

nesdoug

## 9. Drawing a full background

Ok, we learned earlier that you can only draw to the screen... 1. During V-blank or 2. While the screen is off. With option 1, you can only draw 2 rows per frame, and with 30 rows to draw it will take 15-16 frames to do. With option 2 you get 1-2 frames of black screen (or whatever the default BG color is) and the new background is fully shown.

I prefer the second option, as the code is easier. I've used NES Screen Tool to create the backgrounds quickly, and used the Save Nametable as .RLE to compress it. I will then be using the UNRLE asm code to decompress it. (another thing that you don't need to know how it works, it works, forget about it and move on).

I've set it to load a new background every time Start is pressed. I have it also checking against previous frame presses, so that we don't get multiples in a row.

This code...

1. Gets input every frame
2. If Current press = Start, and last frame press != Start...
3. Set flag to turn off screen
4. Next V-blank, turn off screen
5. Draw a new screen
6. Wait till V-blank, turn on screen

Draw the new background...

```
void Draw_Background(void) {
    All_Off();
    PPU_ADDRESS = 0x20; //address $2000 = start of nametable #0
    PPU_ADDRESS = 0x00;
    UnRLE(All_Backgrounds[which_BGD]); //uncompresses our data
    Wait_Vblank(); //don't turn on screen until in v-blank
    All_On();
    ++which_BGD;
    if (which_BGD == 4) //shuffles between 0-3
        which_BGD = 0;
}
```

```
const unsigned char * const All_Backgrounds[]={n1,n2,n3,n4};
// pointers to the addresses of each background
```



Here's the link to the source code. Now 'Start' will change the background...

<http://dl.dropboxusercontent.com/s/e7mktrwqblyb6zr/lesson7.zip>  
 (<http://dl.dropboxusercontent.com/s/e7mktrwqblyb6zr/lesson7.zip>).

November 28, 2015 April 15, 2017 dougfraker

## 7 thoughts on “9. Drawing a full background”

1. **Boring Danger (@fritzvd)** says:  
April 29, 2016 at 8:30 am Edit  
 Hi Doug,

I'm really enjoying these tutorials. I'm going through them a bit. And I actually WAS wondering about RLE.

Do you know what type of RLE is used to encode / decode the background sheets? I'm trying to reproduce it, but come up empty. Any pointers? Or perhaps an explanation of what the UNRLE function does? I'm curious to find out.

Reply

**dougfraker** says:  
May 3, 2016 at 2:21 am Edit  
 See next page for a reply to a similar question.

2. **Boring Danger (@fritzvd)** says:  
May 12, 2016 at 7:32 am Edit

Yes. Sorry. I thought my comment didn't come through. But it was probably just on hold in your comment approve/disapprove section or so.

Reply

3.

[Reply](#)**Adrien** says:[May 26, 2017 at 6:00 pm Edit](#)

Hi doug,

Have you got a tool to edit the positions of the sprites background's in the .h files?

I don't understand all the data in the .h files. Is there a pattern in the array for each range of sprites?

Thank you for this great tuto 😊

4.

[Reply](#)**Sebastian** says:[June 15, 2017 at 3:55 pm Edit](#)

Hi Doug,

I created a background, using tiles of the first 4k of my .chr-file. Unfortunately nothing will be displayed. When I draw a background with tiles from the second 4k, everything is drawn perfectly.

Could you please help me out?

[Reply](#)**dougfraker** says:[June 16, 2017 at 2:38 pm Edit](#)

I don't think I explained very well, but the \$2000 register can change which tiles to use for BG and sprites. xxxB Sxxx. 0 = first set of tiles, 1 = second set. That may be the problem.

[Reply](#)5. Pingback: [BSides 2016 NES Game | Hypn.za.net Edit](#)

[Blog at WordPress.com.](#)