

LESSON 11

Bat Battle

60 minutes

Game Design Studio

WHAT THIS LESSON DOES

In this lesson, you'll learn to build an exciting game with MakeCode Arcade. Create a spaceship that moves, shoots projectiles, and scores points by hitting targets. Watch out-if a target hits your spaceship, it's game over!

WHAT STUDENTS SHOULD BE ABLE TO DO

- Develop skills in using MakeCode Arcade to create an interactive game.
- Understand how to create, control, and position player and enemy sprites.
- Learn to program game interactions such as shooting projectiles and detecting overlaps.
- Gain knowledge on how to keep score and end the game in MakeCode Arcade.
- Enhance problem-solving and debugging skills by experimenting with the code and adding new features.

BEFORE THE LESSON

- Open and test <https://arcade.makecode.com> on the school network beforehand: confirm the site is not blocked by the filter and that the sprite gallery loads.
- One charged device per pupil, since each pupil builds their own game and needs the joystick or arrow keys to play.
- Have your own finished Bat Battle saved so you can demo the target game and show the code blocks pupils are aiming for.

Bat Battle

HOW THE LESSON RUNS

Introduction

Start a New Project

Create your player

Move left and right

Create the targets

Random position at the top

Make them fall

Fire!

Hit the target

Score a point

Game over

Wrap Up

TEACHING MOVES

- 1. Introduction. Show your finished game playing first. Let them see the spaceship, the falling bats and the game over before any code, so they know what they are building towards.
- 3. Create your player. Point out that the block is dragged from the toolbox, not typed. Show where sprites.create lives and how to pick the red spaceship from the gallery.
- 4. Move left and right. Demo moving the sprite before wiring controls, then add the move block. Show that the joystick and the arrow keys do the same thing.
- 5. Create the targets. Stress the onUpdateInterval block runs every 2 seconds forever. Have pupils choose the bat from the gallery so it matches the lesson name.
- 6. Random position at the top. Explain random means a different spot each time. Run it so pupils see bats appearing across the top rather than always the same place.
- 7. Make them fall. Explain vy makes the bat move down. Mention the auto-destroy stops the game slowing as off-screen bats pile up.
- 8. Fire! Test the A button on the simulator with them. Make sure the projectile is created from the sprite so it launches from the spaceship.
- 9. Hit the target. Slow down here. Check pupils select Projectile and Enemy in the overlap block and drag otherSprite, not mySprite, into destroy.
- 10. Score a point. Show the score appear top corner after adding changeScoreBy. Have them play and read their own score aloud to confirm it counts.
- 11. Game over. Check pupils pick Enemy and Player in this overlap block. Have them deliberately let a bat crash to confirm game over triggers.

WHAT TO WATCH FOR

- Pupils drag mySprite into the destroy block instead of otherSprite, so the projectile or player disappears rather than the bat. Stop at the overlap step and explain sprite and otherSprite are the two things that touched. Walk one pupil through swapping it and show the correct behaviour.
- Pupils mix up which kind goes in each overlap block, so hits or game over never fire. Point at the two dropdowns each time: Projectile with Enemy for scoring, Enemy with Player for game over. Have them read the block back to you before testing.
- Pupils think a block placed on the workspace runs even if it is not connected inside an event or start block. Show that loose blocks do nothing. Zoom in on where each block snaps inside its event and have them check theirs is fully attached.

DIFFERENTIATION

- Emerging: Pair them with a partner who has the project open so they can copy the block layout step by step. Sit them near your screen and keep your demo project visible as a reference to match against.
- Developing: Have them explain back to you what onUpdateInterval and vy are doing before moving on, so they build with understanding, not copying. Ask them to fix one deliberate mistake you make in a block dropdown, to practise spotting the Projectile and Enemy settings.
- Proficient: Ask them to add a new feature the lesson invites, such as a second projectile type or a faster fall speed, and explain what they changed. Challenge them to make the game harder over time, then describe to you how their new code behaves.