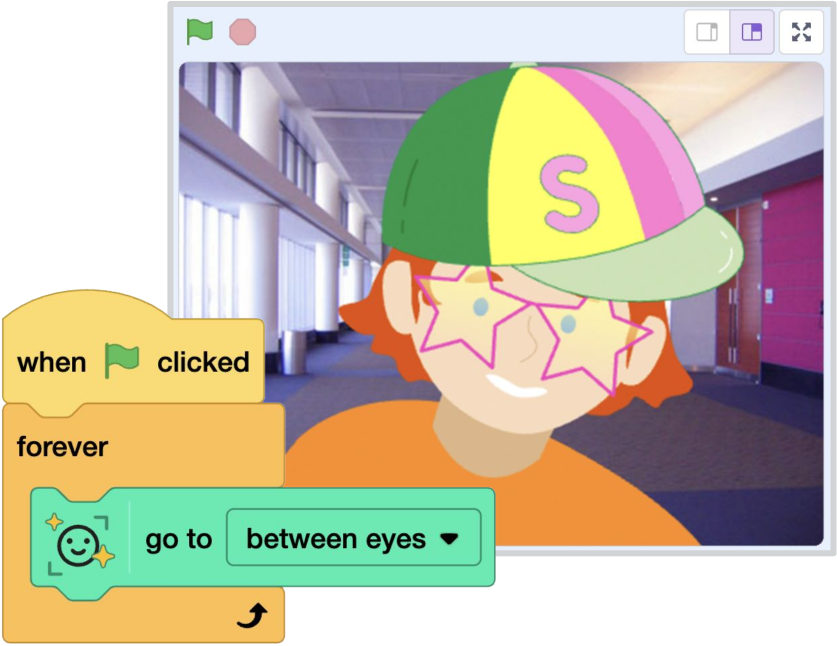
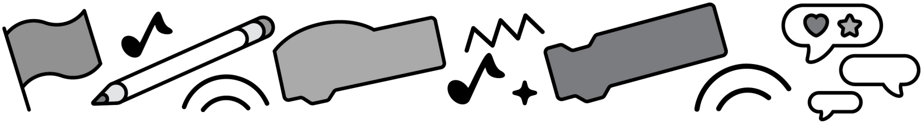
Scratch and AI: Face Sensing



```
when ⚑ clicked
forever
    go to between eyes ▾
```

Use AI face detection to make interactive Scratch projects

scratch.mit.edu SCRATCH™ Set of 10 cards



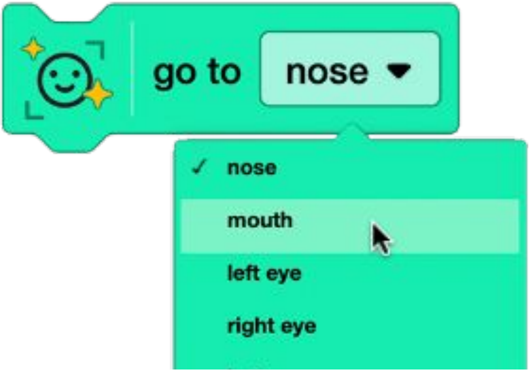
Cards in This Pack

- Try Out Face Sensing
- Create a Face Filter
- Create a Face Sensing Game
- Create a Face Sensing Sound Board
- Use Your Nose As a Pen
- Say It and Spray It
- Fool the AI
- Troubleshooting Tips and Facts
- Advanced Ideas to Try

Face Sensing blocks use a device's camera, so you may be prompted to give permission to use the camera in your browser. ***These blocks do not record or store your video.***

scratch.mit.edu **Set of 10 cards**

Try Out Face Sensing



This AI-powered extension uses a machine learning model to detect if it sees a face and where a nose, eyes, ears, mouth, etc., are. The model was trained using a large data set of images of faces.

Click a “go to nose” block while your face is visible on the stage. Did the sprite go to your nose? What happens if you choose another feature?

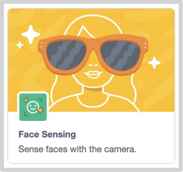
Try Out Face Sensing

scratch.mit.edu

GET READY

Choose any sprite.

