

2

Drawing with Scratch

Find your bearings on the Scratch Stage, learn how to move sprites and draw with them, and discover how loops make it easy to repeat parts of your program. You'll also make an interactive art program called Rainbow Painter.

- 28** Understanding coordinates
- 30** Changing a sprite's position
- 31** Using the pen
- 33** Drawing a house in Scratch
- 35** Using directions to move
- 36** Keeping sprites upright
- 38** Drawing using directions
- 39** Making shapes with Repeat
- 40** Putting loops inside loops
- 41** Creating Rainbow Painter

Hot tip

You can click and drag your sprites around the Stage to reposition them while you're writing your programs. It's a good idea to use the "go to x:0 y:0" block to position them in the program, though, otherwise they might start off somewhere unexpected. If your sprite moves in the program, it won't be in the same position the second time the program is used. Other people can experiment with your program, too, and might move sprites around before it starts.

Don't forget

An easy way to remember which way around x and y should be is that "x is a cross" (and "across").

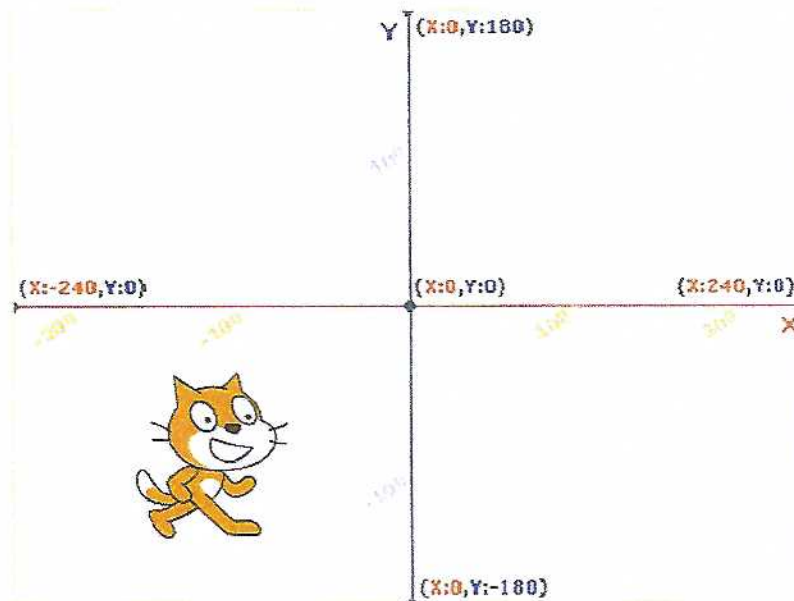
Understanding coordinates

You can use the sprites in Scratch to draw on the Stage. It's a great way to familiarize yourself with how to move sprites, and the technique can be used to create designs that your sprites can interact with, as you'll see when we make a game in Chapter 3.

First, let's take a look at how you position sprites on the Stage. Each position on the Stage has a grid reference, similar to those used on maps. The position across the Stage horizontally is called x, and the position up or down the Stage is called y.

When you start a new project, the cat is in the middle of the Stage, and this position has the grid reference x=0 and y=0.

Here's a map of the grid references on the Stage, after I've moved the cat to the bottom left quarter of the Stage:



As you can see, the numbers for the y coordinate run from -180 at the bottom of the Stage, up to +180 at the top of the Stage.

The numbers for the x coordinate run from -240 at the left edge of the Stage, up to +240 at the right edge of the Stage.

- 1 Click the **Motion** button above the Blocks Palette



- 2 To place a sprite at a particular position on screen, use the **go to x:0 y:0** block. This block is most often used to place a sprite in its starting position. Click it and drag it into the Scripts Area



- 3 The numbers in the block return a sprite to the middle of the screen, but you can change the coordinates. Click the white space beside x: and enter -120. Click the white space beside y: and type -90



- 4 Click the block to see the cat go to its new position, in the bottom left quarter of the screen
- 5 Try changing the numbers in the block and clicking it to see where the cat moves around the Stage

Hot tip



To start a new project, click the File menu and then click New. If the File menu isn't showing (in Scratch 2.0), click Create in the blue menu at the top of the screen.

