

COMO CREAR UN DSK DE TU JUEGO PESTECERO QUE YA TIENES TERMINADO EN CINTA

Este tutorial es para pasar tu juego de Amstrad CPC que ya hayas probado en cinta y tengas acabado a formato disco, o sea DKS.

Todo el mérito hay que dárselo al gran JGN de 4mhz, que tuvo la amabilidad de explicarme paso a paso como hacerlo y poder pasar nuestro Shovel Adventure a disco para gozo de los usuarios de Amstrad disqueteros.

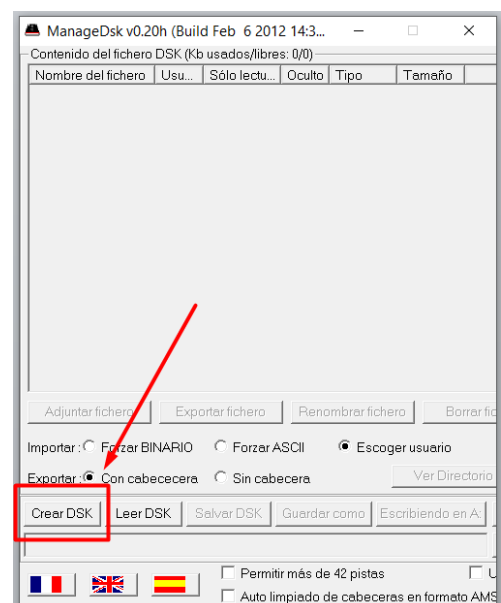
Vamos a ello.

1º CREAR EL ARCHIVO DSK

Para crear nuestro disco empezaremos por ejecutar el programa ManageDSK.exe que se incluye en este tutorial y además se puede descargar de

<https://www.cpcwiki.eu/index.php/ManageDsk>

Pulsamos en “Crear DSK” para comenzar un nuevo disco:



Ahora, con el botón “Adjuntar fichero” vamos a ir añadiendo cada uno de los archivos binarios que componen el juego más la pantalla de carga en binario SIN COMPRIMIR. Si, vamos a hacer un cargador de lo más sencillo, es decir, cargando los archivos tal y como deben quedar en la memoria, no vamos a descomprimir nada, pues en un disco hay sitio de sobra para ello ya que estamos hablando de juegos de 64K.



Al abrir cada fichero nos preguntará si es binario o ascii, seleccionamos binario y pulsamos en “Validar”.

Veremos que se van añadiendo a la lista en el cuadro blanco:



Dependiendo de tu juego tendrás más o menos archivos, en nuestro caso, eran 6 archivos: la pantalla de carga (**loading.bin**), el juego (**game.bin**), el player de música (**Arkos.bin**), los efectos de sonido (**sfx.bin**) y dos canciones que guardamos en zonas concretas de la memoria para aprovecharla al máximo (m1.bin y m3.bin):

| Contenido del fichero DSK (Kb usados/libres: 56/122) | | | | | |
|--|--------|---------------|--------|---------|--------|
| Nombre del fichero | Usu... | Sólo lectu... | Oculto | Tipo | Tamaño |
| ARKOS.BIN | 0 | | | BINARIO | 3 Ko |
| GAME.BIN | 0 | | | BINARIO | 33 Ko |
| LOADING.BIN | 0 | | | BINARIO | 17 Ko |
| M1.BIN | 0 | | | BINARIO | 1 Ko |
| M3.BIN | 0 | | | BINARIO | 1 Ko |
| SFX.BIN | 0 | | | BINARIO | 1 Ko |

Una vez hecho esto, pulsamos en “Guardar como” y salvamos nuestro fichero DSK con todo menos el cargador.

2º CREAR EL CARGADOR ASM A MEDIDA

Ahora toca un poco de edición de texto. Vamos a abrir el archivo “loader_disco_basico.asm” en un editor de código.

NOTA: Este cargador se ejecuta en la dirección 37000, dentro del SUPERBUFFER que usa MK1, ya que hay sitio de sobra ahí.

En primer lugar, vamos a la línea 100 aproximadamente, donde vamos a definir que archivos tiene que cargar el cargador (DEFINICION DE LOS NOMBRES DE LOS BINARIOS A CARGAR).

