LESSON 19

Microbit Hot Potatoe

60 minutes | Microbit Adventures

WHAT THIS LESSON DOES

Follow this guide to build an exciting game on your Microbit. Start by creating a new project, set up a countdown with animations, and add game-over effects. Download the code and play with friends to see who loses!

WHAT STUDENTS SHOULD BE ABLE TO DO

- Develop a basic understanding of creating and manipulating variables in Microbit.
- Apply the concept of random number generation in a practical scenario.
- Understand and implement countdown functionality using loops.
- Integrate visual animations into the code to enhance user experience.
- Understand how to incorporate sound and visual effects to indicate the end of a game.

BEFORE THE LESSON

- Open and test makecode.microbit.org on your own device and confirm the school filter does not block it; check pupils can reach the site and start a new project.
- Have one micro:bit per pupil or pair, each with a power source ready: a battery pack, or a USB cable and power bank.
- Confirm you can download a .hex file to a micro:bit yourself before the lesson, so you can talk pupils through it.
- Clear a little floor or desk space so pupils can safely pass the micro:bit around when they play.

Microbit Hot Potatoe

HOW THE LESSON RUNS

Create a new project
Create a variable
Countdown
Animation
Game over
Play the game

TEACHING MOVES

- 1. Create a new project. Have everyone reach a blank new project before moving on. Point out this is code they build up, not a finished program to run yet.
- 2. Create a variable. Show that the variable seconds is a named box that holds a number. Stress randint(4, 20) picks a fresh random number each time A is pressed, so no two games last the same.
- 3. Countdown. Explain the while loop: it keeps subtracting 1 and pausing one second until seconds reaches 0. Ask pupils to predict when the loop stops before running it.
- 4. Animation. Point out the two icons flashing show the game is live. Read the note aloud that each icon adds 600ms, so a game second is really about 2.2 seconds.
- 5. Game over. Show that the skull and sad sound sit outside the loop, so they only fire once the countdown ends. This is the moment that decides the loser.
- 6. Play the game. Walk pupils through downloading the .hex and attaching power. Remind them to pass the micro:bit carefully and to press A to restart each round.

WHAT TO WATCH FOR

- Pupils place the skull and sound blocks inside the while loop, so they fire every countdown step instead of once at the end. Show them the block sitting inside versus outside the loop and run both. Ask which one only happens when the game is over.
- Pupils expect the timer to run for exactly the number of seconds shown, and think the code is broken when it takes longer. Return to the note about the icons adding 600ms each. Have them count a real game and compare it to the number, so the delay makes sense.
- Pupils think random means the same number keeps coming up, or that they can predict it. Have them press A several times and watch the game length change. Confirm randint gives a new value in the 4 to 20 range each press.

DIFFERENTIATION

- Emerging: Sit beside them for the first block and get one line working before adding more. Pair them with a partner who has the project open, so they can copy the block layout and see it running.
- Developing: Ask them to explain what the while loop is doing in their own words before they add the animation. Have them change the randint range and predict how the game length changes before testing.
- Proficient: Challenge them to add a second animation or a different game-over sound using blocks they can find themselves. Ask them to make the game harder or easier by changing the numbers, and to explain the effect of each change.