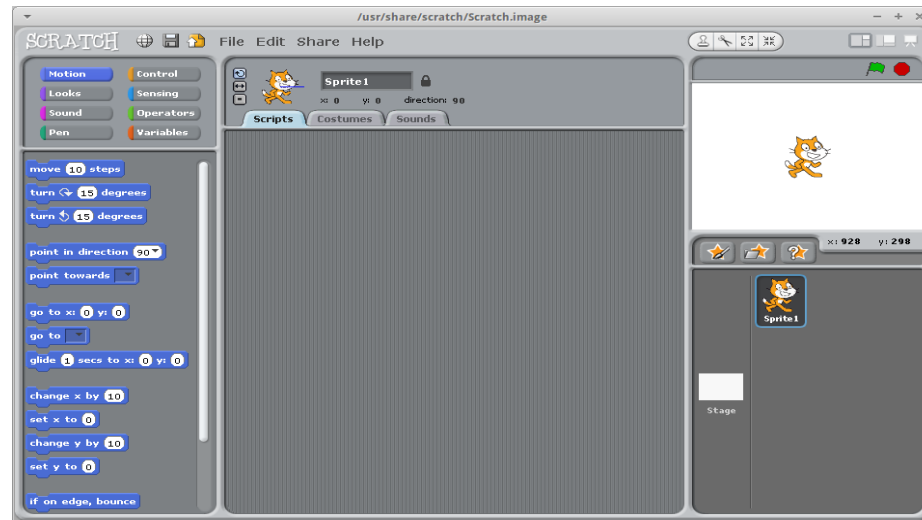


Step 1

Open up Scratch on your computer or online at http://scratch.mit.edu/projects/editor/?tip_bar=getStarted

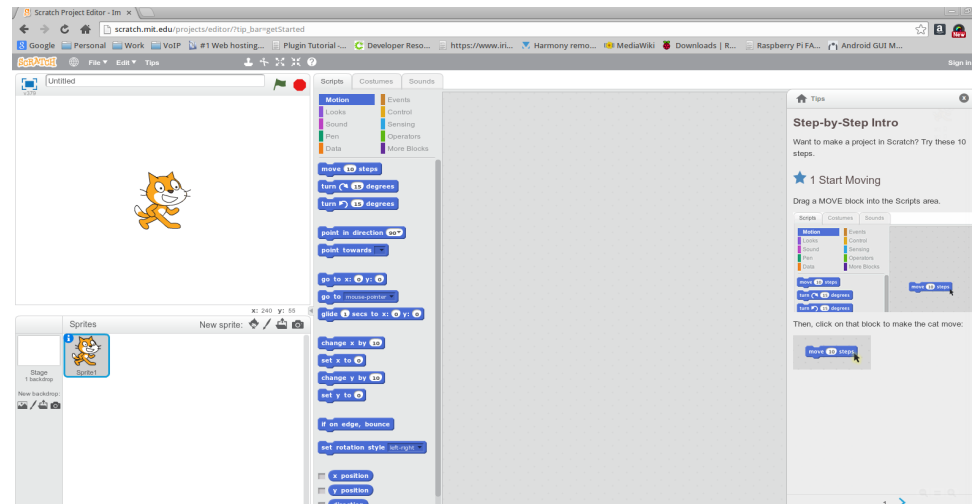
Scratch 1.4 looks like this



Scratch 2.0 Online looks like this

It doesnt matter whether you use Scratch 1.4 or Scratch 2.0 for this exercise but they do look slightly different so please note this.

I will be using Scratch 1.4 for the rest of the screenshots in exercise.



