Neil the seal

Code a game where you help Neil the seal waddle around town



Step 1 What you will make

Neil the seal is loose in town, and he's out to cause chaos! In this project, you'll code a game where Neil waddles around smashing every sign, car, and barrier he can find, scoring points as he goes — all while a ranger chases after him, trying to send him back to the sea.

Have a play with the finished game to see what you'll make.

Step 2 Make Neil move

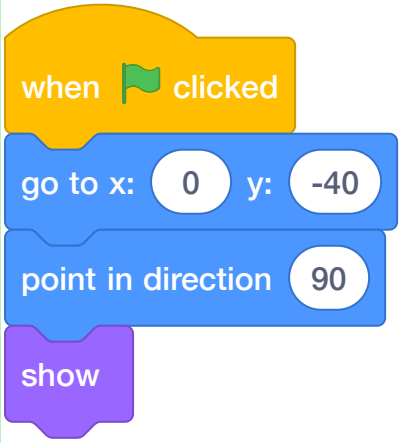
In this step, you'll get Neil the seal moving around the town using the arrow keys.

Open the Neil the seal starter project (https://scratch.mit.edu/projects/1352072422/editor).

The starter project has all the sprites and backdrops you need. ☑

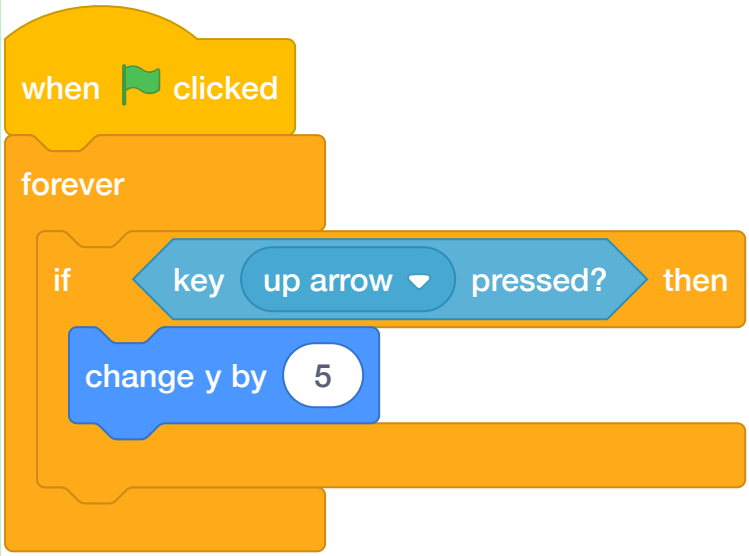
Click on the **Neil** sprite so you can add your code to it.

First, make sure Neil always starts in the same place. Add a `when green flag clicked` block, then send Neil to the middle of the road with a `go to x: y:` block, add a `point in direction` block so he faces right, and `show` him. ☑



Add another `when green flag clicked` block, then add a `forever` loop.

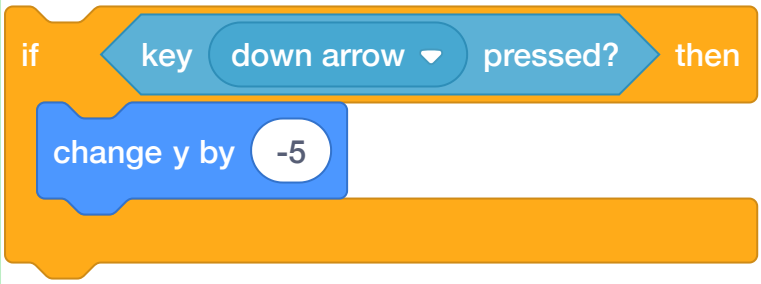
Inside the loop, add an `if ... then` block that checks whether the `up arrow` key is pressed. To make Neil move up when it is, add a `change y by` block.



Click on the green flag and press the up arrow. Neil should move up the screen.

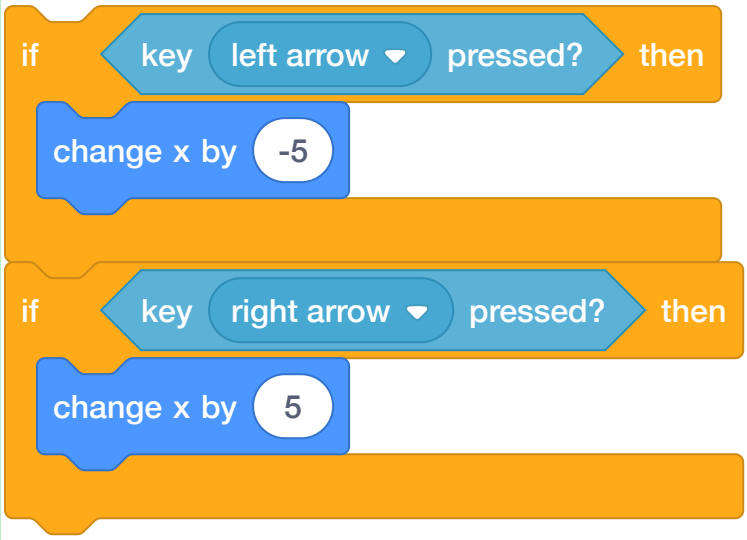
Now you can add movement for the other three arrow keys.

Add another `if ... then` block inside the `forever` loop, below the first one. This time, check for the `down arrow` key and move Neil down.



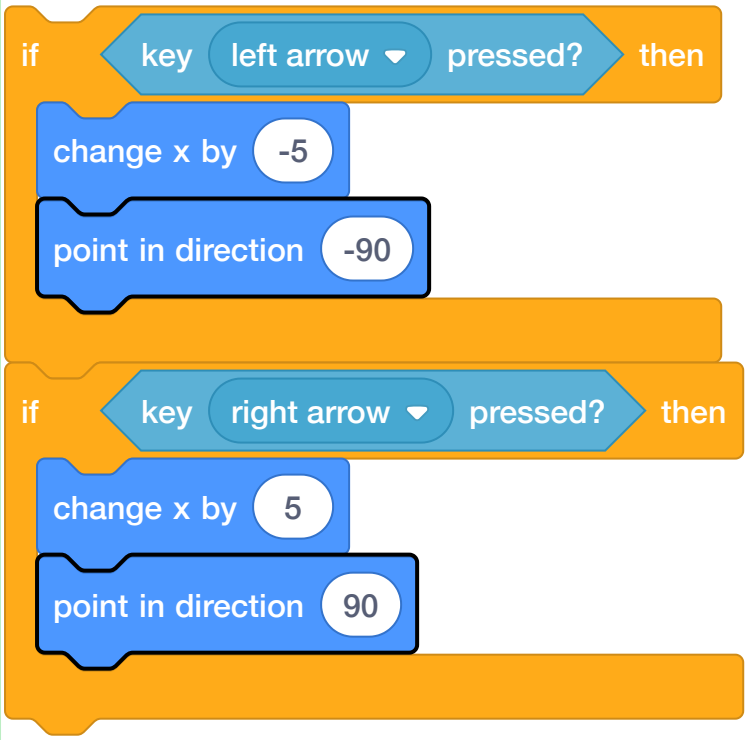
Moving up and down changes Neil's `y` position. To move him left and right, you need to change his `x` position instead.

Add two more `if ... then` blocks to check for the `left arrow` and `right arrow` keys, using `change x by` to move Neil.



Click on the green flag and try all four arrow keys. Neil moves around, but he always faces the same way. You can make him face left and right as he walks.

Add a `point in direction` block inside the `left arrow` and `right arrow` `if ... then` blocks. Point Neil in direction `-90` to face left, and `90` to face right.



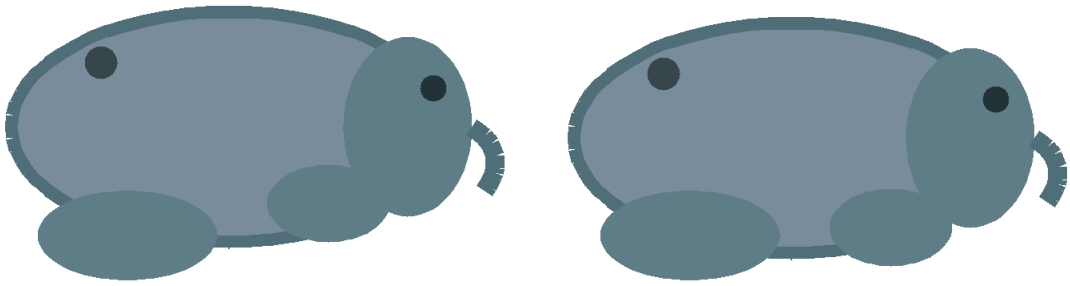
Right now, Neil slides around without moving his body. You can bring him to life by giving him more than one costume and switching between them as he walks.

Click on the **Costumes** tab. Neil has one costume, called **seal1**.

Right-click on **seal1** and choose **duplicate**. This makes a copy called **seal2**.

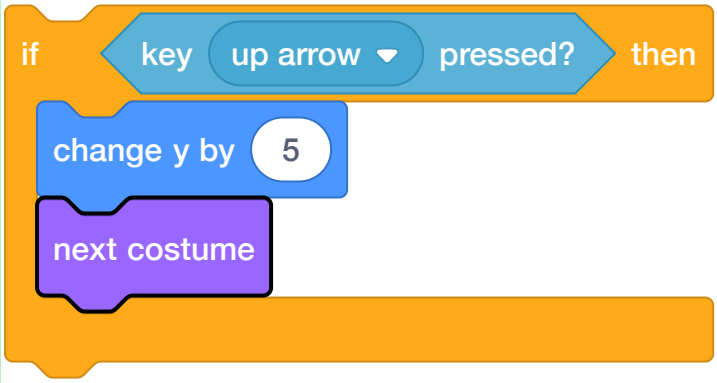
In the Paint editor, click on one of Neil's legs to select it and move it a little. Do the same to his other leg, so **seal2** has the legs in a different position to **seal1**.

