

Making a maze with Scratch

Can you make it to the end?

Student guide

An activity by the Australian Computing Academy

Let's go!

Step 0: Get started

- Go to www.scratch.mit.edu
- Sign in with the username and password your teacher gives you

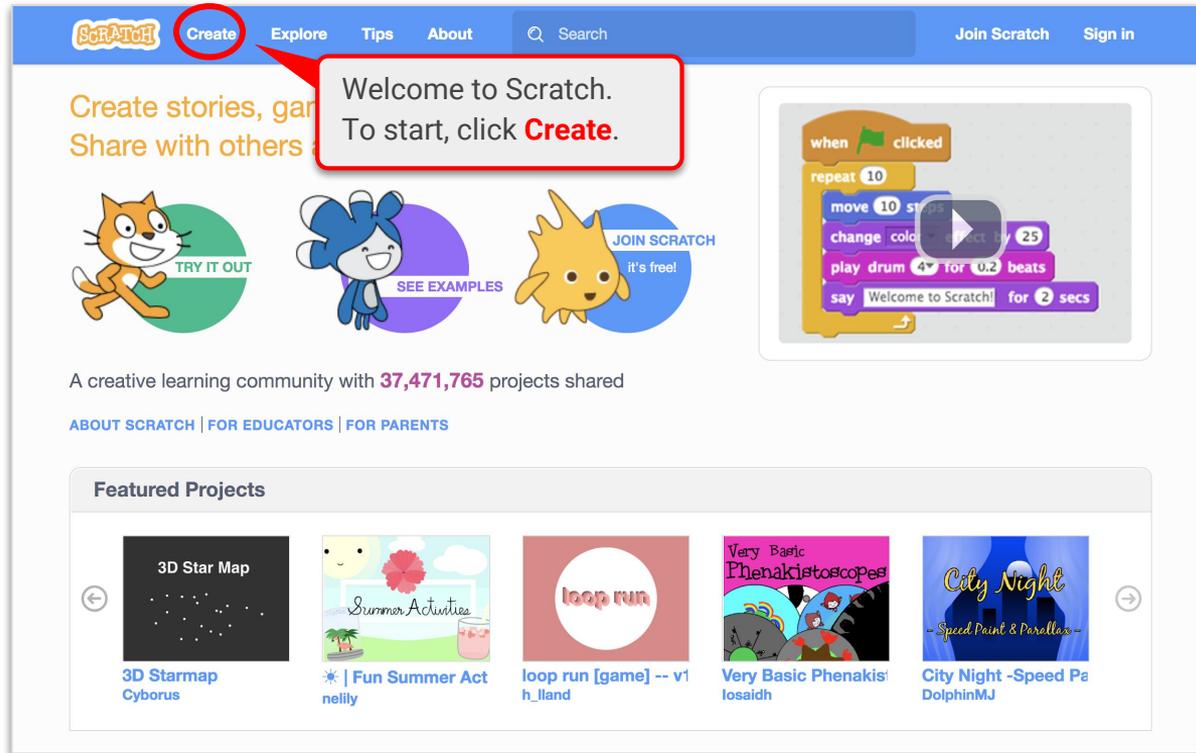


Make a note here:

My username:

My password:

Step 1: Using Scratch



The screenshot shows the Scratch website homepage. The 'Create' button in the top navigation bar is circled in red. A red callout box points to it with the text: "Welcome to Scratch. To start, click **Create**." Below the navigation bar, there is a main heading "Create stories, games, and animations. Share with others." and three character icons: Scratch the cat with a "TRY IT OUT" button, a blue character with a "SEE EXAMPLES" button, and a yellow character with a "JOIN SCRATCH it's free!" button. To the right, a code block preview shows a sequence: "when clicked", "repeat 10", "move 10 steps", "change color by 25", "play drum 4 for 0.2 beats", and "say Welcome to Scratch! for 2 secs". Below this, it says "A creative learning community with 37,471,765 projects shared" and provides links for "ABOUT SCRATCH", "FOR EDUCATORS", and "FOR PARENTS". At the bottom, a "Featured Projects" section displays five project thumbnails: "3D Star Map" by Cyborus, "Summer Activities" by nelly, "loop run [game] -- v1" by h_land, "Very Basic Phenakistoscopes" by losaidh, and "City Night -Speed Paint & Parallax" by DolphinMJ.

