

WRITE YOUR OWN RASPBERRY PI GAME



Follow **Stuart Andrews'** step-by-step tutorial to writing your own game on the Raspberry Pi and enter our fabulous competition to win £250

The £15 Raspberry Pi is a computing sensation, but it was originally designed with one key intention: to inspire a new generation to look beyond games consoles, smartphones and tablets, and embrace the way of code.

The Raspberry Pi comes with a selection of programming environments pre-baked into its Debian-based Linux distribution, but the easiest way to get started is with Scratch. Created by the Media Lab at MIT, Scratch was designed to teach children the fundamentals of programming, without the need to learn complex syntax.

Scratch allows users to drag and drop characters and objects into a game environment, then program their actions using building blocks, which are grouped into colour-coded categories such as Controls, Motion, Operators and Sensing. By dragging and dropping these into place you can create surprisingly sophisticated interactive stories, animations and games, then share them with a global community of students and developers.

We also want you to share your completed games with us. We've teamed up with the creators of the Raspberry Pi to run our very own Scratch game competition (see p42).

First, however, read our tutorial on how to get started.

Getting started with Scratch

In this feature, we're aiming to produce an arcade game – we're calling it Crustacean Storm – while giving you a few pointers on how you may want to take it further. We hope it will encourage budding programmers of all ages to try their hand at coding something bigger and better, leading you on to more complex programming tasks.

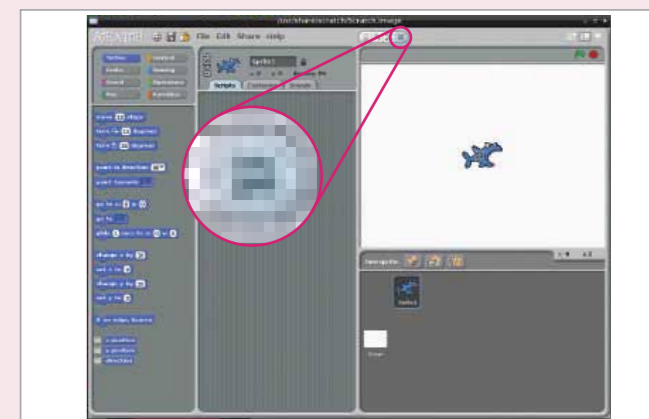
If you've never come across Scratch before, it might be worth looking at our previous feature on programming with it (www.pcprouk.co.uk/links/216scratch), which covers some of the basic functions. However, Scratch is extremely intuitive and easy to pick up; its scripts are comprehensible, and you can learn

a lot just by downloading games and finding out what makes them tick. Don't worry if you don't have a Raspberry Pi: Scratch will run happily under Windows, OS X and Linux, and even on the most modest of PCs. You can download it and find plenty of example programs at <http://scratch.mit.edu>.

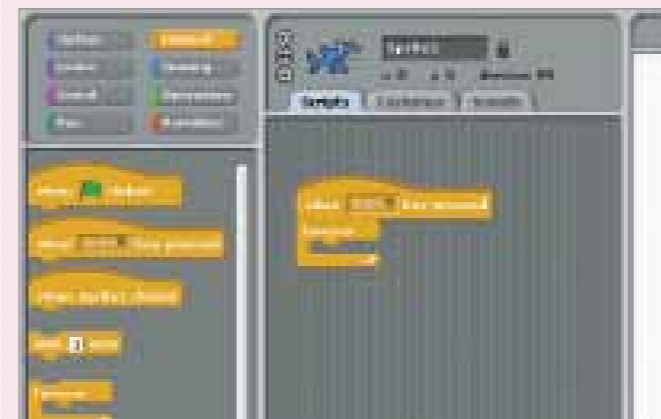
Scratch is only the start. The Raspberry Pi is also great for working with Greenfoot – another highly visual tool that teaches real Java programming – and Python. This general-purpose, high-level programming language remains reasonably accessible, and with the Pygame modules – also preinstalled on the Raspberry Pi – it's possible to create even more complex and fully featured games. If you're interested in learning more, see our Python tutorial on p58. However, Scratch teaches a good range of basic concepts that you can apply to other languages and development environments that you might come across later on. Think of this as the start of a long journey leading to knowledge, enlightenment and, for some lucky youngsters, a future career.



