

# Boat Race

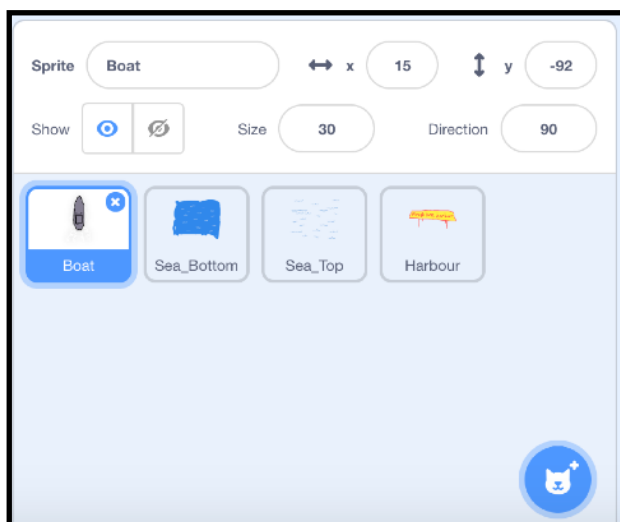
## Setting Up

Access the starter code from here: <https://scratch.mit.edu/projects/364546616/editor/>

Click **Remix** to start

## START

The starter project already has four sprites. You can see them here:



There is a Boat sprite. There are two sprites (Sea\_Top and Sea\_Bottom) that look like water. There is a 'Harbour' sprite that will appear at the end of the game.

**Click the green flag to see what happens.**

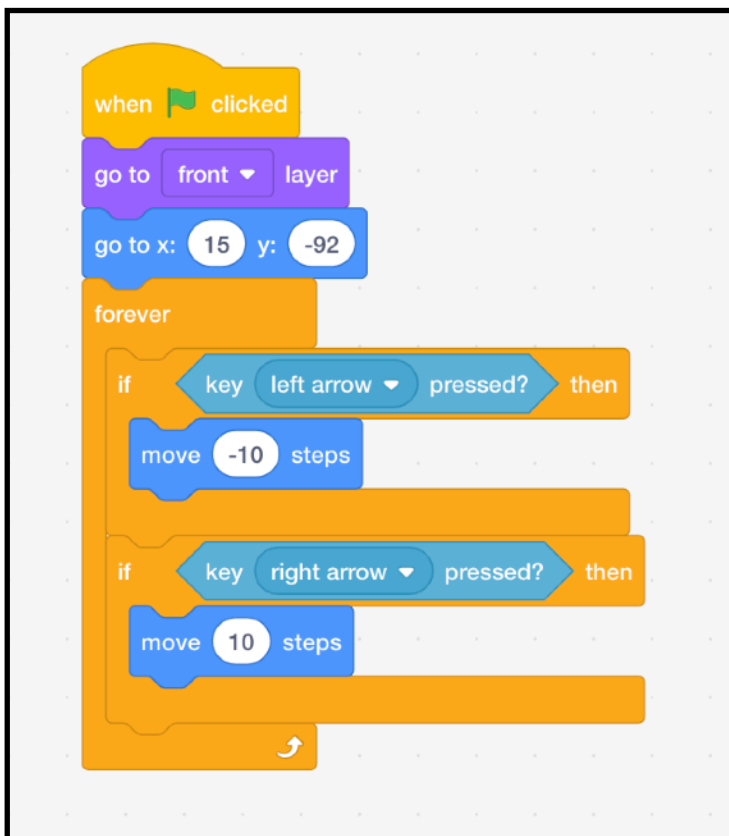
Without making any changes to the code, can you work out how it does this? Look carefully at each squiggle. What are they doing?

This project uses the fact that it's quite easy to write code to make objects move from the top of the screen to the bottom of the screen. We can use this to make it look like the boat is moving through the sea.

### Coding Step 1: Adding code to the boat

First, add these blocks to the boat sprite.

The purple 'go to front layer' block will make sure the boat sits above the water (and not invisible under it).



This makes the boat move left and right.

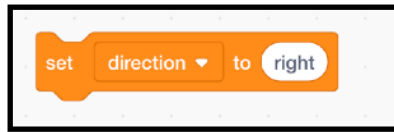
Try it out.

