STEAMENGINEERS

# STEAM ENGINEERS LTD SCRATCH TUTORIAL

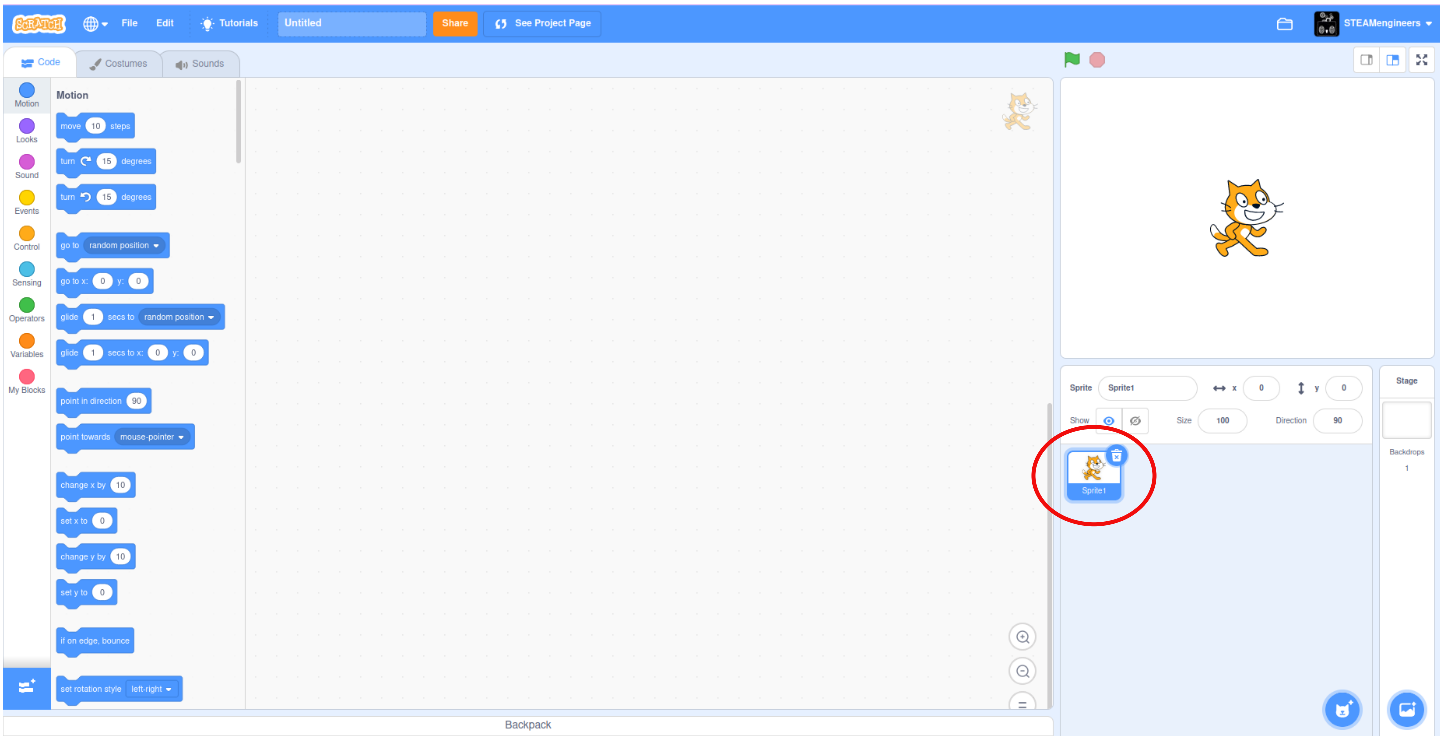
## "BUTTONBOT"

A LITTLE ROBOT YOU CAN TORMENT AND TWIST, JUST TO START GETTING A GRIP ON THE BASICS OF BLOCKCODE

The first thing we need is a sprite. A sprite is a little picture of something, that can move around and do things in your game.

