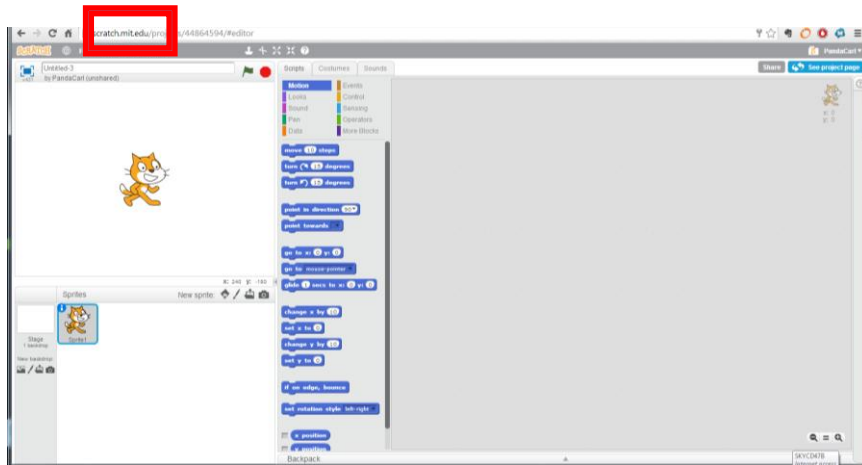


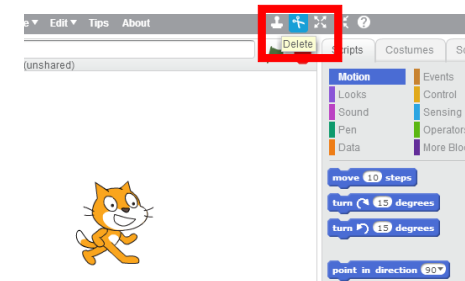
## Let's Open Scratch



- Open Scratch. Go to the website
- <https://scratch.mit.edu>
- If you don't have an account, create one, or just login in
- Once you login in...
- If you have a cat, then you are on the right track!

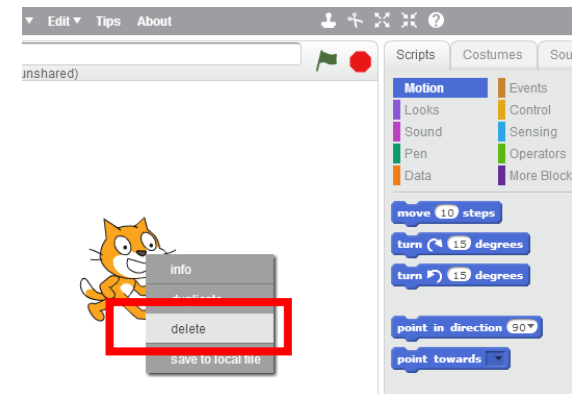
## Get rid of the cat

Click on the scissors, and when your cursor turns into scissor, then click the cat.



OR

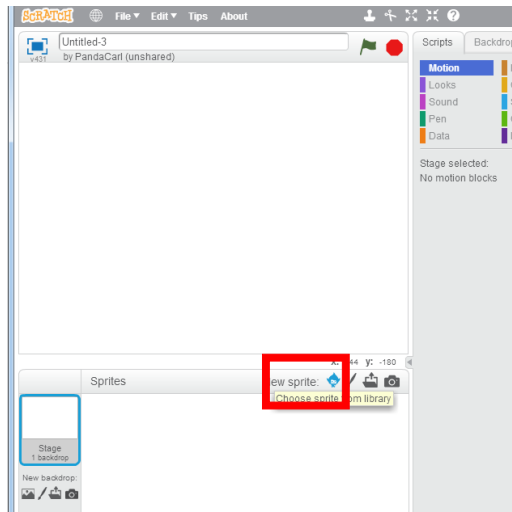
Or right-click the cat, and click delete.



