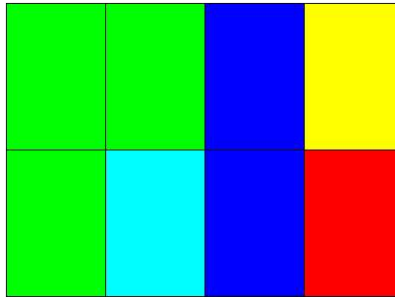


Las imágenes digitales se dividen en columnas y filas, llamadas resolución. Esta división define una gran cantidad de celdas, llamadas pixels, una forma corta de decir "elemento de la imagen." Los pixels son la unidad más pequeña de la formación de una imagen de vídeo.

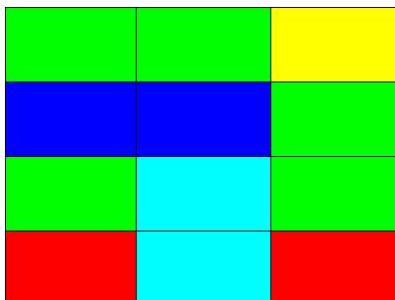
Para entender mejor cuáles son los pixels y cómo afectan a las imágenes, vamos a ver la digitalización del cuadro superior. Para numerar el cuadro arriba, debemos primero dividirlo en un sistema columnas y las filas (de estas columnas y filas definen el número de los pixels usados para representar la imagen, COLUMNAS x FILAS = PÍXELES).

Si, por ejemplo, nosotros decidiéramos dividir el cuadro anterior en 4 columnas y 2 filas, terminaríamos con una resolución de 4 x 2 que contiene 8 pixels, 4 columnas x 2 filas = 8 pixels (ver abajo).



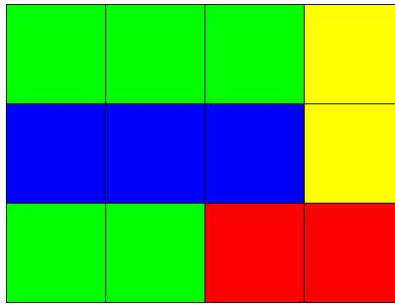
Note que mucha de la información se pierde cuando convertimos nuestro cuadro original en una imagen digital 4 x 2. De hecho, los dos cuadros apenas se parecen. Ya que un pixel puede contener solamente un color, y mucha de la información se pierde en el proceso de la digitalización si solamente utilizáramos 8 pixels. Para reducir esta pérdida, necesitaremos aumentar la resolución (número de columnas y de filas que componen la imagen) que, como puede observar, aumentara el número de pixels.

Pero antes de que continuemos, debo primero precisar algo comúnmente mal entendido sobre los pixels. Si examina el cuadro superior notará que los pixels no son cuadrados, sino que son rectangulares. Una idea comúnmente equivocada es que los pixels tienen que ser cuadrados. Pero como veremos más adelante, muchos juegos usan pixels rectangulares.



Esta es una copia digital 3 x 4 de nuestra imagen original, es algo mejor que nuestra copia 4 x 2, pero note que en esta ocasión los pixels son más anchos que altos, no como en la copia 4 x 2 donde son más altos que anchos.

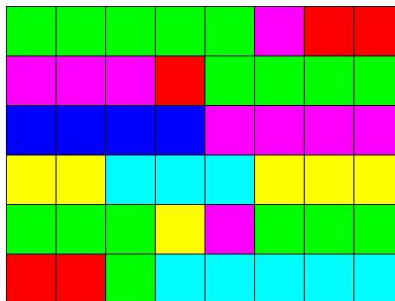
De la imagen anterior, es fácil ver que una resolución de 3 x 4 todavía no es suficiente para capturar con eficacia nuestra imagen original.



La imagen superior es una copia digital 4 x 3 de nuestra original. Compruebe las diferencias entre ellas, aunque ambas resoluciones contengan el mismo número de pixels, 12. Como veremos más adelante, la resolución nos da mucha más información, y no sólo el número total de los pixels usados. Por ejemplo, las resoluciones de 288 x 224 no son iguales que las resoluciones de 224 x 288 aunque ambas resoluciones tienen el mismo número de pixels,

$$224 \times 288 = 288 \times 224 = 64512$$

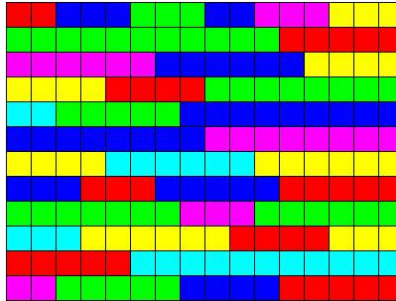
De se cuenta también que los pixels siguen siendo cuadrados en la imagen anterior. Windows, por defecto, utiliza pixels cuadrados, al igual que Linux. Los pixels cuadrados contienen importantes ventajas a los rectangulares, el más importante es el de la compatibilidad. El hecho de que los desarrolladores de un juego arcade no se pusieran de acuerdo en la forma del pixel a usar, harán muy difícil una emulación perfecta en [MAME](#). De hecho, sera la fuente de todos nuestros problemas.



Arriba se muestra una copia digital 8 x 6 de nuestra imagen original. Compruebe que aquí los pixels son también cuadrados. Esto no es un despiste. De hecho es muy fácil “romper” nuestra imagen en pixels cuadrados, si sabemos la “Relación de Aspecto” (aspect ratio) de la imagen.

Una “Relación de Aspecto” (aspect ratio) le dice cuánto es una imagen de ancha respecto a su altura. Si una imagen es cuadrada, entonces tiene una relación de 1:1. Esto significa que si divide en un número igual columnas y las filas que usted conseguirá pixels cuadrados.

Las TV, los monitores de PC, y los monitores arcade son más anchas que altas, y por lo tanto no tienen una “Relación de Aspecto” de 1:1. ¿Cuánto es de ancho? Sugiero que lo descubra por si mismo. Mida su monitor, su anchura y su altura. Con estos dos números usted puede calcular fácilmente la relación correcta de su monitor (necesitará este dato más adelante, así que le sugiero que lo calcule ahora).

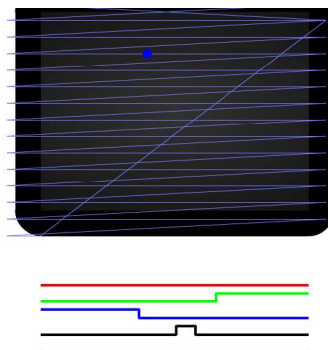


Por fin tenemos un duplicado digital perfecto de nuestra imagen original. Usando una resolución de 16 x 12 podemos capturar nuestra imagen original con 192 pixels cuadrados. Mientras que normalmente 192 pixels no podrían capturar una simple tarjeta de visita y mucho menos una fotografía; nuestra imagen al ser muy simple, se comprime bien. Resumiendo, tomamos una imagen del mundo analógico que tenía un número infinito de pixels, y la digitalizamos en 192 pixels -- que no esta mal --.

¿Qué sería ahora lo importante que tendría que observar?

No es lo bien que está comprimida la imagen, sino cómo las imágenes son representadas por pixels. Debe entender que los pixels (1) solamente muestran un color (2) pueden ser cuadrados o rectangulares y (3) son el producto de las columnas y filas de una resolución. Además, debe saber que puede determinar la forma de un pixel si sabemos el “Relación de Aspecto” (aspect ratio) de la imagen.

Ahora que comprendemos básicamente cómo el mundo digital trabaja con imágenes analógicas, necesitamos explorar un poco más lejos y conocer cómo el mundo analógico trabaja con las imágenes.



La evolución de las CRT giró alrededor de la velocidad en la cual el campo magnético horizontal podría ser manipulado. Si recuerda, el campo magnético horizontal desvía el haz electrónico de izquierda a derecha.

Cuanto más rápidamente una CRT puede manipular este campo, más rápidamente puede escanear una línea horizontal, más las líneas que puede mostrar por segundo (recuerde más líneas = una resolución más grande). Para entender mejor porqué ocurre esto, volvamos con nuestra CRT teórica (representada arriba).

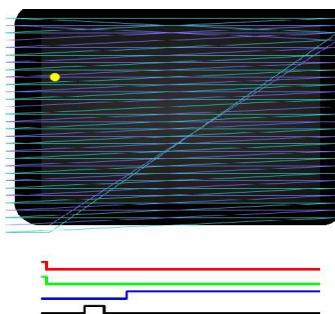
Si nuestra CRT teórica tarda 4 segundos en dibujar cada línea de datos, y dibuja 12 líneas de datos, entonces sabemos que toma exactamente 48 segundos en dibujar la pantalla completa ( $12 \text{ líneas} \times 4 \text{ segundos/línea} = 48 \text{ segundos}$ ). No obstante, notaríamos que tarda más de 48 segundos en refrescar la CRT entera. Esto es debido al espacio en blanco entre la última línea dibujada y la primera. En el mundo digital este espacio en blanco se mide en líneas. En nuestra CRT teórica hay tres líneas en blanco.

Las líneas de borrado dan a la CRT el tiempo que necesita para colocar el haz electrónico en el punto correcto. Encontrará que CRTs más viejos, como algunos monitores arcade, requieren una gran cantidad de líneas de borrado, mientras que CRTs más nuevos no necesitan tantas líneas. Esto es porque los CRTs modernos son más rápidos, y usan componentes más exactos que requieren menos “buffering”.

