



DIGIFLUX TEACHER GUIDE – (EV3 ROBOTICS ACTIVITY)

Designing and Programming Animal-Inspired Robots (EV3)



Short Description: Students learn how to program the EV3 robot base to successfully complete the challenges.

Practical information

Learning objectives

- Apply computational thinking principles (algorithms, loops, conditions, variables)
- Design and program a robotic system to solve a problem
- Understand and apply biomimicry (technology inspired by nature)
- Develop critical thinking and problem-solving skills
- Work collaboratively in pairs or small groups

Alignment with Digiflux Project

This activity contributes to:

- Digital STEM learning and integration of digital tools in education
- Scenario-based and problem-based learning
- Development of critical and creative thinking
- Teacher training in digital pedagogy

STSL Learning Outcomes

SCIENCE.STSL.3.2 – Formulating a hypothesis for a design problem

SCIENCE.STSL.3.3 – Transforming a design idea into a working solution

Required materials

- 1x EV3-sensor kit
- 1 x building kit EV3
- 7 x connection cable
- 7 x charger
- 7x EV3-robot: pre-built
- 7x laptops
- 1x First Lego League mat (with pre-built challenges)

Teacher Role

During the activity, the teacher acts as:

- Facilitator of learning
- Guide in problem-solving processes
- Observer of student behaviour and learning

The teacher should:

- Encourage students to explain their choices
- Support debugging without giving full solutions
- Stimulate reflection and iteration

Colour Coding Used Throughout the Guide



=Introduction / Core content



=Differentiation / wrap-up



=Practice / Consolidation



=Additional information for the teacher

How does the EV3 robot work?

Connecting to the computer

The robot is always connected using a cable or bluetooth.

Plug the cable into the PC port on the robot and into the USB port.

Emergency stop / power off

The grey button in the top-left corner functions as an emergency stop. It can also be used to turn off the robot.

Power-on/ start program

The grey square button in the centre turns the robot on.

This button can also be used to start a program

Charging the robot

The robot battery lasts for approximately two half-days. Charging is done at the back of the robot using the charger.



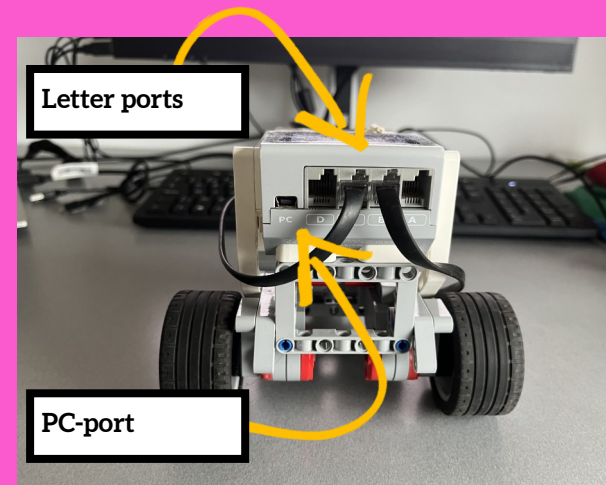
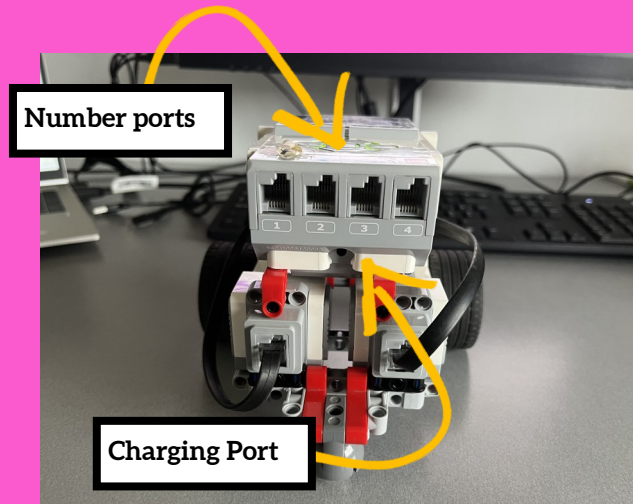
Robot status colours

The robot uses colours similar to a traffic light:

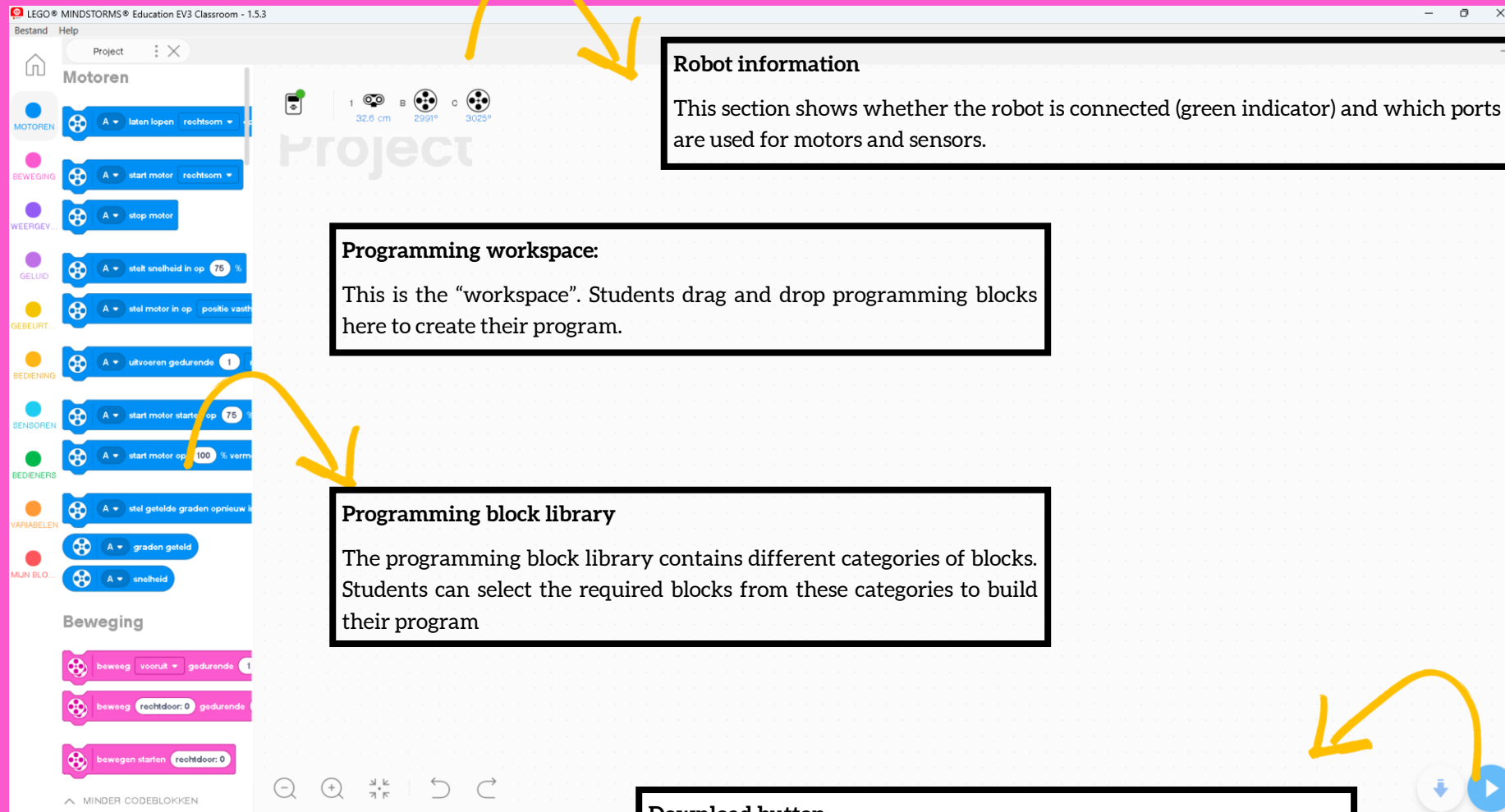
Red = wait, orange = almost ready, green = ready to use

Hardware Guidelines

- Ensure that students work as much as possible on the floor to prevent the robot from falling off tables.
- When connecting components, make sure students: A) connect motors to the letter ports (A-B-C-D) and B) connect sensors to the number ports (1-2-3-4)



The EV3 classroom programming environment

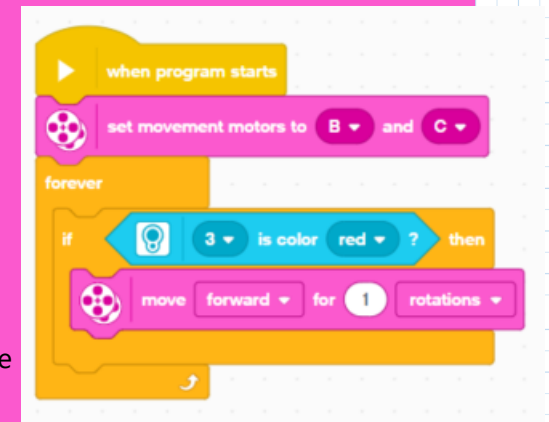


The screenshot shows the LEGO MINDSTORMS Education EV3 Classroom software interface. The interface is divided into several sections:

- Robot information:** This section shows whether the robot is connected (green indicator) and which ports are used for motors and sensors. It includes a green indicator and port numbers (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100).
- Programming workspace:** This is the "workspace". Students drag and drop programming blocks here to create their program.
- Programming block library:** The programming block library contains different categories of blocks. Students can select the required blocks from these categories to build their program.
- Download button:** Always download the program to the robot before running it. Make sure to remember the project name (shown in the top-left corner).

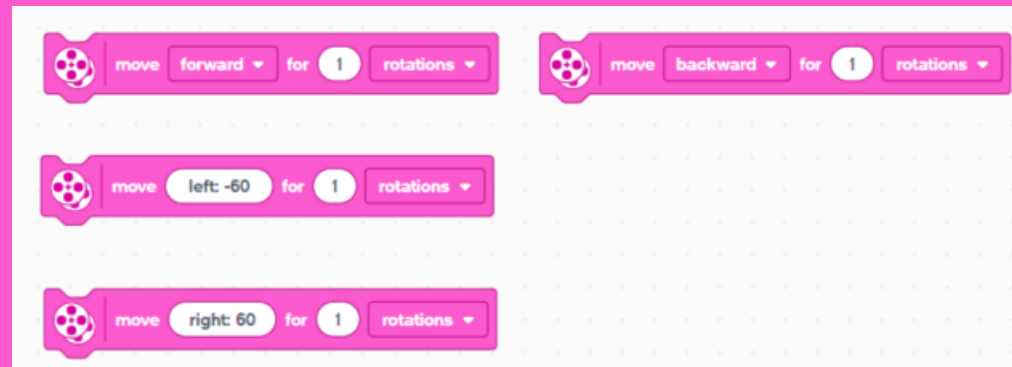
Software guidelines

- The blue **motor** programming blocks are used to control additional motors connected to the robot.
- The pink **movement** programming blocks allow the robot to move.
- The colour sensor works best in darker environments and when detecting the colours of LEGO bricks.
- When programming the pink and blue blocks, students can choose between using **seconds** or **rotations**.
- Always make sure that when students modify their program, they download the updated version to the robot again.
- Ensure that the correct ports are used when programming. For example, if a sensor is programmed to be connected to port 3, but it is not actually connected to that port, the robot will not function.



Basic movement blocks

- Forward / backward movement:
- Turn left (90° angle):
- Turn right (90° angle):





15 min

Getting to Know EV3: Hardware and Software

Whole Class – 15 min

In this project, students will take on the role of **animal caretakers in a futuristic biodiversity reserve**.

In this reserve, the genetic code of all animals is stored in robots, forming a kind of “animal library”.

Students will become animal caretakers and learn how to program these robotic animals.

As part of this introduction, students will receive basic training on how to work as robotic animal caretakers.

Introduction to Hardware and Software

Explain to the students that a robot consists of two main components:

1. **Hardware** – all physical parts of the robot
2. **Software** – the program that controls the robot’s behaviour

Exploring the Hardware

Ask students to observe the robot and identify its most important components.

1. The Hub

The hub sends the program to the different components of the robot. It also provides power to the robot.



2. Large Motors

The two large motors allow the robot to move.

3. Ultrasonic Sensor

The ultrasonic sensor is used to measure the distance to an object. It does this by sending out sound waves and measuring the time it takes for the waves to return.

It works in a similar way to a bat, which uses echolocation to detect objects in its environment.

This is an example of biomimicry: technology inspired by natural systems and animal behaviour.

4. Colour Sensor

A colour sensor is used to detect the colour of a surface.

It works by shining light onto the surface and measuring the reflected light waves.

Based on the intensity of different colours in the reflected light, the sensor can identify the colour of the surface.

Many animals, like humans, have systems in their eyes that function similarly to a colour sensor, allowing them to perceive different colours using photoreceptors and other sensory mechanisms.

This is an example of biomimicry: technology inspired by nature.



5. Touch Sensor

A touch sensor measures the pressure applied to a surface.

Animals are able to detect pressure changes and use this information for navigation, hunting and survival in their specific environments.



This is an example of biomimicry: technology inspired by natural systems and animal behaviour.



Note: Sensors must be connected to the number ports (1-2-3-4), while motors must be connected to the letter ports (A-D).



Activity 1: Bringing a Robot Animal to Life

In Pairs – 15 min

Students use the programming library to create a face for their robot using the **display** and **sound** blocks.

Demonstrate to the whole class how to download the program to the robot using the cable or bluetooth.

An example program is shown on the right.



In pairs

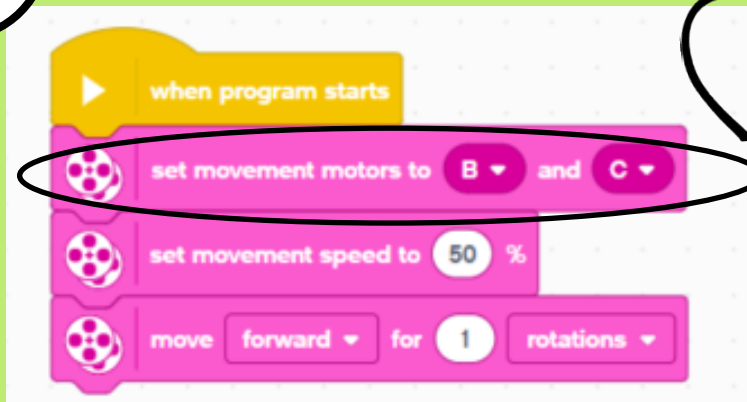
30 min

Activity 2: Making the robot animal move

The teacher places the necessary movement blocks in the programming workspace.

Students must arrange these blocks in the correct order to make the robot move.

At the end of the activity, students present their robot animals to the whole class.



If students do not include this block, the robot will not move.

Make sure that the motors are connected to **ports B and C**.

Activity 3: Building a Rhinoceros Robot

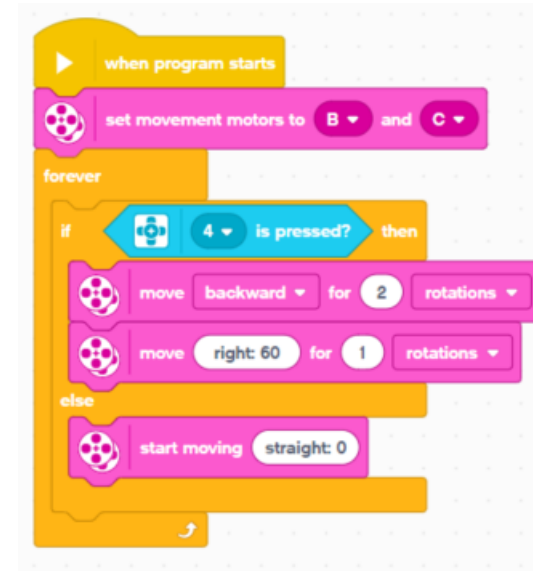
Students rebuild their robot into a rhinoceros robot using the touch sensor.

The teacher places the required programming blocks for the rhinoceros in the programming workspace.

Sensor Task

Example program

Touch sensor Program a rhinoceros robot that moves backwards when it hits a wall.