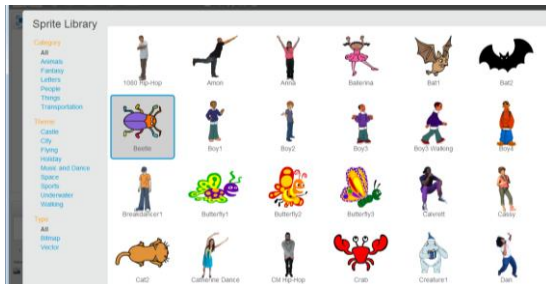
Remember these ways of deleting sprites (images) as you will be needing them in your projects.

# Add Sprite

To add a sprite (we have used Beetle, though use any character you want) import the image from a folder. Click on “Choose sprite from library”.



And then double click on the character you want.



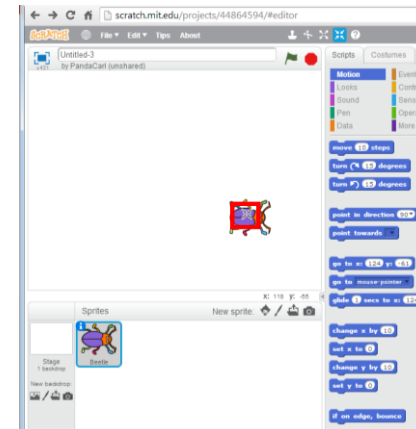
# Resize your Sprite

Is your Santa too small?  
Is your Santa too big?

You can easily **grow** and **shrink** your sprites using the shrink/grow buttons.

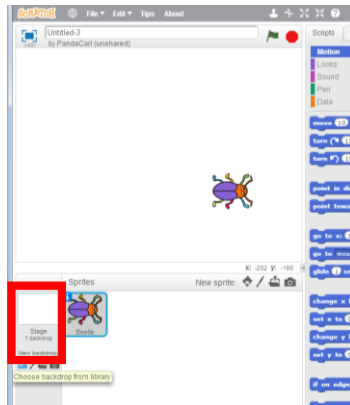


Shrink your sprite so it is a lot smaller

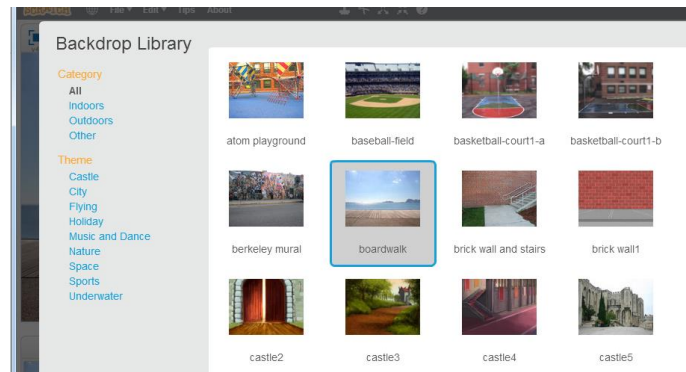


# Add a Background

Click on the Background icon “Choose Background from library”



Pick a background and then double click



# Event Handling (Controlling stuff)

We want to control our sprite using the arrow keys

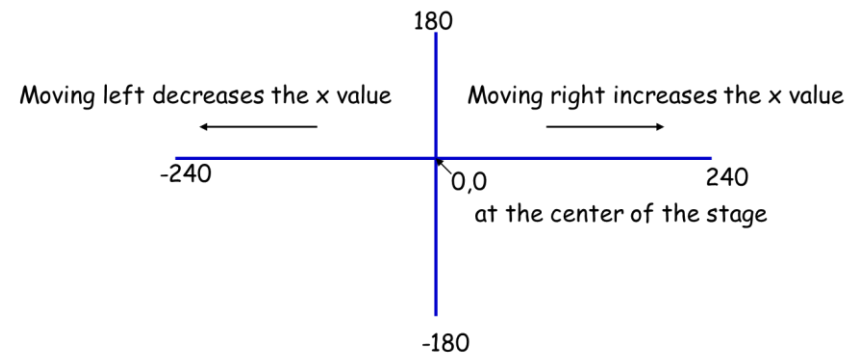
- When we click the left arrow your Sprite should move left
- When we click the right arrow your Sprite should move right

This is a form of event handling

- Responding to user actions like mouse clicks and key presses

## The Scratch Stage

The Scratch stage is 480 pixels wide and 360 pixels high.



