

Simple software sprite to sprite collision

You saw the \$D01E routines and know that with comparison with software collision the \$D01E method is not as handy unless you do something like a simple dodge type of game. Now let us say you were making proper shoot 'em ups, etc, then a software collision would be much better to use.

Depending on the area size of your sprites, this nifty piece of code can calculate the collision areas for the sprites. To be able to use this, you would need to create some labels and values.

Something like:

```
COLLISIONX1 = $02 (or wherever you want it)
COLLISIONX2 = $03
COLLISIONY1 = $04
COLLISIONY2 = $05

XSIZE1 = $06 ;The area of the drawn sprite on the left
XSIZE2 = $0C ;The area of the drawn sprite on the right
YSIZE1 = $0C ;The area of the drawn sprite at the top
YSIZE2 = $18 ;The area of the drawn sprite at the bottom
```

Now we write a routine to calculate the collision routine

```
LDA SPRITEX
SEC
SBC #XSIZE1
STA COLLISIONX1
CLC
ADC #XSIZE2
STA COLLISIONX2
LDA SPRITEY
SEC
SBC #YSIZE1
STA COLLISIONY1
CLC
ADC #YSIZE2
STA COLLISIONY2
```

Now for the main collision detection

```
LDA ENEMYSPRITEX
CMP COLLISIONX1
BCC NOTHIT
CMP COLLISIONX2
BCS NOTHIT
LDA ENEMYSPRITEY
CMP COLLISIONY1
BCC NOTHIT
CMP COLLISIONY2
BCS NOTHIT
```

Last update: 2015-04-17 04:33 base:simple_software_sprite_to_sprite_collision https://codebase64.org/doku.php?id=base:simple_software_sprite_to_sprite_collision

```
HIT      INC $D020
          JMP HIT
NOTHIT   RTS
```

Yes, once again this source uses the flashy border loop if a collision is made. This can easily be changed if you want to change the routine in your game.

From: <https://codebase64.org/> - Codebase 64 wiki

Permanent link: https://codebase64.org/doku.php?id=base:simple_software_sprite_to_sprite_collision

Last update: **2015-04-17 04:33**

