Erasmus+

PROGRAMMING Box

Head In The Clouds Consortium

https://brainsintheclouds.eu

All figures by https://www.pixabay.com

English

TECHNISCHE UNIVERSITÄT WIEN

TECHNICKÁ UNIVERZITA V KOŠICIACH

GAIA KOSOVO SCI

Verein Offenes Lernen

Diakonie

SZS Galakticka KOSICE

SCIO

25. April 2018

OER OPEN EDUCATIONAL RESOURCES

The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

You will notice that the QR codes used for the hand-in application are not present in these tasks. To use the tasks with the hand-in application the collaboration with the project team is necessary, so please contact us by mail.

Table of Content

Topic 1: MakeyMakey

Topic 2: Ozobot

Topic 3: Minecraft

Topic: Lego Education WeDo

Teacher Handout BOX 4: Programming

The Programming box is a box full of hands-on programming and engineering.

It contains 4 independent topic that can be done in any order or even parallel. However, it should not be started before the basics of Scratch-programming had been done (see IT101-box).

NOTE: This document is just a brief overview - there is a detailed teacher's handout for each of the 4 parts!

Topic a: MakeyMakey

MakeyMakeys will teach basics about electric circuits and engage your students in handicraft work and programming in Scratch.

Topic b: Ozobot

Ozobots offer great tasks around drawing and very simple programming (simplier than Scratch). Therefore it is a good option for very young children.

Topic c: Python & Minecraft

As Minecraft is very popular with children of almost all age groups, we want to use it for getting deeper into programming. Python is a very powerful programming language that is used by many IT professionals. Using Python, many activities in Minecraft can be automated, which might have a huge motivational impact on some students. For many students, this might be too difficult - however, we think, it makes sense to have something to offer to those who have the "IT gen" in them. Discovering those can be a stepstone to further professional opportunities.

Topic d: Lego WeDo

Building with Lego-bricks has been a classic favorite occupation for generations of children. We believe that building with Lego itself is highly educational in many points of view. Lego WeDo enriches the pure building with a little programming (in Lego WeDo native programming environment or in Scratch).

Teacher Handout BOX 4: Programming - Topic MakeyMakey

MakeyMakey are small electronic devices, that can be connected to a computer and behave like a keyboard (that's the reason why it's name contains the word KEY: MaKEYMaKEY).

The MakeyMakey box offers tasks that guide children from the basic understanding of electricity to handicraft and programming projects.

Many of the tasks are based on the work that was done by Michael Hielscher and Beat Döbeli Honegger by "pädagogische hochschule schwyz". We thank them for sharing and inspiring us!

Most popular keys

The most popular and most frequently used keys are the space-key and the arrow-keys (up - down - left - right). With those six keys you can already create a game-controller for many simple videogames.

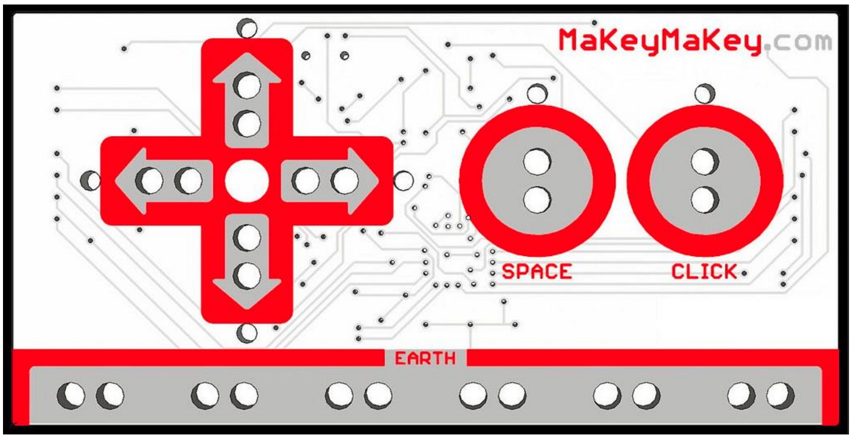


Fig. 1

(If you need more keys, check out the manual that is delivered with the MakeyMakey or the links in the section "Further material")

IMPORTANT:

All tasks only work, if the electric curcuit is closed. Typically this happens by attaching one cable to "Earth" and having one person holding the other end of this cable. (You can also attach it to a ring this person is wearing on the finger!)

Further cables are connected to keys and other objects, e.g. bananas.

For figure sources and references please refer to List of Figures.

Teacher Handout BOX 4: Programming - Topic MakeyMakey

Taskgroups and tasks

There are 2 groups of tasks in this topic:

The first group of tasks teaches the basics about how the MakeyMakey works. This group should be completed before starting the second. All tasks in this group should be done with all students (typically in a group).

The second group of tasks shows some examples of what can be done with the MakeyMakey. It is fun stuff - mostly games that need some handicraft-work. Start with the fruit-piano, as it is the simplest. Bananas usually work well, but you can take any other fruit or vegetable as well. In task 4a1.2 you should have found out which of your available items work best!

If you (and your children) have once understood the principles, then all the other tasks are technically quite simple and more a challenge for handicraft and creativity. However, if students want to deeper engage with programming, they can do so - examples are given in the tasksheets.

Further material

More explainations, tutorials, lesson plans, apps (!) and further ideas can be found here:

https://makeymakey.com/

https://learn.sparkfun.com/tutorials/makey-makey-quickstart-guide

Goals:

- *Engage in hands-on work with electronics*

Skills/Competences addressed:

- *IT, Programming, Handicraft*
- *Logical thinking*
- *Creativity*
- *Working in a team*

Connection to other boxes:

This box should be done after IT101

Teacher Handout BOX 4: Programming - Ozobot

Ozobots are small, simple robots that can “understand” lines and color-patterns in those lines. According to the color-pattern, they show a certain behavior (e.g. going faster, turning right at the next crossroad). This allows children to “program” them just by using paper and colored pens. The tasks in this box cover this “pen and paper”-approach, in order to allow also very young children to have meaningful interaction with the Ozobots.

There are additional possibilities for programming using tablets/PCs, which are also interesting for older children. Links to these resources are given at the end of this document (see part “Further material”).

Many of the tasks are based on the work that was done by Michael Hielscher and Beat Döbeli Honegger by “pädagogische hochschule schwyz”. We thank them for sharing and inspiring us!

Main topics and tasks

The Ozobot-part of Box 4 (“Programming-Box”) includes 4 topics, that can all be performed without computer-assistance.

The topics should be done in their order, which means 1 - 2 - 3 - 4.

Within the topic, tasks can be done in any order, and not all tasks must be done. Tasks should give students inspiration for own projects.

We strongly believe, that own projects are more important than strictly doing one task after the other.

NOTE: In many tasks we have videos included. They can be accessed via the hand-it-in-application (QR-code). Students should be encouraged to use this application: They should answer the questions there, watch the videos and - most important - upload pictures/videos of their own projects!

Topic 1: Getting to know your Ozobot

There are only 2 tasks in this topic. The first one, “Take care of your Ozobot”, shows how to charge, clean and calibrate it. The second one tries to deepen the emotional bonding by encouraging the children to personalize it.

“Take care of your Ozobot” also serves as reference in case of troubles - thus all students (and teachers) should be familiar with it. On the hand-it-in-application page of this task, you will find videos for more detailed information.

Topic 1 can be easily skipped, if you feel that it is too boring to begin with and you do not want to loose the student’s excitement:

- The “Take care of your Ozobot” IS ESSENTIAL, but not IMMEDIATELY; thus students CAN immediately start working with the Ozobot as long as it starts blinking red - then they should know what to do
- The decoration task is a bit tricky, if there are too few Ozobots for too many students.

Topic 2: Ozobots love lines

This topic explores Ozobots primary skill, namely to follow lines. Students are encouraged to draw own maps. One tasks encourages the usage of real maps, thus letting Ozobot travel. This offers a wide range of activities around geography. At this stage, we do not yet use the colors (they are added in topic 3; it's always nice to have something new to show and discover later!)

Topic 3: Ozobot love colors

The other primary skill of Ozobot is to recognise colors and patterns of colors. In topic 3, students will discover those codes and their behaviour. They should document what they have discovered (write a list of codes and behaviour). Nevertheless, the box includes the "Ozobot Colorcodes - Overview", which students may use for tasks in topic 4.

Topic 4: Programming with color-codes

As already mentioned, this allows simple programming and problem-solving-tasks.

For 2 Tasks, 4b4.1 and 4b4.2, students should fill in colors on the task-sheets, thus you will need more copies of it.

Further material

Links

The Ozobot community provides a huge variety of material for further work. If you feel, that your students need more, here are some suggestions where to start searching:

The Official ozobot website offers a lot of material for educators: ready-to-use lessonplans, games, brain-teasers, etc: https://ozobot.com/

The site "Exploring Robotics" also offers great material for Ozobot and other robots: https://www.exploringrobots.com/index.php/robots/ozobot.html

OzoBlockly

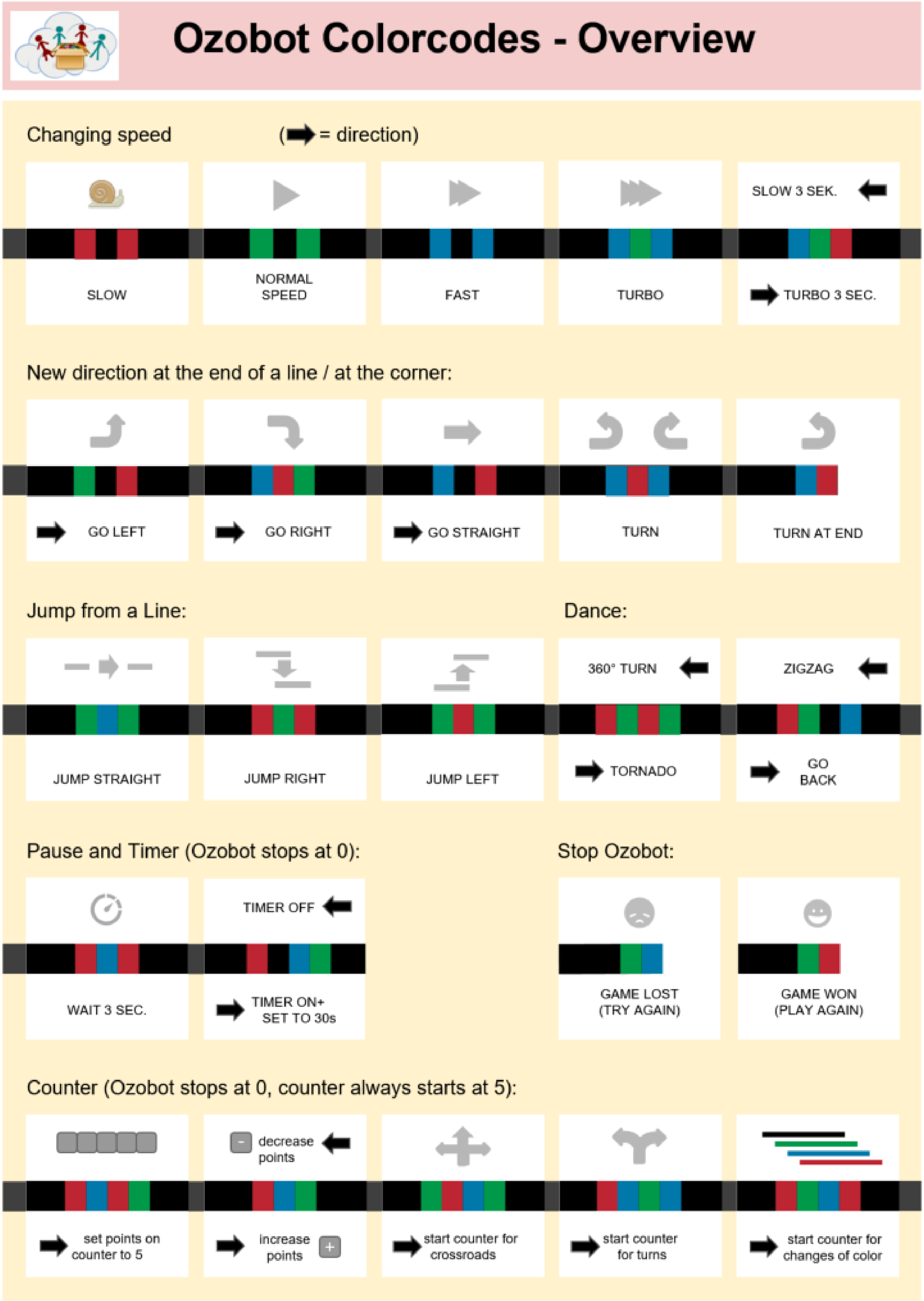
Ozobots can also be programmed using "OzoBlockly", a visual programming language. **There is a great, totally self-explanatory website**, ozoblockly.com. The website presents the programming language as well as the tasks in 5 levels of difficulty, thus addresses students from Kindergarten to high school level. We didn't cover it in the task-sheets because it is so self-explanatory - and a world in its own. It depends on the availability of computer/tablet-resources whether it shall be shown to the students or not.

Apps: Ozobot Evo, OzoGroove, OzoParcours …

There are several apps that can be used with Ozobots. Let your students check them out, if computer/tablet-resources are available.

Teacher Handout BOX 4: Programming - Ozobot

Colorcodes - overview



Ozobot Colorcodes - Overview

Changing speed (➡ = direction)

SLOW

NORMAL SPEED

FAST

TURBO

SLOW 3 SEK. ⬅ / ➡ TURBO 3 SEC.

New direction at the end of a line / at the corner:

➡ GO LEFT

➡ GO RIGHT

➡ GO STRAIGHT

TURN

TURN AT END

Jump from a Line:

JUMP STRAIGHT

JUMP RIGHT

JUMP LEFT

Dance:

360° TURN ⬅ / ➡ TORNADO

ZIGZAG ⬅ / ➡ GO BACK

Pause and Timer (Ozobot stops at 0):

WAIT 3 SEC.

TIMER OFF ⬅ / ➡ TIMER ON+ SET TO 30s

Stop Ozobot:

GAME LOST (TRY AGAIN)

GAME WON (PLAY AGAIN)

Counter (Ozobot stops at 0, counter always starts at 5):

➡ set points on counter to 5

decrease points ⬅ / ➡ increase points

➡ start counter for crossroads

➡ start counter for turns

➡ start counter for changes of color

Fig. 2

For figure sources and references please refer to List of Figures.

Teacher Handout BOX 4: Programming - Ozobot

Goals:

- *Engage in hands-on work with robots*

Skills/Competences addressed:

- *IT, Robotics, Programming*
- *Logical thinking*
- *Creativity*
- *Working in a team*

Connection to other boxes:

This box should be done after IT101

Teacher Handout BOX 4: Programming - Topic Minecraft

Fig. 3

IMPORTANT:

As you certainly do already know: Computers and programs seldomly do exactly what we expect them to do :-(

On the last nine pages of this document you find the Minecraft PI API-documentation

The first 4 pages of this document describe the Minecraft-part of the programming-box - the other 15 pages are the API reference of Minecraft.

What is Minecraft?

Minecraft is a sandbox building game, created by Mohjang and recently bought by Microsoft. The Raspberry Pi comes with a special free version, called Minecraft-PI. This Minecraft-Pi version is more or less the same as the commerical one. Some functions are removed (animals, enemies, pets), but others are added, especially:

- ***Local multiplayer***: This offers your students the opportunity to play together in the same world! If they play together, the world will be saved on the Raspberry, on which is was originally started (the one, that is the „server" for the others). To enter multiplayer mode, have a look at the „game menu" and its 2 buttons: „start game" lets you select the world; „join game" searches the local network for existing, already running games (on other Raspberry Pis) and lets you join one.

- ***Python interface***: this allows programming the minecraft world … see below

For figure sources and references please refer to List of Figures.

Some Minecraft and Python internals

Saving the world

When quitting the game (press ESC and select „Quit to title“), the current world will be saved automatically (there is also an auto-save function that saves from time to time while playing the game). Unfortunately you cannot give names to your worlds (instead a hyphen is added to the word „world“, thus you will have: „world“, „world-“, „world--“, …). You see your worlds in the „Select world“-screen (after having clicked „start game“ in the title screen), by pressing and holding the left mouse-button on a world and moving the mouse to the left or right side. Select then the world that you want to start. (The worlds are shown in order of last usage.)

Positioning in Minecraft

Minecraft uses a coordinate system with x and z as the traditional x- and y-axis. The y-axis in Minecraft represented the height. The coordinates of the player are displayed in the top left corner. If your students are not familiar with the concept of coordinates, you might want to add an activity on a 2-dimensional coordinate system, e.g. using a chessboard, play Battleships, using tiles on the floor, etc.

Python API Libary

Minecraft PI uses a special Python API Libary, which is also compatible with the PC-Minecraft Plugin RaspberryJuice. To get an overview what classes and functions this library has take a look at: http://www.stuffaboutcode.com/p/minecraft-api-reference.html or the appendix of this document.

Minecraft/Python environment

As soon as Minecraft is running, Minecraft holds the mouse. Press tab to let it go, or press Escape for the “game menu”.

While you are in the game menu you can move the minecraft window by moving the Mouse on the Titlebar of the Minecraft-window (Note: This might be hidden behind the game menu.)

What is Python3? (also referred as “the shell”)

Python3 is an interactive tool to tell Python which commands to exectute. After each line (and pressing return!) Python evaluates and executes the command in this line. Sometimes this can „be seen“ (e.g. at the Minecraft chatline), sometimes this happens in silence (e.g. the import-command, which loads code from another module).

NOTE: If your students start to write long programs, than it might be useful to save them. THIS SHOULD **NOT** BE DONE VIA SAVE-AS IN Python3, BUT:

1. start the shell
2. open a new file (menu File / new)

3. write your program

4. save the file

When you want to use the file:

1. start the shell

2. open the existing file (menu File / open)

3. press F5 or menu Run / start module

Taskgroups and tasks

- The first group of tasks (4c1) teaches the basics about Minecraft (start, controls, basic blocks)
- The second group of tasks (4c2) introduces python and shows basic interaction between python and minecraft
- The third group of tasks (4c3) deals with the creation, positioning and some special features of blocks.
- The fourth group (4c4) introduces the concept of loops ("WHILE") and decision making ("IF") by creating a trail of flowers behind the player.
- The fifth group (4c5) introduces TNT, water and lava - together with their speical behavior and remarkable side effects.
- The sixth group (4c6) encourages students to document their best Minecraft experience.

In the following table you will see the programs from the different tasks and a short description of their expected behavior:

4c2.1	The Minecraft chat shows the two posted messages, „Hello Minecraft!" and „The second Test!"
4c2.2	The player is moved 10 blocks in x and 10 blocks in z direction.
4c2.3	The player is moved 20 blocks into the air
4c3.1	Blocks of stone, grass and dirt will be created in a way that the player is between two of them and underneath one
4c4.1	A flower will be planted behind the player (when he moves). The second program produces only one flower.
4c4.2	A flower will be planted behind the player if the block underneath him is grass.