Here is how the two costumes look in the example project. It's up to you how you move the legs — the more different your two costumes are, the more Neil will look like he's moving.



seal1 seal2

Go back to the **Code** tab. Add a `next costume` block inside each of your four arrow key `if ... then` blocks, so Neil changes costume whenever he moves.



Click on the green flag and move Neil around. His two costumes swap back and forth, so he looks like he's waddling along.

Two costumes give Neil a simple wobble. Adding a few more makes his movement much smoother.

Duplicate one of your costumes to make **seal3**, then move the legs again to make a new pose. Keep going until Neil has five costumes, each with the legs in a slightly different position.

In the example project, Neil has five costumes that move his legs up and down as he waddles. Yours can be as different as you like.



seal1 seal2 seal3 seal4 seal5

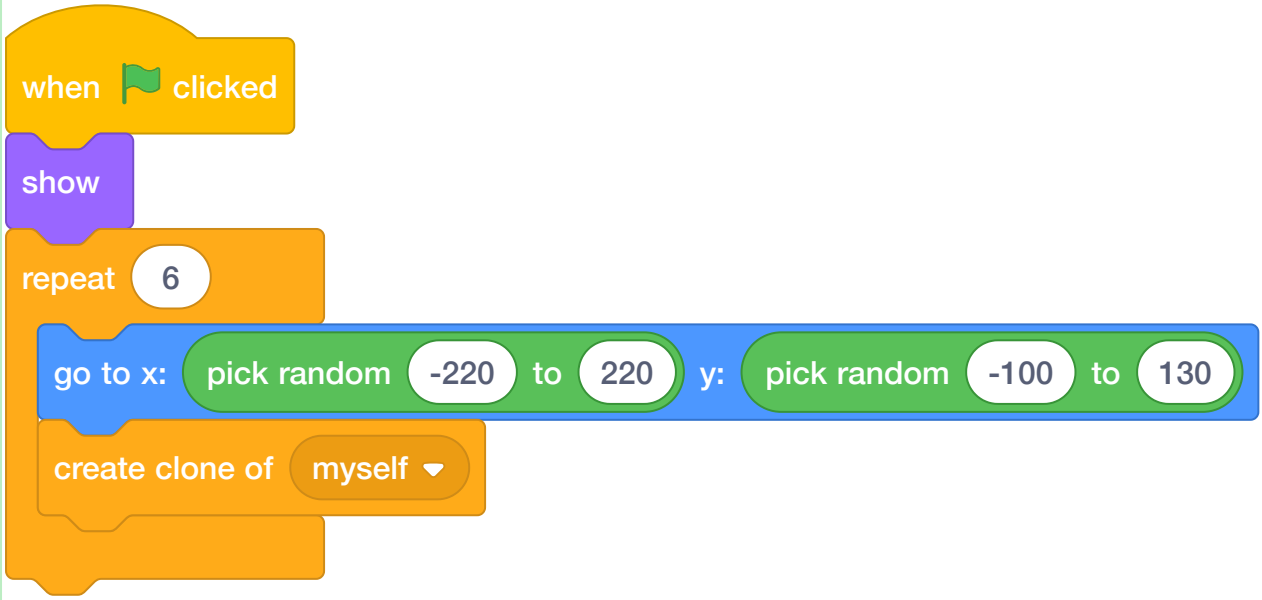
Now the `next costume` block cycles through all five poses, giving Neil a smooth waddle as he explores the town.

Step 3 Add stuff to smash

In this step, you'll fill the town with stop signs for Neil to smash.

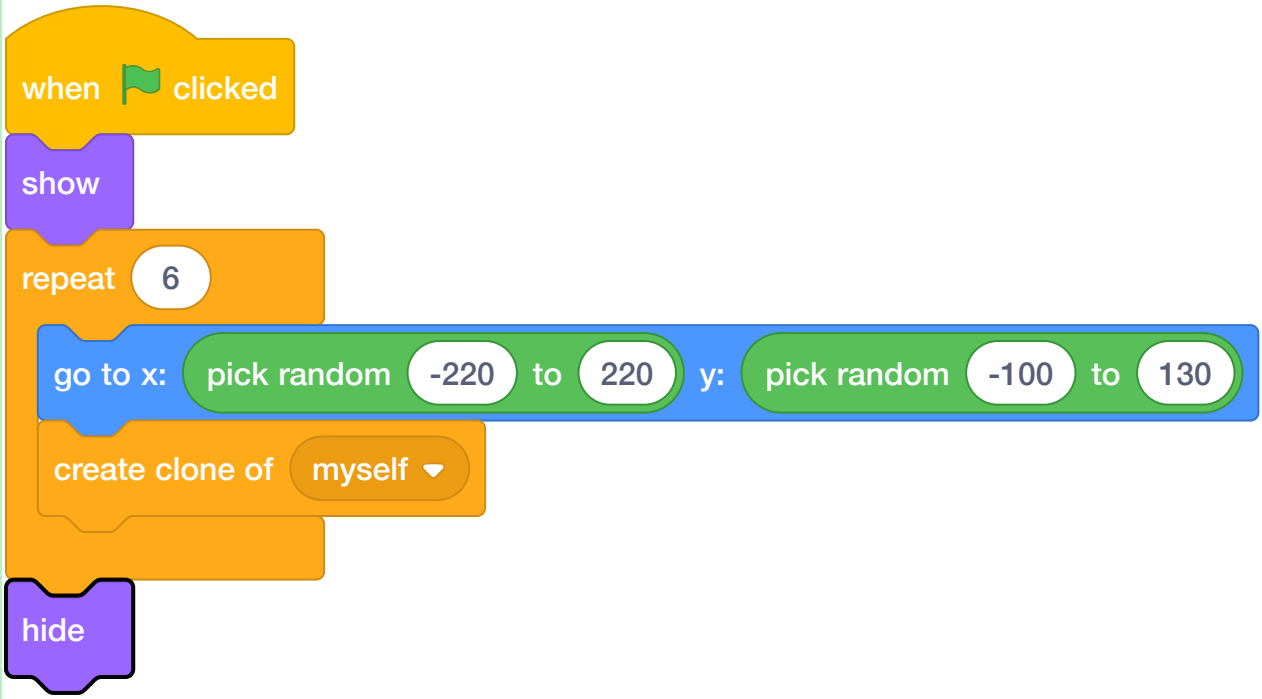
Click on the **Sign** sprite.

Add a `when green flag clicked` block, then a `show` block, because the sign starts off hidden. Then use a `repeat` loop to move to six random spots, making a clone at each one.



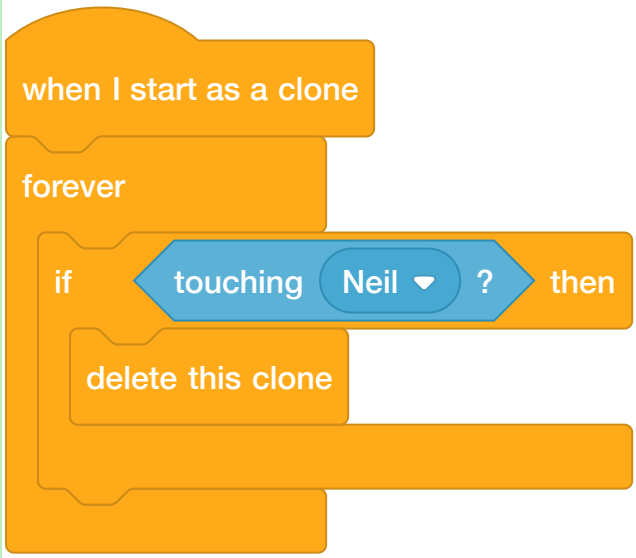
Click on the green flag. Six stop signs appear around the town — but you can still see the original sign that the clones are made from.

Add a `hide` block to the end of the script, so the original sign is hidden once all the clones have been made.



Now only the six clones are visible.

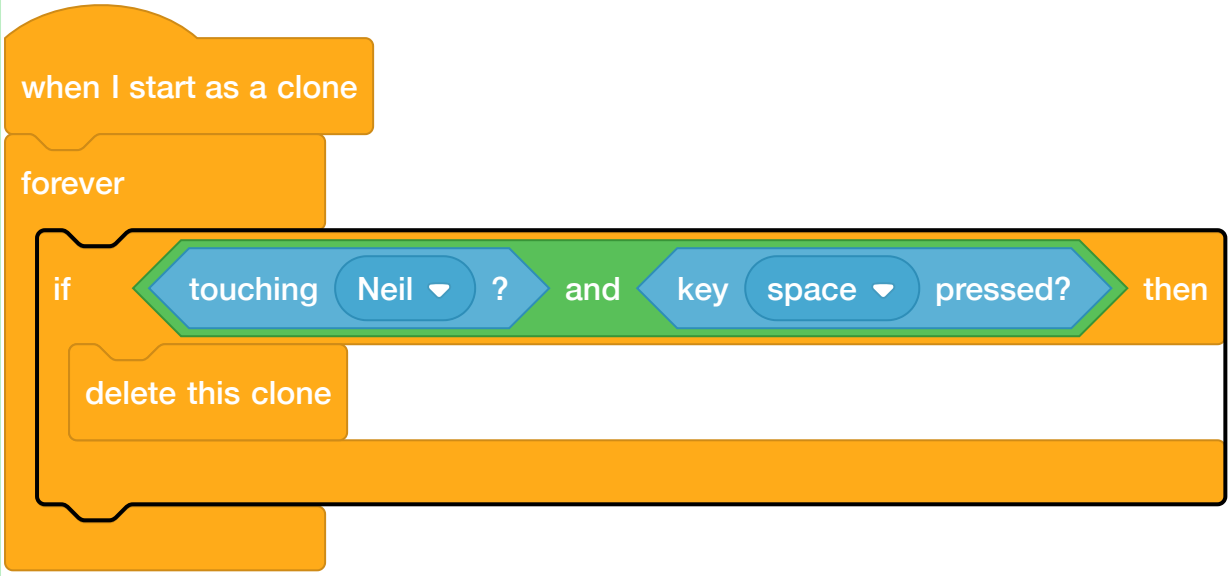
Start a new script with a `when I start as a clone` block. Add a `forever` loop, and inside it use an `if ... then` block to check whether the clone is `touching Neil`. If it is, `delete this clone`.



