

## Activity Sheet:

**Title:** Teach the Micro:bit to Recognize Shaking

**Objective:** In this experiment, you will teach the BBC Micro:bit to recognize a specific gesture (shaking) and program it to respond when it detects the gesture. The objective is to understand the concept of machine learning and practice gesture recognition.

## Materials Needed:

- BBC Micro:bit
- USB cables for Micro:bit connection
- Computers with MakeCode coding environment
- Controlled environment free from excessive vibrations or disturbances

## Instructions:

### Step 1: Introduction (15 minutes):

- Discuss the concept of machine learning and its applications.
- Explain the experiment's objective.

### Step 2: Setup and Training (15 minutes):

- Provide Micro:bits, USB cables, and access to the MakeCode coding environment.
- In this version of the experiment, students will train their Micro:bits to recognize a specific gesture (shaking).

### Step 3: Programming the Response (20 minutes):

- Instruct students to program their Micro:bits to respond with a specific message when the gesture is detected.

### Step 4: Testing (20 minutes):

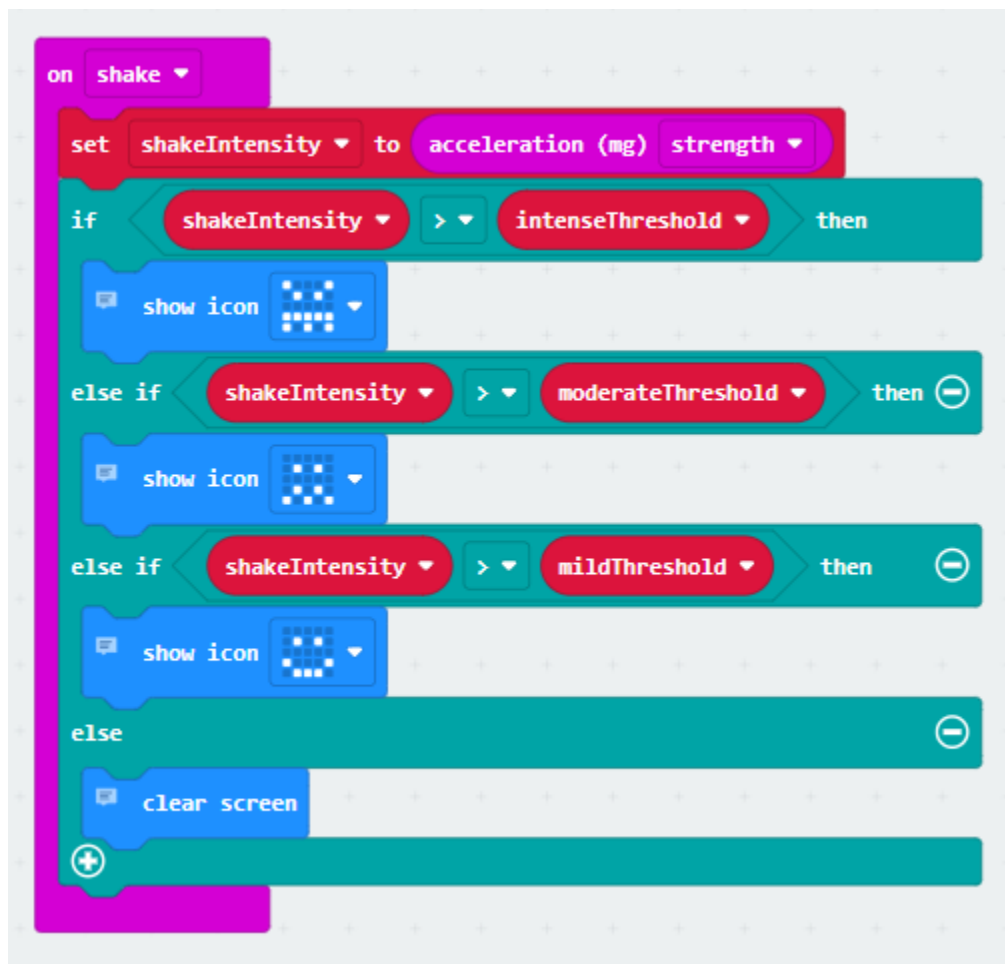
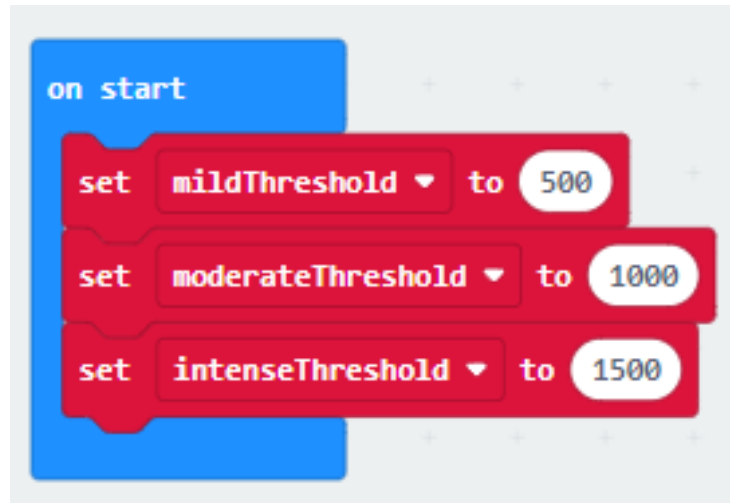
- Students take turns testing their Micro:bits by performing the chosen gesture.
- Observe the Micro:bit's response.

