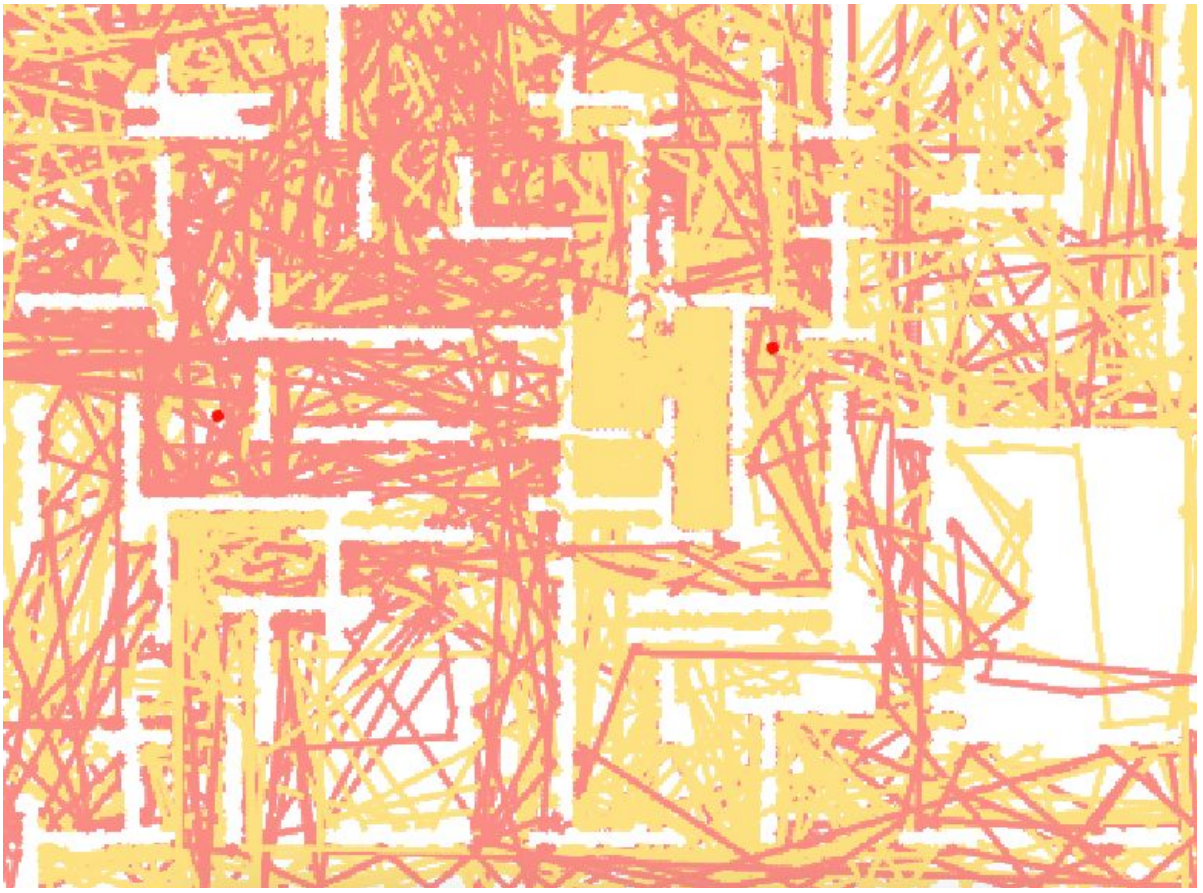
Phoenix



You will need:

Access to the Scratch website, from a computer or laptop.
(It might also work from an ipad or tablet)

This tutorial is best for people who have used Scratch a little bit before.

Setup:

If you haven't used Scratch before, click **join scratch** at the top and create an account. **You might need your parent's email to do this, so ask for help.** If you can't make an account, you can follow along, but you won't be able to save your work.

Once you've made an account follow the Phoenix Scratch studio here. This is where you can find the template versions of all the workshops we create. Feel free to remix them and post the new things you create.

Wait, what is Scratch?

Scratch is a free, online, block-based code editor. You might have used it in school. If you've been to one of our family workshops, you've probably seen us using it there.

There are lots of different tutorials available for using Scratch. Here we are going to focus on showing you ways of using Scratch that link to the work we do at Phoenix. This will include things like animation and drawing.

Doodle Maze

This project combines some of the ideas we covered in the Grow Art tutorial with some new ways of moving sprites around and using clones.

