

Sprite Byte Tutorials Lesson Four: Moving A Sprite

by Alyce Watson <http://alycesrestaurant.com/>

- [Alyce](#)

Table of Contents

[Sprite Byte Tutorials Lesson Four: Moving A Sprite](#)

[Three Easy Ways to Move a Sprite](#)

[Updating the Screen](#)

[Locate a Sprite](#)

[SpriteMoveXY](#)

[SPRITEXY?](#)

[SpriteTravelXY](#)

[SpriteXY](#)

Three Easy Ways to Move a Sprite

Liberty BASIC will move a sprite for you if you issue the SPRITEMOVEXY command or the SPRITETRAVELXY command. You can move the sprite yourself with the SPRITEXY command. You can find out where the sprite is located at any time by issuing the SPRITEXY? command. These commands are explained below and there is a demonstration program for each command.

Updating the Screen

The sprite does not appear in a new location until you update the screen with a DRAWSPRITES command!

The following demonstration programs require bitmaps in your Liberty BASIC sprites folder and they must be run from the Liberty BASIC root directory.

Locate a Sprite

By default, a sprite is located at the upper left corner of the display. If you do not change his location, he is at X equals 0, Y equals 0. The following small demo program simply adds a sprite and places him at X equals 10, Y equals 30. Be sure to notice the two ways this can be done. The values 10 and 30 can be hard coded, like this:

```
'example using hard-coded values
#w.g "spritexy guy 10 30"
```

The values can be variables. If you use variables in sprite commands, be sure to place them outside of the double quotation marks. Set them off with semi-colons. You **must** preserve the blank spaces inside of the quotation marks. It is done like this:

```
'example using variables
spriteX = 10 : spriteY = 30
#w.g "spritexy guy ";spriteX;" ";spriteY
```

Here is the program. Be sure to run it from the Liberty BASIC directory so that it can "find" the bitmaps.

```
'add a sprite and locate it at x=10, y=30

nomainwin
loadbmp "smiley", "sprites\smiley.bmp"
loadbmp "landscape", "sprites\bg1.bmp"
WindowHeight = 300 : WindowWidth = 400

graphicbox #w.g, 0, 0, 400, 300
open "Moving Sprite 1" for window_nf as #w
    #w "trapclose [quit]"
    #w.g "down"
    #w.g "background landscape"
    #w.g "addsprite guy smiley"

    'example using variables
    spriteX = 10 : spriteY = 30
    #w.g "spritexy guy ";spriteX;" ";spriteY
```

```
'example using hard-coded values
'#w.g "spritexy guy 10 30"

#w.g "drawsprites" 'update screen

wait

[quit]
  unloadbmp "landscape"
  unloadbmp "smiley"
  close #w : end
```

SpriteMoveXY

SpriteMoveXy causes a sprite to move a specified number of pixels in the x direction, and a specified number of pixels in the y direction each time a DRAWSPRITES command is issued to update the display. The helpfile gives you examples of usage with hard-coded values and with variables.

```
print #w.g, "spritemovexy SpriteName 5 5";
OR
x=5:y=5
print #w.g, "spritemovxy SpriteName ";x;" ";y
```

The values can be positive or negative. If the X value is negative, the sprite moves to the left, instead of to the right each time the DRAWSPRITES command is issued. If the Y value is negative, the sprite moves up rather than down.

Here is a demo program that uses SPRITEMOVEXY to move the smiley to the left and downward. Each time the display is updated with DRAWSPRITES, the smiley moves 5 pixels in the X direction and 3 pixels in the Y direction.

