

Getting Started With Scratch



Nick Grattan

Kinsale CoderDojo



<http://kinsalecoderdojo.wordpress.com>

<http://coderdojo.com>

Getting Started With Scratch

Contents

Session 1 - Getting Started.....	4
Moving the Cat.....	4
Loading and Saving your Project.....	10
Session 2 - Broadcasting and Costumes.....	13
The Green Flag.....	16
Taking Decisions.....	18
Detecting Touching Sprites	18
Broadcasting Messages.....	23
Costumes.....	26
Session 3 - The Fish Game.....	30
Creating the Shark.....	30
Creating the YellowFish Sprite	34
The Shark Eats the Yellow Fish!	35
Adding Sound	39
Variables and Keeping Score	40
Adding Script Blocks to the Stage	42
Recreating the YellowFish.....	43
Creating more fish!	45
Setting the Stage's Background picture	46
Other Things to Try	47
Session 4 - Dragon Graphics!	48
Drawing a Circle	48
A Circle of Circles!	50
Squares!	51
Spirals – A little more advanced	52
Session 5 - It's a Dog's Life! - Part 1	55
It's a Dog's Life	55
Add Variables	55
The Engine.....	55
The Dog Sprite.....	59
Dog bowl Sprite.....	61

Getting Started With Scratch

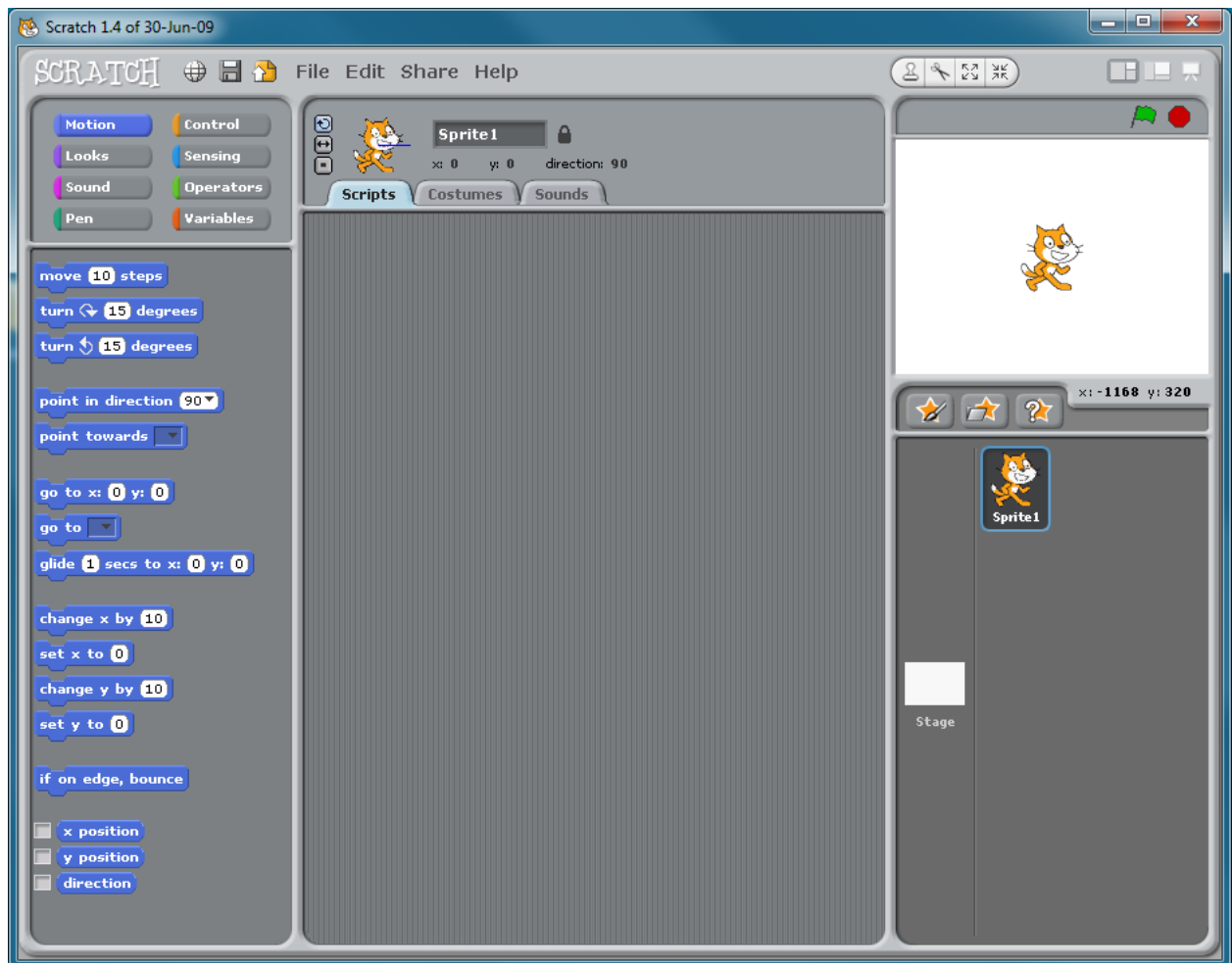
Other things to try	61
Session 6 - It's a Dog's Life! - Part 2	62
How long does your dog live?	62
Bring the dog to feed	65
Session 7 - Scratch Xylophone Part 1	70
Selecting Instruments	70
Add the keys.....	71
Session 8 - Scratch Xylophone Part 2	76
Add the Recording / Playback Buttons	76
Add Recording and Playback.....	79

Getting Started With Scratch

Session 1 - Getting Started

First, download Scratch from http://info.scratch.mit.edu/Scratch_1.4_Download. There are downloads available for the following operating systems: Mac OS X, Microsoft Windows and Debian/Ubuntu. Follow installation instructions appropriate to your operating system.

Run Scratch and the following application window is displayed:



To get started, you need to know about:

- **Sprites:** These are things that do stuff, such as move about, change shape and make noise. When you start Scratch, one sprite is created for you - the Cat.
- **Stage:** This is where the sprites live. It starts out as being a white background. You can set a picture for the stage's background or draw your own background.
- **Blocks:** These are used to make sprites do things, such as moving around, making noise or doing other things.

Moving the Cat

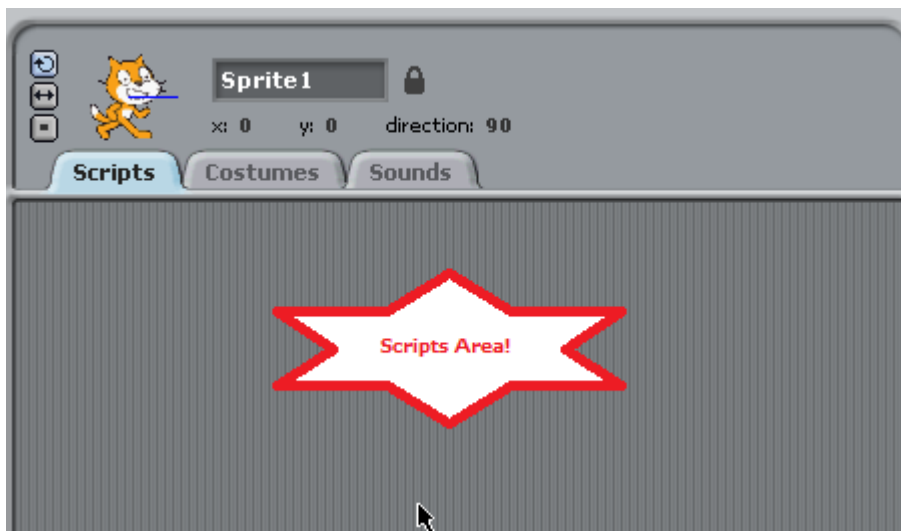
Let's start out by making the Cat move around when the keyboard arrow keys are pressed. To do this you will need to add some code blocks. First, make sure the Cat sprite is selected:

Getting Started With Scratch

1. Single-click the Cat sprite icon:



The Cat sprite will be selected in the Scripts area and you're ready to program!



You now need to select blocks which will do this:

- When an arrow key is pressed....
- ... Move the Cat in the right direction.

You will need to do this four times - once for each of the left, right, up and down arrow keys. Let's start with the up arrow key. First, we need a block that says "when up arrow key pressed":

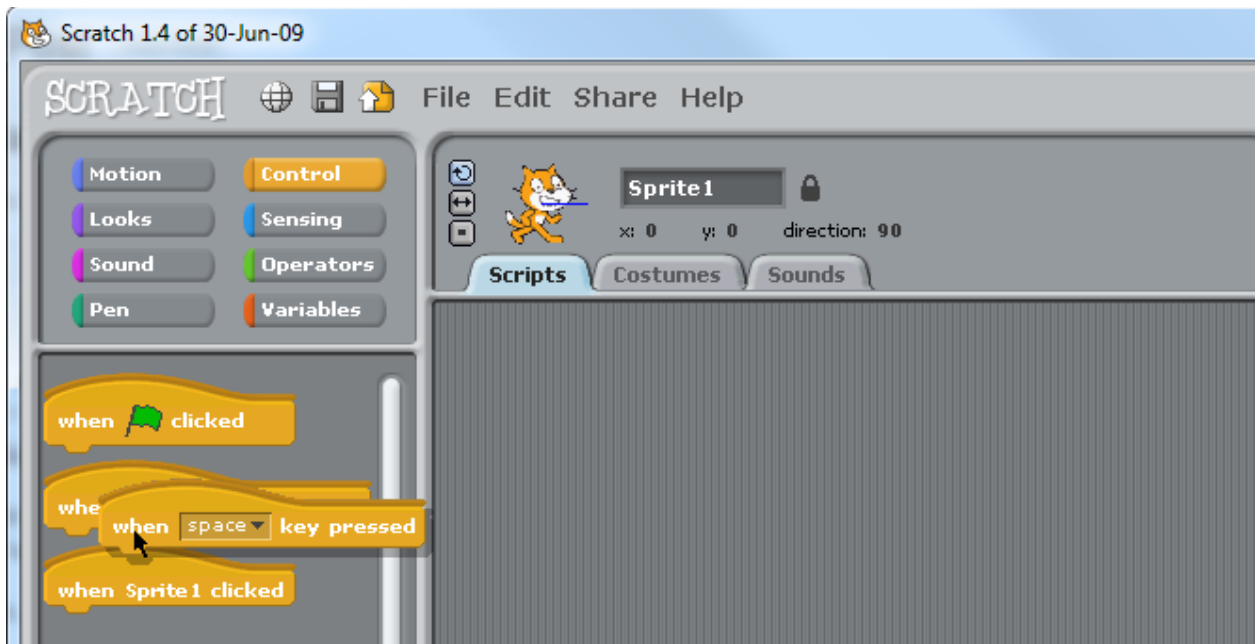
2. Click the **Control** block section:



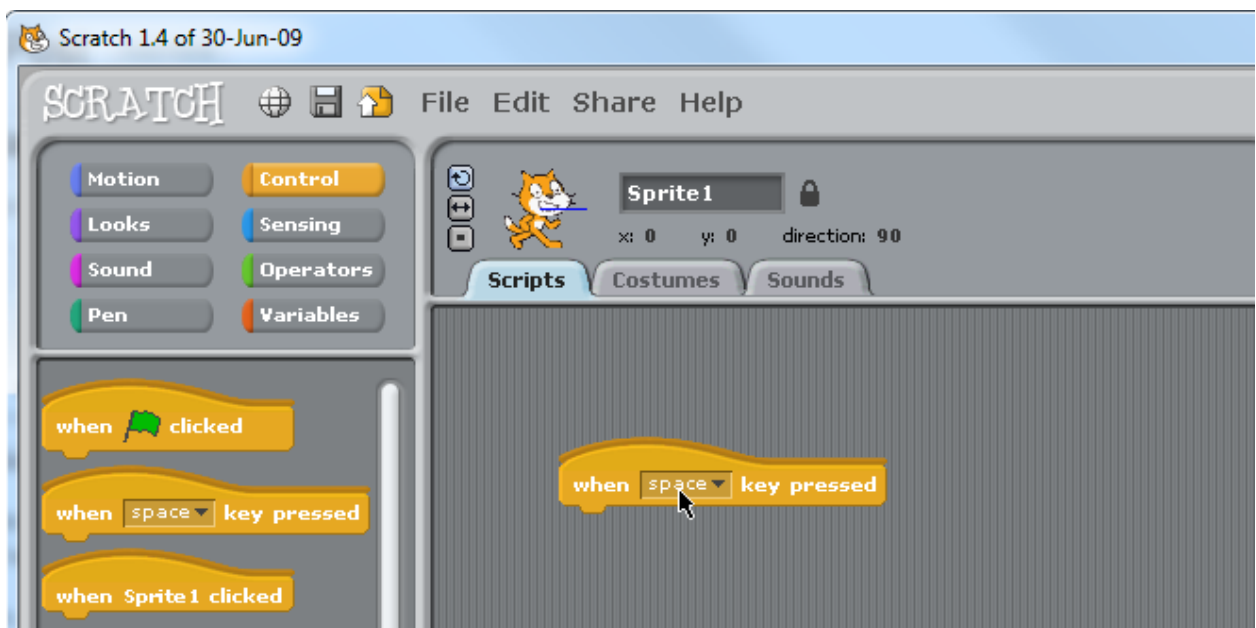
You will now see Control blocks, such as "when space key pressed" and "when sprite 1 clicked". You need to drag the "when Space key pressed" block from the blocks list to the scripts area.

Getting Started With Scratch

3. Drag the **when space key pressed** block ...



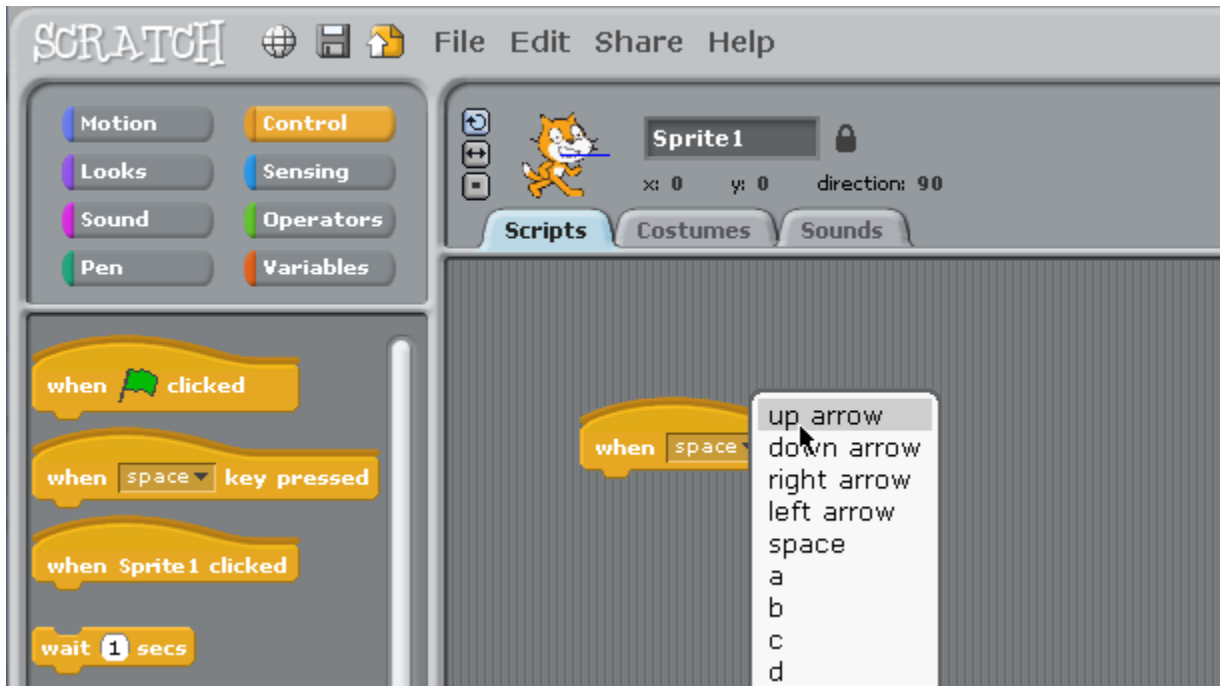
4. ... and drop it onto the scripts area:



But hang on! We want the up arrow key, not the space key!

5. Click the down arrow key next to "space" in the code block you've just added to the scripts area
6. Select **up arrow** from the list of keys:

Getting Started With Scratch



Your code block should now look like this:



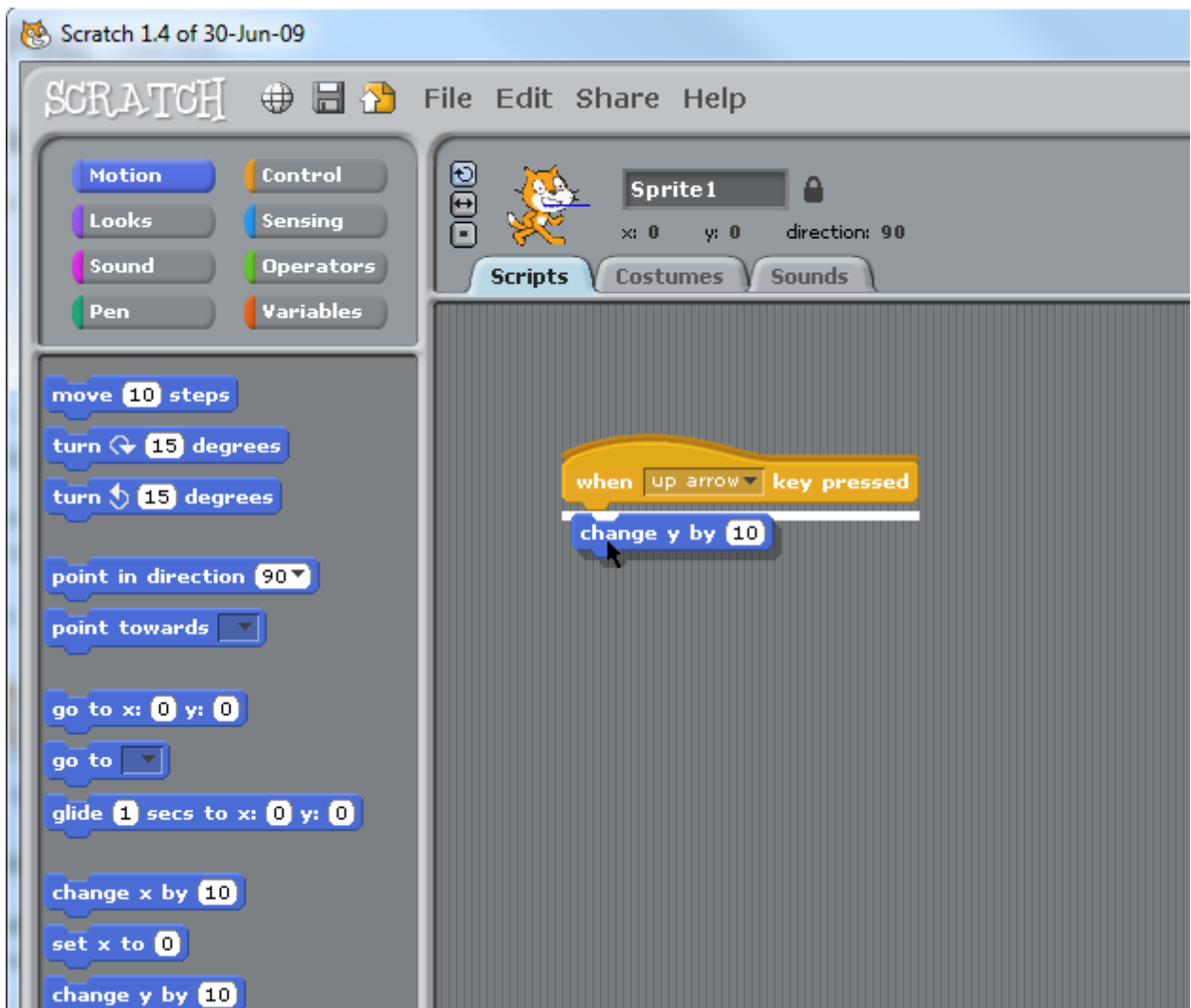
Next, we want the cat to move up when the up arrow key is pressed. We need another code block to do this:

7. Click the **Motion** block selection:

Getting Started With Scratch



8. Drag a **change y by 10** code block and drop it under the "when up arrow key pressed" code block you've already added:



Make sure that it clicks into place so the code blocks are joined together! The code blocks should now look like:

Getting Started With Scratch



Now, you can try this out and see if it works!