Move Neil into a sign and it disappears.

Right now, the signs vanish the moment Neil touches them. To make Neil **smash** them, the player should have to press the space bar too.

Change the `if ... then` condition so it also checks whether the `space` key is pressed, using an `and` block to join the two checks.



Now when Neil bumps into a sign, the player has to press the space bar to smash it.

Step 4 Reward Neil

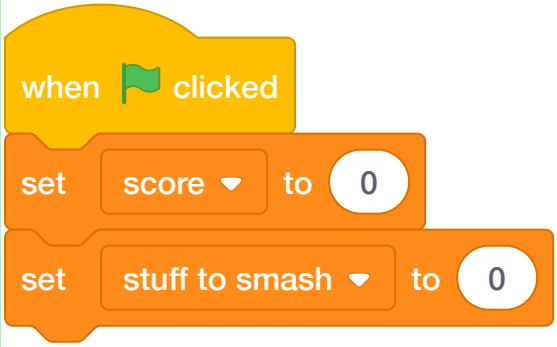
In this step, you'll keep score as Neil smashes things, and keep track of how much is left to smash.

Click on the **Stage**.

Make two new `variables`, `score` and `stuff to smash`, and choose **For all sprites** for each one.

Tick the checkbox next to `score` so the player can see it, and untick `stuff to smash` to hide it.

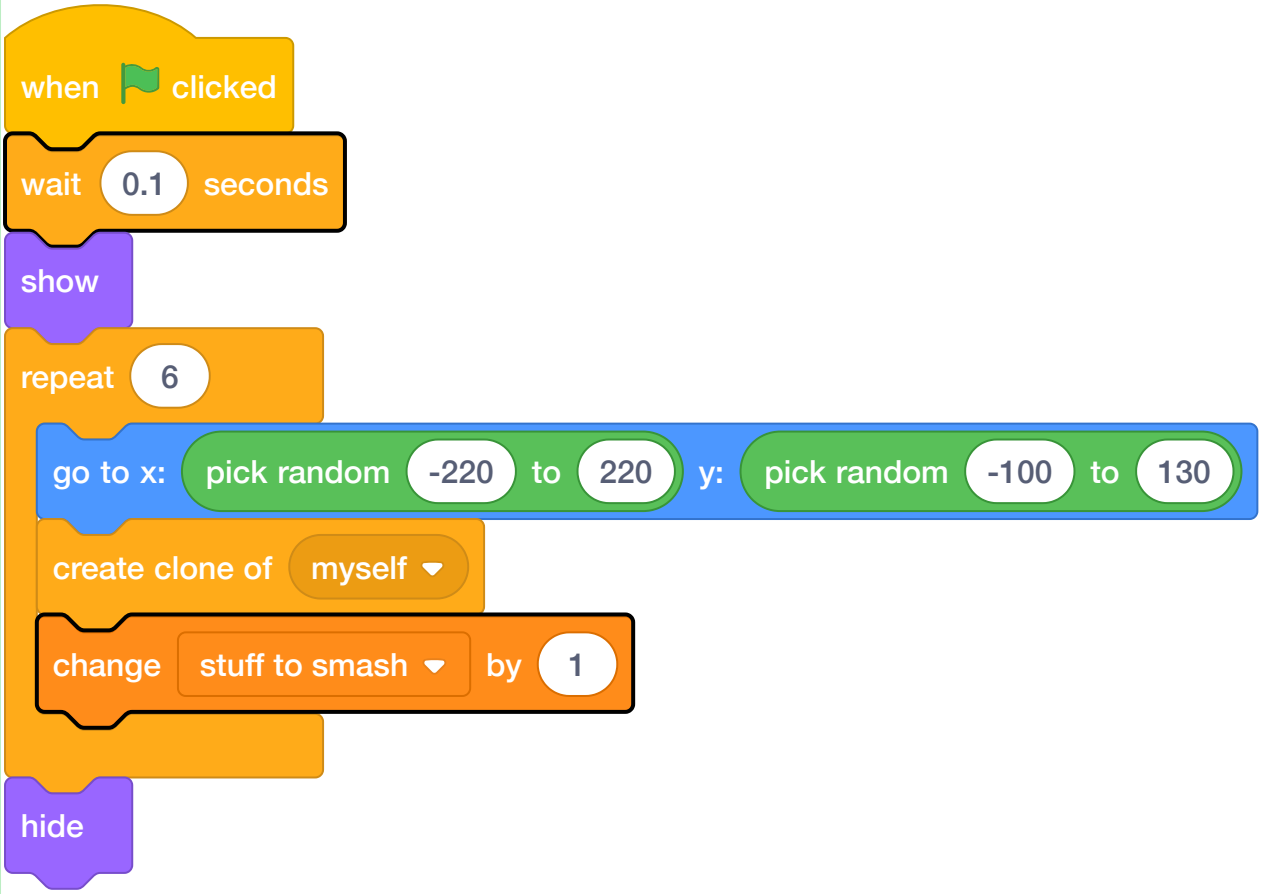
Add a `when green flag clicked` block to the **Stage** that sets both variables to `0`.



Click on the **Sign** sprite. Each sign that gets made should add 1 to the total amount of stuff to smash.

Inside the `repeat` loop, add a `change stuff to smash by` block to count each clone as it's created.

Then add a `wait` block set to `0.1` seconds at the very top of the script. This gives the **Stage** time to reset `stuff to smash` to `0` before the signs start being counted.

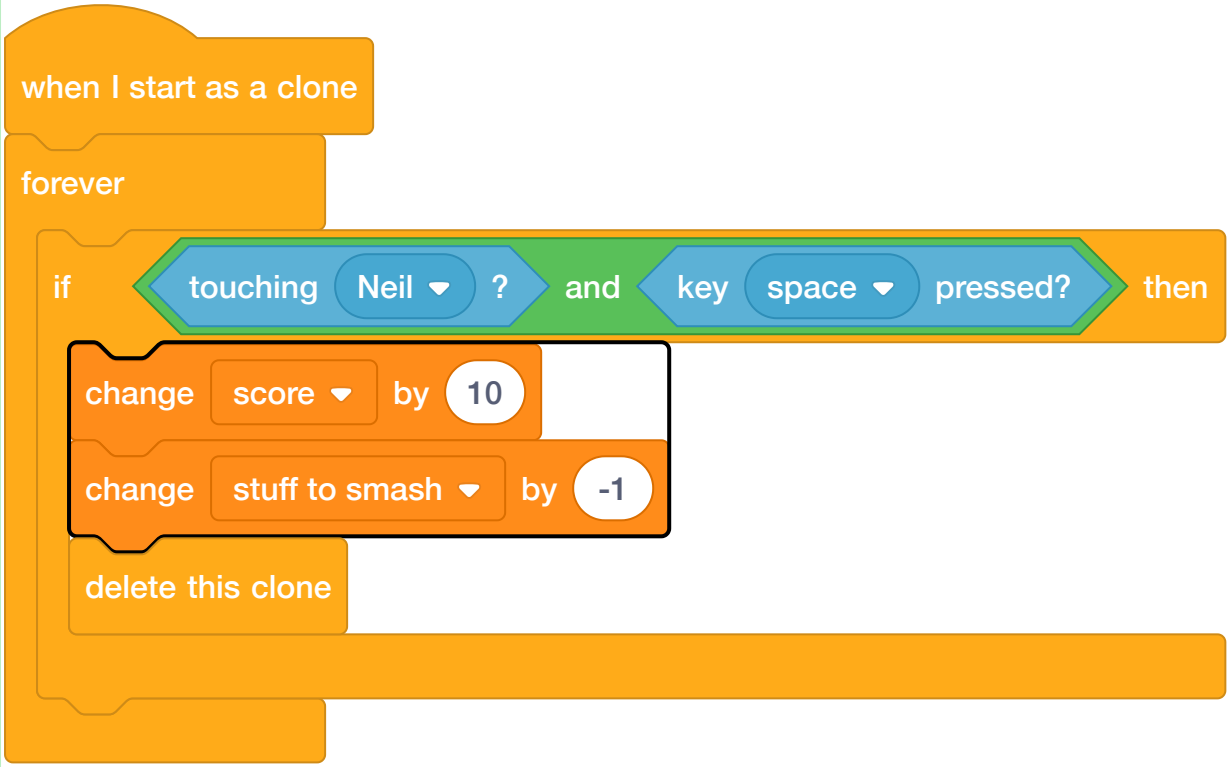


Tip

Both this script and the Stage's setup script start when the green flag is clicked, so they run at the same time. Without the short wait, the signs could start adding to `stuff to smash` before the Stage sets it back to `0`, and the total would come out wrong.

Now, when Neil smashes a sign, reward the player with some points and take 1 off the amount left to smash.

In the clone script, inside the `if ... then` block, add a `change score by` block to add `10` points and a `change stuff to smash by` block to take away `1`.



Click on the green flag and smash some signs. Your score goes up, and the amount of stuff left to smash goes down.