Step 2

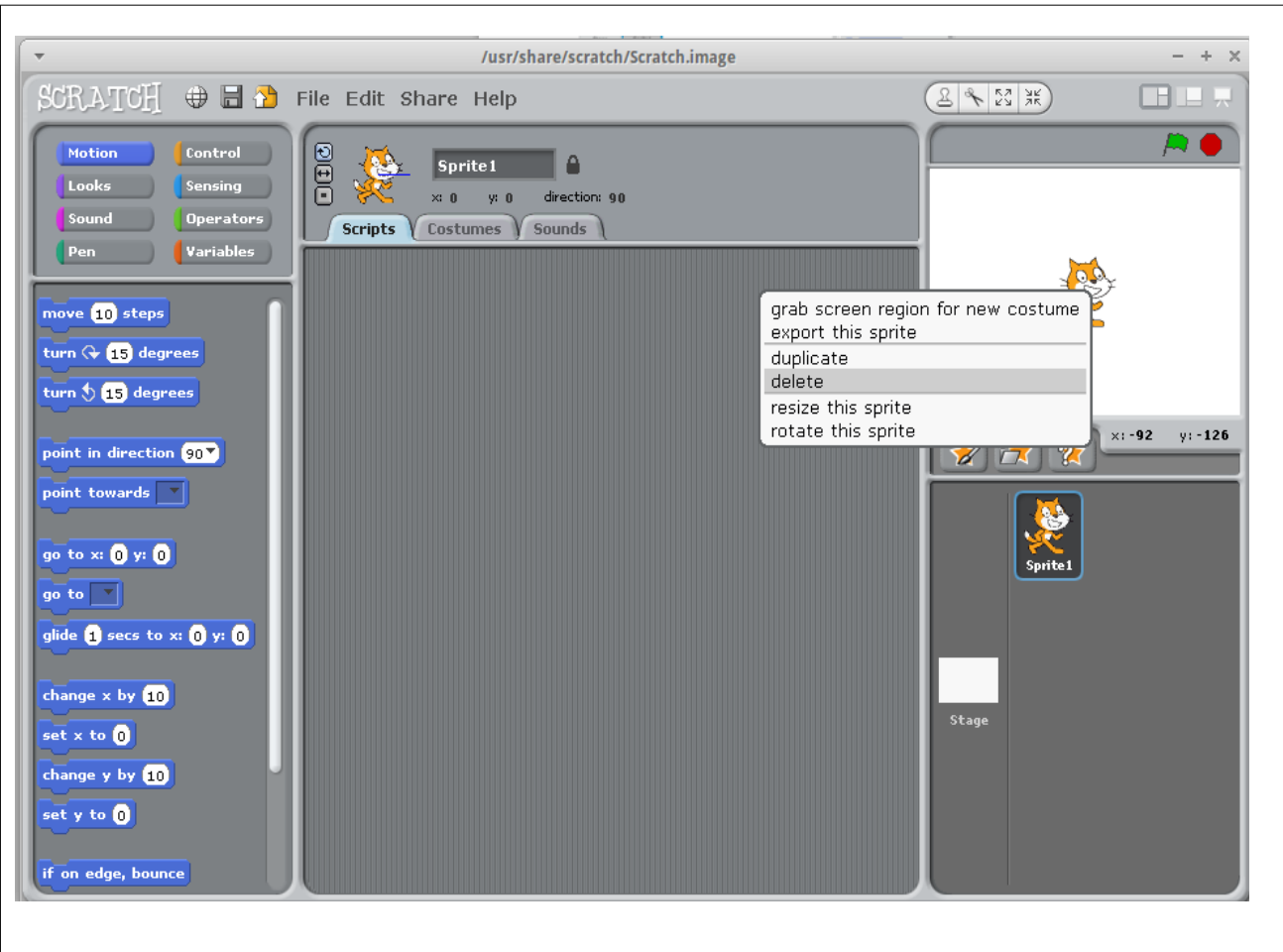
We are going to make the game Pong which is a bit like simple computer tennis. It is one of the very first graphical computer games ever made.

A sprite is like a character in our game. We can make them move and react to different actions from us or other sprites.

We will delete the default cat sprite that Scratch creates and create the sprites that our game will use

Delete the cat sprite by right clicking the mouse on him and then selecting Delete as in the diagram below

He will then disappear!

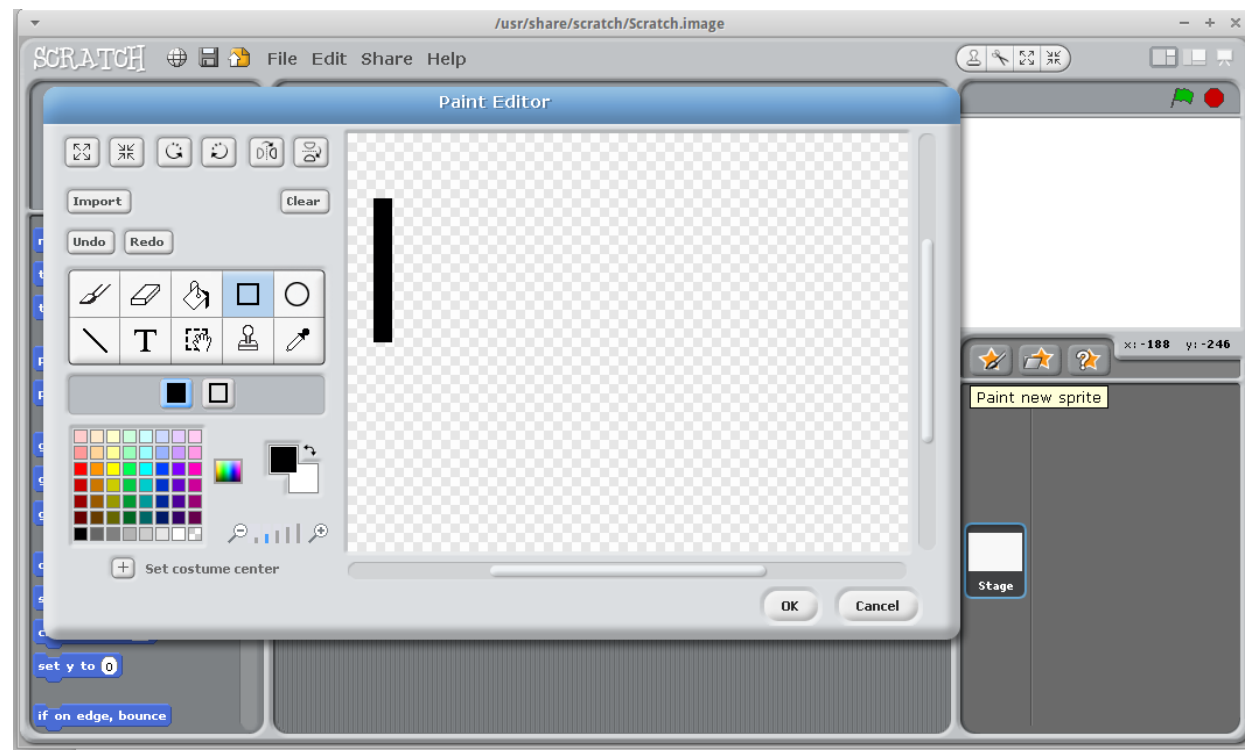


Step 3

We will now create the bat that will be used by each player to hit the ball. Do this by clicking the new sprite button (under the white box where the cat was) and then draw a long black bar using the drawing tool as in the screenshot. Use the rectangle shape to create the bat as its easier than drawing it.

