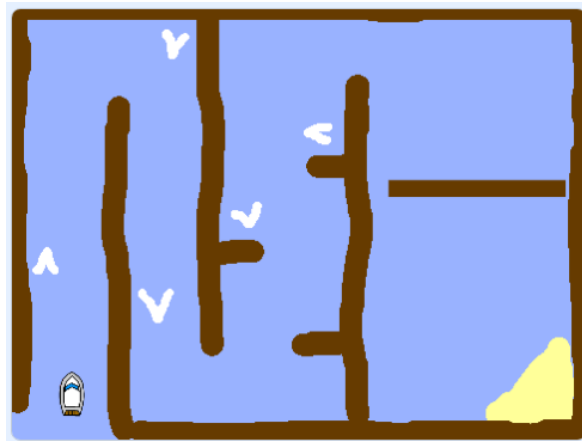 **Projects**

Boat race

Create a boat-racing game in which you have to avoid obstacles

Step 1   Introduction

In this resource, you are going to learn how to make a racing game. The player uses the mouse to navigate a boat to an island without bumping into obstacles.

What you will make

 What you will need

Hardware

A computer capable of running Scratch 3

Software

Scratch 3 (either **online (https://rpf.io/scratchon)** or **offline (https://rpf.io/scratchoff)**)

Downloads

Downloads can be found **here (http://rpf.io/p/en/boat-race-go)**.

What you will learn

- Use operators to compare numbers in Scratch
- Add code to detect when a sprite is touching a colour in Scratch
- Use a variable to record the time in Scratch

Additional information for educators

If you need to print this project, please use the **printer-friendly version (https://projects.raspberrypi.org/en/projects/boat-race/print)**.

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

Step 2   Getting started

Open the starter project.

Online: open the online starter project at **rpf.io/boat-race-starter-on** [(http://rpf.io/boat-race-starter-on)](http://rpf.io/boat-race-starter-on).

If you have a Scratch account you can make a copy by clicking **Remix**.

Offline: download the offline starter project from **rpf.io/p/en/boat-race-go** [(http://rpf.io/p/en/boat-race-go)](http://rpf.io/p/en/boat-race-go), and then open it using the 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)](http://rpf.io/scratchoff).

The project includes a boat sprite, and a race course backdrop with:

- Wood that the boat sprite has to avoid
- A desert island that the boat has to reach

Step 3   Controlling the boat

The player will control the boat sprite with the mouse.

Add code to the boat sprite so that it starts in the bottom left-hand corner pointing up and then follows the mouse pointer.

```
when 🏴 clicked
point in direction  0
go to x:  -190  y:  -150
forever
    point towards  mouse-pointer ▾
    move  1  steps
```

Test your code by clicking the green flag and moving the mouse. Does the boat sprite move towards the mouse pointer?
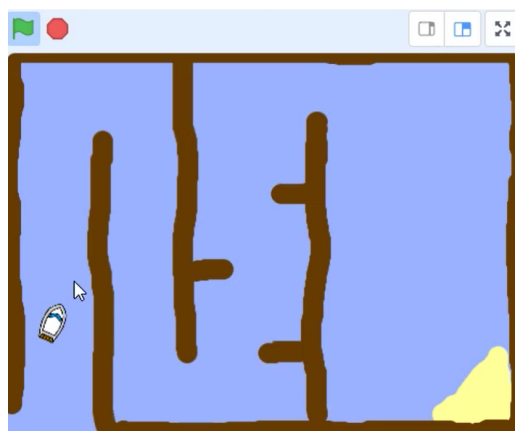
What happens when the boat reaches the mouse pointer? Try it out to see what the problem is. ✓

To stop this from happening, you need to add an 🔶 block to your code, so that the boat sprite only moves if it is more than 5 pixels away from the mouse pointer. ✓

This is what your code should look like:

```
when 🏳 clicked
point in direction 0
go to x: -190 y: -150
forever
    if < distance to mouse-pointer ▼ > 5 > then
        point towards mouse-pointer ▼
        move 1 steps
```

Test your code again to check whether the problem is now fixed. ✓

Step 4   Crashing!

At the moment, the boat sprite can simply sail through the wooden barriers! You're going to fix that now.

You need two costumes for your boat sprite: one normal costume, and one for when the boat crashes. Duplicate your boat sprite's costume, and name one costume 'normal' and the other 'hit'.

