

micro:bit CreateAI: Making a game box!

AI exploration phase

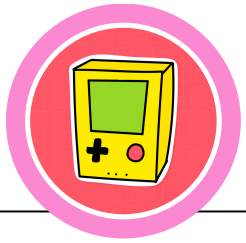
Validating models through Robotics activities

AI and Exploration Topics. What are we exploring?

Participants explore how Machine Learning works by training a micro:bit to recognize motion. They start with programming the Micro:bit and then collect accelerometer data to train a model to distinguish custom movements. They test their model to catch errors and use the trained data to code interactive games on the Micro:bit

Materials Needed and to Create. What must be prepared before the activity?

micro:bit V2
USB cables
Battery packs with 2 AAA batteries
Laptops or tablets



Exploration Steps and Description. What do participants actually do?

Participants begin by creating basic rule-based programs in MakeCode, coding the Micro:bit's display, microphone, and sensors to launch simple games like rock, paper, scissors or digital dice. Next, they connect their Micro:bit to micro:bit Create AI via Bluetooth and record accelerometer samples to train a custom machine learning model on specific gestures. They test their model's live predictions to find out edge cases and correct these errors. As the final step they export their trained AI triggers back into MakeCode, binding each motion to a unique mini-game to construct a functional game box.

Expected Output. What do participants create, produce or gain from the exploration?

Participants collect live sensor data from the accelerometer and use it to train a machine learning model right inside the tool. They make a game box that uses gesture recognition to launch coded mini games like dice or rock, paper, scissors.

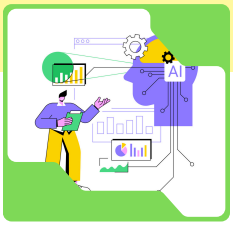
The Hook and the Playfulness. What makes this fun/exciting for young people?

Participants use their own bodies to train a pocket-sized computer. The fun comes from the interesting moment when the Micro:bit recognizes their movements like a custom game controller.

Success indicators. How do we know it worked? What shows participants learned?

When they understand flaws and iterate on their dataset. When they can explain in their own words how data quality controls the accuracy of the AI.

Co-funded by the European Union
CC BY SA



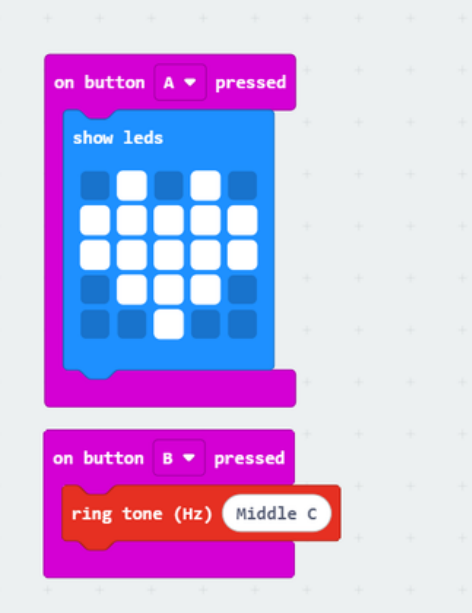
Appendix - Example of material

Appendix 1	Micro:bit CreateAI

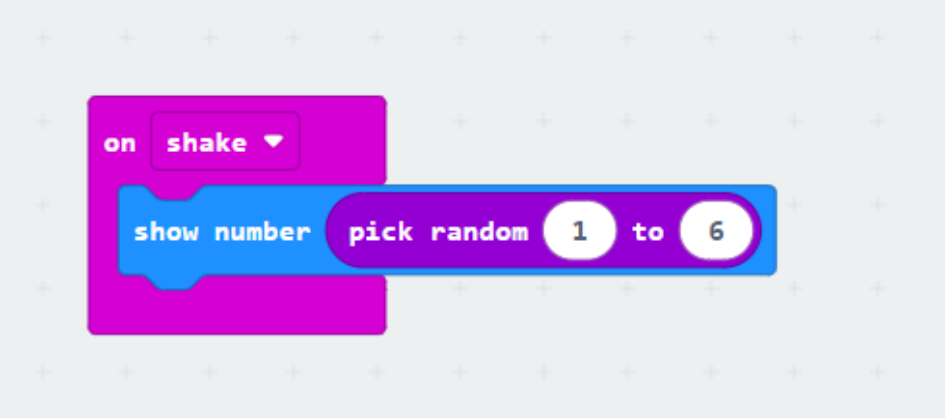
Content of the appendix

Activity 1: Code with Micro:bit
Before starting on the machine learning bit of Micro:bit we start with some small coding activities to get used to the interface.