Hot tip



You can change the numbers in the "go to x:0 y:0" block in the Blocks Palette, and Scratch sometimes changes them, too. If you're often reusing the same coordinates, that might save you time, because you can then drag the same coordinates in with your block. In this book, I'll assume this block is set to x:0 y:0 in the Blocks Palette. You can easily change the values in it, so it doesn't matter if you see something different on your screen.

Hot tip

You can use these blocks for animation. Just change a sprite's position a little bit. It'll appear to move from the old position to the new position.

Beware

If you change the x or y position by too much, you'll break the illusion of animation. It's usually better to use lots of small movements than one big one.

Beware

In your program, any blocks joined underneath the "glide" block won't start until after the glide has finished. Using time values of more than a second can slow down your program significantly.

Changing a sprite's position

There are several blocks you can use to change a sprite's position. To find them, click the **Motion** button above the Blocks Palette:

- **glide 1 secs to x:0 y:0:** This block makes your sprite glide across the screen to its new position. It travels in a direct and straight line. You can change the x and y coordinates in the block, and change how long the movement takes from 1 second to a value like 0.5 or 0.25 (for a faster glide), or to a bigger number for a slower one.

- **change x by 10:** This block makes your sprite move 10 positions to the right on the Stage. It doesn't affect its y position, and works independently of which way the sprite is facing. To increase how far the sprite moves, use a bigger number.

- **change x by -10:** To make your sprite move left, edit the value in the **change x by** block to make it a negative number. To make the sprite move further, use a bigger negative number.

glide 1 secs to x: 0 y: 0

change x by 10

set x to 0

change y by 10

set y to 0

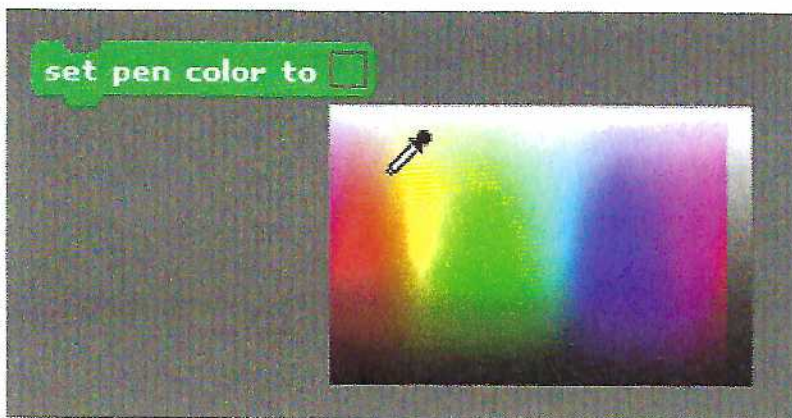
- **change y by 10:** This block makes your sprite move 10 positions up the Stage. It doesn't affect its horizontal position, and also works independently of which way the sprite is facing. Change the 10 to a bigger number for a bigger movement.
- **change y by -10:** Similar to the way you make a sprite move left, you can make a sprite move down the screen by changing the 10 in the **change y by** block to a negative number. For a bigger movement, use a bigger negative number.
- **set x to 0:** This block changes your sprite's horizontal position to a specific number without affecting its vertical position.
- **set y to 0:** This block changes your sprite's vertical position to a specific number without affecting its horizontal position.

