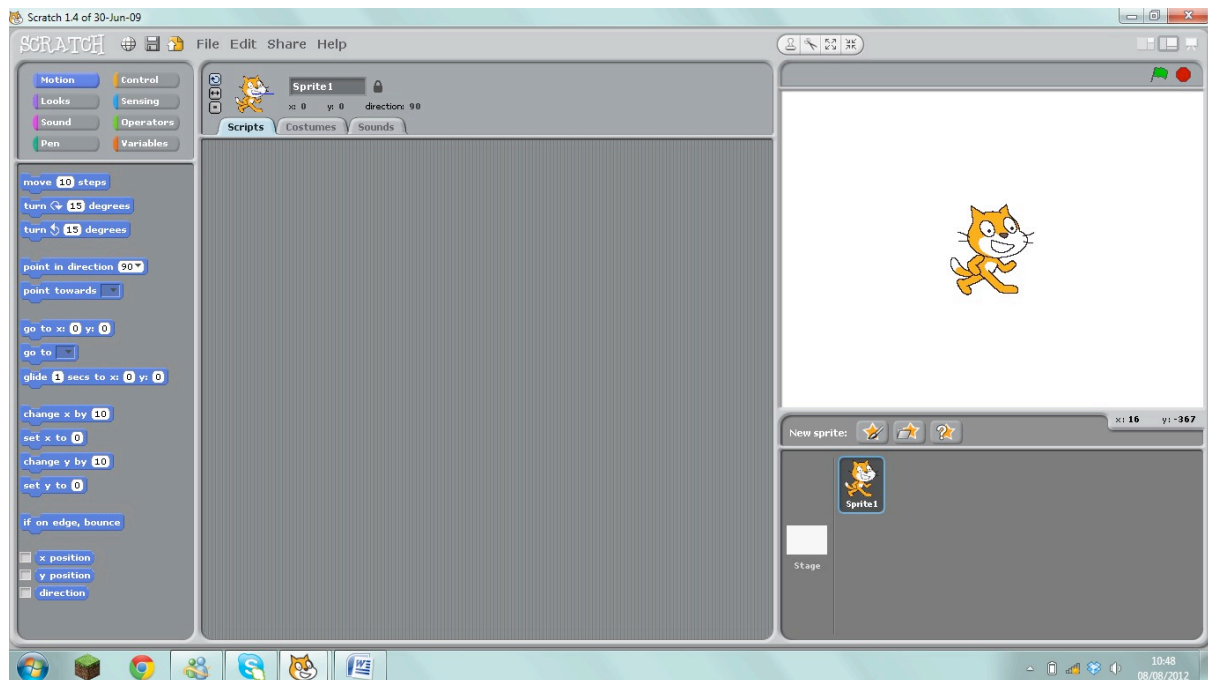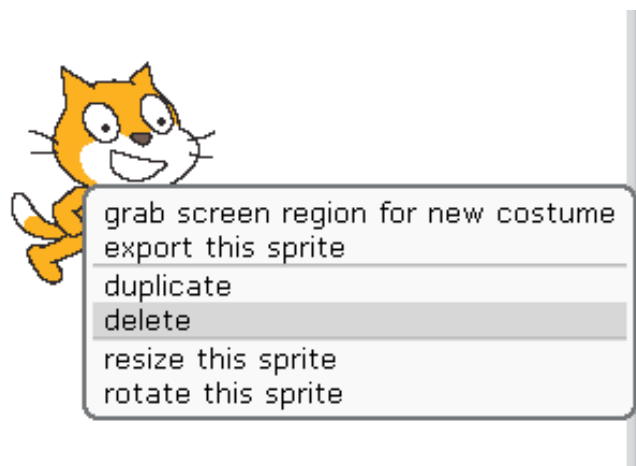# MAKING 'PONG' IN SCRATCH

STEP 0, SETTING UP SCRATCH:

The first thing we need to do when we open Scratch, is remove the cat object from the game screen... to do this we right click on the cat and select 'Delete':


We are now ready to start making a game! In this case, we're going to make one of the most famous old video games 'Pong'. Pong is an arcade game where the player has to hit a ball with a bat, and then get it into the enemy goal. It's similar to table tennis or air-hockey.

MAKING PONG IN SCRATCH
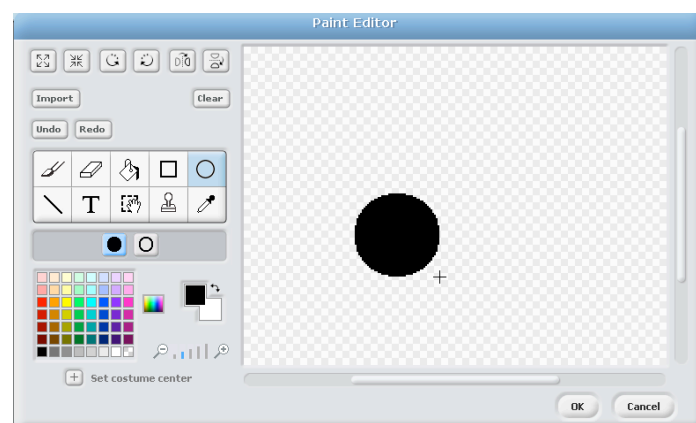
There are five different objects you need to make in scratch in order to create pong:

PlayerGoal

Ball

Player

Enemy

EnemyGoal
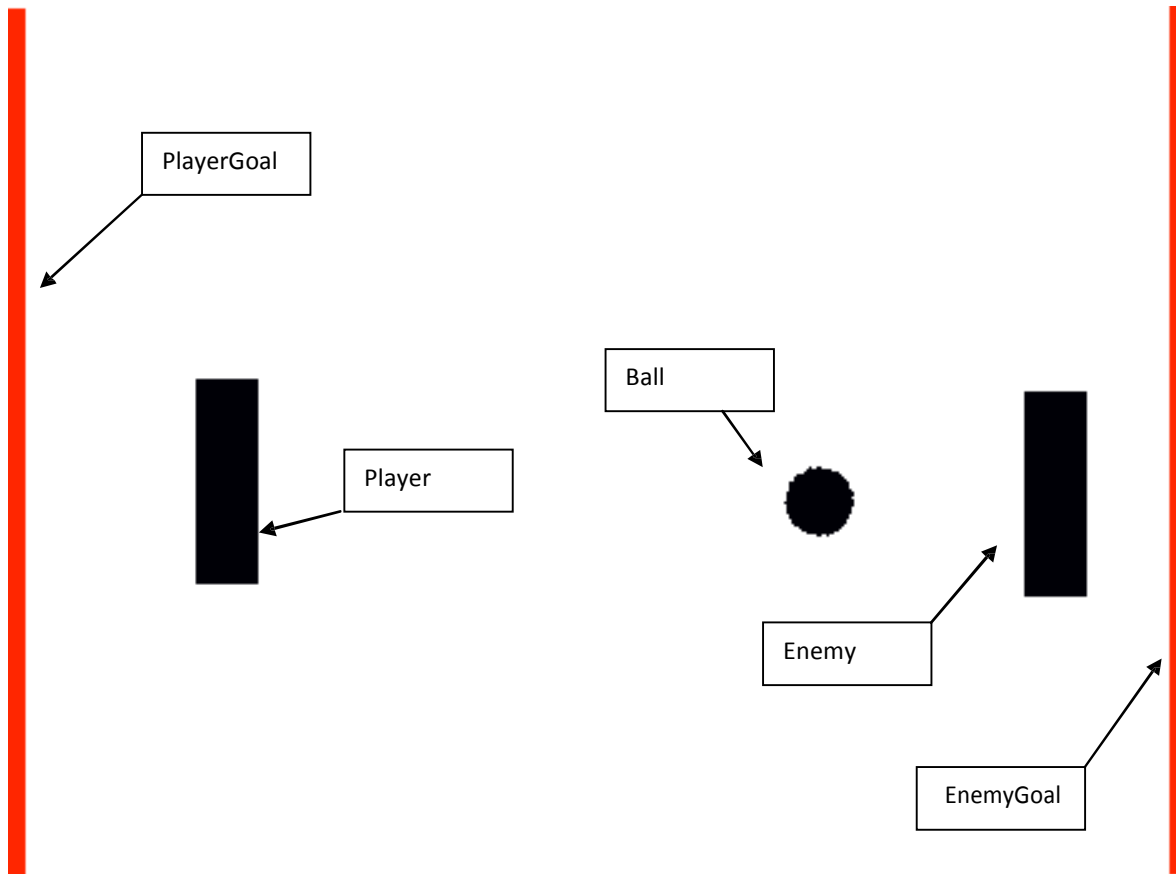
Lets first make the 'Ball' game object, click on the 'paint new sprite' button:

In the paint menu, draw a circle that will represent the ball on the screen, you can make it colorful (just make sure it is round!)

MAKING PONG IN SCRATCH

Hit 'Ok' and then name your ball in the box at the top of the screen:

Do the same thing for the other objects, making sure you name them correctly:

- Player
- Enemy
- EnemyGoal
- PlayerGoal

Well done! You've made all the objects you need to start making Pong! You can click and drag the objects around the screen to put them into the right places.
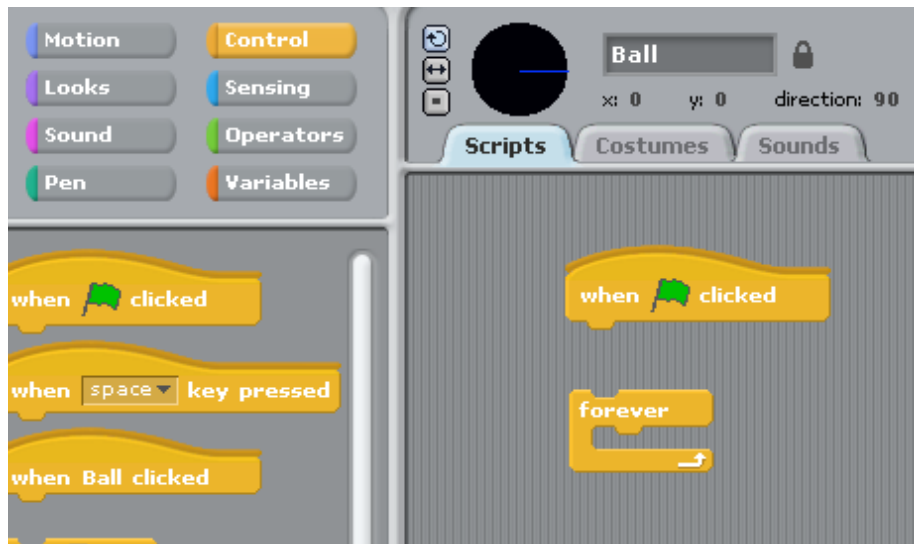
STEP 2, BOUNCING BALLS!

Double click on your Ball object to bring up its information:
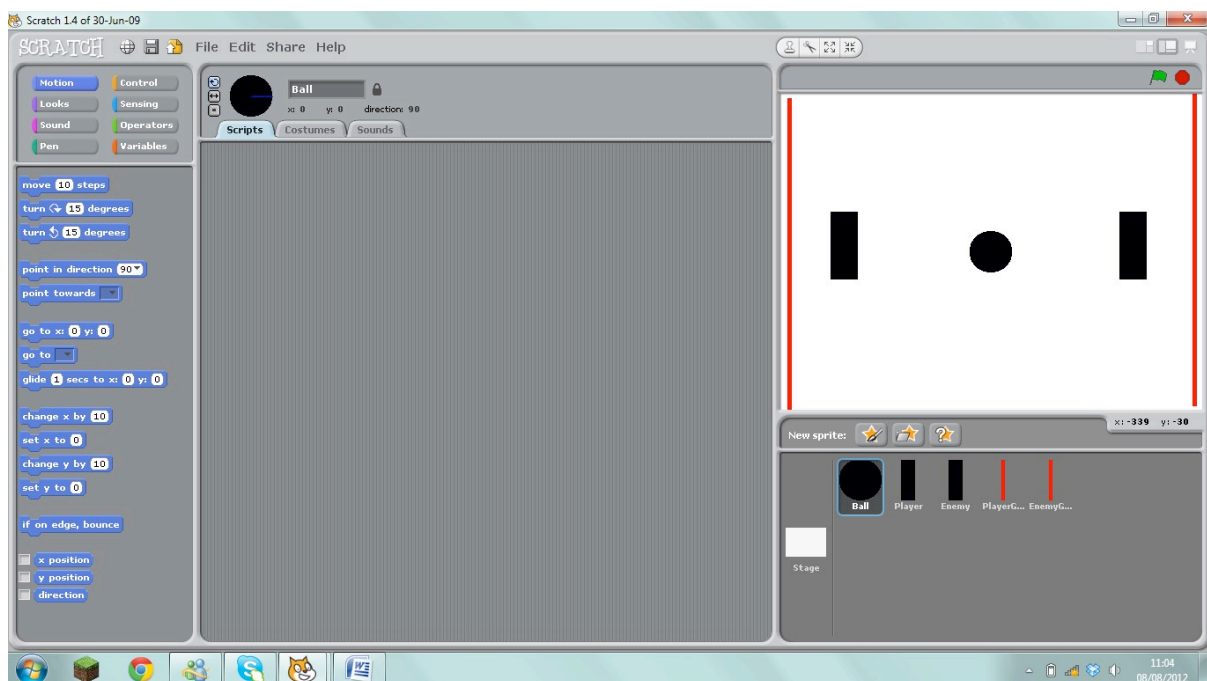
It's a little blank at the moment, so we're going to give it some instructions in order to get it moving around!

