

Start and build a MakeCode Arcade game

Lesson at a glance

Open with a short MakeCode Arcade demo of Coin Catch and hands-up call-outs on the goal and scoring. Pairs name the game's job in plain words, then sketch sprite, left-right movement and coin scoring on the Investigation Journal page. You model the first build chunk by chunk on the IWB—hero sprite, move with buttons, overlap scoring, Share link—running after each chunk. Pairs rebuild the same plan on devices, write the Share link and one next-session addition, then close with a brief fix-talk.

Learning objectives

- Create a MakeCode Arcade project and add a character sprite
- Program the character to move under player control and keep it on the screen
- Run the game as it is built and say what each piece of code makes the character do

Before the bell – prep

Book one tablet or laptop per pair that opens <https://arcade.makecode.com>. Before class, load or rebuild a simple hero-and-coin sample for the demo; open a fresh Arcade project on the IWB. Keep Investigation Journal pages ready for plan, blocks used, Share link and next-step note.

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

Supervise device handling and browser close-down; plug tablets in on pack-away. No physical hazard beyond careful shared-device use.

**Teaching moves**

- **Name the game's job:** Run a two-minute think-pair-share: who the player is, how they move, what scores. Take three or four pairs and revoice into plain algorithm language—player moves the hero with keys; touch coin, score goes up. Pull anyone who jumps to button names back to the job first. Flag this as Project Part 1 of 2.
- **Plan the game:** Model aloud a worked plan on the board: sprite—little hero; move—left and right keys; score—touch a coin; one line left for next session. Show the three Investigation Journal prompts plus Share-link space. Circulate and check every pair has named sprite, movement and scoring before any device opens.
- **Model the first step:** Drive Arcade yourself on the IWB in fixed order: set Player sprite in on start and run; add move mySprite with buttons and run; set Food/coin sprite, on overlap change score by 1 and destroy or respawn, then run into the coin. Model one save routine only—Share, copy link, show where it goes on the journal. Think aloud: place, move, score, run, save.
- **Build the first working version:** Leave the chunk list visible. Pairs rebuild hero, movement, then scoring, running after each chunk. Watch for still heroes missing move with buttons and send them back to chunk 2. Scoring is required today. If devices are short, rotate drivers on the IWB while the class predicts the next block.
- **Record what works so far:** Paper only: what the sprite does so far, main blocks or steps, Share link written clearly, and one thing to add next session. Pairs still stuck on movement record the plan and the stuck-point honestly—useful for Part 2.
- **What was trickiest?:** Three quick contributions only. Revoice fixing language: read the blocks in order, change one thing, run again. Link forward to opening the same project next session. Pack away devices and journals calmly.

What it should show

By the end, each pair should have a Player sprite on screen that moves with buttons, a coin overlap that raises score by 1, and a working Share link on the journal. A common odd result is a visible hero that never moves—almost always the move with buttons block was skipped; rebuild chunk 2 and run again.

Misconceptions & interventions

- **The plan is just decoration—we can skip straight to dragging blocks.** — Hold up the journal page: the plan is the algorithm in plain words; the blocks only write that algorithm for the computer. No editor until sprite, movement and scoring are named.
- **I made a sprite so it should move when I press the keys.** — Point back to chunk 2 on the IWB—without move mySprite with buttons under on start the hero sits still. Add the block, run, and watch the d-pad work.
- **If it breaks, change lots of blocks at once until something works.** — Model calm fixing: read the blocks in order, change one thing, press run. Ask the pair what they predicted before the last run.

Differentiation

Emerging	Developing	Proficient
• Sit stalled pairs at the teacher table and rebuild one modelled chunk at a time from their plain-words plan, running after each add. • If movement still fails, they record the plan and stuck-point on the journal rather than forcing a full score build.	• Prompt a predict-before-run after each chunk: what should the hero or score do when we press play? • Nudge pairs to tidy sprite labels or refine left-right feel while staying inside today's plan.	• Early finishers refine movement feel or labels only—no new on-device side quests away from the class arc. • Ask them to name one concrete addition for next session (loop or new event) and justify it against today's working version.

Cross-curricular hook

Link to English procedural writing—pairs practise precise, ordered instructions the way a recipe or traffic-light sequence works.