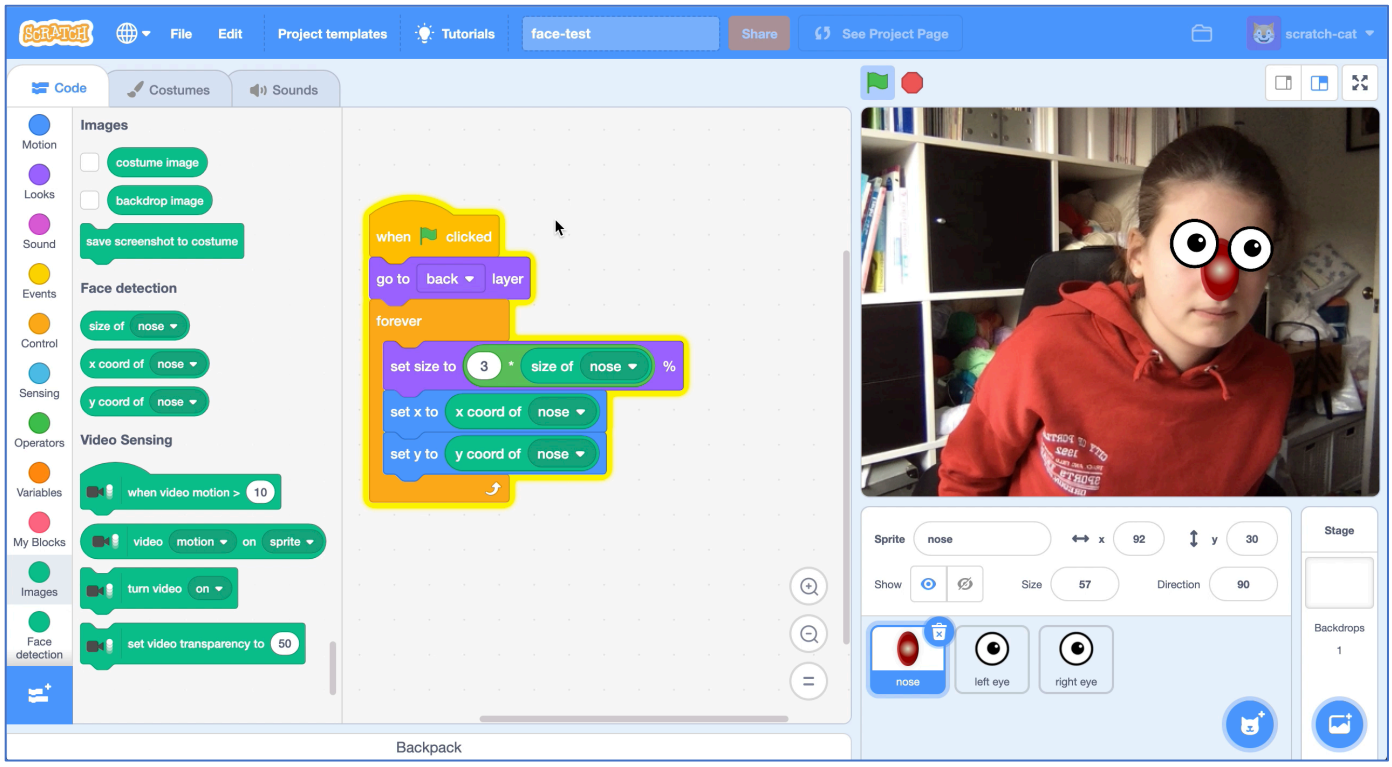
# Face Finder

In this project you will make an AI-powered face filter that adds cartoon eyes to your face.

You will use a pre-trained machine learning model to perform face detection on a live webcam video, and code animated effects using the results.



