LESSON 18

Reaction Timer

60 minutes | Microbit Masterclass

WHAT THIS LESSON DOES

Follow this guide to build a fun project on your Micro:bit. Start by creating a new project, add a welcome message, set up a countdown, introduce a random delay, create variables, and record your reaction time.

WHAT STUDENTS SHOULD BE ABLE TO DO

- Develop skills in creating and managing a new project on the Micro:bit platform.
- Acquire knowledge on how to create and display messages using code.
- Understand and apply the concept of countdowns and delays in programming.
- Learn to create and utilise variables for storing time stamps.
- Gain proficiency in recording and displaying user interactions in real-time.

BEFORE THE LESSON

- Open the Micro:bit editor (makecode.microbit.org) yourself and check it loads through the school filter, and confirm the on-screen simulator runs so pupils without a physical Micro:bit can still test their code.
- One device per pupil or per pair, logged in and able to save a project.
- Have a finished Reaction Timer saved so you can show the target and demonstrate the button press quickly.

Reaction Timer

HOW THE LESSON RUNS

- Create a New Project
- Creating a Welcome Message
- Creating a Countdown
- Adding a Random Delay
- Create Variables
- Recording Player Reaction

TEACHING MOVES

- 1. Create a New Project. Show the New Project button and insist everyone names it 'Reaction Timer' now, so saved work is findable later. Check each pair reaches a blank workspace before moving on.
- 2. Creating a Welcome Message. Run the simulator so pupils see 'Get Ready' scroll across the LEDs. Point out that showString displays letters one at a time, unlike showNumber.
- 3. Creating a Countdown. Stress that the pause of 1000 is what makes each number hold on screen. Remove one pause live so pupils see the numbers flash past and understand why delays matter.
- 4. Adding a Random Delay. Run it two or three times so pupils see the wait length change each time. Ask why an unpredictable delay makes the game fair, not just fancier.
- 5. Create Variables. Model creating the two variables before pasting code. Explain a variable is a labelled box that holds a value, and startTime records the exact moment the full screen lights.
- 6. Recording Player Reaction. Walk through the subtraction slowly: reactionTime minus startTime is the gap between signal and press. Have pairs test and compare their millisecond scores.

WHAT TO WATCH FOR

- Pupils think the number shown is the total time since power-on rather than their reaction time, because runningTime keeps climbing. Show what happens without the subtraction, then with it. Make clear startTime is a marker and only the difference between the two timestamps is the reaction.
- Pupils expect the random delay to be the same each run and think their code is broken when the wait changes. Explain randomRange picks a fresh number every time by design, and that this is exactly what stops players anticipating the signal.
- Blocks are added in the wrong order, so the button check runs before the signal appears. Read the program top to bottom together and confirm the button event sits after the full-screen signal in the sequence.

DIFFERENTIATION

- Emerging: Pair them with a pupil who has the project running, and give them just the showString and countdown blocks to reproduce before adding the rest. Sit beside them for the variable step and name each block aloud as they place it.