2021

PROJECT GIRLSINSTEM

DIGITAL ART TOOLKIT

STEM-ACTIVITIES FOR YOUTH

Version 1.4, October 2023

This instructable was developed by iDrops & De Creatieve STEM for Girls in STEM within the European ERASMUS+ Programme no. 2019-3-BE05-KA205-002788.

Co-funded by the Erasmus+ Programme of the European Union

Lead Author: Valerio

Co-author and review: Marieke Deckers

Graphic Design: Jules De Bruyckere & Marieke Deckers

Review: Marieke Deckers

Photos: Brecht Van den Begin

DIGITAL ART TOOLKIT

TABLE OF CONTENT

GIRLSINSTEM

The EU project

The aim of this project, funded by the European Commission through the Erasmus+ programme, is to empower girls to pursue their interests in Science, Technology, Engineering and Maths (STEM) subjects.

The project will provide young girls and women with support and opportunities to experience the potential of studying STEM subjects and embarking on a fulfilling career in these disciplines.

The project has 4 main actions.

The project team will develop three toolkits for educators, focusing on three subjects; reverse engineering, e-textiles and digital art. These toolkits will provide all the information and resources for educators to be able to facilitate activites with young people on these subjects, in their own environments.

These toolkits will form the basis of three bootcamps which will take place during 2021. These are international bootcamps bringing together young girls and women from across Europe. Each bootcamp will focus on one subject and provide the opportunity for participants to engage with the activities being developed.

The project will also lead a mentoring programme for young females to deepen their knowledge for one specific STEM area. The programme will be implemented in partnership with local NGOs, enterprises and institutional environments, where volunteers from such organisations can accompany the youth to discover more about the areas they work in.

Find out more about the project through the website at:

www.girlsinstem.eu

The partners

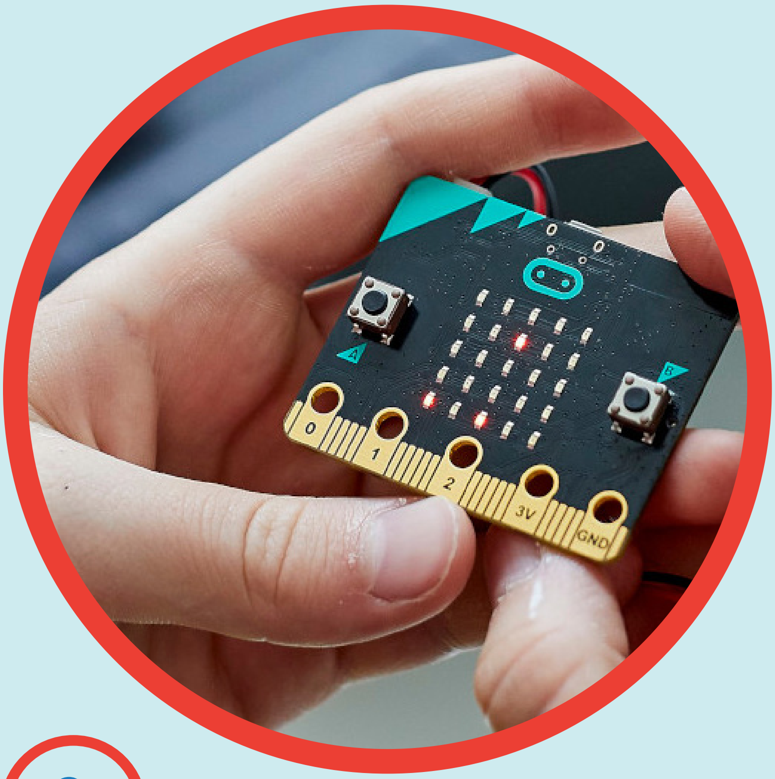
De Creatieve STEM is a non-profit organization dedicated to providing inclusive STEM education and learning activities for future innovators. They offer after-school programs, experiences at FabLabs (digital fabrication workshops).

Dijijeunes is an NGO that promotes literacy, inclusion and the development of digital skills, thus helping to create benefits for society.

iDrops organizes and guides innovation processes related to six social challenges: Care, New Knowledge, Sustainability, Superdiversity, Global Innovation and Community Development.

Fundacja Ad Meritum is a non-profit organization that supports the understanding and development of skills of young people and adults in Europe.

INTRAS Foundation is a non-profit organization dedicated to research and intervention in mental health, with the aim of accompanying people in their recovery process.


A
B
0
1
2
3V
GND

THE MICROBIT

A STARTER GUIDE

4

THE MICRO:BIT

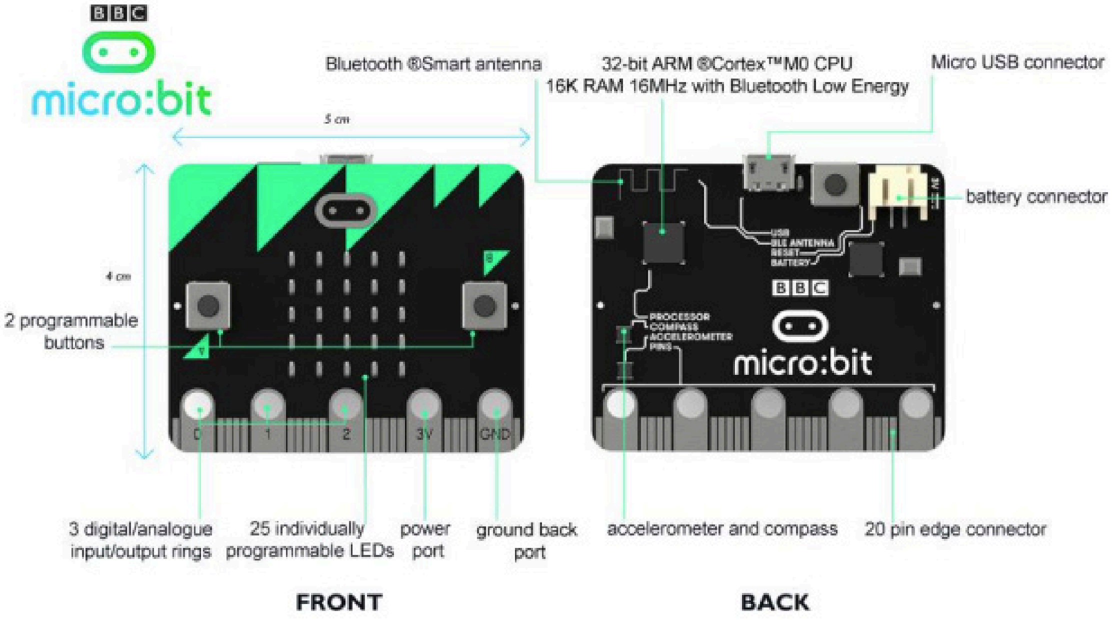
The basics

The micro:bit is a powerful teaching and learning tool that helps younger children to start learning coding and programming, acting as a springboard for further learning and more advanced products such as Raspberry Pi, but is also a great resource for using in lessons across the curriculum including science, maths, D&T, art, music etc.

The BBC micro:bit is a pocket-sized computer that can be coded, customised and controlled to bring digital ideas, games and apps to life. Measuring 4cm by 5cm, and designed to be fun and easy to use, the micro:bit can create anything from games and animations to scrolling stories at school, at home and on the go - all that is needed is imagination and creativity!

It's incredibly versatile with several different coding platforms; Bluetooth connectivity and hard connections that enable a wide range of accessories to be used with it; a 25 LED display screen; and, an on-board accelerometer and compass that allows it to respond to movement. Because the device is suitable for use by across the curriculum, across the age spectrum and for the hobbyist community, the micro:bit is available to purchase from various educational suppliers The platform is cheap, small, and reasonable enough to do some very interesting projects. The challenge will be whether the teachers have been given enough support. Most will be unprepared to exploit or support work with an embedded system.

I haven't been able to drill down into this aspect, but I hope that there's already a critical mass of support material, training, etc. The micro:bit Create Code webpage lists two graphical (CK Javascript and the Microsoft Block Editor) and two text-based IDEs (Microsoft Touch Develop and Python). Unfortunately, many who want to push the technical aspects of a platform like micro:bit often overlook the importance and depth of support required to make such a platform successful in the long run. It does have major corporate support. However, that's no guarantee the efforts will prove fruitful.



MICRO:BIT HAS THE FOLLOWING PHYSICAL FEATURES:

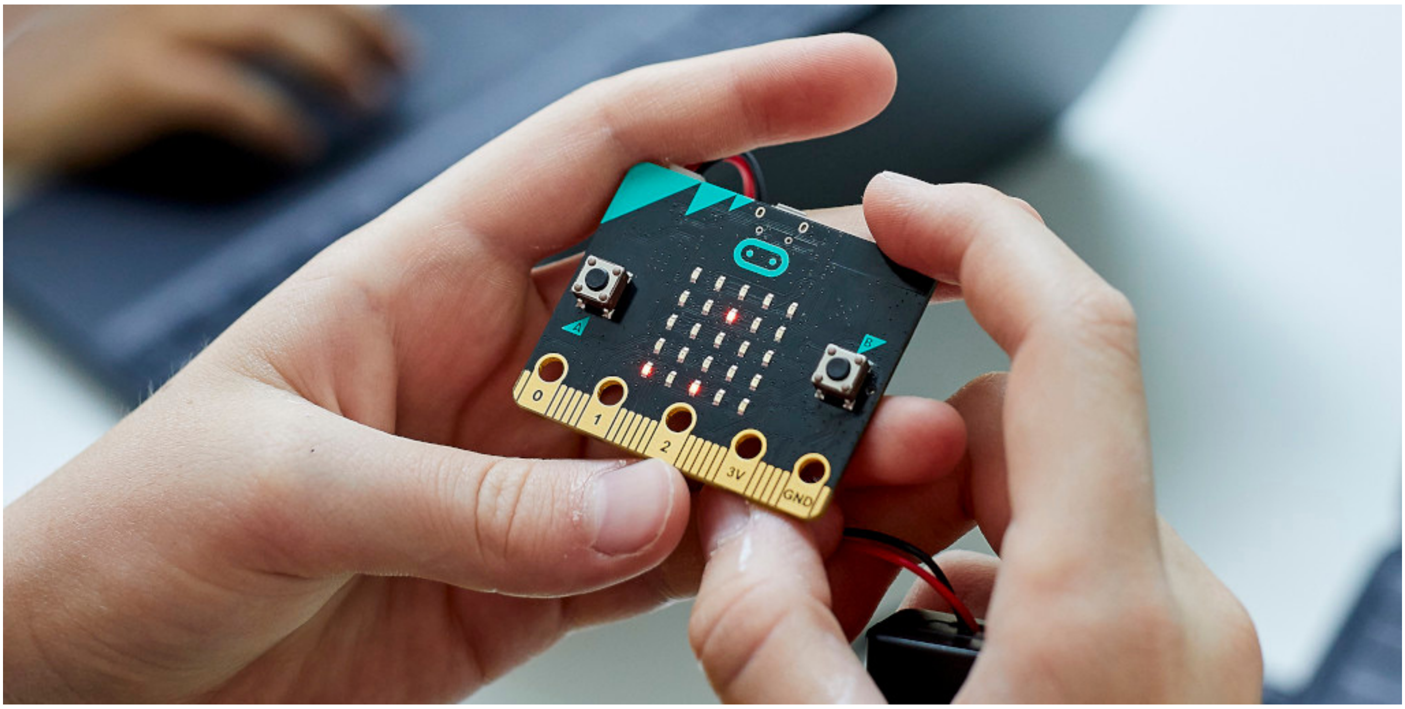
- 25 individually-programmable LEDs.
- 2 programmable buttons.
- Physical connection pins.
- Light and temperature sensors.
- Motion sensors (accelerometer and compass)
- Wireless Communication, via Radio and Bluetooth.
- USB interface.

THE MICRO:BIT ECOSYSTEM

Several suppliers are offering project kits, accessories and sensors for scienceexperiments. The Micro:bit Foundation and its partners is providing a comprehensive range of curriculum-linked teaching resources to support the use of micro:bit in the classroom including Schemes of Work, lesson activities, project ideas, tutorials, quick start guides and teacher training.

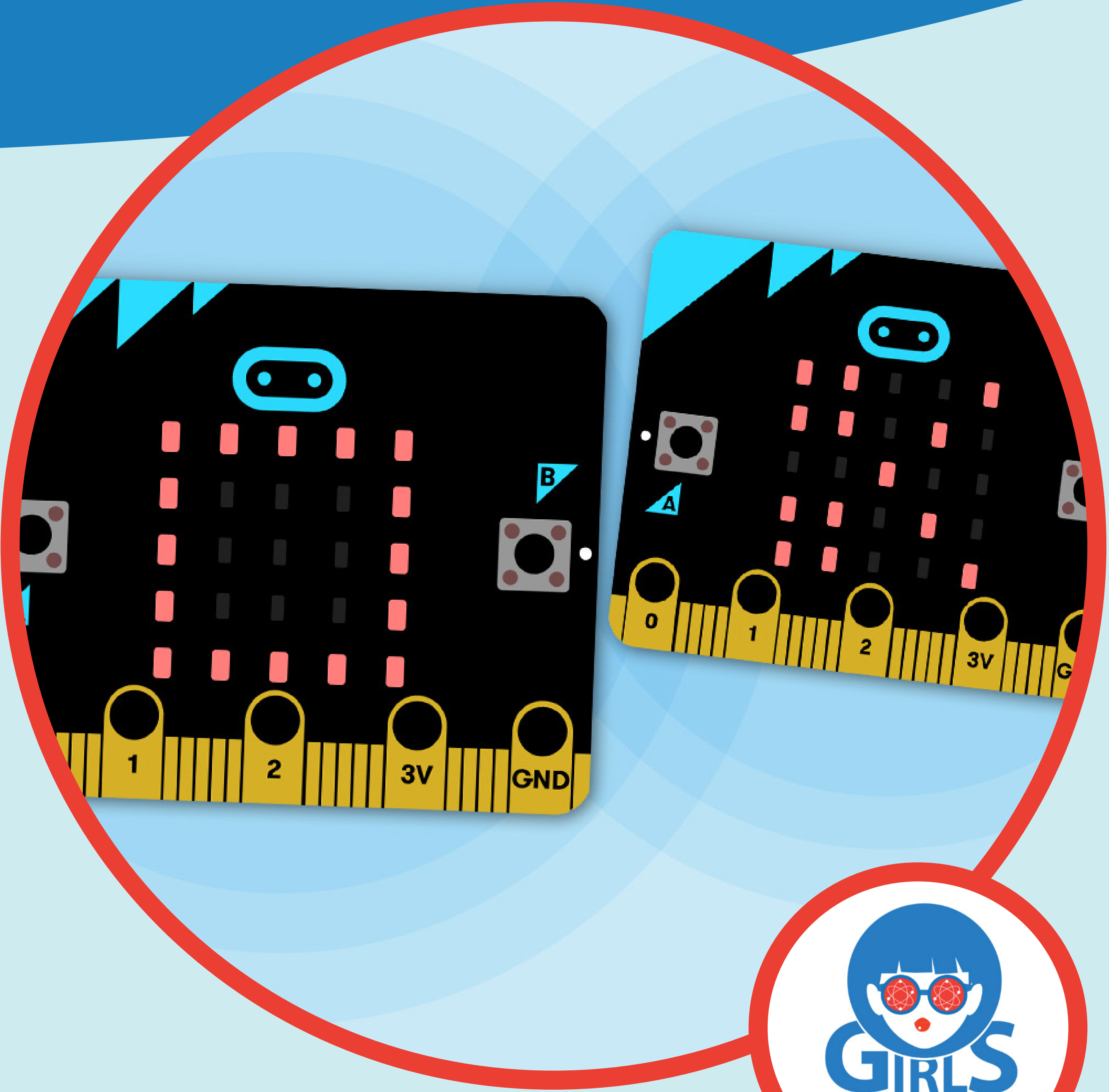
With the UK facing a significant skills shortage, and an extra 1.4m digital professionals needed over the next five years, in 2015 the BBC started 'Make it Digital' - a national campaign to inspire a new generation of coders and programmers.

The digital world is evolving all the time - and with it, the demand for more young people with coding and digital skills. It's great to be supporting BBC Make it Digital to promote the world of digital creativity and inspire the next generation to get involved with what is fast becoming our most exciting and creative industries.



ROCK PAPER SCISSORS

MICROBIT TUTORIAL



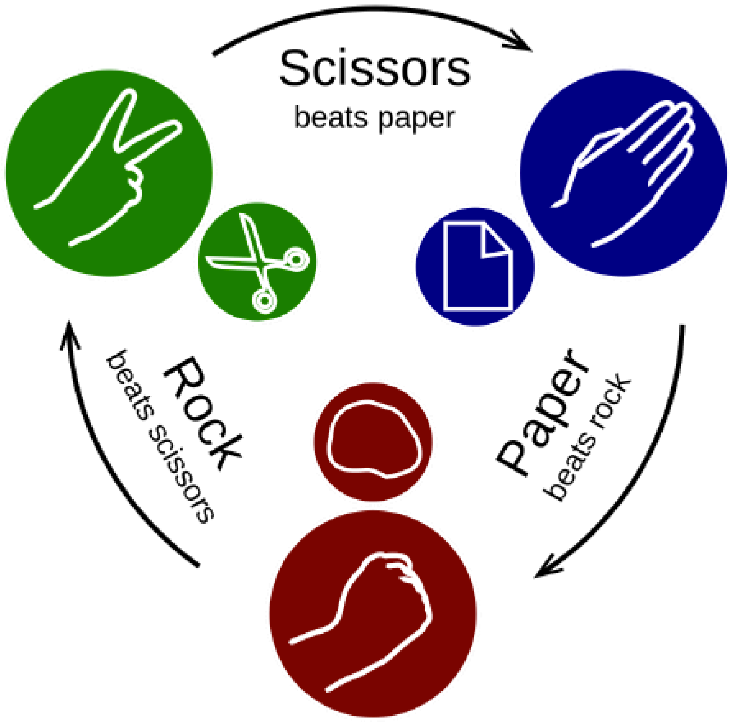
GIRLS in stem

10

ROCK PAPER SCISSORS

Microbit Game

Classic game with two micro:bits and learn about selection, variables and random numbers at the same time.



MATERIALS

- 2 micro:bits (or MakeCode simulator)
- MakeCode or Python editor
- battery packs (optional)
- a partner to play with

HOW IT WORKS

Rock, paper, scissors is a classic game of chance for two people. You and a partner shake your fists 3 times and then make gestures at random to show a rock, paper or scissors. Rock beats scissors, scissors beat paper and paper beats rock (it wraps the rock!).

When the micro:bit accelerometer detects a shake movement, it sets the variable tool to a random number: 0, 1 or 2.

We use 0 because computers start counting at 0, and it's good to remember that 0 is a number!

The program uses selection to decide what image to show on the LED display. If the random number was 0, it shows a rock icon, if it was 1 it shows the icon representing paper. If it wasn't 0 or 1, it must be 2 because we instructed the micro:bit to only pick random numbers between 0 and 2, so in that case it shows scissors.

ACCELEROMETER

An accelerometer is a motion sensor that measures movement. The accelerometer in your BBC micro:bit detects when you tilt it left to right, backwards and forwards and up and down. There are lots of ways you can use the accelerometer in your projects. Find out more about how it works by watching the video, then choose a project to get started.

COMPASS

Your micro:bit has a compass sensor called a magnetometer that measures magnetic fields. It can sense the Earth's magnetic field and so you can use it as a compass.

The Micro:bit

1. Add an on_shake event to run code when you shake the micro:bit. Definethe function on_gesture_shake(). This function will contain all the game's code and execute when you shake the micro:bit.

```
def on_gesture_shake():
        pass
input.on_gesture(Gesture.SHAKE, on_gesture_shake)
```

2. In programming, functions are blocks of statements that perform specific tasks.

Create 2 players and choose a random object (either rock, paper, or scissors) for each player. Each player selects one of the 3 objects (either rock, paper, or scissors). In programming, we prefer to use numbers instead of words and thus we are going to assign the numbers 1, 2 and 3 to the objects rock, paper and scissors respectively. Therefore, you only have to choose a random number in the range of 1 to 3 for each player.

Inside the on_gesture_shake function, create two variables hand1 and hand2 for the first and the second player respectively. Use the randint() function in order to generate 2 random numbers and store them in the variables hand1 and hand2.

```
hand1 = randint(1, 3)
hand2 = randint(1,3)
```

3. Variables are used to store information and data. This data can be numbers, letters, words.

The randint() function is used to generate random integer numbers in a specified range. We use two parameters in order to specify the lower and higher limit. For example, randint(1, 10) will return random integer numbers between 1 to 10.

4. Show players' choice

Use the command:

```
basic.show_number(hand1)
```

to show first player's choice on the LEDs. What do you notice? A number in the range of 1 to 3 appeared and not the actual player's choice (rock, paper, or scissors). This is happening because the computer doesn't know that we symbolize the objects with numbers.

We are going to use the random numbers that are stored in hand1 and hand2 to select a picture to show on the LEDs. If the number is 1, show a picture of a rock. If the number is 2, show a picture of a piece of paper. If the number is not 1 nor 2 (the number is 3), show a picture of scissors.

5. An if statement is a programming conditional statement that performs different actions depending on whether a given condition is evaluated to true or false.

