

WedNESDay

Make Classic Games

&

Emiliollbb

wedNESday

- Qué son los wedNESday
 - NES
 - Herramientas de desarrollo (ASM, C, Gráficos, Trackers).
 - Creación de juegos para NES.
 - Estudio de los componentes de la NES.
 - También probaremos algunos de los juegos
- Todos los miércoles de 17:30 a 18:30.

¿Qué haremos en esta sesión?

- ¿Qué es la NES? ¿Por qué?
- ¿Cómo? ¿ASM?
- Ejemplo de juego desarrollado en NES
 - Inicialización de la consola
 - Interrupciones
 - Fondos
 - Sprites

¿POR QUÉ LA NES? ¿QUÉ TIENE DE ESPECIAL?

**Más de 40 años después de su lanzamiento,
“La Nintendo” sigue siendo divertida de
jugar y programar.**

¿POR QUÉ LA NES? ¿QUÉ TIENE DE ESPECIAL?

- Diseñada para ser un juguete muy barato de fabricar, permitió a los niños de los 80 y 90 tener su propia recreativa en casa sin tener que echar monedas
- Suficientemente potente para ejecutar juegos interesantes, y a la vez suficientemente sencilla para ser fácil de programar
- Su diseño impone una estética de imágenes y sonidos únicos que han creado escuela

Masayuki Uemura (上村雅之, Uemura Masayuki; 20 June 1943 – 6 December 2021)



“Un día de 1971, los jefes de Masayuki le enviaron a preguntar a Nintendo si querían comprar semiconductores. Se citó con Gunpei Yokoi y su equipo, y estos pensaron que estas células podrían ser usadas para hacer nuevos juguetes.[5]

Tras esto, a Masayuki Uemura le comunicaron que le iban a trasladar a otra ciudad, algo a lo que Masayuki se negó, por lo que dejó su trabajo en Sharp y rápidamente fue contratado por Gunpei Yokoi y su equipo de Nintendo.

Trabajando para Nintendo, Masayuki se sentía como cuando era un niño (por eso de hacer juguetes), y se sintió sorprendido que todos esos hombres tan serios hiciesen juguetes” – Extracto de Wikipedia

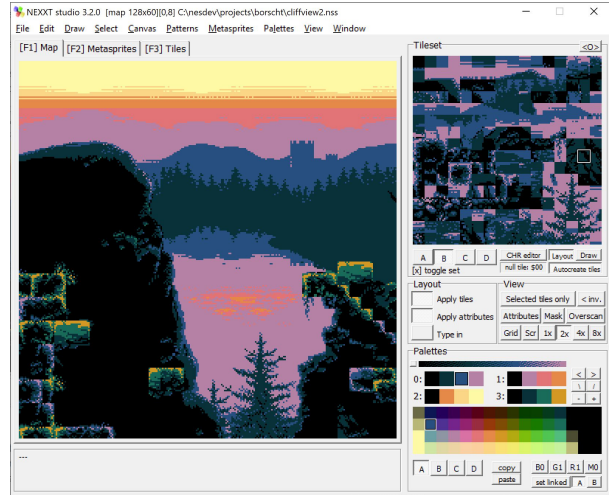
NES (Nintendo Entertainment System)

- CPU Ricoh 2A03 (MOS 65C02) a 1,79Mhz (NTSC)/1,66Mhz (PAL).
- PPU (Picture Processor Unit) 256x240 de resolución y 25 colores simultáneos (paleta de 40 colores) y 64 sprites por pantalla.
- 2KB de RAM y 2KB de VRAM (+256 bytes para oam).
- 4 generadores de tono.
- 2 Controladores, puerto de expansión y slot de cartuchos.
- Cartuchos (ROM) de 60/72 pines.



¿Que necesitamos?

- Un editor de Texto
- Un “Ensamblador” / Compilador.
- Editor Gráfico.
- Tracker.