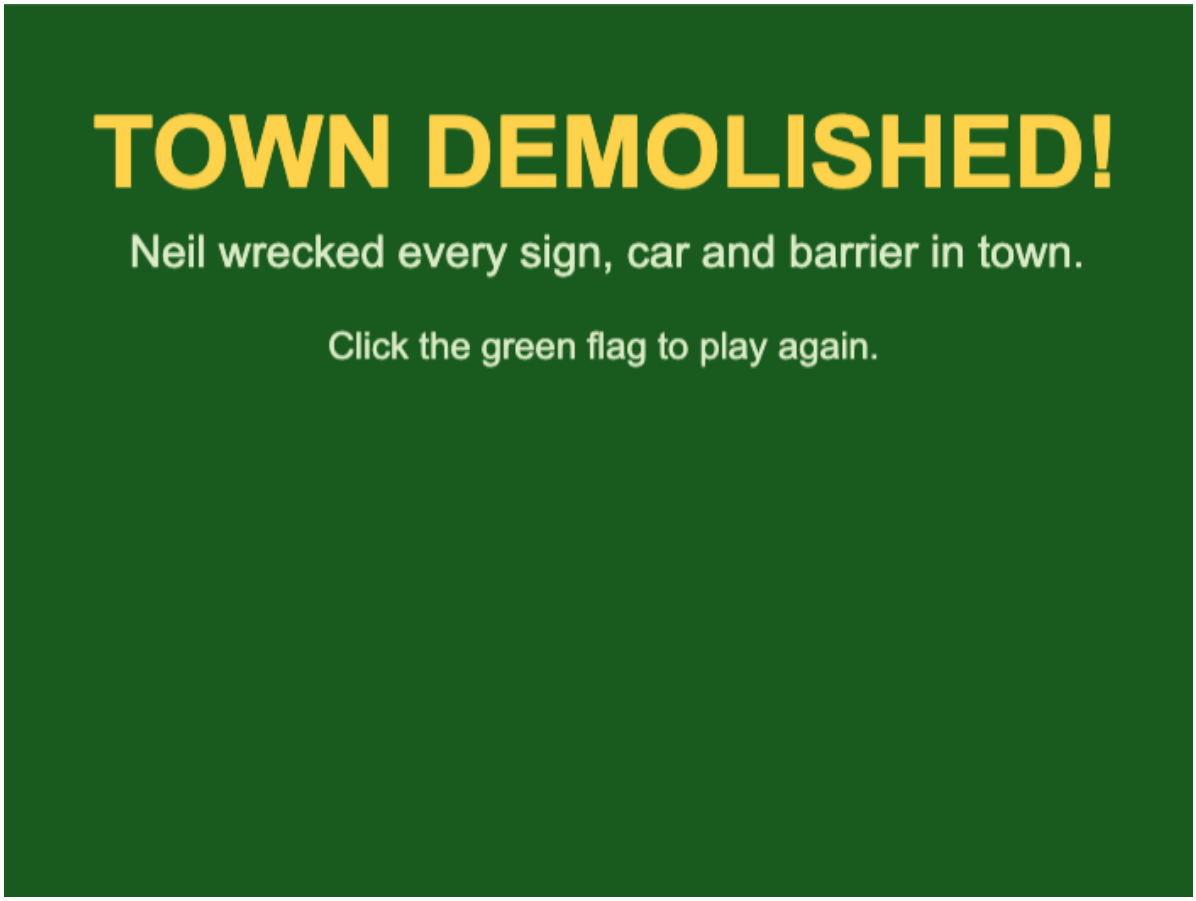
Step 5 Winning

In this step, you'll let the player win by smashing everything in town.

Click on the **Stage**, then click on the **Backdrops** tab.

Hover over **Choose a Backdrop** and click on **Paint** to make a new, blank backdrop. Call it `Win`, then give it a background and some text to celebrate finishing the game — you can write whatever you like.

Here's the **Win** backdrop from the example project:

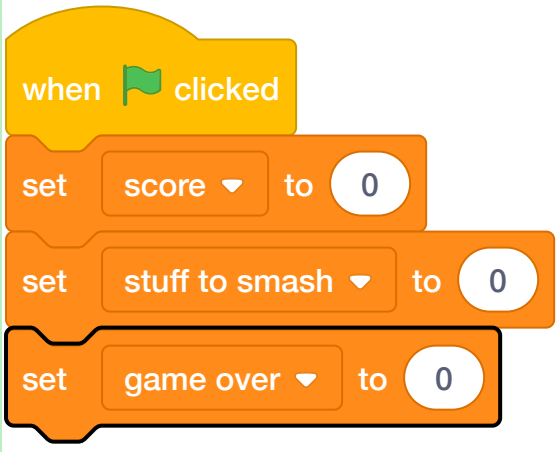


Make a new `variable` called `game over`, and choose **For all sprites**.

This one is just for the code to use, so untick its checkbox to hide it from the player.

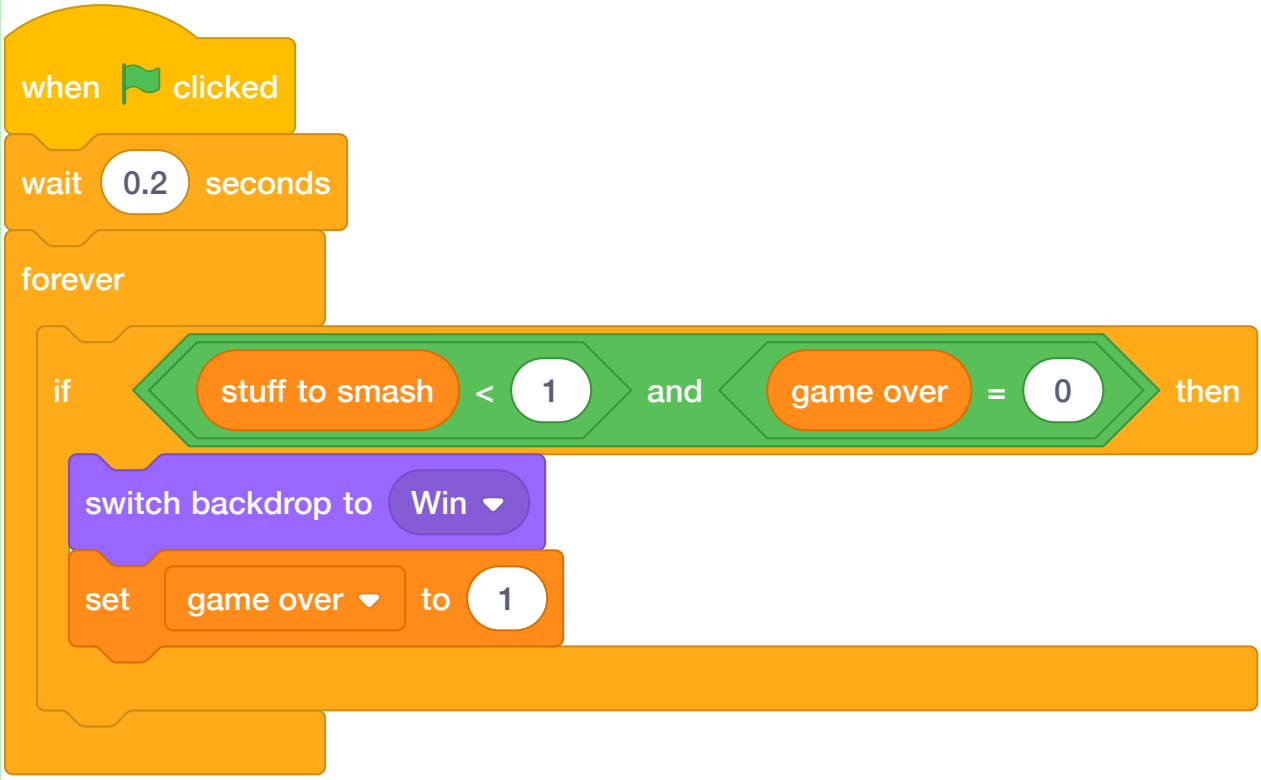


On the **Stage**, add it to your `when green flag clicked` script and set it to `0` at the start.

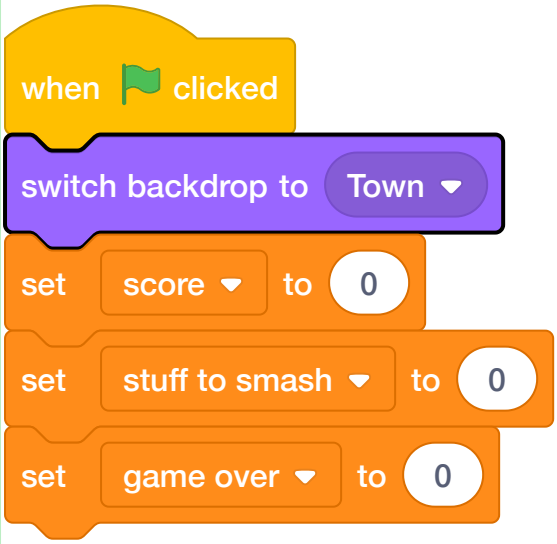


The player wins when there's nothing left to smash. On the **Stage**, add a new script to check for this.

Start with a `wait` block set to `0.2` seconds, so the signs, cars, and barriers have time to be counted before the check begins. Then use a `forever` loop with an `if ... then` block. When `stuff to smash` is less than `1` and `game over` is `0`, switch to your `Win` backdrop and set `game over` to `1`.