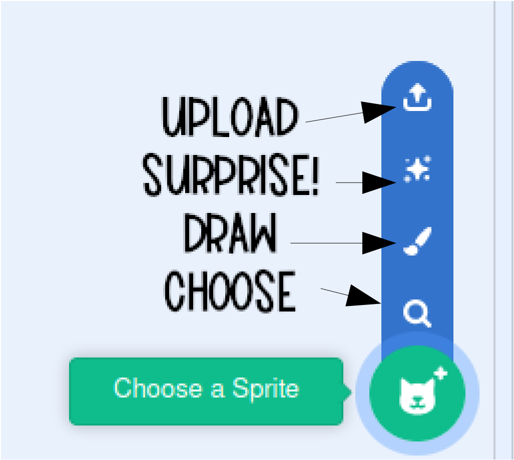
In this example, we've used a little robot Snoof made, but you can pick one of the sprites Scratch lets you pick, or you can even make your own!

When we first start a project in Scratch, there is already a sprite in the project:



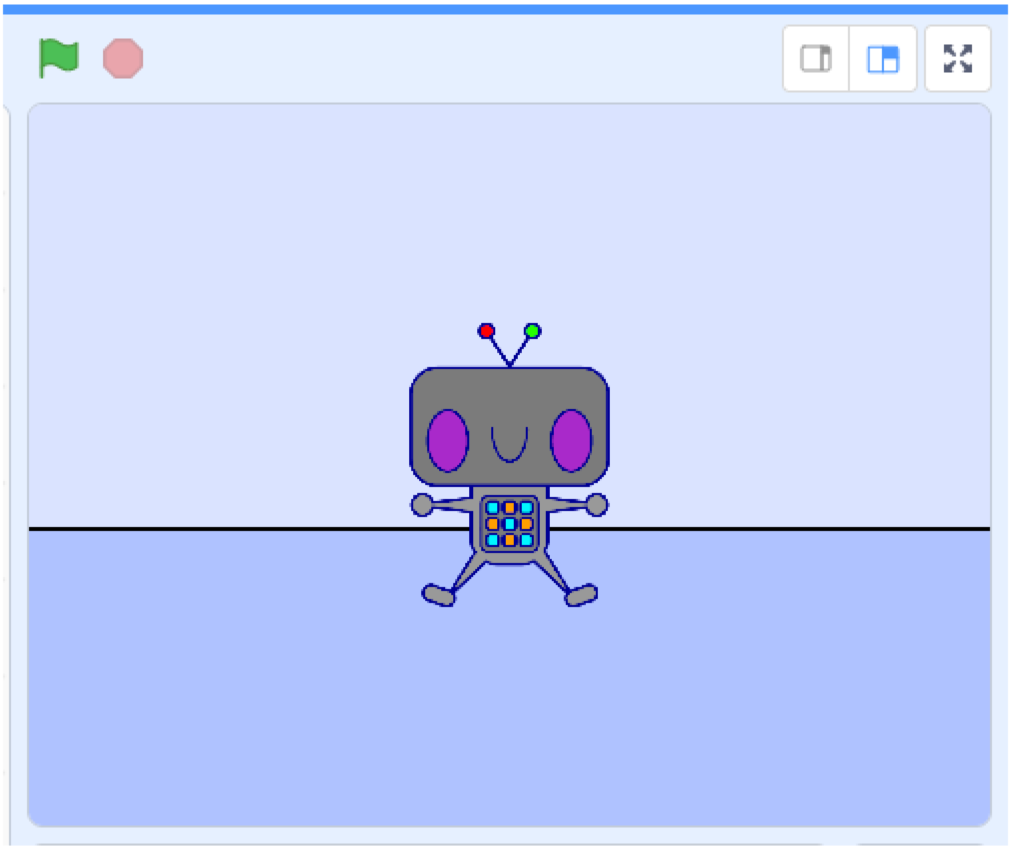
That little orange cat is a sprite. We can delete him if we want, by clicking the little bin on the small picture of him in the red circle.

If we want to pick a different sprite, we can click on the little icon in the bottom right and upload or pick or even make one!



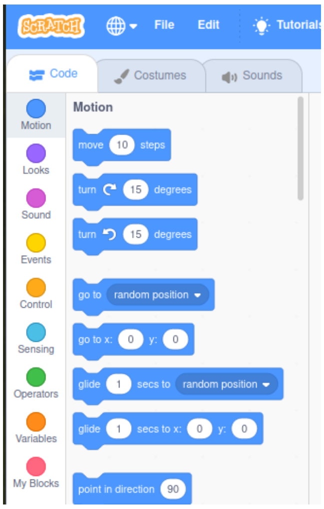
You can do the same thing with the background, using the button just to the right of the sprite button. Again you can pick one from your computer, pick one from scratch, or draw your own. We drew our own here! If your sprite is too big, you can change his size in the sprite window.

