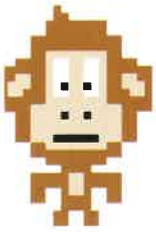
How to build Jumpy Monkey

In the real world there are laws you just can't break. For example, the law of gravity means that something that goes up must always come down again. Jumpy Monkey shows you how to add gravity to your game worlds.

AIM OF THE GAME

The monkey is on a mission to collect bananas. Choose which direction he leaps in and how fast he goes. You need to send him over the palm tree to grab the bananas using the fewest possible jumps.

◁ **Launcher**
Point this arrow in the direction you want to launch the monkey by using the left and right arrow keys.



◁ **Monkey**
Select the monkey's launch speed with the up and down arrow keys, then press the space key to launch him.

◁ **Bananas**
If the monkey touches any of the bananas he will eat them. Keep going until he eats all the bananas.

The instructions appear on the game at the start.

The monkey is launched from the arrow when you press the space key.

This number shows you how fast the monkey will fly once he is launched.

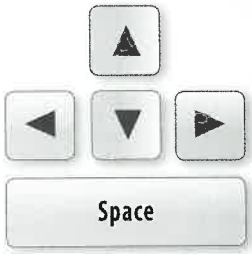
The monkey flies through the air like a cannon ball.

There are three bunches of bananas to collect each time you play the game.

Avoid the tree – the monkey can't fly through it.

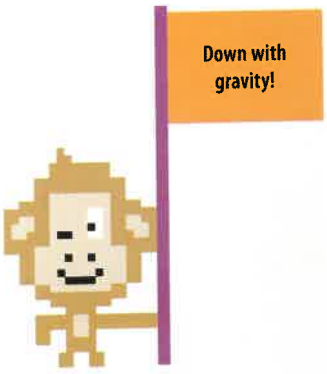
GAME CONTROLS

Players use the arrow keys and space key on the keyboard as game controls.



◁ **Flying monkey**

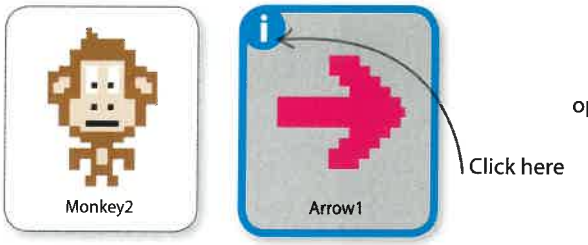
Try to collect all the bananas using as few launches as possible. The game will record how many launches you use.

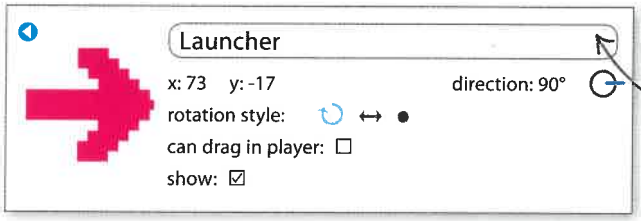


Launching the monkey

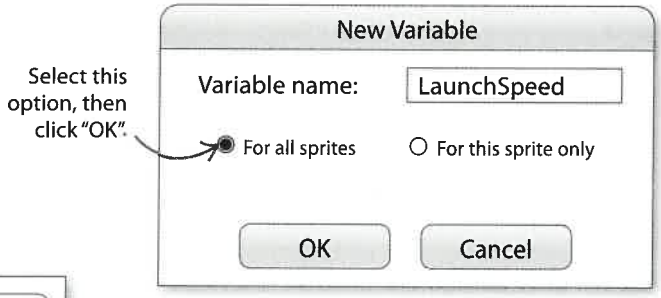
This game uses a big arrow to help the player choose the monkey's precise launch direction. We'll ignore gravity to start off with, but you'll need to add it later to get the monkey past the tree.

1 Start a new project and call it "Jumpy Monkey". Delete the cat sprite and load two sprites from the library – "Monkey2" and "Arrow1". Select the arrow sprite and rename it "Launcher" by clicking on the "i" and typing the new name into the box.

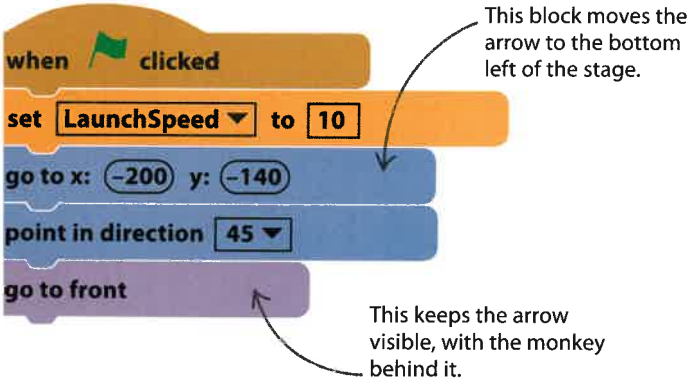


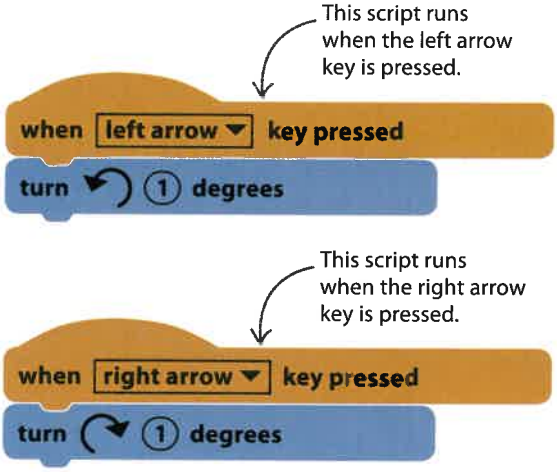


2 Go to Data, select "make a variable", and add a variable called "LaunchSpeed". The new variable will automatically show up on the stage.

