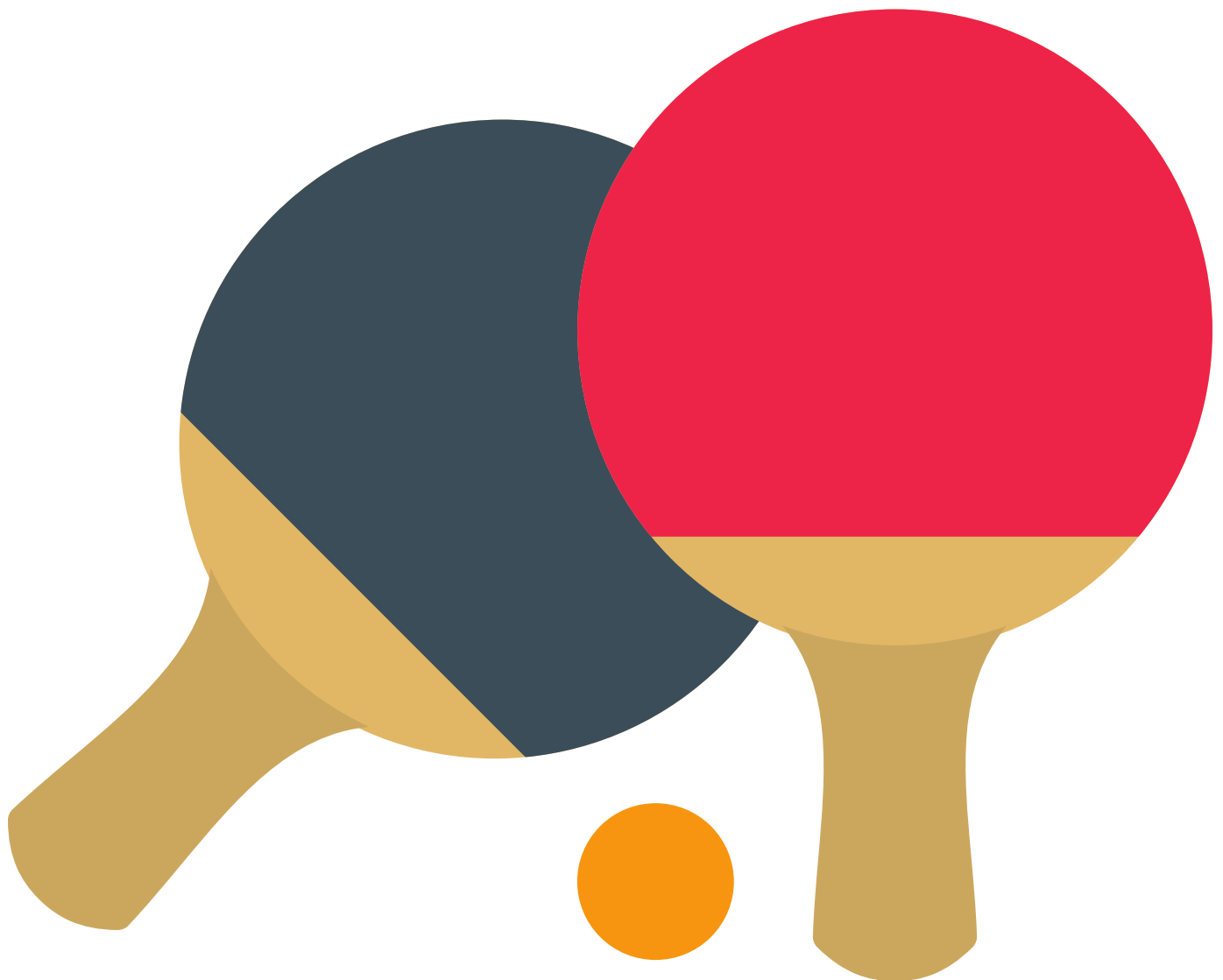




Hello!

My name is Andy and I'm going to explain how to make your very own 'Ping Pong' game in Scratch

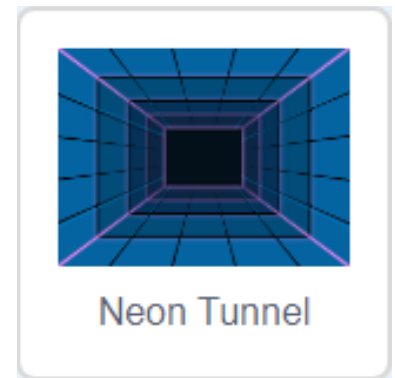


To begin with, we are going to choose a background. To do this click on the **Choose a backdrop** button in the bottom right of the screen. This will open up plenty of options for you to pick from.