So now we have a sprite and a background:



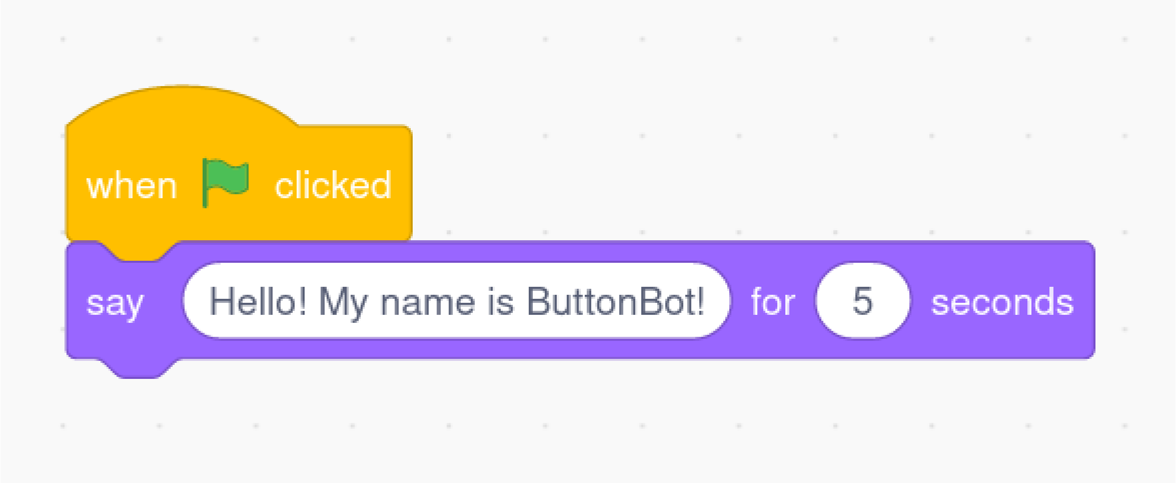
You can see how everything looks in the top right of your screen, and you can test out new bits of code by pressing the green flag, which starts your code running. So let's get started!

If you look to the left of the screen, there are lots of different groups of code blocks, with different colours:



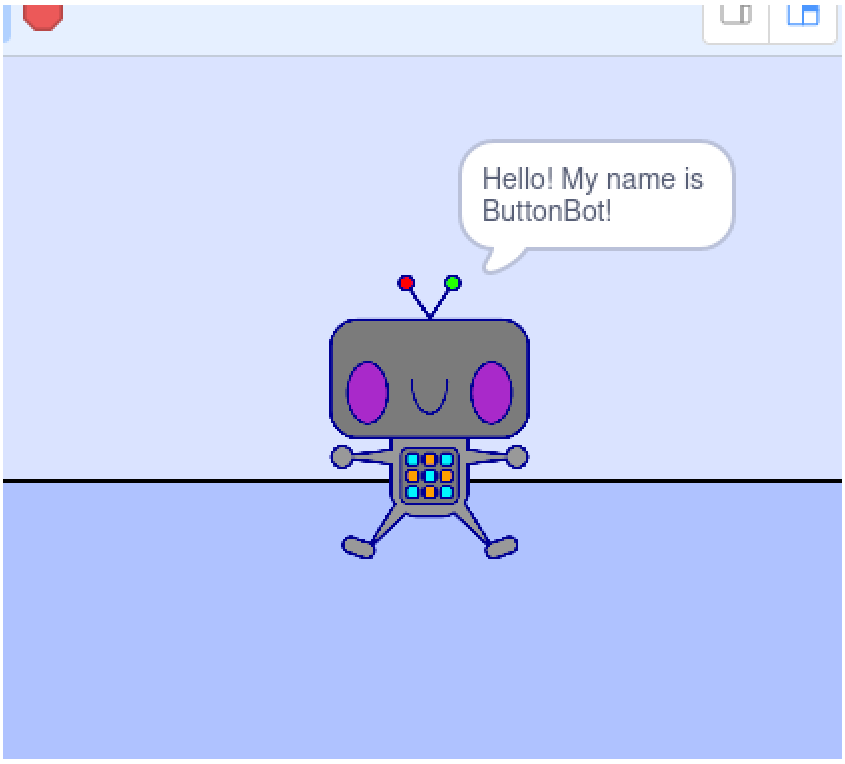
First, we need to tell the code when to start. So we will pick “when (green flag) clicked,” which means the code will start when the game starts.

