

**TFG**

---

**APLICACIÓN Y EMULACIÓN DE ARTE Y  
GRÁFICOS EN DISPOSITIVOS  
ELECTRÓNICOS DE LOS AÑOS 1975-1985**

**Presentado por Carlos Chust Martínez  
Tutor: Francisco Javier Sanmartín Piquer**

**Facultat de Belles Arts de Sant Carles  
Grado en Bellas Artes  
Curso 2021-2022**



**UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA**



**UNIVERSITAT POLITÈCNICA DE VALÈNCIA  
FACULTAT DE BELLES ARTS DE SANT CARLES**

# Resumen

Este trabajo trata de la explicación de la creación de gráficos en dispositivos electrónicos a finales de los años 70 y principios de los años 80, la aplicación de dichos gráficos y/o la creación de directrices teniendo en cuenta las limitaciones técnicas de estos con el propósito de crear gráficos fieles a los productos mencionados. En este trabajo, se detallarán las limitaciones de cada dispositivo, las herramientas y técnicas usadas, la aplicación de fundamentos de arte y animación de los gráficos, los obstáculos que aparecen en este ámbito, la solución a estos problemas, la explicación del valor artístico y el uso aplicado de los conocimientos y prácticas de este proyecto. Los dispositivos mencionados son: Apple II, Atari 2600/VCS, Commodore 64 y NES/Famicom.

# Abstract

This work is about the explanation of the creation of graphics in electronic devices during the late 70s and early 80s, the application of said graphics and/or the creation of guidelines taking into account the technical limitations of these with the purpose of creating graphics loyal to the mentioned products. In this work shall be detailed the limitations of each device, the tools and techniques used, the application of art and animation fundamentals on the graphics, the obstacles that appear in this field, the solution to these problems, the explanation of the artistic value and the applied use of the knowledge and exercise of this project. The mentioned devices are: Apple II, Atari 2600/VCS, Commodore 64 and NES/Famicom.

# Palabras clave

Palabras clave: emulación, píxel, sprite, pixel art, chiptune, Apple, Atari, 2600, VCS, Video Computer System, Commodore, 64, NES, Nintendo Entertainment System, Famicom, Famicom Disk System, Family Computer Disk System, Famitracker.

Keywords: emulation, pixel, sprite, pixel art, chiptune, Apple, Atari, 2600, VCS, Video Computer System Commodore, 64, NES, Nintendo Entertainment System, Famicom, Famicom Disk System, Family Computer Disk System, Famitracker.

# Índice

1. Introducción -----	p.5
2. Objetivos -----	p.10
3. metodologías -----	p.11
4. Desarrollo y resultados del trabajo -----	p.12
4.1. Apple II (1977) -----	p.12
4.2. Atari 2600 (1977) -----	p.23
4.3. Commodore 64 (1982) -----	p.26
4.4. Famicom/NES (1983) -----	p.41
4.5. Famitracker -----	p.51
5. Conclusión -----	p.58
6. Bibliografía -----	p.62
7. Índice de figuras -----	p.64
8. Anexo -----	p.68

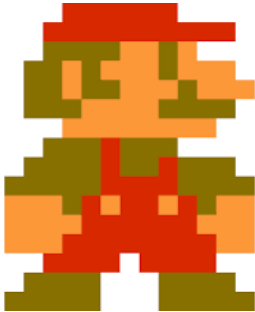


Fig. 1.

# 1. Introducción

Un píxel es la unidad más pequeña en color que una pantalla puede mostrar y puede ser modificada. Su etimología proviene de la combinación de dos palabras. La primera palabra “pix”, es una abreviación de de “pics”, y pics es una abreviación de “pictures”; y “el” es una abreviación de la palabra “element”.

En una pantalla, se usa la disposición de estos bloques en un criterio de píxel a píxel para crear imágenes y animaciones.

En el caso del uso de píxeles en un lienzo reducido y con la visualización definida de dichos píxeles se le conoce como “Pixel Art”.

El primer uso de la palabra píxel fue en 1965 por Frederic C. Billingsley del Laboratorio de Propulsión a Reacción (JPL) en uno de los papeles para describir los elementos pictóricos de imágenes escaneadas por sondas espaciales enviadas a la luna y a Marte. Billingsley aprendió de esta palabra de Keith E. Mcfarland, un ingeniero americano dedicado al desarrollo de técnicas de procesamiento de imágenes digitales. Keith, a su vez, no conoce el origen de esta palabra, tan solo que era un término usado por entonces.

La memoria de los ordenadores y consolas estaba limitada por su tecnología. Esto significa que el arte tiene que ser tan optimizado como sea posible.

Este trabajo se centra en la emulación de gráficos de los años 70 y 80, cuando salió el ordenador Apple II, durante la segunda generación de videojuegos y durante el principio de la tercera generación. El énfasis de este trabajo se trata en la definición de la palabra emulación. Emular, según la Real Academia Española, significa “imitar las acciones de otros procurando igualarlas e incluso excederlas”. En este trabajo, lo que se emulan son los dispositivos elegidos y sus limitaciones y capacidades.

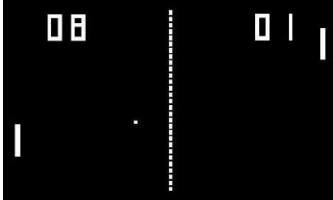


Fig. 2.

El motivo tras elegir este periodo de tiempo es por la limitación de estos productos tanto electrónicamente como artísticamente, creando la necesidad de adaptarse a estos productos como creador y de crear resultados digitales siguiendo unas normas a seguir.

Trabajar con dispositivos más antiguos que en los años 70 implicaría trabajar con productos que, mayoritariamente, son sobre todo analógicos, y aunque pueda haber limitaciones al respecto, se requiere conocimiento electrónico, y es preferible trabajar en un aspecto digital. Y aunque se trabajase en una hipotética situación digital, la limitación analógica de estos es demasiado grande para crear algo más allá que los gráficos de las consolas Pong, o de sonidos demasiado simples.



Fig. 3.

Los dispositivos electrónicos tras los años 80 tienen más capacidad gráfica y sonora para crear obras de arte y productos a partir de estos. Más resolución, más memoria, procesadores más rápidos, chips de sonido con más capacidad sonora y con más entradas para comunicarse con los productos electrónicos.

Idealmente hablando, una persona se debe inclinar a obtener y usar productos que sean mejores que sus versiones anteriores, o aquellos productos que sustituyen, y si uno es artista, ¿no sería mejor que crease obras de arte con los lienzos más grandes, con los bloques y las rocas más grandes para esculpir, con las mejores herramientas posibles? Y si el artista se dedica en el arte digital, ¿no es preferible que sus obras tengan las resoluciones más grandes, usando las herramientas más rápidas y potentes?



Fig. 4.

No necesariamente. Y no es porque un artista no tenga la posibilidad de crear obras más grandes, o mejores herramientas. También incluye el deseo de algunos artistas de crear obras a partir de objetos pequeños. Por ejemplo, hay personas que crean esculturas diminutas a partir de los cantos rodados del río, del grafito quebradizo y frágil del lápiz, o de dibujos de lienzos no más pequeños que la yema de un dedo. Incluso hay artistas que pintan en algo tan diminuto e imperceptible como un grano de arroz.

Por eso, en este caso, se trabaja siguiendo las limitaciones de los dispositivos elegidos, unos dispositivos que aparecieron entre el 75 y el 85. El objetivo en este caso es igual que esculpir en un canto rodado o dibujar en un grano de arroz es crear arte a partir de las limitaciones establecidas.

Un artista crea obras de arte pequeñas, con el mismo propósito que un escalador. Un escalador no escala una montaña porque necesite algo de esta. Lo hace porque puede, y quiere demostrarlo. Y con las limitaciones establecidas el artista, este puede crecer más como artista que de otro modo no sería capaz.

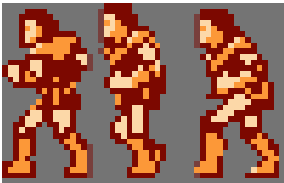


Fig. 5.

Parte del interés de estos estilos artístico es el desafío de trabajar dentro de las limitaciones que un desarrollador afrontaba cuando se creaba un programa o juego. Por ejemplo, la NES dispone de una paleta de 54 colores, de los cuales solo puede haber 25 en pantalla (un color de fondo, cuatro grupos de cuatro colores para los mapas y cuatro grupos de cuatro colores para sprites y personajes), con un tamaño de sprites de 8x8, y una resolución de pantalla de 256x240 píxeles.

Esto implicaba que se debía de usar la memoria disponible teniendo en cuenta la reutilización de los sprites, los tiles, y las paletas de colores.

Como menciona Kiwi en la página web creada para píxel art 2dwillneverdie, aunque la tecnología permite modelos de 3d o imágenes a alta resolución, píxel art es un medio establecido con una enorme variedad de estilos. Incluso si su rol en la cultura pop es mayoritariamente nostalgia, píxel art todavía está en uso hasta hoy. Es el arte más definido y nítido que se puede producir en una pantalla. Optimiza gráficos para dispositivos móviles o con una pantalla reducida.

Además, píxel art es accesible y de fácil introducción, con herramientas disponibles para todos los sistemas operativos e incluso herramientas en línea.

Es fácil de crear y fácil de modificar, y se puede concebir mucho detalle a partir de un píxel.

*“La fuerza del movimiento píxel yace en su simplicidad abstracta. (...) Puede estar estirándonos a nuestros hilos nostálgicos de una infancia romántica de máquinas de arcade y mandos, pero lo mismo se puede decir del movimiento de las manualidades, de la moda vintage, y de las películas en blanco y negro. Entonces, ¿tiene el estilo píxel un valor mayor para gente fuera del mundo de los videojuegos y los ordenadores? A medida que progresamos más y más en la era digital, sea arte, entretenimiento o comunicación, el píxel es nuestro punto de partida. Es nuestro caballete y pintura digital, nuestro cincel y piedra, ladrillos y mortero, papel y bolígrafo. Conectamos con su simplicidad y encontramos un significado más profundo en su abstracción. Tiene su propio lenguaje visual, imaginería, movimiento, interacción y audio. Así que sí, creo que lo tiene.”<sup>1</sup>*

Crear pixel art puede abrir la puerta a otras formas de expresarse, y a aprender a expresarse con limitaciones.

Pixel art es una manera práctica para páginas web y programas de optimizar imágenes y poder reemplazar una imagen de alta resolución por otra que ofrece el mismo propósito con colores y resolución de imagen limitada, a veces ofreciendo un resultado superior debido a la velocidad de procesamiento que estos ofrecen en comparación a una imagen de alta resolución y color.

Pixel art es un estilo artístico muy adaptable, no solo en pantallas, sino también como parte de una cultura que se muestra en tapicería, decoraciones de pared, mosaicos, losas de baño, abstracción geométrica o incluso en modelos de 3 dimensiones. Puede también ser representacional y también



Fig. 6.

---

<sup>1</sup> COTTEE, Simon. Pixel – A pixel art documentary.

<http://simoncottee.blogspot.com/2010/05/pixel-pixel-art-documentary.html> (22 mayo 2010) (consulta: 23 noviembre 2021).



puede ser abstracto, con espacio para la imaginación. Tiene este poder de expresión y de ambigüedad.

Cuanto más específico dibujas algo, menos te identificas con lo que se ha dibujado.

Además, en los años donde se usaba televisores CRT, los píxeles no se representaban de la misma forma que se representa de manera digital, debido a la naturaleza de los televisores CRT. Se veía difuminado, por lo que esto también promovía al estilo de spriting en sí.

Por otra parte, los dispositivos móviles todavía usan una resolución de pantalla relativamente baja, y a medida que pasa el tiempo, se sacan al mercado productos tecnológicos con pantalla digital pequeña, como los recientes relojes digitales deportivos, o relojes inteligentes. Por ejemplo, el Fitbit Sense tiene una pantalla de 336x336 píxeles.



Fig. 7.

Ambos estilos de arte ofrecen su fuerza en su simplicidad abstracta y de la nostalgia que ofrece. Se puede decir que es retrógrado, pero lo mismo se puede decir de las artes tradicionales, moda vintage, y películas o fotografías de blanco y negro. Hay mejores formas de expresar, pero no significa que sea obsoleto. De hecho, a medida que se vayan sacando tecnologías nuevas y más pequeñas, las limitaciones de estas se pondrán a prueba a partir del uso de la programación y del gráfico.

## 2. Objetivos

El objetivo de este trabajo es investigar y comprender las limitaciones y especificaciones gráficas de cada dispositivo electrónico, y la creación de sonido de uno de estos.

Una vez comprendidas las limitaciones de estos, se escribe unas directrices que mantengan fielmente dichas limitaciones y especificaciones gráficas.

Por último, se crea al menos una obra de cada dispositivo siguiendo las directrices creadas. Se explicarán las herramientas y programas usados para la creación de gráficos y/o audio.

Estos tres pasos son los que se va a seguir para poder emular obras de arte según el dispositivo al que se le inspira, teniendo en cuenta que lo que se busca conseguir bajo definición de emular es “imitar una cosa procurando igualarla”.

Los dispositivos electrónicos que se emularán para este proyecto son:

- Apple II.
- Atari 2600 / Atari VCS (Video Computer System)
- Commodore 64
- NES (Nintendo Entertainment System) / Famicom

En este proyecto se incluye también la aplicación de varias formas de experimentación de obras, como la creación de arte PETSCII o la implementación directa de dichas obras dentro de un producto ya licenciado (un proceso conocido como modding).

En el caso de modding, la modificación de un producto licenciado se hace estrictamente bajo un propósito académico de investigación para la demostración de este.







En este proyecto, se tratará de analizar las posibilidades expresivas de estos entornos en la situación contemporánea.

### 3. metodología

El proyecto usará una variedad de herramientas digitales durante el desarrollo del trabajo.