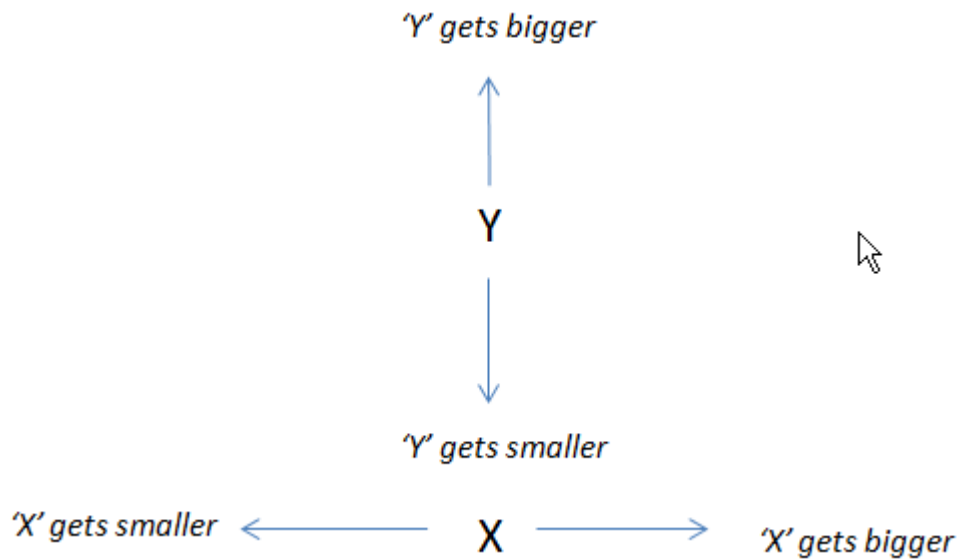
9. Click the up arrow key on your keyboard. The Cat sprite will move up the stage!



So, you can move the Cat up! What about down, left and right? First, you have to know about "x" and "y". These are used to place a sprite somewhere on the stage.

- "Y" values go up and down. Making "Y" bigger makes the sprite go up, and making "Y" smaller makes the sprite go down.
- "X" values go left and right. Making "X" bigger makes the sprite go right, and making "X" smaller makes the sprite go left.

Getting Started With Scratch



10. Now add code blocks to complete moving up/down/left/right:



Notice how Y and X are made smaller by using "-10" (that is "minus ten") for the value to change X and Y by.

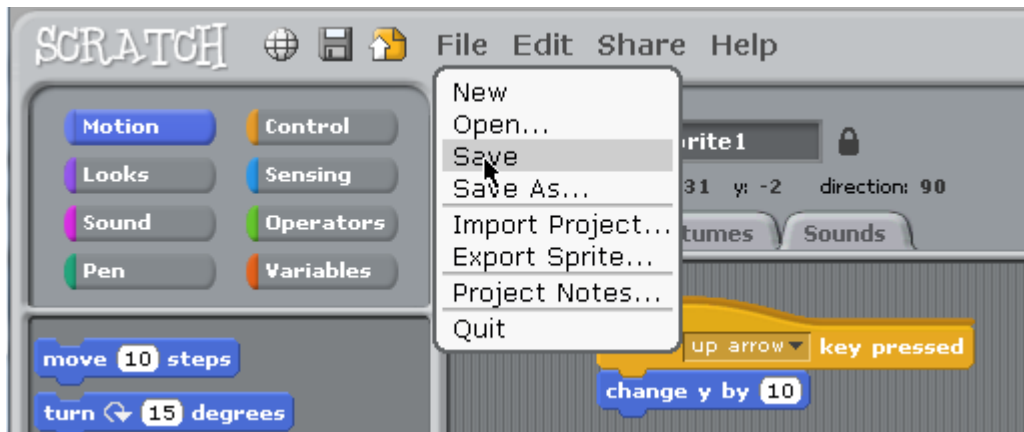
11. Try using the arrow keys to move the cat around the stage.

Loading and Saving your Project

You can practice saving and loading your project - you should save regularly as you're creating a project so you don't lose your work.

Getting Started With Scratch

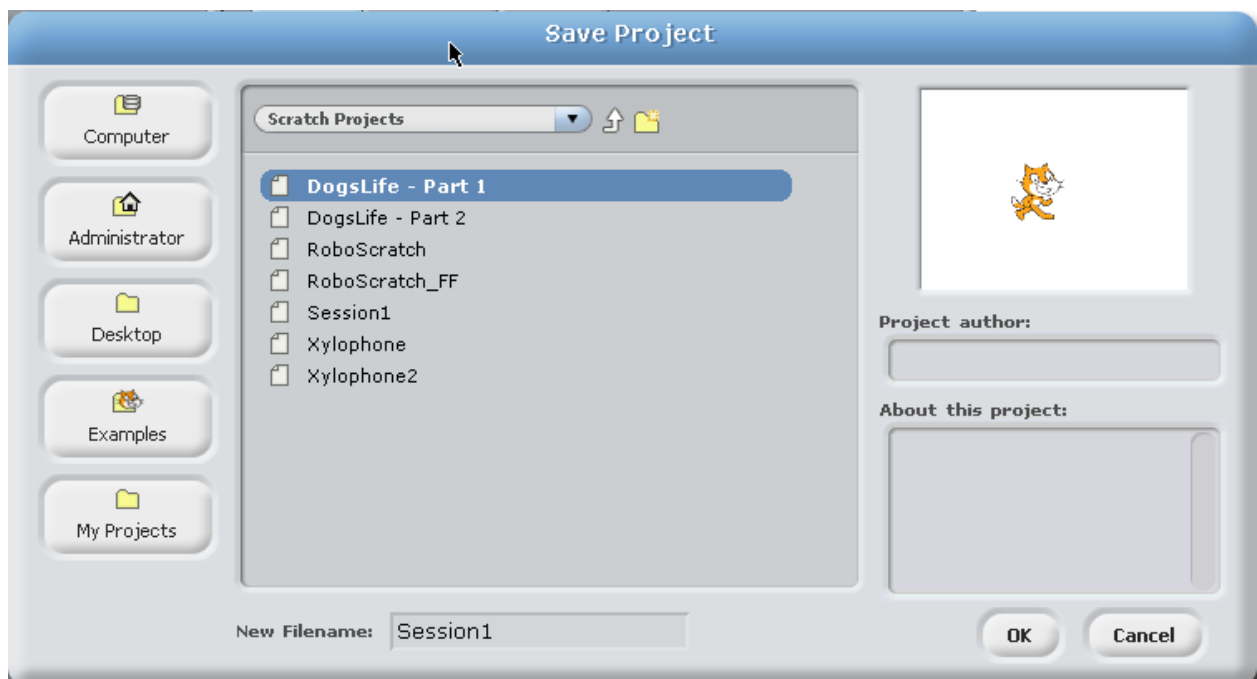
12. To save your project, go to the **File** menu and click on **Save**:



The "Save Project" window is opened.

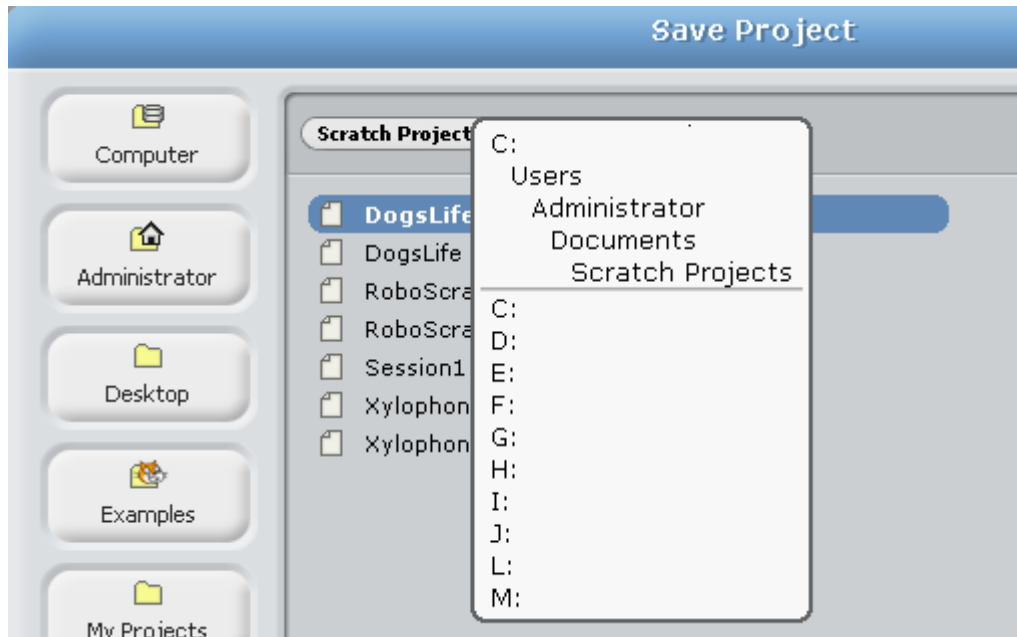
13. Enter a name for your project, such as **Session1**.

14. Click **OK**



You can find out where the project is saved by dropping down the location list:

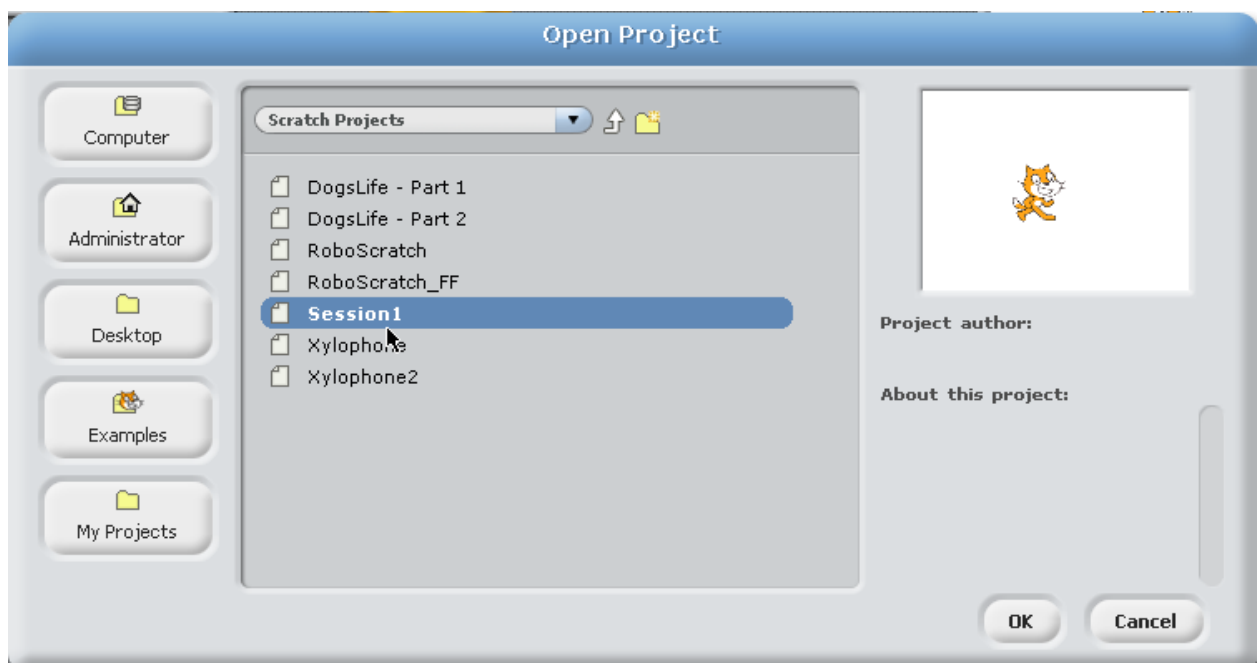
Getting Started With Scratch



This shows that the project will be saved in c:\users\administrator\documents\scratch projects. Your save location will probably be different.

To open a project you had previously saved:

15. Run Scratch (if it's not already running).
16. Select the **File** menu and then select **Open....**
17. Single click the project you want to open (such as "Session1") and click **OK**.



The project is then opened and ready for you to work on. Always remember to save your work!

Getting Started With Scratch

Session 2 - Broadcasting and Costumes

Broadcasting allows one sprite to let other sprites know that something's happened, such as when one sprite hits another sprite.

Each sprite can have more than one costume, and you can change these costumes with code blocks. For example, you might have one costume for being happy and another for being scared.

In this session you will add a trampoline and have the cat jump on the trampoline.

1. Run Scratch if it's not already running.

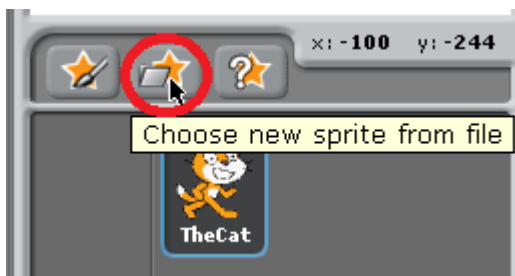
The "Cat" sprite will be automatically added as usual. This has the name "Sprite1" - you will now change the name to "TheCat":

2. Click the sprite's name box, remove "Sprite1" and type "TheCat":



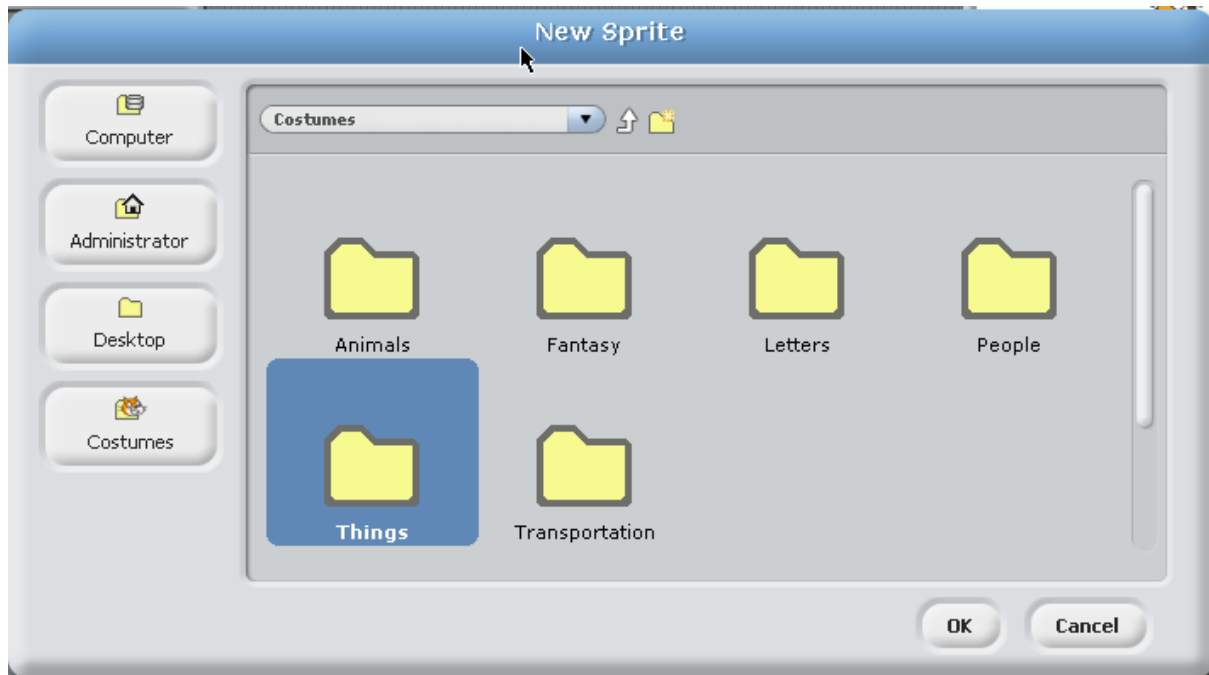
You will now add the trampoline sprite. Scratch comes with some sprites ready for you to use - one of these is a trampoline.

