER FORWARD	LINKS DREHEN TOURNER A GAUCHE TURN LEFT	RECHTS DREHEN TOURNER A DROITE TURN RIGHT
RÜCKWÄRTS ARRIÈRE BACKWARD	RÜCKWÄRTS ARRIÈRE BACKWARD	RÜCKWÄRTS ARRIÈRE BACKWARD
WARTE ATTENDRE WAIT	WARTE ATTENDRE WAIT	WARTE ATTENDRE WAIT
WARTE ATTENDRE WAIT	EINSAMMELN COLLECTER COLLECT	EINSAMMELN COLLECTER COLLECT
ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C	ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C	ABLADEN A-B-C DÉCHARGER A-B-C UNLOAD A-B-C
LASER ABFEUERN LANCER LASER FIRE LASER	LASER ABFEUERN LANCER LASER FIRE LASER	STARTE "ALPHA" LANCER "ALPHA" START "ALPHA"

Gegner im Sichtfeld
 Adversaire dans le champ visuel
 Opponent in field of view

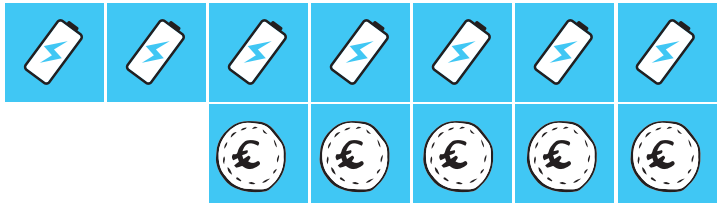
Objekt unter Roboter
 Objet sous robot
 Object under robot

Hindernis direkt vor Roboter
 Obstacle devant robot
 Obstacle in front of robot

Gegner direkt vor Roboter
 Adversaire devant robot
 Opponent in front of robot

Gegner direkt hinter Roboter
 Adversaire derrière robot
 Opponent behind robot

STARTE "BETA"
 LANCER "BETA"
 START "BETA"





MÜLL KARTEN LEVEL 2 & 3
CARTES DÉCHETS LEVEL 2 & 3
WASTE CARDS LEVEL 2 & 3





Feedback MAKERKOFFER - Supervisor

We would be very pleased to receive your feedback! You can fill out the form below and send it by e-mail to info@base1.lu.

Date of the activity				
Age				
Gender	M	W		
Institution	<input type="checkbox"/> Maison relais / Foyer scolaire <input type="checkbox"/> Primary school <input type="checkbox"/> Secondary school <input type="checkbox"/> Youth centre <input type="checkbox"/> Other:			
Role	<input type="checkbox"/> Educator <input type="checkbox"/> Teacher <input type="checkbox"/> Coach, Expert <input type="checkbox"/> Other :			
Makerkoffer	<input type="checkbox"/> Coding <input type="checkbox"/> Wearables <input type="checkbox"/> Electronics			
Level	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3			
Why did you choose the activities of the Makerkoffer?				
Did the Makerkoffer meet your expectations?	<input type="checkbox"/> 😊😊	<input type="checkbox"/> 😊	<input type="checkbox"/> 😐	<input type="checkbox"/> ☹️
What did you like about the activities?				
What was not so interesting?				
What has been difficult to understand?				



Did you learn anything new?	😊😊	😊	😐	😞
Did the activity help you to understand the Makerworld better?	😊😊	😊	😐	😞
Are you interested in researching/making further?	😊😊	😊	😐	😞
Would you do the Makerkoffer activities again?	😊😊	😊	😐	😞
Would you recommend the Makerkoffer to others?	😊😊	😊	😐	😞
Further comments / proposals:				



Feedback MAKERKOFFER - Participant

We would be very pleased to receive your feedback! You can fill out the form below and send it by e-mail to info@base1.lu.

Date				
Age	<input type="checkbox"/> 8-10 <input type="checkbox"/> 11-13 <input type="checkbox"/> 14-16 <input type="checkbox"/> 17-18 <input type="checkbox"/> >18			
Gender				
Institution	<input type="checkbox"/> Maison relais / Foyer scolaire <input type="checkbox"/> Primary school <input type="checkbox"/> Secondary school <input type="checkbox"/> Youth centre <input type="checkbox"/> Other:			
Cycle/Group				
Makerkoffer	<input type="checkbox"/> Coding <input type="checkbox"/> Wearables <input type="checkbox"/> Electronics			
Level	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3			
Did you like the Makerkoffer?				
Did you learn anything new?				
Did you understand the tasks?				
Are you interested in learning more about the theme?				
Would you do the Makerkoffer activities again?				
What else would you like to share with us?				