Como las tarjetas vídeo piensan en términos de pixels (y no de segundos) separan la muestra entre pixels activos (la resolución mostrada) y los pixels de borrado (tiempo del “buffering” para una CRT). Desde la perspectiva de nuestra tarjeta vídeo (es decir el mundo digital) hay realmente 15 líneas de datos, 12 líneas son activas y 3 líneas están en blanco (borrado). Si incluimos el tiempo de borrado vertical encontramos que tardamos 60 segundos para que nuestra CRT refresque la pantalla completamente ( $48 \text{ segundos para dibujar } 12 \text{ líneas de datos} + 12 \text{ segundos para } 3 \text{ líneas de borrado} = 60 \text{ segundos por pantalla}$ ).

Ahora suponga que deseamos mostrar más de 12 líneas de datos en nuestra CRT (deseamos aumentar su resolución). Lógicamente, podríamos deducir que nuestra CRT tardaría más en refrescar su pantalla. Por ejemplo, si deseáramos agregar 4 líneas de datos a nuestra resolución, podríamos deducir que tardaríamos 16 segundos adicionales en refrescar la pantalla completamente ( $4 \text{ líneas} \times 4 \text{ segundos/línea} = 16 \text{ segundos}$ ). De la misma manera, podríamos deducir que tomaría el doble de tiempo en refrescar la pantalla completa en nuestra CRT, si dobláramos el número de líneas que mostramos ( $24 \text{ líneas} \times 4 \text{ segundos/línea} + 24 \text{ segundos para } 6 \text{ líneas de borrado} = 120 \text{ segundos}$ ). Esta relación básica ilustra una realidad de la CRT. Si usted aumenta el número de líneas que una CRT dibuja a una velocidad dada, usted aumentará el tiempo que esta tarda en refrescar su pantalla.

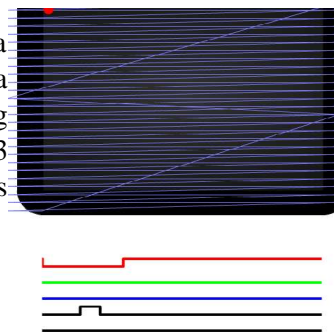
En la imagen inferior, podemos observar esta realidad. Tenemos los mismos 4 segundos por línea CRT dibujada para 24 líneas en vez de 12. Tome nota del inusual patrón que se usa para dibujar estas líneas. La mitad de las líneas se dibujan en el primer pase, y seguidamente se dibuja la otra mitad en un segundo pase. Este entrelazamiento entre las líneas pares e impares es común en las pantallas que funcionan con refrescos muy bajos. En estas ocasiones, se mejoraría el entrelazamiento y se produciría menos parpadeo.



Pero suponga que necesitamos que nuestra CRT se refresque cada 60 segundos, y que si nosotros cambiamos este tiempo nos ocurriría de todo, nuestro juego funcionaría desigual, nuestra imagen "parpadea" demasiado, o nuestra imagen es inestable o esta mutilada. Hay una otra alternativa a disminuir la velocidad de refresco (refresh rate), y es aumentar la velocidad a la cual nuestra CRT explora una línea horizontal.

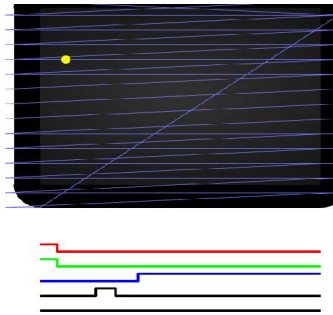
Así pues, por ejemplo, si deseáramos doblar el número de líneas en nuestra CRT a partir del 12 a 24 pero hacer que nuestra pantalla restaure cada 60 segundos, necesitaríamos doblar la velocidad a la cual nuestra CRT explora una línea horizontal. Esto nos permitiría que dibujáramos dos veces tantas líneas en la misma cantidad de tiempo, porque si solamente lleva a 2 segundos el dibujar una línea en vez de 4 segundos, entonces  $24 \text{ líneas} \times 2 \text{ segundos/línea} + 12 \text{ segundos de líneas de borrado} = 60 \text{ segundos}$ .

De manera similar, si deseáramos solamente agregar un par de líneas a nuestra resolución, por ejemplo, por ejemplo, 4 más, de modo que nuestra CRT exhibiera 16 líneas, entonces necesitaríamos aumentar la velocidad de exploración (scanning rate) de 4 segundos/línea a 3 segundos/línea ( $48 \text{ segundos}/16 \text{ líneas} = 3 \text{ segundos/línea}$ ). Esto nos daría  $16 \text{ líneas} \times 3 \text{ segundos/línea} + 12 \text{ segundos para las líneas de borrado} = 60 \text{ segundos}$ .



Ahora podemos ver que si deseamos aumentar el número de líneas de la CRT, necesitamos aumentar el tiempo que esta toma para que nuestra CRT refresque la imagen, o aumentamos la velocidad a la cual esta explora las líneas horizontales. Al revés, si deseamos exhibir menos líneas de datos sobre nuestra CRT. Podemos disminuir el tiempo que esta toma para que nuestra CRT refresque la imagen o disminuimos la velocidad a la cual explora las líneas horizontales. Es absolutamente esencial que entienda esta relación antes de que comencemos con las "modelines".

Ahora que comprendemos básicamente cómo trabajan las CRTs con las imágenes (el mundo analógico) y de cómo las tarjetas vídeo trabajan con las imágenes (el mundo digital), podemos finalmente ver cómo los dos mundos se unen para producir las imágenes que vemos.



Si usted recuerda, cuando escanea una fotografía a color en su ordenador, está tomando básicamente algo del mundo analógico y lo está traduciendo al mundo digital. Un escáner, en este caso, actúa como traductor. Toma la información a partir de un mundo y la traduce al otro. En el mundo de las representaciones gráficas, los “modelines” actúan como este traductor, a excepción que ellas toman la información del mundo digital y la traducen al mundo analógico. Un “modeline” no es nada más que un conjunto de números correlacionados. Esencialmente, es un fórmula para convertir pixels (información digital) a señales de vídeo (información analógica).

		<i>Horizontal Timing (pixels)</i>				<i>Vertical Timing (lines)</i>			
	<b>pclock</b>	<b>Active</b>	<b>SS</b>	<b>SE</b>	<b>Total</b>	<b>Active</b>	<b>SS</b>	<b>SE</b>	<b>Total</b>
16x12	5pixels/seg	16	17	18	20	12	13	14	15

En el corazón del “modeline” está el “Pixel Clock” (reloj de pixel) o el pclock para acortar, un factor de conversión que traduce los pixels (digitales) a los segundos (analógicos). Pixel clock = pixels / segundo

Podemos calcular el reloj del pixel usando el fórmula siguiente:

Pixel Clock = Number of Pixels per Line x Number of Lines x Refresh Rate

(Reloj de pixel = N° de pixels por línea x N° de líneas x Velocidad de refresco

Dese cuenta, que *el número de pixels por línea* incluye el número de los pixels activos (los que muestran datos) y el número de pixels de borrado (tiempo de “buffering” para la CRT). Semejantemente, *el número de líneas* incluye ambas líneas activas (las que muestran datos) y las líneas de borrado (tiempo de “buffering” para la CRT).

Así pues, por ejemplo, conseguimos el siguiente “Pixel Clock” para nuestro “modeline” anterior

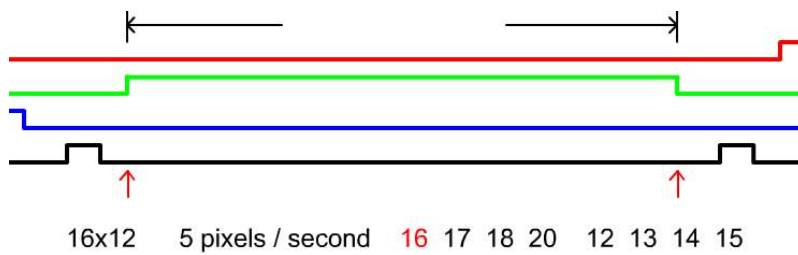
Pixel Clock = 20 Pixels per Line x 15 Lines x 1 refresh per 60 seconds = 5 pixels per second

( Reloj de pixel = 20 Pixels por Linea x 15 Lineas x 1 refresco para 60 segundos = 5 pixels por segundo.)

Para entender la utilidad de los relojes del pixel, necesitamos comprender cómo se utilizan para manipular la

sincronización de la señal. La imagen inferior es una instantánea de las señales de una VGA. Podemos observar como los pixels horizontales están divididos entre activos (los que muestran datos) y los de borrado (tiempo de “buffering” para la CRT).