Como verás, hay 6 definiciones que corresponden a los 6 binarios que vamos a cargar en nuestro ejemplo:

```
96
97 ; DEFINICION DE LOS NOMBRES DE LOS BINARIOS A CARGAR:
98 ; CADA NOMBRE DEBE TENER 8 CARACTERES + BIN,
99 ; SI NO OCUPAN 8 CARACTERES SE RELLENAN CON ESPACIOS EN BLANCO,
100 ; COMO SE VE EN ESTE EJEMPLO (LOADING = 7 LETRAS + 1 ESPACIO + BIN)
101 PANTA:    defb "LOADING BIN"
102 JUEGO:    defb "GAME    BIN"
103 ARKOS:    defb "ARKOS   BIN"
104 SFX:      defb "SFX     BIN"
105 MUSICA01: defb "M1      BIN"
106 MUSICA02: defb "M3      BIN"
107
```

Y como se puede leer en los comentarios, hay que ponerles su nombre + tantos espacios como queden para que sean 8 caracteres en total.

Las etiquetas (PANTA:, JUEGO:, etc) pueden ser las que queramos para identificar cada binario. Ahora veremos donde se usan.

Un poco más abajo, vamos a ir viendo la carga de cada uno de los binarios. Los dos primeros son la pantalla de carga y el propio

juego, veamos como funcionan:

- Cada rutina de carga tiene una etiqueta para llamarla luego (CARGAPANTALLA: y CARGAJUEGO:)
- En la 4ª línea asignamos a hl la etiqueta del binario que vamos a cargar (PANTA y JUEGO en estos dos primeros casos).
- Luego asignamos a de y bc la dirección de memoria donde serán cargados los binarios (C000 la pantalla de carga y 400 el juego).

```
108 CARGAPANTALLA:
109     ld a,0      ;a=Unidad 0, b=Cabeza 0
110     ld b,0
111     call FDCVARS
112
113     ld hl,PANTA
114     ld de,&C000 ;Direccion de carga
115     ld bc,&C000
116     call LOADFILE
117
118     OR    A
119     JP    NZ,CARGAPANTALLA
120     ret
121
122 CARGAJUEGO:
123     ld a,0
124     ld b,0
125     call FDCVARS
126
127     ld hl,JUEGO
128     ld de,&400  ;Direccion de carga
129     ld bc,&400
130     call LOADFILE
131
132     OR    A
133     JP    NZ,CARGAJUEGO
134     ret
135
```

Habría que tener una rutina para cada binario que vamos a cargar indicando, como en los ejemplos anteriores, la etiqueta del binario y su dirección de carga.

Cuando tengamos todas listas pasaremos a ver cómo hacer las llamadas. Si usas este ejemplo de base y no tienes tantos binarios puedes borrar las que te sobren.

Ahora vete a la línea 37, dentro del programa MAIN que se ejecuta y hace las llamadas necesarias:

Por pasos, lo que hace es lo siguiente:

- 37-38: ponemos la paleta en negro
- 40-42: ponemos el modo 0 y activamos la disquetera
- 45-47: llamamos a la primera rutina de carga, la pantalla de carga y ponemos la paleta de colores que le corresponda. Luego lo vemos.
- 49-53: hacemos las llamadas para cargar el resto de binarios.
- 55: apagamos la disquetera.
- 57-61: ponemos el modo 0 y carga la paleta en negro. Esto es opcional.
- 63: ejecutamos el juego

```
32  MAIN:
33  di
34
35  call DISFIRM
36
37  ld hl,PALETANEGRO
38  call SETPALETA
39
40  ld a,0
41  call SETMODO
42  call FDCON
43
44  call CARGAPANTALLA
45
46  ld hl,PALETACARGA
47  call SETPALETA
48
49  call CARGAJUEGO
50  call CARGAARKOS
51  call CARGASFX
52  call CARGAMUSICA01
53  call CARGAMUSICA02
54
55  call FDCOFF
56
57  ld a,0
58  call SETMODO
59
60  ld hl,PALETANEGRO
61  call SETPALETA
62
63  JP    &400      ;ejecutamos el jue
64
```

Ya solo faltaría definir la paleta de colores de la pantalla de carga. Para eso vete a la línea que empiece por la etiqueta PALETACARGA:

```
192
193 ;Paletas de colores - Desde la pluma 15 a la 0 + Borde
194 PALETANEGRO:
195     db 20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20
196
197 PALETACARGA:
198     db 11, 24, 5, 7, 14, 12, 28, 22, 18, 27, 0, 6, 21, 4, 20, 3, 20
199
200
```

