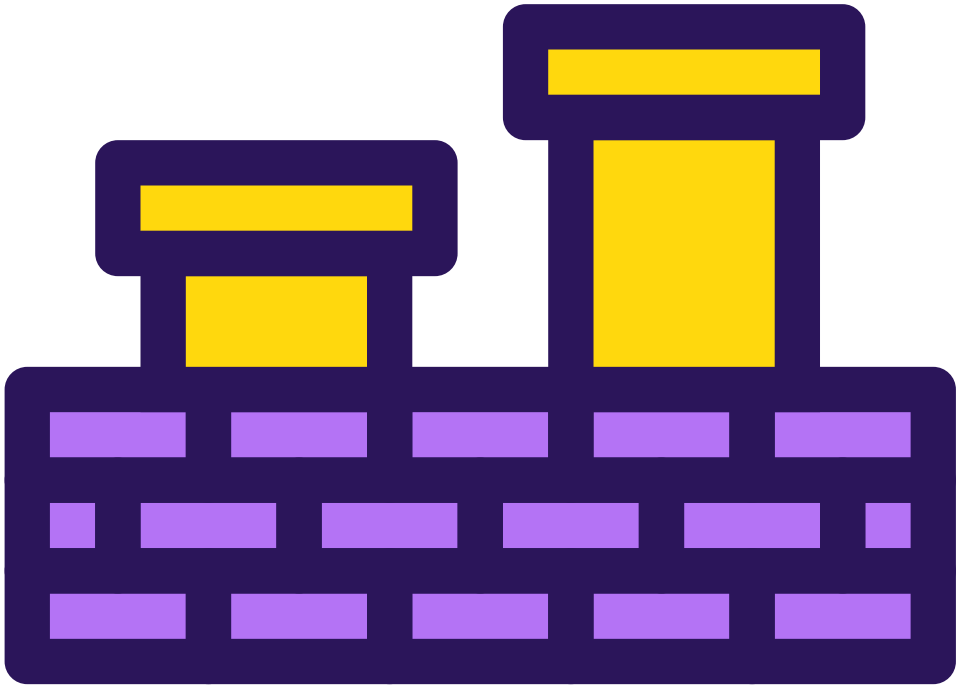
Hello!

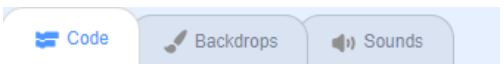
My name is Andy and I'm going to explain how to make your very own 'platforming game' on Scratch- from scratch.



For this first section of our platform game, we're going to focus on building our background. I'm going to set my platforming game in space which will be the basis of my instructions. Of course, you don't have to set yours in space if you don't want to.

In these instructions, I'm going to help you build 2 levels.

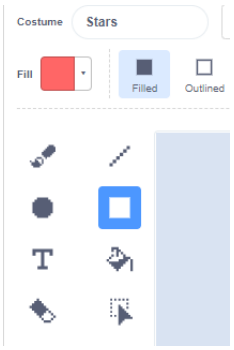
Go to the bottom right of your screen and click the **choose a backdrop button**. Here you will find plenty of backdrops to choose from, the one I'm going to use is just called **stars.** You can scroll through all the options until you find it, or type it into the search box- your choice.



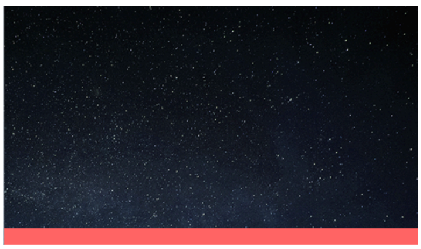
Once you've found your background go to the top left of the screen and click on **backdrops.** This will change the page and give you the option to customize your backdrop, so get your artistic skills ready- don't panic, it's nothing too taxing in these instructions- though if you are more artistically inclined then you can make your levels look even more epic in your own time. You can also import images as well.

Select the rectangle tool and don't do what I accidentally did and make sure it says filled at the top- not outline.

I'm going to start with red (which you can see in the **fill box**. [0,60,100 if you want the exact same colour). Use your rectangle tool and make a rectangle that goes from one end of the screen to the other.