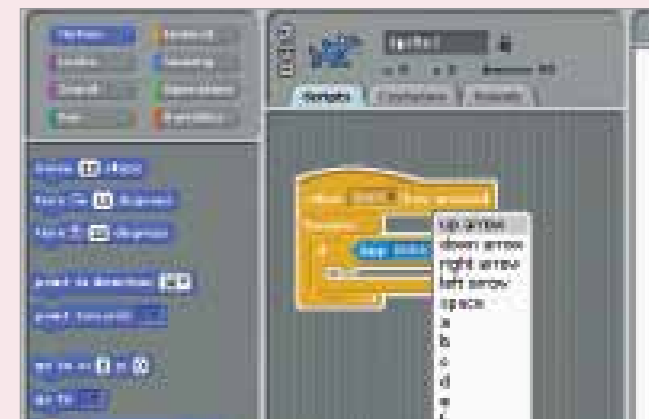
1 First, start up Scratch. Now, you can't have a cat starring in a game called Crustacean Storm. Right-click on the sprite and delete it. Go to the New Sprite bar above the bottom-right-hand window, select the middle button to "Choose new sprite from file", go to the Costumes | Animals folder and select "shark 1-b".



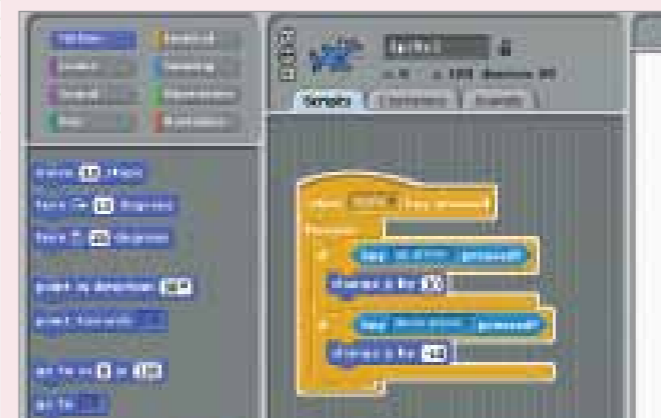
2 As he is, the shark is too big. Let's shrink him. Select the Shrink tool from the toolbar (as shown above) and click about 30 times on the shark until he's roughly the size shown here. Now it's time to name the sprite. Go to the name bar above the central window and type to change its name from "Sprite1" to "Shark".



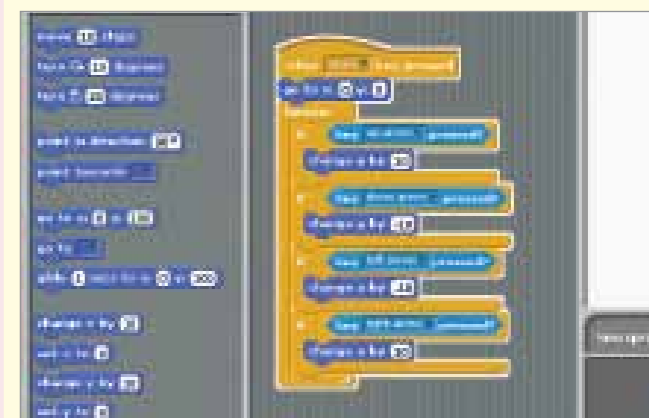
3 It's time to put the player in control. First, select the Control category from the top left of the screen, then drag the blocks "when 'space' key pressed" and "forever" into the shark's script window, as shown here. We'll use the spacebar as the start key for our game.




