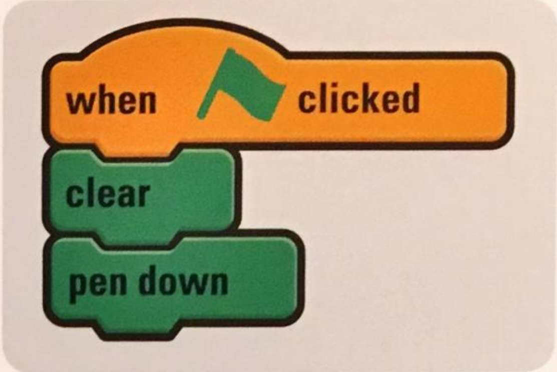
4

Join the **Clear** and **Pen down** blocks together.

Click on the **Events** group. Drag over a **When green flag clicked** block and put it above the **Clear** and **Pen down** blocks.

Now the sprite will be ready to draw a new picture whenever the green flag is clicked.



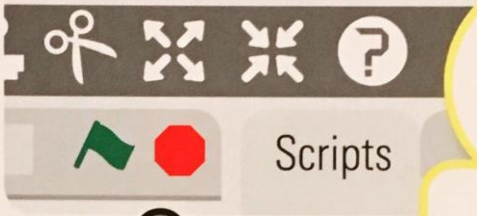
```
when ⚑ clicked
clear
pen down
```

5

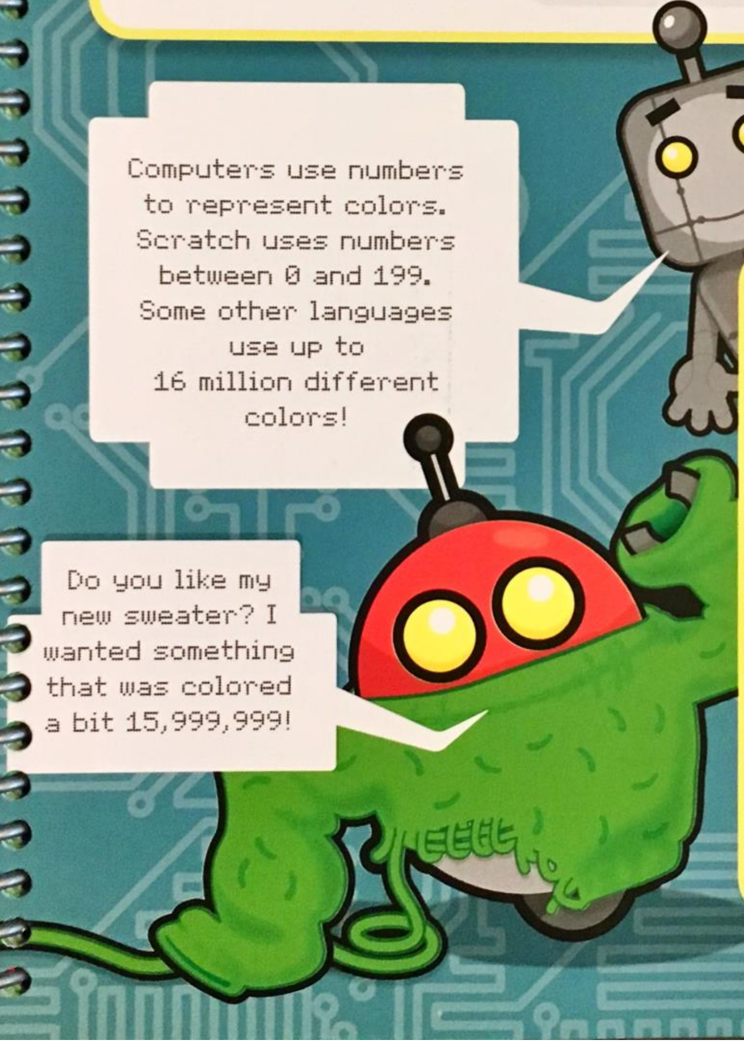
The large sprite can get in the way when drawing.

Shrink it using this icon.

Click the icon, then click on the sprite to make it smaller.