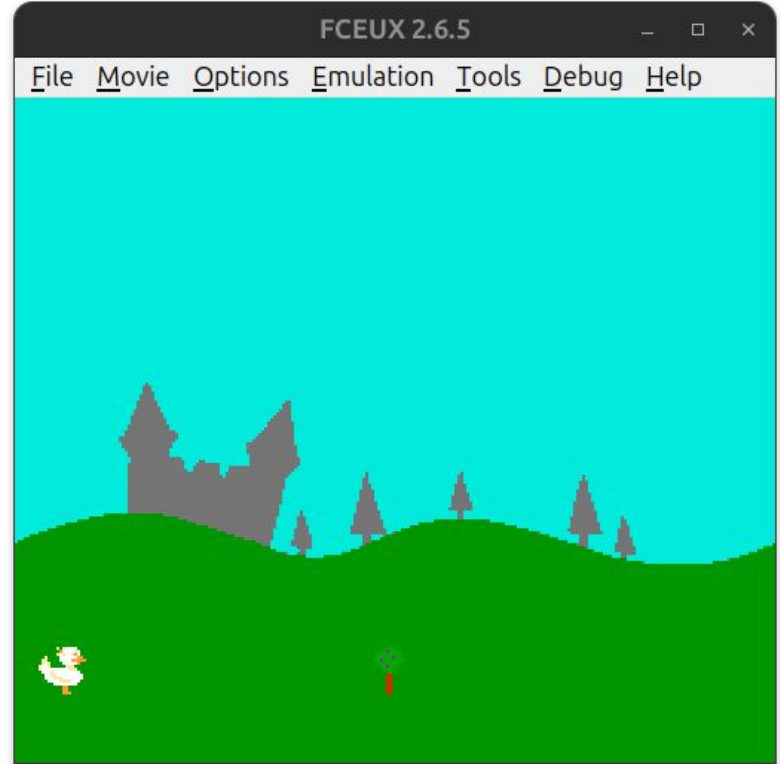
CC65
- the 6502 C compiler

Furnace

Análisis de un Juego de NES (MS Patman Redux)

Analizaremos el juego MS Patman Redux para entender cómo se debe desarrollar un juego.

Veremos tanto las herramientas utilizadas como se ha organizado el código del juego.



Arquitectura NES

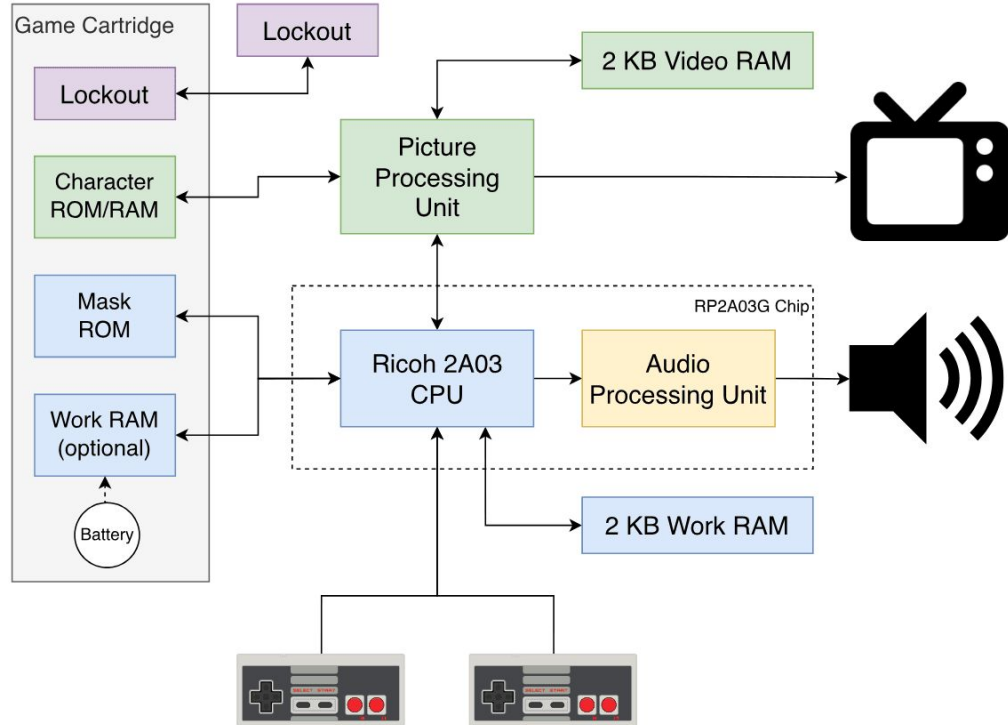


Tabla de colores

\$00	\$01	\$02	\$03	\$04	\$05	\$06	\$07	\$08	\$09	\$0A	\$0B	\$0C	\$0F
\$10	\$11	\$12	\$13	\$14	\$15	\$16	\$17	\$18	\$19	\$1A	\$1B	\$1C	\$0F
\$20	\$21	\$22	\$23	\$24	\$25	\$26	\$27	\$28	\$29	\$2A	\$2B	\$2C	\$2D
\$30	\$31	\$32	\$33	\$34	\$35	\$36	\$37	\$38	\$39	\$3A	\$3B	\$3C	\$3D

La NES sólo puede mostrar 55 colores, programados internamente en la PPU con los códigos mostrados en la tabla.

Capacidades gráficas

La PPU de la NES puede dibujar:

- Fondos
 - 2 pantallas de 256x240 pixels en disposición horizontal ó vertical
 - Scroll x,y a nivel de pixel
- Sprites
 - 64 sprites
 - Máximo 8 sprites por scanline

Tiles

- Todos los gráficos de la NES se basan en tiles de 8x8 pixels
- En el cartucho hay 2 bancos de 256 tiles cada uno
 - Lo habitual es usar un banco para fondos y otro para sprites
 - Alternativamente se puede usar el mismo banco para fondos y sprites
- Cada fondo está compuesto por 32 tiles de ancho y 30 tiles de alto
 - Existen 4 fondos, 2 de ellos mirror
- Cada sprite está compuesto por 1 tile
 - Alternativamente se puede configurar la PPU para usar sprites dobles (1 tile de ancho y 2 de alto)