4c5.1	A TNT block gets spawned behind the user. Note: the last “1” in mc.setBlock(x,y,z,46,1) makes the block explodable when it gets hit. The second program produces TNT blocks when player is moving on grass. If the player hits the last TNT block, a chain reaction of exploding TNT blocks should happen (depending a bit on where the bock-chain was spawned).
4c5.2	A water block gets spawned next to the player. In the second program a square of 3x3 water blocks is created.
4c5.3	A lava block gets spawned near the player. In the second program a square of 3x3 water blocks gets spawned first and then on top of the middle of the water blocks a lava block gets created. The lava starts to flow down into the water, creating some cobblestone blocks.

Further material

Links

http://www.stuffaboutcode.com/p/minecraft-api-reference.html (see Annex)

Some more material for interested students can be found here - it is a step-per-step manual about how to program the “whac-a-mole-2 game with Minecraft and Python: https://www.raspberrypi.org/learning/minecraft-whac-a-block-game/worksheet/

Goals:

- learning the basics of programming

Skills/Competences addressed:

- *IT, Programming, Handicraft*
- *Logical thinking*
- *Creativity*
- *Working in a team*

Connection to other boxes:

This box should be done after IT101

ANNEX

More about the Python Library:

http://www.stuffaboutcode.com/p/minecraft-api-reference.html

Teacher Handout BOX 4: Programming - Lego® Education WeDo 1.0

Lego® Education WeDo 1.0 is a great educational product from LEGO®.

It consists of “normal” LEGO®-bricks and some special bricks like motors and sensors plus a “hub” that allows to program the behaviour of the motors and sensors. The program is written on a computer (or tablet) and transferred to the “hub” either by cable (WeDo 1.0) or bluetooth (WeDo 2.0). We use WeDo 1.0 (the cable-version), which is easier to use: just plug it in and it works!

NOTE: If you search the internet for material, make sure, that you do not get confused between 1.0 and 2.0. As most pages about 1.0 where written BEFORE 2.0 came to the market, WeDo 1.0 is mostly referred to “WeDo” and not “WeDo 1.0”.

Programming

For programming we can either use Scratch or the “native” WeDo-Software, which unfortunably is not available on all operating systems. The tasksheets only cover Scratch.

For programming with Scratch just check out the first task sheet and see how easy it is: Add the motor-blocks and use them in your program. Unfortunately Scratch cannot reliably control the sensors, therefore we recommend to focus on using the motor only. (All models make sense without the sensors as well).

Support

https://education.lego.com/en-us/support/wedo

This is the main support-page for WeDo 1.0: There is a complete User guide available, which you can also download as pdf.

All construction manuals are available online as well:

https://education.lego.com/en-us/support/wedo/building-instructions

Main topics and tasks

The **Lego® Education WeDo 1.0** -tasksheets have 3 topics:

Topic 1: Introduction

A very simple model for a fan, that also shows how to programm with Scratch.

This topic should be done before starting with topic 2.

Teacher Handout BOX 4: Programming - Lego® Education WeDo 1.0

Topic 2: 11 models

After the basic introduction of topic 1, the children can choose which of the 11 model(s) to build. They are different in topic and engineering - and they all include some further exercises.

Some models might be too difficult - at least at the beginning. However, experience shows, that even those children who had not played with LEGO® bricks before, quickly develop LEGO®-building skills.

Topic 3: Your own project

Many children are more interested in realising their own projects instead of following given instructions. However, we would recommend that they do at least one of the 11 models, as this teaches them a lot about the possibilities that are inside those LEGO®-bricks.

Goals:

- *Engage in hands-on work with real-life engineering and programming tasks*

Skills/Competences addressed:

- *IT, Engineering, Programming*
- *Logical thinking*
- *Creativity*
- *Working in a team*

Connection to other boxes:

This box should be done after IT101

	Box: Programming	Topic: MakeyMakey	
	Task 4a1.1:	Electricity	

What is electricity?

What do we need it for?

How does it move?

Take a paper and draw how electricity moves.

Next to it write a list of things you need electricity for.

Upload a video of your paper.

Video: https://www.youtube.com/watch?v=QZPURSF5iH4

Fig. 1

Fig. 2

Fig. 3

For figure sources and references please refer to List of Figures.

	Box: Programming	Topic: MakeyMakey	
	Task 4a1.2:	Basics - Make electrons move	

1. Connect the MakeyMakey with a computer (using a USB-cable). If electricity arrives, the small lamps will turn green.
2.
3. Take two of the cables, that have "alligators" at their ends

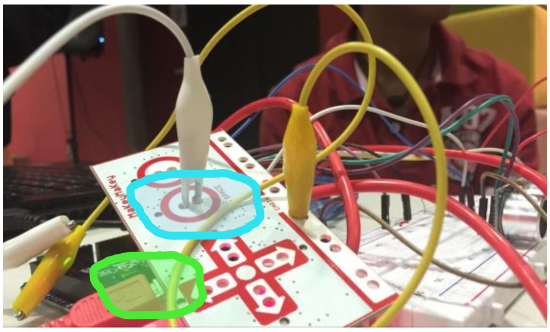


Fig. 4

4. Let the alligator of one cable bite into "Earth" and one of the other cable into "Space".

5. Hold the other 2 alligators together or on an object (e.g. a stone).

6. If the green light next to "Space" lights up, then you have created an electric circuit, which means that electrons move along.

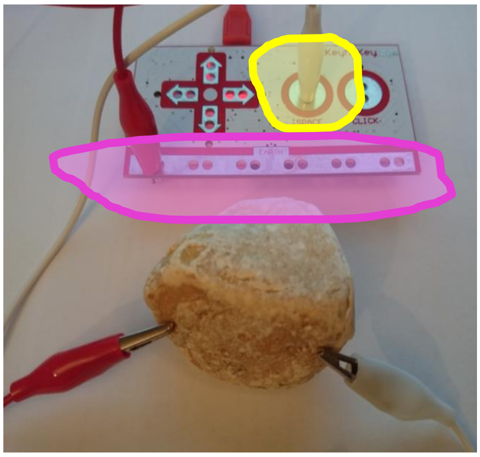


Fig. 5

Replace the stone with something else, e.g. paper. Does this make the electrons flow? What about wet paper? Try this with as many materials as you can find in your classroom!

Make a list of all the materials that make the lamp light up green! These are *conductive materials.* Upload a picture of your list.

Make a video in which you use one of these materials to make the lamp light up. Share this video with us.

Video: https://www.youtube.com/watch?v=TSnanWOe-Yc

For figure sources and references please refer to List of Figures.

	Box: Programming	Topic: MakeyMakey	
	Task 4a1.3:	Basics - Can you be part of an electric circuit?	

Connect the MakeyMakey to a computer via USB.

Then take two “alligator”-cables in two different colors, e.g. red and blue.

Connect two alligators of different cables to “Earth” and “Space” (as you did in the previous task).

NOW: What happens if you take one alligator of one cable in one hand and the other alligator of the other cable in your other hand?

Take a picture of you and the lamp as it lights up.

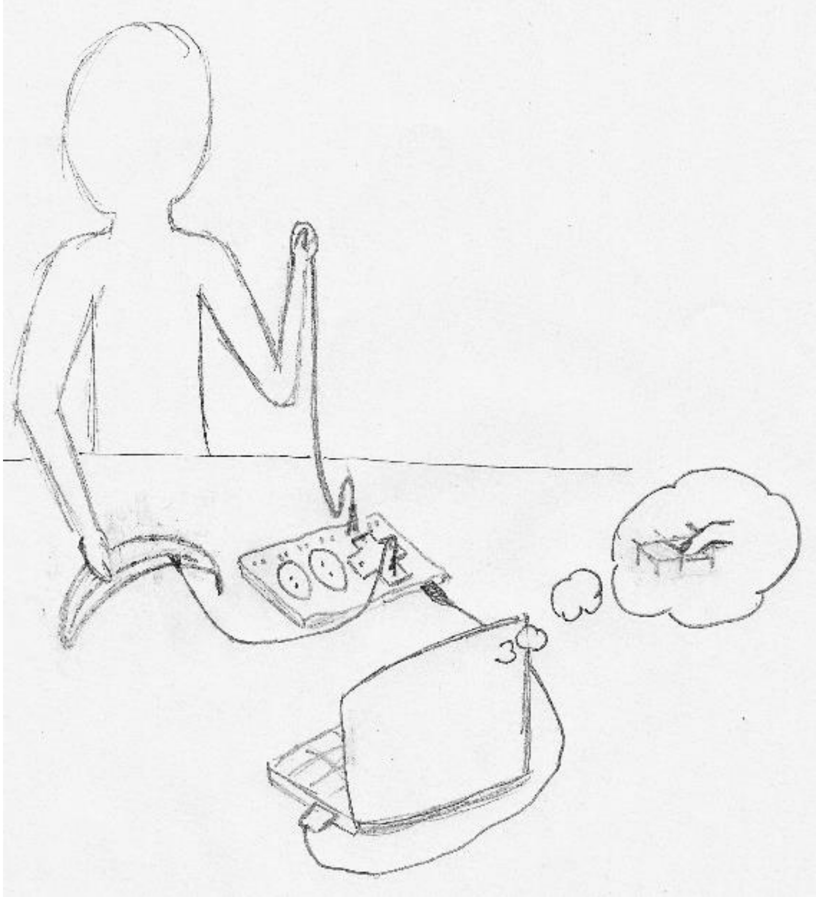
Can you do this together with your friends holding hands or touching each others neck? Take a picture of all of you making the lamp light up together.

NOTE: The amount of electricity that flows via the MakeyMakey is very low - therefore it cannot harm you. Don’t try this with other electronic devices!

Teacher’s note: a full lesson plan on this topic can be found here: https://makeymakey.com/lessons/simple-circuit-challenge/

	Box: Programming	Topic: MakeyMakey	
	Task 4a1.4:	Basics - Through the eyes of the computer	

Look at the picture - it shows what the computer „thinks“ when the electric circuit is closed and runs through the “arrow-up” key:

 Fig. 6	• the MakeyMakey is connected to the computer via USB-cable • 2 alligator-cables are connected to the MakeyMakey: one to “Earth” and one to “Arrow-up” • a person holds one cable, that is connected to “Earth” • the other cable is attached to a banana • the person touches the banana • **the computer thinks, that … ?**

For figure sources and references please refer to List of Figures.

	Box: Programming	Topic: MakeyMakey	
	Task 4a2.1:	Fruit piano	

Step 1:

Attach the MakeyMakey to a computer via USB-cable. Then open a texteditor.

Take a cable: Attach one end to “Space” and the other end to a fruit (e.g. a banana). Take another cable: Attach one end to “Earth” and hold the other end in your hand.

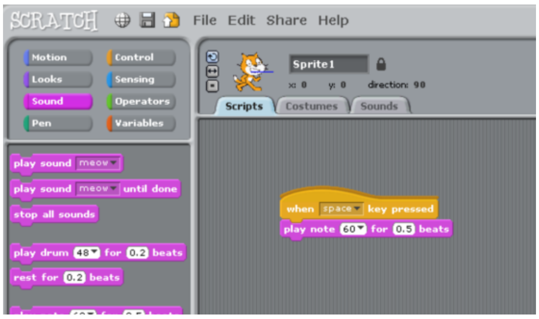
With your free hand: hit the fruit. What happens in the texteditor?

Step 2:

Open Scratch.

Drag “When space key pressed” out (it is in the category “Control”)

Drag “play note … for … beats” out of the category “Sound” and attach it to “when space key pressed”



```
when space key pressed
play note 60 for 0.5 beats
```

Fig. 7

Hit the banana again. What happens ? (Don’t forget to hold the “Earth”-cable in your other hand!)

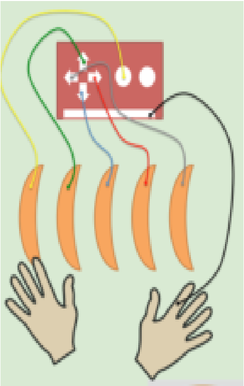
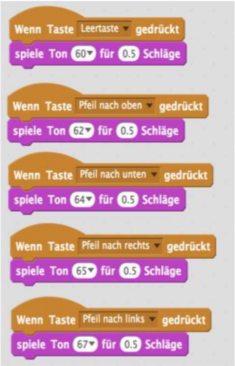


Fig. 8



```
Wenn Taste Leertaste gedrückt
spiele Ton 60 für 0.5 Schläge

Wenn Taste Pfeil nach oben gedrückt
spiele Ton 62 für 0.5 Schläge

Wenn Taste Pfeil nach unten gedrückt
spiele Ton 64 für 0.5 Schläge

Wenn Taste Pfeil nach rechts gedrückt
spiele Ton 65 für 0.5 Schläge

Wenn Taste Pfeil nach links gedrückt
spiele Ton 67 für 0.5 Schläge
```

Fig. 9

For figure sources and references please refer to List of Figures.

	Box: Programming	Topic: MakeyMakey	
	Task 4a2.2:	Game controller	

Fig. 10

Fig. 11

This game controller can be used with any “jump-and-run”-type game. If you don’t have one on your computer, download it (e.g. from supertux.lethargik.org) or use an online game (e.g. http://www.funnygames.in/game/tomb_runner.html). Check which keys are needed to play it! (Usually it is space and arrow-keys).

For Raspberry PI: You can use the Squirrel-game: Goto to: Menu / Games / Python Games, then select “squirrel” from the list.

Now you can build your controller:

In front of the table with the computer put five square fields of strong tinfoil, like in the picture.
There should be enough space between the fields to make sure 5 people can stand next to each other. Connect each field with the Makey Makey using a cable and a clip. After that stabilize the fields using scotch tape

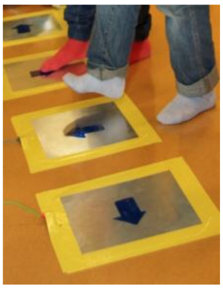


Fig. 12

Hold hands with the players next to you.
The last player holds the “Earth”-cable!

Start the game and play together. Step on the fields on the floor to press the needed keys.
Take a video and share it with us - and be patient with your fellow-players ☺

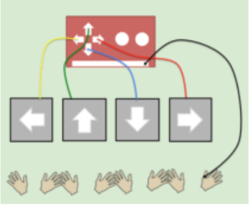


Fig. 13

For figure sources and references please refer to List of Figures.

	Box: Programming	Topic: MakeyMakey	
	Task 4a2.3:	Catching fish	

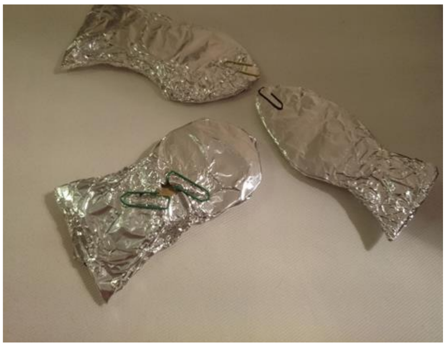


Fig. 14

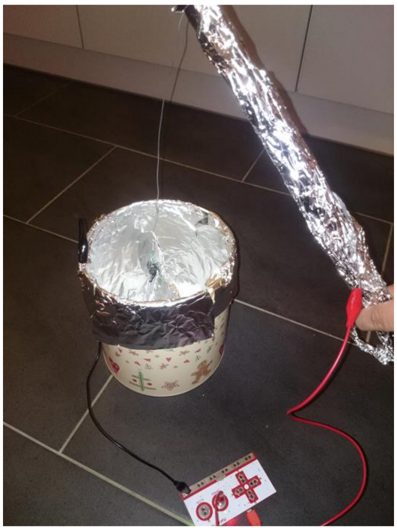
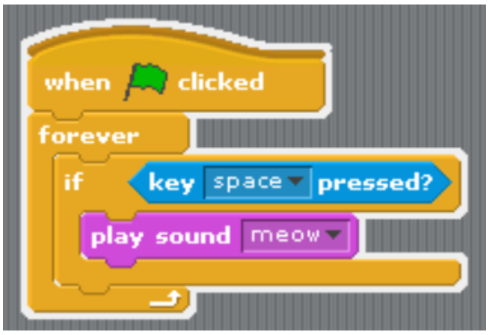


Fig. 15

Create a bowl for the fish. Its walls (not the floor!!) have to be covered with tinfoil. The foil is connected by cable to “Space”. To make the fishing rod you need a long stick and a wire. Connect the two and let the wire hang from the end of the stick. Pack the stick in tinfoil. On the other side of the wire put a magnet.

Cut the fish using carton and also pack them in tinfoil. To be able to catch them the fish also needs to include some metal, like a paper clip. Put the fish at the bottom of the bowl.

Then open scratch. Connect the closing of the electric circuit with a sound, like you did with the fruit piano.



```
when 🏁 clicked
forever
  if key space pressed?
    play sound meow
```

Fig. 16

Now make sure you get all the fish without touching the wall!

Make a picture and/or a video and upload it.

For figure sources and references please refer to List of Figures.

	Box: Programming	Topic: MakeyMakey	
	Task 4a2.4:	Basketball	

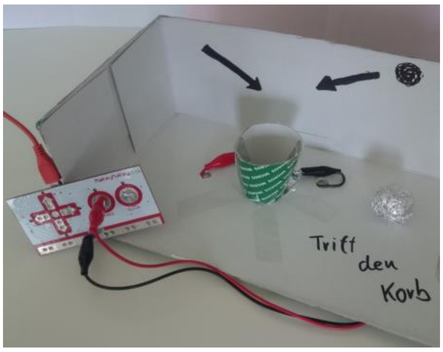


Fig. 17

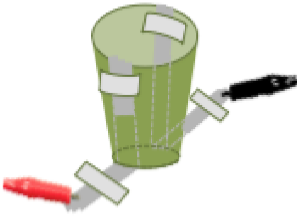


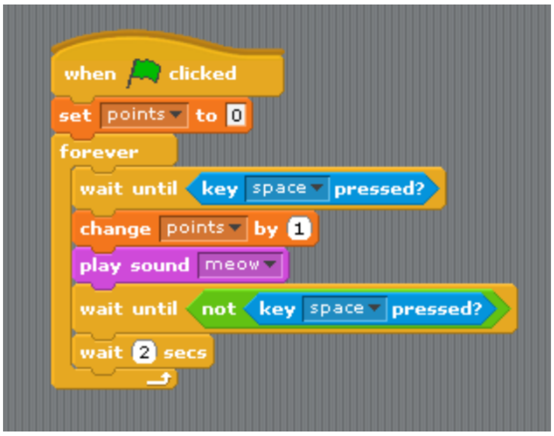
Fig. 18

Take a shoe box and cut out one of its walls. Make a small funnel from paper or cardboard (cut and glue). Take two stripes of tinfoil and fold them in a way that two multi-layered bands remain. From two sides of the funnel add the two stripes by pushing them along the wall and up. On the top of the wall you fold and fix the stripes. Use scotch tape to also fix the other ends of the stripes on the floor of the box. Connect those ends with the MakeyMakey (“Space” and “Earth”).

