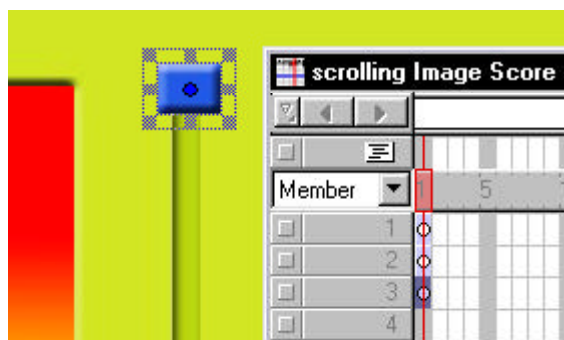


MMAST006 Multimedia Authoring Director 8

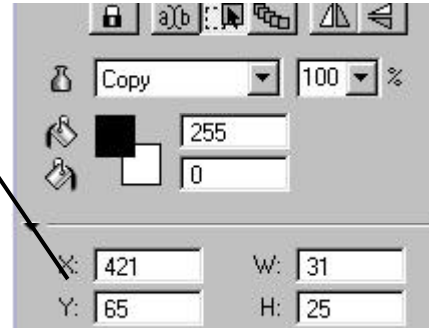
Director Lingo - Variables

In this Lesson we will learn how to use a Lingo Variable to move one image in relation to the users movement of another. The effect will be similar to a scrolling text except we will be scrolling an image instead.


- Open Director and open a new movie and give it a stage size of **640 x 480**. (Modify > Movie > Properties)
- Import the following images from the Director Resources folder: **rainbow.jpg**, **Scrollback.psd** and **Scrollbutt.jpg**. (File > Import) (When prompted import them all as 32 bit images)
- Set the Sprite Preferences to span 1 frame. (File > Preferences > Sprite)
- Drag the, **rainbow**, cast member into frame 1, channel 1 of the **Score**.
- Drag the, **Scrollback**, cast member into frame 1, channel 2 of the **Score**.
- Drag the 3rd cast member, **Scrollbutt**, onto the **Stage** and position it near the top of the scroll slider image. (It should automatically place itself in frame 1 channel 3 of the Score)

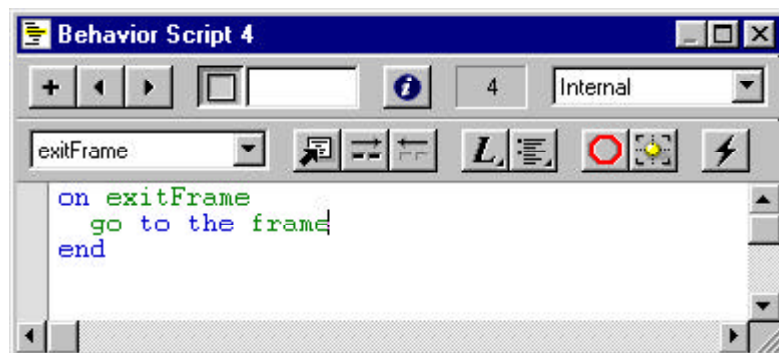


- Save your work as "Scroll" in your *Work in Progress\Director* folder.
- Position your button precisely by selecting it and typing in the following x and y co-ordinates
x: 421, y: 65 in the property inspector



Making a sprite Moveable

- Also make the button sprite moveable by clicking on the Moveable button  on the property inspector
- Double click in frame 1 of the script channel and within the **on exitFrame** handler, type in the command **go to the frame**.



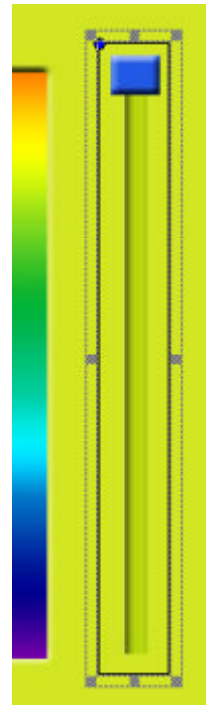
- Close the script window.
- Now test your movie so far by playing it and moving the button. You'll notice it can be dragged anywhere. So we need to constrain its movements.
- Stop the movie and click the rewind button and the button should return to where its meant to be.