Use the if statement.

```
if hand1 == 1:
        basic.show_icon(IconNames.SMALL_SQUARE)
elif hand1 == 2:
        basic.show_icon(IconNames.SQUARE)
else:
        basic.show_icon(IconNames.SCISSORS)
```

Write the same commands for the second player. Don't forget to change the hand1 variable to hand2.

```
if hand2 == 1:
        basic.show_icon(IconNames.SMALL_SQUARE)
elif hand2 == 2:
        basic.show_icon(IconNames.SQUARE)
else:
        basic.show_icon(IconNames.SCISSORS)
```

6. Determine the winner

Show number 1, if first player wins. Show number 2, if second player winsand show 0, if it is a tie. Use multiple if statements (nested if statements) in order to determine the winner.

```
if hand1 == 1: # Rock
        if hand2 == 1: # Rock
                basic.show_number(0)
        elif hand2 == 2: # Paper
                basic.show_number(2)
        else: # Scissors
                basic.show_number(1)
elif hand1 == 2: # Paper
        if hand2 == 1: # Rock
                basic.show_number(1)
        elif hand2 == 2: # Paper
                basic.show_number(0)
        else: # Scissors
                basic.show_number(2)
else: # Scissors
        if hand2 == 1: # Rock
                basic.show_number(2)
        elif hand2 == 2: # Paper
                basic.show_number(1)
        else: # Scissors
                basic.show_number(0)
```

7. Improve user experience

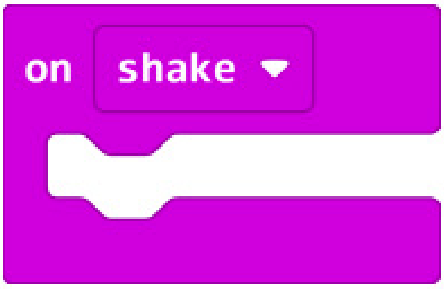
As you may consider, players' icons (choices) appear and disappear quickly. Use the command pause(500) to stop the game for 0.5 seconds (this means that the micro:bit shows the icon for 0.5 seconds). Then use the commands clear_screen() and pause(500) to stop the game for 0.5 seconds before showing the next icon.

```
def on_gesture_shake():
        hand1 = randint(1, 3)
        hand2 = randint(1, 3)
        if hand1 == 1: # Rock
        basic.show_icon(IconNames.SMALL_SQUARE)
    elif hand1 == 2: # Paper
        basic.show_icon(IconNames.SQUARE)
    else: # Scissors
        basic.show_icon(IconNames.SCISSORS)
    basic.pause(500)
    basic.clear_screen()
    basic.pause(500)
    if hand2 == 1: # Rock
        basic.show_icon(IconNames.SMALL_SQUARE)
    elif hand2 == 2: # Paper
        basic.show_icon(IconNames.SQUARE)
    else: # Scissors
        basic.show_icon(IconNames.SCISSORS)
        basic.pause(500)
        basic.clear_screen()
        basic.pause(500)
    if hand1 == 1: # Rock
        if hand2 == 1: # Rock
            basic.show_number(0)
        elif hand2 == 2: # Paper
            basic.show_number(2)
        else: # Scissors
        basic.show_number(1)
    elif hand1 == 2: # Paper
        if hand2 == 1: # Rock
            basic.show_number(1)
        elif hand2 == 2: # Paper
            basic.show_number(0)
        else: # Scissors
            basic.show_number(2)
    else: # Scissors
        if hand2 == 1: # Rock
            basic.show_number(2)
        elif hand2 == 2: # Paper
            basic.show_number(1)
        else: # Scissors
            basic.show_number(0)
input.on_gesture(Gesture.SHAKE, on_gesture_shake)
```

CODING

Step 1

The micro:bit to chose rock, paper, or scissors when you shake it. Place a on shake block so when we shake the micro:bit, it will run part of a program.



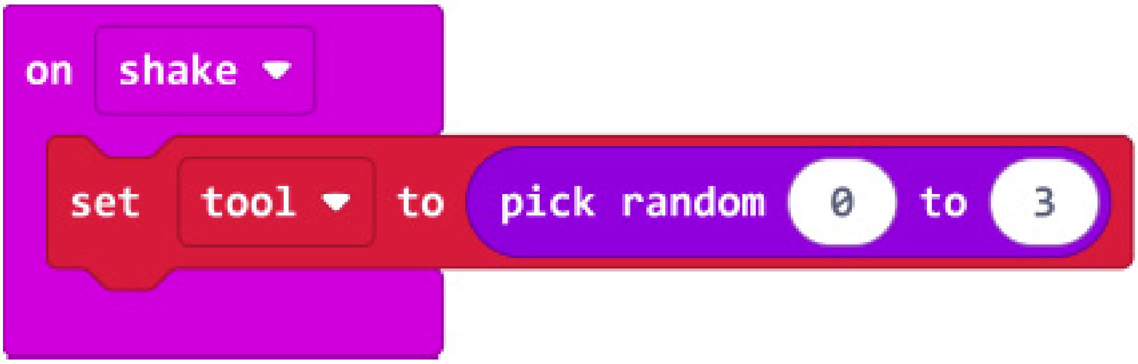
```
on shake ▾
```

```
def on_gesture_shake():

pass

input.on_gesture(Gesture.SHAKE, on_gesture_shake)
```

Step 2

Add a tool variable to store a random number computed with pick random. When we shake the micro:bit, it should pick a random number from 0 to 2 and store it in the variable tool. (This variable is named tool because rock, paper, and scissors are the tools we use to challenge your friends!)

In a later step, each of the possible numbers (0, 1, or 2) is matched to its own picture. The picture is shown on the LEDs when its number is picked.

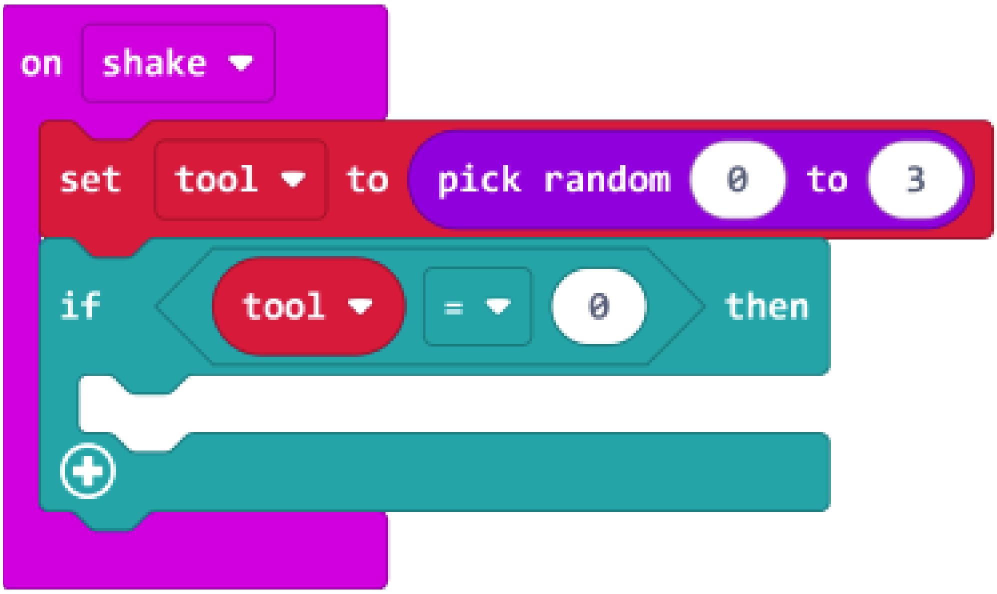


```
on shake ▾
  set tool ▾ to pick random 0 to 3
```

```
tool = 0

def on_gesture_shake():
global tool
tool = randint(0, 3)
input.on_gesture(Gesture.SHAKE, on_gesture_shake)
```

Step 3

Place an if block under the pick random and check whether tool is equal to 0.

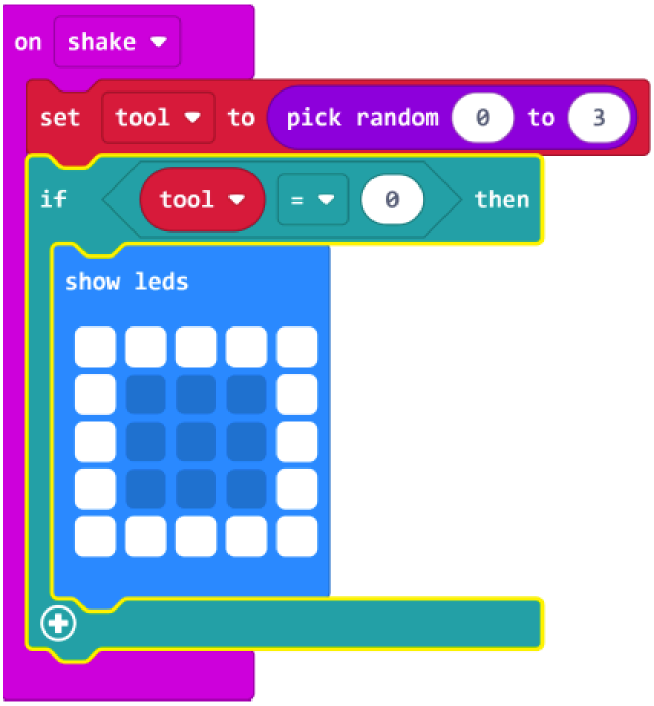


```
on shake ▾
set tool ▾ to pick random 0 to 3
if tool ▾ = ▾ 0 then
```

```
tool = 0

def on_gesture_shake():
tool2 = randint(0, 3)
if tool2 == 0:
pass
input.on_gesture(Gesture.SHAKE, on_gesture_shake)
```

Step 4

In the if block, place a show leds block that shows a picture of a paper.

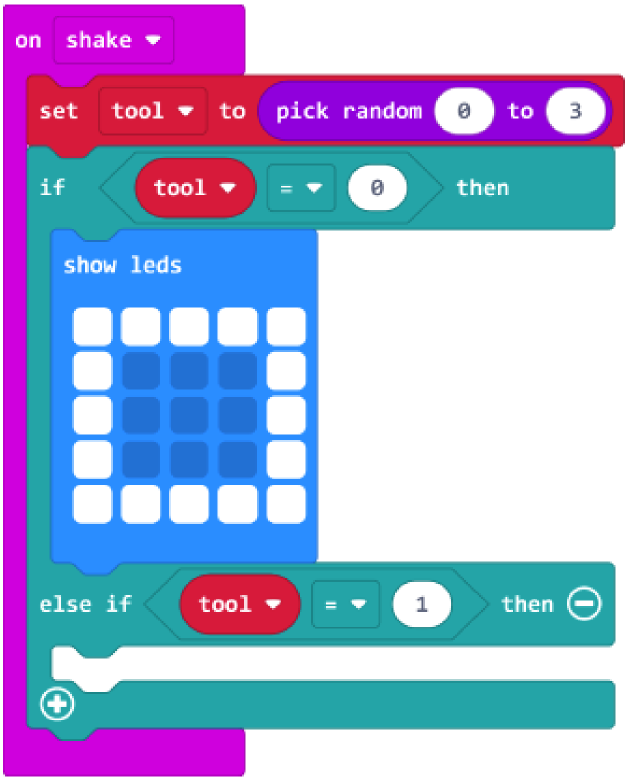


```
on shake
    set tool to pick random 0 to 3
    if tool = 0 then
        show leds
```

```
tool = 0

def on_gesture_shake():
        tool2 = randint(0, 3)
        if tool2 == 0:
                basic.show_leds(“””
                # # # # #
                # . . . #
                # . . . #
                # . . . #
                # # # # #
                “””)
input.on_gesture(Gesture.SHAKE, on_gesture_shake)
```

Step 5

Add an else if block to the if block and check whether tool is equal to 1. Click on the (+) at the bottom of the if block to add an else if section.



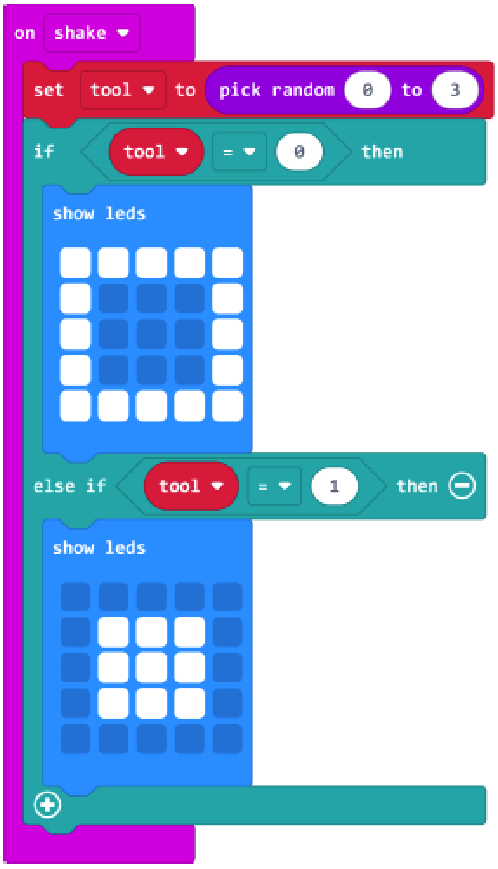
```
on shake
  set tool to pick random 0 to 3
  if tool = 0 then
    show leds
  else if tool = 1 then
```

```
tool = 0

def on_gesture_shake():
        tool2 = randint(0, 3)
        if tool2 == 0:
                basic.show_leds(“””
                # # # # #
                # . . . #
                # . . . #
                # . . . #
                # # # # #
                “””)
        elif tool2 == 1:
                pass
input.on_gesture(Gesture.SHAKE, on_gesture_shake)
```

Step 6

Place a show leds block under the else if and draw a rock image on the screen.

```
tool = 0

def on_gesture_shake():
    tool2 = randint(0, 3)
    if tool2 == 0:
        basic.show_leds(“””
            # # # # #
            # . . . #
            # . . . #
            # . . . #
            # # # # #
            “””)
    elif tool2 == 1:
        basic.show_leds(“””
        . . . . .
        . # # # .
        . # # # .
        . # # # .
        . . . . .
        “””)
input.on_gesture(Gesture.SHAKE, on_gesture_shake)
```

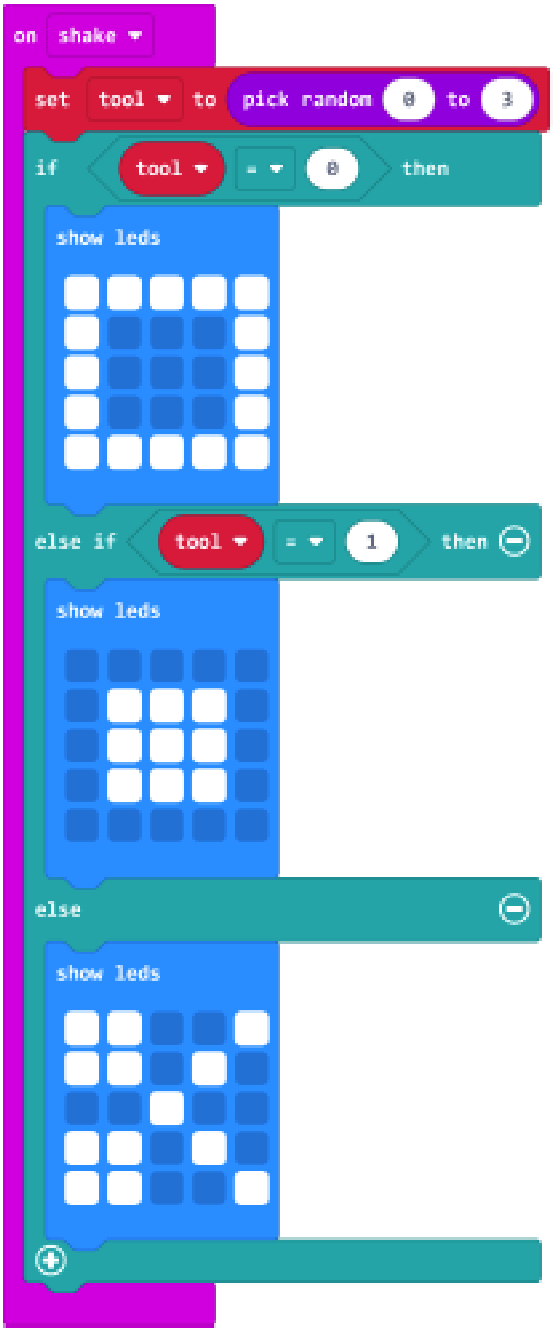


```
on shake ▾
    set tool ▾ to pick random 0 to 3
    if tool ▾ = ▾ 0 then
        show leds
    else if tool ▾ = ▾ 1 then ⊖
        show leds
    ⊕
```

Step 7

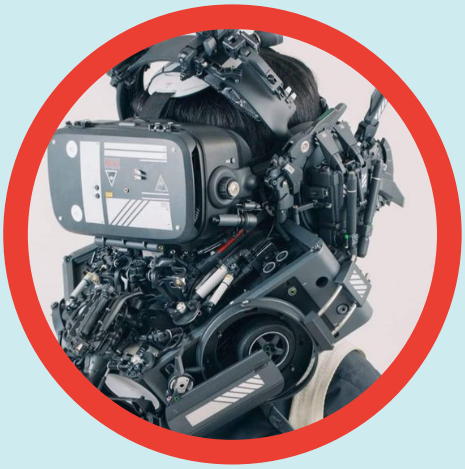
Add a show leds block with a picture of scissors to the else part. No need to check if tool is 2 because 2 is the only number left out of 0,1, and 2. That is why we cause an else instead of an else if.

```
tool = 0

def on_gesture_shake():
        tool2 = randint(0, 3)
        if tool2 == 0:
                basic.show_leds(“””
                # # # # #
                # . . . #
                # . . . #
                # . . . #
                # # # # #
                “””)
        elif tool2 == 1:
                basic.show_leds(“””
                . . . . .
                . # # # .
                . # # # .
                . # # # .
                . . . . .
                “””)
        else:
                basic.show_leds(“””
                # # . . #
                # # . # .
                . . # . .
                # # . # .
                # # . . #
                “””)
input.on_gesture(Gesture.SHAKE, on_gesture_shake)
```



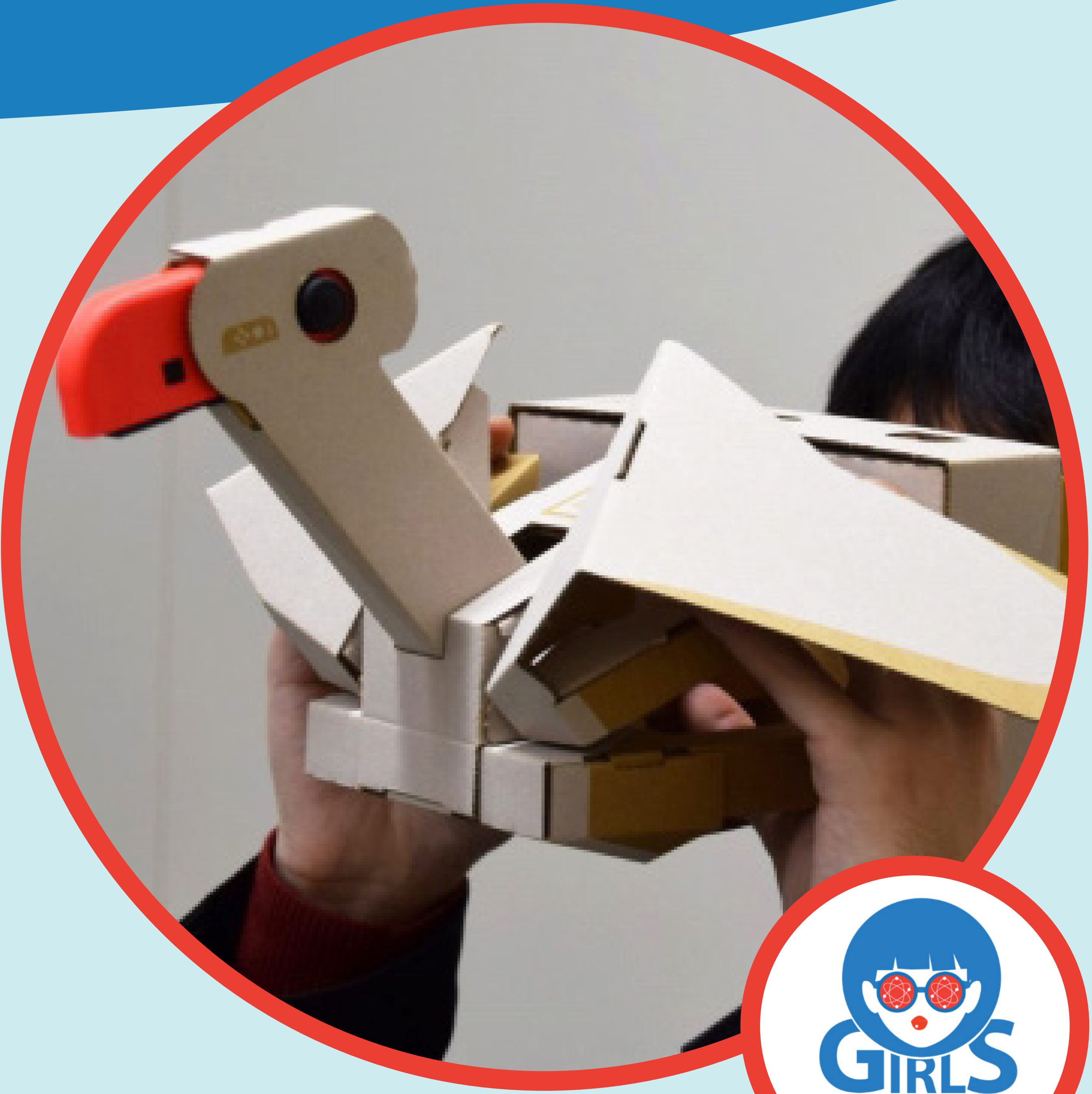
```
on shake
  set tool to pick random 0 to 3
  if tool = 0 then
    show leds
  else if tool = 1 then
    show leds
  else
    show leds
```

The end strategy of a simple Rock Paper Scissors game is to be random and fast. Statistically, each attack will tend to occur just as frequently as another, given that each is equally effective.