The demo uses a TIMER to cause the display to be updated every 200 milliseconds. Remember, the sprite won't appear in its new location until a DRAWSPRITES command is issued, that's why we are updating the display with a timer event. Notice that if we let the program run for a while, the sprite moves off the visible portion of the display. We'll handle that problem in the next demo.

```
'add a sprite and locate it at x=10, y=30
'make sprite move with spritemovexy
nomainwin
loadbmp "smiley", "sprites\smiley.bmp"
loadbmp "landscape", "sprites\bgl.bmp"
WindowHeight = 300 : WindowWidth = 400

graphicbox #w.g, 0, 0, 400, 300
```

```
open "Moving Sprite 2" for window_nf as #w
  #w "trapclose [quit]"
  #w.g "down"
  #w.g "background landscape"
  #w.g "addsprite guy smiley"

  #w.g "spritexy guy 10 30"

  'make sprite move 5 in x direction
  'and 3 in y direction each time
  'the screen is updated with drawsprites
  #w.g "spritemovexy guy 5 3"

  #w.g "drawsprites" 'update screen

  timer 200, [update]

wait

[update]
  #w.g "drawsprites"
  wait

[quit]
  timer 0
  unloadbmp "landscape"
  unloadbmp "smiley"
  close #w : end
```

SPRITEXY?

In the preceding demo, the sprite is free to move off the visible portion of the screen. We can handle that problem by asking Liberty BASIC for its location each time we update the screen.

SPRITEXY? obtains the coordinates of the sprite and places them into the variables specified. The helpfile shows you two ways to do this. The first example issues the command on one line, then issues an INPUT statement on the next line. The second example shows you how to do it all on one line, by placing the variables inside of the quotation marks.

```
print #w.g, "spritexy? SpriteName"
input #w.g, x, y
OR
print #2.g "spritexy? SpriteName x y"
```

In the improved demo below, we check the sprite's location each time **DRAWSPRITES** is issued. When the sprite has reached either the right side or the bottom of the graphicbox, we turn off the timer to stop

him from moving further.

```
'add a sprite and locate it at x=10, y=30
'make sprite move with spritemovexy
'stop sprite when he gets to the edge
nomainwin
loadbmp "smiley", "sprites\smiley.bmp"
loadbmp "landscape", "sprites\bgl.bmp"
WindowHeight = 300 : WindowWidth = 400

graphicbox #w.g, 0, 0, 400, 300
open "Moving Sprite 3" for window_nf as #w
    #w "trapclose [quit]"
    #w.g "down"
    #w.g "background landscape"
    #w.g "addsprite guy smiley"

    #w.g "spritexy guy 10 30"

    'make sprite move 5 in x direction
    'and 3 in y direction each time
    'the screen is updated with drawsprites
    #w.g "spritemovexy guy 5 3"

    #w.g "drawsprites" 'update screen

    timer 200, [update]

wait

[update]
    #w.g "spritexy? guy X Y"
    if (X > 360) or (Y > 240) then
        timer 0 'stop him from moving further
        notice "Sprite has reached the edge!"
        wait
    end if
    #w.g "drawsprites"
    wait

[quit]
    timer 0
    unloadbmp "landscape"
    unloadbmp "smiley"
```

```
close #w : end
```

SpriteTravelXY

Liberty BASIC gives us another way to cause a sprite to move automatically. We can issue a `SPRITETRAVELXY` command to send the sprite from its current location to a specified location.

`SPRITETRAVELXY` causes the sprite called `SpriteName` to travel to the `x, y` location specified at the speed indicated, and to fire an event when it reaches the destination. The helpfile shows us how to use hard-coded values for the desired destination, and how to use variables for those values.

```
#w.g "spritetravelxy SpriteName 200 250 5 [landed]"
or
#w.g "spritetravelxy SpriteName " ;X; " " ;Y; " " ;speed; "
[branchHandler]"
or
#w.g "spritetravelxy SpriteName " ;X; " " ;Y; " " ;speed; "
subHandler"
```