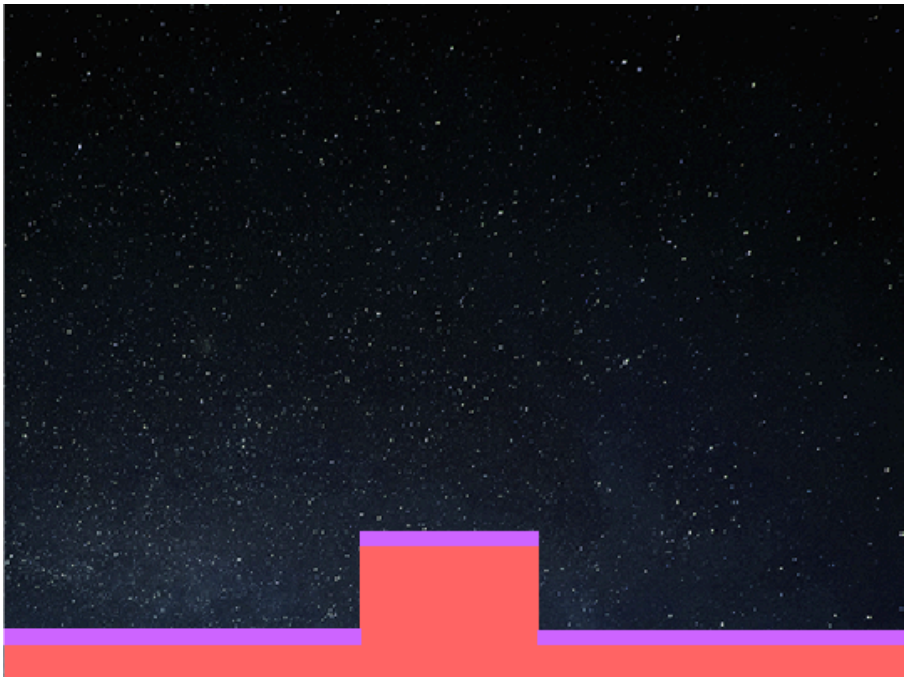
It doesn't need to be very high so make sure you don't fill too much of the screen with it.



Then use the rectangle tool again and make a small box the player has to jump over in the middle of the screen. This is just the first level to teach our player the basic controls, so we don't want anything very difficult... yet.

Now we need to change our fill colour. I'm going to use purple [80,60,100] as I think it looks good next to red- but again you can select whatever colour you like. If you're making your game in a jungle you might prefer brown and green- just make sure the two colours are a different colour than your background otherwise it'll all blend in.

With our new colour, we're going to create a path on top of the first one we made.



Excellent work! Now I'm very aware we haven't actually written any code yet. I think we should probably do something about that...

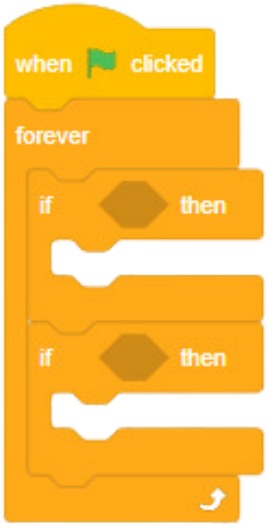
We need to select a sprite our player is going to control during the game. I'm going to use something different from the cat, so let's delete the cat. I'm going to use a sprite called Gobo. Use whatever sprite you want, you add it the same way as a background- just click add sprite rather than add backdrop.

And now it's time to finally do some coding! About time right?

The first thing to do is get our game character to move, otherwise, it'll be very difficult to play. Before we start adding code, just make sure you're clicked back on code (top left) and you've clicked on your sprite in the bottom right of the page.

Now we've selected our sprite, let's add code! The first block we need is in **events** and is **when [green flag] clicked.** This makes it so when the green flag is clicked it will run the code we attach underneath.

Next, go to the **sensing menu** and get the **forever block**. Whilst you are in the **sensing menu** we also need two of the **if-then blocks.** Pop then two **if-then blocks** inside the **forever block.** You should then have something that looks like this.