### Step 5: Discussion and Analysis (15 minutes):

- Conduct a discussion about the experiment's results and the concept of machine learning.
- Reflect on the importance of data and practice.

## Code:

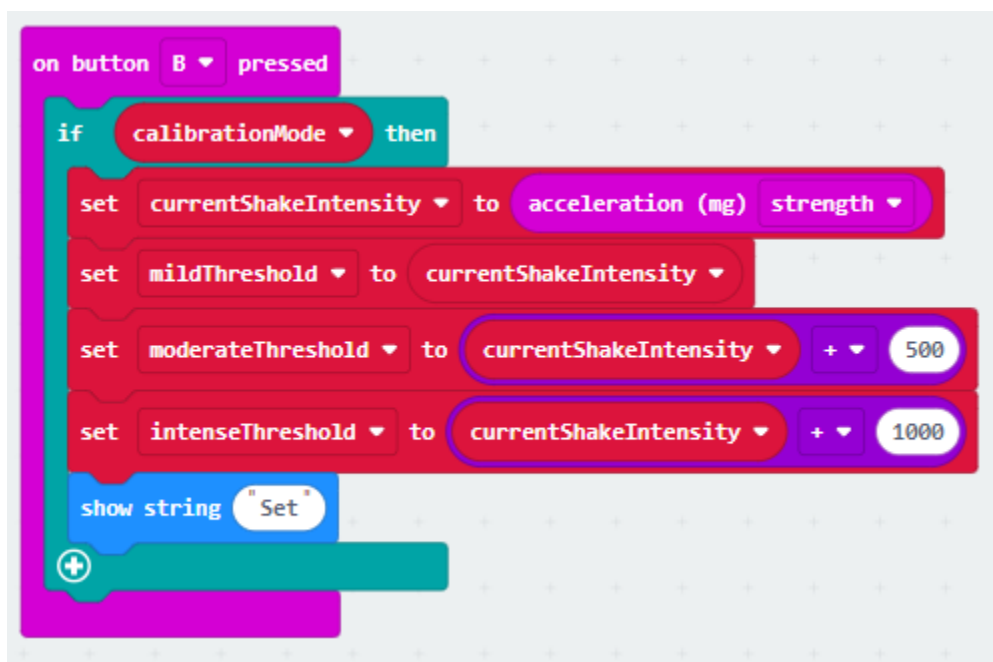
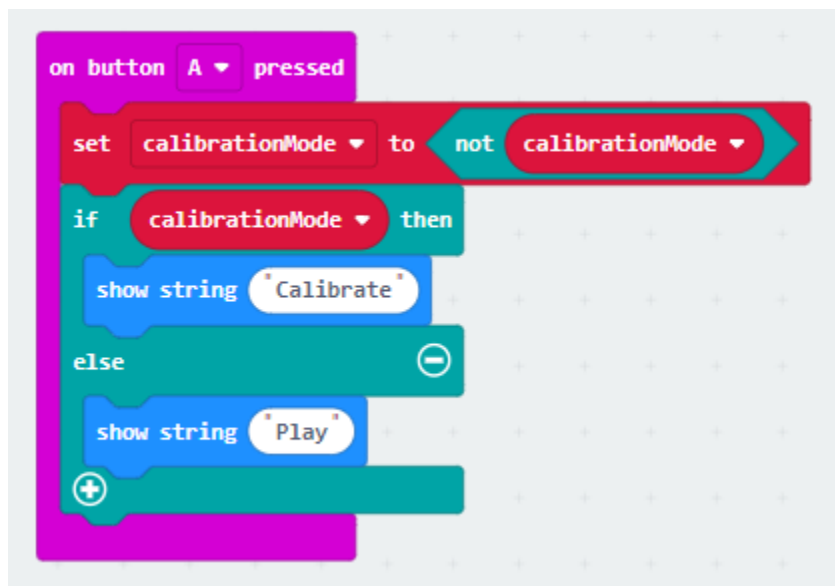
To program the LED grid on the BBC Micro:bit to display information based on the data that was collected (such as the recognition of a specific gesture), you can use the MakeCode environment. Here's how you can do it:

