LESSON 60

Studio: Reaction Timer

40 minutes | Project studios (optional extension sessions) | Built in micro:bit | Outcome 1.7 | Outcome 1.8

WHAT THIS LESSON DOES

Students build a micro:bit reaction timer with a countdown, random delay, and button event to measure reaction time in milliseconds, then run a fair class test of best times.

CURRICULUM OUTCOMES

Outcome	What the specification asks for
1.7	discuss and implement core features of structured programming languages, including variables, operators, loops, decisions, assignment and modules
1.8	test the code to determine its functionality and to identify and manage errors

WHAT STUDENTS SHOULD BE ABLE TO DO

- Build a reaction game on the micro:bit using events and timing (LO 1.7)
- Run a fair test of the class's reaction times (LO 1.8)

BEFORE THE BELL

- Have micro:bits, USB cables or Bluetooth pairing, and charged batteries ready; students logged in to the Coding Ireland platform and the MakeCode micro:bit editor open.
- Open a fresh project on the board projector so you can model the first steps before pairs work independently.
- Agree a simple class save routine (name and location) so work is there next lesson.
- Optional: a quick board space for collecting best fair times later in the lesson.

WHAT YOU NEED

Item	Qty	Per	If you do not have it
micro:bit (with USB lead)	1	pair	run the lesson on the IWB using the MakeCode simulator

Studio: Reaction Timer

HOW THE LESSON RUNS

4 min	introduction	Start: what we're building
3 min	content	Create a New Project
2 min	content	Creating a Welcome Message
5 min	content	Creating a Countdown
8 min	content	Adding a Random Delay
7 min	content	Create Variables
5 min	content	Recording Player Reaction
3 min	content	Make sense: what worked, what broke
2 min	content	Reflect
1 min	summary	What you covered

TEACHING MOVES

- Model the first two build steps on the board, then let pairs continue; circulate and keep the pace on the full reaction loop rather than polishing the welcome string.
- Stress that times are in milliseconds (thousandths of a second) so students read results such as 250 as a quarter of a second.
- When the class plays, agree rules for a fair attempt before collecting scores (for example no pressing early, same hand, seated).
- Write one prediction on the board at the start about when the full screen should appear, and come back to it when the class pulls together at the end to compare it with what the program does.

WHAT TO WATCH FOR

- Setting startTime at the wrong moment so every score looks huge or near zero: the stamp must be taken when the full-screen signal appears, not at the start of the countdown.
- Button handler missing or still empty: the A-button event must store running time, subtract startTime , and show the result.
- Random delay range typed in seconds instead of milliseconds, so the wait feels wrong: keep the range in the 1000 to 5000 band.

DIFFERENTIATION

- Support: Provide a printed block order checklist (welcome, countdown, line, random pause, full screen, start stamp, A-button maths) and sit pairs who need it near you for the variables step.
- Extension: Challenge early finishers to reject early presses, add a short restart path, or try the countdown-as-function idea mentioned in the build.
- Extension: Ask confident pairs to lead the fair-test rules and collate the class best times on the board.