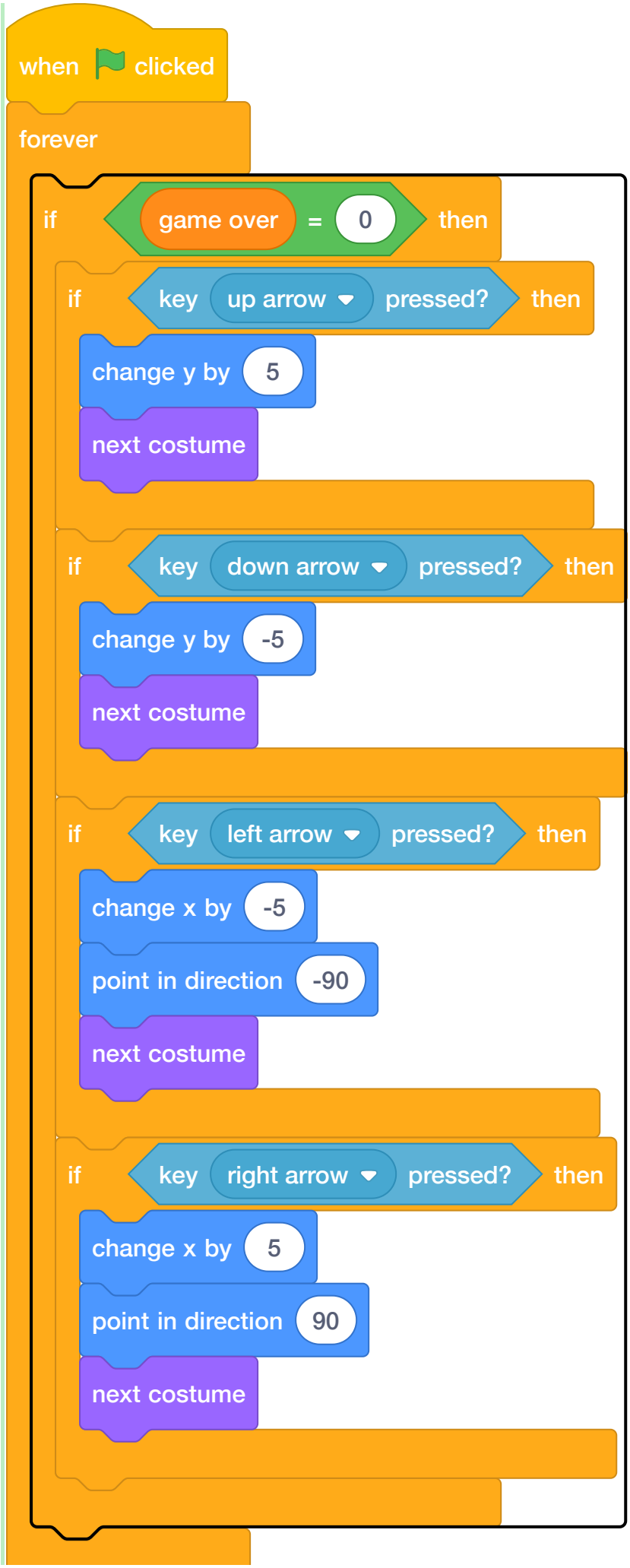


When the game starts, it should always begin on the **Town** backdrop. Add a `switch backdrop to` block to the top of your setup script.



When the game is over, Neil should stop moving. Click on the **Neil** sprite and wrap all of his movement code in an `if ... then` block that only runs while `game over` is `0`.




when clicked
forever
if game over = 0 then
if key up arrow pressed? then
change y by 5
next costume
if key down arrow pressed? then
change y by -5
next costume
if key left arrow pressed? then
change x by -5
point in direction -90
next costume
if key right arrow pressed? then
change x by 5
point in direction 90
next costume

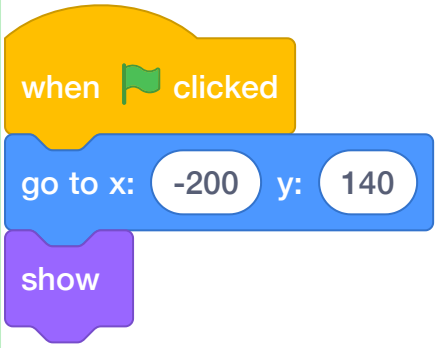
Click on the green flag and smash everything in town. When the last object is gone, your **Win** backdrop appears — and Neil stops moving.

Step 6 Chasing Neil

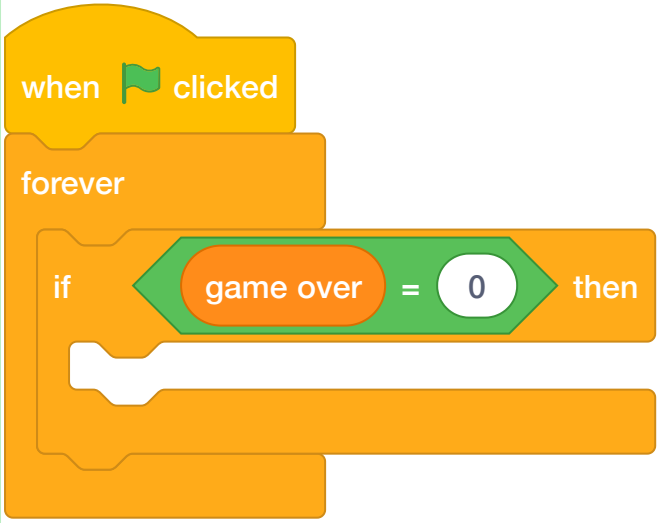
In this step, you'll add a ranger who chases Neil around the town.

Click on the **Ranger** sprite.

Add a `when green flag clicked` block. Use a `go to x: y:` block to place the ranger in the corner, and a `show` block to make sure they're visible.



The ranger should only chase Neil while the game is being played. Use a `forever` loop with an `if ... then` block that checks `game over` is `0`.



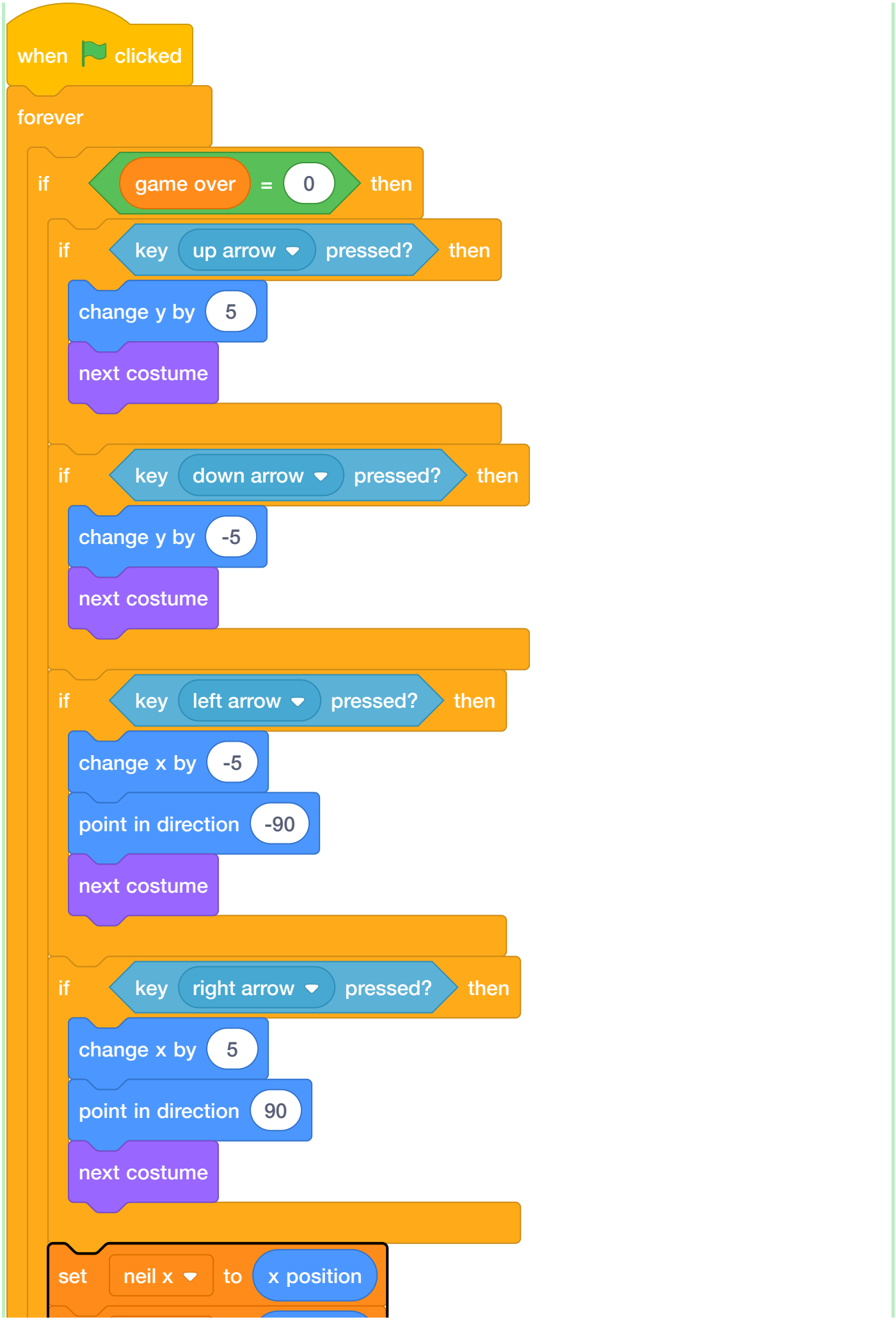
For the ranger to chase Neil, they need to know where Neil is.

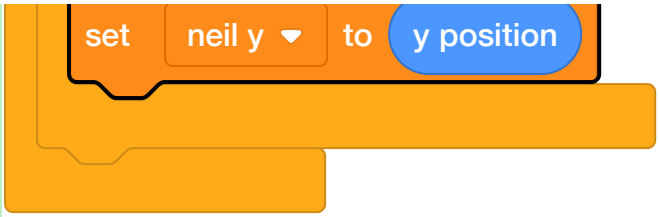
Click on the **Neil** sprite and make two new `variables`, `neil x` and `neil y`. Choose **For all sprites** so the ranger can read them too.

Untick both checkboxes to hide these from the player.