Then make small balls of tinfoil. The have to be big enough to touch both walls of the funnel at the same time.

Now start Scratch. Try to create a program, that counts points, e.g. like this:

- get the variable “points” from the category “Variables”
- let the program wait until “Space” is pressed (until the ball touches both walls and closes the electric circuit.
- then increase the number of points and play a sound.
- wait until “Space” is no longer pressed (the ball is removed), but wait 2 seconds before going on; otherwise the removal of the ball could cause another point



```
when [flag] clicked
set points to 0
forever
  wait until key space pressed?
  change points by 1
  play sound meow
  wait until not key space pressed?
  wait 2 secs
```

Fig. 19

We are looking forward your picture and/or video !

For figure sources and references please refer to List of Figures.

	Box: Programming	Topic: MakeyMakey	
	Task 4a2.5:	Show-jumping course	

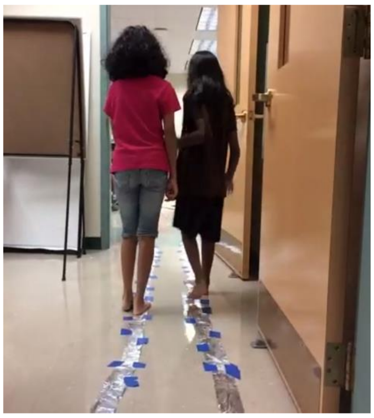


Fig. 20

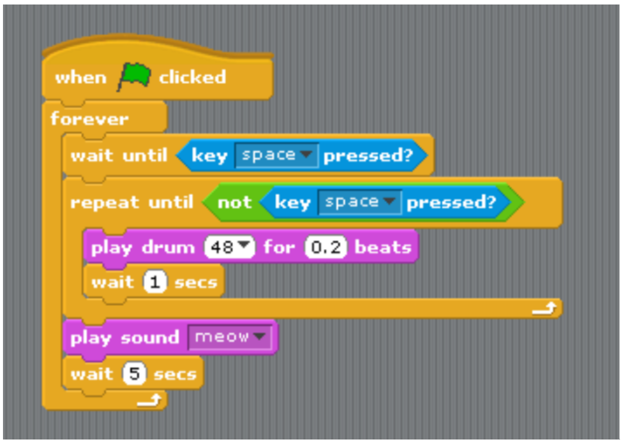
Tape a long line of tinfoil to the floor. Make sure they do not touch each other. You can add obstacles, such as pillows, chairs etc. Attach to each line a cable connecting them to “Space” and “Earth”.

The 2 players shall walk along the parcours. They always have to stay on the tinfoil and must never lose the partner’s hand; otherwise the electric circuit will be interrupted.

Open Scratch and create a program, that waits until both players are ready to play (stand on the tinfoil and take each others hand - this will trigger the “Space”-key).

Then the program should continosly create a noise - until the “Space”-key is no longer pressed. If this happends, another noise shall be created.

This is an example program:



```
when [flag] clicked
forever
    wait until <key [space] pressed?>
    repeat until <not <key [space] pressed?>>
        play drum (48) for (0.2) beats
        wait (1) secs
    end
    play sound [meow]
    wait (5) secs
end
```

Fig. 21

We are looking forward your picture and/or video !

	Box: Programming	Topic: MakeyMakey	
	Task 4a2.6:	Hot wire	

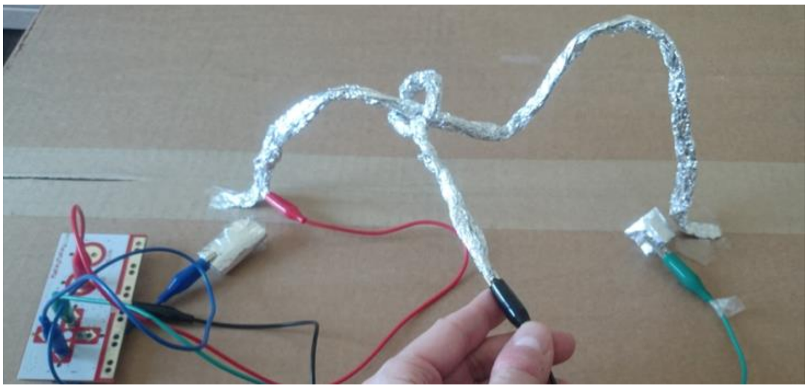
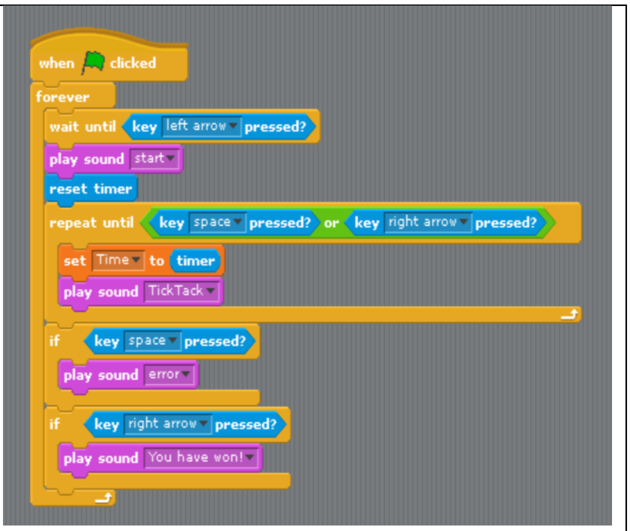


Fig. 22

In this game, you must follow the “hot wire” with a loop on a stick as fast as possible - but you must not touch the wire!

Create the “hot wire”: Take a lot of tinfoil and role it into a wire. Tape the wire to the floor or the table. Then create the stick: With another piece of tinfoil make another wire and wrap it around the first one, like in the picture. Using carton and tinfoil create a start- and endfield. It must be possible to touch these two parts with the stick, without touching the main wire. Connect each part to the Makey Makey: Start and end to e.g. arrow-left and arrow-right; the stick to “Earth”.

What does this program do? Try to understand it and create your own program! You can record your own sounds!	 Fig. 23

```
when [flag] clicked
forever
  wait until <key [left arrow] pressed?>
  play sound [start]
  reset timer
  repeat until <<key [space] pressed?> or <key [right arrow] pressed?>>
    set [Time] to (timer)
    play sound [TickTack]
  if <key [space] pressed?>
    play sound [error]
  if <key [right arrow] pressed?>
    play sound [You have won!]
```

Watch this video for inspiration https://www.youtube.com/watch?v=EZh83tDgWxg … and share your own video with us!

For figure sources and references please refer to List of Figures.

	Box: Programming	Topic: MakeyMakey	
	Task 4a2.7:	Stair piano	

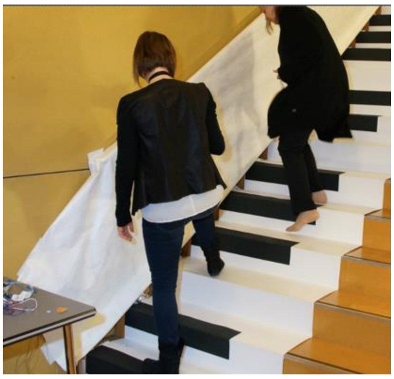


Fig. 24

This is quite a big project!

Try to imitate what happens in this video: https://www.youtube.com/watch?v=21sBo5fP0S8&t=290s

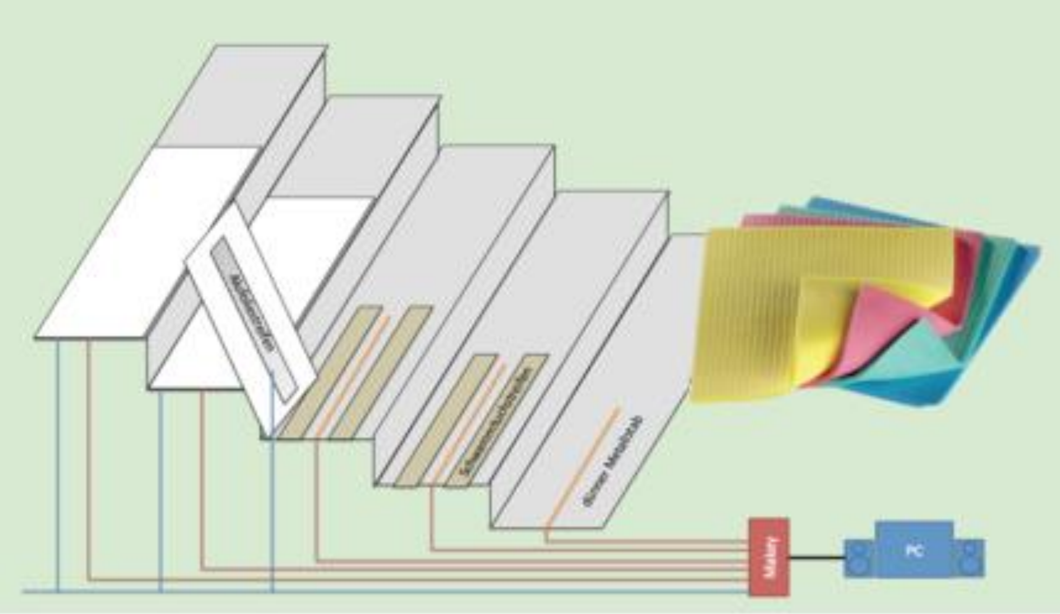


Fig. 25

Make your own rhythm with the stair piano and upload a video of it!

For figure sources and references please refer to List of Figures.

Box: Ozobot	**Topic: Getting to know your Ozobot !**
Task 4b1.1:	**Take care of your Ozobot**

Ozobots are small robots. Their „eyes“ are 5 little openings on the bottom. They have one power-button: If you take care of them, they will function correctly and you will have a lot of fun with them.

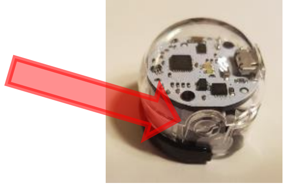


Fig. 30

1. **They need electricity**, which means: You must charge them.

   - As soon as it blinks RED, plug it into a USB-plug via its charger.
   - While loading, it will blink RED/GREEN on low charge, GREEN on normal charge, and SOLID GREEN on full charge.

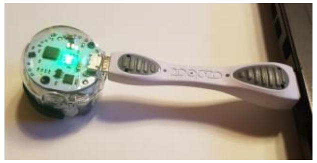


Fig. 31

2. **They need clean wheels**, which means: You might have to clean them from time to time.

   - Take a clean sheet of paper and roll Ozobot back and forth (at least 5 times).

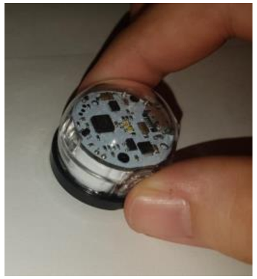
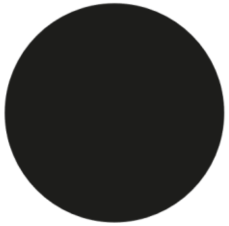


Fig. 32

3. **They need to adjust their cameras** their „eyes“ from time to time (that‘s called „calibration“), which means: You need to calibrate them from time to time:

   - Hold down the power button for 2 seconds, until it blinks white. →
   - Quickly place Ozobot in the middle of the black dot
   - Ozobot will move forward and bink green. If it blinks red, start over again.



NOTE: Follow the QR-code to our „hand-it-in“-app to find links to videos of all 3 activities !

For figure sources and references please refer to List of Figures.

Box: Ozobot	Topic: Getting to know your Ozobot !
Task 4b1.2:	Ozobots love decoration

There are many ways to beautify your Ozobot.

What should **your** Ozobot look like ?

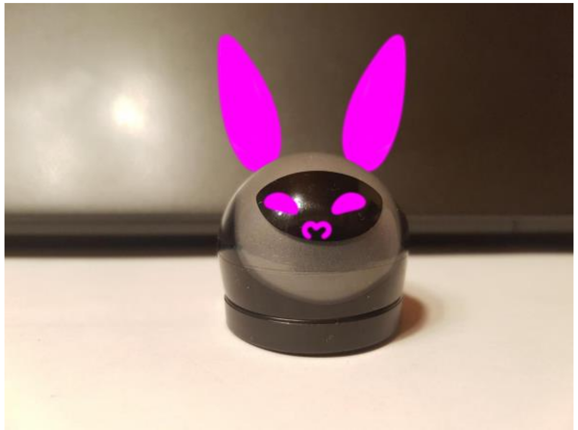


Fig. 33

For figure sources and references please refer to List of Figures.

Box: Ozobot	Topic: Ozobots love lines !
Task 4b2.1	Following a line

Ozobots love lines.
Turn on your Ozobot (push the power-button), place it on the black line, and see what happens.

Draw lines and let Ozobot follow them !

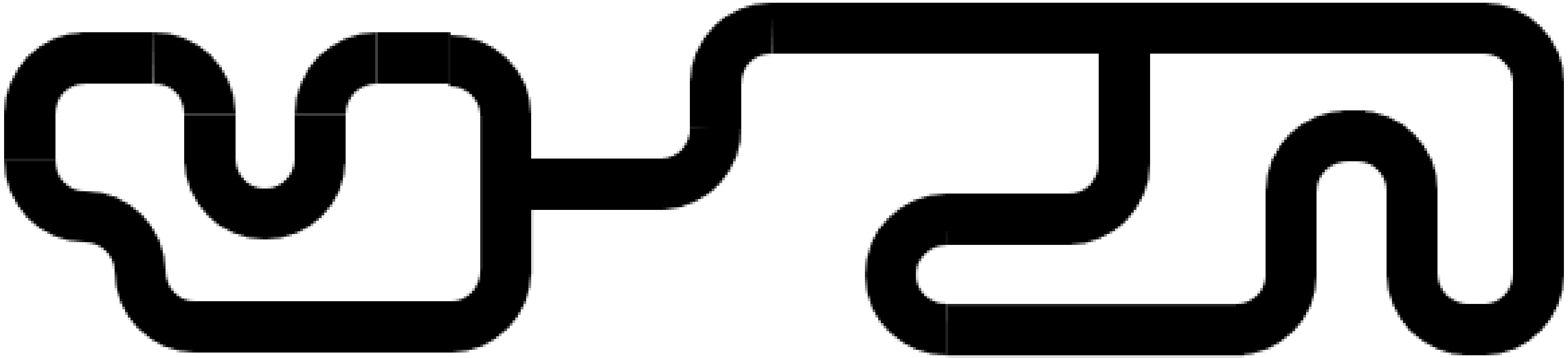


Fig. 34

Does your Ozobot always take the same way ?

For figure sources and references please refer to List of Figures.

Box: Ozobot	**Topic: Ozobots love lines !**
Task 4b2.2	**Your own map!**

Draw your own map !
How big can it be ?

Don‘t forget to upload a picture of it!

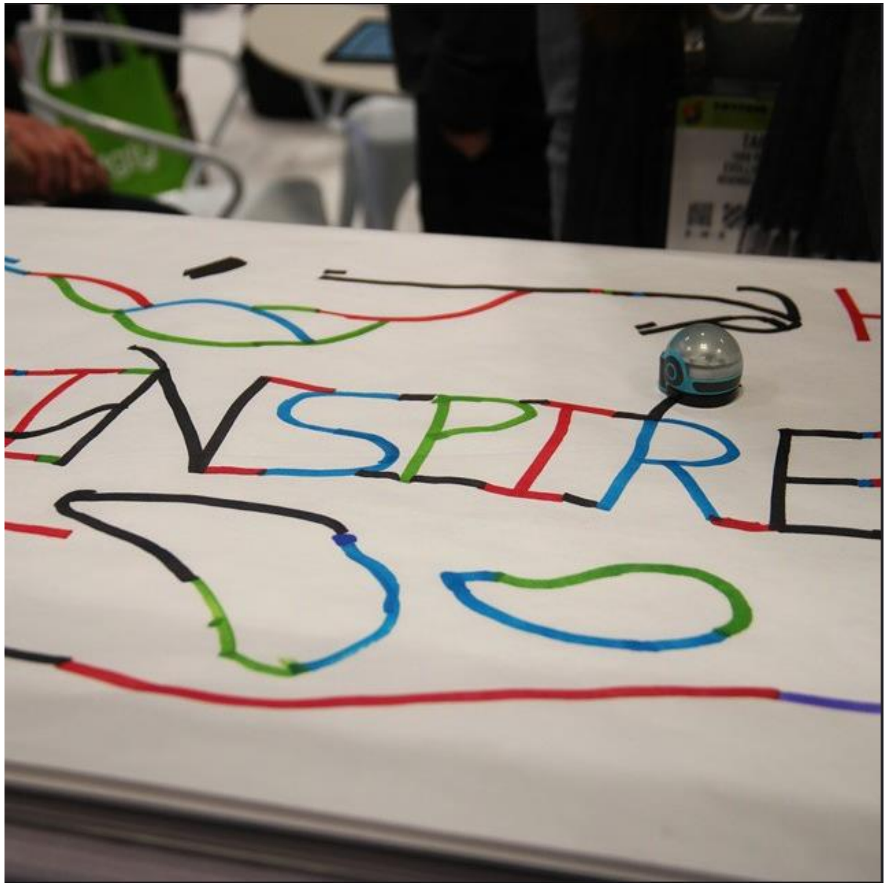


Fig. 35

Do you need ideas for maps?
Your imagination is the limit!

- Atlantis
- Toon city
- The amusement park of your dreams
- the moon,
- …

For figure sources and references please refer to List of Figures.

Box: Ozobot	Topic: Ozobots love lines !
Task 4b2.3	Travelling Ozobot!

Where would your Ozobot like to travel to? To London? To Paris?
Maybe to the USA, as in the picture?