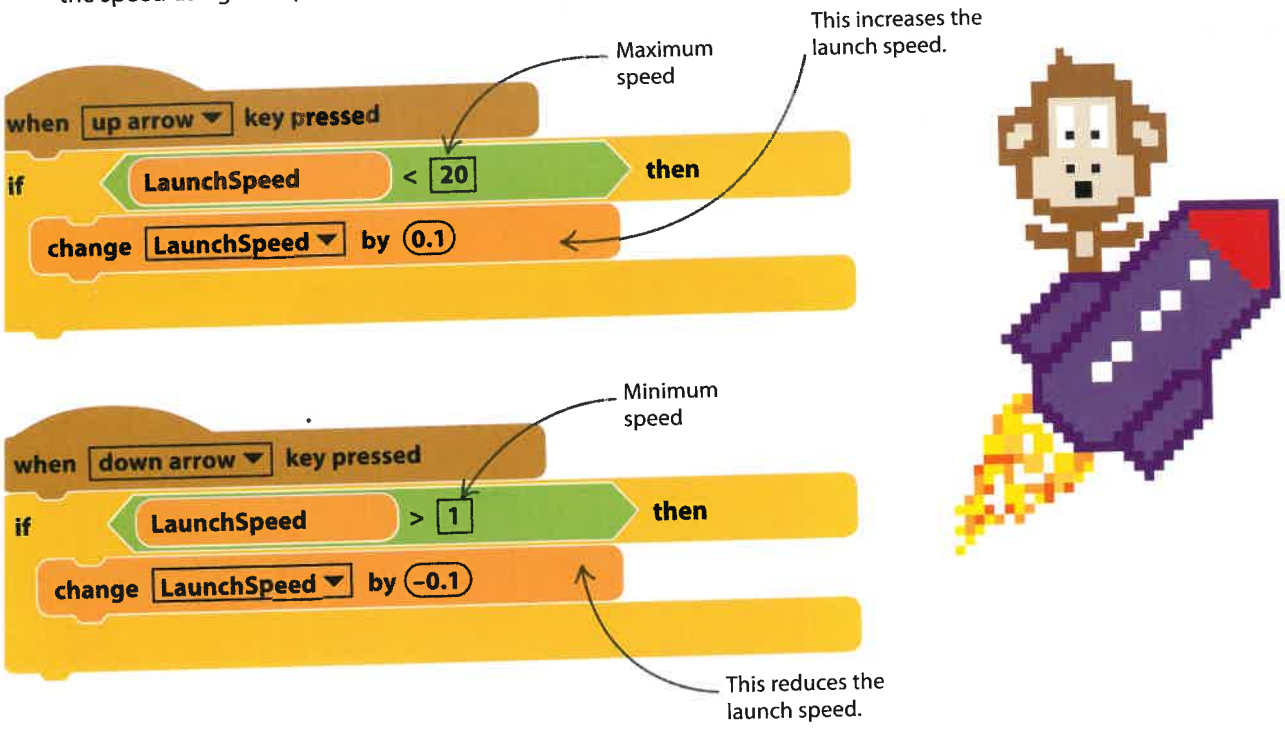


3 Select the Launcher sprite, then add these three scripts to set up the Launcher and allow the player to control its angle using the left and right arrow keys on the keyboard. The direction of the arrow is the direction that he monkey will launch in. Run the scripts and try turning the arrow.





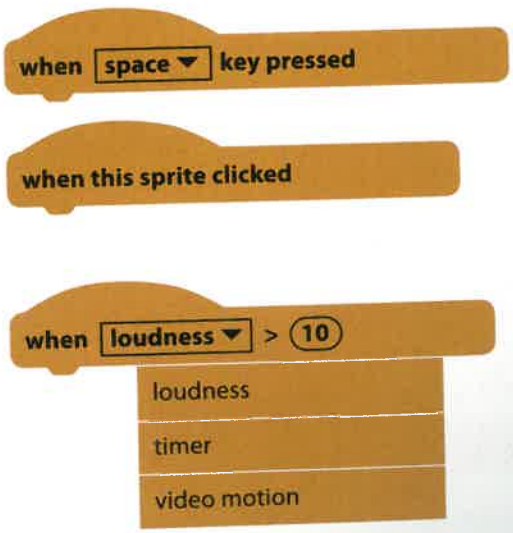
4 Now that you can aim, you need controls to set the speed of the launch. Add these scripts to change the speed using the up and down arrow keys.



LINGO

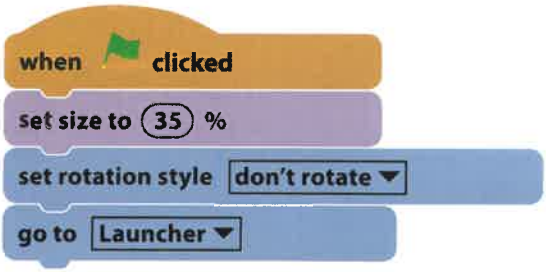
Events

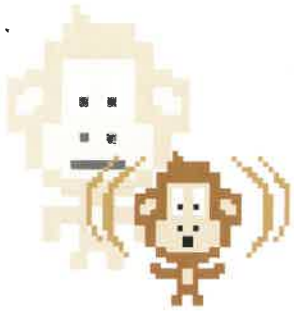
The key presses and mouse clicks that a computer detects are known as events. The brown Events blocks in Scratch trigger a script whenever a particular event occurs. We've seen them used with messages in Cheese Chase, but Scratch also lets you trigger scripts using keys, mouse clicks, sound levels, and even movement detected by a webcam. Don't be afraid to experiment.



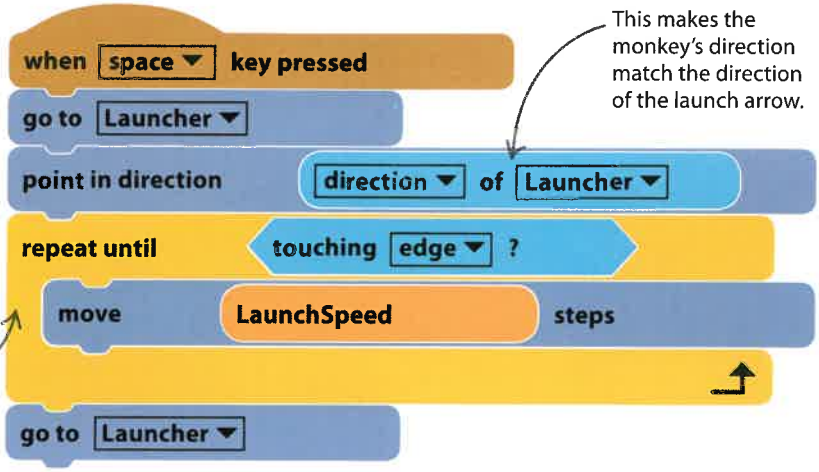
▷ **Setting things off**
Events blocks such as these are used to trigger a script whenever the event they describe occurs.

5 Now select the Monkey sprite. Add this script to shrink him down to the right size and move him behind the Launcher.





6 To launch the monkey when the space bar is pressed, add this new script to the Monkey sprite. "Repeat until" is a new type of loop block that keeps repeating the block inside until the condition becomes true – in this case, the monkey keeps moving until it touches the edge of the stage.



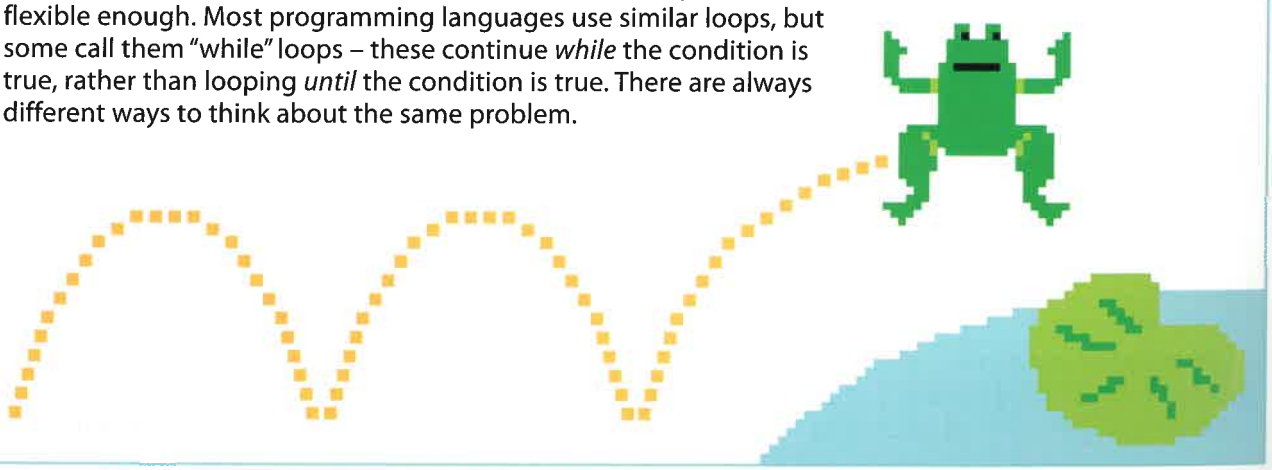
This makes the monkey's direction match the direction of the launch arrow.

The "repeat until" block keeps the monkey moving to the edge of the stage.