Click OK and the bat will appear on the background as Sprite1.

You can move the bat to the left of the screen by clicking and dragging it where it will be the Player 1 bat for the game!



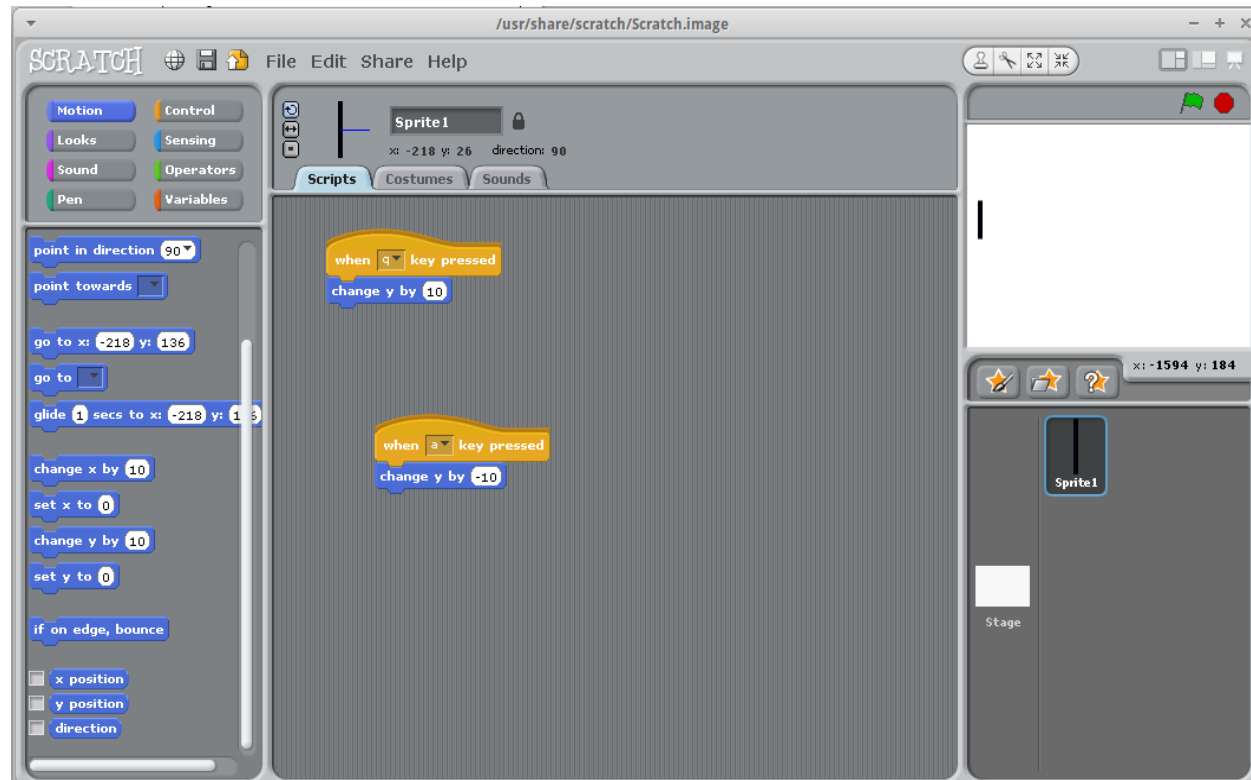
Step 4

Now we will make the bat be controlled by some user action with the keyboard. With Sprite1 highlighted we drag the **Orange** control for “When something is Clicked” onto the script area. We change the Spacebar to the letter Q which will make the bat go up. Then drag the **Blue** Motion for Change Y by 10 and attach it to the **Orange** control.

Change Y by 10 means move 10 pixels up in the Y axis. If we wanted to move right we would have chosen the X axis but we only want to move up and down so Y is what we need.

You can now test your first sprite moving by clicking Q. The sprite should move up!

Repeat this again for the down movement with an **Orange** key press control this time with the letter A and a change y by -10 **Blue** Motion. You click the number to change 10 to -10. -10 means the bat will move down when the letter A is pressed. So now the bat moves up and down when Q and A are pressed.



