Gravity Mouse Chaser

Chris MacGregor for MacGregor Media - chris@macgregor.net

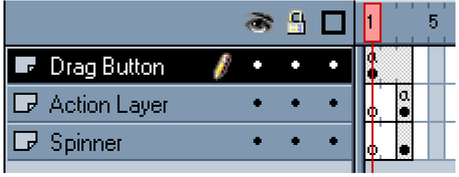
- ***DOWNLOAD THE .FLA***
- **SKILL LEVEL: VERY ADVANCED!!!**
- Must know how to use actions
- Know how to create symbols and use them well

This is a very cool effect that we are starting to see a lot in Flash. It's limitations are endless. Below you will see how to add gravity to Flash using a mouse chaser. The spinner follows your mouse according to how fast your move your mouse in a direction. Very cool!

TUTORIAL

Note: Since this is an advanced tutorial, there won't be many screenshots, as you should already know how to do most of the work. This lesson will provide you with the code you need to know and the concept to make gravity work.

Screenshot of the frames you will be creating in this tutorial:



1. Create a new movie in Flash

2. Create a new button symbol (insert > new symbol) Call this button Drag.

3. In the hit, or fourth frame, of the button image draw a small circle. It should be centered at the middle of the button.

4. In the first frame of the main scene, place this button in the middle of the scene. Because there is nothing on the Up, Over or Down frames of the button, Flash will display this button as a shaded object.

5. With the button selected, convert it to a movie clip symbol (insert > convert to symbol) name the symbol drg1.

6. Give the symbol an instance name (modify > instance) (I named mine drg1).

7. Create an action for the frame with the script: **Start Drag ("/drg1", lockcenter)**. This centers the invisible button at the center of the mouse when the movie starts playing.

8. Create a new layer, let's call it 'Spinner'.

9. Create the symbol that you wish to react to the mouse in the third frame of this layer. For this tutorial I have made the pin-wheel spinner. I called it Spinner.

10. Give the symbol an instance name (modify > instance) (I named mine spin1).

11. Create a new layer, let's call it 'Action Layer'.

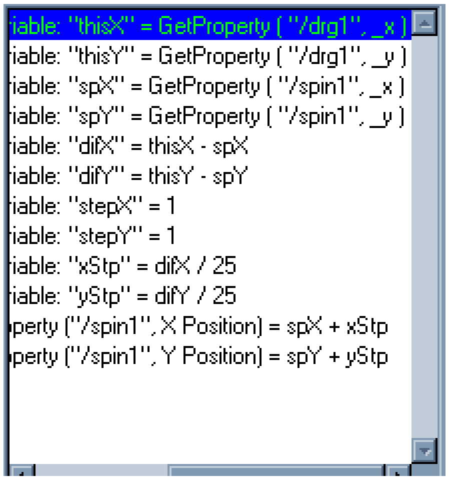
12. In the third frame of the Action Layer, create a new Movie Clip Symbol (insert > new symbol). Name this movie clip 'action'.

13. Open this Movie Clip for editing. Create an action in the first frame.

    In a nut shell this is what the script does:

    a. Where is the Mouse?
    b. Where is the Spinner?
    c. How far apart are they?
    d. How fast should the Spinner move?
    e. Move the Spinner.

    That all seems pretty simple, but converting those commands into Flash's scripting is a little confusing at best. We will take things step by step.



```
iable: "thisX" = GetProperty ( "/drg1", _x )
iable: "thisY" = GetProperty ( "/drg1", _y )
iable: "spX" = GetProperty ( "/spin1", _x )
iable: "spY" = GetProperty ( "/spin1", _y )
iable: "difX" = thisX - spX
iable: "difY" = thisY - spY
iable: "stepX" = 1
iable: "stepY" = 1
iable: "xStp" = difX / 25
iable: "yStp" = difY / 25
perty ("/spin1", X Position) = spX + xStp
perty ("/spin1", Y Position) = spY + yStp
```

NOTE: When setting variables in flash you have **String Literal** abc and **Expression** =. These are VERY important. I would **seriously** suggest [downloading the .fla](#) with this tutorial and paying CLOSE attention to where he uses string & expression statements. Otherwise it won't work!

a. **Where is the Mouse?**

To find the mouse we need to get the x and y properties of the invisible button we made and called drg1. Because we want to remember this information we will assign variables to both the x and y coordinates as such:

Set Variable: "thisX" = GetProperty ("/drg1", _x)

Set Variable: "thisY" = GetProperty ("/drg1", _y)

b. **Where is the Spinner?**

Finding the spinner is identical to finding the mouse, but now we are looking for the x and y coordinates of the instance named spin1:

Set Variable: "spX" = GetProperty ("/spin1", _x)

Set Variable: "spY" = GetProperty ("/spin1", _y)

c. **How far apart are they?**

Remember sleeping through math in High School? This is what you missed. To find the distance of two points on a grid you employ the distance formula.

distance^2 = (X2 - X1)^2 + (Y2 - Y1)^2

Now we are not that concerned with figuring the exact distance, so we are going to use a simplified version of the formula and again save the results as variables:

Set Variable: "difX" = thisX - spX

Set Variable: "difY" = thisY - spY

d. **How fast should the Spinner move?**

This questions is a bit misleading, it should be 'How far should the Spinner move?', as the speed of the Spinner is defined by the size of steps it takes. To get this information we divide the dif* variables by a set number (The smaller the number the faster it moves. For this example I have used 25, however you should try different speeds to see what you like.):

Set Variable: "xStp" = difX / 25

Set Variable: "yStp" = difY / 25

e. **How fast should the Spinner move?**

This questions is a bit misleading, it should be 'How far should the Spinner move?', as the speed of the Spinner is defined by the size of steps it takes. To get this information we devide the dif* variables by a set number (The smaller the number the faster it moves. For this example I have used 25, however you should try different speeds to see what you like.):

Set Variable: "xStp" = difX / 25

Set Variable: "yStp" = difY / 25

f. **Move the Spinner.**

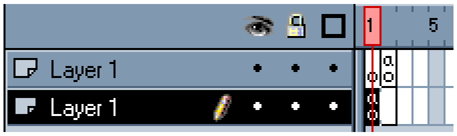
All that code above now gets put into action. These two lines tell the Spinner to reposition itself to a new set of cordinates. We do this by setting the properties of the names instance spin1:

Set Property ("/spin1", X Position) = spX + xStp

Set Property ("/spin1", Y Position) = spY + yStp

14. Add a second layer to the action movie clip, and a second frame to both layers.

15. Insert a blank key frame in the second frame of the second layer.

16. Either repeat all the scripting from step 13, or copy the actions from the first frame of the first layer (where all that scripting went) and paste it into the second frame of the second layer (edit > copy frames) then (edit > paste frames).
    Here is a screen shot of the Action movie clip:
    

17. Return to the main scene and add a stop action to the third frame on the Action Layer. Make sure that your Action movie clip is in the third frame of this scene.

18. Test your movie.

Last Updated 12/09/99