DIY VR GLASSES

TUTORIAL



VR GLASSES

Workshop description

Right now all VR glasses are boring black boxes on your head. In this workshop you are going to change that! Together we are going to think of our own VR game or film and build a custom VR headset to match. When you are going home you will have your own personalized VR headset that you can use to watch VR movies and play VR games.

Workshop progress

Warm Up exercise (15 Min)
Create your own VR hero exercise (30 min)
VR headset inspiration (10 min)
Create your headset design (30 min)
Break (20 min)
Putting the Cardboard VR glasses together (15 min)
Designing your VR headset (45 min)

Watching VR movies with your VR headset (15 min)

- Target audience: 12 -16 year
- Participants: 8-12
- Time: 3 hours

Materials needed

1. WORKSHOP SUPERVISOR

Beamer / TV

2. BUILDING THE HEADSET

12 cardboard VR headset

paper to write on

paper to draw on

pens and markers

Art and craft material

- Glue / tape / glue gun / ...
- Colorful decoration
- Cardboard / styrofoam / play - doh
- paint / spray paint / ...
- cutting knives / scissors
- brushes
- foil / bubble wrap / feathers / …
- ....

3. PARTICIPANTS

Smartphone

4. SOFTWARE

youtube App

within

WARM UP EXERCISE

15 min

INTRODUCTION

VR is a whole new way of seeing and imagining reality. To share your experiences in VR you need to practice your imagination and communication skills.

EXERCISE

Split up the group in pairs of two. One person puts their hands on their head in the form of imaginary VR glasses and closes their eyes. This person is now sitting in the middle of the room of the other person's bedroom. The person with the VR goggles looks straight in front of him and asks what he sees. When the person not in VR has explained the view, the person with the VR goggles can now freely look around and the other has to narrate.

Tip: This exercise is also useful to get to know the other person. So motivate people to ask personal questions about certain objects or items. After 5 min switch rolles.

DIY VR HERO EXERCISE

30 min

INTRODUCTION

Every game / film / experience has a main hero, protagonist. We are going to create our own character so we can design a VR headset to go along with their story.

EXERCISE

Imagine the hero of your story. It can be a fantasy story about a girl saving the world. Or a story about a boy who is running away from home to learn how to dance. Now everybody writes down 5 character traits of their hero, main protagonist. (Shy, strong, beautiful, clever, vegetarian, helpful ...).

Now pass the list with traits to somebody else. This person is going to draw the protagonist based on the character traits. When finished return the list with traits and the drawing of the protagonist.

Tip: Ensure to everybody that the drawings don't have to be pretty or perfect. They just have to get the idea across. You can even draw the surroundings as wel to help people understand the character or situation.

VR HEADSET INSPIRATION

10 min

INTRODUCTION

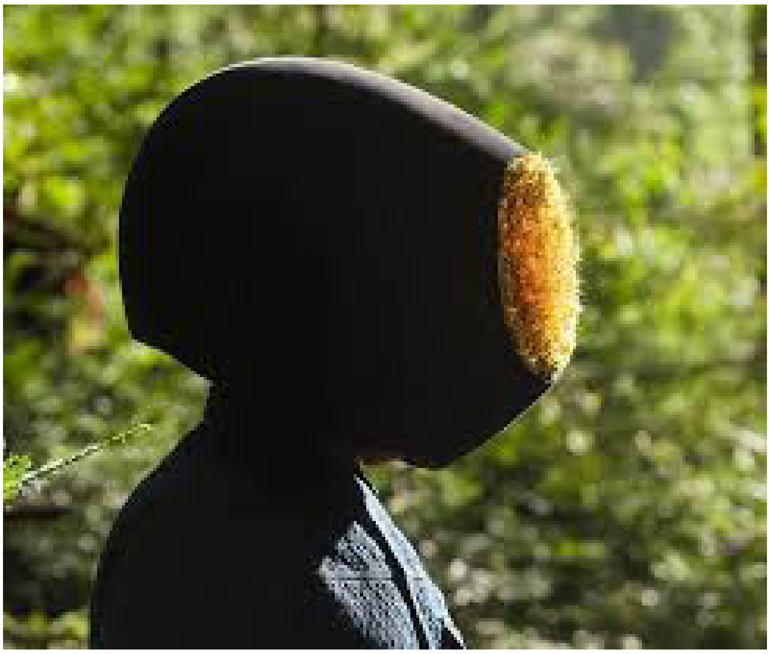
VR glasses have been made by scientists and developers to give us really cool experiences inside the headsets. But the headsets themselves look boring and kinda stupid when you are wearing them. Now it's the job of artists like you to make them beautiful and cool!

EXERCISE

In the eyes of the animal

In the eyes of the animal is a VR film that lets you see the world like animals. To enhance this experience they designed custom VR glasses with living moss on it. While experiencing the VR story you also smell the plants giving you a stronger connection to nature.

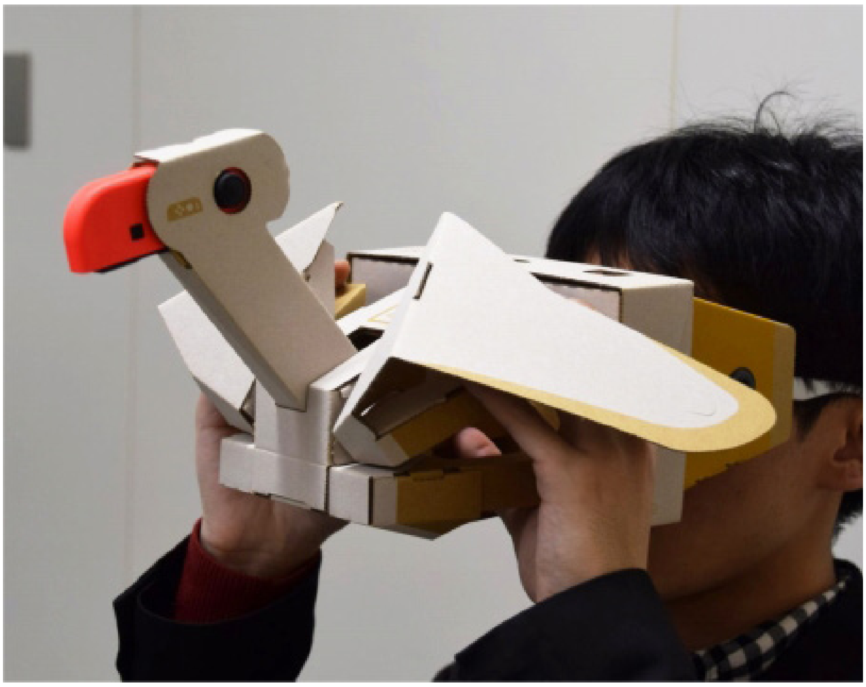
https://youtu.be/SDWuSNPSb9g?t=58



Nintendo VR

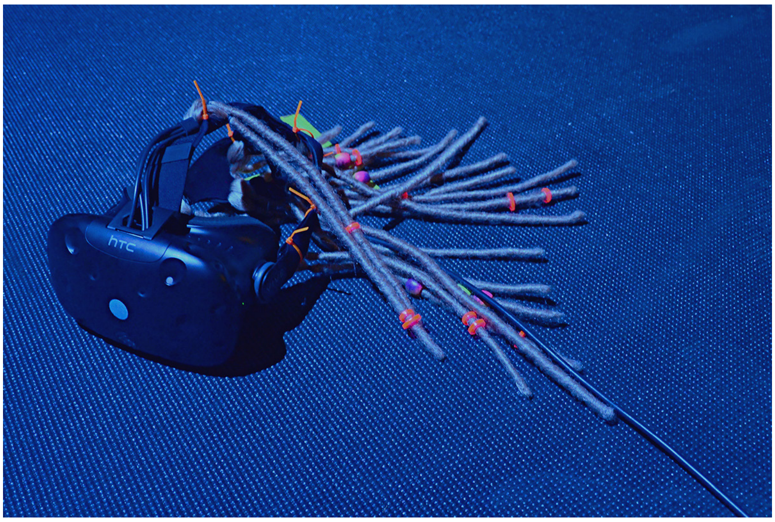
Toycon Bird is a VR game from Nintendo that lets you fly through the sky like a bird. Not only do the VR glasses look like a bird. The functionality of flapping the wings also adds to the feeling of flying because you are flapping air in your face.

https://youtu.be/xmvnENhc0W4?t=178



Antigone

Antigone is a theater piece that uses VR to tell a story in a new way. One of the actors controls a character using the VR headset and controllers. The character looks like a big surfer dude with long dreadlocks. So the headset the actor wears also has dreadlocks to connect the actor and the character in VR.



https://www.youtube.com/watch?v=L9z-z1zMG_s&t=20s

CREATE HEADSET DESIGN

30 min

INTRODUCTION

We have just seen a few examples of how a VR film or game can influence how the design of the VR glasses look like.

EXERCISE

Think of the protagonist and the story you just imagined in the previous exercise. You can take the character traits and the drawing of your protagonist and imagine what kind of VR headset he would have. How can you design VR glasses that may resemble the main protagonist style? How can the VR glasses help you feel more immersed in the story of your main protagonist?

Tip: If a person is stuck try imagining more of the story with that person. How does the protagonist feel? What are his struggles and what are his goals?

MAKE VR GLASSES

15 min

INTRODUCTION

Most cardboard VR glasses come in a package you still have to assemble. Don't be afraid to break something. It's cardboard so you can always tape it back together.

EXERCISE

Follow all the steps to assemble your cardboard VR glasses

https://www.youtube.com/watch?v=5mSW5GyFozg

Tip: Some cardboard VR glasses have glue on them but it's always safer to use some tape to make sure it doesn't fall apart.

DESIGN VR HEADSET

45 min

INTRODUCTION

You have your own VR glasses! Now let's start making them fabulous!

EXERCISE

You've just imagined a protagonist, a story and VR glasses to go with them. You have everything you need to start building your own personalized VR glasses. Are you going to build an amazing construction around your glasses? What are you going to put on your VR glasses so the viewer feels more immersed in the story? Be careful with knives and hot glue!

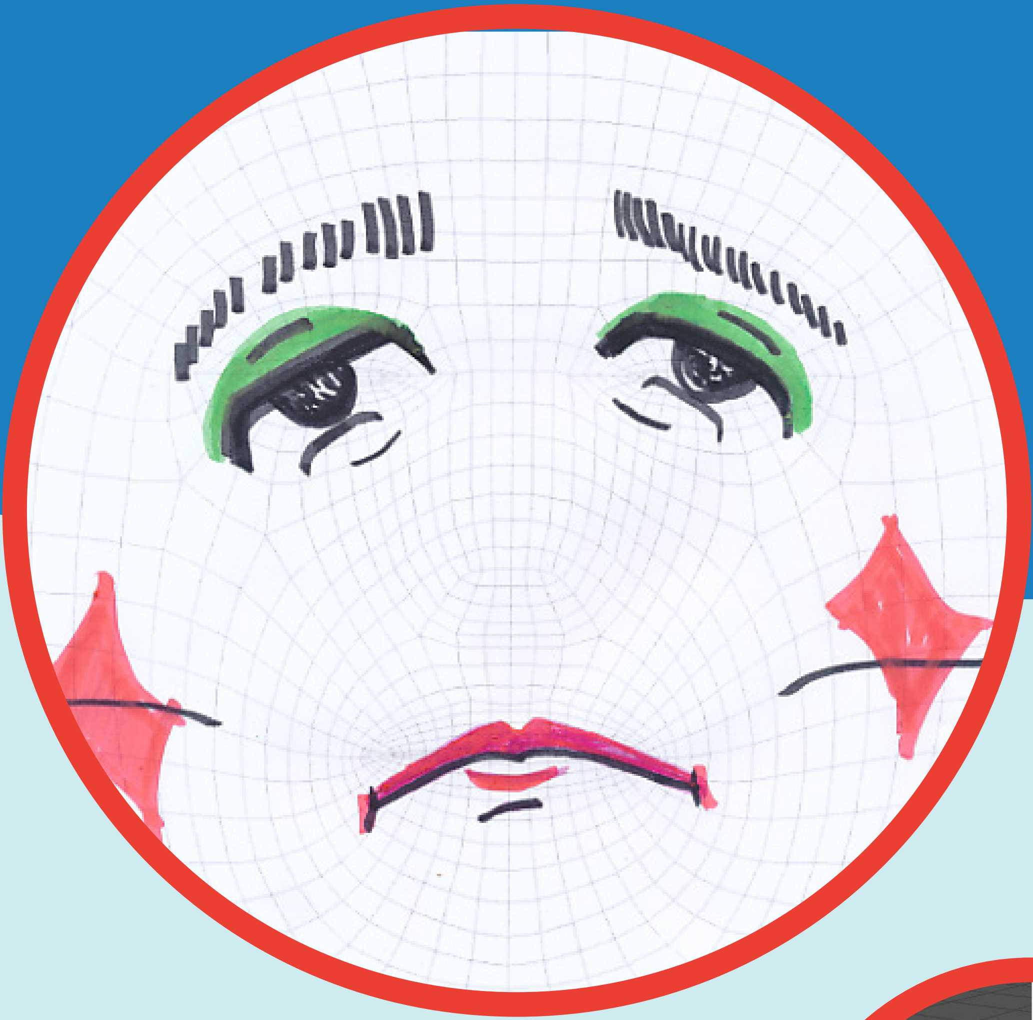
WATCH VR MOVIES

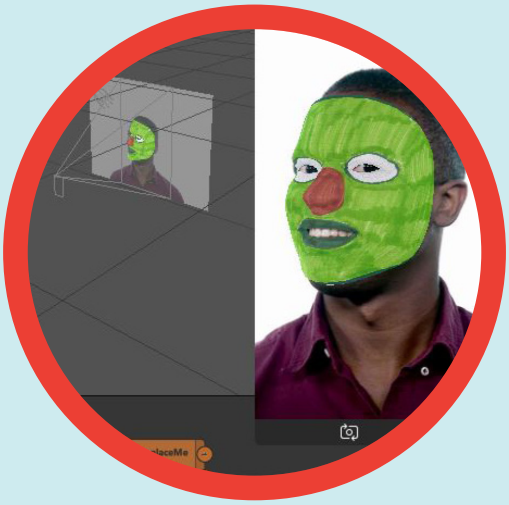
45 min

EXERCISE

Once you are happy with your VR glasses you can install the app WIthin or search voor VR/360° videos on the youtube app and enjoy some VR films. How does the design of the VR glasses influence the way you experience the story? What does the person wearing the VR glasses look like?

Tip: Most people immediately look for rollercoaster or horror videos. This can be fun but maybe for the first time they can try out national geographics wildlife VR film. So to not get sick or scared the first time they try VR.





AI FILTERS

TUTORIAL

AI FILTERS

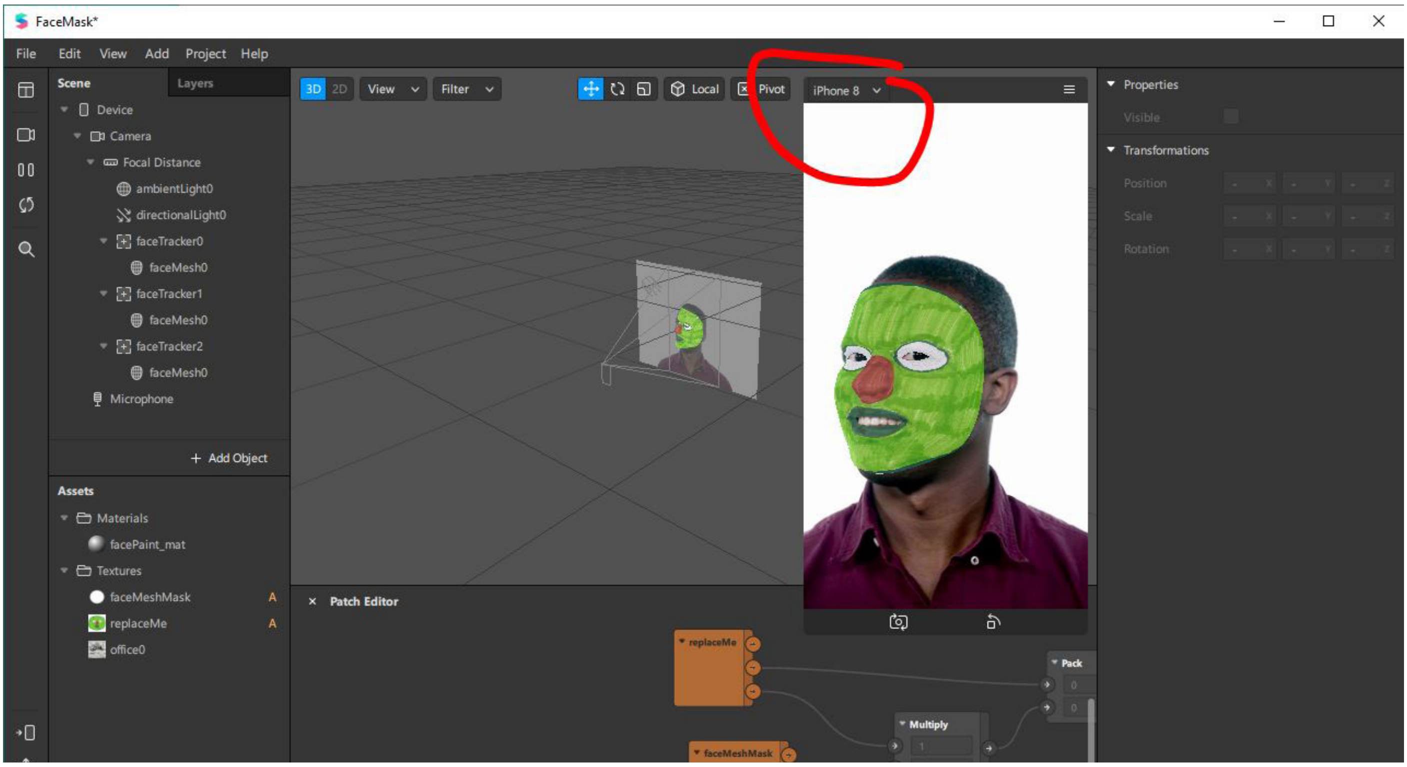
For instagram

Using Spark AR you can now make your own Instagram face filters without needing to know anything about technology. You can start designing your own masks, create your own makeup and share funny filters with your friends.

WORKSHOP PROGRESS

- Warm up exercise (40 min)
- Exploring Instagram filters (20 min)
- Designing Instagram filters (40 min)
- Trying out our own Instagram filters (20 min)

- Target audience: 12 -16 year
- Participants: 8-12
- Time: 2 hours
- Skill level needed: Medium



1. WORKSHOP SUPERVISOR

Beamer / TV
External webcam
Scanner
Wifi

Enough windows for half the participants.

2. MAKING ANALOG FILTERS

- 6 white chalk markers
- 12 colored chalk markers
- 12 animal cards
- 12 painting style cards
- paper to write on
- paper to draw on
- pens and markers
- 12+ Mask templates

3. PARTICIPANTS

Smartphone

4. SOFTWARE

Spark AR studio: https://sparkar.facebook.com/ar-studio/learn/downloads#spark-ar-studio
Facebook account
Instagram

5. VISION MATERIAL

Instagram filter
https://www.youtube.com/watch?v=L8jLaAYbSAk

WARM UP EXERCISE

40 min

INTRODUCTION

Today we are going to make our own Instagram filters. We are going to design masks and virtual make-up that we can share with our friends and family. Before we start designing our own Instagram filters. Let's practice our drawing skills a bit.

EXERCISE

Everybody in the group gets 6 pieces of paper. On these pieces you write down 6 facial features or accessoires (f.e. Glasses, blush, beard, tattoo, makeup etc.)

Now collect all facial features. Divide the participants up in 2 groups and give one group the white chalk markers.The other group is going to stand behind the window and be the models.

The group with chalk markers has 10 minutes to draw the models on the glass. After everybody is ready you switch the participants and now the other group draws.When all participants are drawn on the window you retrieve the chalk markers. Give all the participants colored chalk markers and 2 pieces of paper with a facial feature on each of them.

The participants now have 10 minutes to add the facial features to their drawings of the other participants. Take a moment to look at each other's drawing and question the drawings. Do they still look like the person? How did the filters change their look?