Since Liberty BASIC does the checking for us, we don't need to check the sprite's location each time we update the display. The program invokes an "allDone" branch label when the sprite arrives at its destination. When that event is triggered, we turn off the timer and give the user a notice. In a real game, we'd do other things, of course!

```
'add a sprite and locate it at x=10, y=30
'make sprite move with spritetravelxy

nomainwin
loadbmp "smiley", "sprites\smiley.bmp"
loadbmp "landscape", "sprites\bg1.bmp"
WindowHeight = 300 : WindowWidth = 400

graphicbox #w.g, 0, 0, 400, 300
open "Moving Sprite 4" for window_nf as #w
    #w "trapclose [quit]"
    #w.g "down"
    #w.g "background landscape"
    #w.g "addsprite guy smiley"

    #w.g "spritexy guy 10 30"

    'make sprite move to X = 360, Y = 110
    #w.g "spritetravelxy guy 360 110 4 [allDone]"
```

```
#w.g "drawsprites" 'update screen

timer 200, [update]

wait

[update]
    #w.g "drawsprites"
    wait

[allDone]
    timer 0 'turn off timer
    notice "Sprite has reached 360,110!"
    wait

[quit]
    timer 0
    unloadbmp "landscape"
    unloadbmp "smiley"
    close #w : end
```

SpriteXY

If we want to be in complete control of the sprite's location for each frame of animation, we can use the **SPRITEXY** command. We've already used this command in our demos. We use it to locate the sprite at X equals 10 and Y equals 30 at the start of each program.

SPRITEXY causes the sprite to be drawn at the specified position the next time the display is updated with the **DRAWSPRITES** command. The helpfile shows us how to do this with hard-coded values and with variables, but we'll want to use variables.

```
print #w.g, "spritexy SpriteName 100 137";
OR
x=100:y=137
print #w.g, "spritexy SpriteName ";x;" ";y
```

We need to use variables so that we can change the values easily. This is called "incrementing the value of the variable." Here's how we'll do it. We increment the X value by adding 3 to it each time. We increment the Y value by adding 2 to it each time. After incrementing the values, we issue a **SPRITEXY** command to change the location of the sprite.

```
'increment sprite location each time
spriteX = spriteX + 3
spriteY = spriteY + 2
```

```
'locate sprite at new position
#w.g "spritexy guy ";spriteX;" ";spriteY
#w.g "drawsprites"
```

We don't need to ask Liberty BASIC for the location of the sprite, because we always know it. We keep it in the spriteX and spriteY variables. We can simply check the values of those variables to see if the sprite has reached an edge of the graphicbox.

```
'check to see if sprite is at an edge
if (spriteX > 360) or (spriteY > 240) then
    timer 0 'stop him from moving further
    notice "Sprite has reached the edge!"
    wait
end if
```

This method of moving a sprite is the most complicated of the three methods, but it also gives us the most control over the sprite animation.

```
'add a sprite and locate it at x=10, y=30
'make sprite move with spritexy
'stop sprite when he gets to the edge
nomainwin
loadbmp "smiley", "sprites\smiley.bmp"
loadbmp "landscape", "sprites\bgl.bmp"
WindowHeight = 300 : WindowWidth = 400

graphicbox #w.g, 0, 0, 400, 300
open "Moving Sprite 5" for window_nf as #w
    #w "trapclose [quit]"
    #w.g "down"
    #w.g "background landscape"
    #w.g "addsprite guy smiley"

    spriteX = 10 : spriteY = 30
    #w.g "spritexy guy ";spriteX;" ";spriteY

    #w.g "drawsprites" 'update screen

    timer 200, [update]

wait
```



```
[update]
    'increment sprite location each time
    spriteX = spriteX + 3
    spriteY = spriteY + 2

    'locate sprite at new position
    #w.g "spritexy guy ";spriteX;" ";spriteY

    'check to see if sprite is at an edge
    if (spriteX > 360) or (spriteY > 240) then
        timer 0 'stop him from moving further
        notice "Sprite has reached the edge!"
        wait
    end if
    #w.g "drawsprites"
    wait

[quit]
    timer 0
    unloadbmp "landscape"
    unloadbmp "smiley"
    close #w : end
```

Table of Contents

[Sprite Byte Tutorials Lesson Four: Moving A Sprite](#)

[Three Easy Ways to Move a Sprite](#)

[Updating the Screen](#)

[Locate a Sprite](#)

[SpriteMoveXY](#)

[SPRITEXY?](#)

[SpriteTravelXY](#)

[SpriteXY](#)