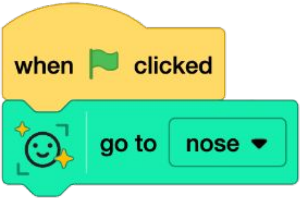
Add the Face Sensing Extension.



Face Sensing

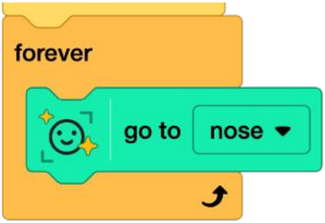
Sense faces with the camera.

ADD CODE



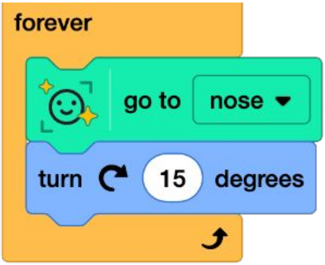
```
when ⚑ clicked
go to (nose ▾)
```

1. Add a “go to [nose]” block. Click it to see the result.



```
forever
    go to (nose ▾)
```

2. Next, add a “forever” loop to have the sprite stick to your chosen feature.

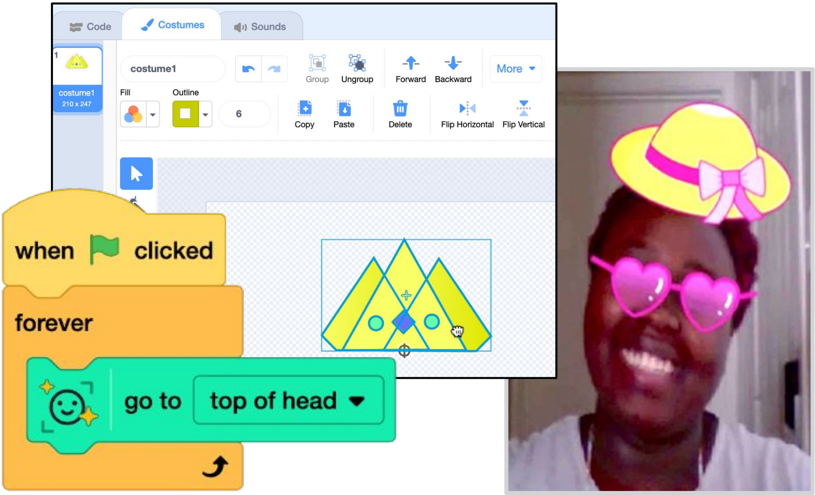


```
forever
    go to (nose ▾)
    turn ↻ (15) degrees
```

3. Try adding additional blocks from the Motion or Looks category to animate the sprite.

What if more than one face is on screen? The blocks can only interact with one face at a time.

Create a Face Filter



```
when ⚑ clicked
forever
    go to (top of head ▾)
```

- Use the Fashion sprites in the sprite library or draw your own hat, glasses, or other accessory with the **Scratch Paint Editor** tools.

- **Code a face filter** that follows your face as you move it around the stage.

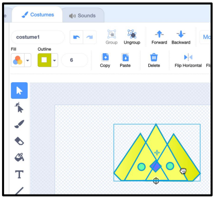
Face Filter

scratch.mit.edu

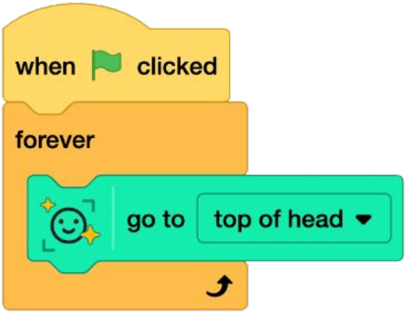
GET READY

Choose any sprite.

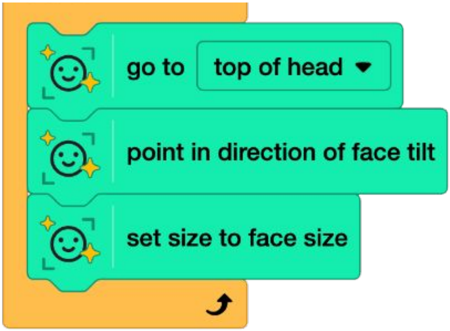
Or use “Paint” to create your own costume or two.