And then we’ll make our little robot do something, so we can pick “say” from the “Looks” menu. If we want the robot to make a sound, we can do that too, but we have to go looking in “Sounds”.



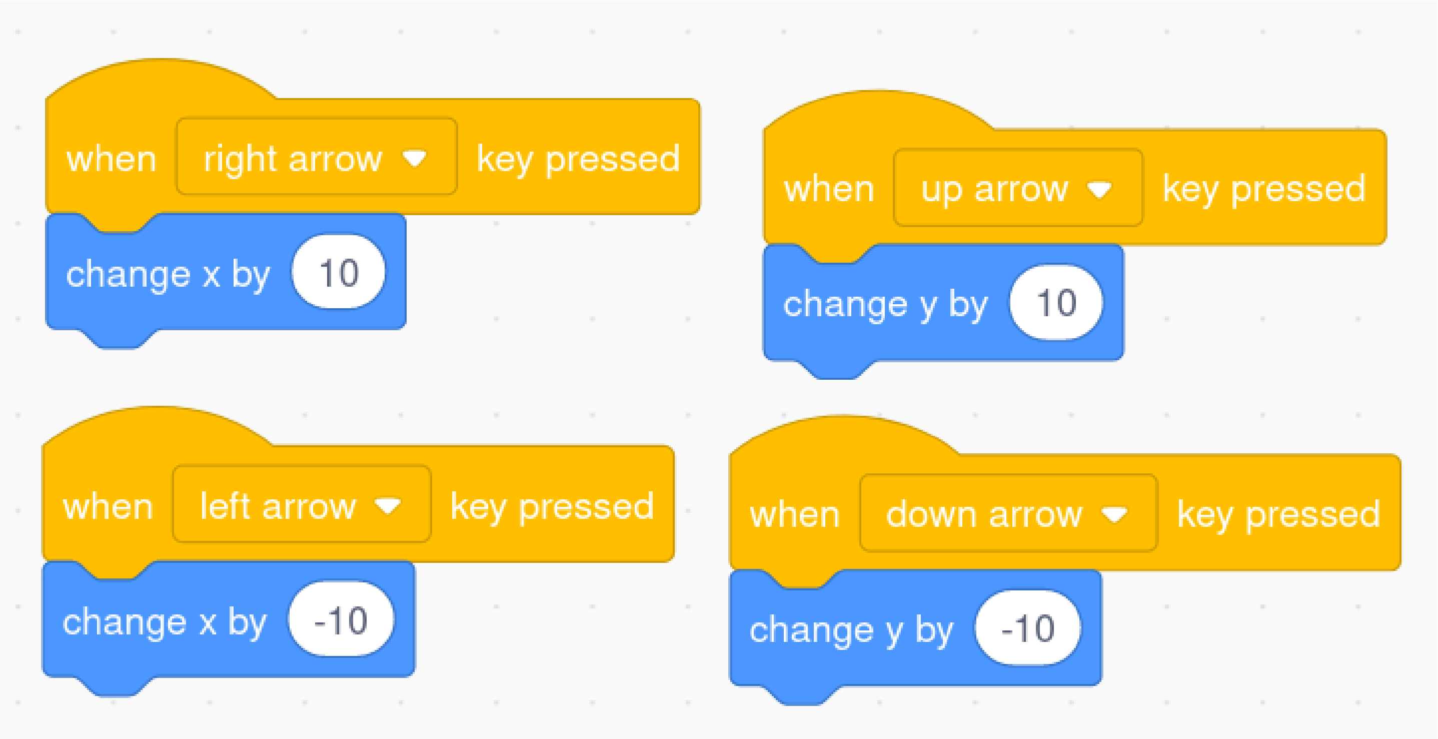
```
when ⚑ clicked
say Hello! My name is ButtonBot! for 5 seconds
```

What happens now when we click our little green flag?