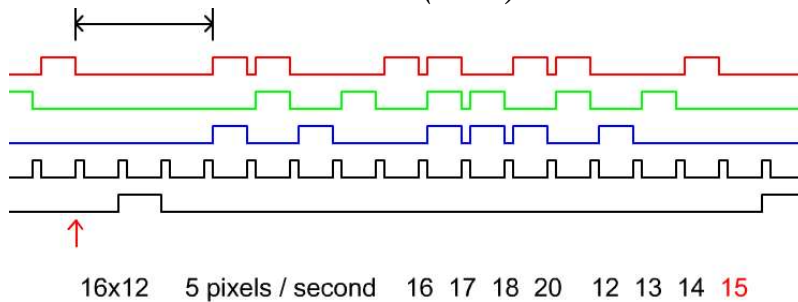
### DIGITAL (Pixels)



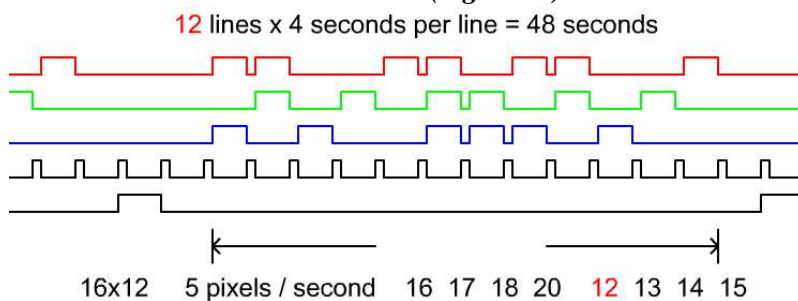
### ANALÓGICO (Segundos)



### DIGITAL (lineas)



### ANALÓGICO (segundos)



Si tiene la documentación técnica de su monitor, es bastante fácil convertir sus sincronizaciones CRT (analógicas) en “modelines”. Puesto que tengo buenas relaciones con el diseñador de nuestra CRT teórica, podría obtener la información siguiente de él.

Horizontal	Vertical	Horizontal	Vertical
Resolución: 16x12 Período De Scanline: 4 segundos Vídeo Activo: 3,2 segundos Pórtico Delantero: 0,2 segundos Sinc.: 0,2 segundos Pórtico Trasero: 0,4 segundos	Resolución: 16x12 Período Del Marco: 60 segundos Vídeo Activo: 48 segundos Pórtico Delantero: 4 segundos Sinc.: 4 segundos Pórtico Trasero: 4 segundos	Resolution: 16x12 Scanline Period: 4 seconds Active Video: 3.2 seconds Front Porch: 0.2 seconds Sync: 0.2 seconds Back Porch: 0.4 seconds	Resolution: 16x12 Frame Period: 60 seconds Active Video: 48 seconds Front Porch: 4 seconds Sync: 4 seconds Back Porch: 4 seconds

Para utilizar esta información, sin embargo, necesitamos convertirla en porcentajes, según lo demostrado abajo.

Horizontal	Vertical	Horizontal	Vertical
Período De Scanline: 100% Vídeo Activo: el 80% Pórtico Delantero: el 5% Sinc.: el 5% Pórtico Trasero: el 10%	Período Del Marco: 100% Vídeo Activo: el 80% Pórtico Delantero: 6,6% Sinc.: 6,6% Pórtico Trasero: 6,6%	Scanline Period: 100% Active Video: 80% Front Porch: 5% Sync: 5% Back Porch: 10%	Frame Period: 100% Active Video: 80% Front Porch: 6.6% Sync: 6.6% Back Porch: 6.6%

Con estos porcentajes podemos crear un “modeline” genérico, que nos permita más adelante hacer cualquier “modeline” que deseemos. En AdvanceMAME este “modeline” especial se llama “device video format”.

	<i>Horizontal Timing (pixels)</i>				<i>Vertical Timing (lines)</i>			
modeline_name	Active	Front	Sync	Back	Active	Front	Sync	Back
device_video_format	.80	.05	.05	.1	.80	.06	.06	.06

Con el “device video format” podemos convertir cualquier resolución en un “modeline” utilizable, todo lo que lo hacemos es aplicar los porcentajes a la resolución deseamos mostrar.

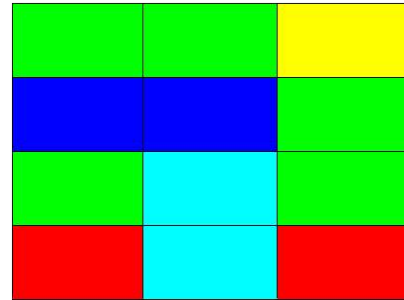
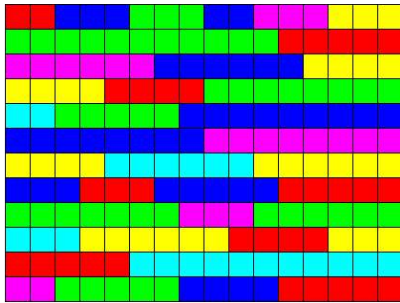
	<i>Horizontal Timing (pixels)</i>				<i>Vertical Timing (lines)</i>			
modeline_name	Active	SS	SE	Total	Active	SS	SE	Total
device_video_format	<i>0.80</i>	<i>0.05</i>	<i>0.05</i>	<i>0.1</i>	<i>0.80</i>	<i>0.06</i>	<i>0.06</i>	<i>0.06</i>
<b>16x12</b>	<b>16</b>	<b>17</b>	<b>18</b>	<b>20</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>
<b>32x24</b>	<b>32</b>	<b>34</b>	<b>36</b>	<b>40</b>	<b>24</b>	<b>26</b>	<b>28</b>	<b>30</b>

<i>Horizontal</i>	<b>16 / 0.80 = 20</b>	<b>( 0.80 + 0.05 ) x 20 = 17</b>			<b>( 0.80 + 0.05 + 0.05 ) x 20 = 18</b>			
<i>Vertical</i>	<b>12 / 0.80 = 15</b>	<b>( 0.80 + 0.06 ) x 15 = 13</b>			<b>( 0.80 + 0.06 + 0.06 ) x 15 = 14</b>			

....

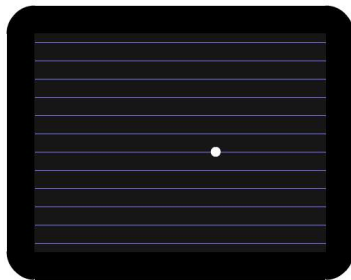


CRTs son definidos por el número de líneas que muestran. Algunos muestran solamente algunas líneas y otros muestran más. ¿Que significa todo esto en términos de calidad de imagen? Bien, si usted recuerda, necesita mostrar cierto número de pixels para mostrar razonablemente bien una imagen. Si usted utiliza demasiadas pocas líneas, simplemente pierde información. Cuando tradujimos el cuadro abajo a una resolución 4x3 encontramos que no se parecía demasiado a la original por esta misma razón.

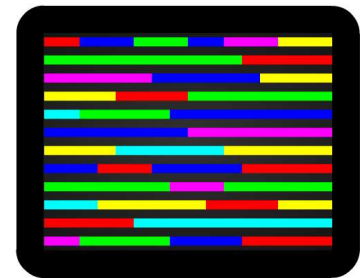


Mientras que hay algunas semejanzas entre estas dos imágenes, apenas se parecen, falta demasiada información. Si usted intenta mostrar una imagen de gran resolución en una CRT de pequeña resolución, terminaría con resultados similares a los de arriba. Normalmente, se hace un promedio del color de las columnas y las líneas juntas o simplemente se pierden. Cuanto mayor es la diferencia de tamaño de la resolución, mayor es la pérdida, y más profundo el efecto.

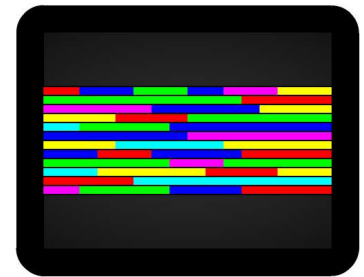
¿Pero qué sucede si usted invierte la situación, y si mostramos una imagen de pequeña resolución en una CRT de gran resolución? Esto sucede muchas veces y es donde todo se vuelve interesante, especialmente cuando usas las “Velocidades de Refresco” de tu CRT.

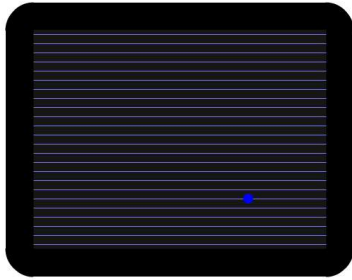


Suponga que un desarrollador de juegos (teórico, por supuesto) escribió un juego para nuestras teóricas *12 líneas por 60 segundos CRT*, y que el cuadro de la derecha representa una instantánea de este juego en acción.



Ahora veremos qué sucede cuando intentamos hacer funcionar este juego en otra CRT, una que no fue diseñado para el, por ejemplo, por ejemplo, la de la izquierda. Las anteriores *24 líneas por 60 segundos CRT* tienen la misma “Velocidad de Refresco” (1 por cada 60 segundos), pero este dibuja el doble de líneas. Si simplemente ejecutáramos nuestro juego de 16x12 en él, obtendríamos algo como el cuadro de la derecha.

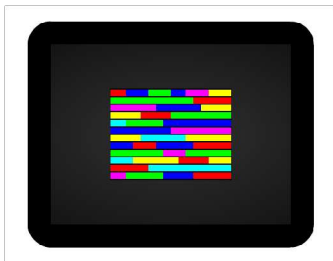




Note aquí que la “Relación de Aspecto” (aspect ratio) es incorrecta. Recuerde, que esta relación nos dice cuánto de ancha es una imagen respecto a su altura.

Si midiera su CRT, su anchura y altura, encontrarías que es 1,3 veces más ancho que de alto. Esto es porque CRTs tienen tradicionalmente una “Relación de Aspecto” de 4:3. Esto es así porque los fabricantes de CRTs le dan la medida diagonal para su monitor, ya que si usted sabe que su cociente de aspecto es 4:3 usted puede calcular sus dimensiones exactas.