**It's not very realistic!**

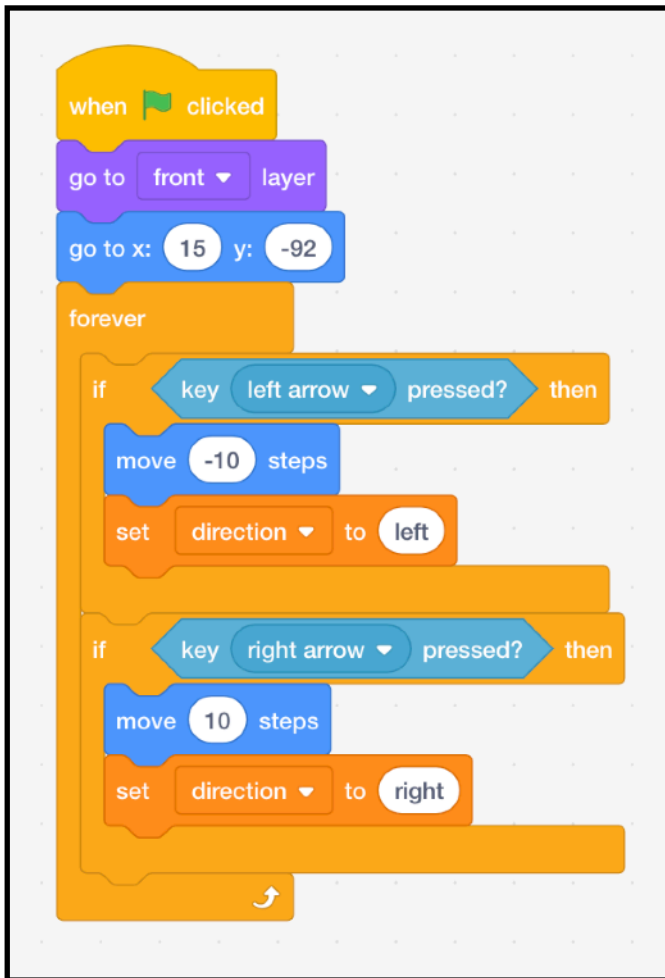
To make it more realistic start by creating a variable called 'direction'.

This is going to store information about the direction the boat is facing: 'left' or 'right'

Add the block



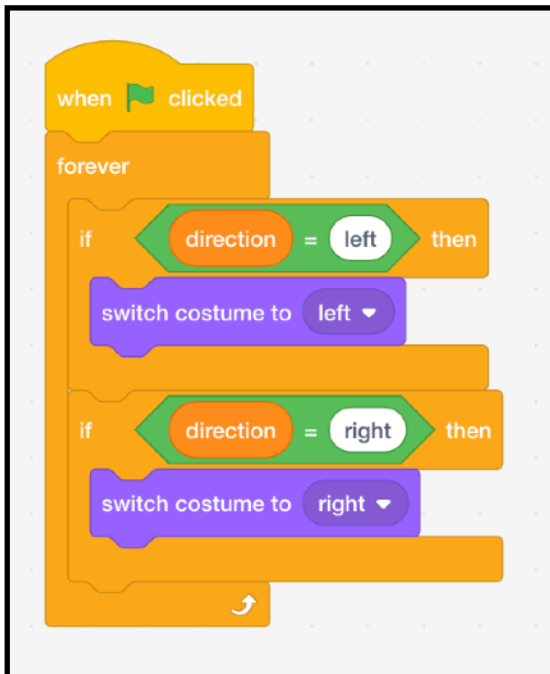
to the code like this:



And don't forget to add one for the 'left' direction.

Next we add these **new** code blocks. (this is a new set of blocks, we're NOT adding more blocks to what we did above)

These will show the boat steering by changing the costume.



```
when green flag clicked
  forever loop
    if direction = left then
      switch costume to left
    if direction = right then
      switch costume to right
```

Run the code again and see what has changed.

It's ok but it's not very responsive. It doesn't straighten up again!

These code blocks do something INTERESTING.