Still on the **Neil** sprite, find his movement script. At the bottom of the movement code, inside the `if ... then` block, set `neil x` and `neil y` to his current position.

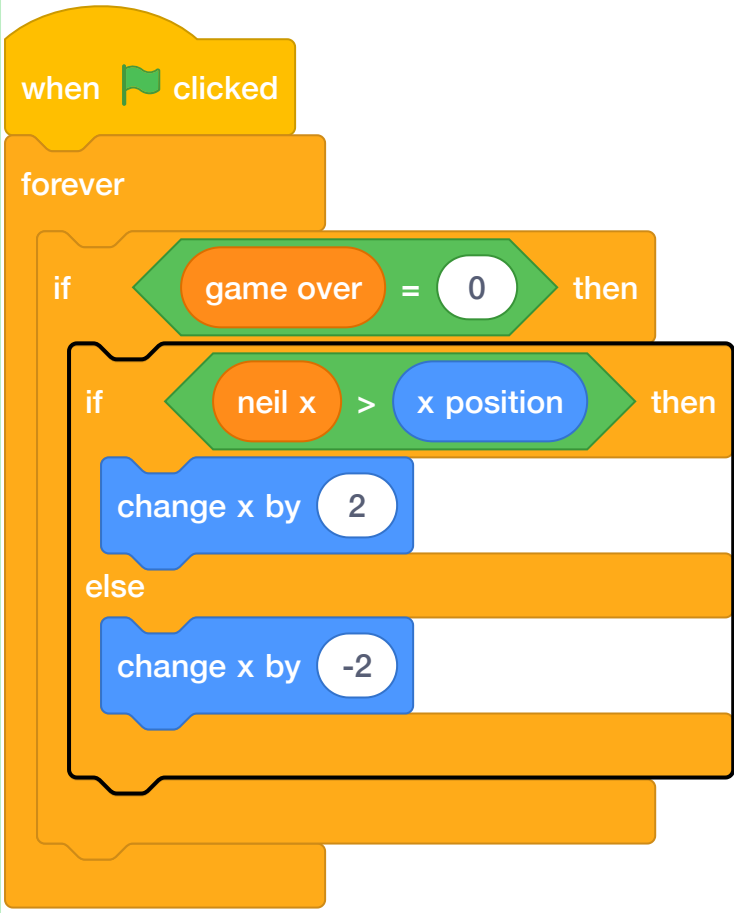



when clicked
forever
if game over = 0 then
if key up arrow pressed? then
change y by 5
next costume
if key down arrow pressed? then
change y by -5
next costume
if key left arrow pressed? then
change x by -5
point in direction -90
next costume
if key right arrow pressed? then
change x by 5
point in direction 90
next costume
set neil x to x position

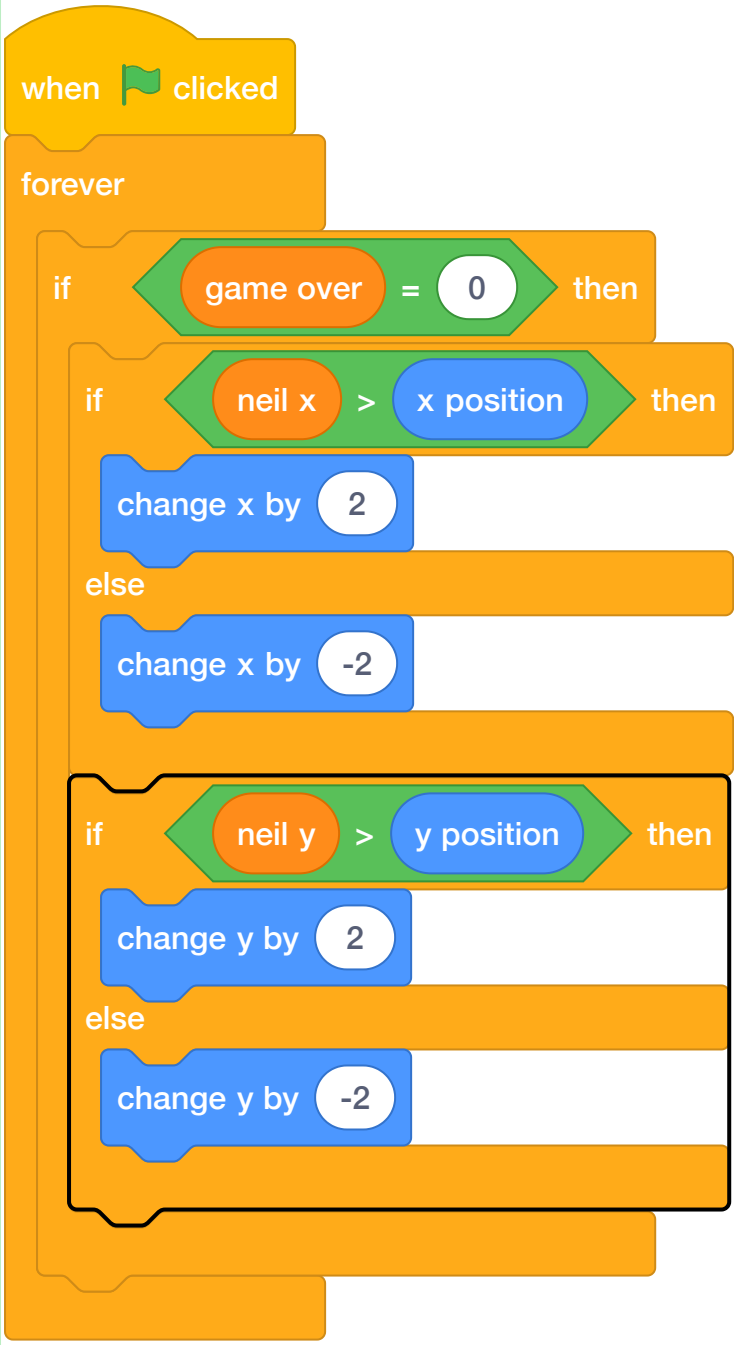


Go back to the **Ranger** sprite. Inside the `if ... then` block, make the ranger move towards Neil's x position.

Use an `if ... then ... else` block: if `neil x` is greater than the ranger's `x position`, `change x by` to move right, otherwise move left.



Now do the same for Neil's y position, so the ranger follows Neil up and down as well.



Click on the green flag. The ranger now chases Neil wherever he waddles.

Step 7 Catching Neil

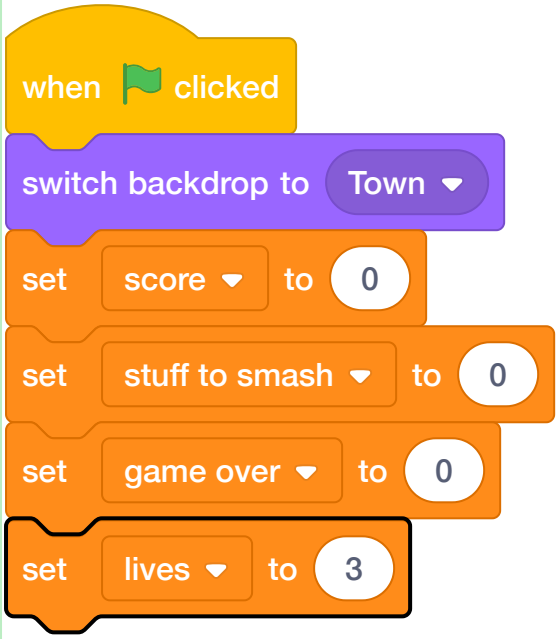
In this step, you'll give Neil some lives to lose when the ranger catches him.

Click on the **Stage**. Make a new `variable` called `lives`, and choose **For all sprites**.

Tick its checkbox so the player can see how many lives Neil has left.

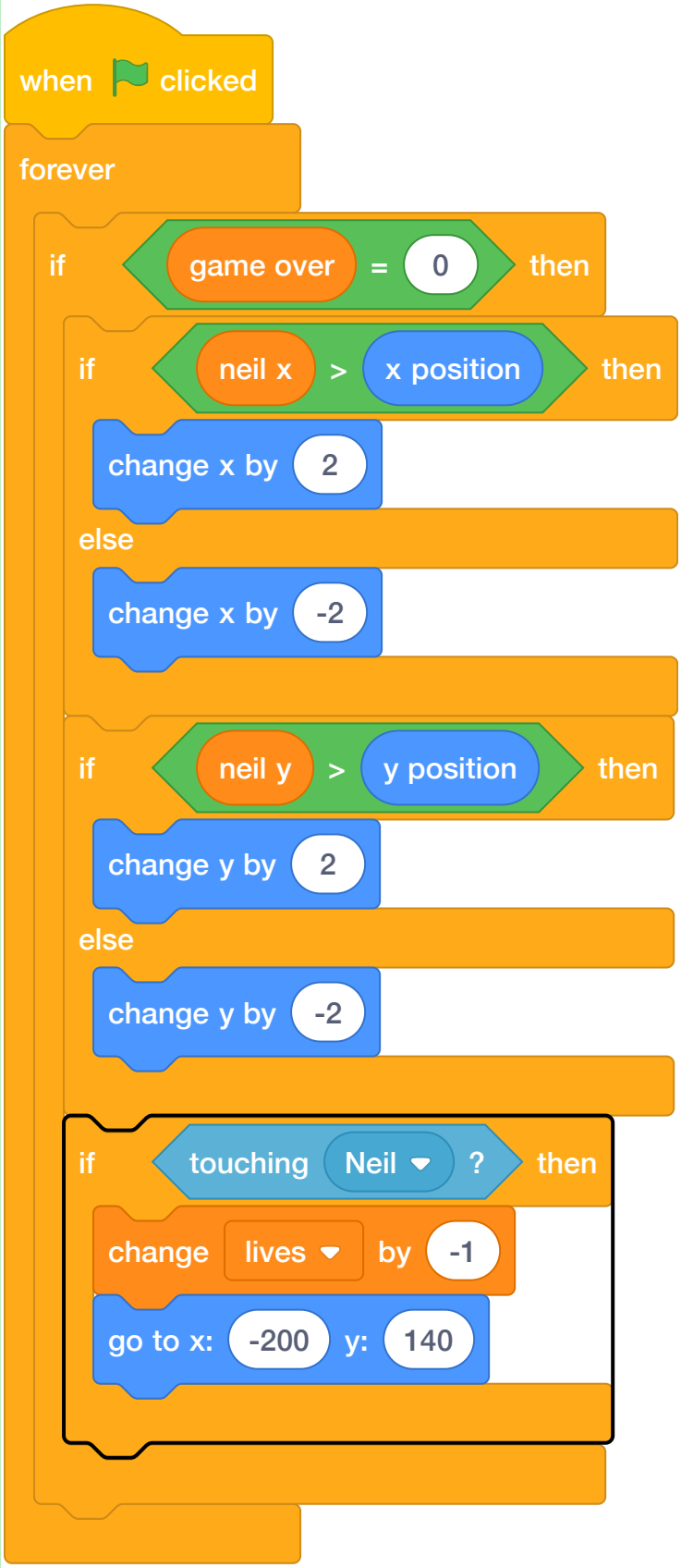


Add it to your `when green flag clicked` script and set `lives` to `3` at the start.