El cuadro anterior tiene un cociente de aspecto de 32:12. Es 2,6 veces ancho que alto, simplificándolo tenemos 1,3. Si corregimos la “Relación de Aspecto” (aspect ratio), de la imagen anterior obtendríamos...



Ahora nuestra imagen original se muestra de forma correcta; tiene un “aspect ratio” de 4:3. Pero note que esta es solamente mitad del tamaño de nuestra pantalla. Es exactamente mitad del tamaño de la imagen original.

En la CRT que estamos viendo básicamente estamos mostrando una imagen de 16x12 en una resolución 32x24, tendríamos un duplicado perfecto, no obstante sería de diferente tamaño. ¿Pero por qué?

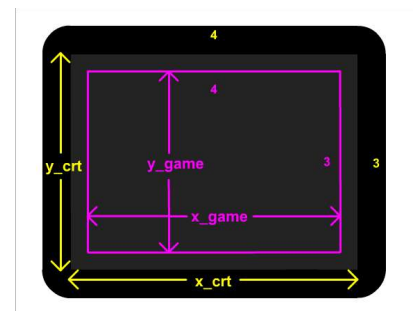
Si usted recuerda, Windows y Linux utilizan pixels cuadrados por defecto. Hay una razón muy buena de esto. Si cada uno utiliza pixels cuadrados, las imágenes nunca se distorsionaran. Técnicamente hablando, usted podría elegir cualquier forma del pixel, siempre y cuando utilice esta forma de pixel, sin ninguna distorsión. Puesto que la imagen 16x12 utiliza pixels cuadrados y una imagen de resolución 32x24 también usa pixels cuadrados, si mostramos nuestra imagen 16x12 en una resolución 32x24, no ocurre ninguna distorsión. Así pues, con esto en mente, ahora sabemos que podríamos también dibujar nuestra imagen 16x12 en una 20x15, 24x18, o una resolución 28x21, o en cualquier otra resolución que utiliza pixels cuadrados, y nunca distorsionaremos nuestra imagen. Con un poco de álgebra, podemos ver porqué esto es cierto.

Conociendo:

```
y_crt
x_game
y_game
4 : 3 = x_crt : y_crt = x_game : y_game
```

Desconocemos:

```
x_crt
```



Resuelva  $x_{crt}$  en términos de  $x_{game}$ ,  $y_{game}$ , y  $y_{crt}$ :

```
x_crt / y_crt = x_game / y_game
x_crt * y_game = y_crt * x_game Multiplicación Cruzada
x_crt = (y_crt * x_game) / y_game El resultado para x_crt
```

*De la fórmula anterior podemos ver que cuando fijamos el número de pixels mostrados por línea a 32, nuestra forma del pixel se empareja exactamente a la forma del pixel de nuestro juego, que es cuadrada (antes la forma del pixel de nuestra muestra era rectangular). Nota, que los modos ya mencionados eran calculados usando esta fórmula, según mostramos abajo.*

$$x_{crt} = (24 * 16)/12 = 32$$

$$x_{crt} = (21 * 16)/12 = 28$$

$$x_{crt} = (18 * 16)/12 = 24$$

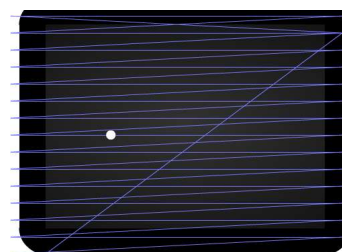
$$x_{crt} = (15 * 16)/12 = 20$$

Utilizaremos este mismo truco más adelante para mostrar correctamente juegos de 224 líneas en resoluciones de 240 líneas.

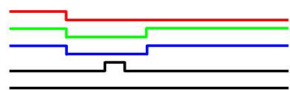
Por alguna razón, muchos juegos arcade utilizan pixels rectangulares. Esto hace una emulación perfecta más difícil, especialmente en Windows y Linux donde los pixels son cuadrados por defecto. Por lo tanto si deseamos emular perfectamente juegos, vamos a tener que hacer algo compensar esta discrepancia. La técnica más directa, y probablemente la más popular, es traducir la resolución de la imagen a una resolución mas popular, **640x480** o **800x600**. Mientras que ocasionalmente este proceso puede tener poca perdida, a menudo, no ocurre así. Si la forma del pixel mostrado es un múltiplo de la forma del pixel de la imagen, entonces teóricamente no debería de haber perdida.

Debajo de los pixels fueron estirados para caber en la pantalla. Note que cierta información se perdido. Es imposible decir cuánto de esta información se ha perdido en cada estiramiento una imagen con su ajuste en pantalla, pues depende de la forma del pixel de la imagen original y de la resolución la imagen que se está traduciendo. Además este proceso necesitara también mas demanda de proceso.

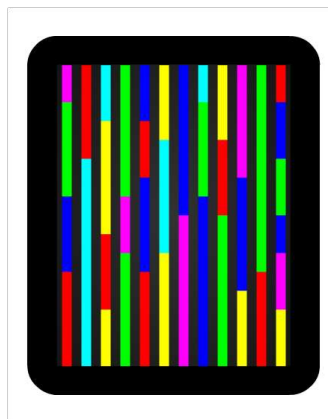
Antes de seguir adelante, usted debe ahora entender que (1) el número de líneas afecta al tamaño del scanline de la imagen (2) el estiramiento del número de columnas no tiene ningún efecto en la imagen en si (3) si la forma del pixel de la muestra difiere de la forma del pixel del aspecto de la imagen, la “Relación de Aspecto” de la imagen será incorrecta, a menos que la forma del pixel de la muestra sea un múltiplo de la forma del pixel de la imagen (**32x24** – **16x12**).



Suponga que nuestro juego teórico no fue diseñado para funcionar horizontalmente. Es decir, que fue diseñado para funcionar en un monitor montado verticalmente. Mire el cuadro abajo de la derecha que representa este juego vertical en la acción.



¿Qué sucedería si intentáramos hacer funcionar este juego en nuestra pantalla montada horizontalmente a 12 líneas por 60 segundo CRT mostrada a la izquierda?

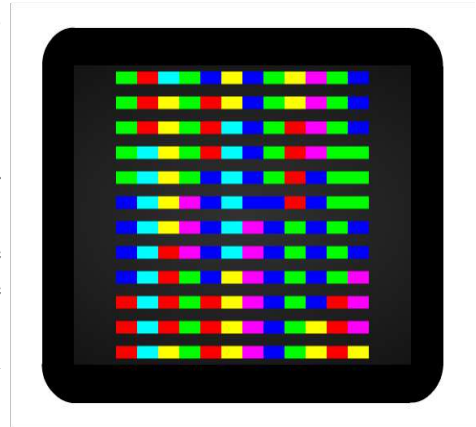


		<i>Horizontal Timing (pixels)</i>				<i>Vertical Timing (lines)</i>			
	<b>pclock</b>	<b>Active</b>	<b>SS</b>	<b>SE</b>	<b>Total</b>	<b>Active</b>	<b>SS</b>	<b>SE</b>	<b>Total</b>
16x12	5pixels/seg	16	17	18	20	12	13	14	15

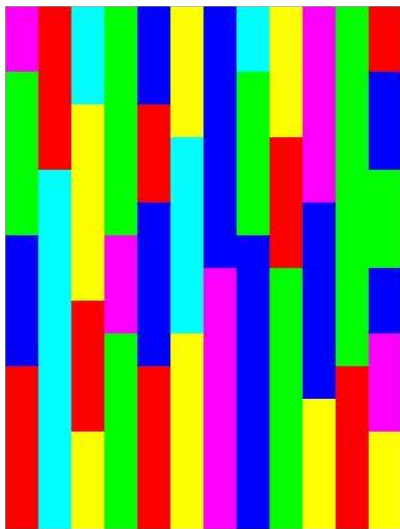
El “modeline” anterior trabaja sin embargo perfectamente en nuestra CRT si se montara verticalmente, si

intentamos utilizarla en la misma CRT montada horizontalmente, nos encontraremos con un gran numero de problemas.

Si usted compara cuidadosamente la imagen de la derecha con nuestro cuadro original (mostrado arriba), usted notará algo curioso. A saber, las dos filas superiores de pixels y las dos filas inferiores de pixels faltan de nuestro juego. ¿Qué les sucedió?



Bien, si usted intenta hacer funcionar un juego vertical en un monitor montado horizontalmente, las columnas se convierten en filas y las filas se convierten en columnas. Nuestro juego vertical 16x12 se convierte realmente en un juego 12x16. Es decir, las 16 columnas se convierten en 16 filas, y las 12 filas se convierten en 12 columnas. Puesto que nuestras teóricas *12 líneas por 60 segundos CRT* muestran solamente 12 líneas de datos y nuestro juego ahora necesita 16 líneas, nos faltan 4 líneas de datos (16 líneas - 12 líneas = 4 líneas). Por lo tanto, después de que hayamos centrado nuestro juego, encontramos que las dos filas de pixels superiores y las dos filas de pixels inferiores no están allí.



Si usted recuerda, nuestras *12 líneas por 60 segundos CRT* toman 4 segundos para dibujar cada línea de datos. Si quisiéramos que exhibiera 4 más líneas de datos, podríamos deducir lógicamente que necesitaría 16 segundos adicionales restaurar su exhibición (4 líneas x 4 segundos/linea = 16 segundos).

