

⚽ GCC Sports with Code!

Activity: Code your favorite sport!

In this activity, we'll code (and play!) a sports-themed game and teach a computer to play it, too. To begin, go to the Machine Learning for Kids website.

<https://machinelearningforkids.co.uk/>

Setting up your model

Create a new machine-learning model called “score or miss” recognizing numbers. You'll want to add two values: (1) **x**, which is a **number** indicating the starting position of the ball and (2) **angle**, which is a **number** indicating the direction in which the ball is launched. Click “create” when you're done.

Start a new machine learning project

Project Name *

score or miss

Give your project a name to describe what sort of thing you'll try to teach the computer to recognise.

Recognising *

numbers

Value 1 *

x

Type of value *

number

Value 2 *

angle

Type of value *

number

ADD ANOTHER VALUE

CREATE

CANCEL

This will take you to the main **project page**. Set up the categories needed to train your model.

"score or miss"

Train

Collect examples of what you want the computer to recognise

Train

Learn & Test

Use the examples to train the computer to recognise numbers

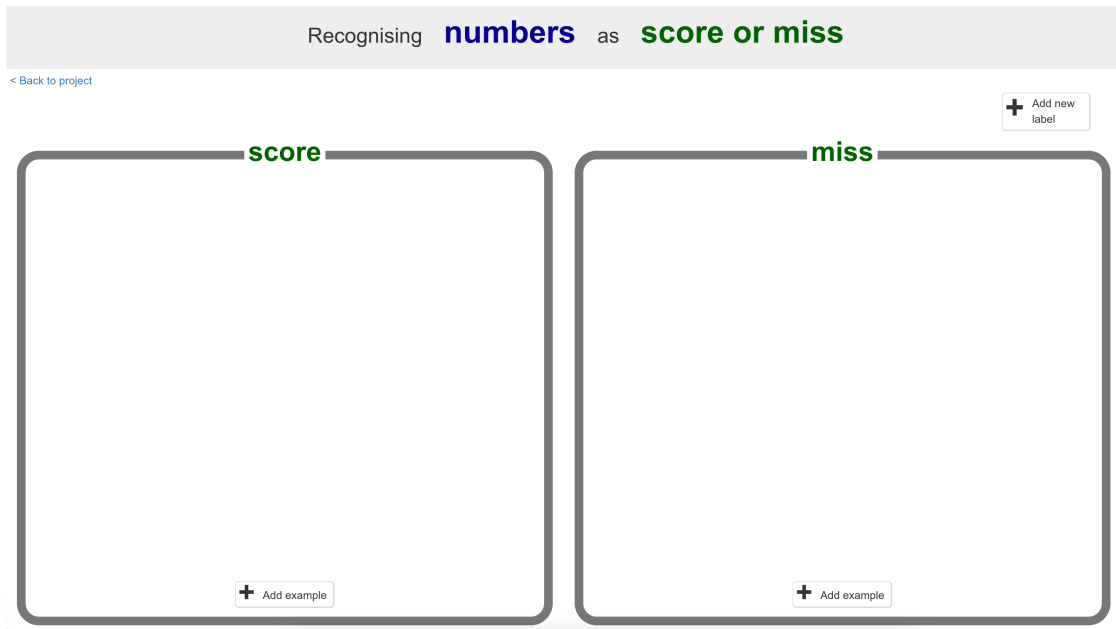
Learn & Test

Make

Use the machine learning model you've trained to make a game or app, in Scratch, Python, EduBlocks, or App Inventor