ADD CODE



```
when [flag] clicked
forever
    go to (top of head ▾)
```

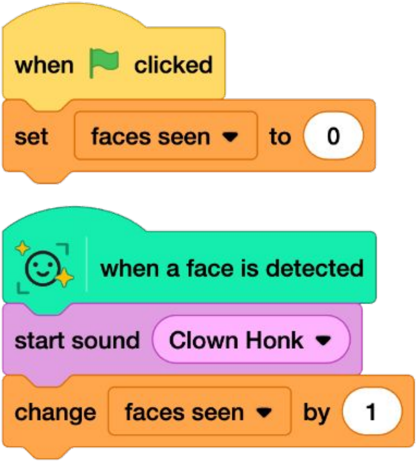


```
    go to (top of head ▾)
    point in direction of face tilt
    set size to face size
```

1. Add code so it sticks to a feature of your face, like the top of the head.
2. Experiment with other blocks to make the sprite scale to match the size of your face and point in the direction of your face.

Need to adjust the sprite’s position on your face? Try moving on the Costumes tab relative to the <u>costume center.</u> You could also add code to switch costumes.

Create a Face Sensing Game



```
when 🏳 clicked
set [faces seen ▾] to (0)

when a face is detected
start sound (Clown Honk ▾)
change [faces seen ▾] by (1)
```

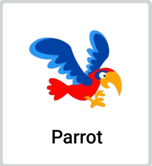
- Create a **face detection counter**
- Code a game that uses your face to **score points**, like a clicker game that uses your nose instead of a mouse
- Code a game that uses your face to **control a player sprite**, triggering jumps or helping it move around the stage or point in a direction

Face Sensing Game

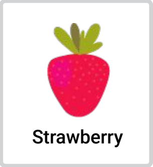
scratch.mit.edu

GET READY

Choose a first sprite.

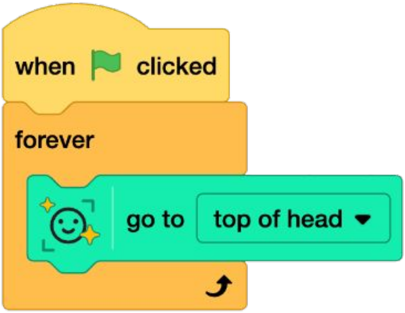


Choose a second sprite.



ADD CODE

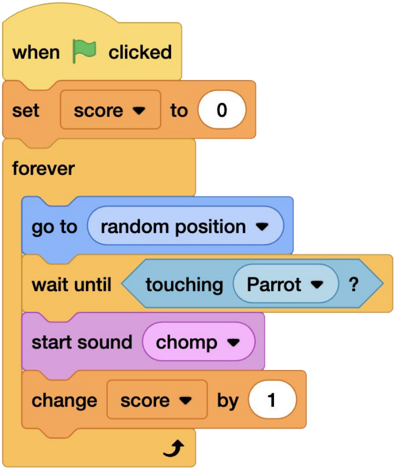
1. Add code to the first sprite so you can control it with your face. This will be the player.



```
when ⚑ clicked
forever
    go to (top of head ▾)
```

2. Create a score variable to track points. Don’t forget the program will need to reset it each time a new game is started.

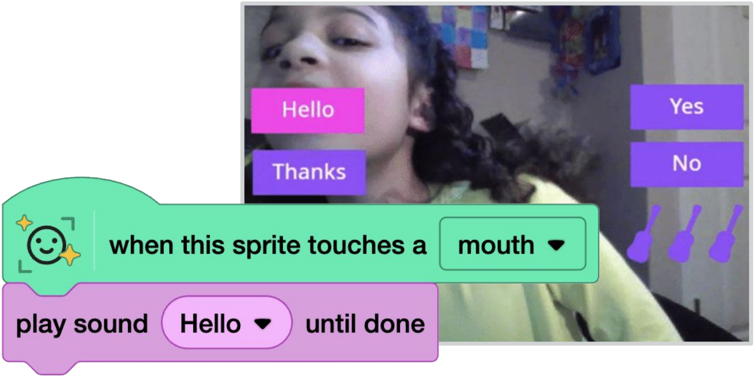
3. Add code to the second sprite so it moves to a random position on the stage and gives the player a point when sprites touch.



```
when ⚑ clicked
set (score ▾) to (0)
forever
    go to (random position ▾)
    wait until <touching (Parrot ▾) ?>
    start sound (chomp ▾)
    change (score ▾) by (1)
```

Try making the game harder by moving the second sprite after a time, if not touched yet.

