

LESSON 8

Target Test

60 minutes

Advanced Game Development

WHAT THIS LESSON DOES

Get ready to create an exciting game with MakeCode Arcade! You'll learn to build a player sprite, move it around, create targets, shoot projectiles, and track your score. Follow the steps to start coding and have fun!

WHAT STUDENTS SHOULD BE ABLE TO DO

- Develop skills in creating and moving sprites using MakeCode Arcade.
- Understand how to generate targets at random positions within a game environment.
- Gain proficiency in programming game controls, such as shooting projectiles.
- Learn to implement a scoring system within a game.
- Encourage creativity and problem-solving by adding additional features to the game.

BEFORE THE LESSON

- Open <https://arcade.makecode.com> yourself first and confirm the school filter does not block it. Sign in or start a new project so you can demo the block editor.
- One device per pupil, or one per pair if devices are tight. The A-button firing works with keyboard as well as an on-screen joystick.
- Have your own finished game saved and ready to show, so pupils see the target when they get stuck mid-build.
- Project your screen. Much of this lesson is watching where a block snaps in.

Target Test

HOW THE LESSON RUNS

Introduction to Target Test

Start a New Project

Create your player

Move left and right

Create the targets

Random position at the top

Fire!

Hit the target

Score a point

Wrap Up

TEACHING MOVES

- 2. Start a New Project. Walk pupils through naming and saving the project now, before any code goes in. Losing work to a closed tab is the most common frustration here.
- 3. Create your player. Point out the `sprites.create` block sits in an on start context. Show where to click to open the sprite gallery and pick the red ship rather than drawing pixels by hand.
- 4. Move left and right. Have them test movement straight away. If the ship drifts off-screen or will not move, check the controller move block is attached to the correct sprite variable.
- 5. Create the targets. Explain `onUpdateInterval 2000` means a new target every two seconds. Let them watch targets pile up before adding random positioning, so the next step's purpose is obvious.
- 6. Random position at the top. Show that setting a random x with a fixed y near the top is what scatters targets across the sky. Ask pupils to predict what changing the y value would do.
- 7. Fire! Emphasise the projectile is created from the player sprite, so it launches upward from the ship. Test the A button before moving on. A silent nothing usually means the wrong sprite is selected.
- 8. Hit the target. Slow right down. Check three things aloud: kind Projectile and kind Enemy in the overlap block, `otherSprite` in the destroy block not `mySprite`, and fire selected as the effect.
- 9. Score a point. Show that `changeScoreBy(1)` goes inside the same overlap block, below the destroy line. Then play it as a class and celebrate the first score appearing.
- 10. Wrap Up. Ask a few pupils to name one feature they would add next. Let finishers actually start building it rather than sitting idle.

WHAT TO WATCH FOR

- Pupils put mySprite into the destroy block instead of otherSprite, so shooting destroys the player rather than the target. Remind them the overlap block hands you two sprites: sprite is the projectile, otherSprite is the target. It is otherSprite you want to blow up.
- Pupils select the wrong sprite kinds in the overlap block, so hits never register and nothing happens when they fire. Have them read the block back to you: Projectile overlaps Enemy. If either dropdown says something else, no collision will ever be detected.
- Pupils think blocks can go anywhere and drop the scoring line outside the overlap block. Point out that changeScoreBy only runs when it sits inside the on overlap block. Outside it, the score would either never change or change constantly.

DIFFERENTIATION

- Emerging: Pair them with a partner and give them one job at a time: get the ship on screen first, test it, then move on. Do not let them race ahead of a working step. Keep your saved example open on the projector so they can match block by block.
- Developing: Let them build each step then test it before you confirm, rather than checking every block for them. Ask them to explain why the projectile comes from the ship.
- Proficient: Point them at the wrap-up ideas: a second target type, multiple lives, faster spawning. Have them explain their new code back to you before they play it. Ask them to change the two-second interval and predict the effect on difficulty before testing.