Activity 4: Bat

In pairs

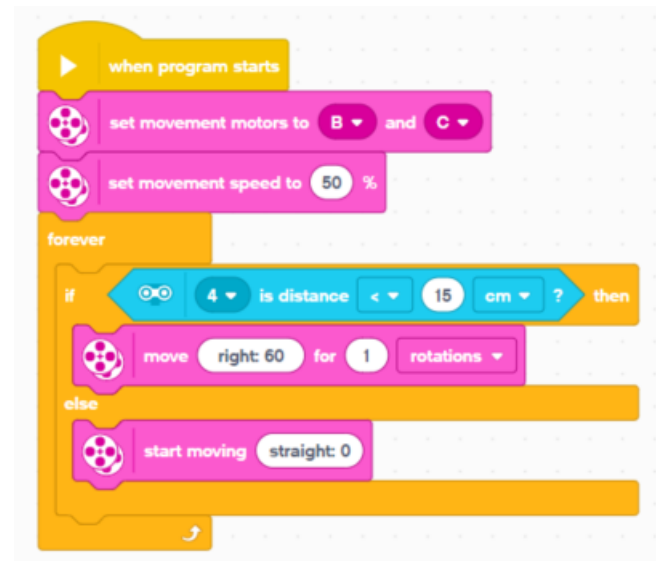
35 min

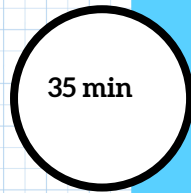
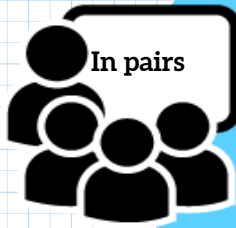
The teacher briefly reviews how the distance sensor works. Students then build and program a bat robot. The teacher places the necessary programming blocks for the bat in the programming workspace. Students must arrange these blocks in the correct order.

Sensor Task

Example program

Distance sensor
Program a bat that turns when it reaches a wall.





Missions on the FLL-mat

The teacher reviews the challenges on the mat together with the students.

Students then decide as a group how many challenges they aim to complete during the session (recommended: 12–15).

The teacher writes this number on the board.

Each time a challenge is completed, the total can be updated.

The goal is for the group to solve as many challenges as possible together.

Note:

The mat may not be complete. Check which challenges are available before starting the activity.

Tips

- The maximum speed on the mat is 70%. Make sure students are always ready to catch the robot.
- Only one group is allowed on the mat at a time. Building and programming take place at the laptop.
- Students who want an additional challenge can program the sensors more functionally or complete multiple challenges in sequence.
- The mat may curl at the edges. You can fix this by taping it down underneath.

Basic Rules

- The robot starts in the protractor area in the bottom-left corner (LEGO logo), behind the black starting line.
- Once a challenge is completed correctly, the team earns one point. The robot does not need to return to the starting position.

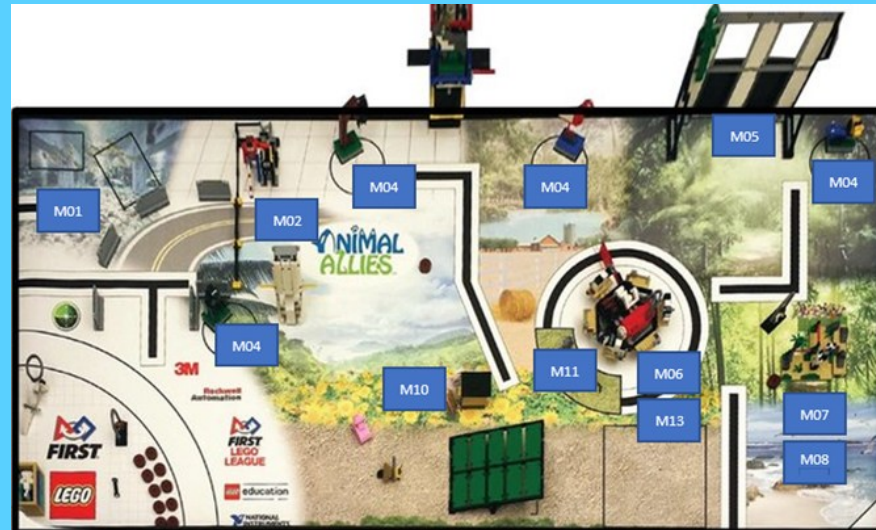
Mission overview

Scan the QR code for an overview of the missions.



Note:

The content of some missions has been simplified in this guide.



Building the Challenges

Students work in pairs and choose a mission.

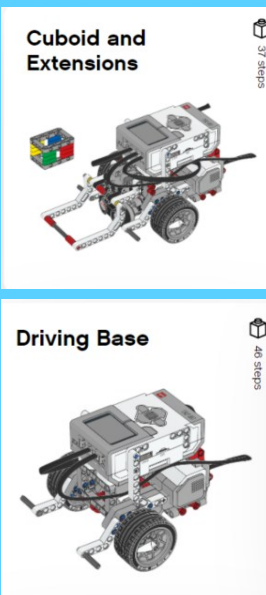
They build additional components for their robot, for example the **medium motor driving base** or their own design.

After building, they create a program.

Each program may be tested multiple times on the mat.

The teacher stays near the mat and provides feedback after each test.

All building instructions can be found in the software under **"Building"**.



Programming the challenges

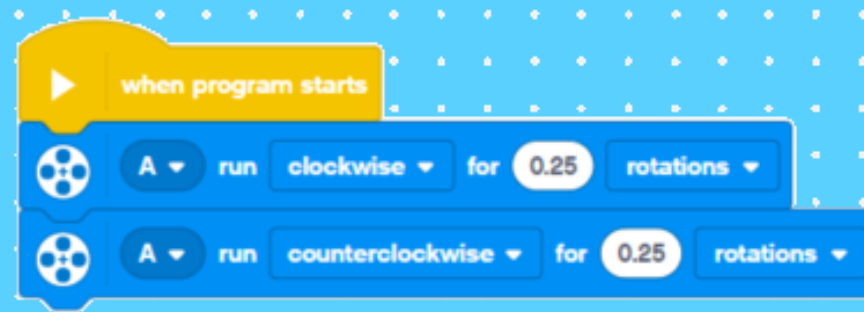
Objective: Driving

Students use the pink programming blocks to control the movement of the robot, as in the previous activities.

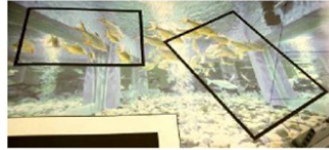
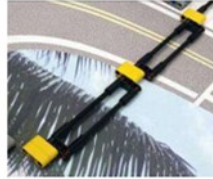
Objective: Lifting the Arm

The teacher demonstrates this program to the whole class and lets the students test it.

Make sure that the additional motor is connected to port A.



Overview of Programming Challenges

Num ber	Goal	Image	Difficulty
M01	Move the shark to its new home.		Easy
M02	Drive over the black barrier with the robot to activate the dog.		Medium

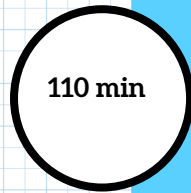
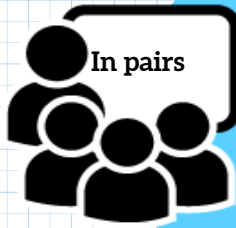
Activity 5: Chameleon

In pairs

35 min

The teacher briefly reviews how the colour sensor works. Students then build and program a chameleon robot. The teacher places the necessary programming blocks for the chameleon in the programming workspace. Students must arrange these blocks in the correct order.

Sensor	Task	Example program
Colour sensor	Program a chameleon that shows a different emotion for each colour (using sound and the display). This behavior mimics how chameleons respond to changes in their environment.	



Missions on the FLL-mat

The teacher reviews the challenges on the mat together with the students.

Students then decide as a group how many challenges they aim to complete during the session (recommended: 12–15).

The teacher writes this number on the board.

Each time a challenge is completed, the total can be updated.

The goal is for the group to solve as many challenges as possible together.

Note:

The mat may not be complete. Check which challenges are available before starting the activity.

Tips

- The maximum speed on the mat is 70%. Make sure students are always ready to catch the robot.
- Only one group is allowed on the mat at a time. Building and programming take place at the laptop.
- Students who want an additional challenge can program the sensors more functionally or complete multiple challenges in sequence.
- The mat may curl at the edges. You can fix this by taping it down underneath.

Basic Rules

- The robot starts in the protractor area in the bottom-left corner (LEGO logo), behind the black starting line.
- Once a challenge is completed correctly, the team earns one point. The robot does not need to return to the starting position.

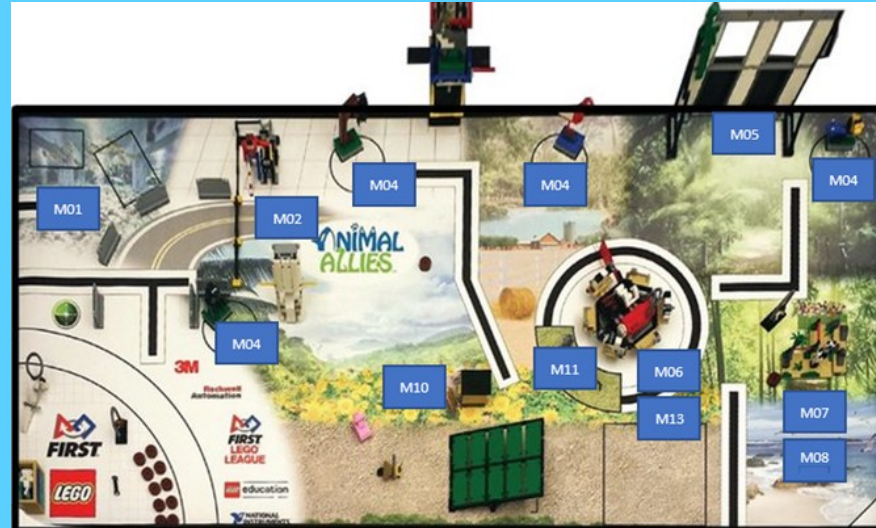
Mission overview

Scan the QR code for an overview of the missions.



Note:

The content of some missions has been simplified in this guide.



Building the Challenges

Students work in pairs and choose a mission.

They build additional components for their robot, for example the **medium motor driving base** or their own design.

After building, they create a program.

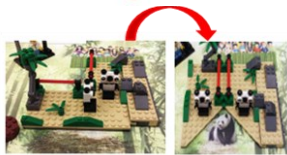
Each program may be tested multiple times on the mat.

The teacher stays near the mat and provides feedback after each test.

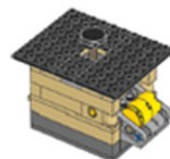
All building instructions can be found in the software under **"Building"**.



Overview of Programming Challenges

Num ber	Goal	Image	Difficulty
M04	Move the animals and the coloured food to the feeding areas using the robot.		Easy
M06	Turn the machine to remove the waste.		Hard
M07	Transform the panda scene from controlled care and observation to a natural wilderness environment.		Hard

Overview of Programming Challenges

Num ber	Goal	Image	Difficulty
M05	Place the gecko on the wall.		Hard
M08	Bring the animal caretaker to the base camp.		Medium
M09	Place the beehive and collect the honey.		Medium



DIGIFLUX TEACHER GUIDE – (MICRO:BIT ACTIVITY)

Designing Interactive Systems using Micro:bit



Students act as digital designers who develop interactive systems using the Micro:bit. They design and program games, circuits and interactive solutions that respond to user input and environmental conditions.

Practical information

Learning objectives

- Apply computational thinking using Micro:bit programming
- Understand input and output systems (sensors and actuators)
- Design and test interactive digital systems
- Apply basic principles of electricity and circuits
- Develop creative solutions through game design

Alignment with Digiflux Project

This activity contributes to:

- Digital STEM learning and integration of digital tools in education
- Scenario-based and problem-based learning
- Development of critical and creative thinking
- Teacher training in digital pedagogy

STSL Learning Outcomes

SCIENCE.STSL.1.1 – Understanding the role of science and technology in society

SCIENCE.STSL.1.3 – Using digital technologies to investigate and solve problems

SCIENCE.STSL.2.1 – Understanding how systems respond to inputs and produce outputs

SCIENCE.STSL.2.2 – Analysing interactions within technological systems

SCIENCE.STSL.3.2 – Formulating a hypothesis or solution for a design problem

SCIENCE.STSL.3.3 – Transforming ideas into a working product or prototype

Assessment

Students are evaluated based on:

- Functionality of their programs
- Understanding of input and output systems
- Creativity in design
- Problem-solving approach
- Collaboration

Teacher Role

During the activity, the teacher acts as:

- Facilitator of learning
- Guide in problem-solving processes
- Observer of student behaviour and learning

The teacher should:

- Encourage students to explain their choices
- Support debugging without giving full solutions
- Stimulate reflection and iteration

Colour Coding Used Throughout the Guide



=Introduction / Core content



=Differentiation / wrap-up



=Practice / Consolidation



=Additional information for the teacher

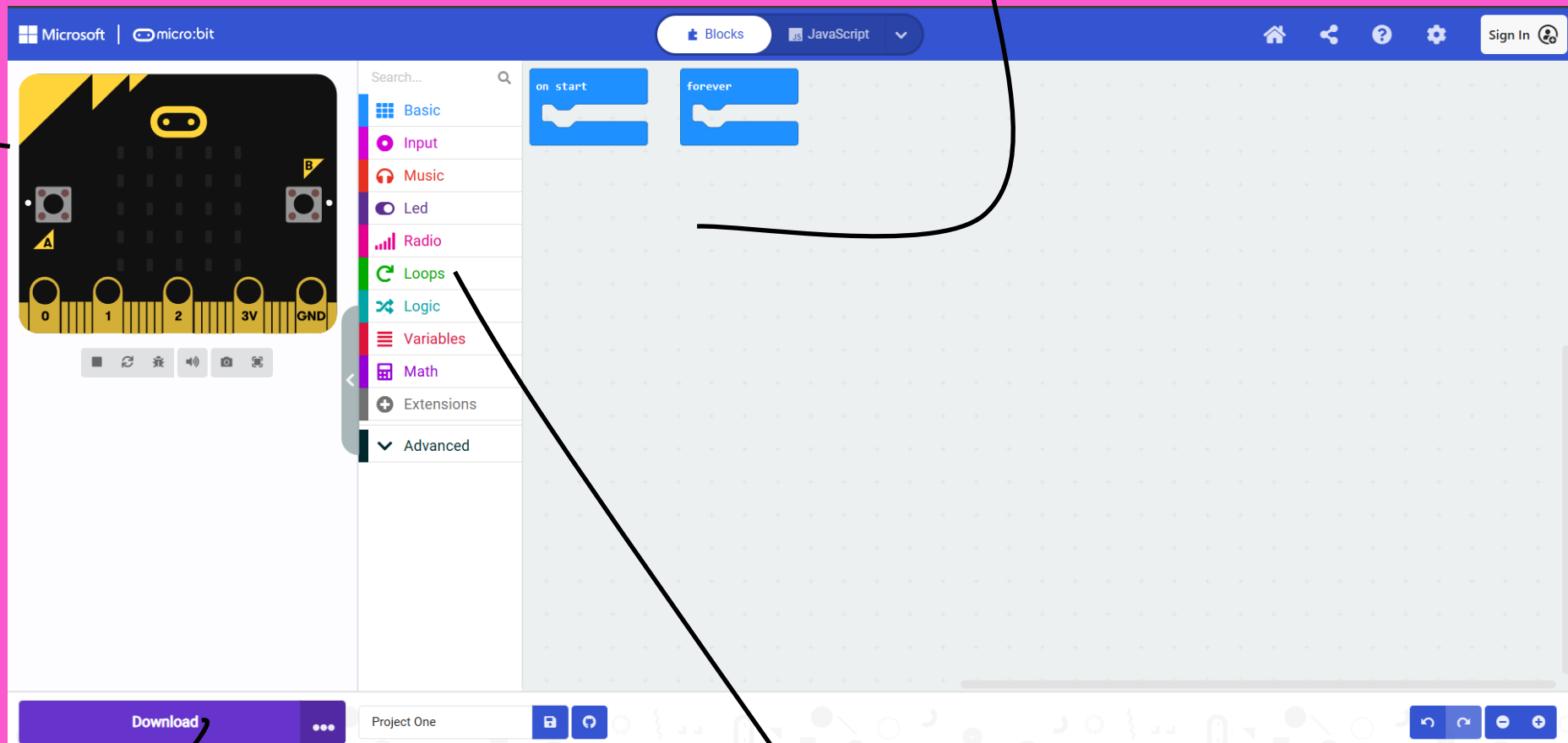
Introduction to the Micro:bit

Simulator

Students can test their program here before downloading it to the Micro:bit. The buttons on the simulator are functional.