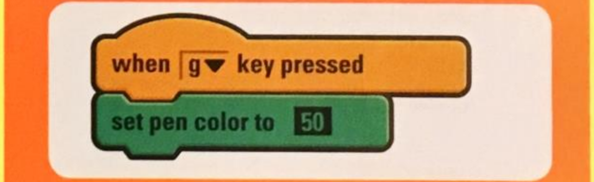


Your program is now complete! Click the green flag icon to start using it to draw.


Computers use numbers to represent colors. Scratch uses numbers between 0 and 199. Some other languages use up to 16 million different colors!
Do you like my new sweater? I wanted something that was colored a bit 15,999,999!

Challenge

Can you add some commands that will let players change the pen color? You will need to use **When key pressed** blocks from the **Events** group, and **Set pen color to** blocks from the **Pen** group.



```
when [g ▾] key pressed
set pen color to (50)
```

You will need to add a **When key pressed** block and **Set pen color to** block for each color you want to use. Experiment with different numbers and keep testing your code.

DEBUGGING

Coding can be a process of trial and error—testing out ideas and seeing if they work. It is pretty common to make mistakes when doing this. A bug is another name for a mistake in a piece of code that stops it from working correctly. Debugging means fixing those mistakes. You'll find the answers to these questions on page 33.

1

Bugs for breakfast

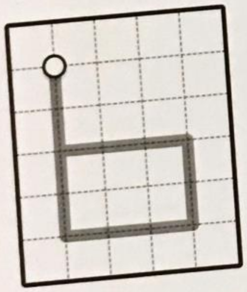
Get started with debugging by finding the mistake in these instructions for making a piece of toast:

1. Get a piece of bread.
2. Put it in the toaster.
3. Spread butter on the toast.
4. Take the toast out of the toaster.

2

B wrong

Here are some commands to draw a letter "b," like the one shown here. But what's wrong with the commands?

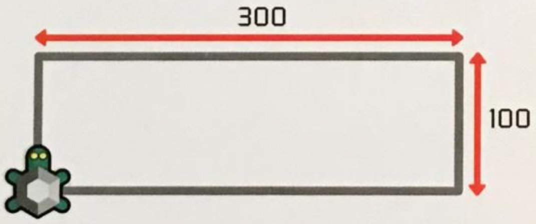


Start at the circle.
D4 R3 D2 L3

3

Logo bug

This rectangle is 100 pixels high and 300 pixels wide.

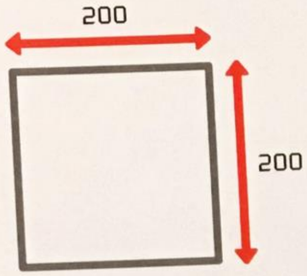


This code should draw the rectangle—but there's a bug or two somewhere!

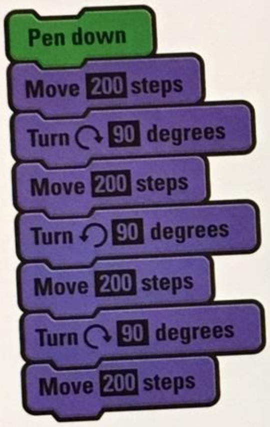
```
fd 100
rt 90
fd 300
righrtt 90
fd 100
rt 90
fd 90
```

4 Scratch bug

Here is some Scratch code to draw a square. The square should be 200 pixels by 200 pixels.



Find the bug in the code:



```
Pen down
Move 200 steps
Turn ↻ 90 degrees
Move 200 steps
Turn ↺ 90 degrees
Move 200 steps
Turn ↻ 90 degrees
Move 200 steps
```

Key word

Debugging: Getting rid of mistakes that stop your code from working correctly

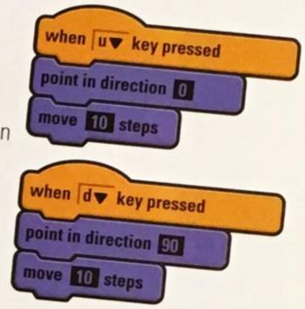
5 Broken game

This Scratch program should make a sprite do two things:

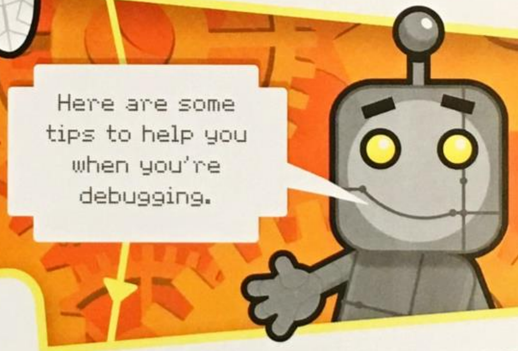
Move **up** when U is pressed.

