

technology

Build, debug and share a fuller game

Lesson at a glance

Pairs reopen the MakeCode Arcade game they saved earlier and name one thing a visitor would still wish it could do. After a short board look at the forever and on-A block shapes, you model Route B: gravity plus jump on A, a deliberate wrong-button bug, the fix, and deleting one unused block. Pairs then add one loop or event to their own save, run-fix-tidy, record the change on the Investigation Journal page, and demo to a partner pair before saving.

Learning objectives

- Build a fuller Arcade game with a player sprite and enemies that move
- Find and fix at least one bug while building, and say what was wrong
- Demo the finished game and explain one thing the code does

Before the bell – prep

Before the lesson, open MakeCode Arcade on the IWB with a sample player sprite that already moves left/right. Leave Route B and Route A labels ready on the board. Check every pair can find their earlier save; if not, plan neighbour forks. Investigation Journal pages and working devices ready.

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

Carry devices with two hands; no food or drink at desks; coil trailing cables so nobody trips. Shut lids and return chargers at pack-away.



Teaching moves

- **Getting Started:** Get every pair into their earlier save before you talk. Ask the hook once: what still cannot the game do that a visitor would notice? Take three quick answers only, then move on — keep loop and event for the next step.
- **Loops and events:** On MakeCode Arcade, point only at the purple forever block and the on A button block. Leave both shapes visible while you say the two plain definitions. Do not build a full feature yet and do not put bug or simplify on the board glossary.
- **Watch the next steps:** Deep-model Route B only: set ay to 300, then on A set vy to -150. Before running, ask what they will notice if A does nothing or if gravity is missing. First wire the jump to B on purpose, name bug and fix aloud, then delete a leftover play-sound block that never runs and name simplify.
- **Extend, debug and simplify:** Steer movement-only saves to Route B by default; Route A only if coin and score already exist. Circulate for look-fors: they can point at the new loop or event, name a bug or careful check, and still run after removing one unused block. At about 15 minutes, freeze for a 30-second partner whisper on what still feels buggy.
- **Record the bug and the change:** Send pairs to the Investigation Journal starters: We added... / The bug was... (or check we ran) / We removed... because.... Accept honest half-fixes. Prompt concrete detail — what they saw on screen, not only 'we made it better'.
- **Demo to a partner:** Pair of pairs, one direction only: about 2 minutes silent try-out with makers off the keyboard, 1 minute explain, 30 seconds for one steal-worthy idea. Protect the close — skip role swap or board demos if time is tight.
- **Reflect and save:** Display-only talk: trickiest bug in the room and what testing helped notice. Revoice that a first version is a start and tidying unused blocks is real maker work. Then save to the school account, shut lids, and clear trailing cables.

What it should show

Most pairs add Route B (jump on A with gravity) or, if they already had coin and score, a forever overlap that scores once and moves or destroys the coin. A working demo plus one named bug/check and one removed unused block meets the bar. Score rocketing on a coin usually means the coin was never moved; sprite flying off means ay was skipped.

Misconceptions & interventions

- **'I found a bug — I'm bad at coding.'** — During the deliberate wrong-button demo, revoice: a bug is information, and real coders expect them and hunt them on purpose. Praise the notice before the fix.
- **'More blocks make a better game.'** — When you delete the leftover sound block, run again and show the game still works. Say plainly: a short clear program that works beats a long messy one that almost works.
- **On Route A, pairs leave the coin in place so the score rockets every frame.** — Point at the board warning and the if stack: move or destroy the coin inside the overlap (or score only once per touch), then re-run.

Differentiation

Emerging	Developing	Proficient
• If lost, have them say the game's job in one sentence, then point them at the matching board label (jump needs gravity + on-A; score needs a forever check). • If their save will not open, let them fork a neighbour's working base and still complete one extend-debug-tidy cycle.	• Stick to one route only; either a named bug fix or one careful check both count. • Use the 15-minute freeze whisper to name one thing still buggy before the journal write.	• After the first change works and unused blocks are gone, add a second small event (sound on score or a simple win screen). • Movement-only pairs who finish Route B cleanly may try a basic coin + score and Route A as stretch.

Cross-curricular hook

Link the partner demo to English oral language: makers explain the change, bug or check, and tidy-up in plain sentences.