LESSON 25

Digital Dice Roller

40 minutes | Introduction to Python | Outcome 3.11 | Outcome 3.13 | Outcome 3.14 | Outcome 2.6 | Outcome 2.7 | Outcome 1.22 | Outcome 2.16

WHAT THIS LESSON DOES

In this lesson, you'll build a digital dice roller with a Micro:bit using Python. Follow step-by-step instructions to code shake detection, random number generation, and LED displays, creating a fun, interactive project from scratch.

CURRICULUM OUTCOMES

Outcome	What the specification asks for
3.11	use and control digital inputs and outputs within an embedded system
3.13	develop a program that utilises digital and analogue inputs
3.14	design automated applications using embedded systems
2.6	construct algorithms using appropriate sequences, selections/conditionals, loops and operators to solve a range of problems, to fulfil a specific requirement
2.7	implement algorithms using a programming language to solve a range of problems
1.22	read, write, test, and modify computer programs
2.16	use data types that are common to procedural high-level languages

WHAT STUDENTS SHOULD BE ABLE TO DO

- Understand the basics of programming with Micro:bit using Python.
- Develop skills in handling random number generation and event detection.
- Master the use of lists and loops for data storage and animation.
- Apply functions to organise and reuse code effectively.
- Create an interactive project simulating a real-world object.

BEFORE THE LESSON

- Open the online Python editor for micro:bit and confirm it loads behind the school filter; the simulator's shake button lets students test without hardware.
- If using physical micro:bits, charge or cable them up and check each shows up over USB, then walk through downloading a .hex once yourself so you can talk students through it.
- Have the full working code saved somewhere the class can reach, since the multi-line Image strings are easy to mistype.

Digital Dice Roller

HOW THE LESSON RUNS

Introduction to the Project
Import Modules
Define Dice Faces
Detect Shake Event
Animate Rolling Dots
Select and Display Random Face
Wrap in a Function
Download your Code
Wrap Up and Extensions

TEACHING MOVES

- 2. Import Modules. Point out that running this does nothing visible on purpose; a clean run with no error is the success signal. Remind them import errors are almost always a spelling slip.
- 3. Define Dice Faces. Show how the five rows of digits map to the LED grid and how 9 means full brightness, 0 means off. Have them read one face aloud before typing so the pattern makes sense.
- 4. Detect Shake Event. Draw the shape of while True with an if inside: the board checks over and over, acting only when shaken. Stress this loop never ends, which is what we want here.
- 5. Animate Rolling Dots. Explain the animation is just a second list of images shown in quick sequence to fake motion before the real result appears. Let them slow or speed it up to see the effect.
- 6. Select and Display Random Face. Connect random.randint to picking a list index, not the number itself. Ask what index range matches six faces and why it starts at zero.
- 7. Wrap in a Function. Frame this as tidying, not new behaviour: the same rolling code now lives in one named block that can be called again. Nothing the dice does should change.
- 8. Download your Code. Circulate as students flash to hardware; the shake gesture feels different on a real board than the simulator button, so let them try it physically.
- 9. Wrap Up and Extensions. Have each student name which of lists, loops, random and functions did which job. Offer the extension of showing the number as text after the image for those ready.

WHAT TO WATCH FOR

- Students expect the index range to be 1 to 6 to match a dice, so they use random.randint(1,6) and hit an index error or skip a face. Have them count the list positions out loud starting at zero, then work out that six faces occupy indices 0 to 5, so the call is random.randint(0,5).
- Students think the animation and the final face are the same thing, or that the animation determines the result. Point out the animation is decorative only; the actual outcome comes from the random pick afterwards. Comment out the animation to show the dice still works.
- Students believe a working simulator means it will work on the board, then are surprised the shake behaves differently. Contrast the simulator shake button with physically shaking hardware, and have them test on the board before declaring it finished.

DIFFERENTIATION

- Emerging: Give the completed code to copy and run first, then remove one section at a time and have them retype it, so a broken environment is not blocking the whole task. Pair them with someone whose board is flashing correctly so USB and download trouble does not stall the coding.
- Developing: Ask them to predict what each new step will add before they paste it, then check against what happens. Have them explain in their own words why random.randint uses 0 to 5 rather than 1 to 6.
- Proficient: Set the extension of displaying the rolled number as scrolling text after the image, or adding more animation frames. Ask them to justify why wrapping the logic in a function is better practice, framing reuse and readability as they would in a coursework write-up.