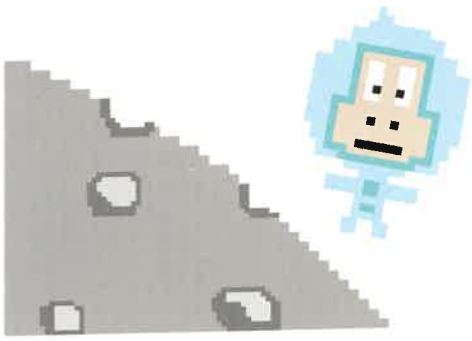
EXPERT TIPS

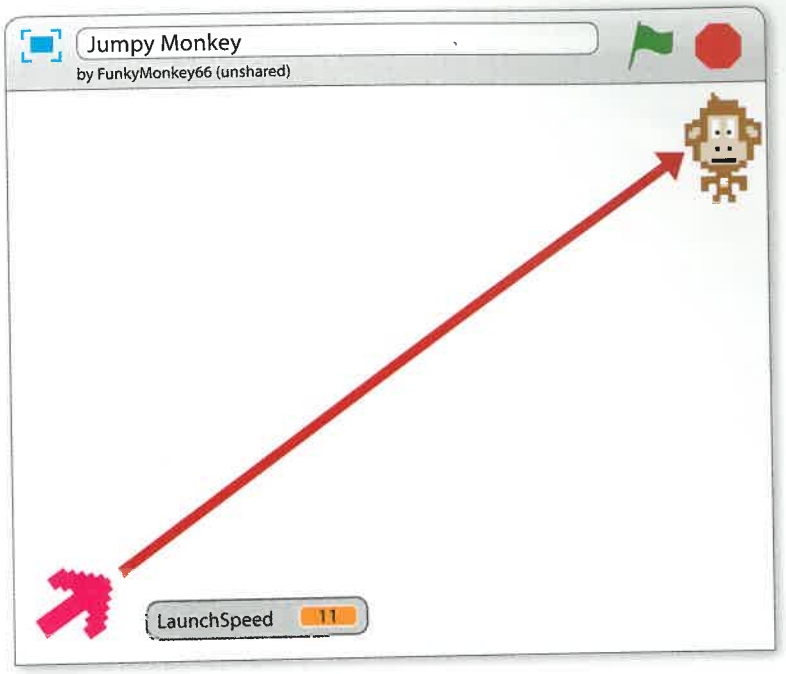
"repeat until"

Do you want to keep repeating an action only until something happens and then get on with the rest of the script? The "repeat until" block can help your code when "forever" and "repeat" loops aren't flexible enough. Most programming languages use similar loops, but some call them "while" loops – these continue *while* the condition is true, rather than looping *until* the condition is true. There are always different ways to think about the same problem.



7 Try setting the Launcher angle and speed using the arrow keys, and pressing the space bar to fire the monkey. He goes in a completely straight line until he hits the edge of the stage. Real things don't do this – they fall back towards the ground as they move. We'll add gravity to the game later to make the monkey behave realistically.

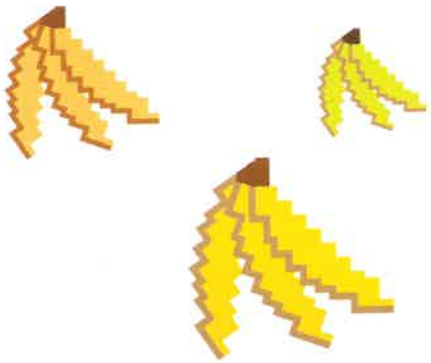


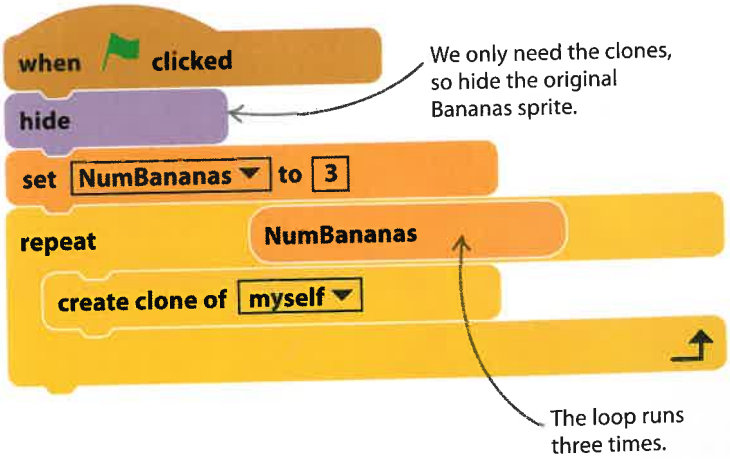


Bananas and palm trees

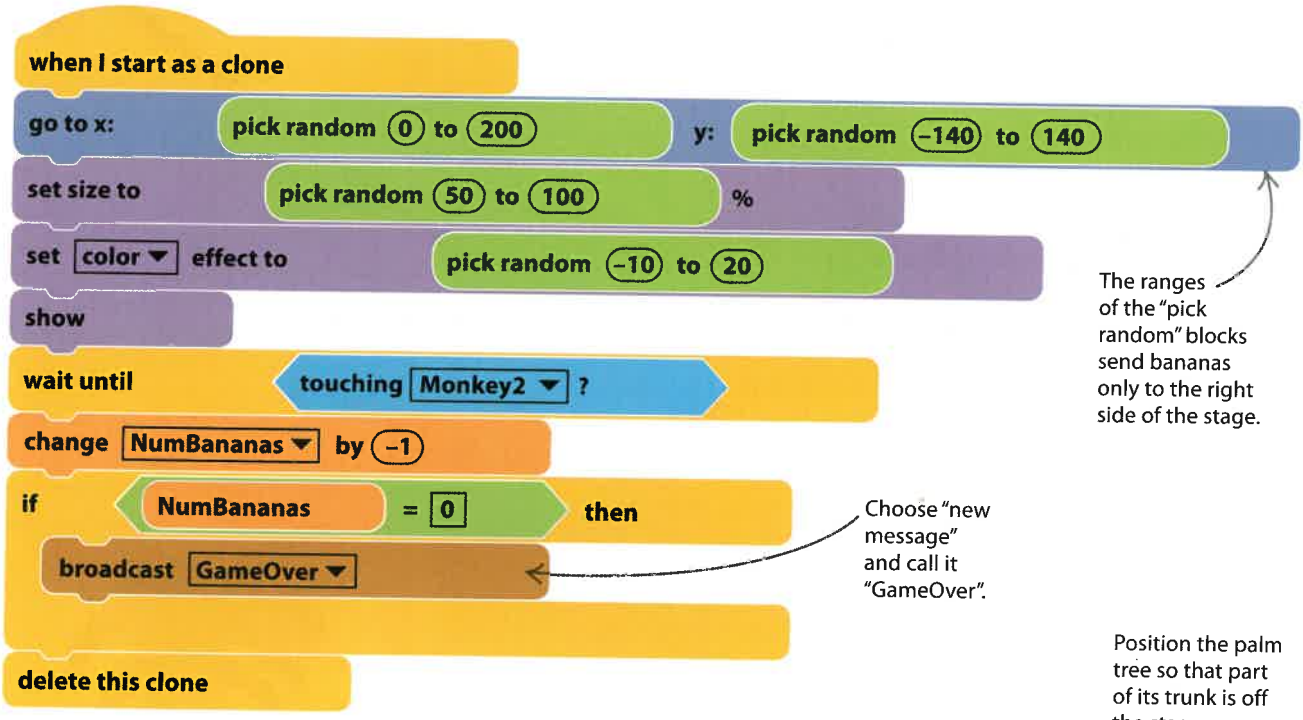
The point of this game is for the monkey to collect bananas. By using clones, you can add just one Bananas sprite but give the monkey plenty of fruit to aim for.

8 Add the Bananas sprite to the project. Make a variable for all sprites called "NumBananas" to keep track of the number of bananas on the stage – start with three. Build the following script to clone the bananas, but don't run it yet as you still need to tell the clones what to do.





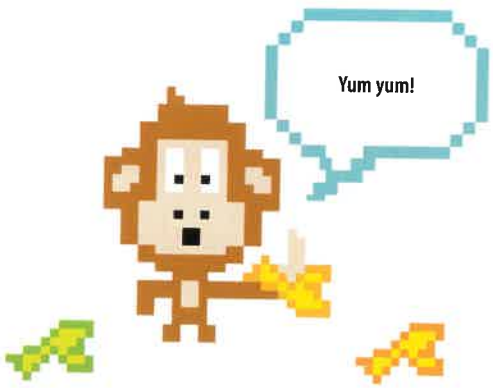
9 Add the next script to place each banana clone in a random spot on the right of the stage, change how it looks, and make sure it's not hidden. The clone will wait for the monkey to touch it and then disappear. If it's the last banana, it sends a "GameOver" message, which you need to create as a new message.