Constraining the movement of a sprite



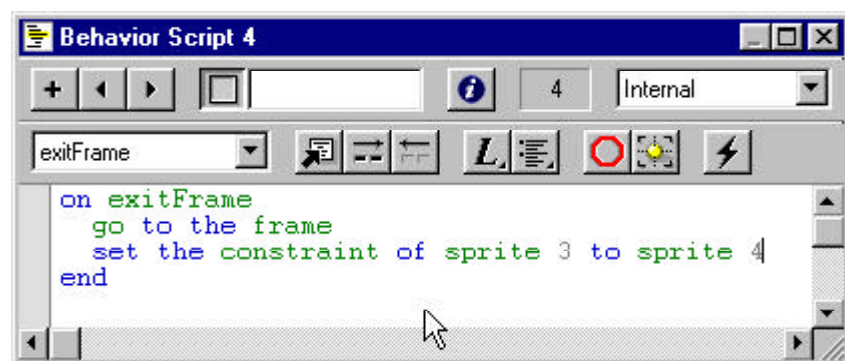
- Create an invisible rectangle around the button and the slider track in the image using the **Unfilled Rectangle** tool with the **No Line** button depressed.

If you can't find this tool Palette, go to the menu bar and select Window > Tool Palette. (Ctrl+7)



- Now add a new line of script in the Frame Script to constrain the button to the new sprite we've just created:

set the constraint of sprite 3 to sprite 4



- Now test the movie again and you'll find that the button can't be dragged outside the rectangle. You'll also notice that the rectangle is way too large for our purpose. It is the centre point of the image that is constrained and not the whole image.

Therefore the width of the rectangle needs to be 1 pixel and its top corner needs to be exactly over the position of the button.

- To work that out I set the position to x: 420, y: 65 and then the width to 1 pixel. I then visually dragged out the bottom area to suit the image. This is what I ended up with:



- Now test the movie again and see how well it works.

Now we have that sorted out we need to set up the "rainbow" image (Sprite 1) to a position from where to start.

- select sprite 1 in the score and use the Down arrow key to move it as far down as it will go without revealing its edges.
(If you try to select it and move it on the stage you will end up moving sprite 2 because it is layered on top of sprite 1)

Creating a Variable

Good, now we are all set up to write some lingo to get the rainbow image moving in relation to the movement of the scrollbar button.

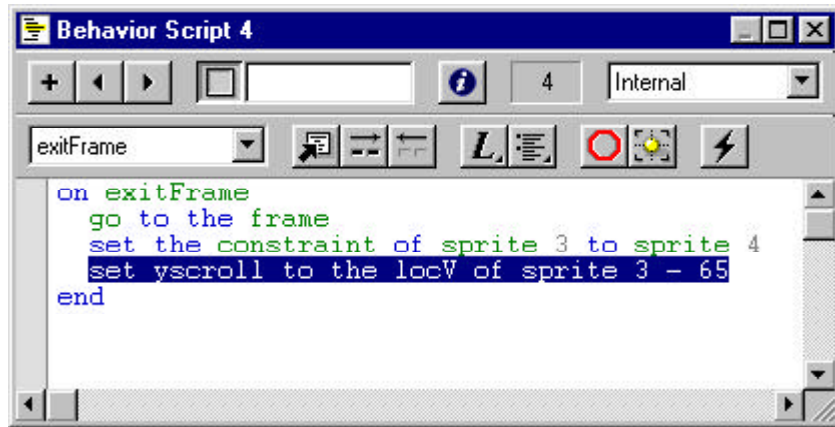
To do this we need to create a **variable** which will change value as the button is moved and therefore affects the other image.

A variable is a mathematical equation that can change if other ingredients of the equation vary. We are going to call our variable **yscroll**. (You can call it whatever you like, it is just a name) Lingo programmers often put a letter in front of their variables so they are easily recognised and prevents them from being confused with other lingo functions of commands.


We want **yscroll** to be a value of 0 when the scrollbar button is at the very top. The vertical location (locV in lingo) of the button is 65 (Remember the **y** co-ordinate)

Therefore to set the variable we add the following into the score script:


set yscroll to the locV of sprite 3 - 65



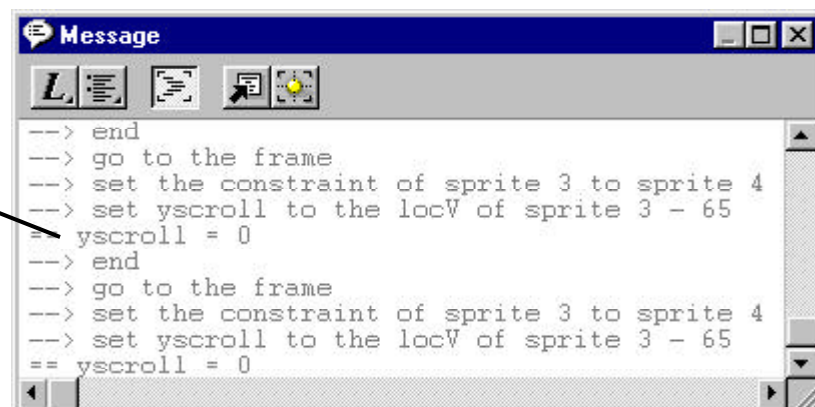
If sprite 3 is located properly then yscroll should equal 0. Lets test this...