I've personally gone for this backdrop called Neon Tunnel.

Feel free to choose the backdrop you like the most.

With your background selected, we come to the brutal part... time to delete the cat sprite.



With the cat out of our hair select a new one by clicking on the **choose a sprite** button towards the bottom of the screen and get the ping pong ball... it's just called ball.

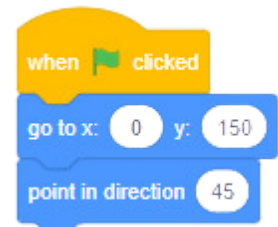
Let's get to some actual coding, shall we?

Go to the **events menu** and get a **on green flag clicked** block. Then go to the **motion menu** and get a **go to X: Y:** block. Staying in the **motion menu** get a **point direction 90** block.

With those blocks replaced, you want to change the numbers to read **go to x:0 Y:150** this will start the ball at the top of the screen in the middle of the background. And change 90 to 45 in the **point in direction 45**. This will make the ball point at a 45-degree angle.

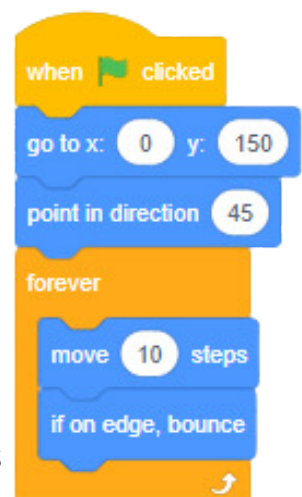
It's not very exciting code at the moment so let's keep going.

Go to the **control menu** and get a **forever block**. Whatever we put inside the forever block will happen forever... surprisingly. Go back to the **motion menu** and get a **move 10 steps** and an **if on edge bounce** block. These two blocks go inside the **forever block** which goes underneath **point in direction 45**.



This code will make the ball just bounce around the screen forever. If you wanted to slow the ball down you can make the **10** in **move 10 steps** a smaller number or a bigger number if you want to speed it up.

My lack of gaming ability means that I stick with 10 but you might be better than me so feel free to increase and decrease the number to match your skill level. I'll show you how to get the code to change the number automatically when players get a certain level of points a little later on.



Okay, our ball is flying around the screen with no signs of stopping so let's add a paddle so we can hit the ball and stop it reaching the bottom of the screen.

Add a new sprite called Paddle. It looks more like a line to be honest, but it's the one I like to use. Adding the new sprite will make it look like all your code has disappeared but before you panic and do something dramatic like rage quitting, I promise all your code is still there.

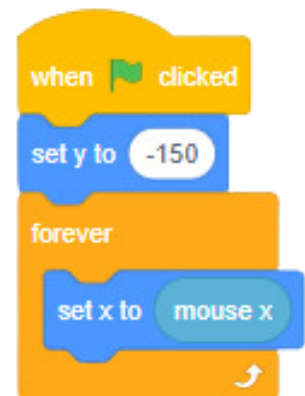
Each sprite can contain different code so if you click on the ball in the bottom right of the screen you will see your code come back.

You need to click on the paddle for this next bit of code though.

To keep things simple we're going to make the **paddle stick follow the mouse pointer** so we can use our mouse to play.

With the paddle sprite selected go to the **events menu** and get a **on green flag clicked** block, go to the **control menu** for a **forever block** and to the **motion menu** for a **set Y to** and a **set X to**. The final stop is the **sensing menu** for a **mouse X block**.

What we're going to do is tell the paddle to go to the bottom of the screen and move left and right depending on where the mouse point is. The code blocks you just got need arranging like this.



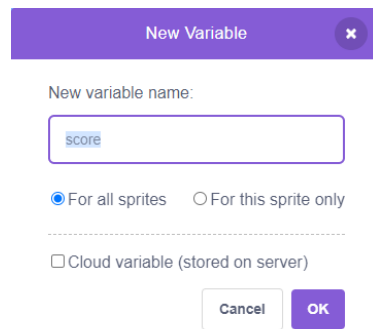
What this tells the paddle to do is go to Y -150 which is close to the bottom of the screen and follow the mouse to the left or right without going up or down.

The mouse x block looks at what X number the mouse is at and makes the paddle travel to the same number... just lower down. The **set Y to -150** block isn't essential to have in as you can click and move your paddle where you want it on the screen and it will always start there.

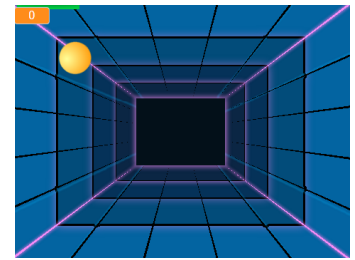
I only add the **set Y** block to protect the game from potential cheaters.