The ranges of the "pick random" blocks send bananas only to the right side of the stage.

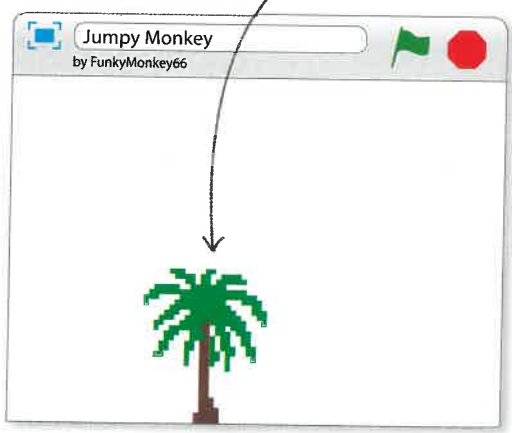
Choose "new message" and call it "GameOver".

10 Run the project. You should be able to get the monkey to collect all the bananas. There is no script run by the "GameOver" message yet.



11 The game is too easy – we need an obstacle. Add the Palmtree sprite to the project. Drag and drop the tree at the bottom of the stage.

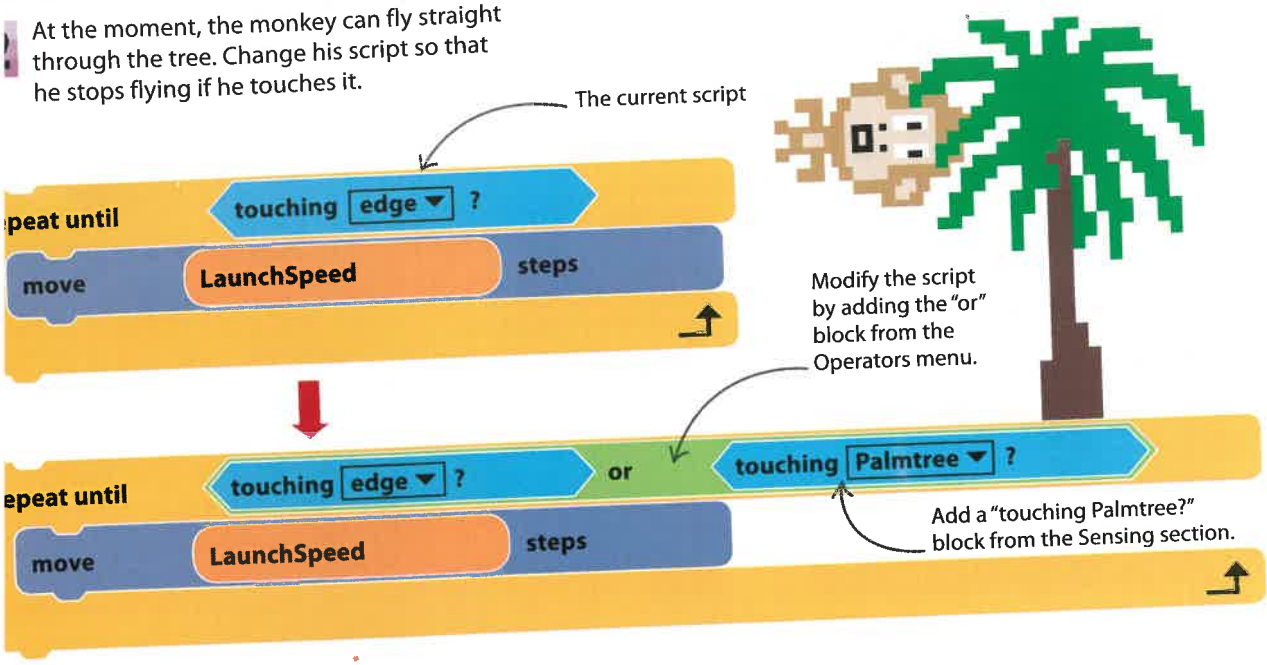
Position the palm tree so that part of its trunk is off the stage.



△ **Tree on stage**

Make sure your palm tree is slightly off-centre, towards the left of the stage, or the bananas will get stuck behind the tree and the game won't work.

2 At the moment, the monkey can fly straight through the tree. Change his script so that he stops flying if he touches it.

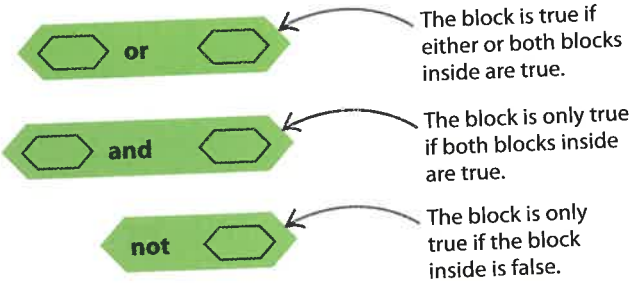


3 Run the project. The monkey should stop flying when he hits the tree, which makes any bananas to the right of the tree impossible to reach. Don't worry, gravity will come to the rescue soon.

EXPERT TIPS

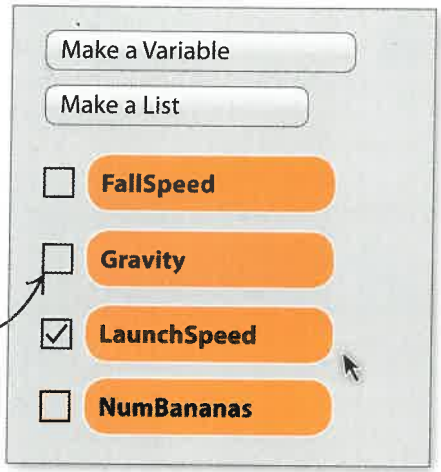
"or", "and", "not"

So far, most of the "if then" blocks in this book have tested only a single condition, such as "if touching cat" in the Star Hunter game. In this chapter, however, you need to test two conditions at once: "touching edge or touching Palmtree". Complex sets of condition like this occur a lot in coding, so you need a way to combine them. In Scratch, the green Operators blocks do the job. You'll see words like "or", "and", and "not" in almost every programming language, or special symbols that do the same job.



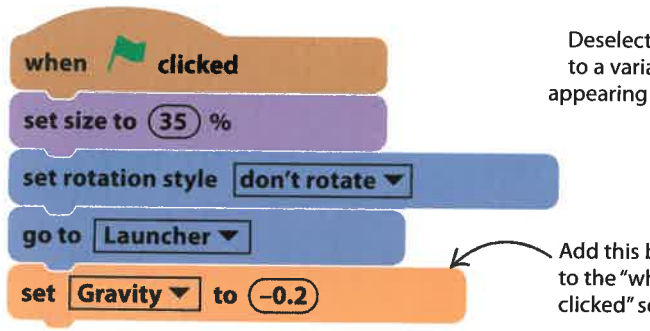
△ **Logic blocks**
Logical Operators blocks such as these three let you test for complex sets of conditions.

14 Make two more variables for all sprites: "FallSpeed" and "Gravity". Then add a "set Gravity" block to the monkey's "when clicked" script and amend his "when space key pressed" script as shown below. The new blocks use variables to simulate gravity. "FallSpeed" keeps track of how many steps the monkey needs to be moved down by gravity. The value of "Gravity" is how much "FallSpeed" increases by each time the monkey moves.