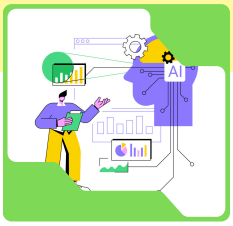
First we explain that you can program input like buttons do create different output. For example via the leds or the buzzer.



Challenge one: Dice



Co-funded by the European Union
CC BY SA

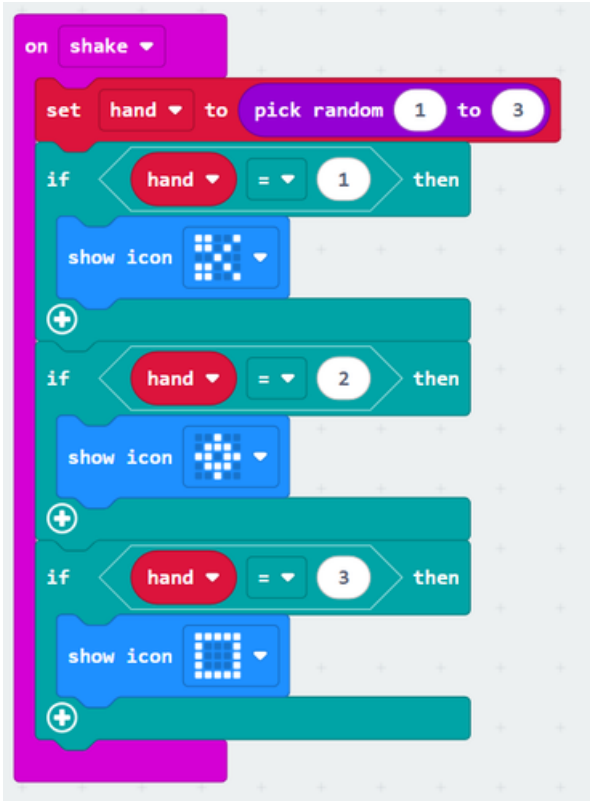


Appendix - Example of material

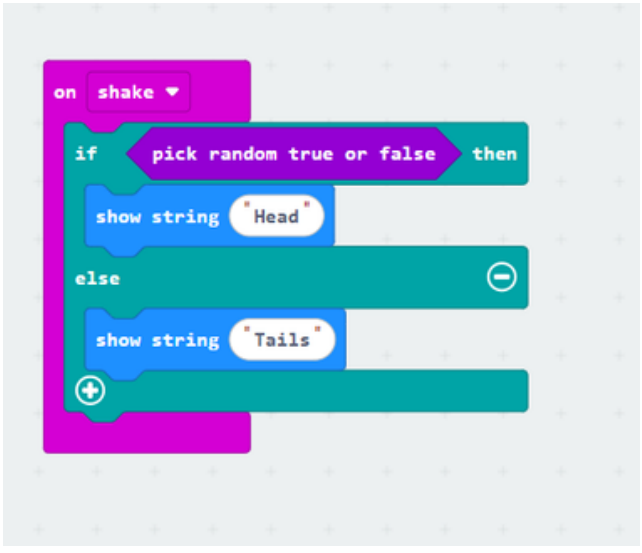
Appendix 1	Micro:bit CreateAI

Content of the appendix

Challenge two: Rock, paper, scissors

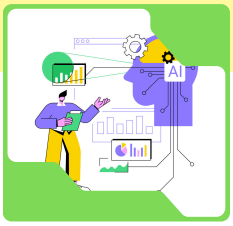


Challenge three: Head or tails



Co-funded by the European Union

CC BY SA



Appendix - Example of material

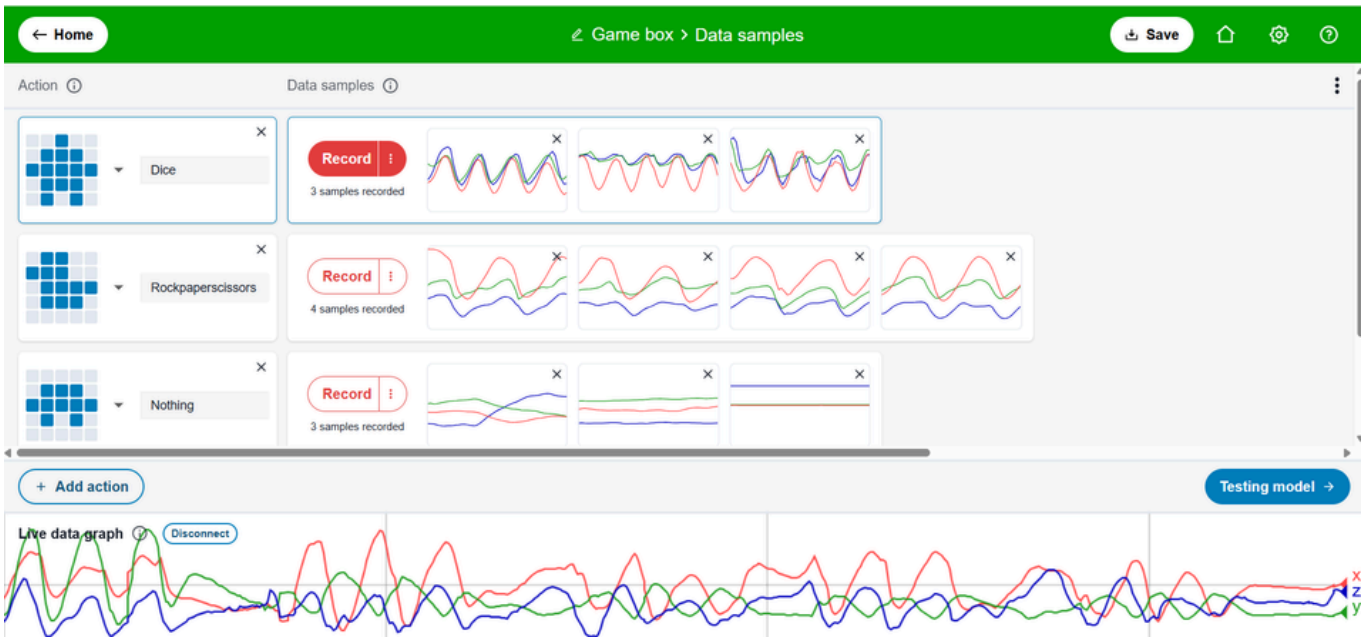
Appendix 1 | Micro:bit Make ai

Content of the appendix

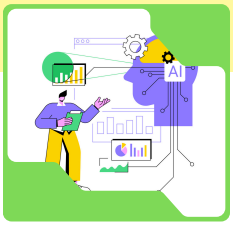
Activity 2: train the AI to recognise movements
Now that we know how to code with a micro:bit we're going to take it a step further and train movement data with micro:bit CreateAI. As you can see in the code on the previous pages, we programmed three games all with the shake input block. With CreateAI we will link those games to a specific gesture. To do this, we will gather motion samples with the accelerometer and label them for each game and train your custom model. Once exported to MakeCode, your own unique gestures will act as the live triggers for every game in your arcade box!

Step one: Start a new project
Open createai.microbit.org in your browser, start a new project and pair your Micro:bit V2. First with a cable and follow the steps on the website to connect wirelessly so it can stream live motion data. Make sure to turn your bluetooth on

Step two: add your movements and labels
Create action categories for each gesture you want to detect. For example: fast shaking for the dice, turning the device to flip a coin and sideways shaking for rock, paper, scissors.
Important! Be sure to add a Still category for when the device is at rest, otherwise the program will think standing still is closest to a graph with as little movement as possible.



Co-funded by the European Union
CC BY SA



Appendix - Example of material

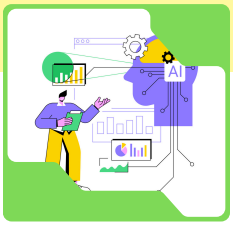
Appendix 1	Micro:bit Make ai

Content of the appendix

Step 3: Train and test your data
Click “train model” and test your movements live on screen to check the accuracy of the movements. Record more samples or look if a movement is very similar to another movements if it misinterprets a gesture.