- Un software de edición de imágenes (Clip Studio Paint).
- Un software en el que se pueda dibujar con píxeles ( Clip Studio Paint).
- <http://petscii.krissz.hu/> , herramienta en línea usado para la creación de gráficos de la Commodore 64.
- Un software de edición de archivos de solo lectura, o .ROM (YY-CHR).
- Herramienta de lupa.
- Un emulador de .ROMS (NESTOPIA).
- Herramienta de producción de sonidos y música, o tracker, para NES/Famicom (Famitracker).



0/1	00	Negro	0	0	0	#000000	
0/1	11	Blanco	255	255	255	#FFFFFF	
0	01	Verde	20	245	60	#14F53C	
0	10	Violeta	255	68	253	#FF44FD	
1	01	Naranja	255	106	60	#FF6A3C	
1	10	Azul	20	207	253	#14CFFD	

Sin embargo, Apple tiene dos modos de resolución de pantalla, y la mencionada actualmente es el modo de resolución alta.

En el modo de resolución baja, el ordenador ofrece una limitación de 40-48 píxeles a cambio de la capacidad de usar un total de 16 colores, con la opción de ofrecer 4 líneas de texto con 40 píxeles, o sin texto con 48 píxeles. Para ofrecer la capacidad de 16 colores, el ordenador hace uso de los 8 bits de color, incluido el bit de paleta, pero en vez de usar 1 o 2 píxeles, usa toda la sección de 7 píxeles para crear un píxel en este modo, lo que explica que una pantalla de 280x192 se vuelva 40x48 en modo de baja resolución: 280 dividido por 7 (el número de píxeles por sección) es 40.

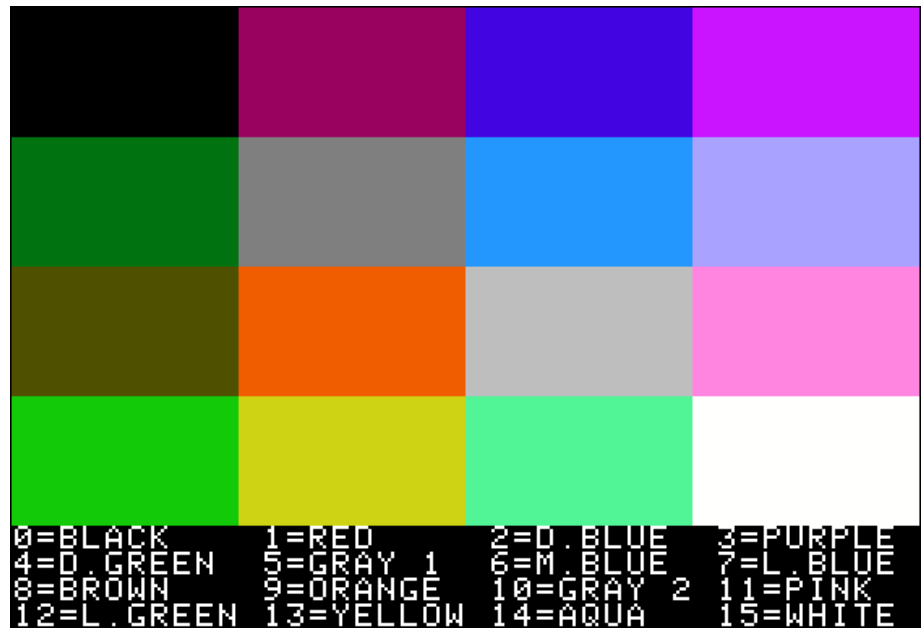


Fig. 8.

Debido a la complejidad y la falta de practicidad de el modo de resolución baja, la mayoría de los desarrolladores del ordenador creaban programas en modo de resolución alta (280x192, píxeles) y con pantalla monocroma y/o de color.

Estas son las directrices para crear gráficos:

- El tamaño de la pantalla es de 280x192 píxeles.
- La pantalla está dividida en filas de 7 píxeles que contienen 8 bits de información.
- Si la pantalla es monocroma, el primer píxel no tiene uso, y dependiendo del estado de los 7 bits restantes, le indica a la pantalla si los píxeles se vuelven blancos o negros.
- Si la pantalla es de color, los píxeles adquieren doble de anchura, con la opción de estos píxeles anchos de cambiar a 2 colores extra, dependiendo de qué bit de paleta se haya elegido al principio.

(x-0000000)



Fig. 9.

En esta parte del proyecto, se usará una fotografía para la creación de una imagen monocroma o de 1 bit, con la misma resolución que una pantalla de Apple.

Antes de explicar el proceso, hay ciertos aspectos que aclarar en este proceso. Entre ellos la capa de binarización, y el uso y organización de carpetas y máscaras.

La capa de corrección de binarización es una capa que divide la capa actual en blanco puro y/o negro puro. La opción deslizante del umbral, con un valor entre 1 a 255, decide en qué valor de iluminación la imagen se separa en blanco y negro. Este proceso también se le conoce en otros programas como “umbral” o “threshold”.

La capa de corrección funciona siguiendo tres pasos: comprueba el valor de iluminación de un píxel, y lo compara con el número del umbral. Si el número del valor de iluminación de píxel es mayor que el umbral, ese píxel se vuelve blanco. Si el valor es inferior al umbral, ese píxel se vuelve a negro.

En umbral de valor 1, la binarización separará la mayoría de los píxeles de la imagen a blanco puro, con la excepción del negro puro.

En umbral de valor 255, la binarización separará la mayoría de los píxeles de la imagen a blanco, a excepción de negro puro.

En un umbral con valor 127, el valor medio, los valores dependerán de si los píxeles son más claros, o más oscuros, que el gris neutro.

En cuanto a las carpetas, el motivo por el cual se organizan de este modo durante el proceso es debido a que todas las capas que permanezcan por debajo de la capa de corrección se verán afectadas por la misma. Mientras que dentro de las carpetas, las capas de corrección solo afectará a las capas inferiores que estén tan solo dentro de dicha carpeta.

Esto ofrece la ventaja de que se puede modificar el umbral de la capa de corrección la capa de la imagen, en caso de que se quiera.

En caso de que el proceso sea demasiado complicado, se puede quitar la máscara de la carpeta y se combina las carpeta con sus capas para crear una sola capa de imagen. El inconveniente que conlleva este paso es que esa capa no se podrá modificar su umbral.

Por último, debido a la resolución de pantalla de Apple II, se pueden usar imágenes de baja resolución. Sin embargo, independientemente de la resolución, deben ser imágenes adecuadamente definidas e iluminadas para un buen resultado. En caso de que se quiera crear una imagen con más resolución, el 3er paso se puede pasar por alto. De ser así, se ha de tener en cuenta que el tamaño de la imagen elegida sea al menos igual o mayor que el tamaño del lienzo.

Este es el proceso de cómo crear una imagen con las especificaciones de blanco y negro de Apple II usando fotografía.

1. Encontrar o crear una imagen.
2. Abrir un programa de edición de fotografías e imagen.

En este caso se usa Clip Studio Paint.

3. Reducir y/o recortar el tamaño de imagen a la resolución de pantalla de Apple II (280 x 192).

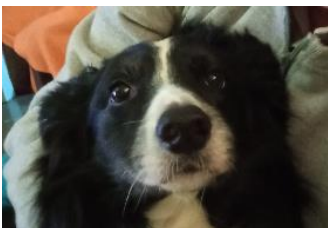


Fig. 10.



4. Crear una carpeta de capa
5. Añadir una capa de máscara en la carpeta de capa
6. Insertar una copia de la fotografía
7. Insertar dentro de la carpeta y por encima de la fotografía una capa de corrección de binarización con un umbral estándar.

8. Llamar a la carpeta "BASE".

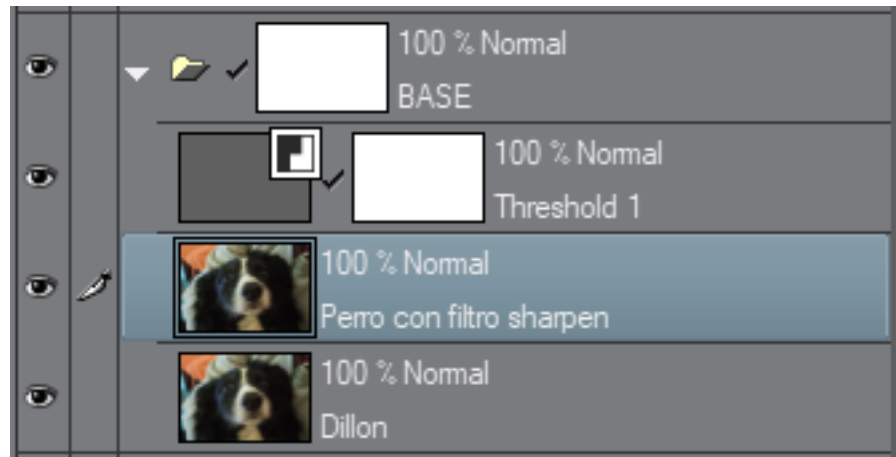


Fig. 11.

9. Copiar la carpeta BASE tantas veces como se necesite.
10. En cada carpeta copiada, cambiar el umbral de la capa de corrección a un número deseado, y nombrar cada carpeta copiada el número de umbral de su capa de corrección.
11. Ordenar las carpetas en orden ascendente o descendente, dependiendo del proceso de trabajo de cada fotografía, a excepción de la carpeta BASE como última capa y desactivada. En este caso, se ordenaron las capas en orden descendente.
12. Enmascarar todas las carpetas de umbral enumeradas a excepción de la última capa enumerada.

13. Comenzar a desenmascarar cada capa hasta recibir el resultado deseado. Usar capas con detalles que ayuden a delimitar cada capa.

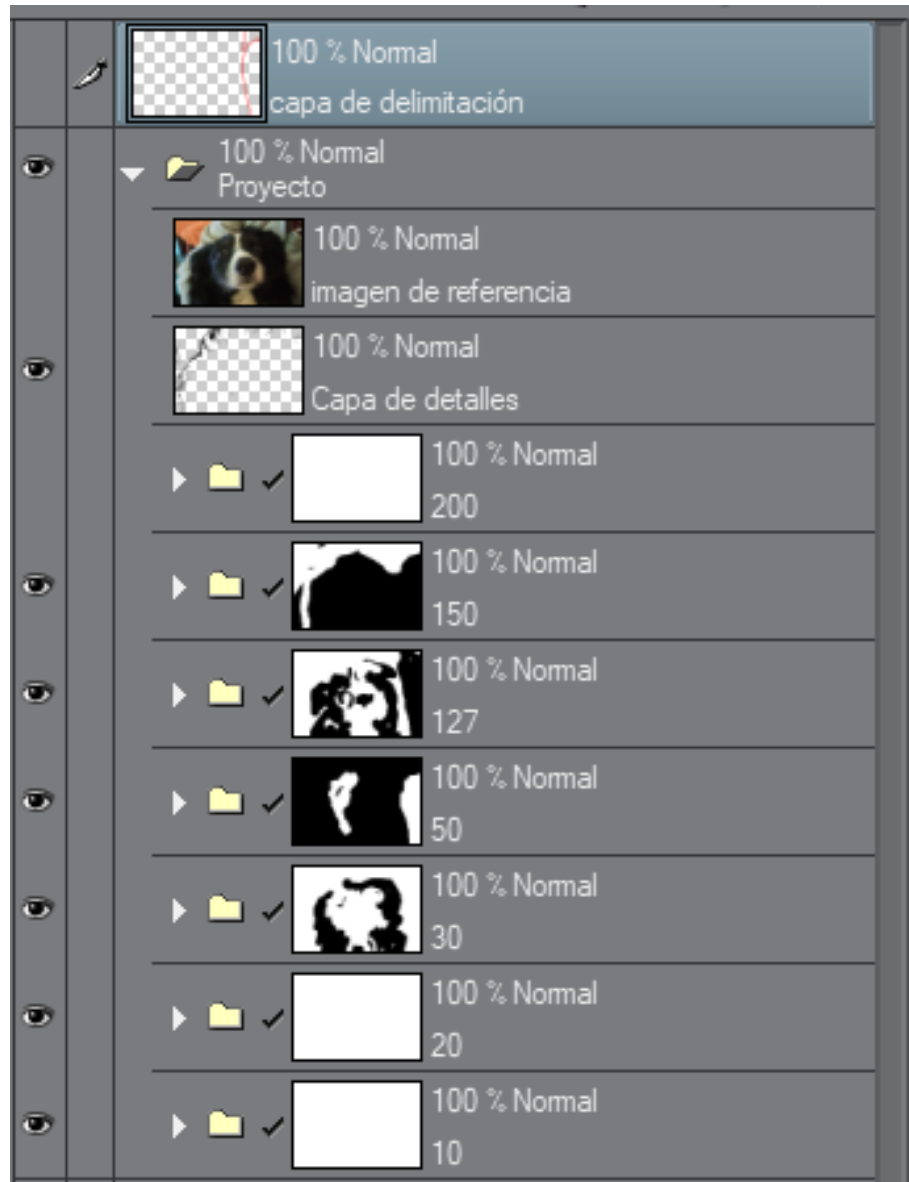


Fig. 12.

14. Añadir toques finales.

En ciertos casos, se pueden crear más detalle a partir de las capas con los umbrales más oscuros. En este caso, se puede usar los pliegues de

la ropa de la capa de umbral más oscuro como un elemento para añadir más contraste a la imagen actual. Para ello, se elige la herramienta de varita mágica, o herramienta de autoselección para elegir los píxeles blancos de toda la capa, se deselecciona las partes de la capa que no se quiera usar, y esa selección se puede usar para otras capas. En este caso, la selección de los pliegues blancos se usa para dibujar en negro para ofrecer a los pliegues más contraste al respecto.



Fig. 13.



Fig. 14.



Fig. 15.



Fig. 16.