Draw a map of your neighbourhood, your city, your country or of Europe and send your Ozobot on a long, long journey …

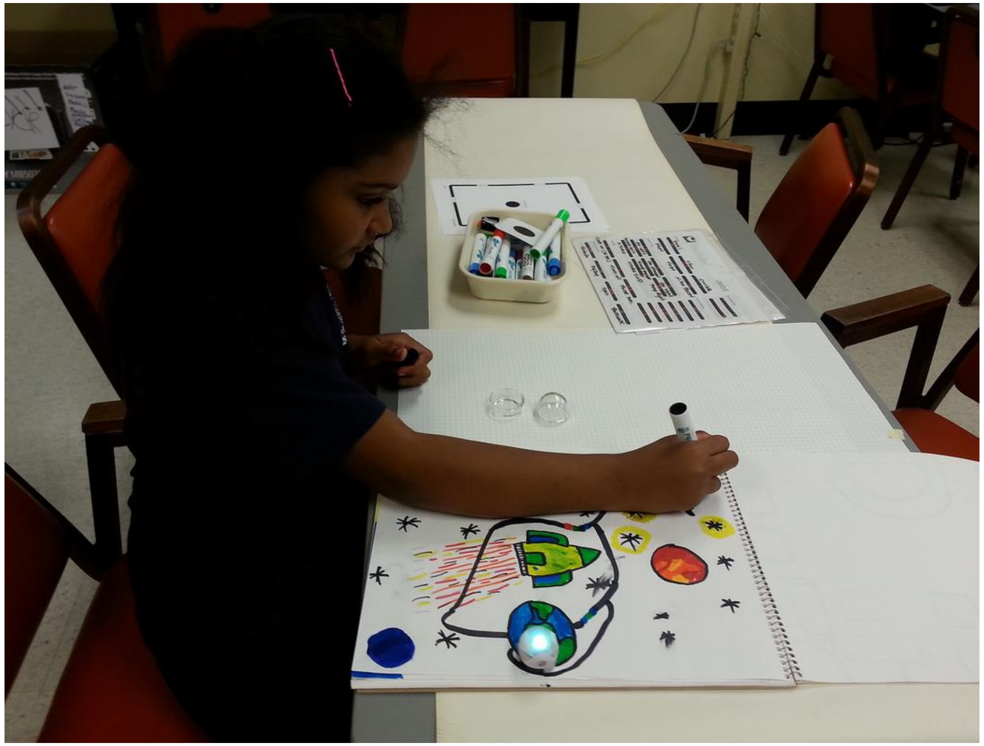


Fig. 36

For figure sources and references please refer to List of Figures.

Box: Ozobot	Topic: Ozobots love lines !
Task 4b2.4	Puzzle-map

When you create your own map, if might happen, that your sheets of paper become too small.
It also might happen, that you have a very nice part, which you would like to use in another map as well.

Why not build up the map out of smaller parts, like a puzzle!

Try it out!

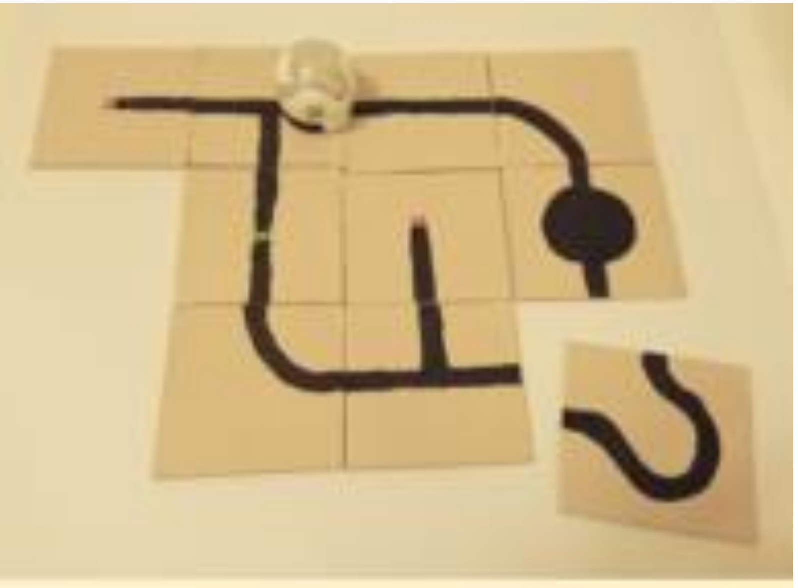


Fig. 37

For figure sources and references please refer to List of Figures.

Box: Ozobot	Topic: Ozobots love colors !
Task 4b3.1	Color-codes for speed

<u>By using special combinations of certains colors</u>, you can tell Ozobot what to do.
What happens at A, B, C, D and E ?

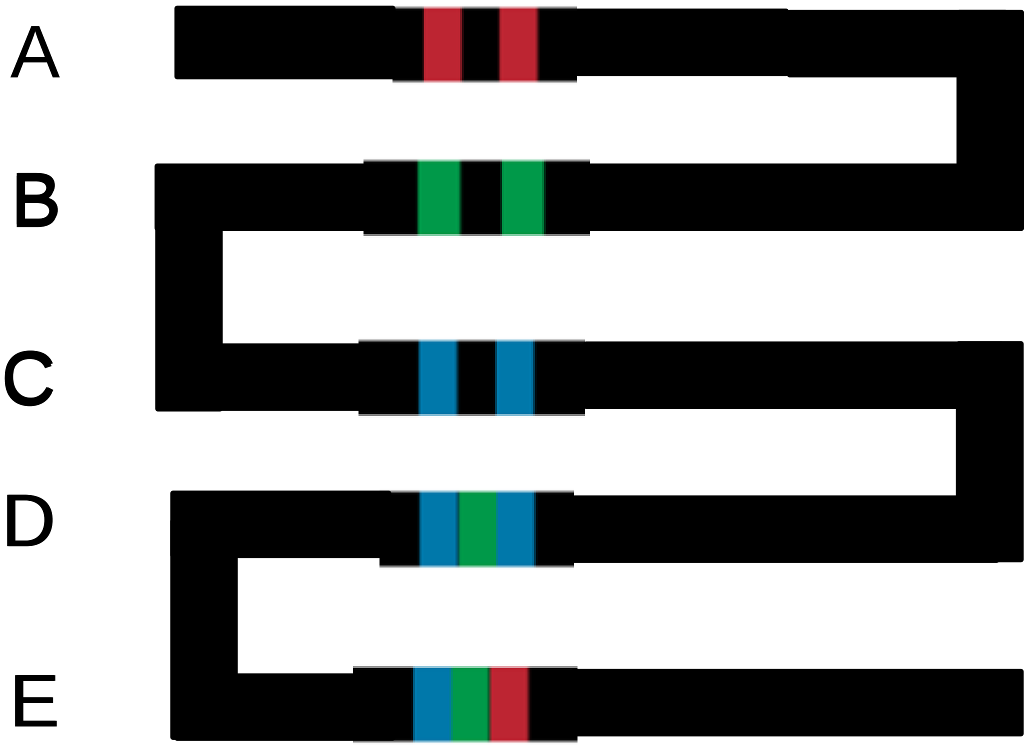


Fig. 38

Upload a picture of your maze or city or racing track or whatever you invent for your Ozobot !

For figure sources and references please refer to List of Figures.

Box: Ozobot	Topic: Ozobots love colors !
Task 4b3.2	Color-codes for direction

What happens, if your Ozobot get to a crossroad?

With color-codes you can tell it what to do!
What do the color-codes tell?

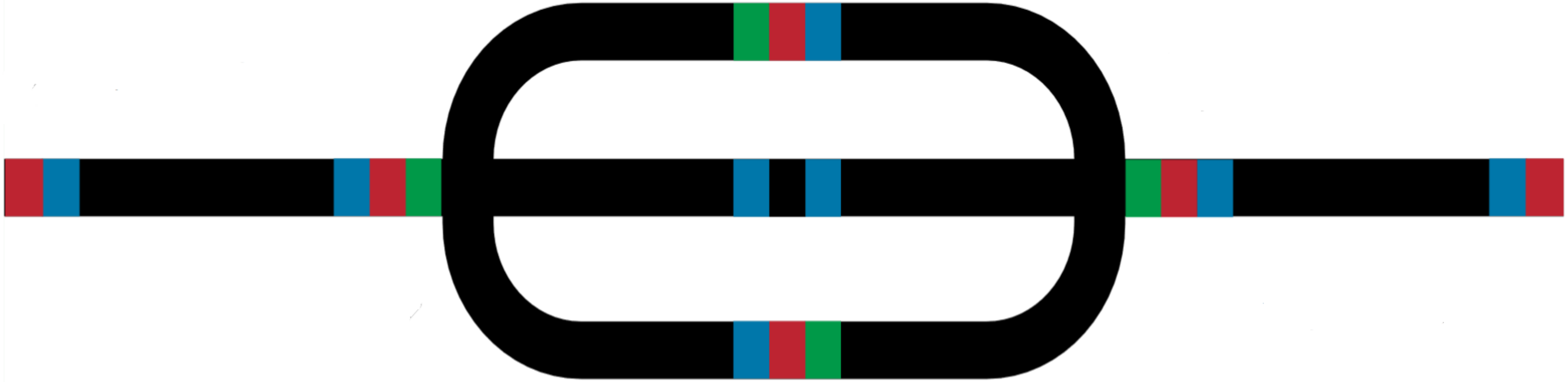


Fig. 39

Upload a picture of your maze or city or racing track or whatever you invent for your Ozobot !

You can use these color-codes to make him behave correctly !

For figure sources and references please refer to List of Figures.

Box: Ozobot	Topic: Ozobots love colors !
Task 4b3.3	Color-codes for reaching the goal - Treasure hunt

Here we have some new color-codes that show, if Ozobot has reached the goal.
Does your Ozobot find the treasure?

Can you create your own treasure hunt?

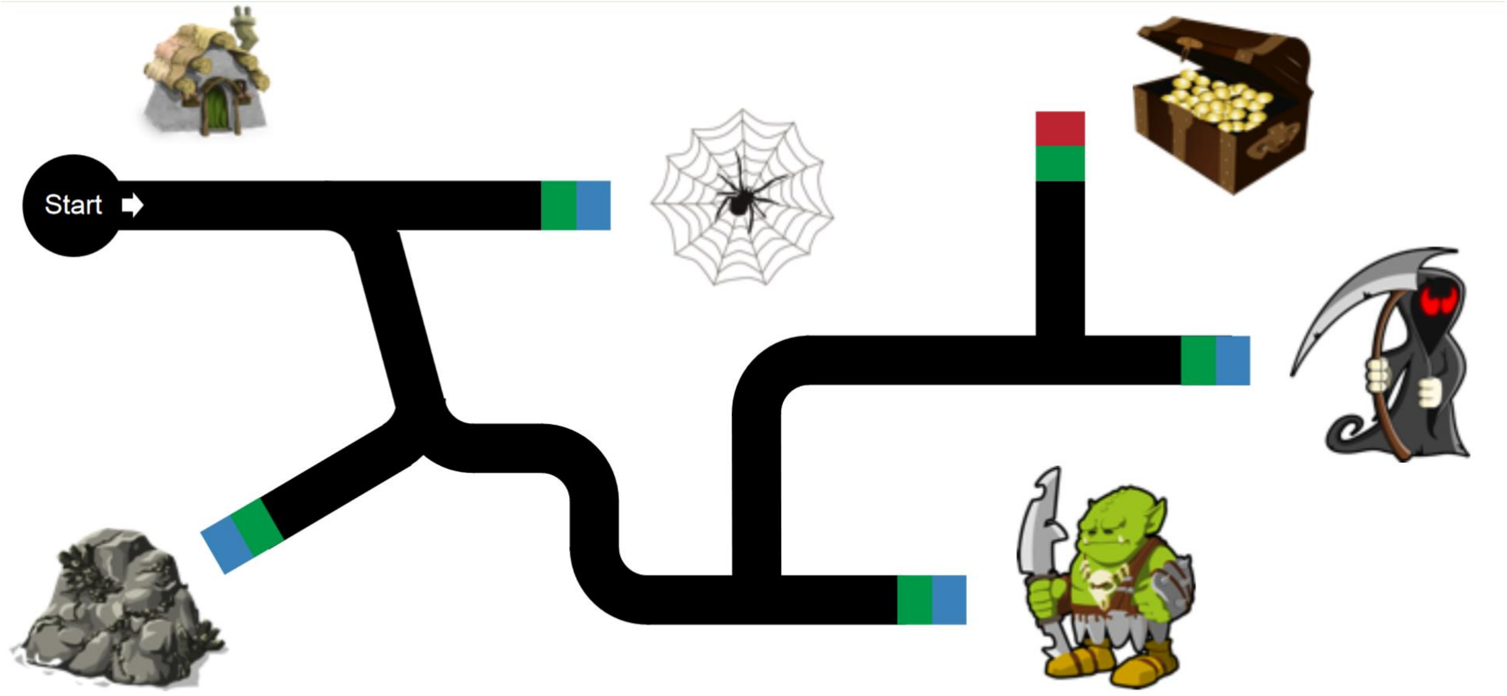


Fig. 40

For figure sources and references please refer to List of Figures.

Box: Ozobot	Topic: Programming with color-codes
Task 4b4.1	Find the way out!

In the previous tasks you learned some of the color-codes that Ozobots can understand. Ask your teacher for a complete list of color-codes!

Help your Ozobot to find its way by filling in the correct color-codes !

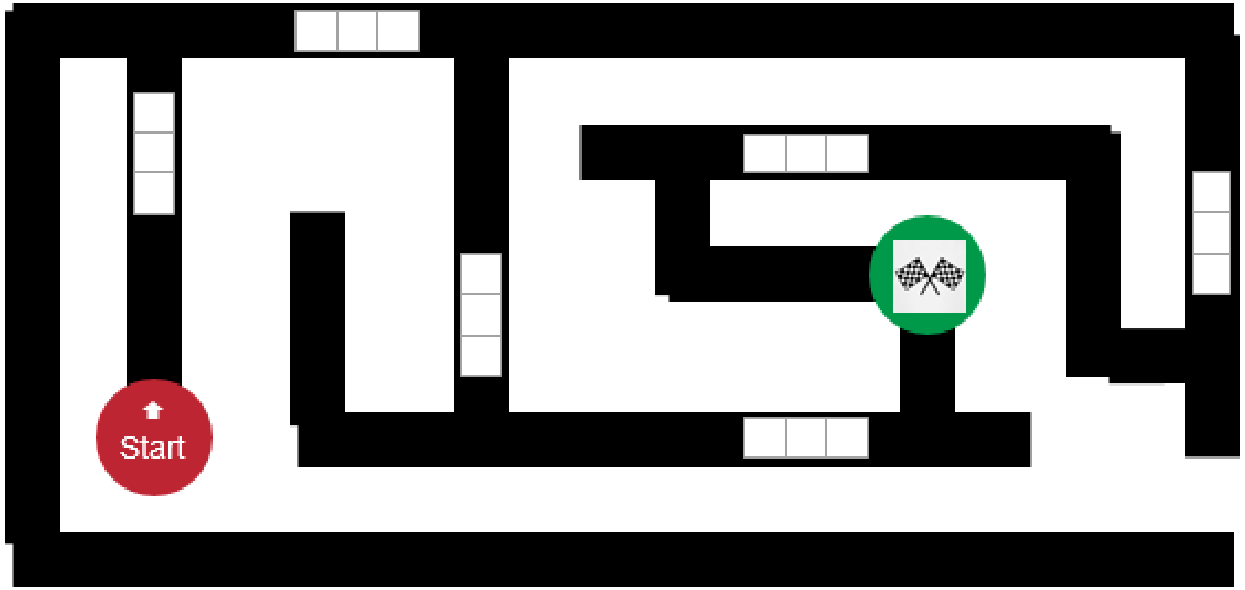


Fig. 41

For figure sources and references please refer to List of Figures.

Box: Ozobot	Topic: Programming with color-codes
Task 4b4.2	Going shopping

Help Ozobot to find its way from home to the shop. Use the right color-codes!

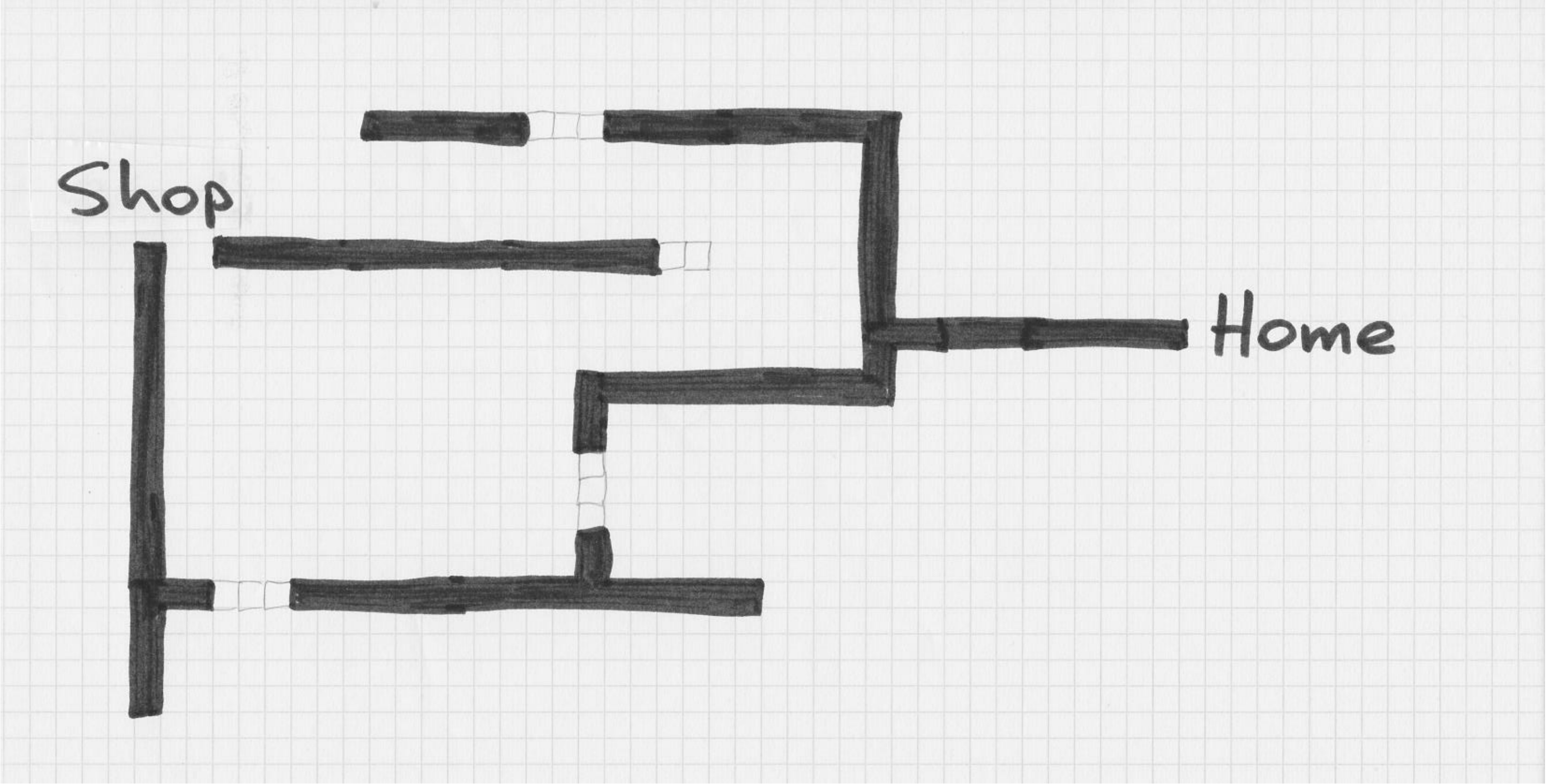


Fig. 42

For figure sources and references please refer to List of Figures.

Box: Ozobot	Topic: Programming with color-codes
Task 4b4.3	Ozobot dance

What about letting 2 or more Ozobots dance together?
Draw a line for each of them – don‘t forget to use color-codes!