4 Now drag the "if" block from the Control category into the "forever" block, then go to the Sensing category and drag the "key 'space' pressed" block to the control point on the "if". Use the dropdown menu to change "space" to "up arrow".

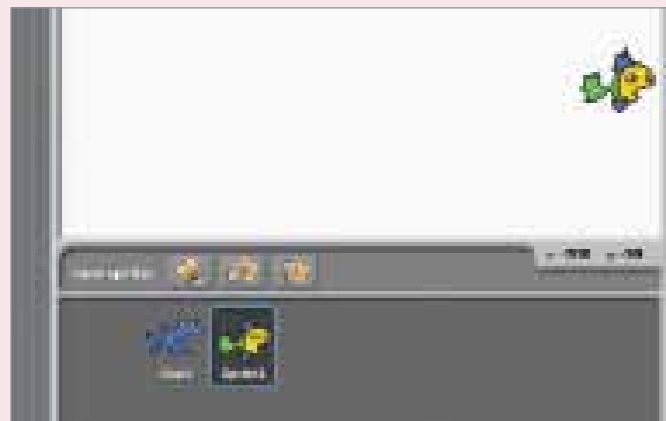


5 That tells Scratch to look out for the Up key, but now we need to assign the response. Go to the Motion category and drag the "change y by 10" block into the "if" block. If the spacebar then Up arrow is pressed, our shark now moves. You can now repeat this for the Down arrow, replacing the "change y by" value to -10.



6 To make the left and right arrows work, repeat steps 4 and 5, but replace the "change y by" block with the "change x by" block and set the values -10 and 10 respectively. Finally, our star needs a start position. Grab the "go to x: 0 y: 100" block from Motion and drag it into the place shown. Change both values to 0.

 **Raspberry Pi – advanced class**
Now tackle our In Depth tutorial on p58



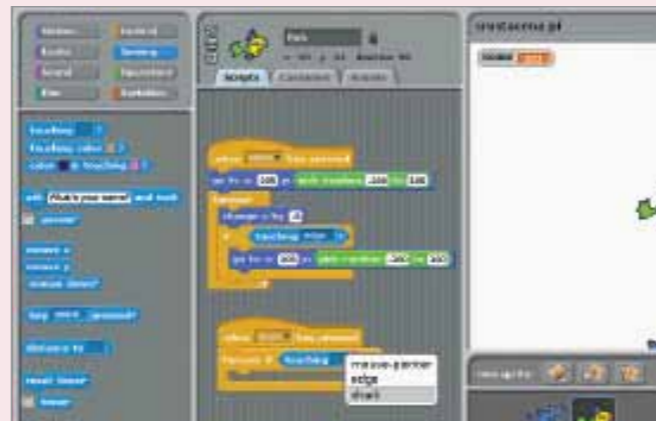
7 Now click again to “Choose new sprite from file” and select a fish, before shrinking it as we did in step 2 and renaming it “Fish”. Ensure it’s selected, as shown, and drag “when ‘space’ key pressed” into the central scripts window. Go to Motion and drag the “go to x:0 y:0” block into place below.



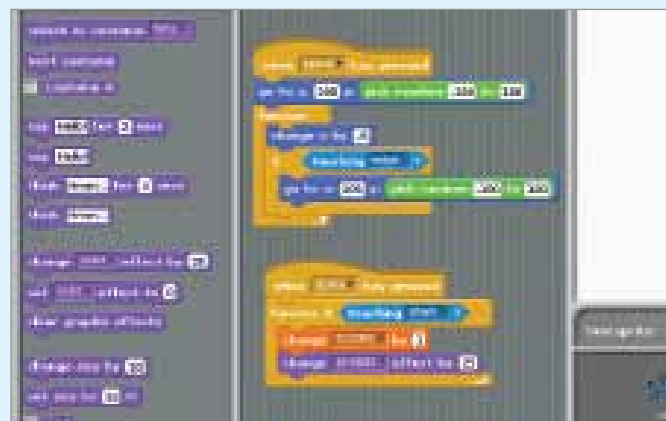
8 Set “x” to 205. To stop the fish always appearing in the same place, go to Operators and drag the “pick random 1 to 10” block into the “y” value. Set values to -180 and 180. Next, go to Control and drag in “forever” to snap in below “go to”, then go to Motion and pull the “change x by” block inside it. Change x to -5.



