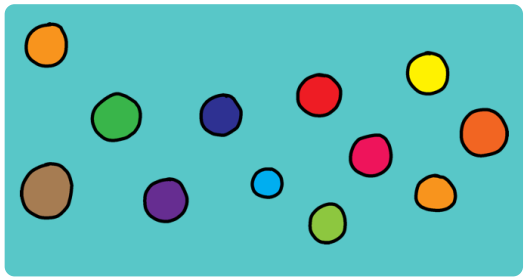
Projects

Catch the dots

Make a dot-catching game

Scratch



Step 1 Introduction

In this project you'll learn how to create a game in which the player has to match up coloured dots with the correct colour of the controller wheel.

What you will learn

- How to choose random items from a list
- How to use variables to track speed, lives, and the player's score

What you will need

Hardware

- A computer capable of running Scratch 2.0

Software

- Scratch 2.0 **offline** (http://rpf.io/scratchoff)

Downloads

- **Offline Scratch 2 resources** (http://rpf.io/p/en/catch-the-dots-scratch2-go)

Additional notes for educators

You can find **the completed project here** (http://rpf.io/p/en/catch-the-dots-scratch2-get).

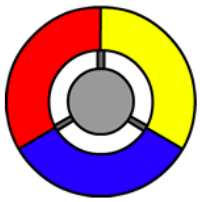
Step 2 Create a controller

Start by creating a controller that the player will use to collect dots.

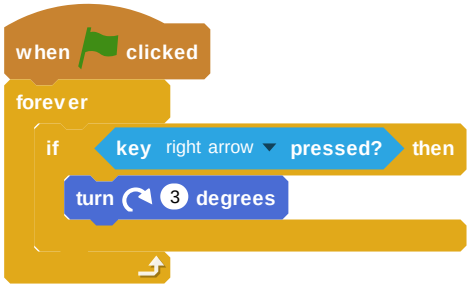
Open the 'Catch the dots' Scratch starter project from **rpf.io/p/en/catch-the-dots-scratch2-go** (http://rpf.io/p/en/catch-the-dots-scratch2-go), and then open it in the Scratch offline editor.

If you need to download and install the Scratch offline editor, you can find it at **rpf.io/scratchoff** (http://rpf.io/scratchoff).

You should see a controller sprite:



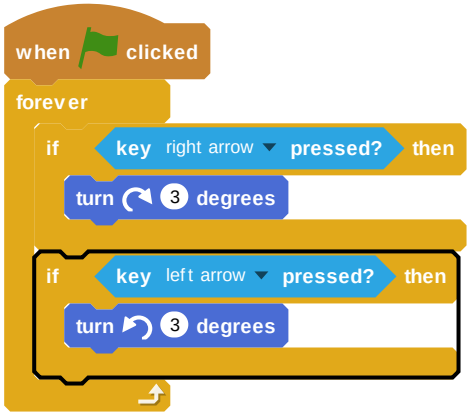
Add some code to the controller sprite to make the sprite turn right if the player presses the right arrow key:



Test your code. The controller should spin to the right when you press the right arrow key.

Add code to the controller sprite to make the sprite turn left if the player presses the left arrow key.

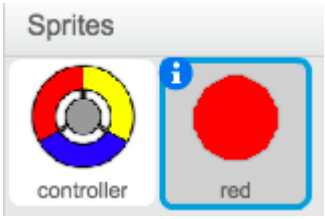
Here is what your code should look like:



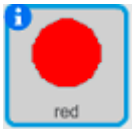
Step 3 Gain points or lose lives

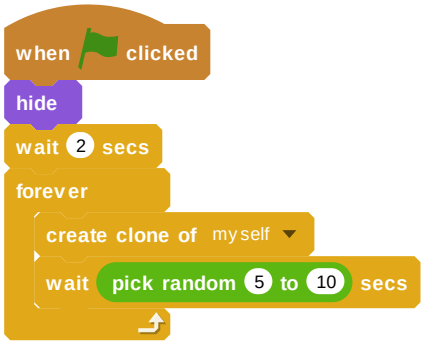
Now you're going to add some dots that the player needs to collect.