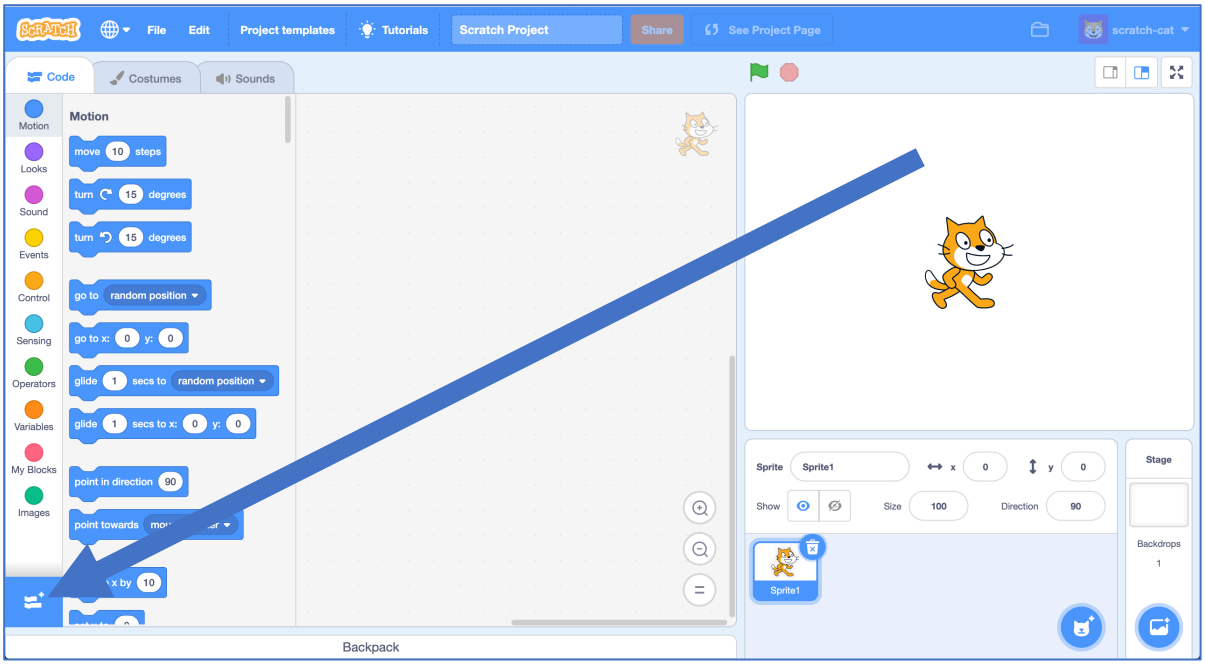
1. Go to https://machinelearningforkids.co.uk/pretrained/ in a web browser

*This page displays some of the pretrained machine learning models that are available to you. For this project, we'll be using the Face Detection model.*

2. Click on "**Get started**"

3. Open the **Extensions** window

*Click on the blue button with the plus icon in the bottom left.*




4. Click on the **Video Sensing** extension

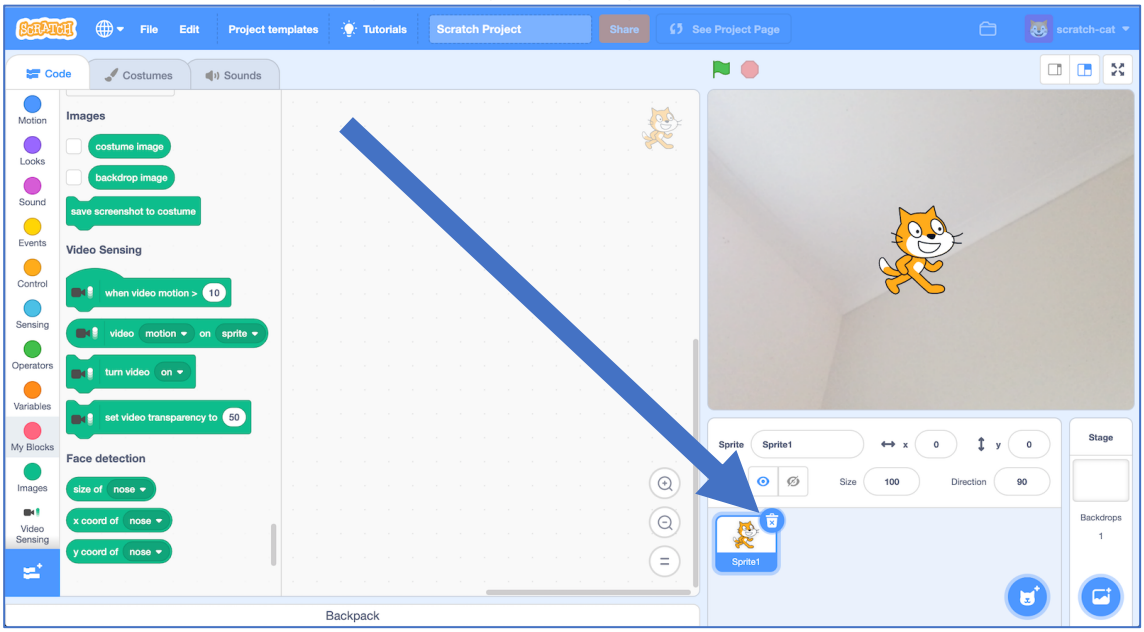
*You will need this extension to use the webcam in your project.*