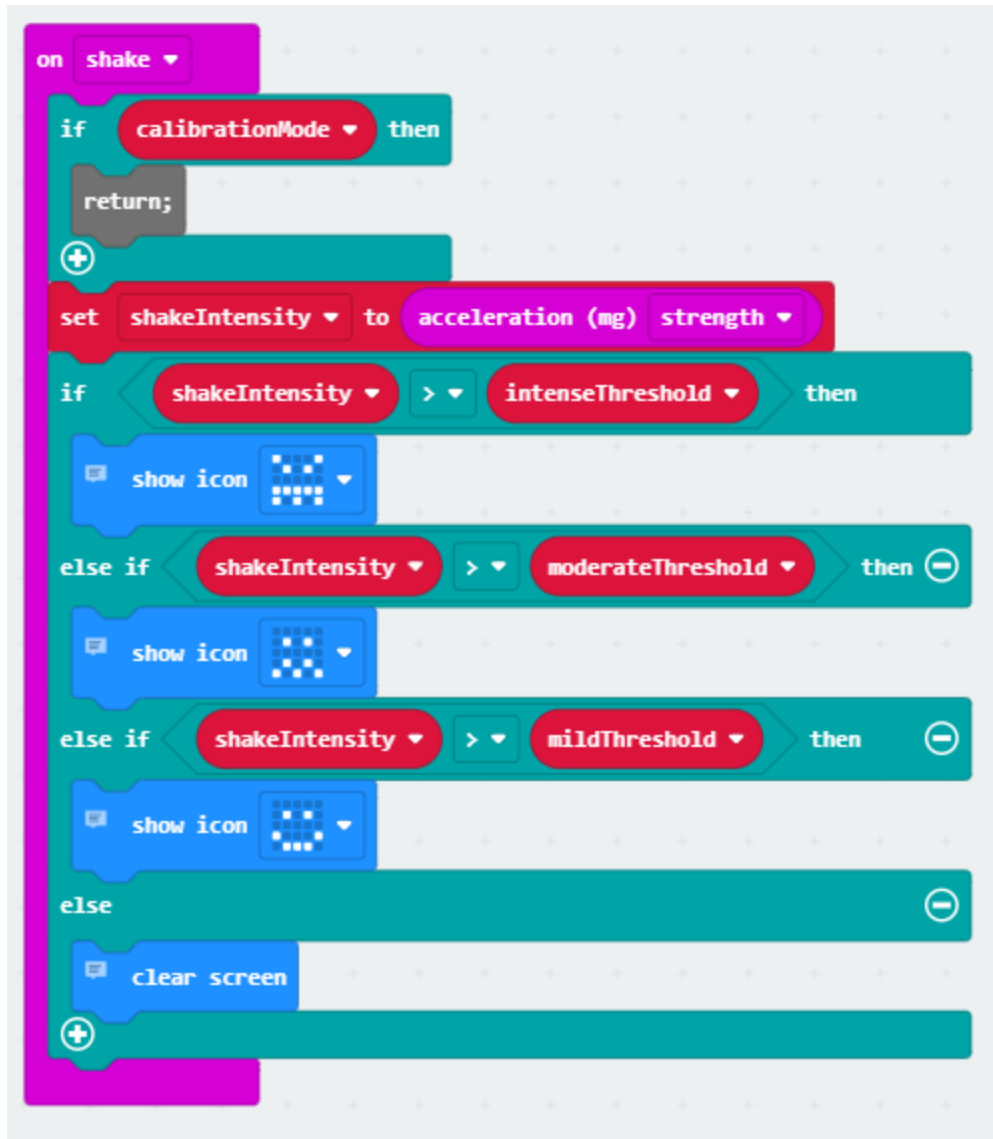


You can use LED grid patterns to create simple images, messages, or icons to display on the Micro:bit when the gesture is recognized. Test your code by shaking the Micro:bit to see how it responds. If you've programmed it correctly, it should display the information or pattern you've specified on the LED grid. You can further experiment with different patterns or messages in response to the recognized

gesture by modifying the code. For example, you can change the displayed message or create more complex LED grid patterns.

You can also expand this project by implementing a calibration feature where the Micro:bit adjusts its sensitivity based on the player's typical shaking intensity and by introducing a scoring system where players try to achieve specific shaking patterns or intensities. You can try the following code:





Explanation of the code:

#### Variables:

- 'mildThreshold', 'moderateThreshold', 'intenseThreshold': Variables to store the threshold values for categorizing shake intensities as mild, moderate, or intense.
- 'calibrationMode': A boolean flag to indicate whether the game is in calibration mode ('true') or normal play mode ('false').

#### Shake Detection Handler

This event handler is triggered when the Micro:bit detects a shaking motion.

Inside the handler:

- If 'calibrationMode' is 'true', the function exits without doing anything, allowing calibration settings to be adjusted without interference from the shake detection logic.
- If 'calibrationMode' is false, the function measures the current shake intensity using the accelerometer (set shakeIntensity to acceleration (mg) strength).
