
The Skinny On NES Scrolling

Documento original: The Skinny On NES Scrolling

Autor: Loopy

Traduzido por: <Hyde> (líder do projeto *Translatore*).

E-Mail: Hyde@programmer.net

Página WEB: <http://hydesprojects.cjb.net/>

Versão: 1.0

-Primeira versão desta tradução.

Parte I

A informação disponível sobre o mecanismo de rolagem de cenário (ou tela de fundo) é suficiente para a maioria dos jogos. Porém, há outros que usam este mesmo mecanismo para produzir uma variedade de efeitos especiais que, até agora, jamais haviam sido explicados. Para que possamos entender como alguns destes efeitos são produzidos, devemos explicar como o mecanismo de rolagem funciona na sua totalidade.

Os seguintes registradores estão relacionados com rolagens:

(V) Endereço da VRAM (16 bits), o qual é determinado modificado através do registrador \$2006.

(T) Endereço temporário da VRAM (16 bits). Na verdade este endereço é formado por 15 bits, dado que o último bit não é usado.

(X) Índice de deslocamento horizontal (X) dentro de uma telha.

A PPU usa a o endereço da VRAM para (1) ler e guardar valores na VRAM através do registrador \$2007, e (2) para retirar dados das tabelas de referências necessários para desenhar o cenário. A PPU atualiza o endereço da VRAM à medida que o cenário é desenhado, de tal maneira que este sempre aponta para dados existentes em alguma tabela de referência. Os bits 0-11 indicam o endereço (menos \$2000) da tabela de referência em uso e os bits 12-14 representam o índice de deslocamento vertical (Y) dentro de uma telha (Obs: O número deste deslocamento é relativo a primeira linha da telha em uso. Por exemplo, se a PPU tiver que desenhar dados de uma telha e o valor dentro destes últimos bits for sete, a PPU desenhará a ultima linha desta telha).

Ações que afetam o conteúdo de alguns registradores:

(Aqui será usada notação lógica, já que esta simplifica todas as explicações).

(D) Byte escrito/lido.

\$2000 (Escrevendo):

(T):0000110000000000 = (D):00000011

\$2005 (Escrevendo por primeira vez):

(T):0000000000011111 = (D):11111000

(X) = (D):00000111

\$2005 (Escrevendo por segunda vez):
 (T):0000001111100000 = (D):11111000
 (T):0111000000000000 = (D):00000111
\$2006 (Escrevendo por primeira vez):
 (T):0011111100000000 = (D):00111111
 (T):1100000000000000 = 0
\$2006 (Escrevendo por segunda vez):
 (T):0000000011111111 = (D):11111111
 (V) = (T)
Começo de uma linha de escaneamento (supondo que o cenário e os sprites estão ativados):
 (V):0000010000011111 = (T):0000010000011111
Começo de um quadro (linha 0) (supondo que o cenário e os sprites estão ativados):
 (V) = (T)

Parte II

Você pode interpretar os bits 0-4 do endereço da VRAM como o valor de rolagem horizontal (X) multiplicado por 8, o qual é incrementado pela PPU durante o processo de desenho. Quando estes bits mudam de 11111 (31 em decimal) para 00000 (esta mudança se deve ao fato que o endereço da VRAM está sempre sendo incrementado), o bit 10 do endereço é permutado, causando uma modificação no número da tabela de referência sendo utilizada.

Você pode interpretar os bits 5-9 do endereço da VRAM como o valor de rolagem vertical (Y) multiplicado por 8. A atualização destes bits ocorre diferentemente: quando estes mudam para 00000, o bit 11 só é permutado se o valor representado por eles (prévio à mudança) for incrementado desde 29 (em vez de 31). Se você coloca (manualmente) um valor maior que 29 nestes bits (através dos registradores \$2005 ou \$2006), dados de tabela de atributo serão usados como dados de tabelas de referência e o valor de rolagem vertical (Y) mudará de 31 a 0 sem permutar o bit 11. Isto explica o porquê de quando um valor maior que 240 é escrito à \$2005, este mesmo número aparece como uma quantidade de rolagem negativa.

Observações:

- \$2005 y \$2006 compartilhem um "switch", ou um interruptor, que determina a quantidade de vezes que algum valor foi escrito nestes registradores. Lendo o registrador \$2002 fará com que este interruptor seja desligado.

Envie-me suas sugestões!