3. Click the **Choose new sprite from file** button in the sprite list window:

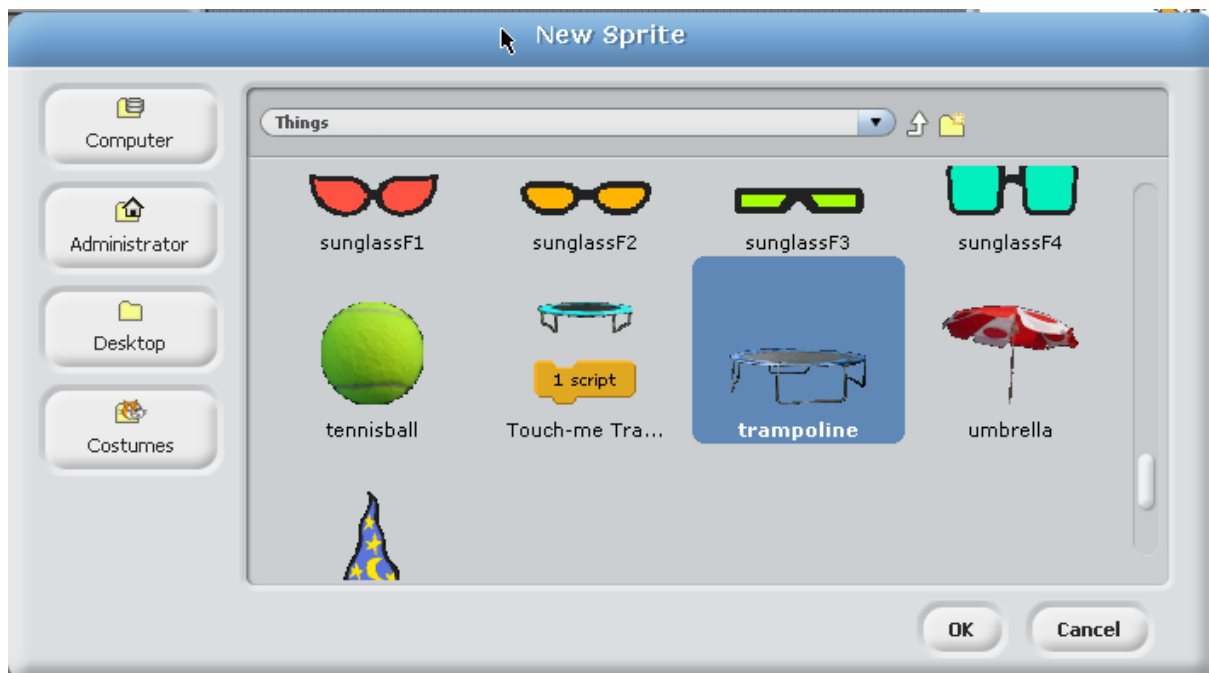


4. Double-click the **Things** folder in the "New Sprite" window:

Getting Started With Scratch

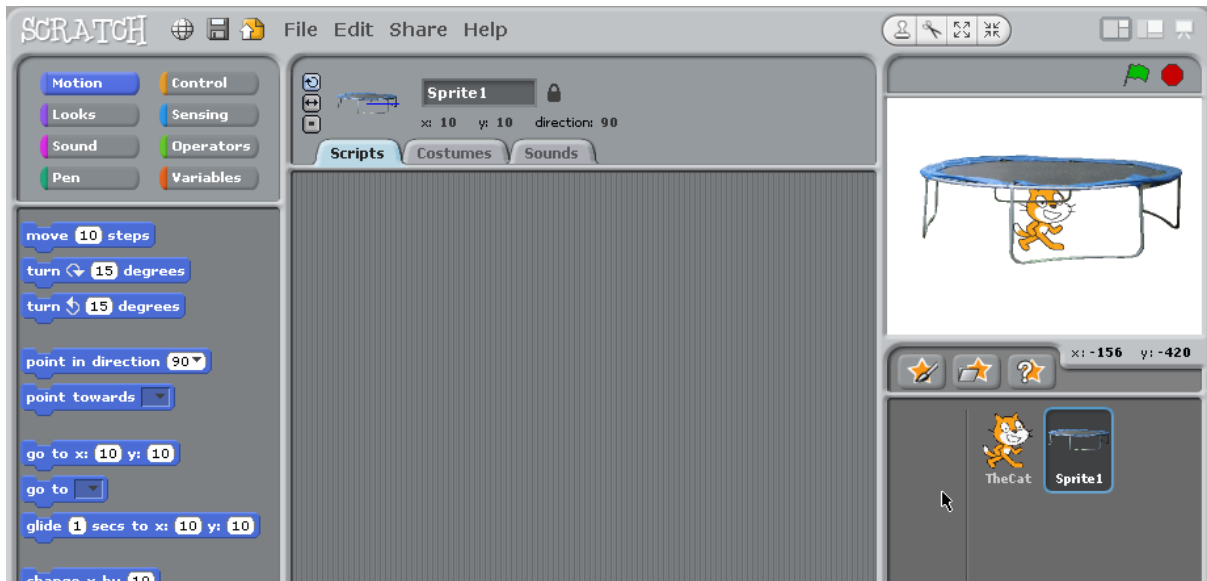


5. Scroll down the list of sprites and click **Trampoline**. (Be careful not to select the "Touch-me Trampoline" sprite) and click **OK**.



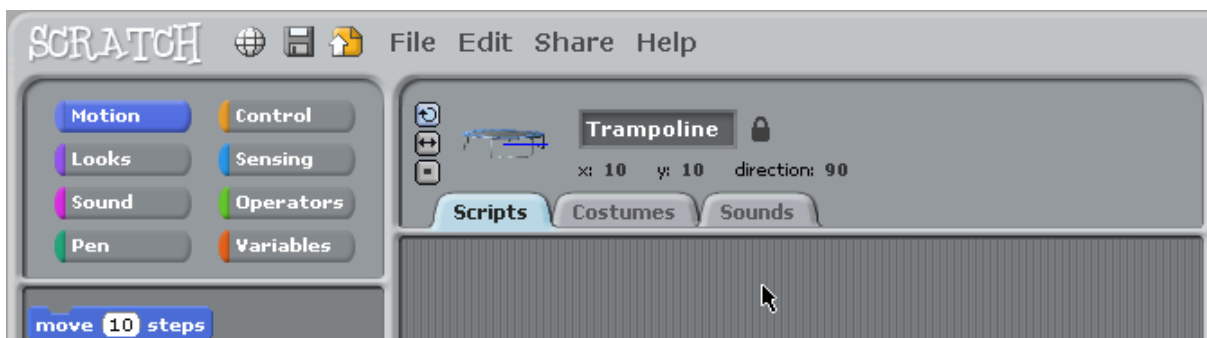
This adds the new sprite with the name "Sprite1":

Getting Started With Scratch



You will now change the name of the sprite to "Trampoline":

6. Change the name of the sprite you have just added to **Trampoline**:



Now re-arrange the sprites on the stage to put them in the right places:

7. Drag the Trampoline sprite down to the bottom of the stage:



8. Drag TheCat sprite above the trampoline like this:

Getting Started With Scratch



9. Save your project with the name **Session2**.

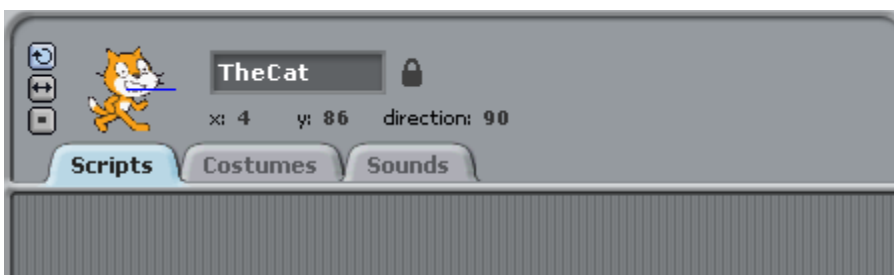
Now you'll get the cat to bounce up and down on the trampoline. You will need to add blocks to the sprite "TheCat".

Make sure you select the correct sprite when adding code blocks, otherwise they'll end up in the wrong place and the project won't work properly.

10. Select **TheCat** sprite from the sprites list:



11. Check TheCat sprite is selected:



The Green Flag

The green flag button will start running any blocks in a Green Flag code block. You can find the green flag button just above the stage:

Getting Started With Scratch



The "when green flag clicked" code block is in the "Control" section:



Clicking the red Stop button (which is next to the green flag button) will halt execution of your blocks.

12. Add a "when green flag clicked" block to TheCat sprite:



We want the cat to start bouncing when the green flag is clicked, and to continue bouncing until the red Stop button is pressed. You will need to add a "forever" block to the "when green flag clicked" block to do this:

13. Add a **forever** block under the "when green flag clicked" block:

Getting Started With Scratch



Now, any blocks you add inside the "forever" block will be run many, many times - in fact, until the red Stop button is pressed. But wait! There are no blocks in the "forever" block yet so it will do nothing!

Taking Decisions

You need to add code blocks into the "forever" block that say:

- If TheCat is touching the trampoline, bounce up.
- Else if the TheCat is not touching the trampoline, fall down.

You will notice that you're taking a decision - "*if this then* do that *else* do something else". We use a "if else" block to do this.

14. Add a **if ...else** block inside the "forever" block:



Be careful: Don't select the "if" block, make sure you choose the "if..else" block. It's in the "Control" block section.

Detecting Touching Sprites

The hexagon in the "if" blocks is where you put your decision:

Getting Started With Scratch



We want to say "if TheCat sprite is touching the trampoline". You do this by adding a "touching" block to the hexagon:

15. Select the **Sensing** code block section:

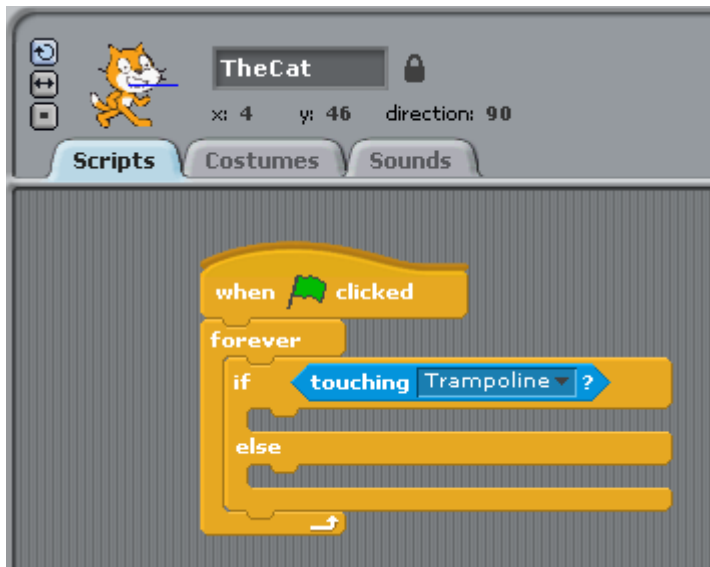


16. Add the **touching** code block into the hexagon:



17. Click the drop-down arrow in the "touching" block and select **trampoline**:

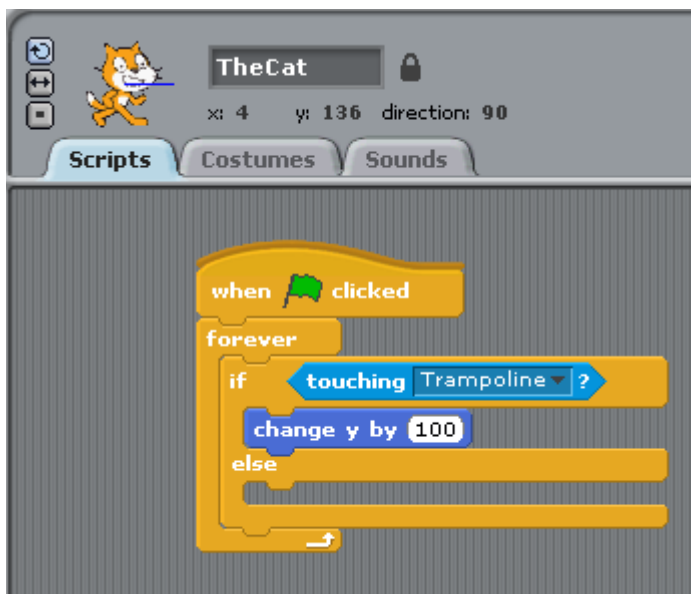
Getting Started With Scratch



There are two empty places in the "if" code block where you need to add code blocks to make TheCat move up and down.

First, when TheCat touches the trampoline it needs to jump up to the top of the stage.

18. Add a **change y by** code block into the first part of the "if" block and change the value to **100**:



19. Now add another **change y by** block to the second part of the "if" block and change the value to **-10**:

Getting Started With Scratch



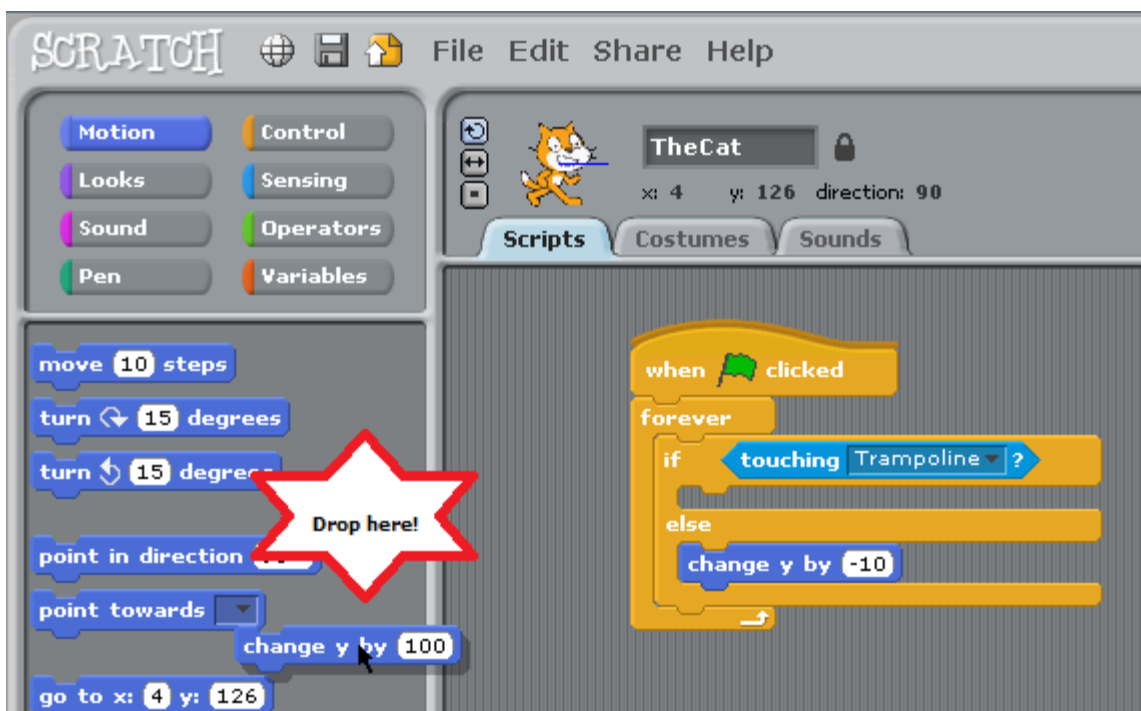
20. Click the Green Flag to start your project. You can click the Red Stop button to halt execution.

TheCat will jump up and down, but it all happens rather quickly. That's because when TheCat moves immediately up by 100 units when it hits the trampoline, and then falls down more slowly. We really want TheCat to bounce up more slowly.

We can do this by adding a **Repeat** block. This allows another code block to be repeated a number of times.

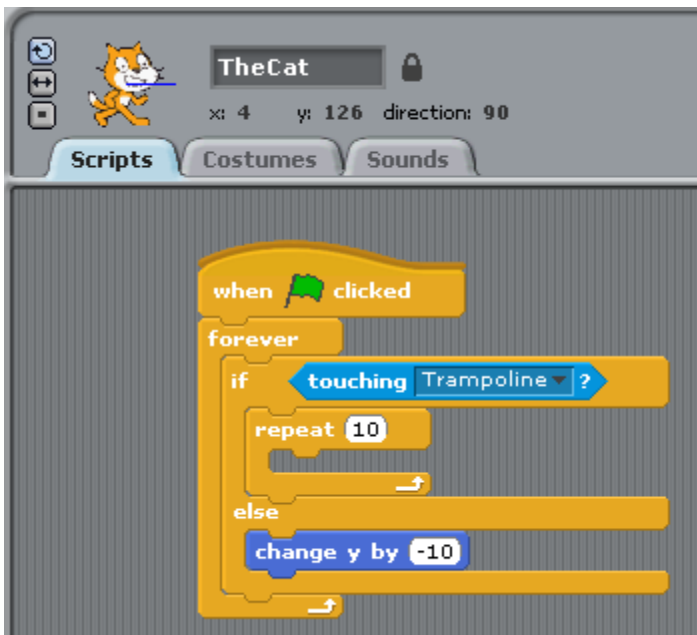
You will need to remove the "change y by 100" block. To do this:

21. Drag the **change y by 100** block from the "if" block and drop it into the block library:



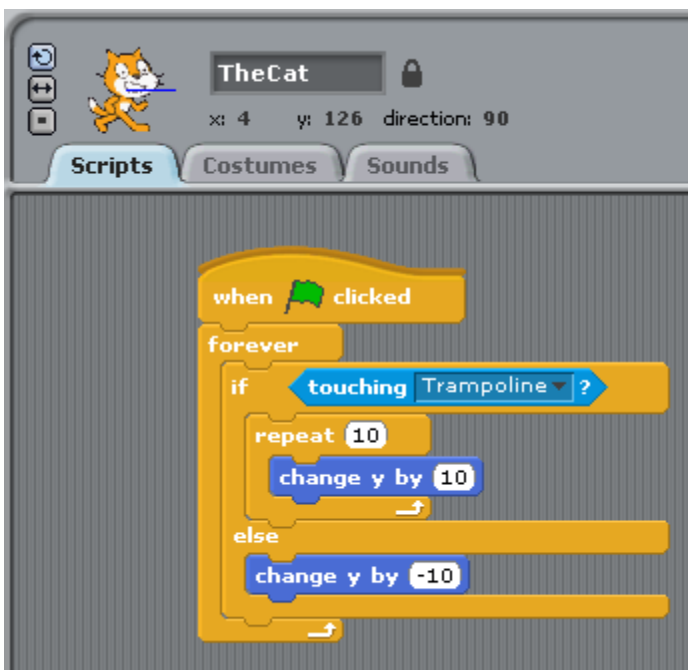
Getting Started With Scratch

22. Add a **repeat** block into the first part of the "if" block. It's located in the "Control" block section:



The "10" value tells the block how many times to repeat.

23. Add a **change y by 10** block inside the "repeat" block:



Since TheCat now moves up in small steps of 10 units rather than one step of 100 units the animation is much smoother.

24. Save your project with the name "Session2".
25. Click the Green flag button to try out your project.

Getting Started With Scratch

Broadcasting Messages

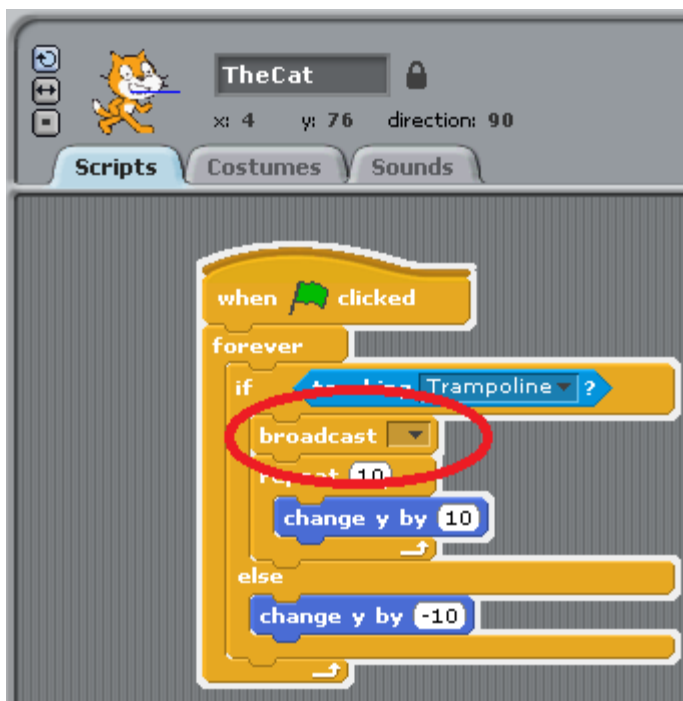
Often, when something happens (like the cat hitting the trampoline) you want to tell other sprites. This is done by *broadcasting messages*. You need to:

- Have one sprite broadcast the message
- Have other sprites listen for the message, and do something when the message is received.

You can have many different messages and each has its own name.

You will create a message that will be broadcast by TheCat when it hits the trampoline. The trampoline will listen for this message, when it's received, it will change its shape to look like the cat's jumped on it.