- Open the message window by clicking on the **Message Window** button  in the toolbar. (Or from the menu bar select **Window > Message**)



- Play the movie, click the **Trace**  button. You'll see lots of lingo script flying past. This is because every time the head exits the frame it applies all the script in the frame script. Stop the movie again and read what it put into the message window.

Notice it gives you the value of yscroll



- Now play the movie again and this time move the slider button. Then stop the movie and check the new values of yscroll in the message window.

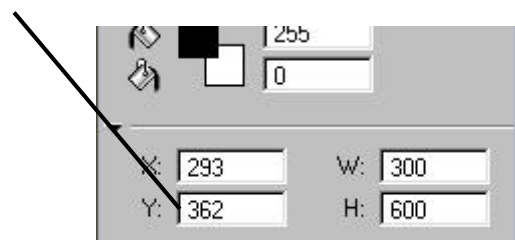
```

--> go to the frame
--> set the constraint of sprite 3 to sprite 4
--> set yscroll to the locV of sprite 3 - 65
== yscroll = 50
--> end
--> go to the frame
--> set the constraint of sprite 3 to sprite 4
--> set yscroll to the locV of sprite 3 - 65
== yscroll = 50
--> end

```

Now we know our variable is working, lets apply the value of our yscroll variable to the rainbow image:

- Check the y co-ordinate of the rainbow image. (Click on sprite 1 in the score and check the value in the property inspector)



..Mine is 362.

This value is the present vertical location (locV) of sprite 1.

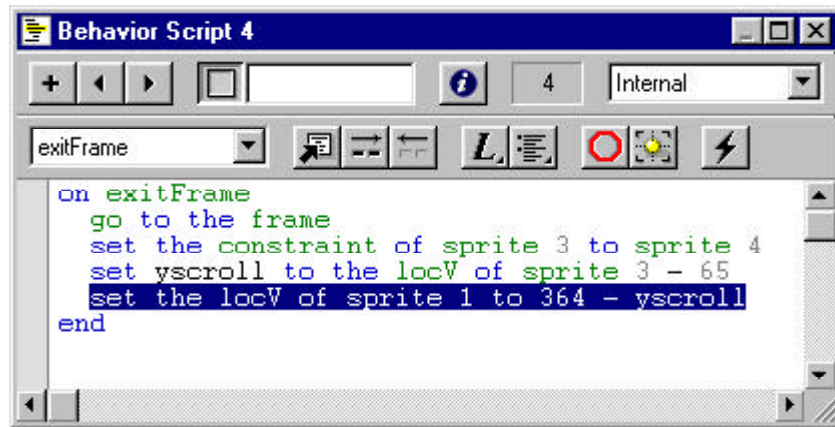
- Now for the application of the variable equation:

set the locV of sprite 1 to 362 - yscroll

This means that the vertical location of sprite 1 is 362 minus the value of yscroll. So when the value of yscroll increases, the vertical location changes to the new value.

(Eg: if yscroll = 10 the locV of sprite 1 = 352)

- Lets try it..



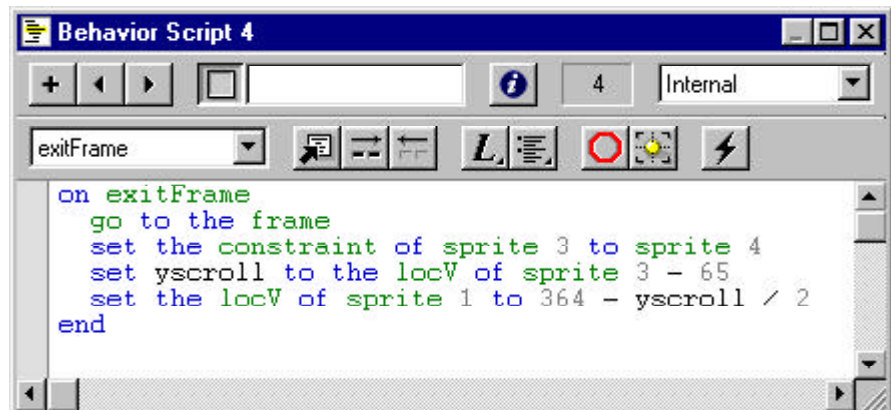
- Play your movie and see if it works.

You will now notice that the rainbow image is too short and goes exposes the background.

There are a couple of ways I can suggest to overcome this:

- One is to **divide** the value of yscroll by 2.