Tip: closing one eye makes it easier to draw the other person on the glass.

Mi

EXPLORING INSTA FILTERS

20 min

INTRODUCTION

Now is the time to try out some Instagram filters. To find new Instagram filters you can create stories or go to a creators instagram profile.

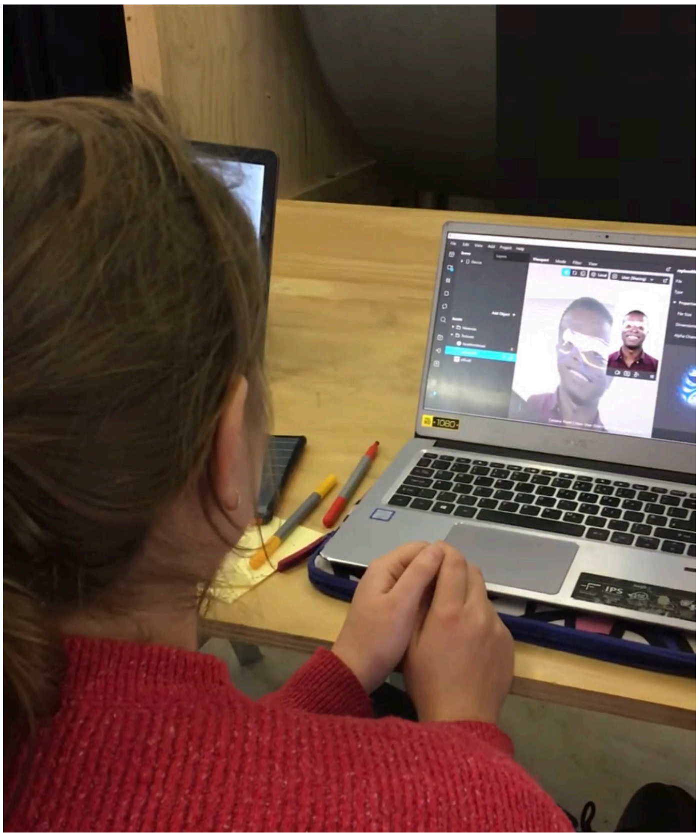
You can find a step by step guide how to find filters in the drive. Underneath is a collection of really inspiring and cool creators.

EXERCISE

Every participant can try out some filters, look at some different creators and some different types of filters.

While the participants are trying out filters, write down their insta-username and one emotion (happy, sad, crazy, lonely, in love, hungry, ...) on a piece of paper. When you have collected everybody's names, mix the pieces of paper and distribute them to the participants.Now invite everybody to create a short insta-story using a filter inspired by the feeling written on the paper and send it to the other person.

Tip: Some filters are world filters. You can always help the participants make a fun story by being their cameraperson.



DIY FACE FILTER

40 min

INTRODUCTION

We just tried out many funny and weird filters. Some of them are really complex and use 3D models to change the shape of our face or even distort reality itself.

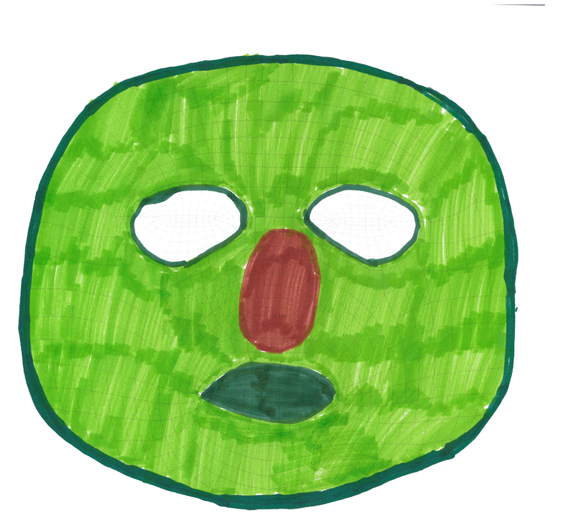
Now we are going to design our own animal mask using a very easy face-template.

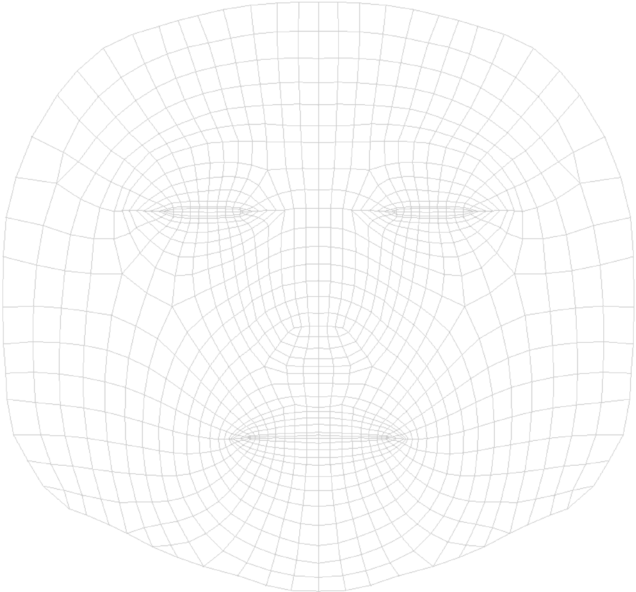
EXERCISE

Give every participant a face-template and let them pick an animal and art-style. Inspired by the art style they can design their own animal mask.

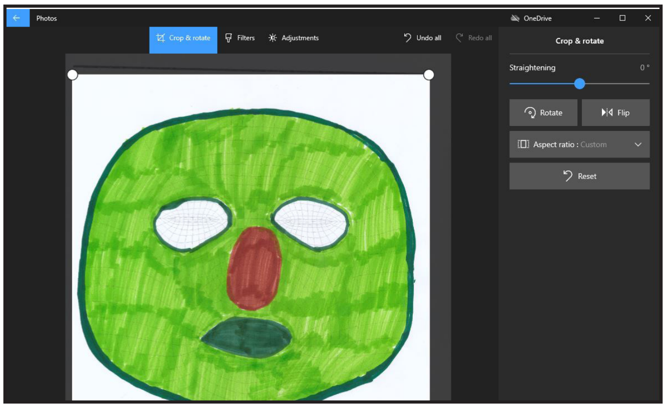
Inform the participants to color in their entire face, but everything that is drawn outside of the face template won't be on the filter. So no need to be carefull and draw inside the lines

Zie template op de volgende pagina..

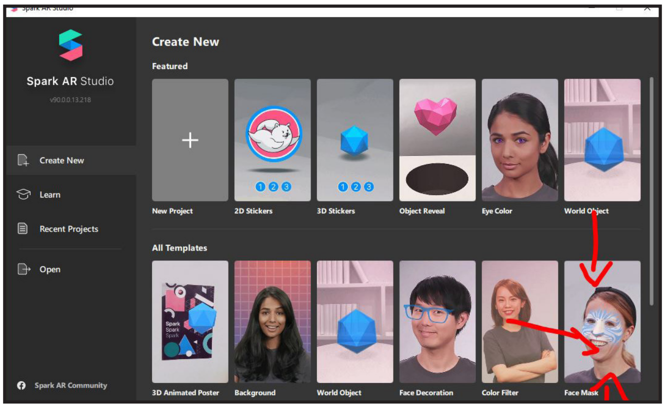




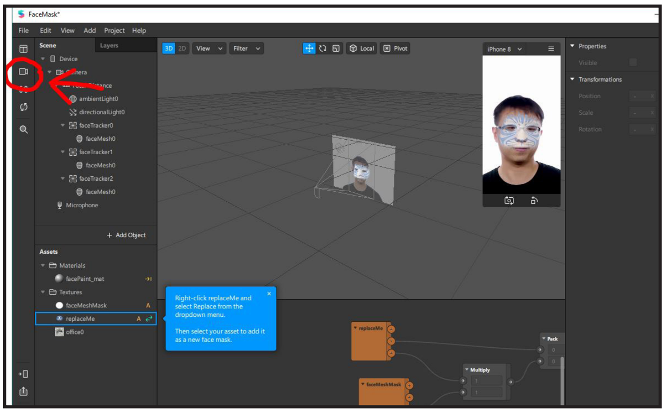
STEP BY STEP



1. Scan the drawing. Crop the drawing into a square on the black lines on the template



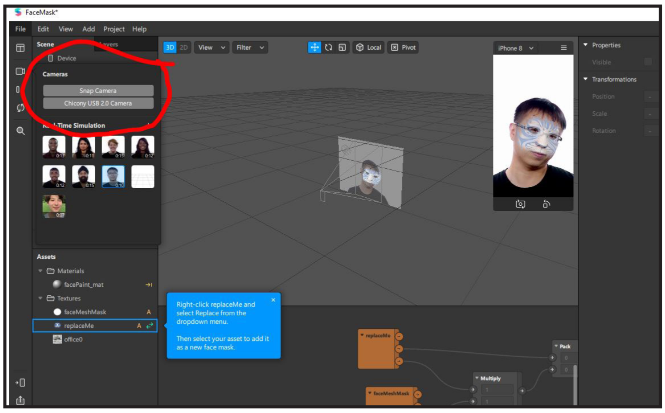
2. Open Spark AR studio. Pick the Face Mask template



Right-click replaceMe and select Replace from the dropdown menu.

Then select your asset to add it as a new face mask.

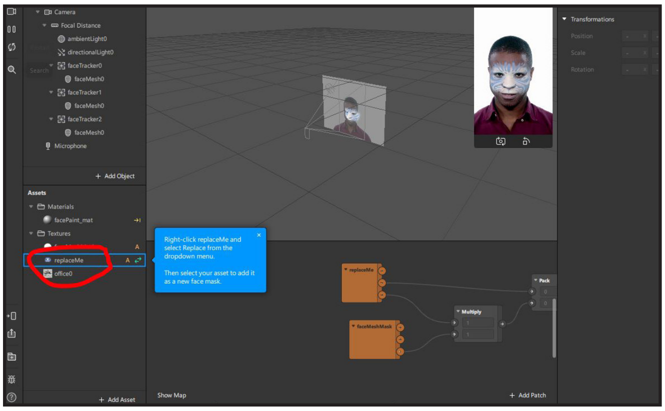
3. Click on the little camera icon on the left of the screen



Right-click replaceMe and select Replace from the dropdown menu.

Then select your asset to add it as a new face mask.

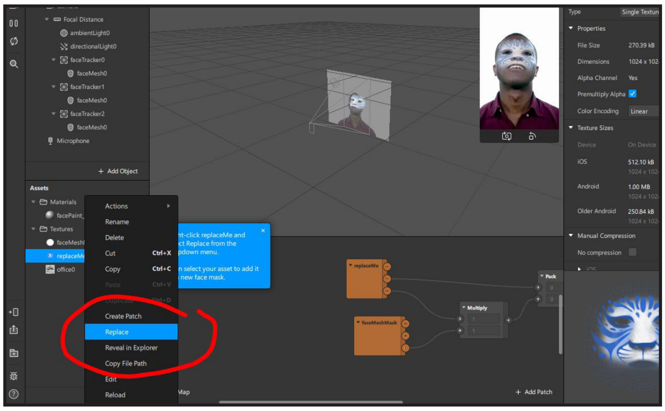
4. Select your webcam.



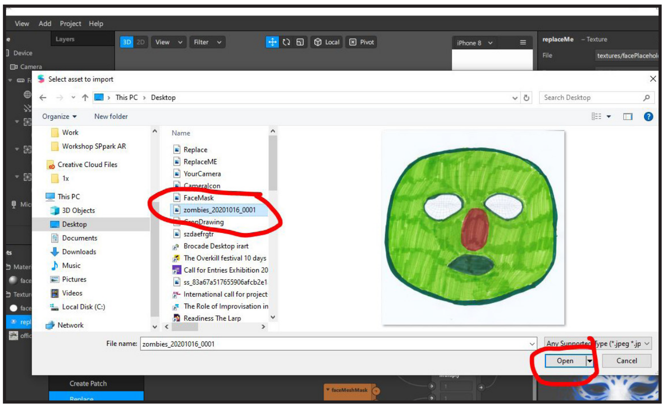
Right-click replaceMe and select Replace from the dropdown menu.

Then select your asset to add it as a new face mask.

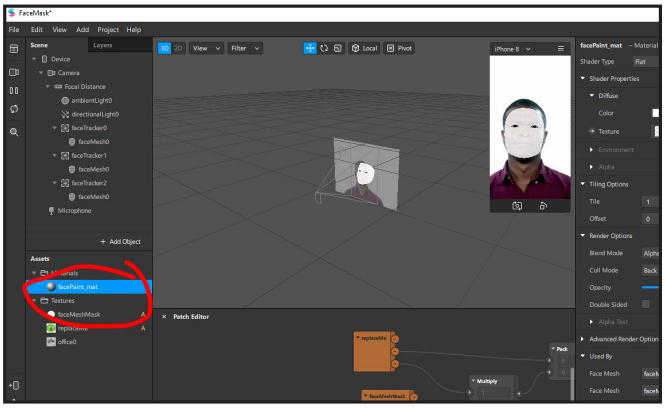
5. Now right click on the file called replaceMe



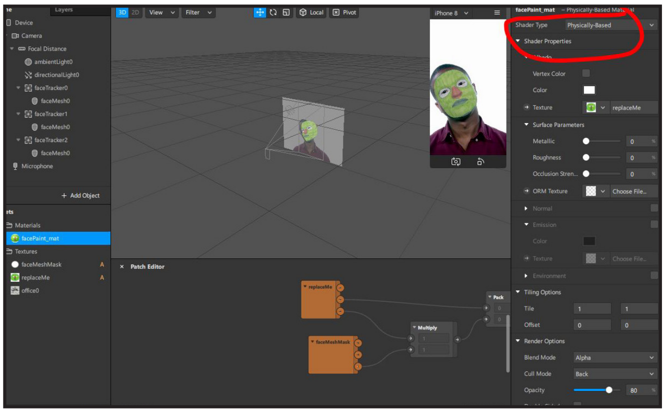
6. Choose replace in the drop down menu



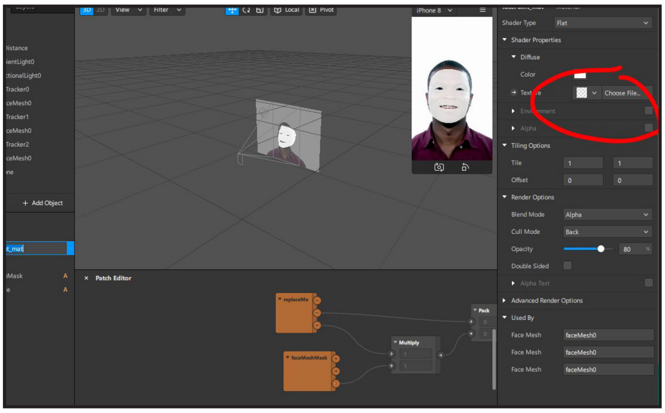
7. Add the drawing you just scanned



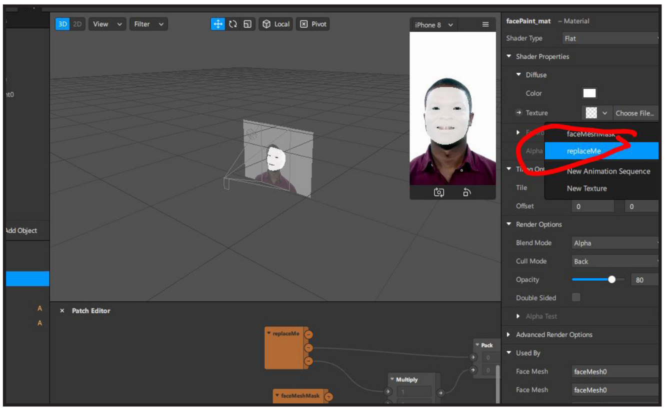
8. Click on the material facePaint_mat



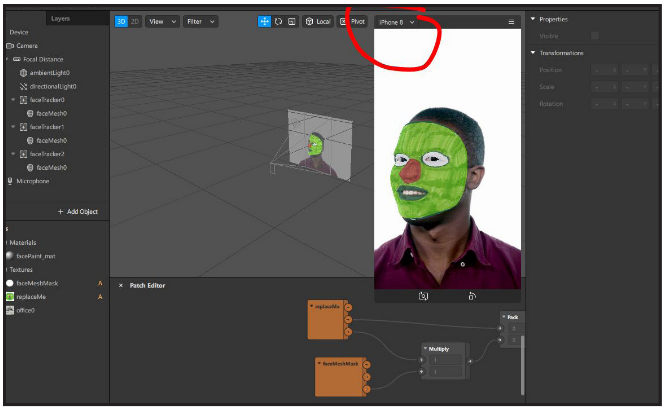
9. On the right side



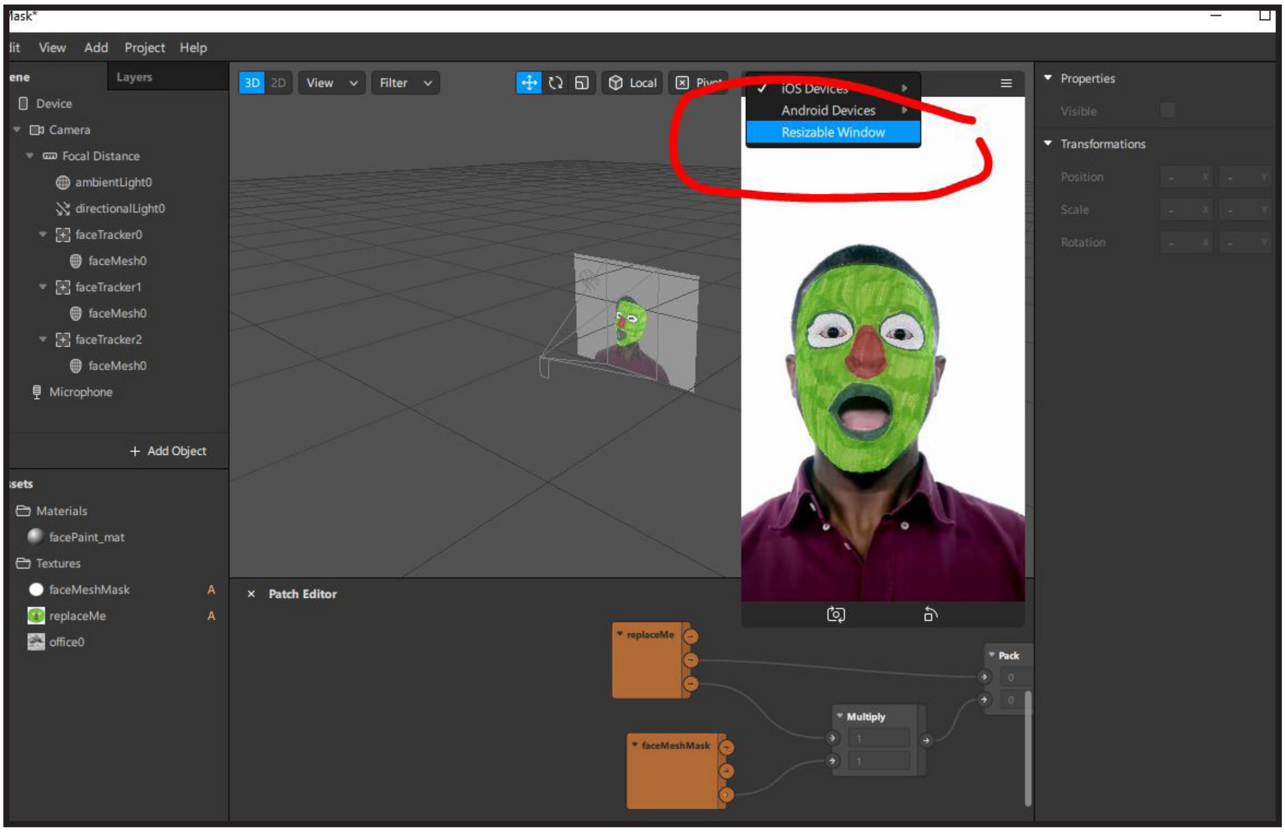
10. Change the shader type from physically based to Flat



11. Change the texture to replaceMe



12. To make the window bigger click on Iphone 8 and change it to resizable window



13. Press control/command S to save your filter.

To change the filter you can replace the drawing again.If everybody's drawings are done and scanned. You can create one filter using everybody's designs.

1. Click add assets underneath the screen
2. Choose Animation sequence
3. Select all the drawings
4. On the right change FPS (frames per second) 24 to 1
5. Select Randomize
6. On the left click on the material facePaint_mat
7. On the right change the texture replace me to the animationSequence0

Tip: If not the entire mask template is colored, it is best to remove the white background in order to only cover the part of your face that is colored.

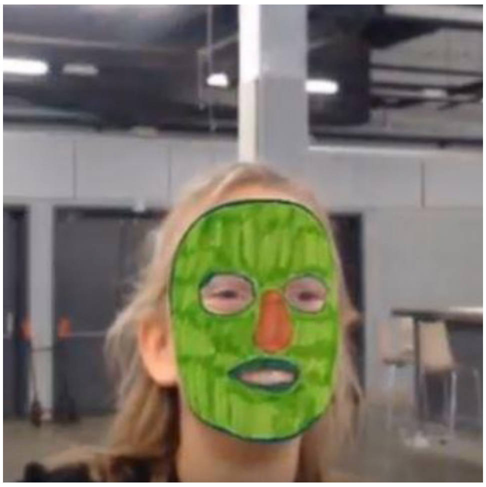
TRY IT OUT!