Using the pen

The pen in Scratch enables a sprite to draw a line as it moves around the Stage. The blocks are:

- **clear:** This clears all the drawing on the Stage, but doesn't disturb the background or any sprites.
- **pen down:** This is like putting the pen down on a piece of paper. After you use this, your sprite leaves a line wherever it goes.
- **pen up:** This block stops your sprite from drawing as it moves. Sprites start with the pen up, so they won't leave a line unless you put the pen down.
- **set pen color to [color]:** This block enables you to use a pipette to choose which color you'd like to use with the pen. Click the box of color inside the block and then click a color somewhere else on the screen to use it. In Scratch 1.4, you're also shown a palette to choose colors from (see below).

Scripts Costumes Sounds



Beware



When you come to add multiple sprites in your projects later, each sprite will have its own pen and its own settings for the pen color and size, and for whether the pen is up or down.

Hot tip



If you're using the pen in your program, it's a good idea to start with the "clear" block so you always start with a clean Stage.

Hot tip

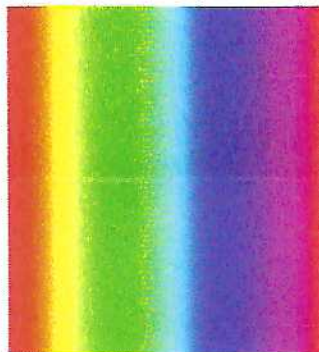
Click the numbers in the blocks and you can edit them to different numbers.

Don't forget

The Pen blocks are colored dark green, and you show them in the Blocks Palette by clicking the Pen button above it.

- **change pen color by 10:** Each color has a number, and the colors are arranged like a rainbow. Use this block to change the pen color to a higher number. Using a negative number, will change the pen color to a lower number.

- **set pen color to 0:** You can also set the color to a specific number, between 0 and 200. If you use higher numbers, they do work, but the same colors repeat. Color 350 is the same as 150, for example. To the right, you can see the pen colors from 0 on the left to 200 on the right.



- **set pen shade to 50:** You can change how bright or dark the color you use is. The starting value is 50, and the range of numbers goes from 0 to 100. Lower numbers darken the color, with 0 being black. Higher numbers lighten the color, with 100 being white. If you use numbers above 100, they do work, but the scale reverses every 100 numbers, so numbers from 101 to 200 go from light to dark, and from 201 to 300 go from dark to light again.
- **change pen shade by 10:** Use this block to change the shade, relative to its current value. Use a positive or negative number, depending on whether you want the number of the pen shade to go up or down.
- **change pen size by 1:** This block increases the pen size. Use a negative number to decrease it.
- **set pen size to 1:** Use this block to set the pen size to a specific width. Anything larger than 5 leaves a thick line. The maximum size is 255, but this is too big to be useful often.
- **stamp:** This block prints a copy of the sprite on the Stage, so when the sprite moves on, it will leave a picture of itself behind. It's just a picture of the sprite, not a copy of the sprite that you can control. In Chapter 4, you'll learn how to clone and duplicate sprites, so you can control the copies.

Drawing a house in Scratch

You can use what you've learned about moving sprites and the pen to draw pictures on the Stage. Follow these steps to draw a house:

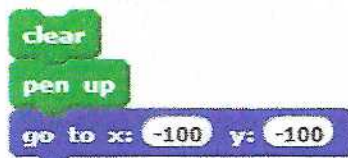
1 Change the background to xy-grid, which puts a grid onto the Stage to help you work out coordinates. You can find it in the Other category in Scratch 2.0, and it's in the top folder with all the other background folders in Scratch 1.4

2 Click your cat sprite, and click the **Scripts** tab to open the Scripts Area

3 There's a potential pitfall here: if the pen is down, then the sprite will draw a line you don't want as it moves into place. To fix that, lift the pen before moving the sprite. Click the **Pen** button above the Blocks Palette and drag the **clear** and **pen up** blocks into the top of your script



4 Click the **Motion** button above the Blocks Palette and drag in the **go to x:0 y:0** block. Change the numbers in it to x:-100 y:-100, so it puts the sprite in the right starting position



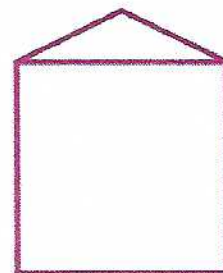
5 Next, we should get ready to draw. Use **pen down** to start drawing, and set the pen color and size to something that will stand out from the grid pattern. Add these blocks to the bottom of your script



Hot tip



The picture is drawn almost instantly. To slow it down so you can see what's happening, add some "wait" blocks between the movement blocks. You can find the "wait" block among the Control blocks.