Workspace

This is the workspace where students drag and drop programming blocks to create their program.



Download:

By clicking the download button, the program is transferred to the Micro:bit.

Programming Block Library:

Students can find different categories of programming blocks here. They can click on the categories to expand them and access the blocks.



10 min

Introduction to the Micro:bit

Activity 1: Exploring the micro:bit hardware

Each student receives a Micro:bit, a notebook and a pen. The teacher explains what the Micro:bit is: A microcontroller, a small programmable device used to control other systems and machines.

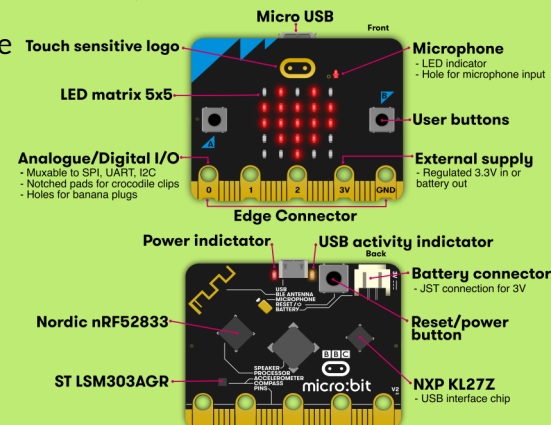
Students are given 3 minutes to explore the Micro:bit and write down:

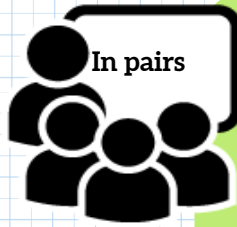
- what they notice
- - what questions they have

The teacher then discusses the students' observations and introduces the key hardware components.

The teacher also explains the difference between:

- Input components (provide information to the device)
- - Output components (display or produce information such as text, sound or images)



 10 min

Introduction to the Micro:bit

Activity 2: Exploring the micro:bit software

Students work in pairs at a laptop using their Micro:bit.

The younger student starts as the “driver”. The teacher explains the two roles:

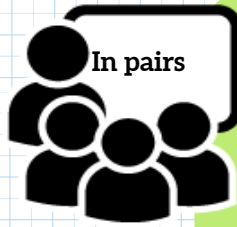
- Driver: controls the laptop and types
- Navigator: gives instructions to the driver

Students open a new program in the software and explore the environment for 3 minutes.

They write down:

- what they discover
- what questions they have

Students then switch roles and repeat the activity for another 3 minutes. The teacher asks: “What did you discover?” “What questions do you have?” The teacher then provides a short introduction to the software, including: - programming blocks - the simulator - how to download a program to the Micro:bit



In pairs

15 min

Introduction to the Micro:bit

Activity 3: Getting to Know Each Other

Students create their first program by programming their name on the Micro:bit. The teacher prepares the program blocks on the board (not in the correct order).

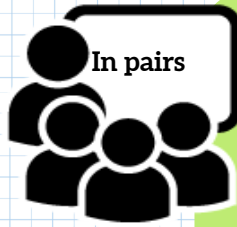
Students must arrange the blocks in the correct sequence and download the program to their Micro:bit.

Each student programs their own name on their device. The teacher demonstrates how to connect the external battery.

All Micro:bits are collected and then randomly redistributed to the students.

Students have 2 minutes to find the correct owner of the Micro:bit without reading the name or showing the screen.

They can only describe what they see (for example: number of letters, first letter, etc.). Important: When removing the external battery, do not pull too hard. Demonstrate this to the whole class.



In pairs

15 min

Introduction to the Micro:bit

Activity 3: Getting to Know Each Other

Students create their first program by programming their name on the Micro:bit. The teacher prepares the program blocks on the board (not in the correct order).

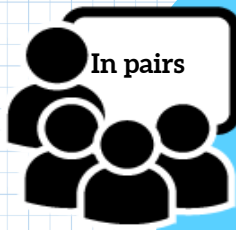
Students must arrange the blocks in the correct sequence and download the program to their Micro:bit.

Each student programs their own name on their device. The teacher demonstrates how to connect the external battery.

All Micro:bits are collected and then randomly redistributed to the students.

Students have 2 minutes to find the correct owner of the Micro:bit without reading the name or showing the screen.

They can only describe what they see (for example: number of letters, first letter, etc.). Important: When removing the external battery, do not pull too hard. Demonstrate this to the whole class.



30 min

Activity 4: introduction tasks

The teacher projects the introduction tasks on the board.

Students work in pairs and complete the following programming tasks. They take turns in their roles (driver and navigator).

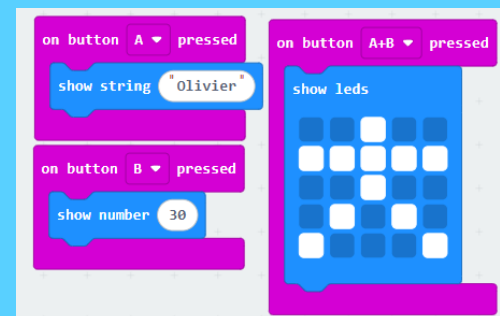
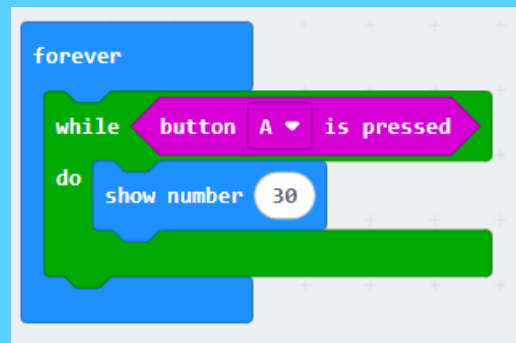
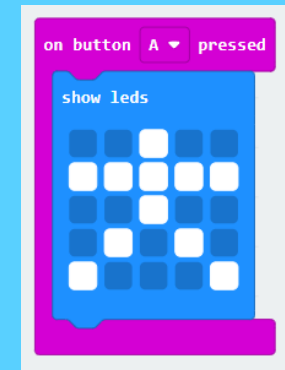
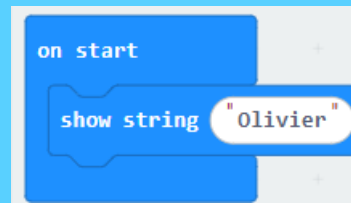
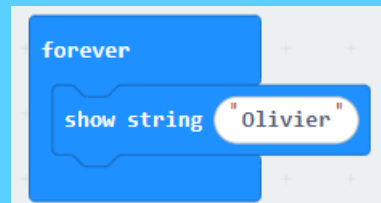
Each student programs the tasks on their own Micro:bit and shows the result using the external battery.

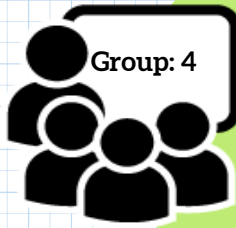
Tasks:

1. When the Micro:bit starts, it continuously displays my name.
2. When the Micro:bit starts, it displays my name once.
3. When button A is pressed, it shows an icon related to my hobby.
4. As long as button A is pressed, it shows my age.
5. When button A is pressed, it shows my name. When button B is pressed, it shows my age. When buttons A and B are pressed together, it shows the icon.

The teacher observes student progress and provides support where needed.

The examples below are functioning solutions for each task





Activity 5: Station Work – Micro:bit Games

Students work in groups of four at one laptop.

Each station contains:

- a partially completed program
- a specific task Within each group.

students take on two roles:

- Programming team: builds and programs the solution
- Execution team: tests and plays the game

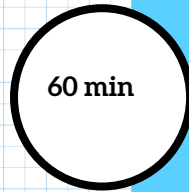
Students switch roles regularly. Each group works for 15 minutes at a station before moving to the next one.

The programming team:

- combines the given blocks
- downloads the program to the Micro:bits (using the external battery)

Afterwards, the group plays the game together, guided by the execution team. At the end of each rotation, students reset the program using the answer sheet for the next group. The teacher explains all stations at the start of the activity.

Important: At the Twister station, the teacher explains the concept of a variable (a piece of memory that stores and changes data).



Game 5.1: Bingo

Task

When pin 0 is touched, the Micro:bit counts down from 3 to 1 and then shows a random number between 0 and 9.

Game Description

Each student selects a bingo card and fills in numbers between 0 and 9.

Players take turns touching pin 0.

Each time, a random number appears on the Micro:bit.

Students cross off the number if it appears on their card.

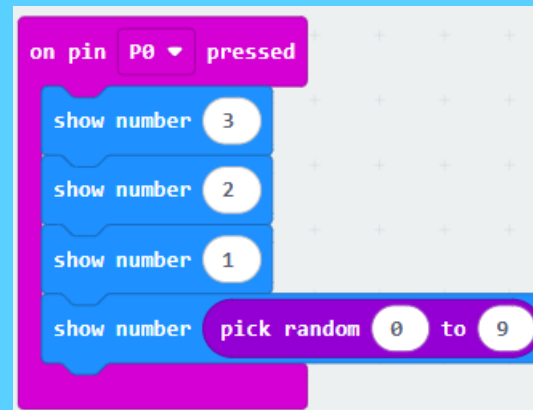
If the same number appears multiple times, it can only be crossed off once.

The first player to complete their card wins.

Materials

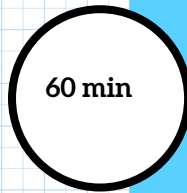
- laminated bingo cards
- whiteboard markers
- cloth for erasing

Example





Group: 4



60 min

Game 5.2: Twister

Task

Micro:bit 1: When shaken, it displays a random colour (red, green, yellow, blue).

Micro:bit 2: When shaken, it displays a random body part (left hand, left foot, right hand, right foot).

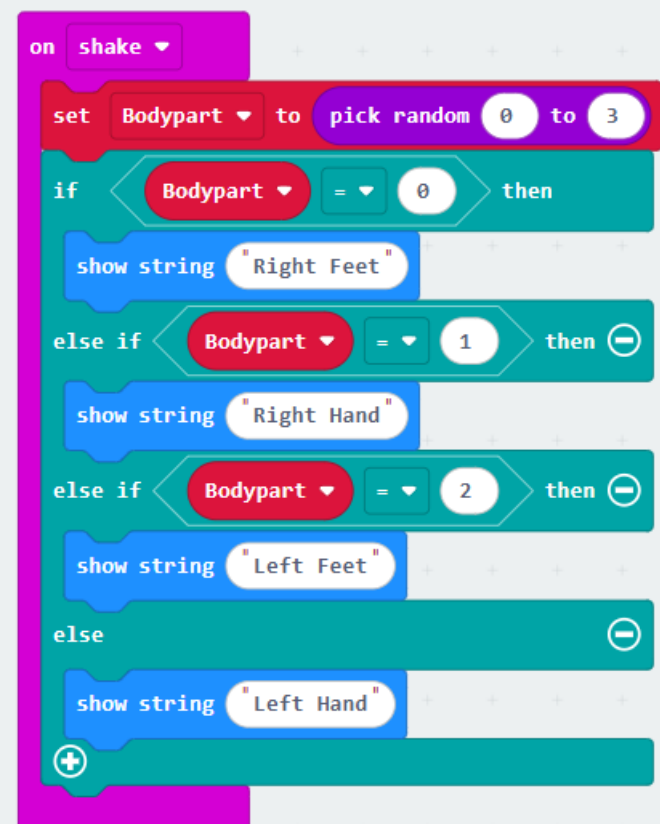
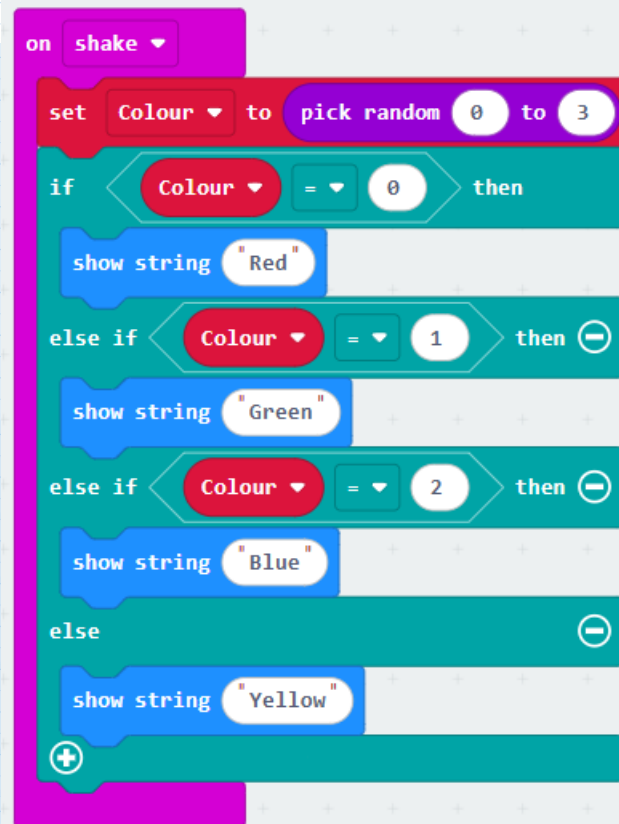
Game Description

One student shakes the Micro:bits. The other students follow the instructions. The student who first touches a “white” space must switch roles with the player who is shaking.

Materials

- Twister mat

Examples

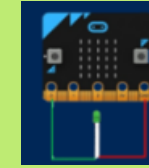




15 min

Advanced: Circuit Design and Boardgames

Students build a basic electrical circuit (see example).



The teacher explains the correct setup and highlights that the **anode (positive side, long leg)** of the LED must be connected to pin 0. Otherwise, the LED may break.

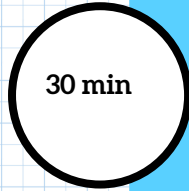
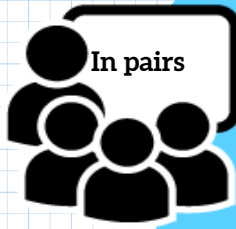
Students are given 3 minutes to observe the circuit and write down:

- what they notice
- what questions they have

The teacher then discusses the students' questions and explains the key concepts.

Key Concepts Discussed

- **Components of a circuit** Battery, LED, wires
- **Role of the Micro:bit** Works as a switch controlling the flow of electricity
- **Energy source** The battery provides energy and moves electrons
- **Polarity** The positive (+) and negative (–) sides of components
- **Flow of electricity** Electrons move through conductors from negative to positive
- **Materials** Conductors (e.g. metal wires) allow electricity to pass Insulators (e.g. plastic) do not allow electricity to pass
- **Closed vs open circuit** Electricity only flows in a closed circuit



Advanced: Circuit Design and Boardgames

Students work in pairs using a circuit challenge booklet.

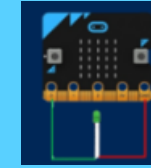
They complete the challenges in the booklet.

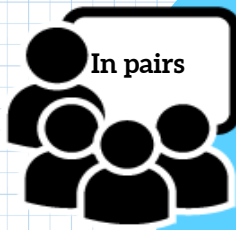
Students can choose between:

- basic challenges (blue—blauw)
- advanced challenges (red –rood)

The advanced blocks can be found under: “Advanced” → “Pins” in the programming environment.

The answers for the challenge booklets can be found on the next page.





