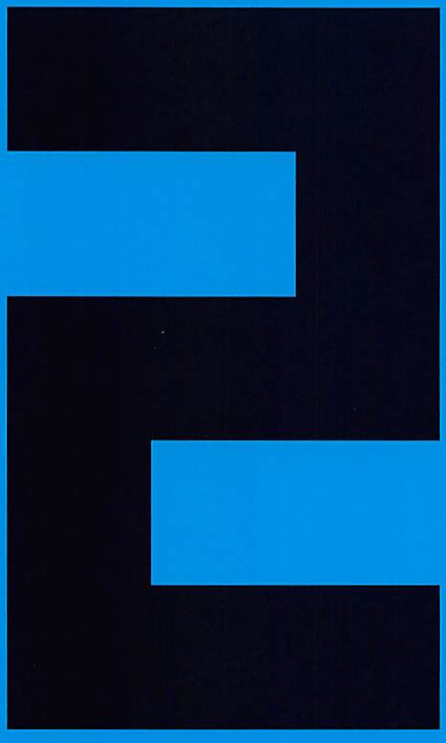
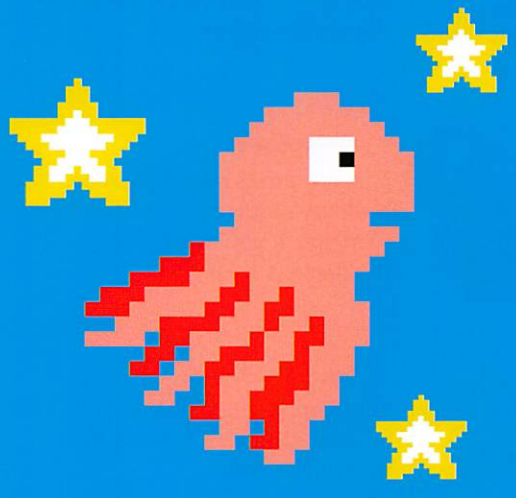


Star Hunter



How to build Star Hunter

Welcome to your first Scratch game: Star Hunter, a fast-paced, underwater treasure hunt. Just follow the simple steps in this chapter to build the game, then challenge a friend to beat your score.

AIM OF THE GAME

The aim of this game is to collect as many gold stars as you can. Use the cat to collect the stars, but watch out for deadly octopuses. You'll need to move quickly to succeed. The main sprites in the game are shown below.

◁ **Cat**
Move the cat around the screen with your computer mouse—the cat sprite follows the mouse-pointer.

◁ **Octopuses**
The octopuses patrol the seas but they swim more slowly than you. If you touch one, the game is over!

◁ **Stars**
These appear one at a time in random places. Touch a star to score a point.

GAME CONTROLS

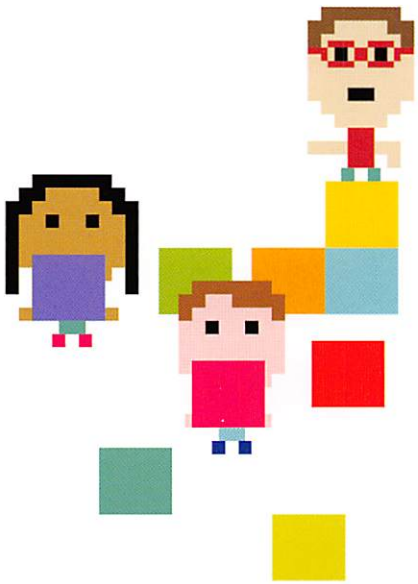
Use a computer mouse or touchpad to control this game.

◁ **Under the sea**

Star Hunter is set in the deep sea, but you can change the backdrop to anything you like, from outer space to a picture of your bedroom.

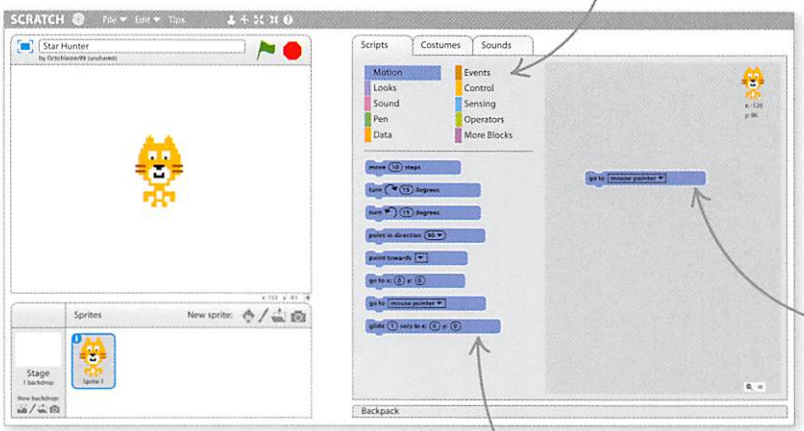
Building scripts

Like any Scratch program, Star Hunter is made by joining colored blocks like the pieces of a jigsaw puzzle. Each block is an instruction that tells a sprite what to do. Let's start by programming the game's main sprite: the cat.



1 Start Scratch and choose either "create" or "New Project". You'll see a screen like the one below, with the cat sprite in place. In the middle is a set of blue instruction blocks.

Clicking the buttons here reveals different sets of blocks.

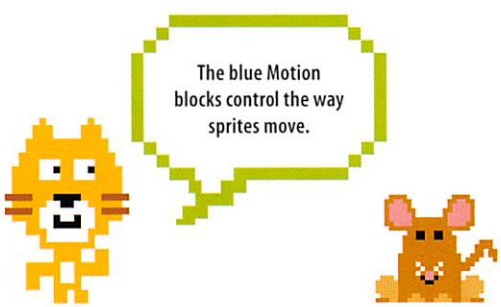


Drag your chosen blocks here to build a script.

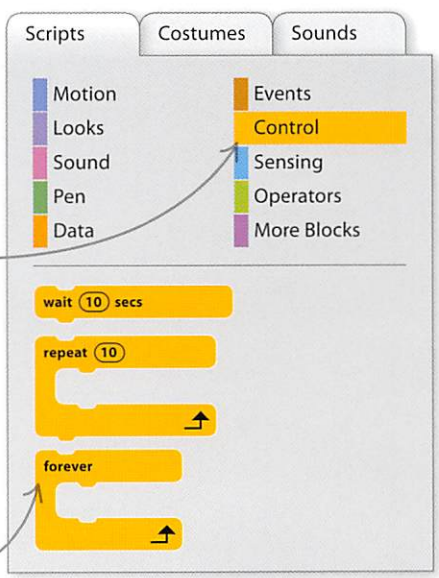
Choose blocks from the list in the middle.

2 We'll program the cat to move wherever the player moves the computer mouse. Click on the "go to mouse-pointer" block and drag it to the right part of the screen—the scripts area.

Some blocks include a drop-down menu.



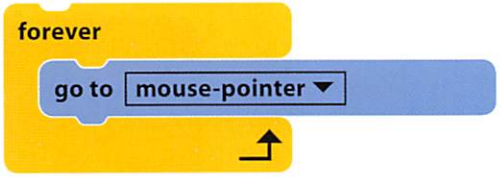
3 Now select the yellow Control button and look for a "forever" block.



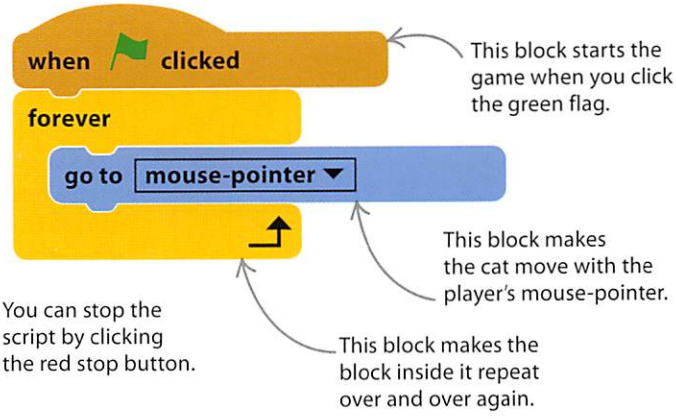
Click Control to reveal the yellow blocks.

Drag the "forever" block to the scripts area.

