



Create a quick game to help people learn what can be 'identifying information' in photos.

INFORMATION

What you will learn

- Using hide and show blocks
- Using sounds
- Using a score system
- Extension - creating scrolling text

What you will need

HARDWARE

A computer capable of running Scratch 3

SOFTWARE

Scratch 3:
either online - [rpf.io/scratchon](https://scratch.mit.edu/)
or offline - [rpf.io/scratchoff](https://scratch.mit.edu/)

Project Links

STARTER PROJECT

<https://scratch.mit.edu/projects/1330291808/>

COMPLETED PROJECT

<https://scratch.mit.edu/projects/1330291951/>

Additional notes

To learn more about staying safe online, explore the range of resources from the [eSafety Commissioner](#), including [eSafety Kids](#).

To learn more about the project read our [blog](#).



CHALLENGES

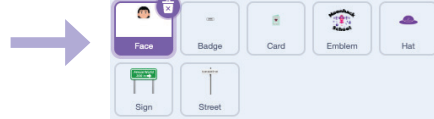
- Create more levels. Will each level get more difficult? Will it have harder items to find? Will it be different photo locations and events?
- Photos contain data inside them such as time, date, and location. Could this become part of the project?
- Add game elements like a timer.
- Can you animate parts of the projects to act as distractors?

STEP 1 - SETTING UP THE PROJECT

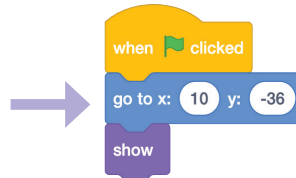
Start by opening the starter project - <https://scratch.mit.edu/projects/1330291808/> The first step is to let the player know how to play the game.



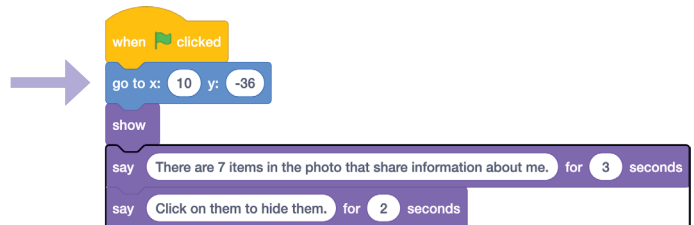
Select the Face sprite



Start an algorithm with the event **when green flag clicked**. Add a motion block to set its **starting location**. Add a looks block to **show** the sprite on screen.



Add 2 **say blocks** that will tell the player how to play the game. Having 2 blocks separates the text. You could also choose to record your own voice saying the instructions.

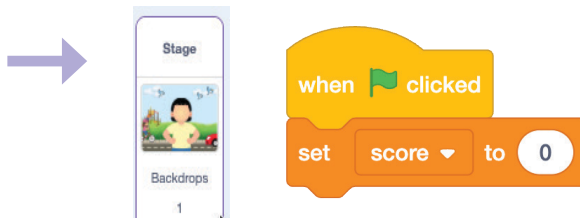


Test your project and make sure the speech bubbles work.

STEP 2 - CREATING A SCORING SYSTEM

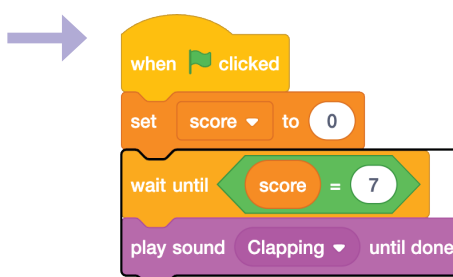


The score code block will be attached to the **backdrop**. Click on the Stage. Start with the **green flag** event. Create a new variable and call it score. Add a **set score** block.



A score of 7 means the player has found all of the items hidden in the photo.

Add a **wait until** control block. Use the **equals operator** to create the condition of checking the score **variable**. Then add a play sound block. You can choose any sound that you like.



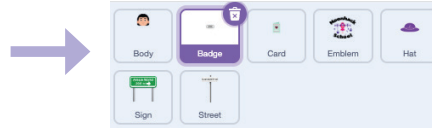
You can test your project again, but you won't see the result of the coding you've just done. This is because we haven't added code to award points.

Will you choose to leave the score on your screen or hide it?

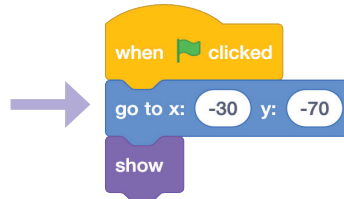
STEP 3 - ADDING THE ITEMS

Each item that the player has to identify is a separate sprite. Each sprite will have the same algorithms with some small differences.

- Select the badge sprite. The first algorithm will control the sprite at the start of the project.



