technology

Build and debug the core features

Lesson at a glance

Frame pupils as game programmers who expect bugs. Model each block on the IWB — sprite, left/right movement, gravity, tile map, jump, danger and goal — while pairs build the platform game in MakeCode Arcade on their devices and press play after each piece. The pivot is the mid-air jumping bug: pupils spot it, swap in the ground-check code and test the fix. They log the bug and fix in the Investigation Journal, then a code talk revoices how they debugged it.

Learning objectives

- Add core features such as scoring, loops or events to the saved game
- 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 pair's device into arcade.makecode.com and confirm a fresh, empty project before the bell — the biggest time-sink is pupils opening old work. Have the IWB on the same site so you model from a matching screen. Investigation Journals out for the bug-and-fix recording. No headphones needed; keep game sound low.

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

Keep game celebrations calm so devices stay flat on desks — the only hazard is a knocked or dropped tablet. Confirm work is saved before logging off.

Lesson 24

Teaching moves

- **Create a new Arcade project:** Model creating a new project on the IWB so every pair sees the identical empty screen. Walk the room and check each device shows a fresh project — anyone in an old file will get lost by step 4.
- **Create your sprite:** Open the sprite editor on the IWB and draw the character pixel by pixel. Give a firm time cue — pupils over-decorate. Let a struggling pair copy your modelled design exactly rather than stall here.
- **Add gravity:** Add the forever block setting vy to 50 and press play — the sprite falls off screen, which is correct now. Say so out loud: 'positive pulls it down, the ground comes next step.' Head off the 'it's broken' panic before it spreads.
- **Draw the map:** This is the fiddliest step — model the Draw Walls tool slowly for ground tiles. Keep maps short. If a sprite falls straight through the floor, the ground was drawn without the wall setting.
- **Only jump when touching ground:** Swap in the ground-check jump code on the IWB, then have pupils try to jump in mid-air to prove the endless-jump bug is fixed. This is the strongest bug-and-fix moment — flag it for the Journal now, before you move on.
- **Danger tiles:** Add the danger event and stress that the lava tile named in the code must match the tile placed on the map. Have pupils step on lava on purpose to confirm Game Over. Keep the reaction calm so devices stay on desks.
- **Code talk — what worked and what we'd change:** Draw out one real bug a pair hit (usually the mid-air jump) and have them revoice how they found and fixed it. Hold the focus on the debugging process, not the finished game.

What it should show

The finished game should let the sprite move left/right, fall under gravity, jump only when touching the ground, die on danger tiles, and show You Win on the goal tile. The most common 'it doesn't work' is a mismatch: the lava or crystal tile named in the code isn't the tile placed on the map, so nothing triggers — have pupils re-check the tile matches. A sprite falling through the floor means the ground wasn't set as a wall.

Misconceptions & interventions

- **Pupils think a bug means they've done something wrong or the game is ruined.** — Reframe on the spot — every programmer builds, runs and fixes. Point to the mid-air jump: 'that's a bug, and finding it is the job.' Have them fix it and see it work.
- **Pupils expect the sprite to jump forever mid-air because pressing up keeps working.** — Ask 'why should you only jump when you're on the ground?', then swap in the isHittingTile ground-check code and let them test that mid-air presses now do nothing.
- **Pupils place a danger or goal tile but nothing happens when the sprite touches it.** — Show that the tile named in the onOverlapTile block must be the exact tile placed on the map. Line the code tile up against the map tile on the IWB and re-place if they differ.

Differentiation

Emerging	Developing	Proficient
• Pair with the teacher table and copy the modelled sprite and short map exactly. • Give a very short map with one clear jump so the build stays achievable and testable.	• After the ground-check fix, add a second danger tile and test the Game Over still triggers. • Ask 'where else could a bug hide?' and have them run their game to find one.	• Move the goal tile to make a harder level, then test the jump is still reachable. • Critique a partner's map: is there a jump that's too big, or a tile that doesn't match its code?

Cross-curricular hook

Link to Maths — the jump timer (750 milliseconds) and vy values are about number and time, and pupils reason about positive/negative direction.