Time to add the score.

Go to the variable menu and create a new variable called score.



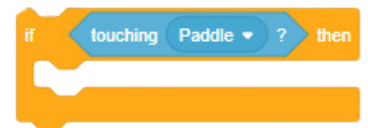
This will make a score box appear in the top corner of the game screen. If you double-click on it- it will make the box a little bigger and just look a little neater.



Come to think of it, we should probably fix a fairly big issue in our game... you know the ball just passing through the paddle and not doing anything.

Click back on the **ball sprite** in the bottom right corner of the screen and we'll add some code to it.

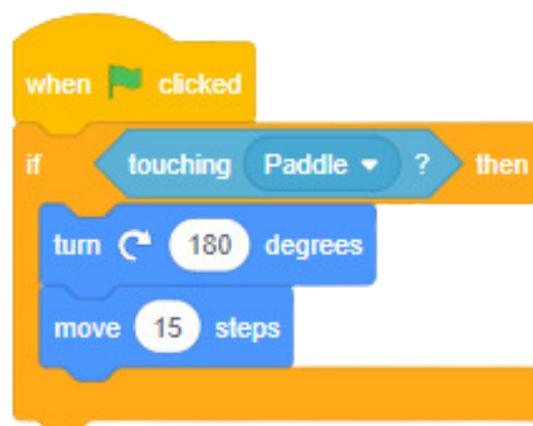
Now you've clicked on the ball sprite go to the **control menu** and get an **if then** block. We then need a way for the program to check if anything is touching the paddle and for the we go to the **sensing menu** and grab a **touching mouse pointer?** block. This is a strangely worded block it is basically saying, "Is this sprite touching something". Change the bit that says **mouse-pointer** to **paddle** by clicking on the arrow once you've dropped the block in the gap between **if** and **then**.



This is a slightly strange-looking block of code as it's not worded very well. The block asks itself a question. *Am I touching the paddle?* If I am I need to do what is inside this block. If I'm not touching the paddle I can ignore it.

To make the ball bounce off the paddle we can go back to the **motion menu** and get a **turn direction 15 degrees** block. Let's change the **15** to a **180** which means it'll bounce upwards when it hits the paddle. Grab a **move 10 steps**. This number needs to change to **15**.

Pop a **when green flag clicked** block on top and test your game.

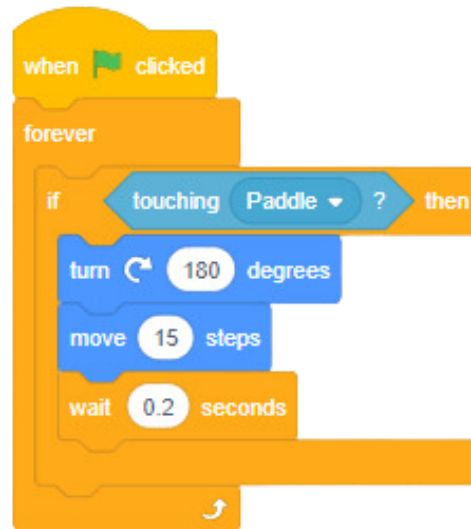


Let me guess. The ball doesn't bounce, and you're all shouting loudly probably saying rude words. Put down the keyboard and don't throw it away. It's not supposed to work.

Let me explain why...

Currently when we hit the green flag the ball checks if it is touching the paddle. It isn't when the game starts as it is at the complete opposite end of the game board. It checks once and only once throughout the whole game so we need to add a **forever block** around the code so the ball is constantly asking itself if it is touching the paddle.

Let's also get a **wait block** from the **control menu** to help prevent people from trapping the ball and getting infinite points.



Let's add the score now.

Add another **on green flag clicked** (you should know where it is by now) then go to the **variable menu** and get a **set my variable to 0**. We can change the **my variable** to **score** by clicking the arrow next to it. Also from the **variable menu** get a **change my variable by 0** - this is how we are going to add a point. This will also need changing to **score**.

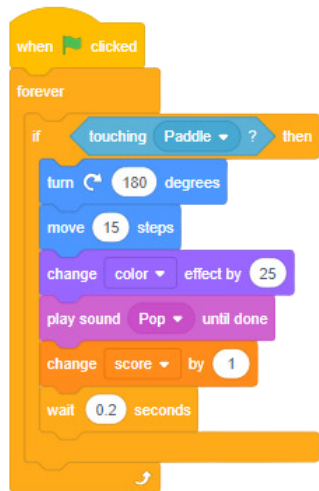
Add the **change my score by 1** (notice I changed 0 to 1) block into the **if touching paddle block** and the **green flag clicked and set score to 0** creates it's own code block.