Create a new sprite called 'red'. This sprite should be a small red dot.



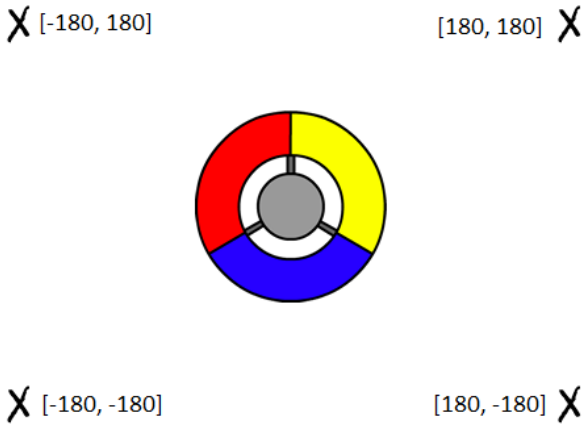
Add this script to your 'red' sprite to create a new clone of the sprite every few seconds:



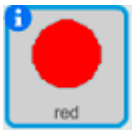


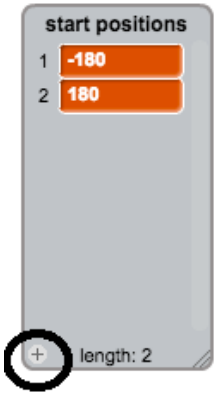
If you click the green flag now, it looks like nothing is happening. This is because all of the cloned sprites are hidden, and they appear in the same place.

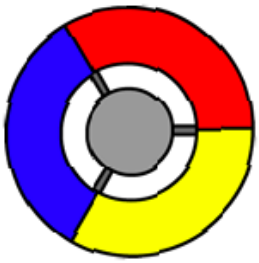
You are going to add code to make each new clone appear in one of the four corners of the Stage.



Create a new list called `start positions`, click the list's `(+)` icon to add the values `-180` and `180`.



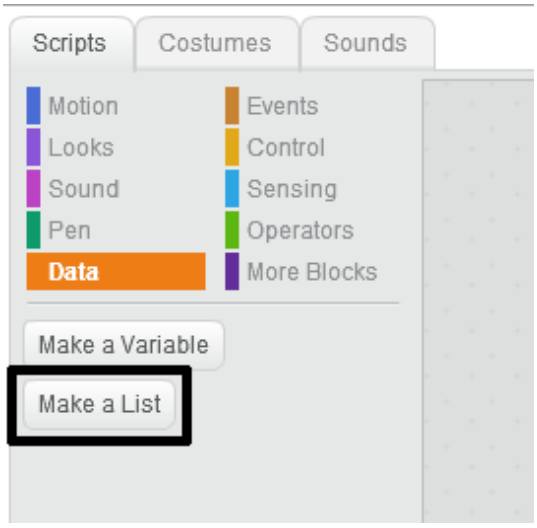




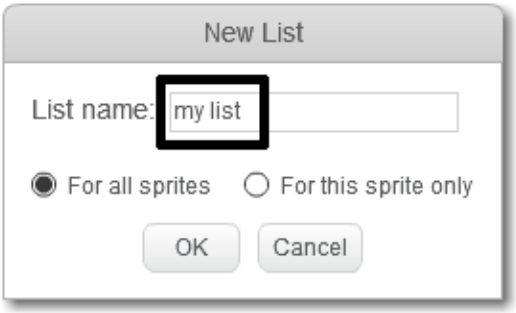
Make a list

Make a list

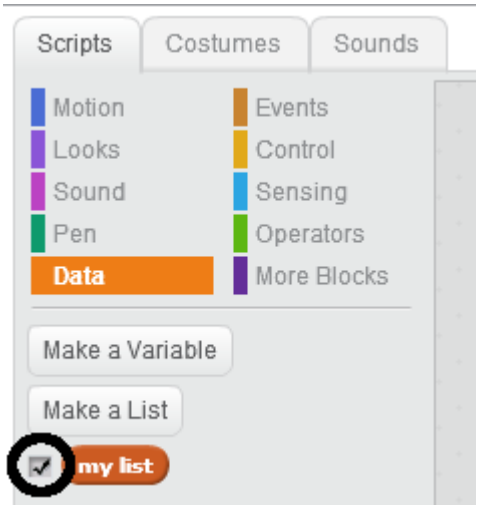
- Click on **Data** in the Scripts tab, then click on **Make a List**.