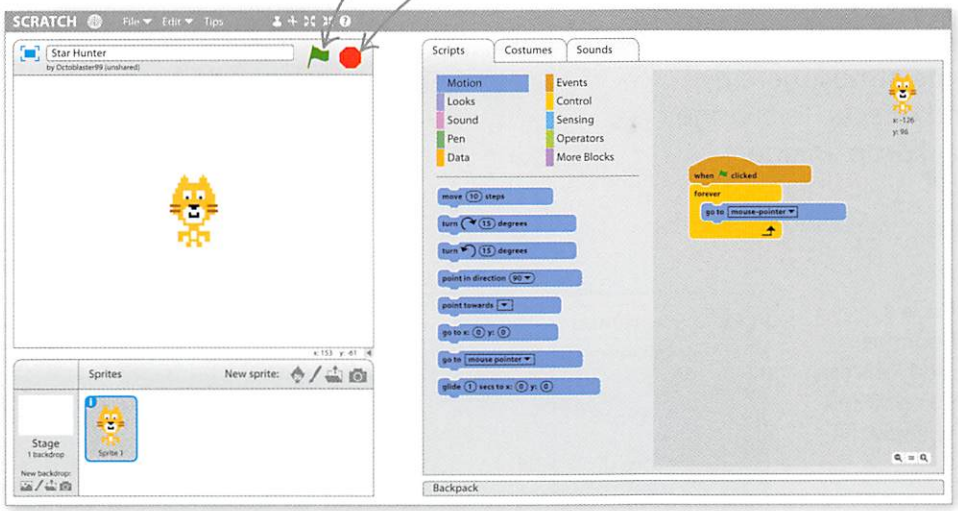
4 Drag it to the right and drop it over the blue block. It will wrap around it like this:

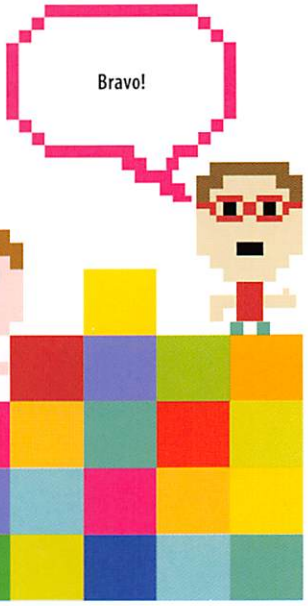


5 Next, select the brown Events button. Look for a block with a green flag. Drag it to the right and add it to the top of your script. Read through the script and think about what each block does.

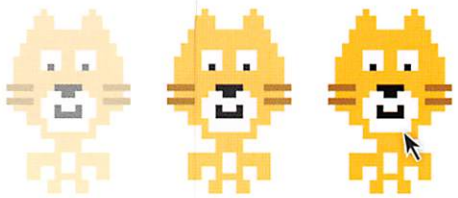


6 Now look at the top right of the stage—you'll see a green flag. Click this to run your script.





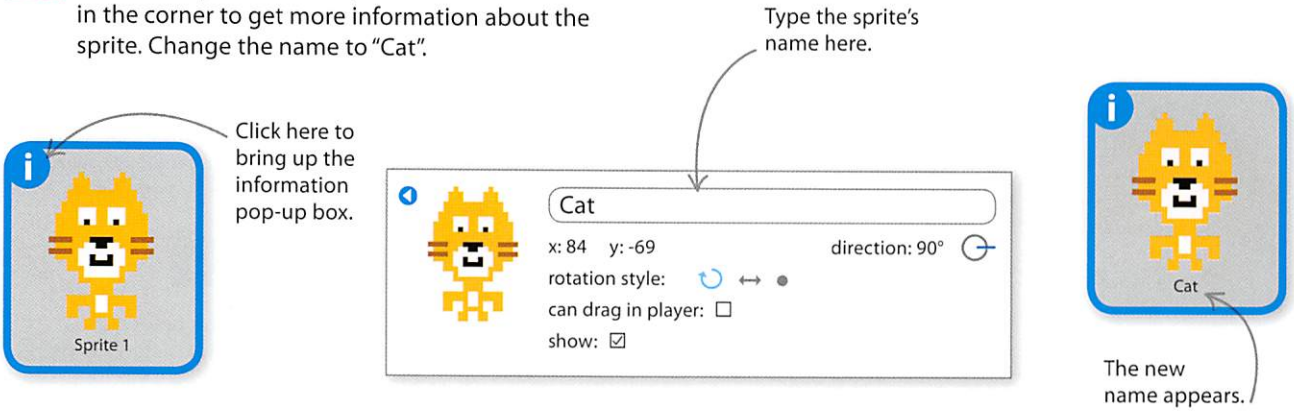
7 Move your mouse and watch what happens. If you followed all the steps, the cat will move with the mouse-pointer around the stage.



▷ **Well done!**

You have created your first Scratch project. Let's add some more things to the project to build a game.

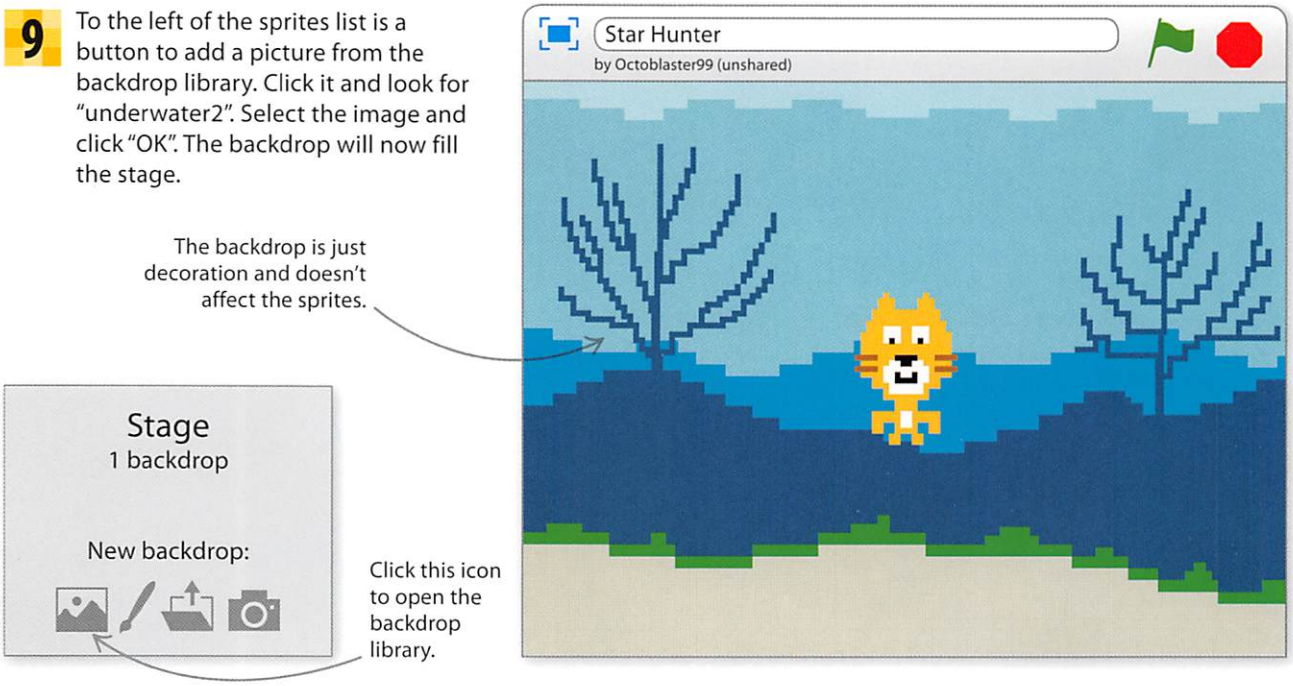
8 The cat is called "Sprite1". Let's fix that. In the sprites list, select Sprite1 (the cat) and click on the blue "i" in the corner to get more information about the sprite. Change the name to "Cat".



Setting the scene

At the moment, the stage is just a boring white rectangle. Let's create some atmosphere by adding scenery and sound effects. To change the scenery, we add a "backdrop" image.

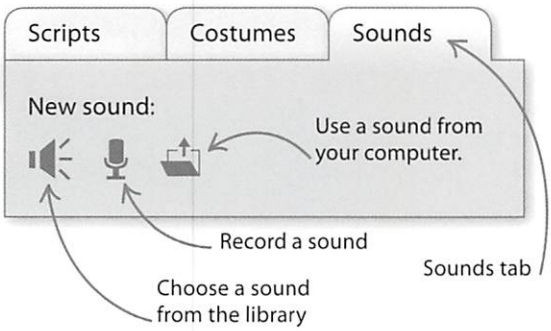
9 To the left of the sprites list is a button to add a picture from the backdrop library. Click it and look for "underwater2". Select the image and click "OK". The backdrop will now fill the stage.