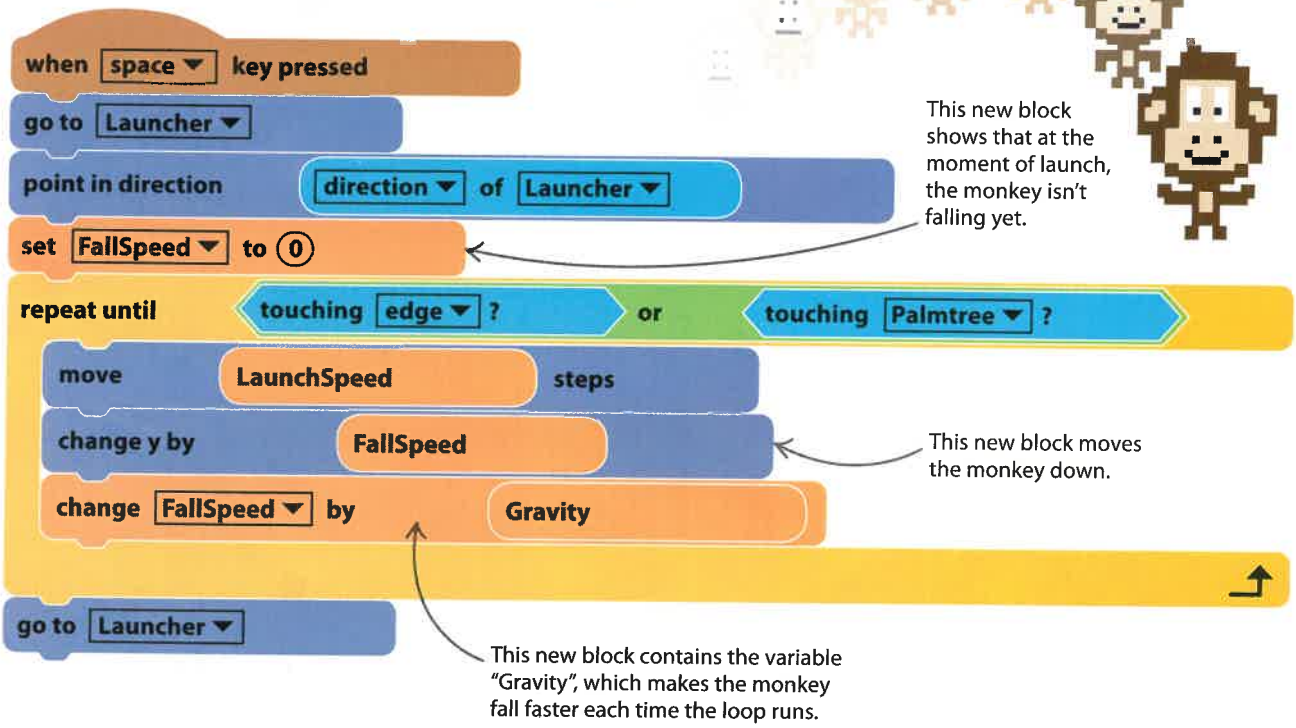


Deselect the box next to a variable to stop it appearing on the stage.

△ **Hiding variables**
If you don't want variables to appear on the stage, you need to untick the box next to them in the Data section. Do this for these two new variables.



Add this block to the "when clicked" script.



This new block shows that at the moment of launch, the monkey isn't falling yet.

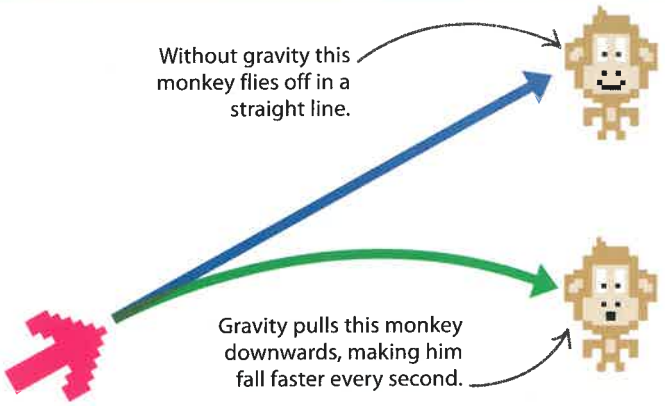
This new block moves the monkey down.

This new block contains the variable "Gravity", which makes the monkey fall faster each time the loop runs.

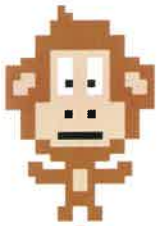
EXPERT TIPS

Real world gravity

In the real world, when you try to throw something in a straight line it curves slowly back towards the ground as gravity pulls it down. To make the game work in the same way, you move the monkey along the straight line, but also add a downward move after each shift along that line, to create the same effect as the constant downward tug of gravity. This allows the monkey's movement to seem natural, making the game more engaging.

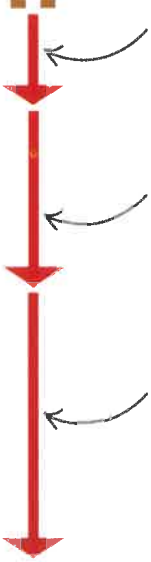


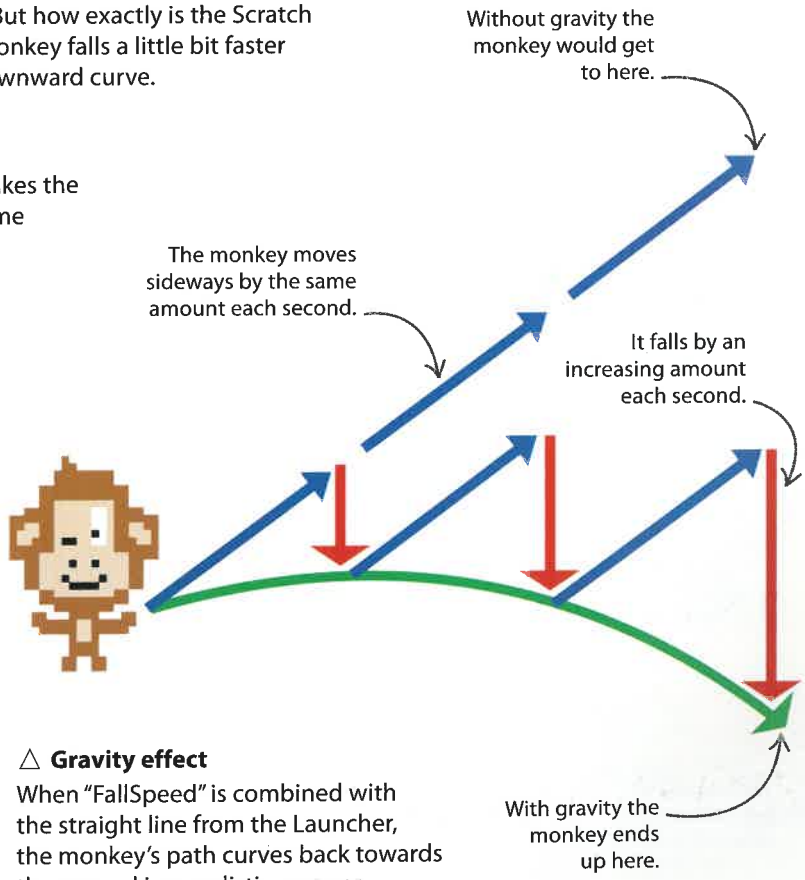
15 Run the project again – you can now direct the monkey over the tree to reach the tricky low bananas. But how exactly is the Scratch gravity working? Every second, the monkey falls a little bit faster than the second before, creating a downward curve.



◁ **Falling faster**

The "FallSpeed" variable makes the monkey fall further each time the "repeat" loop runs.





△ **Gravity effect**

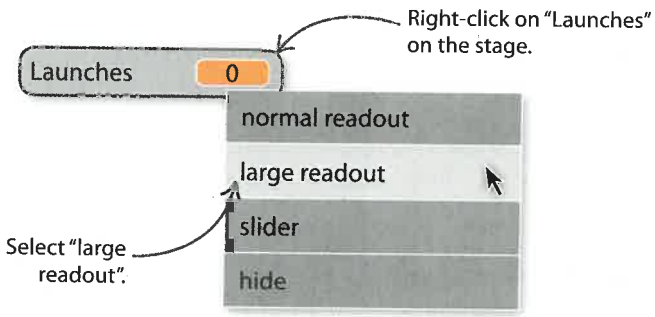
When "FallSpeed" is combined with the straight line from the Launcher, the monkey's path curves back towards the ground in a realistic manner.

