

LESSON 8

Chase the Dot

60 minutes (2 periods)

Intermediate Projects

WHAT THIS LESSON DOES

Get started with creating a fun game on your Microbit where you control a chaser dot to catch a target dot. Follow step-by-step instructions to set up the project, code the dots, and score points within 10 seconds.

WHAT STUDENTS SHOULD BE ABLE TO DO

- Develop skills in creating and managing Microbit projects on makecode.com.
- Understand and apply the concept of variables in coding to create game sprites.
- Learn to create and use functions for repetitive tasks in a game scenario.
- Develop skills in using gestures to control game elements in a Microbit project.
- Understand and implement the concept of scoring and round systems in game development.

BEFORE THE BELL

- Open makecode.com on the school network and confirm a new micro:bit project loads in the browser.
- Count out one micro:bit per student or pair plus USB cables. Charge them the day before and check each powers on.
- Build the finished Chase the Dot project yourself once so you can project the blocks or flash a working demo if the class stalls.

Chase the Dot

HOW THE LESSON RUNS

- Create a new Microbit project
- Create 2 new variables
- Create a function called newRound
- Call newRound at the start
- Move the chaser dot
- Score a point

TEACHING MOVES

- 1. Create a new Microbit project. Project the empty MakeCode editor. Name the game aloud, then restate the aim in one line: tilt the centre chaser to catch the edge target, one point each catch, ten seconds on the clock.
- 2. Create 2 new variables. Watch them create target and chaser before any blocks. Point to Advanced, Game for create sprite. Stress blink and low brightness on target only so the two dots stay visually distinct.
- 3. Create a function called newRound. Say why a function: same random-edge setup runs at the start and after every catch. Have them build newRound fully before anyone calls it. Check random x and y both sit in the 0 to 4 range.
- 4. Call newRound at the start. On the board, show newRound(), startCountdown at 10000, then the start sound, in that order at the bottom of on start. Ask why calling the function matters if the blocks already exist.
- 5. Move the chaser dot. Build one tilt branch live, then let pairs finish the other three. Read each gesture against its x or y change. Students test on the real micro:bit, not only the simulator.
- 6. Score a point. Place the isTouching check at the top of forever, before the tilt branches. On touch: addScore 1, happy sound, newRound. Play a full ten-second round and read the score aloud.

WHAT TO WATCH FOR

- Students define newRound but never call it, so the target stays at the centre and the game looks broken. What to do: Ask them to point to where newRound runs in on start and again after a catch. If either call is missing, add it before changing other blocks.
- Students may think: They treat a single tilt as a one-step move, but forever keeps applying the gesture, so the chaser flies off the screen. What to do: Have them tilt briefly then level out while watching x and y. If it still races, check they are not stacking extra change blocks outside the gesture checks.
- Students may think: They expect scoring when the dots look close, without isTouching, or they put the touch check outside forever so it only runs once. What to do: Trace one loop aloud: touch first, then tilts. Confirm isTouching sits inside forever and that a catch calls newRound so the target moves.

DIFFERENTIATION

- Emerging: Pair them with a neighbour whose sprites already show. Have them recreate only target, chaser, and the newRound call before adding tilts. Stand with them at variable creation. Tick off blink and brightness on target before they touch any gesture blocks.
- Developing: If tilts work but scoring does not, ask them to read the forever block top to bottom and insert isTouching above the gestures. Have them flash to the micro:bit and narrate one catch: score sound, then target jumps, then countdown still running.
- Proficient: Ask them to explain newRound to a stuck pair using only the board, no typing for the other students. Challenge them to tune feel only with blocks already in the project: blink rate, brightness, or countdown length, then justify the change in a quick playtest.