# Make your Sprite move

Let's make your Sprite move by using the right and left arrow keys on your keyboard.



Choose event needs to take place.



Choose the steps that takes place when the event occurs

BUT WHAT HAPPENS WHEN THE SPRITE GOES TOO FAR ON THE SCREEN?

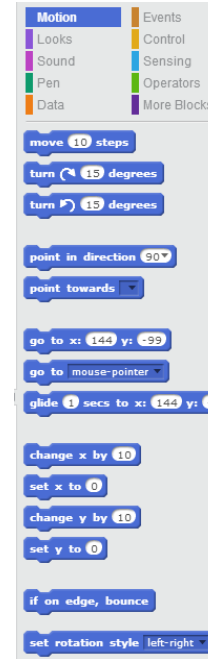


# Understanding Scripts

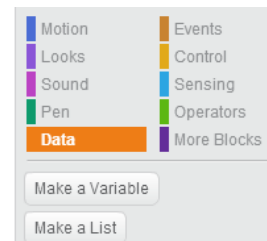


A script is a collection blocks that all interlock with one another - they determine how sprites interact with each other on the stage.

For this game we use 6 different types of Scripts:



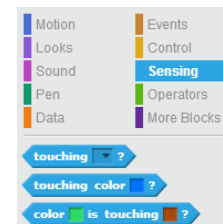
**Motion:** what direction do you want your Sprite to take?



**Data:** when you need to add a variable to your game



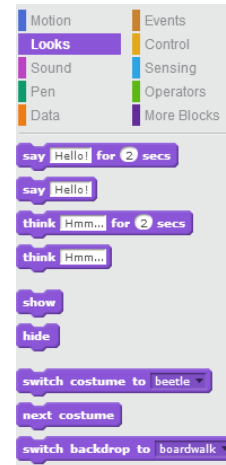
**Operators:** what math operations do you want your Sprite to do?



**Sensing:** what do you want your Sprite to do when it touches something?



**Events:** what event needs to take place for a follow on action to take place?



**Looks:** how do you want your Sprite to look?

## Make your Sprite stay on the screen

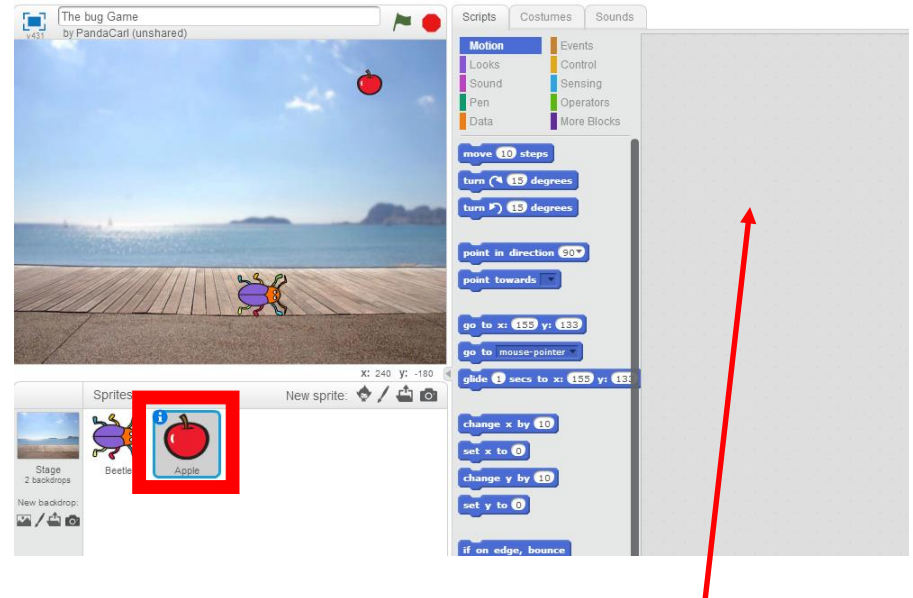
Make it not go over the stage's limits



Set point in direction to 90, so it doesn't flip over.