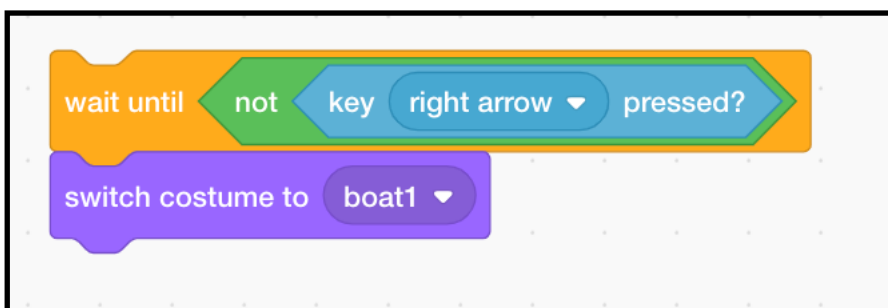
There's a light blue sensing block that reacts to the **right arrow key** being pressed.

But we REVERSE that by using the green '**not**' block.



This combination means it waits until you take your finger **off the key** before changing the costume back to boat1

finger **off the key**



```
wait until not key right arrow pressed?
  switch costume to boat1
```

This is how the code should look after the changes you just made.

```
when clicked
  forever
    if direction = left then
      switch costume to left
      wait until not key left arrow pressed?
      switch costume to boat1
    if direction = right then
      switch costume to right
      wait until not key right arrow pressed?
      switch costume to boat1
```

The image shows a Scratch script starting with a yellow 'when clicked' block. Below it is an orange 'forever' loop block. Inside the loop, there are two conditional 'if' blocks. The first 'if' block checks if 'direction' is 'left'. If true, it switches the costume to 'left', waits until the 'left arrow' key is not pressed, and then switches the costume to 'boat1'. The second 'if' block checks if 'direction' is 'right'. If true, it switches the costume to 'right', waits until the 'right arrow' key is not pressed, and then switches the costume to 'boat1'. The 'forever' loop block has a white arrow at the bottom indicating it repeats indefinitely.

Try the code and see how the boat responds to the keys being pressed.

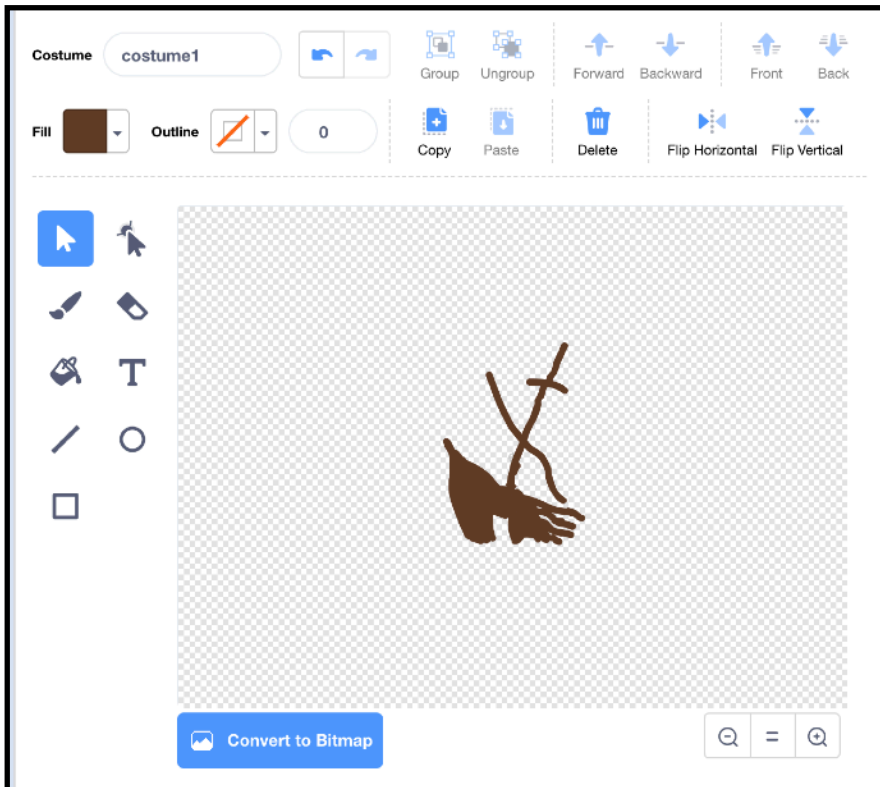
It should straighten up when you take your finger off the key.