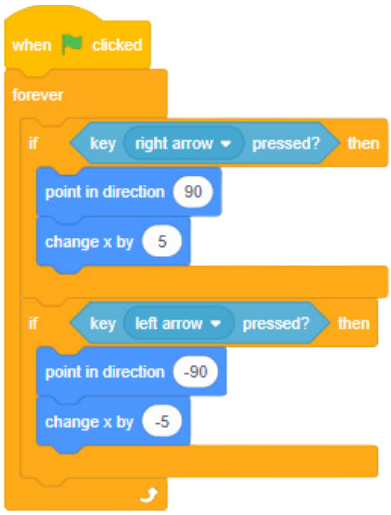
Head to the **motion menu**, and get two **point in direction 90 blocks,** and two **change x by blocks.** You need a point in direction and change x by in each **if then block.** You need the point in direction at the top and change x by blocks underneath.

Change the top set of numbers to 90 and 5 and the lower loop numbers to -90 and -5. You can do that by clicking on the numbers.

Do this and you will have something that looks like this

Go to the **sensing menu** and select **key space? pressed-** You're going to need two of them. Put them in the blank spaces next to the two places it says **if.** Click on the **space** word and change the top one to **right arrow** and the bottom one to **left arrow.**

Your code will look like this.



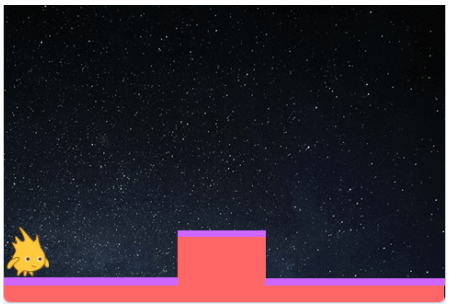
For the next part, we need to make sure our player can't just scroll through the level without having to jump on the platforms.

Head to the **events menu** and get a **when [green flag] clicked.** Then down to **looks** where you need a **set size to 100.** Change this **100** to **50** and clip it onto the bottom of the **when [green flag] clicked block.**

If you've tested your code (which you should have done, even though I don't remind you to) - you'll have noticed that when your sprite moves to the left it goes upside down- which is annoying. To stop that from happening go to the **motion menu** and select **set rotation style left-right** and add it underneath **set size to 50**. This will stop our sprite rotating and make it just face left or right.

Back to the **motion menu** and get a **point in direction 90,** so our character always starts facing in the right direction.

Next, you need to click on your sprite in the top right screen and drag it to where you want the player to start- on the layer you made across the bottom. You need to be touching your top layer colour (purple for me!)



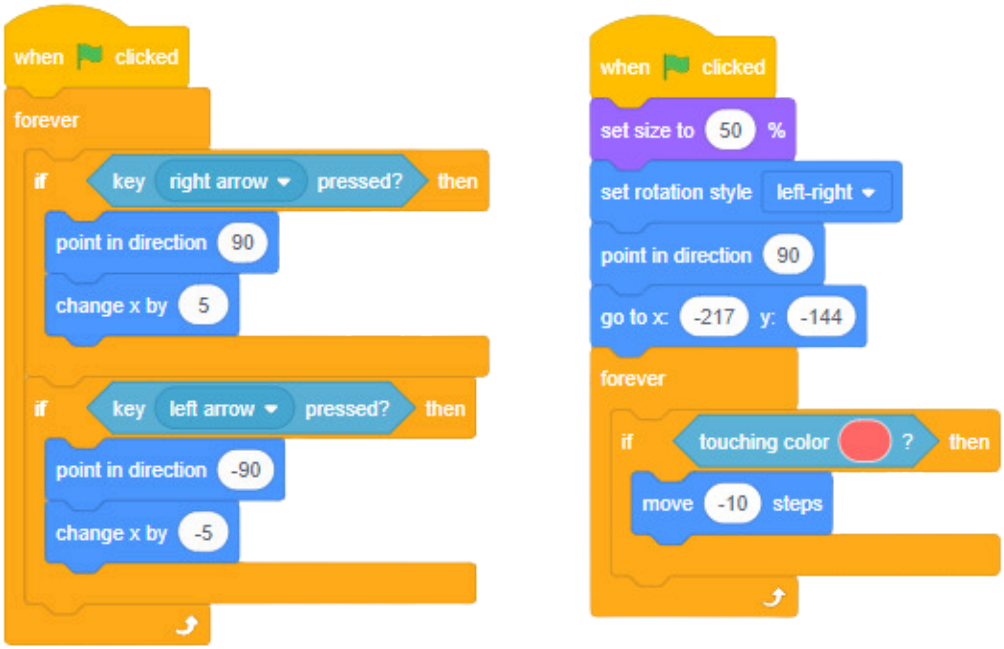
Go to the **motion menu** and select **go to x (#) y (#).** Hopefully, where I've put a hashtag (#), you should have the same numbers that your sprite is on. To check look at the coordinates underneath your game box in the top right corner. If the numbers aren't the same you can click on the block numbers and manually change them. This goes underneath **set rotation style.**

