

Ink Effects

You can apply ink effects to individual sprites in the Score, or use Lingo to change the ink effect of a sprite on the fly. The Lingo usage is described below.

What the different ink effects do:

Copy is the default ink, and is useful for backgrounds or for sprites that do not appear in front of other artwork. If the cast member is not rectangular, a white box appears around the sprite when it passes in front of another sprite or is displayed on a non-white background. *Sprites with the **Copy** ink animate faster than sprites with any other ink.*

Matte removes the bounding box (rectangular area) around a sprite. Artwork within the boundaries is opaque. **Matte** functions much like the Lasso in the Paint window, in that the artwork is outlined rather than enclosed in a rectangle. ***Matte**, like **Mask**, uses more memory than the other inks, and sprites with this ink animate more slowly than other sprites.*

Background Transparent makes the pixels in the background color of the selected sprite appear transparent and permits the background to be seen. *This effect uses more memory and may make your sprite animate more slowly.*

Transparent makes all colors transparent so you can see the artwork through it.

Reverse reverses overlapping colors. A pixel that was originally white becomes transparent and lets the background show through unchanged. **Reverse** is good for making custom masks.

Ghost is useful for reversing black and white. When it is applied to the foreground sprite, any black pixel turns the pixel beneath it white. Anything white becomes transparent.

Not Copy, Not Transparent, Not Reverse, and Not Ghost are variations of the effects above. The foreground image is first reversed, then the **Copy, Transparent, Reverse, or Ghost** inks are applied. These are good for odd effects. Like **Transparent**, the **Not Transparent** ink is good for reversing black and white. Just choose **Not Transparent**, select a white fill, then draw a rectangle on Stage on top of the artwork you want to reverse.

Mask ink allows you to define exactly what parts of a sprite are transparent and opaque. For **Mask** ink to work, you must place a 1-bit mask cast member in the Cast window position immediately following the cast member to be masked. The black areas of the mask make the sprite opaque and the white areas make the sprite transparent. This ink is especially useful for sprites in which you want some white areas to be transparent, and some opaque.

For example, to show a white car you would want the white body of the car to be opaque and the windows to be transparent. To create a mask, make a copy of the car in the next cast position, convert it to 1-bit color depth with the Transform Bitmap command, and then fill in the body of the car with black. In the Score, apply **Mask** ink to the sprite of the car. The body of the car becomes opaque and the windows transparent.

Blend ensures that the sprite uses the blend percentage specified in the Sprite Properties dialog box.

Darkest compares pixel colors in the foreground and background, and uses whichever pixel color is darkest.

Lightest compares pixel colors in the foreground and background and uses whichever pixel color is lightest.

Add creates a new color that is the result of adding the color value of the foreground sprite with the color value of the background sprite. If the value of the two colors exceeds the maximum color value, the numbering begins again at 1.

Add Pin is similar to **Add**. The foreground sprite's color is added to the background sprite's color, but the value of the new color cannot exceed the maximum color value.

Subtract subtracts the value of the foreground sprite's color from the value of the background sprite's color to arrive at the new color. If the color value of the new color is less than the minimum color in the color scale, the new color is determined by wrapping around and starting at the top of the color scale.

Subtract Pin subtracts the color value of the foreground sprite from the value of the background sprite. The value of the new color does not wrap around the color scale.

Tip:

Mask and **Matte** use twice the memory of any other ink because Director has to internally create a duplicate of the artwork.

(from Director Help)

Lingo usage:

You can assign an ink effect to a sprite using a Lingo script. This script may be in the Script channel of the Score, or in a behavior or sprite script, or in a global Movie script.

Syntax: `the ink of sprite spriteNum`

A numeric value indicates the ink effect applied to the sprite ***spriteNum***. May be read or set.

Code	Ink
0	Copy
1	Transparent
2	Reverse
3	Ghost
4	Not copy
5	Not transparent
6	Not reverse
7	Not ghost
8	Matte
9	Mask
32	Blend
33	Add pin
34	Add
35	Subtract pin
36	Background transparent
37	Lightest
38	Subtract
39	Darkest

Examples of usage:

Stores the ink effect applied to sprite 3 in the variable `currentInk` :

```
put the ink of sprite 3 into currentInk
```

Sets sprite (`i + 1`) to the **matte** ink effect (code **8**):

```
set the ink of sprite (i + 1) to 8
```