Fig. 43

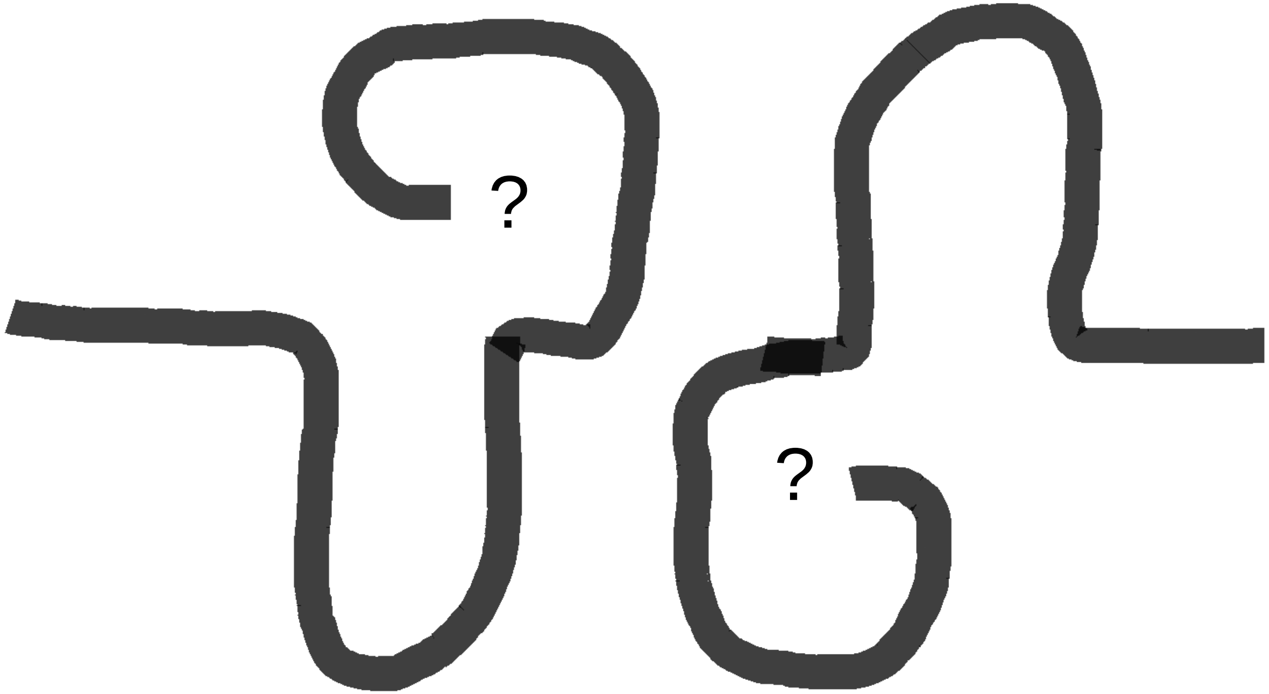


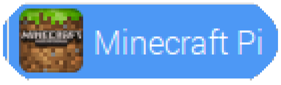
Fig. 44

For figure sources and references please refer to List of Figures.

Box: Programming	Topic: Minecraft
4c1.1	Welcome to Minecraft - controls

Fig. 50

To start Minecraft:



+

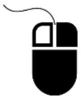


Fig. 52

Fig. 51

Controls:

KEY	ACTION
ESC	Game Menu / close Inventory
TAB	Letgo of the Mouse
W	Go forward
S	Go backward
A	Go left
D	Go right
E	Open Inventory
Space	Jump
1,2,3,… 8	Select item from itembar
Double press Space	Fly / stop flying
Left Mouse	destroy Block
Right Mouse	create Block

For figure sources and references please refer to List of Figures.

Box: Programming	Topic: Minecraft
4c1.2	Welcome to Minecraft - inventory

Minecraft Blocks

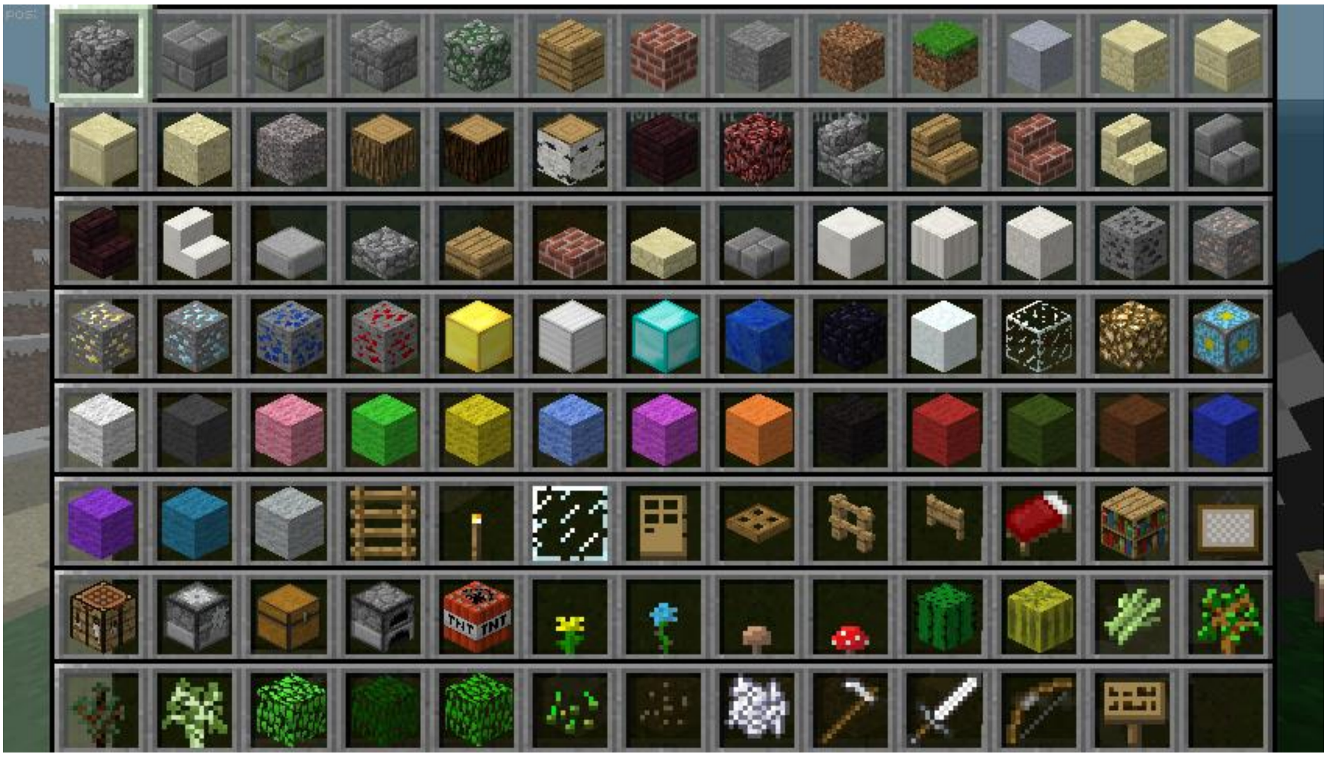


Fig. 53

Discover each block.
How are they different ?
What can you do with them ?

Can you create a block with
the sword in your hand ?

For figure sources and references please refer to List of Figures.

Box: Programming	Topic: Minecraft
4c2.1	Hello Python

-) Start Minecraft, press „start game“, and select a worlds. Then press Tab to let the mouse go.

-) Afterwards start the Python3 Shell:

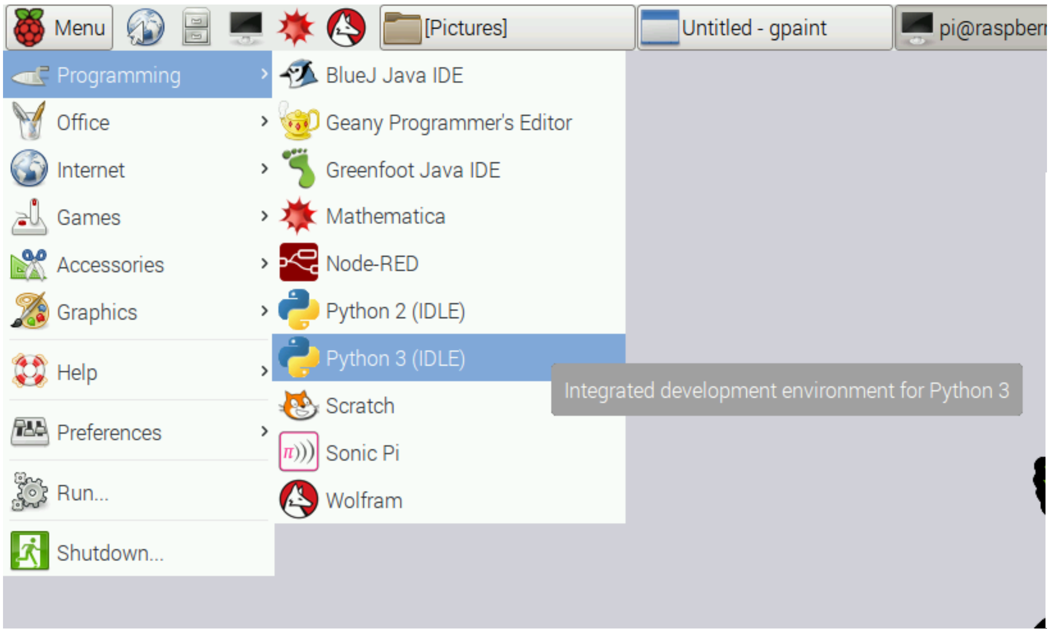
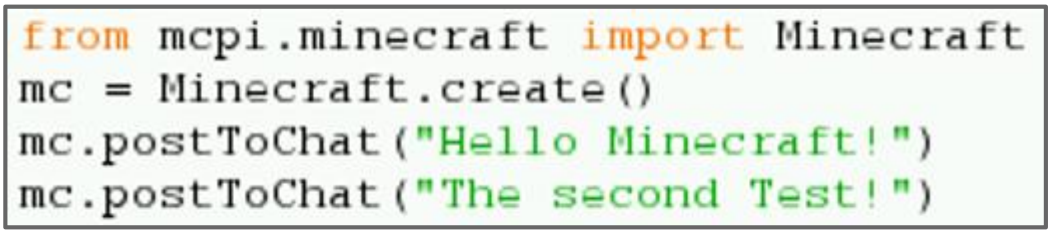


Fig. 54

Enter the following lines:



```
from mcpi.minecraft import Minecraft
mc = Minecraft.create()
mc.postToChat("Hello Minecraft!")
mc.postToChat("The second Test!")
```

What happens in Minecraft?

For figure sources and references please refer to List of Figures.

Box: Programming	Topic: Minecraft
4c2.2	Hello Python - Where am I

-) Start Minecraft and
-) go to some place without hills
-) press Tab
-) Start the Python3-Shell
-) Enter the following text in the Python3 Shell:

```
from mcpi.minecraft import Minecraft
from time import sleep
mc = Minecraft.create()
mc.postToChat("Where are you?")
x, y, z = mc.player.getPos()

mc.player.setPos(x+10, y, z+10)
```

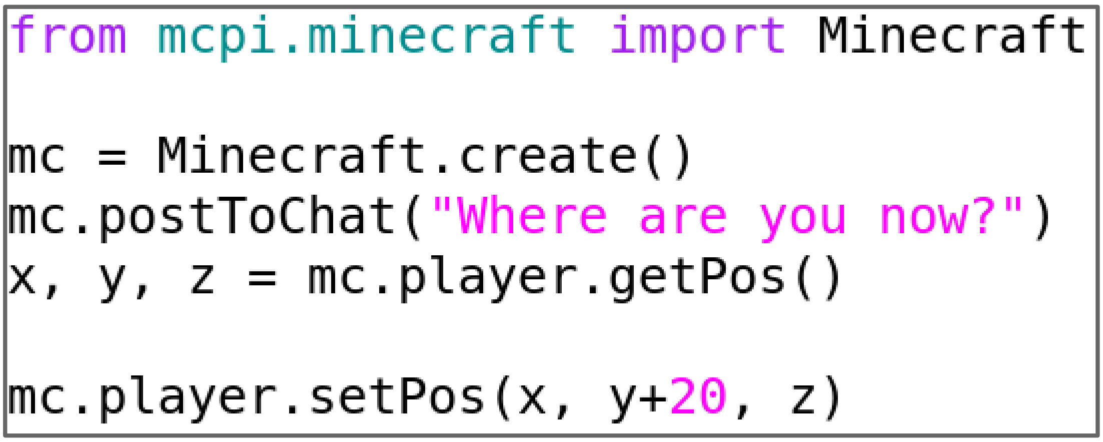
What happened to your Minecraft character?

Remember, all Minecraft-python programs are starting with these first two lines:

```
from mcpi.minecraft import Minecraft
mc = Minecraft.create()
```

Box: Programming	Topic: Minecraft
4c2.3	Hello Python - Where am I now?

-) Start Minecraft, start a game and press Tab
-) Start the Python3-Shell
-) Enter the following text in the Python3 Shell:



```
from mcpi.minecraft import Minecraft

mc = Minecraft.create()
mc.postToChat("Where are you now?")
x, y, z = mc.player.getPos()

mc.player.setPos(x, y+20, z)
```

What happened to your Minecraft character?

NOTE: In Minecraft *x* and *z* represent walking directions (forward/backward and left/right) whereas *y* is up/down

Can you move your character a little bit to the left (with python code)?

For figure sources and references please refer to List of Figures.

Box: Programming	Topic: Minecraft
4c3.1	Blocks - creation

-) Start Minecraft, start a game and press Tab
-) Start the Python3-Shell
-) Enter the following text in the Python3 Shell:

```
from mcpi.minecraft import Minecraft
mc = Minecraft.create()

x, y, z = mc.player.getPos()

mc.setBlock(x+1, y, z, 1)
mc.postToChat("what happened?")

x, y, z = mc.player.getPos()
mc.setBlock(x-1, y, z, 2)
mc.postToChat("what happened now?")

x, y, z = mc.player.getPos()
mc.setBlock(x, y, z+1, 3)
mc.postToChat("what is this?")
```

Which blocks where created and where did they spawn?

For figure sources and references please refer to List of Figures.

Box: Programming	Topic: Minecraft
4c3.2	Blocks – and their numbers

Here you see all the possible Block numbers:

AIR	=	0	IRON_BLOCK	=	42
STONE	=	1	STONE_SLAB_DOUBLE	=	43
GRASS	=	2	STONE_SLAB	=	44
DIRT	=	3	BRICK_BLOCK	=	45
COBBLESTONE	=	4	TNT	=	46
WOOD_PLANKS	=	5	BOOKSHELF	=	47
SAPLING	=	6	MOSS_STONE	=	48
BEDROCK	=	7	OBSIDIAN	=	49
WATER_FLOWING	=	8	TORCH	=	50
LAVA_FLOWING	=	10	FIRE	=	51
SAND	=	12	STAIRS_WOOD	=	53
GRAVEL	=	13	CHEST	=	54
GOLD_ORE	=	14	DIAMOND_BLOCK	=	57
IRON_ORE	=	15	CRAFTING_TABLE	=	58
COAL_ORE	=	16	LADDER	=	65
WOOD	=	17	STAIRS_COBBLESTONE	=	67
LEAVES	=	18	REDSTONE_ORE	=	73
GLASS	=	20	SNOW	=	78
LAPIS_LAZULI_BLOCK	=	22	ICE	=	79
SANDSTONE	=	24	SNOW_BLOCK	=	80
BED	=	26	CACTUS	=	81
COBWEB	=	30	CLAY	=	82
GRASS_TALL	=	31	SUGAR_CANE	=	83
WOOL	=	35	FENCE	=	85
FLOWER_YELLOW	=	37	GLOWSTONE_BLOCK	=	89
FLOWER_CYAN	=	38	BEDROCK_INVISIBLE	=	95
MUSHROOM_BROWN	=	39	STONE_BRICK	=	98
MUSHROOM_RED	=	40	GLASS_PANE	=	102
GOLD BLOCK	=	41	MELON	=	103

Look at the next Page for the Task

Box: Programming	Topic: Minecraft
4c3.3	Blocks - as a Picture

Here you see all the possible Blocks:

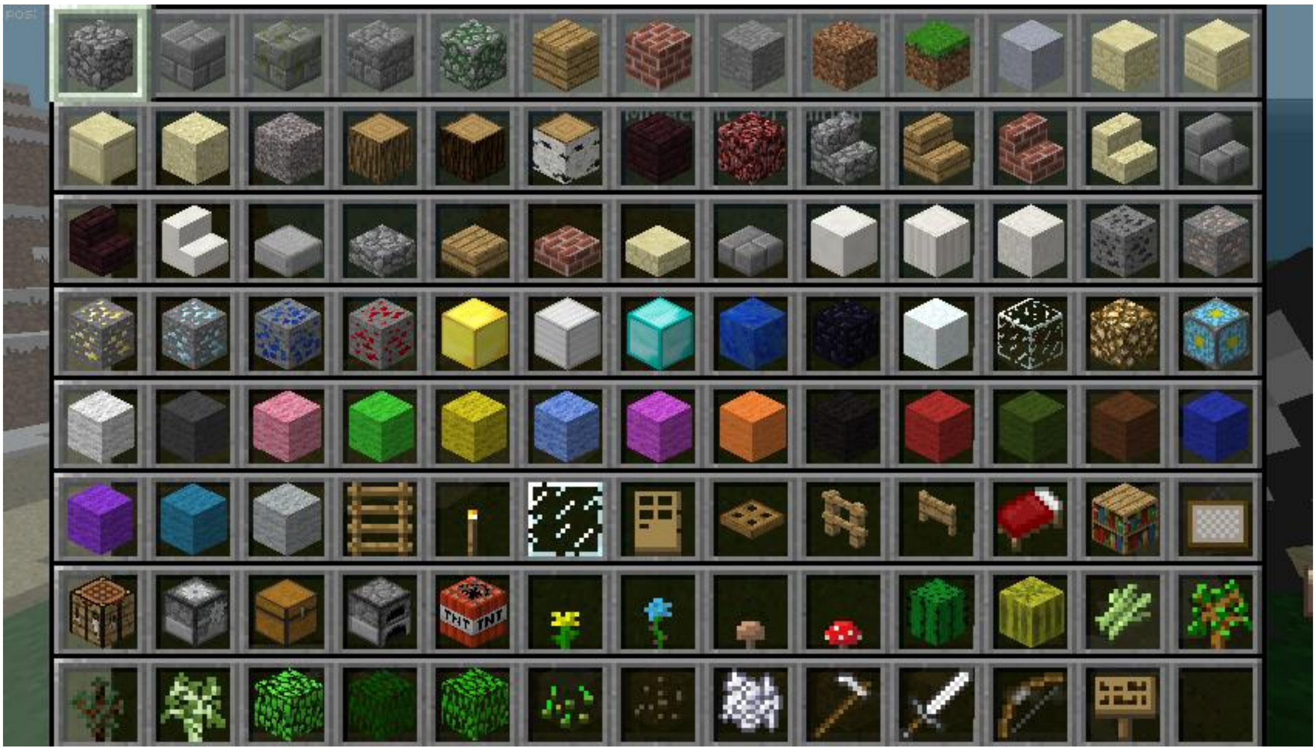


Fig. 55

Remember from the last Tasksheets:
to get the Player position:

```
x,y,z = mc.player.getPos()
```

To create the block with the number 46 in front of the player:

```
mc.setBlock(x+1,y,z,46)
```

Try to create some blocks with python !