Step 1a: Creating your first project

The image shows the Scratch IDE interface with several instructional callouts:

- Code Drawer:** A vertical sidebar on the left containing various code blocks categorized by color: Motion (blue), Looks (purple), Sound (pink), Events (yellow), Control (orange), Sensing (teal), Operators (green), Variables (orange), and My Blocks (pink).
- Script Area:** A large white workspace in the center where code blocks are dragged and dropped.
- Stage:** A preview area on the right showing the current scene with a cat sprite.
- Sprite Area:** A panel below the stage for managing sprites, including a 'Cat' sprite and a 'Backpack'.

Callouts and their locations:

- Top Left:** A red callout box with the text "Explore different kinds of blocks by clicking the coloured dots." pointing to the code drawer.
- Center:** A red callout box with the text "Drag code blocks here to write code, drag them back to the drawer to delete them." pointing to the script area.
- Top Right:** A red callout box with the text "This is where you see your code run." pointing to the stage.
- Right Side:** A red callout box with the text "Right click a sprite to delete it." pointing to the cat sprite in the stage.
- Bottom Right:** A red callout box with the text "Add new sprites" pointing to the 'Add Sprite' button in the sprite area.
- Bottom Right:** A red callout box with the text "Choose a backdrop" pointing to the 'Add Backdrop' button in the stage area.

Step 1b: Working with sprites

The image shows a screenshot of the Scratch software interface. On the left, the 'Code' area is visible, showing a list of code blocks categorized by color: Motion (blue), Looks (purple), Sound (pink), Events (yellow), Control (orange), Sensing (light blue), Operators (green), Variables (orange), and My Blocks (pink). A red box highlights the 'Looks' category, with an arrow pointing to the 'say Hello! for 2 seconds' block. Another red box highlights the 'Sound' category, with an arrow pointing to the 'say Hello!' block. A third red box highlights the 'Costumes' tab, with an arrow pointing to the 'switch costume to cat-a' block. A fourth red box highlights the 'Code' tab, with an arrow pointing to the 'say Hello! for 2 seconds' block. A fifth red box highlights a colored dot in the 'Looks' category, with an arrow pointing to the 'say Hello! for 2 seconds' block. The main workspace shows a cat sprite on a stage. The bottom right corner shows the 'Sprite' and 'Stage' panels, with the 'Cat' sprite selected and its properties (Size: 100, Direction: 90) visible.

Click **Sounds** to add sound.

Click **Costumes** to change how sprites look.

Click **Code** to code your sprites.

Click a coloured dot to find code blocks for your sprite or backdrop.

Step 1c: Adding some code

Drag out these blocks to see what will happen.

The screenshot shows the Scratch code editor interface. On the left is a palette of block categories: Code (selected), Costumes, and Sounds. Below these are icons for Motion, Looks (selected), Sound, Events, Control, Sensing, and Operators. The main workspace contains three stacked blocks: a yellow 'when green flag clicked' block, a blue 'move 10 steps' block, and a purple 'say Hello! for 2 seconds' block. Three red callout boxes provide instructions: 'Always start with an event.' points to the yellow block; 'Blocks are colour coded so you can easily find them.' points to the purple block; and 'Anything in a white field can be changed by clicking and typing a word or number.' points to the 'Hello!' text field in the say block.

Step 1d: Designing sprites

The screenshot shows the Scratch costume editor interface. The top menu bar includes 'Scratch!', 'File', 'Edit', 'Tutorials', 'maze', 'Share', 'See Community', and 'Give Feedback'. The main workspace is divided into several sections:

- Costume List:** On the left, there are two costume thumbnails labeled 'cat-a' (93 x 101) and 'cat-b' (90 x 110). A red callout box points to the 'cat-a' thumbnail with the text: "Right click to copy a costume and edit it to create animations."
- Costume Properties:** The 'cat-a' costume is selected. It has a 'Costume' dropdown set to 'cat-a', a 'Fill' color picker, and an 'Outline' color picker set to black with a thickness of 4. A red callout box points to the 'Group' and 'Layer' buttons with the text: "Group and layer elements".
- Tools:** A vertical toolbar on the left contains icons for selection, eraser, fill, text, line, circle, and rectangle.
- Canvas:** The central canvas shows a drawing of the Scratch cat character on a checkered background. A 'Convert to Bitmap' button is at the bottom of the canvas.
- Stage:** On the right, the 'Stage' area shows the 'Sprite' dropdown set to 'Cat', 'x' and 'y' coordinates at 0, 'Size' at 100, and 'Direction' at 90. Below this is a 'Backdrops' list with 3 items.