Move **down** when D is pressed.

Find the bug in the code.



```
when u ▾ key pressed
point in direction 0
move 10 steps

when d ▾ key pressed
point in direction 90
move 10 steps
```



Debugging tips

When your code doesn't do what you want it to:

1. Go through your code step by step, thinking about what each command does.
2. Draw a picture or diagram to help you.
3. Take a break for a few minutes!

Think about these guidelines when you are coding:

1. Plan your program carefully first, either with a diagram or some written notes.
2. When you are learning to code, it is better to write lots of small, simple programs rather than one larger and more complex program.
3. Test your program as you are building it—don't wait until you have put in all the commands.

CHAPTER 2
CHAPTER 3
CHAPTER 4

LOOPS

Computers are very good at doing things over and over again. A loop is a way of making your program do something repetitive—like count up to 20, draw a shape with lots of sides, or make a spaceship orbit around and around a planet.

Why use loops?

Imagine you want to write a program to draw a square. You could do it like this:

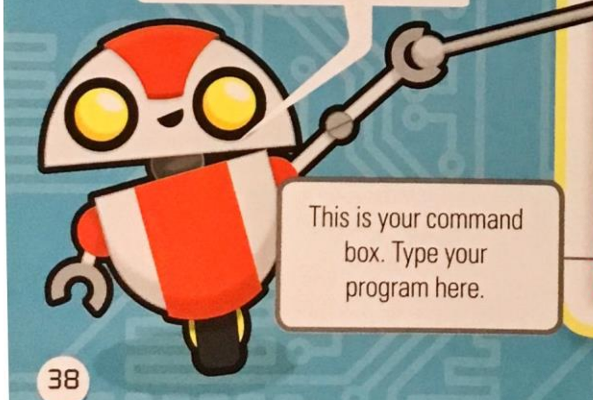
1. **Draw the first side.**
2. **Turn 90 degrees.**
3. **Draw the second side.**
4. **Turn 90 degrees.**
5. **Draw the third side.**
6. **Turn 90 degrees.**
7. **Draw the fourth side.**
8. **Turn 90 degrees.**

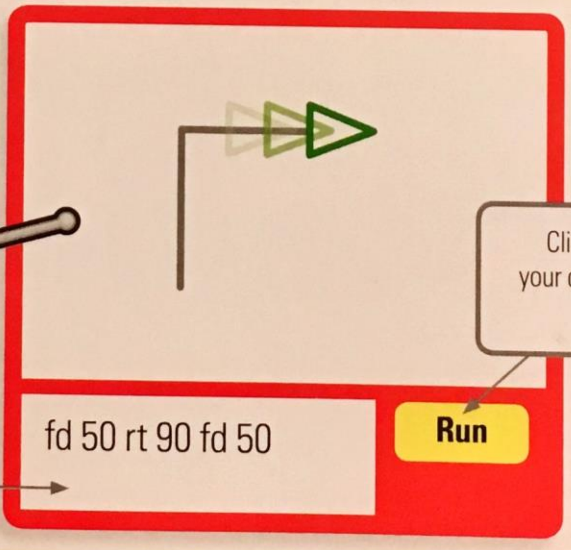
It would take 8 separate instructions. A loop makes this much simpler. With a loop we just need 3 instructions:

1. **Repeat this 4 times:**
2. **Draw a side.**
3. **Turn 90 degrees.**

Logo loops

We're going to write some Logo programs to try out making repeat loops. First of all, in case you're new to Logo, let's get to grips with how to use it.


This is the drawing box. The output of your program will show up here.



```
fd 50 rt 90 fd 50
```

This is your command box. Type your program here.

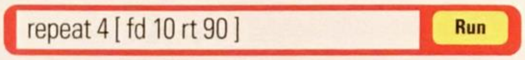
Click **Run** to test your code or press the **Enter** key.

LOOPS IN SCRATCH

Now we're going to look at how we can use loops in Scratch. Loops work the same way in Scratch as in Logo, but instead of typing your commands, you drag and drop them. Let's give it a try.