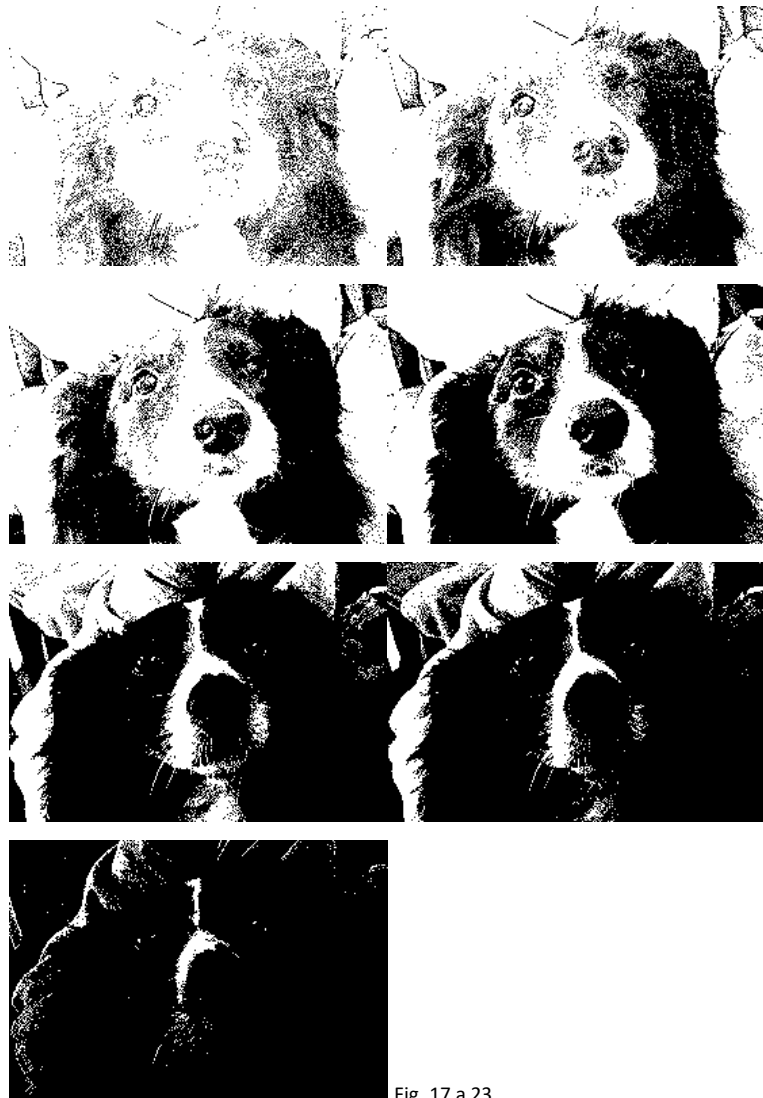


Fig. 17 a 23.



Fig. 24.

## 4.2. ATARI 2600 (1977)

La mayoría de los ordenadores tienen un chip de video especializado para enviar información al monitor de manera sincronizada para crear imagen o video en la pantalla. Sin embargo, la Atari 2600 era una consola sin un chip de video. Lo que esta consola hace, igual que otros productos sin chip gráfico, es crear la imagen a partir del CPU directamente a la pantalla. El problema que conlleva la falta del chip gráfico es el uso intensivo del CPU para crear imagen y video, con la posibilidad de producirse errores técnicos, como se muestra en dos juegos de la Atari 2600: Pacman y Mangia.

La Atari 2600 tiene una pantalla de 192 píxeles de alto por 160 de ancho, con una paleta de 128 colores (8 colores de blanco a negro, y 15 columnas de tono, cada uno con 8 colores de la misma tonalidad y diferentes grados de luminosidad y saturación). Un detalle importante a mencionar sobre la Atari 2600 es que la anchura de la pantalla es realmente 320 píxeles de ancho, pero en este caso, la anchura de los píxeles en esta consola dobla el tamaño de anchura. Por este motivo, la Atari tiene un estilo gráfico ancho, y los gráficos están creados con ese detalle en cuenta.

Antes de poder mencionar cómo se emula los gráficos de la consola, se debe aclarar que en el proceso de creación de gráficos de la Atari 2600, la consola usa la definición de scanlines. Un scanline, o línea de exploración, es una línea de píxeles dentro de la pantalla. Y debido al diseño de la consola, comprender la naturaleza gráfica de esta al detalle requiere conocimiento técnico. E incluso teniendo en cuenta la información técnica, varios juegos de la consola llevan el límite gráfico a partir de trucos de programación y el uso adecuado de las limitaciones para sacar el mejor provecho de la consola.

Teniendo esto en cuenta, para emular los gráficos de la consola, se siguen estas directrices:

- Los píxeles tienen el doble de anchura.
- El tamaño de la pantalla es de 320x192 píxeles, usando píxeles de doble anchura.
- El elemento de fondo utiliza toda la pantalla, y se pueden cambiar los colores de los scanlines.
- El elemento de playfield está en frente del elemento de fondo, y se usa para mostrar paredes o el fondo. Usa la mitad de la pantalla, pero se puede repetir y copiar este elemento para cubrir la pantalla.
- Los sprites tienen 8 píxeles anchos de anchura (o 16 píxeles normales)
- Los sprites pueden tener hasta 192 píxeles de altura.
- Cada objeto puede tener un color predeterminado, con la posibilidad de añadir más colores y con la limitación de que solo se puede cambiar a un color por scanline.
- Hay 2 recursos de bola y 2 recursos de misil.
- Las bolas tienen un tamaño de 1x1 píxeles.
- Los misiles tienen el mismo tamaño que las bolas, con la inclusión de que añade un trazo o línea a esta, desde el punto de origen programado hasta la bola.
- Debido a las limitaciones técnicas de la consola, el patrón, las imágenes deben tener un aspecto simple, y que se incline a un aspecto horizontal/vertical.

Siguiendo estas directrices, para crear un sprite de Atari:

1. Antes de comenzar, se debe tener en cuenta que al final del proceso, el resultado se ensanchará el doble de anchura para seguir las mismas limitaciones que la consola Atari.
2. Se crea un lienzo con un máximo de anchura de 8 píxeles.



3. Se crea una capa de fondo para el lienzo de programa.
4. Se crea una segunda capa con un color diferente.
5. Se elige una herramienta con la que poder dibujar en la máscara de la capa. Esta herramienta de dibujo debe tener un 100% de opacidad y con un borde duro. En la mayoría de los programas, la herramienta se suele llamarle lápiz (pencil) o “herramienta binaria” (binary tool).
6. Se aplica una máscara en la segunda capa, y se enmascara la capa hasta que no se vea el color de la capa (gris oscuro).
7. Se dibuja en la máscara de la capa, creando la silueta de la imagen deseada. (Recordar paso 1.).
8. Se elige, a partir de la paleta de la consola Atari, los colores a elegir de la imagen. Señalizamos cada scanline dibujando un píxel con el color deseado.
9. Se desactiva la máscara de la capa y se comienza a colorear horizontalmente los colores, con una herramienta de selección y cubo de relleno o con la herramienta de lápiz con 1 píxel de densidad. Se vuelve a activar la máscara de capa una vez terminado.
10. (Opcional) En caso de querer animar, volver a seguir los pasos 1 a 8 en capas separadas.
11. Una vez obtenido el resultado deseado, se dobla la resolución del proyecto en los ajustes de resolución.
12. Se añade los toques finales, teniendo en cuenta el paso 1.

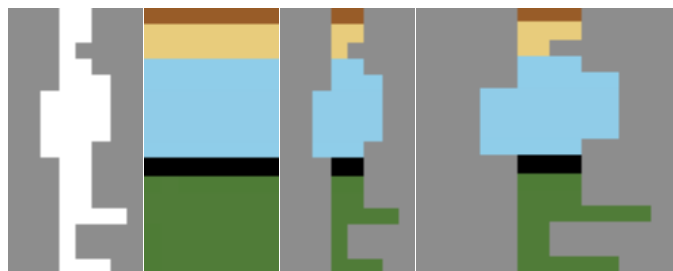


Fig. 25.

### 4.3. COMMODORE 64 (1982)

La Commodore 64 es uno de los ordenadores más populares gracias a su lenguaje de programación BASIC y su uso práctico como de entretenimiento, con una resolución de 320 por 200 píxeles. Incluso hasta hoy se usa el ordenador, o programas que emulan dicho ordenador, para crear juegos o demostraciones que ponen al límite la capacidad del ordenador.

En el caso de la C64, usa el sistema de células de colores como secciones para determinar el tipo de color de una zona de píxeles. A diferencia del sistema de secciones de Apple 2, cuyas secciones eran de 7 píxeles horizontalmente, en la Commodore 64 se usa células de color de 8x8 píxeles de altura por anchura. Puedes elegir qué color para los píxeles de primer plano y los píxeles de fondo, y cada célula de color solo requiere 1 byte (8 bits) de memoria para definir el color de ambos píxeles, primer plano y fondo, usando 9kb para toda la pantalla.

Además, la Commodore 64, igual que el ordenador Apple II, ofrece también un modo llamado "modo multicolor", que ofrece la capacidad de aplicar 4 colores en una célula de color a cambio de usar 2 píxeles a la vez en vez de uno, dividiendo la anchura de la pantalla por la mitad, a 160 píxeles.

El ordenador ofrece una cantidad de 8 sprites, de 24x21 píxeles de tamaño, cada uno enumerado de 0 a 7 y cada uno requiere 64 bytes, con un byte sin usar. En el ordenador, se guardaban en los números de memoria 2040 a 2047.

Un sprite requiere 63 bytes para definir su forma, y cada fila de píxeles tiene 3 bytes cada uno (63 bytes = 21 filas x 3 bytes). Esto se aplica tanto en el modo normal como en el modo multicolor.

La paleta de la Commodore 64 consta de 16 colores. Y con las observaciones de Philip Timmerman [Fig. 26], estos son los colores que usa el ordenador doméstico:

#	name	r	g	b
0	black	0,0 ( 0)	0,0 ( 0)	0,0 ( 0)
1	white	254,999999878 (255)	254,999999878 (255)	254,999999878 (255)
2	red	103,681836072 (104)	55,445357742 ( 55)	43,038096345 ( 43)
3	cyan	111,932673473 (112)	163,520631667 (164)	177,928819803 (178)
4	purple	111,399725075 (111)	60,720543693 ( 61)	133,643433983 (134)
5	green	88,102223525 ( 88)	140,581101312 (141)	67,050415368 ( 67)
6	blue	52,769271594 ( 53)	40,296416104 ( 40)	121,446211753 (121)
7	yellow	183,892638117 (184)	198,676829993 (199)	110,585717385 (111)
8	orange	111,399725075 (111)	79,245328562 ( 79)	37,169652483 ( 37)
9	brown	66,932804788 ( 67)	57,383702891 ( 57)	0,0 ( 0)
A	light red	153,690586380 (154)	102,553762644 (103)	89,111118307 ( 89)
B	dark grey	67,999561813 ( 68)	67,999561813 ( 68)	67,999561813 ( 68)

C	grey	107,797780127 (108)	107,797780127 (108)	107,797780127 (108)
D	light green	154,244479632 (154)	209,771445903 (210)	131,584994128 (132)
E	light blue	107,797780127 (108)	94,106015515 ( 94)	180,927622164 (181)
F	light grey	149,480882981 (149)	149,480882981 (149)	149,480882981 (149)

Fig. 26. Tablero de información de los colores por Philip Timmerman <sup>2</sup>

En la página web de Philip se encuentra el proceso por el que se descubre el espectro de los colores pasados a números de valor en RGB, con su valor aproximado en paréntesis. Debajo está el mismo tablero de colores con solo el valor aproximado, su color hexadecimal y el color en sí.

<sup>2</sup> <http://unusedino.de/ec64/technical/misc/vic656x/colors/> (consulta: 23 noviembre 2021)

















#	Nombre	R	G	B	HEX	Color
0	Negro	0	0	0	#000000	
1	Blanco	255	255	255	#FFFFFF	
2	Rojo	104	55	43	#68372B	
3	Cian	112	164	178	#70A4B2	
4	Morado	111	61	134	#6F3D86	
5	Verde	88	141	67	#588D43	
6	Azul	53	40	121	#352879	
7	Amarillo	184	199	111	#B8C76F	
8	Naranja	111	79	37	#6F4F25	
9	Marrón	67	57	0	#433900	
A	Rojo claro	154	103	89	#9A6759	
B	Gris oscuro	68	68	68	#444444	
C	Gris	108	108	108	#6C6C6C	
D	Verde claro	154	210	132	#9AD284	
E	Azul claro	108	94	181	#6CSEB5	
F	Gris Claro	149	149	149	#959595	

Fig. 27. Tablero tabla anterior con valores redondeados con RGB y código HEX. <sup>3</sup>

Esta paleta destaca sobre todo por su aspecto único y fácil de identificar.

Cuando se ve una imagen de la Commodore 64, se identifica fácilmente por la paleta de colores, la separación exacta en tiles de 8x8 de los tiles y sprites y de la limitación de tener un máximo de 2 colores en modo normal, y 4 colores en modo multicolor.

<sup>3</sup> <http://unusedino.de/ec64/technical/misc/vic656x/colors/> (consulta: 23 noviembre 2021)

### Creando un Sprite a partir de un sprite sheet

Un Sprite sheet, u hoja de Sprite, es una hoja con una cuadrícula y con información en sus lados para indicar el número de la columna y el valor que tiene cada cuadrícula.

Bitwert	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1
1. Reihe																								
2. Reihe																								
3. Reihe																								
4. Reihe																								
5. Reihe																								
6. Reihe																								
7. Reihe																								
8. Reihe																								
9. Reihe																								
10. Reihe																								
11. Reihe																								
12. Reihe																								
13. Reihe																								
14. Reihe																								
15. Reihe																								
16. Reihe																								
17. Reihe																								
18. Reihe																								
19. Reihe																								
20. Reihe																								
21. Reihe																								

Fig. 28.