5. Open the **Extensions** window again

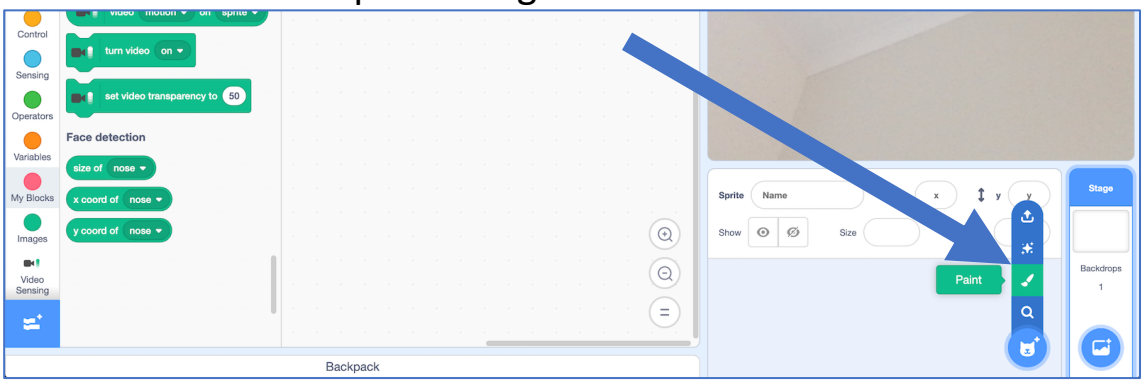
6. Click on the **Face detection** extension

*You will need this extension to use the pre-trained machine learning model that identifies the location of your face in the webcam feed.*