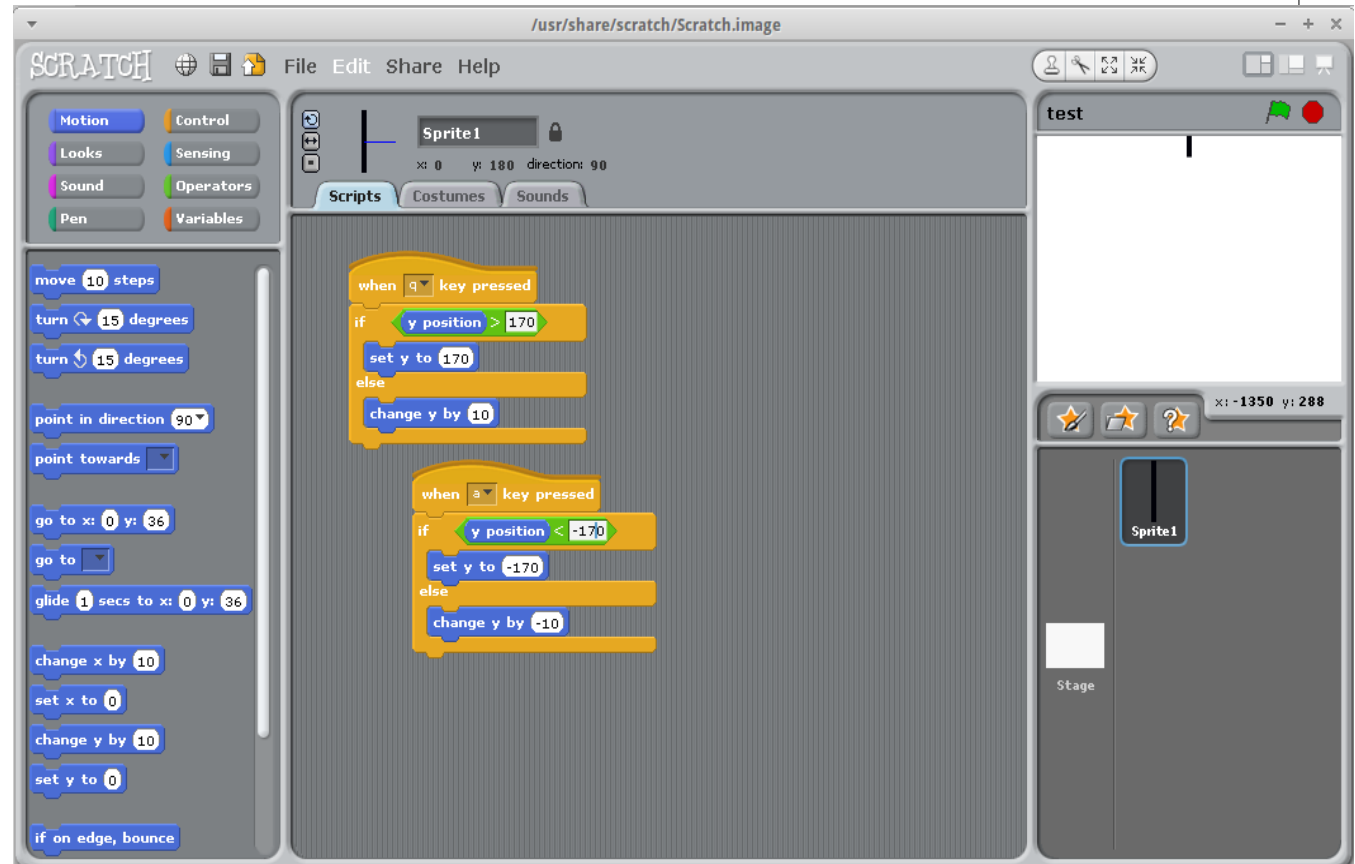
Step 5

You will notice that if you keep pressing Q or A that the bat goes off the screen.

To stop this happening we are going to put an **Orange** “If Else” Control around the Y movement. This allows us to detect where on the screen the bat is and then put it back on the screen

For the Up Q letter movement put a **Green** > than Operator into the box beside IF. Then put the Motion block Y position to the left of the > and 170 to the right. 170 is the top of the screen so when this is true it means the bat is off the screen. The blocks in the first part of the If are executed when the If is true and the part after the else when false. So for this example when True we want the bat to stop going off the screen so we put **Blue** control “Set Y to 170” in this part. We put back in the Change Y by 10 to the else part.

We do the exact same for the A button except with a < **Green** operator and -170 as the bottom of the screen. See screen shot.