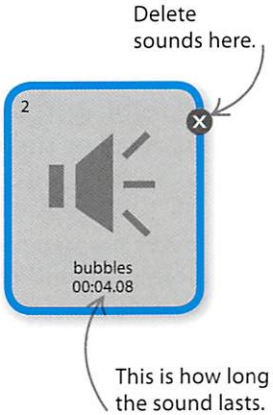
Sound effects

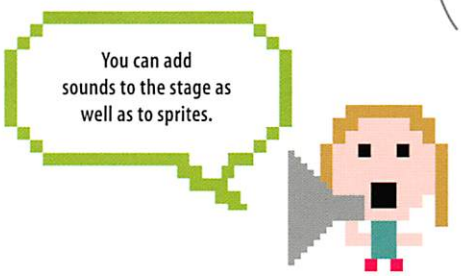
Now we'll add a bubbling sound to the cat sprite to make it sound like we're underwater.

10 Highlight the cat in the sprites list and then click the Sounds tab above the blocks palette. Click the speaker icon to choose a sound from the library.

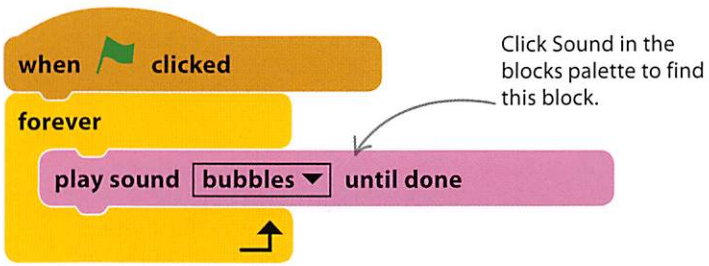


11 Look for "bubbles" in the library. You can preview sounds by clicking the play symbol. To load a sound into the game, click the speaker icon and then "OK". Now you'll see bubbles in your list of sounds.





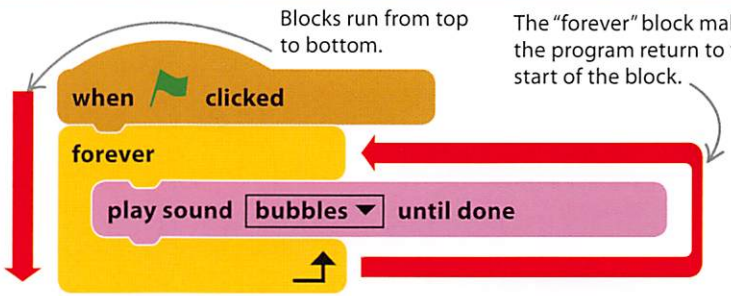
12 Click the Scripts tab and add the following script to the cat sprite, but leave the old script in place because you need both. The new script repeats the bubbles sound. The "play sound ... until done" block waits for the sound to finish before letting it start again. Run the game to hear the sound effect.



EXPERT TIPS

Loops

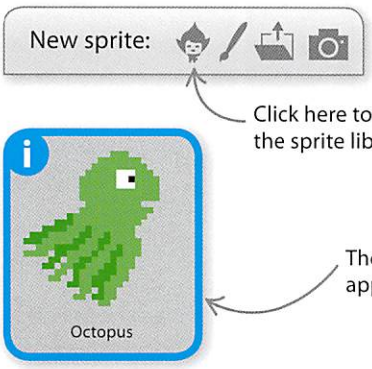
A loop is a section of code that repeats over and over again. The "forever" block creates a loop that carries on forever, but other types of loop can repeat an action a fixed number of times. Loops are very common in almost all computer programming languages.



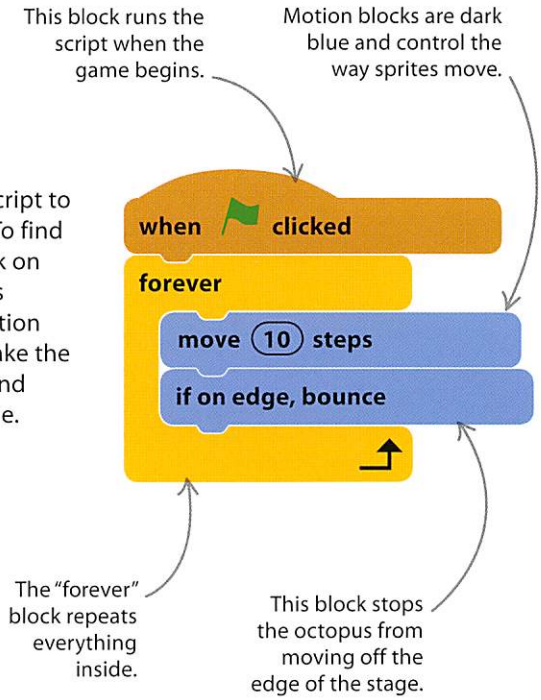
Add an enemy

The game needs an enemy to make things more interesting. Let's add an octopus with a deadly sting. The octopus will patrol the stage, moving left and right, and the player will have to keep out of its way or the game is over.

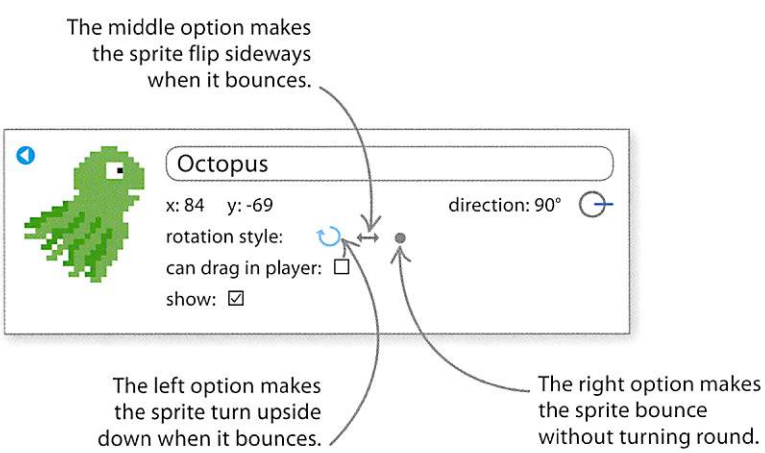
13 To add a second sprite to the project, click the icon shown below to open up the sprite library. Choose the octopus and click "OK".



14 Add the following script to the octopus sprite. To find the blue blocks, click on Motion in the blocks palette. The two Motion blocks used here make the octopus move left and right across the stage.



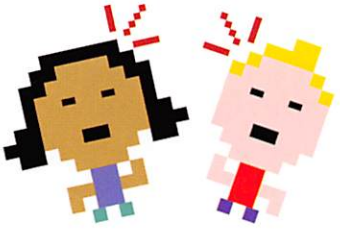
15 Now run the script. The octopus will patrol left and right, but you'll notice it's upside down half the time. We can fix this by changing the way the sprite turns around when it changes direction. Highlight the octopus and click the blue "i". In the pop-up box, there are three options after "rotation style".



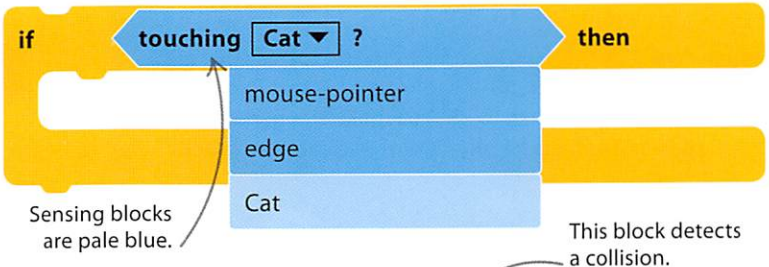
16 Choose the middle option and run the project. The octopus should now stay right side up and facing forward all the time. You can adjust its starting position on the screen by dragging it with the mouse.

Collisions

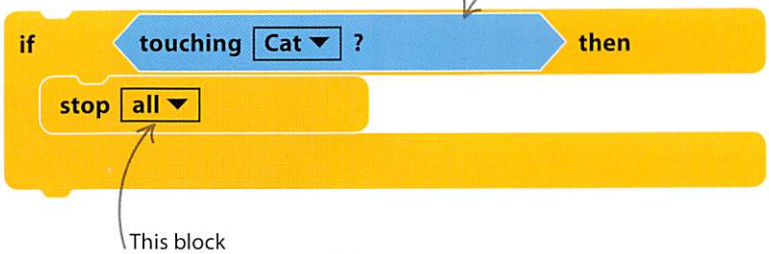
So far the octopus and cat move through each other without anything happening. We need to add a script to make them stop moving when they collide. Collision detection is very important in computer games.



