

technology

Build and debug the core features

Lesson at a glance

Open by framing pupils as game programmers who expect bugs, then model a fresh MakeCode Arcade project on the IWB. Pairs build step by step: sprite, left/right movement, gravity, tile map, jump, ground-check, danger and goal events. They play-test, spot and fix at least one bug, then record the bug and fix on the Investigation Journal page after a hands-off code talk.

Learning objectives

- Build a platform game with movement, gravity, a ground-checked jump, danger tiles and a goal
- Run, find and fix at least one bug in the program
- Describe the bug and how it was fixed in the Investigation Journal

Before the bell – prep

Log every paired device into arcade.makecode.com before the bell and open a blank project on the IWB. Have Investigation Journals ready for the bug-and-fix page. Bookmark the Arcade site; offline kits won't substitute for this build.

Materials

Item	Qty	Per	Source	Low-cost substitute
tablet or laptop with MakeCode Arcade	1	pair	school devices	run the lesson on the IWB with pupils taking turns

Safety watch-point

Devices stay flat on desks during win/lose celebrations — no lifting or high-fiving over open laptops. Calm testing only.



Teaching moves

- **What we're building:** Show the finished-level image on the IWB. Ask 'What does a programmer do when a game doesn't work first time?' Reassure that bugs are expected today, then confirm every pair is on a fresh empty Arcade project.
- **Create a new Arcade project:** Model New Project on the IWB so every pair sees the same start screen. Walk the room and check no pair has reopened an old project before you move on.
- **Create your sprite:** Open the sprite editor on the IWB and draw a simple character pixel by pixel. Give a firm time cue so pairs don't decorate forever; struggling pairs may copy your design exactly.
- **Move left and right:** Add the left/right controller block live, press Play, and have every pair try both buttons. If nothing moves, check the block is snapped in and not left detached in the workspace.
- **Add gravity:** Drop $vy = 50$ inside a forever block and press Play so the class watches the sprite fall off-screen. Ask 'Why is it falling?' — reassure them this is correct until the map gives ground.
- **Draw the map:** Model ground (Draw Walls), lava danger and the goal crystal on the IWB. Key check: ground tiles must be walls or the sprite falls through. Keep maps short for pairs who struggle.
- **Jump:** Model the jump variable and up-button event, then let pairs test. Watch for anyone who skips creating the jump variable — without it the code fails.
- **Move through the map:** Give play-through time. Circulate asking 'Can your sprite actually reach the goal, or is a jump too big?' Praise pairs who spot and fix a gap themselves.
- **Only jump when touching ground:** Swap in the `isHittingTile` bottom-check on the IWB. Have pairs try jumping in mid-air — it should now fail. Flag this as the strong bug-and-fix moment for the Journal.
- **Danger tiles:** Add the `onOverlapTile` lava event. Remind pairs the tile in code must match the map tile. Have them deliberately step on lava to confirm Game Over.
- **Goal:** Add the red-crystal overlap for the You Win screen. If nothing happens at the end, re-check the goal tile matches the code exactly.
- **Code talk — what worked and what we'd change:** Hands off keyboards. Draw out one real bug — mid-air jumping is the classic — and revoice how that pair noticed and fixed it. Then send pairs to the Investigation Journal to record the feature, the bug, and the fix.

What it should show

A working platformer: left/right move, gravity fall, jump only on ground, lava ends the game, goal shows You Win. The common intentional bug is endless mid-air jumping before the ground-check. Surprising fails usually mean ground not set as walls, jump variable missing, or danger/goal tile in code not matching the map.

Misconceptions & interventions

- **Pupils think jumping in mid-air is fine because the up button always works.** — After they add the bare jump, ask them to jump three times without landing. Then add the isHittingTile bottom check and have them try again so they feel the fix.
- **Pupils place ground tiles but forget Draw Walls, so the sprite falls straight through.** — On the IWB, toggle wall on one tile and press Play — sprite stands. Toggle off — it falls. Have the pair re-paint ground with walls on.
- **Pupils expect danger or goal code to work even when the map uses a different tile than the one named in the overlap block.** — Point at the exact tile image in the code and on the map side by side. Swap one to match, re-test, and watch the event fire.

Differentiation

Emerging	Developing	Proficient
• Pair copies the modelled sprite and a short three-platform map; teacher table helps snap the forever gravity and wall ground first. • Journal scaffold: sentence starters for 'what the game did' and 'what we changed'.	• After the ground-check works, ask 'What other button event could you add?' and let them try one small change. • Re-test danger and goal deliberately and note both outcomes in the Journal.	• Move the goal or add a wider gap and re-balance jump timing so the level still fair-plays. • Critique a neighbour's fair feel: is any jump impossible, or any danger unavoidable?

Cross-curricular hook

Links to English oral language — pairs explain a bug and fix in clear sentences before writing the Journal entry.