Step 6

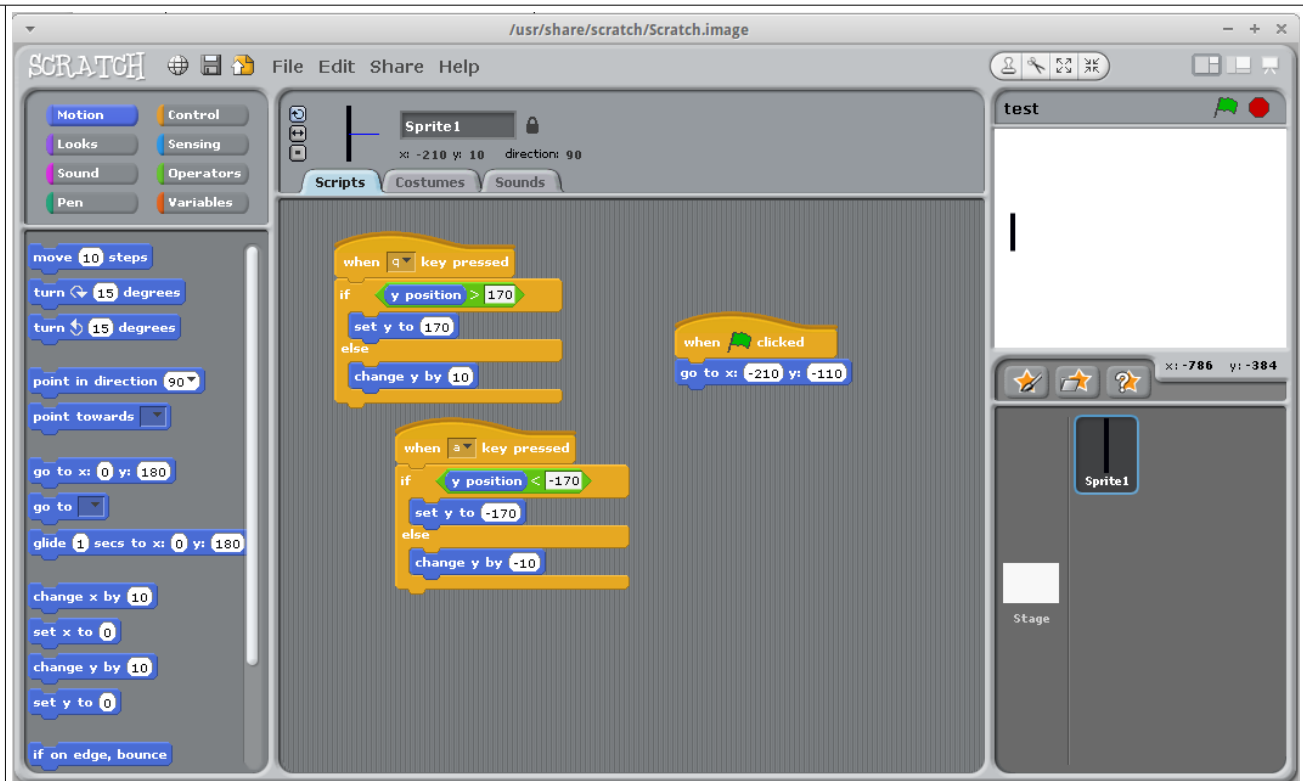
Lets set the start value for the Bat. Add the **Orange** control with the Green flag which gets executed when the game starts.

Noe put the left bat in the starting position you want. Howver your mouse over it and note the X and Y values under the little screen

Then add a **Blue** motion “Go to x and y” and enter the x and y values you noted. I will choose -220 for x and -110 for y for the left bat.

Click the Green flag and then move the bat using Q and A then press the Green flag again and it should go back to starting position.

Now we are finished with the first bat lets rename it to Player1



Step 7

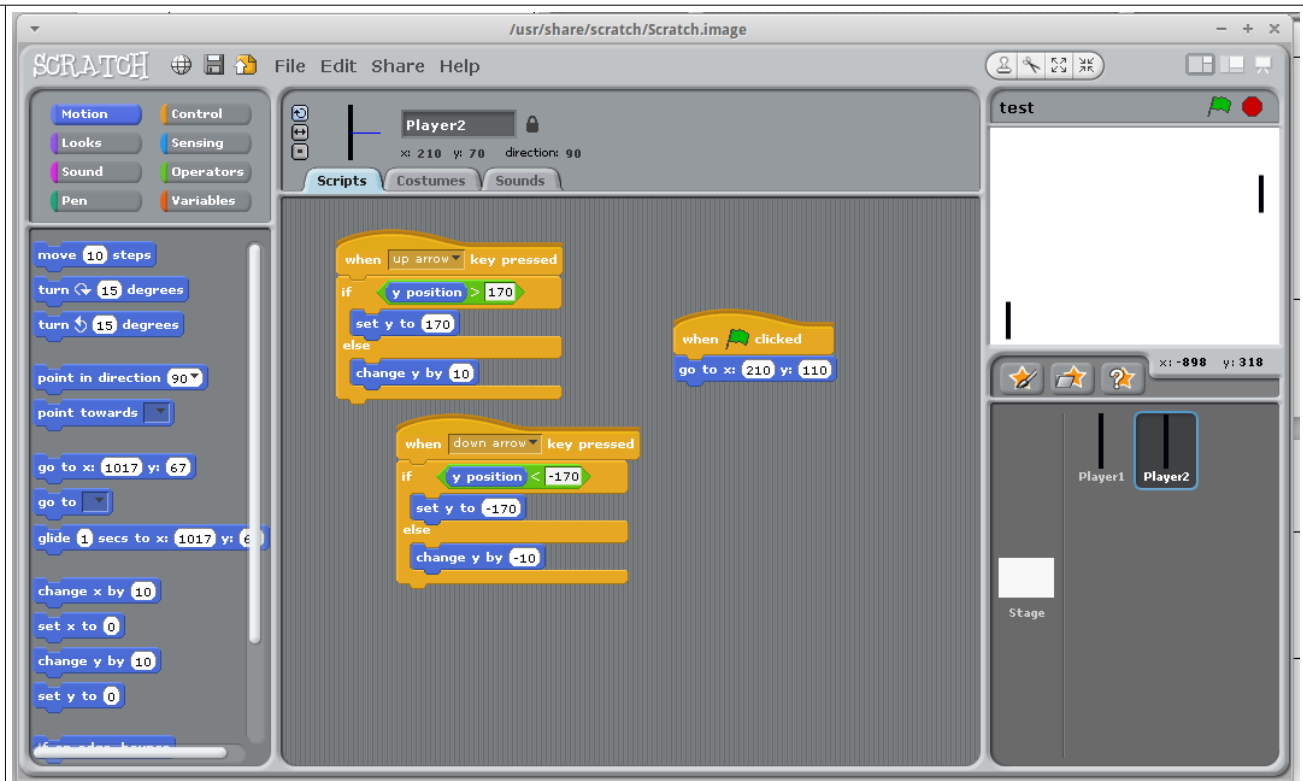
Make a copy of Player1 bat by right clicking on the sprite and clicking Duplicate Sprite. Rename this copied sprite to Player 2 and place him on the right of the screen