Now you need to go to the **control menu** and select **forever** which goes underneath **go to x: (#) Y: (#).** Then go to the **control menu** again and get an **if then** block which we'll sandwich in between our **forever** block.

Go to the **motion menu** and get **move 10 steps.** This is going between our **if then block.** and the number needs to change to **-10.**

You'll have noticed that there is a blank space in between **if** and **then,** which we now need to fill. Go to the **sensing menu** and select **touching colour ?** which we're going to place in the blank space. Then click on the colour (it'll be green) and select the **pipette** at the bottom. You then need to click on the bottom colour of the length we created.

The code will now look like this.



So, how are we going to get over the platform in the middle? We need to make our character jump.

Go to the **sensing** menu and select **When space is clicked.** You can use the space bar as your jump button if you like, but because I'm lazy and want to play with one hand, I'm going to change it to **when up arrow is clicked.**

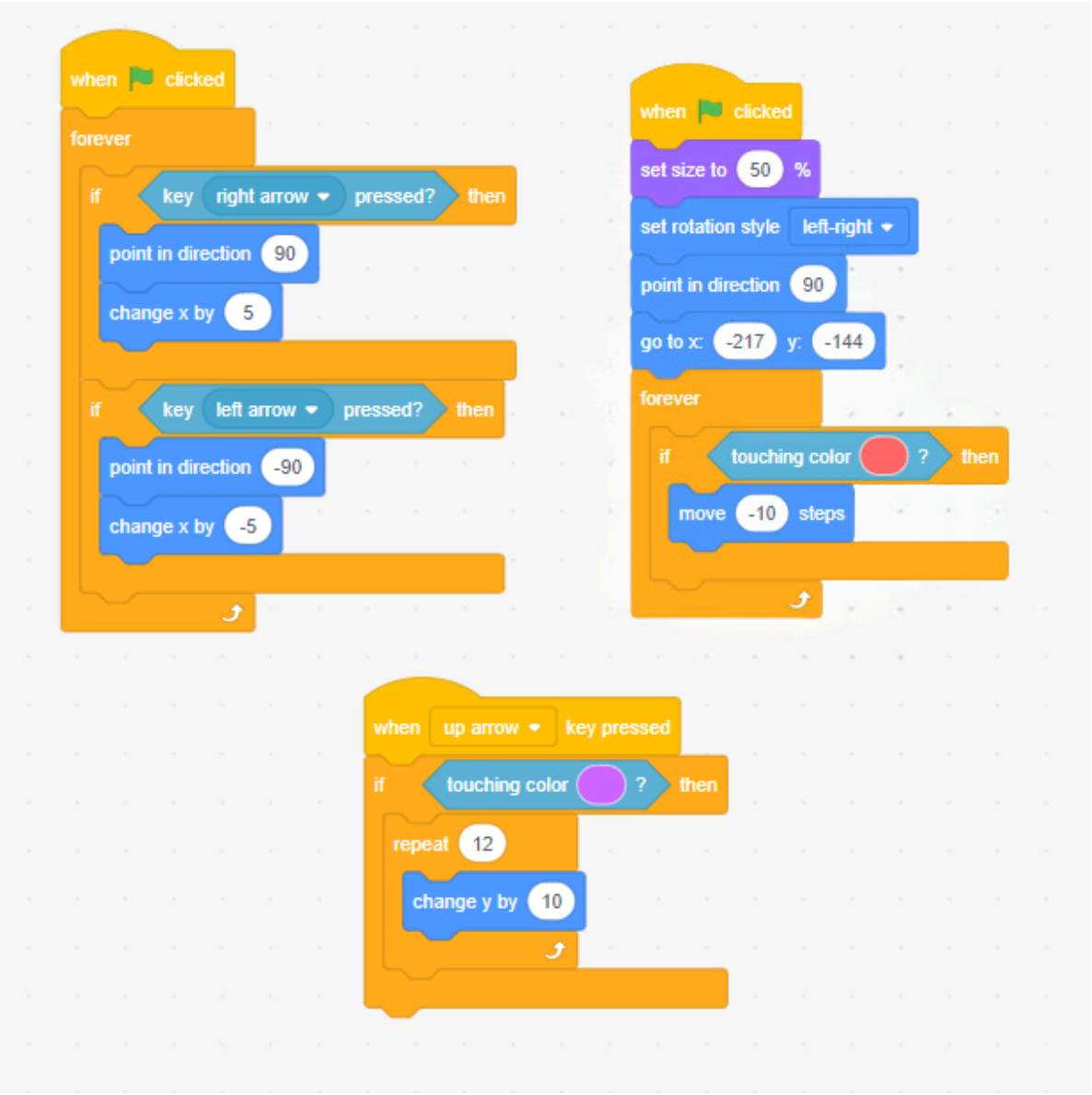
Then go to the **control menu** and select our old friend, which I feel like we've used a lot in the making of this game, the **if then block.** I bet you can't guess where we're putting it.