Esta cuadrícula se hacía en papel con unas proporciones de 24x21, el tamaño de un sprite. Cada sprite se divide en secciones de 3 verticalmente , por lo que cada fila contiene 8 píxeles, y esos 8 píxeles se pueden representar con un byte, dependiendo del número que se le dé. Cada píxel es un bit del byte, y cada bit tiene un valor establecido, de 1 a 128, con una combinación de sección de píxeles de 0 a 255. Dependiendo de si se dibuja en ese píxel o no en la pantalla, se le suma su valor a la sección que le corresponde.

Binario	0	0	1	0	1	1	0	1	→→	00101101
Valor	124	64	32	16	8	4	2	1		
Píxel	■	■		■			■			
Suma	0	0	32	0	8	4	0	1	32+8+4+1 →	45

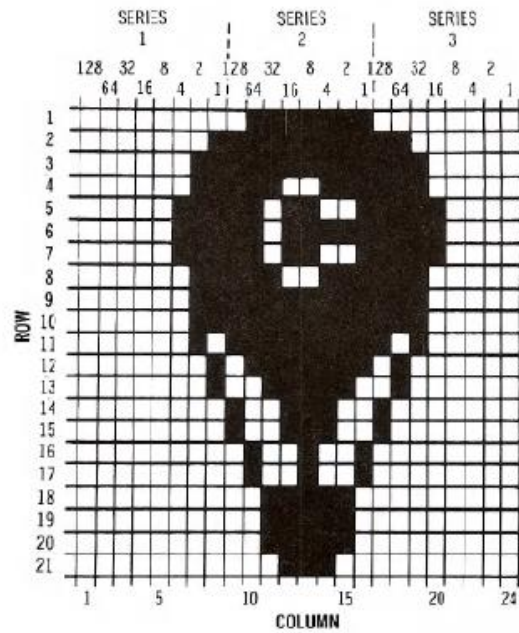
También se puede usar el valor hexadecimal para describir el valor de la sección de píxeles. Para hacerlo, se divide la sección en 2, y cada 4 píxeles indica un valor hexadecimal.

Decimal	0	0	1	0	1	1	0	1		
Valor	8	4	2	1	8	4	2	1		
Píxel	■	■		■			■			
Suma	0	0	2	0	8	4	0	1	2 / 8+4+1	2 / D

En C64, este valor se representaría como \$2D, con \$ siendo el símbolo que indica al ordenador que trata con un valor hexadecimal.

Para el modo multicolor, el sistema para crear el mismo sistema que en el modo normal, con la diferencia de que cada píxel usa 2 píxeles de anchura, y puede usar 4 colores. Este sistema emula el mismo sistema de modo multicolor del ordenador Apple II.

Esto implica que, en la cuadrícula, se debe dibujar y colorear los píxeles en pares de dos, y dependiendo del número de los bits encendidos y apagados de cada pareja, representa su píxel y color correspondiente.



The series, that make up row two are calculated like this:

Series 1: 

0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---

 $\frac{1}{1} = 1$

Series 2: 

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

  
 $\uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow$   
 $128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255$

Series 3: 

1	1	0	0	0	0	0	0
---	---	---	---	---	---	---	---

  
 $\uparrow \quad \uparrow$   
 $128 + 64 = 192$

For row 2, the data would be:

DATA 1,255,192

In the same way, the three series that make up each remaining row would be converted into their decimal value. Take the time to do the remainder of the conversion in this example.

Now that you have the data for your object, how can it be put to use? Type in the following program and see what happens.

Fig. 29.



```

1 REM UP, UP, AND AWAY!
5 PRINT "{CLR/HOME}"
10 V=53248 : REM START OF DISPLAY CHIP
11 POKE V+21,4 : REM ENABLE SPRITE 2
12 POKE 2042,13 : REM SPRITE 2 DATA FROM 13TH BLK
20 FOR N = 0 TO 62: READ Q : POKE 832+N,Q: NEXT
30 FOR X = 0 TO 200 : GETS ITS INFO FROM DATA*
40 POKE V+4,X: REM UPDATE X COORDINATES
50 POKE V+5,X: REM UPDATE Y COORDINATES
60 NEXT X
70 GOTO 30 : INFO READ IN FROM Q*
200 DATA 0,127,0,1,255,192,3,255,224,3,231,224
210 DATA 7,217,240,7,223,240,7,217,240,3,231,224
220 DATA 3,255,224,3,255,224,2,255,160,1,127,64
230 DATA 1,62,64,0,156,128,0,156,128,0,73,0,0,73,0
240 DATA 0,62,0,0,62,0,0,62,0,0,28,0

```

\*FOR MORE DETAIL ON READ & DATA SEE CHAPTER 8.

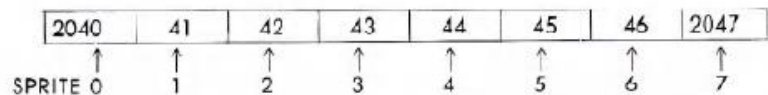
If you typed everything correctly, your balloon is smoothly flying across the sky (page 72).

In order to understand what happened, first you need to know what picture making locations control the functions you need. These locations, called registers, could be illustrated in this manner:

Register(s)	Description
0	X coordinate of sprite 0
1	Y coordinate of sprite 0
2 - 15	Paired like 0 and 1 for sprites 1-7
16	Most Significant Bit—X Coordinate
21	Sprite appear: 1=appear 0=disappear
29	Expand sprite in "X" Direction
23	Expand sprite in "Y" Direction
39 - 46	Sprite 0 - 7 color

In addition to this information you need to know from which 64 byte section sprites will get their data (1 byte is not used).

This data is handled by 8 locations directly after screen memory:



Now let's outline the exact procedure to get things moving and finally write a program.

Fig. 30.

Estas son las directrices para crear gráficos fieles a la Commodore 64:

- La pantalla tiene un tamaño de 320x200 píxeles, con una paleta de 4 colores.
- La pantalla está dividida por cuadrículas por 8x8 píxeles.
- En el modo estándar, cada cuadrícula puede tener un máximo de 2 colores.
- En el modo multicolor, las cuadrículas pueden tener un máximo de 4 colores, a cambio de que usa el doble de píxeles para adquirir información.

La Commodore 64 fue creada con el propósito de ser un ordenador personal asequible, por lo que en su memoria tiene una galería con bitmaps de 8x8 píxeles que se usan para los caracteres. Sin embargo, esta galería aprovecha la mitad del espacio restante para crear símbolos y formas que pueden servir para programas creados por el usuario. Estos símbolos pueden servir para crear interfaces o gráficos.

<http://petscii.krissz.hu/> (2021)

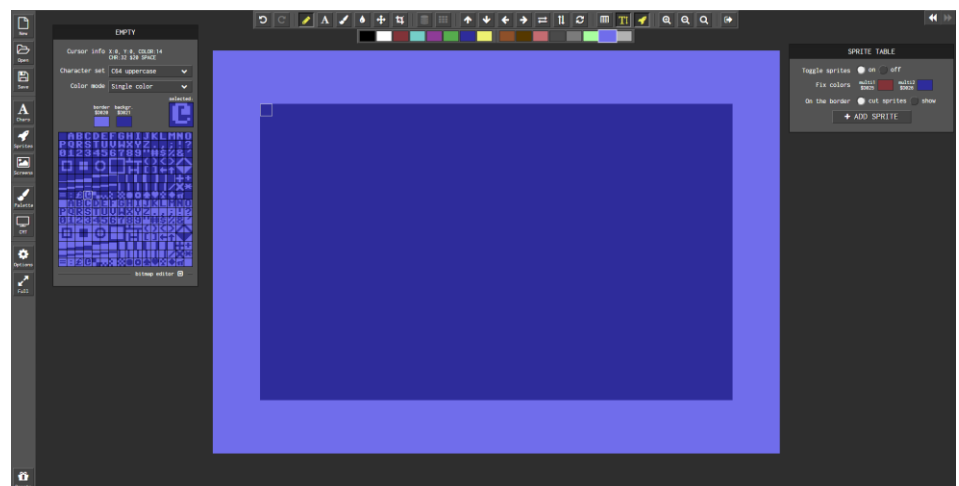


Fig 31.

Esta página web es un editor que usa dicha galería de caracteres para crear imágenes fieles a la Commodore 64. Además de poder usar los caracteres con los colores deseados, estos mismos caracteres se pueden editar

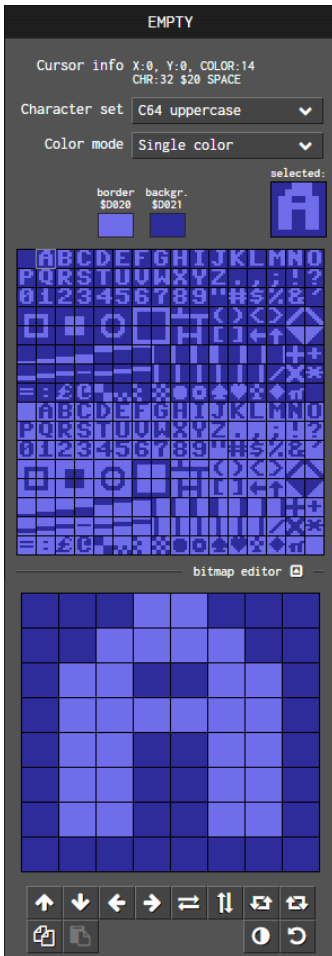


Fig. 32.

en caso de que se desee, se pueden crear sprites a partir de la pantalla de sprites con varias opciones, incluyendo la animación de dichos sprites, el uso de varias herramientas para el dibujo; la capacidad de importar recursos dentro del editor, sean caracteres, sprites o escenas, y, por último, la capacidad de exportar los proyectos en formatos compatibles para la C64, en imagen .PNG, o incluso raw bytes y/o código de assembly.

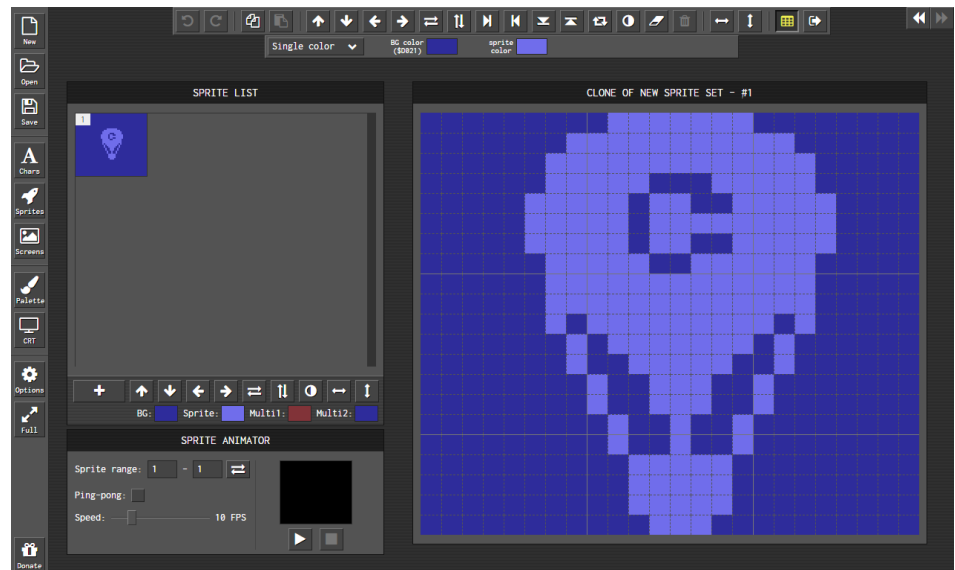


Fig. 33.

Además, tiene la opción de elegir entre varias paletas que emulan el color y el aspecto visual de las pantallas de CRT.

La barra de herramientas al lado izquierdo de la pantalla es la barra de archivos, pantallas y configuración:

- A. Crear un proyecto nuevo.
- B. Abrir un proyecto.
- C. Salvar el proyecto.
- D. Edición de caracteres.
- E. Edición de sprites.
- F. Edición de pantallas
- G. Selección de paletas

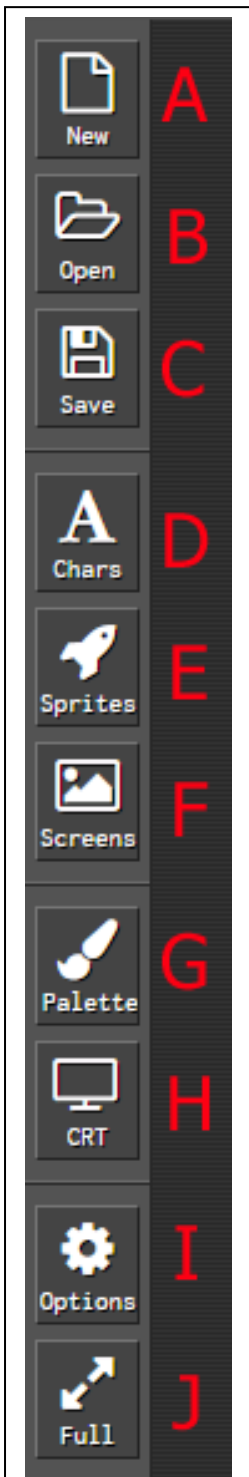


Fig. 34.

- H. Efectos de pantalla CRT..
- I. Opciones y configuración.
- J. Pantalla completa.

En este proyecto lo que se desea es crear una imagen de pantalla a partir de los caracteres predeterminados. Una vez dentro del proyecto de pantalla, nos aparecerán esta pantalla, con otra barra de herramientas en el lado superior de la ventana del navegador.



Fig 35.