We only need to change a few items in the Player2 scripts to make this bat work separately from Player1.

Change Q to Up Arrow and A to Down Arrow

Change the Green flag start location to x 210 and y 110 to be the opposite of the the Player1 bat.

Now both bats can go up and down when their up and down buttons are pressed



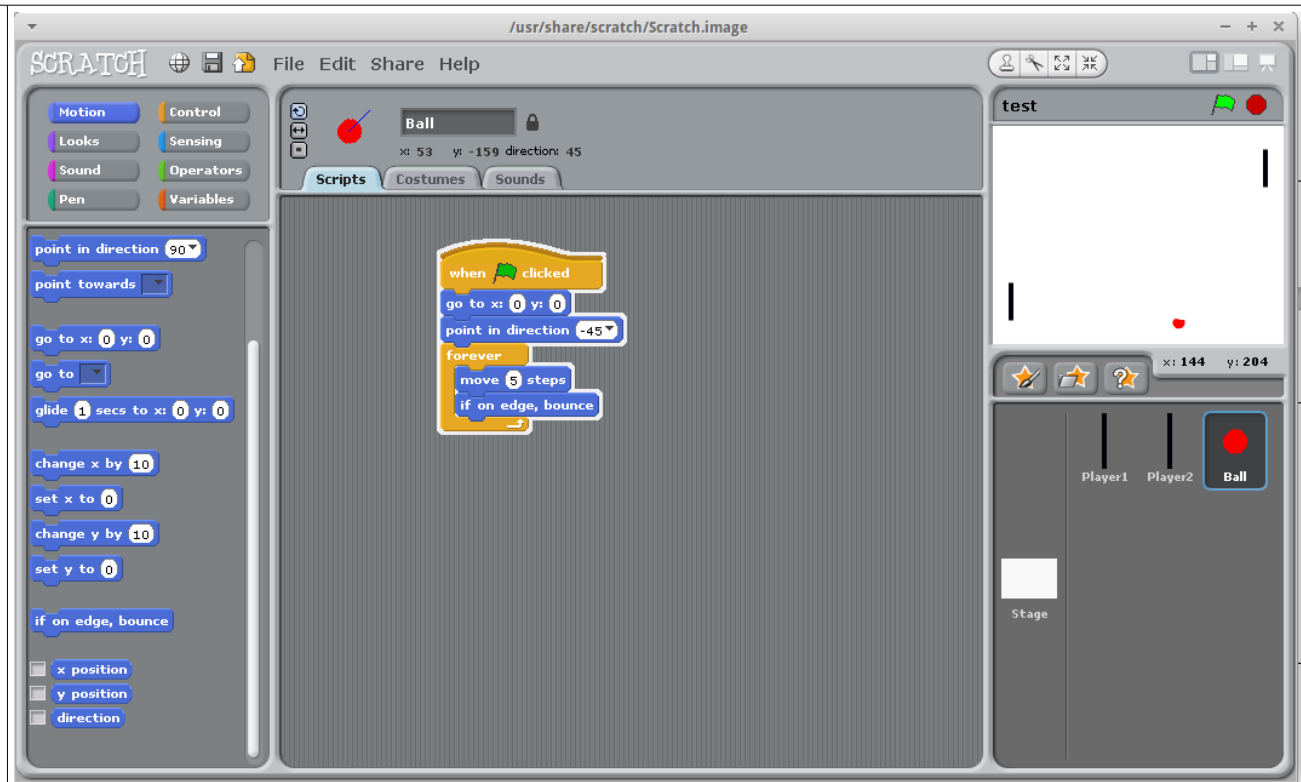
Step 8

Now we will create the ball. Create a new sprite and choose the circle drawing tool and then choose the red color. Draw a circular small ball and click ok. Make the ball quite small.

Now add the **Orange** Green flag control and add a **Blue** motion Goto X 0 and Y 0 underneath so that when the green flag is pressed it starts in the middle. Also lets make the ball point at an angle by putting a **Blue** Motion "Point in Direction -45 degrees" under the Goto XY.

Then add an **Orange** Forever control and put a **Blue** motion Move 5 steps withn the forever control. Under the Move 5 Steps add a If on Edge Bounce control.

Now press the Green flag and the ball should move around the screen bouncing off the walls.



