

LESSON 19

Introducing MakeCode Arcade

60 minutes (2 periods)

Introduction to Game Design

WHAT THIS LESSON DOES

Get started with creating arcade games using a fun and easy-to-use tool. Follow step-by-step instructions to build your first project, design a character, control its movements, and add cool effects to bring your game to life.

WHAT STUDENTS SHOULD BE ABLE TO DO

- Understand the basic functionality and usage of MakeCode Arcade.
- Develop skills in using the MakeCode Arcade code editor.
- Create and customise a sprite in MakeCode Arcade.
- Implement controls to move a sprite and keep it within the screen boundaries.
- Apply and experiment with different effects in MakeCode Arcade.

BEFORE THE LESSON

- Open arcade.makecode.com yourself first and confirm the school filter does not block it. It runs in the browser with no login needed.
- One device per pupil, or per pair if devices are short. Have the site loaded and ready before pupils arrive.
- A large screen or projector so you can model creating a project and adding blocks as the class follows along.
- Note that MakeCode saves projects in the browser, so pupils who move between devices may not find their work again.

Introducing MakeCode Arcade

HOW THE LESSON RUNS

What is MakeCode Arcade?

The Code Editor

Create a new Arcade project

Create a sprite

Design your character

Control your character

Make them stay on the screen

Add an effect

Change the effect

Time to Explore

Wrapping Up

TEACHING MOVES

1. What is MakeCode Arcade? Draw the link to Scratch out loud: same coloured drag-and-snap blocks, new tool. Pupils who know Scratch will settle faster once they hear it is familiar.
2. The Code Editor. Point to each of the three areas on the projected screen as you name it: simulator, toolbox, code area. Ask pupils to point to the same three on their own screens before moving on.
3. Create a new Arcade project. Model creating and naming a project once on the big screen, then wait until every pupil has a named project open before you continue.
4. Create a sprite. Show where the sprite block lives in the toolbox and drag it in yourself first. The block starts as an empty grid, which is expected: reassure pupils nothing is broken.
5. Design your character. Give a firm time limit for painting. Tell pupils now that there is time at the end to polish, so they stop perfecting and move on with the class.
6. Control your character. After they add the movement block, have every pupil test the arrow keys in the simulator straight away so they see cause and effect before the next step.
7. Make them stay on the screen. First let them drive the sprite off the edge on purpose, then add the stay-on-screen block. Feeling the problem first makes the fix make sense.
8. Add an effect. Model adding one effect and watching it in the simulator. Keep it to one effect for now so everyone sees the same result before they experiment.
9. Change the effect. Now open it up: let pupils swap the effect in the block and compare what they see. Ask two or three to describe their favourite to the class.
10. Time to Explore. This is the polish and stretch time you promised earlier. Set a clear expectation: add a second sprite or a new effect, not start over from scratch.
11. Wrapping Up. Recap the build order aloud, sprite, design, control, boundaries, effect, so pupils can name the steps they took rather than just the result.

WHAT TO WATCH FOR

- Pupils add each new block in the wrong place or as a separate loose block, so the code does not run as expected. Model exactly where each block clicks into the existing code on the big screen, and check the code area of a few pupils as they build.
- Pupils think painting a character is the main task and spend the whole lesson in the Sprite Editor. State the time limit before they open the editor and remind them the coding, not the drawing, is today's goal. Free-play time comes at the end.
- Pupils expect the sprite to move before the control block is added, and think the tool is broken when it sits still. Explain that a sprite only does what its code tells it. No control block means no movement yet, which is exactly right for that step.

DIFFERENTIATION

- Emerging: Sit them beside a pupil who has the project open and let them mirror the toolbox location for each block. Break the sprite-creation step into find the block, drag it in, then paint, checking after each part.
- Developing: Ask them to say aloud what each block does before they run it, so they connect the code to the result. Have them fix a deliberately misplaced block on your screen to check they understand where code goes.
- Proficient: In the explore step, challenge them to add a second sprite with its own effect and explain the difference to a partner. Ask them to try every effect and report back which one they would use for an explosion and why.