## Add the moving objects

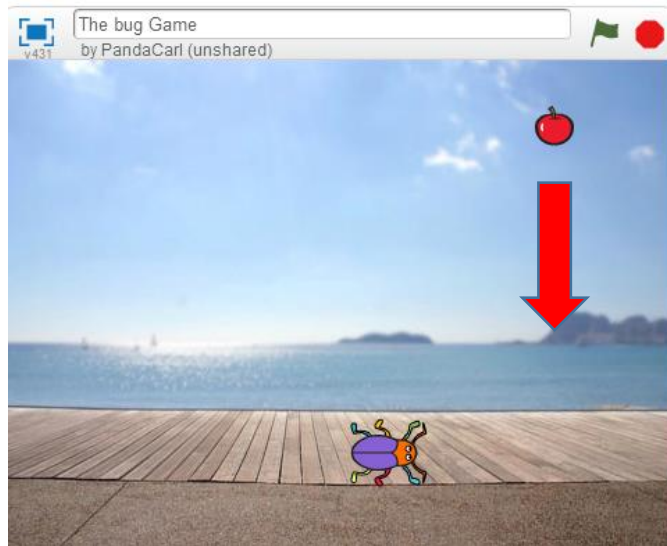
- Adding the moving object is the same as adding the Beetle. It is also a sprite (we have used an apple because Beetles love to eat apples)
- Resize the object so that it is at a good size.



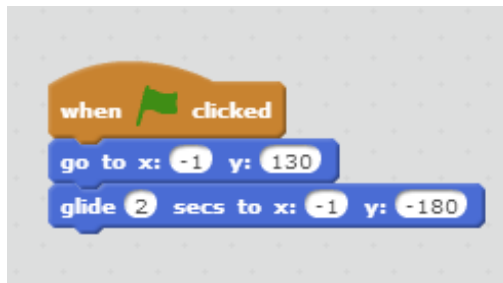
You will also notice that there are NO instructions for the new Sprite

# Make it Fall

We want the apple to fall down whenever we click the green flag.

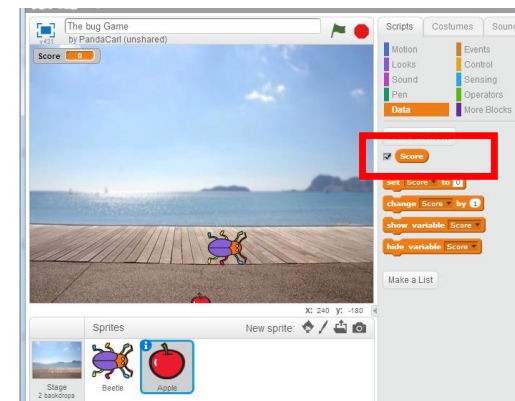
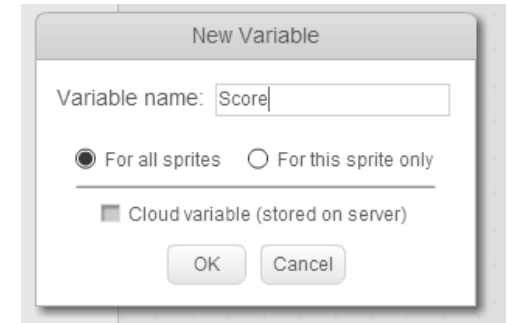
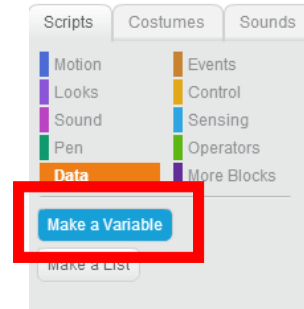


Stay on the same value of x as you started “x = -1”,  
then drop to “y = -180”



# Keep track of the score

Variables are values that can be changed.  
We will use a variable to keep track of our score.



