

LESSON 24

Microbit Bop It Game

60 minutes

Microbit Innovators

WHAT THIS LESSON DOES

In this step-by-step lesson, you'll create a 'Bop It' game on your Microbit. You'll learn to create a new project, define variables, set a countdown, generate random numbers, display images, and program buttons. You'll also learn how to handle game over scenarios and add extra actions for an added challenge.

WHAT STUDENTS SHOULD BE ABLE TO DO

- Understand and apply the concept of variables in coding.
- Develop skills in creating and manipulating countdowns in a game context.
- Implement conditional statements to control game actions.
- Programme Microbit buttons to interact with the game.
- Enhance problem-solving skills by adding new features to the game.

BEFORE THE LESSON

- Microbits charged and ready, one per pupil or pair
- Computers or tablets ready to create a Microbit project and download code to the devices
- Test download to a Microbit before the lesson so pairing and cables are working

Microbit Bop It Game

HOW THE LESSON RUNS

- Create a new Microbit project
- Create a variable called 'action'
- Add a start countdown block
- Set 'action' to a random number
- Display the image
- Program the buttons
- Check Game Over
- Challenge!

TEACHING MOVES

- 1. Create a new Microbit project. Pitch the game briefly: a random LED image, the matching button action, as many correct presses as possible in 20 seconds. Get every pupil or pair starting a new Microbit project.
- 2. Create a variable called 'action'. Show how to create the action variable and set it to -1 in on start. Stress that -1 means "no action yet, pick a new one".
- 3. Add a start countdown block. Add startCountdown set to 20000 milliseconds. Say aloud that this is 20 seconds on the clock from the moment the program runs.
- 4. Set 'action' to a random number. In forever, only when action equals -1, set it to a random number from 0 to 2. Confirm those three values map to the three later button actions.
- 5. Display the image. Before coding, map 0 to left arrow (A), 1 to right arrow (B), 2 to both arrows (A+B). Then add the if blocks that show each LED pattern.
- 6. Program the buttons. For each button: if action matches, addScore and reset action to -1; else set action to -2 and gameOver. Suggest duplicating blocks to save time.
- 7. Check Game Over. Place if game.isGameOver at the top of forever and set action to -2 so the loop stops issuing new images once time is up or the game has ended.
- 8. Challenge! Download and play on the real Microbit first. Then challenge finishers to widen the random range to 0-3, show an X, and handle shake when action is 3.

WHAT TO WATCH FOR

- Pupils treat action as the score or leave it stuck on 0, 1 or 2 after a correct press, so the next prompt never appears. Remind them: correct press must set action back to -1 so the forever loop knows to pick a new random action.
- Pupils put button checks inside forever instead of onButtonPressed, or forget the else branch that ends the game on a wrong press. Show that button code lives in its own input block, and walk one full correct-path and wrong-path example together.
- When extending the challenge, pupils widen the random range but forget the new image and the onShake handler for action 3. Checklist: range 0-3, LED for 3, onShake that scores only when action is 3 and otherwise ends the game.

DIFFERENTIATION

- Emerging: Pair with a partner and complete on start, the countdown, and the first action image together before tackling buttons. Have them explain what action = -1 means before they write more code.
- Developing: Use block duplication for the three button handlers so only the condition and matching detail change. Ask them to test one correct press and one wrong press on the simulator before downloading.
- Proficient: Set the challenge: add action 3 with an X image and an onShake handler that scores or ends the game. Invite a further feature of their own, such as a different gesture or a shorter countdown, using the same pattern.