- The shake intensity is then compared against the set thresholds to determine its category (mild, moderate, intense).
- Depending on the category, an appropriate icon is displayed on the LED screen (happy for mild, confused for moderate, angry for intense).
- If the shake intensity is below the mild threshold, the screen is cleared.

#### Button A Event Handler:

This event handler is triggered when button A is pressed.

Inside the handler:

- Toggles the 'calibrationMode' between 'true' (on) and 'false' (off).
- Displays "Calibrate" when entering calibration mode and "Play" when exiting calibration mode.

#### Button B Event Handler:

This event handler is triggered when button B is pressed.

Inside the handler:

- Active only in calibration mode ('if (calibrationMode)').
- When in calibration mode, pressing button B sets new thresholds based on the current shake intensity:
  - 'mildThreshold' is set to the current shake intensity.
  - 'moderateThreshold' and 'intenseThreshold' are set to progressively higher values (500 and 1000 units above mildThreshold, respectively).

In normal play mode, the Micro:bit detects and responds to shakes by categorizing their intensity and displaying corresponding icons. This demonstrates how AI systems can use sensory data (in this case, motion data) to make decisions.

In calibration mode, players can adjust the sensitivity of the shake detection by setting new threshold values. This feature illustrates the concept of calibration in AI systems, where parameters are adjusted to suit different conditions or preferences.