To get started **remix** this project,

https://scratch.mit.edu/projects/383415776

Step 1

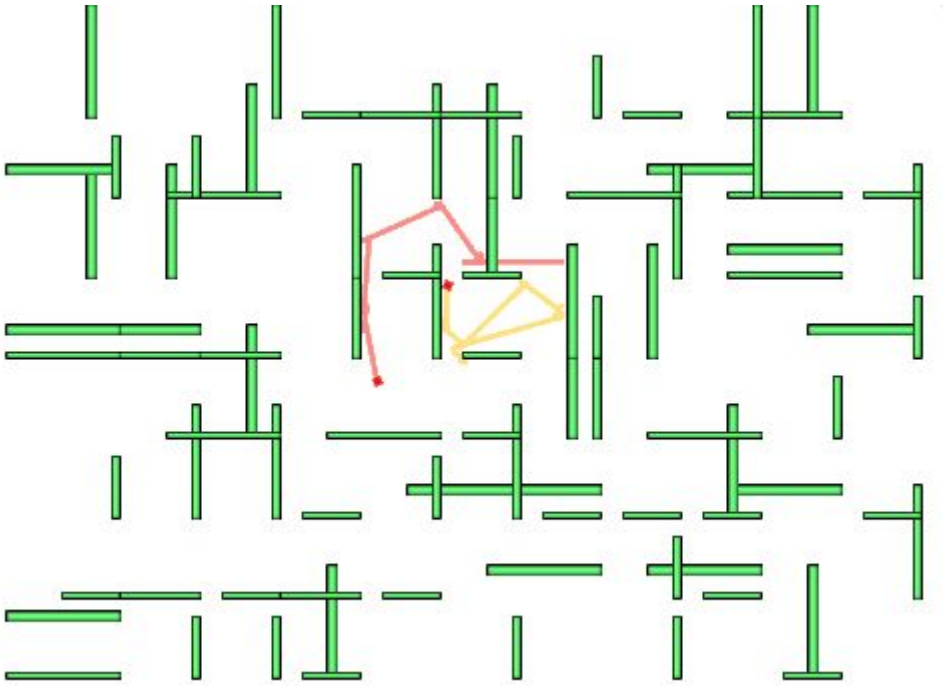
There is quite a lot going on in this project.

Before taking a look at the code, run the project, using the green flag.

With just the three sprites, we are achieving quite a lot.

To have a closer look at this, open up the code for the MazeWall sprite. At the moment we are hiding it using an effect. Find the two set ghost effect to 100 blocks and change the value to 0.

Now when you press the green flag, you should have something that looks like this.