30 min

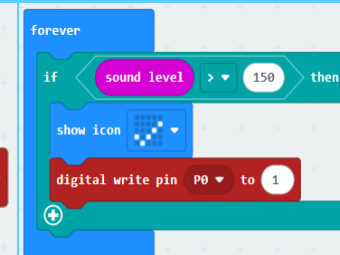
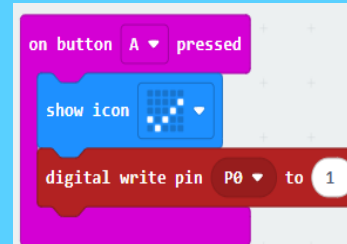
Advanced: Circuit Design and Boardgames

Challenge

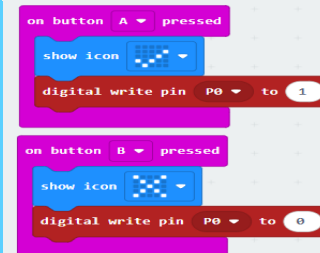
Basic

Advanced

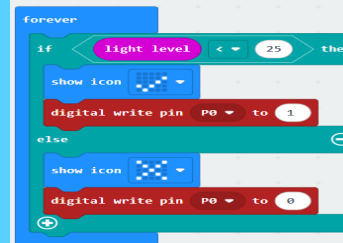
1



2



3



4



The LED lights are connected in parallel, as shown in the image.

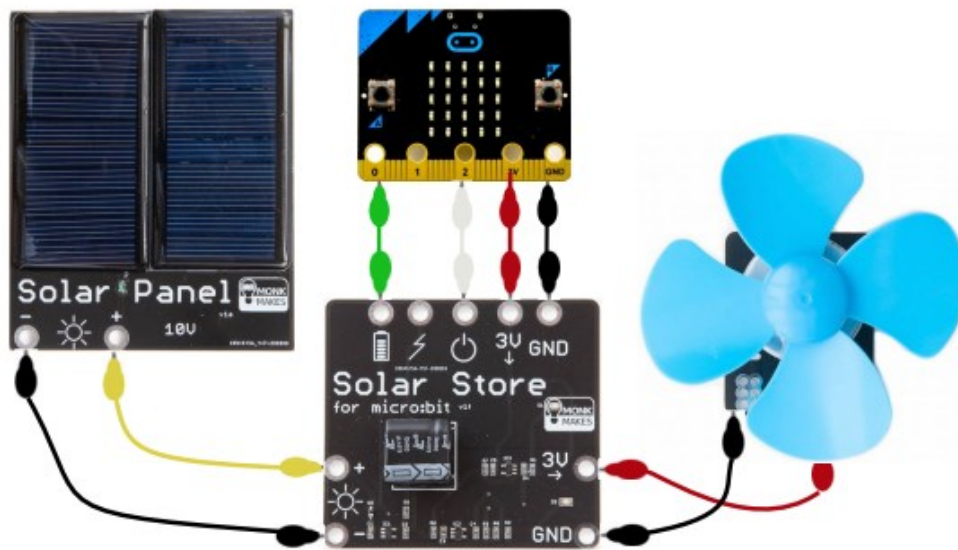
All conductive materials can be used, for example: all metals.

Different solutions are possible, for example using a switch as shown in the image.



MICROBIT SUNCATCHER

Students explore solar, wind and water energy by using the Micro:bit to measure light and create interactive systems.



Short description

Working with solar energy and the Micro:bit

- Introduction to the Micro:bit sensor
- Introduction to solar panels and bar charts
- Designing your own solar panel system using Micro:bit and LEGO Education SPIKE prime

Practical information

Learning objectives

- SCIENCE.STSL.1.1 – Understanding the role of science and technology in society
- SCIENCE.STSL.1.3 – Using digital technologies to investigate and solve problems
- SCIENCE.STSL.2.1 – Understanding how systems respond to inputs and produce outputs
- SCIENCE.STSL.2.2 – Analysing interactions within technological systems
- SCIENCE.STSL.3.2 – Formulating a solution for a design problem
- SCIENCE.STSL.3.3 – Developing a working prototype

Materials

- 7 laptops
- 7 Micro:bits with accessories
- 7 Micro:bit introduction workbooks
- 7 puzzle sets of solar panel setups
- 7 solar experiment kits for Micro:bit
- 7 experiment worksheets
- 7 large batteries
- 1 box of crocodile clips
- 7 LEGO Education SPIKE Prime kits
- 1 extra SPIKE Prime kit
- A4 sketch paper bundle
- Pencils, sharpeners and erasers
- 7 cardboard plates (approx. size A3)

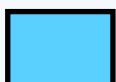
Visual legend



=Introduction and core content



=Differentiation and wrap-up



=Practice and consolidation

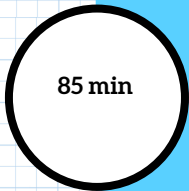
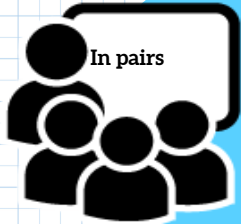


=Additional information for the teacher

Workshop flow

Preparation

Ensure that students work in pairs and that each pair has: one Micro:bit with accessories and one laptop with the software opened.



Creating the program

Introduction

During this activity, students focus on the **light sensor** of the Micro:bit:

- where to find it
- how it works
- how to use it in a program

The LEDs on the Micro:bit can function not only as an output device, but also as an input device (light sensor) that measures light intensity.

This means that Micro:bit programs can respond differently depending on how light or dark it is.

Step 1: Test Program

Before starting the activity booklet, the teacher creates a test program together with the students.

Students program their team name on the LED display of the Micro:bit.

The characters appear one by one on the screen.

During this activity, extra attention is given to basic programming skills:

- adding blocks
- dragging and positioning blocks correctly
- deleting blocks
- downloading the program to the Micro:bit

Step 2: Working with Pseudocode

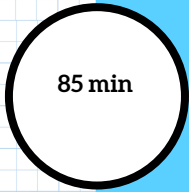
In the activity booklet, students find pseudocode that describes the steps of the program they need to create.

Some words are highlighted in different colours.

Each colour corresponds to a specific programming block.

Students translate the pseudocode into a working program.

Workshop flow



Step 3: Puzzle Element

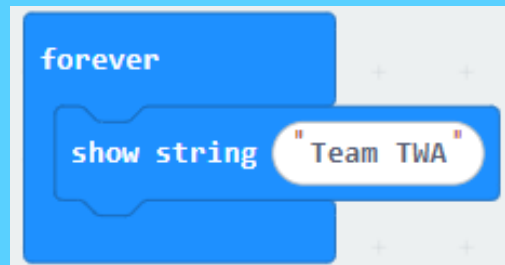
When students successfully complete a program, they receive a puzzle piece.

They will need these pieces later when designing their own solar panel system.

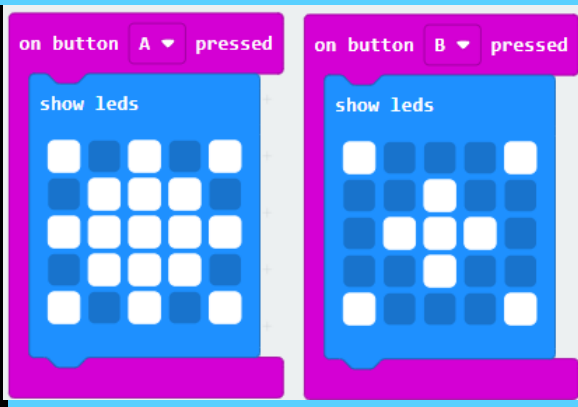
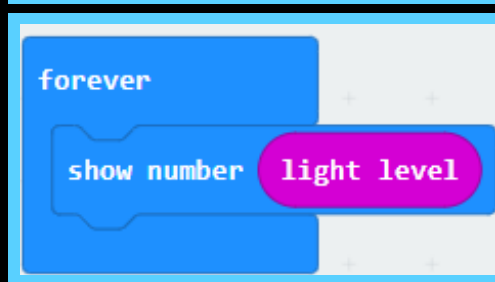
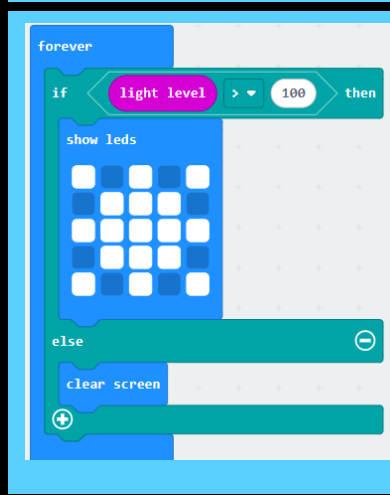
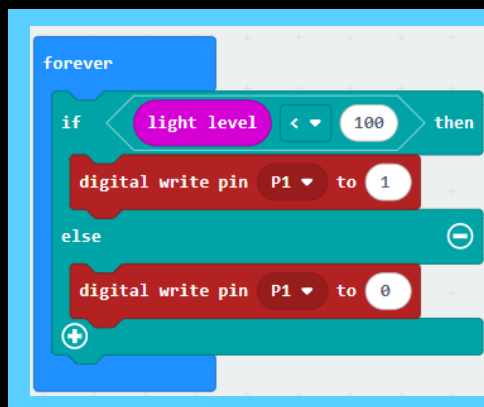
Differentiation

If students find the challenge too difficult, they can ask for a hint.

They will then receive an image showing the programming blocks they need to complete the task.



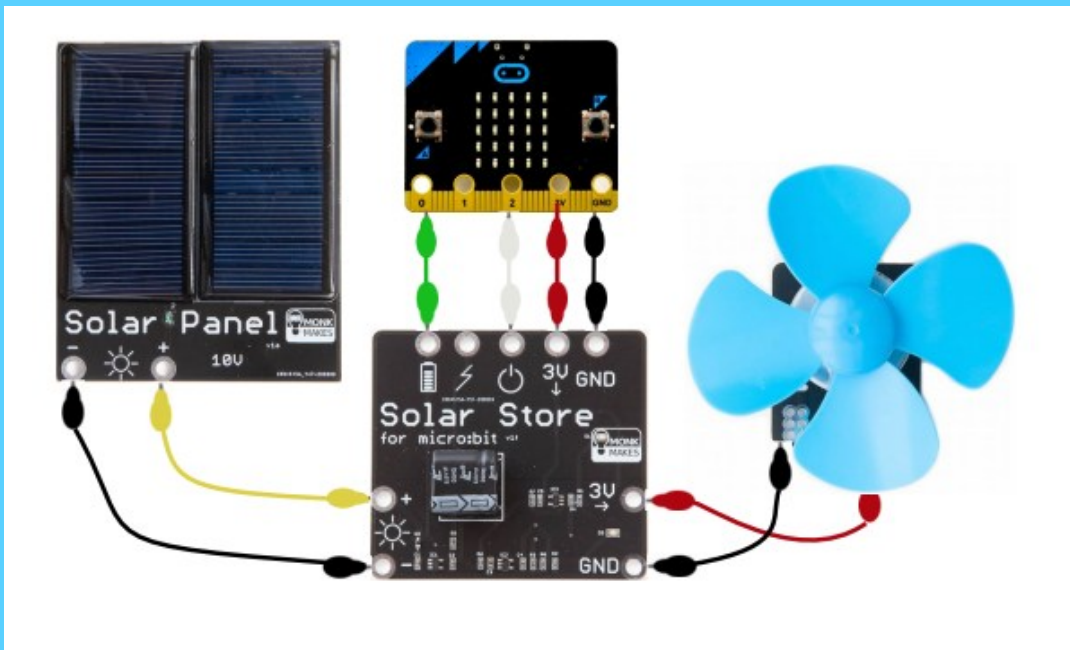
Solutions

1		
2		
3		• If it doesn't work, try changing the light level code to light level > 50, 60, ...
4		



Building the solar system puzzle

Students now have collected all puzzle pieces needed to complete the system. The teacher shows the final system configuration (see example image).



Construction Phase

Students build the setup using their own Micro:bit.

They construct the system on a solid piece of cardboard to make it easier to move.

Make sure that all crocodile clips are correctly and securely connected.

Understanding the System

The teacher discusses the different components with the students:

- **Solar panel** Used to collect solar energy
- **Solar energy storage (battery)** Stores the collected solar energy for later use
- **Fan (output device)** Acts as a consumer of energy. If enough energy is stored, the fan rotates. If there is not enough energy, the fan stops.
- **Micro:bit** Used to monitor how much energy is being collected. This is measured by detecting how quickly the battery charges.

Conclusion

Before analysing the system, students will need to create a program to visualise and interpret the collected data.



Programming the solar panel

Students create a program with the Micro:bit to visualise the amount of energy generated by the solar panel.

The teacher builds the first part of the program together with the students.

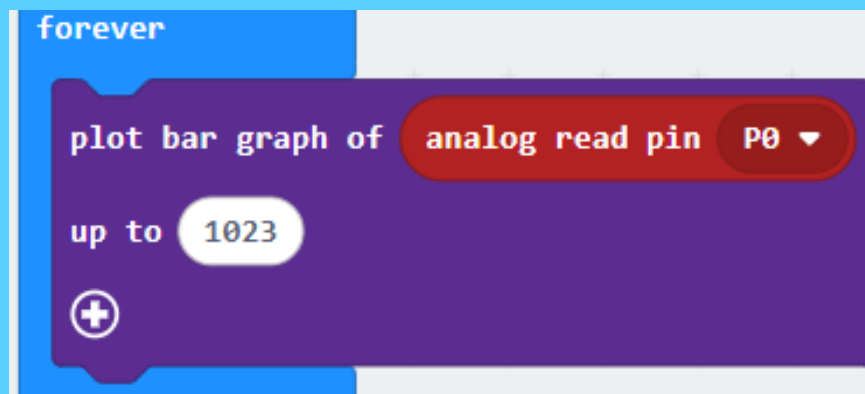
Tip:

The bar graph block can be found under the LED category.

Step 1: Completing the Program

Students complete the program independently. They must determine which pin (P) is connected to the energy storage module (battery).

Students should carefully analyse their physical setup. **Correct answer:** Pin 0 (P0)

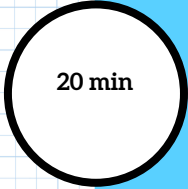


Step 2: Understanding the Data

The program uses the value **1023**, which represents the maximum value that can be displayed in the graph of energy generated by the solar panel.



In pairs



20 min

Programming the solar panel

Step 3: Using the Data Simulator

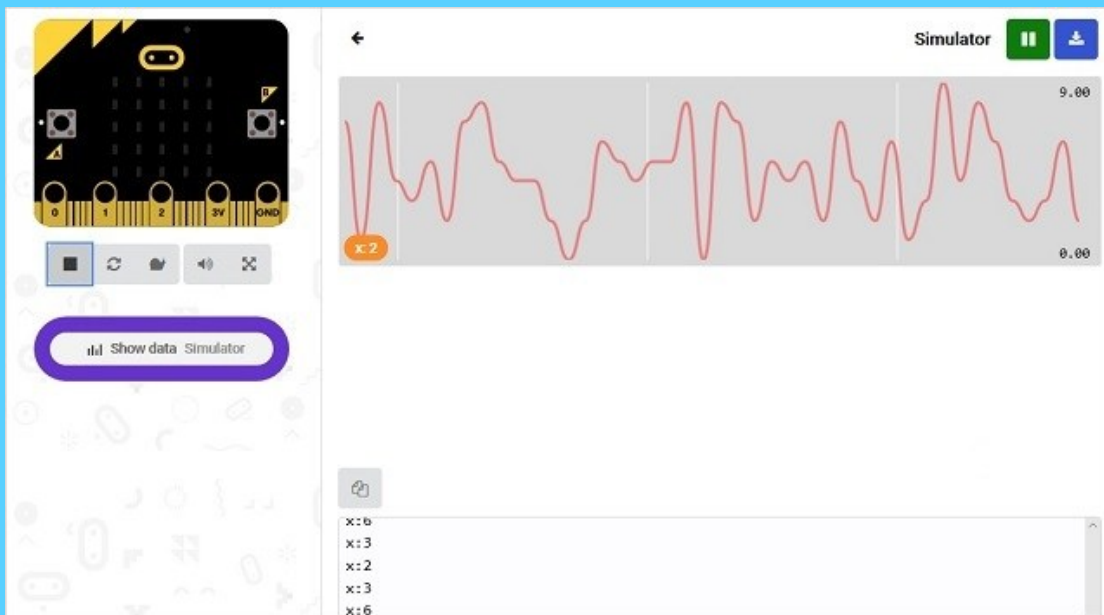
After downloading the program to the Micro:bit and connecting the device, students can open the **"Show Data Simulator"**.