Draw a square

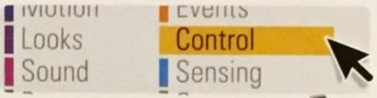
If you wanted to draw a square in Logo, you would type:



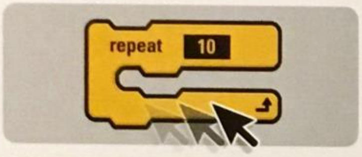
```
repeat 4 [ fd 10 rt 90 ]
```

In Scratch, we can create the same code by dragging **Repeat**, **Move**, and **Turn** blocks:

1 Go to the Scratch website then click **Create** or **Try it out**. Turn to page 4 for help. Now click on the **Scripts** tab in the center of the Scratch screen. Choose the **Control** group.

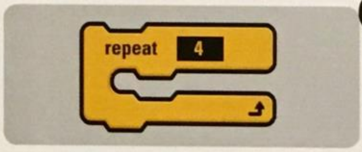


2 Drag a **Repeat** block onto the scripts area on the right.



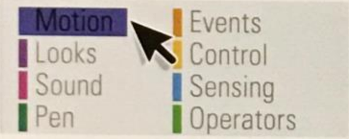
```
repeat 10
```

3 Change the number of loops to repeat to 4.

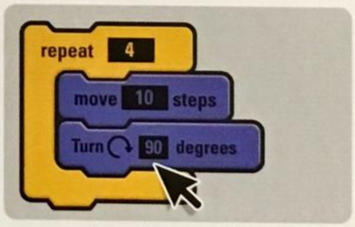


```
repeat 4
```

4 Click on the **Motion** group.

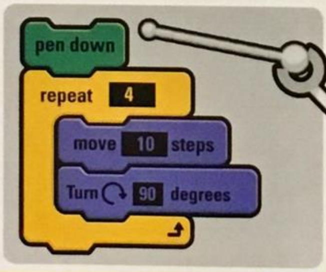


5 Drag in a **Move** block and a **Turn** block. Then change the amount to turn to 90 degrees.



```
repeat 4
    move 10 steps
    Turn ↻ 90 degrees
```

6 Drag a **Pen down** block from the **Pen** group.



```
pen down
repeat 4
    move 10 steps
    Turn ↻ 90 degrees
```

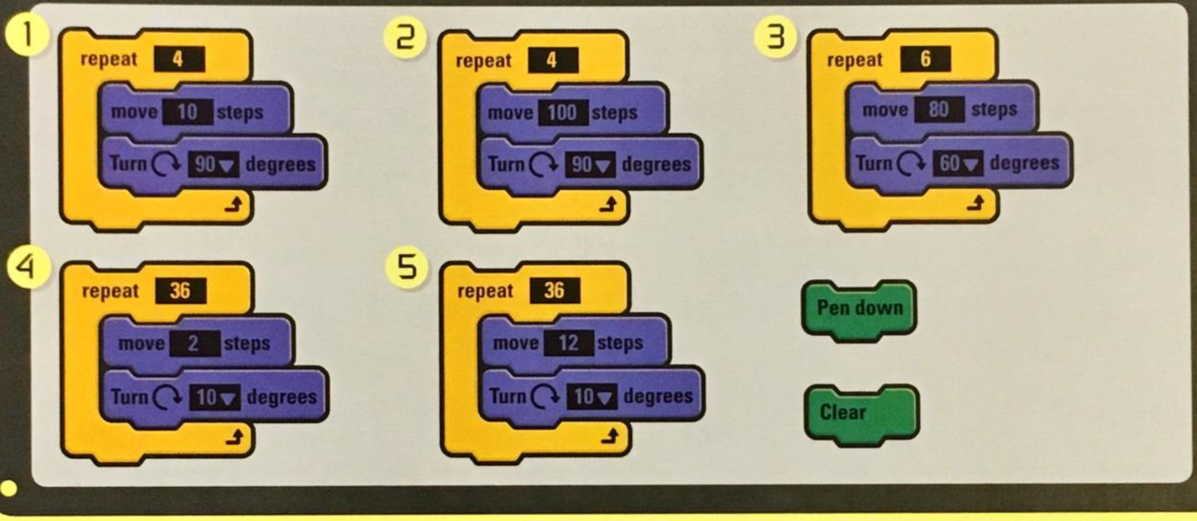
Click the **Pen down** block to run the loop. Drag the cat sprite out of the way. Your program will draw a square on the screen.