Step 9

Now we want to make the bats bounce the ball. Add an **Orange** If block with a **Cyan** Sensing “If Touching Colour” block as the condition. Put black as the colour.

Then within the IF block put a **Blue** motion “Turn by 90 degrees” to make the ball bounce (like the **Blue** “If on edge bounce” but you can see that for the bats as they aren't a edge)

The screenshot shows the Scratch IDE interface. The main workspace displays a red dot representing the 'Ball' sprite at coordinates (50, 138) with a direction of 45 degrees. The Scripts area contains the following code:

- when green flag clicked
- go to x: 0 y: 0
- point in direction -45
- forever loop:
 - move 5 steps
 - if on edge, bounce
 - if touching color black?
 - turn 90 degrees

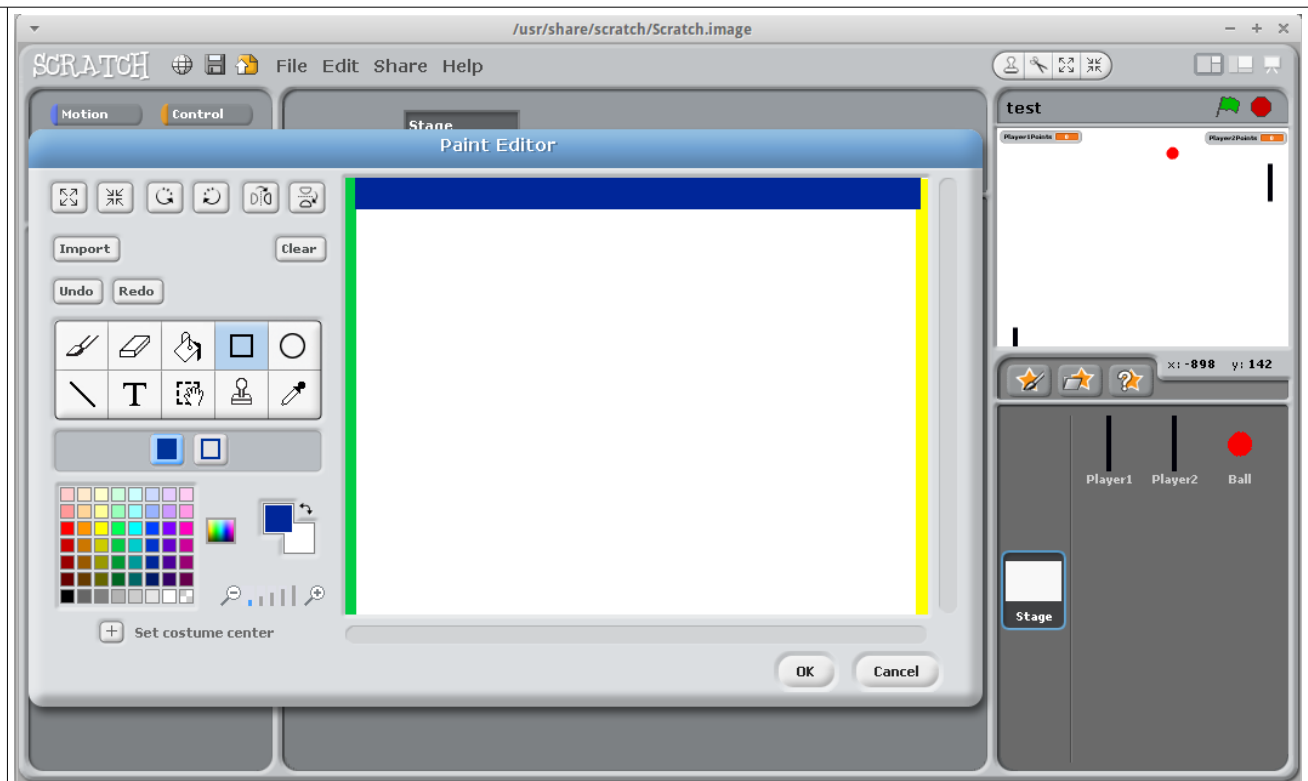
The left sidebar shows the 'Sensing' category selected, with the 'If Touching Colour' block highlighted. The right sidebar shows the 'test' area with a red dot on the stage and a 'Ball' sprite in the bottom right corner.

Step 10

Next we want to add scoring. So we go to the **Dark Orange** Variables section and add 2 variables – Player1Points and Player2Points. Move the labels on the screen to the left and the right as this is where we will track scoring.

We are going to check scoring by checking if the ball goes behind the bat and hits the wall. To do this we make each back wall a different colour. You do this by clicking on the Stage and then go to Background and edit. Then add the 3 rectangle colours like in the screenshot.

Click OK and then move the Points labels into the blue rectangle.