You’ve already done some coding! Awesome!

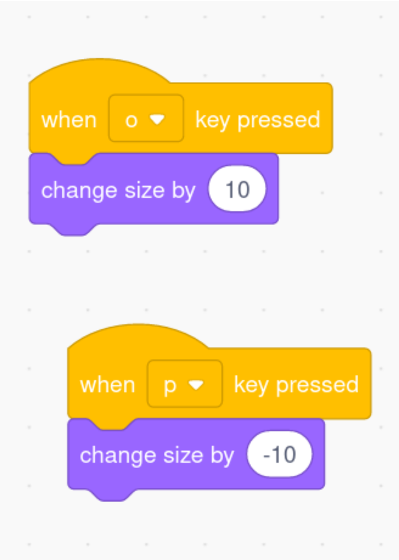
Next, we need our little robot to be able to move around the screen, or he'll never get anything done. The robot can move side to side, which is called x, or up and down, which is called y. If we add these code blocks, he will move when we press the arrow buttons on the keyboard.



```
when right arrow ▾ key pressed
change x by 10

when up arrow ▾ key pressed
change y by 10

when left arrow ▾ key pressed
change x by -10

when down arrow ▾ key pressed
change y by -10
```

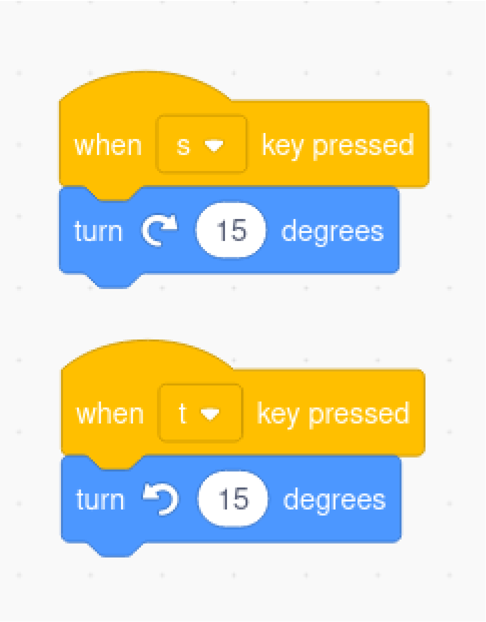
Remember to press the green flag to reset your code to the new settings, before you try. Does the robot move too fast or too slow? What happens if you change the 10 to 5, or to 20?

How about if we want to make the robot bigger, or smaller? Maybe we should be able to see if he can change size through the code?

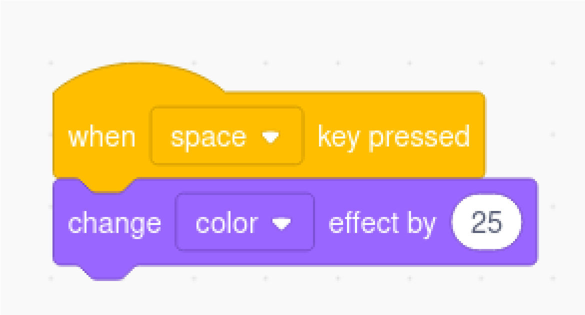


```
when o ▾ key pressed
change size by 10

when p ▾ key pressed
change size by -10
```

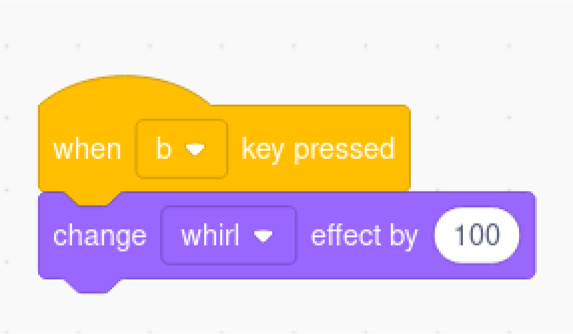
Or maybe he needs to spin around, or get a bit twisted? Try these bits of code and see what happens when you press the buttons:



```
when s key pressed
turn ↻ 15 degrees

when t key pressed
turn ↺ 15 degrees
```