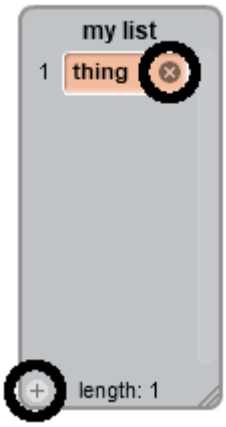
- Type in the name of your list. You can choose whether you would like your list to be available to all sprites, or to only a specific sprite. Press **OK**.



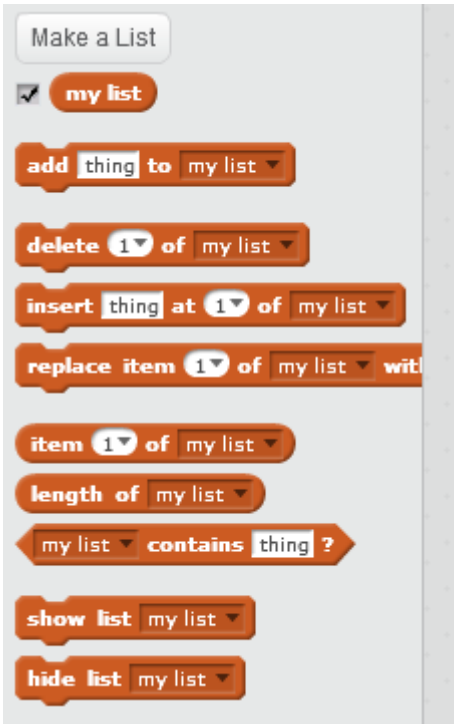
- Once you have created the list, it will be displayed on the stage, or you can untick the list in the Scripts tab to hide it.



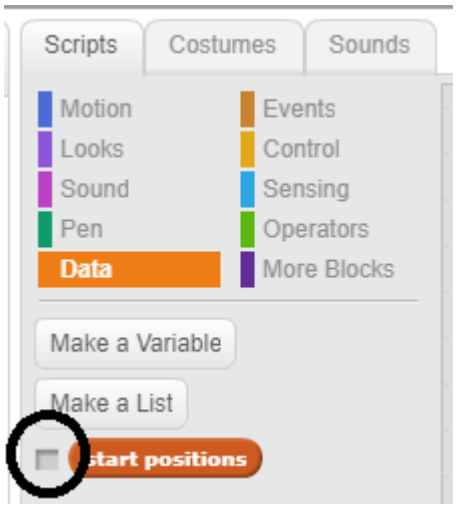
- Click the + at the bottom of the list to add items, and click the cross next to an item to delete it.



- New blocks will appear and allow you to use your new list in your project.

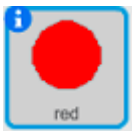


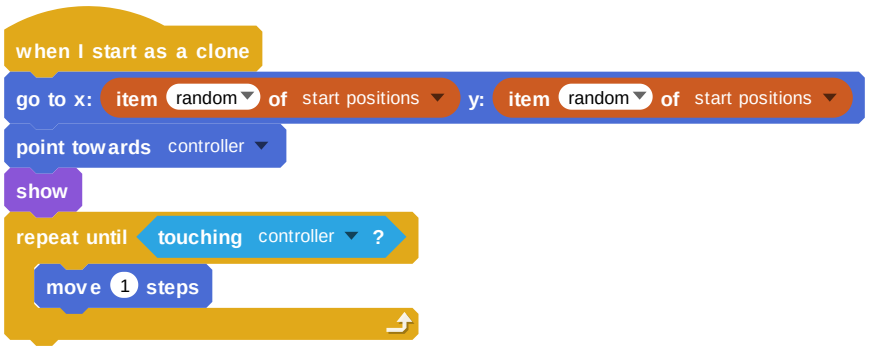
Then you can hide the list by unselecting this box:



Notice that the coordinate for each corner of the Stage is a combination of `180` and `-180`. This means you can use the list to pick a corner of the Stage at random.