To make our sprite jump smoothly we are going to go back to the **sensing** menu and select **repeat 10** which we'll put in between our **if then block.** Change the number 10 to the number **12.**

Go to **motion** and get a **change y by 10 block** which needs to go in between **repeat 12.** Depending how high you want your sprite to jump, depends how big that number is- I would recommend 10 though, as you might be able to add a double jump at some point as a power up so you don't want to make it too easy for them.

We want to make sure our player can't jump anywhere we need to add a **touching colour** from the **sensing menu** to the blank space between **if and then.** Click on the colour in there (it'll be green again), select the pipette and this time change the colour to the top layer of your platform. (In my case purple).

Let's compare code.



The more observant amongst you will notice my numbers have changed in the top right piece of code- that's because I wasn't quite touching the purple so my code didn't work, if your sprite isn't jumping, chances are that's why. I make mistakes so you don't have to!

Now you've tested your code, you'll notice that our sprite just stays in the air and doesn't come back down again.

I'm going to save you some reading as we've found everything you need for this next bit before so I'm going to leave you to it. You need a **when [green flag] click block, a forever block, a if else block, a touching colour block and two change y by blocks.**

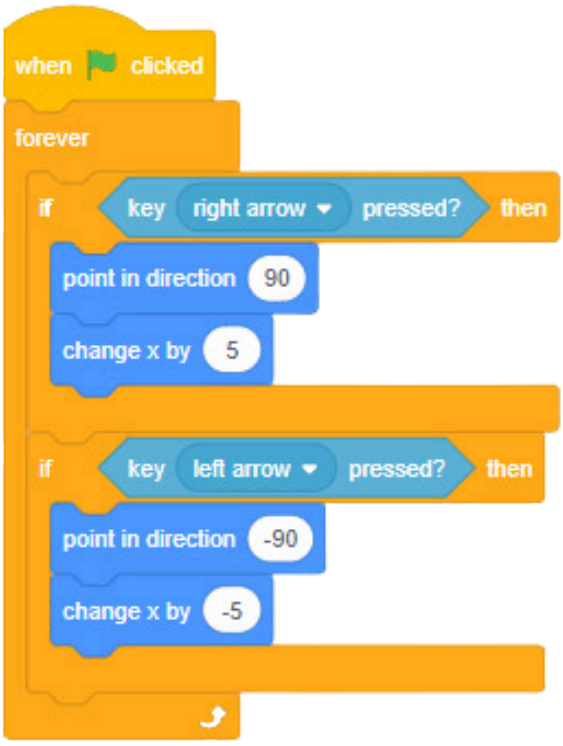
These blocks will give your sprite the ability to float back down. So place the **when [green flag] clicked on top of the forever block,** with the **if then else block** in between **forever.** In between **if then** and **else** place a **change y by block** and add the **touching colour block** to the blank space in between **if** and **then.**

The colour needs to be the same colour as the top layer of your floor- do you remember how to do it? *hint* the pipette *hint*

Change the top number to 0 and the bottom number to -3. This means when the sprite is touching the top level it won't do anything and when it's not, it will fall down forever until it's touching the top level.