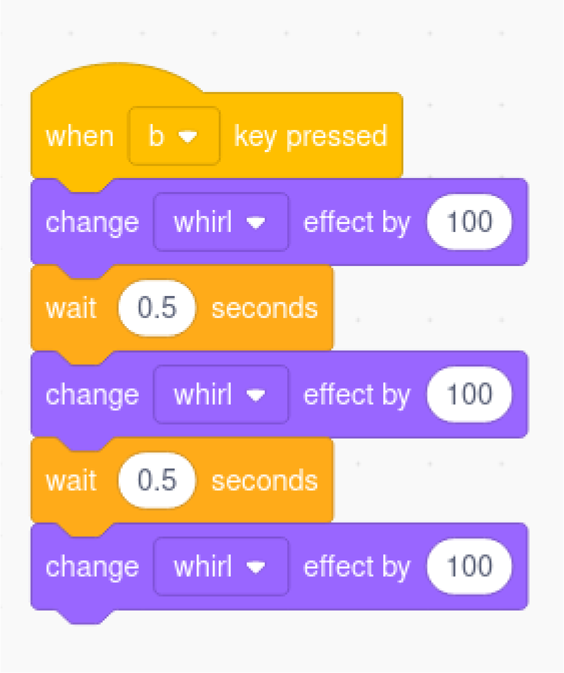


```
when space key pressed
change color effect by 25
```



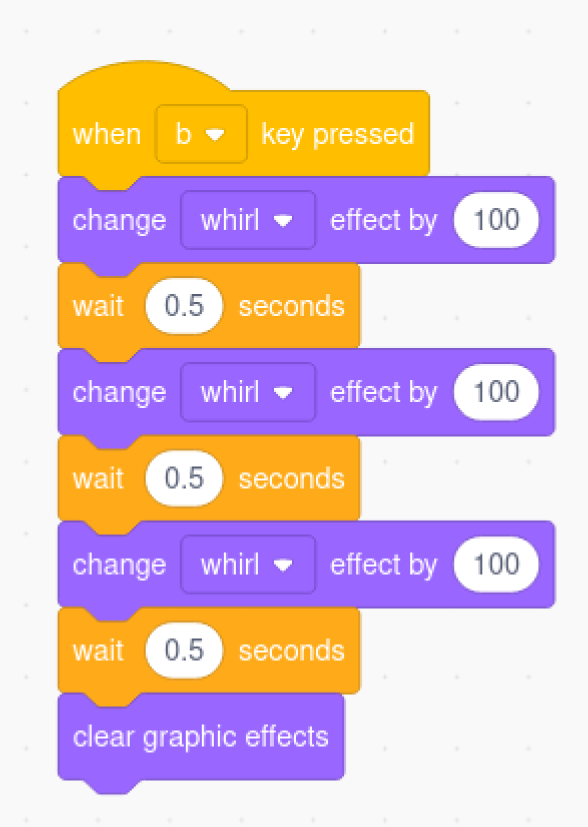
```
when b key pressed
change whirl effect by 100
```

The whirl effect is very cool, but it doesn't really twist the robot as much as Snoof would like. What would happen if we changed it to this?



```
when b key pressed
change whirl effect by 100
wait 0.5 seconds
change whirl effect by 100
wait 0.5 seconds
change whirl effect by 100
```

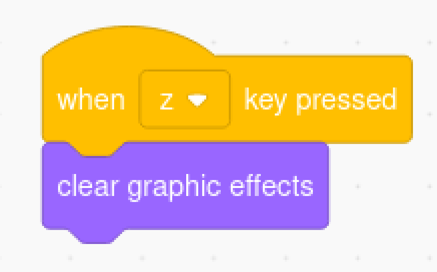
Whoa! That's a bendy twisty robot! What happens when you change the numbers? Does he twist more, or less? How can we change it to look how we want it to look? And now he's twisted… How do we untwist him?



```
when b key pressed
change whirl effect by 100
wait 0.5 seconds
change whirl effect by 100
wait 0.5 seconds
change whirl effect by 100
wait 0.5 seconds
clear graphic effects
```

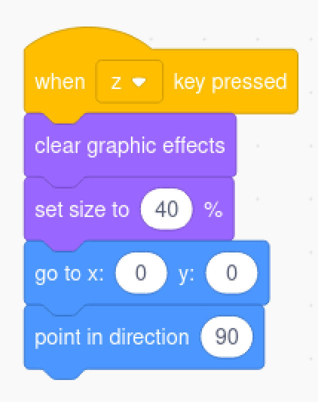
Can you make some code to twist him in the other direction?

