

nesdoug

## 4. What's a V-blank?

The PPU is a graphics processor that either sends a signal to the TV, or gets new information from the CPU (through a very slow 1 byte at a time bus). But, it can't do both at the same time. So, the only time you can send it new information is when it's 'off', or during v-blank.

When is that? Well, the PPU spends 90% of its time sending pixels to the TV, one line at a time. It goes from top left, to top right, drops down 1, left to right, drop down 1, left to right, etc, until it reaches the bottom, then it pauses, jumps back up to the top left and starts all over again.

It does this 60 times a second. The slight pause at the bottom is the vertical blank period. When the screen is 'on', it is the only time you can send new information (without drawing weird glitchy streaks across the screen).

V-blank period is annoyingly short. So short, you really only have time to write 2-4 columns worth of 8x8 tiles, and update the sprites. For a scrolling game, you will need to update BG tiles as you go, but, only during V-blank (so you don't have the screen flashing black every few seconds). So, it's very important to time it.

There are 2 ways to know when you're in V-blank...the PPU sets a flag that you can check (the high bit of \$2002). Or, you can turn on NMI interrupts. When NMI is on, the program will stop what it's doing when it gets a signal that V-blank has started... And it will jump to the NMI code. You can use this to time writes to the PPU, and time your music code, and time movements of sprites. We know that it will jump here every 1/60th of a second, so the whole game will use that as a reference for how much time has passed between events.

We are going to write a new program that writes "HELLO WORLD" while the screen is on, 1 letter at a time. It will wait 30 frames, then during v-blank write 1 letter. Repeat until all letters are drawn. Then it will clear them and start over.

Inside the reset.s module, I inserted (basically) ++NMI\_flag and ++Frame\_Count, in the NMI handler. So, at the beginning of each V-blank, they will both go up one. An NMI\_flag > 0 will then trigger the next action [inside the main() function]. I'm waiting 30 frames (0.5 seconds) to trigger the next letter to be loaded. Here are the changes...

```
void Load_Text(void) {
    if (Text_Position < sizeof(TEXT)){
        PPU_ADDRESS = 0x21;    // set an address in the PPU of 0x21ca
        PPU_ADDRESS = 0xca + Text_Position; //about the middle of the screen
        PPU_DATA = TEXT[Text_Position];
        ++Text_Position;
    }
    else {
        Text_Position = 0;
        PPU_ADDRESS = 0x21;
        PPU_ADDRESS = 0xca;
        for ( index = 0; index < sizeof(TEXT); ++index ){
            PPU_DATA = 0; // clear the text by putting tile #0 in its place
        }
    }
}
```

```
void main (void) {
    All_Off(); // turn off screen
    Load_Palette();
    Reset_Scroll();
    All_On(); // turn on screen

    while (1){ // infinite loop
        while (NMI_flag == 0); // wait till NMI
        NMI_flag = 0;
        if (Frame_Count == 30){ // wait 30 frames = 0.5 seconds
            Load_Text();
            Reset_Scroll();
            Frame_Count = 0;
        }
    }
}
```

```
// inside the startup code, the NMI routine will
// ++NMIflag and ++FrameCount at each V-blank
```



Here's the source code...

<http://dl.dropboxusercontent.com/s/c3fbfranz5gcafz/lesson2.zip>  
(<http://dl.dropboxusercontent.com/s/c3fbfranz5gcafz/lesson2.zip>)

(again, must go in a folder in the cc65 directory to compile)

(on a side note...ideally you would wait till you're in the V-blank period before you turn rendering back on [All\_On()], or you'll get 1 frame where the screen is misaligned, which will look a little odd. It's not noticeable here, since the only time the screen is turned on outside V-blank, the entire screen is black, so a misalignment isn't noticeable.)

[November 18, 2015](#)[April 15, 2017](#) [dougfraker](#)

## 5 thoughts on “4. What's a V-blank?”

1.

**Iury Bessa** says:

[October 4, 2016 at 3:46 am](#) [Edit](#)

On line 51 you forgot to add the ++ to Frame\_Counter. There is a reset to 0 if Frame\_Counter were to get to 30, but no way to increment Frame\_Counter in the source code.

[Reply](#)

**dougfraker** says:

[October 4, 2016 at 2:51 pm](#) [Edit](#)

Frame counting must be done in the NMI code...which it is...in reset.s

With NMIs turned on, it will automatically go to the NMI code 60 times a second.

2.

**chris** says:

[October 13, 2016 at 1:07 pm](#) [Edit](#)

[Reply](#)

I get the error  
ld65.exe: Error: nes.cfg(71): Attribute expected  
when compiling. What is wrong ?

1.

[Reply](#)

**BLumia** says:

[November 20, 2016 at 2:54 pm Edit](#)

Maybe u missed "ONCE: load = PRG, type = ro, optional = yes;" in your .cfg file ?

3.

[Reply](#)

**plin** says:

[January 16, 2017 at 1:19 am Edit](#)

Awesome tutorials!

[Reply](#)

[Create a free website or blog at WordPress.com.](#)