26. Drag a **broadcast** block from the "Control" section and put it just under the "if" block like this:

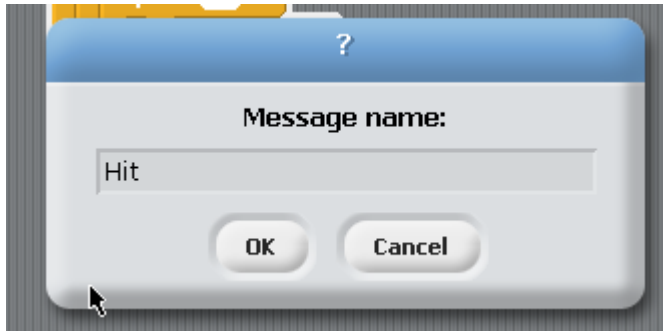


27. Click the down arrow in the "broadcast" block and select **new**.

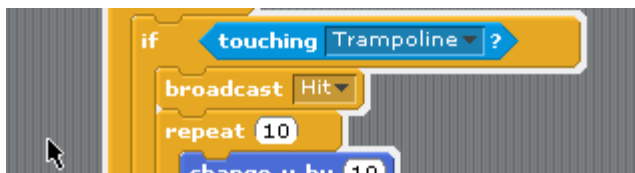


28. Type in the **Hit** for the name and click **OK**:

Getting Started With Scratch

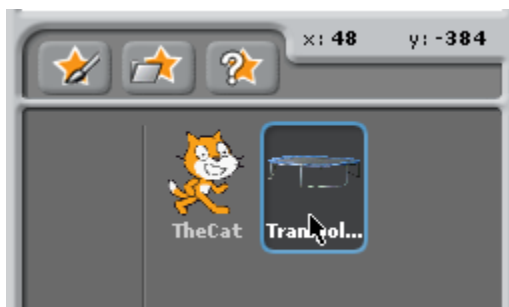


The block will now look like this:



You have now created a message called "Hit" and it's broadcast whenever TheCat hits the Trampoline. You now need to add blocks to the Trampoline that respond to this message.

29. Click the Trampoline sprite in the sprite list to select it:



30. Add a **when I receive Hit** block from the "Control" section to the Script for the Trampoline:



31. Add a **set color effect to 0** code block from the "Looks" section:

Getting Started With Scratch



32. Click the down arrow key and select **fisheye**:



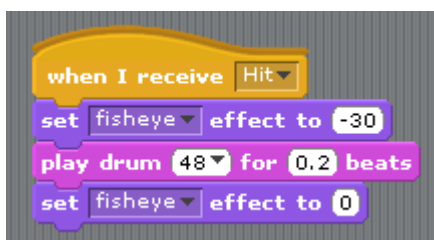
33. Enter **-30** for the value:



34. Add a **play drum 48 for 0.2 beats** block from the "Sound" section:



35. Add another **set color effect to 0** code block from the "Looks" section and select **Fisheye**.
Leave the value at "0".



Getting Started With Scratch

The "fisheye" effect changes the appearance of the sprite as if it's viewed using a fisheye lens on a camera. This works for any sprite, not just a trampoline.

36. Save your project.
37. Click the Green Flag to try out your project.

Costumes

Costumes allow you to change the way your sprites look. You can change the sprite's costume as things happen in your project. For example, you may want to change TheCat when it hits the trampoline.

To do this, you first need to add a new costume to TheCat:

38. Select TheCat sprite by clicking **TheCat** in the list of sprites.
39. Click the **Costumes** tab:



There are already two costumes for this sprite - you will now delete the second one.

40. Click the **X** button for the second costume:



41. Click the **Import** button to select a new costume from the library.
42. Click the down arrow for the "Things" box and select Costumes:

Getting Started With Scratch



43. Double-click **Animals** from the folder list:



44. Select **cat4** from the list of costumes and click **OK**.

Getting Started With Scratch



You will now have two costumes for TheCat:



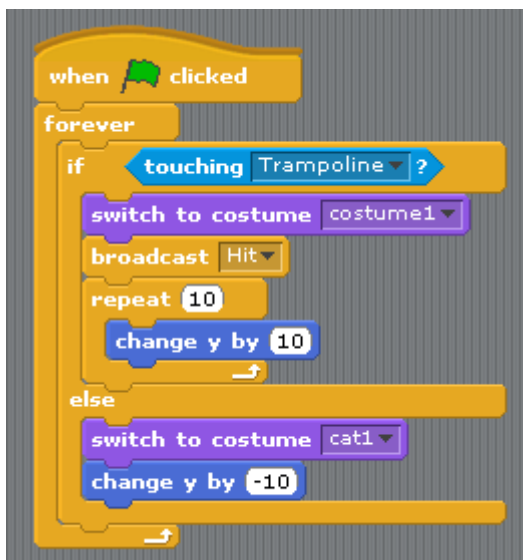
You can use code blocks to choose which costume to display. When TheCat hits the costume you will change the second costume, and change back to the first costume when TheCat starts falling again.

45. Add a **switch to costume cat1** block from the "Looks" section to below the "if" block.
46. Click the down arrow and select **costume1**:

Getting Started With Scratch



47. Add a **switch to costume cat1** block to just below the "else".
48. Make sure the "cat1" costume is selected.



49. Save your project.
50. Click the Green Flag to try out your project.

Getting Started With Scratch

Session 3 - The Fish Game

This game was created by "Wong-Sir", see this web site for more details: <http://www.wong-sir.com/scratch/> which is modified from the Fish Game tutorial (<http://scratch.redware.com/project/fish>, created by Redware Limited).

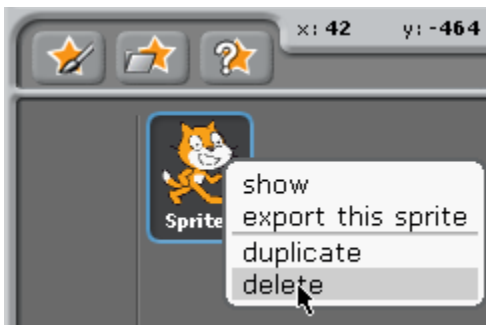
The game creates an aquarium with a shark. The shark goes around eating other fish in the aquarium!



Creating the Shark

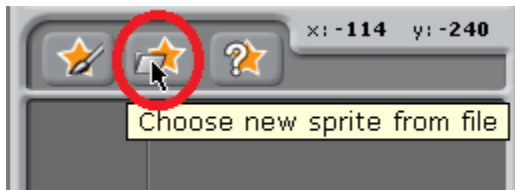
You will create a new Scratch project and add the Shark sprite from the sprite library. Blocks will be added to make the Shark follow the mouse pointer.

1. Start Scratch. You should have a new project with the Cat sprite.
2. Delete the Cat sprite by right-clicking it and selecting **delete**.



3. Create a new sprite by clicking the **Choose new sprite from file** button:

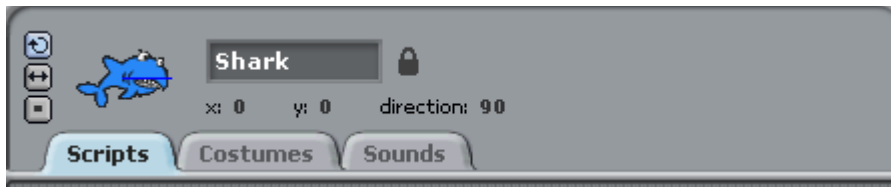
Getting Started With Scratch



4. Select the **shark1-a** sprite from the Animals folder and click **OK**.



5. Rename this sprite to **Shark**:



You will now create a block that will:

- Start executing when the green flag is clicked.
- Make the Shark smaller to 40% of its original size.
- Create a loop that will run forever.
- In this loop, make sure that the Shark sprite points to the mouse-pointer.
- Moves the Shark sprite towards the mouse pointer
- Waits a short time before moving again (this slows it down).

The first attempt to do this will use this code:

Getting Started With Scratch

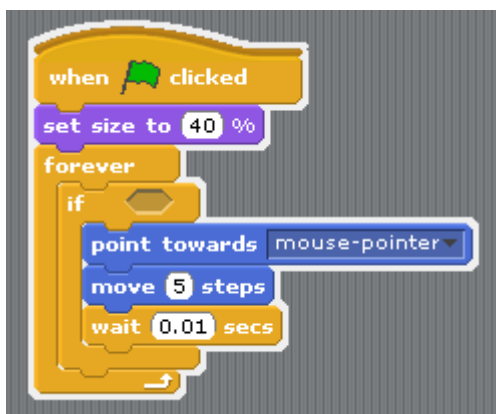


To add these blocks to the Shark sprite do the following:

6. Add a **when green flag clicked** block from the "Control" section.
7. Add a **set size to 100%** block from the "Looks" section and change to **40%**.
8. Add a **forever** block from the "Control" section.
9. Add a **point towards** block from the "Motion" section and select **mouse-pointer**.
10. Add a **move 10 steps** block from the "Motion" section and change to **5**.
11. Add a **wait 1 secs** block from the "Control" section and change to **0.01**.
12. Save the project as **Session3** and click the green flag to run.

You will find that the shark follows the mouse pointer quite well until it catches up with the mouse pointer. When this happens the Shark sprite goes mad - this is because it keeps moving 5 steps in a circle. An **if** block can be used to stop the Shark moving when it catches up with the mouse pointer.

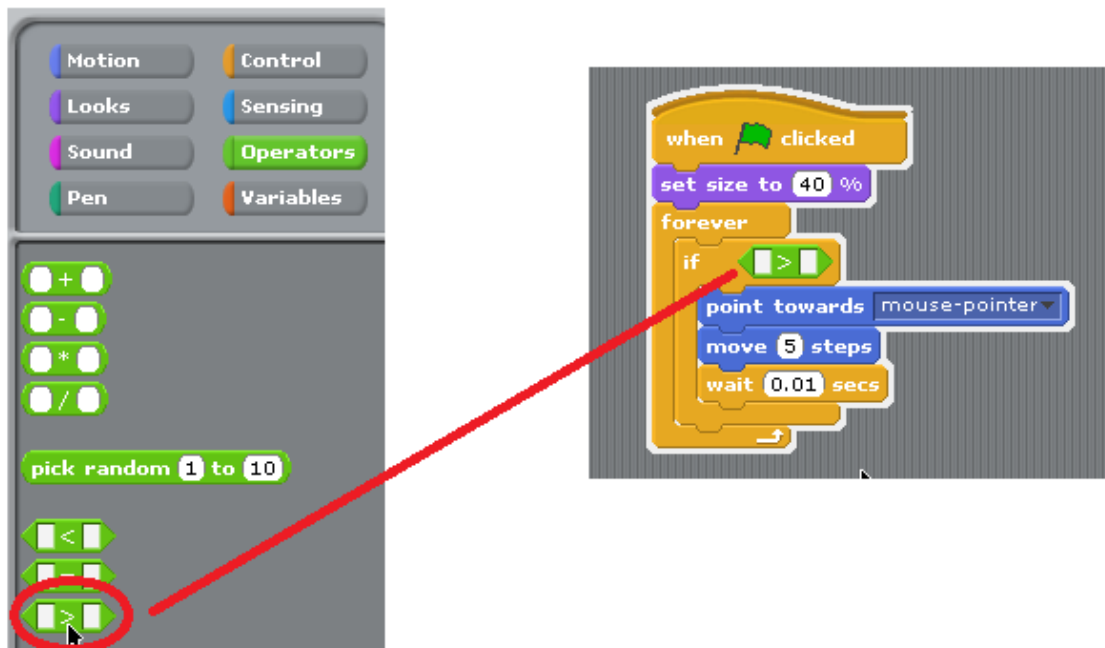
13. Add a **if** block under the "forever" block:



We need to put an *operator* in the **if** block that says "if the distance to the mouse pointer is greater than 5" so that the Shark will only move if it's not near the mouse pointer.

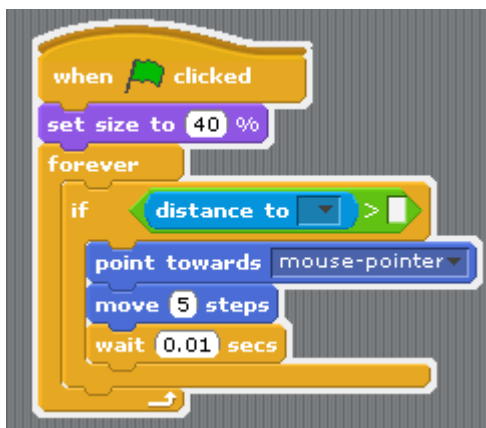
14. Select the **Operators** code section (the green one).
15. Drag a **>** (greater than) operator and put it into the **if** block's hexagon:

Getting Started With Scratch



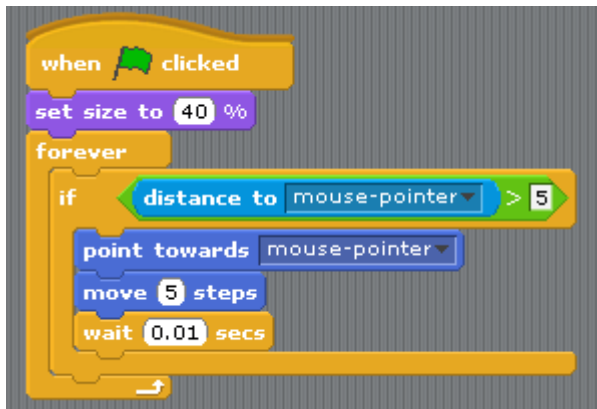
The greater than operator block has two spaces for the numbers being compared. The first value for us will be the distance from the Shark to the mouse pointer and the second will be the value to compare to, which will be 5 (this is the distance the Shark moves when following the mouse).

16. Select the **Sensing** block section (light blue) and drag a **distance to** block and drop it in the first of the two rectangles in the > operator, like this:



17. Drop down the list for "distance to" and select **mouse-pointer**.
18. Type in **5** into the second rectangle. The blocks should look like this:

Getting Started With Scratch



19. Save your project and click the green flag to test.

Creating the YellowFish Sprite

The YellowFish sprite will move around the fish tank and the Shark will try and eat this fish!

20. Create a new sprite by clicking the **Choose new sprite from file** button.



21. Select **fish4** from the Animals folder and click **OK** in the "New Sprite" window.

22. Rename this sprite to **YellowFish**:



You will now create a block that will move the yellow fish around the stage. This code will, when the green flag is clicked:

- Make the yellow fish smaller (20% of its original size)
- Create a forever loop
- In this loop, make the fish turn left or right in a random direction.
- Move the fish on three steps.
- If the fish hits the side of the stage, make it bounce off the edge.
- Wait a small amount of time to slow the fish down.

The most complex bit of this is the **turn** block. There are two types of turn block:

Getting Started With Scratch



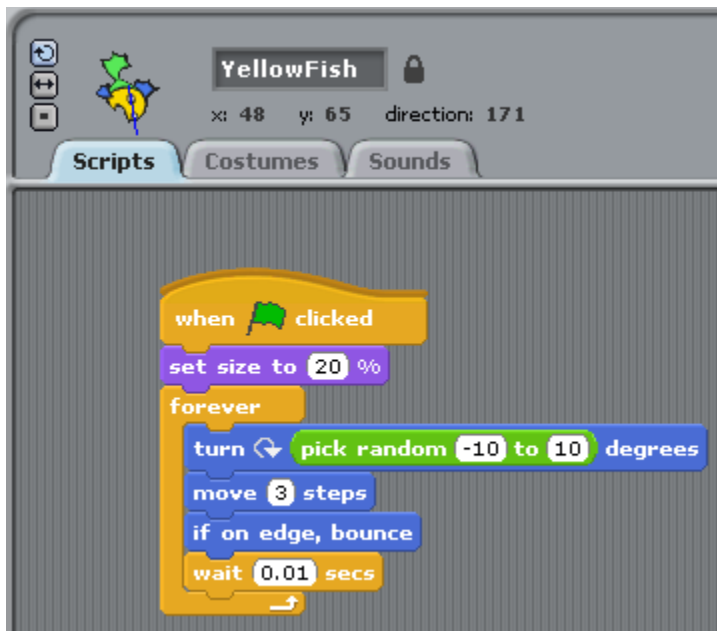
The first moves clockwise by a number of degrees and the second moves anti-clockwise by a number degrees. You will use the first.

For this block a negative number (less than zero) for degrees always moves anti-clockwise and a positive number (greater than zero) for degrees will move clockwise.

Remember: There are 360 degrees in a circle!

We want the fish to turn a random number of degrees each time, either anti-clockwise (a negative value) or clockwise (a positive number). We use the **pick random** block from the operators section. By entering **-10** and **10** we tell the block we want a random number between -10 and 10.

23. Add this block to the YellowFish sprite:



24. Save the project and click the **Green Flag** to try it out.

You should find that the YellowFish sprite moves around the stage and the Shark sprite still follows the mouse pointer. Not much eating going on here yet!

The Shark Eats the Yellow Fish!

The Shark most open it's mouth to eat the YellowFish! To do this, you need to add a costume to the Shark.

25. Click the Shark sprite in the list of sprites to select it.
26. Click the **Costumes** tab for the Shark sprite.
27. Click the **Import** button to import a new costume:

Getting Started With Scratch

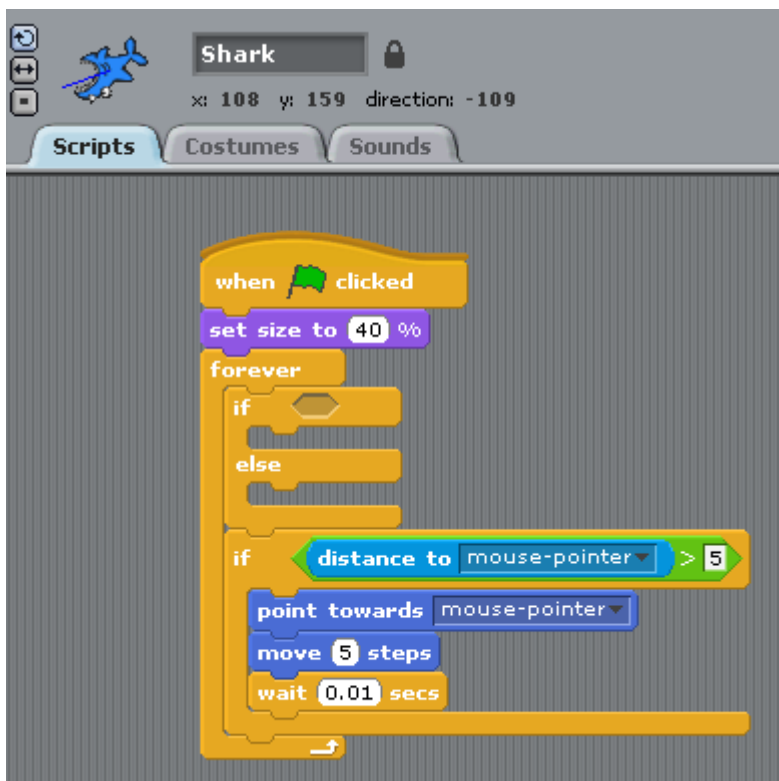


28. Select the **shark1-b** sprite from the **Animals** folder and click **OK**.

A script block now needs to be added which will open the Shark's mouth when it's near the YellowFish.

