LESSON 44

Scratch AI Pose Space Game

60 minutes | Discovering Artificial Intelligence

WHAT THIS LESSON DOES

In this lesson, you'll create an exciting space game using Scratch and Google Teachable Machine. Learn to control a spaceship with tilt poses using a Teachable Machine Pose model, or use the provided model if needed.

WHAT STUDENTS SHOULD BE ABLE TO DO

- Develop skills in using Scratch and Google Teachable Machine to create a game.
- Understand how to control a sprite using pose models.
- Learn to set up and adjust the confidence threshold for an AI model.
- Gain knowledge on how to create and manipulate clones of sprites in Scratch.
- Apply creativity to enhance and personalise the game with additional features.

BEFORE THE LESSON

- Open and test https://stretch3.github.io on your own machine first: confirm the TMPose2Scratch extension loads and the school filter does not block it.
- Each pupil needs a device with a working webcam, and camera permission will have to be allowed in the browser when the extension loads.
- Have the fallback pose model URL to hand (https://teachablemachine.withgoogle.com/models/R9i0aRHth/) for anyone without their own model.
- Check pupils have space at their desks to tilt left and right in front of the camera.

Scratch AI Pose Space Game

HOW THE LESSON RUNS

Introduction
Setup Scratch and TMPose2Scratch
Add the Rocketship Sprite
Load Teachable Machine Model
Display Pose Label
Set the Confidence Threshold
Move Left and Right
Label every x seconds
Add the Star sprite
Show and Make Stars Fall
Challenge

TEACHING MOVES

- 1. Introduction. Reassure the room that nobody is stuck: a ready-made pose model is provided, so a pupil without their own model can still play. Set the goal as controlling a spaceship by tilting.
- 2. Setup Scratch and TMPose2Scratch. Walk them through Extensions in the bottom-left, then TMPose2Scratch. Warn that the camera prompt appears once: they must click Allow, or the stage stays blank and nothing detects.
- 3. Add the Rocketship Sprite. Point out the set size and go to x/y blocks put the rocketship small and low. Check everyone deleted the Cat sprite before adding Rocketship.
- 4. Load Teachable Machine Model. Show exactly where the model URL goes in the pose classification block. Have the fallback URL on the board so no one is copying it letter by letter.
- 5. Display Pose Label. Have every pupil tick the pose label block on and tilt to watch the label change. This is the quickest way to prove their camera and model are working before they code movement.
- 6. Set the Confidence Threshold. Explain 0.6 in plain terms: the model must be about sixty percent sure before it acts. Too high and it ignores real tilts, too low and it fires by accident.
- 7. Move Left and Right. Check the hat blocks read Tilt Left and Tilt Right, and that change x is minus ten for left, plus ten for right. Test with a tilt straight away.
- 8. Label every x seconds. Have them add label once every 0.1 seconds, then tilt again and feel the difference: the rocketship now responds fast because the model checks ten times a second.
- 9. Add the Star sprite. Point out hide plus the forever loop that keeps making clones. Stress the wait pick random 3 to 8 controls how often stars appear.
- 10. Show and Make Stars Fall. Read the clone code with them: when I start as a clone runs for each copy, it drops from y 150, and delete this clone at the edge stops clones piling up and slowing the game.
- 11. Challenge. Let finishers pick one improvement to build fully rather than starting three. Ask them to describe what should happen before they code it.

WHAT TO WATCH FOR

- Pupils think the code decides when a pose is detected, and forget the webcam is doing the recognising, so they sit still and wonder why nothing moves. Send them back to the pose label block. When they tilt and see the label change, the link between their body, the camera and the code becomes clear.
- Pupils believe a higher confidence threshold makes the game better because it is more accurate, then find their tilts are ignored. Have them try 0.9 and then 0.6 and compare. Draw out that the right value is a balance, not the highest number.
- Pupils place clone code under the green flag instead of under when I start as a clone, so the star never falls or falls only once. Show that when I start as a clone is a separate hat block that runs for every copy. Movement and delete belong there, not in the create-clone loop.

DIFFERENTIATION

- Emerging: Give the fallback model URL and the exact block list so they can copy the setup and reach a working tilt before worrying about the stars. Pair them with a partner who has camera permission sorted, so a stuck webcam does not stall the whole build.
- Developing: Ask them to change one value on their own, such as the fall speed or the confidence threshold, and predict the effect before testing. Have them explain their Tilt Left and Tilt Right blocks back to you to confirm they understand the hat block, not just copied it.