Game over

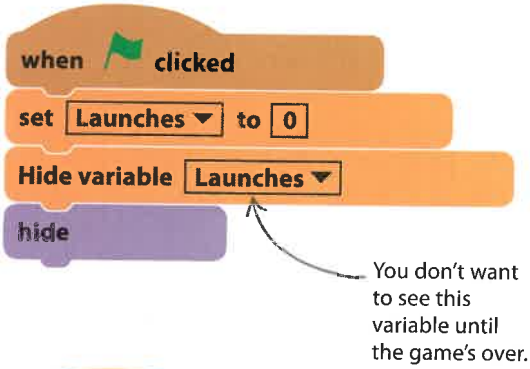
When the monkey has collected all the bananas, a "GameOver" message is broadcast, ending the game. Make a sign to go with it to tell the player how many launches they used to collect the bananas.

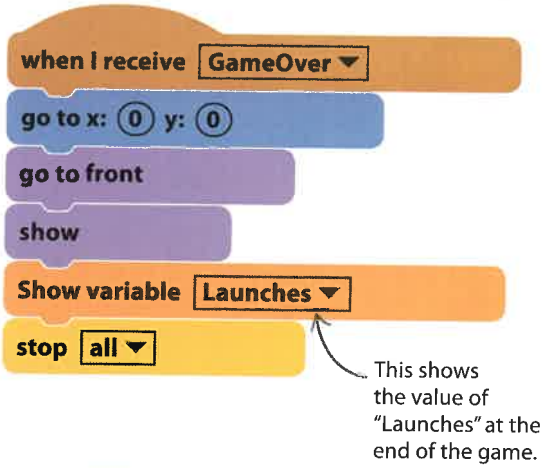
16 Click the paintbrush symbol to paint a new sprite and make a sign like the one below, leaving a gap in the text where the number of launches will go. You can make the sign as plain or as decorative as you like. Name the new sprite "Game Over".

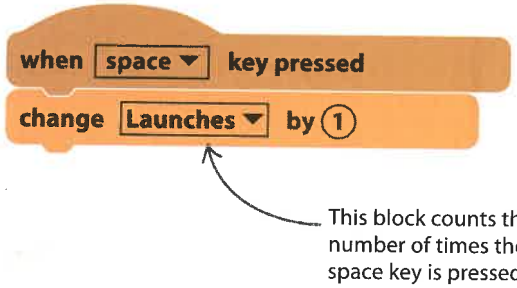
17 Now add a variable for all sprites to count the number of launches. Call this variable "Launches", show it on the stage, and right-click on it to change it to "large readout". This shows just the value and not the name of the variable. You'll reposition the launch counter later.



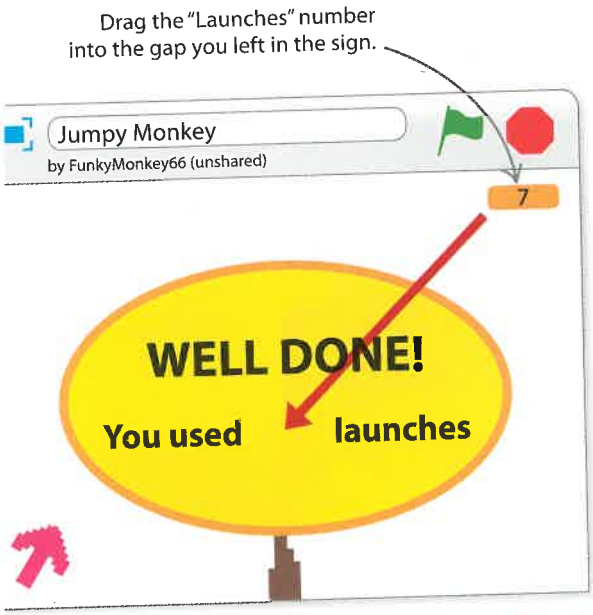
18 Now add these scripts to your sign. Together, they will count the number of times you launch the monkey and will display that number at the end of the game.



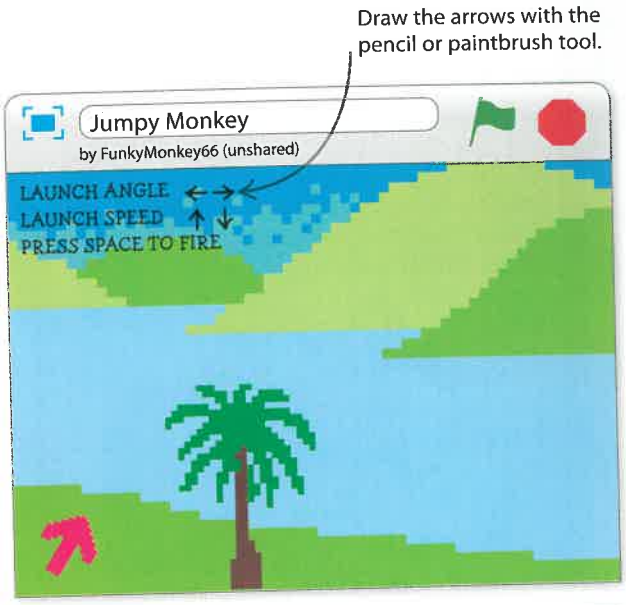




Run the game and collect all the bananas. When you see the "WELL DONE!" sign on the stage, drag the "Launches" counter into the gap in the sign. Scratch will remember its position in future games, so the sign will always be in the right place.



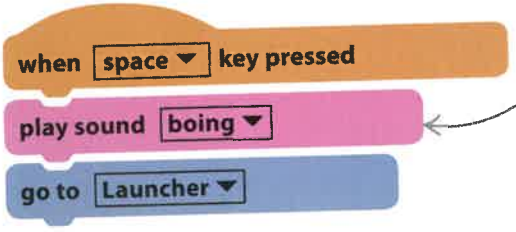
20 To add a backdrop, click on the stage information area in the bottom left and then choose the Backdrop tab at the top. Either paint your own scenery or load an image from the library. Use the text tool to add the game's instructions to the image, as shown below.



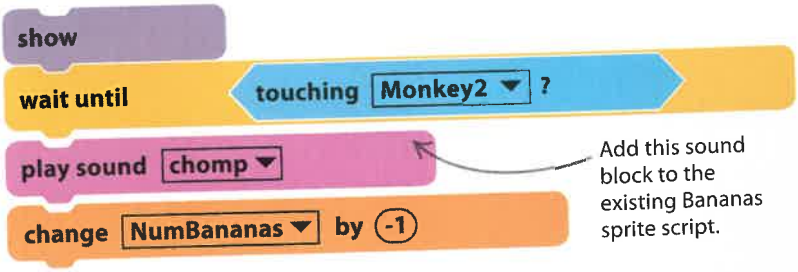
Make some noise

To make the game more interesting, you can add some sound effects. Follow the instructions below to play different sounds when the monkey is launched and when he eats the bananas.

21 Click the Monkey sprite, select the Sounds tab, and load "boing" from the library. Then add a "play sound" block to the existing monkey script in the position shown here. This will make the "boing" sound play every time the monkey jumps.



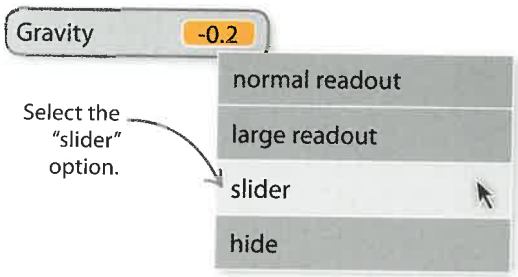
22 Click the Bananas sprite and load "chomp" from the sound library. Then add a "play sound" block to the existing banana script in the position shown here. Now the "chomping" sound will play each time the monkey gets a banana.