## STEP 6 - Obstacles

Next you will make an obstacle for the boat to avoid.

Draw your own obstacle. Keep it simple. Make sure the size makes sense.

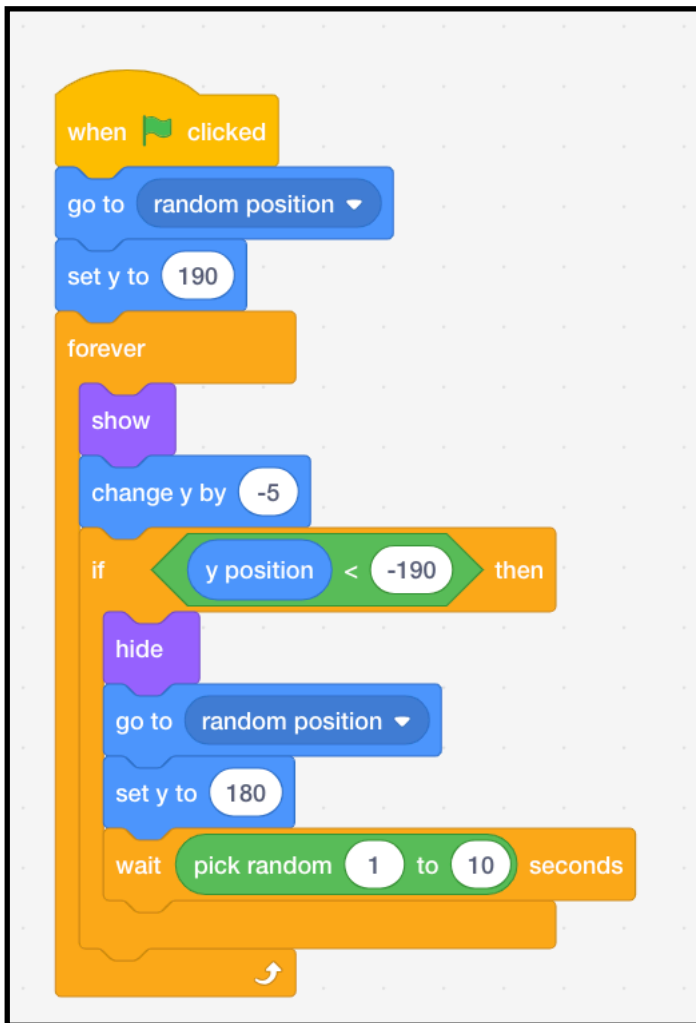
An obstacle could be a rock or a buoy or a shipwreck or a sandbar.  
Here's an example of a wrecked ship:



Make sure the sprite you draw sits over the middle point of the drawing screen; over the little + symbol in the centre of the checked rectangle.

Click on the code tab and we'll add some code to make this obstacle behave in the right way.

Add this code to the obstacle:



Can you work out what it will do before you run the program?

Run the code and check to see if it's working.

**Things to check:**

Does the obstacle start at the top of the screen?

Is it a sensible size?

**Things to play around with:**

Try changing the number of seconds the sprite waits before appearing again.

Is your sprite too big to avoid? Try some other sizes.

