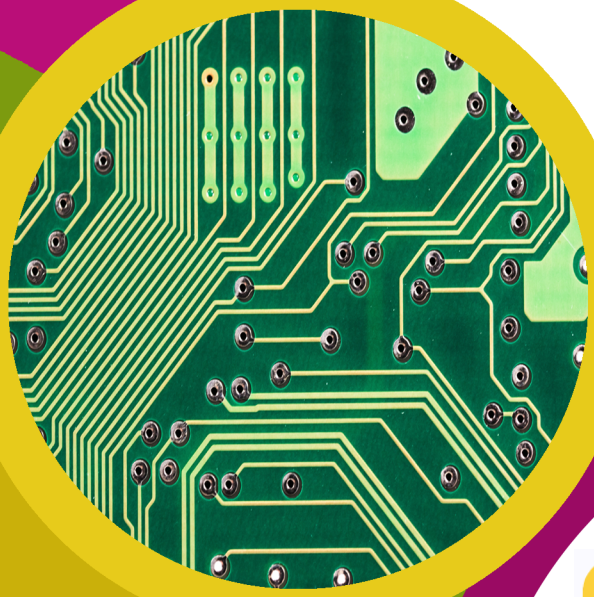
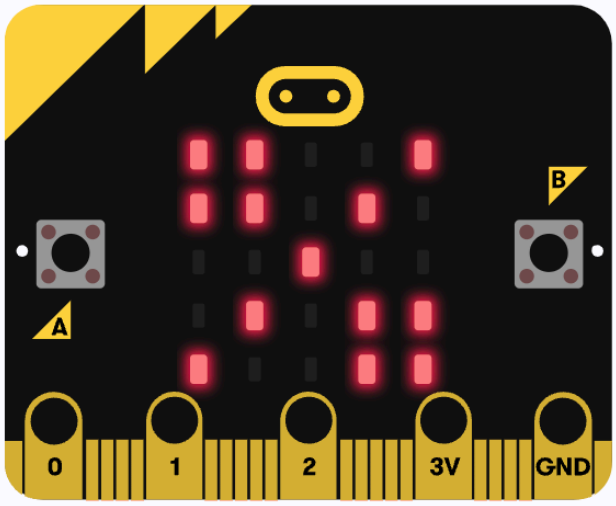
technocamps

Micro:bit Math Game

Teacher Guidance





UNDEB EWROPEAIDD EUROPEAN UNION
Llywodraeth Cymru Welsh Government
Cronfa Gymdeithasol Ewrop
European Social Fund

Prifysgol Abertawe Swansea University
CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD
PRIFYSGOL BANGOR UNIVERSITY
Cardiff Metropolitan University
Prifysgol Metropolitan Caerdydd
i.t.wales
PRIFYSGOL ABERYSTWYTH UNIVERSITY
PRIFYSGOL glyndŵr Wrecsam
Wrexham glyndŵr UNIVERSITY
University of South Wales Prifysgol De Cymru

Links to Science and Technology AoLE

Computation:

(PS2) I can create simple algorithms and am beginning to explain errors.
(PS2) I can follow algorithms to determine their purpose and predict outcomes.
(PS2) I can explore how different component parts work together.
(PS2) I am beginning to explain the importance of accurate and reliable data to ensure a desired outcome.
(PS3) I can use conditional statements to add control and decision-making to algorithms.

Links to Other AoLEs

Mathematics and Numeracy:

(PS2) I have explored additive relationships, using a range of representations. I can add and subtract whole numbers, using a variety of written and mental methods.
(PS3) I can use the four arithmetic operations confidently, efficiently and accurately with integers and decimals, and I can combine these using distributive, associative and commutative laws where appropriate.
(PS3) I can fluently recall multiplication facts up to at least 10 x 10 and use these to derive related facts.

Languages, Literacy, and Communication:

(PS3) I can listen to and understand information about a variety of topics,

The Four Purposes and Cross-Curricular Skills

This resource provides **Critical Thinking and Problem-Solving** opportunities. Students are required to follow and design an algorithm using block-based programming. They are able to analyse errors in the code, identify solutions, and deduce the next steps in the code.

Learners also use **Creativity and Innovation**. They are encouraged to discuss and implement strategies to improve their program. They can use their own strategies to code the micro:bits and play Salute, a math game.