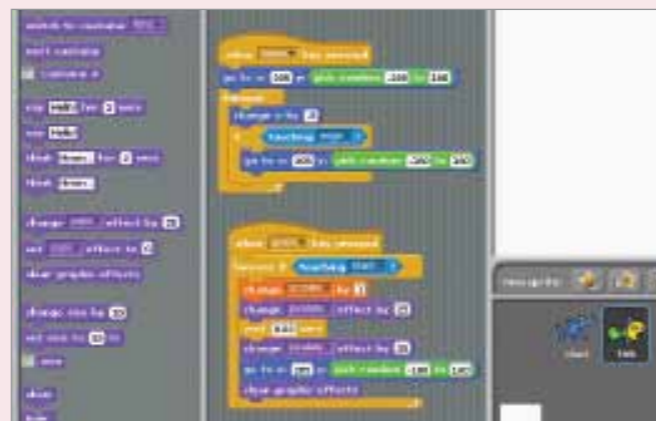
9 We want our fish to reappear if it hits the side of the screen. Drag an “if” block from Control inside the “forever” block, then go to the Sensing category and drag the “touching” block onto the “if”. Select “Edge” from the “touching” dropdown. Drag in a “go to x & y” block from Motion and configure it as in step 8.



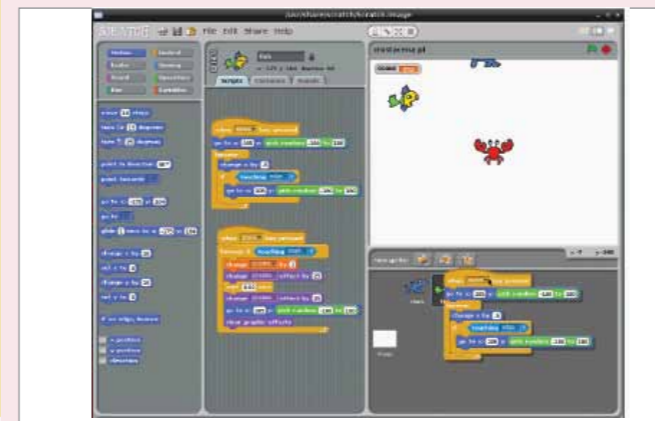
10 Click on Variables and then the “Make a Variable” button. Call it “Score”. The score will now appear in the top-left corner of the game screen. Add a new “when ‘space’ key pressed” block from Control, clip on a “forever if” block; now go to Sensing and drag in a “touching” block. Choose “shark” from the dropdown.



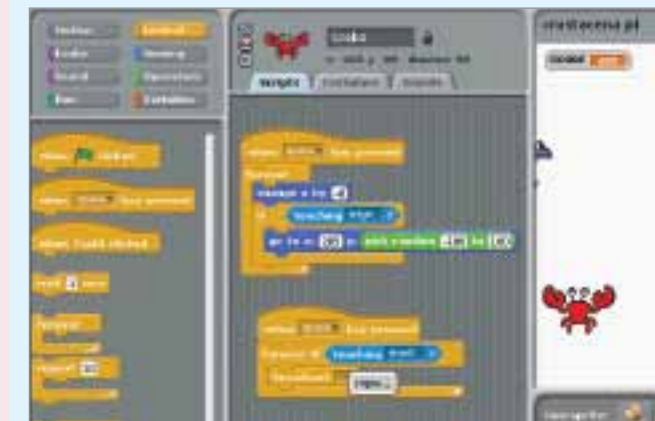
11 What happens when the fish meets the shark? Our piscine friend gets eaten and the score creeps up. Go to Variables and drag in the “change score by 1” block, then go to Looks and drag in the “change ‘color’ by 25” block. Change the “color” option to “pixelate”.