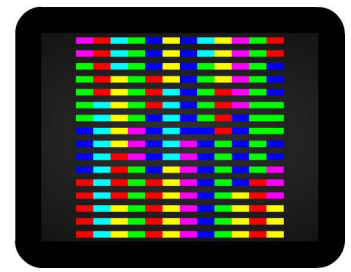
Por tanto si deseáramos exhibir las 4 líneas que falta de nuestro juego, necesitaríamos aumentar el tiempo que toma para que nuestra CRT restaure su pantalla por 16 segundos. Esto daría a nuestra CRT el tiempo que necesita exhibir las 16 líneas de datos. Aunque esto no es recomendable y hay que evitarlo para que haya una correcta velocidad y visualización del juego.

Recuerde que cuanto más rápidamente una CRT puede explorar una línea horizontal, más las líneas que puede exhibir en la misma cantidad de tiempo. Con esto en mente, si deseáramos exhibir las 4 líneas que falta de nuestro juego, y deseáramos guardar la constante la “Velocidad de Refresco” (refresh

rate), entonces necesitaríamos aumentar nuestra velocidad de refresco horizontal en la CRT.

Con un poco aritmética podemos ver que si cambiamos el índice de exploración de nuestra CRT a partir de 4 segundos/linea a 3 segundos/linea podremos exhibir las 16 líneas en los mismos 48 segundos (48 segundos/16 líneas = 3 segundos/linea) y en hacer así que se mantenga en nuestro juego el vsync correcto, como 16 líneas x 3 segundos/linea + 12 segundos de líneas que se esconden son nuestros 60 segundos. Obviamente, esto es una solución mejor que cambiar la “Velocidad de Refresco” (refresh rate) de nuestros CRT. Desafortunadamente, muchos monitores no soportan múltiples velocidades de refresco horizontales; es decir, varias velocidades para explorar una línea horizontal. Estos monitores son fijos en una frecuencia de exploración particular. Si estas en esta situación, debes entre elegir una frecuencia de imagen perfecta o un vsync correcto. Pues no hay mejor solución para estos CRT, y ya la opción sería, más bien, una preferencia personal.

Afortunadamente para nosotros, nuestras teóricas *12 líneas por 60 segundos CRT* soportan velocidades de refresco horizontales múltiples, así que podemos configurarla para exhibir 16 líneas de datos en vsync perfecto, según lo representado a la derecha.



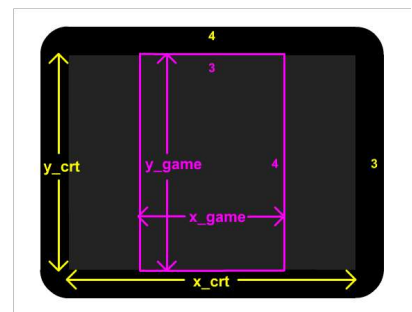
Mientras que la CRT antedicha ahora exhibe las 16 líneas de nuestro juego, puede observar como todavía algo parece incorrecto. La imagen parece "estar aplastada." En verdad es demasiado ancha. La "Relación de Aspecto" es incorrecta y necesita ser ajustada.

Sabiendo:

$$\begin{aligned} y_{crt} &= (3/4) (x_{crt}) \\ x_{game} &= (3/4) (y_{game}) \\ y_{crt} &= (y_{game}) \end{aligned}$$

Desconocemos:

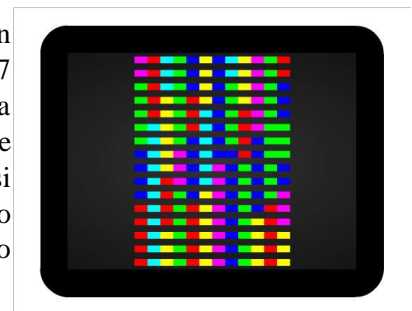
$$x_{crt}$$



Resolvemos para  $x_{crt}$  en términos de  $x_{game}$ :

$$\begin{aligned} x_{crt} &= (4/3) (y_{crt}) \text{ Definimos } x_{crt} \text{ en términos de } y_{crt} \\ x_{crt} &= (4/3) (y_{game}) \text{ Substituimos } y_{game} \text{ para } y_{crt} \\ x_{crt} &= (4/3) ((4/3) (x_{game})) \text{ Substituimos } (4/3) (x_{game}) \text{ por } y_{game} \\ x_{crt} &= (16/9) (x_{game}) = x_{game}/0.5625 = (1.77) (x_{game}) \end{aligned}$$

Con un poco de álgebra podemos ver que para mostrar un juego vertical en una CRT horizontal, necesitamos multiplicar el número de columnas por 1,77 (o dividirlo por 0,5625). Esto nos dará el número de los pixels de la muestra que necesitamos utilizar para que nuestro juego tenga la "Relación de Aspecto" correcta. Puesto que nuestro juego vertical tiene 12 columnas, si multiplicamos ese número por 1,77, conseguimos aproximadamente 21. Por lo tanto si configuramos nuestra CRT teórica para exhibir 21x16 en 1 refresco por 60 segundos, terminaremos obteniendo lo siguiente.



Ahora tenemos un duplicado sin pérdidas de nuestro juego vertical original que funciona perfectamente en nuestra CRT montada horizontalmente. Pero mientras que la imagen es un duplicado perfecto a nivel de pixel, puede notar que hay algunas diferencias sutiles entre las dos exhibiciones, a saber la dirección del "scanline". En la CRT vertical el "scanline" va de arriba a abajo y en nuestra CRT horizontal el "scanline" va de izquierda a derecha. Obviamente, hay también un cierto espacio perdido, pues los juegos verticales funcionando horizontalmente no ocupan la pantalla entera.

Los CRTs avanzaron rápidamente. Cada año se consiguieron más grandes y más rápidos. Con CRTs más rápido, nos referimos a que puede mostrar más líneas. Existen varios tipos de estándares de monitor arcade debidos a esta evolución, en el que cada uno soporta más líneas.

Se enumeran abajo.

*Resolución estándar = 262 líneas refrescadas 50 - 60 veces por segundo.*

*Resolución extendida = 312 líneas refrescadas 53 veces por segundo.*

*Resolución media = 416 líneas refrescadas 50 - 60 veces por segundo.*

Puesto que los monitores arcade son definidos por su velocidad de funcionamiento, conseguimos el siguiente cuadro.

*Resolución estándar = 262 líneas x 60 por segundo = 15720 líneas por segundo*

*Resolución extendida = 312 líneas x 53 por segundo = 16536 líneas por segundo*

*Resolución media = 416 líneas x 60 por segundo = 24960 líneas por segundo*

Y puesto que las medidas en "por segundos" se dan generalmente en hertzios (hertzio), podemos convertir los cálculos anteriores en datos más familiarmente técnicos que se enumeran abajo.

*Resolución estándar = 262 líneas x 60 hertzio = 15,7 kilociclos*

*Resolución extendida = 312 líneas x 53 hertzio = 16,5 kilociclos*

*Resolución media = 416 líneas x 60 hertzio = 25 kilociclos*

Si has leído alguna vez sobre monitores arcade, estos números le deben parecer familiares. De hecho, es absolutamente común para referirse a monitores arcade por su velocidad de reloj horizontal (horizontal clock rate, el número de líneas que una CRT dibuja por segundo). Debe también entender que un monitor arcade de 15,7 kilociclos, por definición, puede dibujar hasta 15.700 líneas por segundo y que un monitor arcade de 25 kilociclos, por definición, puede dibujar hasta 25.000 líneas por segundo (estos números son aproximaciones redondeadas).

Para emular perfectamente cada juego de arcade necesitamos entender lo siguiente.

262 líneas ( 192 + 70 activos en blanco) x 60 hertzio = 15,7 kilociclos (algunos juegos)

262 líneas ( 224 + 38 activos en blanco) x 60 hertzio = 15,7 kilociclos (la mayoría de los juegos)

262 líneas ( 240 + 22 activos en blanco) x 60 hertzio = 15,7 kilociclos (muchos juegos)

416 líneas ( 384 + 32 activos en blanco) x 60 hertzio = 25,0 kilociclos (algunos juegos)

525 líneas ( 480 + 45 activos en blanco) x 60 hertzio = 31,5 kilociclos (juegos del vector)

Los primeros 3 modos, x192, x224, y x240, funcionan en monitores arcade de 15,7 kilociclos. La mayoría de los juegos de arcade funcionan en 224 líneas, aunque los más modernos utilizan típicamente 240 líneas.

El 5º modo de abajo son, x384, que funciona en un monitor arcade de 25 kilociclos. Los monitores arcade de 25 kilociclos, nunca fueron muy populares, entre los desarrolladores de juegos arcade (probablemente debido a su coste). Tan solo un par de juegos utilizan "modelines" de 384 líneas, juegos como 720 grados. Y el modo último es, x480, que es realmente una adaptación para la emulación de juegos de vectores en tramas CRTs. Hay un par de los modos del *Weirdo* que no pude adaptar, pero por ahora nosotros haremos como si no existieran.

De los modos anteriores, el x224 y el x240 suponen la mayoría de los juegos, algo así como el 85% - 90% de todos los juegos arcade usan estos dos modos.

Si volvemos al álgebra, obtenemos las definiciones de los modos usando las fórmulas siguientes.