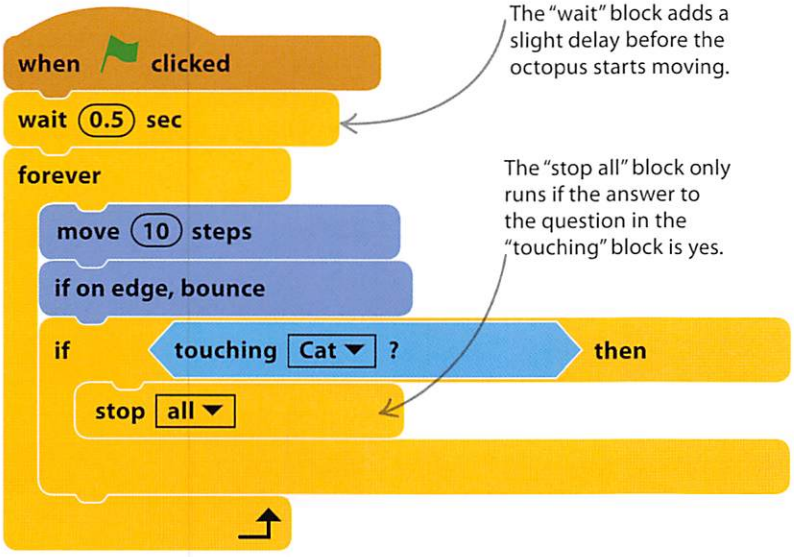
17 Highlight the octopus and drag a yellow "if then" block to an empty part of the scripts area. Now add a pale blue "touching" block to the top of the "if then" block. Click the drop-down menu and choose "Cat". This script will help the octopus detect the cat.



18 Choose Control in the blocks palette again, and add a "stop all" block to the middle of the "if then" block. This will stop all action if the octopus is touching the cat, ending the game.



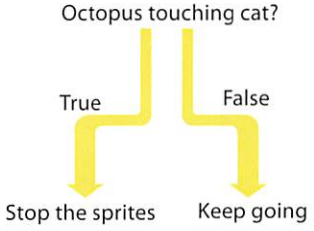
19 Now add the "if then" blocks you've built to the octopus's main script, placing it carefully after the blue Motion blocks. Also, add a "wait 0.5 sec" before the loop. Run the project and see what happens.



EXPERT TIPS

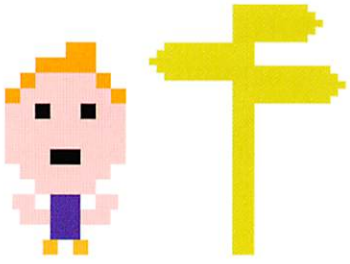
"if then"

You make decisions every day. If it's raining, you might use an umbrella. If it isn't, you don't. Computer programs do the same thing by using what programmers call conditional statements, such as "if then". When Scratch reaches an "if then" block, it runs the blocks inside only if the statement is true.



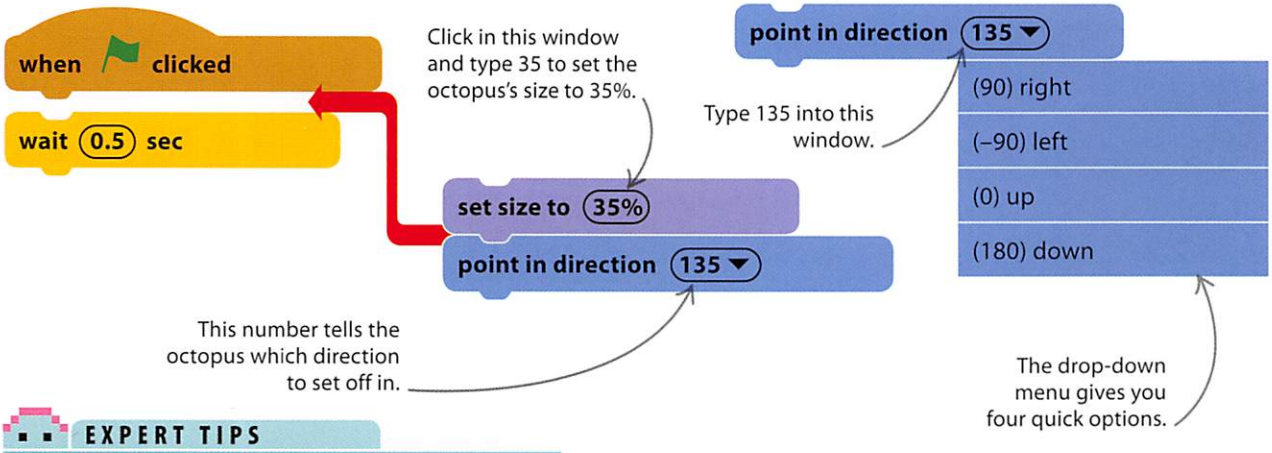
More enemies

Let's add more enemies to the game, but to make things more challenging, we'll make them move in different directions. We can tell each sprite exactly which way to go by using a block that works like a compass.



20 Add a purple "set size" block to the top of the octopus's script, after the "when clicked" block. Set the octopus's size to 35% to make the game a bit easier. Then add a blue "point in direction" block.

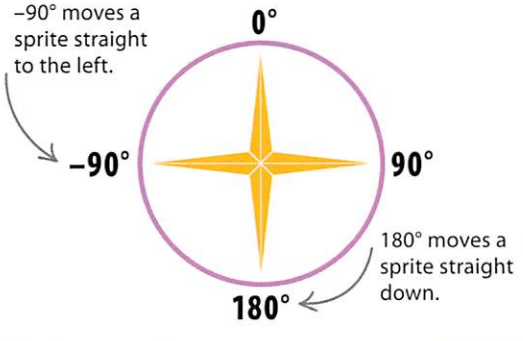
21 To change the octopus's direction, click on the window in the "point in direction" block and type 135 in place of 90. This will make the octopus move diagonally.



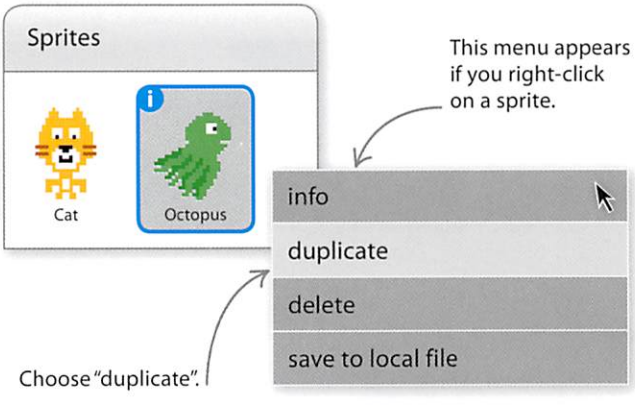
EXPERT TIPS

Directions

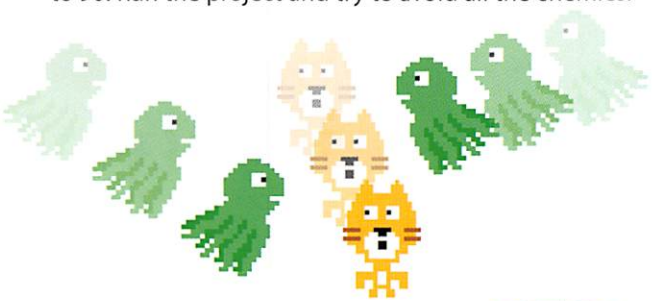
Scratch uses degrees to set direction. You can choose any number from –179° to 180°. Negative numbers point sprites left; positive numbers point them right. Use 0° to go up and 180° to go straight down.



22 Now we can duplicate our octopus to create more enemies. Right-click on the octopus in the sprites list (or control-click if you have a Mac) and choose "duplicate". Copies of the Octopus sprite will appear in the sprites list, named Octopus2 and Octopus3. Each will have a copy of the first octopus's script.

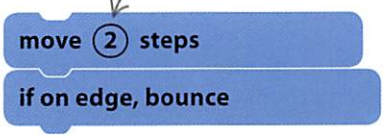


23 To make the octopuses move in different directions, change the number in the "point in direction" block for each new octopus. Leave the first Octopus sprite's direction as 135, but set Octopus2 to 0 and Octopus3 to 90. Run the project and try to avoid all the enemies.