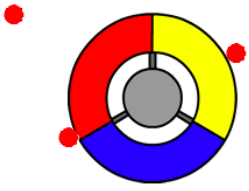
Add this code to the 'dot' sprite to make each new sprite clone appear in a random corner and then slowly move towards the controller sprite.





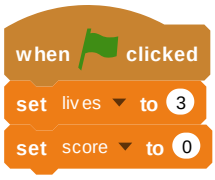
This new code chooses either `-180` or `180` for the x and y positions, meaning that each 'dot' sprite clone starts in a corner of the Stage.

Test your project. You should see red dots appear in the corners of the Stage and move slowly towards the controller.



Create two new variables called `lives` and `score`.

Add code to your Stage to set the `lives` variable to `3` and the `score` to `0` at the start of the game.



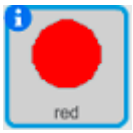
Add this code to the end of the Stage's script to make the game end when the player loses the last of the lives:

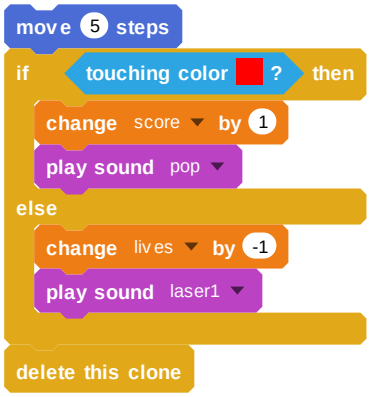
The player should win points for catching dots, and should lose lives for failing to catch dots. A dot can only be caught by matching the colour of the controller to the colour of the dot.

Go back to the 'red' dot sprite's Scripts area to add some code blocks to the end of the sprite's `when I start as a clone` script.

First, make the dot clone `move 5 steps` so that it overlaps the controller.

Then add code to either add `1` to `score` if the colour of the dot clone matches the colour of the controller when they touch, or to take `1` away from `lives` if their colours don't match.





Test your game to make sure that:

1. You lose a life if you don't match a dot with the correct colour
2. You score a point if you match a dot correctly

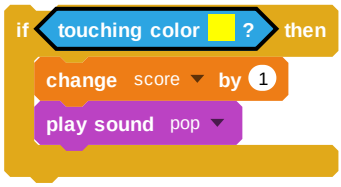
Step 4 More dots

Duplicate your 'red' dot sprite twice, and name the two new sprites 'yellow' and 'blue'.

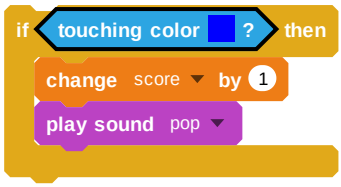
Change the costume of each new sprite so it is the correct colour: the 'yellow' sprite should be yellow, and the 'blue' sprite should be blue.

Change the code of each sprite so that the player has to match dot clone to the correct colour on the controller to score points.

This is how you need to change the code for the yellow sprite:

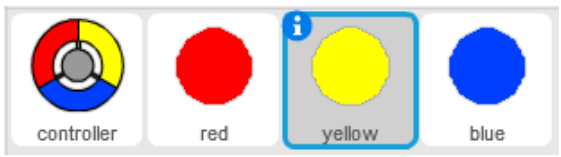


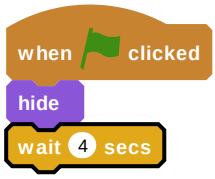
This is how you need to change the code for the blue sprite:



If you play the game now, you can see that the dots sometimes get created one top of each other.