Co-funded by the European Union



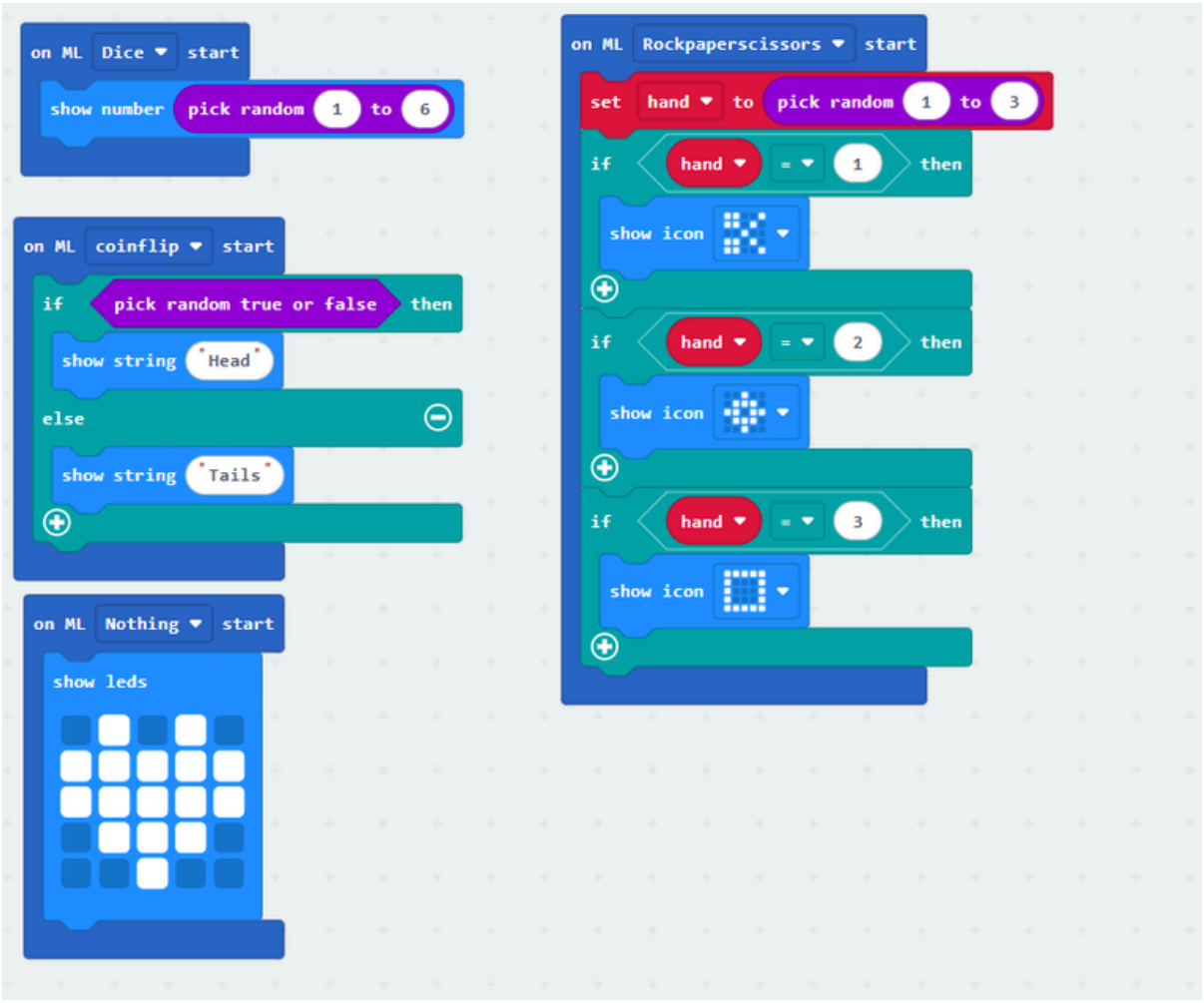
Appendix - Example of material

Appendix 1	Micro:bit Make ai

Content of the appendix

Step 4: Export to MakeCode

Click Edit in MakeCode to export your custom machine learning blocks, then attach them to the code logic we have made in activity 1to trigger your mini-games



CC BY SA