LESSON 13

Maze Master: Help Your Character Escape!

30 minutes | Adventures in Digital Creation

WHAT THIS LESSON DOES

In this lesson, you'll guide your students through creating and navigating a maze in Scratch Jr. They'll learn to edit the background to design their maze, place their character, and use movement blocks to navigate. Extend the challenge by encouraging more complex mazes. Conclude with students sharing their creations.

WHAT STUDENTS SHOULD BE ABLE TO DO

- Understand and apply the concept of mazes in Scratch Jr.
- Develop skills in editing and designing backgrounds in Scratch Jr. for maze creation.
- Master the use of movement blocks to guide a character through a maze.
- Enhance problem-solving abilities by creating and navigating through complex mazes.
- Share and discuss their mazes, reflecting on challenges encountered and solutions developed.

BEFORE THE LESSON

- Check every device has Scratch Jr installed and opening, with one device per pupil or per pair.
- Open the Background Editor yourself first so you can point straight to the background icon above the stage and the paintbrush edit icon.
- Have a simple barrier maze already drawn on your own device to show what the finished start point should look like.

Maze Master: Help Your Character Escape!

HOW THE LESSON RUNS

Introduction
Mazes
Designing Your First Maze
Placing the Character
Coding the Maze Solution
Test and Debug
Code a Complex Maze!
Wrap up

TEACHING MOVES

- 2. Mazes. Take a few answers before giving your own definition. Land on the key idea: a maze is a path with walls, and you follow it from start to end without hitting a dead end.
- 3. Designing Your First Maze. Model the two taps slowly: background icon above the stage, then the paintbrush to edit. Keep the first barrier to a single straight line so the coding stays achievable.
- 4. Placing the Character. Show them the drag onto the X, then the checkmark to save. Warn them that leaving without the checkmark loses the background.
- 5. Coding the Maze Solution. Build the sequence aloud on the board: green flag, move forward, then turn and move to go around the line. Let them copy your block order first, then adapt.
- 6. Test and Debug. Name debugging as you do it. When a character clips the line, ask which single block or number to change rather than rebuilding the whole sequence.
- 7. Code a Complex Maze! Set a limit on the number of walls so mazes stay solvable in the time left. Remind them to mark a fresh X and end point before coding.
- 8. Wrap up. Ask two or three pupils to name which of the three skills, design, sequencing or debugging, they used most today. Keep it spoken.

WHAT TO WATCH FOR

- Pupils think one 'move forward' block will carry the character all the way through, expecting the program to steer around the wall by itself. Show that the character only does exactly what each block says. Break the path into a move, a turn, another move, and count the steps aloud together.
- Pupils add or repeat blocks randomly when the character hits the wall, instead of finding the one that is wrong. Slow them down. Run the code one block at a time and ask them to point to the exact moment it goes wrong before changing anything.
- Pupils leave the editor without the checkmark and lose their maze, then blame the app. Make the checkmark the last thing everyone does aloud together before returning to code.

DIFFERENTIATION

- Emerging: Give them a barrier you have already drawn so they can start straight on the coding without the design step. Sit with them for the first move-and-turn sequence and count the steps together.
- Developing: Ask them to predict where the character will stop before they press the green flag, then check. Have them fix one deliberate error you place in a sequence on your device.
- Proficient: Point them at the complex maze early, with turns and dead ends. Ask them to explain their sequence back to you block by block, then swap devices with a partner and solve each other's maze.