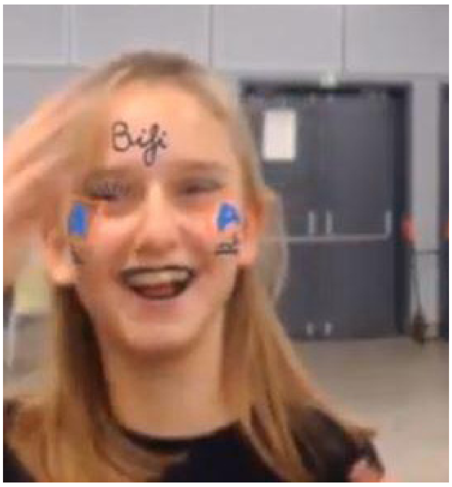
20 min

EXERCISE

Up to 3 people can stand in front of the webcam and experience all the face filters. Ask them to make a cool facial expression to celebrate their creation. You can make screenshots or use screen recording software to make some cool footage.

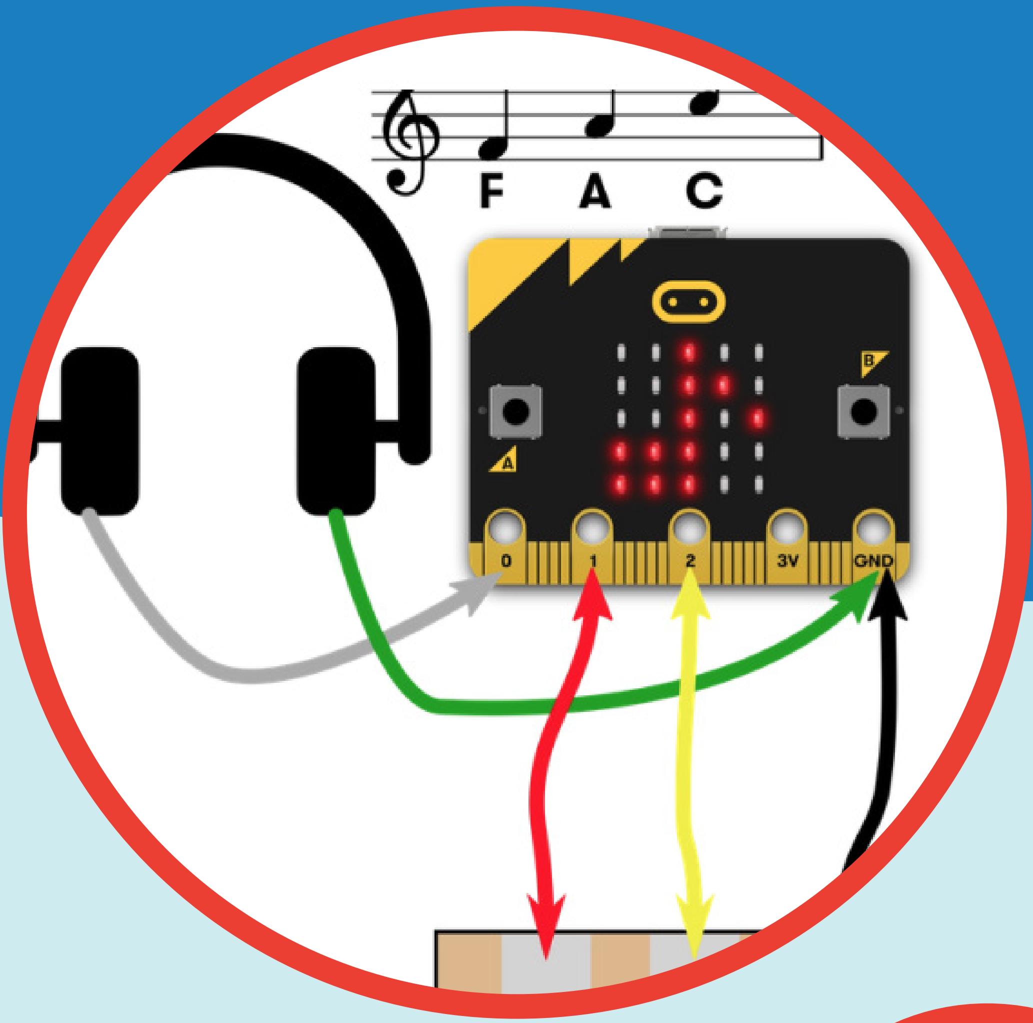
Ask the participants what they think of their designs. If anything is different now they see it on their own face. What went well and what would they change? What kind of masks would they want to design next?





HOW TO UPLOAD INSTAGRAM FILTERS

https://sparkar.facebook.com/ar-studio/learn/docs/submitting/


F
A
C
B
A
0
1
2
3V
GND

DIGITAL MOVEMENT INSTRUMENT

TUTORIAL

DIGI MOVEMENT INSTRUMENT

Workshop description

The goal of this workshop is for girls to have fun playing with music and programming. The girls will learn how to create a music instrument by programming it to make sound by movement. To enable this they will use a micro:bit and its compass and accelerometer

Workshop progress
Warm up exercise (40 min)
Making a digital movement instrument (40 min)
Designing your digital movement instrument(40 min)
Trying out our own digital movement instrument (20 min)
Extra exercises (30)

- Target audience: 12 -24 year
- Participants: 8-12
- Time: 3 hours
- Skill level: Medium

Materials needed

1. WORKSHOP SUPERVISOR

Beamer / TV

WiFi

2. WARM-UP EXERCISE

Kristal glasses or glass bottles to make sound (at least 5 of them)

3. DIGITAL MUSIC MACHINE

Micro:Bit (1 per participant or team)

Battery pack

Batteries

Crocodile clips (5 per participant or team)

Wired headphones, buzzer or speaker (1 per participant or team)

Cardboard

Tin Foil

Pens, markers, glue sticks, scissors, ...

4. SOFTWARE

Make Code Editor

https://makecode.microbit.org

WARM UP EXERCISE

40 min

INTRODUCTION

Today we are going to make our very own Digital Movement Instrument. We will learn about inputs and outputs, how to code with blocks and how to program music.

To practice we will experiment with the basic components of music.

EXERCISE

Ask the question: Who knows some of the basic components of music? (There are more, but the focus today lies on the elements below. Via this link you can find more information)

- Rhythm
- PITCH or NOTE or TONE
- Melody (A coherent succession of pitches.)
- Harmony (The relationship of pitches as they sound simultaneously.)

As a warm up we will explore each of these elements with some little exercises together.

Lets start with rhythm

The participants will create music by clapping to a certain rhythm.

You start off with telling one participant to clap her hands to a certain rhythm. Then you ask the next participant to clap her hands to a double rhythm. Then ask the next participant to join in, either on the same rhythm as 1 of the first 2, or propose another rhythm. They can also freestyle. The goal is just to have some fun searching to a good rhythm together.

PITCH or NOTE or TONE

Watch this short clip of the comedy series 'The big bang theory' to explain the difference in high or low tone. https://www.youtube.com/watch?v=Supu0lB71d4 After you watch this video practice talking in higher and lower tones.

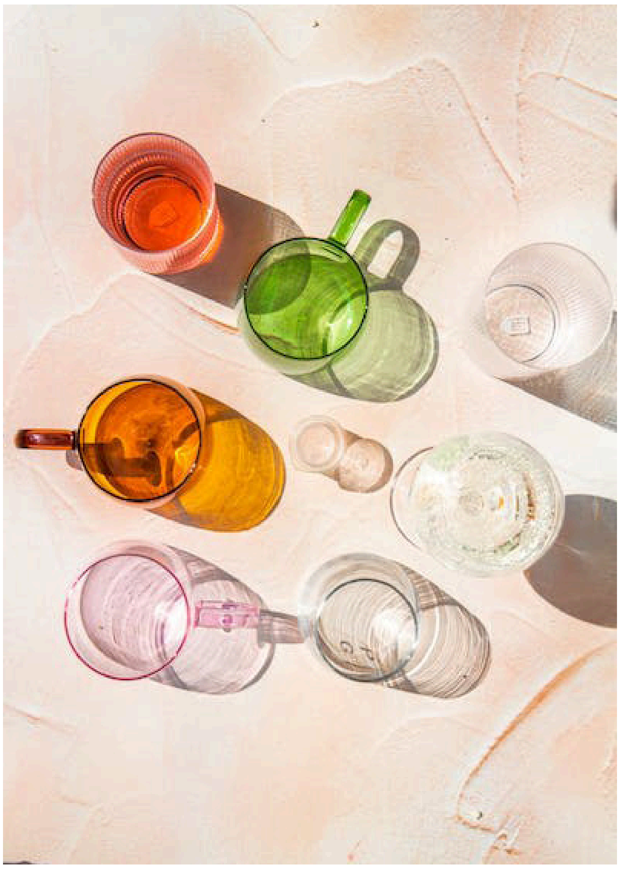
To demonstrate tone height in another way, use water bottles or kristal glasses with different levels of fluid in them. This is shown in the video below: https://www.youtube.com/watch?v=0YSyrotk3SI You can also blow on the bottles instead of using a spoon.

Melody (A coherent succession of pitches.)

Once you have the bottles or glasses set up, you can demonstrate a melody. You can even play a little song. Let the participants play with the setup.

Harmony (The relationship of pitches as they sound simultaneously.)

To demonstrate harmony, you can let the participants use the bottles or glasses to play different sounds at the same time. Do they sound good together or not? Try to find sounds that sound good together, "Now that's harmony!".



CREATING SOUND

40 min

INTRODUCTION

Now is the time to create our DINSTRUMENT – Direction music INSTRUMENT! First we have to program the Micro:Bit so that it makes sound when it moves. We will first simulate this on the computer and then test it on the micro:bits themselves. To start, surf to the MakeCode editor and create a new project.

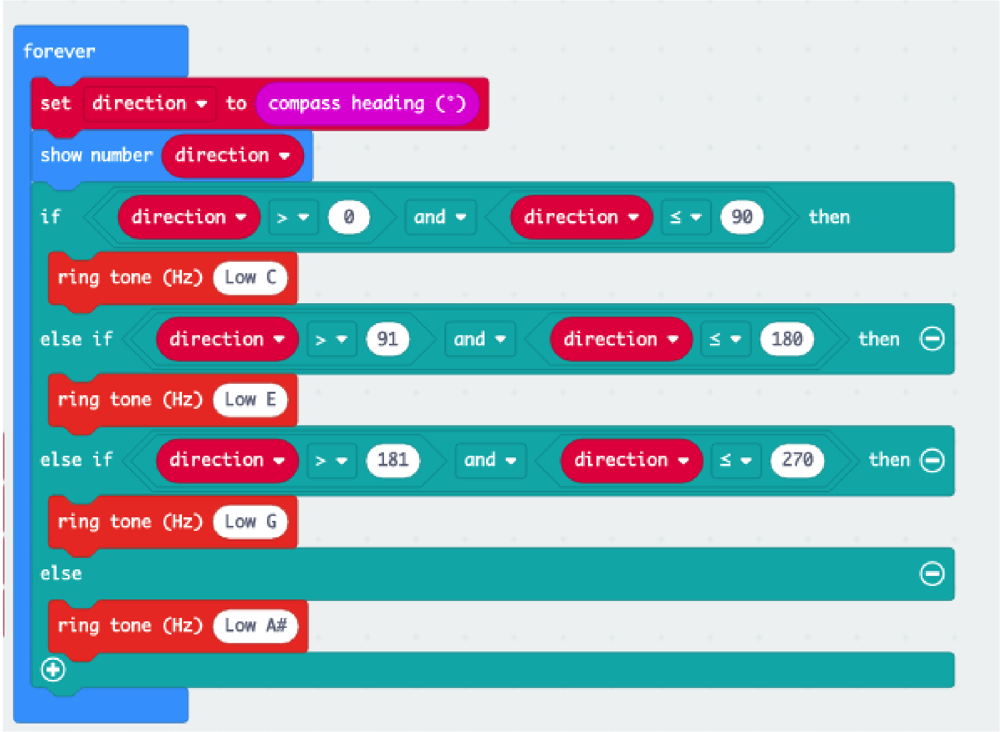
Tip: You can first do the other exercises from the Micro:bit workshop to practice working with the Micro:bit and MakeCode editor.

EXERCISE

To make music with the microbit we will start off with programming it to play 4 notes, one for each direction.

Show the 4 notes on the microbit

- Use blocks: Forever
- Set 'Variable' to map 'input' from low … high ... to low … high …
- if 'Variable' > .. and 'Variable' < ... then …
- Ring tone = note 1
- Else if ..
- Ring tone = note 2



```
forever
  set direction to compass heading (°)
  show number direction
  if direction > 0 and direction ≤ 90 then
    ring tone (Hz) Low C
  else if direction > 91 and direction ≤ 180 then
    ring tone (Hz) Low E
  else if direction > 181 and direction ≤ 270 then
    ring tone (Hz) Low G
  else
    ring tone (Hz) Low A#
```

Explanation

In order to show the notes they first need to define a variable, which we named 'direction'. We will write the compass heading (°) to this variable.

We use the function 'set direction to …'.

Since one circle is 360°, we need to split it into 4 pieces for 4 notes. That means that for every 90° there is 1 note.

To split the 360° in 4 we use the function 'map compass heading (°) This maps:

1 as 0° to 90° note C

2 as 91° to 180° note E

3 as 181° to 270° note B

4 as 271° to 360° note A#

Tip: You can test this in the makecode virtual simulator or when you download the code to the microbit.

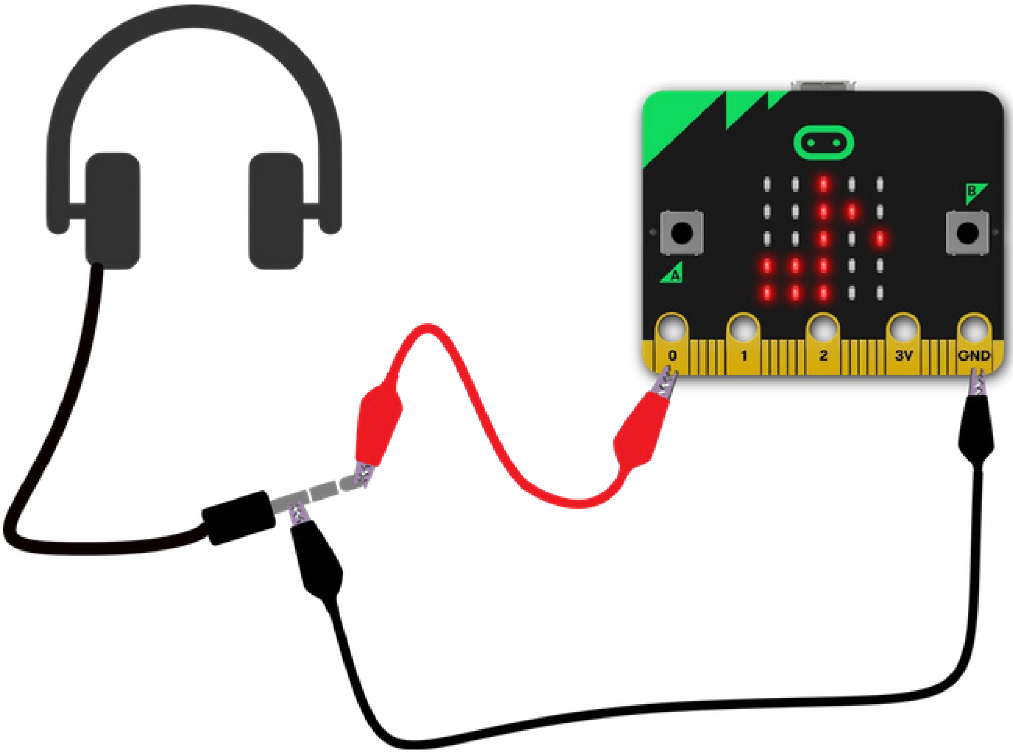
CONNECT MICRO:BIT TO HEADPHONE OR SPEAKER

Connect the micro:bit to headphones or a speaker

Connect your micro:bit to headphones or a speaker so you can hear sound.

The gold pins on the bottom of the micro:bit are used for inputs and outputs. Here we use pin 0 as an output. The micro:bit sends pulses of electrical signals from pin 0 when it plays the tune. The headphones must also be connected to the GND pin on the micro:bit to complete the electrical circuit.

Use 2 crocodile clips to clip the tip of the headphone plug to pin 0 on micro:bit. Clip the longer part of the headphone plug to the GND pin on micro:bit.



TEST THE DINSTRUMENT - DIRECTION MUSIC INSTRUMENT

Now we can test DINSTRUMENT by putting on the headphones or speaker and facing the microbit in different directions while connecting pin P1 with the GND pin.

You can make the connection with a crocodile clip.

DIY THE INSTRUMENT

40 min

INTRODUCTION

We just tried out the microbit with headphones or speaker. But now we want to integrate it into a full Digital Music Instrument. We will show this by making a guitar-like instrument, but the participants can get creative and create their own version of the instrument.

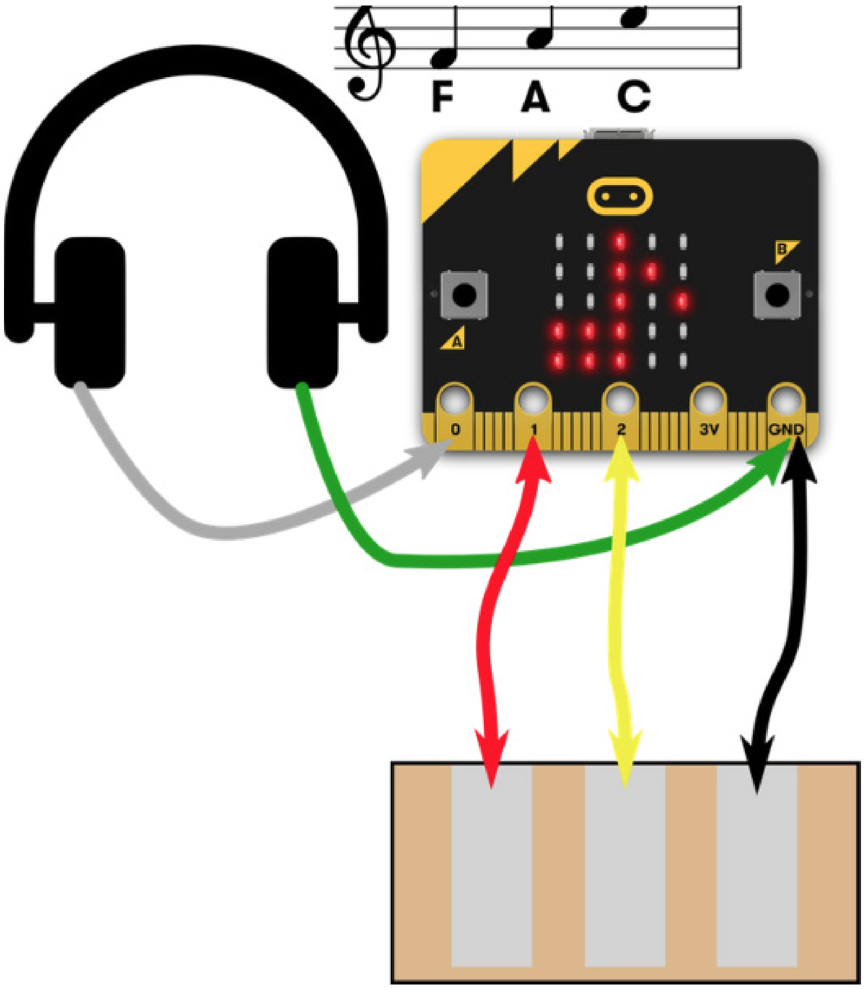
EXERCISE

Cut out a guitar shaped instrument and use 3 stripes of silver paper at the neck to use as frets. It's important that these silver paper frets don't touch each other. Also make sure there is room to clip the crocodile clips to the frets.

Tip: You can first make a prototype of the instrument before creating a beautiful one.

CONNECT THE FRETS TO THE MICRO:BIT

Connect a battery to the Micro:Bit so that you can play around without being connected to the computer. When touching pint 1 or pin2 make sure you also press pin 0.