Create a Face Sensing Sound Board



```
when this sprite touches a [mouth ▾]
play sound (Hello ▾) until done
```

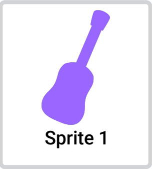
- **Choose a variety of fun sounds** or record your own.
- **Code a sound board** that uses parts of your face to play sounds.
- **Code effects** that add visual confirmation, like color changes.
- Create a **mystery button** that picks a sound to play at random.

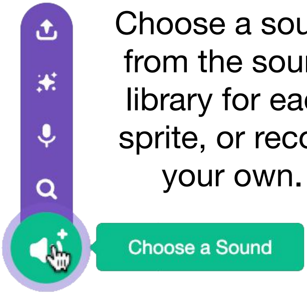
Sound Board

scratch.mit.edu

GET READY

Choose a few sprites, or draw your own.

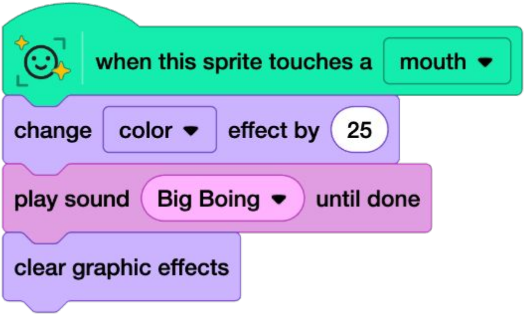




Choose a sound from the sound library for each sprite, or record your own.

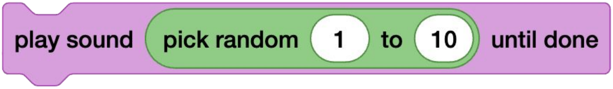
ADD CODE

1. Add code to each sprite to play a sound, change an effect, or perform another animation when parts of your face touch them.



```
when this sprite touches a [mouth ▾]
change [color ▾] effect by (25)
play sound (Big Boing ▾) until done
clear graphic effects
```

2. Try adding multiple sounds to a sprite. Use the “pick random” operator so each time is a surprise.



```
play sound (pick random (1) to (10)) until done
```

Use Your Nose As a Pen

- Combine blocks from the Face Sensing and the **Pen Extension** for a wild way to draw!

- Stick a sprite to your nose or eyes, put the **pen down**, and move it around the stage to create a unique art piece.

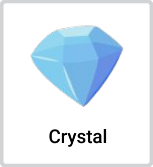
- Want more control? Try using your head **tilt to put the pen up and down**. Or try adjusting the **pen size based on your face size.**

Use Your Nose As a Pen

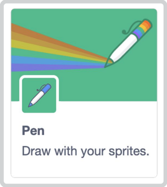
scratch.mit.edu

GET READY

Choose any sprite to act as the Pen.

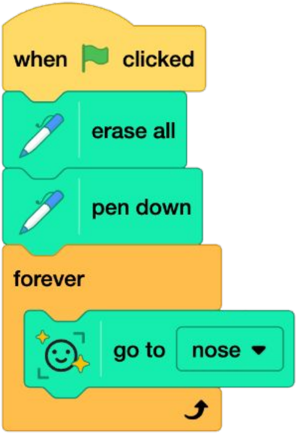


Add Pen Extension.



