Sprite Byte Tutorials Lesson Three: Changing the Sprite Image

by Alyce Watson http://alycesrestaurant.com/

-

Alyce

Table of Contents

Sprites with more than one image.

Did you know that your sprites can have more than one image? You can add images when you add the sprite. The **ADDSPRITE** command allows you to give the sprite a name and a list of images. The names of the images are the names you've given them when they are loaded with **LOADBMP**. The following example code loads three bitmaps, then adds a sprite that includes those three images in its image list.

```
loadbmp "image1","first.bmp"
loadbmp "image2","second.bmp"
loadbmp "image3","third.bmp"
#game "addsprite spritename image1 image2 image3"
```

Animation

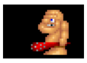
You can use this method to add frames of animation for a sprite. For instance, an animal can be running with a realistic look by including images of the animal at different parts of his stride. The **CYCLESPRITE** command causes the sprite to cycle through the list of images, either once, or over and over, giving a realistic look to the running animal.

Here is the syntax for **CYCLESPRITE**:

```
CYCLESPRITE
print #w.g, "cyclesprite SpriteName 1"
print #w.g, "cyclesprite SpriteName -1"
print #w.g, "cyclesprite SpriteName 1 once"
```

This causes a sprite to cycle through its image list automatically. Using "1" will cause the list to cycle forward. Using "-1" will cause the list to cycle backwards. Using the optional "once" parameter will cause the sprite to cycle through its image list only one time, otherwise it cycles continuously.

This article presents an animated demo of a little caveman swinging a club. He'll look like this:



To create this in Liberty BASIC, you must add the sprite for the caveman and include three variations on his image. One has the club pointing forward, another has it over his head, and another has the club pointing downward. When you issue the **ADDSPRITE** command, include the names from the **LOADBMP** commands for these three frames, then issue the **CYCLESPRITE** command. Each time the frame is updated with **DRAWSPRITES**, the displayed sprite image cycles to the next image on the list. Here is what it looks like in a program:

```
    loadbmp "club1","club1.bmp"
    loadbmp "club2","club2.bmp"
    loadbmp "club3","club3.bmp"

    #main.g "addsprite guy club1 club2 club3"
    #main.g "spritexy guy 85 50"
    #main.g "cyclesprite guy 1"
```

Another use for an image list.

There are other reasons to change the sprite's image. Perhaps an object has magical powers at some point in the game, and it begins to glow red. Perhaps an enemy looks different as his power fades. Perhaps a missile explodes into a ball of fire. All of these variations can be included in a single sprite by adding their images to the image list in the **ADDSPRITE** command.

Why not use multiple sprites?

You might ask, "Why not just use multiple sprites instead of one sprite with multiple images?" You can use different sprites, and I've done this. I've moved a missile offscreen and placed an explosion sprite in its place. The explosion sprite is later moved offscreen, and the missile is placed onscreen again when the user fires his weapon. If, instead, you use a single sprite and call up different images from its list, you don't need to write separate routines for collision detection, movement, etc. If you used multiple sprites, you'd have to switch them back and forth and write movement and collision detection routines for each of them.

This article includes an example of a caveman carrying different weapons... a rock, a club, and a spear. He looks like this:

To add the sprite, give it a name and include the list of images that were loaded with **LOADBMP**.

```
    loadbmp "rock","rock.bmp"
    loadbmp "spear","spear.bmp"
    loadbmp "club","club.bmp"

    #main.g "addsprite guy rock spear club"
    #main.g "spritexy guy 85 50"
    #main.g "drawsprites"
```

SPRITEIMAGE

Change the image displayed on the sprite with the **SPRITEIMAGE** command. The syntax for the command is:

```
SPRITEIMAGE
print #w.g, "spriteimage SpriteName BmpNameX"
```

This causes the sprite called SpriteName to be shown as the image from its image list called BmpNameX.

In the second demo, there is no **CYCLESPRITE** command. Instead, to demonstrate **SPRITEIMAGE**, there are buttons that allow the user to change the image displayed as the sprite. Here is the code for the buttons:

```
[rock]
    #main.g "spriteimage guy rock"
    #main.g "drawsprites"
    wait

[club]
    #main.g "spriteimage guy club"
    #main.g "drawsprites"
    wait

[spear]
    #main.g "spriteimage guy spear"
    #main.g "drawsprites"
    wait
```

DRAWSPRITES

Don't forget that a **DRAWSPRITES** command must be issued to update the screen for each new frame of animation.

CHALLENGE

If you'd like to learn more about using these techniques, take the sample code and develop it into a game!

Download the images for these demos here:

sprite3images.zip

- Details
- Download
- 7 KB

DEMO of CYCLESPRITE

```
[WindowSetup]
    NOMAINWIN
    WindowWidth = 238 : WindowHeight = 200
    UpperLeftX = INT((DisplayWidth-WindowWidth)/2)
    UpperLeftY = INT((DisplayHeight-WindowHeight)/2)

[ControlSetup]
graphicbox  #main.g, 1, 1, 228, 180

Open "Sprite Cycling" for Window_nf as #main
    #main "trapclose [quit]"
    #main.g "down; fill darkgreen; flush"
    #main.g "getbmp bkg 0 0 228 178"
    #main.g "background bkg"

[SpriteSetup]
    loadbmp "club1","club1.bmp"
    loadbmp "club2","club2.bmp"
    loadbmp "club3","club3.bmp"

    #main.g "addsprite guy club1 club2 club3"
    #main.g "spritexy guy 85 50"
    #main.g "cyclesprite guy 1"

[loop]
    scan
    #main.g "drawsprites"
    calldll #kernel32, "Sleep", 200 as long, re as void
    goto [loop]

[quit]
    unloadbmp "club1"
    unloadbmp "club2"
    unloadbmp "club3"
    close #main : END

'Game artwork created by Ari Feldman ari@arifeldman.com
```

DEMO of SPRITEIMAGE

```
[WindowSetup]
```

```
    NOMAINWIN
    WindowWidth = 238 : WindowHeight = 252
    UpperLeftX = INT((DisplayWidth-WindowWidth)/2)
    UpperLeftY = INT((DisplayHeight-WindowHeight)/2)

[ControlSetup]
graphicbox  #main.g, 1, 1, 228, 180
button      #main.rock, "Rock",[rock],UL, 5, 190, 70, 24
button      #main.club, "Club",[club],UL, 80, 190, 70, 24
button      #main.spear, "Spear",[spear],UL, 155, 190, 70, 24

Open "Sprite Images" for Window_nf as #main
    #main "trapclose [quit]"
    #main.g "down; fill darkgreen; flush"
    #main.g "getbmp bkg 0 0 228 178"
    #main.g "background bkg"

[SpriteSetup]
    loadbmp "rock","rock.bmp"
    loadbmp "spear","spear.bmp"
    loadbmp "club","club.bmp"

    #main.g "addsprite guy rock spear club"
    #main.g "spritexy guy 85 50"
    #main.g "drawsprites"

[loop]
    Wait

[quit]
    unloadbmp "rock"
    unloadbmp "spear"
    unloadbmp "club"
    close #main : END

[rock]
    #main.g "spriteimage guy rock"
    #main.g "drawsprites"
    wait

[club]
    #main.g "spriteimage guy club"
    #main.g "drawsprites"
    wait

[spear]
```

```
    #main.g "spriteimage guy spear"
    #main.g "drawsprites"
    wait
```

Table of Contents