**7.** Delete the cat sprite by clicking on the trash can icon

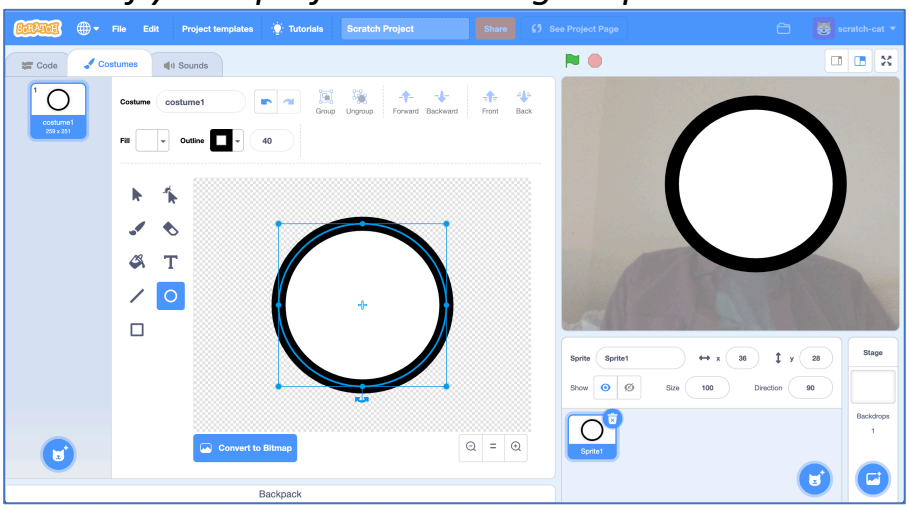



**8.** Create a new sprite using the **Paint** brush button



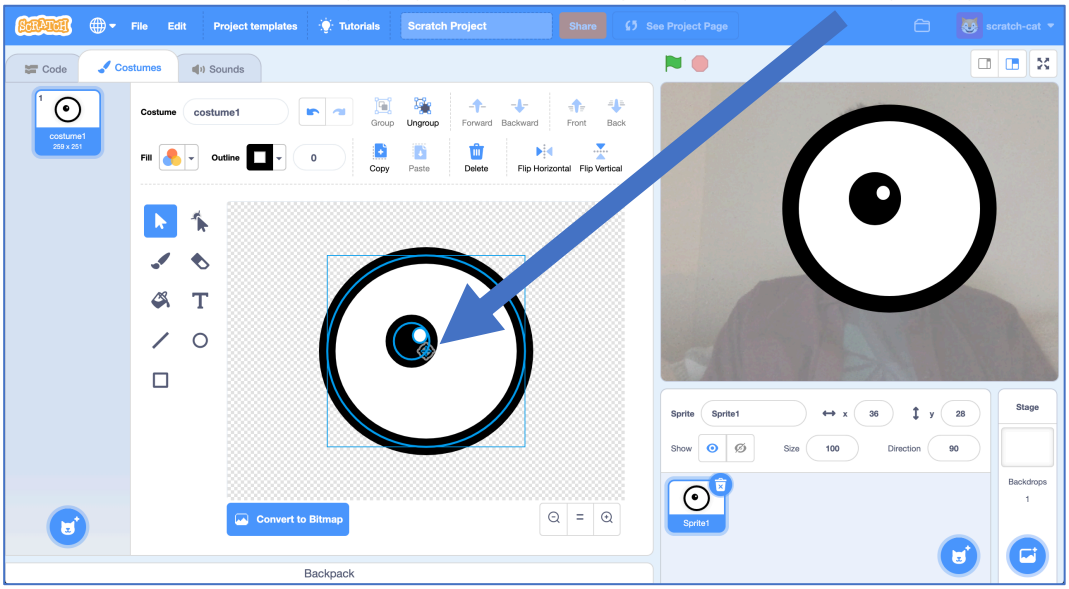

**9.** Draw a cartoon eye in the **Costume** tab

*You can draw it freehand if you like, or use the circle tool to draw two circles if you'd prefer something simpler.*



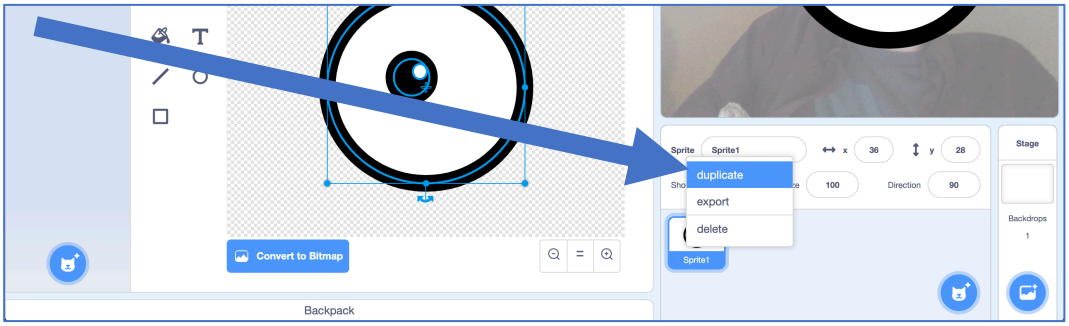
**10.** Drag your eye so the centre matches the sprite's centre crosshairs