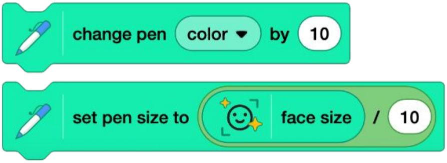
ADD CODE

1. Add a Pen block to put the pen down. Then have the pen follow your nose.



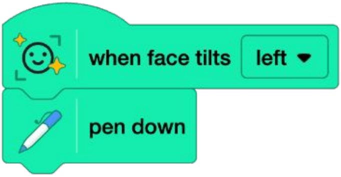
```
when ⚑ clicked
erase all
pen down
forever
    go to nose ▾
```

2. Try variations like changing the pen color or setting the pen size based on your face size.

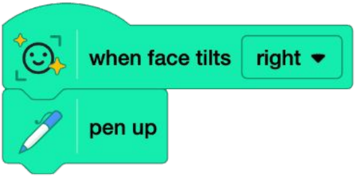


```
change pen color ▾ by 10
set pen size to (face size / 10)
```

3. Want more control? Use “when face tilts” to control when the pen is up and when it is down.

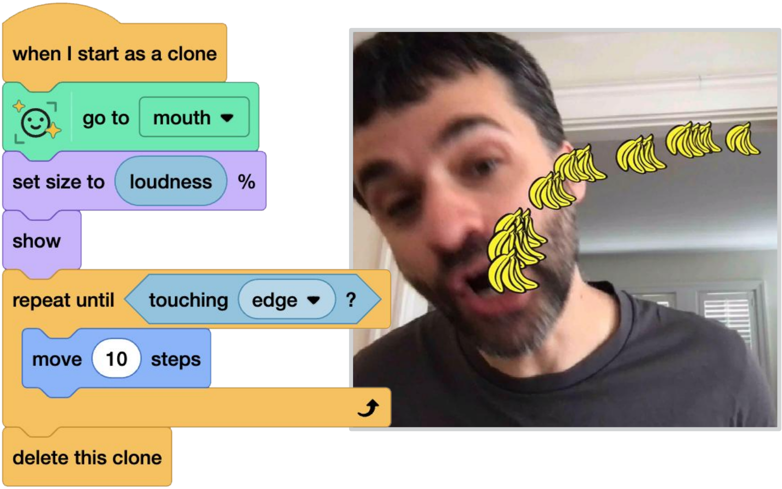


```
when face tilts left ▾
pen down
```



```
when face tilts right ▾
pen up
```

Say It and Spray It



```
when I start as a clone
go to mouth ▾
set size to (loudness) %
show
repeat until <touching edge ▾ ?>
    move (10) steps
end
delete this clone
```

- Code sprite **clones** to spray from your mouth!
- Control the size and visibility of clones using the "**loudness**" block.
- For even more fun, set up different sprite **costumes that are chosen at random**, so what sprays out is a continual surprise.

Say It and Spray It

scratch.mit.edu

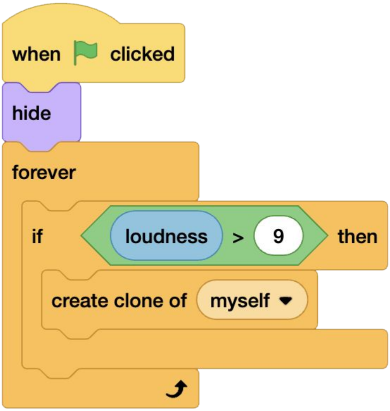
GET READY

Choose any sprite.

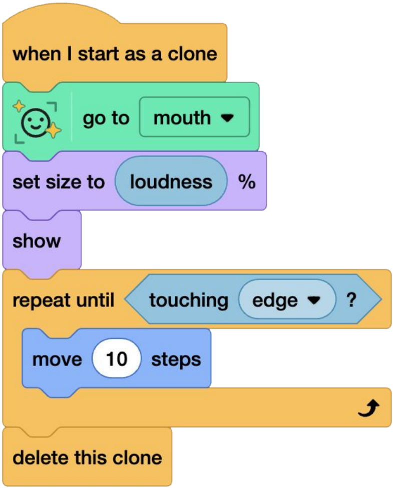
ADD CODE

1. To have multiple copies of a sprite spray from your mouth, you can create clones and use “loudness” to trigger their creation.

2. Have clones go to your mouth, then move repeatedly. You could delete clones when they reach the edge of the stage, and set or change size based on loudness.



```
when ⚑ clicked
hide
forever
    if <(loudness) > (9)> then
        create clone of (myself ▾)
    end
end
```



```
when I start as a clone
go to (mouth ▾)
set size to (loudness) %
show
repeat until <touching (edge ▾) ?>
    move (10) steps
end
delete this clone
```

Fool the AI

Face Sensing blocks try to detect if a face exists, but they are not able to identify who the face is, or even if it is a human face! That means sometimes the AI makes interesting mistakes. Identifying these mistakes can help us see the difference between our own human intelligence and AI.

