

## LESSON 7

# Space Shooter

60 minutes

Advanced Game Development

### WHAT THIS LESSON DOES

Get ready to build your own space-themed game with MakeCode Arcade! In this lesson, you'll create a spaceship, dodge and shoot asteroids, and programme exciting gameplay features. Start your coding adventure now!

### WHAT STUDENTS SHOULD BE ABLE TO DO

- Develop understanding of MakeCode Arcade for game creation.
- Gain proficiency in creating and controlling game sprites.
- Learn to implement game mechanics such as scoring and lives.
- Understand how to detect and respond to sprite interactions.
- Apply coding skills to create a complete Space Shooter game.

### BEFORE THE LESSON

- Open [arcade.makecode.com](https://arcade.makecode.com) yourself first and confirm the school filter does not block it; the site needs to load and run in the browser.
- One device per pupil, each able to reach [arcade.makecode.com](https://arcade.makecode.com). Pupils build one game each, so pairing is a fallback rather than the plan.
- Decide how pupils get the code blocks: display each step's block on the board or have the lesson open on their own screens so they can copy accurately.

## Space Shooter

### HOW THE LESSON RUNS

Introduction

Create a new Arcade project

Create your spaceship sprite

Control the spaceship

Set the number of lives

Create the asteroids

Program the asteroids

Fire rockets

Destroy asteroids

Lose a life

Game On!

### TEACHING MOVES

- 3. Create your spaceship sprite. Stress that the spaceship must point right, because that is the side asteroids fly in from. Have them rename the sprite to spaceship straight away so later code refers to the right thing.
- 4. Control the spaceship. Get everyone to test movement before moving on. Check the ship stops at the screen edge rather than vanishing off it, then have them try all four arrow keys.
- 5. Set the number of lives. Point out this sets the starting lives shown on screen. Keep it at three for now so everyone's game behaves the same when you troubleshoot later.
- 6. Create the asteroids. Explain the `onUpdateInterval` block fires every 1000 milliseconds, so a new asteroid appears every second. Let them design the asteroid in the sprite editor before adding more code.
- 7. Program the asteroids. Watch that asteroids move left across the screen and get destroyed at the edge. If they pile up and slow the game, the destroy-off-screen block is likely missing or wrong.
- 8. Fire rockets. Tell pupils the A button fires, and on the simulator that is the space key. Have them confirm a rocket launches from the ship before moving on.
- 9. Destroy asteroids. Walk the room as they set the overlap kinds to Projectile and Enemy. The common slip is leaving `mySprite` in the destroy block instead of `otherSprite`. Have them test a hit scores a point.
- 10. Lose a life. Same overlap pattern, now Player and Enemy, again with `otherSprite` in the destroy block. Have them deliberately let an asteroid hit the ship and check a life disappears and the camera shakes.
- 11. Game On! Give a few minutes of play and encourage them to beat a high score. Ask what they would change: faster asteroids, more lives, a different ship.

### WHAT TO WATCH FOR

- Pupils leave mySprite in the destroy block instead of swapping it for otherSprite, so the wrong sprite is destroyed or nothing happens on a hit. Point at the two objects named in the overlap block. Explain sprite is our ship or rocket, otherSprite is the asteroid we want gone. Have them read the block aloud before testing.
- Pupils pick the wrong kinds in the overlap blocks, mixing up Projectile, Enemy and Player, so rockets do not destroy asteroids or the ship never loses a life. Have them match each overlap to its job: rocket hits asteroid is Projectile and Enemy, asteroid hits ship is Player and Enemy. Get them to test each one separately.
- Pupils assume typing code once means it works and skip testing, then cannot tell which step broke the game. Insist on a quick play-test after every step that adds behaviour. Finding one broken thing is easy; finding five at the end is not.

### DIFFERENTIATION

- Emerging: Sit beside them for the first sprite creation and rename so the setup is right before anything is built on it. Let them copy the block for a step, run it, and see the result before you explain what it does.
- Developing: Have them read each new block back to you and say what it should make happen before they run it. When something breaks, ask which step they last tested that worked, so they learn to narrow down the fault themselves.
- Proficient: Once the game plays, set a challenge: make asteroids faster, add a second rocket type, or change the starting lives. Ask them to explain their overlap blocks to a pupil who is stuck, naming sprite and otherSprite as they go.