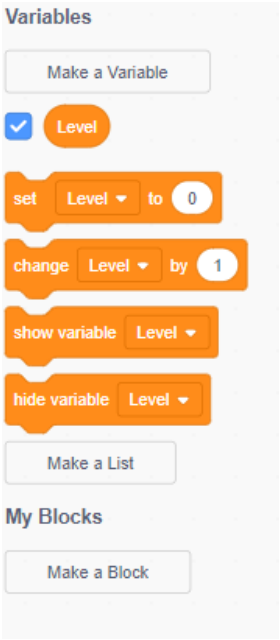
As an experiment see what happens when you change that 0 to a higher number.

Here's a code update.



Creating more levels

To begin with, let's create a way to tell the player know which level they're on. To do this we're going to create a **variable.** To this go to **variables** and click make **a variable.**



You could call this variable whatever you like, but I'd recommend calling it **level** to keep things easy to remember. One of the big problems with code is that people make it too complicated by naming their variables something silly.

A variable is a value that can change depending on conditions or information. Simply put, the number can constantly change.

Once you've created the variable you will see a level box appear on your game screen in the top right corner. If you don't want to tell players what level they're on, just untick the blue arrow to make it disappear and you can add it again later if you change your mind.

These instructions are already getting a little bit long for my liking. It's almost an entire book at this point so I'm not going to go through how to make every little bit of a new level here.

You need to create your second level by clicking on create a backdrop. (Check the instructions further up if you need some help)'

Feel free to make this level more difficult. But remember it's only level 2 so it shouldn't be that much more difficult than your first level, but it's your game so you can honestly do whatever you like.

Rightyho here we go.

Now you have your second level let's make it so our player goes to it once they've completed the first level. But first, we need to make sure our game always starts from our first level, which means adding a couple of code blocks to some code we've already made.

Remember this code?

This is the code we are going to add a couple of blocks to.

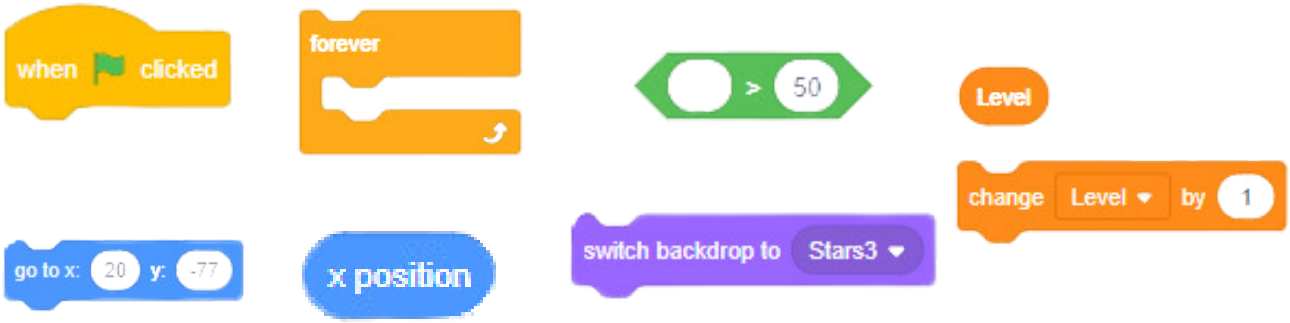
You'll find the first block in **variables.** And it's the second one down in the menu **set level to 0.** Get yourself one of these and add it underneath **point in direction.** What this tells our game to do is when our player restarts the game the level counter (if you're showing it) will be set to 0 - oh, change that 0 to 1... so our level counter will start at 1.

The second block we need is found in **looks**. It's called **switch backdrop to** (7th one down- or 14th up, depending how complicated you want to make it.) Add this underneath the **set level to 1**. And as if by magic (the magic being your mouse clicks) you will have code which looks like this.