29. Click the **Scripts** tab for the Shark sprite.

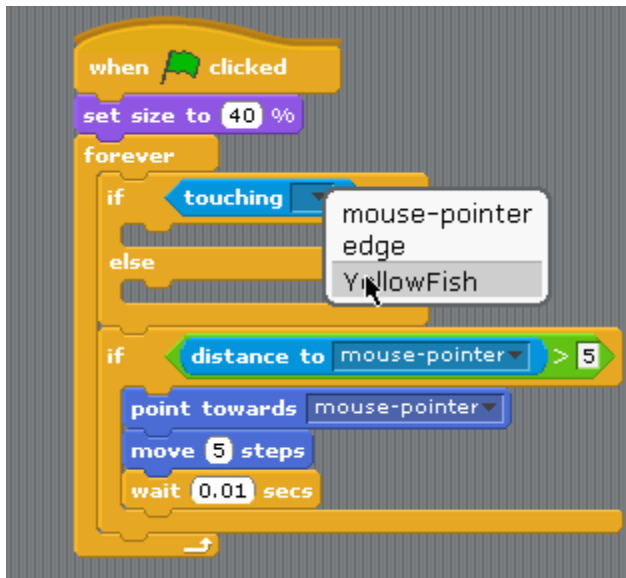
30. Add an **if else** block just after the forever block:



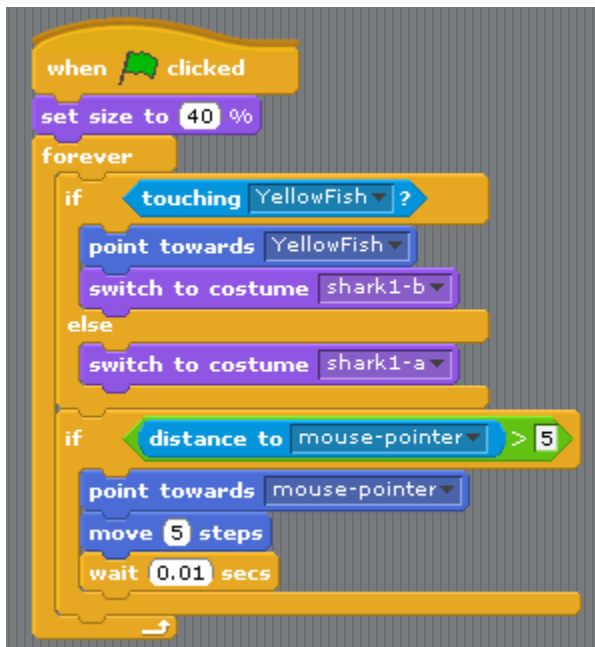
31. Add a **touching** block from the "Sensing" section into the hexagon for the "if" block you just added.

32. Drop down the list for the "touching" block and select **YellowFish**:

Getting Started With Scratch



33. Add the three other blocks for the "if else" block shown here:



34. Save your project and run by clicking the Green flag.

You should now see the Shark opening its mouth when near the YellowFish sprite.

The YellowFish sprite will be hidden when eaten by the Shark. You will need to add a block to the YellowFish sprite that detects when it's near the Shark, and if it is, to hide itself. Also, when the Green flag is clicked the YellowFish must make sure it's shown.

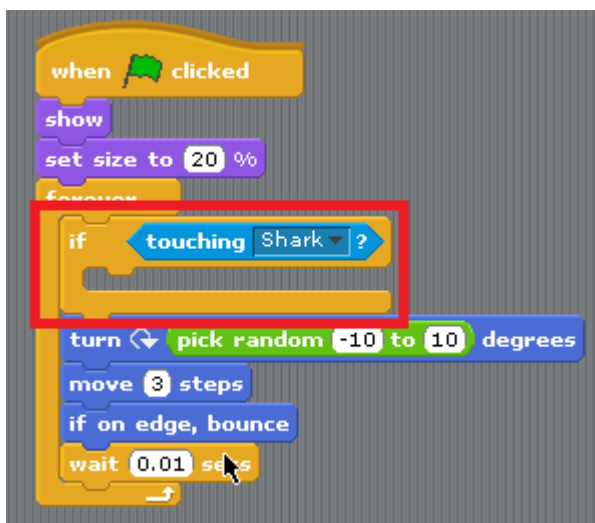
35. Click the YellowFish sprite in the sprite list to select it.

36. Add the **show** block shown below *to the YellowFish sprite*:

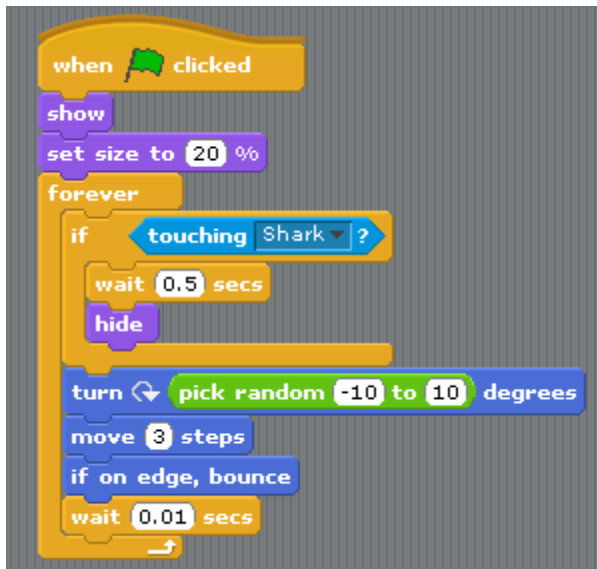
Getting Started With Scratch



37. Add a **if** block under the "forever" block. Make sure that the existing blocks (starting with "turn") are *outside* the if block you've added:



38. Add the **wait** and **hide** blocks into the "if" block as shown here:



39. Save your project and click the **Green Flag** to test.

You should find that the Shark can now eat the YellowFish sprite! The shark's mouth should open and the YellowFish sprite disappears!

Adding Sound

We now want to make a "pop" sound when the YellowFish sprite is eaten! You need to import a sound for the YellowFish and then play the sound when its eaten. To do this:

40. Make sure the YellowFish sprite is selected.

41. Click the **Sounds** tab for the YellowFish:



42. Click the **Import** button.

43. Select **Pop** from the "Effects" folder and click **OK**.

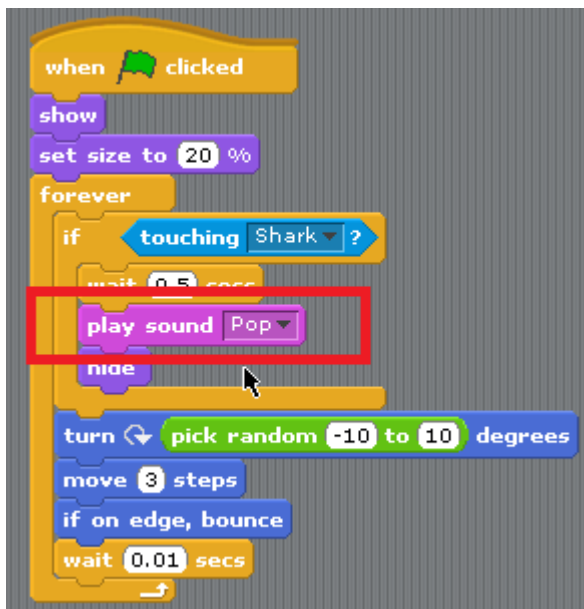
You should now see the imported sound:

Getting Started With Scratch



44. Click the **Scripts** tab.

45. Add a **play sound** block from the "Sound" block section and ensure **pop** is selected:



46. Save your project and click the **Green Flag** to test.

Variables and Keeping Score

We need to keep a score of how many fish the Shark has eaten. In Scratch we use *variables* to do this. A variable is a place where we can keep, say a number, and get that number back again sometime later. We can also do things to numbers in a variable, such as add another number to it.

First, you need to add a variable for the score:

47. Select the **variables** code section (it's orange).

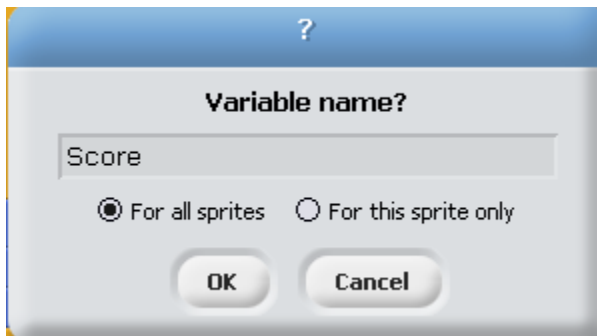
Getting Started With Scratch



48. Click the **Make a variable** button.

49. Type in **Score** for the variable's name and make sure **For all sprites** is selected.

50. Click **OK**.



You should now see some new blocks added to the "Variables" section that you can use with this new variable:



Also, the variable is displayed in the stage so you can see the number it contains:

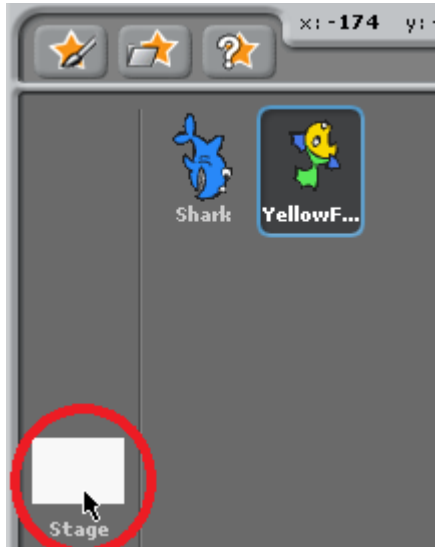


Getting Started With Scratch

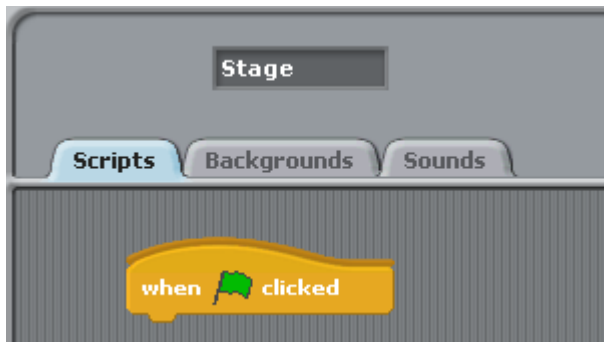
Adding Script Blocks to the Stage

You can add script blocks to the stage when these blocks are not linked to particular sprites. You will add a block to the stage to set the score when the game is run. To do this:

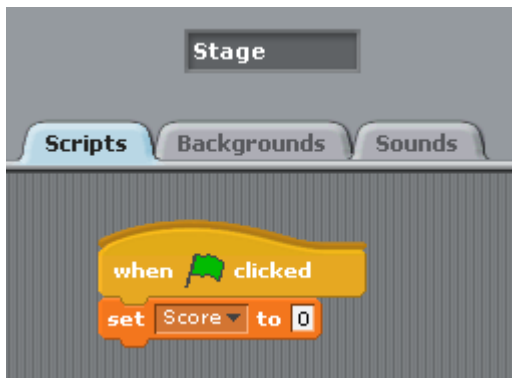
51. Click the stage icon in the sprites list window:



52. Add a **when green flag clicked** block to the stage's "Scripts" section:



53. Add a **set score to 0** block under this code block:

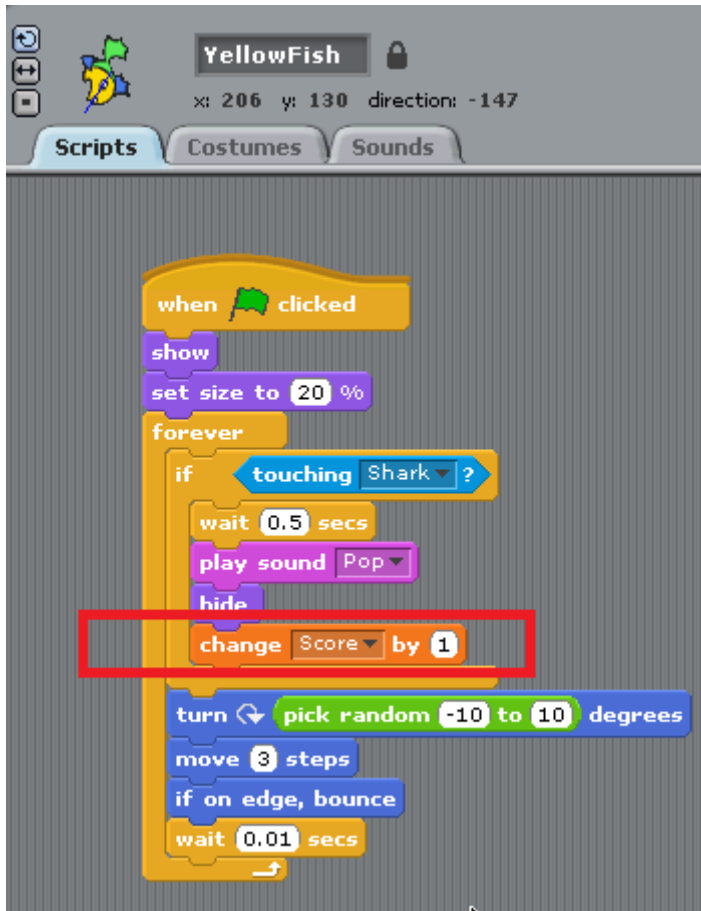


This makes sure that the score is set to "0" whenever the green flag is clicked. You now need to add one to the score when the shark eats the fish. You will add a code block to the **YellowFish** sprite:

Getting Started With Scratch

54. Select the **YellowFish** sprite.

55. Add a **change score by 1** block under the "hide" block you added earlier:



This will add 1 to the "Score" variable when the fish is eaten.

56. Save your project and click the **Green Flag** to test.

You should now see the "Score" change from 0 to 1 when the fish is eaten.

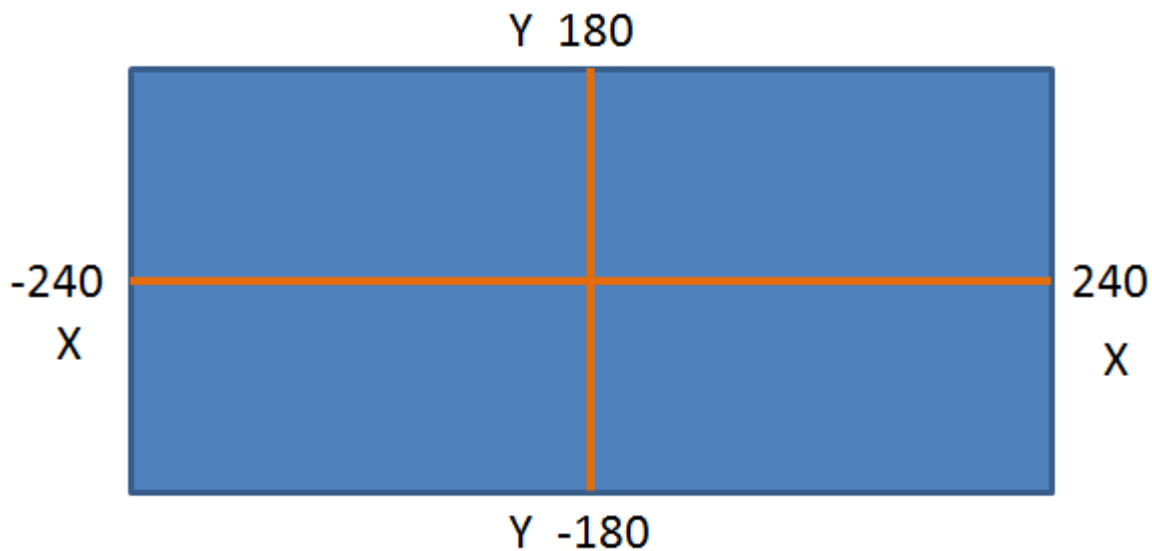


Recreating the YellowFish

There are no more fish to eat once the shark has eaten the yellow fish. You will now add code that will recreate the yellow fish after it's been eaten. Also, the yellow fish will be put into a random location and face a random direction.

You have to know how big the stage is to put the fish in a random location. Here are the sizes:

Getting Started With Scratch

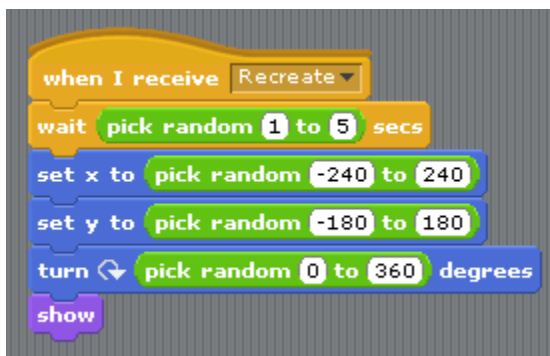


The program will pause for a short time and yellow fish will need to be put in a random location when:

- The game first starts
- When the fish is recreated after it's been eaten.

Rather than write the code twice you'll use a broadcast message to do this.

57. Add the following block to the **YellowFish** code scripts:



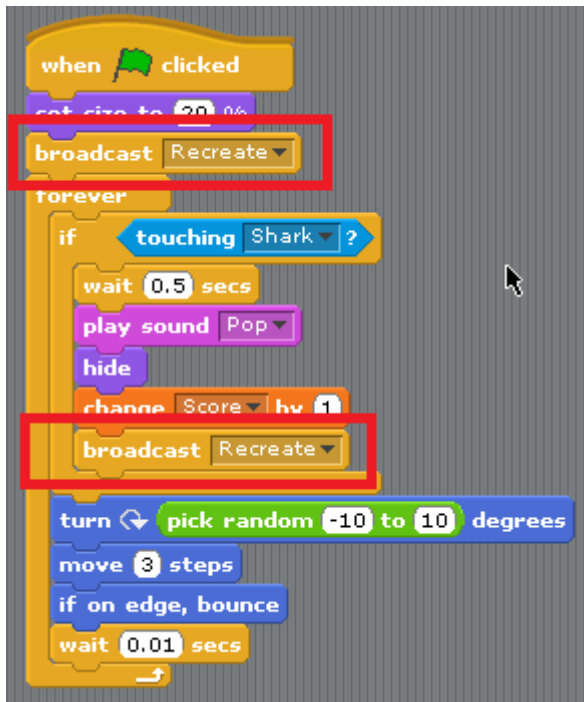
This code does the following:

- Waits for a random period between 1 and 5 seconds.
- Sets the X location to a random number between -240 and 240.
- Sets the Y location to a random number between -180 and 180.
- Sets the direction the fish points in to a random number between 0 and 360 degrees.

The broadcast message must be sent from two places in the **YellowFish's** code block.

58. Add two **broadcast** blocks for the "Recreate" message and remove the **show** block that's just below the "when green flag clicked" block as shown below:

Getting Started With Scratch



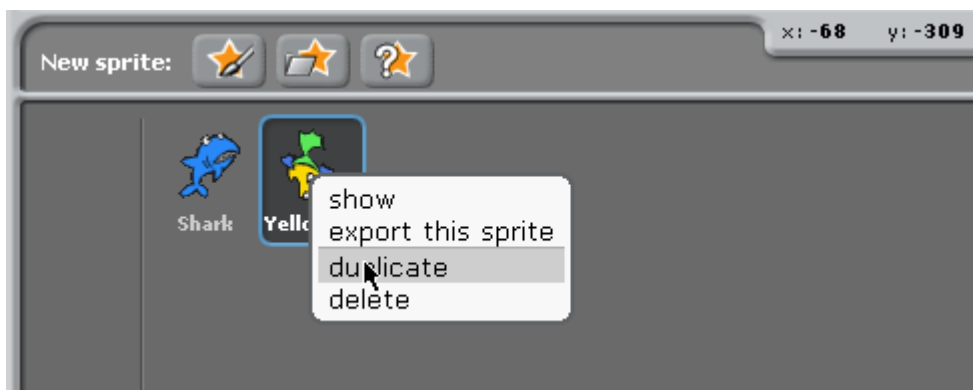
59. Save your project and click the **Green Flag** to test.