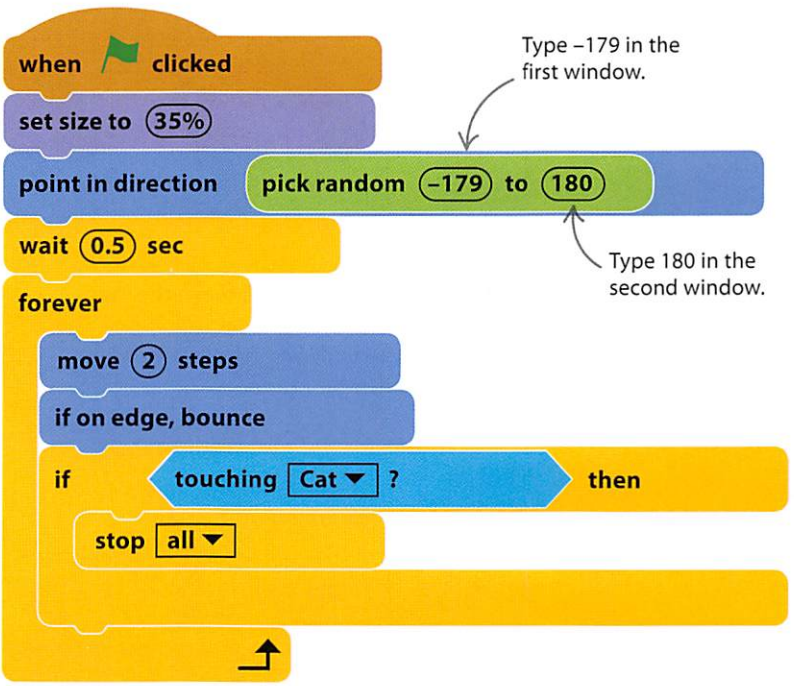


24 If it's too hard to stay alive, make the octopuses slower by lowering the number of steps in their "move" blocks to two. Remember to change the script for all three octopus sprites.

Changing this number adjusts the octopus's speed.



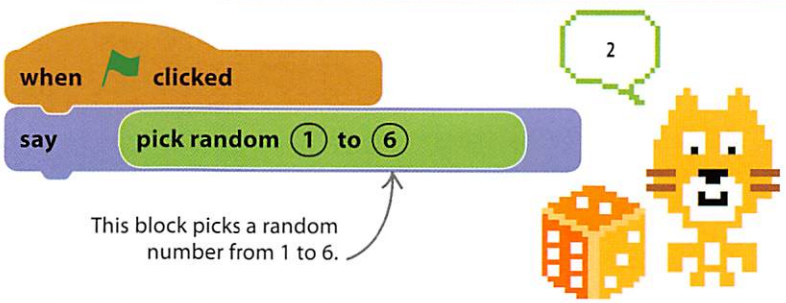
25 For more variety, let's make one of the octopuses set off in a random direction. To do this, we use a green "pick random" block. This is Scratch's way of rolling a dice to generate a random number. Choose Operators in the blocks palette to find the block and add it to the first octopus's script. Run the project a few times to see the octopus choose different starting directions.



EXPERT TIPS

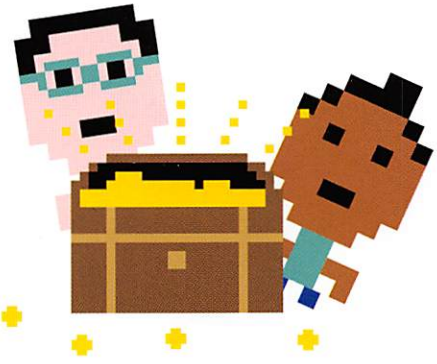
Random numbers

Why do so many games use dice? Dice create surprises in a game because they make different things happen to each player. A random number is one you can't predict in advance, just like the roll of a dice. You can get the cat to say a random dice roll using this simple code.

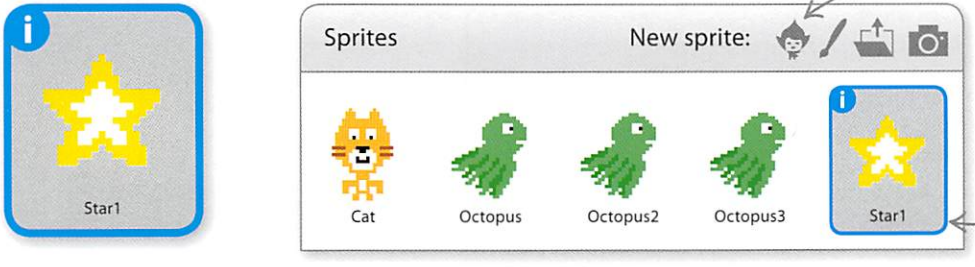


Collecting stars

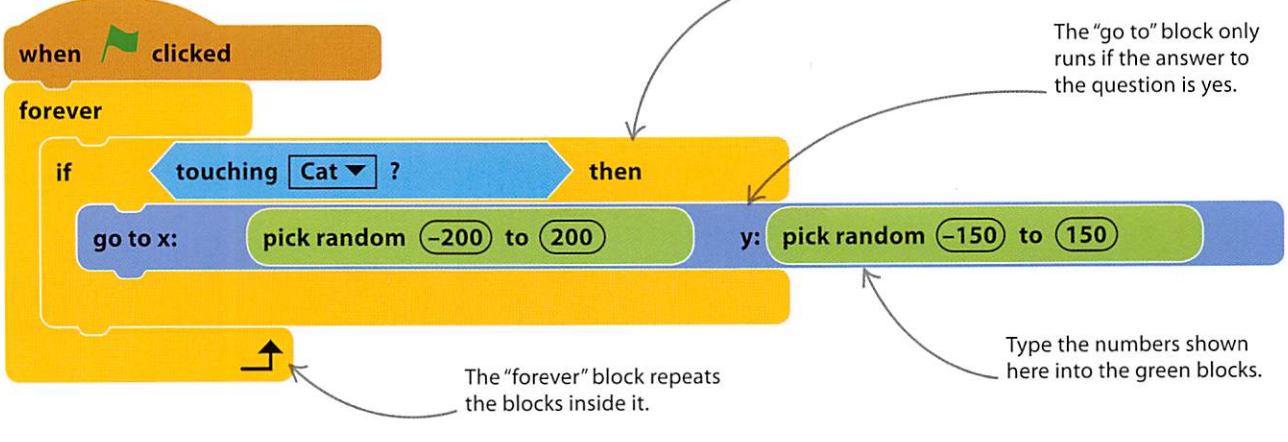
In many games, the player has to collect valuable items to win points or to stay alive. In Star Hunter, we use gold stars as underwater treasure that the player has to collect. We'll use random numbers again to make each star appear in a new place.



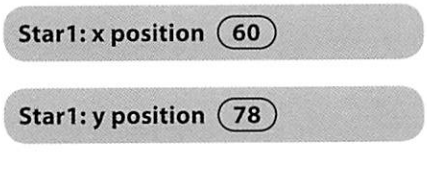
26 Click the "choose new sprite" symbol in the sprites list and choose the "Star1" sprite from the library.



27 Add the following script to Star1. This script will make the star move to a random new location whenever the cat touches it. The green blocks create random numbers called coordinates, which Scratch uses to pinpoint locations on the stage.



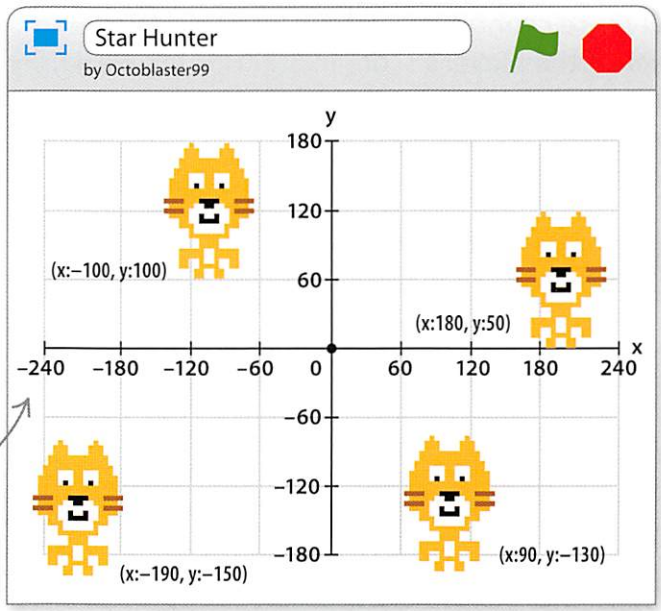
28 To see the star's coordinates change when it moves, choose Motion in the blocks palette and put ticks by "x position" and "y position". Now run the game: you'll see the star's x and y coordinates update each time the cat makes it move. Untick both boxes before you carry on.



