

LESSON 1

Banana Jump

60 minutes

Coding and Interactive Creators

WHAT THIS LESSON DOES

In this lesson, students will create an engaging game using Scratch, a visual programming language. They will learn to control character movements, create clones, and detect collisions. The game involves a cat that must jump to avoid falling bananas. The lesson covers creating a new project, adding and animating sprites, programming movements, and detecting collisions.

WHAT STUDENTS SHOULD BE ABLE TO DO

- Understand and apply the basics of Scratch programming including character movements, creating clones, and detecting collisions.
- Create a new project in Scratch and add elements such as backdrops and sprites.
- Use loops and conditional statements to animate sprites and control their movements.
- Implement collision detection to trigger game events.
- Develop problem-solving and computational thinking skills through game development.

BEFORE THE LESSON

- Open scratch.mit.edu on your own screen and check the school filter is not blocking it. If pupils are to save, confirm they can log in to their Scratch accounts beforehand.
- One device per pupil, or per pair if numbers are tight.
- Have a finished Banana Jump project open yourself so you can show what the game does before pupils build it.

Banana Jump

HOW THE LESSON RUNS

Introduction

Create a new project

Add the Blue Sky backdrop

Animating the Cat

Add the bananas sprite

Adding Bananas

Making Bananas Move

Jumping Cat

Detecting Collisions

Wrap up

TEACHING MOVES

1. Introduction. Show your finished game running for ten seconds so pupils know exactly what they are building: a cat that jumps to dodge falling bananas.
2. Create a new project. Walk pupils to scratch.mit.edu and to a fresh project together. Do not let anyone race ahead until everyone has a blank stage open.
3. Add the Blue Sky backdrop. Point out that the brown strip at the bottom is the ground. Everything the cat and bananas do sits on that line.
4. Animating the Cat. Explain that the forever loop swapping costumes is what makes the cat look like it is running. Have pupils click the green flag and check the cat is bottom left and moving.
5. Add the bananas sprite. Make sure pupils select the correct sprite in the sprite list before adding code, so the banana blocks do not land on the cat by mistake.
6. Adding Bananas. Explain that the banana hides itself and makes clones, so the original vanishing on the green flag is expected, not a bug. The action happens in the next step.
7. Making Bananas Move. Name the parts: a clone starts on the right, shows itself, and the repeat block nudges it left in small steps. Get pupils to trace one banana's journey aloud.
8. Jumping Cat. Point out the two repeats: ten steps up, then ten steps down. Have pupils press space and check the cat returns to the ground rather than drifting up.
9. Detecting Collisions. Use the watchman idea in the notes: the forever loop keeps asking is the cat touching a banana, and stops everything when the answer is yes. Test by letting the cat get hit.
10. Wrap up. Let pupils play, then set one challenge: change a single number and predict what will happen before clicking the flag.

WHAT TO WATCH FOR

- Pupils think the banana code is broken because the banana disappears when they click the green flag. Explain the hide block is deliberate. The visible bananas are clones made later, so nothing is wrong until the movement code in the next step is added.
- Pupils add blocks to whichever sprite is showing, not the sprite the step names, so the cat gets the banana's code. Before every new block, have pupils say out loud which sprite is selected and check it is highlighted in the sprite list.
- Pupils change y by only once instead of inside a repeat and wonder why the cat jumps in one jerk or not at all. Show that the jump is many small steps inside the repeat block, not one big move. Drag the block inside the loop together.

DIFFERENTIATION

- Emerging: Give the exact blocks one line at a time and check each one runs before moving on. Pair them with a partner who can already drag and snap blocks confidently.
- Developing: Ask them to explain back what one block does before they add the next, so the code is understood rather than copied.
- Proficient: Set the wrap-up challenges: make the cat jump higher, speed up the bananas, or make them appear more often, and have them predict the effect first.