The shark can now eat the fish and the fish will be recreated after a short period of time. Your score can now be larger than 1!

Creating more fish!

The game would be more fun with more fish! You can use the Scratch *sprite duplicate* command to create more fish with the same script blocks.

60. Right-click the **YellowFish** sprite and select the **duplicate** command.



61. Single-click the new sprite, which is called "Sprite1" and rename it as **YellowFish2**.



Getting Started With Scratch

62. Do this twice more to create **YellowFish3** and **YellowFish4**.



63. Save your project and click the **Green Flag** to test.

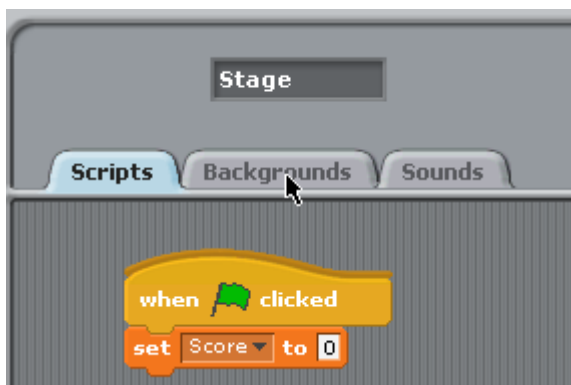
Setting the Stage's Background picture

To make the game look more like an aquarium you can now set a picture for the background:

64. Click the **stage** to select it:



65. Click the **Backgrounds** tab for the stage:

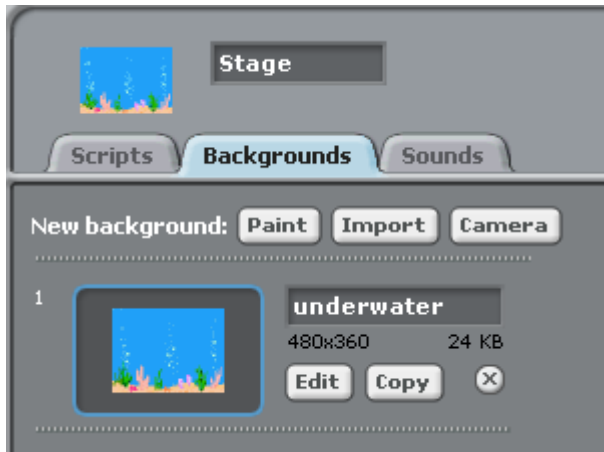


66. Click the **Import** button and select the **underwater** image from the "Nature" folder.

67. Click the **X** delete button for the "background1" background.

Getting Started With Scratch

Your background should now look like this:



68. Save your project and click the **Green Flag** to test.

Other Things to Try

See if you can work out how to:

- Change the costumes for fish so they look different.
- Make fish that move at different speeds.
- Create different types of animals, perhaps a crab?

Getting Started With Scratch

Session 4 - Dragon Graphics!

Turtle Graphics was originally developed as part of a teaching language called Logo. The idea is simple - you have a Turtle, and it has a pen stuck to its tail. You can tell the turtle to:

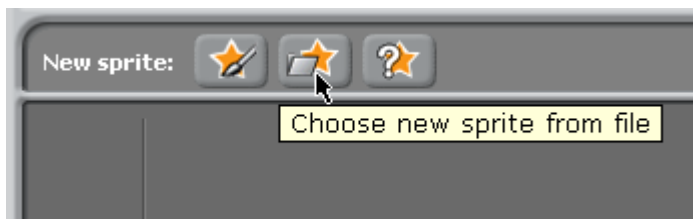
- Move forward a number of steps
- Turn left or right
- Lift up the pen (no drawing) or lower the pen (drawing) as the turtle moves

Now, we're going to use a dragon rather than a turtle (it moves faster!) and we're going to do it in Scratch. So, what types of images can we draw?

Drawing a Circle

Squares and triangles are easy (perhaps you want to try?), so we're going to start with drawing a circle. To do this:

1. Start a new Scratch project (select **File + New**).
2. Delete the Cat if it's added for you (we're *not* doing Cat Graphics!)
3. Add a new sprite from a file and **Choose new sprite from file**:



4. Select **dragon-1b** from Fantasy

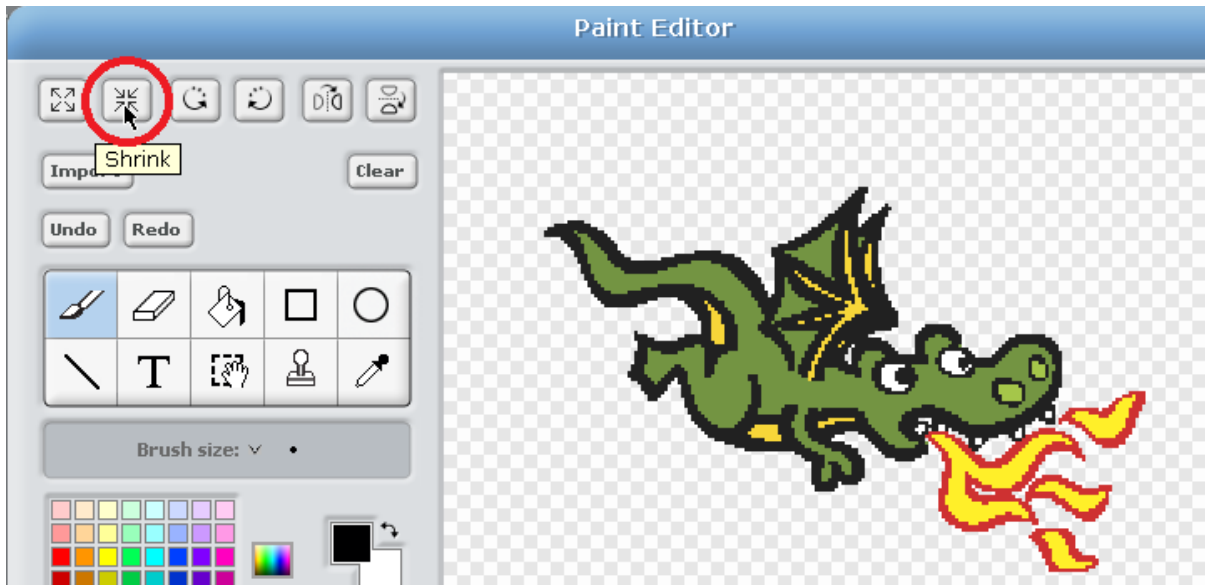
You will now make the dragon a little smaller:

5. Make sure your sprite is selected, choose the **Costumes** tab and click **Edit**.



6. Click the Shrink button three times.

Getting Started With Scratch



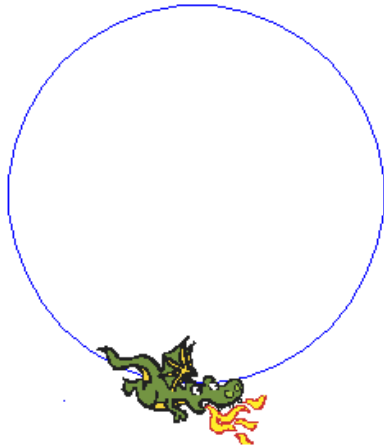
7. Now add the following steps to the dragon sprite:



The **hide** and **show** blocks are in the "Looks" section - these hide and show the sprite. The **clear** block (from the "Pen" section) clear any drawing that's been done on the stage. The **pen down** block causes the sprite to draw on the background whenever it's moved.

8. Click the **green flag**.

You will find that a circle will be drawn:



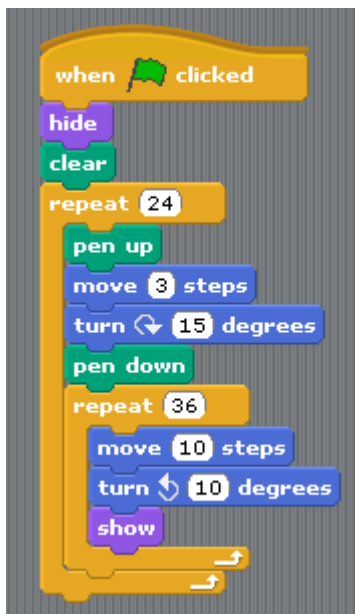
Note:

- Try moving the dragon around on the stage before clicking the green flag if your circle is not all visible on the stage or if a circle is not drawn.
- We turn 10 Degrees in each repeat loop ($10 \times 36 = 360!$).

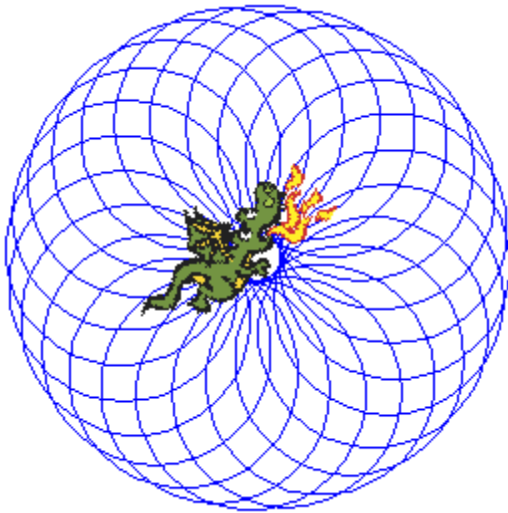
A Circle of Circles!

You will now draw lots of circles around an imaginary circle.

9. Change the code block for the dragon to this:



10. Click the green flag to draw the following pattern:

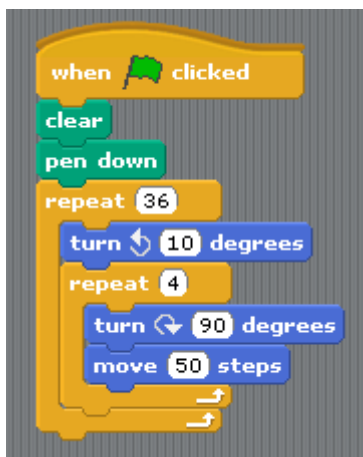


The code adds another “repeat” around the one you wrote before - it just moves the starting point for each circle it draws. Look carefully, some of the values from your existing programme have changed to make it fit.

Squares!

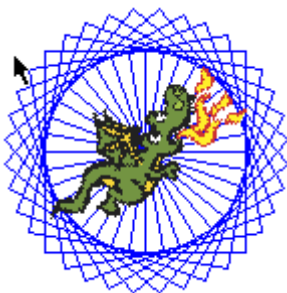
You can use squares to draw a circle to:

11. Modify the block for the dragon to this:



12. Click the Green Flag.

This draws 36 squares, each time moving a little around an imaginary circle:



Getting Started With Scratch

Spirals – A little more advanced

Here's how to draw an Archimedean Spiral (see: http://en.wikipedia.org/wiki/Archimedean_spiral).

13. Delete the block for the dragon.

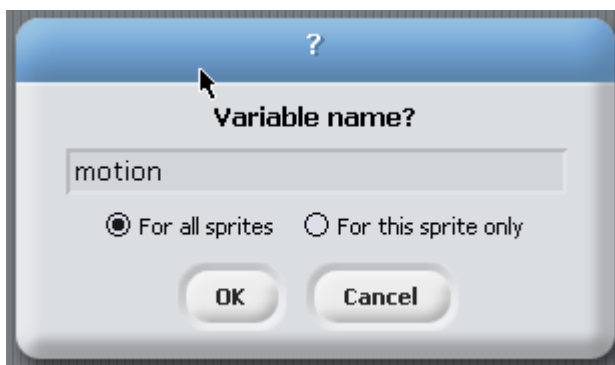
You will need to add a variable called **motion**:

14. Select the **variables** block section (the orange one).

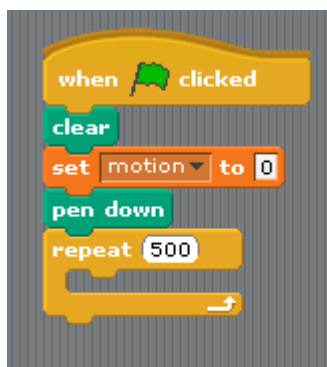
15. Click **Make a variable**.



16. Enter **motion** for the name of the variable and click **OK**



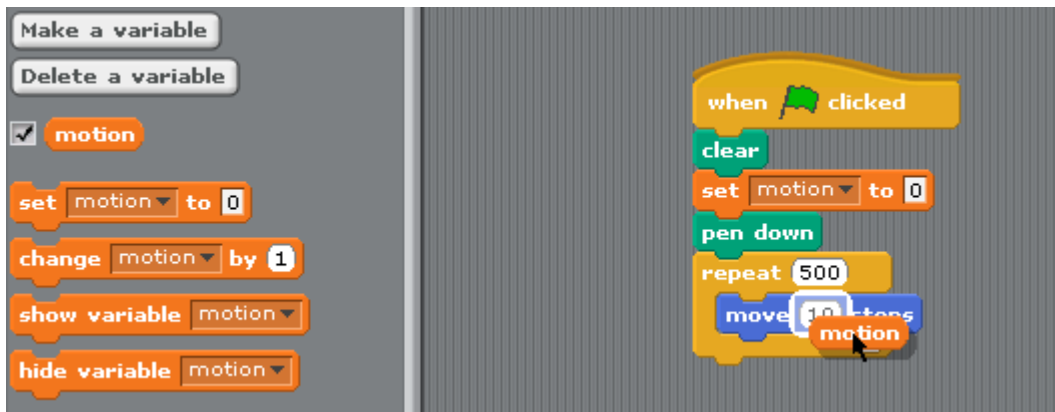
17. Add the following blocks for the dragon sprite:



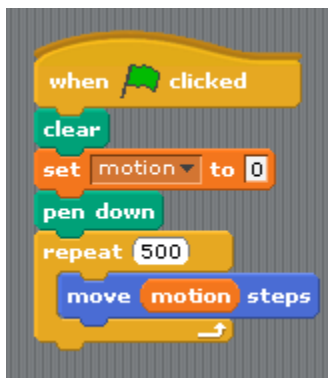
18. Add a **move 10 steps** block inside the "repeat block".

Getting Started With Scratch

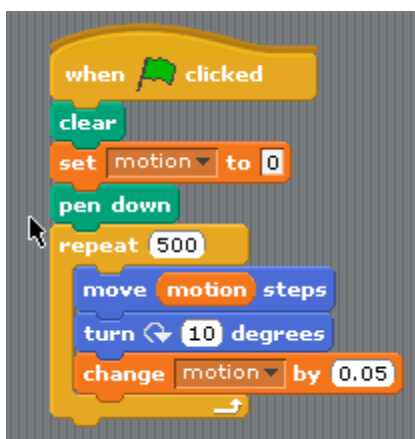
19. Drag the **motion** variable block (from the "Variables" section) and drop it into the "motion" block:



Your block should look like this:



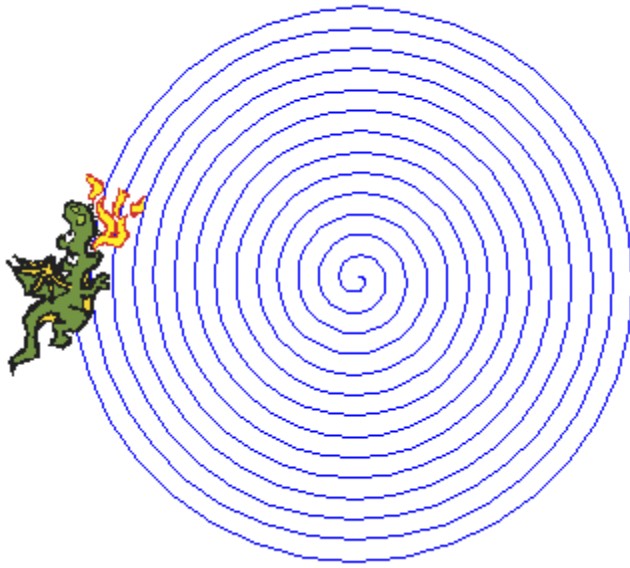
20. Complete the block by adding the following blocks:



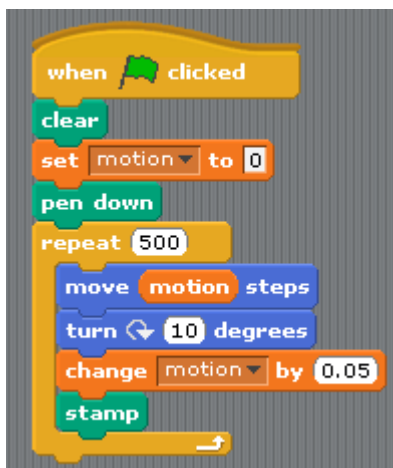
21. Click the green flag. Your dragon should draw the following pattern:

Getting Started With Scratch

motion 25.0



Now try experimenting with different shapes and drawing commands. For example, try adding a **Stamp** (in the "Pen" section) and try and work out what it does:



Session 5 - It's a Dog's Life! - Part 1

It's a Dog's Life

The dog has three things that are measured: **Weight**, **Fitness** and **Happiness**. These start out with a value of 50, and decrease every second. If any of these three values gets to zero your dog dies. If its weight goes above 100 it also dies.

You can do three things to the dog: **Feed**, **Pet** and **Walk**. The following table shows how these change the weight, fitness and happiness:

	Weight	Fitness	Happiness
Walk	-1	+1	
Pet			+1
Feed	+1		

You must keep your dog alive as long as possible!

To implement this game:

1. Run Scratch and delete the "Cat" sprite.

Add Variables

You need to add the following variables to your project **Fitness**, **Happiness**, **Weight**. To do this, repeat this instruction for each of the three variables:

2. Select **Variables** in the top left.
3. Click **Make variable**.
4. Enter the variable name (e.g. "Fitness") and click **OK**.