MAKING PONG IN SCRATCH

Using the orange 'Control' menu at the side of the screen, add the following pieces of code to your balls logic, you can click and drag it into place:

Then, add this code from the blue 'Motion' menu, the code should all fit together like a puzzle. Make sure you change the numbers in the puzzle pieces so they are like the picture below too:
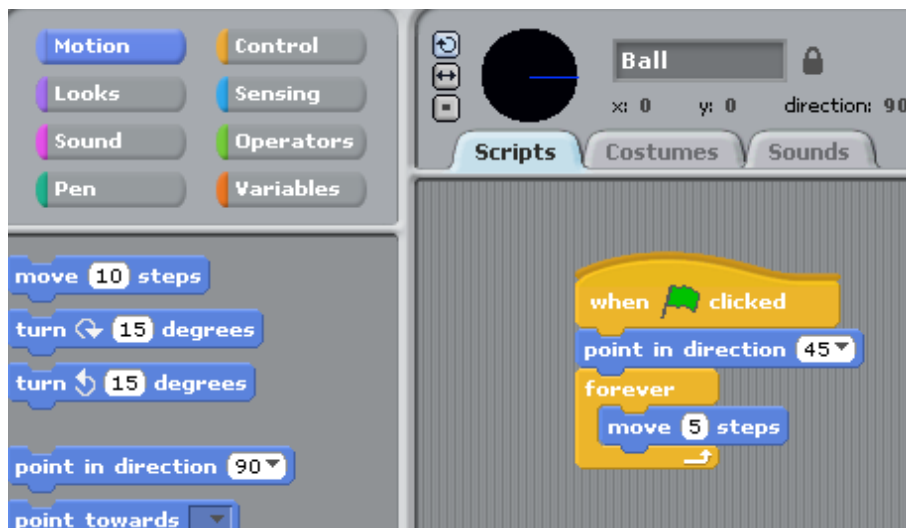

This code will tell the ball to point at 45 degrees when you hit the green 'start' flag, and then it will move 5 steps in that direction over and over again. This gives the illusion it is moving! We want the ball to bounce off the edges of the game though, so now we'll add the following code piece into the forever loop:

What do you think this means? Click the green flag to see what happens, then remember to hit the red stop sign afterwards!

MAKING PONG IN SCRATCH

You may have noticed that the ball doesn't go back to where it originally was in the middle of the screen, so you can change that by adding this puzzle piece to your code:
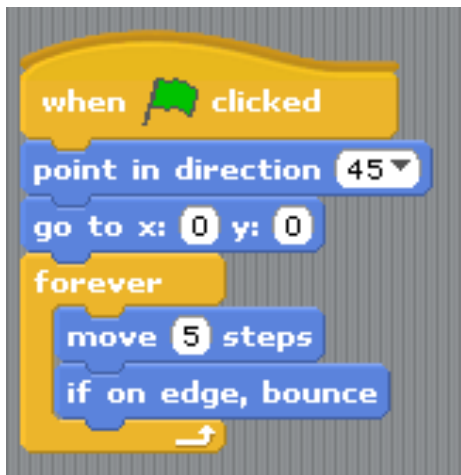

This means that when you click the green flag, the ball will move back to 0,0 (the middle of the screen) before it starts moving around!

STEP 3, HITTING THE BAT!

Now we want the ball to bounce off of the player and the enemy 'bats' this will mean that the player and enemy can defend their goals by pushing the ball away from themselves. To do this, we're going to make it so when the ball is touching one of the bats, it will bounce off in a random direction.

In the orange 'control' menu, drag two 'if' statements into your code:
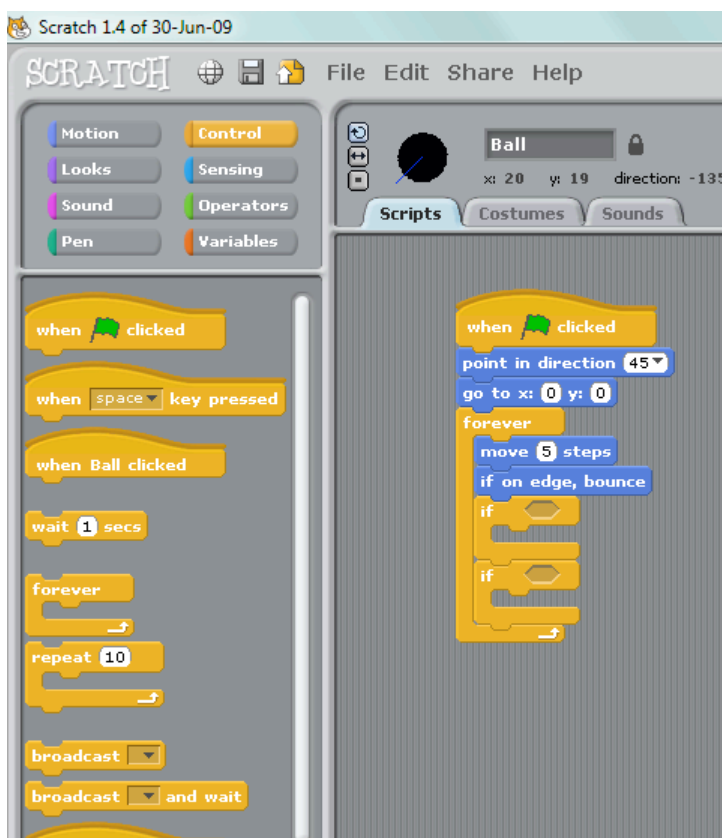
MAKING PONG IN SCRATCH