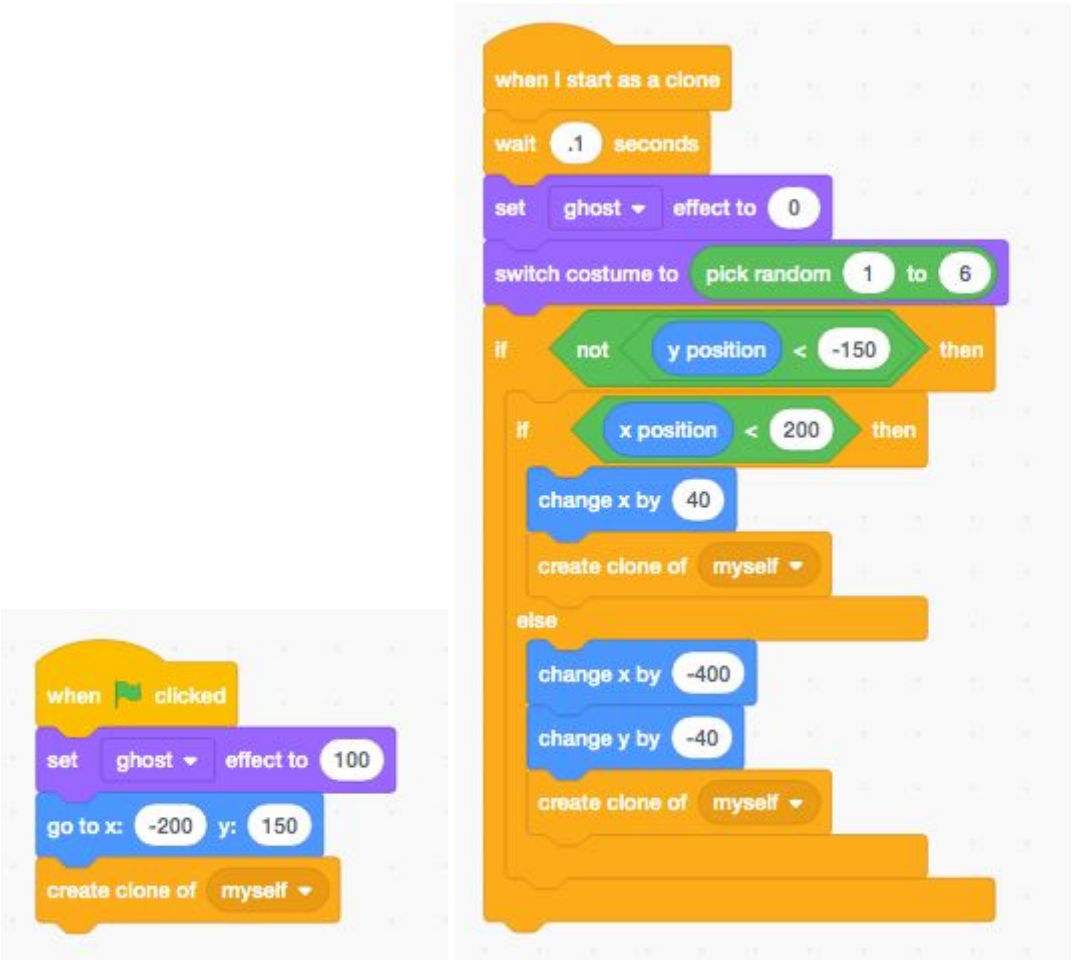


Now, if you stop the scene, and restart it, you should start to notice what is happening.

Each time the scene starts a new maze is drawn.

Step 2

We can take a closer look at this by adding in a wait .1 seconds block.



Now we can see how the maze is built out of the clones. The sprite creates a clone of itself and moves, over and over again, until it reaches the edge of the stage.

At each step it chooses a costume. If you open up the costumes you can see that there are different costumes that each have a different type of wall in them.

Try adjusting these and see what kind of mazes you can start to make.

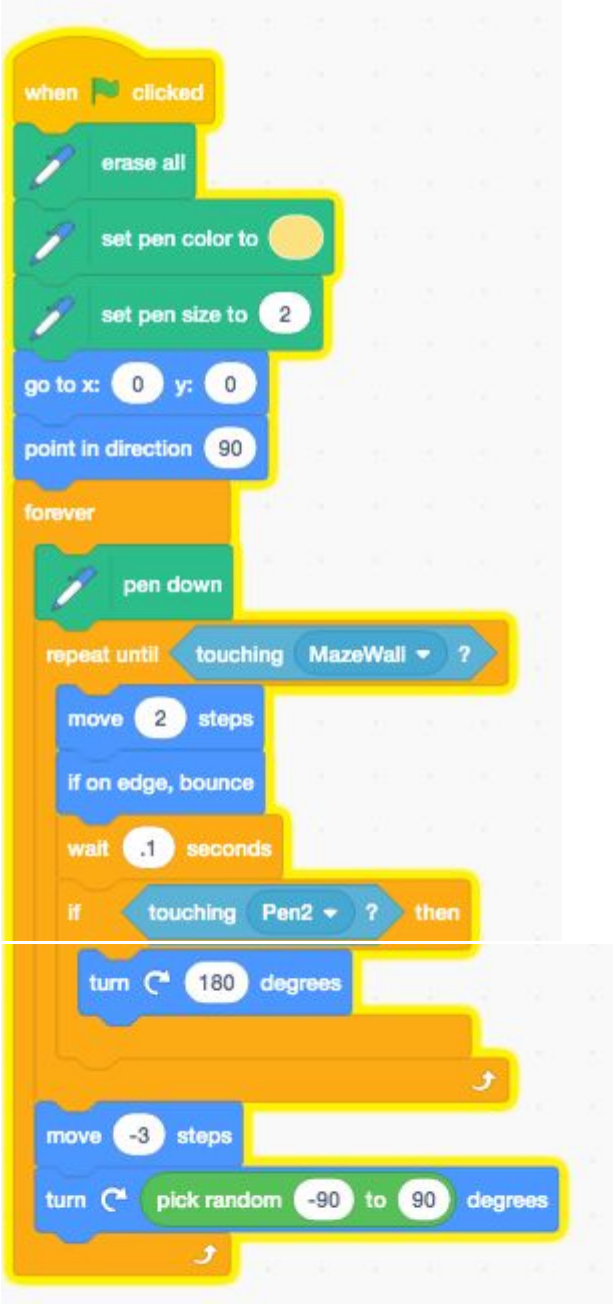
As you do this you will probably have to start adjusting the values in the code block. The change x by 40 and change y by -40 blocks, in particular will need careful adjustments.