The variables should look like this:



The Engine

You will now create the code that will implement the "engine" that works out the dog's state of fitness, happiness and weight:

Getting Started With Scratch

5. Select the **Stage**.
6. Add a **when I receive** block, select the down arrow and select **new...**



7. Enter **Walk** for "Message name" and click **OK**.
8. Add a **change** block from "Variables" to decrease "Weight" by 1.
9. Add a **change** block from "Variables" to increase "Fitness" by 1:



10. Repeat this to create a new message called **Pet** and add this **change** block:



11. Repeat this again to create a message called **Feed** and add this **change** block:



You should now add a **forever** block that will work out the new values for "Fitness", "Happiness" and "Weight" every second:

Getting Started With Scratch



Here's some help with writing this long block: The complex bits here are creating the "if" block. To do this:

12. Add an **If / Else** block:

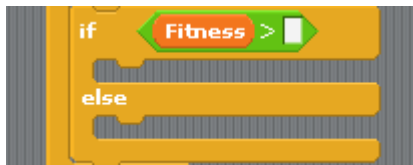
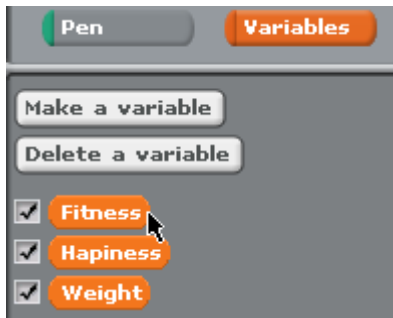


13. Add a **>** operator to the "if" part:

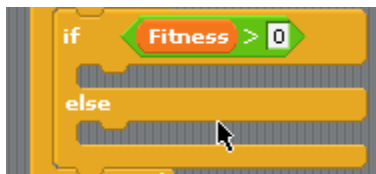


14. Drag a variable, e.g. Fitness from the "variables" section into the first space in the **>** operator:

Getting Started With Scratch



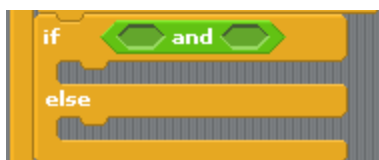
15. Enter **50** for the second box in the > operator:



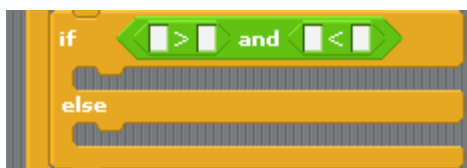
16. You will need to add a new broadcast message called "Died" as well:



17. The **if/else** block for **Weight** is a little different as it uses an **and** block. Start by adding the **and**:



18. Then use a < operator and a > operator:



19. Now fill in the blocks as shown above.

Getting Started With Scratch

The Dog Sprite

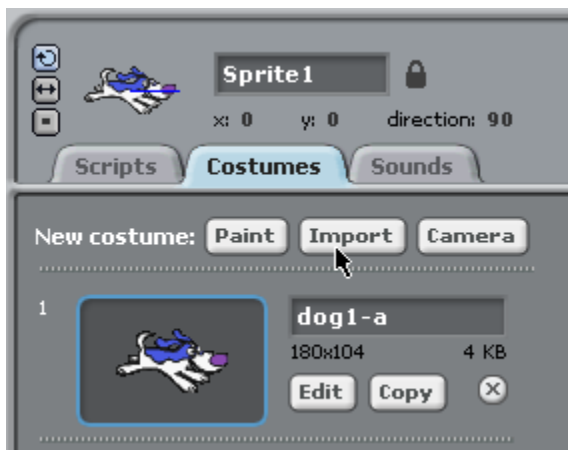
20. Choose to add a new sprite by clicking this button:



21. Select sprite **dot1-a** under "Animals".



22. Add a new costume for this sprite. Click **Costumes** and click **Import**.

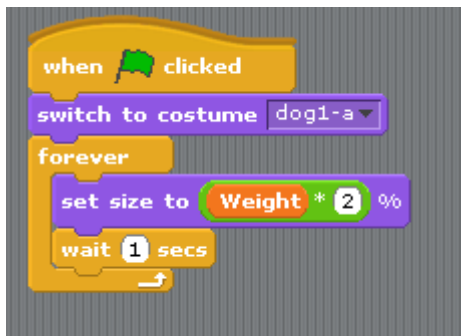


23. Select **ghost1** under "Fantasy".

Getting Started With Scratch



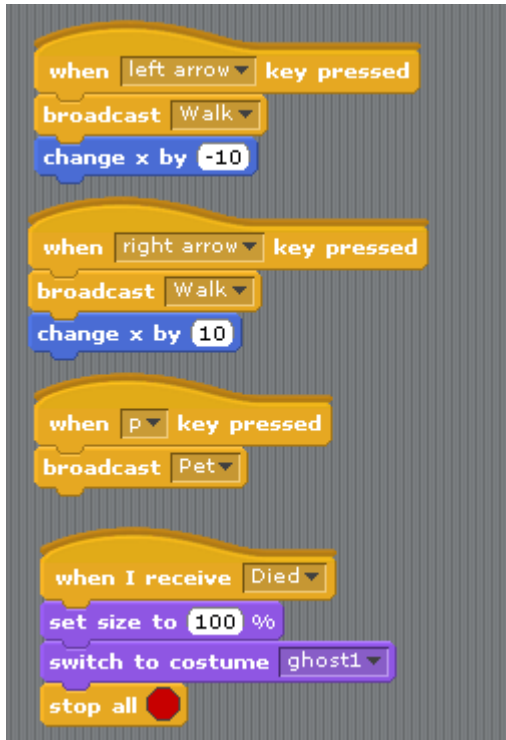
24. Add this block for the dog spite:



25. Notice that the "set size to" block uses a * operator to multiple the **Weight** variable by 2.

26. Add these other blocks to the dot spite:

Getting Started With Scratch



Dog bowl Sprite

27. Now add a feeding bowl sprite. The sprite is **cheesy-puffs** under "Things".

28. Add the following code block:



22. Click the Green Flag and try out the game.

Other things to try

Now you have the game working, you might try changing it to:

- Make sure that walking the dog does not go off the screen.
- Can you pet the dog by clicking him/her?
- How would you add a "sleep" state when the dog is asleep (can you feed/pet/walk a sleeping dog?)

Session 6 - It's a Dog's Life! - Part 2

This project follows on from the "Dog's Life" game created for Scratch in the last session. Here you will add some extra functions.

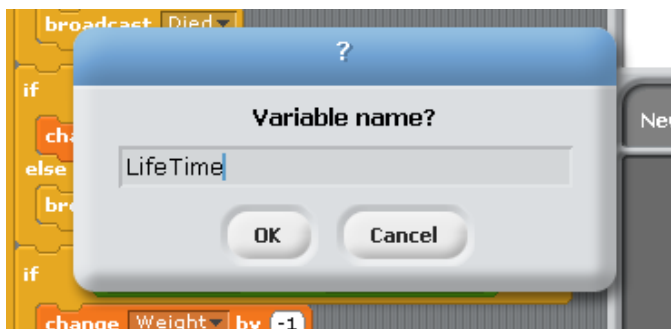
How long does your dog live?

You will now add some code to display how long your dog has been alive.

1. Make sure you select the **Stage**.

You will now add a new variable called **Timer**. To do this:

2. Select the **Variables** code block section (the orange one).
3. Click **Make a variable**.



4. Enter **Lifetime** for the "Variable name" and click **OK**.

Code blocks will be added to use the Scratch Timer. To do this you will need to reset the timer, and set the timer value into your variable "Lifetime".

Your code will be added to the stage, *not* the dog sprite.

5. Select the **Sensing** code block section (blue).
6. Put a **Reset Timer** block at the top of the green flag block for the stage.



Note: You should already have the "Green flag" block.

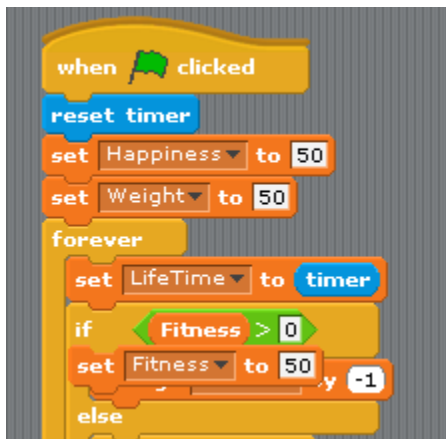
7. Select the **Variables** code block section (orange).
8. Add a **Set** variable code block under "forever" code block. Select the *Lifetime* variable in this code block:

Getting Started With Scratch



The *Lifetime* variable will be set to have the value "0". However, we want to set the value to the current value of the timer:

9. Select the **Sensing** code block section (blue).
10. Drag the **Timer** variable to replace "0" in the "set" block for "LifeTime":



You can now run your game. You will notice that the "Lifetime" variable will record how long your dog is alive for in seconds. If the dog dies, the Lifetime variable will show how long the dog lived for:



You will notice Lifetime shows the time in seconds and tenth of a second, but the tenths do not change like a stop watch. You can change the code to just display seconds:

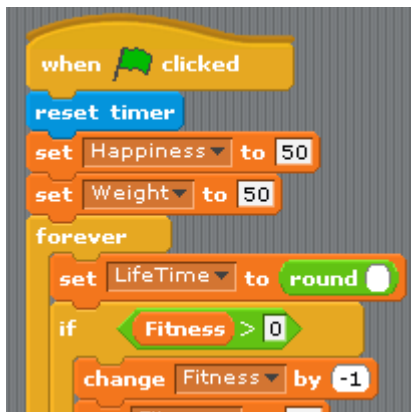
11. Remove the blue "timer" block from the set lifetime block:

Getting Started With Scratch



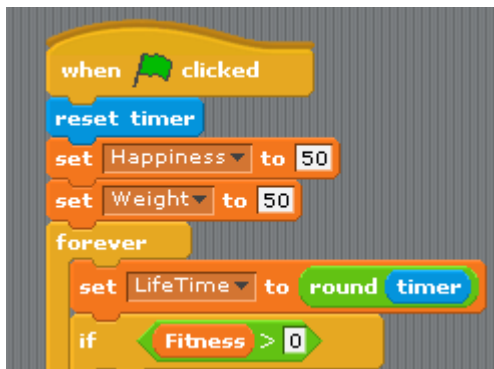
12. Select the **Operators** code block section (green).

13. Add a **round** code block into the set lifetime block:



14. Select the **Sensing** code block section (blue).

15. Drag the **Timer** variable to replace "0" in the "round" block you just added:



You should now see the time being displayed in seconds:

Getting Started With Scratch



How long does your dog live for if you don't pet, walk or feed it?

Bring the dog to feed

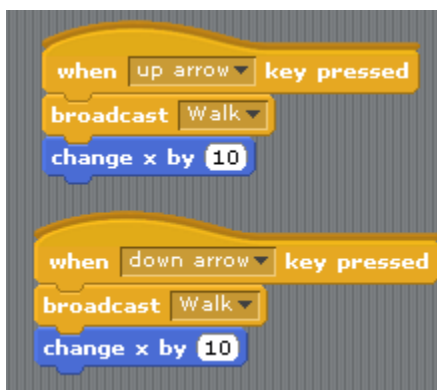
At the moment, you feed the dog by pressing the space bar. You will now feed the dog by walking to towards the bowl. When the dog is close to the bowl it will feed.

First, you will need to add "when" statements for when the up and down arrows are pressed. The code blocks for these are the same as for the left and right arrows.

16. Select the dog sprite and click the **scripts**.
17. *right-click* the "when left arrow" block and select **duplicate**:

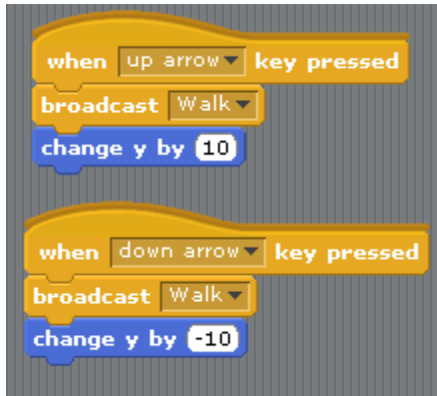


18. Click to place the copy of the code block in the "Scripts" section for the dog sprite.
19. Change the "left arrow" selection to "up arrow".
20. Repeat this to create a code block for "down arrow". You should have added these two code blocks:



21. Remove the "change x" blocks for "up arrow" and "down arrow" and add **change y by** blocks with the values shown below:

Getting Started With Scratch



Test that you can now walk the dog up and down as well as left and right.

You will now make changes to the code blocks for the *stage*:

22. Select the **stage**.
23. *Delete this code block:*



24. Add a "broadcast feed" code block to the "when I receive walk" block:



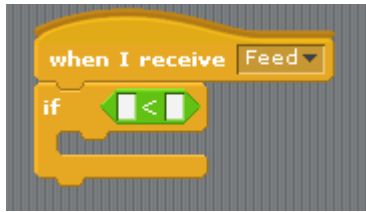
Now you will add code to the dog sprite:

25. Select the *dog sprite* and the "Scripts" tab.
26. Add this new code block:

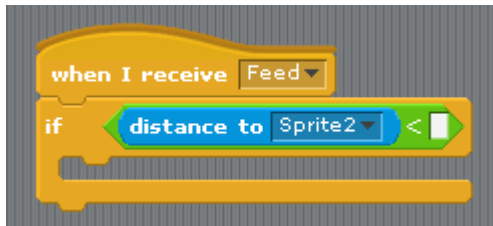


27. Add a < operator to the if block:

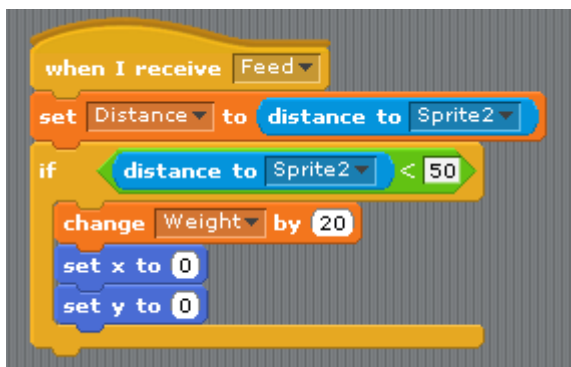
Getting Started With Scratch



28. Add a **distance to** block from the "sensing" section (blue) to the first box in the "<" operator. Select the name of the cheezy-puffs bowl (in this case, it's "Sprite2"):



29. Enter the value "50" into the second white box:
30. Increase the weight by 20 and move the dog to the center of the stage by adding a **change** code block as follows:

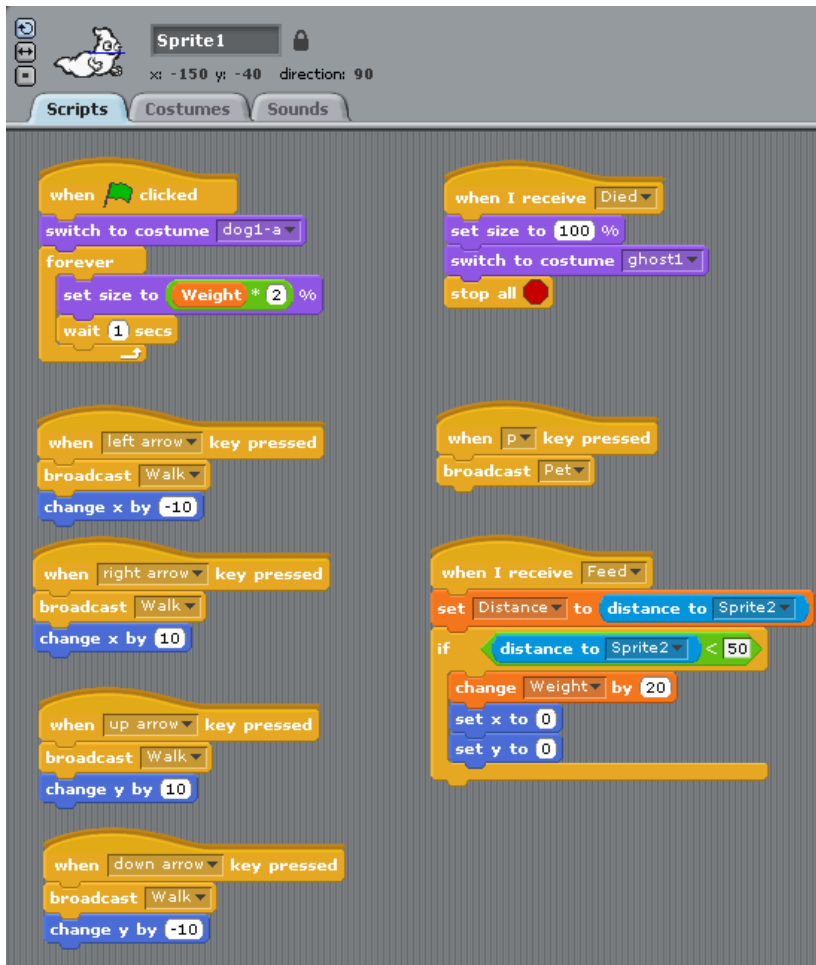


This code finds out how close the dog is to the bowl. If the distance is less than 50 the dog is fed.

31. Try running the game. Is it too difficult? How can you make it more easy?

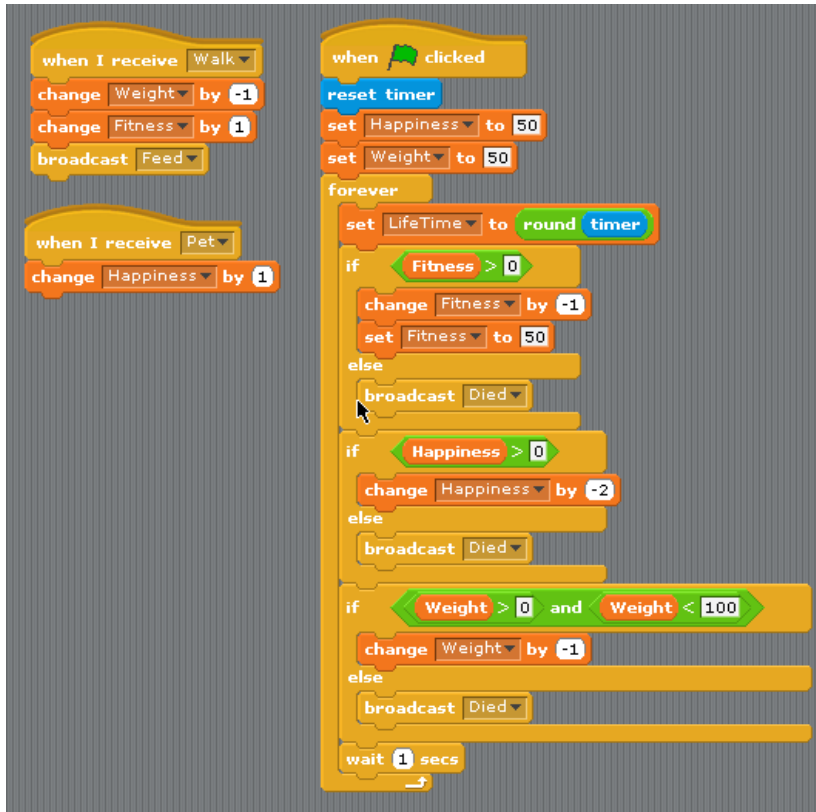
So you can check your code, here's all the code for the dog sprite:

Getting Started With Scratch



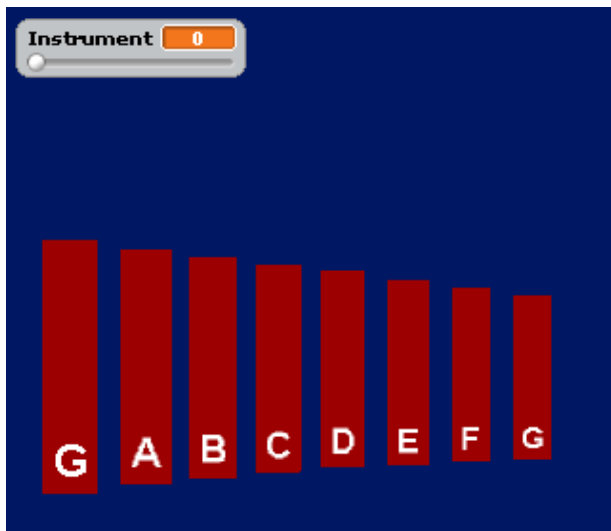
And for the stage:

Getting Started With Scratch



Session 7 - Scratch Xylophone Part 1

In this session you will create a Xylophone using Scratch. This is based on the project <http://nebomusic.net/scratchxylophone.html> by Mr. Michaud. The stage will look like the following:



1. Start Scratch. Delete the Cat.

Scratch allows you to select different instruments. You will create a new variable to control which instrument is used.