These if statements will only do the code we put inside them if whatever they check equals true. So now we go to the light blue 'Sensing' menu, to get something for the if statement to check. Add the following sensing code into your if statement puzzle pieces:
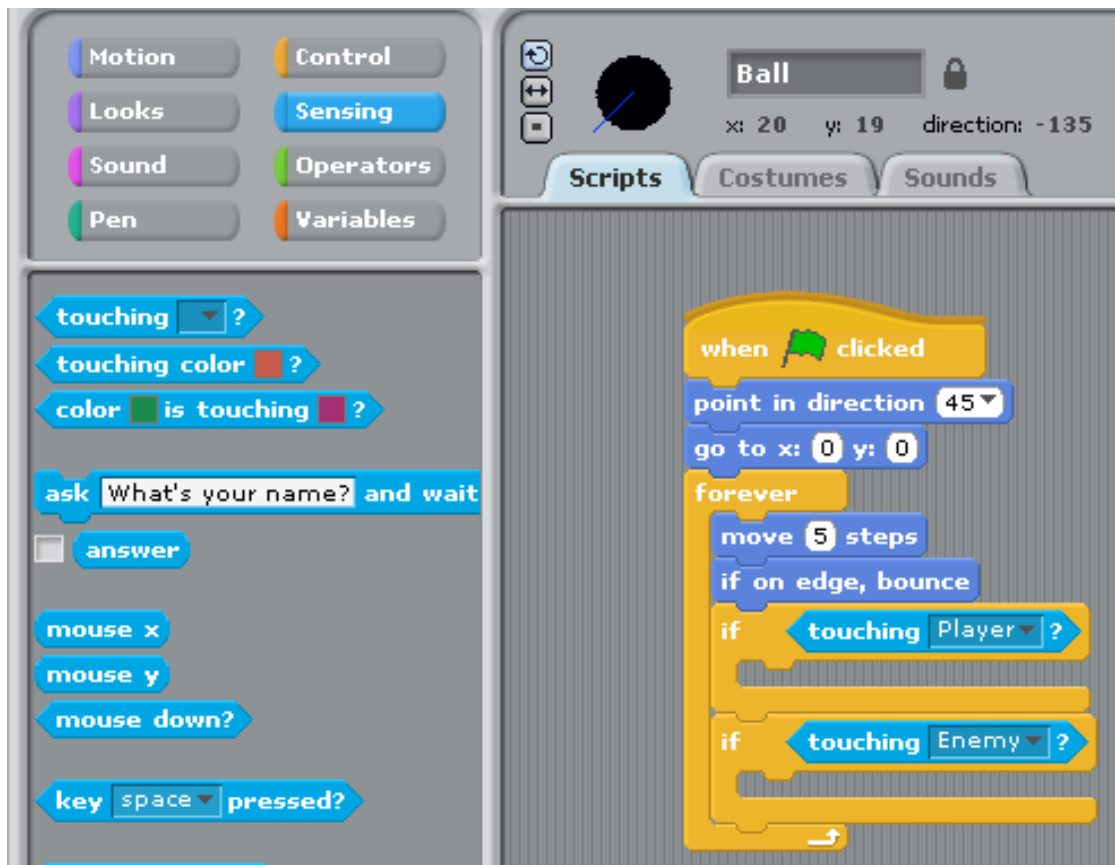

