

LESSON 42

Read, document and debug game code

40 minutes

Strand 3: Coding at the next level

Built in MakeCode Arcade

Outcome 3.9

Outcome 3.6

Outcome 3.7

WHAT THIS LESSON DOES

Students build a Hot Air Balloon game in MakeCode Arcade, reading each block of code carefully, finding and fixing bugs, and adding comments so another person can follow the program.

CURRICULUM OUTCOMES

Outcome	What the specification asks for
3.9	analyse code to determine its function and identify errors or potential errors
3.6	document programs to explain how they work
3.7	present the documented code to each other in small groups

WHAT STUDENTS SHOULD BE ABLE TO DO

- Read unfamiliar game code and work out what it does, then find and fix a planted error (LO 3.9)
- Comment and document the code so another person can follow it (LO 3.6), and explain a section to a partner (LO 3.7)

BEFORE THE BELL

- Devices ready, students logged in to MakeCode Arcade in the browser
- Open MakeCode Arcade on the board so you can model the first steps in the text or blocks view
- Have a simple plan for how pairs will swap and read each other's projects near the end
- Remind students how the class saves and reopens Arcade projects

WHAT YOU NEED

Item	Qty	Per	If you do not have it
tablet or laptop with MakeCode Arcade	1	pair	run the lesson on the IWB with students taking turns

Read, document and debug game code

HOW THE LESSON RUNS

4 min	introduction	Start: what we're building
2 min	content	Create a New Project
4 min	content	Set Up the Game Environment
3 min	content	Create the Player
4 min	content	Control the Player
5 min	content	Create the Obstacles
4 min	content	Collide with Obstacles
4 min	content	Read code you did not write, and find the planted error
2 min	content	Leave a note for the next person
2 min	content	Test Your Game
3 min	content	Make sense: what worked, what broke
2 min	content	Reflect
1 min	summary	What you covered

TEACHING MOVES

- Model reading each new section aloud before students type: what does this line set up, and what should we see when we run?
- Push the text view at least once so students practise tracing unfamiliar code, not only dragging blocks
- When the game runs, pause and ask where a deliberate or accidental bug might hide (interval timing, velocity, overlap kinds)
- Leave two minutes for pairs to swap devices or projects: each student points at one section and explains what it does

WHAT TO WATCH FOR

- Player moves up and down as well as left and right: check the move values so vertical speed stays at 0
- Clouds never appear or pile in one spot: check the update interval, random x range, and that velocity points downward
- Overlap does not end the game: confirm player and clouds use Player and Enemy kinds that match the overlap handler

DIFFERENTIATION

- Support: Model the first two build steps on the board and give a short comment starter, e.g. what the sprite is and how it moves
- Extension: Ask stronger students to add clearer comments on the obstacle timer and overlap, or to adjust speed and spacing and explain the change
- Extension: Invite a pair to trace one tricky section in the text view and teach it back to a neighbouring pair