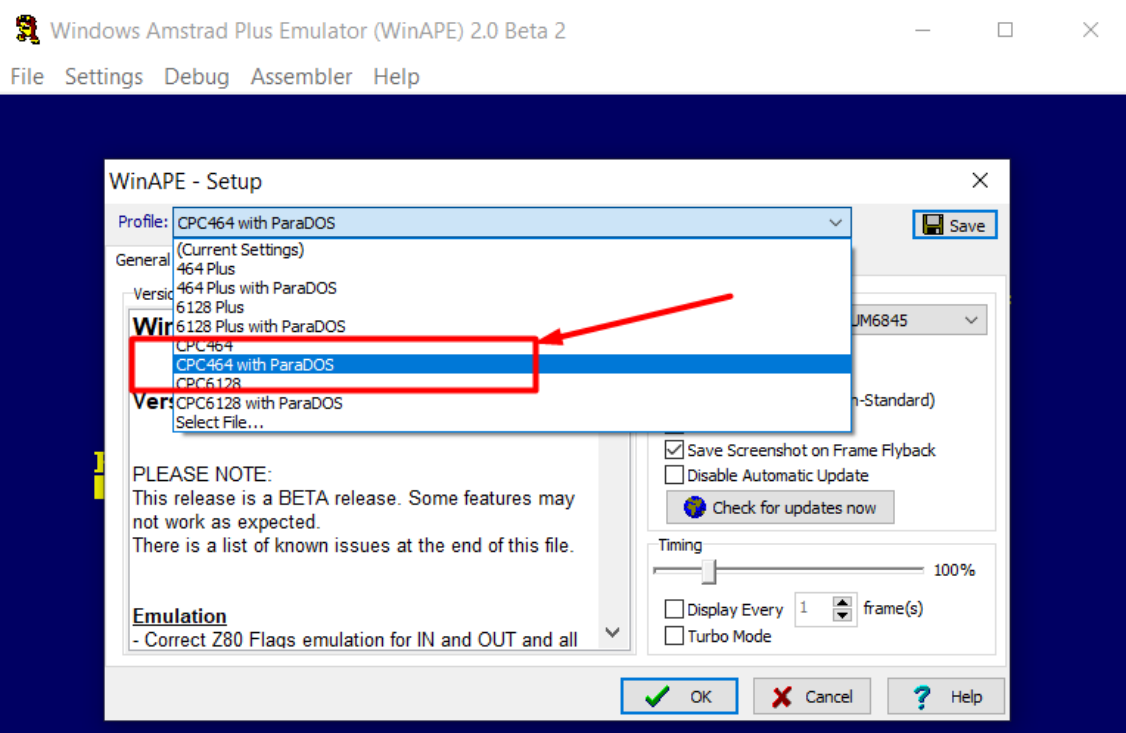
Como se explica en el comentario, debemos poner los 16 colores de la paleta, del 15 al 0, en ese orden, y luego el color del borde.

Con esto tendríamos el cargador listo, **guarda los cambios**. Ahora vamos a añadirlo al DSK.

3º AÑADIR CARGADOR AL DSK

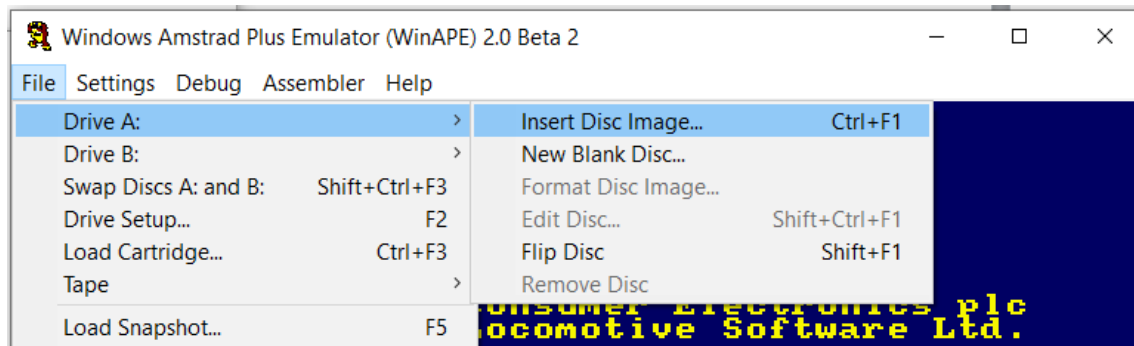
Para este último paso vamos a abrir el emulador **Winape**.

Pinchamos en “Settings” -> “General” y seleccionamos el perfil llamado “**464 with PARADOS**”. Luego pulsa OK.



Primero, vamos a cargar el DSK que hicimos con el ManageDSK en el primer paso.