*You should notice it jump into the right place when you get close.*

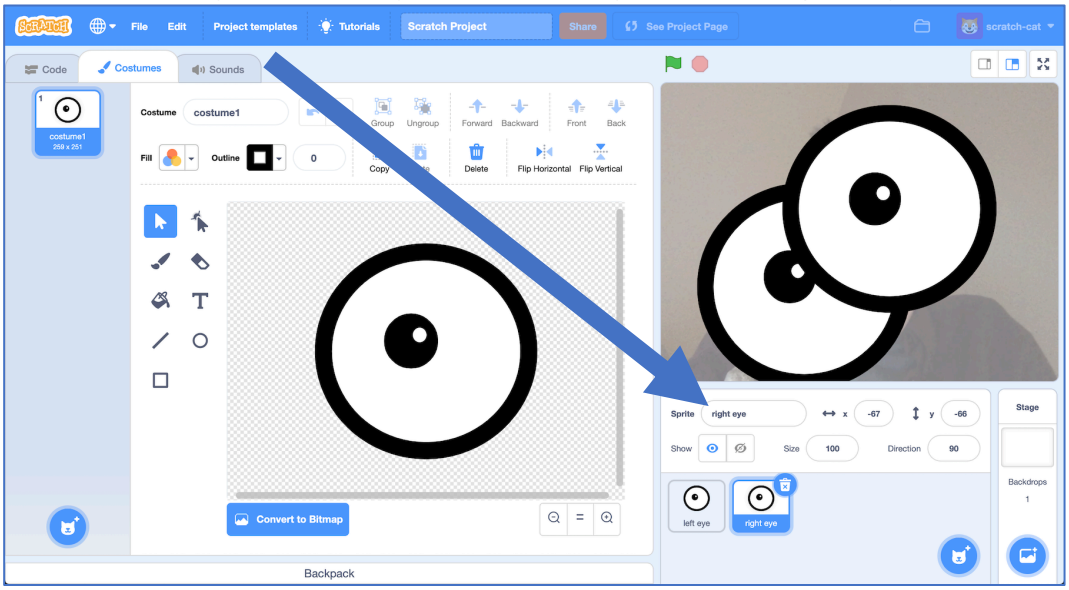



**11.** Duplicate your eye sprite

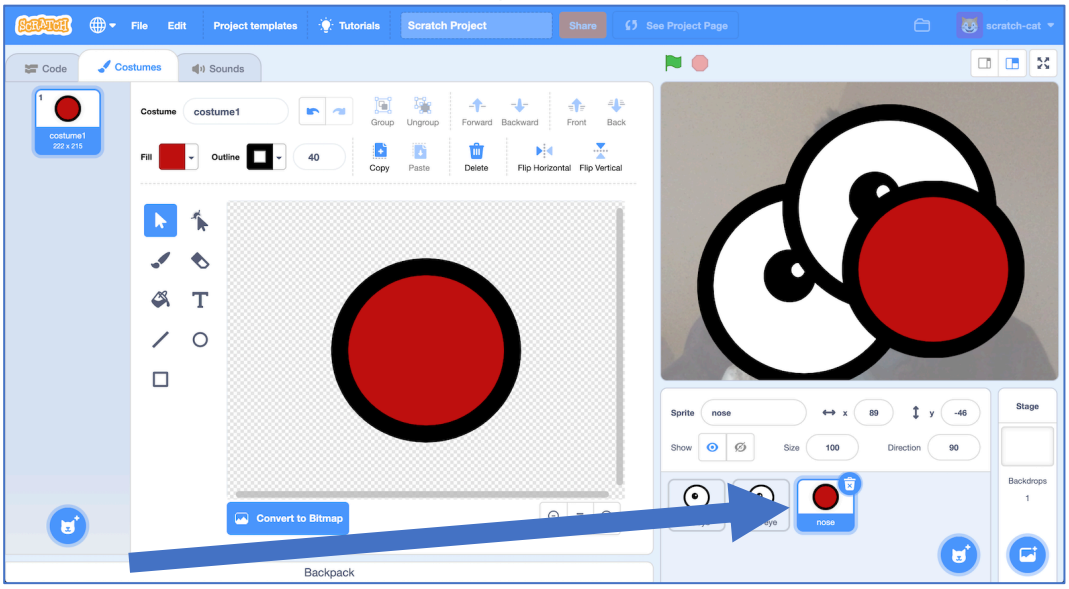
*Right-click on the sprite and then click on **Duplicate***