Saving your work

Click the **File** menu at the top of the page on the left. Then click:
New – to start some new work.
Download to your computer – to save a file on to your computer.
Upload from your computer – to open a file you saved earlier.

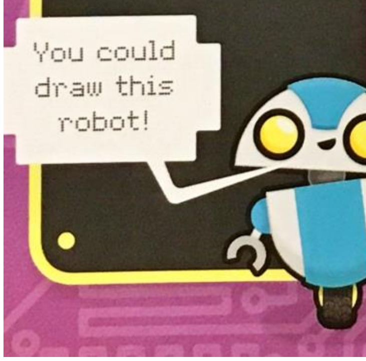
Practicing Scratch loops

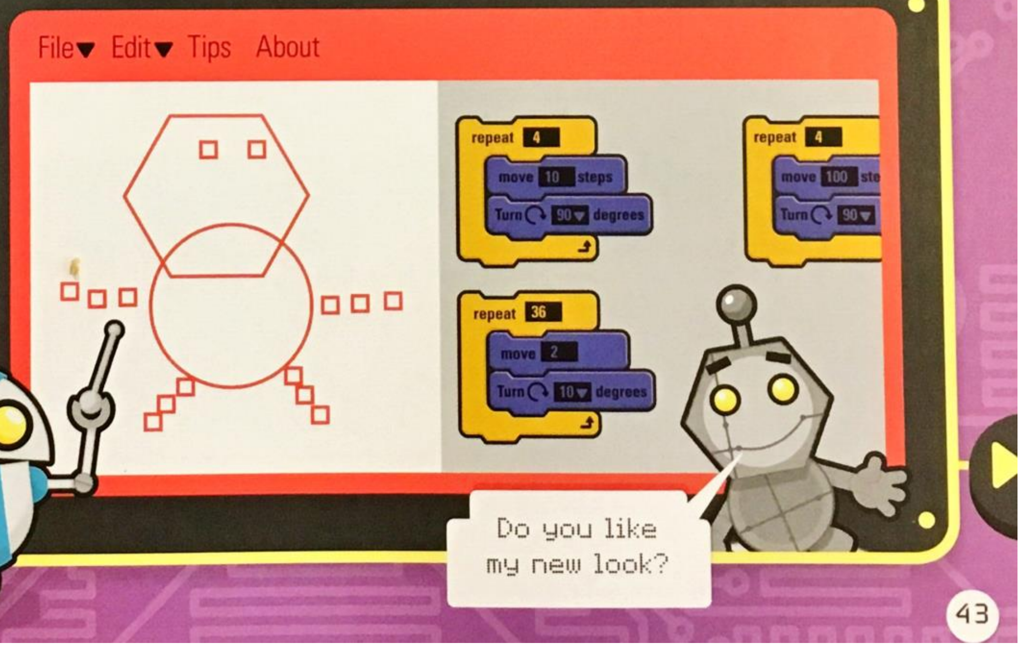
Create these 5 loop blocks on the scripts area. You'll also need to drag over a **Pen down** and a **Clear** block. Try clicking on the **Pen down** code block and then on each of the **Repeat** blocks in turn. Click **Clear** to erase your shapes. Test out what each of the loops draws. Check the answers on page 62.



```
1
repeat 4
    move 10 steps
    Turn ↻ 90 degrees

2
repeat 4
    move 100 steps
    Turn ↻ 90 degrees

3
repeat 6
    move 80 steps
    Turn ↻ 60 degrees

4
repeat 36
    move 2 steps
    Turn ↻ 10 degrees

5
repeat 36
    move 12 steps
    Turn ↻ 10 degrees

Pen down

Clear
```

After drawing each of the shapes above, try dragging the Scratch sprite to a new space on the screen. You can start building up a picture or pattern.