For this exercise we don't need to create a perfect maze, but it should be something that our Pen sprites can navigate.

Step 3

Let's switch over to the Pen1 sprite.

This code is a bit longer than the blocks that build the maze, but should still be things that we are familiar with from earlier tutorials.



First of all we are setting up our Pen. We choose a size and a colour. We also use erase all to make sure we start with an empty stage each time.

The idea with our Doodle Maze is that the Pen sprite will explore the maze on its own and discover a path, whilst leaving an interesting trail.

To make this happen we need a way of checking for collisions between the Pen Sprite and the MazeWall Sprite.

In different projects you might find slightly different ways of solving this. For our project, because each part of the wall is a clone of the original MazeWall Sprite, we can use the touching MazeWall ? block to check for a collision.

We are saying that the Pen sprite should move 2 steps and then check if it is touching the MazeWall. It will also bounce if it hits the edge of the stage after two steps, and turn if it hits the other Pen sprite.

When the Pen sprite does hit a MazeWall sprite it moves backward 3 steps. This is to make sure it gets out of the wall.

Finally we tell the Pen sprite to turn around by giving it a range of -90 to 90 degrees. You can experiment with different values here. We chose this range as it makes sure the Pen Sprite has a good chance of moving away from the wall and continuing it's path.

Step 4

When you are happy with how the maze works you can start to experiment with some other ideas.

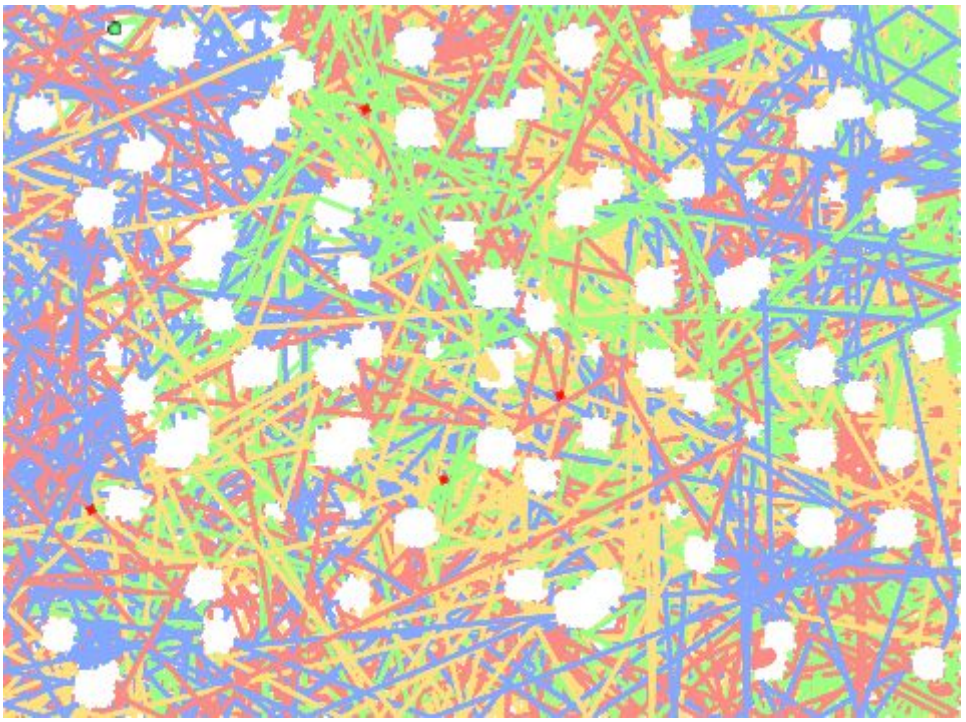
If you like the effect, remember to change the MazeWall code back to how it started, without the delay, and with set ghost effect to 100.