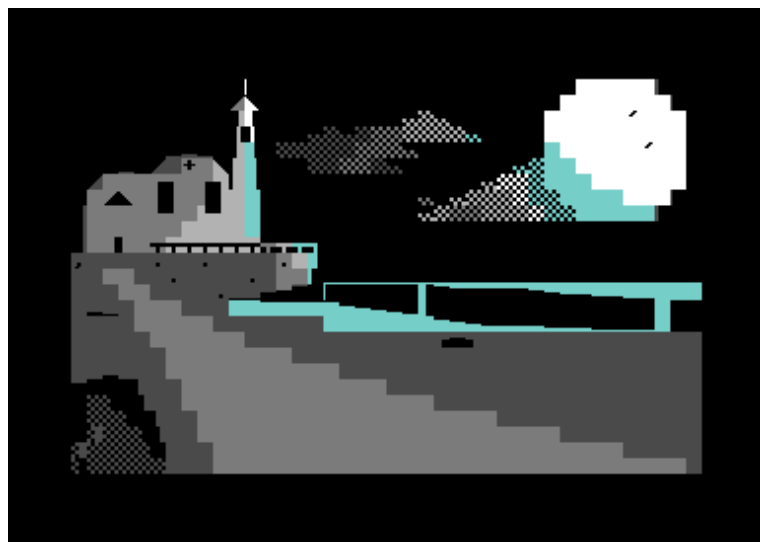
Estas son las funciones de dicha barra:

1. Deshacer.
2. Rehacer.
3. Dibujar con el color seleccionado.
4. Dibujar sin el color seleccionado.
5. Pincel para cambiar color.
6. Relleno.
7. Mover lienzo.
8. Selección.
9. Indefinido (no usado).
10. Indefinido (no usado).
11. Mover hacia arriba.
12. Mover hacia abajo.
13. Mover hacia la izquierda
14. Mover hacia la derecha.
15. Voltear horizontalmente.
16. Voltear verticalmente.
17. Reemplazar un carácter o color con otro.
18. Cuadrícula
19. Ventana de caracteres
20. Ventana de Sprites

21. Acercar vista.
22. Alejar Vista.
23. Reiniciar vista.
24. Exportar.

Una vez comenzamos a dibujar con el programa, se ha de tener en cuenta que la Commodore tendrá una imagen de fondo del color seleccionado, y los caracteres y símbolos con los colores que se seleccionen, por lo que el color de fondo debe de ser aquel que más prolifere dentro de la imagen. En este caso es el negro.

Una vez familiarizado con el area de trabajo, se comienza a bloquear la escena con caracteres cuadrados, y poco a poco se le da forma usando los recursos disponibles hasta obtener el resultado deseado.



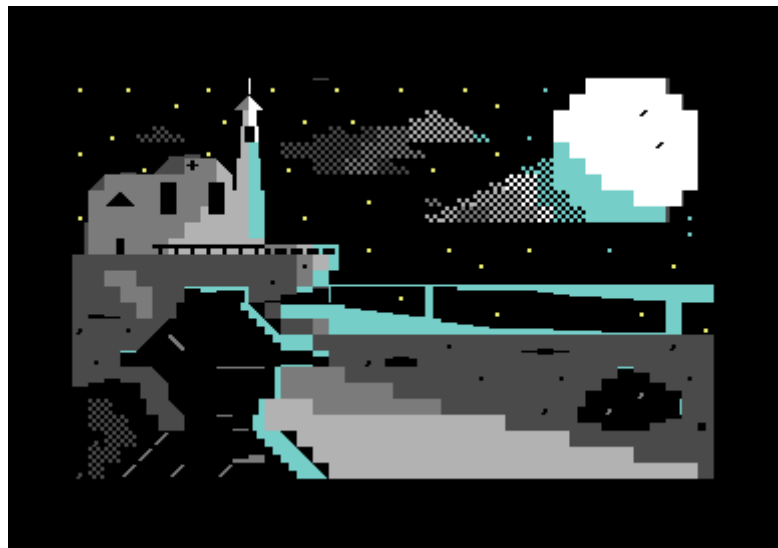
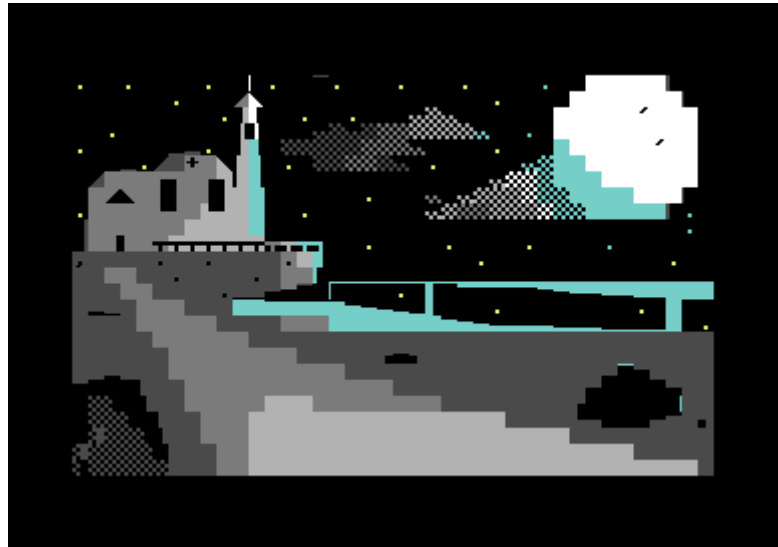


Fig. 36 a 39.

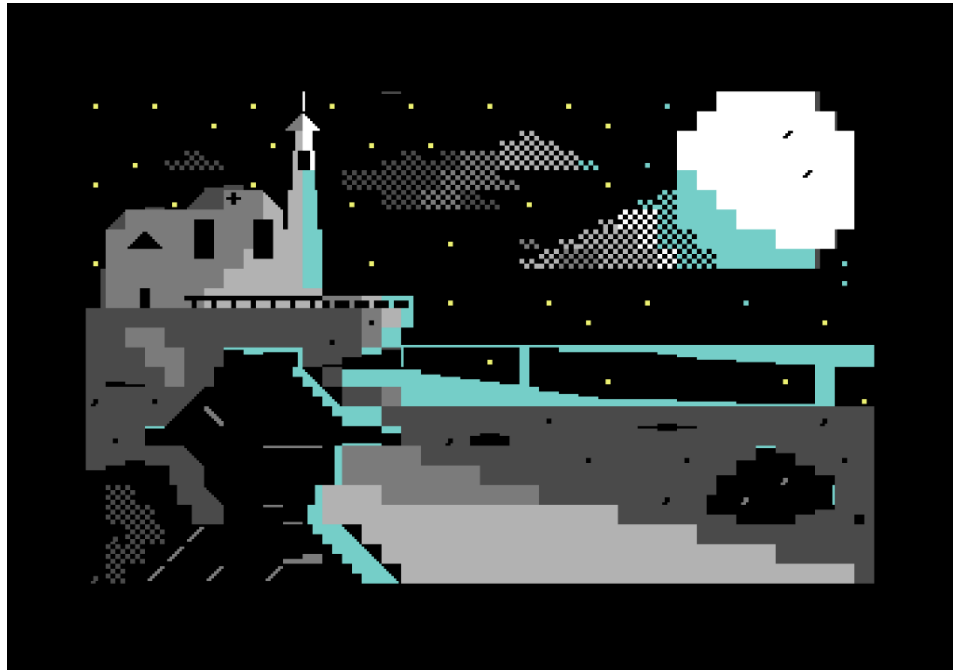


Fig. 40.

El motivo por el que en algunas zonas como la carretera tienen un aspecto cuadrado es debido a que los caracteres usan el negro como fondo, por lo que para “alisar” la carretera implica que una parte de ese carácter sea negro, como se muestra en la silueta del hombre con sombrero, en el hombro derecho.



#### 4.4. FAMILCOM/NES (1983)

El tamaño de la pantalla de la NES es de 256x240 píxeles. La consola usa un tamaño de sprites y tiles de 8x8.

La capacidad de sprites son 64, aunque el tamaño de los sprites se ve reducido a 8x8, en comparación al tamaño de 21x24 de la C64 y el límite de 7 sprites. Para circundar este problema, los desarrolladores usan varios sprites para agruparlos y darles un tamaño más grande.

En esta parte del proyecto, además de crear gráficos fieles para la Nintendo FAMILCOM/NES, se aplicarán estos gráficos en una ROM de la misma consola para demostrar la capacidad de edición de un producto de la NES.



Fig. 41 y 42.

Este es un concepto de un personaje pingüino creado para crear en gráficos de la NES, y para aplicarlo en la ROM. En este caso, se aplicará en la ROM de Super Mario Bros. + Duck Hunt de la NES.

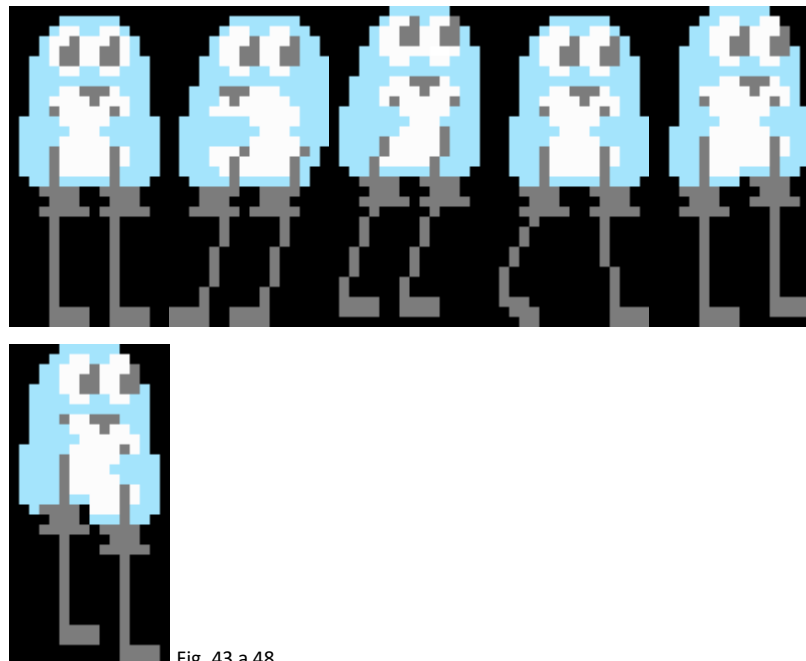


Fig. 43 a 48.

Para crear gráficos de la NES se siguen estas directrices:

1. Los gráficos se dividen en cuadros de 8x8 píxeles.
2. Si se quieren añadir gráficos más grandes, se agrupan más cuadrados de 8x8 juntos.
3. Cada bloque usa una paleta de 4 colores, del cual 1 de los 4 colores se puede usar para transparencia.

Este pingüino sustituirá a personaje de Mario cuando ingiere una seta o flor de fuego. El tamaño de super Mario (Mario alto), al igual que el de pingüino, es de 32x16 píxeles, o 4x2 bloques.

Esta imagen está dividida por una cuadrícula de 8x8 píxeles usando un máximo de 4 colores cada uno.

Para poder sustituir un personaje con otro, se utilizará un programa llamado YY-CHR.

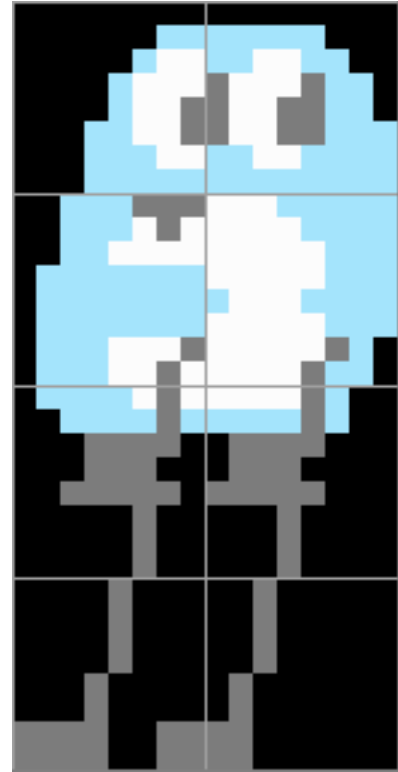


Fig. 49.

YY-CHR es un programa usado para la edición de gráficos para varios archivos ROM. En caso de este proyecto, YY-CHR se usará para la edición de gráficos en ROMs de la consola FAMILICOM/NES.

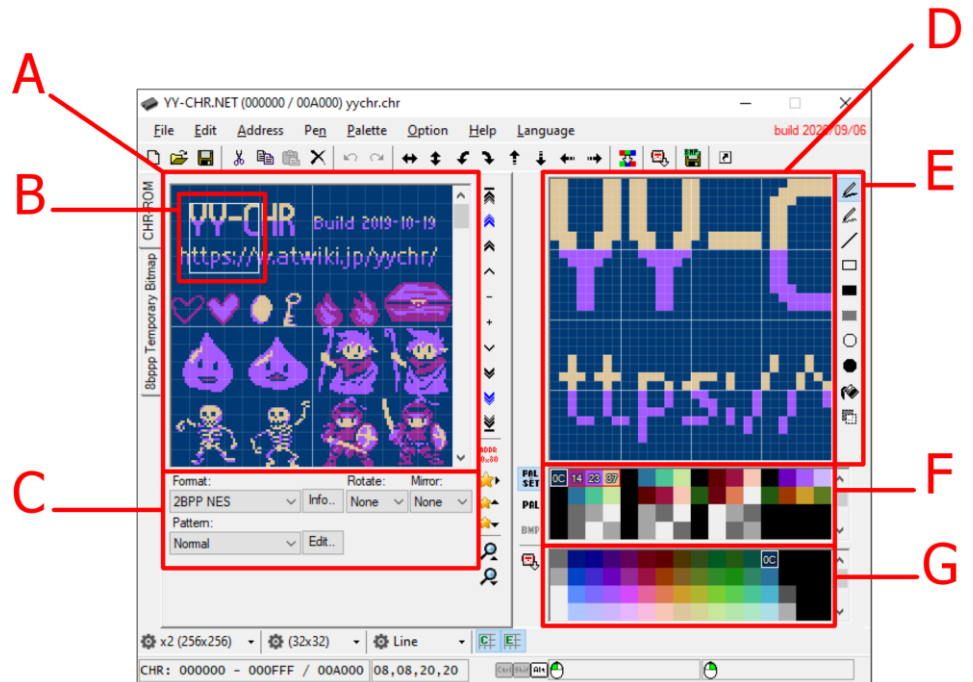


Fig. 50.