Click on your 'hit' costume, and use the **Select** tool to grab pieces of the costume and move and rotate them to make the boat look like it has crashed to pieces.
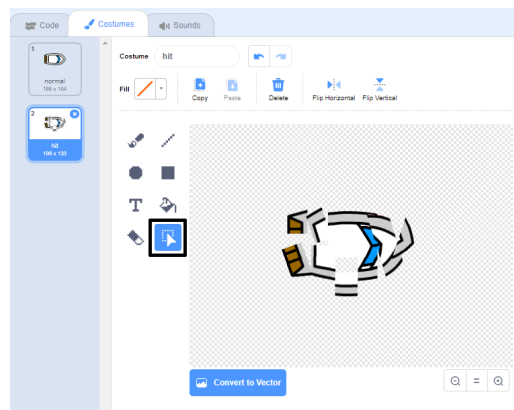
Now add code to your boat so that it crashes and breaks up when it touches any brown wooden barriers. ✅

Here's what your code should look like:

```
when 🏳 clicked
point in direction 0
go to x: -190 y: -150
forever
    if  distance to mouse-pointer ▼ > 5  then
        point towards mouse-pointer ▼
        move 1 steps

    if  touching color ● ?  then
        switch costume to hit ▼
        say Noooooo! for 2 seconds
        switch costume to normal ▼
        point in direction 0
        go to x: -190 y: -150
```

You should also add code to make sure that your boat sprite always starts out looking 'normal'. ✅

Test your code again. If you try to sail the boat through a wooden barrier now, the boat should crash and then move back to its starting position.

Noooooo!

Step 5 Winning!

Now add another **if** statement to your boat sprite's code so that the player wins when they make the boat arrive at the yellow island.

When the boat gets to the island, the game should say 'YEAH!', and then it should end.

Here's what your new code should look like:


```
if    touching color ( ) ? then
    say  YEAH!  for  2  seconds
    stop  all ▼
```

Don't forget that this new code needs to be inside the **forever** loop.

Challenge: sound effects

Can you add sound effects that play when the boat crashes or reaches the island?

You could even add background music!

Step 6   Adding a timer

Now you will add a timer to your game, so that the player has to get to the island as quickly as possible.

Add a new variable called `time` to your Stage.

You can also choose a look for your timer by changing how your new variable is displayed.

Now add code blocks to your Stage so that the timer counts up until the boat reaches the island.

Here's what your new code should look like:


```
when 🏳 clicked
set  time ▾  to  0
forever
    wait  0.1  seconds
    change  time ▾  by  0.1
```

Test out your game and see how quickly you can get the boat to the island!

Step 7 Obstacles and boosters

Right now the game is **far** too easy, so you will add some things to make it more interesting.

First, you'll add some boosters to speed up the boat.

Edit your Stage backdrop by adding in some white booster arrows.
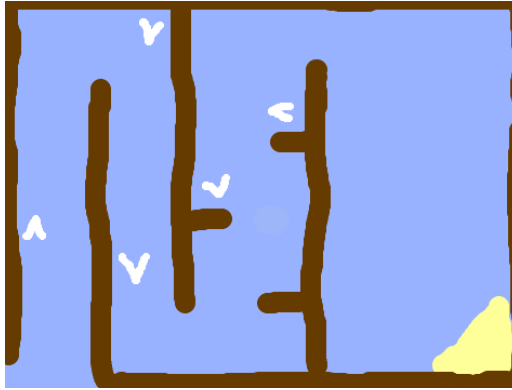
Now add more code blocks to your boat's `forever` loop so that the boat sprite moves three extra steps when it touches a white arrow.
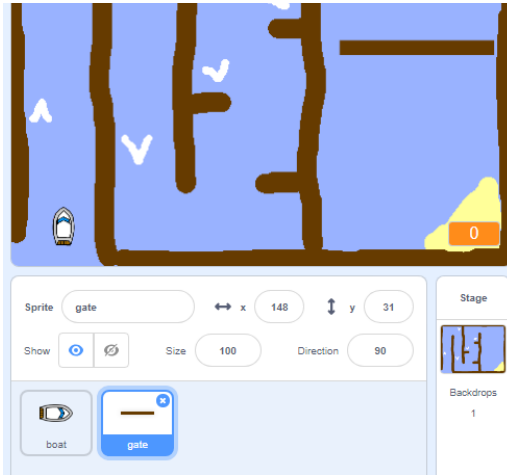
```
if  touching color (  ) ?  then
    move  3  steps
```