Clear graphic effects is an interesting one, because we’ve put on some of those effects and it would be good to know how to remove them, right? We can make it so one button press resets the robot, like this:



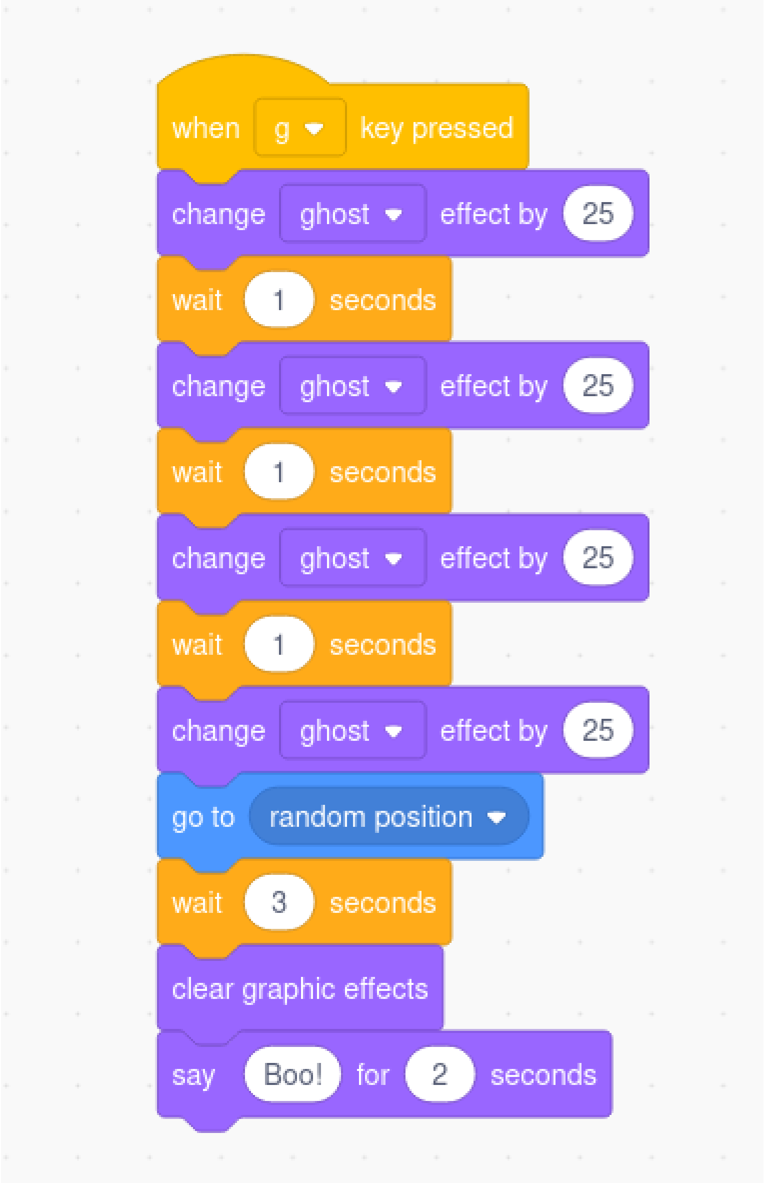
```
when z key pressed
clear graphic effects
```

And we can add some blocks to reset his size, and where he is on the screen, and whether he’s still spun around:



```
when z key pressed
clear graphic effects
set size to 40 %
go to x: 0 y: 0
point in direction 90
```

Now let’s try something more complicated – have a look at this code and think it through. Can you see what it will do? If you can’t work it out, you can always ask for help working it out!



```
when g key pressed
change ghost effect by 25
wait 1 seconds
change ghost effect by 25
wait 1 seconds
change ghost effect by 25
wait 1 seconds
change ghost effect by 25
go to random position
wait 3 seconds
clear graphic effects
say Boo! for 2 seconds
```

Now insert the code and try it out. Did it do what you thought it would? How can we change this code to make the robot behave differently?

What other code can you think of for your robot?