(Nota del Traductor: Horizontal Clock = Reloj Horizontal | Refresh Rate = Velocidad de Refresco)



Reloj-Horizontal = Número de líneas (activas + borrado) x Velocidad de Refresco  
Velocidad de Refresco (o el reloj vertical) = Reloj-Horizontal / Número de líneas (activas + borrado)  
Número de líneas (activas + borrado) = Reloj-Horizontal/Velocidad de Refresco

(Nota del Traductor: Pixel Clock = Reloj de pixel )

Reloj de pixel = Número de pixels (activos + borrados) x Numero de líneas (activas + borrados) x Velocidad de Refresco  
Reloj del pixel = Número de pixels (activos + borrados) x Reloj Horizontal

Si sabemos la Resolución y la Velocidad de Refresco de un juego arcade, entonces es bastante fácil calcular en qué clase de monitor se montó. Así pues, por ejemplo, si deseáramos saber en qué clase de monitor funcionó la serie *Mortal de Kombat*, todo lo que necesitaríamos hacer seria obtener la Resolución y la Velocidad de Refresco del juego. Esta es una tarea bastante fácil en [MAME](#), pues se da la información cuando arrancamos un juego. En este caso, es 400x254 a 53,2 hertzios.

Con esta información a disposición, hacemos:

### **TIPO DE MONITOR**

x254 a 53Hz = 85.636132315522% Lineas Activas  
262 líneas (224 activas + 38 de borrado) x 60 Hz = 15.7 KHz

(las líneas activas x Velocidad de Refresco) / Número total de líneas = % de las líneas activas

254 líneas x 53,2 hertzio = 13619,2 total de líneas activas

Puesto que la resolución más baja 25KHz es 384 líneas, podemos deducir fácilmente que el MK funciona en una CRT de 15,720 kilociclos y por lo tanto

13512,8 líneas activas/15720 líneas = 86% de líneas activas

De nuestra lista abajo podemos ver que el 85% es el modo más cercano.

192 líneas/262 total de lineas activas = 73%

224 líneas/262 total de lineas activas = 85%

240 líneas/262 total de lineas activas = 91%

384 líneas/416 total de lineas activas = 92%

Así pues, podemos ahora concluir con seguridad que la serie Mortal de Kombat fue diseñada para una CRT que funcionó del modo popular: 262 líneas ( 224 + 38 activos en blanco) x 60 hertzio = 15,7 kilociclos

Si su monitor de 15,7 kilociclos soporta varios modos entonces realmente dependerá de sus preferencias. Si usted elige x224 a 60 hertzios entonces sus juegos estarán más con la sincronización vertical. Si usted elige x240 a 60 hertzios, luego los juegos verticales (funcionando horizontalmente) se verán más agradablemente, especialmente los más viejos, como *Pacman*, *Mappy*, *Galaga*, el etc. ¿Por qué?

Bien, todos estos juegos utilizaron 288 columnas, y si usted juega un juego vertical en un monitor horizontal, entonces las columnas se convierten en filas, así que 288 columnas se convierten en 288 filas. Y si hacemos unas pocas de matemáticas

(15720 lineas / (50 - 60)Hz) x 91% lineas activas = 288 - 240 lineas

Vemos que con una configuración de 240 líneas puede hacer funcionar todos sus juegos verticales de < 288 columnas en un monitor horizontal. Sabemos, que no estarán en su correcta Velocidad de Refresco, ya que

algunos juegos estarán fuera de su sincronización vertical, pero por lo menos ninguno ellos requieran entrelazados, etc...

Si usted elige 224 líneas en 15,7 kilociclos entonces usted tiene una gama de...

$$(15720 \text{ líneas} / (50 - 60)\text{Hz}) \times 85\% \text{ líneas activas} = 268 - 224 \text{ líneas}$$

Como puedes observar nos sobran 20 líneas que difícilmente podremos solventar, pero si bajamos el Reloj Vertical hasta 47Hz entonces...

$$(15720 \text{ líneas} / (47 - 60)\text{Hz}) \times 85\% \text{ líneas activas} = 286 - 224 \text{ líneas}$$

De nuevo todos sus juegos de 288 columnas aparecen perfectos otra vez. Si usted hiciera lo mismo en una disposición de 240 líneas, obtendría un máximo de 306 líneas. Con algunos cálculos más podemos observar también

$$240 \text{ líneas} \times 60 \text{ hertzios} - 224 \text{ líneas} \times 60 \text{ hertzio} = 1 \text{ kilociclo}$$

Un rango de Reloj Horizontal de 1KHz nos permite emular perfectamente cualquier juego entre 224 - 240 líneas.

Sería complicado a base de underclocking y overclocking conseguir modos de 192 líneas, así que estos juegos se estirarán o tendrán bordes. Dese cuenta que el tamaño del borde en esta situación será grande. En una configuración de 240 líneas, sera alrededor del 20% de su pantalla ( 192 / 240 el = 80%) y en 224 líneas sera alrededor del 15% de su pantalla ( 192 / 224 el = 85%).

Para emular los esos modos en un monitor de 15,7 kilociclos, usted necesitará configurar el “Velocidad de Refresco” a la mitad 25Hz - 30Hz (es decir entrelace), el número de líneas doblara a cualquier hclock, así que si usted tiene las típicas 240 líneas configuradas entonces entrelazará usando este rango

$$2 \times (288 - 240) \text{ lines} = 576 - 480 \text{ lines}$$

Note que esto hace que los juegos vectoriales se vean agradablemente, pero los modos de 384 líneas también requerirán estirar o tener bordes (sea que el tamaño del borde es grande aquí también, alrededor 20% de los mostrados en pantalla ( 384 / 480 = 80%).

¿Y que pasa con los otros cuatro modos?

Los modos de 192 funcionaran perfectamente con el doble explorado (double scanned) o doble pixel ( 192 líneas x 2 = 384). Sin embargo, los juegos del vector requerirán sacrificar algunas líneas. Alrededor 20 líneas se perderán (whacked) ( 480 - 460 = 20). Aunque si usted baja su Velocidad de Refresco a 48 hertzios, entonces no tendrá que sacrificar ninguna de las líneas

$$(24960 \text{ líneas} / 48 \text{ Hz}) \times 92\% = 480 \text{ líneas}$$

Por la misma razón, bajar a los 48Hz todos sus juegos de 240 líneas funcionarán correctamente usando doble explorado (double scanned), aunque, la sincronización vertical estará fuera obviamente. Si usted desea que sus juegos de 240 líneas con la “Sincronía Vertical”, tendrá que estirarlos o usar un borde enorme, casi del 40% de



su pantalla ( 240 / 384 el = 62%). Los modos de 224 líneas funcionarían bien usando doble explorado a 448 líneas, pero como 240 modos de la línea, si usted desea la “Sincronía Vertical” usted tendrá que estirarlos o adquirir un borde incluso más grande, levemente sobre el 40% ( 224 / 384 el = 58%).

Hay dos formas bien diferenciadas de hacer funcionar AdvanceMAME. Puedes dejar que AdvanceMAME automáticamente te genere un “modeline” para cada juego (recomendado) o puedes crear y asignar manualmente los suyos propios. El método de la generación es el método más potente de hacer funcionar AdvanceMAME, pues es capaz de proporcionar una emulación perfecta. Si, sin embargo, usted no alcanza unos resultados satisfactorios dejando a AdvanceMAME generar sus “modelines”, puede crear y asignar el suyo propio usando la utilidad de *advv.exe*. Sin importar cómo usted planea hacer funcionar su configuración usted necesita comenzar con la utilidad de *advcfg.exe*. Este crea las entradas del device\_video\_format en su archivo de *advmame.rc* para que la utilidad *advv.exe* las utilice para generar sus “modelines”.

El “device video format” de AdvanceMAME es un modeline especial que utiliza porcentajes en vez de pixels. Estos porcentajes pueden cambiar levemente en cada rango de frecuencias de reloj, AdvanceMAME, por defecto, generará un “device video format” para su “hclock” más bajo y para el más alto. Entonces utilizará una interpolación lineal de estos dos formatos para crear “modelines” en los diferentes “hclocks” soportados. Si usted está utilizando un monitor de frecuencia fija, como un monitor arcade de 15 kilociclos, o un monitor arcade D9200, puede tener mayor suerte usando apenas un “device video format”. Solo borre la mayor y vea si esto le ayuda.

Recomiendo encarecidamente el usar uno de los modos estándares enumerados abajo para crear un “device\_video\_format”.

262 líneas (192 activas + 70 en blanco) x 60 Hz = 15.72 KHz

262 líneas (224 activas + 38 en blanco) x 60 Hz = 15.72 KHz

262 líneas (240 activas + 22 en blanco) x 60 Hz = 15.72 KHz

416 líneas (384 activas + 32 en blanco) x 60 Hz = 24.9 KHz

525 líneas (480 activas + 45 en blanco) x 60 Hz = 31.5 KHz

Si no está seguro del tipo de monitor que tiene puedes intentar descubrirlo haciendo el cálculo para saber que tipo de monitor llevaba su juego en concreto. Esta es una tarea bastante fácil si su juego se emula en [MAME](#), pues la información aparece al comienzo de la emulación. Si no siempre puede acudir a GOOGLE. Una vez que tenga la información a su disposición use la fórmula para averiguar su “Tipo de Monitor”. Si obtiene un porcentaje de líneas activas desconocidas, entonces la selección correcta será aquel que esté más cercano. Note que si configura su monitor con un modo no estándar pocos (o prácticamente ninguno) juegos tendrán una sincronización vertical correcta.