+ A, B buttons

+ movement

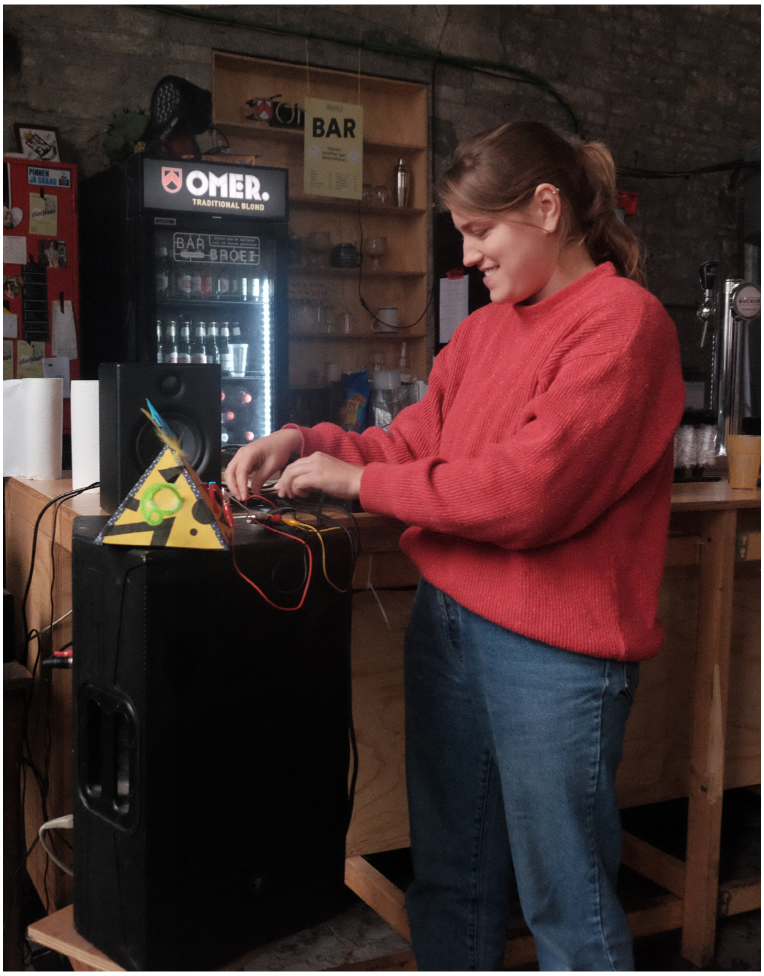
+ light sensor

TRY IT OUT

20 min

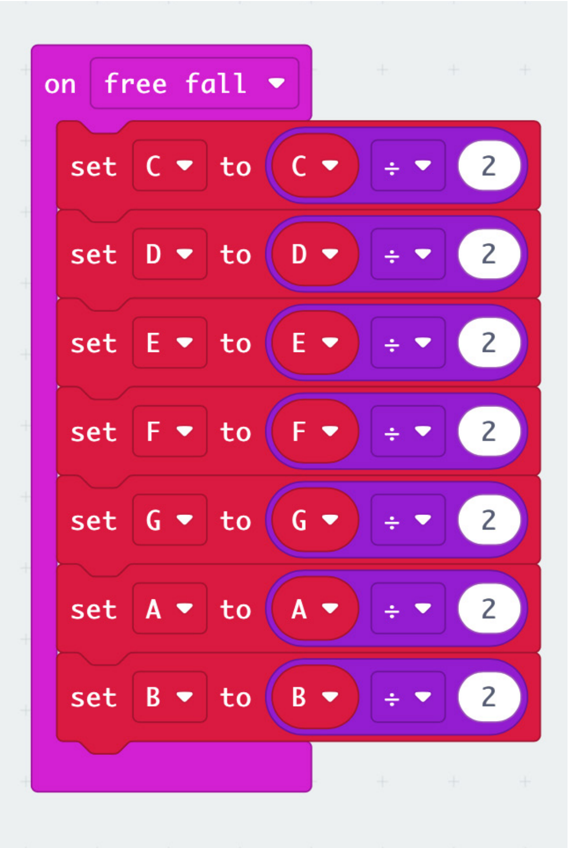
Try playing music with the digital movement instrument.

Ask the participants what they think of their designs and the sound it makes. Can they make it sound even better?

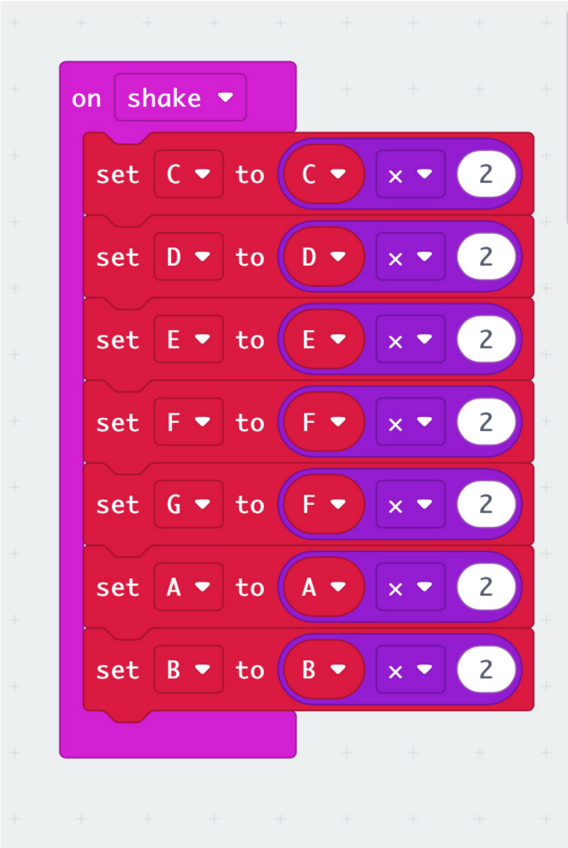


MORE EXERCISE

Change the pitch of the Digital Music Instrument.

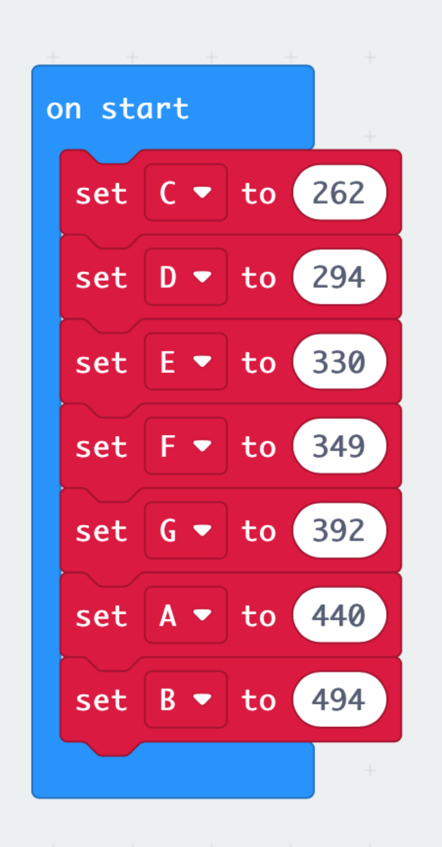


```
on free fall
    set C to C ÷ 2
    set D to D ÷ 2
    set E to E ÷ 2
    set F to F ÷ 2
    set G to G ÷ 2
    set A to A ÷ 2
    set B to B ÷ 2
```

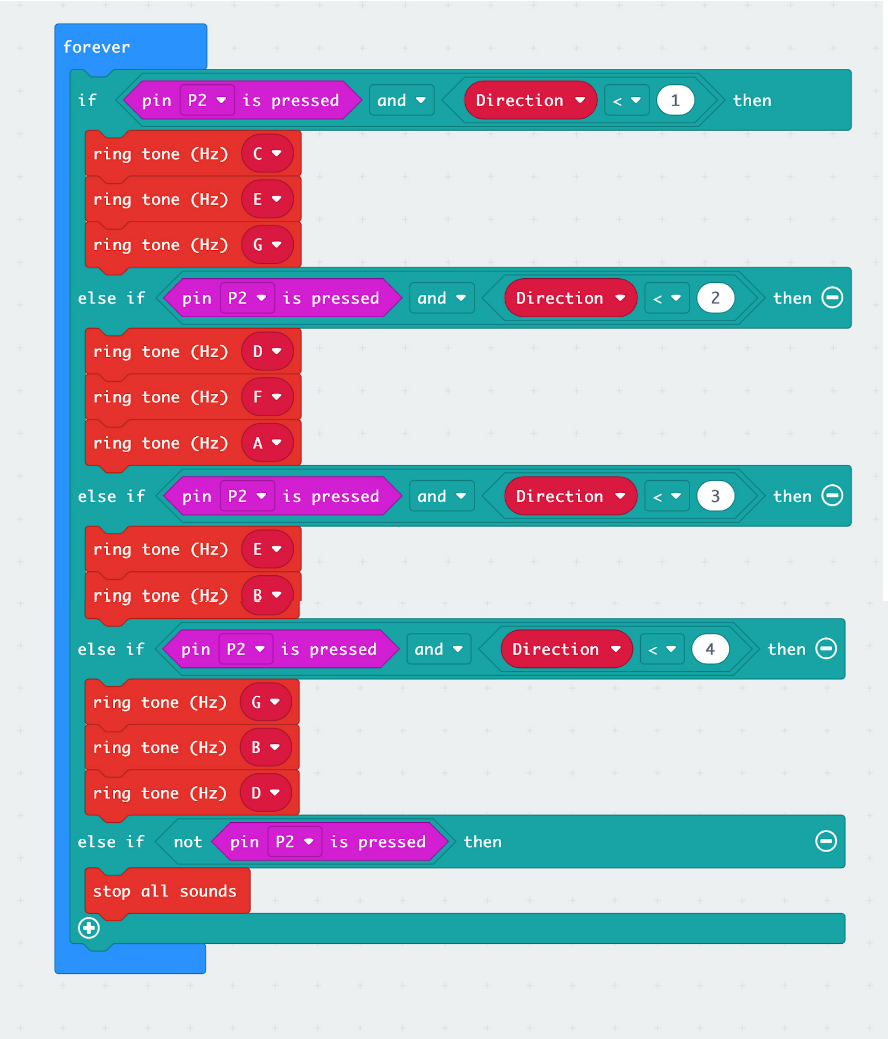


```
on shake
    set C to C × 2
    set D to D × 2
    set E to E × 2
    set F to F × 2
    set G to F × 2
    set A to A × 2
    set B to B × 2
```

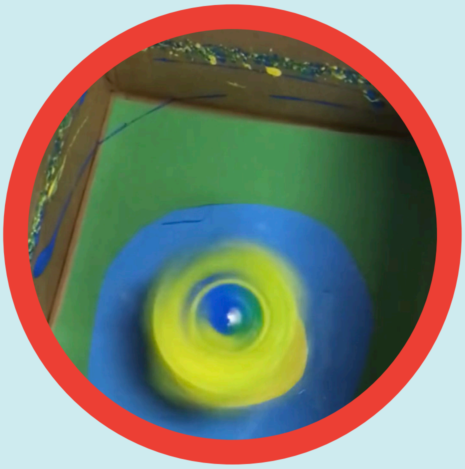
Play chords on the Digital Music Instrument.



```
on start
    set C to 262
    set D to 294
    set E to 330
    set F to 349
    set G to 392
    set A to 440
    set B to 494
```



```
forever
    if pin P2 is pressed and Direction < 1 then
        ring tone (Hz) C
        ring tone (Hz) E
        ring tone (Hz) G
    else if pin P2 is pressed and Direction < 2 then
        ring tone (Hz) D
        ring tone (Hz) F
        ring tone (Hz) A
    else if pin P2 is pressed and Direction < 3 then
        ring tone (Hz) E
        ring tone (Hz) B
    else if pin P2 is pressed and Direction < 4 then
        ring tone (Hz) G
        ring tone (Hz) B
        ring tone (Hz) D
    else if not pin P2 is pressed then
        stop all sounds
```



SER AVENTURERAS

PAINTBOT

TUTORIAL

PAINTBOT

Introduction

In this workshop you will make an art robot with which you can create unique paint masterpieces. The PaintBot consists of a turning table, rotating at varying speeds depending on a song you selected. On the turning table you can place a sheet of paper or small canvas. By carefully dropping paint drops above the rotating sheet, you get a unique masterpiece created by you and the PaintBot. Each song creates a different pattern of rotation speeds, resulting in a unique creation.

WORKSHOP PROGRESSION

- Creating the PaintBot hardware: Case + Electronics
- Programming the Micro:bit to steer the PaintBot
- Creating the artwork

- Target audience: 17 -25 year
- Participants: 8-12
- Time: 8 hours

Materials

BOX

- Cardboard box (or any other suitable material) with sides of at least 10 cm in height. (The size of the box depends on the size of the paper(s) or canvas you want to put on the turning table)
- Flat cardboard for the turning table
- An egg carton
- Tape
- Scissors
- Fluid glue
- Crafts material: to fixate your sheet or canvas on the turning table (f.e: a toothpick) and to decorate your PaintBot.
- Paint (fluid so you can drop it on your canvas)
- Paper/canvas

ELECTRONIC CIRCUIT

- Small electric DC motor capable of withstanding 9V (you might want to check your old motorized toys to see if you can reuse one of theirs)
- Motor shaft gear
- Motor shaft connection piece
- Micro:bit + battery pack and AAA 1,5 V batteries
- Block battery 9V with clip
- Breadboard
- MOSFET switching under 3 V and able to withstand 3A
- NPN- transistor
- Diode
- Resistor between 1 K and 10 K ohm.
- Breadboard wires
- Longer wires to connect the motor
- 2 crocodile clips

PROGRAMMING THE MICRO:BIT

- PC with internet connection
- Micro:bit with usb cable
- Make Code Editor: https://makecode.microbit.org/#

CREATING THE ARTWORK

- Paint tubes of different colors
- Decorative material
- Your imagination!

MAKE

The paintbot hardware

CREATE THE BOX

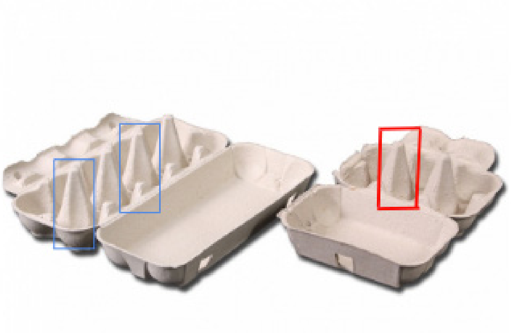
Take a red and black piece of wire of at least 30 cm each and connect it to the motor. (Be sure not to cut the pieces too small as this complicates connecting the motor in the PaintBot with the rest of the electronic circuit. The bigger your box, the longer the wires need to be)

Use the egg carton as motor stand

Remove the lid of the egg carton, we do not need it.

We need one of the tops of the egg carton and its surrounding 4 egg spots. Cut between the 3rd/4th egg spot and the adjacent top.

Cut the tip of the top. It is better to start cutting a small piece and adjusting afterwards rather than cutting too much.

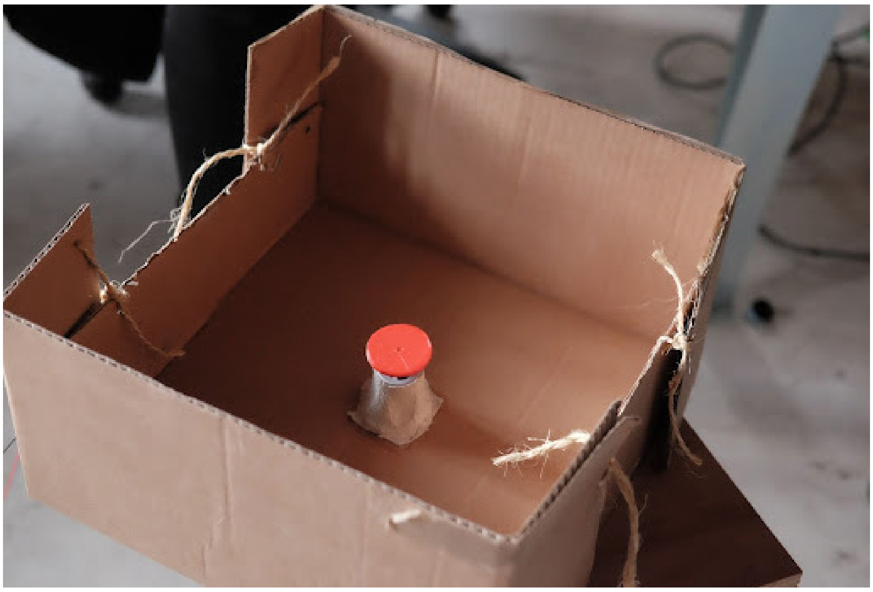


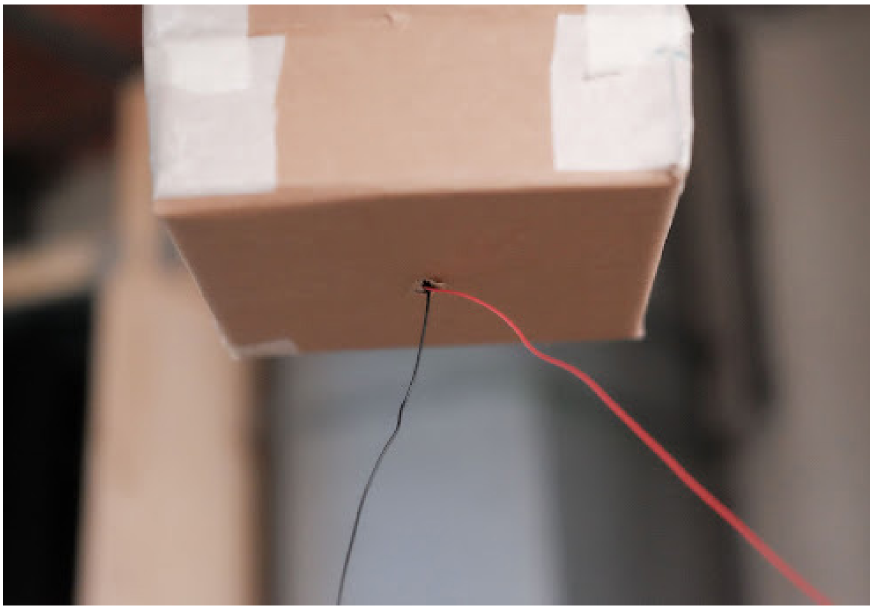
Put the motor from the bottom to the top with the motor shaft upwards and the wires pointing down. The motor should fit tightly in the hole you just cut. You can try to push it carefully or to cut a bit more from the top. It is important that the motor fits firmly and the construction is robust as it forms the base of the turning table. To give extra stability and support you can use tape or you can put toothpicks underneath the motor through the egg carton.

Make a little hole in the middle of the cardboard box. Put the wires of the motor through the hole going from the inside of the box to the outside or bottom of the box. Place the egg carton holder on top of it.

Ensure that the egg carton holder is placed in the middle of the box and firmly attach it to the box. This is very important as it needs to withstand the force of the rotating motor.

Place a carton disc (round or any other shape) on top of the red (3D printed) motor connection disc. Make sure it is smaller than the box so it can freely rotate.





ELECTRONIC CIRCUIT

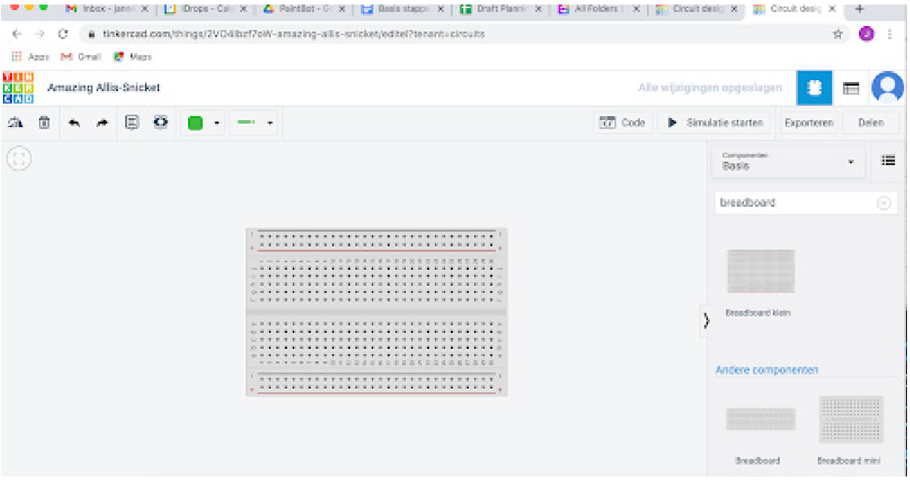
Now it is time to build the electronic circuit. The Micro:bit together with the electronic circuit regulates the flow of current coming from the battery to the motor.