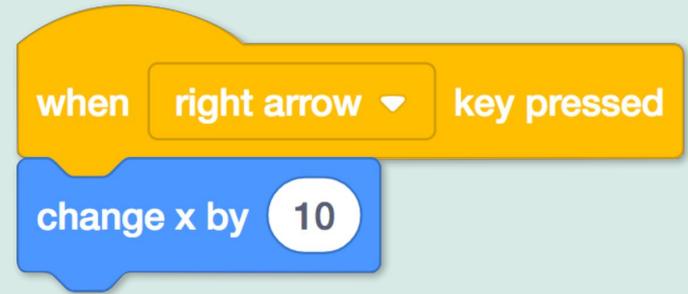
Add lines, shapes, colour and text

Group and layer elements

Right click to copy a costume and edit it to create animations.

Step 2a: Code the arrow keys

- Choose a new sprite (small and kind of round is good)
- Pull out the **When space key pressed** block from the **events** drawer
- Change the block so it says **When right arrow key pressed**
- From the **motion** drawer, pull out the **change x by 10** block.
- Click the blocks together.
- Test: press the right arrow key and see what happens.

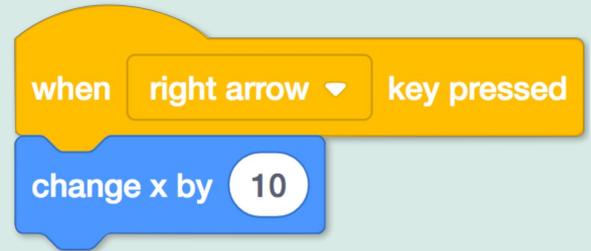


Step 2b: Code the arrow keys

- Drag a new **when space key pressed** block from the **events** drawer
- Drag out a **change y by 10** block from the **motion** drawer
- Click them together
- Test: press the right arrow key and see what happens.

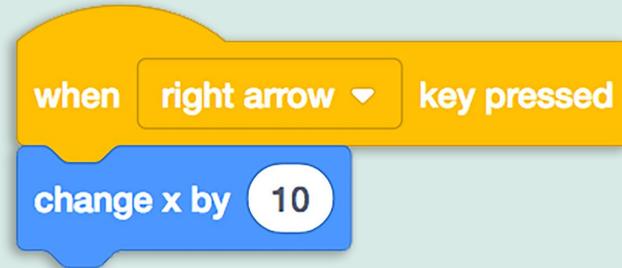
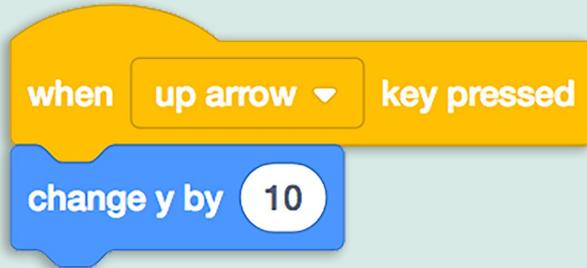


If you are ready, go ahead and code the down arrow and the left arrow (if you're not sure, you can see all the code on the next page).



Checkpoint

Test: your sprite can move left, right, up and down using the arrow keys



Step 3: Draw the maze

Now it's time to draw a maze using the paint tools in Scratch.

c Just use one colour for your maze

d Using the paintbox, use filled in rectangles and circles to make a quick maze.

e Make sure your sprite can fit through the maze

a Click the backdrop tab

b Create a new backdrop by clicking the paintbrush.

