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:
Welcome to Scratch. To start, click Create.
Step 1a: Creating your first project

- Drag code blocks here to write code, drag them back to the drawer to delete them.
- This is where you see your code run.
- Right click a sprite to delete it.
- Add new sprites
- Choose a backdrop
- Explore different kinds of blocks by clicking the coloured dots.
Step 1b: Working with sprites

- Click a coloured dot to find code blocks for your sprite or backdrop.
- Click **Sounds** to add sound.
- Click **Costumes** to change how sprites look.
- Click **Code** to code your sprites.
Step 1c: Adding some code

Drag out these blocks to see what will happen.

Always start with an **event**.

Blocks are colour coded so you can easily find them.

Anything in a white field can be changed by clicking and typing a word or number.
Step 1d: Designing sprites

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

Add lines, shapes, colour and text.

Group and layer elements.
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

1. When the up arrow key is pressed, change y by 10.
2. When the right arrow key is pressed, change x by 10.
3. When the down arrow key is pressed, change y by -10.
4. When the left arrow key is pressed, change x by -10.
Step 3: Draw the maze
Now it’s time to draw a maze using the paint tools in Scratch.

- Click the backdrop tab
- Create a new backdrop by clicking the paintbrush.
- Using the paintbox, use filled in rectangles and circles to make a quick maze.
- Just use one colour for your maze
- Make sure your sprite can fit through the maze

Classroom use
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 if...then 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.
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:

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

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?
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.
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.

Gather the blocks above, and add them to your existing code as shown.
Step 11: What’s next? It’s over 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?
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.