Paletas

- La NES utiliza 8 paletas de 4 colores cada una
 - 4 paletas para fondos
 - Cada cuadrado de 2x2 tiles usa la misma paleta
 - 4 paletas para sprites
 - Cada sprite puede usar una de las 4 paletas disponibles
 - El primer color de cada una de las 8 paletas debe ser el mismo, y es considerado color de fondo / transparencia
 - Las paletas se encuentran en la dirección \$3F00 de la RAM de la PPU

Un poco de informática básica

- Una memoria la podemos imaginar como una tabla bidimensional (como una estantería) en la que guardar datos
 - El espacio de memoria del 6502 nos podemos imaginar una estantería, con 256 baldas, numeradas de \$00 a \$FF, dónde en cada balda tiene hueco para 256 números, también etiquetados de \$00 a \$FF.
 - No todas las baldas son memoria RAM, algunas baldas nos permiten interactuar con la PPU, o leer los mandos.

Un poco de ensamblador 6502

- Un registro es “una cajita” dentro de la cpu en la que podemos guardar un número de 8 bits (%00000000 - %11111111), (\$00, \$FF) (0,255)
- La CPU 6502 tiene un registro principal llamado A (Acumulador)
- Para cargar un valor, lo haremos con:
 - LDA #16 ; En decimal
 - LDA #\$F0 ; También lo podemos hacer en Hexadecimal, que será lo habitual
 - LDA #%00010000 ; Algunas veces será conveniente hacerlo en binario
- La CPU sólo entiende binario, en todos los casos anteriores, el ensamblador generará el mismo código binario para la CPU, el poder usar distintas notaciones es una comodidad que nos ofrece el ensamblador.

Un poquito más. Es sencillo, lo prometo.

- Para escribir el valor que tenemos guardado en “la cajita A”, usaremos:
 - STA \$2002
- No es necesario aprender de memoria las posiciones de memoria. Podemos usar constantes:
 - STA PPUADDR

Ejemplo de Ensamblador

Veamos el siguiente ejemplo:

```
LDA #2  
STA $20  
LDX $20
```

Probemoslo en un simulador:

<https://tony-cruise.github.io/6502Simulator.html>

Primer ejemplo: Cargar colores en las paletas

```
1 LDA PPUSTATUS
2 LDA #$3F
3 STA PPUADDR
4 LDA #$00
5 STA PPUADDR
6 ; Enviar los codigos de color a la PPU
7 LDA #$01 ; Azul
8 STA PPUDATA
9 LDA #$14 ; Violeta
10 STA PPUDATA
11 LDA #$16 ; Rojo
12 STA PPUDATA
13 LDA #$29 ; Verde
14 STA PPUDATA
```

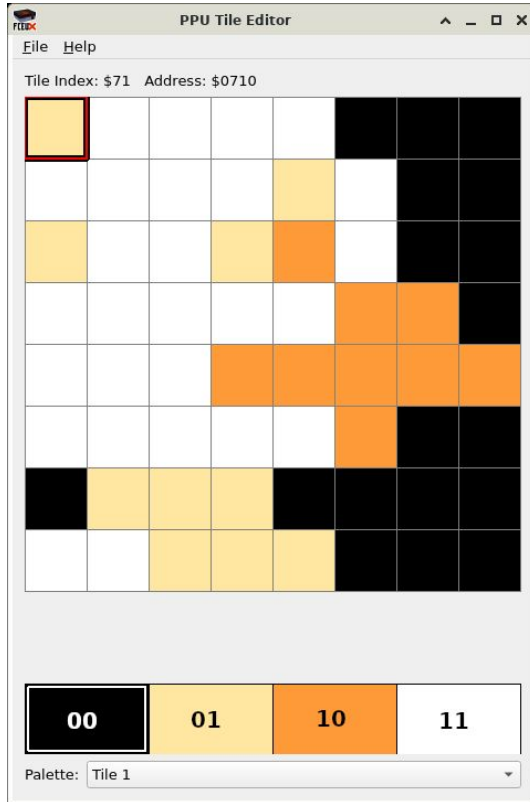
Para enviar datos a la PPU, primero indicamos a qué dirección de memoria queremos mandarlos, y luego los vamos mandando byte a byte. En este ejemplo enviamos los 4 colores de la primera paleta de fondos.

Primer ejemplo: Cargar las paletas

Veamos un ejemplo para cargar las paletas en la NES:

https://github.com/makeclassicgames/wedNESdaysExamples/blob/main/01_paletas.s

Creando Tiles



Los tiles los podemos crear con cualquier herramienta que nos sea cómoda, siendo la más habitual NEXXT Studio.

<https://frankengraphics.itch.io/nexxt>

Referencias

- <https://famicom.party/book/>
- <https://www.copetti.org/writings/consoles/nes/>
- <https://nerdy-nights.nes.science/>
- https://www.nesdev.org/wiki/Nesdev_Wiki
- <https://kh-labs.org/concepts/xa65/>
- <https://cc65.github.io/>