- Start an algorithm with the event **when green flag clicked**. Add a motion block to set its **starting location**. Add a **show** block from the looks menu. We need the sprite to be on the screen when the game begins.



- The second algorithm will control the sprite when it is found by the player.
Start with the event **when this sprite clicked**. Add a **hide** block. Add the **play sound** block. You can keep this sound or choose your own.



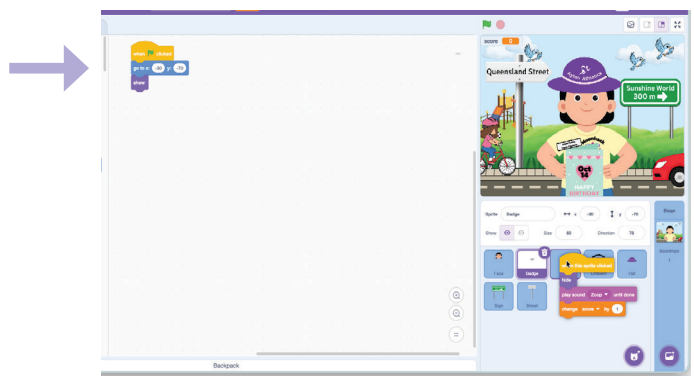
- Finish the algorithm by adding a **change score block**. When the sprite is 'selected' this will add 1 point, working towards collecting 7 points to finish the game.



Test your project by clicking the green flag and then click on the badge. Does it disappear? Does your score change by 1?

- For each of the other six sprites, they need to have the same 2 algorithms: one that controls how to start the game and one that controls what to do when the sprite is clicked.

You can create these algorithms on each sprite, or you can copy your algorithm across to each sprite by dragging the code on top of the other sprites.

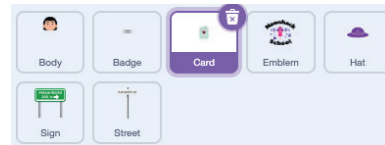


- Did you know?** There are different ways you can copy algorithms to other sprites. You can drag it on top or you can drop it into your backpack (at the bottom of your screen) and then drag it to other sprites.

Your algorithms for the Card sprite. The card sprite also needs a **go forward** block so that it sits on top of the emblem and badge.

```

when green flag clicked
  go to x: 5 y: -125
  show
  go forward 2 layers
  
```



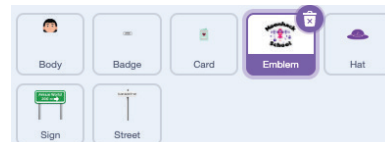
```

when this sprite clicked
  hide
  play sound Zoop until done
  change score by 1
  
```

Your algorithms for the Emblem sprite. A school emblem gives identifying information about where you go to school.

```

when green flag clicked
  go to x: 25 y: -70
  show
  
```



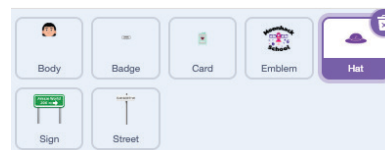
```

when this sprite clicked
  hide
  play sound Zoop until done
  change score by 1
  
```

Your algorithms for the Hat sprite. A sports club emblem lets people know where you spend your spare time. This also needs to sit on the top layer of the sprites.

```

when green flag clicked
  go to x: 5 y: 90
  show
  go to front layer
  
```



```

when this sprite clicked
  hide
  play sound Zoop until done
  change score by 1
  
```

Your algorithms for the Sign sprite. A sign that directs people to a landmark gives information about the area that you live in.

```

when green flag clicked
  go to x: 160 y: 20
  show
  
```



```

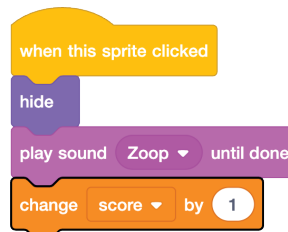
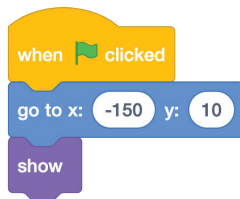
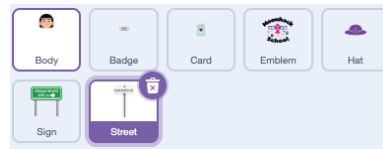
when this sprite clicked
  hide
  play sound Zoop until done
  change score by 1
  
```



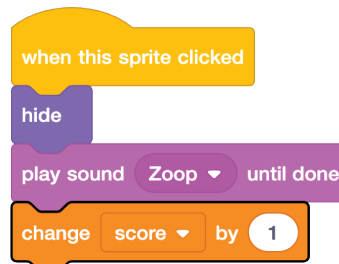
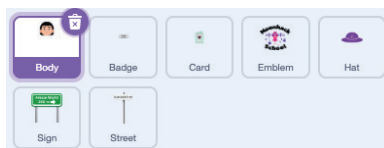
Now is a good time to test your program so far, if you haven't been testing after each sprite. Make sure each sprite is on screen when the green flag is clicked, and hides when clicked.



