LESSON 42

Scratch AI Rock, Paper, Scissors Game

60 minutes | Discovering Artificial Intelligence

WHAT THIS LESSON DOES

In this lesson, you'll build an exciting Rock, Paper, Scissors game using Scratch and Google Teachable Machine. Learn to integrate your Teachable Machine Image model with hand gestures to compete against the computer.

WHAT STUDENTS SHOULD BE ABLE TO DO

- Develop a Rock, Paper, Scissors game using Scratch and Google Teachable Machine.
- Understand and apply the use of variables in Scratch for storing player's choice, computer's choice, and the result of the game.
- Implement randomisation in Scratch to simulate the computer's choice in the game.
- Integrate Google Teachable Machine Image models in Scratch projects for gesture recognition.
- Understand and adjust the confidence threshold for AI model to improve accuracy of gesture recognition.

BEFORE THE LESSON

- Open stretch3.github.io yourself first and confirm it loads and is not blocked by the school filter. The TM2Scratch extension lives here, not on the standard Scratch site.
- One device per pupil or pair, each with a working webcam. Test that the browser is allowed to access the camera, since pupils will need to click Allow.
- Have the link to a ready-made Teachable Machine image model on hand for any pupil who has not built their own, so nobody is stuck at the model step.

Scratch AI Rock, Paper, Scissors Game

HOW THE LESSON RUNS

- Introduction
- Setup Scratch and TM2Scratch
- Add a Sprite
- Create Variables
- Randomize Computer Choice
- Load Teachable Machine Model
- Display Image Label
- Set the Confidence Threshold
- Get Player Choice
- Determine If It's a Draw
- Determine If You Win
- Else You Lose
- Challenge

TEACHING MOVES

- 1. Introduction. Reassure the class that no one needs their own model to take part. Show the provided model link so pupils without one know they can still play the full game.
- 2. Setup Scratch and TM2Scratch. Walk them to stretch3.github.io, not the usual Scratch site. Model deleting the Cat, adding TM2Scratch from Extensions, and clicking Allow so the camera feed appears in the stage.
- 4. Create Variables. Stress spelling the three variable names exactly: playerChoice, computerChoice, result. A capital or a typo now causes silent failures later that are hard to find.
- 5. Randomize Computer Choice. Write the code 1=Rock, 2=Paper, 3=Scissors on the board and leave it there. Pupils will reference this number key repeatedly through the win and lose logic.
- 6. Load Teachable Machine Model. Pause here and check every pupil has pasted their own model link, or the provided one, into the model URL block. A blank link means no gesture recognition.
- 7. Display Image Label. Turn on the image label so pupils can see live what the camera thinks their hand is. Use it as a quick check that the model is actually working.
- 8. Set the Confidence Threshold. Explain 0.6 in plain terms: the model must be at least 60 percent sure before it reacts. Let pupils raise or lower it if recognition feels twitchy or unresponsive.
- 9. Get Player Choice. Point out the label names in the hat blocks must match the model's labels exactly. If Rock, Paper, Scissors are named differently in the model, the blocks will not fire.
- 10. Determine If It's a Draw. Build the forever loop with the first if together as a class. Establish the pattern of nesting before pupils add the win and lose branches themselves.
- 11. Determine If You Win. Slow right down. Read the three winning pairs aloud and show how each and block checks two things at once, joined by or. This is the hardest step of the lesson.
- 12. Else You Lose. Show that You lose is simply the final else: if it is not a draw and not a win, the player loses. Test all three outcomes before moving on.
- 13. Challenge. Remind pupils the green flag must be clicked for each round. Offer the extensions as choices: more model images, a Space bar restart, or sound effects.

WHAT TO WATCH FOR

- Pupils think the game replays each round automatically and get confused when the computer keeps the same choice. Show that the computer only picks a new number when the green flag is clicked. Have them click it before each round, and flag the Space bar restart as the fix in the challenge.
- Pupils assume the gesture recognition failing means their code is wrong, when it is really a labelling or lighting issue. Send them to the image label display first. If the label is wrong or jumping, it is the model or the light, not the blocks. Adjust the confidence threshold or improve the model.
- Pupils get the win logic backwards, pairing choices that actually lose. Return to the board key and walk one pair through slowly, for example computer Rock beats nothing here, Paper beats Rock. Have them trace one full round by hand before trusting the code.

DIFFERENTIATION

- Emerging: Give them the provided model link straightaway so they are not held up building their own, and pair them with a partner for the camera and extension setup. Provide the win and lose logic pre-built to drop in, so they focus on getting gesture recognition working and playing the game.
- Developing: Let them build the code step by step but check in at step 11, the nested win conditions, where most stumble. Ask them to explain their number key back to you, 1 is Rock and so on, to confirm they follow their own logic before testing.
- Proficient: Send them to the challenge: add a Space bar restart so the game replays without clicking the green flag, or add sound effects. Have them improve their own Teachable Machine model with more example images and compare recognition before and after.