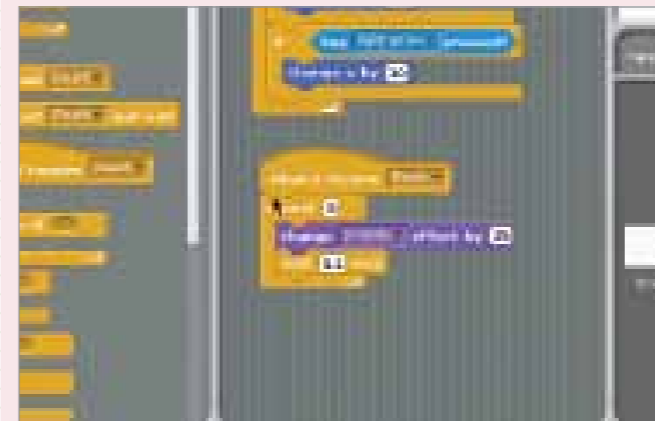
12 It’s a nice effect, but over far too soon. Drag in a “wait ‘1.00’ secs” block from Control and change the wait duration to 0.02. Then drag in another “change ‘color’” block and change the “color” to “pixelate”. Add the “go to x” block with a random “y” position (as in step 8), and a “clear graphic effects” block from Looks.



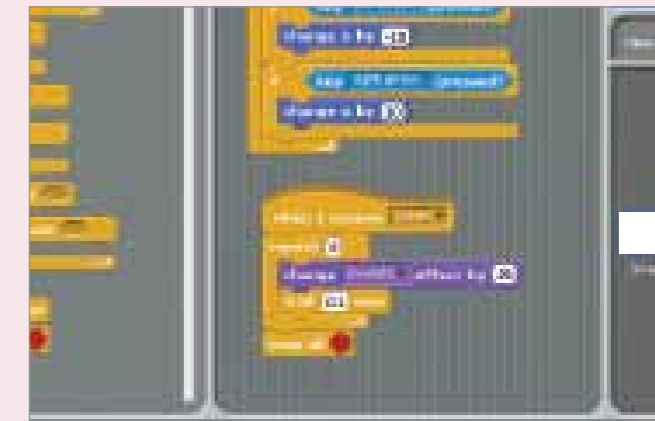
13 It’s crab time! Add a new sprite – crab 1-a – and shrink him. We can make the crab move by duplicating the fish script. Click on the Fish sprite, then click the Duplicate icon on the top toolbar, click on the first script for this fish, and drag it over to the crab. Go to the crab’s script and amend “change x by” value so it reads -8.



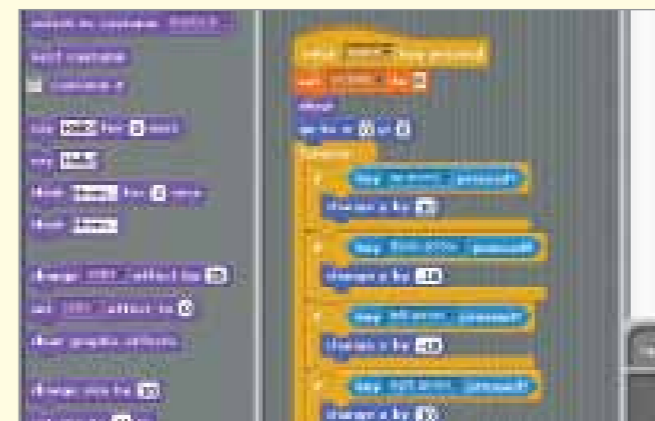
14 The crab moves, but we want him to be deadly. Drag in a new “when ‘space’ key pressed” block, then add a “forever if” block. Go to Sensing and add a “touching” block; use the dropdown to select the Shark sprite. Go to Control and drag in a “broadcast” block. Select New from the dropdown and call it “Death”.