(Don't worry if the X and Y coordinates are different to mine!)

This means that our player will always start the game from level 1. If you fancied making a unique game I suppose they could start from the last level and work backwards.

Our final step (yep, the final step- you'll soon be free of my instructions) is to tell our game to change levels when the player reaches the end of a level. Here's a shopping list of blocks for you to get.



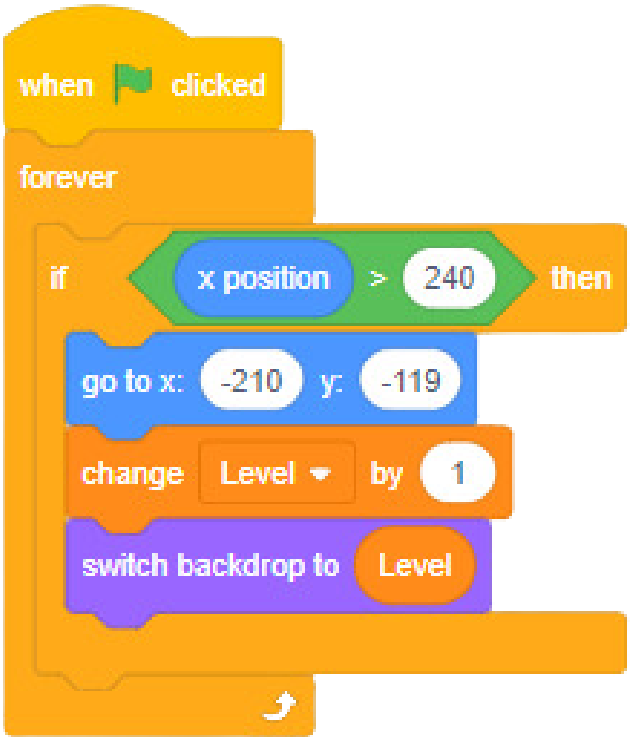
Now to alter some numbers and drop some things into place. First that **x position** needs to go into the gap before **50** in the green block. And change that number from **50** to **240**. (240 is the edge of the game map on the right-hand side)

Next, that **go to X and Y** numbers need to be the coordinates you are starting your second level on. I always try and start my levels from the same spot, but this might change. So find the coordinates on your second level start point and add them in.

And that **level block** needs to go over the option next to **switch backdrop to**. (As a note that I probably should made earlier, yours probably won't say stars3 next to it)

It doesn't look like it yet, but this code will tell our game to change our level when our player reaches the right-hand side of the screen. (This is the game I'm making- but you might prefer them to have to get to the top screen, all you need to do is change the numbers around a bit- and if you want the level to change when your player gets the top - that X position will need to be Y position instead).

That was a lot of instructions and could get confusing, so here's a picture.



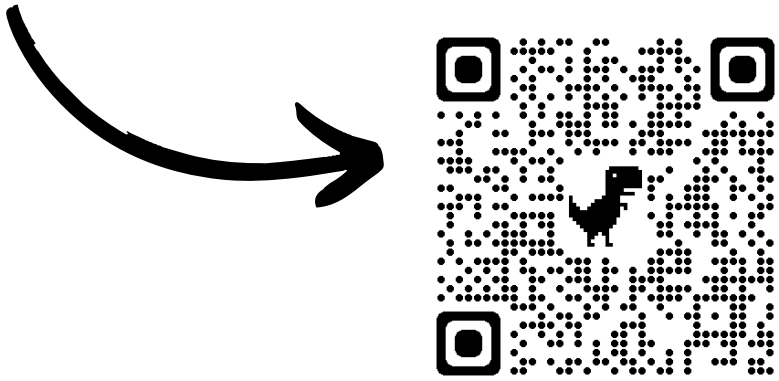
Beautiful right?

This code tells our game to go through the levels in the order they are laid out in our backdrops. You can, of course, add bonus levels and things but that's a coding idea for another day.

Challenge time!

Congratulations on successfully creating the beginnings of an epic platforming game.

Here are a few extra challenges if you want to really test yourself. If you want to complete the challenges with me, then scan this QR code.



- Create collectables to collect during the game
- Add even more levels with increasing difficulty
- Get your character to make a noise when it jumps