These two if statements will check if the ball is touching the Player or the Enemy object in the game. Right now they don't do anything though, so go to the blue 'Motion' menu and add the following:
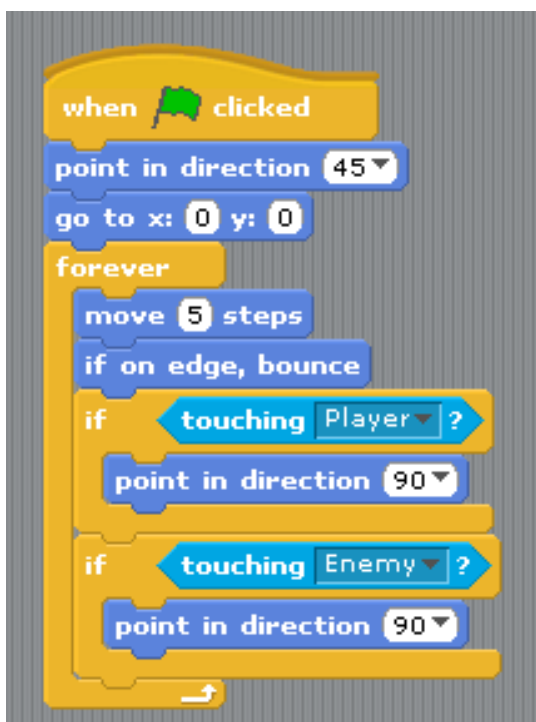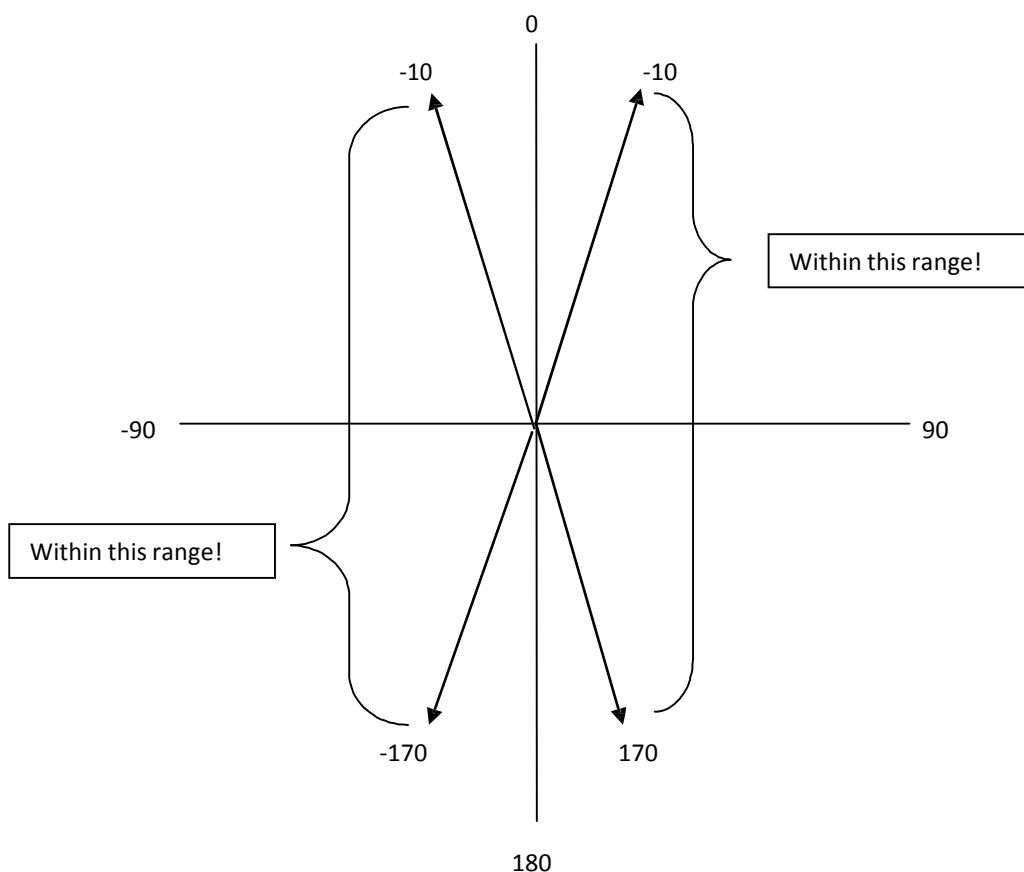
MAKING PONG IN SCRATCH

We want the ball to choose a RANDOM direction to bounce towards though, so to add that to the code we go to the green 'Operators' menu, and then add the following to our code:
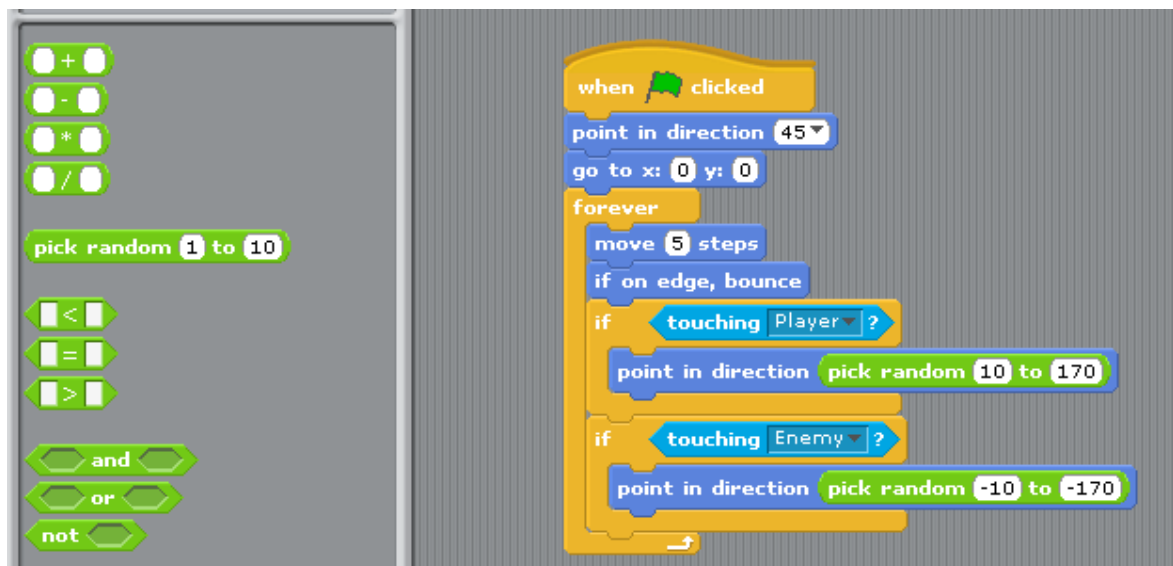
The 'Pick Random' operator should fit nicely into the 'point in direction' command. When it is used, it will choose a random number between the minimum and maximum you give it. In this case, we tell it to bounce towards the right if it hits the olayer, and to the left it bounces off of the enemy!
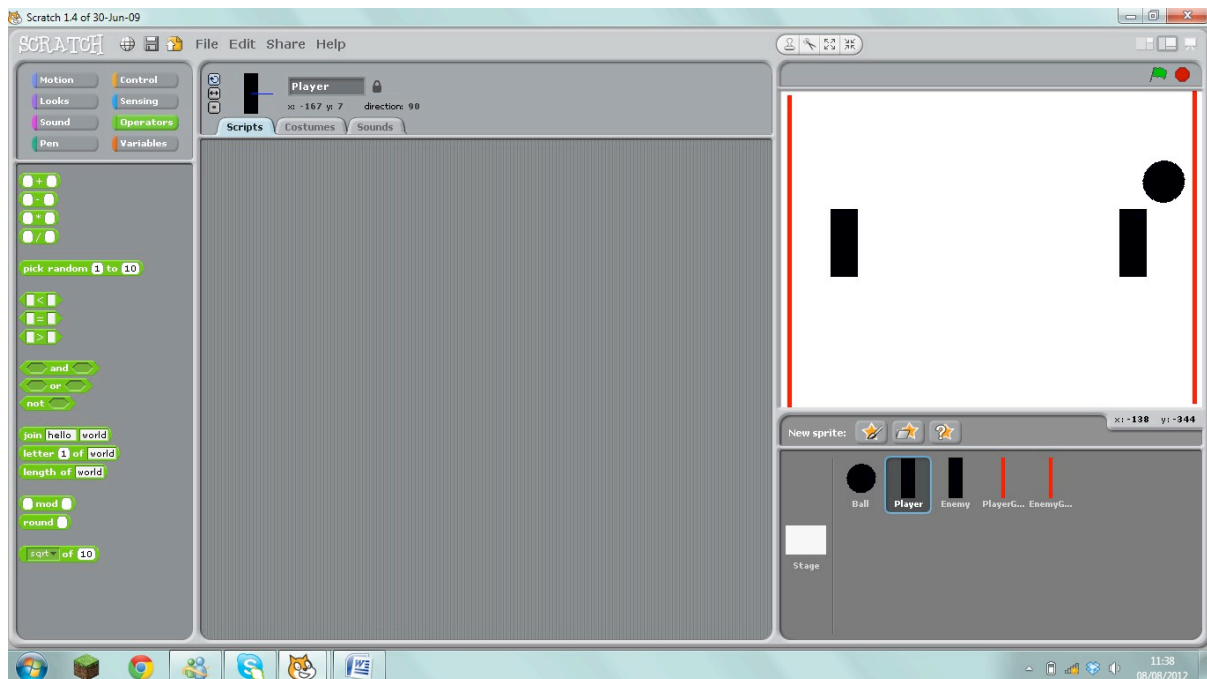