Above: The house we're drawing.

...cont'd

Hot tip



Why not see if you can add windows and a door? Remember to lift the pen before repositioning the pen inside the square.

6 Our sprite is now positioned in the bottom left corner of our house. To draw the square, move the sprite up (increasing the y coordinate), right (increasing the x coordinate), down (decreasing the y coordinate), and left (decreasing the x coordinate). Each line has a length of 200, so the sprite ends where it started. Click the **Motion** button above the Blocks Palette and add these blocks to your script

```
change y by 200
change x by 200
change y by -200
change x by -200
```

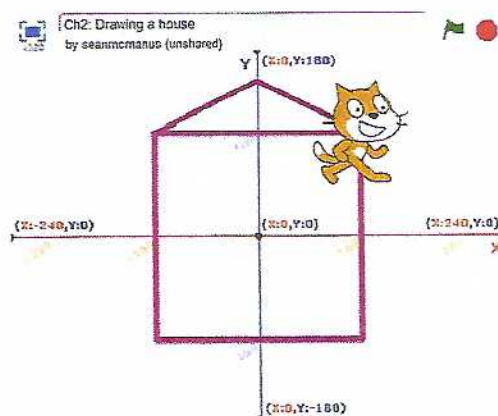
7 To draw the roof, we need to move the cat to the top left corner of the house first. If we change the x position and y position separately, we'll leave two lines in an L shape. To draw a straight diagonal line, we use the **go to x:0 y:0** block to move to the tip of the roof, and use it again to move to the right corner of the house. Drag in these blocks

```
go to x: -100 y: 100
go to x: 0 y: 150
go to x: 100 y: 100
```

8 Click your script to run it, and you should see the scene below on the Stage

```
clear
pen up
go to x: -100 y: -100
pen down
set pen color to red
set pen size to 5
change y by 200
change x by 200
change y by -200
change x by -200
go to x: -100 y: 100
go to x: 0 y: 150
go to x: 100 y: 100
```

Above: the finished program



Using directions to move

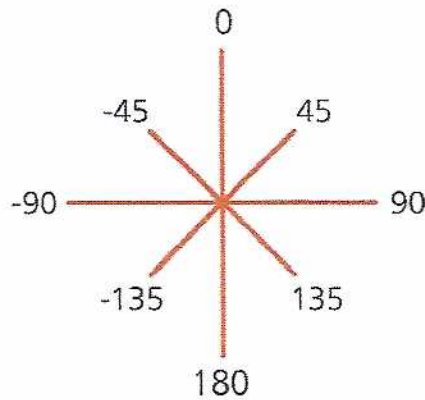
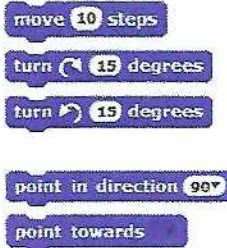
As you saw last chapter, there is another way you can move sprites in Scratch, which is to point them in a particular direction and then move them forwards in that direction. Here are the blocks you use for that:

- **move 10 steps:** This moves your sprite forwards. In the case of the cat, this usually means the direction it's facing. You can change the number of steps, and a negative number makes the sprite go backwards.
- **turn clockwise 15 degrees:** This rotates your sprite clockwise by 15 degrees. You can change the number of degrees. You can use a negative number to turn the other way, but you will rarely need to, because of the next block.
- **turn anti-clockwise 15 degrees:** This block turns your sprite in the other direction.
- **point towards:** This block is used to point a sprite towards another sprite, or the mouse pointer.