This allows them to:

- observe a live graph
- analyse how quickly the battery charges
- understand how solar energy input affects the system

Conclusion

Students interpret the data and understand how light intensity influences the amount of energy generated by the solar panel.





In pairs

35 min

Controlling the solar panel system

Students extend their existing program to control the fan using the Micro:bit.

The teacher introduces the task using pseudocode and explains it step by step.

Step 1: Understanding the Pseudocode

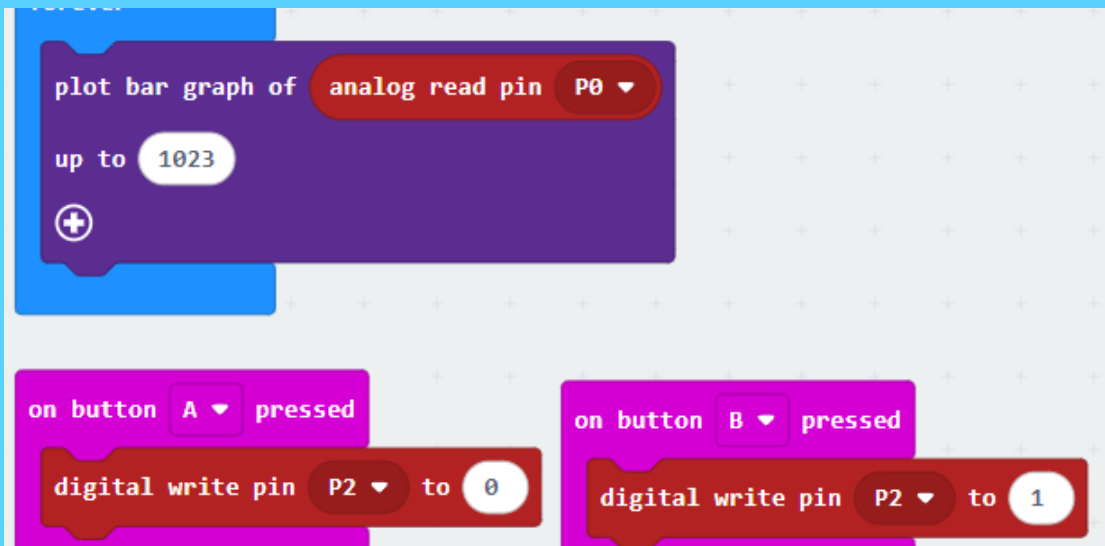
Pseudocode:

- When button A is pressed, the fan turns on
- When button B is pressed, the fan turns off

The teacher displays the pseudocode on the board and discusses it with the students.

Step 2: Building the Program

Students translate the pseudocode into a working program.



They must determine which pin is connected to the fan.

If students find this difficult, the teacher can provide the individual programming blocks as a hint.

Step 3: Testing the Full System

Students complete the full program, including:

- the bar graph (light measurement)
- the fan control (on/off)

Students test their setup and observe how the system behaves.

Step 4: Experimentation

Students carry out experiments using their setup, following the instructions on the worksheet.

Experiment sheet

Step 1: Charging

- Place the solar panel in the sunlight and observe how long it takes to fully charge
- Write down what you see on the display when the battery is fully charged. (Tip: in bright sunlight, the battery charges faster).

Step 2: Discharging

- Disconnect the solar panel
- Press button B so that the fan starts running.
- Measure how long it takes before the fan stops.
- Write down what you see on the display when the fan stops.

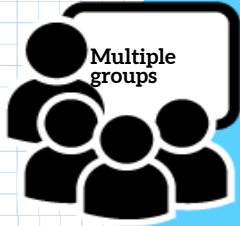
Step 3: Charging while the fan is running

- Try to charge the system while the fan is still running
- Use bright light
- Measure how long it takes to fully charge the system while the fan is running

The bar graph works best when the Micro:bit has new batteries or is connected to a computer via USB. Low batteries may result in incorrect measurements.

Measurement table

Test	Time (minutes)	What do you observe?	Location of the experiment
1. Charging			
2. Discharging (fan stops)			
3. Charging while fan is running			



Multiple
groups

25 min

Analysing and discussing results

Step 1: Comparing Results

Students compare their experimental results with other groups.

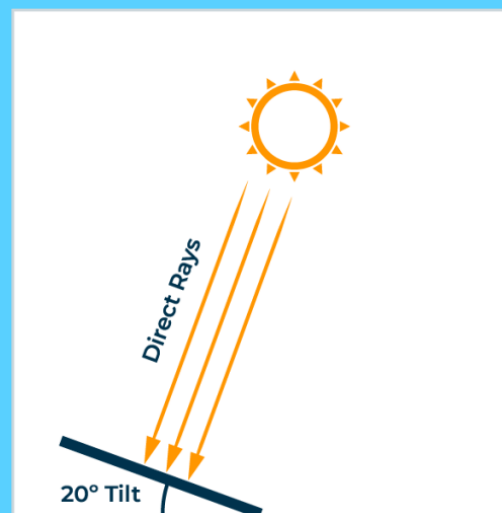
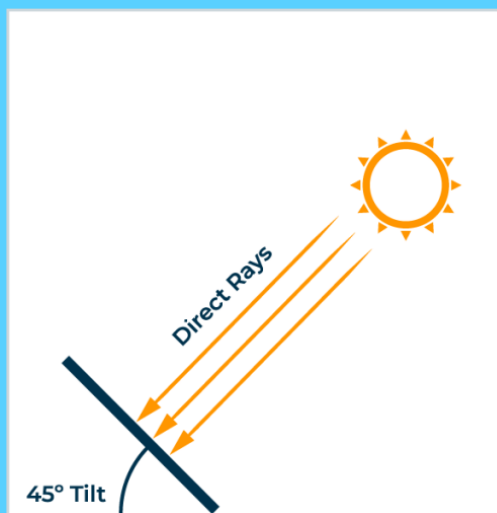
The teacher guides the discussion using the following questions:

- Are there clear differences between the groups?
- What could be the cause of these differences?
- Which observations are the same?
- What conclusions can be drawn from the results?

Step 2: Drawing Conclusions

Possible conclusions include:

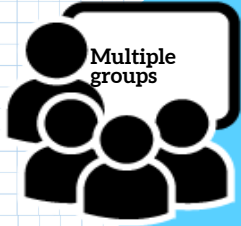
- The battery charges faster when no energy consumer is active.
- The intensity of sunlight or artificial light influences the charging speed. The more light available, the faster the battery charges.
- Light intensity is not the same in every location.
- Multiple factors influence the efficiency of the system.



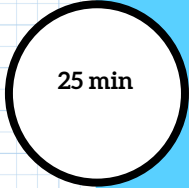
Step 3: Understanding the Role of Angle

The teacher explains that not only the location of the solar panel is important, but also its angle.

The teacher briefly discusses the visual examples.



Multiple
groups



25 min

Analysing and discussing results

Step 4: Extension Questions

The teacher encourages deeper thinking by asking:

- How should the sun be positioned relative to the solar panel to achieve optimal energy absorption?
- How are real solar panels positioned on rooftops? At which angle?
- Is a fixed angle always the best solution? (Consider seasonal changes such as summer and winter.)

Conclusion

Students understand that both light intensity and panel orientation influence energy efficiency and system performance.



25 min

Designing and building

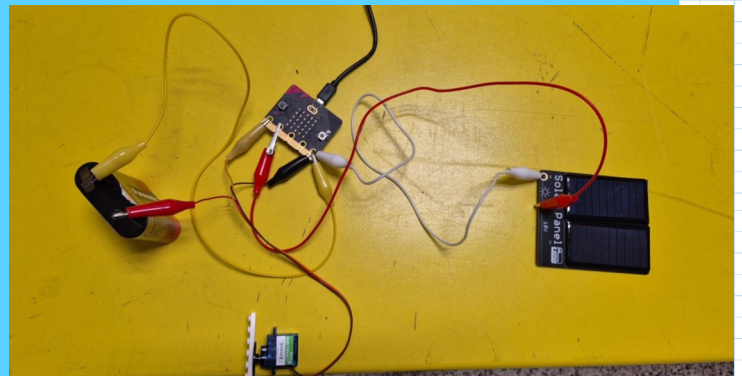
Students work in pairs to design and develop an improved solar panel system.

Their goal is to create a system that automatically adjusts the position of the solar panel based on the position of the sun.

Materials

Each group can use the following materials:

- 1 motor
- 1 LEGO Education SPIKE Prime set
- 1 Micro:bit
- Solar panel extension kit
- 1 battery
- Crocodile clips



Design Challenge

Students design a solar panel system that:

- is programmed using the Micro:bit
- automatically checks if the panel is in the optimal position
- adjusts its position when needed
- uses LEGO SPIKE Prime to create the prototype
- is able to rotate independently

Step 1: Design Phase

Students start by creating a design sketch of their system.

They must consider:

- placement of cables and connections
- positioning of the motor
- stability of the construction

The teacher reviews and approves the design before students start building.



In pairs

100 min

Designing and building

Step 2: Building Phase

Students build their prototype using the available materials.

They assign roles within their team, such as:

- construction
- programming

Step 3: Programming Phase

Students program their system so that:

- the solar panel detects light intensity
- the motor adjusts the panel angle accordingly

Step 4: Testing and Iteration

Students test their system and improve it where necessary.

The teacher:

- regularly checks progress
- asks questions about their approach
- provides support when needed

Reflection

Students reflect on their design:

- How does your system respond to light?
 - Does your design improve energy efficiency?
- What would you improve in a next iteration?



90 min

Differentiation

In Groups – up to 90 min

If students experience difficulties while programming the solar panel system, the teacher can provide a partially completed program.

In this version:

- all blocks are already in the correct position
- only the values and data still need to be adjusted

The goal for students is to improve the program **without adding or removing blocks**, but by modifying the existing parameters.

Support Strategy

The teacher can:

- provide a simplified version of the program
- show the individual programming blocks on the screen as hints
- guide students step by step when necessary

Example Solution

An example solution can be found online:

<https://makecode.microbit.org/62654-11815-62811-57709>

Note:

Multiple correct solutions are possible. This example is only one possible approach.

Extension for Advanced Students

If students finish building the solar panel system faster than expected, they can extend their design.

Students can:

- redesign their solar panel system to improve efficiency
- add additional functionalities
- explore more complex solutions

Students can use the **LEGO SPIKE Prime hardware** for these extensions.

Important Note

The system consists of two separate programs:

- controlling the solar panel using the Micro:bit software
- controlling the vehicle or movement using SPIKE Prime software



In pairs

Exhibition preparation

At the end of the workshop, students prepare their project for presentation in a small exhibition.

Step 1: Preparing the Explanation

Students prepare how they will explain their solar panel system to visitors.

They reflect on:

- What information will we share about our solar panel system?
- Who will explain which part?

Step 2: Demonstration

Students plan a short demonstration of their system.

They decide:

- how to show how the system works
- how to demonstrate the effect of light and panel position
- how to explain the role of the Micro:bit

Step 3: Visitor Interaction

Students think about how visitors can engage with their project.

They consider:

- What can visitors try out themselves?
- What questions can we ask the visitors?

Teacher Role

The teacher:

- supports students in structuring their explanation
- ensures all group members are involved
- provides feedback on clarity and demonstration

Conclusion

Students present their work and demonstrate their understanding of:

- solar energy systems
- programming and control system behaviour and optimisation



VR WORKSHOP

This activity focuses on designing sustainable solutions for endangered species by analysing their needs and creating adapted habitats.



STSL learning outcomes

- SCIENCE.STSL.2.3 – Making evidence-based predictions about relationships between living organisms and their environment
- SCIENCE.STSL.3.2 – Formulating a hypothesis or solution for a real-life design problem
- SCIENCE.STSL.3.3 – Transforming a design idea into a functional solution
- SCIENCE.STSL.4.1 – Investigating global environmental problems using scientific reasoning
- SCIENCE.STSL.4.3 – Developing a product or solution that addresses environmental challenges

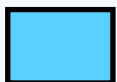
Kleurgebruik doorheen de handleiding



=Introduction and core content



=Differentiation and conclusion



=Practice and consolidation



=Additional information for the teacher

Tips

- Create a free account on Sketchfab in advance (www.sketchfab.com). You can upload up to 10 models using the same account.
- Scan the QR code to watch the introductory video about the Oculus Quest.

Basic Controls

- **Zoom in / Zoom out:** Use the scroll wheel on the computer mouse.
- **Changing the Perspective:** Click on the model and drag the mouse in the direction you want to view.
- **Changing Colours:** Select a block first. In the workspace, click on the block and choose the desired colour.
- **Connecting Blocks:** The selected block is highlighted (in red). When moving the mouse over the block, it turns blue and connection points appear. These connection points show where the next component can be added. When hovering over a point, it turns green. Click with the left mouse button to confirm the connection.
- **Finding Blocks:** At the bottom of the screen, you will find the library with different options.
- **Made a mistake?** Use **Ctrl + Z** to undo.



Oculus



Additional Information

More information can be found here:

<https://www.mecabricks.com/en/forum/topic/65>



15 min

Animal habitat challenge

Activity 1: Introduction to the Design Challenge

Students are introduced to the main challenge of the day.

The teacher explains that students will work in pairs to design a **protected habitat** for an endangered animal of their choice.

Step 1: Introducing the Concept

The teacher asks the following questions:

- What is a protected habitat? (A habitat that is adapted to the needs of the animal, without threats such as poachers or other animals.)
- Why is this important? (Animals in the wild are often threatened, and animals in captivity may not live in suitable conditions.)

Step 2: Context and Inspiration

The teacher briefly introduces a real-world example:

LEGO builders David and Gus created a protected habitat for a deer in LEGO Masters Australia (2021).

The deer had previously been listed as an endangered species but was removed from the list in 2007.

Students watch an inspiration video.

Step 3: Identifying Endangered Species

The teacher asks:

- Which animal species are currently endangered?

Examples may include:

jaguar, orangutan, panda, rhinoceros, lion, polar bear, shark, whale, sea turtle, tiger, dolphin, elephant, leopard, coral.

Students select one endangered animal from the list.

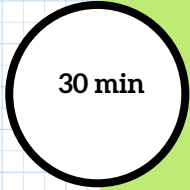
Step 4: Explaining the Challenge

The teacher explains the different steps of the design challenge:

1. Choose an endangered animal (keep it secret from other groups).
2. Brainstorm ideas and complete the "animal passport" using online information.
3. Develop a solution and present a sketch or initial concept to the teacher.
4. Design and improve the habitat (optionally build using LEGO).
5. Communicate results (Students explore habitats in VR and guess which animal belongs to which habitat – this takes place the next day (it takes time to upload))



In pairs



30 min

Animal habitat challenge

Activity 2: Investigating Animal Needs

Students investigate the needs of their chosen animal using the questions in the animal passport.

The animal passport forms the basis for the design of their habitat in a later phase.

1. At the end of the activity, students present their initial habitat design or sketch to the teacher.
2. Encourage students to justify their design choices based on the needs of the animal.



In pairs

75 min

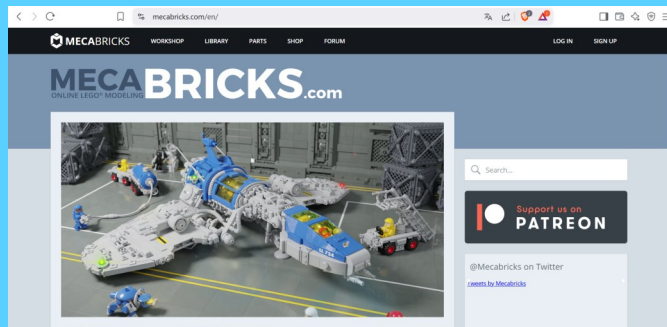
Building the habitat in 3D software

Students create a digital version of their habitat using 3D design software.