- A. Visualización del archivo.
- B. Cuadrado de selección para la edición del archivo.
- C. El espacio de formato, rotación, espejo y patrón
  - a. Opción de formato. Esta opción cambia el tipo de visualización que se puede obtener a partir del programa. Normalmente el programa reconoce el archivo que se usa y utiliza el formato adecuado, pero en caso de que no lo haga, se puede cambiar adecuadamente en esta opción.
  - b. Opción de rotación. Gira los gráficos dependiendo de la opción elegida.
  - c. Opción de espejo. Voltea los gráficos vertical u horizontalmente.
  - d. La opción de patrones. Cambia la manera en que se organiza la visualización del archivo. En ocasiones se

debe de usar esta opción para poder visualizar los archivos adecuadamente.

- D. La ventana de edición. Es la que se usa para cambiar los gráficos del archivo ROM. Con el botón izquierdo del ratón, se dibuja con el color seleccionado. Y con el color derecho, se usa la opción de cuentagotas para elegir el color seleccionado.
- E. Herramientas de dibujo, y se pueden seleccionar también los números del teclado para cada herramienta, teclado numérico incluido.
- 1) Lápiz. Es el más simple de todos, pero el que más se usa. Funciona como el lápiz de cualquier otro programa.
  - 2) Lápiz de patrón. Aplica en el gráfico un patrón de tablero de ajedrez.
  - 3) Línea recta.
  - 4) Rectángulo. Crea líneas para formar un rectángulo.
  - 5) Rectángulo con relleno. Hace lo mismo, pero rellena el interior con el mismo color.
  - 6) Rectángulo con relleno de patrón. Delimita un rectángulo, y dentro de este, lo rellena con un patrón de tablero.
  - 7) Círculo. Crea un círculo pixelado.
  - 8) Círculo relleno.
  - 9) Cubo de relleno. Rellena una parte seleccionada con un color diferente.
  - 10) Estampa. Se usa arrastrando para seleccionar una parte de los sprites y aplicarlo al seleccionar o hacer clic en la ventana.

- F. Ventana de paletas. Aquí se puede elegir qué color usar para visualizar mejor el contenido del archivo ROM. Estas paletas **no cambian** el color de los elementos dentro de juego. Su propósito es de visualización dentro del archivo. Para cambiar el color dentro del juego, se requiere otro proceso.

En el caso de la NES, escoges una paleta de 4 colores, y seleccionando el color que se quiere usar, se edita en el archivo los gráficos. Una vez terminado, cuando se abra el juego, este aplicará una paleta de color ya creada para aplicar cada gráfico, tile o sprite, su color correspondiente. También se puede cambiar de color con las teclas N y M.

- G. Paleta de colores del sistema o consola. De nuevo, estos colores sirven para visualizar los elementos dentro del archivo. Una vez creada o editada una paleta, se puede guardar la paleta en la pestaña de paleta, y se puede cargar para un futuro uso.

Sabiendo el proceso para crear gráficos de la NES, se puede usar el programa YY-CHR para implementar los gráficos. Debido a que la ventana de YY-CHR no se puede cambiar de tamaño, para trabajar se usa YY-CHR, con la referencia de gráfico que se quiere implementar al lado, la aplicación de lupa abierta y la ventana de configuración del ratón en caso de que la sensibilidad del ratón sea demasiado alta.

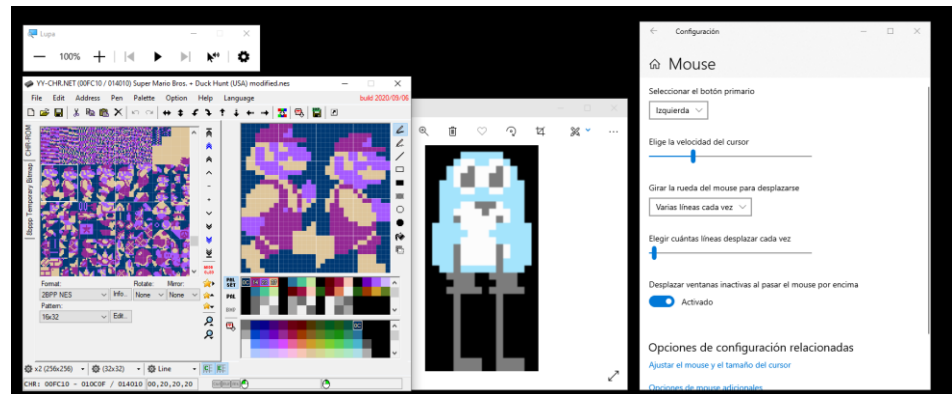


Fig. 51.

Para poder aumentar y disminuir la vista del programa con el teclado se mantiene pulsada la tecla de Windows y la tecla de suma (+) o resta (-).

Para aplicar al pingüino dentro de los límites de Mario, se ha de tener en cuenta que no todos los gráficos tienen la apariencia completa dentro del programa, y esto es debido a que esta es la memoria del juego. Debido a que el espacio de la memoria era limitado, diseñaban los gráficos con la intención de modularidad y reutilización.

Fig. 51. Vista previa del archivo de la NES y los gráficos que van a ser modificados.

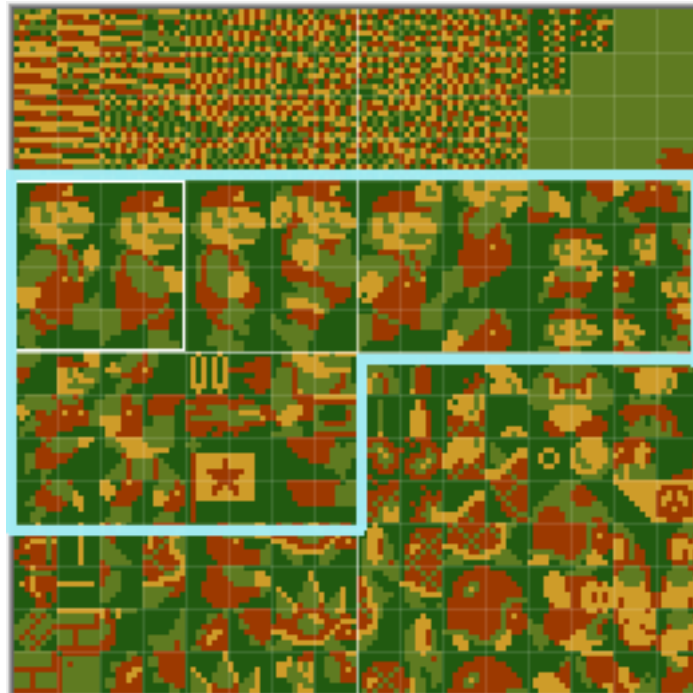


Fig. 52.

Por ejemplo, cuando Mario alto está quieto, utiliza dos sprites de cabeza de una de los sprites para correr, y otros 4 sprites para la pose de inactividad, dos de los cuales están repetidos y volteados en el otro lado.

Algunos problemas que pueden aparecer en la implementación de sprites hechos a manos es que Mario en este caso es considerado un personaje de forma humanoide. Esto implica que el juego está diseñado con esto en mente para varias acciones como nadar, lanzar bolas de fuego, trepar o deslizarse en el mástil de la bandera, etc. Así que, como demostración, los nuevos sprites de pingüino tendrán las poses de inactividad, correr, saltar y derrapar .



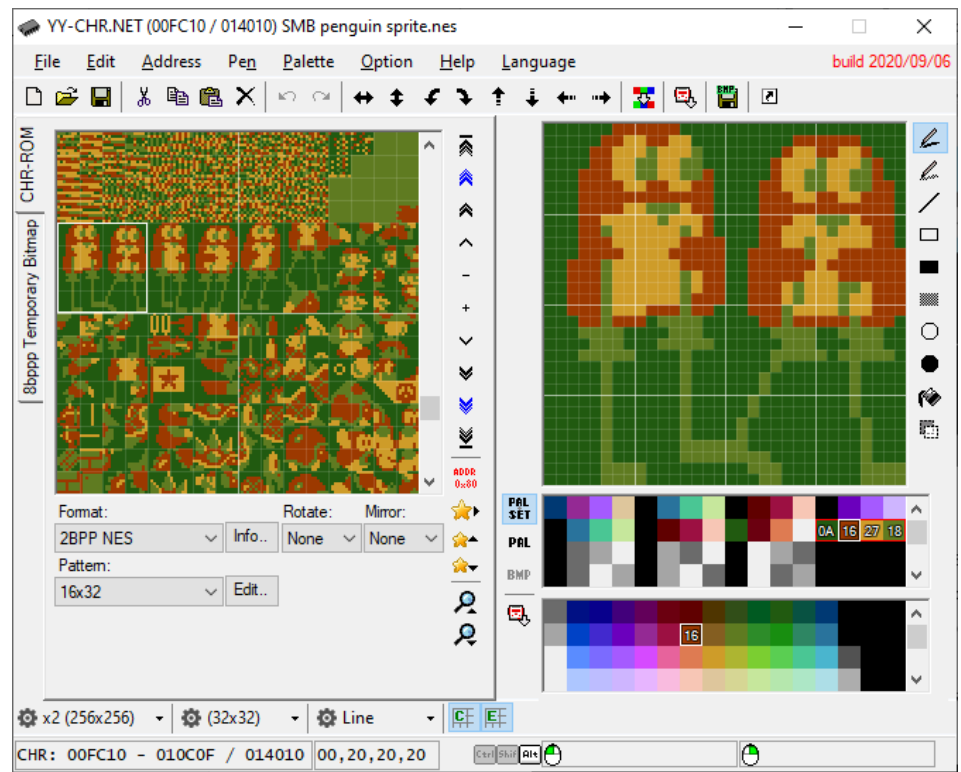


Fig. 53.

Una vez se tiene un archivo modificado, se requiere del uso de un emulador para poder comprobar los resultados. En este proyecto se utiliza un emulador llamado Nestopia. Emula tanto ROMs de NES como archivos de música NSF.

Debido a la palabra “emulador”, ha de haber una aclaración sobre el programa en sí. Estos programas utilizan la palabra emulación y sus variantes bajo el contexto de que el programa funciona con el propósito de imitar aquello que se desea. En este caso, el emulador Nestopia emula las capacidades de una consola NES.

El emulador tiene muchas configuraciones. Sin embargo, para lo que nos importa usar, que es la comprobación de los gráficos, tan solo necesitamos saber los controles, y que se debe de abrir el archivo de rom a través de la pestaña “File” (File > Open > ROM)

Los controles para el emulador con teclado son:

Flechas del teclado	Direcciones de la cruceta
Tecla mayúscula derecha	SELECT
Tecla entrar o enter	START
Coma ( , )	B
Punto ( . )	A

Una vez familiarizados con la configuración y el programa, se abre el archivo ROM, se comienza el programa usando los comandos de los controles, y se comprueba el resultado.



Fig. 54.

## 4.5. Famitracker

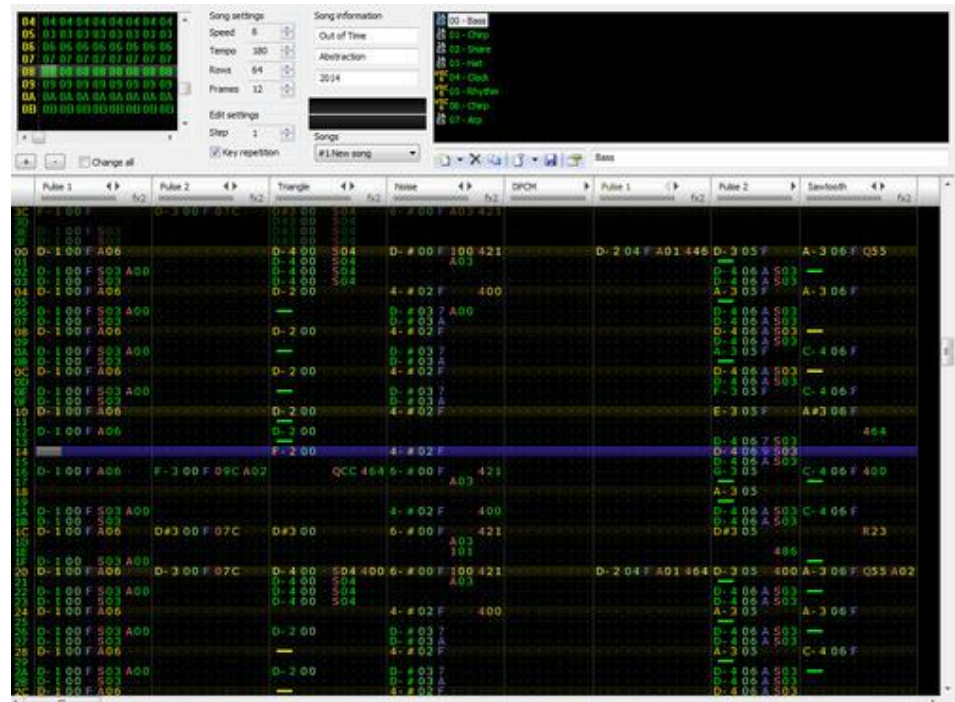


Fig. 55.

Famitracker es un programa de creación de sonidos y música especializado para la consola NES/Famicom. Este programa sigue las limitaciones de la misma consola para crear música, además de ofrecer varias opciones de añadir más canales de sonido gracias a la introducción de chips de sonido adicionales en los cartuchos de ciertos juegos (ejemplo: Castlevania 3, Japón).

En la sección de Song Settings, hay 6 opciones que determinan la velocidad de la pista musical.

