Spring Hill Tech Club Volume 2, # 4 11 Oct 2007

Tech News

Flashlight simulation

Kevin Karplus

Today's *Tech News* is dedicated to explaining two simple programs that Abe and I wrote together to get a particularly nice animation effect—the illusion of a flashlight beam lighting up an image. Scratch does not provide sophisticated lighting simulation, so we fake it with a clever trick (thanks to Scratch user Jens, who wrote a program using a similar effect).

The sprites can be thought of as a stack of clear plastic sheets with opaque images on them. What you see is the topmost image of the stack of sheets.

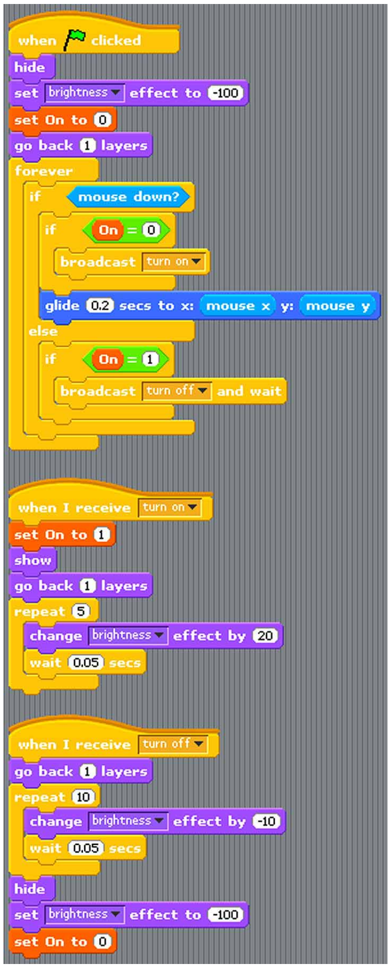
The idea is to use a plain black background for the stage, and to have a black-on-transparent image on a sprite in front of it. Since everything is black, nothing can be seen. But if we slip a white sprite behind the image sprite, the image will appear where the white sprite is.

There are two parts to the program: one is for having the white sprite follow the mouse around, the other is for turning the flashlight on when we press the mouse button, and off when we release the button. The two variants of the program differ mostly in how we combine these two parts.

Both programs have to make sure that the image sprite is in front of the white sprite, and they use glides for following the mouse pointer so that the beam will appear to move smoothly.

Which program do you like better?

What limitations do you see to this way of producing the illusion of a flashlight beam?



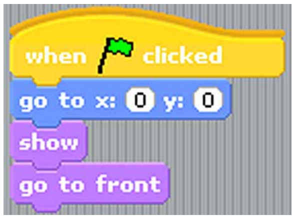
```
when ⚑ clicked
hide
set brightness effect to -100
set On to 0
go back 1 layers
forever
  if mouse down?
    if On = 0
      broadcast turn on
    glide 0.2 secs to x: mouse x y: mouse y
  else
    if On = 1
      broadcast turn off and wait

when I receive turn on
set On to 1
show
go back 1 layers
repeat 5
  change brightness effect by 20
  wait 0.05 secs

when I receive turn off
go back 1 layers
repeat 10
  change brightness effect by -10
  wait 0.05 secs
hide
set brightness effect to -100
set On to 0
```

Flashlight 1 uses one script to do all the interaction with the mouse. It has a variable to keep track of whether the light is on or not, and sends broadcast messages to turn the light on when the mouse is pressed and the light is off, and to turn the light off when the mouse is pressed and the light is on. Separate scripts turn the light on slowly or off slowly.

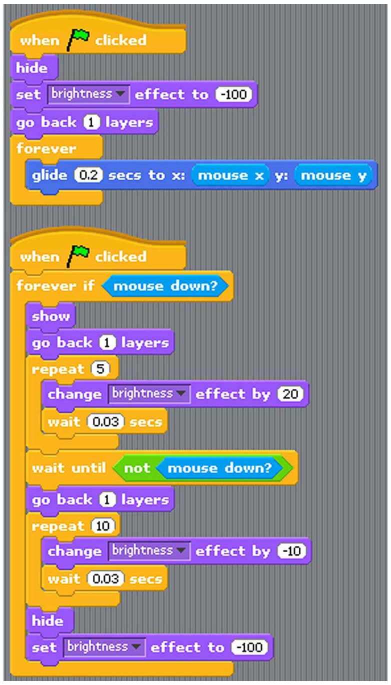
If we use a radial gradient fill for the white sprite (white in the center, black on the edges), we can make it look a bit more like a flashlight beam.



```
when ⚑ clicked
go to x: 0 y: 0
show
go to front
```

The image sprite has a simple initial script to make sure that it is in the right position and visible at the front.

Flashlight 2 uses one script to move the white sprite and another to turn the light on and off. It uses wait blocks to figure out when the light needs to be turned on and off, rather than polling the mouse button every time it changes position.



```
when ⚑ clicked
hide
set brightness effect to -100
go back 1 layers
forever
  glide 0.2 secs to x: mouse x y: mouse y

when ⚑ clicked
forever if mouse down?
  show
  go back 1 layers
  repeat 5
    change brightness effect by 20
    wait 0.03 secs
  wait until not mouse down?
  go back 1 layers
  repeat 10
    change brightness effect by -10
    wait 0.03 secs
  hide
  set brightness effect to -100
```