Step 4a: make the maze solid

- To check if your sprite touches the wall, choose an **if...then** block from the **control** drawer
- From the **sensing drawer**, bring out the **touching colour** block and add it to the block as shown
- Click inside the coloured oval (purple here) then hover over your maze so that the oval changes colour to be the same as the colour of your maze walls
- From the **motion** drawer, choose the **go to x y** block and add the numbers shown



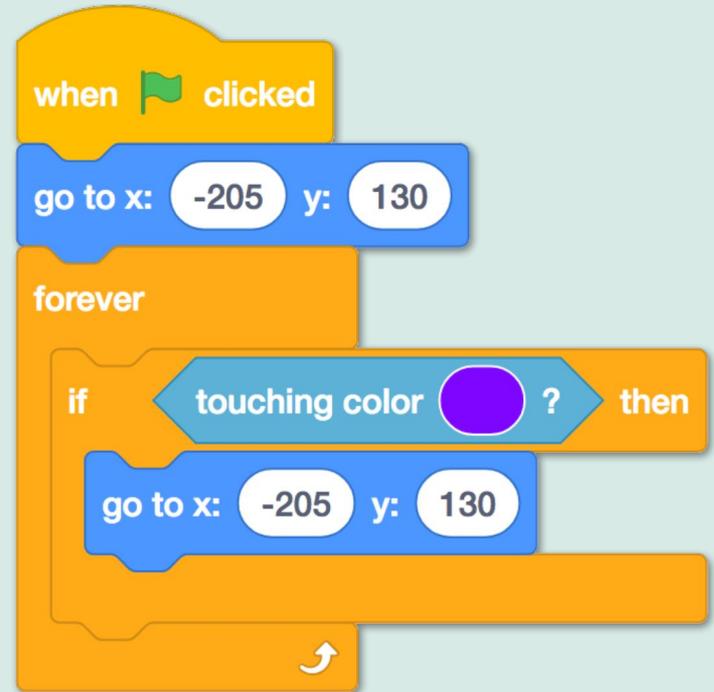
Nothing happens yet! Keep going on the next page.



Step 4b: make the maze solid

- To make sure the code checks all the time if the sprite is touching purple:
- Pull out a **when green flag is clicked** block from the **event** drawer.
- To start in the same place each time, pull out a **go to x y** block from the **motion** drawer
- Pull out a **forever** block from the **control** drawer
- Put the **if...then** code you already have inside the **forever** block
- Here is the final code. 

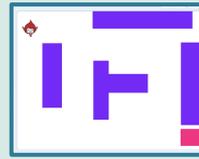
- **Test:** does your sprite move around the maze and can't pass through the walls?



Step 5: Add a beginning and end

To make an end point, the code is a bit like the code to make the walls solid.

- Draw a coloured rectangle onto your backdrop - ours is pink
 - Use an **if...then** block with a **touching color** block to check if the sprite has touched your end point
 - Decide what happens: we used a **go to x y** block to return to the start,
 - Add some words: find a **say** block in the **looks** drawer and add your own message
 - You can also add a sound effect or change the way the sprite looks using more blocks from the **looks** and **sound** drawer
- **Test:** guide your sprite through the maze. It should bounce off all, and when it reaches the end return to a go and deliver your message.



```
when green flag clicked
  go to x: -205 y: 130
  forever
    if touching color purple ? then
      go to x: -205 y: 130
    if touching color pink ? then
      go to x: -205 y: 130
      say You win! for 2 seconds
```

Level it up

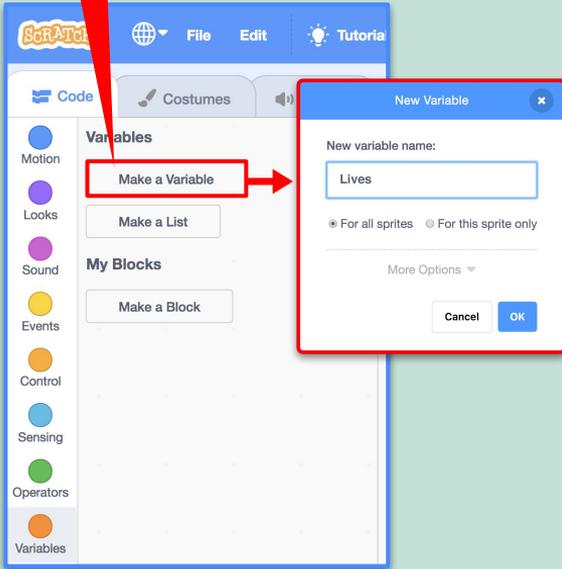
Your maze works!