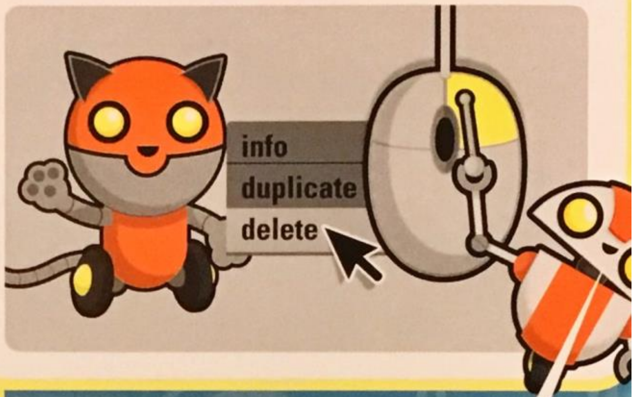
```
repeat 4
    move 10 steps
    Turn ↻ 90 degrees

repeat 4
    move 100 ste
    Turn ↻ 90

repeat 36
    move 2
    Turn ↻ 10 degrees
```

LOOPS FOREVER

Sometimes we need loops that run forever. This is particularly useful in games where we want something to keep happening, like moving a sprite around. We're going to code a game where a fish keeps swimming around the screen, following the mouse pointer.

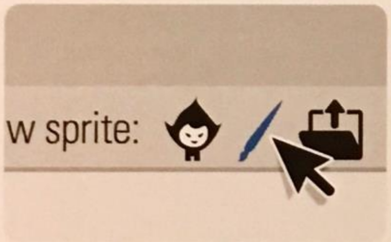
How to code a swimming fish

1 Start by going to the Scratch website. Delete the main sprite by right-clicking on it and then choosing **Delete**.



Right-clicking means press this button if you're on a PC. If you're on a Mac, press **Control** and click.

2 Now make your own fish sprite. Start by clicking **Paint new sprite**.

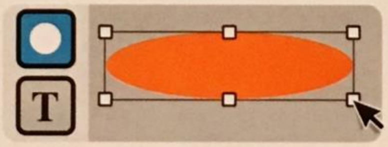


Select the **Ellipse** tool.

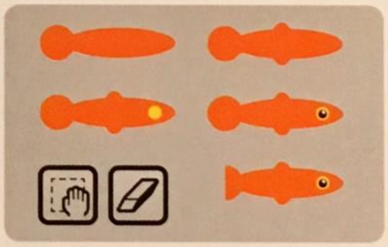
Set to fill in.

Choose orange.

Draw a wide ellipse.

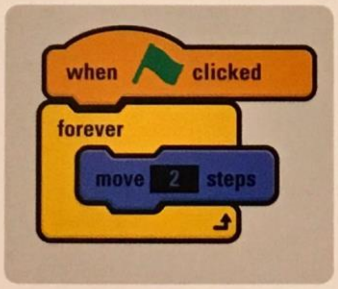


Draw more ellipses...and use the **Erase** or **Select** tool to delete the back of its tail.



3 Now click on the **Scripts** tab next to the red stop button. You're going to drag some code to the scripts area to make the fish swim forward once the program starts.

Drag the **When green flag clicked** block from the **Events** group.



```
when ⚑ clicked
forever
    move 2 steps
```

Drag the **Forever** loop block from the **Control** group and the **Move** block from the **Motion** group.

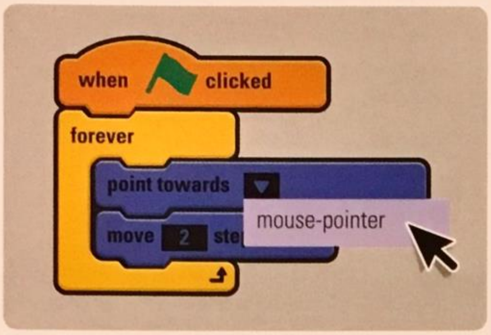
Change the **Move...steps** value to 2 to slow the fish down.

Click the green flag (near the top of the screen) to test your code.

4

To make the fish change direction, drag the **Point towards** block from the **Motion** group into the loop. Set it to "mouse pointer."

Test your code by clicking the green flag!



```
when 🏁 clicked
forever
    point towards ▼
    move 2 ste[ps]
```

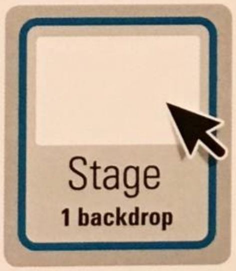
Download our robots to use as sprites on Scratch! Go to http://www.qed-publishing.co.uk/extra-resources.php or scan this:

Every time the loop repeats, it makes the fish point towards the mouse pointer. It also moves it every loop. Without loops, the game wouldn't work!