Can the AI find the parts of a face if:

- you are in disguise, your face is covered, or your face is tilted or upside down?
- the lighting in the room is very bright or very dark?
- you step out of frame and hold up a drawing of a smiley face? a stuffed animal? a pet? two googly eyes attached to fingertips? or another facelike object made of different materials or from nature?

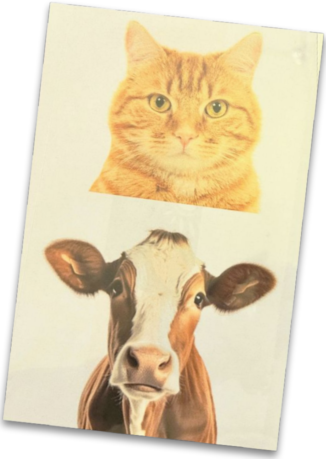
What variables can you change to try to fool it into thinking it sees a face? What limitations can you find?

Fool the AI

scratch.mit.edu

The AI that we're using is trained to detect human faces.

False positives are things that are not actually a human face (false) that it detects as a face (positive). What are some false positives you can find?

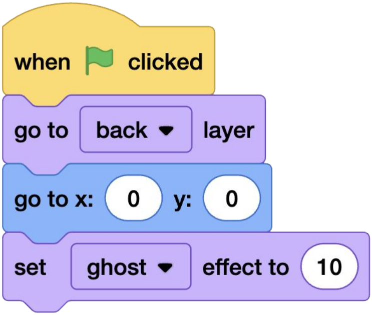


Why might some animal faces (like cats) be recognized while others (like cows or dogs) may not be seen as a face? Think about the proportions of a human face and how those are different from a cow's longer snout.

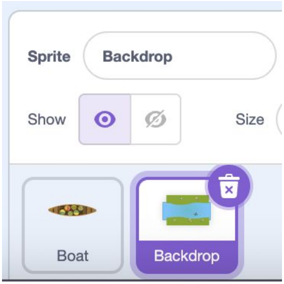
A **false negative** when working with the Face Sensing blocks is when a human face is in the frame, but the AI does not detect it (perhaps because it is too small or upside down). What are some false negatives you can find? What might be causing their detection failure?

Troubleshooting Tips and Facts

What if I want to hide my face/hide the visible video feed without removing the Face Sensing blocks? Any “Backdrop” created in the Sprite Area on your Face Sensing project will be hidden by default and replaced with your video feed, but you can create backdrops/ backgrounds as sprites instead. If the backdrop sprite covers the entire screen, none of your video feed will be visible, but the program will still be able to “go to nose” or recognize face tilt and face size. You can also use the “set ghost effect” block if you want your video feed somewhat visible through the backdrop.



```
when ⚑ clicked
go to back ▾ layer
go to x: 0 y: 0
set ghost ▾ effect to 10
```



Sprite: Backdrop

Show ● ⊘   Size

Troubleshooting Tips and Facts

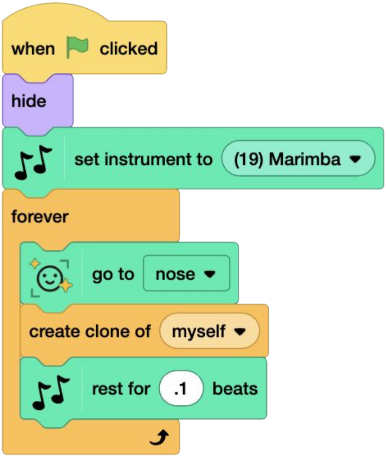
If I use the "Set Thumbnail" button while my face is on-screen, will it capture an image of my face? No. When using the Face Sensing extension, none of your data is stored or sent to Scratch or any other site, and your camera view is not captured when setting your project thumbnail, so your data remains private.