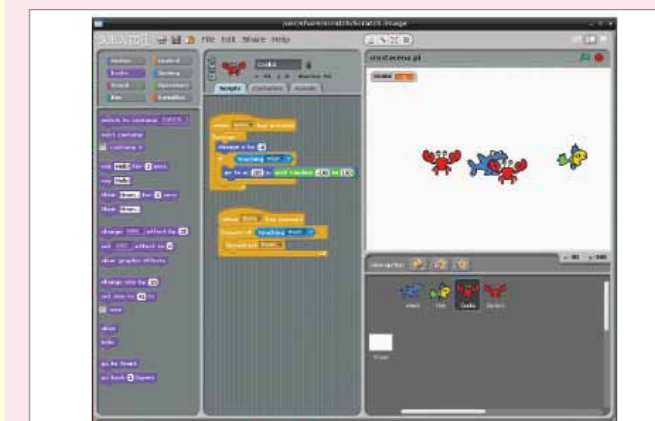
15 Death equals Game Over. Select the Shark sprite, then go to the Control category and add the “when I receive” block. Select “Death” from the dropdown. Now add a “repeat” block from Control, go to Looks and add the same “pixelate” block as in step 12, and then a “wait” block.



16 Let’s stretch the shark’s demise out a bit. Set the “wait” value to 0.1 seconds, and set the value in the “repeat” block to 6. Add a “stop all” from Control to the end of this script. That will end the game when the shark is hit.



17 We need the score to reset every time the game is started, so drag a “set score to 0” block from Variables and a “show” block from Looks to the first shark script, placing them just below the initial “when ‘space’ key pressed” block.



18 We have one killer crustacean, but what about a swarm? In Scratch you can easily duplicate crabs. Click the Duplicate tool, then the crab, and make another crab or two. When you’re done, click the button in the top right to enter the full-screen presentation mode, press the green flag, then space – and start playing.

What next?

Where do you go from here? Adding a background would be one easy step (here we've simply edited the default background and used the Paintbucket tool to fill it with blue). Another would be to add sound effects, or even simple animation; both the shark and the crab have alternative costumes you can use, and it's simple to do using the "switch to

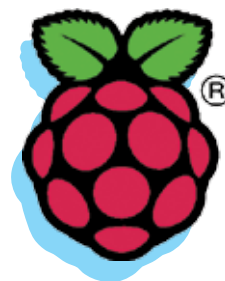
costume" block in the Looks category. By altering values it's also possible to speed up the crabs and fish, while you can duplicate both to make for a busier game. Why not try adding more hostile sea creatures with different movement patterns, or change the theme altogether? Or add titles and a proper Game Over screen. Scratch makes all this and more possible – the rest is up to you. ◆



It's easy to add more sophisticated graphics

COMPETITION

Enter our Raspberry Pi coding competition and you could win our £250 first prize!



We want you to **design your own computer game** – either using the Scratch software from this feature, or the more advanced Python used for our In Depth tutorial on p58. You can create your own shoot-em-up, a brain-taxing puzzler, or even a text adventure game for the 21st century – the choice is yours.

Raspberry Pi founder Eben Upton will help *PC Pro* editor Barry Collins to select the winning entry.



HOW TO ENTER

Simply email your finished game to competitions@pcpro.co.uk

If you're sending a Scratch-based game, simply attach the .rb file to your email. If you're coding your game in Python, send us a zip file containing all the code and other game assets, such as images.

The winner will receive a cheque for £250 and have their game published on the *PC Pro* website.

The closing date for entries is 10 October 2012.