# Scratching our heads...

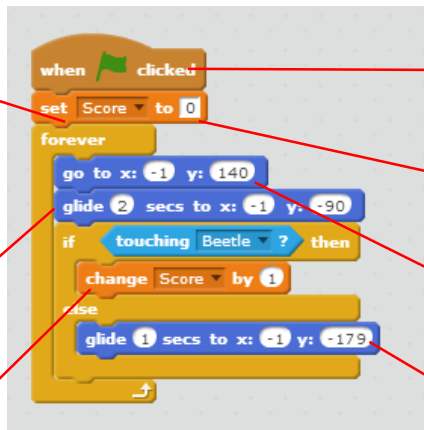
Setting the start score to 0

When we start our game we have to set its score to default of 0.



Know when the Sprite catches the present

To check if the Sprite caught an object we have to use a condition and collision, see below



Forever Loop will run the blocks inside of it until you stop your game (press stop).

Glide until it reaches the Santa's hat height

Condition if present touches Santa (Sprite1) then add 1 to the score

Starting our game (Clicking the green flag)

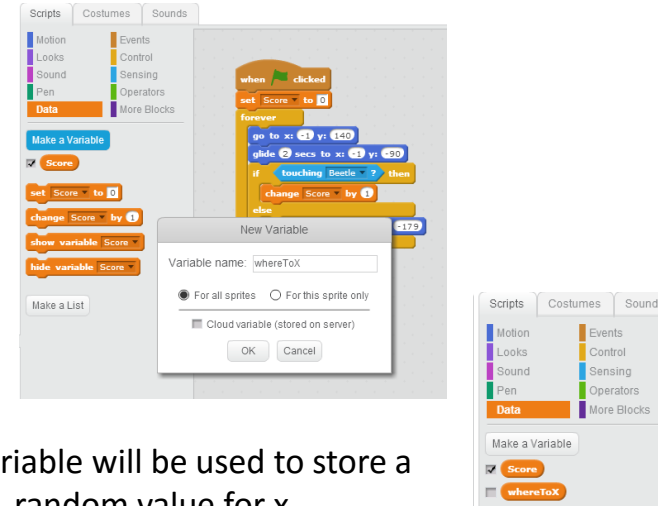
Setting the score to 0

Make our present appear at x = -1 and y = 140

If it does not (else) touch, then glide out of the stage. (y = -200)

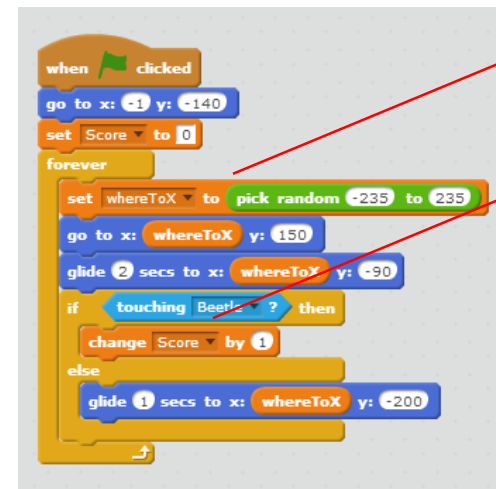
# Make it random...a bit harder

The game is no fun if the present keeps falling on the same spot. Let's add something called a random variable for our value of x.



This variable will be used to store a random value for x.

And therefore we can create the following

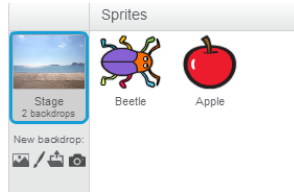


This is where we get the random value for x

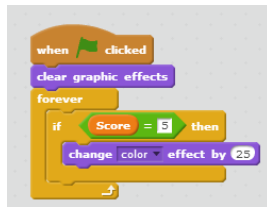
This is where we use the value for x (whereToX).

