

Catch the Dots



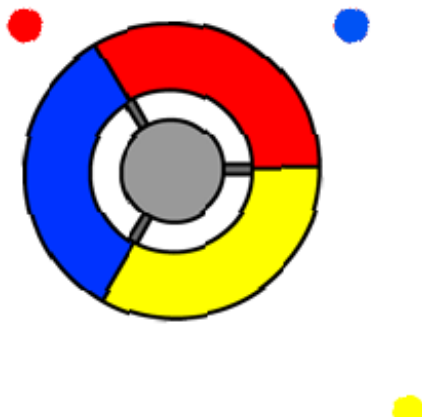
Introduction

In this project you'll learn how to create a game, in which you have to match up coloured dots with the correct part of the controller.

score 6

high score 0

lives 3



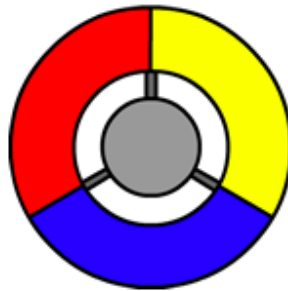
Step 1: Creating a controller

Let's start by creating a controller, that will be used to collect dots.

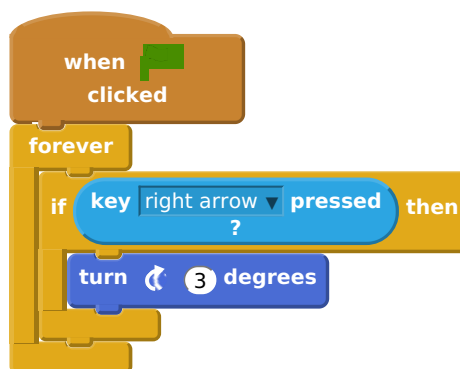
✓ Activity Checklist

- ☐ Open the 'Catch the Dots' Scratch project online at jump.to/cc/dots-go or download from jump.to/cc/dots-get and then open if you are using the offline editor.

You should see a controller sprite:



- ☐ Turn your controller to the right when the right arrow key is pressed:



- ☐ Test out your controller – it should spin to the right.

Save your project

Challenge: Spinning left

Can you make your controller spin to the left when the left arrow key is pressed?

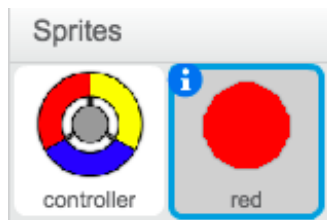
Save your project

Step 2: Collecting dots

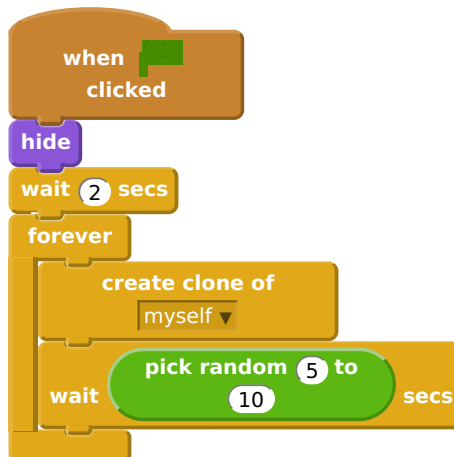
Let's add some dots for the player to collect with their controller.

✓ Activity Checklist

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



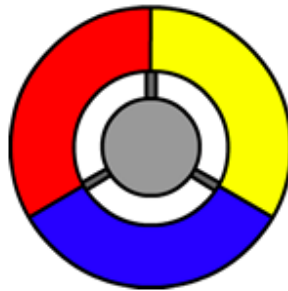
- ☐ Add this script to your 'red' dot sprite, to create a new dot clone every few seconds:



- ☐ When each clone is created, you want it to appear in one of the 4 corners of the stage.

X

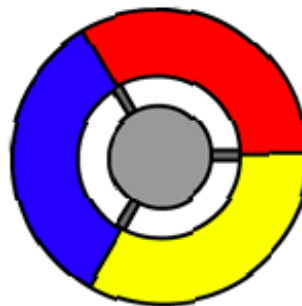
X



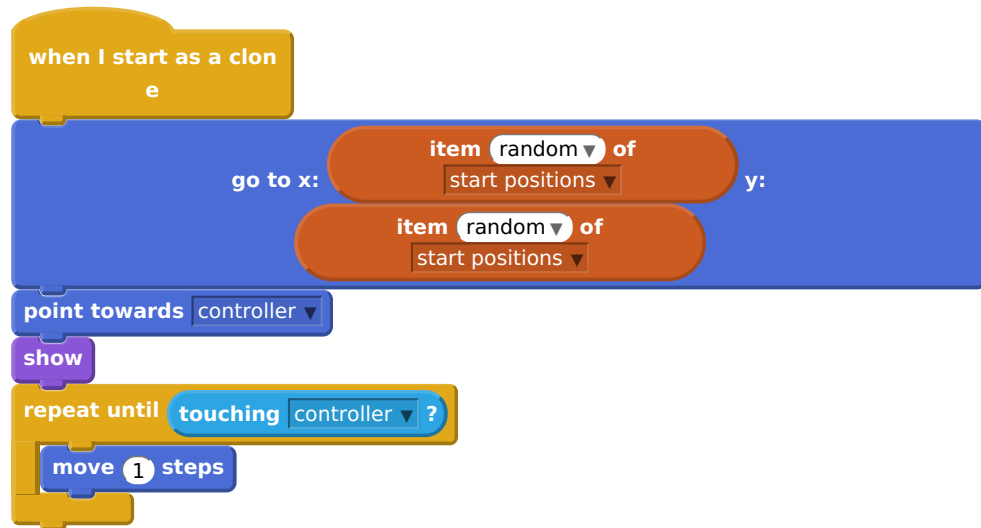
X

X

To do this, first create a new **list** called **start positions** and click the (+) to add in the values **-180** and **180**.