You can hit the green flag to test if this works!

MAKING PONG IN SCRATCH

STEP 4, PLAYER CONTROL!

The player needs to be able to control where their bat is, so now we're going to add that in!

Double-Click on the Player Object to bring up its information screen, don't worry about your Ball's code - its safely stored inside that object.


We're going to now set it so that the user can move the Player object up and down on the Y axis. To do this we have to add the following code from the menus that we have already become familiar with:
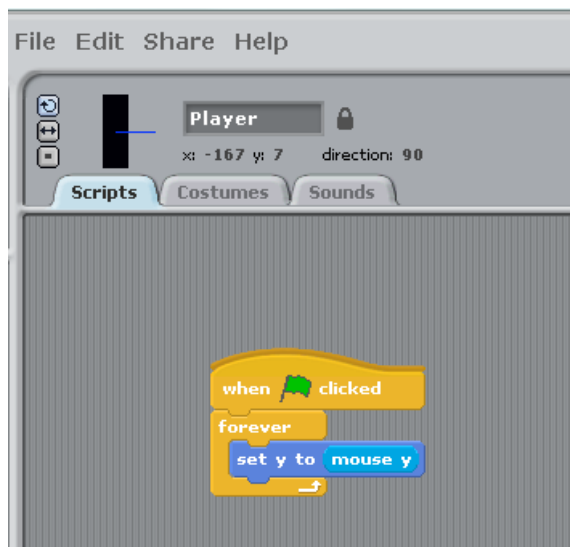

This will mean that the Y position of the Player will always move to the Y position of the mouse, you can test this using the Green Flag.

Artificial intelligence is simply a computer making decisions based on information given to it. In this case, we want the enemy to try and block the ball from getting into its goal. So when the ball is higher than the bat, we want to move it up, and when the ball is lower than the bat, we want it to move down.

We're going to add this now... click on the Enemy object to access its information menu:
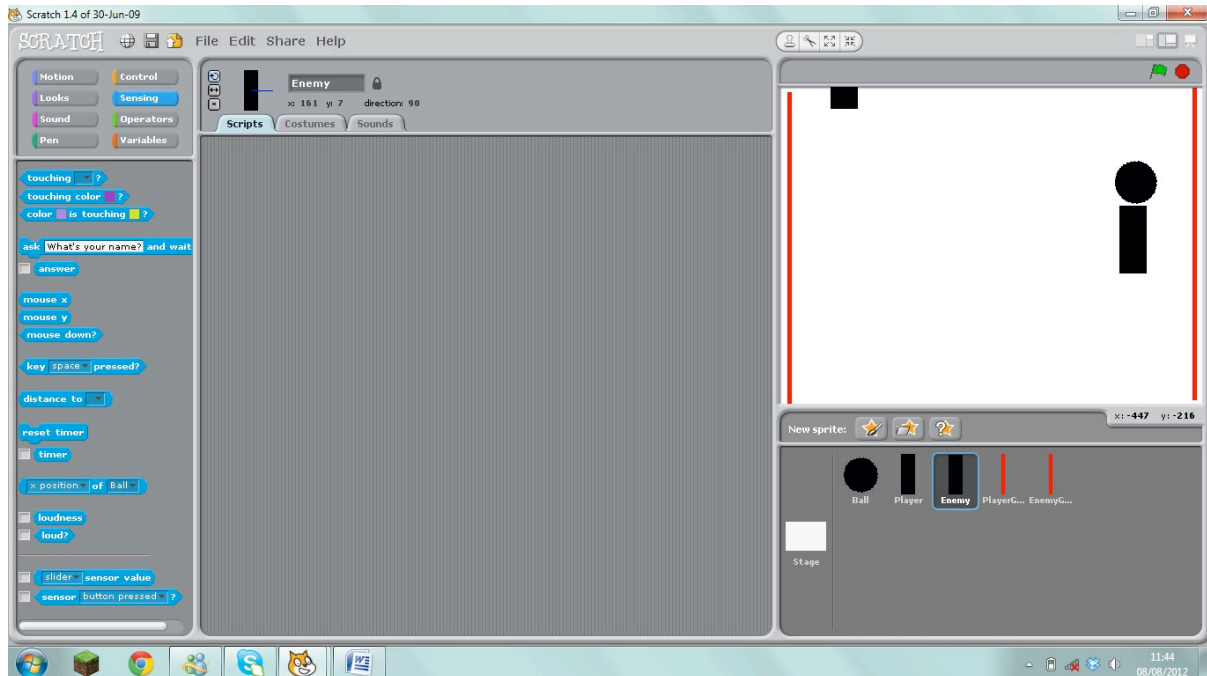
We're going to use two new green 'Operator' menu functions this time around, along with some if statements and a forever loop! The operators we are adding are the LESS THAN and GREATER THAN operators (> and <) these will test if where the position of the ball is relative to the position of the enemy:
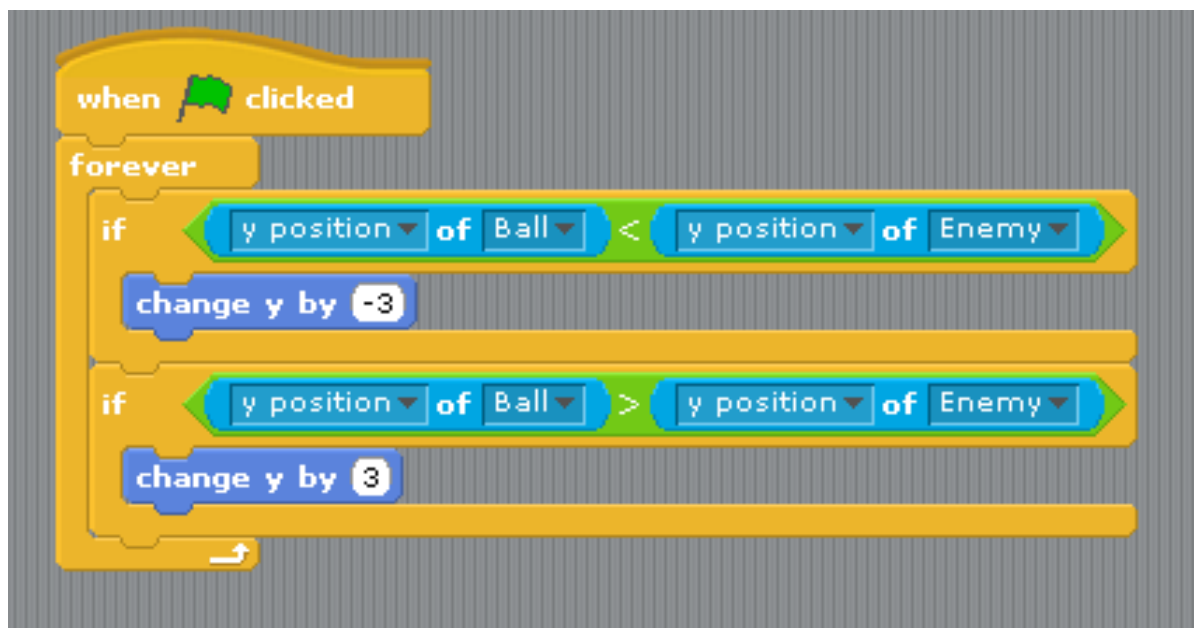

