Lesson 25

technology

Start and build a MakeCode Arcade game

Lesson at a glance

The class builds their first MakeCode Arcade project, step by step, on their own devices: a sprite they draw themselves in the sprite editor, joystick control so they can drive it around, a block that stops it wandering off the screen, and a visual effect they can swap. The learning is that a program is a set of steps and each block makes one thing happen, so run it after every block and ask what changed. Model each piece on the board first, then let them build and play.

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

One tablet or laptop per pupil or pair, on arcade.makecode.com, and the board linked to your device. Build the project yourself once beforehand, especially the sprite editor, because drawing a character is where the time goes and where pupils most want help. Agree a time limit for drawing before you start, or half the class will still be colouring when the movement block goes in.

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

Normal care with devices, and supervise packing away at the end. Nothing else on this lesson, the work is all on screen.

Lesson 25

Teaching moves

- **What we're building:** Show a finished sprite moving around the screen with an effect on it, then close it. Say that every game starts as a list of steps and that they will add one step at a time and watch what each one does. Do not open the editor yet.
- **Create a sprite:** Model this and run it before anyone draws. An empty square on the screen is the sprite, and seeing it appear from one block is the moment the idea lands.
- **Design your character:** Give this a firm time limit, five minutes is plenty, and say so before they start. The sprite editor is a different screen from the block editor, so show the way back before you let them loose in it.
- **Control your character:** Run it the moment the block is in and let them drive around for a few seconds. Ask what the block did, in their own words. This is the step that turns a picture into a game.
- **Make them stay on the screen:** Ask first what happens if they keep driving in one direction. Let them lose the sprite off the edge, then add the block, so the fix answers a problem they have just had rather than one you described.
- **Change the effect:** A good place to let them experiment, and a good place to name the habit: change one thing, run it, see what happened. That is what they will do for the rest of the year when something goes wrong.
- **Code talk:** Talk only, nothing typed and nothing saved. Take two or three pupils on where the program did something they did not expect, and how they spotted which block was at fault. Ask one to say what a single block makes happen, then finish on what they would add next.

What it should show

Most pupils finish with a character they drew, driving around the screen under joystick or arrow-key control, staying inside the play area, with an effect on it. A pupil with a sprite that moves has met the objectives even without the effect. Expect drawing the character to take longer than every block put together.

Misconceptions & interventions

- **Pupils think the sprite is gone for good when it drives off the edge of the screen.** — Ask where they last saw it and which direction they were pushing. It is off the edge, not deleted, and the next block is what keeps it in view.
- **Pupils expect a score or a way to win, because that is what a game means to them.** — Say plainly that today is about making a character they control, and that scoring comes later in the project. Keep their ideas for the code talk.
- **Pupils get lost in the sprite editor and cannot find their code again.** — Show the way back to the block editor on the board before anyone opens it, and again for anyone who gets stranded. It is a navigation problem, not a coding one.
- **When something misbehaves, pupils start a new project rather than look for the block at fault.** — Stop the pupil and ask which block they added last. Change one thing, run it again, and name that as how a programmer finds a bug.

Differentiation

Emerging	Developing	Proficient
• Aim for a sprite that moves. Let them pick a simple shape rather than draw a detailed character, so the time goes on the code. • Sit them where they can see your screen and run the project after every block.	• Build through the stay-on-screen block and the effect. Ask them to say what each block made happen before they add the next. • When something misbehaves, have them name the block they think is responsible before changing it.	• Try several effects and describe the difference between them, then keep the one that suits their character. • Ask what they would need to add for the sprite to do something when it reaches the edge instead of stopping.

Cross-curricular hook

Visual arts, through the sprite editor. Designing a character on a small grid of pixels is a real constraint, and choosing which few squares carry the character is the same decision as choosing what to leave out of a drawing.