The **Interacting and Collaborating** and **Data and Computational Thinking** sections of the **DCF** apply to this resource. Students will learn to code using loops and events to create an algorithm that takes user input to display an output as well as uses random number generation as a game event. Learners will work together, solve puzzles and play math games.

Why Is Learning This Important?

This resource provides learners with the opportunity to create simple algorithms with a demonstrable application, using a block-based programming language. It introduces concepts such as conditionals, loops, and event-based programming which are critical to most common programming languages. The resource also allows students to practice their math skills and allows for collaborative and interactive activities to showcase this. This can be expanded to introduce students to text-based programming such as Python.

Suggested Approaches Key

In this suggested approach we use the following colours to differentiate the types of activities:

- **Yellow - Explain.** Teachers should explain the slide/example to the class.
- **Green - Discuss.** Teachers should start an open discussion with the class to get them to feedback some answers/ideas.
- **Purple - Activity.** Students are expected to complete an activity whether it be in their workbooks or on the computer, followed by a discussion of their solutions.
- **Green - Introduction/Conclusion.** The introduction/conclusion is also colour coded green. Teachers should hand out materials in the introduction and conclude the session and collect materials at the end.

Introduction

Begin with introductions, and a brief explanation of the Technocamps programme, before handing out any resources required by learners and any additional aids for learners with additional learning needs.

Explain: Topics Covered Today

We will be learning how to use micro:bits and block-based programming to play a fun math game called Salute.

Discuss: Micro:bit

Provide each student with a micro:bit and ask them what they know about it. Have they used them before? Can they tell you what some of the components are (e.g. buttons, LED lights, USB connector)?

Explain: What is micro:bit?

The micro:bit is a very small computer that is used to teach how hardware and software work together.
It has several components: 25 Led lights that can be used to display images, sensors that can detect light/temperature/movement, buttons, and radio and bluetooth antenna.
We can program the micro:bit to take input, display output, process information, communicate with other micro:bits and many more things.

Explain: What is programming?

Programming is telling a computer what to do using a set of ordered instructions. The set of ordered instructions is called an **algorithm**. The language used to tell the computer what to do is called a **programming language**.
Introduce the students to the MakeCode editor and explain how to connect

Display Output

Explain: Downloading Programs

Each time you make changes to your program you need to click on the download button before the micro:bit can run the program.

Activity: Numbers!

Program the micro:bit to display a number of your choice. Drag the **show string** block into the **forever** loop and choose the icon from the dropdown menu.

Explain: LED lights

The **show icon** blocks allow us to display default images on the micro:bit. Using the **show leds** block, we can customise the image that is displayed.

Activity: Custom Icons

Ask the students to create their own icons using the **show leds** block. Clicking on each individual square turns that light on or off.

Explain: Loops

Loops can be used to repeat commands in a program without typing out each action every time. They can be repeated forever, for a certain number of times, or for a given condition.
In MakeCode, these are found in the **Loops** section.
For a loop to have a purpose, an action command needs to be placed inside it.
All MakeCode programs start with a default **forever** loop. This loop will run a set of commands until the micro:bit is unplugged or reset. You can only have one **forever** loop in a micro:bit code.

Activity: Changing Numbers

Ask the students to create a program that displays two alternating numbers of their choice. They can do this in the default **forever** loop. Suggest that they add a **pause** command between the two icons.

Discuss: Forever vs. On Start

Explain the difference between the **on start** block and the **forever** loop. Ask the students what they think would happen if they put the code from the 'Changing Numbers' activity in the **on start** block instead of the **forever** block.
Answer: The code would run once and then stop without repeating.

Activity: Counting with Forever Loops

Hint to students that they can chain together many **show string** and **pause** blocks inside a loop, which allows them to count using micro:bits. Remind them to click **Download** to update the code on their micro:bits.

Explain: Events

Each micro:bit has two buttons: A on the left and B on the right.
These buttons allow us to choose which action to take without reprogramming the micro:bit each time.
For example, we can show one number when we press button A and another number when we press button B.
These commands are found in the **input** section.

Activity: Events

Create a program that displays one number when button A is pressed and a different number when button B is pressed.

Explain: Salute!

Explain the rules of the math game Salute to students. Salute is a fun math game for 3 players.
First, two players hold up a number to their foreheads— they can see the other person's number, but not their own! A third player looks at the two numbers and decides either to multiply them or add them. They say the result out loud. Based on that number and the other person's number, the two other players try to guess their own number. The fastest player to guess their own number wins the round! (An

Explain: Micro:bit Movement and Random

Micro:bits have a special feature that allows them to detect when they are being moved or rotated. This lets us put certain code into an event block that will run only when the micro:bit is rotated or moved in a specific way.