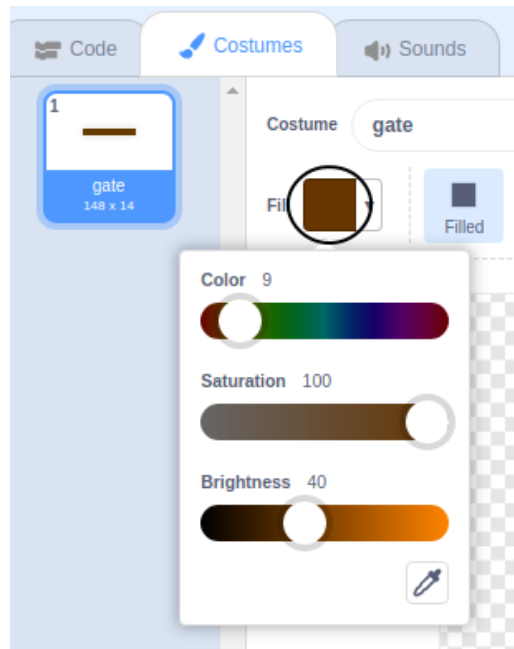Test your game to see whether your new booster arrows speed up the boat.

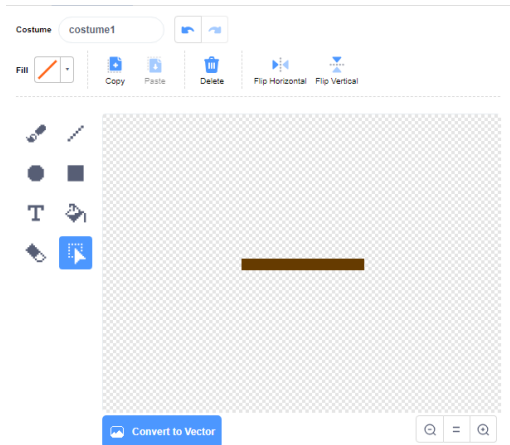Next you'll add a spinning gate that the boat has to avoid.

Add a new sprite that looks like this, and call it 'gate':

Make sure that the colour of the gate sprite is the same as the colour of the wooden barriers.
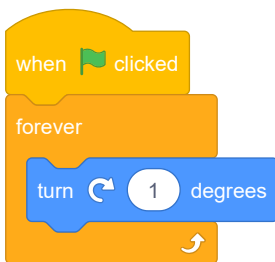
Make sure that the centre of the gate sprite is positioned in the middle.

Add code to your gate sprite to make it spin slowly forever.

Here's what your new code should look like:

Test your game again. You should now have a spinning gate that you need to stir your boat around.

🏆 **Challenge!**

Challenge: improving your game

- Can you add more obstacles to your game? For example, you could add green slime to your backdrop and make changes to the code so that the slime slows the boat down when the player lets them touch.

- You could add a moving obstacle, for example a log or a shark!
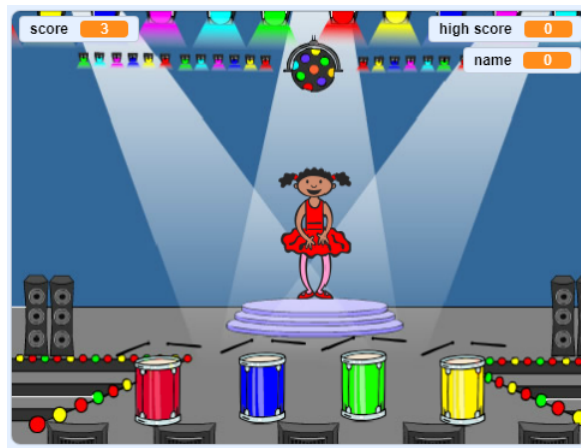

- Can you turn your game into a race between two players? The second player will need to control their boat using the up arrow to move forward and the left and right arrow keys to turn.

- Can you create more levels by adding different backdrops, and can you then allow the player to choose between levels?

Step 8   What next?

Congratulations on completing the 'Boat race' project! Would you like to try something a little more challenging?

You could try out the **Memory (https://projects.raspberrypi.org/en/projects/memory?utm_source=pathway &utm_medium=whatnext&utm_campaign=projects)** project.