Playing with gravity

Add a slider to the game to allow you to experiment with the "Gravity" variable. The slider will allow you to tweak the "Gravity" value – you can even make the monkey fall upwards.

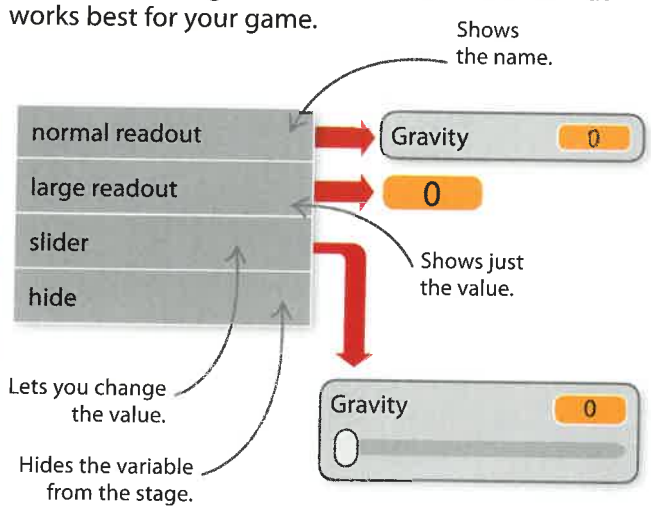
23 To adjust gravity in your game world, show the "Gravity" variable on the stage by ticking its box in the Data section. Then right-click the variable display on the stage and select "slider". The slider lets you change the value of a variable on the stage.



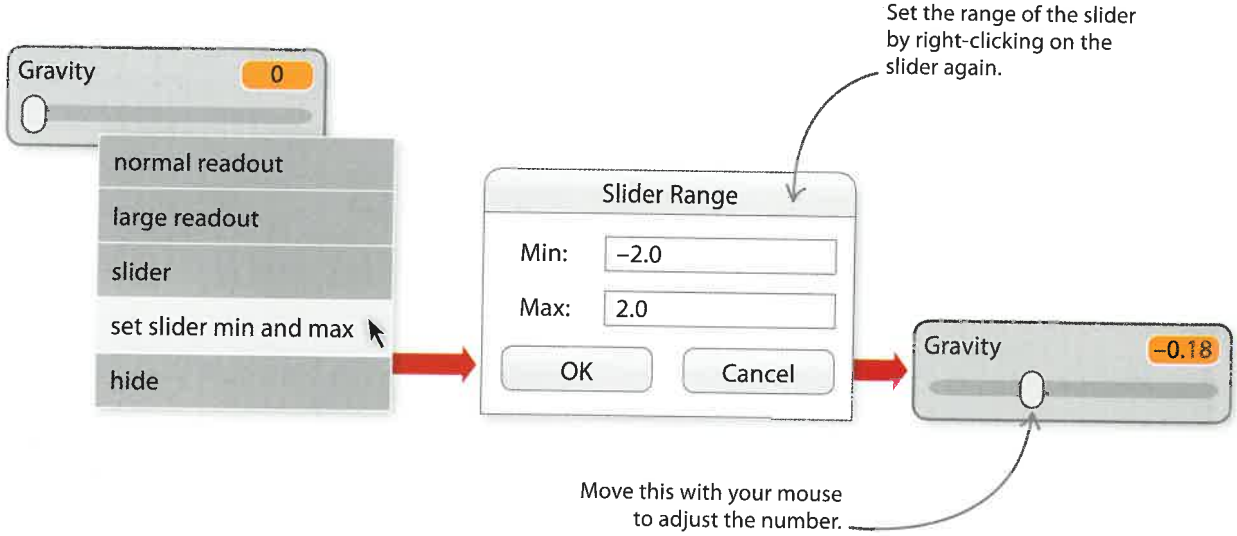
EXPERT TIPS

Displaying variables

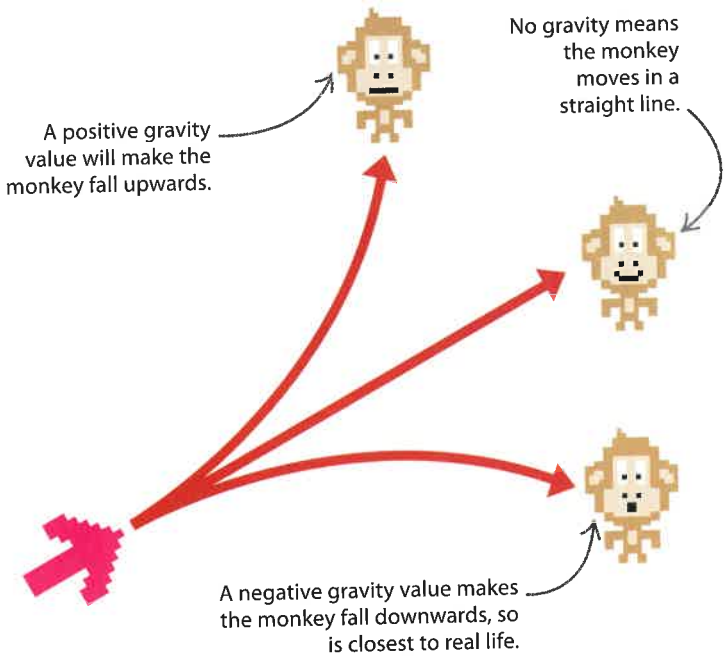
You can change how a variable is shown on the stage. There are three different options: normal readout, large readout, and slider. You can also hide the variable using this menu. Choose the look that works best for your game.



24 To set the range of the variable, right-click on the slider and type in the minimum and maximum values – for this game use –2.0 and 2.0. Make sure you type 2.0 not just 2, or the slider will only allow you to select whole numbers within the range.



25 Now play around with the gravity settings in this game using the slider. Using the suggested value of –0.2 works well, but have a look at what happens when you increase or decrease this number – if it is positive the monkey will fall upwards.



26 When you've finished experimenting with gravity, right-click on the slider and select "hide" to return the game to normal. Now you know how gravity works, you could try making a version of the game with reverse gravity so the monkey falls upwards. Think about what changes you'd need to make to the game for this to work, like moving the Launcher to fire downwards.

GAME DESIGN

Game physics

Physics is the science of forces and movement in the real world. Game physics is all about getting that science into games, so that things react and move around in realistic ways – being pulled down by gravity, for instance, or bouncing. Programmers have to solve all sorts of physics problems to make games more realistic or fun. When objects collide, should they bounce or crunch? How should objects move when they go underwater or into space?

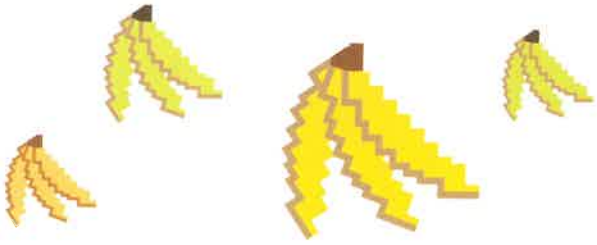
△ **Defying gravity**
Game physics doesn't have to be like real-world physics – you can create worlds with gravity that makes things fall upwards or even sideways. Gravity can be much stronger or weaker than in real life – perhaps balls fly higher with each bounce, until they shoot off into space.

Hacks and tweaks

Congratulations – you've built your first game with gravity. Once you've tried the game a few times, you can start to play around with the code to make the game your own. Here are a few ideas to try out.