Make

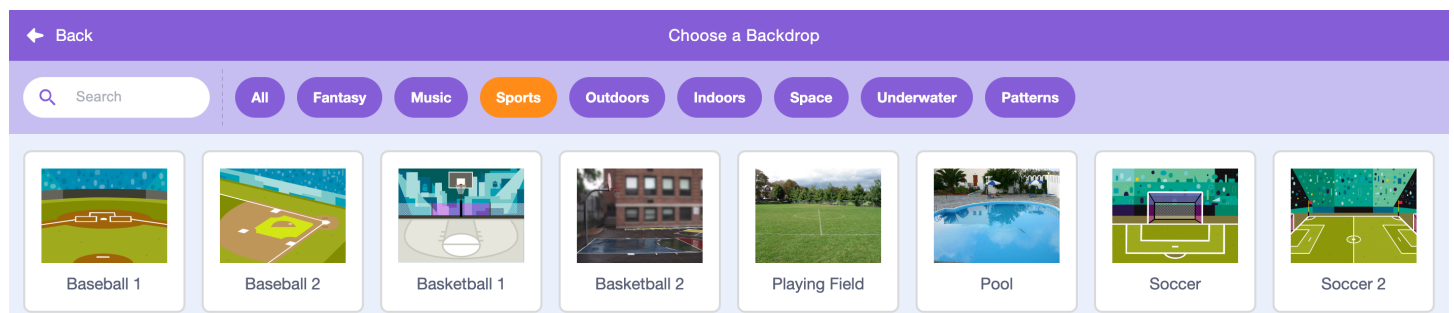
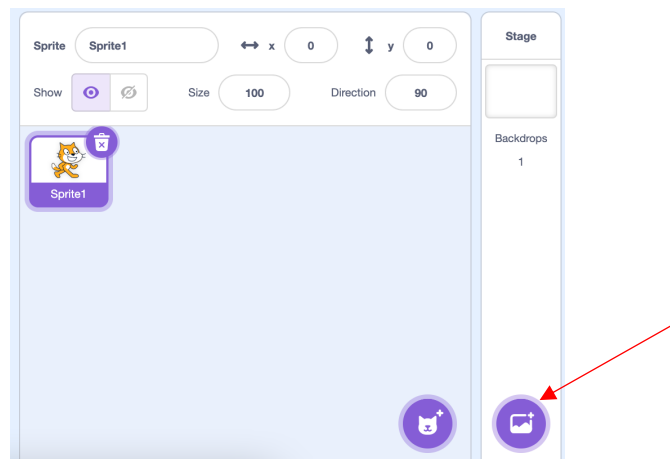
We would like our model to determine whether we “score” a goal or “miss” the net. Add one label for “score” and another for “miss”. We'll add training data later when we play our game.



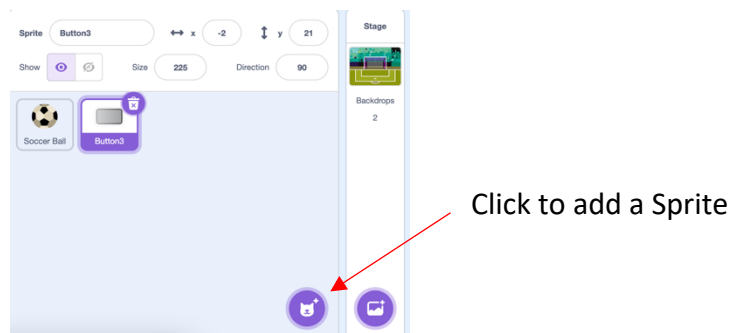
Click “Back to project” to return to the project page. On the project page, click “Make” and select Scratch 3. On the next screen, select “go straight into Scratch”.

Coding your game

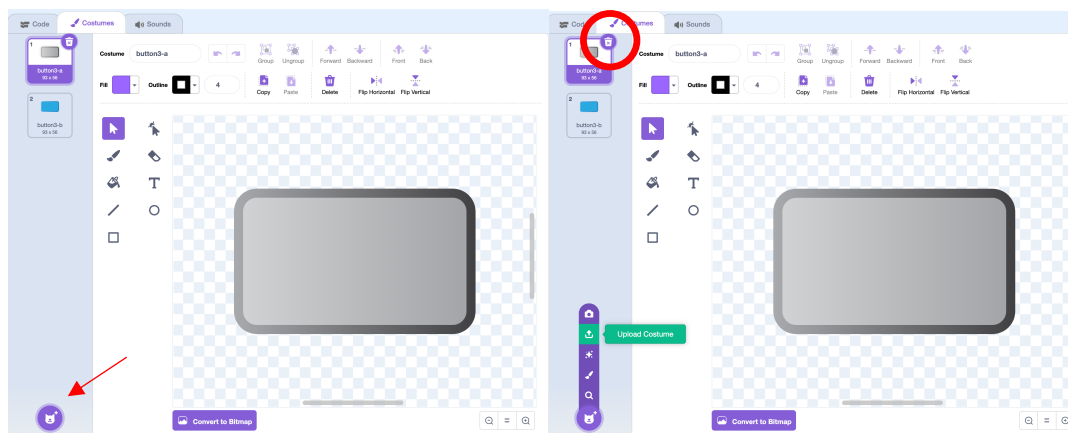
To start, we’ll code a soccer game! In the bottom right corner, click the “Choose a Backdrop” icon. This will take you to a list of backdrops. Select the Soccer backdrop.