Box: Programming	Topic: Minecraft
4c3.4	Blocks – some can be special

Some of the blocks can have special attributes (like the color of wool for example). If you want to create a block with a special attribute you have to use this command:

```
mc.setBlock(x,y,z,<Blocknumber>,<Attribute>)
```

Try out the different attributes with the Wool Block (number 35).

Try to build your country´s flag !

```
mc.setBlock(x+1,y,z,35,1)   1: Orange
mc.setBlock(x+1,y,z,35,2)   2: Magenta
mc.setBlock(x+1,y,z,35,3)   3: Light Blue
mc.setBlock(x+1,y,z,35,4)   4: Yellow
mc.setBlock(x+1,y,z,35,5)   5: Lime
mc.setBlock(x+1,y,z,35,6)   6: Pink
mc.setBlock(x+1,y,z,35,7)   7: Grey
mc.setBlock(x+1,y,z,35,8)   8: Light grey
mc.setBlock(x+1,y,z,35,9)   9: Cyan
mc.setBlock(x+1,y,z,35,10) 10: Purple
mc.setBlock(x+1,y,z,35,11) 11: Blue
mc.setBlock(x+1,y,z,35,12) 12: Brown
mc.setBlock(x+1,y,z,35,13) 13: Green
mc.setBlock(x+1,y,z,35,14) 14: Red
mc.setBlock(x+1,y,z,35,15) 15:Black
```

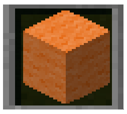


Fig. 56

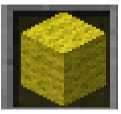


Fig. 57

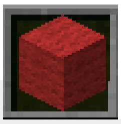


Fig. 58

Post a picture of your flag!

For figure sources and references please refer to List of Figures.

Box: Programming	Topic: Minecraft
4c4.1	Loops and Flowers

-) Start Minecraft, start a game and press Tab
-) Start the Python3-Shell
-) Enter the following text in the Python3 Shell:

```
from mcpi.minecraft import Minecraft
from time import sleep
mc = Minecraft.create()

flower = 38

while True:
 x, y, z = mc.player.getPos()
 mc.setBlock(x, y, z, flower)
 sleep(0.1)
```

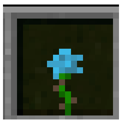


Fig. 59

Press Enter after the last line! Run around in Minecraft and look around (behind you!). What can you see?

Restart the shell
and enter the following, modified code:

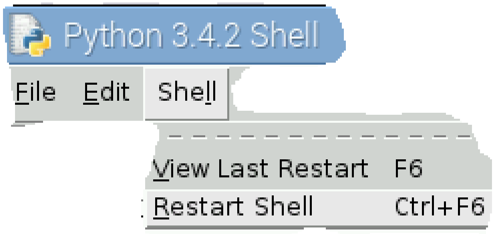


Fig. 60

```
from mcpi.minecraft import Minecraft
from time import sleep
mc = Minecraft.create()
flower = 38
x,y,z = mc.player.getPos()
mc.setBlock(x,y,z,flower)
sleep(0.1)
```

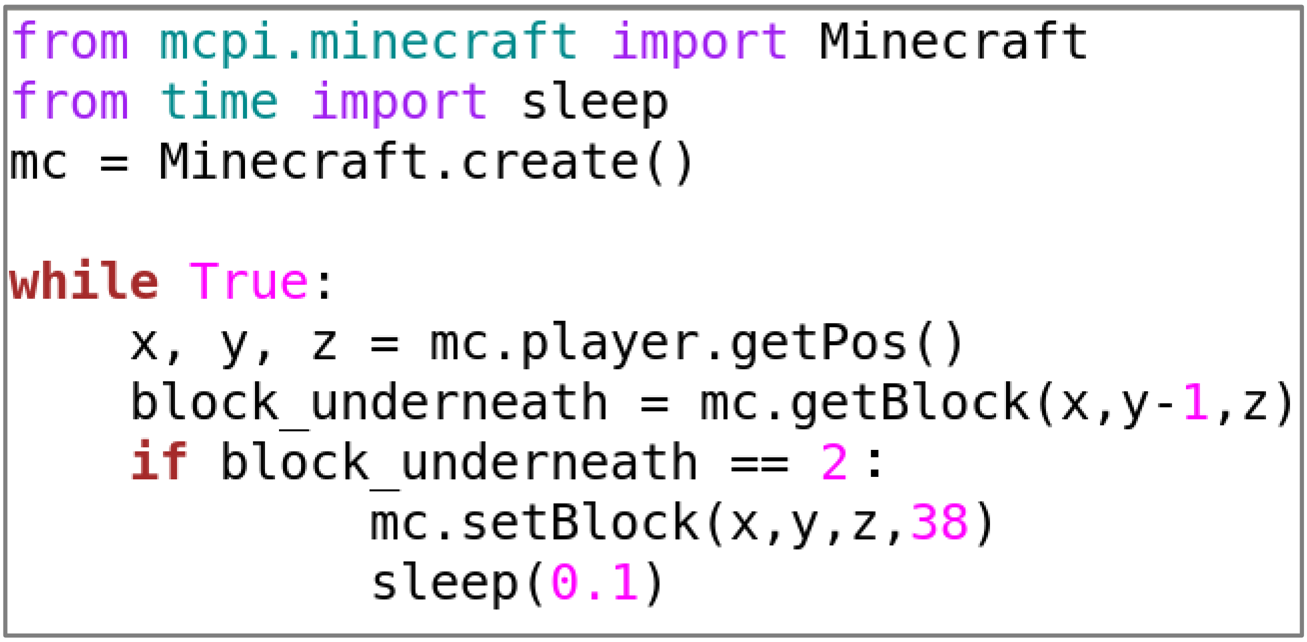
What´s the difference between these two programms?

For figure sources and references please refer to List of Figures.

Box: Programming	Topic: Minecraft
4c4.2	Loops and more Flowers

-) Start Minecraft, start a game and press Tab
-) Start the Python3-Shell
-) Enter the following text in the Python3 Shell:

As flowers are only growing on earth (or dirt or grass in Minecraft), we should modifiy our last programm a litte bit. IF there is a grass-block underneath our player a flower should spawn.



```
from mcpi.minecraft import Minecraft
from time import sleep
mc = Minecraft.create()

while True:
    x, y, z = mc.player.getPos()
    block_underneath = mc.getBlock(x,y-1,z)
    if block_underneath == 2:
            mc.setBlock(x,y,z,38)
            sleep(0.1)
```

NOTE: Because of the WHILE and the IF, you have to press Enter twice after you have entered the text.

Do you know which number the grass-block has? What do you need to change to let flowers spawn on sand or on stone? Can you let grow another flower?

Box: Programming	Topic: Minecraft
4c5.1	Fun with TNT

-) Start Minecraft, start a game and press Tab
-) Start the Python3-Shell
-) Enter the following text in the Python3 Shell:

```
from mcpi.minecraft import Minecraft
mc = Minecraft.create()

x, y, z = mc.player.getPos()
mc.setBlock(x,y,z,46,1)
```

Which block was created? Have you tried to hit it? What happens?

What happens if you try the program below and walk a little bit around in Minecraft?

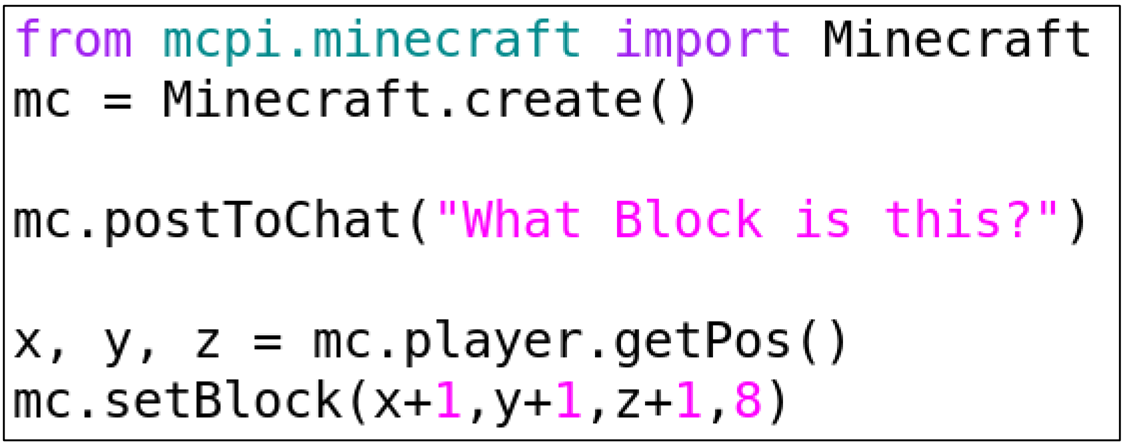
```
from mcpi.minecraft import Minecraft
mc = Minecraft.create()

while True:
    x, y, z = mc.player.getPos()
    block_underneath = mc.getBlock(x,y-1,z)
    if block_underneath == 2 :
            mc.setBlock(x,y,z,46,1)
            sleep(0.1)
```

Try to hit the last block. What happens?

For figure sources and references please refer to List of Figures.

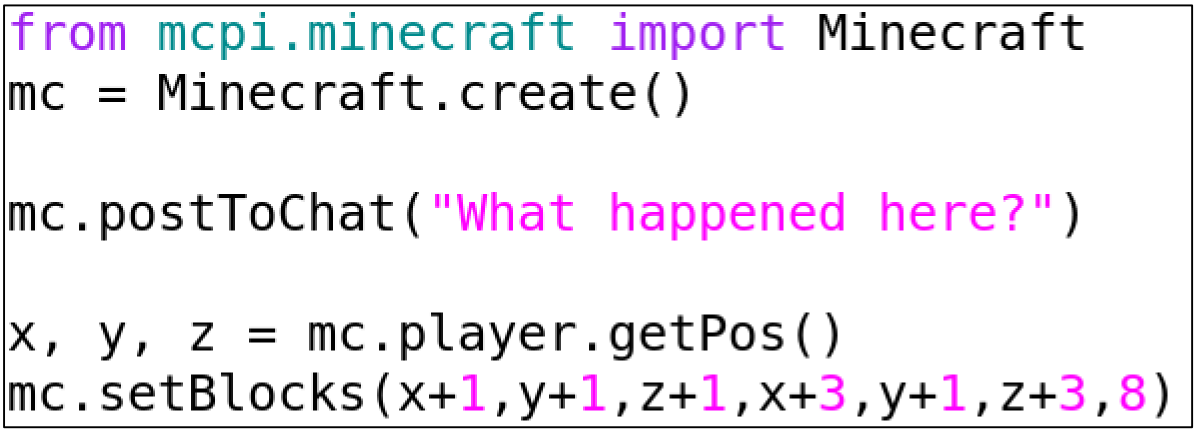
Box: Programming	Topic: Minecraft
4c5.2	Fun with Water

-) Start Minecraft, start a game and press Tab
-) Start the Python3 shell
-) Enter the following text in the Python3 shell:



```
from mcpi.minecraft import Minecraft
mc = Minecraft.create()

mc.postToChat("What Block is this?")

x, y, z = mc.player.getPos()
mc.setBlock(x+1,y+1,z+1,8)
```

Which block was created?



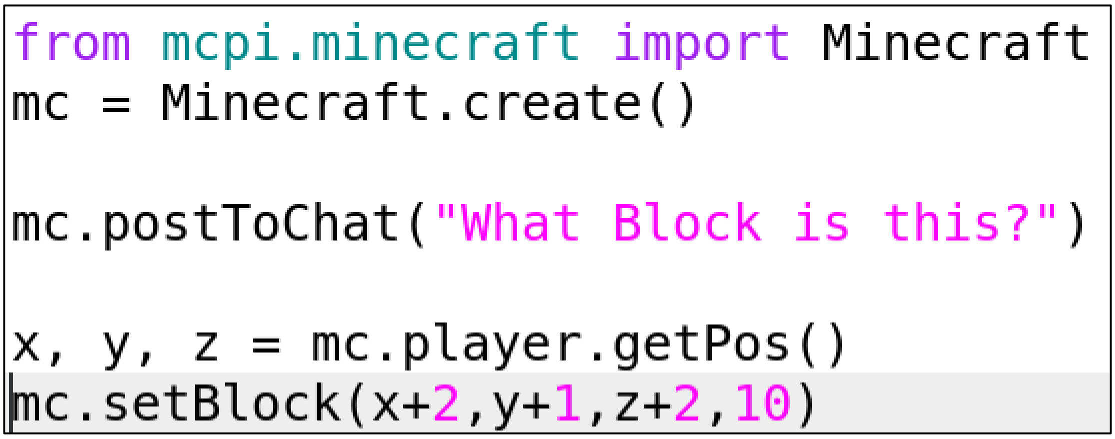
```
from mcpi.minecraft import Minecraft
mc = Minecraft.create()

mc.postToChat("What happened here?")

x, y, z = mc.player.getPos()
mc.setBlocks(x+1,y+1,z+1,x+3,y+1,z+3,8)
```

How many Blocks where created now? Try mc.setBlocks out with different coordinates. What happens?

For figure sources and references please refer to List of Figures.

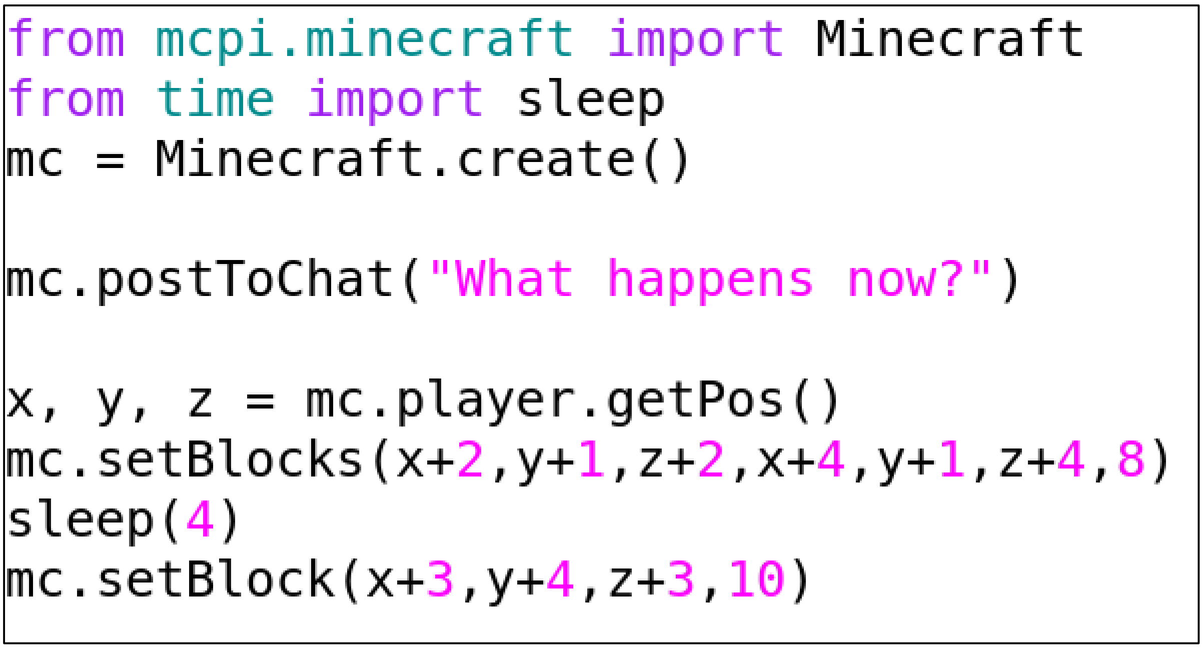
Box: Programming	Topic: Minecraft
4c5.3	Fun with Lava

-) Start Minecraft, start a game and press Tab
-) Start the Python3 shell
-) Enter the following text in the Python3 shell:



```
from mcpi.minecraft import Minecraft
mc = Minecraft.create()

mc.postToChat("What Block is this?")

x, y, z = mc.player.getPos()
mc.setBlock(x+2,y+1,z+2,10)
```

Which block was created?



```
from mcpi.minecraft import Minecraft
from time import sleep
mc = Minecraft.create()

mc.postToChat("What happens now?")

x, y, z = mc.player.getPos()
mc.setBlocks(x+2,y+1,z+2,x+4,y+1,z+4,8)
sleep(4)
mc.setBlock(x+3,y+4,z+3,10)
```

What happend here?

For figure sources and references please refer to List of Figures.

Box: Programming	Topic: Minecraft
4c6.1	My best Minecraft experience

By now, you have learned a lot about how to use Minecraft:

You can build things manually or with Python,
you can use a hundred different blocks,
you can create explosions and other natural disasters,

show us what you do with it!

Fig. 61

Please post a picture or video of your best Minecraft experience!

For figure sources and references please refer to List of Figures.