## STEP 7 - MORE OBSTACLES:

**Duplicate** this sprite

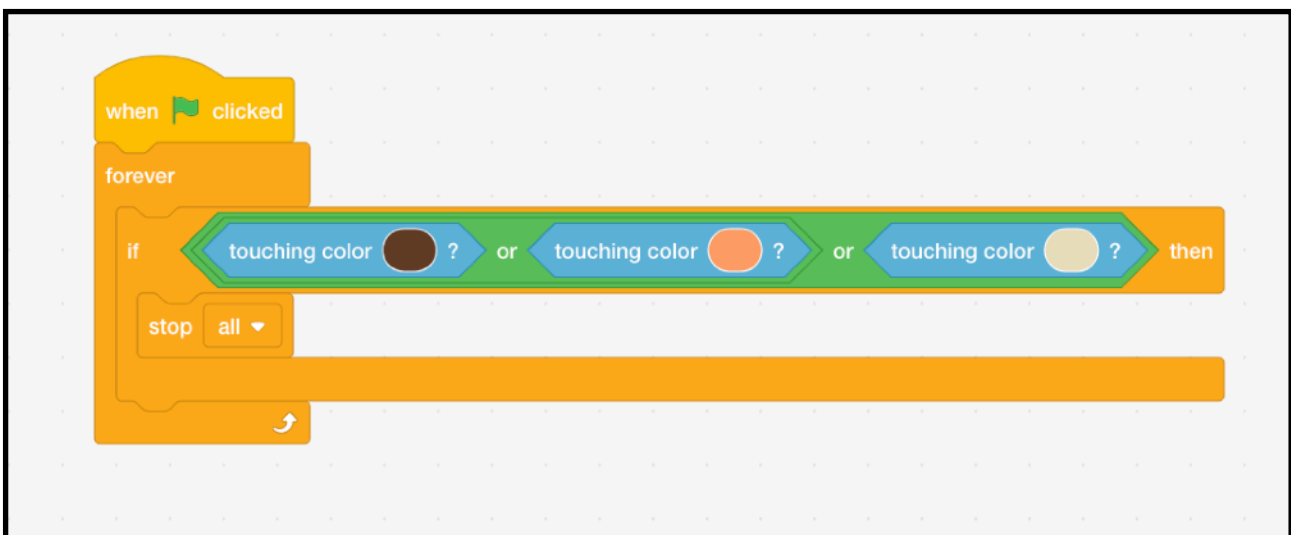
Next, draw a new obstacle in the costume tab (deleting the old costume).

Give it some different values for how long it waits before appearing so they don't all arrive at once!

**STEP 8 - make more obstacles using the instructions above.**

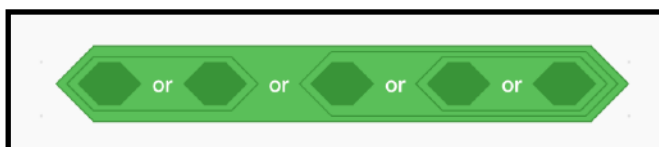
## STEP 9 Last Task!!

THIS CODE GOES ON THE BOAT SPRITE



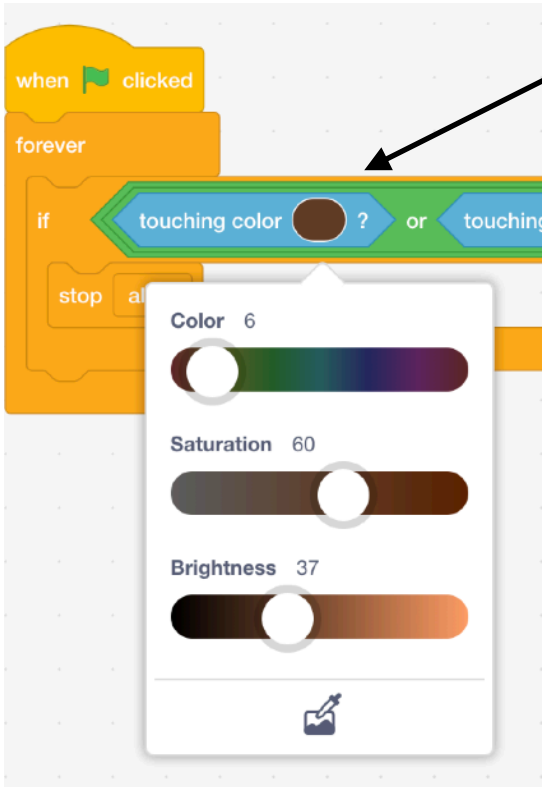
\* You will need one '**touching color**' block for each different obstacle.

String together as many green 'or' blocks as you need



\* You need to change the colour values to the colour of the obstacles you have created. See next page for how to do that.