Add that code to your project using the methods you have learnt from the other objects. When the balls y position is < the Enemy, the Enemy moves down. When it is > than the Enemy, the Enemy moves up!

At this point your game works! You can play against the enemy and try to hit the ball into his goal. But you can add in a score to make it more interesting...

STEP 6, VARIABLES

A variable is a way of storing data in a computer, in the case of our game we want two variables to store the score of the player and the enemy.

Go to the red 'Variables' menu and click 'Make a variable':
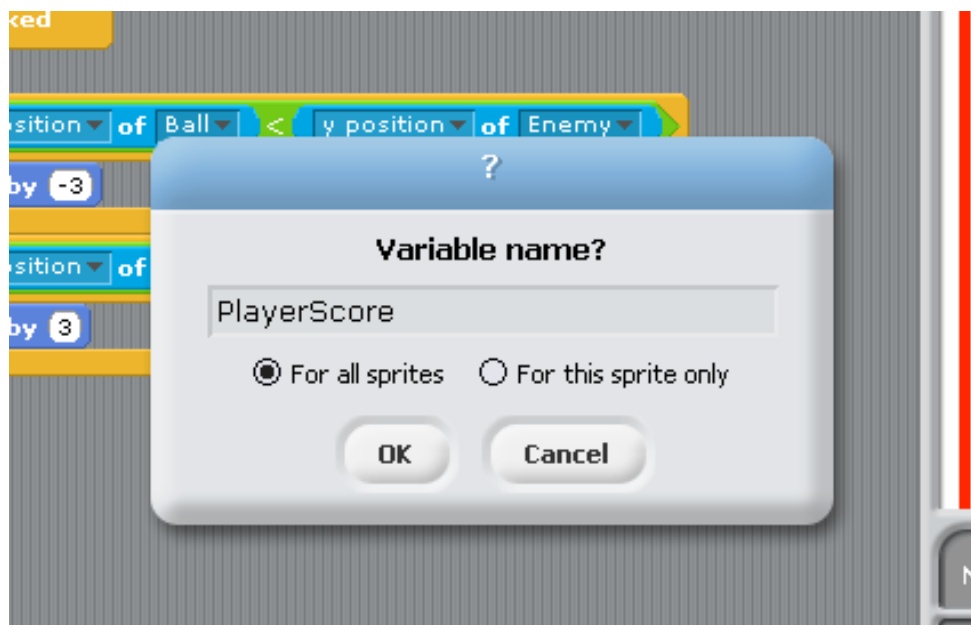

Call the variable PlayerScore, and make it 'For all Sprites' then hit OK. Make another variable in the same way called 'EnemyScore'. You'll also notice that it automatically adds a little counter to your game screen:
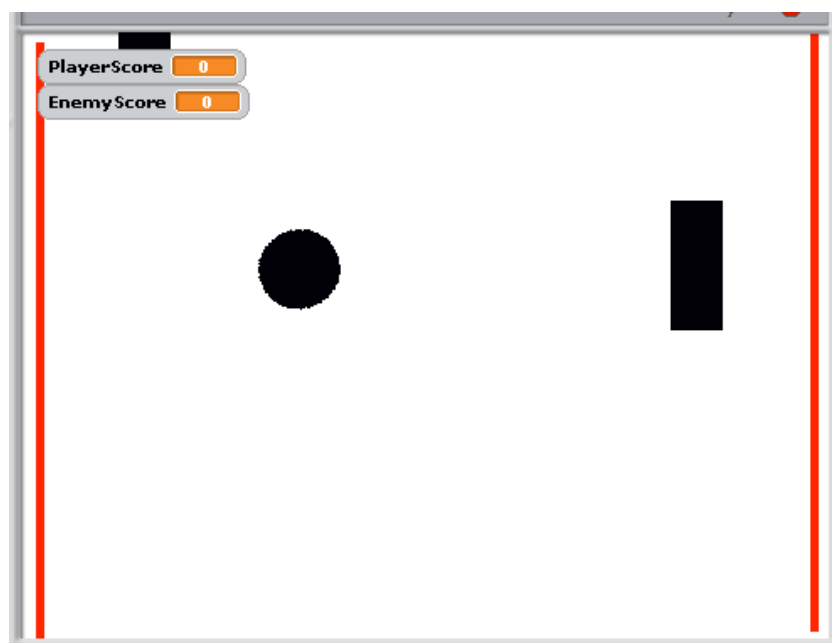
Double click on the ball object to bring its code back up again to your screen. The variables menu has some code snippets in it that we can use to change the score depending on what is happening in the game.