- **point in direction 90:** This makes your sprite point in a particular direction. The direction numbers go from -179 to 180. There is a menu in the block you can click to choose one of the most common directions: right (90), left (-90), up (0), or down (180). You can also type in a number of your choice.

Scripts Costumes Sounds

Motion
Looks
Sound
Pen
Data
Events
Control
Sensing
Operators
More Blocks



Don't forget



Which is better? Moving using coordinates, or using directions? There's no right answer to that: it depends on your program. Part of the art of programming is to choose the best approach for each program you write. We'll use both methods in this book, sometimes in the same program.

Hot tip



When you turn your sprite, Scratch makes sure the numbers make sense automatically. For example, when the direction is 180 (down) and you rotate clockwise by 90 degrees, Scratch turns the resulting direction into -90 (left) instead of 270, which would be mathematically correct, but is outside the range of directions Scratch uses.

Keeping sprites upright

One of the problems with rotating sprites is that it can look strange, even in the context of a game. When you rotate the cat to move up (direction 0), it looks like it's climbing the walls:

Hot tip



Although you can change the rotation style in Scratch 2.0 using the information panel, it's better to do it using a block in your program, just in case someone has changed the rotation style before using the program.



Worse still, when its direction is set to left (-90), the cat looks like it's walking on its head.



To avoid this problem, you can change the rotation style of the sprite. There are three styles to choose from:

- **all around:** This is the “normal” setting, that makes the sprite turn all the way around depending on its direction, and can make the sprite appear to defy logic and gravity. The sprite turns to face the direction it will move in.
- **left-right:** This keeps your sprite's feet on the ground and makes it face either left or right, but never up or down or at an angle. You can move the sprite in any direction, but it will always face right or left.
- **don't rotate:** This stops the sprite from visibly changing when its direction changes. You can still change its direction and move it in that direction, but the sprite will always look exactly the same.

Hot tip



The right way up for a sprite in Scratch is facing right. That means you should always draw your own sprites facing right (see Chapter 4).

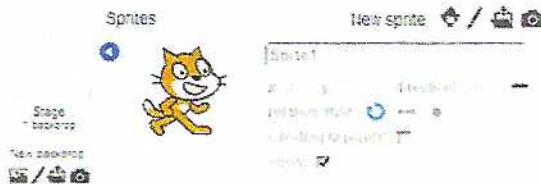
...cont'd

Setting rotation style in Scratch 2.0

There is a new block in Scratch 2.0 called **set rotation style**. Click the **Motion** button above the Blocks Palette and then use the scrollbar (on the right edge of the Blocks Palette) to find this block near the bottom.



You can also change the rotation style of the sprite by clicking the sprite in the Sprite List and then clicking the **i** (for information) button in the corner of it. Click one of the three rotation style options to select it, and then click the back arrow (white triangle on blue circle) in the top-left of the information panel to close it.



Setting rotation style in Scratch 1.4

To change the rotation style in Scratch 1.4, click the sprite in the Sprite List to select it (if necessary), and then click one of the three buttons above the Scripts Area. The buttons are on the left of the picture of your sprite. The first button changes the rotation style to all around, the second button to left-right, and the third button turns off rotation.



Hot tip



You can rotate your sprite without using blocks if you need a quick fix. In Scratch 2.0, use the circle icon beside the direction in the information panel. In Scratch 1.4, use the picture of your sprite above the Scripts Area. In both cases, click and drag in a circle to rotate the sprite.



Above: The information button on the cat sprite.

Drawing using directions

You can draw pictures by moving the sprite with directions too. Earlier in this chapter, you saw how to draw a house by moving the cat to specific points on the Stage.