EXPERT TIPS

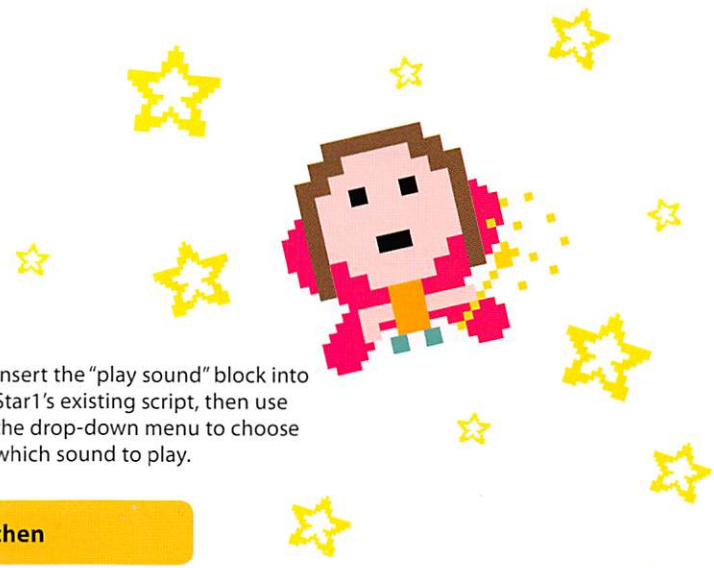
Using coordinates

To pinpoint a location on the stage, Scratch uses numbers called coordinates. These work just like graph coordinates, with x numbers for horizontal positions and y numbers for vertical. To find the coordinates for a spot on the stage, just count the steps across and up from the center of the stage. Positive coordinates are up or right, negative coordinates are down or left. Every spot on the stage has a unique pair of coordinates that can be used to send a sprite to that position.

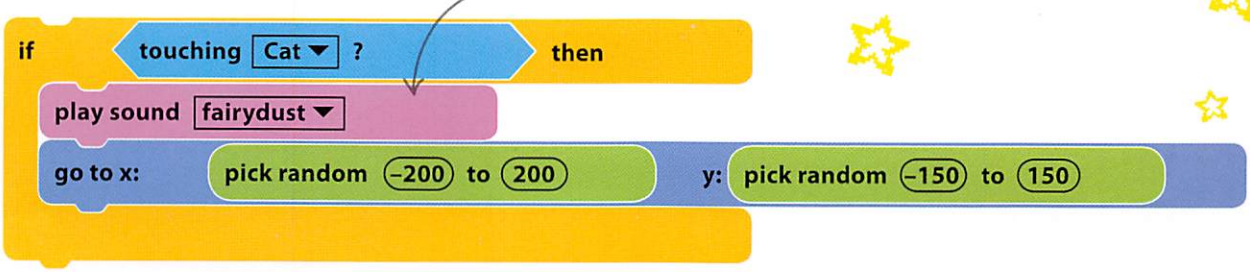


The x axis is longer than the y axis and extends from −240 to 240.

29 You can add a sound effect that plays when the cat touches a star. First make sure that the star is selected in the sprites list, then click the Sounds tab above the blocks palette. Click the speaker symbol to open the sound library. Choose "fairydust" and click "OK". Now add the pink "play sound" block to the star's script and choose "fairydust" in the drop-down list.

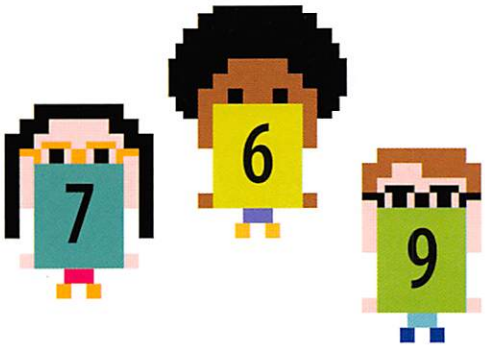


Insert the "play sound" block into Star1's existing script, then use the drop-down menu to choose which sound to play.

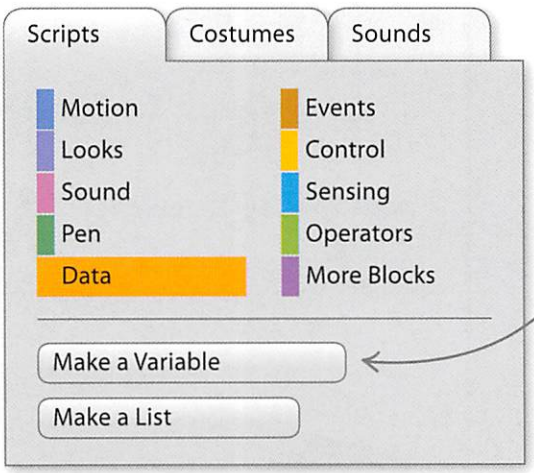


Keeping score

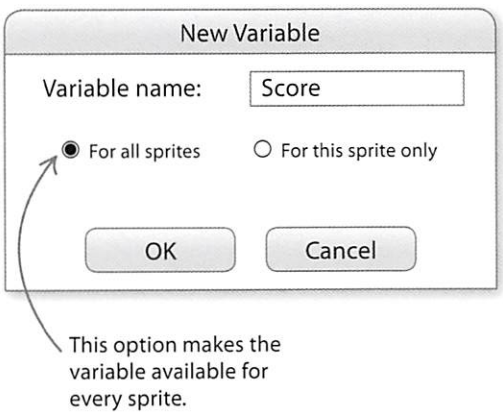
Computer games often need to keep track of vital statistics such as the player's score or health. We call these changing numbers "variables". To keep track of the player's score in Star Hunter, we'll create a variable that counts the number of stars the player has collected.



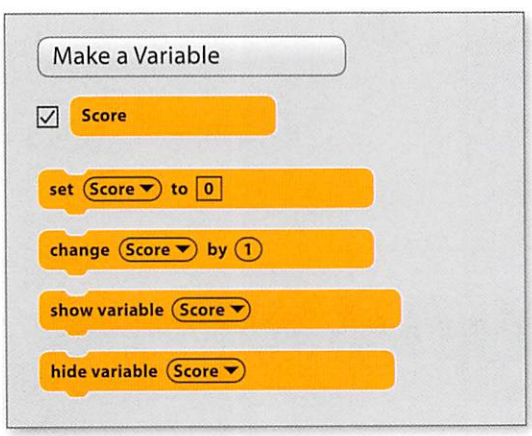
30 With any sprite selected, choose Data in the blocks palette. Click on the button "Make a Variable".



31 A pop-up box appears asking you to give your variable a name. Type "Score" in the box. Make sure the option "For all sprites" is selected and hit "OK".

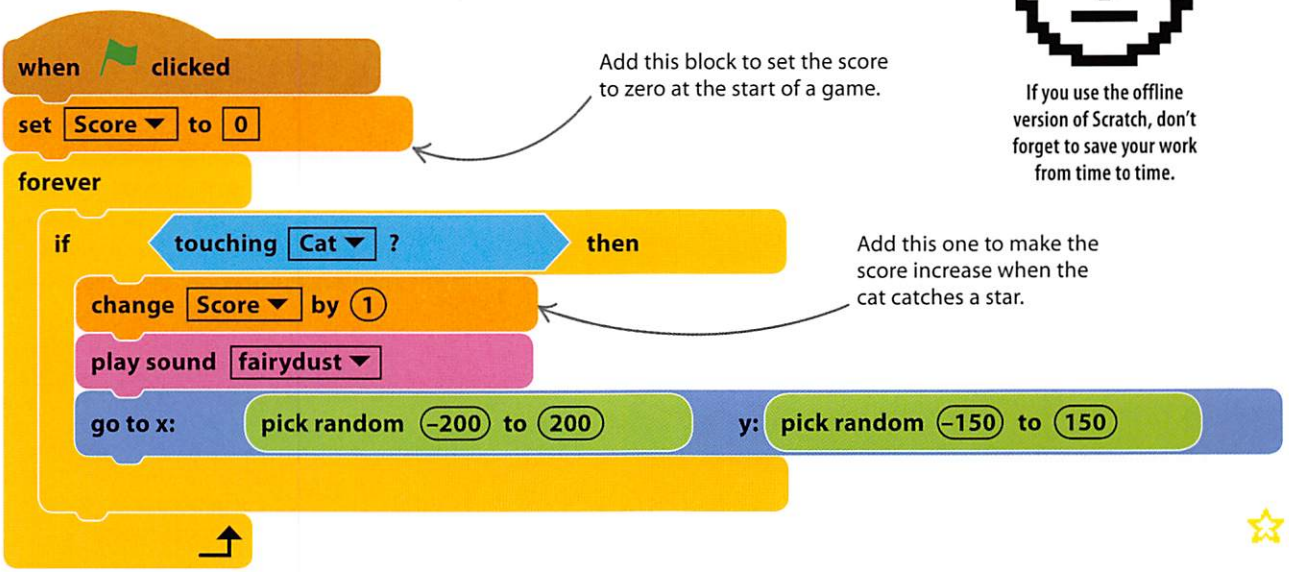


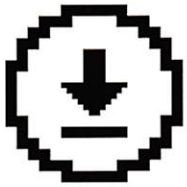
32 You'll see a new set of blocks appear, including one for the score. Make sure the box next to it is checked to make the score appear on the stage.