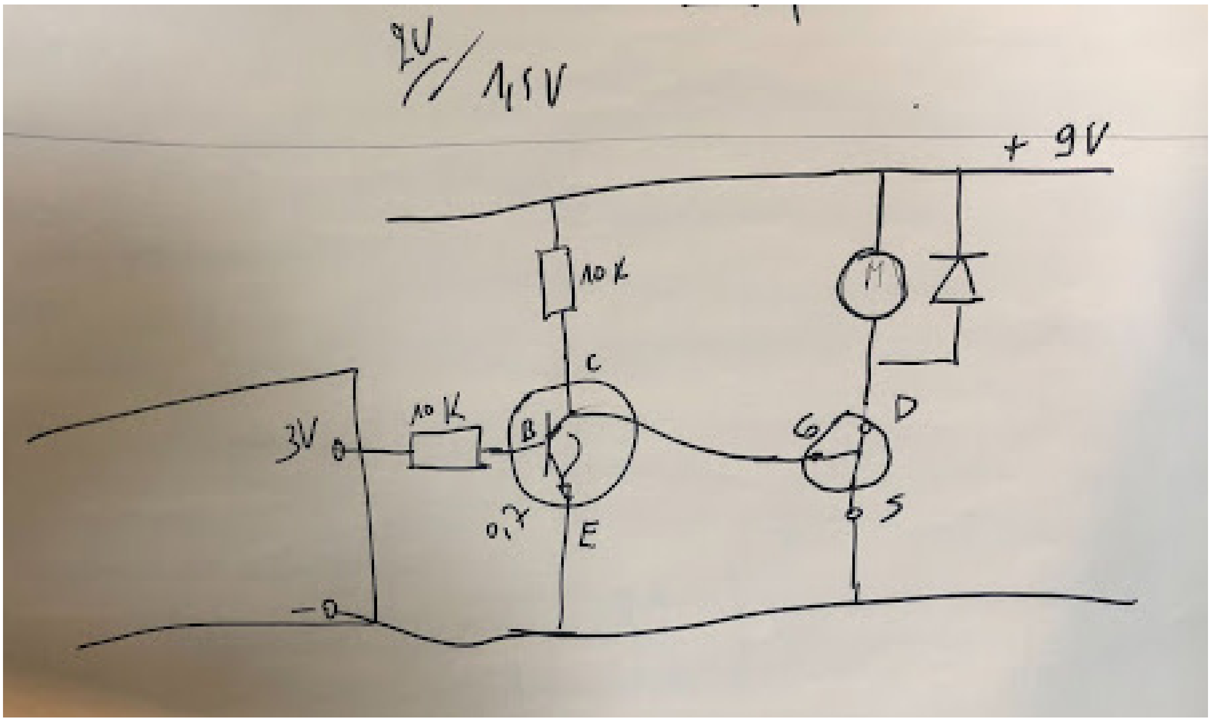
The building of the electronic circuit is very important and needs to be done with great care as we are working with delicate electronic components and batteries. Be sure to follow the instructions carefully and ask for help if necessary. We advise you to build the circuit on the computer first (TINKERCAD/circuit) before building the real one.

Tinkercad offers an easy to use and free tool to simulate your electronic circuit. You can even connect a virtual Micro:bit to test the whole setup. Please do not blow up a real Micro:bit, blow up a virtual one.

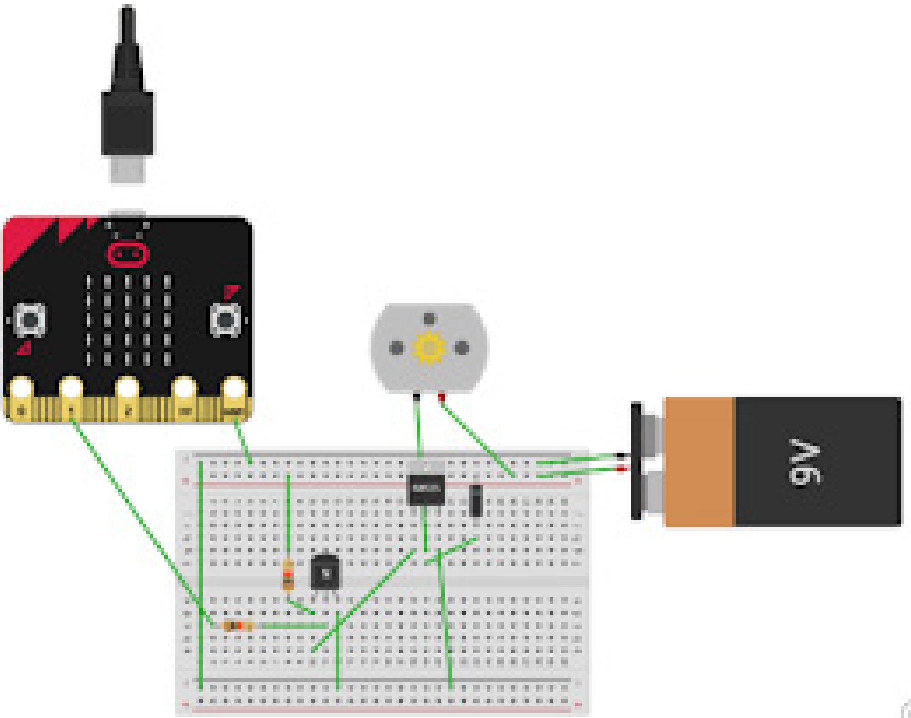
1. Go to tinkercad.com
2. Log in or create an account
3. On the left side on your dashboard, click on 'Circuits'.
4. Create a new circuit.
5. You can rename your project by clicking on the name at the top.
6. On the right side you find the components you can use. The menu above them offers even more components which you can also find using the search bar.
7. To use a component, click on it and click on the workspace in the middle, here you can build your circuit.

When finished building, compare your circuit with the one you made on the computer and the one in this manual before connecting the batteries.



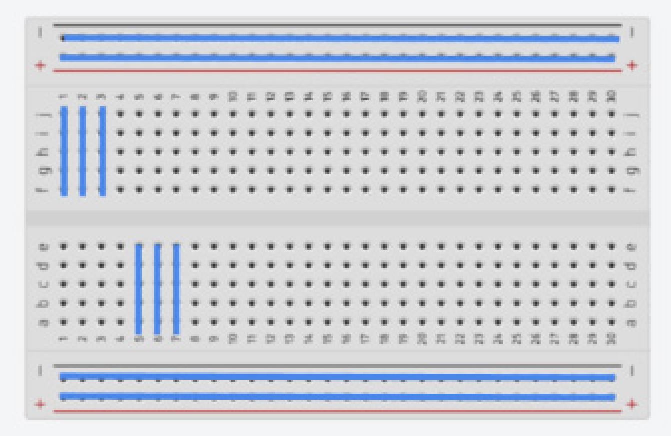


This is a schematic drawing of the electronic circuit. The instructions on the next page will guide you step by step.

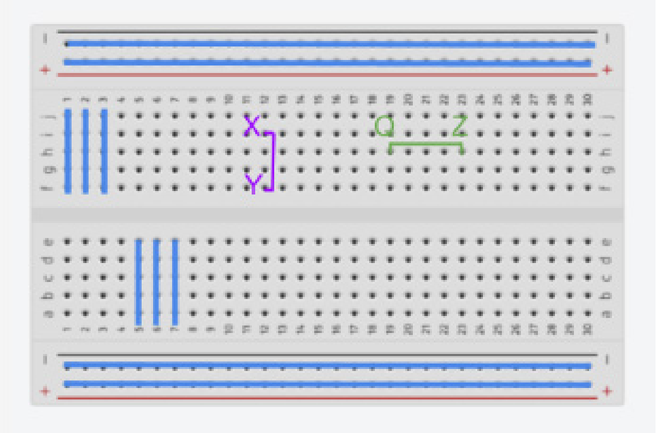


STEP BY STEP

A breadboard is used to easily build a (prototype) circuit without the need to solder the components. Instead, you place the components in the holes of the breadboard to connect them. These connections are essential to let the current flow from one point to another.

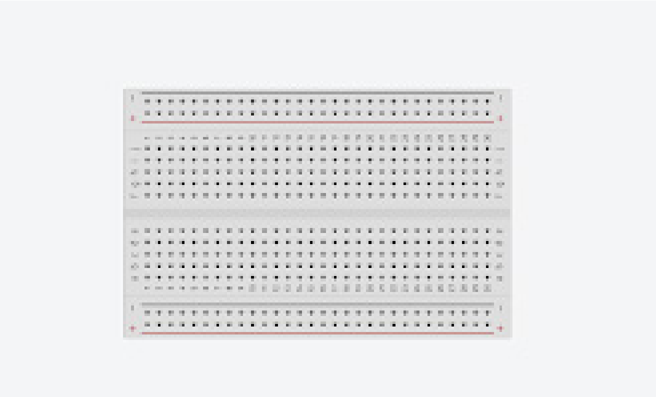


The holes next to the black and red lines are horizontally connected, the holes in the middle vertically, but the upper ones are not connected with the lower ones. In the picture below, the blue lines indicate the connection of the holes.

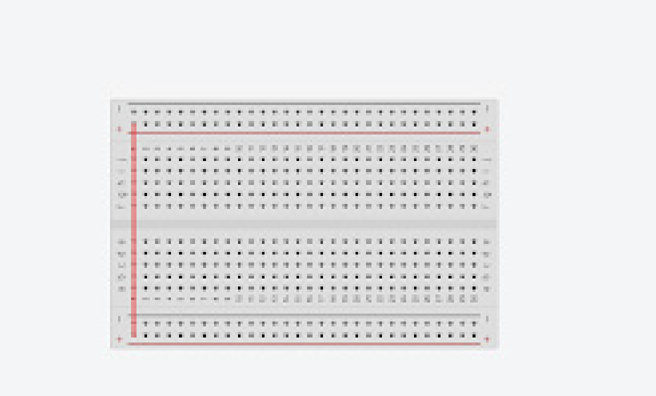


Placing a wire from point X on a blue line to point Y on the same blue line has no effect as the holes were already connected. On the other hand, by placing a wire from point Q on a blue line to point Z on a different blue line we make a connection between those two lines.

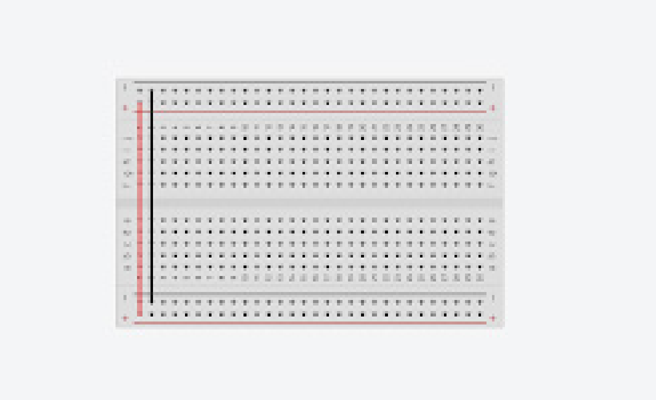
1. Place the breadboard in front of you. Ensure the black and red lines are oriented horizontally.



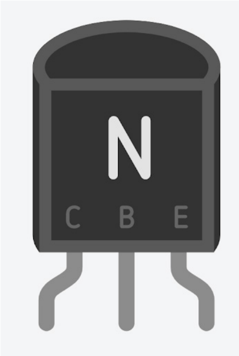
2. For your own ease, connect the plus at the top of your breadboard with the one at the bottom. If possible, use a red wire.



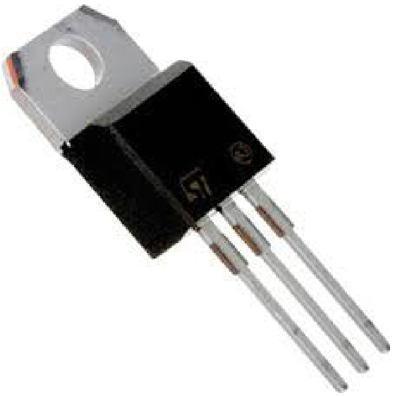
3. Do the same for the min and connect the min from the top with the min from the bottom, if possible use a black wire.



4. We need to use 2 types of TRANSISTORS in our circuit:



Bipolar NPN

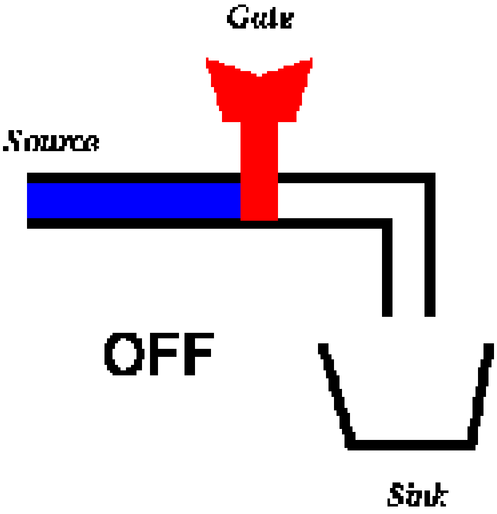


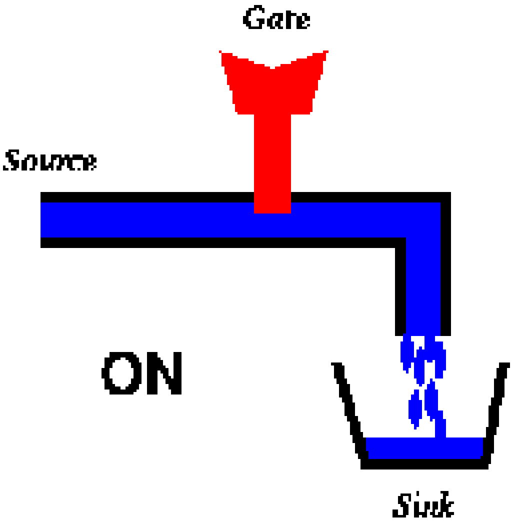
field-effect MOSFET

The transistor regulates how much current flows through the transistor itself. Only when the current flowing through the gate surpasses a certain threshold, the current will flow through the transistor.

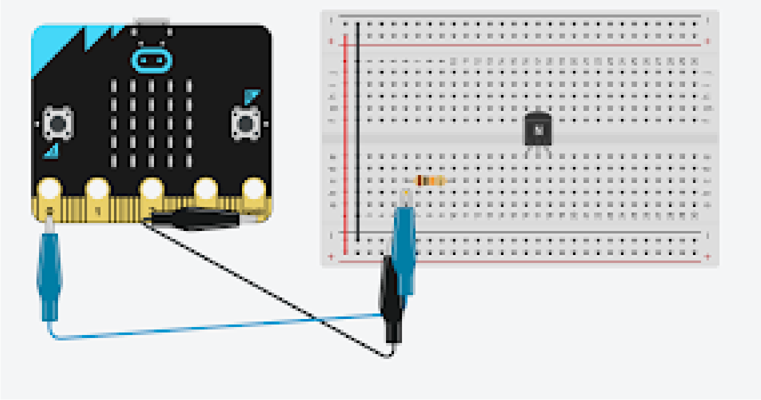
Compare it with a hydraulic valve: just by turning the valve open water will flow through the tube. You can regulate the amount of water just by modifying slightly the opening of the valve. In the case of the transistor a small current at the "gate" (or "base" depending on the kind of transistor) can regulate the larger flow of current passing through the circuit.

Let's start with the NPn transistor



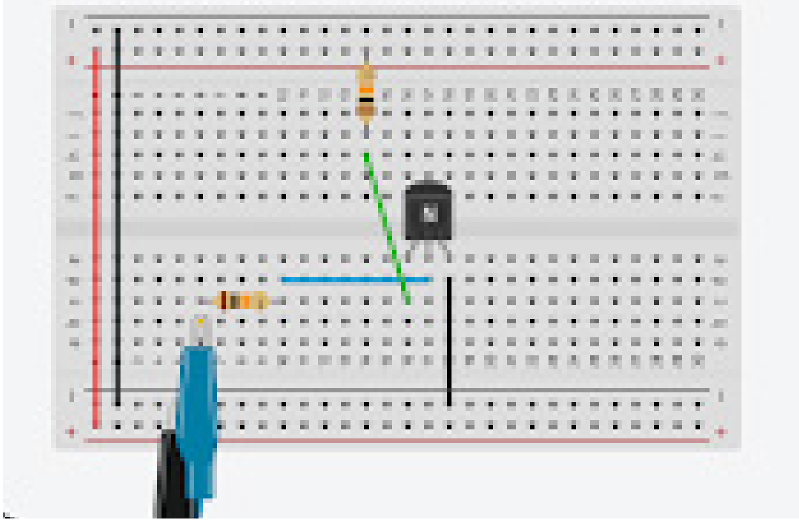


5. Connect the Ground of the Micro:Bit (GND) to the min (-). Use crocodile clips to connect to the Micro:Bit, connect the other side to a cable and insert it in the breadboard. Connect the pin 0 of the Micro:Bit to another line on the breadboard. Now take a resistor (10K) and put one leg in a hole above or below the cable connected to pin 0. Insert the other leg in another hole more to the right.

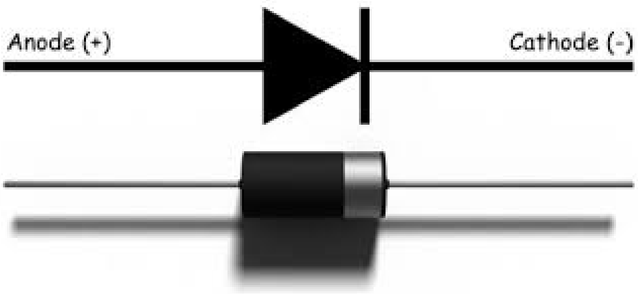


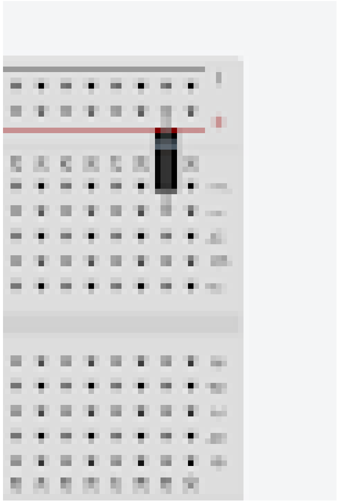
6. Connect the right side of the resistor to the middle leg of the transistor (BASE)

7. Now take another resistor and inert one side on the positive line above and connect the other side with a wire to the left side of the transistor (collector). Connect the right leg of the transistor (emitter) to the negative line at the bottom as shown hereunder.



8. Now let’s insert the DIODE.





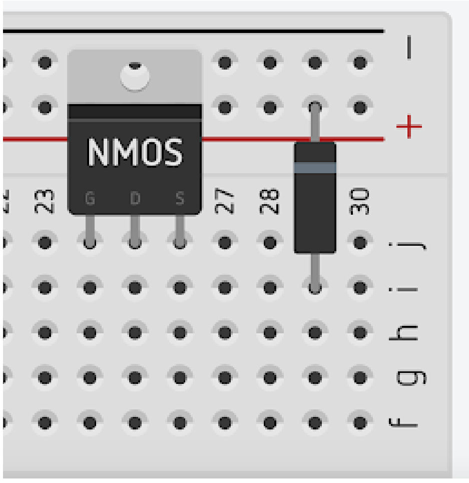
The diode lets current pass only in one direction and we use it to protect the battery from current going back (from left to right in this case).

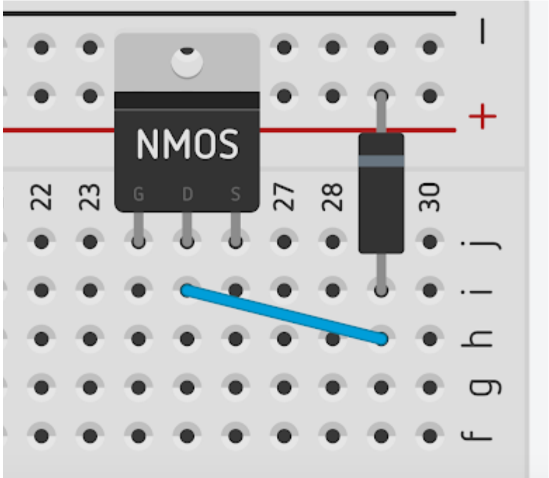
ATTENTION: the direction of the diode is very important! Make sure that the grey ring of the diode is place above towards the + line.

9. Now it is time to place the other transducer: the MOSFET.

The MOSFET is a field effect transducer that can carry higher current than the bipolar NPN transistor. The metal part serves to cool down the transistor that can become very hot if large currents are flowing...RELAX!! This is not our case!

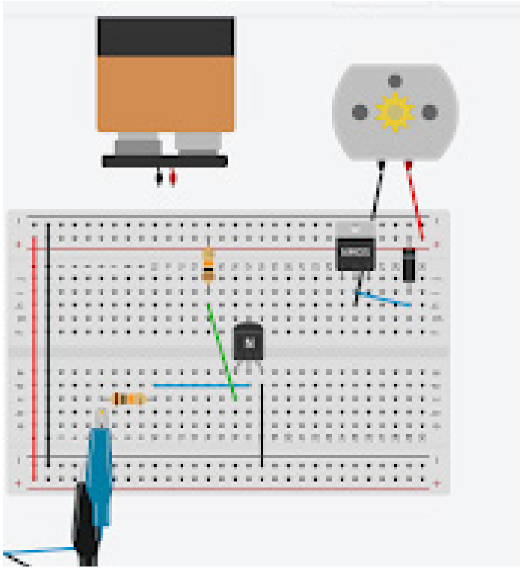
Make sure the flat part of the MOSFET is facing the lines and that you can read the descriptions above the legs (G-gate on the left, D-drain in the middle and S-source on the right)



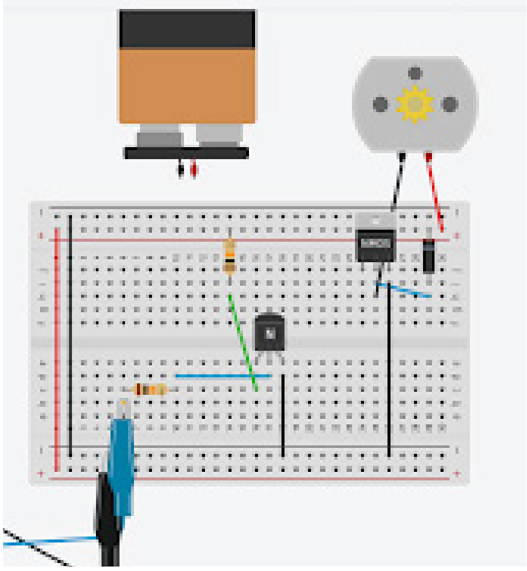


10. Connect the middle leg of the MOSFET (drain) to the bottom of the diode.

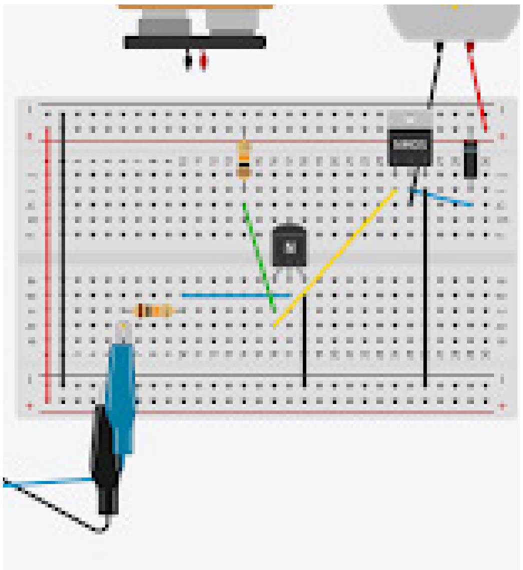
11. Take the motor and insert the cables in the circuit. The red one (positive) in the positive line (+) and the negative (black) under the middle leg of the MOSFET.