También recomiendo tener una copia de seguridad de repuesto de cualquier formato de video que cree. Esto asegurará que nunca accidentalmente borre su formato que consiguió con todo su esfuerzo. También podría agregar algo como 15.72KHz/o 24KHz/ o cualquier cosa al principio del formato, asegurándose que no usa espacios. AdvanceMAME no hará caso de su línea. Debajo tenemos un ejemplo.

x224/device\_video\_format 15720 0,744186 0,0697674 0,0697674 0,116279 0,854962 0,0725191 0,0114504 0,0610687

AdvanceMAME apenas hará caso de esta línea y la utilidad de *advcfg.exe* nunca la sobrescribirá. Es también una buena idea hacer una copia de seguridad de sus formatos en un diskette, y guardarlo en lugar seguro.

Cuando le arranque la utilidad de *advcfg.exe*, se le presentarán algunas opciones. La primera será para seleccionar un sistema de sincronizaciones de una lista de las mas populares. Recomendando encarecidamente usar de los que están proporcionados, naturalmente, el que describe lo mejor posible a su monitor. Por ahora, solo vamos a escoger una Monitor Estándar de 15,72 kilociclos (para otras configuraciones, cambie simplemente mis instrucciones). Así pues, con 15,7 kilociclos seleccionaríamos **La Resolución Estándar Arcade CGA (15KHz)** y la presionamos INTRO.

Usted ahora tiene que seleccionar sus frecuencias de monitor y de funcionamiento de la tarjeta vídeo. Primero se le mostraran un rango de “pclock”. ¿Así pues, qué debe poner aquí?

Si recuerda, los monitores son analógicos (segundos), las tarjetas vídeo son digitales (los pixels). El “pclock” (reloj de pixel) es la herramienta de la conversión que traduce segundos a pixels -- él define el número de pixels por segundo que la tarjeta de vídeo envía al monitor.

( Pixel Clock = Number of Pixels per Line x Number of Lines x Refresh Rate )

Reloj de Pixel = Numero de Pixels por Linea x Numero de Lineas x Velocidad de Refresco.

Dese cuenta que el *número de pixels por línea* incluye ambos pixels activos (los que muestran datos) y los pixels que están escondidos (los que dan al CRT el tiempo necesita para colocar su haz electrónico en el punto correcto), esto también se aplica al *número de líneas*. Con la formula anterior podremos saber que relojes de pixels (altos) son necesarios para las altas resoluciones (o los rápidos), y que los relojes de pixel (bajos) son necesarios para las resoluciones bajas (o lentas).

Los monitores arcade funcionan generalmente aproximadamente 7MHz (aunque podrían bajar a los 5MHz para las resoluciones mas bajas). Esta es la razón por la cual las configuraciones del monitor arcade requieren una tarjeta vídeo que pueda manejar “pclocks” muy bajos. De mi experiencia personal, las tarjetas de ATI y la Blade T64 de Trident pueden manejar los “pclocks” de tan solo 5 (y hasta posiblemente bajar mas, aunque no se puede realmente), la G400 de Matrox pueden ir sin problema bajo los 7 (núcleo fb o svgalib), y las tarjetas de Nvidia pueden llegar hasta los 8 en Linux usando el fb del núcleo o a 5 usando svgalib (DOS o Linux). Si corre AdvanceMAME en un monitor de 15 kilociclos debe poseer una tarjeta que soporte un “pclock” de por lo menos 7, pues cualquier cosa de 8 o más alto podría restringir la emulación. ¿Por qué es 7MHz tan especial?

Cuando configure AdvanceMAME en su monitor hace dos cosas. Primero usted configura su tarjeta vídeo para un cierto tiempo de borrado (la utilidad de *advcfg.exe*), después configura en segundo lugar su monitor para un cierto tiempo de borrado (controles del monitor arcade). Si estos dos tiempos no se emparejan, entonces la imagen no toma la pantalla completa (es overscanned / underscanned). Naturalmente, en un monitor arcade de la alta calidad tienes controles para configurar esto, especialmente en los mas modernos como el D9200 o el Polo. En monitores mas baratos, no hay tanta configuración posible (las TVs, por ejemplo). Esto significa que si usted tiene un monitor problemático que no permita mucha variación entonces usted tendrá que hacer funcionar juegos entre 5-8MHz, que es para lo que fueron diseñados estos monitores originalmente.

Las tarjetas vídeo de hoy día soportan “pclocks” muy altos, así que su límite superior no tiene ningún efecto verdadero en su configuración de AdvanceMAME. Esto significa que el factor limitador para una alta refresco

del “pclock” sería su monitor. Pero es justo decir que puede no hacer mucho caso del número más alto del “pclock” en la mayoría de las configuraciones, pues la mayoría de los monitores de calidad no pueden mostrar demasiado resoluciones altas. Así pues, recomendaría el dejar su límite superior del “pclock” a 90 a menos que esté teniendo problemas.

El límite por abajo del “pclock” es sobre todo un asunto de los fabricantes de las tarjetas video que no entienden que haya gente que intenta hacer funcionar sus tarjetas vídeo en resoluciones muy bajas. Su interés está mas en mostrar altas resoluciones, que es lo que busca la mayoría de la gente en una tarjeta moderna. Si está teniendo problemas con su tarjeta video, entonces debe intentar aumentar el extremo inferior de su “pclock” antes de aumentarlo. Si su monitor no funciona bien con un “pclock” más alto, entonces debe considerar el substituir su tarjeta video por una que soporte un “pclock” más bajo.

Para un monitor estándar arcade de 15,7 kilociclos, usted usaría;

pclock 5-90

hclock 15,72

vclock 50-60

Si fue todo bien, debe tener ahora una pantalla como la de la derecha. Si usted no consigue esto, intente aumentar el tiempo de borrado horizontal. Si eso falla, presione salga e intente cambiar su ajuste del “hclock”, hacia arriba o abajo en incrementos de 0.01 kilociclos. Si todavía no tiene ningún éxito aumente el extremo inferior de su “pclock” en incrementos de 0.5



Es absolutamente esencial que utilice su teclado y sus controles de monitor arcade correctamente aquí. Cualquier modificación del teclado hecha a esta pantalla tendrá un efecto profundo en toda su configuración, especialmente algún cambio al tamaño vertical. Si usted tiene una CRT moderna de alta calidad, después de los ajustes por defecto que deben ir bien, centre su pantalla usando sus controles del monitor arcade.

Obviamente, si usted está teniendo problemas aquí, y los ajustes de defecto entonces no están funcionando bien, necesitará ocuparse de esto de nuevo. Los cambios que se hagan sobre el reposicionado horizontal y/o vertical se pueden hacer sin problema. Todo lo que está haciendo es cambiar la localización del pulso de la sincronización, que afecta solamente al tamaño de los “pórticos” delanteros y traseros, no es ningún grande problema. No se preocupe excesivamente de hacer algunos ajustes aquí, aunque solamente hágalos cuando sean necesarios.

Los cambios realizados al tamaño horizontal afectan al activo horizontal y al tiempo del borrado de su monitor. Si desea usar una configuración de “pclock” alto para usarlo en tarjeta video actual, entonces necesita disminuir el tamaño horizontal de la resolución ahora hasta que alcance un número que su tarjeta soporte.

Si busca con cuidado el pclock más alto que podría funcionar en su configuración, debería ensanchar su pantalla tanto como sea posible con sus controles de monitor y después disminuir la resolución horizontal en la utilidad de **advcfg.exe** hasta que quepa en la pantalla. Esto es todo. Si sube mas la imagen estará “underscanned”. La diferencia entre la imagen más grande que puede conseguir y la imagen más pequeña puede conseguir que quepa en la pantalla define básicamente el tamaño del desborde (slop) en su configuración.

Si necesita volver a cambiar el tamaño vertical, intente usar un número de líneas entre 240 a 224 . Una configuración de 224 líneas no es mala. De hecho, es un ajuste bueno, si me lo permite, pues la mayoría de los

clásicos funcionarán como el original (es decir emulación perfecta). Este es un ajuste maravilloso para una recreativa clásica, pues muchas de obras clásicas de la tapa de tabla utilizan 224 líneas. La cuál, nos trae una consideración importante. Puede desear configurar su CRT para utilizar 224 líneas de todos modos. Si no esta seguro de que configuración es la mejor para usted, relea la página anterior.

Pero ¿por qué no puedo configurar mi disposición para utilizar ambos modos, 224 líneas y 240 líneas? Bien, no diré que es imposible hacerlo así, sólo que personalmente lo consideraría poco práctico. A menos que desee recentrar su pantalla usando sus controles de monitor arcade cada vez usted cambie entre juegos de 224 líneas y juegos de 240 líneas, debe entender que solamente puede configurar su monitor para un tiempo de borrado. Para emular perfectamente ambos modos sin el ajuste de su pantalla entre los juegos usando sus controles de monitor de la arcada, usted necesitaría forzar (“underclock” u “overclock”) su monitor de modo que usted tuviera una gama horizontal del reloj 1KHz (según lo explicado en la página anterior). Si usted lo configura para 240 líneas, después los juegos de 224 líneas tendrán bordes o requirieran ser estirados. Si usted lo configura para 224 líneas, después los juegos de 240 línea estarán “overscanned” (display\_adjust generate\_clocky), o la pantalla completa estará fuera del vsync (generate\_yclock del display\_adjust).