Selecting Instruments

2. Create a Variable Called **Instrument**:



Getting Started With Scratch

3. Right-click the variable on the stage and select "Slider".



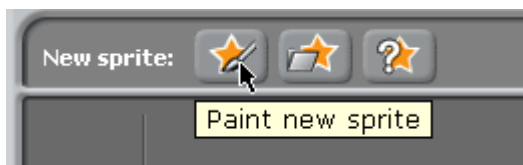
The instrument variable should now look like this on the stage:



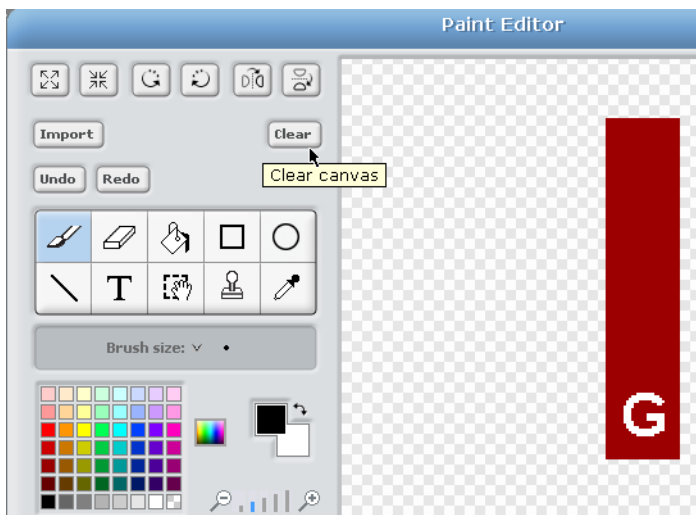
Add the keys

You will now add the first key for the Xylophone:

4. Click Paint a New Sprite.



5. Draw a Large Rectangle and fill with a color.
6. Use text tool and put a "G" in the lower part of the "Bar."



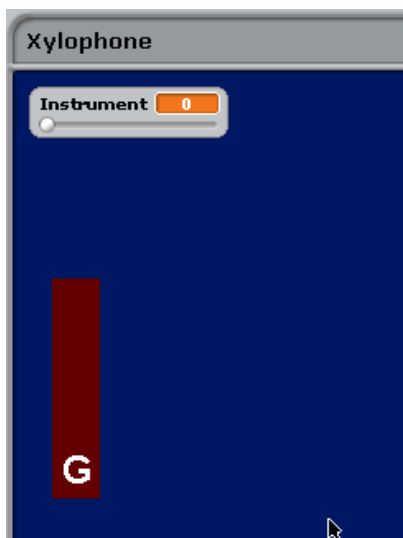
7. Click **OK**
8. Name this Sprite "**LowG**"
9. Click the "Costumes" Tab on the "**LowG**" sprite.
10. Click **Copy**
11. Click **Edit** on "costume2"
12. Use Flood Fill to make the bar a darker colour.
13. Click **OK**.

Getting Started With Scratch

The costumes should now look like this:



14. Click on the "Stage" icon in the sprite list.
15. Click on **Backgrounds** Tab.
16. Click "Edit" to edit the background for the stage.
17. Flood Fill the stage using a solid color.
18. Click **OK**.

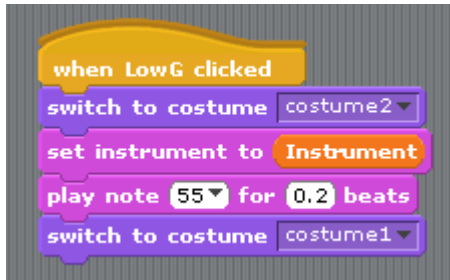


19. Add This Script to the "LowG" Sprite



20. Add this Script to "LowG" Sprite (to play when the sprite is clicked).

Getting Started With Scratch



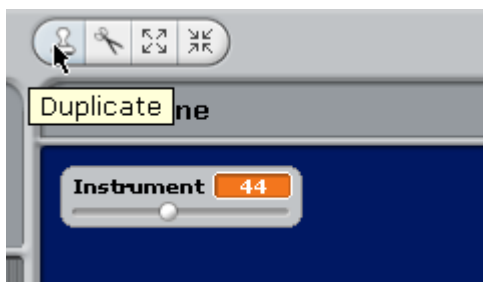
21. Add this Script to "LowG" Sprite (to play when 1 key is pressed)



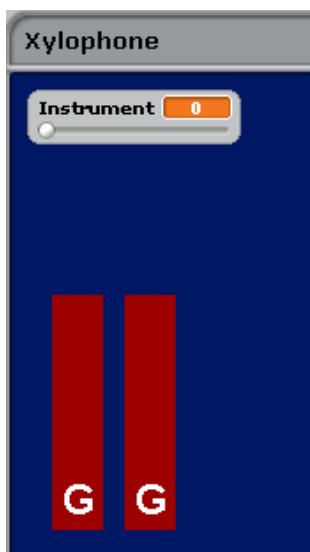
22. Click the Green Flag and Test by pressing "1" or clicking the sprite - it should play 1 note

You will now duplicate the "LowG" sprite to create the other keys:

23. Use stamp tool to copy the "LowG" Sprite:



You should now have the following sprites:

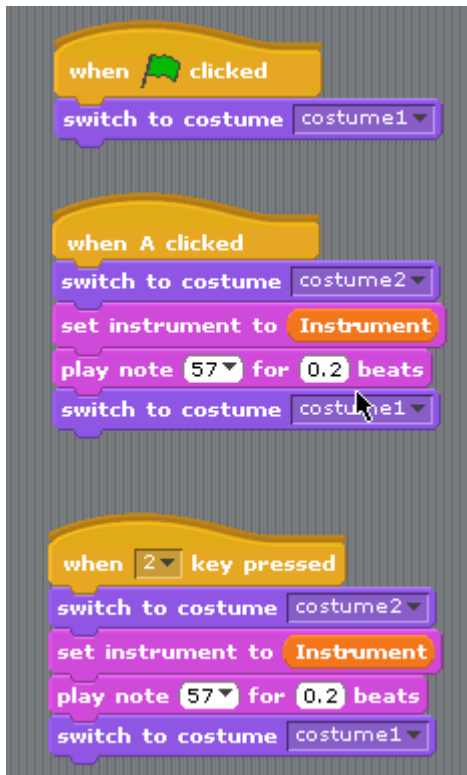


Getting Started With Scratch

24. Make the Following Changes to new Sprite:

- Use Shrinker Tool to make it smaller
- Change the sprite's name to "A"
- Change the "play note" to 57 (Pitch name A) in when clicked
- Change when 1 key pressed to when 2 key pressed
- Change the "play note" to 57 (Pitch name A) in when 2 key pressed
- Edit the Costumes so "A" is on the Bar

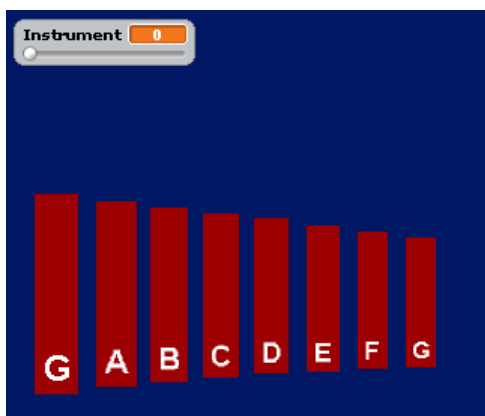
Finished Script for A Bar should look like:



25. Click the Green Flag and test the project. You should be able to play two notes.

26. Use the stamp tool and create the rest of the Xylophone You will need to carefully change the blocks to play different notes using the correct keys.

The finished stage should look like this:



Getting Started With Scratch

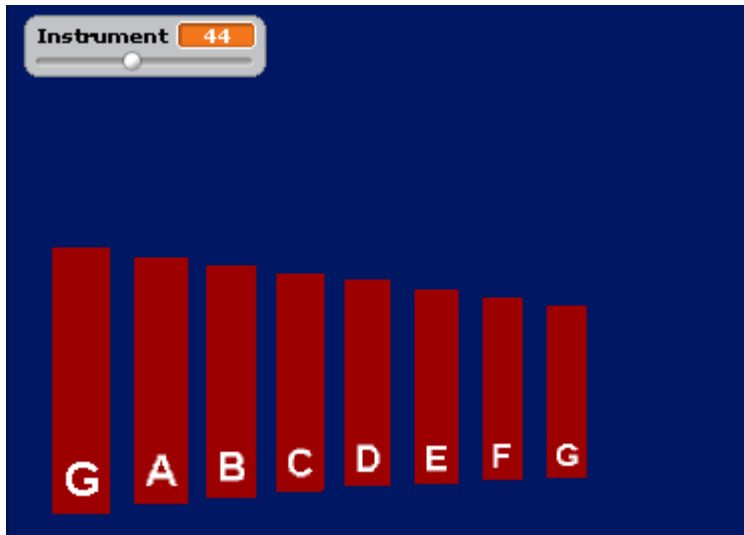
27. You should now be able to play "Mary had a little lamb" with the following notes:

B A G A B B B A A A B D D B A G A B B B B A A B A G

Session 8 - Scratch Xylophone Part 2

Using the Xylophone you created in Session 8 you will now add the ability to record the notes you press and then play them back. This will use a Scratch *list*.

The final stage will look like this:



To do this you will:

- Add a Red button sprite that starts recording the user playing. Notes are stored in a list.
- Add a Square button sprite that stops the recording.
- Add a Green Triangle sprite that triggers the Program to play back what the user had recorded

You should start with the completed Scratch project from Session 7.

Add the Recording / Playback Buttons

1. Add the following variables **NoteListItem** and **RecordOn** the Xylophone Project (you already have the "Instrument" variable):

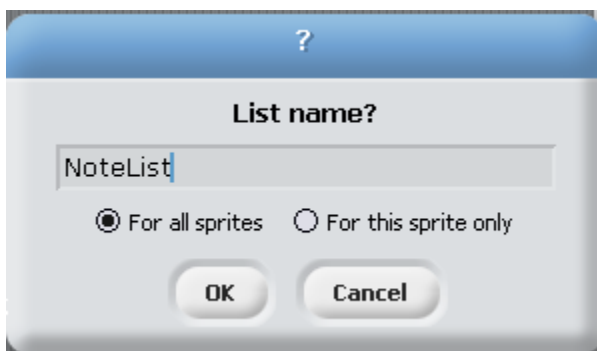


2. Add a list variable by clicking Make a list in the "Variable" block section:

Getting Started With Scratch

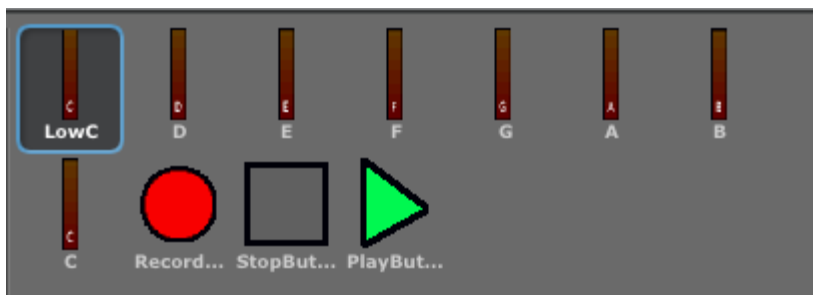


3. Type **NoteList** for the list name and click **OK**:

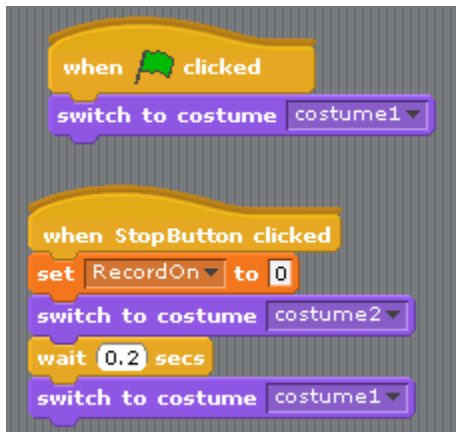


4. Add three new sprites as follows:
 - Green Play Triangle with the name **PlayButton**.
 - Red Circle for Record with the name **RecordButton**.
 - Grey Square for Stop with the name **StopButton**.

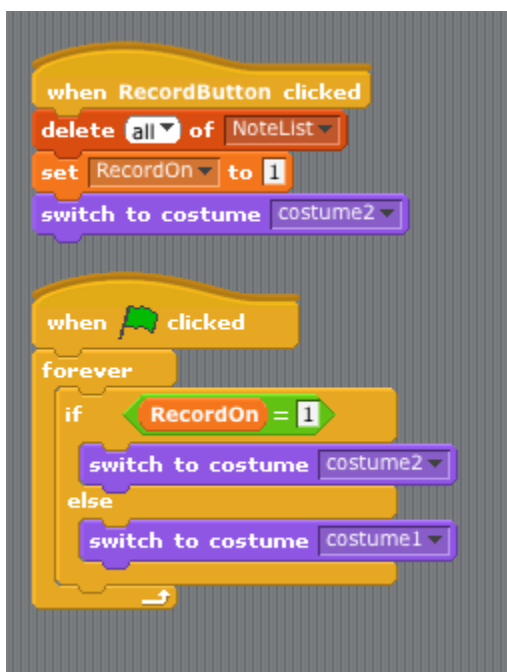
You will use the costume painter to draw them so they look like this:



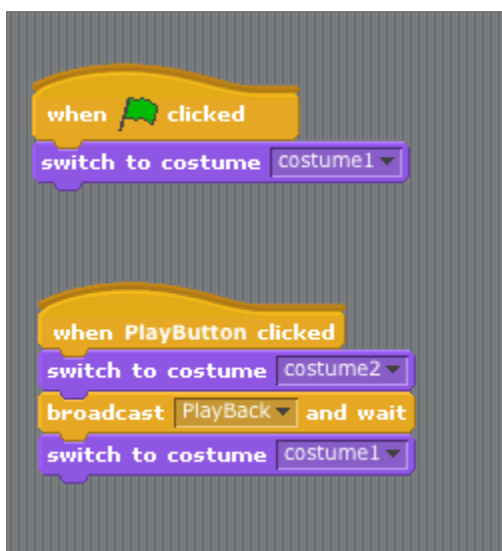
5. Add this block for the **StopButton**:



6. Add this block for the **RecordButton**:

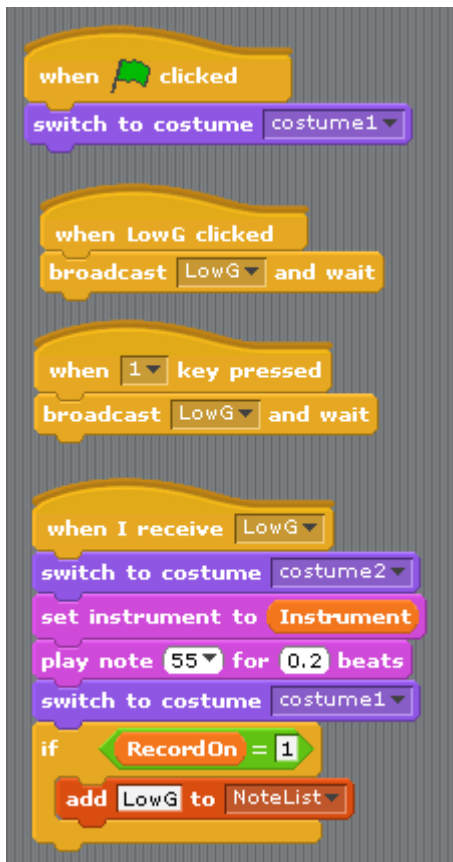


7. Add this block for the **PlaybackButton**:



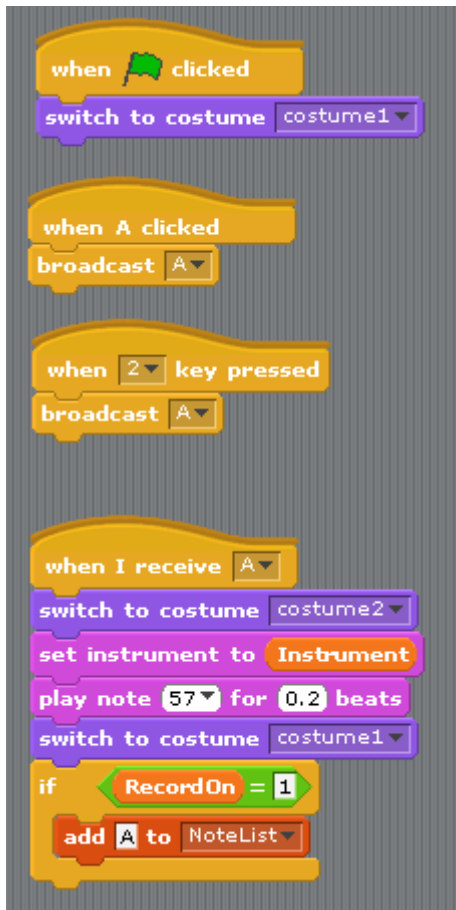
Add Recording and Playback

8. Change the scripts for the **LowG** Sprite:

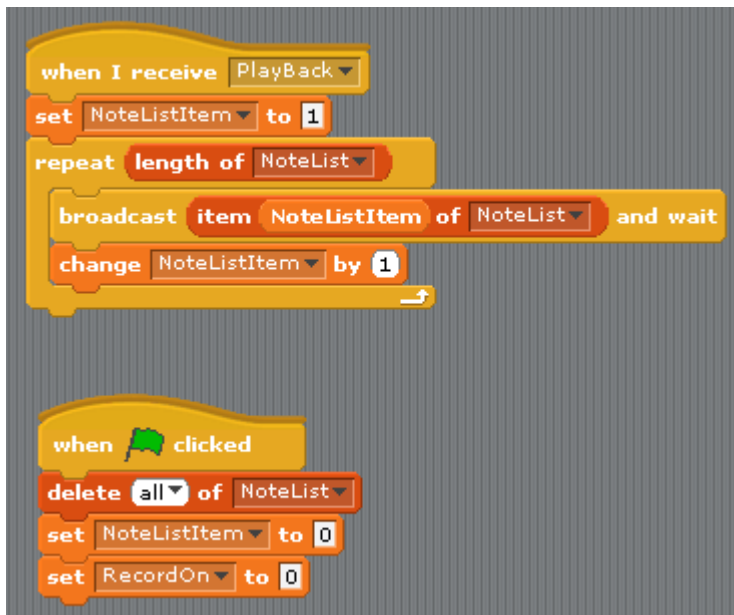


9. Change the scripts for the **A** Sprite as follows:

Getting Started With Scratch



10. Change the scripts for the rest of the bars in the program.
11. Put these scripts on the stage to control recording and playback.



12. Save and test your program

You should now be able to start recording, play some notes, stop recording and then play back!