12. Connect the source (right leg) to the min line (-) at the bottom (or above)



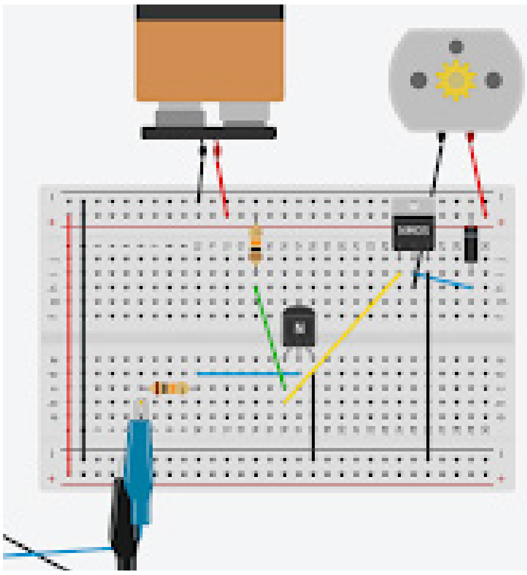
13. Connect the left leg of the MOSFET (gate) to the left leg of the NPN transistor



14. Now it is time to connect the battery.

IMPORTANT: Check if you connected everything correctly, preferably ask someone to double check your work, because now we need to connect the battery and there will be 9V in the circuit. If everything is correct, you can proceed with the next step.

15. Clip the battery clip on the 9V battery and connect the positive side of the battery (red!) to the plus. Connect the negative side of the battery (black!) to the min.



PROGRAM

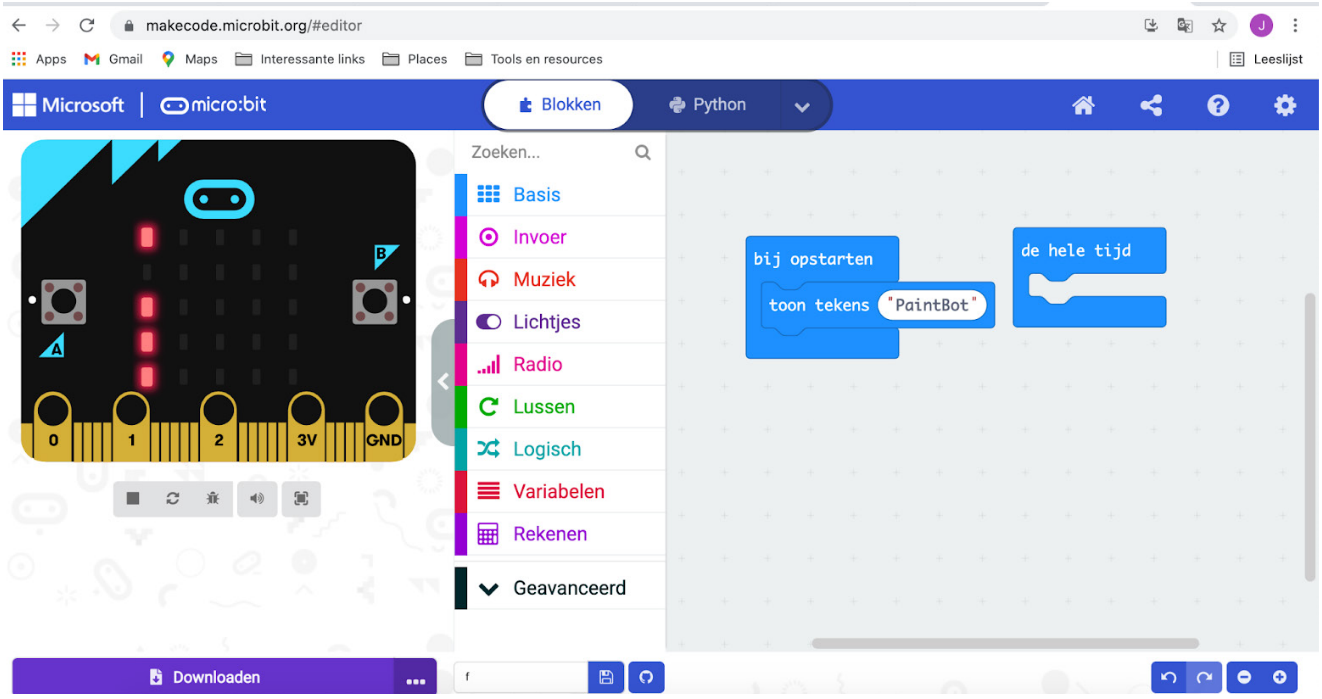
The Micro:bit

The Micro:bit sends electrical signals to the motor telling how fast it should rotate, but first we need to tell the Micro:bit how and when to send these signals. We will do so by programming the Micro:bit using the Make Code Editor. It's an online application specially made to make coding as accessible as possible, using colorful code-blocks.

We assume here that you are already familiar with the Micro:Bit.

Go to the Make Code Editor and create a new project. Name your project.

Your screen should look like this.



bij opstarten
toon tekens "PaintBot"

de hele tijd

You can rename the project at the bottom of your screen (PaintBot or any other funny names), next to the save disk.

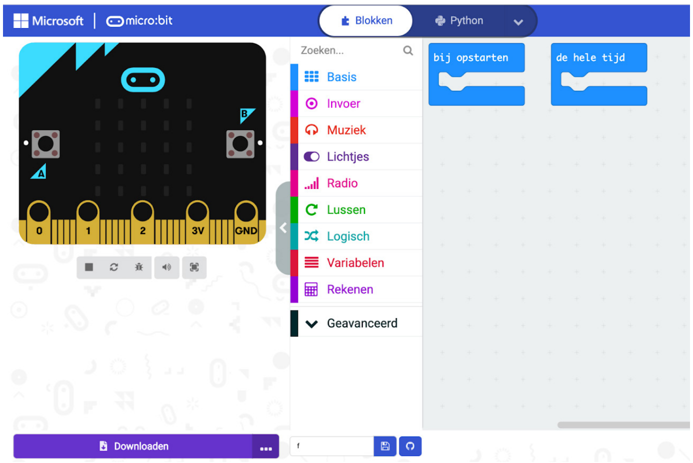
MAIN CODE

This is the algorithm to program the Micro:Bit

- Text “PaintBot” appears on start
- After pressing button A the PaintBot starts to spin
- The Micro:Bit sends electrical signals (one number between 0 and 1023) via the analogue pin P0. This signal determines the rotational speed of the disc.
- Every cifer disappears after one second.

Let’s make it

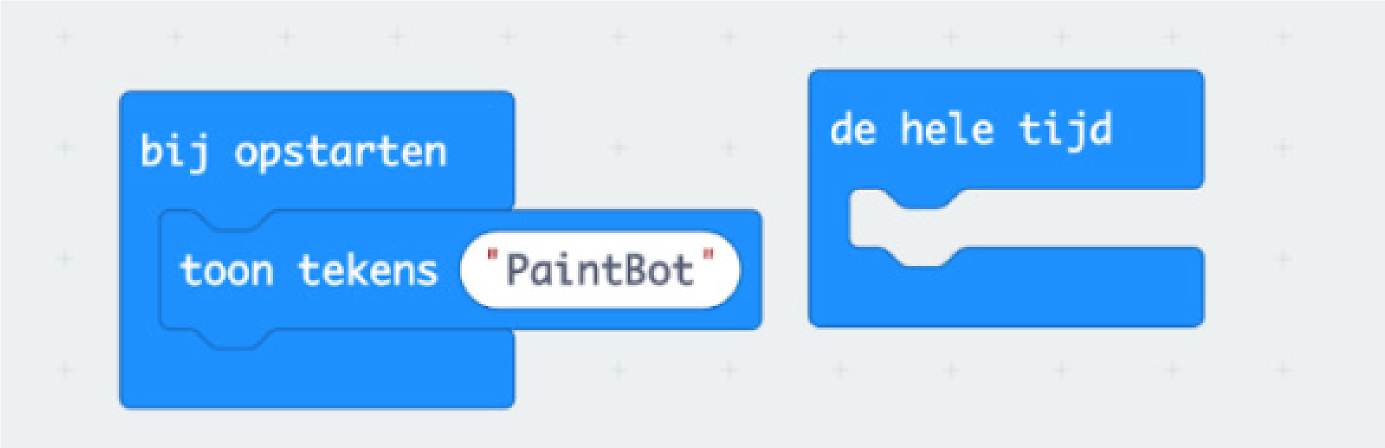
1. Open the editor



In the code editor you see two blue blocks; “on start” and “forever”. Code you place in these blocks will respectively be executed when starting the program (i.e. when starting up the Micro:bit) and during the whole operation of the program.

All the blocks inside the ‘on start block’ will be executed once. The blocks inside the forever block will be looped until the micro:bit is restarted or shut down.

2. Display the word “PaintBot” at start-up.

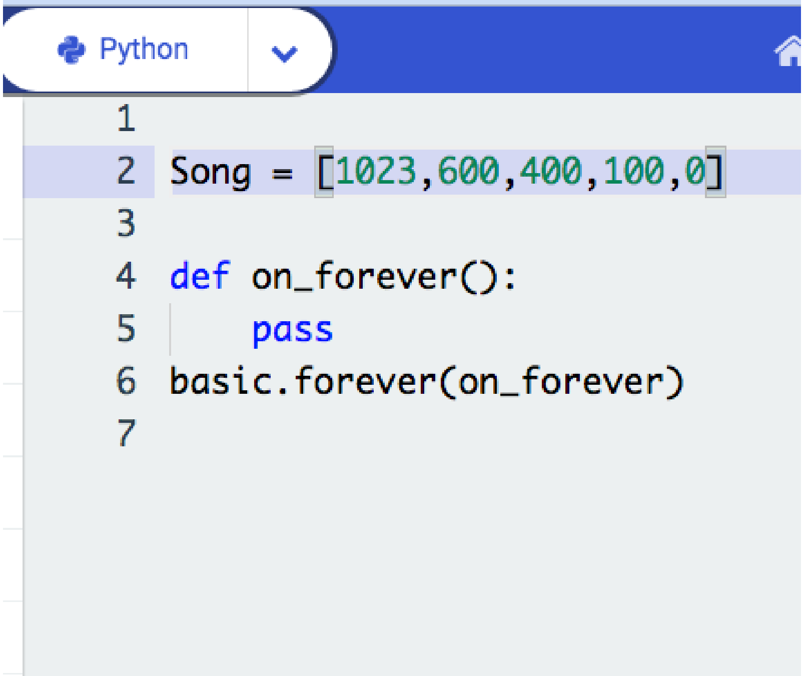


```
bij opstarten
    toon tekens "PaintBot"

de hele tijd
```

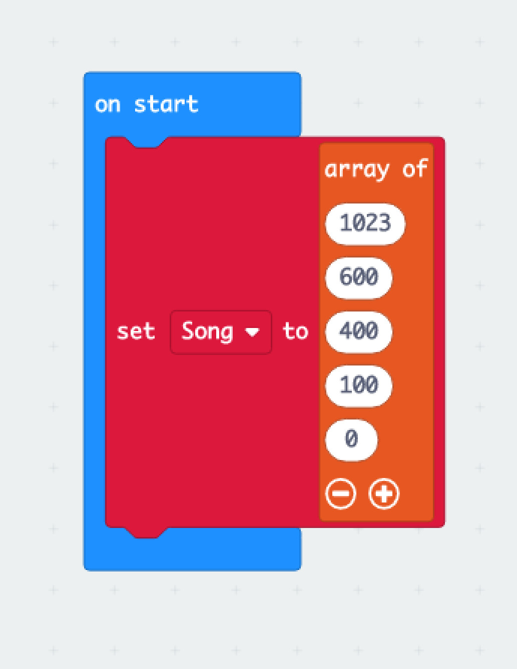
3. At the top of the page switch the editor from Block to Python and type the code below: Song = ….

This will create a new variable called Song. This variable represents the values of rotation of the motor and will depend on the beats of one song you give!

Tip: check the parenthesis and the commas!



```
Song = [1023,600,400,100,0]

def on_forever():
    pass
basic.forever(on_forever)
```



```
on start
  set Song to array of
    1023
    600
    400
    100
    0
```

4. Go back to "blocks editor". You should see a new variable. We will change the values of this variable using the number generator.

5. Now create the rest of the code.

6. Use a block "on start"

7. Use a logic operation: if condition 1 (do this)-- else if condition2 (do this): On this case condition1 is pressing button A condition 2 is pressing button B

8. If button A is pressed go through the values of the variable Song using a loop

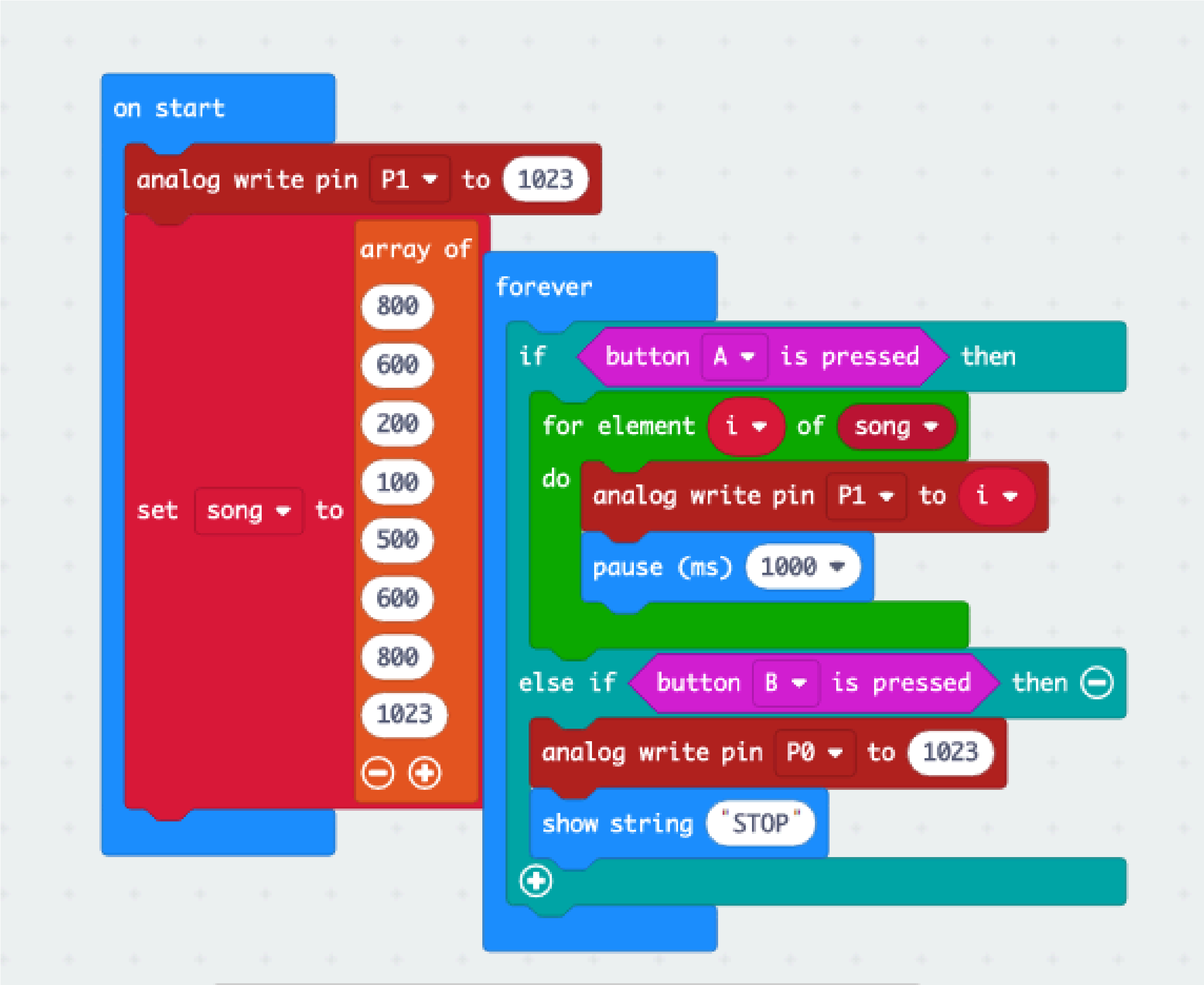
9. For every element of the variable Song use a pause of 1000 ms = 1 second

10. If button B is pressed set pin0 to 1023

IMPORTANT pin0 = 1023: means the motor doesn't move

pin0 = 0 : motor maximum rotations speed

SOLUTION



```
on start
    analog write pin P1 to 1023
    set song to array of
        800
        600
        200
        100
        500
        600
        800
        1023

forever
    if button A is pressed then
        for element i of song
        do
            analog write pin P1 to i
            pause (ms) 1000
    else if button B is pressed then
        analog write pin P0 to 1023
        show string "STOP"
```

The code is now ready but.. what values for the Variable Song shall we choose, i.e what rotational speed for our motor?

The answer to this question is given by the number generator.

NUMBER GENERATOR

We want to use a song as input to determine the rotation speed. Therefore we need to convert the song into numbers which the Micro:bit can understand and use as output for pin 0. We created a special number generator to do this for you.

1. First pick a nice melody or your favorite song and download it as an mp3 file (e.g. https://onlinevideoconverter.pro/nl/youtube-converter-mp3/). Save it as “song.mp3”.
2. Go to the number generator. If you see a lot of code, ignore it and focus on the instructions. You can double click on the textfield of the instructions to hide the code. On the left side, click on the folder to open a panel.
3. To use the number generator you need to be logged in with your Google account. If the students have their own, they can use it, otherwise make a general one for the class.
4. Go to the location on your computer where you saved your mp3 file. Drag and drop the file to the left side of the number generator into the panel. A pop-up will appear saying the document will only be stored temporarily and deleted afterwards. Accept to continue. Now your song.mp3 should upload and appear in the folder. If the upload is finished, you can press the play button on the left of the instructions. Be patient and a list of numbers will appear. Wait until the program finishes. Underneath the program you see the output, copy the list, including the square brackets [..]. The list starts underneath the text “COPY THE LIST BELOW”.
5. Go back to your Micro:Bit Make Code. In the upper middle, you see “blocks” is selected. Switch to “Python”, now you see the python code. Now at the right of “ Song = “ paste the list you copied earlier.
6. Switch back to “Blocks” mode. Now, a list should appear under the show “PaintBot” block, in the starting block. Now the variable “Song” contains the values relative to the BPM of the song.

Now you are all done with setting us the PaintBot now it is your time to create the artwork.

CREATE ARTWORK

With all the components

Let the students play and experiment with the system

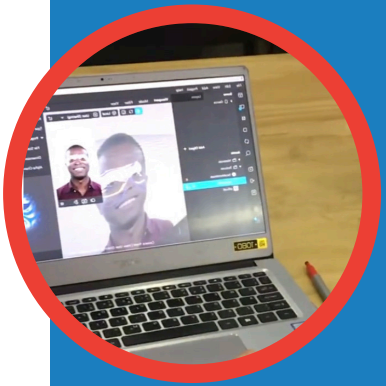
Place the PaintBot on a desk and place some paint in the middle of the disk. You can use different colors. Press the button A of the Micro:Bit to activate the motor. When the Motor spins the paint will be spread on the disk following the BPM of the music. Repeat the process as much as you want, maybe adding different colors.

VERY IMPORTANT: ...HAVE FUN!!

Draaischijf
Put the connection part on top of the motor
Make sure it is firmly attached .
Use glue if needed.

LOVE

SER AVENTURERAS



CODE, DESIGN, PRODUCE!

Learn what it takes to be a digital artist. Focus on robotic arts and digital sound or music making. Our Digital Art Toolkit guides you to combine coding with visual art in step by step instructions, gives you basic theoretical background and ideas for fun and easy exercises.

Find out more about the project through the website at:

www.girlsinstem.eu

Co-funded by the
Erasmus+ Programme
of the European Union

DE CREATIEVE STEM

DIGIJEUNES

FUNDACIÓN intras

Ad Meritum FOUNDATION

IDROPS