On the next slides you will find ideas to make your maze better.

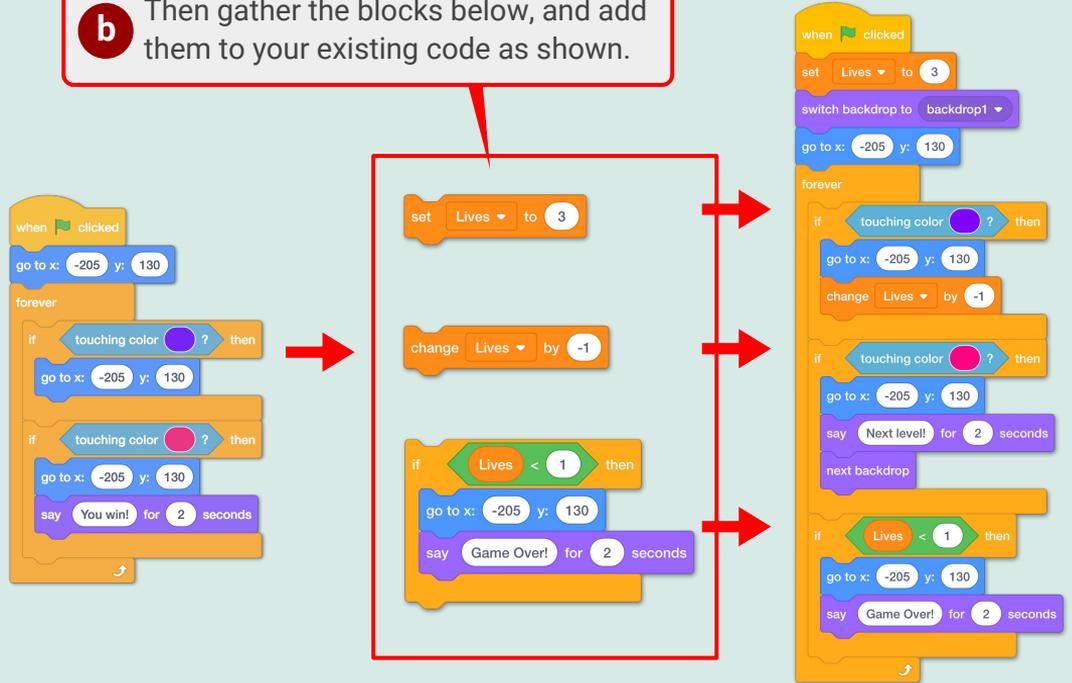
Step 6: Add variables

If you want your sprite to have lives, have a go adding this code to your project

a Head to the **variables code** drawer, select **make a variable**, and call it **Lives**.



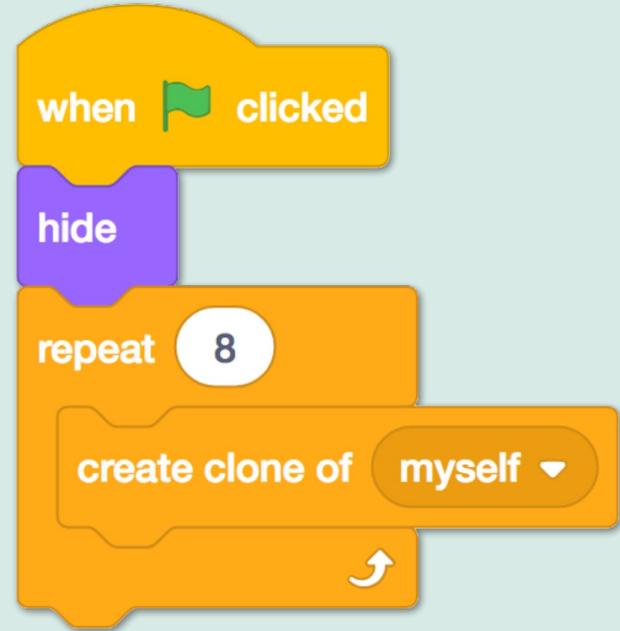
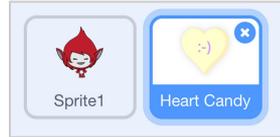
b Then gather the blocks below, and add them to your existing code as shown.