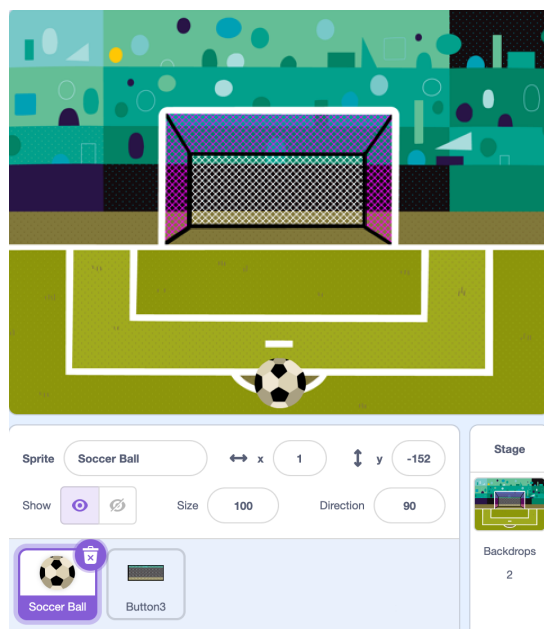
Now, we'll need to add some interactive components to our game. We'll need a ball and a target at which to kick the ball. To create the target, select Button3. Delete Sprite1. To create the target, select Button3. Delete Sprite1.



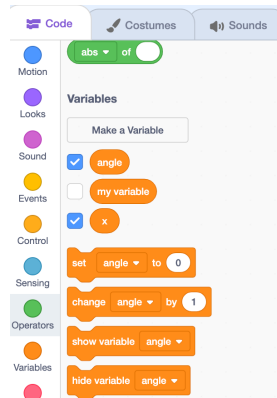
Then, click Button3 in the Sprite list and go to the Costumes tab. Click, "Choose a Costume". Upload the image **soccer_goal.png**. Delete any other costumes for Button3 by clicking the trash can in the upper right corner of the costume.



Now, change the size of Button3 to 50 and place it over the goal, so it blends into the scene. Move the soccer ball to the foot of the penalty box.



Let's make the ball move! Click on the "Soccer Ball" in the Sprite list. We'll first need to add two variables to track the ball's initial position and launch angle. Go to the Variables tab and click **Make a Variable** and create variables for all Sprites called **angle** and **x**.

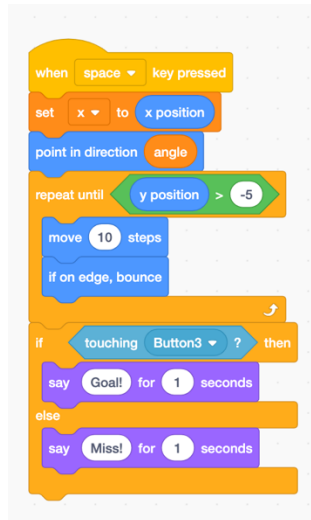


Now it's your turn! Write a block of code to kick the ball towards the goal. Use the description below to help structure your code.

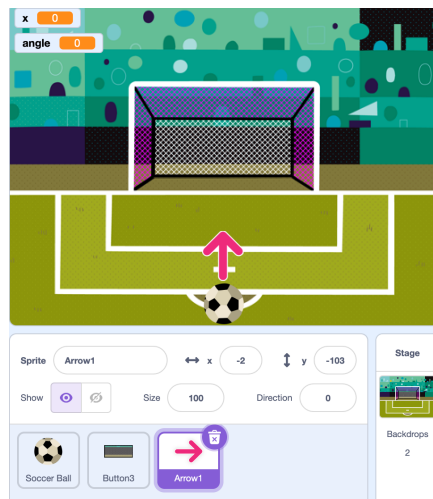
1. When the spacebar is pressed, we need to set our **x** variable to the initial x position of the ball.
2. We would like our ball to start moving with direction **angle**.
3. Then, we want our ball to move 10 steps at a time until it goes out of bounds or enters the goal (for the soccer backdrop, this occurs when $y > -5$).
4. For fun, we'd like the ball to bounce if it is on the edge of the screen.
5. **If** the ball is touching Button3, say "Goal!" for 1 second or **else**, say "Miss!" for 1 second.
6. Once the ball is moving, try changing the angle value with the "set angle to _" piece. Make sure to move the ball outside the penalty box before each try.