Click on the **Ranger** sprite. When the ranger catches Neil, Neil should lose a life.

At the bottom of the `if ... then` block, below the movement code, add an `if ... then` block that checks if the ranger is `touching Neil`. If they are, `change lives by -1` and send the ranger back to their starting position.

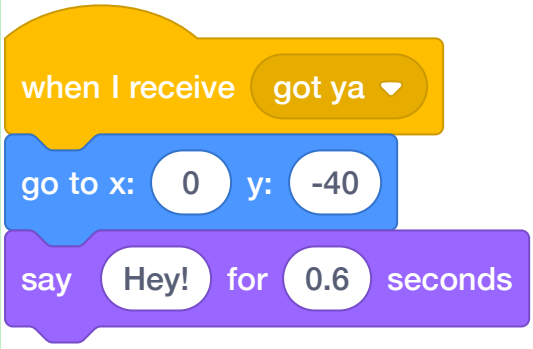


Now tell the rest of the game that Neil has been caught. Make a new `broadcast` message called `got ya`, and add a `broadcast and wait` block inside your new `if ... then` block.



Click on the **Neil** sprite. Neil needs to react when he's caught.

Add a `when I receive` block for the `got ya` message. Send Neil back to his starting position, then use a `say () for () seconds` to make him say `Hey!`.



Click on the green flag and let the ranger catch Neil. Neil says "Hey!", pops back to the start, and loses a life.

Step 8 Back to the sea, Neil

In this step, you'll end the game when Neil runs out of lives.

Click on the **Stage**, then click on the **Backdrops**** tab.

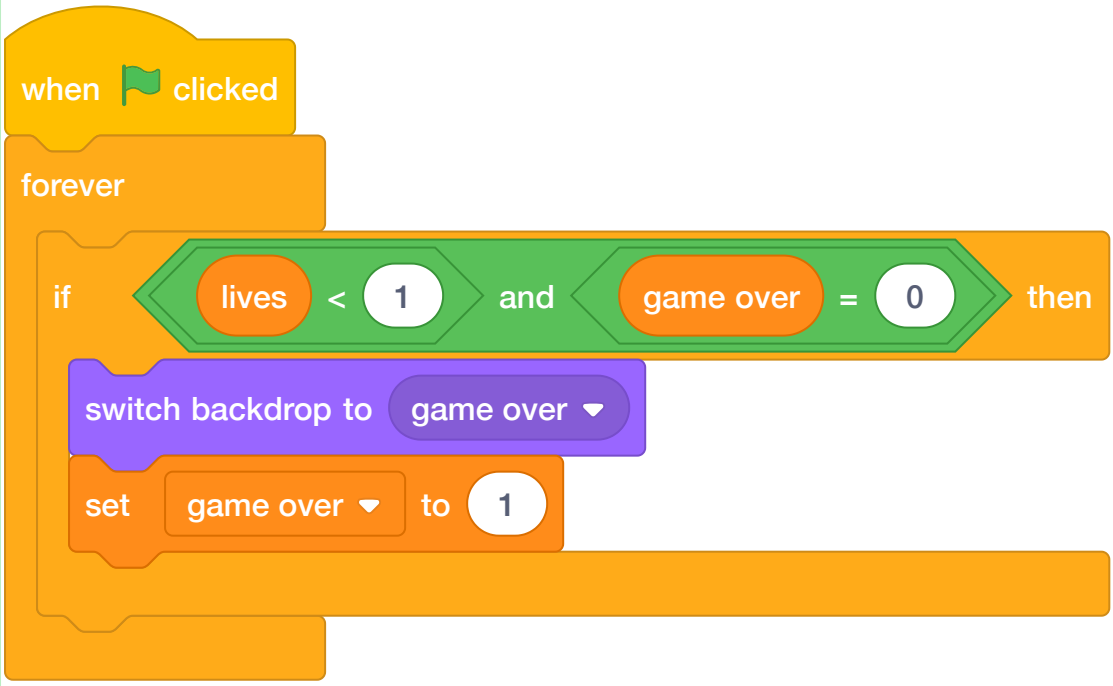
Hover over **Choose a Backdrop** and click on **Paint** to make a new, blank backdrop. Call it `game over`, for when Neil runs out of lives. As before, you can design it however you like.

Here's the **game over** backdrop from the example project:

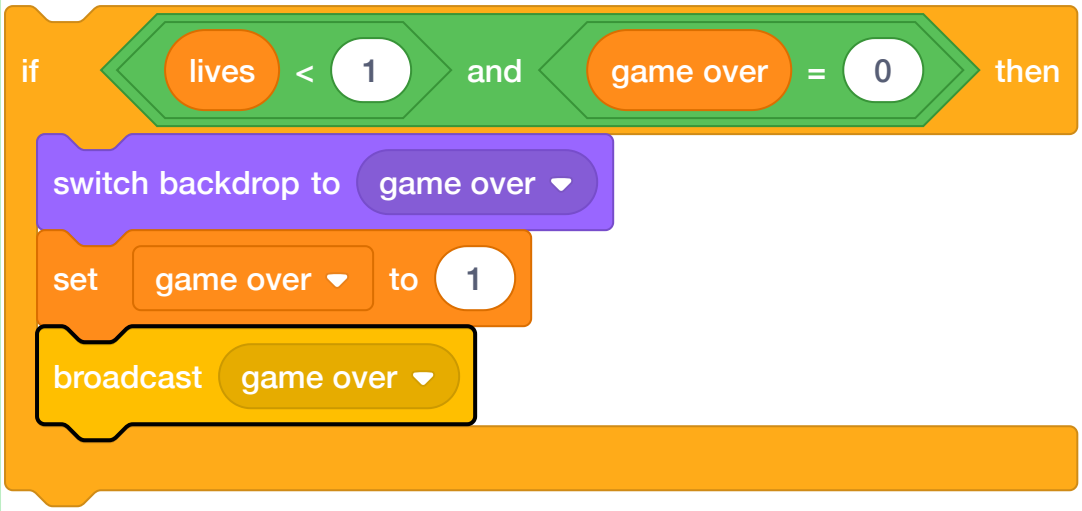


Neil loses when he runs out of lives. On the **Stage**, add a new script — much like your winning one — using a `forever` loop and an `if ... then` block.

When `lives` is less than `1` and `game over` is `0`, switch to your `game over` backdrop and set `game over` to `1`.

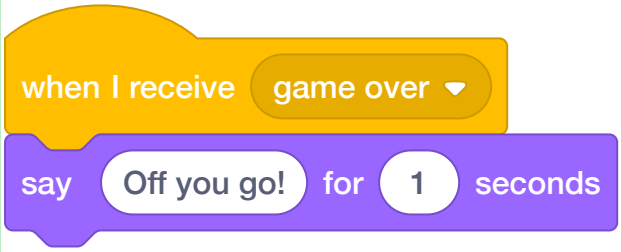


Make a new `broadcast` message called `game over`. Add a `broadcast` block to your new script, so the other sprites know the game has ended.