I changed my mind about using Face Sensing blocks in my project, but the video feed is still on my screen. How can I get rid of it? Remove all of the extension's blocks from the script area (drag blocks to the block palette to delete). Make sure to turn off any stage monitors by ensuring all checkboxes are unchecked in the blocks palette (next to blocks like "face tilt"), too. Save your project. Then, reload it by refreshing the browser or closing and reopening the project.

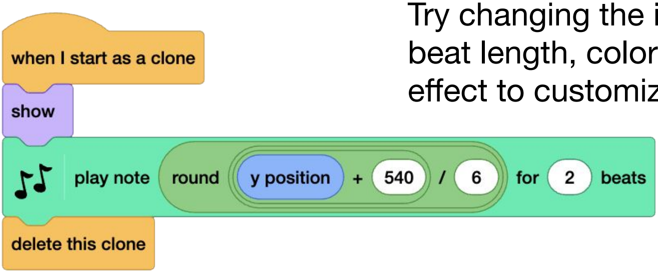
Advanced Ideas to Try

Nose Piano

Remix one of our starter projects, like “Musical Droplets,” and try using your nose, instead of a mouse, to play notes.



```
when ⚑ clicked
hide
set instrument to (19) Marimba
forever
    go to nose
    create clone of myself
    rest for .1 beats
```



```
when I start as a clone
show
play note round ((y position + 540) / 6) for 2 beats
delete this clone
```

Though “nose x” and “nose y” reporter blocks aren’t available, you can create a sprite that follows your nose and use the sprite’s x and y position in your code.

In this example, a clone is created periodically and its position determines the note played.

Try changing the instrument, beat length, color, size, or ghost effect to customize.

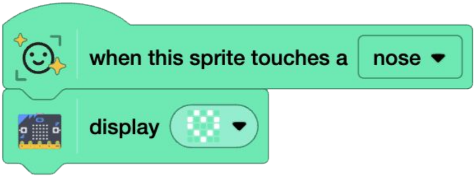
Or remix starter projects like “Musical Buttons” and “Piano” with “when this sprite touches a nose.”

Advanced Ideas to Try

Communication Device

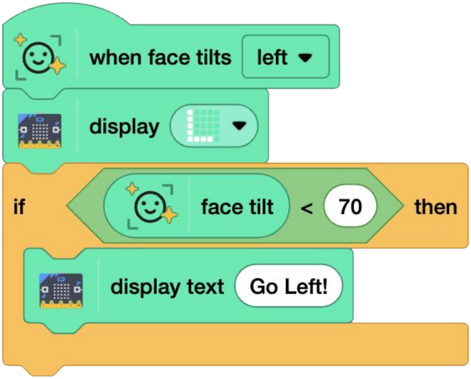
Use “when this sprite touches a nose” or “when face tilts” to send pre-programmed messages.

Connect to a micro:bit and text messages can be displayed by the LED panel. Connect to text to speech blocks and messages can be audible.

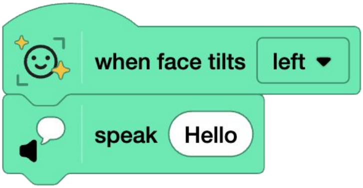


```
when this sprite touches a [nose ▾]
display [ ▾]
```

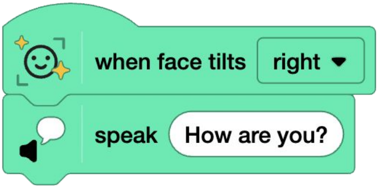
Install the micro:bit or text to speech extensions to try.



```
when face tilts [left ▾]
display [ ▾]
if <(face tilt) < (70)> then
    display text (Go Left!)
```



```
when face tilts [left ▾]
speak (Hello)
```



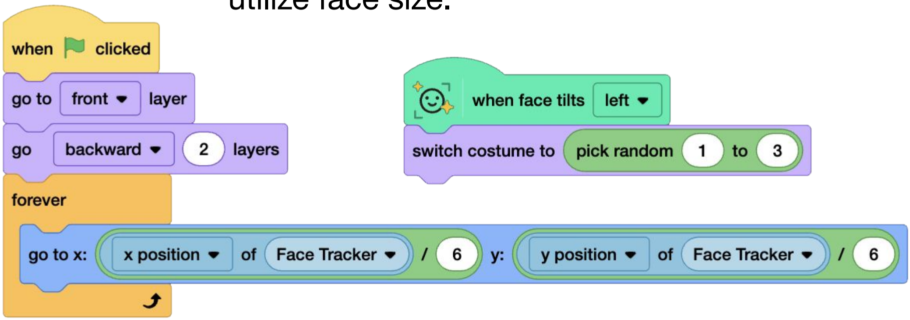
```
when face tilts [right ▾]
speak (How are you?)
```

