This code checks which sprite triggered the hardware sprite collision detection when bit 1 of $D01E is turned on. Assuming sprite 0 is the player sprite and sprites 1…7 are the “enemies”. A collision is considered TRUE if a sprite is within +/-20px on the Y axis and +/-23px on X axis from the player sprite. SPR_COLL_DETECT returns in .X the sprite number which is in that area. If multiple collisions are in “TRUE” condition, the higher sprite value is returned first. If you need to check multiple collisions, you should continue the SPR_COLL_DETECT code calling SPR_COLL_LOOP without altering .X register.

```
|----->---->----->---------->24 pixels
|     |    |      |___________
##################            |
##################            |
##################            |
##################            |
######xxxxxx######____________|-> 21 pixels
######x****x######            |
######x****x######            |
######x****x######            |
######x****x######____________|
######xxxxxx######            |-> 21 pixels
##################            |
##################            |
##################            |
##################____________|
|     |     |     |
|----->----->----->---------->24 pixels
# = Detect area around player sprite
x = Overlayed player/enemies pixels for checking purposes
* = Player sprite central pixels
;---------------------------------------
```

```
;---------------------------------------
; Collisions detection
; Hybrid hardware/software collision
; detect code.
; By Flavioweb 2018.
;-------------------
CHECK_PLR_COLLISION
    LDA $D01E
    AND #%00000001                  ; Some HW detected collision with
player sprite?
    BEQ CHECK_PLR_COLLISION_EXIT    ; No -> Exit.
    JSR SPR_COLL_DETECT             ; Check which sprite collided with
player.
    BNE CHECK_PLR_SPRITE_01         ; If .X==$00 no collision detected
(which is almost impossible...)
CHECK_PLR_COLLISION_EXIT
    RTS
CHECK_PLR_SPRITE_01
    CPX #$01                        ; Check if sprite X is in "collision
area"
```

```
    BNE CHECK_PLR_SPRITE_02             ; No -> check next sprite
;   Do something here...
    RTS
CHECK_PLR_SPRITE_02
    CPX #$02
    BNE CHECK_PLR_SPRITE_03
;   Do something here...
    RTS
CHECK_PLR_SPRITE_03
    CPX #$03
    BNE CHECK_PLR_SPRITE_04
;   Do something here...
    RTS
CHECK_PLR_SPRITE_04
    CPX #$04
    BNE CHECK_PLR_SPRITE_05
;   Do something here...
    RTS
CHECK_PLR_SPRITE_05
    CPX #$05
    BNE CHECK_PLR_SPRITE_06
;   Do something here...
    RTS
CHECK_PLR_SPRITE_01
    CPX #$06
    BNE CHECK_PLR_SPRITE_07
;   Do something here...
    RTS
CHECK_PLR_SPRITE_07
    CPX #$07
    BNE CHECK_PLR_SPRITE_EXIT
;   Do something here...
CHECK_PLR_SPRITE_EXIT
    RTS
;---------------------------------------
; Sprite collided detect
; If .X<>$00 = Sprite collided with player
; Plr = spr $00
; From $01 enemies sprites.
;-------------------
SPR_COLL_DETECT
    LDX #$07
SPR_COLL_LOOP
    LDA SPRITEY,X                       ; Load Enemy Y position
    SEC
    SBC SPRITEY                         ; Subtract Player Y position
    BPL CHECK_Y_NO_MINUS
    EOR #$FF                            ; Invert result sign
CHECK_Y_NO_MINUS
    CMP #$15                            ; Check for enemy sprite distance Y
```

```
    BCS CHECK_PLR_NO_COLL
    LDA SPRITEX,X                       ; Load Enemy X position
    SEC
    SBC SPRITEX                         ; Subtract Player X position
    BPL CHECK_NO_MINUS
    EOR #$FF                            ; Invert result sign
CHECK_NO_MINUS
    CMP #$17                            ; Check for enemy sprite distance X
    BCS CHECK_PLR_NO_COLL
    RTS
CHECK_PLR_NO_COLL
    DEX                                 ; Goes to next sprite/enemy
    BNE SPR_COLL_LOOP
    RTS
;--------------------------------------
; Sprites data tables examples
;--------------------
SPRITEY
    .BYTE $64, $80, $A0, $D0, $50, $70, $90, $A0
SPRITEX
    .BYTE $A0, $80, $A0, $D0, $50, $70, $90, $A0
SPRITEMSB
    .BYTE $00, $00, $00, $00, $00, $00, $00, $00
;--------------------------------------------------------------------------------
----
```

From:
https://codebase64.org/ - **Codebase 64 wiki**

Permanent link:
https://codebase64.org/doku.php?id=base:hybrid_hardware_software_sprite_collision_detection&rev=1517084864

Last update: **2018-01-27 21:27**