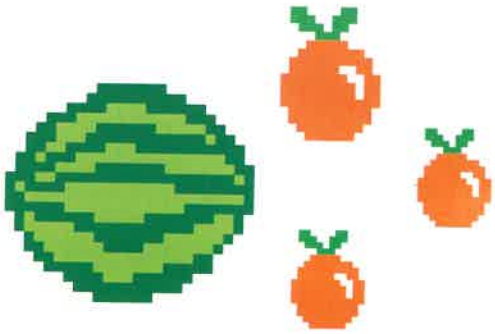
◁ Banana bonanza

Try adding more bananas, making then bigger or smaller, and put them in different places on the screen.



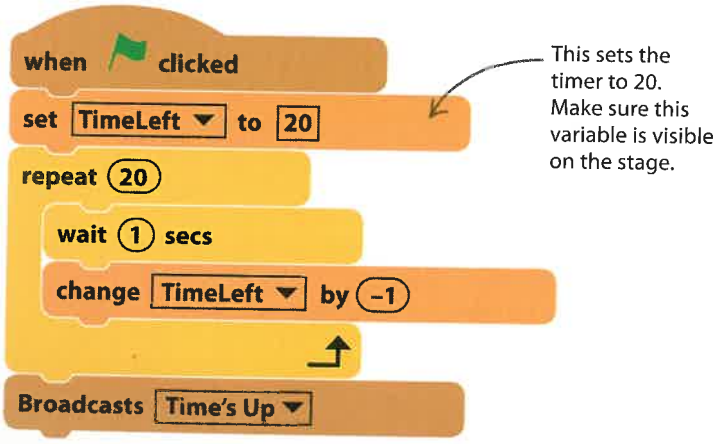
▽ Fruit salad

Add more fruit with a different score for each type. You'll need to make a "Score" variable and add extra sprites – there are oranges and watermelons in the Scratch sprite library.

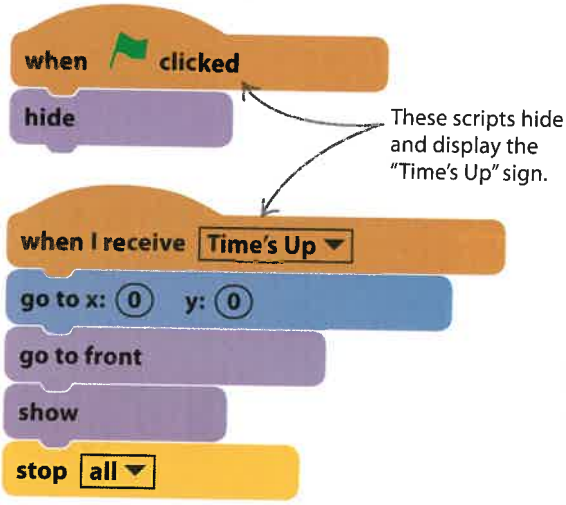


▽ Beat the clock

You can add a timer to make the player complete the game in a set time. Create a new variable called "TimeLeft" and add the script below to the Monkey2 sprite. Then create a new sprite, click on the Costumes tab, and make a sign that says "Times Up!" Finally, add the two scripts on the right to this sprite.



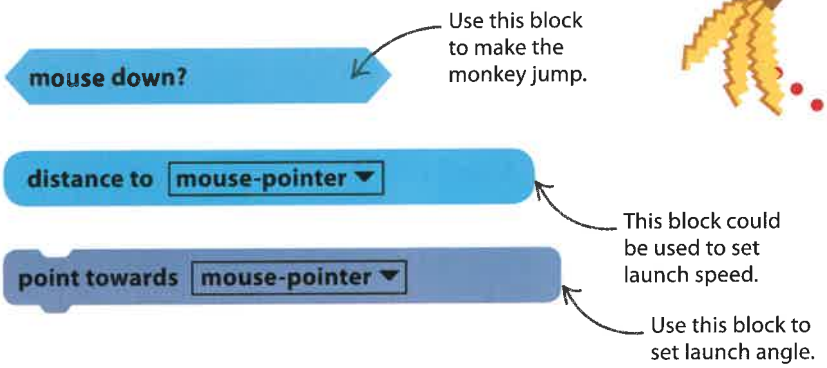
This sets the timer to 20. Make sure this variable is visible on the stage.



These scripts hide and display the "Time's Up" sign.

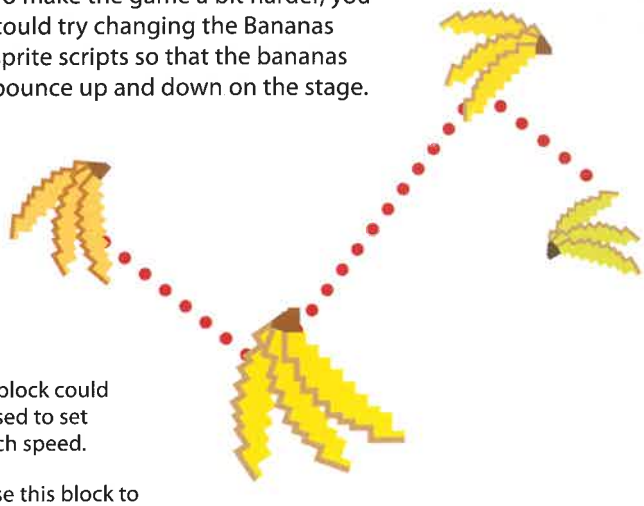
▽ Mouse control

You could use a computer mouse as the controller for this game instead of the keyboard. The three blocks below allow you to set the launch angle and speed as well as making the monkey jump. See if you can figure out some code to use them.



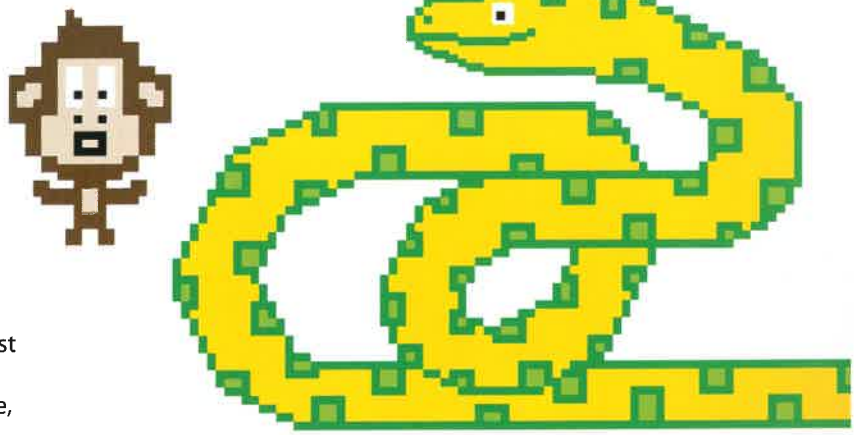
▷ Bouncing bananas

To make the game a bit harder, you could try changing the Bananas sprite scripts so that the bananas bounce up and down on the stage.



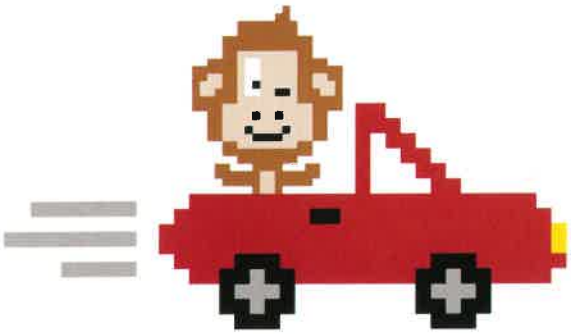
▷ Danger! Snake!

Add another challenge by creating an obstacle that gets in the monkey's way or maybe ends the game – perhaps a giant monkey-eating snake or spider?



▽ Bug or bonus?

You might have discovered that you can adjust the monkey's speed in flight with the arrow keys. You can fix this by adding a new variable, "MonkeySpeed", and copying the value of "LaunchSpeed" into it at launch. Then use MonkeySpeed not LaunchSpeed in the move block for the monkey. Or, if you enjoy being able to change the monkey's speed, leave the game as it is.



▽ Launch speed slide

You've already tried adding a slider to control gravity. You could also add a slider to adjust launch speed.