Velocidad (speed ) y tempo son las opciones que determinan los BPM (Beats Per Minute, pulsaciones por minuto).

La fórmula que determina BPM es:

$$BPM = \frac{(Tempo * 6)}{Speed}$$

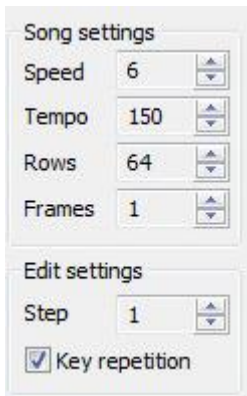


Fig. 56.

La información que explica el funciona la fórmula de BPM aparece en la documentación del programa. En la configuración estándar, los BPM son 150.

El tiempo base para las pistas musicales son de 60 o 50 tics por segundo usando la configuración NTSC/PAL estándar, que es usado para calcular el tempo medio de la canción. Esto significa que algunas filas de la pista sonarán más largas o más cortas que otras, algo que puede ser audible. Para evitar esto, se deja la configuración de tempo máximo a 150 (125 para PAL) y usar solo la velocidad speed para cambiar el bpm para asegurar que todas las filas de notas adquieran la misma duración.

El BPM máximo en el programa es de 900 en configuración NTSC, y 750 para la configuración PAL (igual al tempo 150 y 125 en velocidad 1). Ir más allá de esto no funciona correctamente.

Para evitar problemas e inconvenientes, los proyectos se trabajarán en configuración NTSC.

La opción de rows indica el número de filas de notas que adquiere una página, en este caso como frame.

La opción de frame configura la cantidad total de páginas del archivo.

La opción step es el número de filas que se salta durante la navegación de la partitura usando las flechas del teclado.

Por último, la opción "Key Repetition" cambia la forma en la que se aplica las notas cuando se pulsa la tecla. Cuando está activada, si se mantiene pulsada la tecla, se creará la misma nota en las siguientes filas hasta que se suelte. Preferiblemente, es mejor que esta opción esté desactivada para evitar la creación accidental de notas.

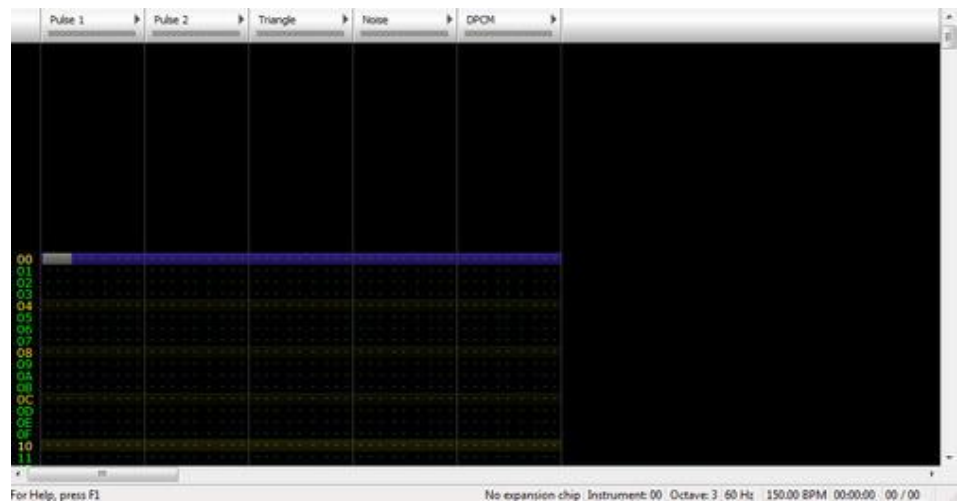


Fig. 57.

La NES/Famicom tiene un número mínimo de 5 canales de sonido. Estos canales de sonido producen cada uno un sonido por, y cada sonido puede adquirir un volumen y tono diferente de otro. Esto es porque en cada canal se recibe instrucciones del tipo de instrumento, el volumen y el tono de este.

Cada canal, sin embargo, tiene tipos distintos de sonido: Los dos primeros canales son canales de pulso, o canales de señal cuadrada. En un osciloscopio, estas señales vistas desde el estetoscopio tienen forma de cuadrado.

El tercer canal es un canal de triángulo. Su vista en el estetoscopio es triangular.

El cuarto canal es un canal de ruido. Este canal usa una onda de ruido, usando uno de dos modos tonales, con 16 tonos cada uno, ofreciendo 32 tonos diferentes. El primer modo ofrece un ruido estándar, mientras que el segundo modo le da una variación metálica, eléctrica y ligeramente más melódica. Debido a la naturaleza de este canal, se usa sobre todo para percusión.

El último canal es el de muestras de sonido. Este canal es especial, porque es el que se usa para implementar sonidos grabados fuera de la consola. Esto le da a la consola la capacidad de añadir instrumentos, o incluso voces, que el chip de sonido por sí solo no sería capaz de hacer. Sin embargo, a pesar de las ventajas, este canal no es popularmente usado durante la época de desarrollo de los videojuegos de la NES debido a la falta de memoria de los cartuchos, y del alto requerimiento de memoria para implementar una muestra de sonido. Esto se puede aliviar optimizando el sonido bajo un editor especializado, crear un sonido corto.

Estos son los 5 canales mínimos que contiene la consola, pero con la implementación de chips de sonido de cartuchos, se pueden añadir más canales. Sin embargo, aunque la consola esté limitada a 5 canales de sonido, los cartuchos de juego podían añadir configuraciones y capacidades más allá de las limitaciones de la consola. Estas configuraciones se llamaban “mapper”.

Una de las ventajas de los cartuchos creados con mapper es la inserción de chips de sonido con la capacidad de añadir más canales de sonido a la consola, aumentando el repertorio de música de estos. En caso de Famitracker, se usan los 5 mappers más usados:

- Konami VRC6
- Konami VRC7
- Nintendo FDS Sound
- Nintendo MMC5
- Namco 163

Sin embargo, para evitar complicaciones, y para comenzar simple, se usará Famitracker, con la configuración estándar 2A03.

Para crear una pista musical de chiptune se crea un proyecto nuevo, se guarda en una carpeta deseada y se crea un instrumento base para aplicar la melodía. Se puede crear un instrumento con clic derecho del ratón en la ventana de lista de instrumentos (clic derecho > Add), o con el botón “Add Instrument”.

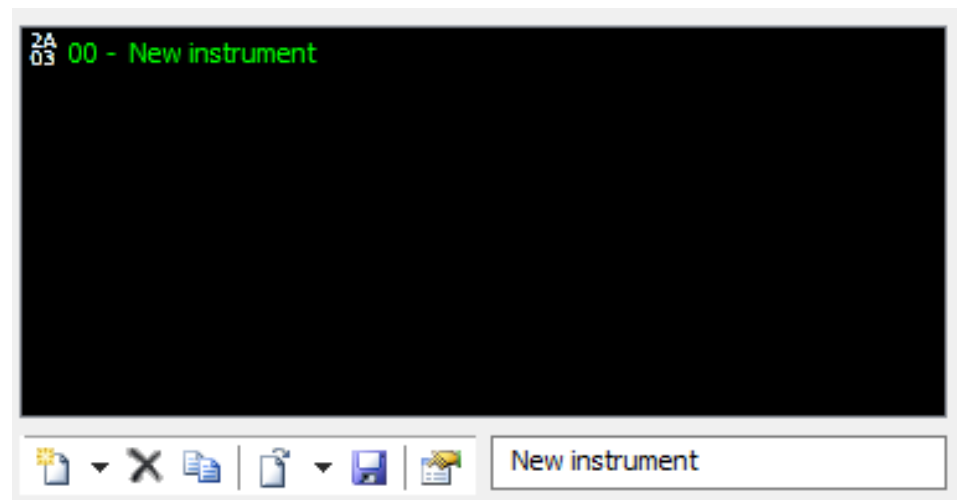


Fig 58.

Todavía no se puede crear música, aún con el instrumento creado. Aunque se graben notas con el instrumento actual, este mismo instrumento no está configurado para hacer sonido todavía. Para que pueda crear sonido, se hace doble clic en el instrumento creado, y en la zona de volumen, añadir en size una o más barras de volumen. A menos que se configure para que se repita en bucle, el instrumento repetirá el último volumen de la nota.

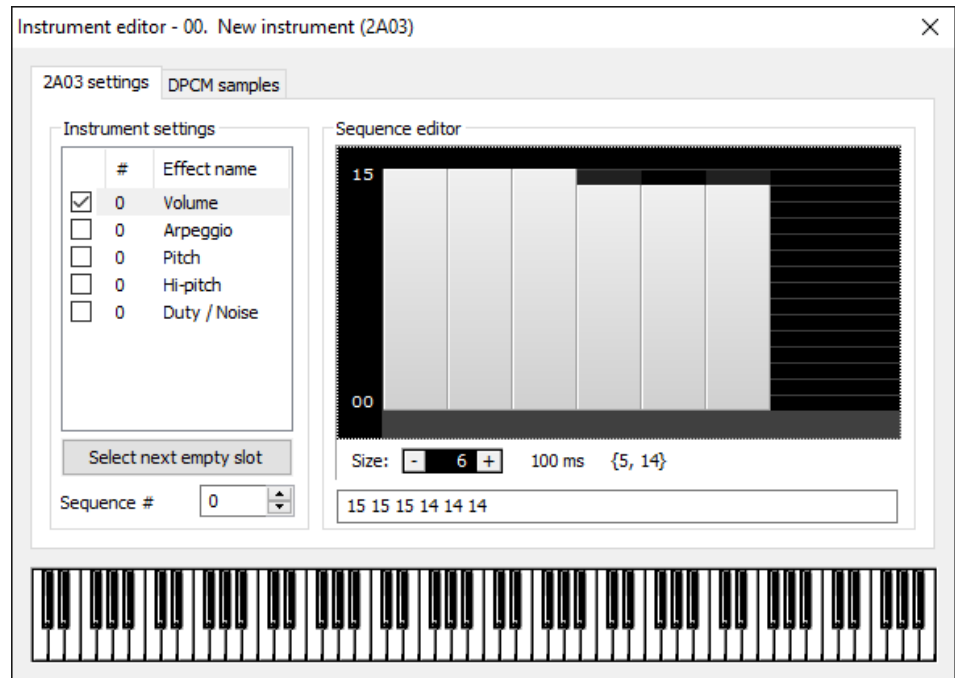


Fig 59.



Fig. 60.

Con un instrumento configurado para crear sonido, se puede comenzar a llenar la partitura, llamada “frame”, con notas. Una vez llenado un frame, se pueden crear más frames a través de la ventana de editor de frames (clic derecho > Add Frame), ofreciendo más espacio para la partitura. Se sigue trabajando hasta que se obtiene el resultado deseado.

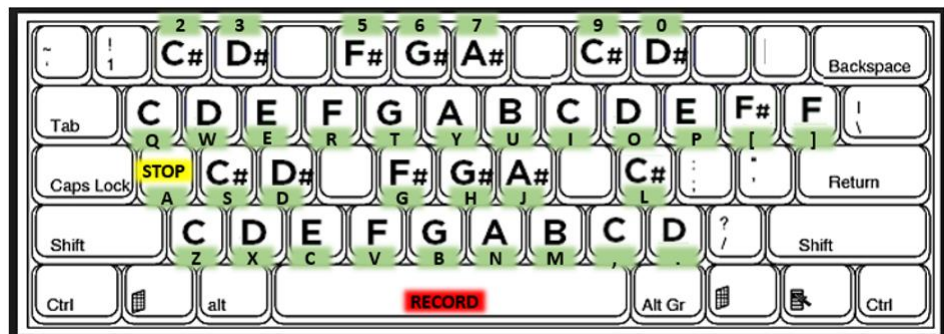


Fig 61.

Cada frame de la pista musical tiene un número específico, y este número tiene 5 números adicionales, cada uno representando cada canal de sonido. Cuando se graban notas en dichos frames, se pueden volver a reutilizar



en un futuro en caso de que se nombren los frames de canales, o del mismo frame, con el mismo número especificado por el editor.

Como la pista musical creada en este proyecto tiene partes que se repiten en ciertos instrumentos, se usan varios frames repetidos a lo largo de la canción, por lo que no hace falta copiar la misma partitura de nuevo.

El resultado de la pista musical creado para este trabajo se puede encontrar en el siguiente enlace. La pista musical usa la melodía de Rock Around the Clock de Bill Haley:

<https://carchuma.blogs.upv.es/2021/11/05/version-de-nes-famicom-de-rock-around-the-clock-de-bill-haley/>

## 5. Conclusión

Durante este proceso se ha conseguido entender de manera práctica cómo funcionan los 4 dispositivos mencionados para desarrollar directrices que se pueden usar para gráficos fieles a los dispositivos que se emulan.

Después de 1983, los productos electrónicos adquieren mayor popularidad y mayor influencia, por lo que tienen más herramientas e información para personas sin conocimiento de programación.

Lo aprendido de este proyecto no solo ayuda para crear gráficos de los dispositivos mencionados, sino que sirve como un paso más para introducirse en el campo de los romhacks, modding, creación de programas para productos antiguos, o para arte de otras generaciones de consolas u ordenadores.

El arte creado a partir de estos ordenadores y consolas fueron diseñados con un propósito útil: para crear interfaces, sonidos de aviso en los ordenadores; y crear gráficos, música y efectos de sonido para las consolas.

Y del mismo modo que dichos gráficos y sonidos fueron desarrollados con esto en mente, aunque se creen obras digitales por el hecho en sí, también se puede utilizar como una parte de una obra de arte aún mayor. Pueden tener un aspecto práctico.

E incluso con las limitaciones de distintos dispositivos electrónicos, hay productos que están en el mercado actualmente, siguiendo dichas limitaciones. Incluso si algunos de estos no están estrictamente limitados, están ciertamente inspirados por estos, siguiendo un un estilo y estética similares.