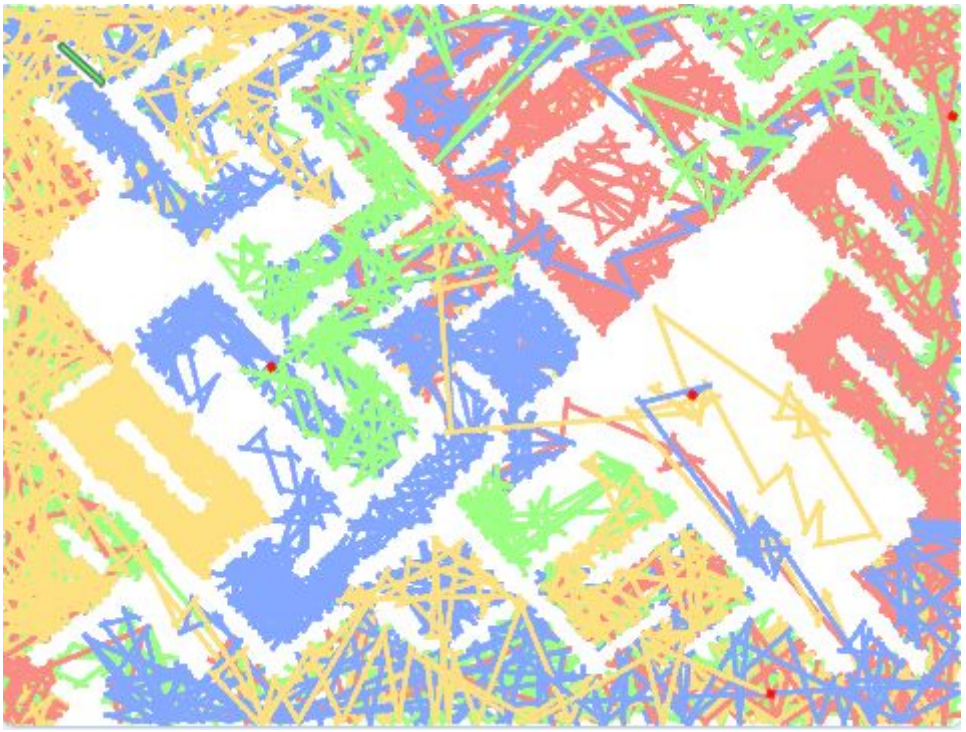
Extra steps

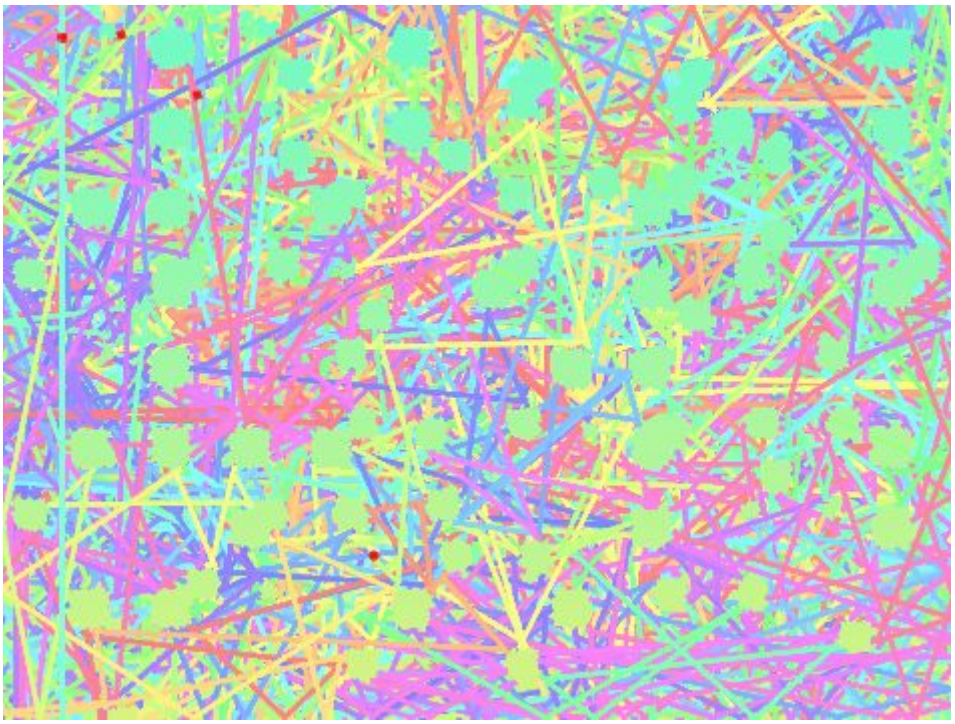
Wow, you've just learned the basics of how this sound box works. There are still a few switches we didn't mention though.

- Can you create a MazeWall sprite that has more costume options?
- Can you create extra Pen sprites to explore the maze?
- Can you make the Pen sprites move in a different way?
- Can you create a new sprite that the Pen sprites can collect?

Or, you could see if you can recreate something similar to these pictures.







Remember to share your project to the Phoenix Scratch studio!

Hopefully you found this exercise interesting! Here we have explored how collision can be used. This is very useful if you are starting to build games.

Phoenix.org.uk
@PhoenixLeic