33 The score counter will appear in the top left of the stage but you can drag it anywhere you like.

34 We want the score to start at zero and increase by one each time the cat touches a star. Select the star sprite and add the two orange Data blocks below to its script.





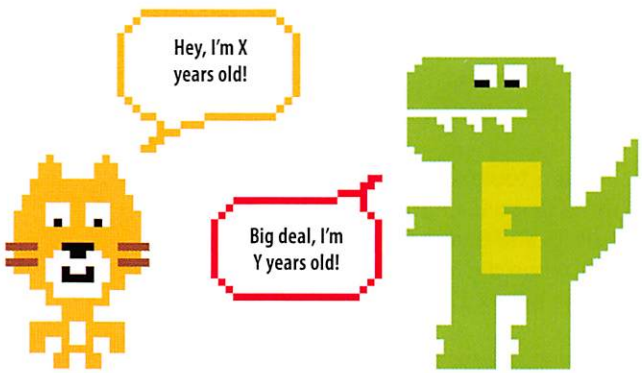
If you use the offline version of Scratch, don't forget to save your work from time to time.

35 Now click the green flag to run the script and see what happens when the cat collects each star. See if you can collect 20 stars without bumping into an octopus.

EXPERT TIPS

Variables

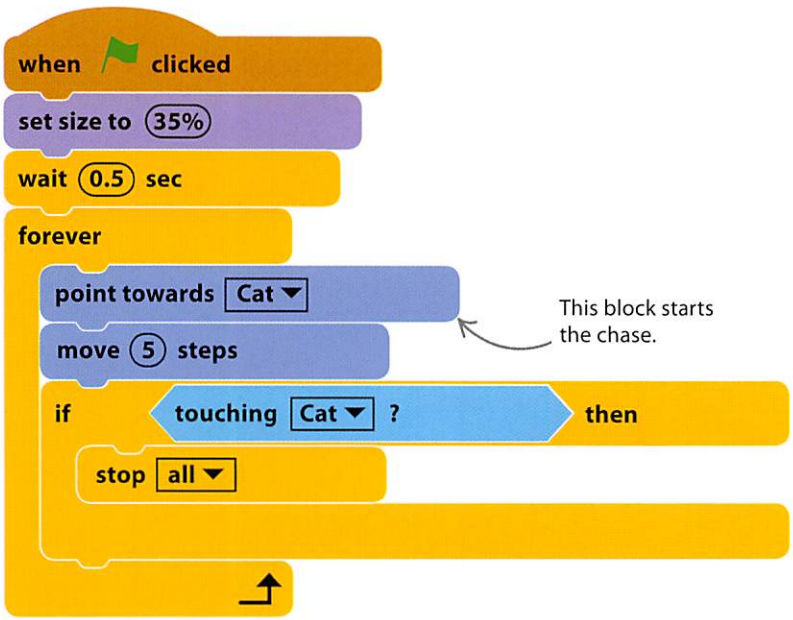
A variable works like a box that you can store information in, such as a number than can change. In math, we use letters for variables, such as x and y. In computer programming, we give variables names such as "Score" and use them for storing not just numbers but any kind of information. Try to choose a name that tells you what the variable is for, such as "Speed" or "Score". Most computer languages won't let you put spaces in the names of variables, so a good tip is to combine words. Instead of using "dog speed", for instance, type "DogSpeed".



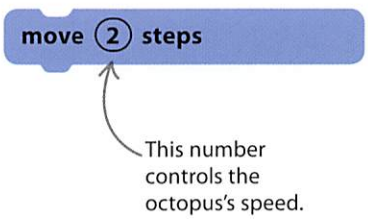
Better enemies

Now we have a working game, we can test it and experiment with changes that make it easier, harder, or—most important—more fun. One way to make the game more interesting is to make the three octopuses do different things.

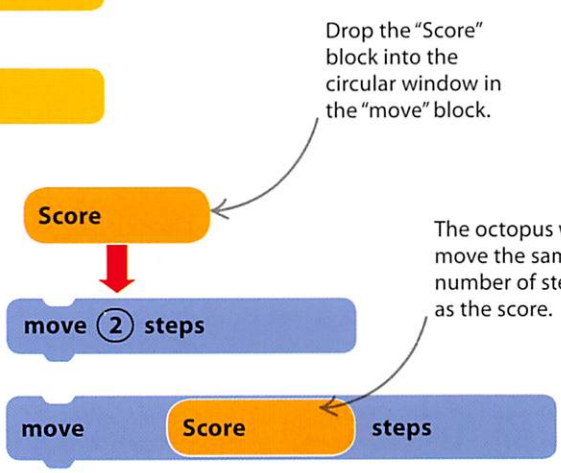
36 Right-click the script for Octopus2 and select "delete" to remove it. Replace it with the following script. This will make the octopus chase the cat.



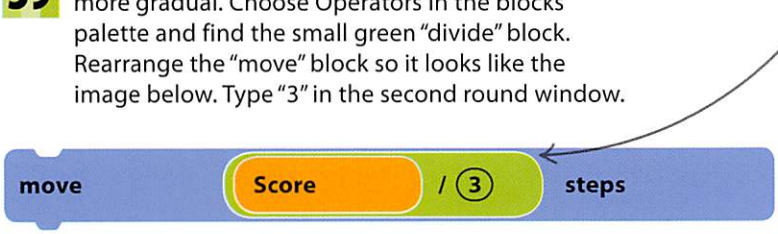
37 Run the project and see how the game plays. You'll probably find it hard to escape the octopus because it moves quickly. To slow it down, change the number of steps to two.



38 You can make the game get harder as you play. Select the original octopus sprite and click Data in the blocks palette. Drag the "Score" block into the octopus's "move" block. Now try the game. The more points you get, the faster the octopus swims.



39 If it gets too hard too quickly, we can make things more gradual. Choose Operators in the blocks palette and find the small green "divide" block. Rearrange the "move" block so it looks like the image below. Type "3" in the second round window.



40 Now we'll make Octopus3 patrol in a regular pattern. To do this, we'll use a new Motion block that makes it glide smoothly from point to point, rather than moving in steps. Replace the script for Octopus3 with the following two scripts. These run at the same time, one checking for collisions and the other moving the octopus around its patrol route.

The two scripts are separate in the scripts area.

Scripts | Costumes | Sounds

Motion, Looks, Sound, Pen, Data, Events, Control, Sensing, Operators, More Blocks

move (10) steps
turn (15) degrees
turn (15) degrees
point in direction (90)
point towards
go to x: (0) y: (0)
go to (mouse-pointer)
glide (1) secs to x: (0) y: (0)

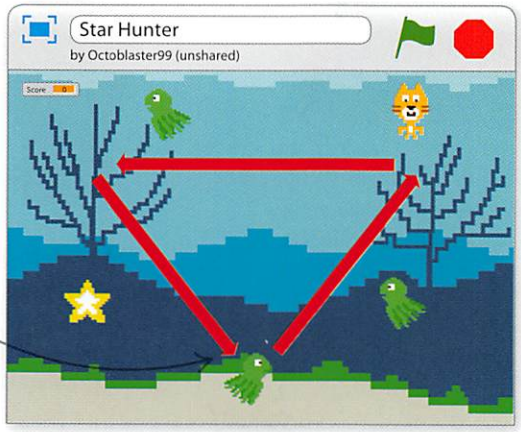
when clicked
forever
glide (3) secs to x: (0) y: (-150)
glide (3) secs to x: (200) y: (100)
glide (3) secs to x: (-200) y: (100)

x: -126
y: 96

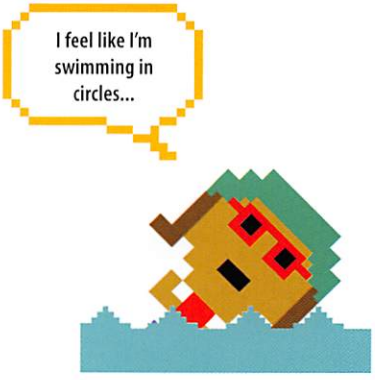
Type these numbers into the "glide" blocks.

when clicked
set size to (35%)
wait (0.5) sec
forever
if touching (Cat) ? then
stop (all)

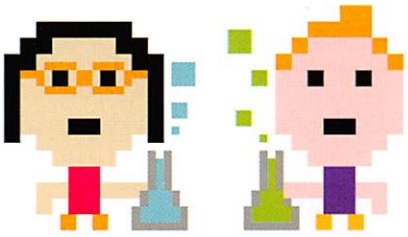
41 Now run the project and watch Octopus3. It should swim in a repeating triangle pattern.