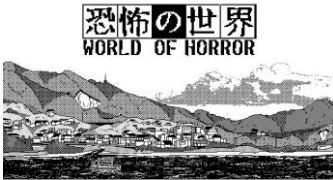


Fig. 62.



Fig. 63.



Fig. 65.

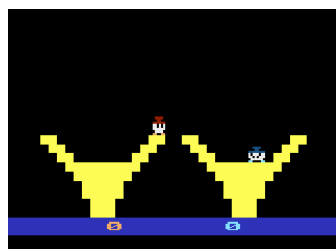


Fig. 66.

## Apple II

“World of Horror” (恐怖の世界), creado por “Panstasz” y Paweł Koźmiński, es un juego de terror que usa una paleta de 4 colores, o de 2 colores, dependiendo de cómo se configure.

“Who’s Lila”, creado por los desarrolladores “Garage Heathen”, es un juego de detectives (o de detectives inverso, según la descripción del mismo juego) que usa un estilo gráfico monocromo, con un estilo llamado “ditherpunk”.

“Return of the Obra Dinn”, creado por Lucas Pope, es un juego de detectives que usa el estilo de ditherpunk en un entorno de tres dimensiones.

Shroud House es un juego creado por “Ravancloak” y “Tophat” para un concurso de Halloween de 2011. Es un juego RPG monocromo con elementos de terror.<sup>4</sup>

Downwell es un juego que tiene una paleta de 2 bits y de 3 colores, cuyo estilo visual se centra en el negro como color predominante, el blanco para representar los personajes, con rojo para contrastar e indicar.

## Atari

La serie de juegos BIT.TRIP, creados por Choice Provisions y QubicGames, aunque tengan mayores gráficos de cualquier consola y ordenador de segunda o tercera generación, tienen definitivamente una influencia en estos, como la interfaz de estos, que tienen una apariencia similar a los juegos de Atari, o el sonido creado por estos.

Debido al tamaño pequeño de la memoria de los juegos Atari, estos eran simples por naturaleza y teniendo en cuenta de que su jugabilidad fuese repetible. Hay gente que se dedica a crear juegos caseros para consolas,

<sup>4</sup> <https://www.slimesalad.com/forum/viewtopic.php?f=4&t=8280> (consulta: 23 noviembre 2021)

llamados “homebrew”. Juegos como “ Cannonhead Clash creado por “bluswimmer” o “A Roach in Space” por Vladimir Zuñiga.

### Commodore 64

Debido a la fácil información ofrecida para programar en la Commodore 64, su código BASIC, su capacidad gráfica y su chip de sonido, la Commodore 64 sigue siendo una consola que se usa o emula para crear arte, música y programas a diario.



Fig. 67.

David Murray, también conocido en las redes sociales como “The 8-Bit Guy”, ha creado varios juegos para la Commodore 64. Entre ellos, varios juegos de la serie de juegos de estrategia llamados “Planeta X” y “Attack of the PETSCII Robots”. Este último juego es una adaptación del mismo juego bajo el mismo nombre para la Commodore PET. Usa la fuente de caracteres de la Commodore PET, y tiene colores, con la capacidad de jugarse en monocromo, o “modo PET”.



Fig. 68.

Además, la Commodore 64 tiene en su comunidad grupos que se dedica a crear programas que las capacidades de la Commodore 64 a su límite, en programación, gráficos y sonido. Estos programas se les llama demoscenes, o demos. Y es un aspecto popular de la Commodore 64.

Por otra parte, y siguiendo la misma vena que los grupos de demoscenes, la comunidad de artistas de la Commodore 64 sigue creando obras de arte y música con sus limitaciones para crear obras inspiradoras.

### NES

La NES, o Famicom, es una de las consolas retro que más popularidad tiene en el mundo. Muchas personas conocen la consola, aunque solo sea por nombre, y aquellos que han jugado en ella sienten nostalgia. Por ello, se crea muchos medios relacionados a la consola.



Fig. 69.



Fig. 70.



Fig. 71.

Shovel Knight es un juego que se inspira en los gráficos y música de la NES, con ciertas libertades gráficas. De hecho, los compositores de música Jake Kaufman y Manami Matsumae usan el mismo programa Famitracker, creandopistas musicales en formato 2A03 con VRC6.

Micro Mages es otro juego, siguiendo todas y cada una de las limitaciones de la NES. Esto es porque, a pesar de poder jugarse en ordenador, estaba apuntado también a jugarse en la consola NES a partir de un cartucho.

Y con el programa de Famitracker, hay una comunidad de compositores que, a pesar de que no creen música para juegos, hay una lista extensa de pistas musicales a partir del programa de Famitracker.

Incluso a falta de aplicación de gráficos en estos dispositivos, se han creado unas directrices y procesos claros para crear gráficos fieles a estos, y con la cultura consumidora actual, se puede usar para crear productos gráficos de fácil venta, dando una salida de trabajo independiente como freelancer.

Aunque las herramientas digitales tanto de hardware como de software tengan una obsolescencia rápida, dentro del campo de experimentación multimedia se aplica y se adquiere un nivel expresivo a con los softwares digitales creativos, debido a la necesidad de experimentar con formas de hacer y programar que nuestra curiosidad nos lleva a recordar.

## 6. Bibliografía

BENJAMINSSON, Klas. The Masters of Pixel Art, Volume 1: 5-bit to 8-bit colour depth images – Amiga & Atari & PC. Nicepixel Publications, 2016.

BENJAMINSSON, Klas. The Masters of Pixel Art, Volume 2: Commodore 64, ZXpectrum, Atari XL-XE, Amstrad CPC, Plus/4, MSX. Nicepixel Publications, 2016.

BENJAMINSSON, Klas. The Masters of Pixel Art, Volume 3: Contemporary Pixel Art. Nicepixel Publications, 2019

BANNERT, Robert, y BAUER, Christine, The SNES Pixel Book, BitMap Books, 2019.

OMAGARI, Toshi. Arcade Game Typography: The Art of Pixel Type. Thames & Hudson, 2019.

NES/Famicom: A Visual Compendium. BitMap Books, 2019.

Atari 2600/7800: A Visual Compendium. Bitmap Books. 2020

FINNIGAN, David. The New Apple II User's guide. Mac GUI. 2012.

LAPEDINO, Tim. Art of Atari. Dynamite Entertainment, 2016.

WILLIAMS, Richard. The Animator's Survival Kit: A Manual of Methods, Principles and Formulas for Classical, Computer, Games, Stop Motion and Internet Animators. Farrar, Straus and Giroux. 2012.

D'AMELIO, Joseph. Perspective Drawing Handbook. 2004.

EDWARDS, Betty. Color: A Course in Mastering the Art of Mixing Colors. TarcherPerigee, 2004.

PENTINAT, Carrión, y ANTONIO, José. Polaris: Desarrollo de un videojuego con estética Pixel Art. 2018.

ALBALA, Mitchell. Landscape Painting: Essential Concepts and Techniques for Plein Air and Studio Practice.

HUGHES, John F, et al. Computer Graphics: Principles and Practice, Addison-Wesley Professional. 2013.

3dtotal Publishing, The Ultimate Concept Art Career Guide. 2018

BOGOST, Ian y MONTFORT, Nick, Racing the Beam: The Atari Video Computer System, 2009.

## 7. Índice de Figuras

Fig. 1. Sprite de Mario de Super Mario Bros. Para la Nintendo Entertainment System. El tamaño de Mario es de 16x16 píxeles.

Fig. 2. Pantalla de consola Pong.

Fig. 3. Matthew Simons, Chapter house. Piedra caliza, 2007.

Fig. 4. Autorretrato de Da Vinci en un Grano de arroz, Andrew Rykovanov.

Fig. 5. 24 Sprites formando 3 imágenes de Simon Belmont caminando, Castlevania, Konami (1986).

Fig. 6. Icono de pestaña de la página Web de la UPV, tamaño aumentado por 4.

Fig. 7. Fitbit Sense, de Fitbit.

Fig. 8. Imagen de Wikipedia por Mausmalone de la paleta de colores de Apple. Il de baja resolución con 4 líneas de texto.

[https://commons.wikimedia.org/wiki/File:Apple\\_II\\_Low\\_Res\\_-\\_no\\_scanlines.png](https://commons.wikimedia.org/wiki/File:Apple_II_Low_Res_-_no_scanlines.png) (consulta: 23 noviembre 2021).

Fig. 9. Retrato de mascota en monocromo, 290 x 192, trabajo propio. 2020.

Fig. 10. imagen original, reducida a 280x192 px.

Fig. 11. detalle de la ventana de capas mostrando carpeta base, pasos 3 a 8.

Fig. 12. Detalle de ventana de capas mostrando pasos 9 a 14.

Fig. 13. Capa con umbral 200.

Fig. 14. Capa editada para mostrar los pliegues de la ropa.

Fig. 15. Foto sin la capa de pliegues

Fig. 16. Foto con la capa de pliegues

Fig. 17 a 23. imágenes con filtro de umbral de varios valores.

Fig. 24 . resultado final del proceso de imagen. Segundo retrato de mascota, 280x192px, trabajo propio.

Fig. 25. proceso de creación de un sprite de Atari. 8x16 y 16x16 , Trabajo propio. 2021



Fig. 26. Tablero de información de los colores por Philip Timmerman.  
<http://unusedino.de/ec64/technical/misc/vic656x/colors/> (consulta: 23 noviembre 2021).

Fig. 27. Tablero limpio de la tabla anterior con RGB y código HEX.  
<http://unusedino.de/ec64/technical/misc/vic656x/colors/> (consulta: 23 noviembre 2021).

Fig. 28. Recurso de C64-wiki.com de cuadrícula para crear sprites en alemán. Reihe es línea. Bitwert es valor de bit.  
<https://www.c64-wiki.com/wiki/Sprite> (consulta: 23 noviembre 2021).

Fig. 29. Página original del manual de instrucciones de la Commodore 64 (capítulo 6) explicando del proceso y creación de los sprites.  
[https://www.commodore.ca/manuals/c64\\_users\\_guide/c64-users\\_guide-06-sprite\\_graphics.pdf](https://www.commodore.ca/manuals/c64_users_guide/c64-users_guide-06-sprite_graphics.pdf) (consulta: 23 noviembre 2021).

Fig. 30. Página original del manual de instrucciones de la Commodore 64 (capítulo 6) con código BASIC que renderiza un sprite volando. El sprite está representado a partir de las líneas de DATA 200-240.  
[https://www.commodore.ca/manuals/c64\\_users\\_guide/c64-users\\_guide-06-sprite\\_graphics.pdf](https://www.commodore.ca/manuals/c64_users_guide/c64-users_guide-06-sprite_graphics.pdf) (consulta: 23 noviembre 2021)

Fig. 31. Página web del editor PETSCII.

Fig. 32. ventana de caracteres con editor Bitmap.

Fig. 33. Editor PETSCII, pantalla de sprites.

Fig. 34. barra vertical de Herramientas del Editor PETSCII.

Fig. 36 a 39. proceso de imagen de "La Iglesia".

Fig. 40. "La Iglesia", 384x272 px, trabajo propio. 2021.

Fig. 41 y 42. Conceptos del diseño de pingüino para editar dentro de un ROM.

Fig. 43 a 48. Gráficos del pingüino en varias poses para el juego.

Fig. 49 Gráfico del pingüino usando 8 sprites de 8x8 píxeles.

Fig. 50. Interfaz de YY-CHR.

Fig. 51. Captura de pantalla del escritorio con la ventana de YY-CHR, lupa, imagen de referencia y ventana para configurar la sensibilidad del ratón

Fig. 52. Vista previa del archivo de la NES y los gráficos que van a ser modificados.

Fig. 53. El archivo del juego ya modificado con gráficos hechos a mano.

Fig. 54. Captura de pantalla del juego modificado con el gráfico modificado de pingüino en pantalla.

Fig. 55. Ventana de Famitracker

Fig. 56. Ventana de configuración de tema.

Fig. 57. Sección de Famitracker de la ventana de partitura con los canales estándar 2A03.

Fig. 58, lista de instrumentos.

Fig. 59, editor de instrumentos.

Fig. 60, editor de frames.

Fig. 61. Imagen de la distribución de notas a lo largo de un teclado EN.

<https://github-wiki-see.page/m/nroflmao/Famitracker/wiki/Keyboard-Cheat-Sheets> (consulta: 23 noviembre 2021).

Fig. 62. World of Horror (恐怖の世界), Panstasz, 2020.

Fig. 63. Who's lila?, Garage Heathen, 2021.

Fig. 64. Return of the Obra Dinn, Lucas Pope, 2018.

Fig. 65. Bit.Trip Runner, Choice Provisions & CubiqGames, 2010.

Fig. 66. Cannonhead Clash, "bluswimmer", 2020

Fig. 67. Attack of the PETSCII Robots, David Murray, 2021.

Fig. 68. We are Demo, OFFENCE, FAIRLIGHT, NOICE, 2016.

Fig. 69. Frontier, James Svård, 2013.

Fig. 70. Shovel Knight, Yacht Games, 2014.

Fig. 71. Micro Mages, Morphicat Games, 2019.

## 8. Anexo

1. Editor de PETSCII para Commodore 64:

<http://petscii.krissz.hu/> (consulta: 23 noviembre 2021).

2. Página oficial de YY-CHR:

<https://w.atwiki.jp/yychr/> (consulta: 23 noviembre 2021).

3. Página oficial de Famitracker:

<http://famitracker.com/> (consulta: 23 noviembre 2021).

4. ROM modificado de Super Mario Bros. & Duck Hunt, trabajo parcial propio:

<https://carchuma.blogs.upv.es/2021/07/20/rom-de-super-mario-bros-duck-hunt/> (consulta: 23 noviembre 2021).

5. Adaptación NES 2A03 de “Rock Around the Clock” de Bill Haley.

Trabajo propio:

<https://carchuma.blogs.upv.es/2021/11/05/version-de-nes-famicom-de-rock-around-the-clock-de-bill-haley/> (consulta: 23 noviembre 2021).