# Winning

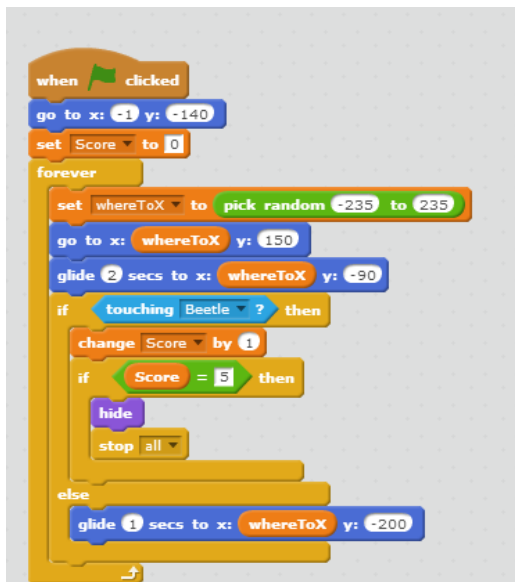
A good way to win is if the player has successfully caught 5 bugs. This involves an if statement in the stage's script window – the background image – you need to click on it



Then create the following script

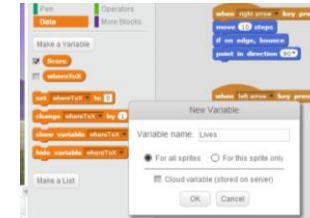


Then create the following script by going to the Apple sprite



# Losing

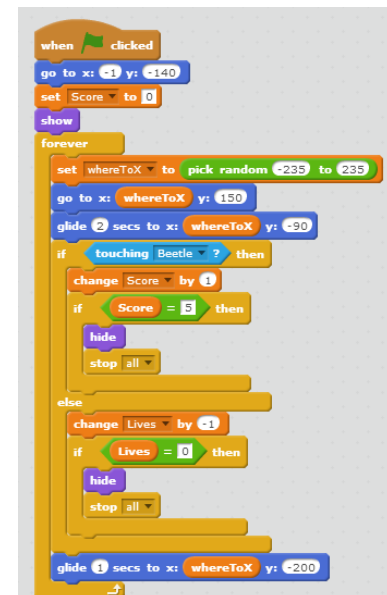
We could program the losing as having lives. If the player does not catch 3 bugs then they lose. Or how many do you think is far? Let create Variable called "Lives"



Then create the following script on the background image



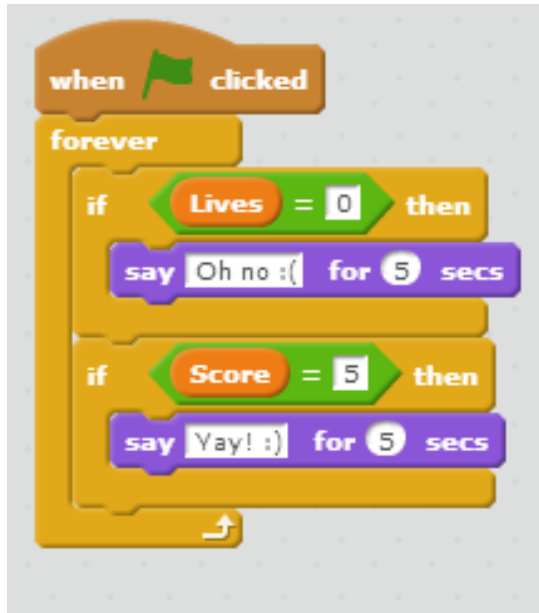
Then create the following script by going to the Apple sprite





## Making our Beetle Speak

Let's make our Beetle say something whenever the user wins or loses.



What else can you make your Beetle do when you win or lose?

Losing