Let's add some cosmetic and sound code to really bring this part of the game to life.

Go to the **looks menu** and get a block called **change colour effect by 25**. This is going in the **if touching paddle then block**. What this will do is change the balls colour when it hits the paddle which will look awesome.

Next, go to the **sound menu** again and get the **play sound pop until done** which is also going in the **if touching paddle then block**. They can both go just above the score.



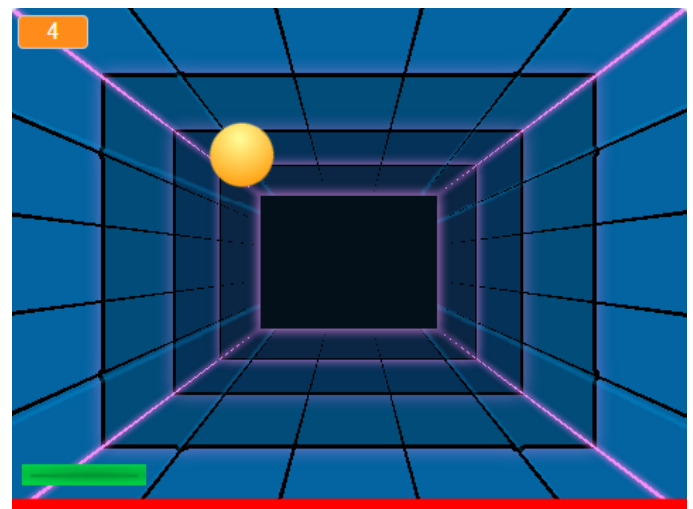
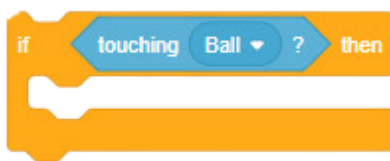
So, we now have a game that will go - literally forever- so let's add a game over mechanic.

For the game-over mechanic... we're going to need another sprite.

Add a new sprite called "line" - there is one called that (long red line) or you can draw your own if you really want to and we're going to place this right at the bottom of the screen, underneath the paddle.

We're going to use this line as the **game over** line.

So, whilst clicked on the new line sprite, go to the **control menu** and get a **if then** block and repeat the code we've already made so we have this...



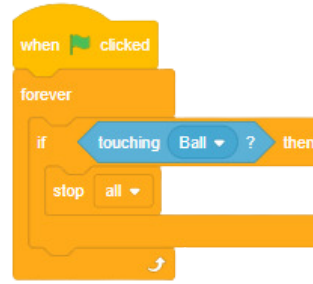
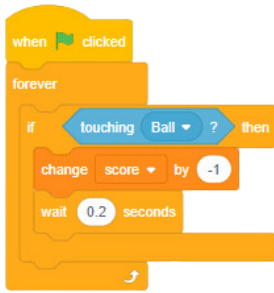
Notice how it says ball instead of paddle like we used before. This means the red line will be asking itself *am I touching the ball* and so it's asking it all the time but a **forever block** around it- again I'm not going to tire my typing fingers out telling you where to find that block.

Oh... let's start it when the **green flag is clicked** as well.

The final bit to add is inside the **if touching ball then** is found in the **control menu** and it's called **stop all**.

Now if the ball hits that red line... the game is over.

As an alternative you could add **change score by -1** in there so the player loses a point when they hit the line- rather than the game ending. It's up to you. Just add a wait to the **change score by -1** otherwise you'll be losing loads of points.



Right. Theoretically, now you have a game. But shall we make it a little more interesting?

We could add another level perhaps. I also talk about adding collectables into games in other instructions and videos so check them out and you might be able to use them here as well.

Right. Let's add a level first - which is basically just a new background.

If you click back on the **ball sprite** we need to add some extra code.

First... let's add a **switch backdrop to** (the name of your backdrop) underneath the **set score to 0** block. This means when your game starts it will always start on the backdrop we've been using this whole time.

Next you can add another backdrop of your chose... you remember how to do that? Because I'm too lazy to tell you.

With a new background chosen go to the **control menu** and get a **wait until** block. Then to **operators** for a **blank > 50**. This is called a **greater than** block. Drop that into the **wait until** blank space then go to the **variables menu** get a **score block** and add that over the blank space next to the >. Then, finally, go back to the **looks menu** and get a **switch backdrop to** [put in second background name] and piece it all together. Oh- maybe make the number smaller than 50... I normally go with 5.

This means that when the player gets a score greater than 5 ... 6 basically ... it will change the backdrop.