Add code to the 'yellow' dot sprite so that it waits 4 seconds after the flag is clicked before appearing.





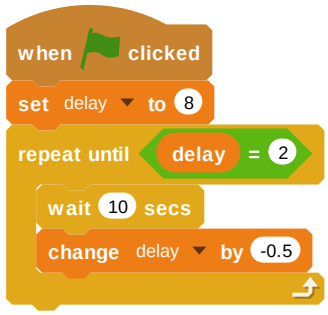
Then add code to the 'blue' dot sprite so that it waits 6 seconds after the flag is clicked before appearing.

Step 5 Increase the difficulty

Now you're going to make the game more difficult the longer the player plays it. You will do this by making the dots appear faster and faster over time.

Create a new `variable` called 'delay'.

Go to the Stage's Scripts area and create a new script that sets the `delay` variable to `8` and then slowly reduces the value of `delay` while the game runs.



Notice that this code is very similar to the code you would use to create a countdown timer!

Next, use the `delay` variable in the code scripts of the 'red', 'yellow', and 'blue' sprites.

Remove the code block that makes the game wait a random number of seconds between making the dot sprite clones. Replace the block you've removed with your new `delay` variable:

Do this for all three dot sprites.

Test the game, and check whether the dots begin to appear more quickly as the game goes on.

- Does this work for all three coloured dots?
- Can you see that the value of the `delay` variable decreases?

Step 6 Challenge: faster dots

Can you improve your game by adding a `speed` variable and using this new variable to make the dot change their speed over time? The clones should start by moving one step at a time, and then steadily get faster and faster.

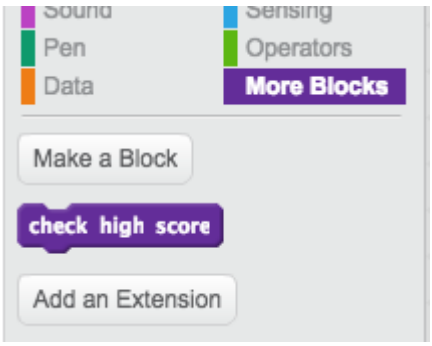
The code you need for this is very similar to the code in which you've used the `delay` variable.

Step 7 High score

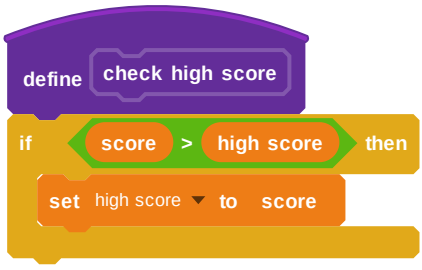
You're going to save the game's high score, so that players can see how well they are doing.

Create a new variable called `high score`.

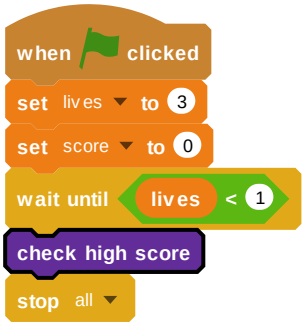
Select the Stage, and create a new custom block called `check high score`.



Add code to your custom block so that the block checks if the current value of `score` in larger than the value of the `high score` variable, and then stores the value of `score` as the new value of `high score`.



Add your new custom block to the Stage script before the end of the script.



Play your game twice to check whether your score gets correctly saved as the `high score`.

Step 8 Challenge: improve your game

Can you think of ways to improve your game? For example, you could create special dots that:

- Double your score
- Slow down the dots
- Hide all the other dots on the screen

Can you add a menu with buttons to your game? You could add an screen with instructions or a separate screen for showing the high score.

Step 9 What next?

Try the **Clone wars** (https://projects.raspberrypi.org/en/projects/clone-wars-scratch2) project to make a game in which you have to save the Earth from space monsters. In that project, you will be able to use what you have learned about cloning sprites and adding a score!