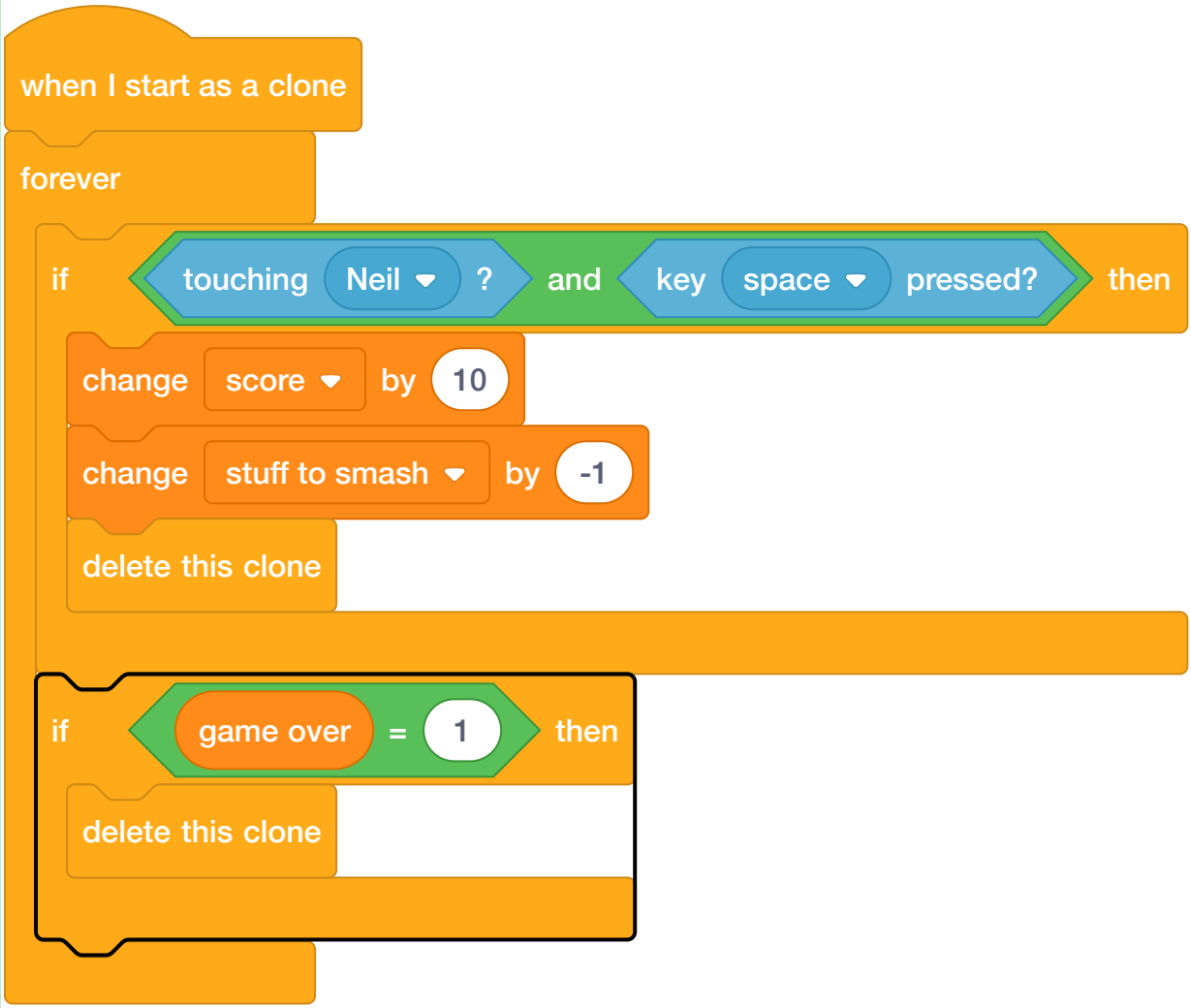
Click on the **Ranger** sprite. When the game is over, the ranger sends Neil back to the sea.

Add a `when I receive` block for the `game over` message, and use a `say () for () seconds` to make the ranger say `Off you go!`.



When the game ends, all the stop sign clones are still lying around town. Get them to tidy themselves up.

Click on the **Sign** sprite. In the clone's `forever` loop, add an `if ... then` block that checks whether `game over` is `1`. If it is, `delete this clone`.



Click on the green flag and let the ranger catch Neil three times. The **game over** backdrop appears, the ranger sends Neil back to the sea, and the leftover stop signs disappear.

Step 9 Add cars and barriers

In this step, you'll add cars and barriers for Neil to smash too. Instead of coding them from the beginning, you'll reuse the code you wrote for the stop signs.

Click on the **Sign** sprite. Drag each of its two scripts onto the **Car** sprite in the Sprite list, then onto the **Barrier** sprite, to copy the code across.

Both sprites already have the `create clone of` and `touching` blocks pointing at the right things, so the code works straight away.

Click on the **Car** sprite. In the `repeat` loop, change the number to decide how many cars appear around town. It's up to you!

Then do the same on the **Barrier** sprite.

Now decide how many points each one is worth. On both the **Car** and **Barrier** sprites, change the number in the `change score by` block.

Maybe some things are trickier to reach and should be worth more?



You can also choose where each type of object appears. In the `go to x: y:` block, change the `pick random` numbers to set the area they spawn in — for example, keeping the cars low down so they stay on the road.



Click on the green flag. The town is now full of signs, cars, and barriers for Neil to smash — each worth the points you chose, in the places you picked.

Step 10 Give Neil a voice

At the moment, the player can just hold down the space bar to smash everything at once. In this step, you'll add a short delay — and give Neil something to say each time he smashes something.

Make a new `list` called `neil's words`, and choose **For all sprites**.

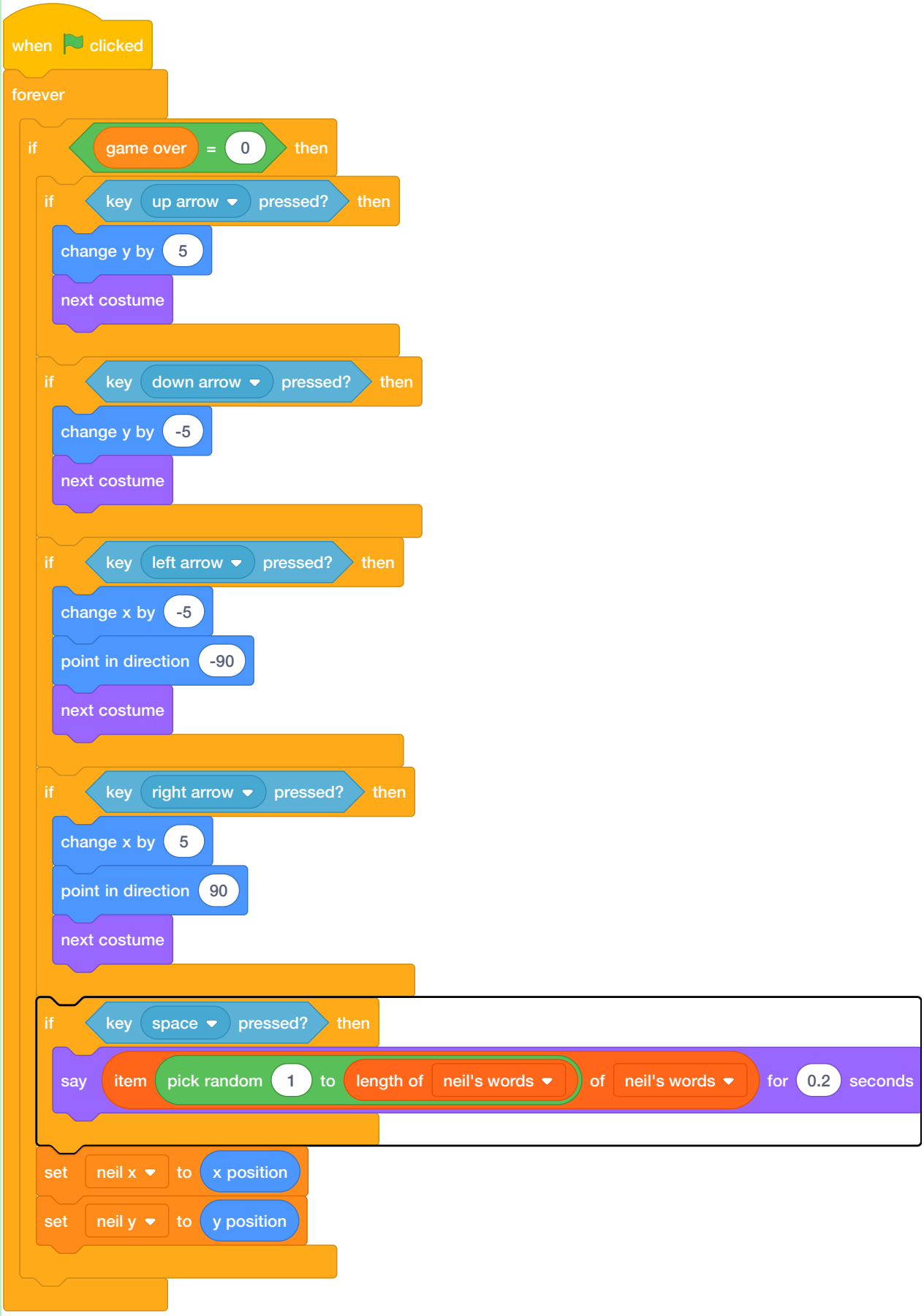
Using the list on the Stage, click on the **+** to add a few things for Neil to shout when he smashes something. You can add as many as you like, and they can be anything you want!

Once you've added all your words, untick the list's checkbox to hide it from the player.



Click on the **Neil** sprite and find his movement script.

With the rest of the movement code, inside the `if ... then` block, add another `if ... then` block that checks for the `space` key. Use a `say () for () seconds` block to make Neil say a random item from your list for `0.2` seconds when the `Space` key is pressed.



Using `length of` means Neil will always pick from however many things you added to the list.

Click on the green flag and smash away. Neil shouts something each time, and holding down the space bar no longer clears the whole town in an instant.

Challenge!

Make it your own

You've finished the game — now make it your own! Here are some ideas to take Neil the seal further. You don't have to try them all; pick the ones that sound the most fun.

- Can you add more levels, each with different things for Neil to smash?
- Can you turn the ranger into clones, so more than one ranger chases Neil at once?
- Can you give each level its own backdrop, so the town changes as you play?
- Can you change how fast Neil or the rangers move, to make the game easier or harder?
- Can you add a patch of sea where Neil can hide, so the ranger can't reach him?

Save your project

Published by (https://www.raspberrypi.org) under a (https://creativecommons.org/licenses/by-sa/4.0/).
(https://github.com/RaspberryPiLearning/neil-the-seal)

Published by Raspberry Pi Foundation (https://www.raspberrypi.org) under a Creative Commons license (https://creativecommons.org/licenses/by-sa/4.0/).
View project & license on GitHub (https://github.com/RaspberryPiLearning/neil-the-seal)

Accessibility (https://www.raspberrypi.org/accessibility/)

Cookies Policy (https://www.raspberrypi.org/cookies/)

Privacy Policy (https://www.raspberrypi.org/privacy/)

Translate for us (/en/projects/translating-for-raspberry-pi)