Step 1: Accessing the Platform

Students open the website:

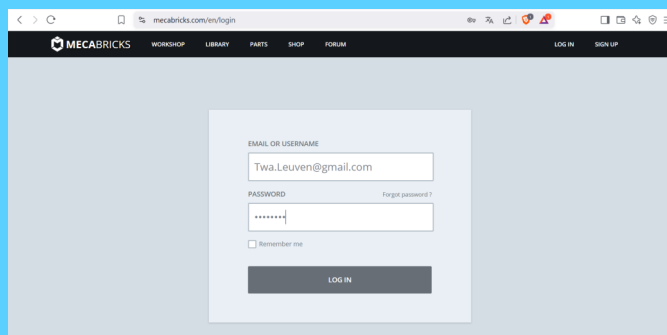
<https://www.mecabricks.com/>



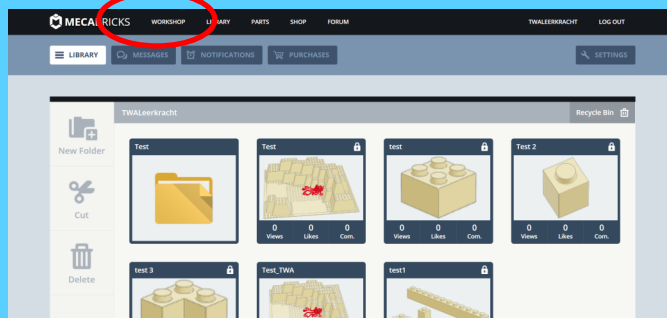
Step 2: Logging In

Students click on “**Log in**” in the top right corner and sign in.

The teacher provides the login details.



Building the habitat in 3D software

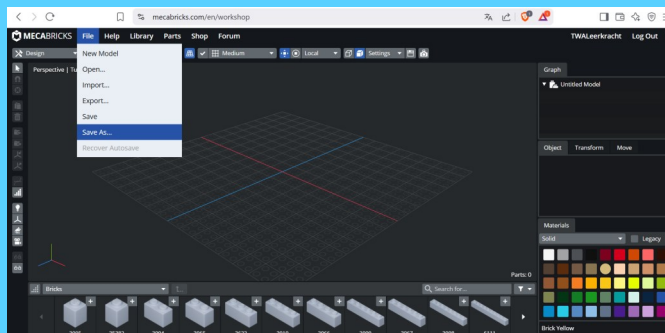


Step 3: Creating the 3D Design

Students build their habitat digitally based on their design sketch.

They:

- recreate their habitat in 3D
- apply their design choices from the previous activity
- further refine their solution



Step 1: Introduction to the Software

The teacher provides a short introduction to the 3D software.

During this introduction, the teacher demonstrates the basic controls and key functions.

Step 2: Creating the Base Environment

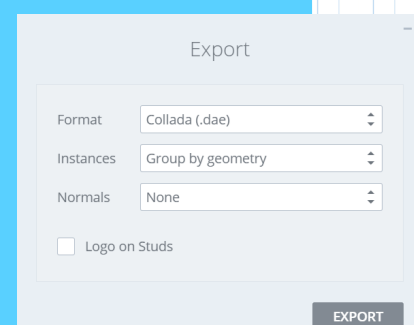
The teacher and students create the base of the world together.

Students place **5 × 5 baseplates (32 × 32)** to form the floor.

Important: Students must not exceed this building area, otherwise they will not be able to view the environment properly in VR.

Differentiation

- **Students who need more time:** focus on completing the basic habitat structure
- **Advanced students (early finishers):** create a LEGO-style avatar ("LEGO selfie") in the software



Building the habitat in 3D software

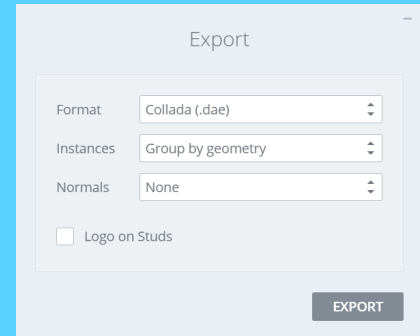
Step 4: Exporting the Models

At the end of the activity, all models are exported.

The teacher:

- clicks on the export function
- saves each model as a **.dae file** on a USB drive

The files can later be uploaded to Sketchfab (e.g. from home).

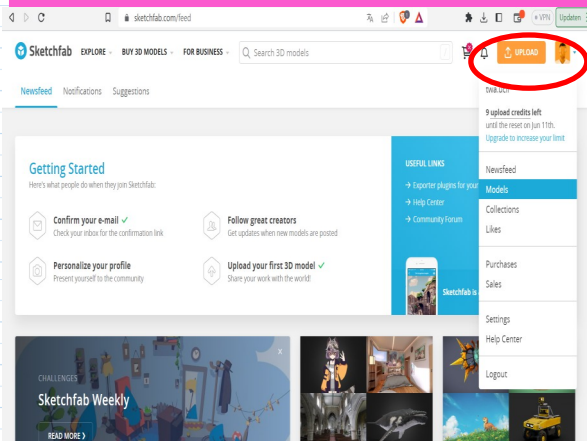


Teacher Role

The teacher:

- demonstrates the software
 - supports students with technical challenges
 - monitors progress and collaboration within the groups
- checks whether students stay within the defined design constraints

Uploading to Sketchfab and Viewing in VR



Step 1: Uploading the Models

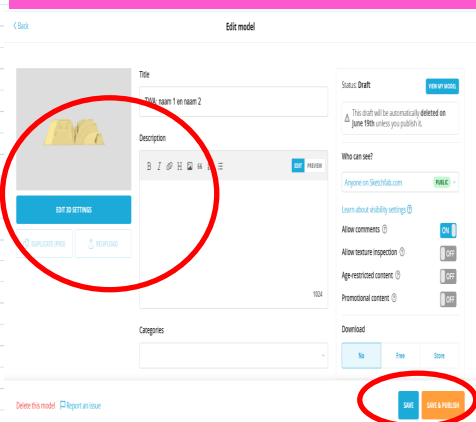
- Log in to Sketchfab.
- Click on **“Upload”** and upload one of the students’ models. These files may be stored in zip folders.

Step 2: Checking and Editing the Project

- Check the name of the project.
- Use the following naming convention:
TWA: name1 & name2 (enter the students’ names)

If you make changes to the name:

- first save the project using the **“Save”** button at the bottom
- then click on **“Edit 3D settings”**

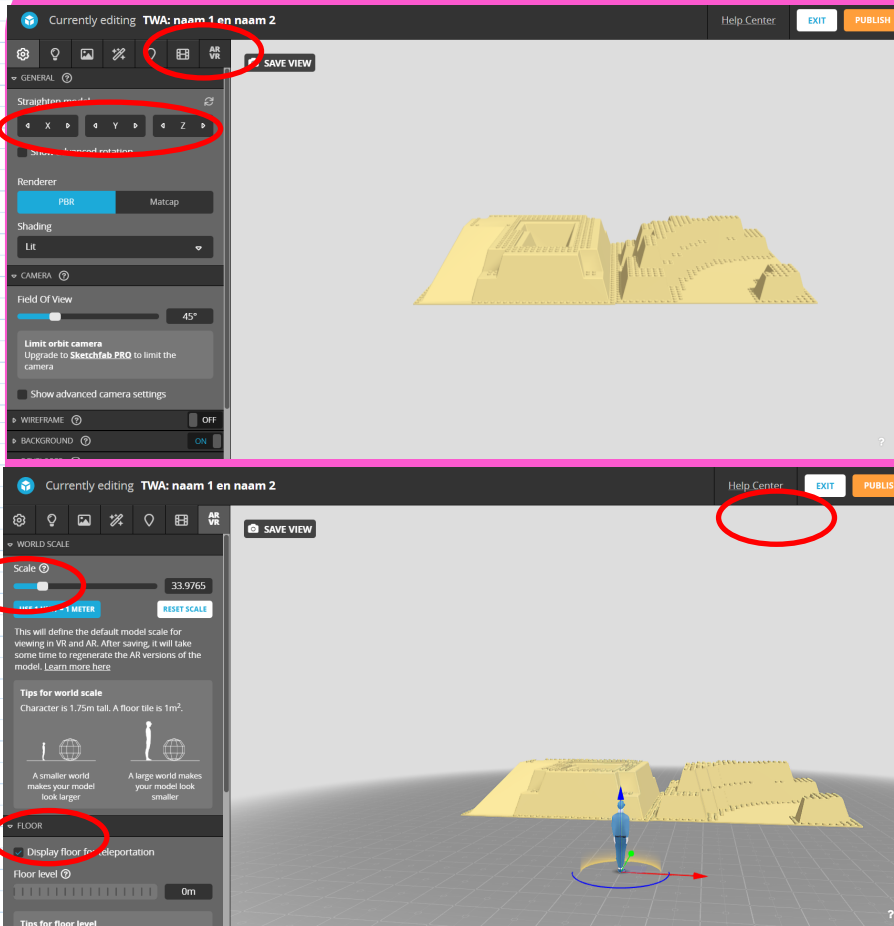


Teacher Note

Ensure that:

- all student models are correctly uploaded
- project names are consistent
- files are properly organised for later use

Encourage students to explore their models in VR and compare their designs with others.



Step 3: Adjusting Orientation

In the X, Y and Z settings, the axes can be rotated.

This may be necessary if the model appears incorrectly oriented.

If the model appears correctly positioned (as in the example image), click on **"VRVR"** to continue.

Step 4: Adjusting Scale and Floor Level

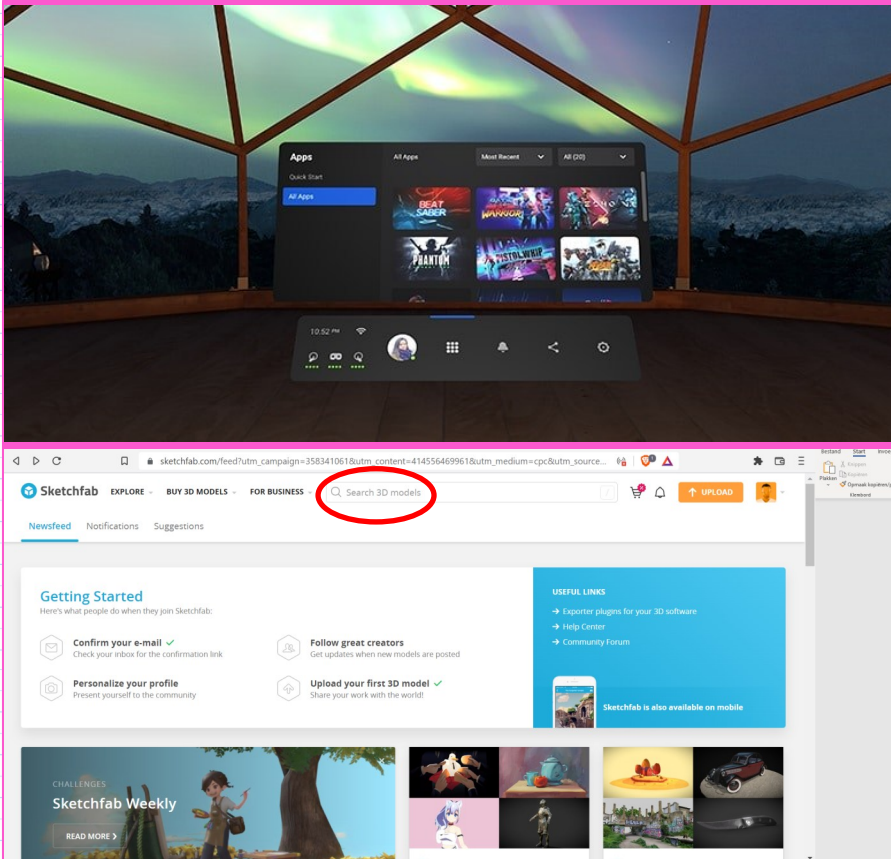
Adjust the following settings:

- **Scale (30)**
- **Floor level**

This ensures that the avatar can move properly within the environment.

Once the settings are correct:

- click on **"Publish"**
- Repeat this process for all student projects.

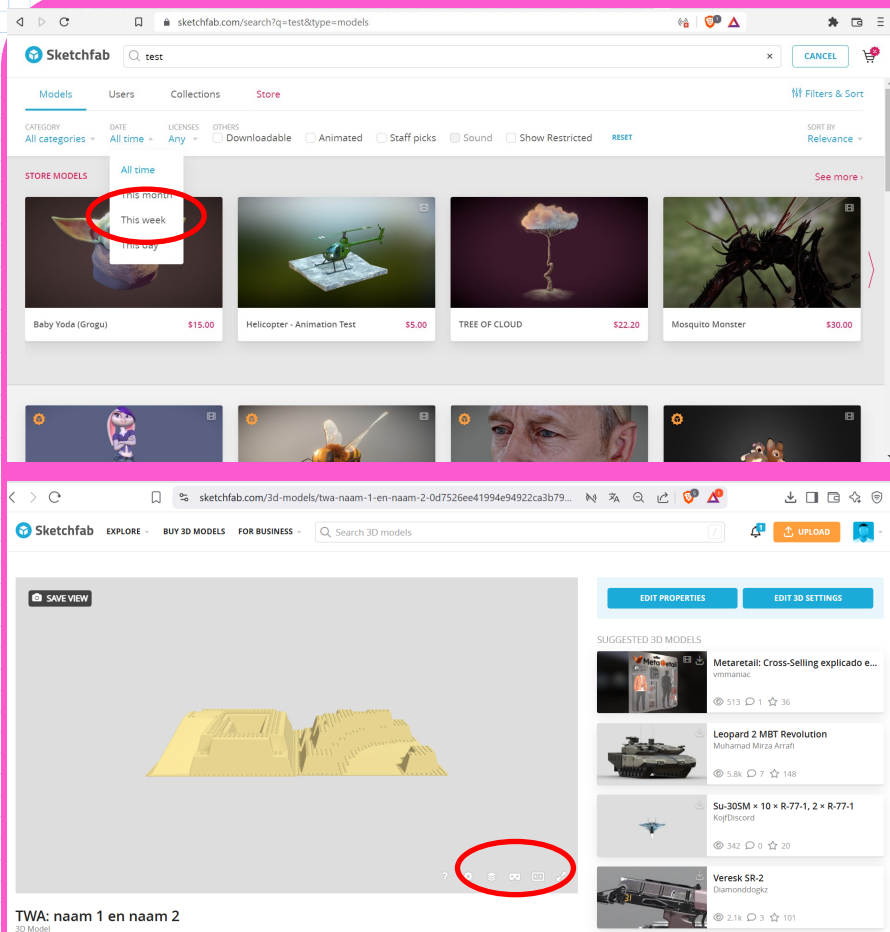


Step 5: Preparing the VR Environment

- Put on the Oculus Quest headset
- Set up a **guardian boundary** (using the QR code or internet connection)
- Open the internet browser using the controller
- Enter the following website in the search bar:
www.sketchfab.com

Step 6: Opening the 3D Models

- Click on **"Search 3D models"**
- Enter the name of the student project
- Select the correct model and open it in VR



Step 7: Filtering the Models

• Use the filters in Sketchfab to find the correct models. Select:

- **"This week"** or
- **"Today"**

This helps to quickly locate the recently uploaded student projects.

Step 8: Viewing the Model in VR

- Open the selected model.
- Click on the **VR icon**.
- Explore the habitat in the VR environment.



Exploring the projects in VR

While some students are still building and programming their projects, others can explore the designs of all groups in VR.

Game Element

A game element is integrated into this activity.

The teacher writes the selected animals of each group on the board in advance.

After exploring a VR environment, students must:

- guess which habitat they visited
- write down the name of the animal that belongs to that habitat

The goal is to identify as many habitats correctly as possible.

Exploration Rules

- Each student is allowed a maximum of **5 minutes per environment**
- Students work in pairs:
- one student wears the VR headset
- the partner ensures safety and guides the user

Safety Guidelines

- Use hygiene masks under the VR headset (students keep these throughout the day)
- Assist students when putting on or adjusting the VR headset
- Students may wear their own lenses or glasses under the headset if necessary
- Clearly mark the play area (guardian boundary) using tape on the floor
- Students must **not physically walk around the room** → they should use teleportation via the controller
- Never touch students wearing a VR headset unexpectedly → always communicate clearly
- Ensure that other students keep a safe distance from the VR user

Conclusion

Students explore, compare and evaluate different habitat designs in VR and connect them to the needs of endangered animals.





LASER SAFARI

Design technology using inkscape



Short Description

In this workshop, students create six animal magnets using the Inkscape software.