Box: Programming	**Topic: Lego® Education WeDo 1.0 - Introduction**
Task 4d1.1	**It's hot – let's build a fan !**

1)

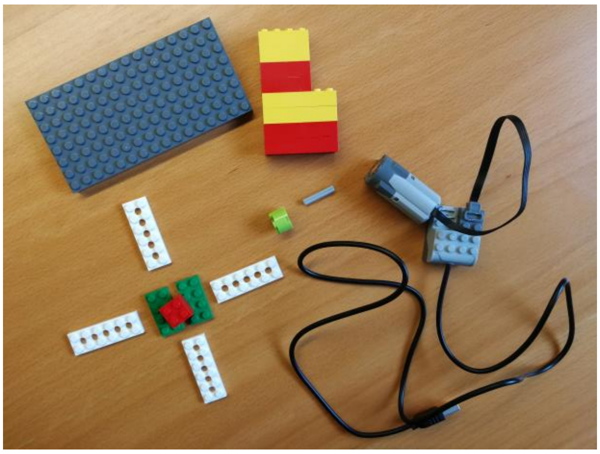


Fig. 70

2)

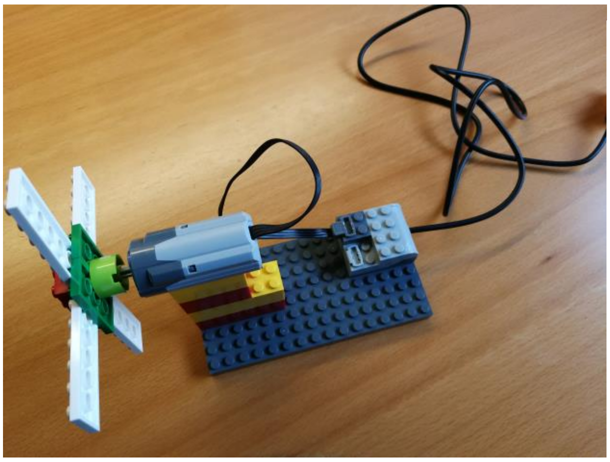
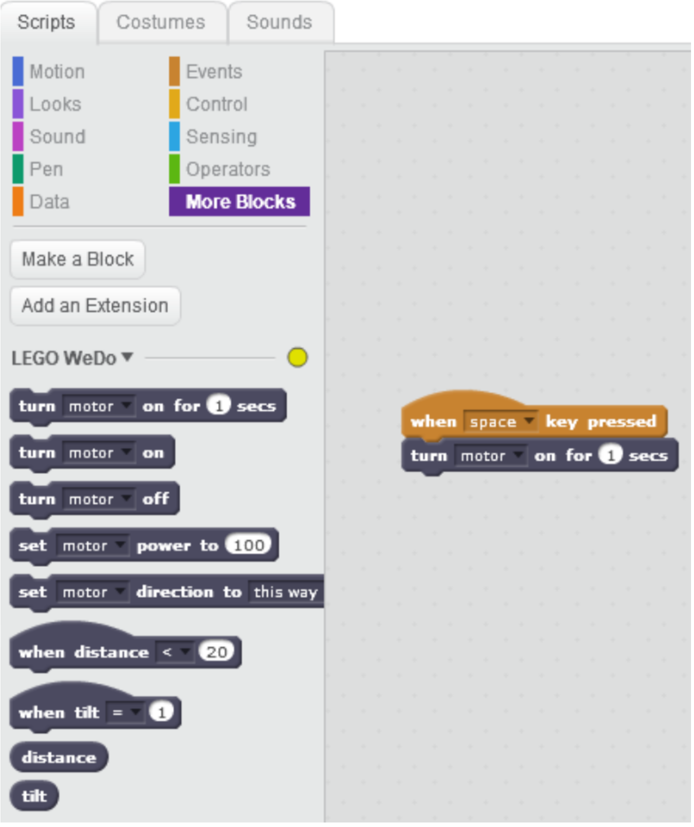


Fig. 71

3) Start Scratch on Raspberry PI (see IT-box)

4) Scratch



```
when space key pressed
turn motor on for 1 secs
```

Fig. 72

Can you make the fan blades more efficient ?
(You can also use other material than Lego® ☺)

For figure sources and references please refer to List of Figures.

Box: Programming	Topic: Lego® Education WeDo 1.0 – guided projects
Task 4d2.1	Hungry alligator

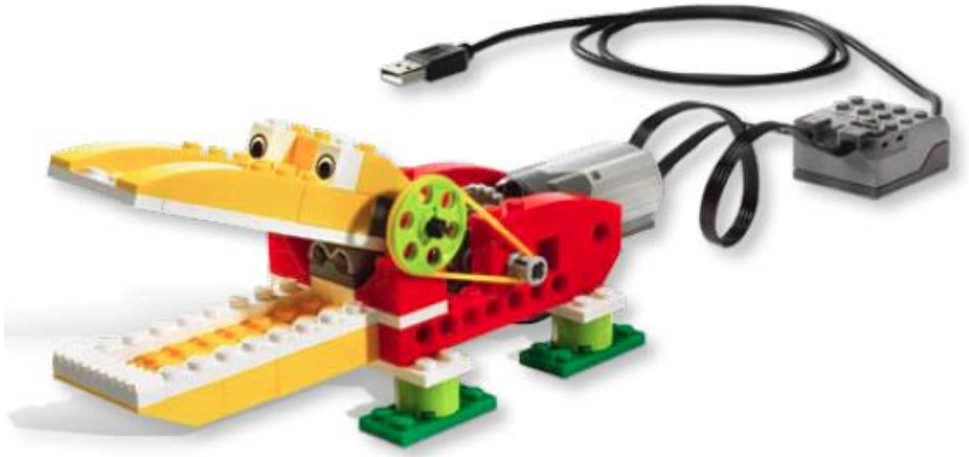


Fig. 73: © All rights at the LEGO® group.

Build an alligator, that can open and close its mouth!

Use the QR-Code in the top right corner or this link https://education.lego.com/en-us/support/wedo/building-instructions to navigate to the construction manual (if you need it) and upload a picture or video of your alligator !

Where do alligators live ?
What do alligators eat?

For figure sources and references please refer to List of Figures.

Box: Programming	Topic: Lego® Education WeDo 1.0 – guided projects
Task 4d2.2	Airplane

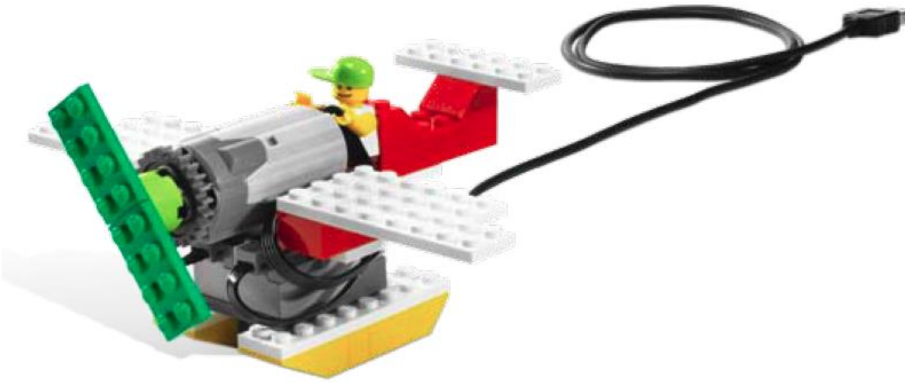


Fig. 74: © All rights at the LEGO® group.

Build an airplane!

Use the QR-Code in the top right corner or this link https://education.lego.com/en-us/support/wedo/building-instructions to navigate to the construction manual (if you need it) and upload a picture or video of your plane!

Can your plane make sounds?

For figure sources and references please refer to List of Figures.

Box: Programming	Topic: Lego® Education WeDo 1.0 – guided projects
Task 4d2.3	Dancing birds

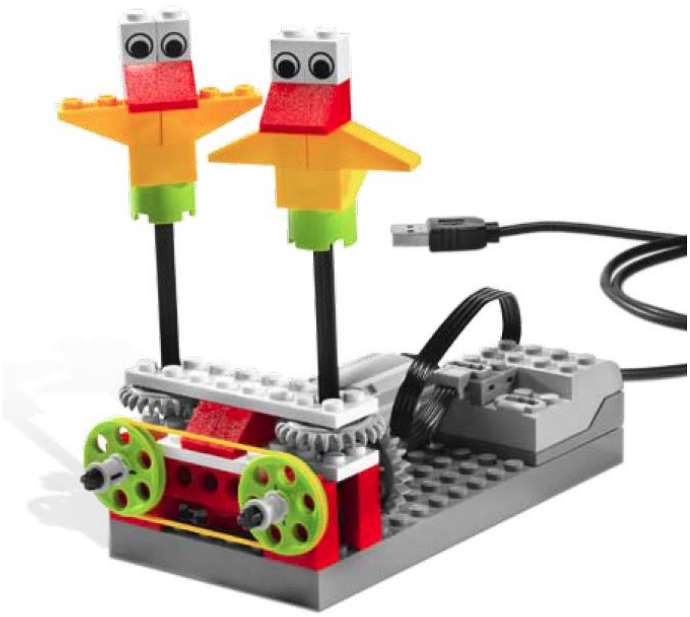


Fig. 75: © All rights at the LEGO® group.

Build one ore more dancing birds!

Use the QR-Code in the top right corner or this link https://education.lego.com/en-us/support/wedo/building-instructions to navigate to the construction manual (if you need it) and upload a picture or video of your birds!

Do your birds turn in the same direction?
How can you change the direction?

Can they make noises as well?
What is their preferred music?

For figure sources and references please refer to List of Figures.

Box: Programming	Topic: Lego® Education WeDo 1.0 – guided projects
Task 4d2.4	Goal-keeper

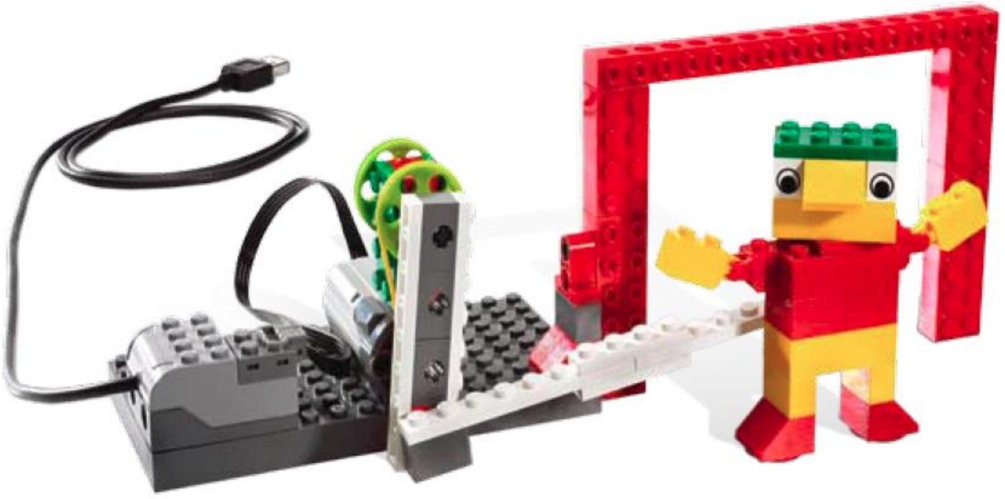


Fig. 76: © All rights at the LEGO® group.

Build a goal-keeper !

Use the QR-Code in the top right corner or this link https://education.lego.com/en-us/support/wedo/building-instructions ner to navigate to the construction manual (if you need it) and upload a picture or video of your goal-keeper !

How many times did it catch the ball?
How can you improve it?

For figure sources and references please refer to List of Figures.

Box: Programming	**Topic: Lego® Education WeDo 1.0 – guided projects**
Task 4d2.5	**Roaring lion**

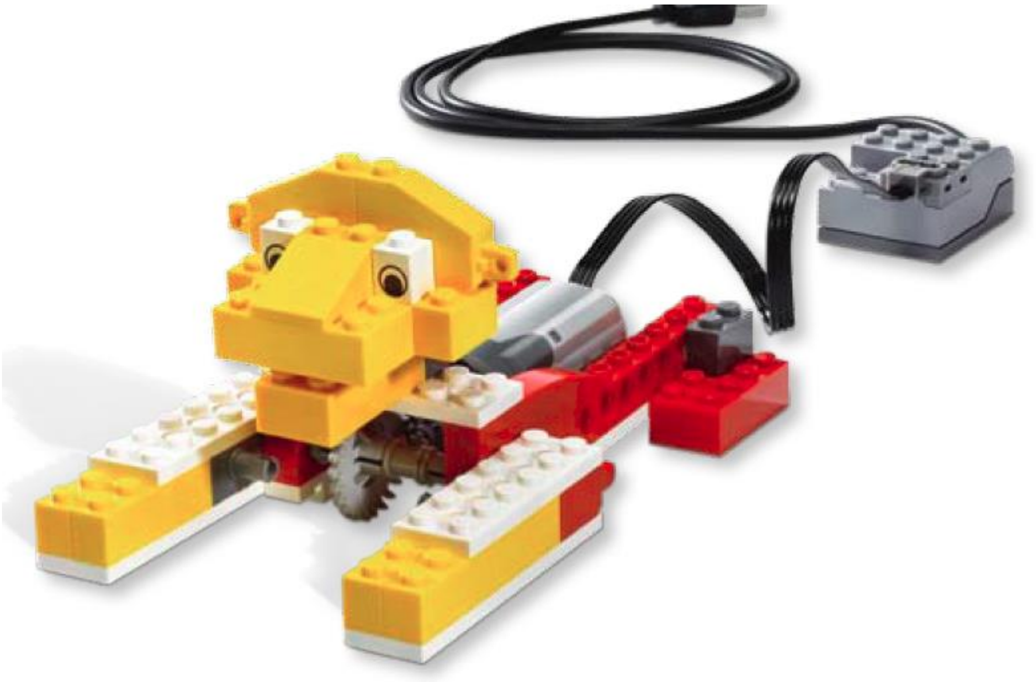


Fig. 77: © All rights at the LEGO® group.

Build a lion, that can stand up and roar!

Use the QR-Code in the top right corner or this link https://education.lego.com/en-us/support/wedo/building-instructions r to navigate to the construction manual (if you need it) and upload a picture or video of your lion!

Where do lions live?
And what do they eat?

For figure sources and references please refer to List of Figures.

Box: Programming	Topic: Lego® Education WeDo 1.0 – guided projects
Task 4d2.6	Drumming monkey

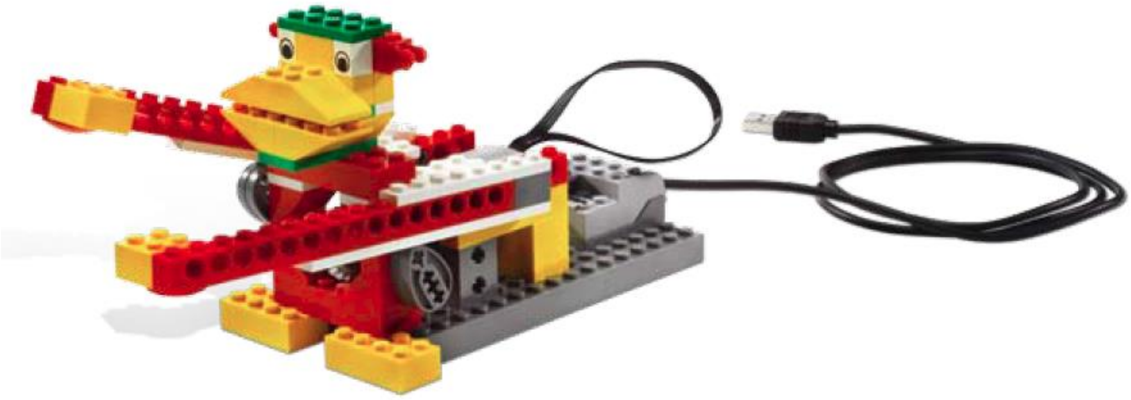


Fig. 78: © All rights at the LEGO® group.

Build a monkey, that play the drums!

Use the QR-Code in the top right corner or this link https://education.lego.com/en-us/support/wedo/building-instructions to navigate to the construction manual (if you need it) and upload a picture or video of your monkey!

Which rhythms can your monkey play?
Can you change them?

For figure sources and references please refer to List of Figures.

Box: Programming	Topic: Lego® Education WeDo 1.0 – guided projects
Task 4d2.7	Smart spinner

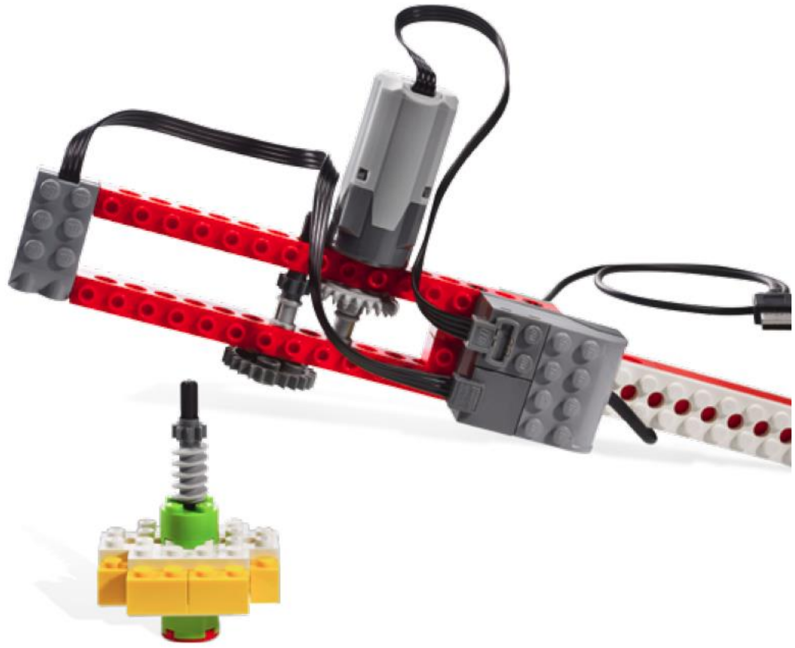


Fig. 79: © All rights at the LEGO® group.

Build a machine, that starts a spinner!

Use the QR-Code in the top right corner or this link https://education.lego.com/en-us/support/wedo/building-instructions to navigate to the construction manual (if you need it) and upload a picture or video of your spinner!

How long does the spinner turn?
How can you make it turn longer?

For figure sources and references please refer to List of Figures.

Box: Programming	Topic: Lego® Education WeDo 1.0 – guided projects
Task 4d2.8	Cheerful fans

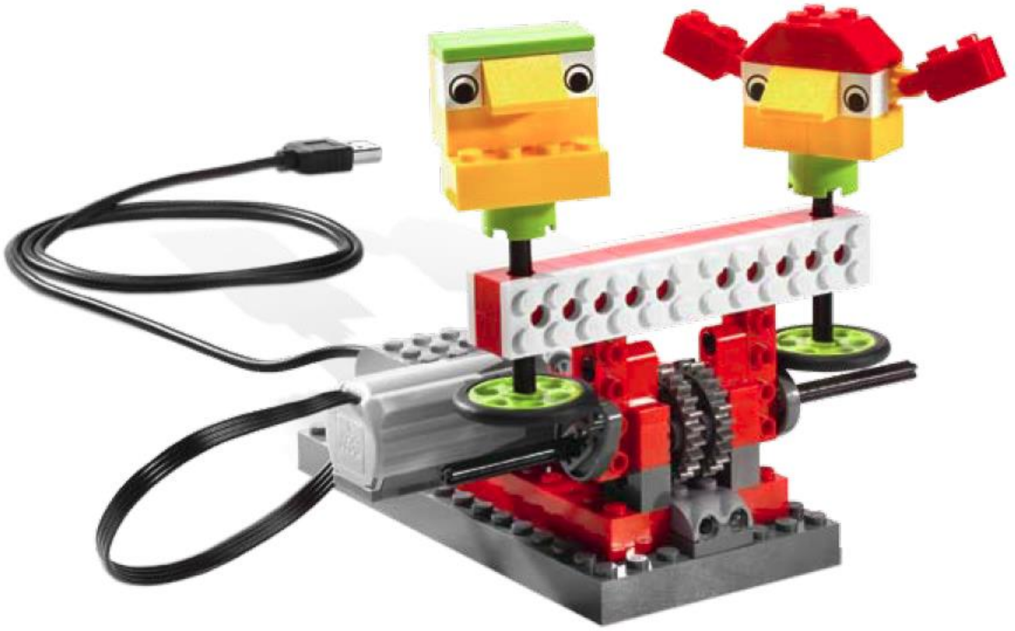


Fig. 80: © All rights at the LEGO® group.