You can also use directions to achieve the same effect. Here's a program to draw a square:

Don't forget



You can duplicate blocks by right-clicking them (See Chapter 1).

```

clear
pen up
go to x: -100 y: -100
pen down
set pen color to red
set pen size to 5
point in direction 0
move 200 steps
turn 90 degrees
move 200 steps
turn 90 degrees
move 200 steps
turn 90 degrees
move 200 steps
turn 90 degrees
  
```

- The first six blocks are the same as we used when drawing the house, and just prepare our sprite for drawing.
- The next block turns it to point up the screen.
- Then we move it forwards 200 steps, which draws a line up the screen.
- We rotate 90 degrees right (so the sprite is facing right), move it again to draw the top of the square.
- Then we make two more turns and lines.

There's a lot of repetition in there: we have included the same two instructions four times in a row. It works, but it's laborious to create and hard to read. Luckily, there is a better solution.

Making shapes using Repeat

In our square drawing program, it would be far simpler to tell the program to go forward and turn right four times, than it is to list out every turn and movement.

The **repeat 10** block makes this possible. It is one of the Control blocks, which determine whether and how often things happen in a program. It's shaped like a bracket, and you put the blocks you want to repeat inside it. You change the number in the block's frame to say how often you want it to repeat the blocks inside.

Here is how you can use the **repeat 10** block to draw a square:



The blocks to move and turn sit inside the **repeat** block's bracket, and the **repeat** block is set to repeat them four times.

Repeating sections of program like this are often called loops. It's easier to see what's happening in this program than it was when we had so many drawing instructions, and it's easier to modify the program too.

What if you decide you need a hexagon instead of a square? You can just increase the number of times the sprite moves and turns, from 4 to 6 (for the 6 sides and angles), and also change the size of the turn to 60 degrees.



Hot tip

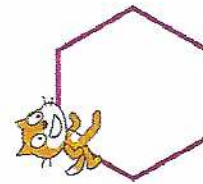


You can modify this loop to draw any shape with sides of the same length. To work out the angle of the turn, divide 360 by the number of sides. To alter the size of the shape, change the distance the sprite moves for each line.

Hot tip



For best results, add these example loops after the first six blocks from the previous example that set up the Stage.



Above: The square becomes a hexagon with a small change to the loop.

Beware

Take care with where you put your blocks. For example, if you put the block to change the pen color inside the inner "repeat" block that draws the hexagon, the pen color will change with each line. That doesn't matter so much with abstract demonstrations like this, but it can cause huge problems in more elaborate programs.

Hot tip

A loop inside a loop is called a nested loop.

Hot tip

The outer loop repeats 72 times because it turns 5 degrees each time. 72 times 5 degrees makes 360 degrees, a full circle.

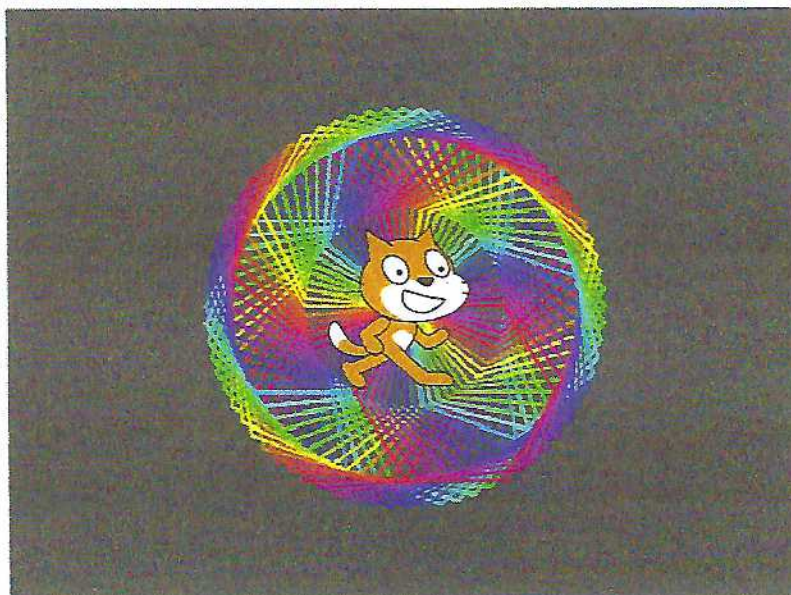
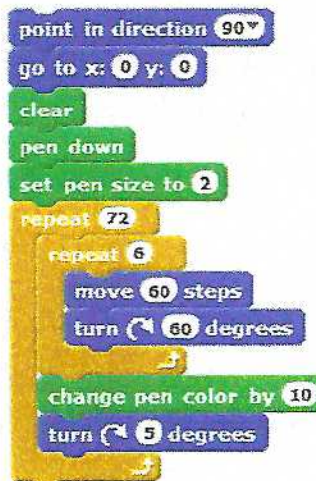
Putting loops inside loops