Your algorithms for the Street sprite. A street sign indicates where you live.



Select the Face sprite. The algorithm to place the sprite is already in place. Add the algorithm to hide it when clicked.

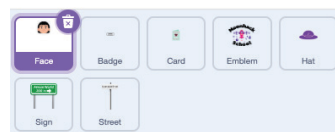


Test your program to know that all 7 sprites are doing what they should. When you collect all 7, did your final sound play? You can finish the project here.

STEP 4 - EXTENSION, SCROLLING SCRIPT (OPTIONAL)



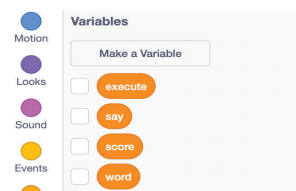
Select the Face sprite.



Find your algorithm that has the 2 say blocks and delete these 2 blocks.



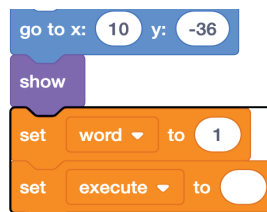
Create 3 new **variables** called say, word, and execute.



word - will keep track of the letter it is up to
execute - stores all the letters that have been collected



Add blocks to **set word** to 1 so it will begin with the first letter. **Set execute** to be empty as we have not collected any of the letters yet.



This block is tricky and made up of multiple parts. The say variable will join the execute and word variables of the text. Every time the loop runs, the program will find one letter from the message, add it to the letters already collected, update the message the sprite is saying, and then save that message. This keeps repeating until the message is complete.

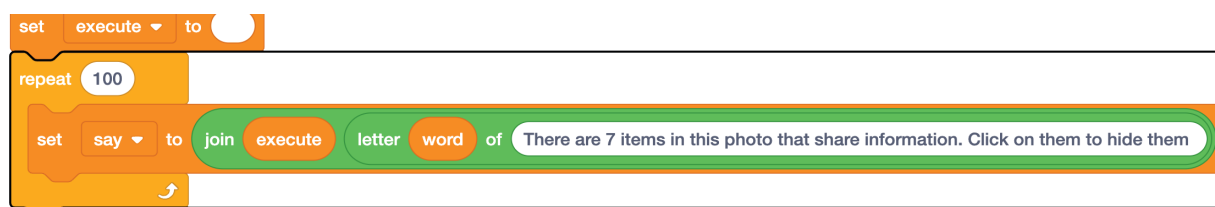
Loop 1 - T

Loop 2 - Th

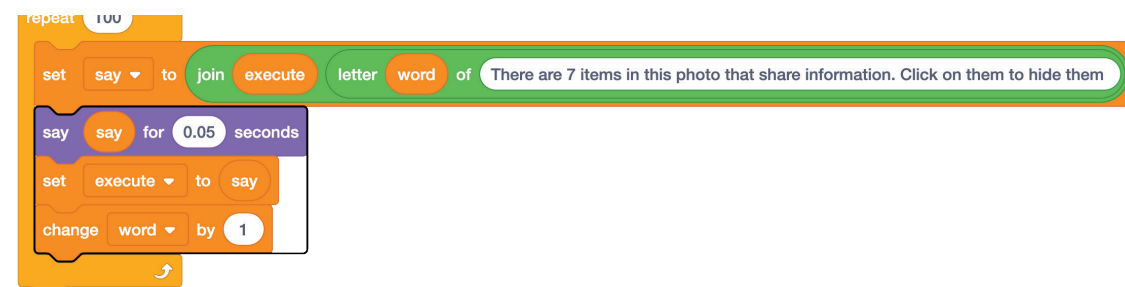
Loop 3 - The

Loop 4 - Ther

Inside a **repeat loop**, add a **set variable** block. Inside the set variable, add a **join operator**. In place of the word banana, add a **letter of operator**. From the variables menu, bring across the **execute** and **word** variables to add them in.



Add a **say** block from the looks menu, adding in the **say** variable. This will show the message on the screen. **Set execute** to be the say variable. Add a block to **change the word variable** by 1.



Test your completed game. The instructions should appear as if they are being typed on the screen, one letter at a time, and then your game should play successfully. Which sprite do you think will trick everyone the most?

Congratulations! You're a Moonhack changemaker. Will you try another project or spend time making this one better?

Don't forget to talk to an adult about registering your participation and downloading your certificate at moonhack.com