Micro:bits also have a special feature that allows them to display random numbers. Combining this with the movement feature, we can program the micro:bits to play Salute.

Activity: Salute!

Ask the students to get into groups of 3. Two players will act as the guessers, and one player will choose to add or multiply the numbers of the other two players. Later on, the third player will also be in charge of the scoreboard. Make sure that the students switch up roles!

Programming Salute!

Explain: Programming Salute

Demonstrate to students how to use the **on logo up** event block to program the micro:bits to run specific code when the micro:bit is rotated. Then, use the **show string** and **pick random** block to program the micro:bits to display a random number when the micro:bit is held up to the forehead.

Activity: Salute!

Ask the students to recreate the code on the slide. Now, their micro:bits are programmed to play Salute! Have the students spend some time playing.

Discuss: Sending Messages

What do they think the next step will be?

Hint: How will the other micro:bit know what to do with the received message? Tell the students to look in the **radio** section for possible commands to use.

Variables and a Counter

Explain: Variables

Explain to students that variables allow us to store values for later use. A variable is like a box: you can put anything you want inside of the box, and take it out later to use. In the case of Salute, we can use variables so the micro:bit remembers the scores of each player. This will allow us to program a scoreboard for the micro:bit.

Activity: Making a Counter

Using the default variable **a**, it is possible to create a counter based on events. By starting with the **on button A pressed** block, students should add a **change a by 1** block. This will increase the variable a by one every time button A (the left button) is pressed. Remember to add a **show string** block so we can see the new value of a.

Hint to students that they can use the **on button B pressed** block to decrease the value of a. Remind them that a negative value for the **change a by** block will cause it to decrease by that amount instead of increase.

Discuss: Scoreboard

Discuss with students how they could use two variables to program a scoreboard. Hint that a scoreboard is, essentially, two counters operating at the same time. Remind them that they could use one button to increment Player A's score and the other button to increment Player B's score.

Explain: Creating Variables

Instruct students to click on the **Variables** section, then on the **Make a Variable** button. They should create two variables: one called Player A Score and the other called Player B Score.

Explain: Creating a Scoreboard

First, students should create three blocks: one **on start** block, and one **on pressed** block for each button. In the **on start** block, they should use **set** blocks from the Variables section to set Player A Score and Player B Score to 0.

Then, in the **on button A pressed** block, students should add a **change Player A Score by 1** block and a **show string Player A Score** block. This will make it so pressing button A will add 1 to player A's score and show the new score. Hint that they should do the same for button B.

Discuss: Resetting the Screen

After the buttons are clicked, the score shown simply stays on the screen until the button is pressed again. We should clear the screen after a score is displayed, by using a **wait** block and a **clear screen** block. Hint to students the process they should use, then allow them to attempt it themselves.

Explain: Checking Scores

The micro:bits also have an event for when both Button A and Button B are pressed. This event allows us to check both scores when both buttons are pressed. Instruct students to add an **on button pressed A+B** block and use a **show string** block to show player A's score. Then, after a 1 second delay using a **pause** block, do the same for player B's score. Insert another **pause** block delay, then clear the screen with a **clear screen** block.

Activity: Playing Salute!

Students can now play Salute again with their new scoreboards. Two out of three players should program their micro:bits with the game code, while the third player should use the scoreboard code. This player is the one who adds or multiplies the numbers together and is in charge of the scoreboard.

Differentiating for Learners

- More advanced students might program features like detecting which player wins the game. Hint to these students that they should use **if** blocks to decide which student has more points.

- Some learners may need more guidance in assembling the code than others. Since this workshop is aimed at ages 9-11, there might be several learners with little coding experience so will require step-by-step instructions in assembling the blocks. Other learners can use their previous coding experience to write the algorithm without any code provision.

- The micro:bit website and editor allow for navigation using accessibility features such as a screenreader, or speech recognition software.

Where To Go Next

- If students are ready for an advanced challenge, introduce them to **if** blocks to create a feature to detect which player wins the game. A third variable to track the number of rounds played might be needed.

- Micro:bit also has a user-friendly Python editor. This workshop can be adapted to introduce learners to a text-based programming language.

technocamps

@Technocamps

Find us on Facebook