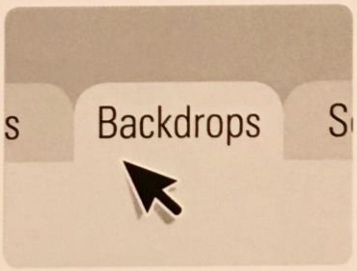
5

Now draw your own background picture for the game.

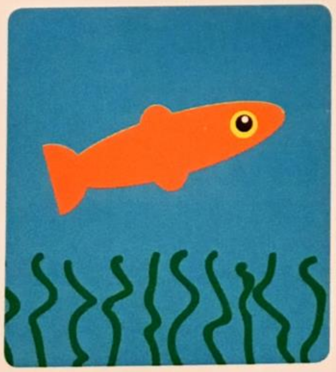
First click **Stage**.



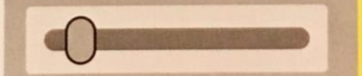
Then click **Backdrops**.



Click the **Fill** tool and choose a blue color. Now click the background to color it in.



Use the **Brush** tool to draw some reeds.



Use the **Line width** slider to change the size of the reeds.

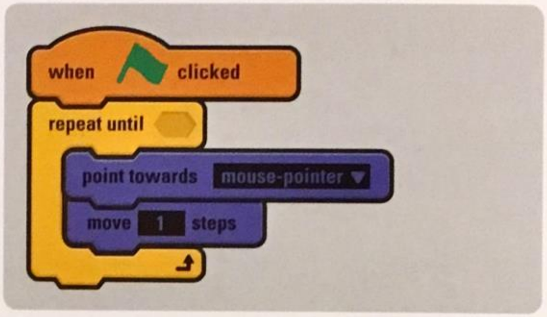
Your program is now complete! Click the green flag icon at the top of the screen to start playing.

REPEAT UNTIL...

Sometimes we need to stop loops when something happens—for example, if a player in a game bumps into a wall. To program things like this, we use a "repeat until" loop. We are going to code a simple maze game to learn how to use this technique.

How to code a maze game

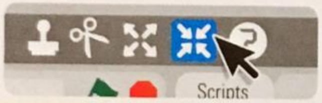
1 Start Scratch. Drag code to the scripts area to make the Scratch sprite move slowly across the screen, pointing towards the mouse pointer.



```
when ⚑ clicked
repeat until < >
    point towards [mouse-pointer ▼]
    move (1) steps
```

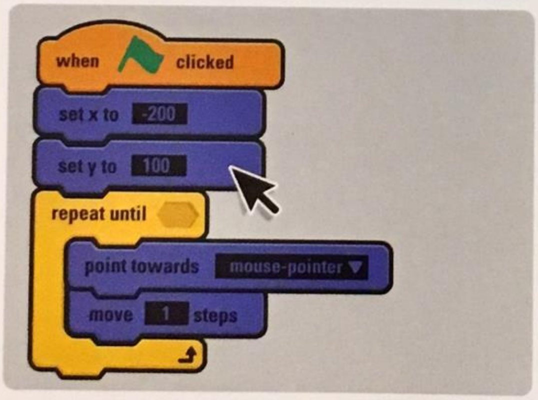
For help, see steps 3 and 4 on pages 44–45. But use a **Repeat until** block instead of a **Forever** block. Change the speed of the sprite to move 1 step each loop.

2 Make the Scratch sprite smaller by clicking the **Shrink** icon at the top of the screen, and then clicking the Scratch sprite several times.