Inspired by a poem by Joke Van Leeuwen, students combine different animal parts in a playful and creative way, resulting in unusual and imaginative compositions. Students explore creativity and design by experimenting with shapes, composition and digital tools.

STSL learning outcomes

STSL Learning Outcomes

- SCIENCE.STSL.1.3 - Using digital technologies to create and modify visual content -
- SCIENCE.STSL.2.2 - Analysing how components can be combined to create new systems or designs
- SCIENCE.STSL.3.2 - Formulating a solution for a creative design problem
- SCIENCE.STSL.3.3 - Developing a design into a functional or visual product
- SCIENCE.STSL.4.2 - Exploring creativity and innovation in the use of science and technology

Tips and tricks voor Inkscape

Differentiation

You can prepare parts of the activity in advance.

For example:

- prepare the oval shapes in Inkscape in advance (up to step 4)
- provide these templates to the students

This allows students to focus on assembling and designing their animals instead of creating all shapes from scratch.

Keyboard Shortcuts

- **Made a mistake?** Press **Ctrl + Z** to undo.
- **Zoom overview of the page:** Press **Ctrl + 5**
- **Zoom in / Zoom out:** Hold **Ctrl** and scroll with the mouse wheel
- **Exit a drawing tool or menu:** Press **Escape**

Encourage students to experiment with different shapes and combinations to enhance creativity.

Workshop flow

Activity 1: Listening to the Poem

The teacher or student reads the poem aloud.

Guiding Questions

The teacher asks the following questions:

- What happens in the poem? (Animals are combined in unusual ways. In science, researchers are also studying the combination of genes. This is called genetic modification, for example using CRISPR-Cas9. With this technology, scientists can modify, remove or add specific genes to an organism in order to change or improve certain characteristics.)
- Which animals would you like to combine?



5 min

Introduction to the Challenge

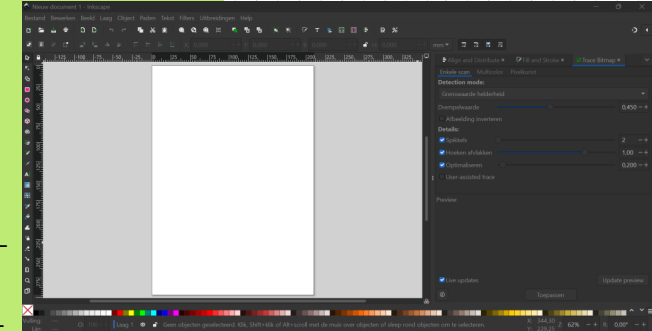
The teacher explains that students will have the opportunity to combine three animals of their choice.

Students will create **six animal magnets** based on these combinations.

Teacher Role

The teacher:

- facilitates discussion
- encourages students to think creatively
- guides students in making connections between science and design





Following the Design Steps

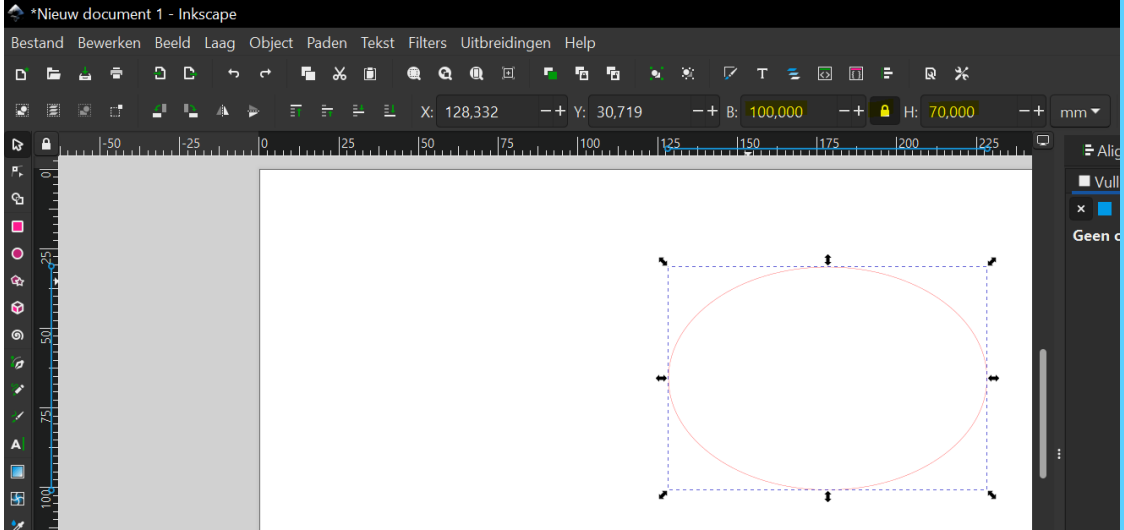
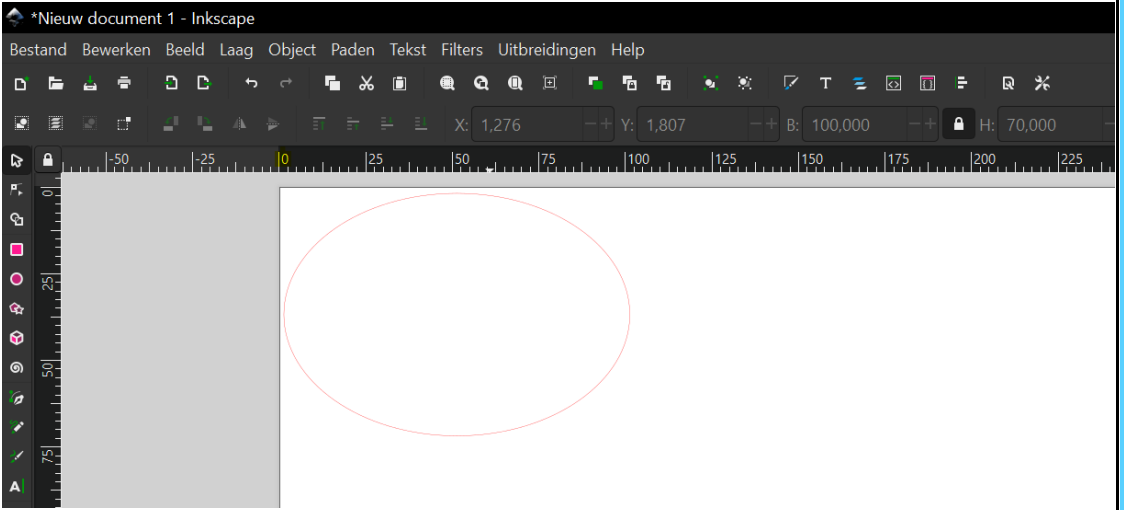
Students design three new animals by following a structured step-by-step plan. Before starting, the teacher ensures that students agree on their roles:

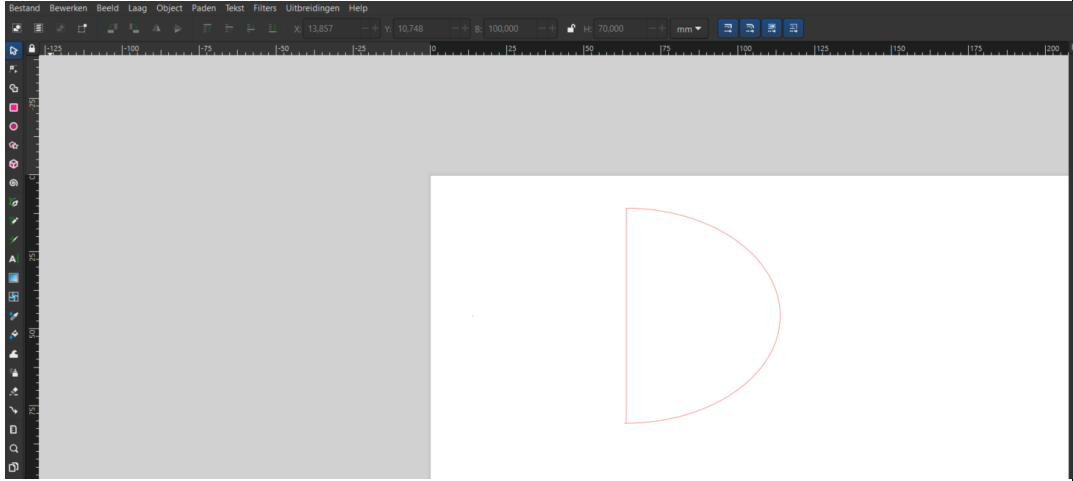
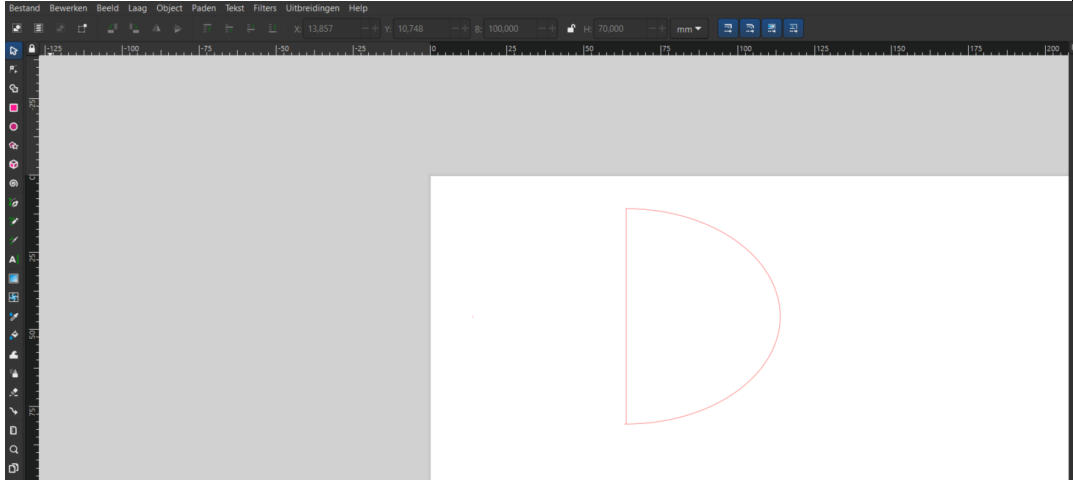
One student acts as the driver (controls the computer and performs the actions). The other student acts as the navigator (gives instructions and guides the process)

The teacher reviews each step with the whole class before students continue individually.

Designing in Inkscape

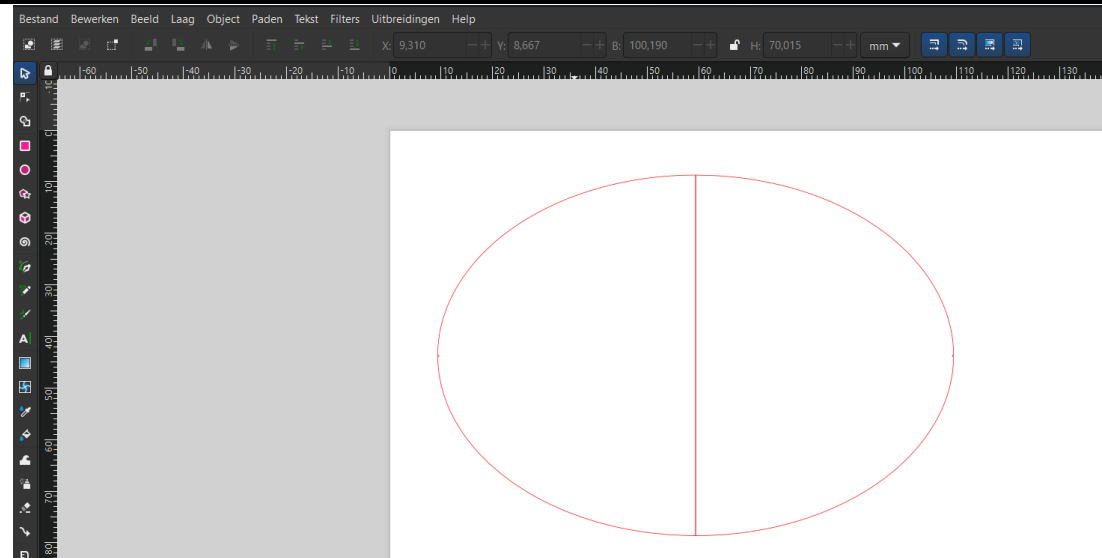
Stap	Uitleg	Foto
1	<u>Adjusting the document size</u> 1. Click on "File" in the top left corner 2. Click on "Document Properties" 3. Enter the values as shown in the example (Width: 600mm, Height: 300mm) 4. Close the window 5. Press shift + 5 to display the full workspace	

Stap	Uitleg	Foto
2	<u>Drawing the oval</u> 1. Select the ellipse tool and draw an oval. The current colour of the shape is not important 2. Adjust the size of the oval: Width: 100 mm (W: 100), Height: 70 mm (H: 70). To do this: Click on the shape and modify the values in the toolbar at the top of the screen 3. Lock the proportions by clicking on the lock icon to fix the shape	
3	<u>Adjusting fill and stroke</u> 1. Select the oval 2. Click on "Object" -> "Fill and Stroke" 3. Apply the following settings: "Fill: none", "Stroke colour: red (255)", "Stroke width: 0.1 mm" 4. Encourage students to check their measurements carefully to ensure accuracy before continuing	

Stap	Uitleg	Foto
4	<u>Splitting the oval—part 1</u> 1. Draw a rectangle with the following dimensions: Width: 50 mm, Height: 70 mm 2. Select both the oval and the rectangle: Hold the shift key and click both shapes 3. Go to “Object” -> “Align and Distribute” 4. Select the appropriate alignment options  4. Then go to “Path” -> “Difference”	
	<u>Splitting the oval—part 2</u> 1. Copy the oval: Press Ctrl + C, Press Ctrl + V 2. Select both shapes using the shift key 3. Go to “Object” -> “Align and Distribute” 4. Choose the correct alignment options (see below) 	

Splitting the oval—Part 3

1. Rotate the top half of the oval by 180° using the rotation tool
2. Go to “Object” -> “Align and Distribute” and select the correct alignment option 
3. Move the half oval using the arrow keys until the shapes overlap correctly
4. Select both shapes by holding the shift key
5. Go to “Path” -> “Combine”

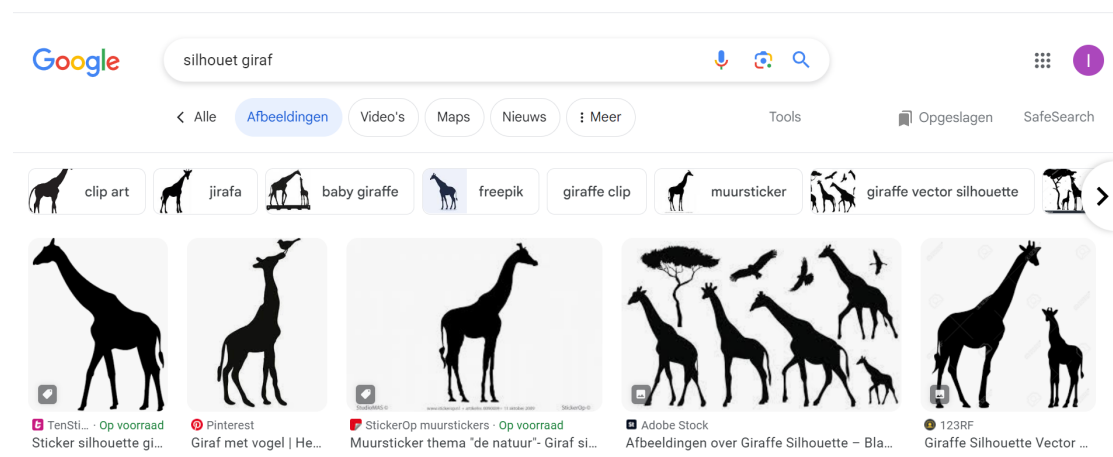


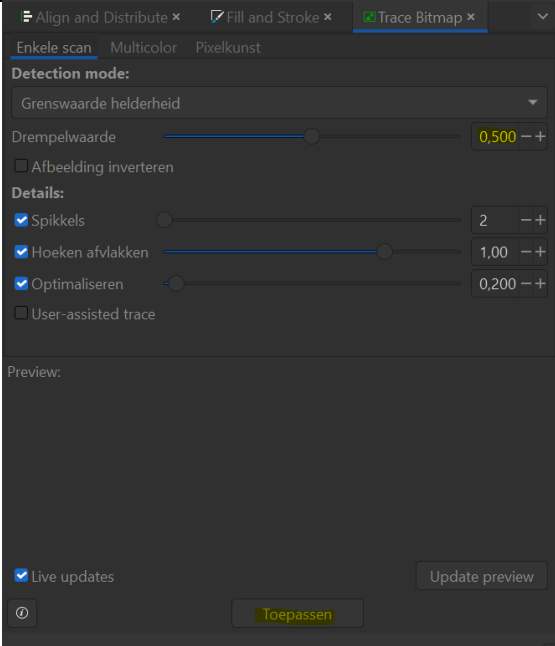
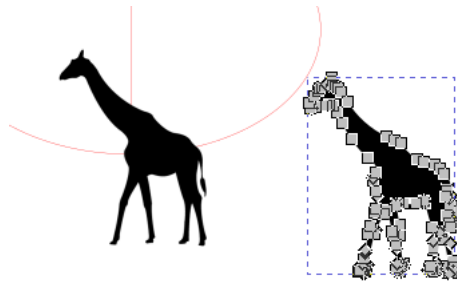
5