Hint: You might want to use the orange "repeat until" control piece to keep the ball moving.

Your code should look something like this.



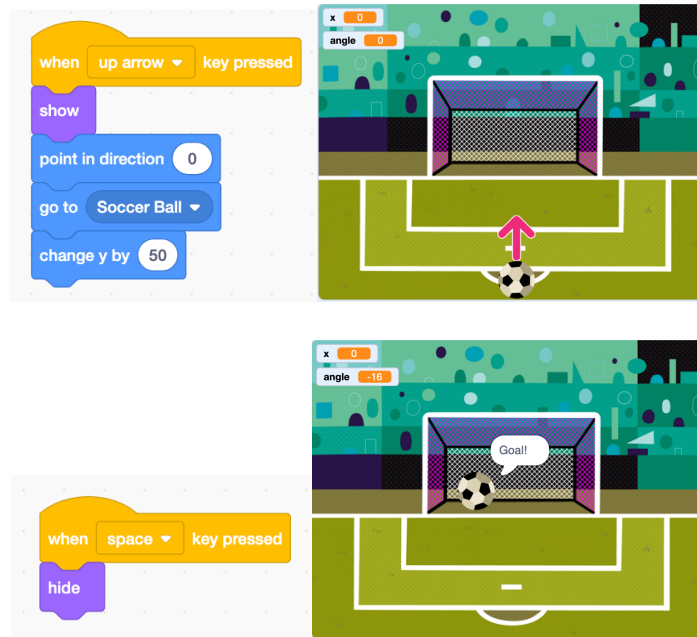
To make our game more fun, we'd like to control the direction in which we kick the ball without typing in a number every time. To do this, we'll need to add another Sprite called Arrow1. Drag the arrow above the ball and change its direction to 0.



Selecting Arrow1, add some code to change the direction of the arrow using the arrow keys on your keyboard. We then want to set our **angle** variable for the soccer ball to match the direction in which the arrow is pointing.



We would also like our arrow to disappear once the ball is kicked and to reappear before we kick our next goal. We can control these functions with our keyboard using the code blocks below.



To reset our game after each goal, we'll need to add code to both the Soccer Ball and Arrow1 Sprites. For Arrow1, add code to reset the direction of the arrow and the **angle** variable to 0.



Then, click on the Soccer Ball in the Sprite list and add code that will return the ball to the center of the penalty arc and reset the ball's direction. Also, we should reset our **x** variable to 0.

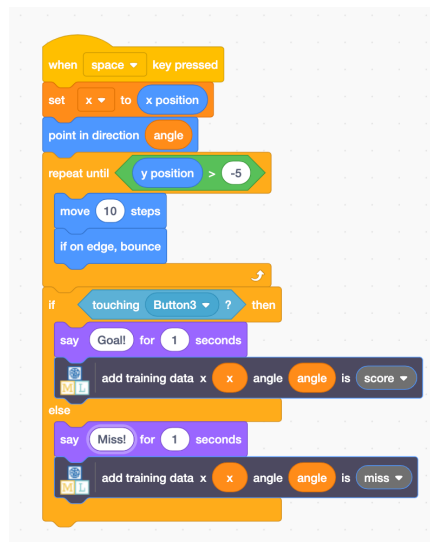


Hooray! Now we have a complete game! Take some time to test out your new game. You're welcome to make improvements and add fun features like sound effects. Be creative and have fun!

Training an ML model to play your game

Now that you're a soccer pro, let's teach the computer to play your game! Using **machine learning (ML)**, you can **train** (teach) your **model** whether a particular initial ball position and kicking angle will result in a goal or not.

Click the Soccer Ball Sprite and add your "score or miss" model pieces to your **if/else statement** to add the results of each kick to your model's training data.