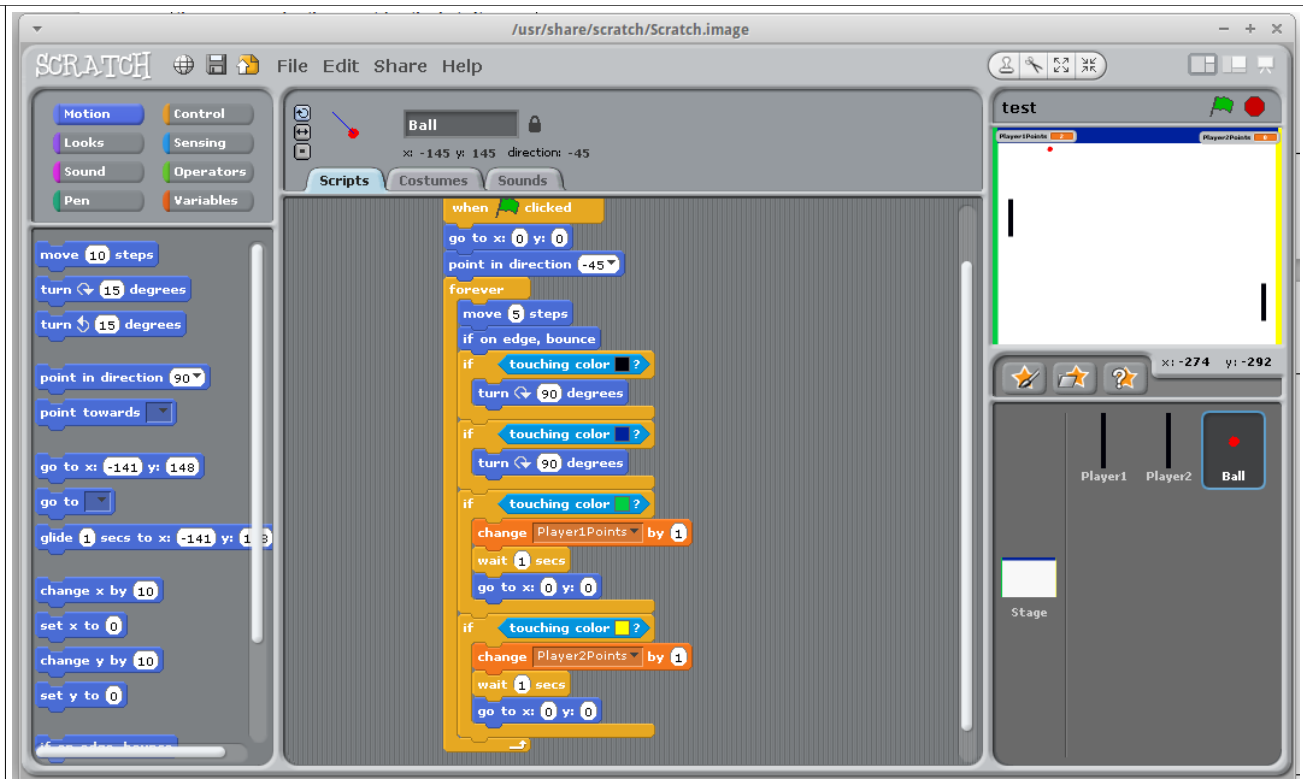
Step 11

Next we go back to the Ball script and add the bounce logic for when it hits the blue part of the screen under the script for the bat. Its exactly the same so you can just right click the block and choose duplicate. Then just change the colour to to the blue by clicking the colour and then use the colour dip to get the blue colour.

Next we add the points logic. Put another If block under the last and another Touching Colour this time with the Green wall colour. Go to the **Dark Orange** variable section and choose “Change Player1Points by 1” and put it in the IF statement. Then add an **Orange** “Wait 1 Seconds” and a **Blue** Motion “Goto X0 Y0”. This will bump the points by 1 and put the ball back in the middle.

Copy this block for the other wall replacing the colour and Player1Points for Player2Points.

Now the balls should bounce of the top and bottom walls and the bats. When it hits the back walls a player gets a point!



Step 12

The game now works but we will add a few final tidy ups to make it even better.

Replace the Forever script block in the ball with a **Orange** Repeat Until with a **Green** OR operator. In the OR operator we will check if either Player1Points = 5 or Player2Points = 5 see screenshot using 2 **Green** = operators.

When this condition is met someone has won so we will check who it is by adding an **Orange** If Else block underneath the Forever Until checking if Player1Points =5 and if so then getting the Ball to say “Player 1 Wins!” using a **Purple** Sensing Say block. If its not Player 1 then its Player 2 so the Else says “Player 2 Wins!”

Also add a **Dark Orange** Set Player1Points to 0 before the Repeat until block and the same for Player2Points.

Also, move the “Change Player1Points by 1” after the Goto x0 y0 as it stops points being added twice. Same for Player2Points

The game is now fully working.



Step 13

Some expert steps for those who want to make the game even more challenging. We can make the ball go faster everytime there is a score.

Add a variable speed and then before the Forever loop starts Set Speed = 5 using the **Dark Orange** block. Change the Move x by 5 to Move x by Speed. Add a **Dark Orange** block Change Speed by 1 when there is a score for Player 1 or Player 2.

You will notice the bats struggle to keep up. This is because there are too many forever loops going on between the bats and the ball. If we just have one forever loop in the ball and use that to send a message when buttons are pressed it works much better. This is done using **Orange** Broadcast and Receive. In the ball Forever loop we add Broadcast Bat1Up, Bat1Down, Bat2Up and Bat2Down for each of the key presses. Then in the Bats we get rid of the loops and replace with “When I Receive” the messages above.

This is better code and the game works better

