



Construct your own racing game

Wanting to add a bit of jeopardy to a simple foot race? This micro:bit racing game allows you to do just that.

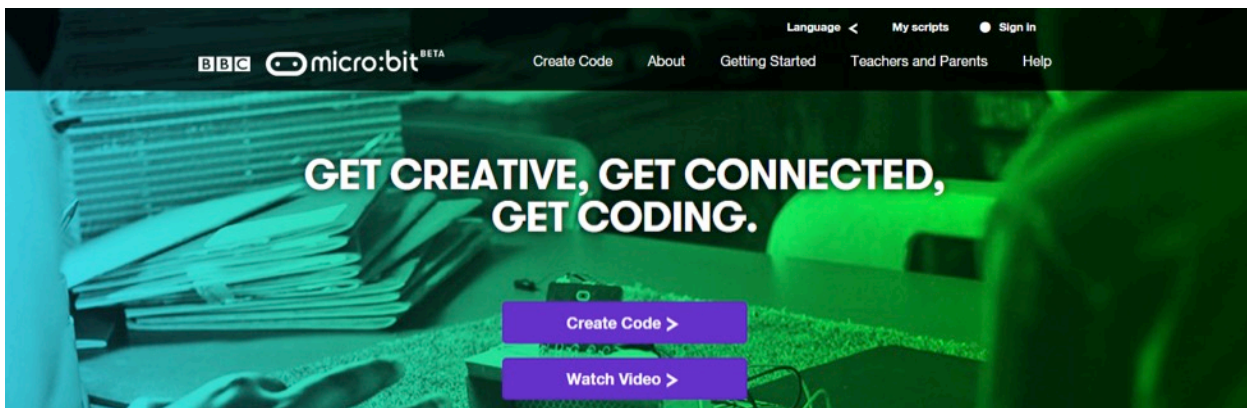
When you've got the game on your micro:bit, simply set a start and finish point that both you and a friend have to get to. Then, shake your micro:bit and take the number of steps that comes up on the LED screen. Who's going to get to the finish point first?

In this exercise, you can learn how to change the code of the racing game and make it your own.

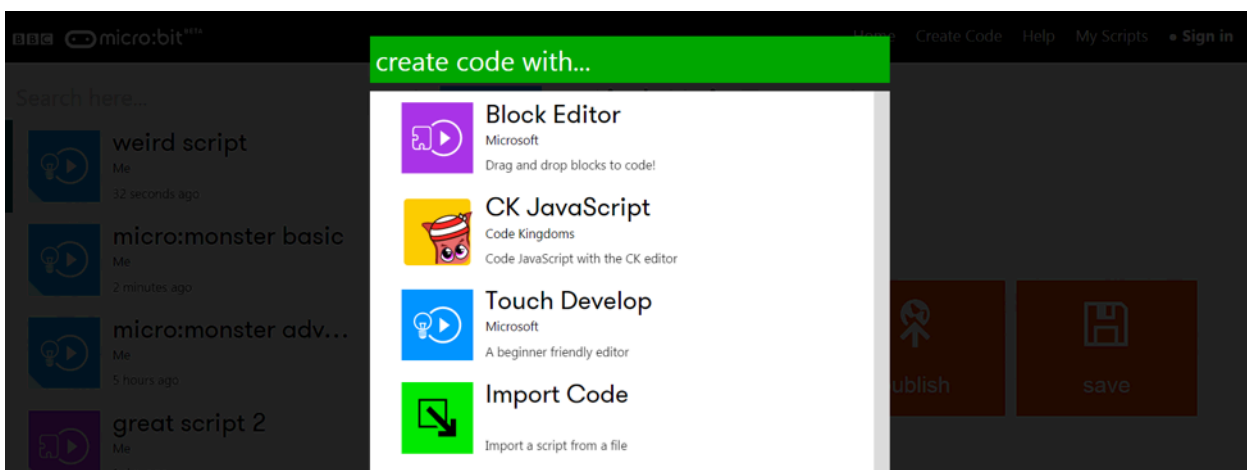
Step 1: Import the code

Download the hex file from our Live Lessons website by clicking on the **racing game hex file** link.

Select **'My scripts'** on the top navigation on the micro:bit website (www.microbit.co.uk), and choose **'Create code'**.



Choose **'Import Code'** and upload the hex file that you've downloaded from the Live Lessons website.



The script for your racing game should now appear in your code window.

Hit 'run' to see it in action on the simulator, or plug in your micro:bit, hit 'compile' and drag your hex file onto your micro:bit to try out your racing game.

Step 2: Understanding the code

my scripts run compile convert help **Race Game**

Basic
LED
Images
Input
Logic
Loops
Math
Game
Music
Pins
Variables
Notes

on shake
do
note: When microbit shaken
note: set result variable to random number up to 7
set result to pick random 0 to 7

What happens when I shake my micro:bit?

This is the block of code that generates the numbers when you shake your micro:bit.

Here we've said that when you shake your micro:bit (**on shake**), we're setting a variable called **result** to a random number from **0** to **7**.

my scripts run compile convert help **Race Game**

Basic
LED
Images
Input
Logic
Loops
Math
Game
Music
Pins
Variables
Notes

on shake
do
note: When microbit shaken
note: set result variable to random number up to 7
set result to pick random 0 to 7

forever
note: show the result on screen
if result = 7
do
show string "Back Four!"
else
show number result

What's displayed on the screen?

This is the block of code that makes the result display on the screen.

Here we're inserted a **forever loop**, which means that the program will keep running and not only run once.

We've introduced a **conditional statement**, which states that IF the result is 7, then the screen shows the string, or message, "Back Four", which tells you to take four steps back. This is a 'forfeit' that comes up randomly.

Otherwise (ELSE), the screen will simply display the number - which tells you how many steps forward you can take.

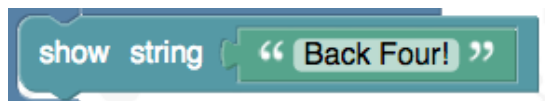
Step 3: Changing the code

You can adapt your racing game by changing the replies that you get when you shake your micro:bit.

To change the number of steps you can take, simply change the number **7** in the block below to any number you like.



You can also change what you have to do when you encounter the 'forfeit'. Simply click within the quotation marks (" ") for the string and type in the message that you'd like showing. This could be anything from "jump up and down" to "turn around".



Test, play and show us what you've done

Now that you've made your very own racing game, click '**run**' to test it on the simulator and '**compile**' to see it working on your micro:bit.

Click '**export**' to save off your code and send it to us at live.lessons@bbc.co.uk. You could see your codes featured on our **micro:bit Live Lesson** in February.