You can put **repeat** blocks inside each other, for example if you want to draw lots of hexagons, each one slightly offset from the previous one. Start a new project and try making this program. It creates a pattern of overlapping hexagons as shown below.

The first few blocks set the sprite up, facing right, in the middle of the Stage. Then we set the pen up, by clearing any previous digital doodles, putting the pen down and setting the pen size.

After that, we have the first of our loops, which will repeat 72 times. If you look inside it, you can see the first thing it does is use another loop which repeats 6 times to draw a hexagon.

After that loop has finished and the hexagon is drawn, the pen color is changed and the sprite is turned by 5 degrees. Then the process repeats, with another hexagon drawn, the color changed and the sprite rotated until 72 hexagons are on screen.



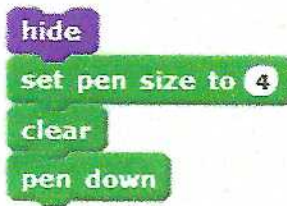
Creating Rainbow Painter

To close this chapter, here's a simple art program called Rainbow Painter that enables you to paint with a striped pen across the starry night sky. It shows you how simple it can be to make a program that interacts with the user. Once you've written it, you can use this program to make your own digital designs:

1 Start a new project, and change the background to stars. It's in the Space theme in Scratch 2.0 and the Nature folder in Scratch 1.4

2 Click the cat in the Sprite List and click the **Scripts** tab to open the Scripts Area

3 It's possible to hide a sprite, so it's not shown on the screen but it can still move around and draw on the Stage. We'll use a hidden sprite that follows the mouse pointer to draw on the Stage. You can use the cat sprite that stars in every new project for this. Click the **Looks** button above the Blocks Palette to find the **hide** block



4 Drag in the following blocks to set up the program

5 As well as the **repeat** block, there is a **forever** block which repeats whatever is inside it endlessly. It's another Control block, so click the **Control** button above the Blocks Palette to find it



6 Drag it in and join it to the program so far

Don't forget



The color of the blocks provides a hint about where you can find them in the Blocks Palette. To find the green blocks, click the green button above the Blocks Palette.

Hot tip



Notice that there's no slot on the bottom of the "forever" block for other blocks to lock into. That's because a forever loop never ends, so any blocks underneath it would never be used.

Hot tip



You can stop a program that runs forever like this, or any other program, by clicking the red hexagonal stop button above the Stage.

...cont'd

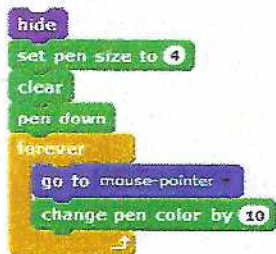
- 7 Drag in the Motion block **go to** and click the menu in it to choose the mouse-pointer. This moves your invisible sprite to the mouse pointer. As you move the mouse, the sprite will keep moving to the mouse's new position and will leave a line behind it

go to mouse-pointer

- 8 Drag in the block to change the pen color by 10, so the color is continuously changing

change pen color by 10

When you run the program and move your mouse pointer over the Stage, you'll leave a rainbow colored line. Below you can see a face drawn using Rainbow Painter.



Above: The final program looks like this.