Advanced Ideas to Try

Parallax

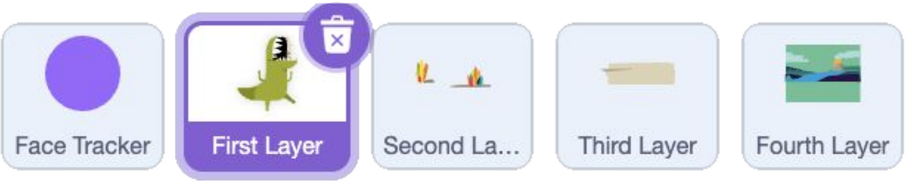
A parallax is where the movement of the controller (a mouse, a nose…) adjusts the viewer's perspective.

Though "nose x" and "nose y" reporter blocks aren't available, you can create a sprite that follows your nose and use the sprite's x and y position in your code for each parallax layer. Add variety with costume changes triggered by face tilts. Or utilize face size.



```
when ⚑ clicked
go to front ▾ layer
go backward ▾ 2 layers
forever
    go to x: ((x position ▾ of Face Tracker ▾) / 6) y: ((y position ▾ of Face Tracker ▾) / 6)
↺
```

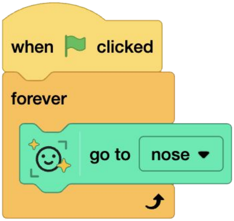
```
when face tilts left ▾
switch costume to (pick random 1 to 3)
```



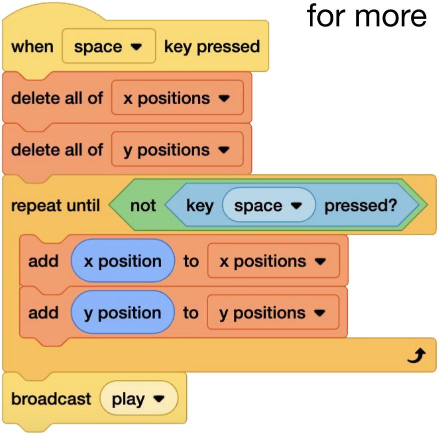
Advanced Ideas to Try

Animation Recorder

The entertainment industry often uses motion capture of actors for a more realistic feel to their computer animated character's movement. Try recording the movements of a sprite that follows your face, by storing its x and y positions in a list. Then, have the animation play back.



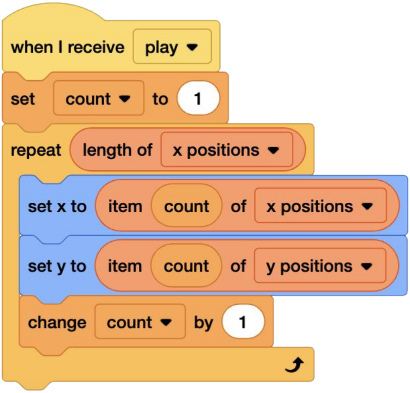
```
when ⚑ clicked
forever
    go to (nose ▾)
end
```



```
when (space ▾) key pressed
delete all of (x positions ▾)
delete all of (y positions ▾)
repeat until <not <key (space ▾) pressed?>>
    add (x position) to (x positions ▾)
    add (y position) to (y positions ▾)
end
broadcast (play ▾)
```

Create lists to store x and y positions repeatedly while a key is pressed. Use a counter variable to iterate through the lists when playing back.

See our cards on Variables and Lists for more information.



```
when I receive (play ▾)
set (count ▾) to (1)
repeat (length of (x positions ▾))
    set x to (item (count) of (x positions ▾))
    set y to (item (count) of (y positions ▾))
    change (count ▾) by (1)
end
```