Step 7: Create tokens

Add tokens to your maze for your sprite to collect.

- Choose a new sprite - we used a heart
- Each sprite has its own space to write code. Make sure you put code for your heart in the right place - the heart should have a blue rectangle around it
- Use a **when green flag clicked** block
- From the **looks** drawer pull out a **hide** block
- In the **control** drawer pull out a **repeat** block
- Also in **control** , find a **create clone of myself** block
- Put them together as shown



Using cloning with a repeat block lets you create lots of copies of the same sprite without having to write lots of the same code.

Step 8: Send tokens to random spots

This code tells each token where to go, and whether it should show or not.

- Pull out a **when I start as a clone** block from the **events** drawer
- Find a **point in direction** block and **go to random position** block in the **motion** drawer
- We use a different if block this time - **if then else** so that the token will hide if it's on the wall, or show if it's not
- The **wait** block is in the **control** drawer
- The **operators** drawer contains the **pick random 1 to 10** block

- Test your code: are there any tokens on the wall?
- Test your code again: how long do the hearts stay for before they hide?

```
when I start as a clone
  point in direction 90
  go to random position
  if touching color purple ? then
    hide
  else
    show
  wait pick random 1 to 10 seconds
  hide
```

The image shows a Scratch script for a clone. It starts with a 'when I start as a clone' block. This is followed by a 'point in direction 90' block, then a 'go to random position' block. An 'if then else' block follows, with the condition 'touching color purple?'. The 'then' branch contains a 'hide' block, and the 'else' branch contains a 'show' block. Below the if block is a 'wait pick random 1 to 10 seconds' block, and finally a 'hide' block at the end of the script.

Step 9: Collect tokens

You can decide what happens when a sprite touches a token.

In our example the token spins around, changes size and colour, before hiding.

You can choose what happens to your token!



Make sure this code goes on to your heart sprite.

```
when I start as a clone
  forever
    if touching Sprite1 ? then
      repeat 30
        turn 12 degrees
        change size by -2
        change color effect by 5
      hide
```

The image shows a Scratch code block structure. It starts with a 'when I start as a clone' block. Below it is a 'forever' loop. Inside the loop is an 'if touching Sprite1 ? then' block. Inside the 'if' block is a 'repeat 30' loop. Inside the 'repeat' loop are three blocks: 'turn 12 degrees', 'change size by -2', and 'change color effect by 5'. Below the 'repeat' loop is a 'hide' block. The 'forever' loop and the 'if' block both have small curved arrows at the bottom indicating they are loops.

Step 10: Add another level

If you want to add a second level to your maze, you can design another backdrop (with the same colour) and add the extra code shown. You can add as many extra backdrops as you like as long as you use the same colour, and the same colour for the end point.

```
when green flag clicked
  go to x: -205 y: 130
  set Lives to 3
  forever loop
    if touching color purple then
      go to x: -205 y: 130
      change Lives by -1
    if touching color pink then
      go to x: -205 y: 130
      say You win! for 2 seconds
    if Lives < 0 then
      say Game Over! for 2 seconds
```



```
switch backdrop to backdrop1
say Next level! for 2 seconds
next backdrop
```



```
when green flag clicked
  switch backdrop to backdrop1
  go to x: -205 y: 130
  set Lives to 3
  forever loop
    if touching color purple then
      go to x: -205 y: 130
      change Lives by -1
    if touching color pink then
      go to x: -205 y: 130
      say Next level! for 2 seconds
      next backdrop
    if Lives < 0 then
      say Game Over! for 2 seconds
```



Gather the blocks above, and add them to your existing code as shown.

Step 11: What's next?

Now it's up to you now to add more features to your maze.



Well done!

What did you think?

- Is there anything in the maze that you would like to change or improve?
- Is there anything that you found really tricky?
- Are there parts of this project that you would use again in different ways?
- Play someone else's game. What do you like about it? Is there anything you don't understand or that could be improved?

Creative commons



This work is licensed under the Creative Commons Attribution 4.0 International License.

To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>

or send a letter to:

Creative Commons,
PO Box 1866, Mountain View,
CA 94042, USA.