To change the shape of the triangle, try different numbers in the "glide" blocks.



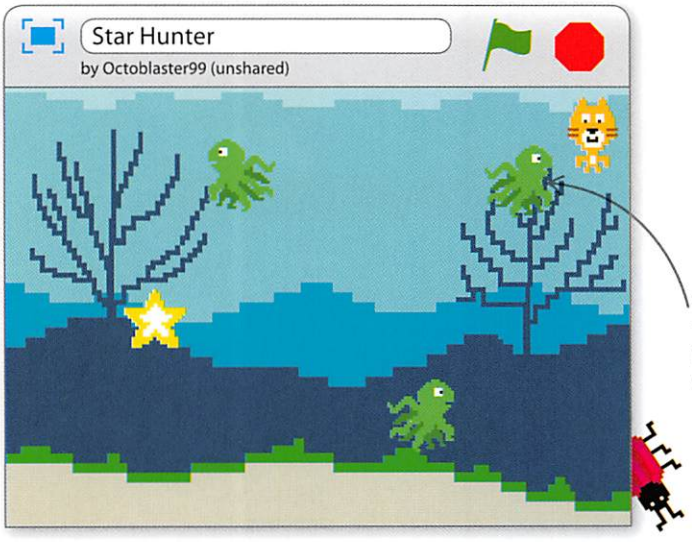
Hacks and tweaks



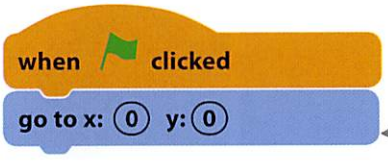
You've built a fun game, but that's just the beginning. Scratch makes it easy to change and adapt games as much as you want. You might find bugs that need fixing, or you might want to make the game harder or easier. Here are some suggestions to get you started.

▽ Debug Octopus2

If Octopus2 ends up in the top-right corner at the end of a game, it can trap the player in the next game and end it too quickly. This is a bug. To fix it, you could drag the octopus away from the corner before starting, but it's better to use a script that moves it automatically. Insert a "go to" block at the start of the script for Octopus2 to send it to the center of the stage.



Octopus2 can trap the player in the top-right corner.

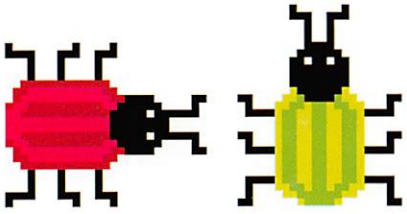


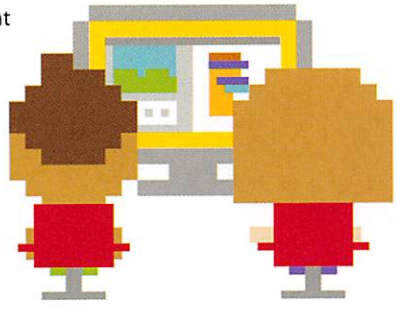
Add this block to make Octopus2 start in the center of the stage.

LINGO

Bugs

A bug is an error in a program. The first computers made mistakes when real insects, or bugs, got in their circuits. The name stuck. Today, programmers often spend as much time finding and fixing bugs as they do writing code in the first place.



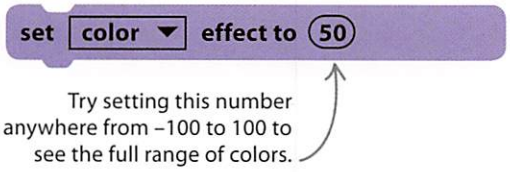


△ Fine-tuning

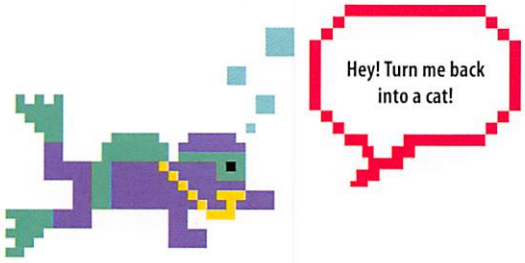
The best games have been carefully tested to make sure they play well. Test every change you make and get friends to play your games to see how well they work.

▽ Different colors

Make your octopuses different colors by using the "set color" block from the Looks section. Place it under the "set size" block at the start of the script.



Try setting this number anywhere from –100 to 100 to see the full range of colors.



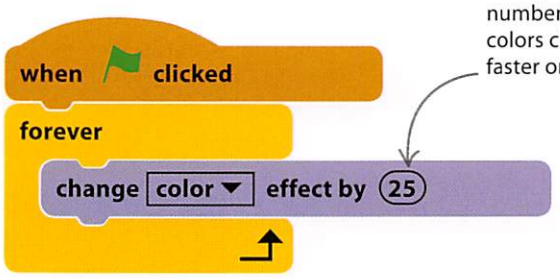
△ Scuba diver

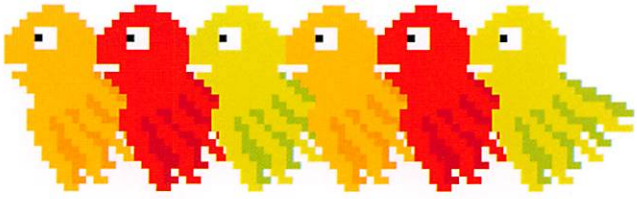
To make the underwater theme more convincing, replace the cat with a diver. Click on the cat in the sprites list, then open the Costumes tab and click on the sprite symbol to open the library. Load the costume called "diver1".

▽ Flashing colors

You can make an octopus change color continually to create a flashing effect. Add the script below to any octopus. Try experimenting with different numbers in the "change color" block.

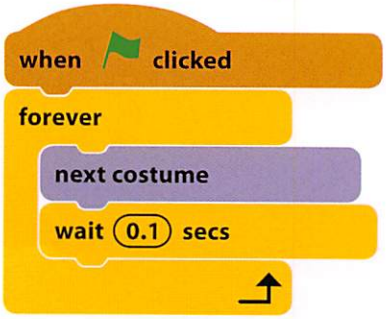
Change this number to make colors change faster or slower.

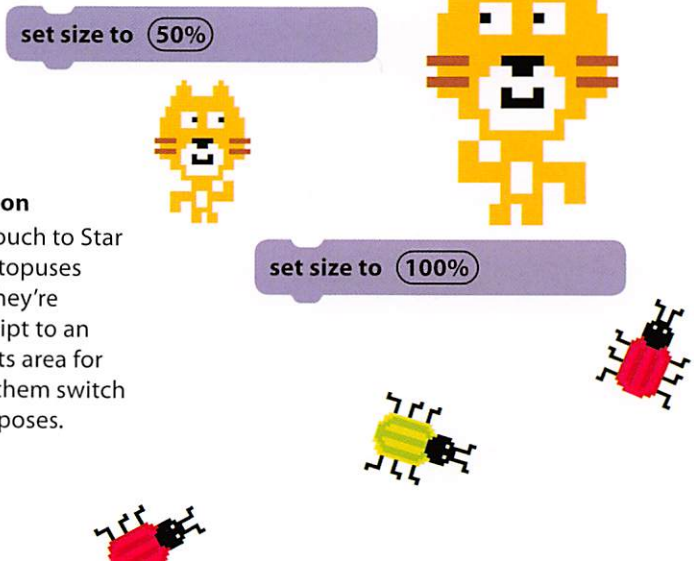




▽ Play with size

You can change how easy the game is by adjusting the size of the sprites. Change the number in the octopuses' blue "move" blocks to alter their speed. Change the purple "set size" blocks to make sprites larger or smaller. Fine-tune the numbers until the game is just hard enough to be fun.



◁ Swimming animation

To add a professional touch to Star Hunter, animate the octopuses so that they look as if they're swimming. Add this script to an empty part of the scripts area for each octopus to make them switch between two different poses.