

## **USING THE STOS SPRITE, MOVE AND ANIM COMMANDS**

Sprites are the little graphical images that move about on-screen such as ships, aliens, bullets, little men, etc. Let's look at how we can do this.

### **THE SPRITE COMMAND**

Before you can use the sprite command you will need some sprites in memory. For example, try loading the animals1.mbk sprite bank from the STOS disk.

EG: load"animals1.mbk"

As it's a sprite bank, STOS automatically loads them into bank one. Now let's put it on screen.

```
10 sprite 1,100,100,14
```

This put a monkey sprite on the screen. Now, let's take a closer look at the sprite command.

```
sprite NUMBER,X,Y,IMAGE
```

NUMBER is the number of the sprite which ranges from 1 to 15. You can only place 15 sprites on the screen at any one time.

X AND Y are the X and Y coordinates where you want to put the sprite. So in the above example, the sprite is placed 100 pixels across the screen and 100 pixels down the screen.

IMAGE is the number of the sprite from the sprite bank you want to place on the screen. In the above example its image 14 which is the monkey.

Let's try two sprites on the screen.

```
10 sprite 1,100,100,14
```

```
20 sprite 2,160,100,14
```

So we now have two copies of the monkey on screen. Try changing the image number and the second sprite will be another one from the bank. Sprites can be designed using the sprite editor accessory. Refer to the STOS manual on how to create sprites.

### **MOVING A SPRITE:**

The sprite can be moved across the screen by using the MOVE command. It goes like this.

move X 1,(1,2,300) or move Y 1,(1,2,300)

Let's look at the command in more detail.

move DIRECTION,NUMBER,(SPEED,STEPS,PIXELS)

DIRECTION can be X or Y, it tells STOS to move the sprite either X (across the screen) or Y (up or down the screen).

NUMBER is the number of the sprite to move.

SPEED is the speed to move the sprite which ranges from 1 (fast) to 4 (slow).

STEPS tells STOS how many pixels to move the sprite at a time.

PIXELS tells STOS how many pixels to move the sprite.

Look at this example

```
10 sprite 1,10,10,14
```

```
20 move x 1,(1,2,100)
```

```
30 move on
```

So the sprite is moving across the screen at a speed of 1, in steps of 2 pixels at a time, and 100 pixels across the screen. The move-on command tells STOS to start the movement. Note that sprites work on interrupt so your program can be doing over things while the sprite is working.

We can add extra letters to the move command to make it do other things like L and E. Normally the sprite will move and stop at the PIXELS parameter you choose but you can fix STOS to keep moving the sprite along its set strings by adding the letter L at the end of the move command. Using the E command allows you to stop the movement at a certain point.

EG:

```
10 sprite 1,10,10,14
```

```
20 move x 1,"(1,2,100)(1,-2,100)L" :rem keep moving sprite
```

```
30 sprite 2,10,50,14
```

```
40 move Y 1,"(1,2,100)E50?:rem stop sprite at Y co-ordinate 50
```

50 move on

## **ANIMATING THE SPRITE**

We can step through the images of a sprite by using the ANIM command. The format goes like this...

```
anim NUMBER,(IMAGE1,DELAY),(IMAGE2,DELAY) etc
```

NUMBER is the number of the sprite to animate.

IMAGE is the image number to show.

DELAY is the time to pause between each image.

Let's animate the monkey.

```
10 sprite 1,100,100,14
```

```
20 anim 1,"(14,5)(15,5)(16,5)(17,5)L" : anim on
```

So this command flicks the sprite between images 14 to 17 and then loops back to the start. We now have a walking monkey.

## **OTHER COMMANDS**

```
MOVON(NUMBER)
```

This command checks to see if the sprite is still moving. It returns 0 if it isn't and 1 if it is.

```
UPDATE, UPDATE ON, UPDATE OFF
```

STOS updates sprites every 50th of a second. This can be awkward when you want to do other things and STOS is using processor time for its updating. STOS is always checking a sprite and updating its position and you can turn the feature off by using the Update Off command, and Update On can return things to normal. You can then update the sprite when you want freeing some processor time like so...

```
10 update off
```

```
20 for X=10 to 100
```

```
30 sprite 1,X,10,1 : update
```

```
40 next X
```

Try removing the update command from this example and the sprite will not be displayed on the screen. It's still there but just not being copied to the screen.