First of all, we need to check that the scores of both players are 0 when the game starts, so drag in the following code snippets:
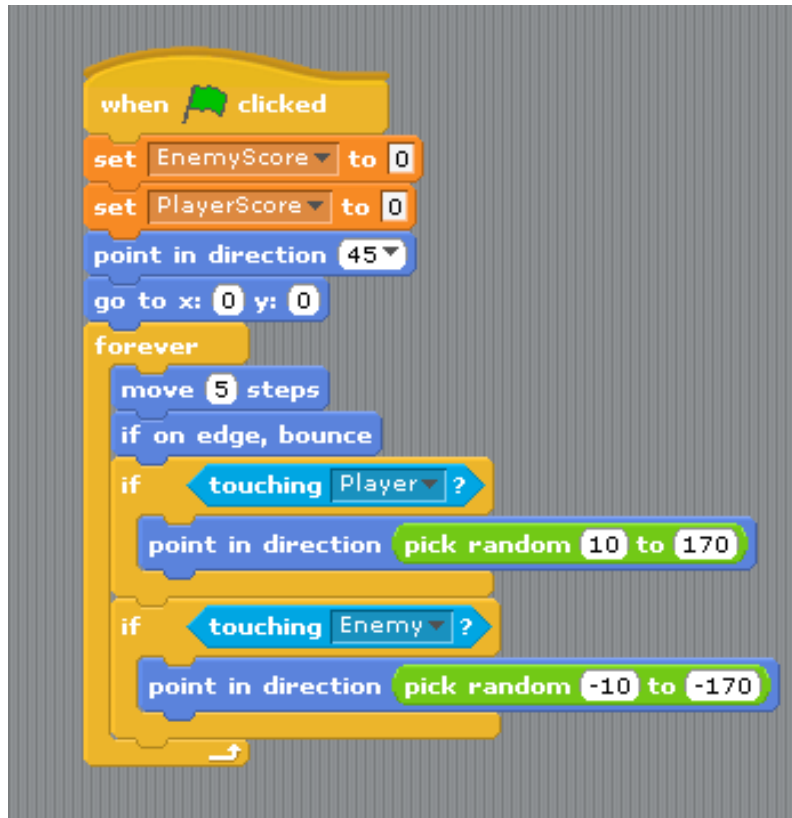
Now we need to test if the ball hits the goal areas in the game, this is done in the same way as checking if it hits the player or the enemy, so add this code to the bottom of balls forever loop:

This code tests to see if the ball hits one of the goals, then changes the score by 1. It then moves the ball back to the middle of the screen and waits 1 second before starting the game again! (This stops the player getting confused when the ball is reset to the center of the screen.)

Finally, add two more IF statements to the bottom of the ball's logic loop, these will check to see if the scores reach 5, and then the game will end! We'll be using the = operator... which checks if an item equals another one:

The 'stop all' function ends the game. But this only happens when one of the scores is high enough!

COMPLETED!

Well done, you have finished all the tasks for this workshop! You can now go back and edit your drawings and make the game look better, you can do this by clicking on an object and selecting the 'costumes' tab:
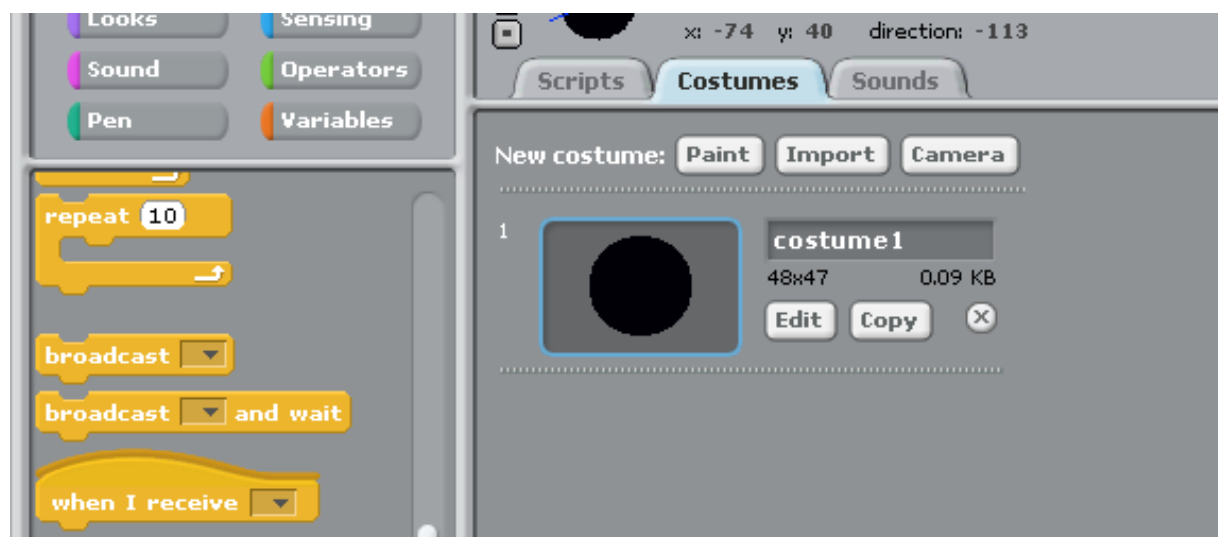
Then you can click edit and change what the object looks like! You've done very well!