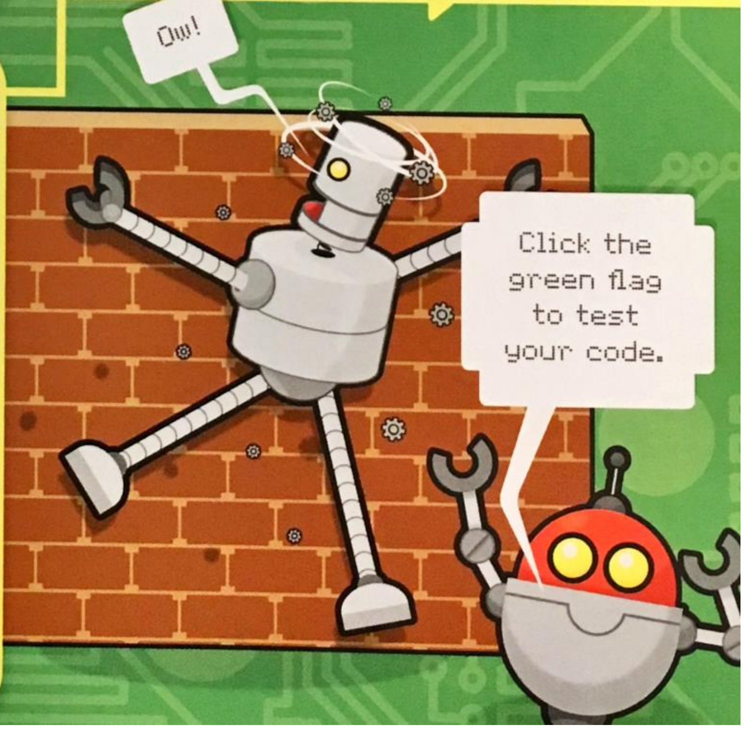


3 To make the sprite start in the same place each time, drag the **Set x to** and **Set y to** code blocks from the **Motion** group to the scripts area.

Experiment with changing the **Set x** and **Set y** values.



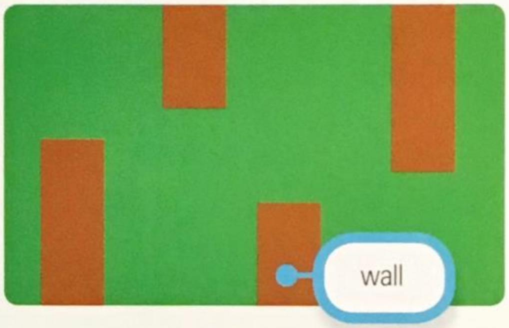
```
when ⚑ clicked
set x to (-200)
set y to (100)
repeat until < >
    point towards [mouse-pointer ▼]
    move (1) steps
```



4

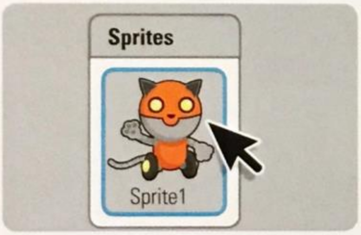
Draw a simple background for the game. Look at step 5 on page 45 for help getting started.

Use the **Rectangle** tool to draw some walls. Make them all the same color.

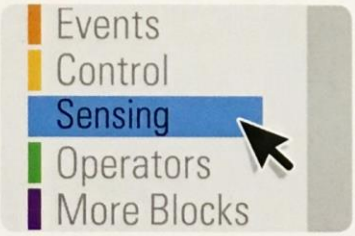


5

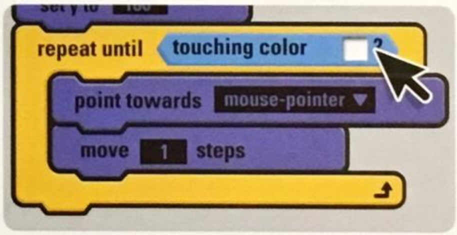
The **Repeat until** block will loop forever because we haven't told it when to stop yet. It needs to repeat until the sprite touches a brown color—the wall color.



Click on the Scratch sprite icon and then the **Scripts** tab to bring your code back.

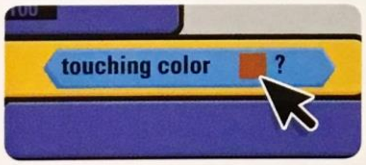


Click the **Sensing** group.



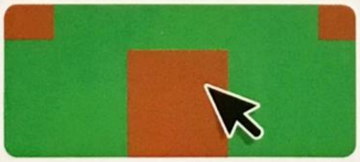
```
set y to 100
repeat until <touching color ?>
    point towards mouse-pointer ▼
    move 1 steps
```

Drag a **Touching color** block onto the top of the **Repeat until** block.

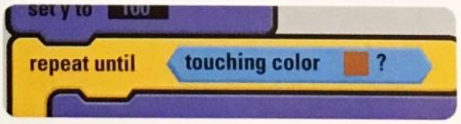


```
touching color ?
```

Click the colored square then choose the color to check for...



...by clicking one of the walls.



```
set y to 100
repeat until <touching color ?>
```

Now your game will play until the sprite hits a wall. Test it by clicking the green flag at the top of the screen. To play again, drag your sprite away from the wall.