LESSON 9

First Arcade Project

60 minutes | Game Design Studio

WHAT THIS LESSON DOES

Get started with creating your own arcade game using MakeCode Arcade. Follow step-by-step instructions to build a project, add sprites, design maps, and programme movement and interactions. Create, test, and play your game in a browser or on a handheld device.

WHAT STUDENTS SHOULD BE ABLE TO DO

- Understand and utilise MakeCode Arcade for creating games.
- Manipulate the Code Editor to build and modify game elements.
- Create and customise sprites for use in a game.
- Develop a tile map and implement walls for game navigation.
- Implement game mechanics such as projectiles, sprite movement, and life count.

BEFORE THE LESSON

- Open and test arcade.makecode.com on your own device first, and check your school filter does not block it. Sign in if you want work saved to accounts.
- One device per pupil or pair. Sound on or headphones ready, as later steps play a power-down effect on a hit.
- If you have BrainPad or GameGo handhelds, have them charged and a cable ready for the final download step. The lesson works fully without them.

First Arcade Project

HOW THE LESSON RUNS

- What is MakeCode Arcade?
- The Code Editor
- Create a new Arcade project
- Add a sprite
- Choose a sprite from the gallery
- Move your sprite
- Draw the tile map
- Draw the walls
- Camera follow sprite
- Add some projectiles
- Set their direction and speed
- Detect overlap
- Lose a life
- Send the code to your handheld

TEACHING MOVES

- 1. What is MakeCode Arcade? Draw the link to Scratch: same colourful blocks, new playground. Keep it to a minute so pupils get hands on the editor quickly.
- 2. The Code Editor. Point out the three areas on screen: simulator, toolbox and code area. Name them once so pupils can follow your later instructions.
- 3. Create a new Arcade project. Walk the class to arcade.makecode.com together and have everyone create and name a project before anyone codes. Catch login trouble here.
- 4. Add a sprite. Explain a variable can hold a whole sprite, not just numbers or text. Model dragging the create-sprite block once on the big screen.
- 5. Choose a sprite from the gallery. Send pupils to the gallery rather than the paint editor for now, so a design detour does not eat the build time.
- 6. Move your sprite. Add the move-with-buttons block and have pupils test the arrow keys in the simulator immediately. Movement working is the first win.
- 7. Draw the tile map. Keep the first map simple: a border around the edge only. Show how to open the tile map editor before setting them loose.
- 8. Draw the walls. Stress that walls are drawn with a separate Draw Walls tool over the same tiles. Have pupils test that the sprite now stops at the edge.
- 9. Camera follow sprite. Show the sprite going off screen first, then add the camera-follow block so pupils see exactly what problem it fixes.
- 10. Add some projectiles. Explain the two-second interval fires a projectile automatically. Test it before adding any consequences so pupils see it moving.
- 11. Set their direction and speed. Use the vx and vy table on the board. Positive and negative values decide which side and direction, so change one value and watch what moves.
- 12. Detect overlap. Introduce overlap as the game noticing two sprites touch. Test the sound and spray effect fire on a hit before adding lives.
- 13. Lose a life. Add set-life to 3, change-life-by minus one, and destroy the projectile together. Point out the destroy line stops one hit costing many lives.
- 14. Send the code to your handheld. Invite pupils to add one thing of their own. Only run the handheld download for pupils who have finished and have hardware.

WHAT TO WATCH FOR

- Pupils think the projectile spawns on the side they expect regardless of the numbers, and are confused when it starts elsewhere. Return to the vx and vy table. Have them change one value from positive to negative and watch the start side flip, so the rule becomes visible.
- Without the destroy line, a single projectile overlapping the player keeps firing the effect and can drain several lives at once. Show the difference: run it without destroy, then add otherSprite.destroy() and run again. Make clear the projectile must be removed once it lands a hit.
- Pupils draw the tile map but forget walls are a separate tool, so the sprite walks straight through the border. Have them reopen the editor and switch to Draw Walls, then test movement against the edge. A wall is a tile you cannot pass, drawn separately.

DIFFERENTIATION

- Emerging: Pair with a partner and get movement working end to end before moving on, so there is an early success to build from. Give the exact block name to find in the toolbox rather than the group, to cut the search time.