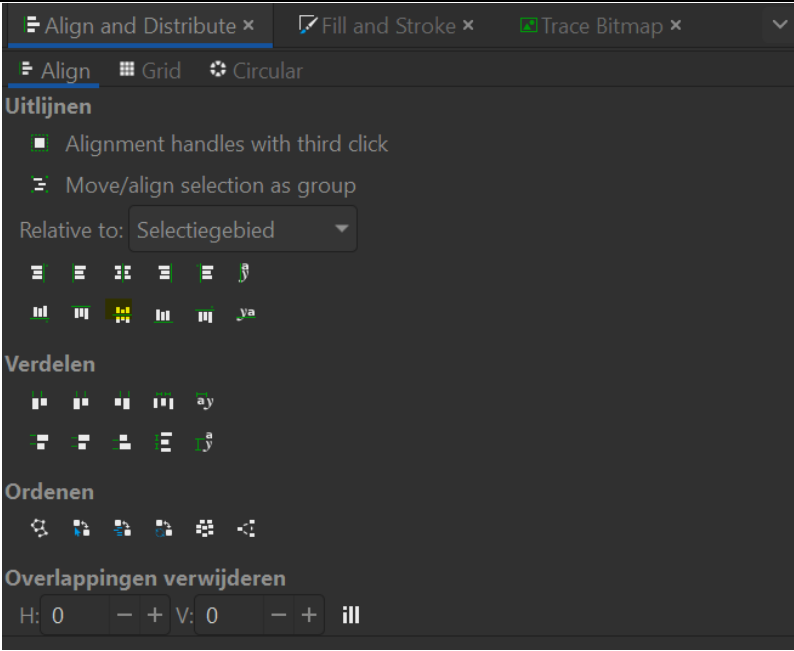
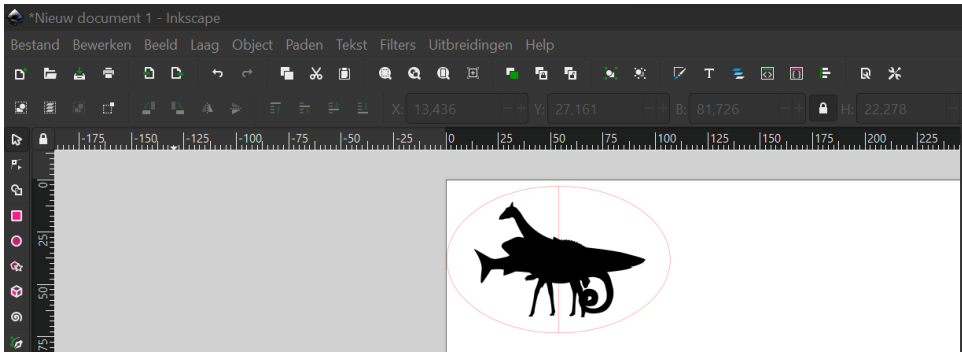
Searching for line drawings

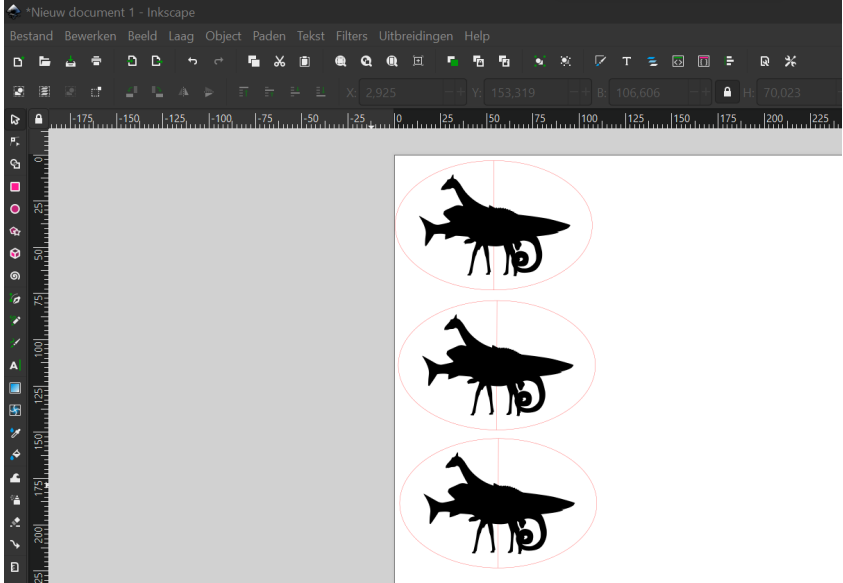
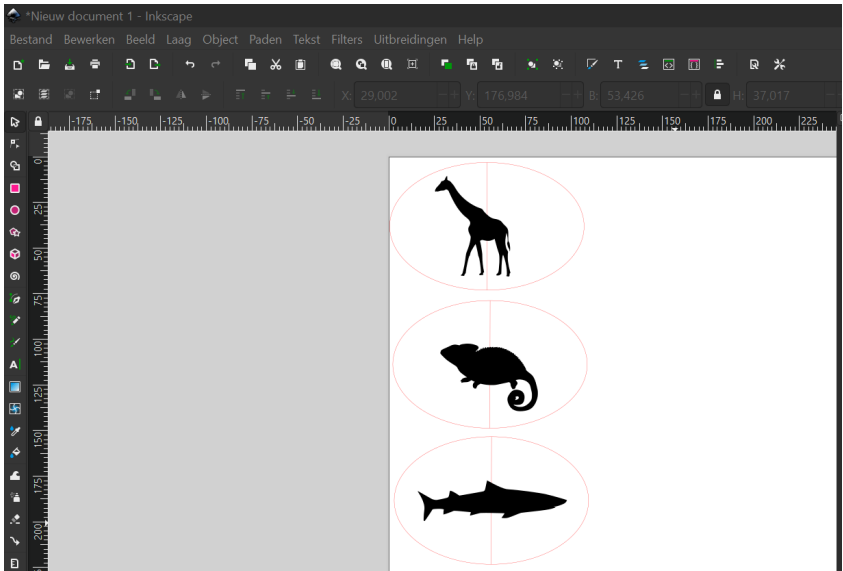
1. Ask students to search online using the term: Silhouette + name of their chosen animal
2. When students find a suitable image:
3. Copy the image and paste it into Inkscape using Ctrl + V

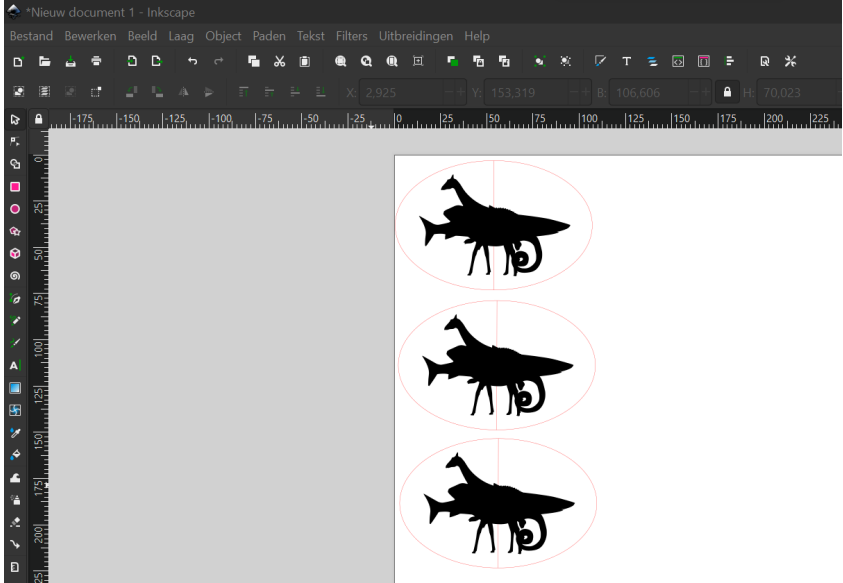
Encourage students to choose clear and simple silhouettes to make the design process easier.



Stap	Uitleg	Foto
6	<u>Converting a PNG to a SVG</u> 1. Resize the image so that it fits within the workspace 2. Select the image and go to: "Path" -> "Trace Bitmap" 3. In the new window: ensure that the threshold value is at least 0.5 4. Click on "Apply"	
7	<u>Cleaning the image and positioning</u> 1. Click on the image and drag it slightly to the side, you will now see two versions of the image 2. Double-click on one of the images. If you see many small nodes (points), this is the correct vector version. Delete the other version 3. Position the correct animal inside the oval so that it is cut exactly in half	

Stap	Uitleg	Foto
8	<u>Aligning the giraffe and the oval</u> 1. Select both the oval and the giraffe 2. Go to "Object" -> "Align and Distribute" 3. Select only the bottom alignment options (as shown in the example)	
9	<u>Repeating the steps</u> 1. Repeat steps 5 to 7 for the two other animals 2. Place the animals on top of the giraffe 3. The additional animals do not need to be aligned	

Stap	Uitleg	Foto
10	<u>Duplicating the animals</u> 1. Select the oval and the three animals 2. Press: Ctrl + C , Ctrl + V 3. Drag the duplicated oval to the side 4. Repeat this process until you have three ovals	
11	<u>Selecting the final animals</u> 1. Make sure each oval contains only one animal.	

Stap	Uitleg	Foto
12	<u>Saving the document</u> 1. Insert the USB drive into the laptop 2. Click on "File" -> "Save as ..." 3. Save all documents on the usb drive 4. Hand the USB drive back to the teacher or supervisor	
13	<u>Attaching the magnets</u> 1. Cut small pieces of magnetic tape (approx. 1 cm each) 2. Attach the magnetic pieces to the back of the wooden shape 3. Test the magnets to make sure they work properly	

DRONE WORKSHOP

Introduction to Drones and Their Applications



Learning objectives

Students will be able to:

- Explain what a drone is and how it works
- Identify different applications of drones in society -
- Understand basic safety rules and regulations for drone use
- Reflect on how drones are used in real-world contexts



STSL Learning Outcomes

- SCIENCE.STSL.1.1 – Understanding the role of science and technology in society
- SCIENCE.STSL.1.3 – Using digital technologies to investigate and understand applications
- SCIENCE.STSL.2.1 – Understanding how systems respond to inputs and produce outputs
- SCIENCE.STSL.3.2 – Formulating ideas about technological solutions in real-world contexts

Materiaal:

- 2 drone models
- 7 tablets
- 7 drones
- Obstacles (optional)



Workshop flow

Preparation

Before the activity:

- Turn on the projector and main computer connected to it
- Prepare **6 computers**, distributed across the room
- Students work in pairs at each computer
- Ensure that all students have sufficient space to work comfortably
- Log in to the computers using the provided credentials
- Start the **Kahoot quiz**



Activity 1: Introduction and Exploration

Students gather around the central table.

The teacher starts the workshop with a name round and an exploratory discussion.

Guiding Questions

The teacher asks:

- Who has ever flown a drone before?
- Can someone explain what a drone is?
- Are there different types of drones?

Discussion

Students share their experiences and ideas.

The teacher builds on their answers and introduces basic concepts about drones.

Safety Agreements

The teacher explains the safety rules:

- Drone controllers must be handled carefully
- When flying:
 - stay above the designated flying area (e.g. mats)
 - avoid potential damage
 - A maximum of **2 drones** can be in the air at the same time
 - Drones must not fly near other students

Reflection: Why do you think safety rules are important when flying a drone?

Activity 2: Kahoot Quiz – Drone Basics

Students participate in an interactive Kahoot quiz using their smartphones.

Activity Description

The teacher starts the quiz and reads each question aloud.



Students answer individually or in small groups.

After each question, the teacher discusses:

- why the correct answer is correct
- why the other answers are incorrect

Teacher Guidance

The teacher:

- encourages students to explain their reasoning
- allows time for discussion after each question
- connects the answers to real-world drone use

Discussion Questions

After selected questions, the teacher asks:

- Why is this answer correct?
- What would happen if this rule is not followed?
- How does this apply in real-life situations?

Example Explanations

The teacher can use simplified explanations such as:

A drone must be:

- able to fly
- unmanned
- remotely controlled

Drones must respect:

- privacy rules
- safety regulations
- flight restrictions

Propellers influence movement:

- equal speed → drone goes up
- differences in speed → direction changes

Students reflect on their learning:

- What new knowledge did you gain about drones?
- Which rule or concept surprised you most?

Activity 3: Building a Drone Model

Students work in groups of 4 or 6.

Each group receives the necessary materials:

- 1 drone model
- 2 A propellers
- 2 B propellers
- 1 control panel
- 1 adapter cable



Step 1: Building the Model

Students attempt to assemble the drone model correctly.

The only instruction provided is:

- The drone must be able to generate lift and move air downward.
- Students explore and experiment with the components to find a working solution.

Safety Guidelines

The teacher explains:

- Keep a safe distance from the propellers when they are rotating
- Students can feel the airflow from a distance (approx. 10 cm), but should not touch the propellers
- Handle all materials carefully

Step 2: Understanding the Structure

Correct configuration:

- A propellers are placed on **motors 1 and 3**
- B propellers are placed on **motors 2 and 4**
- Motor cables:
 - motors 1 and 3 → numbers facing towards the user
 - motors 2 and 4 → numbers facing away
 - Propellers rotate in **opposite directions**

This creates:

- downward airflow
- stability
- the ability to move and rotate

Discussion

The teacher asks:

- Why do some propellers rotate in opposite directions?
- What happens if all propellers rotate in the same direction?
- How does airflow influence the movement of the drone?

Teacher Role

The teacher:

- observes how students build their model
- guides students without giving immediate answers
- encourages trial and error
- supports understanding of how the system works



Reflection

Students reflect on their model:

- What did you change to make the drone work?
- Why is correct propeller placement important?

What did you learn about how drones generate lift?

Activity 4: Understanding How Drones Work

Students analyse how their drone model functions.



Step 1: Reflecting on the Building Process

The teacher asks:

- Why didn't the drone work at first?
- What did you have to change?
- Which part was most important for stability?

Students discuss their answers.

Step 2: Explaining the System

The teacher explains:

- Lift is created by propellers pushing air downward
- Stability depends on balanced rotation
- Movement depends on differences in propeller speed

Discussion

- What happens if one propeller spins faster than the others?
- Why do drones need multiple propellers?

Activity 5: Regulations and Safety

Step 1: Exploring the Rules

The teacher introduces drone regulations.

Students learn about:



- maximum flight height (120 m)
- visual line of sight (VLOS)
- no flying above crowds
- respecting privacy

Scenario-Based Thinking

The teacher asks:

- If you want to fly in a busy park, what rules apply?
- What could go wrong if rules are ignored?

Students discuss possible situations.

Reflection

Why do these rules exist?

Which rule do you think is the most important?

Activity 7: Applications and Innovation + Flying the drones

Step 1: Exploring Applications

The teacher shows examples:



- delivery (Amazon, bpost)
- agriculture (crop monitoring)
- rescue operations
- construction and mapping

Step 2: Future Thinking

Students explore:

- AI drones
- autonomous drones
- flying taxis

Discussion

Which application do you find most useful and why?

Which application would you improve?