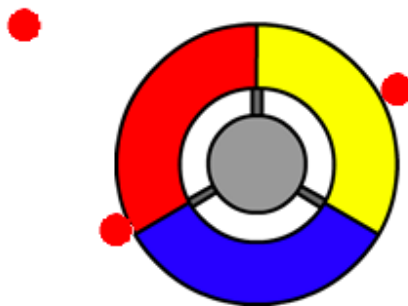


You can use these 2 list items to pick a random corner of the stage. Add this code to the 'dot' sprite, so that each new clone moves to a random corner and then slowly moves towards the controller.



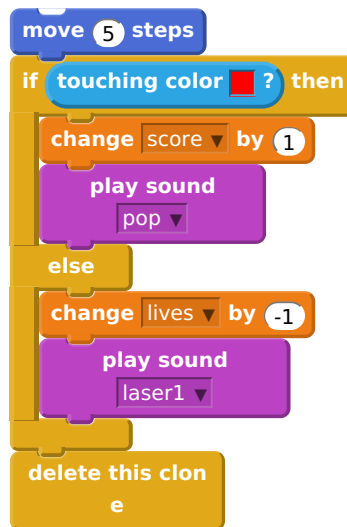
The code above chooses either `-180` or `180` for the x *and* y positions, meaning that each clone starts in one corner of the stage.

- ☐ Test your project. You should see lots of red dots appear in each corner of the screen, and move slowly towards the controller.



- ☐ Create 2 new variables called `lives` and `score`.
- ☐ Add code to your stage to set the `lives` to 3 and the `score` to 0 at the start of the game.
- ☐ You need to add code to the end of your red dot's `when I start as a clone` code, so that either 1 is added to the player's `score` if the colours match,

or 1 is taken from the player's **lives** if the colours don't match.



- ☐ Add this code to the end of your stage's script, so that the game ends when the player loses all of their lives:



- ☐ Test your game to make sure this code works as expected.

Save your project

Challenge: More dots

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



Edit these sprites (including their code), so that each coloured dot has to match the correct colour on the controller.
Remember to test your project, making sure you gain points and lose lives at the right times, and that your game isn't too easy or too hard!



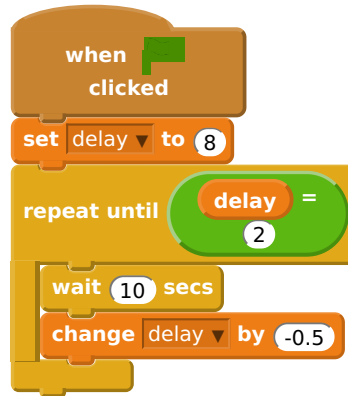
Save your project

Step 3: Increasing the difficulty

Let's make the game get more difficult the longer the player survives, by slowly reducing the delay between dots appearing.

✓ Activity Checklist

- ☐ Create a new variable called **delay**.
- ☐ On your stage, create a new script that sets the delay to a high number, and then slowly reduces the delay time.



Notice that this is very similar to how a game timer works!

- Finally, you can use this **delay** variable in your red, yellow and blue dots' scripts. Remove the code that waits a random number of seconds between creating clones, and replace it with your new **delay** variable:



- Test your new **delay** variable, and see whether the delay between dots reduces slowly. Does this work for all 3 coloured dots? Can you see the value of the **delay** variable reducing?

Save your project

Challenge: Faster moving dots

Can you improve your game by adding a **speed** variable, so that the dots start off moving 1 step at a time, and steadily get faster and faster? This will work in a very similar way to the **delay** variable used above, and you can use this code to help you.

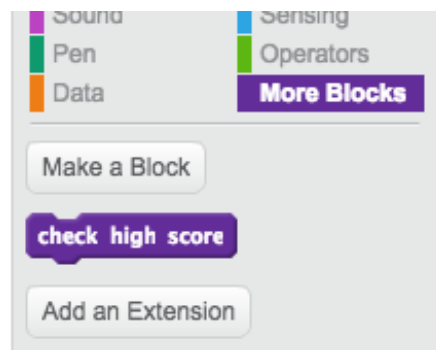
Save your project

Step 4: High score

Let's save the high score, so that players can see how well they're doing.

✓ Activity Checklist

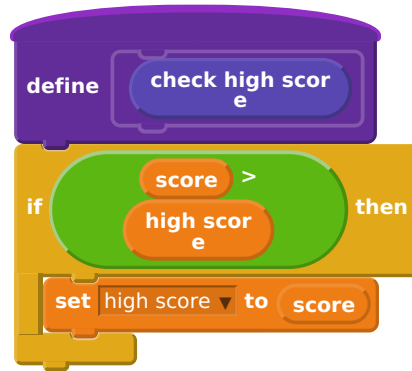
- ☐ Create a new variable called **high score**.
- ☐ Click on your stage, and create a new custom block called **check high score**.



- ☐ Just before the end of the game, add in your new custom block.



- ☐ Add code to your custom block to store the current **score** as the **high score** **if** it's the highest score so far:



- ☐ Test the code you've added. Play your game to check whether the **high score** is updated correctly.

Save your project

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!

Save your project

Challenge: Game menu

Can you add a menu (with buttons) to your game? You could add an instructions screen, or a separate screen for showing

the high score. If you need help with this, the 'Brain Game' project will help you.