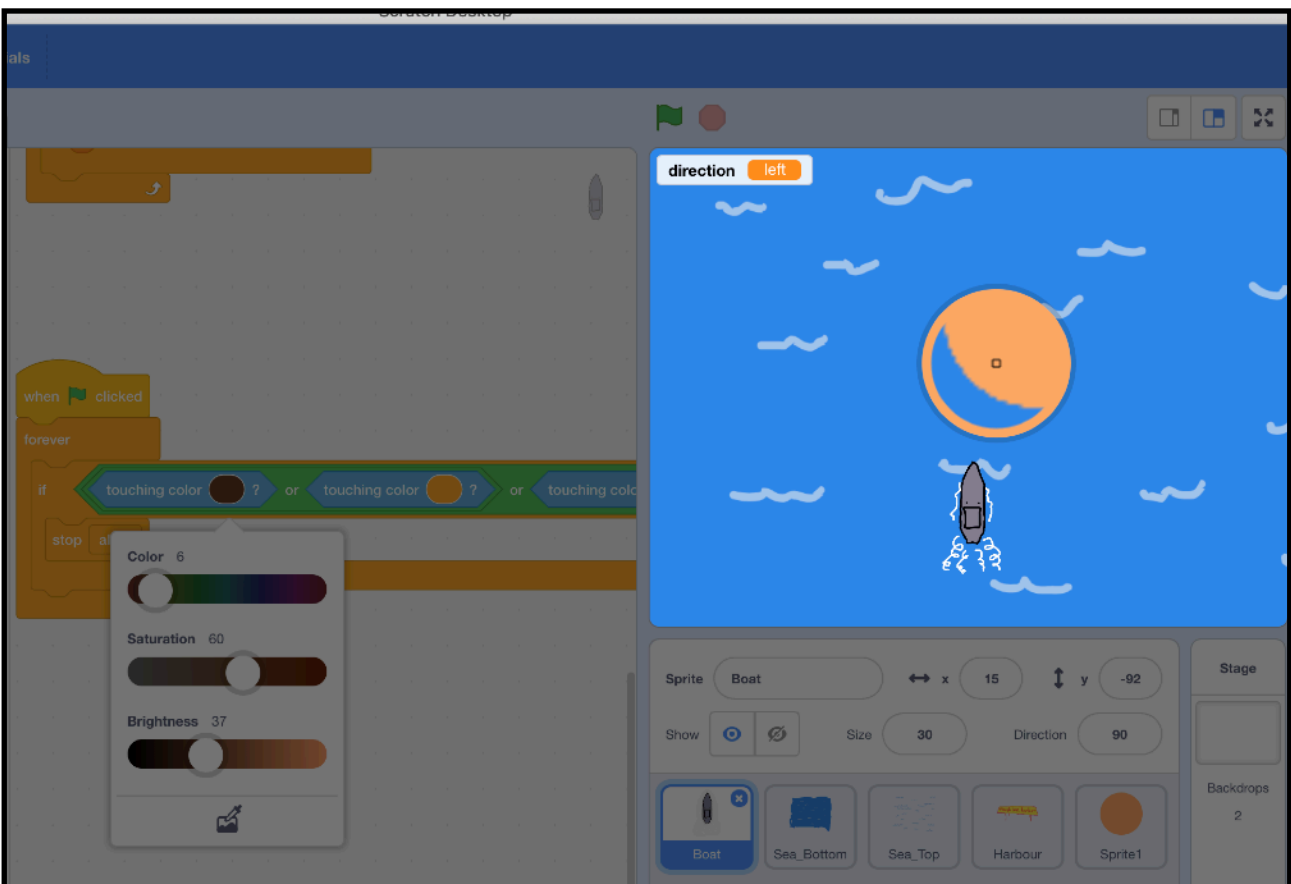
Click on the colour field to open the colour menu.

Then click this button:



It will make a colour picker tool that you can hover over colours on the scratch screen in order to 'pick them up'

**Click the sprite you want to avoid to get its exact colour**



## STEP 10

Run the program by clicking the green flat.

See if you can reach the harbour!

If anything doesn't work, check your code against the instructions.

////////////////////////////////////

Copyright © 2020 Neil Davidson