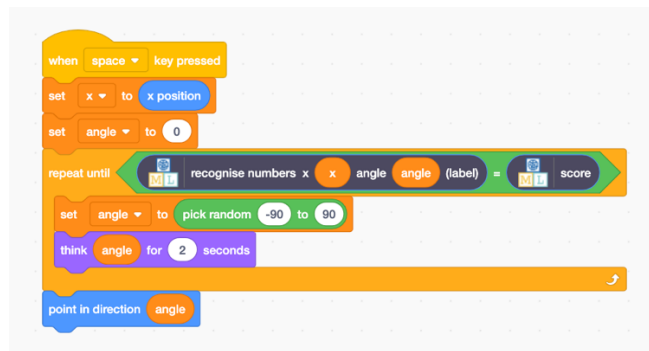


Now, collect training data by moving the ball to different x positions (be sure to stay behind the penalty box!). Start by collecting 20 examples of "score" and 20 examples of "miss". Go to the Machine Learning for Kids "Train" menu, and you should see the **x** and **angle** values for each attempt under each label. You might need to delete duplicates.

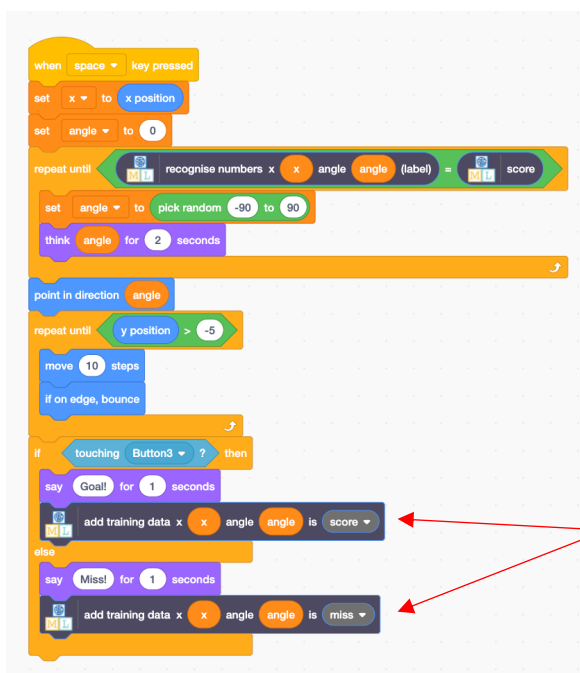
score					miss				
x 207 angle -49	x -168 angle 53	x 0 angle 26	x -112 angle 50	x 147 angle -21	x 130 angle -4	x -202 angle 0	x 188 angle -24	x 63 angle 38	x -162 angle 66
x -119 angle 35	x 0 angle -13	x 85 angle -17	x 172 angle -39	x 0 angle -31	x -87 angle -15	x 85 angle 23	x -220 angle 25	x 75 angle 22	x -186 angle 70
x 169 angle -52	x 109 angle -31	x 63 angle 0	x -93 angle 22	x 66 angle -31	x 0 angle 84	x 127 angle 71	x 141 angle 70	x 79 angle 72	x -108 angle -20
x 0 angle 0	x 0 angle -81	x 0 angle -31	x -165 angle -62	x -154 angle 40	x -152 angle 10	x -104 angle -26	x -133 angle -32	x 173 angle 15	
					x -107 angle 0				
+ Add example					+ Add example				
20					20				

It's time to test your model to see whether it can select the correct angle at which to kick the ball. Go back to your Scratch project. We'll need to add a new block of code that will randomly sample different launch angles for a particular starting position.

We first need to initialize our **x** variable to the ball's starting position and clear the previous value of **angle**. We'll then set the angle variable to random angles from -90 to 90 and repeat until the (**x**, **angle**) pair is recognized by our model as "score". Once the model chooses a direction, we'll point our ball in that direction.



You will need to remove the “when space key pressed” piece from your original code block that makes the ball move. (Bonus challenge: You can also “Make a Block” that allows you to assign a name to each block and call only the blocks you want to run.) For now, though, we will just copy our motion code from above.



This allows your game to keep training your model as the computer plays. Over time, you should see that the model gets better at soccer!

Now, try it out! Drag your ball to a position behind the penalty box and hit the space bar. Does your model find an angle that results in a goal? How many angles does it guess before it guesses correctly? Try to count how many times your model is correct. Is your model good at soccer?

If not, that’s okay! An important lesson in machine learning is that a model is only as good as the data used to train it. Just as you practice your favorite sport, your model will keep gathering training data to improve its performance.

Bonus challenges:

1. Automate your game by inserting your game code into an orange “repeat” piece and having the computer randomly select starting x positions for the ball.
2. Adapt the Arrow1 code to have the arrow point in the direction selected by the model.
3. Change the backdrop and Sprite costumes to turn your soccer game into a basketball, softball, or baseball game.

Have fun playing sports with code!