Build cheering fans (maybe next to the goal-keeper of task 4b.4)!

Use the QR-Code in the top right corner or this link https://education.lego.com/en-us/support/wedo/building-instructions to navigate to the construction manual (if you need it) and upload a picture or video of your fans!

Compare your fan-modell with those of other teams. Can they move differently? Cheer differently? Look differently?

For figure sources and references please refer to List of Figures.

Box: Programming	Topic: Lego® Education WeDo 1.0 – guided projects
Task 4d2.9	Flying bird

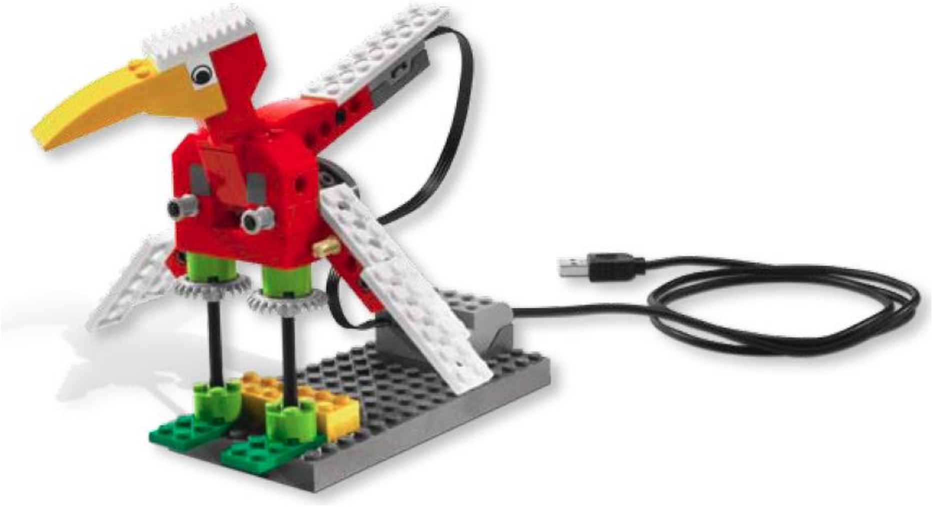


Fig. 81: © All rights at the LEGO® group.

Build a bird, that flaps its wings and cries!

Use the QR-Code in the top right corner or this link https://education.lego.com/en-us/support/wedo/building-instructions to navigate to the construction manual (if you need it) and upload a picture or video of your bird!

What kind of bird did you build?
Where does it live?

For figure sources and references please refer to List of Figures.

Box: Programming	Topic: Lego® Education WeDo 1.0 – guided projects
Task 4d2.10	Rising giant

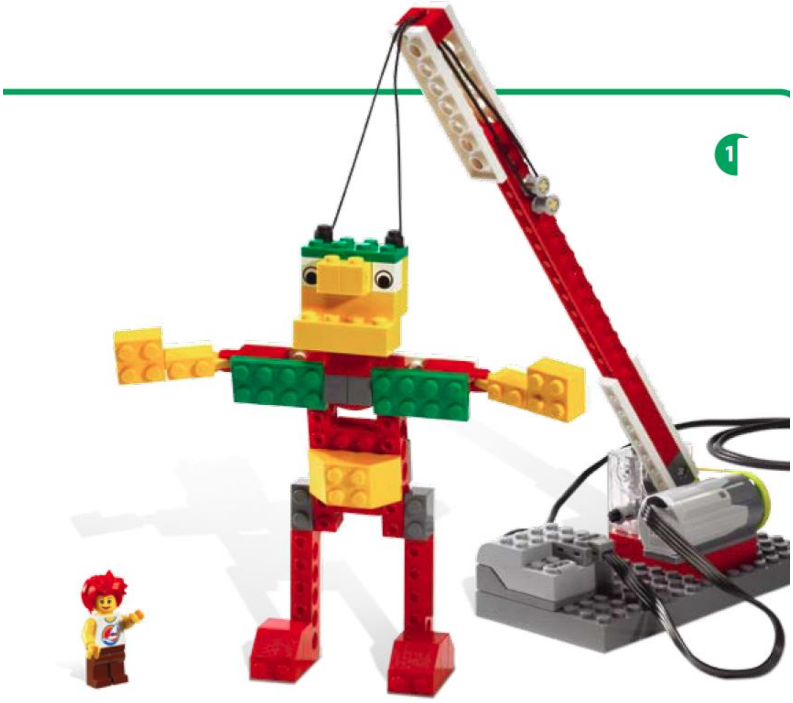


Fig. 82: © All rights at the LEGO® group.

Build giant that it lifted by a crane!

Use the QR-Code in the top right corner or this link https://education.lego.com/en-us/support/wedo/building-instructions to navigate to the construction manual (if you need it) and upload a picture or video of your giant!

Play theatre with the giant and some other Lego-figures. Why does the giant rise? What does it do?

For figure sources and references please refer to List of Figures.

Box: Programming	Topic: Lego® Education WeDo 1.0 – guided projects
Task 4d2.11	Sailboat in the storm

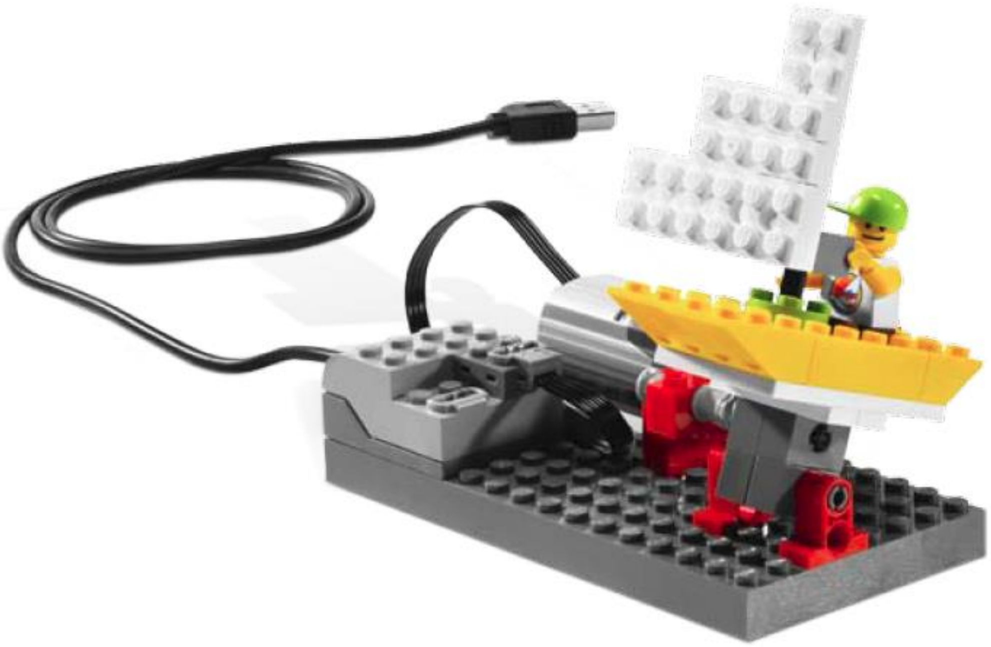


Fig. 83: © All rights at the LEGO® group.

Build a sailboat at stormy weather!

Use the QR-Code in the top right corner or this link https://education.lego.com/en-us/support/wedo/building-instructions to navigate to the construction manual (if you need it) and upload a picture or video of your boat!

Write the captain‘s log: How did the he survive the storm?

For figure sources and references please refer to List of Figures.

Box: Programming	**Topic: Lego® Education WeDo 1.0 – own projects**
Task 4d3.1	**Your own project**

?

Build … your own model!

What is it?
Upload a picture or video of your model!

List of Figures - Teacher Handouts

Figure -Nr.	Copyright holder (if applicable)	Reference / Source	Last accessed
Fig. 1	jayahimsa	https://www.flickr.com/photos/49822796@N00/7167031572	2018/04/25
Fig. 2	Hielscher and Döbeli Honegger CC BY-SA	Hielscher M., Döbeli Honegger B. (2016): *Ozobot Projektideen*. pädagogische Hochschule Schwyz, 2016, p.6.	---
Fig. 3	Raspberry Pi Foundation CC By-SA	https://raw.githubusercontent.com/raspberrypilearning/getting-started-with-minecraft-pi/master/images/minecraft-pi-banner.png	2018/04/25

List of Figures - Tasksheets

Figure -Nr.	**Copyright holder (if applicable)**	**Reference / Source**	**Last accessed**
4a: MakeyMakey			
Fig. 1	Kai Pilger	https://www.pexels.com/photo/buildings-city-city-view-cityscape-597909	2018/04/25
Fig. 2	bruce mars	https://www.pexels.com/photo/yellow-and-white-cable-train-947937/	2018/04/25
Fig. 3	Santeri Viinamäki	https://commons.wikimedia.org/wiki/File:Electric_hand_mixer_20170514.jpg	2018/04/25
Fig. 4	Padaguan, CC BY-SA	https://en.wikipedia.org/wiki/Makey_Makey#/media/File:Makey_Makey.jpg	2018/04/25
Fig. 5	Hielscher and Döbeli Honegger CC BY-SA	Hielscher M., Döbeli Honegger B. (2015): *MaKey MaKey Projektideen*. pädagogische Hochschule Schwyz, 2015, p.1	---
Fig. 6	Stefanie Meissl		---
Fig. 7	Lifelong-Kindergarten-Group, MIT Media-Lab	software - screenshot	---
Fig. 8	Hielscher and Döbeli Honegger CC BY-SA	Hielscher M., Döbeli Honegger B. (2015): *MaKey MaKey Projektideen*. pädagogische Hochschule Schwyz, 2015, p.2	---
Fig. 9	Lifelong-Kindergarten-Group, MIT Media-Lab	software - screenshot	---
Fig. 10	Hielscher and Döbeli Honegger CC BY-SA	Hielscher M., Döbeli Honegger B. (2015): *MaKey MaKey Projektideen*. pädagogische Hochschule Schwyz, 2015, p.3	---
Fig. 11	Hielscher and Döbeli Honegger CC BY-SA	Hielscher M., Döbeli Honegger B. (2015): *MaKey MaKey Projektideen*. pädagogische Hochschule Schwyz, 2015, p.3	---
Fig. 12	Hielscher and Döbeli Honegger CC BY-SA	Hielscher M., Döbeli Honegger B. (2015): *MaKey MaKey Projektideen*. pädagogische Hochschule Schwyz, 2015, p.3	---

Fig. 13	Hielscher and Döbeli Honegger CC BY-SA	Hielscher M., Döbeli Honegger B. (2015): *MaKey MaKey Projektideen*. pädagogische Hochschule Schwyz, 2015, p.3	---
Fig. 14	Hielscher and Döbeli Honegger CC BY-SA	Hielscher M., Döbeli Honegger B. (2015): *MaKey MaKey Projektideen*. pädagogische Hochschule Schwyz, 2015, p.4	---
Fig. 15	Hielscher and Döbeli Honegger CC BY-SA	Hielscher M., Döbeli Honegger B. (2015): *MaKey MaKey Projektideen*. pädagogische Hochschule Schwyz, 2015, p.4	---
Fig. 16	Lifelong-Kindergarten-Group, MIT Media-Lab	software - screenshot	---
Fig. 17	Hielscher and Döbeli Honegger CC BY-SA	Hielscher M., Döbeli Honegger B. (2015): *MaKey MaKey Projektideen*. pädagogische Hochschule Schwyz, 2015, p.5	---
Fig. 18	Hielscher and Döbeli Honegger CC BY-SA	Hielscher M., Döbeli Honegger B. (2015): *MaKey MaKey Projektideen*. pädagogische Hochschule Schwyz, 2015, p.5	---
Fig. 19	Lifelong-Kindergarten-Group, MIT Media-Lab	software - screenshot	---
Fig. 20	Hielscher and Döbeli Honegger CC BY-SA	Hielscher M., Döbeli Honegger B. (2015): *MaKey MaKey Projektideen*. pädagogische Hochschule Schwyz, 2015, p.6	---
Fig. 21	Lifelong-Kindergarten-Group, MIT Media-Lab	software - screenshot	---
Fig. 22	Hielscher and Döbeli Honegger CC BY-SA	Hielscher M., Döbeli Honegger B. (2015): *MaKey MaKey Projektideen*. pädagogische Hochschule Schwyz, 2015, p.8	---
Fig. 23	Lifelong-Kindergarten-Group, MIT Media-Lab	software - screenshot	---
Fig. 24	Hielscher and Döbeli Honegger CC BY-SA	Hielscher M., Döbeli Honegger B. (2015): *MaKey MaKey Projektideen*. pädagogische Hochschule Schwyz, 2015, p.10	---
Fig. 25	Hielscher and Döbeli Honegger CC BY-SA	Hielscher M., Döbeli Honegger B. (2015): *MaKey MaKey Projektideen*. pädagogische Hochschule Schwyz, 2015, p.10	---
4b: Getting to know your Ozobot!			
Fig. 30	Head in the Clouds - Consortium	own picture	---
Fig. 31	Head in the Clouds - Consortium	own picture	---
Fig. 32	Head in the Clouds - Consortium	own picture	---
Fig. 33	Head in the Clouds - Consortium	own picture	---

Fig. 34	Hielscher and Döbeli Honegger CC BY-SA	Hielscher M., Döbeli Honegger B. (2016): *Ozobot Projektideen*. pädagogische Hochschule Schwyz, 2016, p.1.	---
Fig. 35	BenSpark	https://www.flickr.com/photos/abennett96/16361576469	2018/04/25
Fig. 36	Learn 2 Teach, Teach 2 Learn	https://www.flickr.com/photos/28629285@N02/22027121115	2018/04/25
Fig. 37	Hielscher and Döbeli Honegger CC BY-SA	Hielscher M., Döbeli Honegger B. (2016): *Ozobot Projektideen*. pädagogische Hochschule Schwyz, 2016, p.5.	---
Fig. 38	Head in the Clouds - Consortium	own illustration	---
Fig. 39	Hielscher and Döbeli Honegger CC BY-SA	Hielscher M., Döbeli Honegger B. (2016): *Ozobot Projektideen*. pädagogische Hochschule Schwyz, 2016, p.3.	---
Fig. 40	Hielscher and Döbeli Honegger CC BY-SA	Hielscher M., Döbeli Honegger B. (2016): *Ozobot Projektideen*. pädagogische Hochschule Schwyz, 2016, p.4.	---
Fig. 41	Hielscher and Döbeli Honegger CC BY-SA	Hielscher M., Döbeli Honegger B. (2016): *Ozobot Projektideen*. pädagogische Hochschule Schwyz, 2016, p.8.	---
Fig. 42	Head in the Clouds - Consortium	own illustration	---
Fig. 43	jeff medaugh	https://commons.wikimedia.org/wiki/File:Grandjete.jpg	---
Fig. 44	Head in the Clouds - Consortium	own illustration	---
4c: Minecraft			
Fig. 50	Raspberry Pi Foundation CC By-SA	https://raw.githubusercontent.com/raspberrypilearning/getting-started-with-minecraft-pi/master/images/minecraft-pi-banner.png	2018/04/25
Fig. 51	Head in the Clouds - Consortium	software – cropped screenshot of Raspberry PI Menu	---
Fig. 52	Sarfraz Shoukat – CC BY	Icons made by https://www.flaticon.com/authors/sarfraz-shoukat from https://www.flaticon.com	2018/04/25
Fig. 53	Raspberry Pi Foundation CC By-SA	https://raw.githubusercontent.com/raspberrypilearning/getting-started-with-minecraft-pi/master/images/mcpi-inventory.png	2018/04/25
Fig. 54	Head in the Clouds - Consortium	Own Picture	---
Fig. 55	Raspberry Pi Foundation CC By-SA	https://github.com/raspberrypilearning/getting-started-with-minecraft-pi/blob/master/images/mcpi-inventory.png	2018/04/25
Fig. 56	Raspberry Pi Foundation CC By-SA	Cropped Image – Original: https://github.com/raspberrypilearning/getting-started-with-minecraft-pi/blob/master/images/mcpi-inventory.png	2018/04/25
Fig. 57	Raspberry Pi Foundation CC By-SA	Cropped Image – Original: https://github.com/raspberrypilearning/getting-started-with-minecraft-pi/blob/master/images/mcpi-inventory.png	2018/04/25
Fig. 58	Raspberry Pi Foundation CC By-SA	Cropped Image – Original: https://github.com/raspberrypilearning/getting-started-with-minecraft-pi/blob/master/images/mcpi-inventory.png	2018/04/25

Fig. 59	Raspberry Pi Foundation CC By-SA	Cropped Image – Original: https://github.com/raspberrypilearning/getting-started-with-minecraft-pi/blob/master/images/mcpi-inventory.png	2018/04/25
Fig. 60	Head in the Clouds - Consortium	software – cropped screenshot of Python Shell	----
Fig. 61	CC0	https://5dwallpaper.com/pt/minecraft-wallpaper-hd-2/minecraft-hd-backgrounds-wallpaper-wp3808551/	2018/04/25
4d: Lego® Education WeDo 1.0			
Fig. 70	Head in the Clouds - Consortium	own picture	---
Fig. 71	Head in the Clouds - Consortium	own picture	---
Fig. 72	Lifelong-Kindergarten-Group, MIT Media-Lab	software - screenshot	---
Fig. 73	The LEGO® group	https://education.lego.com/en-us/support/wedo/building-instructions	2018/04/25
Fig. 74	The LEGO® group	https://education.lego.com/en-us/support/wedo/building-instructions	2018/04/25
Fig. 75	The LEGO® group	https://education.lego.com/en-us/support/wedo/building-instructions	2018/04/25
Fig. 76	The LEGO® group	https://education.lego.com/en-us/support/wedo/building-instructions	2018/04/25
Fig. 77	The LEGO® group	https://education.lego.com/en-us/support/wedo/building-instructions	2018/04/25
Fig. 78	The LEGO® group	https://education.lego.com/en-us/support/wedo/building-instructions	2018/04/25
Fig. 79	The LEGO® group	https://education.lego.com/en-us/support/wedo/building-instructions	2018/04/25
Fig. 80	The LEGO® group	https://education.lego.com/en-us/support/wedo/building-instructions	2018/04/25
Fig. 81	The LEGO® group	https://education.lego.com/en-us/support/wedo/building-instructions	2018/04/25
Fig. 82	The LEGO® group	https://education.lego.com/en-us/support/wedo/building-instructions	2018/04/25
Fig. 83	The LEGO® group	https://education.lego.com/en-us/support/wedo/building-instructions	2018/04/25