LESSON 6

Shark Attack

60 minutes | Advanced Game Development

WHAT THIS LESSON DOES

Get ready for an exciting coding journey! You'll create a thrilling game where you control a character avoiding chasing enemies. Using MakeCode Arcade, follow simple steps to build and test your game from start to finish.

WHAT STUDENTS SHOULD BE ABLE TO DO

- Develop skills in using MakeCode Arcade platform for game creation.
- Understand and apply coding concepts to create and control sprites.
- Implement game logic to create dynamic gameplay, including enemy creation and movement.
- Apply coding techniques to detect sprite overlaps and trigger game events.
- Enhance problem-solving and creativity by modifying and expanding game features.

BEFORE THE LESSON

- Open arcade.makecode.com yourself first and check your school filter is not blocking it. This is the single thing most likely to stop the lesson.
- One device per pupil, or per pair if devices are short. Building the game solo works best where possible.
- Build the finished game once yourself so you can demonstrate each code block and troubleshoot fast.
- Have the site open on the projector so you can model creating a project and adding blocks live.

Shark Attack

HOW THE LESSON RUNS

Introduction
Create a new Arcade project
Create your player sprite
Move the sprite
Make it stay on the screen
Create the enemy sprites
Set the enemy position
Make them chase the player sprite
Game over
Game Over, Well Done!

TEACHING MOVES

- 2. Create a new Arcade project. Model creating and naming a new project on the board before pupils touch their devices. Show them where the code workspace and the game preview sit on screen.
- 3. Create your player sprite. Show pupils that the grey box in the create block opens the sprite gallery. Let them pick any character, the code is identical whichever they choose.
- 4. Move the sprite. After adding movement, have pupils test straight away with joystick or arrow keys. Fixing one block and retesting is the core rhythm of the whole lesson.
- 5. Make it stay on the screen. Ask pupils to first push their sprite off the edge, then add the block and watch it get stuck at the wall. Seeing the problem before the fix makes the block meaningful.
- 6. Create the enemy sprites. Stress selecting Enemy, not Player, as the sprite kind here. Choosing the wrong kind breaks the chase and game-over steps later, so check this before moving on.
- 7. Set the enemy position. Point out that without this block sharks spawn on top of the player. Setting them to the top left gives the player a fighting chance.
- 8. Make them chase the player sprite. Have pupils test and watch a shark track them. If they want harder gameplay, the speed value of 10 is the number to change.
- 9. Game over. Explain the overlap block as a watchman checking whether player and enemy touch. Make sure pupils pick LOSE in the game-over block, then let them play.
- 10. Game Over, Well Done! Invite pupils to extend the game: more enemies, faster sharks or a power-up. Circulate and ask what they changed and why.

WHAT TO WATCH FOR

- Pupils set the enemy sprite kind to Player instead of Enemy, so the chase and game-over blocks never fire. Check the sprite kind dropdown in step 6 as you circulate. Have pupils read their own block aloud: it should say SpriteKind.Enemy.
- Pupils forget to select LOSE in the game-over block, so nothing meaningful happens when a shark catches them. At step 9, point to the dropdown inside the game.gameOver block and have every pupil confirm it reads LOSE before they test.
- Pupils think a block is wrong when the game does not react, but they have simply not pressed play or their test window is closed. Remind pupils to run the game preview after every new block. Build a test-after-each-step habit early rather than debugging ten blocks at once.

DIFFERENTIATION

- Emerging: Pair them with a partner and let them mirror the projected code block by block. Sit with them for step 2 so the project is created and named before the class moves ahead.