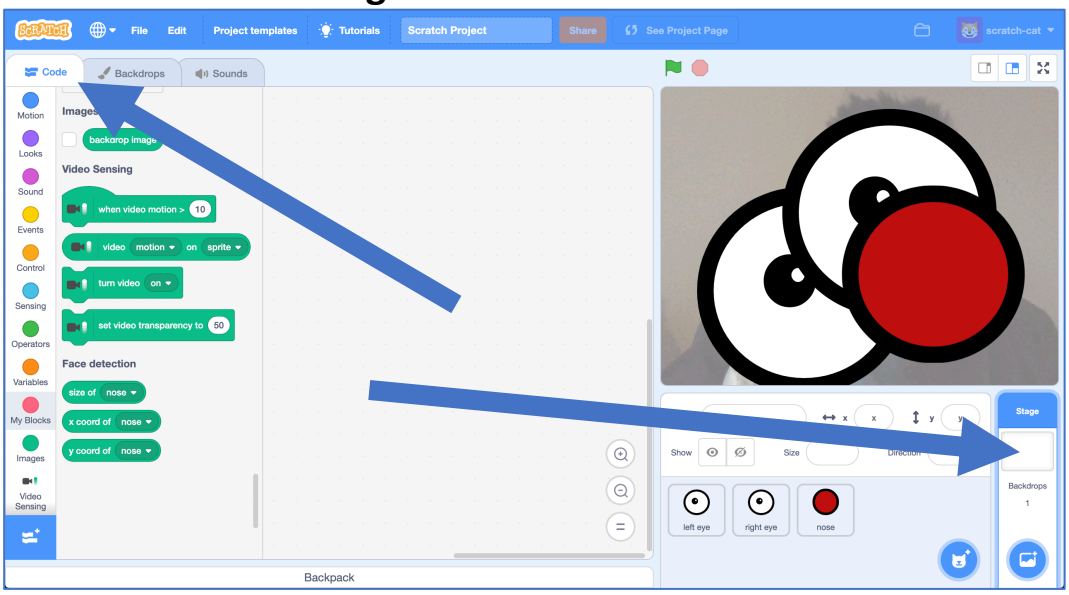

**12.** Name your two sprites "left eye" and "right eye"



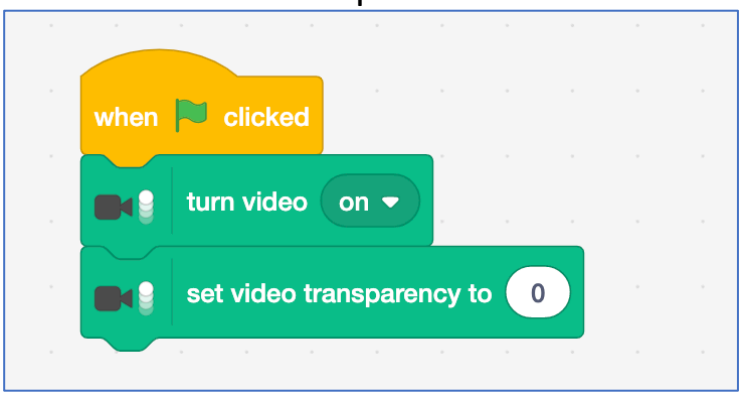
## 13. Use the **Paint** button again to draw a cartoon nose sprite