Eg: **set the locV of sprite 1 to 364 - yscroll/2**



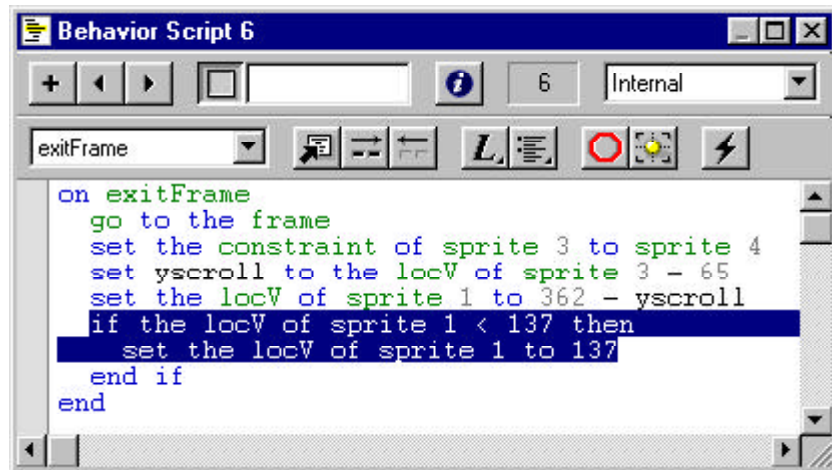
...But this leaves it a bit short to the end.

- Try playing with the divide figure to get a more accurate scroll.

• The other way would be to stop the movie and move the rainbow image up to its upper limit and then put that value into the following **If and Then** statement:

**if the locV of sprite 1 < 137 then
set the locV of sprite 1 to 137**

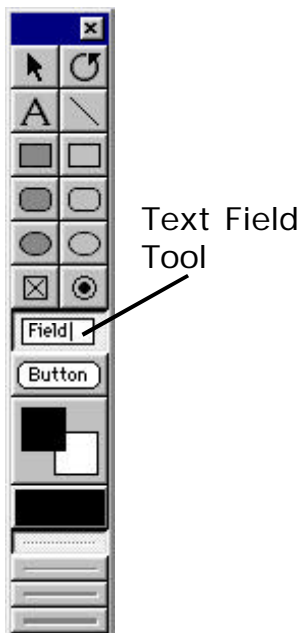
- Try this one and see the difference...



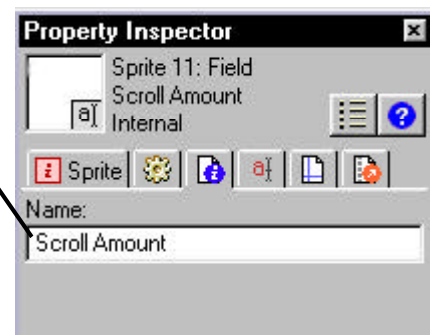
I prefer the previous way because it is a smooth scroll from top to bottom.

After a bit of experimentation I ended up with a value of 1.55

Now to finish off I will show you how to put a variable value into a text field for the user to see. This has endless possibilities for multimedia content such as keeping the track of a students progress or the score of a game.

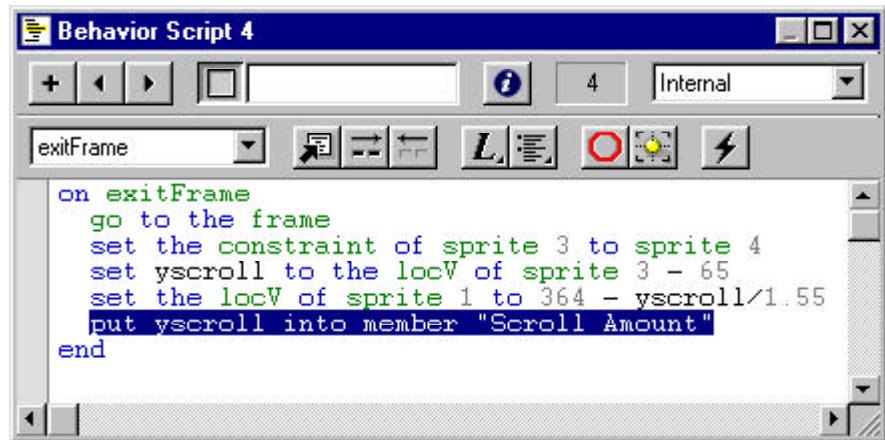


- Create a text field on the stage in the top left hand corner and leave it empty.
- Make its background transparent. (Sprite properties)
- Find and select the field sprite in the Cast Window and click on the Cast Member Properties tab. Name it Scroll Amount.



- Add the following line to the Frame Script:

put yscroll into member "Scroll Amount"



Notice the quotation marks around the word Scroll Amount. This is to distinguish the name of a cast member from other lingo commands.

- Test your movie and see how it works.
- Save your movie for future reference.

In this lesson you have learnt how to:

- Precisely position and size sprites using the properties inspector.
- Make a sprite moveable.
- Constrain the movements of a sprite to the boundary of another sprite.
- Create a variable equation.
- Have the movement of a sprite alter the variable.
- Have the variable to affect other sprites.