Pulsamos Control+F1 (Insert disc image en la unidad A):

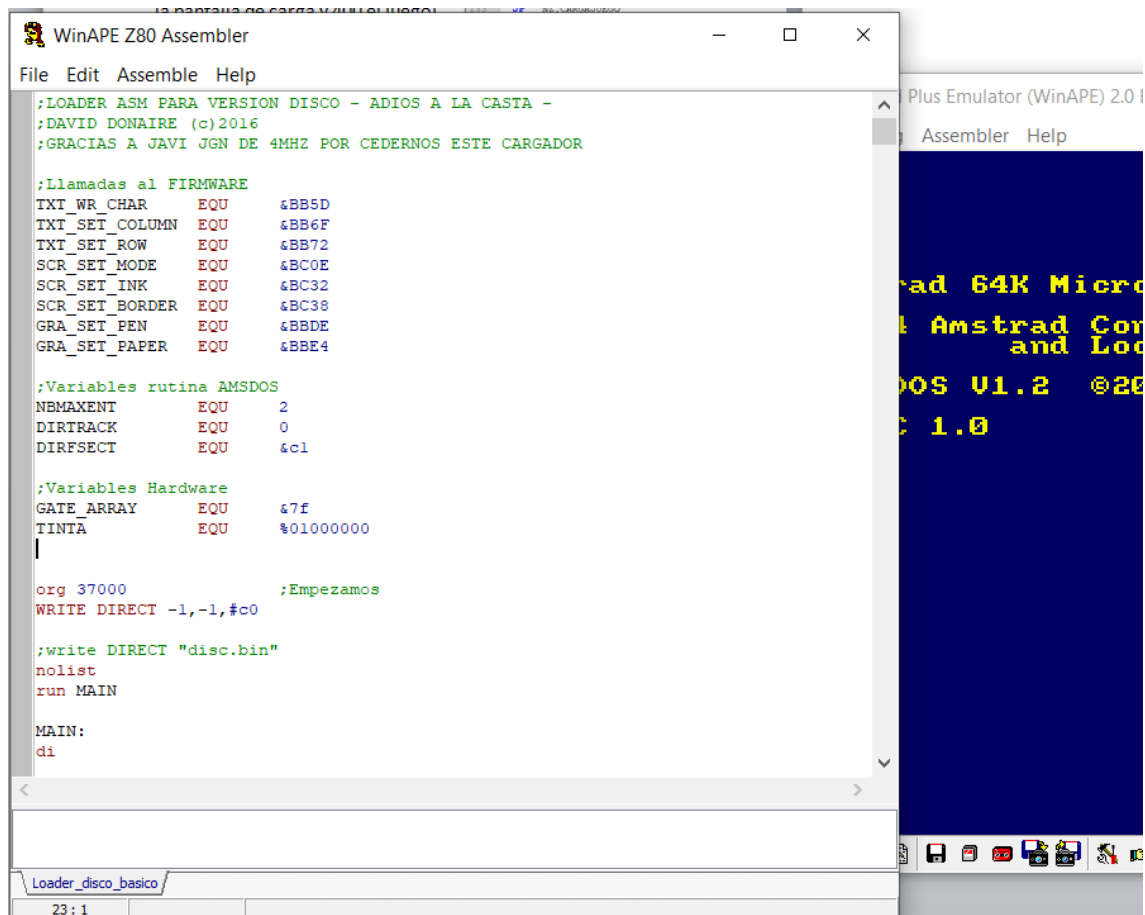


Y elegimos nuestro fichero DSK.

Ahora vamos a pulsar F3 y se nos abrirá el cuadro de Assembler.

Pulsamos “File” -> “open” y abrimos nuestro fichero asm del cargador.

Nos quedará algo como esto:



Vamos a probar si el cargador funciona. Para ello, pulsamos “Assemble” → “Run” en el menú de Assembler o bien F9.

Nos saldrá una ventanita como esta, al terminar de ensamblar pulsamos OK y el juego se empezará a cargar con nuestro cargador.

Si todo ha ido bien, el juego se ejecutará, si no, revisa el tutorial y busca en que te has podido equivocar.

Posibles fallos:

Repasa direcciones de memoria, nombre de los archivos binarios, nombres de etiquetas usadas, etc. También comprueba que hayas usado los binarios adecuados, recuerda **que no hay que usar binarios comprimidos**, ni siquiera la pantalla de carga.

Si el juego arranca y está funcionando correctamente, solo te queda añadir el cargador al disco. Vuelve a la pantalla Assembler de emulador. Y en el editor descomenta las líneas siguientes:

```
;write DIRECT "disc.bin"  
;nolist
```

Pulsa de nuevo F9 y al terminar de ensamblar, si hacemos un CAT a nuestro DSK veremos que se ha añadido un nuevo fichero llamado “disc.bin”.

Ya podemos arrancar nuestro juego con el típico **RUN”DISC**
☺☺☺☺ CHIMPÚN.