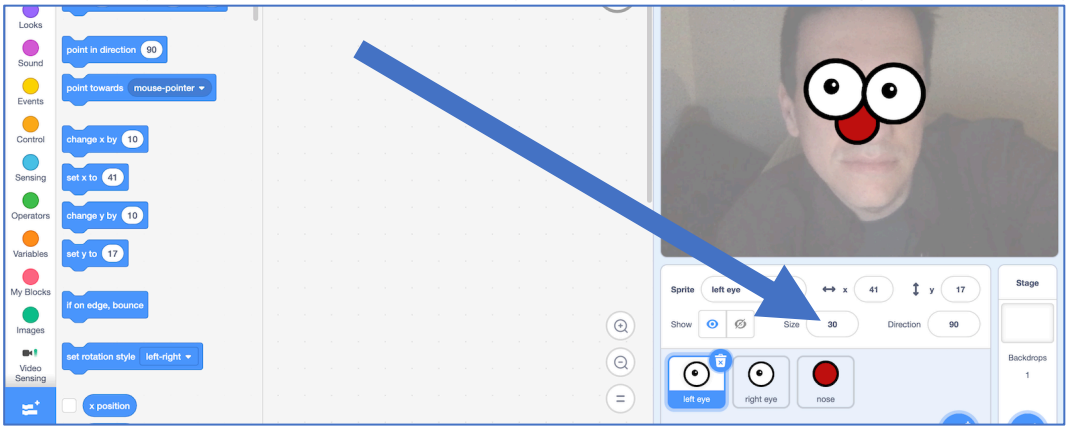

## 14. Click on the **Stage** and then click the **Code** tab



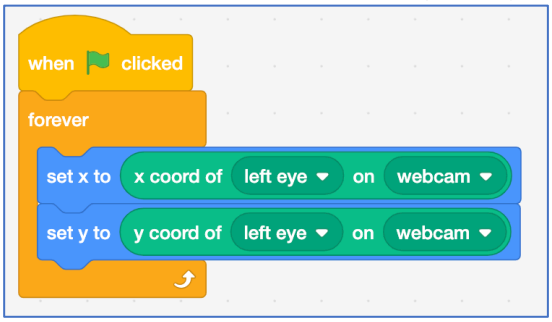

## 15. Create this script to enable the webcam



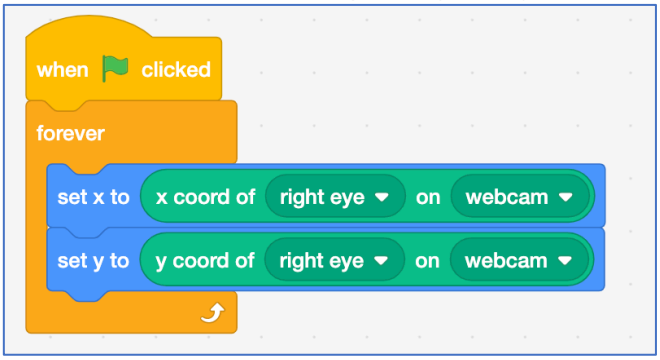
**16.** Adjust the size of your sprites so they are a good fit for your face



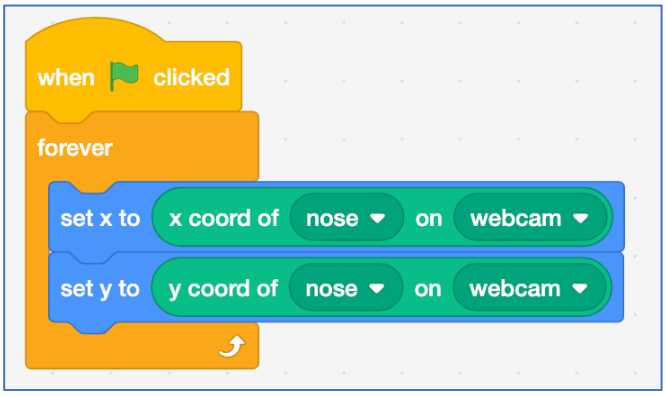

**17.** Click on the "**left eye**" sprite and create this script




**18.** Click on the "**right eye**" sprite and create this script




**19.** Click on the "**nose**" sprite and create this script



## 20. It's time to test! Click on the **Green Flag**

### What have you done so far?

You've made a project using a pre-trained machine learning model.

Over 32,000 photos were collected by academics at a university, who went through them all and noted the location of the 390,000 faces they found in them. All of those examples of what bits of photos look like faces were used to train a machine learning model how to recognize faces in photos.

Real-world machine learning projects often use models already trained by other people. It is a good way to quickly make a project when you don't have the time to collect your own training data.

## 21. Update your three scripts to change the size of the sprites

*This will update the size of your sprites based on the size of your face, so they'll get bigger if you move your face closer to the webcam. Experiment with different numbers until you're happy with how it is working (e.g. try 1, 1.5, 1.75, 2, etc.)*

**22.** Test your project again by clicking on the **Green Flag**

## What have you done?

You've made a Scratch project using a machine learning technique known as face detection: detecting the location of faces in photos.

There are two stages to how it does this.

First stage: "object detection". It finds the part of the photo that looks like it contains a face. Think of it as the computer drawing a box around where it predicts a face is.

The second stage: shape prediction. It predicts where the eyes, nose and mouth are most likely to be in the box drawn in the first stage. This is sometimes described as detecting "facial landmarks".

## How is this technology used?

What you're doing is **not** "facial recognition". Your project isn't recognizing whose face is in the photo. That is because the pre-trained model that you're using wasn't trained with photos of a particular person.

It is just looking for something that looks like a human face, because it has been trained with examples of photos of lots of different faces.

"Face detection" is a useful capability. You might've seen mobile apps use video face filters to add fun effects to video, like you did in this project.

Other real-world uses include being able to automatically blur people's faces in photos when you don't have permission to publish their faces, or automatically counting the number of people that a video camera can see.