Como ultimo recurso, podría también intentar usar los ajustes para el juego que funcionaba en su monitor. Si este modo no funciona entonces algo estará mal probablemente. Si esta seguro de que su monitor esta conectado correctamente a su tarjeta vídeo, entonces sera probablemente de la tarjeta que utiliza o de la versión de AdvanceMAME. Como los controladores de las tarjetas vídeo para AdvanceMAME se van perfeccionando constantemente puede intentar usar una versión diferente de AdvanceMAME (en caso de algún fallo de driver) o intentarlo con otra tarjeta vídeo (alguna que tenga mejor soporte. Vea mis artículos sobre tarjetas para los chipsets más populares para saber como funcionan en nuestro entorno).

Recomiendo encarecidamente usar sólo 224 o 240 lineas, tanto como si quiere usar otro tipo de lineas para todos sus otros juegos, como sincronización vertical para 15.72 KHz. La única vez que puede usar una configuración diferente de lineas es si planea jugar con el reloj horizontal. Por ejemplo, Yo he configurado mi D9200 para ejecutar juegos de 230 lineas, para las cuales uso 224 lineas alrededor de 15.2 KHz y 240 lineas alrededor de los 16.3 KHz. Si su CRT no soporta un rango de 1KHz, entonces no hay razón para jugar con estos temas. Además esto podría acortar la vida de su monitor. Asegúrese que si usa este tipo de configuración, es consciente de los riesgos; ya que fácilmente podría estropear su monitor.

Una vez que ha conseguido centrar, ajustar y estabilizar la pantalla en la utilidad *advcfg.exe*, presione [ENTER]. Compruebe las resoluciones disponibles. De, a cada resolución, medio minuto o más, para asegurarse que es estable. No se preocupe si la imagen no esta centrada en este momento (\*NT.: No estoy de acuerdo con esto), solo asegúrese que las resoluciones son estables.

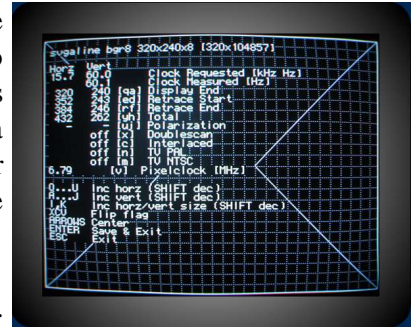
Si se toma su tiempo en perfeccionar su “device video format” usando la utilidad *advcfg.exe*, luego no solo sus juegos se verán y funcionaran mejor, sino que podrán ser fácilmente corregidos usando la utilidad *advv.exe*.

La utilidad advv.exe es un sofisticada calculadora de “modelines”. Usa su “device video format” para calcular un perfecto “modeline” para tu CRT, la cual puede ser “tweakeada” y asignada a cualquier juego. Cuida que si cambias el tamaño de tu imagen en advv.exe, lo estarás cambiando las configuraciones de reloj y lineas activas, por lo tanto solo deberías de centrar el “modeline” y no intentar cambiar su tamaño. Su modo de referencia (o modos en el caso del D9200) debería ser centrado usando solo los controles del monitor, todos los demás modos deberías ser centrados usando solo el teclado. Por ello es una buenísima idea elegir algún modo popular como referencia, como 224 lineas a 60 Hz o 240 lineas a 60Hz.

Tenga en cuenta también que muchas tarjetas ATI muestran la imagen movida hacia una dirección dependiendo de su configuración de color, por lo tanto si no cambia la opción de “display\_color auto” a “bgr16” o a “bgr32”,

muchos juegos no le aparecerán centrados. Para configuraciones ATI, centre su pantalla para el modo de referencia usando sus controles del monitor y ajustando advv.exe para que usa 16/32Bits de color. La utilidad advcfg.exe se ejecuta también por defecto a 8bits, pero puede usar otra profundidad de color añadiendo "--bit 16" o "--bit 32".

En una configuración de ATI Radeon 7500 y un monitor D9200 ejecute juegos a **320x240** a 60 Hz, **512x384** a 60 Hz y **640x480** a 60Hz. Por ejemplo pruebe *Gauntlet*, *720 Degrees* y *Asteroids*. Un D9200 puede guardar tres modos diferentes de centrado, uno para cada "hclock" estándar, la cual es una fantástica característica. Muchos monitores "Multi-sincronía" pueden guardar también muchos tipos de centrados para los variopintos "hclocks" que soportan, lo cual también es una gran ventaja.



Puede fácilmente "tweakear" su modo de referencia centrando en advv.exe. Presione F5 y ponga su modo de referencia. Por ejemplo, si lo configuro a **224** líneas debería poner

288 x **224** a 60 Hz

Para una configuración de **240** líneas debería poner

320 x **240** a 60 Hz

Busque su nuevo "modeline" y renómbrelo a algo como "referencia", luego selecciónelo y presione [ENTER]. Como este es su modo de referencia: USE SOLO SUS CONTROLES DEL MONITOR PARA CENTRAR LA PANTALLA. Para todos los demás "modelines" use su teclado, a menos, por supuesto, que su monitor soporte diferentes modos de centrado, que deberían ser centrados cada uno usando sus controles del monitor arcade. Si usas una tarjeta ATI, no olvides seleccionar la profundidad de color correcta y luego centrar su imagen. Solo ignore todas las demás profundidades de color y use solo el que haya elegido (16 o 32 bits).

Llegados a este punto, debería poder ejecutar cada juego, pero muchos estarían sin centrar, entonces debería crear un modeline para cada 16 líneas. Es una buena idea incluir los cinco modos mas populares que puede ver abajo. Dese cuenta que a menos que pueda "overclock" o "underclock" su reloj horizontal de su monitor de 15.7 KHz, solo su modo de referencia tendría un correcta "Velocidad de refresco" (suponiendo que haya forzado estos por para que no tengan bordes o los haya estrechado)

256 x **192**

256 x **224**

256 x **240**

512 x **384**

640 x **480**

Por ejemplo, si su monitor esta configurado para:

262 líneas (**224** activas + 38 borrado) x 60 Hz = 15.72 KHz

con los siguientes rangos de frecuencia

pclock **5-90**

hclock **15.72**

vclock **47-60**

Debería tener un rango de 286 - 224 lines. Por lo tanto en *advv.exe*, presione F5 y use

256 x 224 60Hz (Este seria su modo de referencia. Céntrelo con los controles del monitor)

256 x 192 60Hz (centrado solo con el teclado)

256 x 240 60Hz (centrado solo con el teclado)

256 x 256 60Hz (centrado solo con el teclado)

256 x 272 60Hz (centrado solo con el teclado)

256 x 288 60Hz (centrado solo con el teclado)

512 x 384 60Hz (centrado solo con el teclado)

640 x 480 60Hz (centrado solo con el teclado)

Si configuro su monitor para

262 lines (240 activas + 22 borrado) x 60 Hz = 15.72 KHz

con los siguientes rangos de frecuencia

pclock 5-90

hclock 15.72

vclock 47-60

Debería tener un rango de 306 - 240 lines. Por lo tanto en *advv.exe*, presione F5 y use

256 x 240 60Hz (Este seria su modo de referencia. Céntrelo con los controles del monitor)

256 x 192 60Hz (centrado solo con el teclado)

256 x 224 60Hz (centrado solo con el teclado)

256 x 256 60Hz (centrado solo con el teclado)

256 x 272 60Hz (centrado solo con el teclado)

256 x 288 60Hz (centrado solo con el teclado)

256 x 304 60Hz (centrado solo con el teclado)

512 x 384 60Hz (centrado solo con el teclado)

640 x 480 60Hz (centrado solo con el teclado)

No se preocupe por el reloj vertical que advv.exe le asigne, y por lo tanto tampoco se preocupe sobre la resolución horizontal que use, cuando debería usar 256. La idea general es crear un “modeline” por defecto para todo tipo de relojes verticales, cada 16 lineas. La resolución horizontal es irrelevante configure la para que se ajuste a la resolución del juego. Asegúrese de no modificar el tamaño de la resolución con el teclado. Solo céntrela usando los cursores. Para tarjetas ATI de nuevo asegúrese de la profundidad de color que esta usando. Una vez que guarde sus modelines (F2) en su fichero *advname.rc*, presione [ESC] para salir de *advv.exe*. Ahora debería de poder jugar a todos sus juegos sin problema.

Si la mayoría de sus juegos se muestran correctamente, puede rápidamente “tweakear” los pocos que no lo estén (lea mas abajo). Si la mayoría de los juegos no se muestran correctamente, puede forzar a AdvanceMAME a usar modelines guardados en “advname.rc”. Simplemente edite su “advname.rc” y cambie “display\_adjust generate\_yclock” a “display\_adjust xclock”.

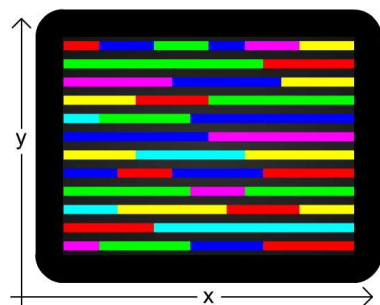
Para “tweakear” la imagen de cualquier juego en AdvanceMAME, ejecute el juego y presione la tecla [TAB]. Desde el menú seleccione “Video”. El modo usado por su juego actual se puede ver en la parte superior del menú. Para cambiarlo baje hasta la opción “mode” y presione [ENTER]. Las mejores resoluciones se muestran



en la parte superior, por lo tanto empieza por ahí y vaya bajando hasta que encuentre una resolución con la que se encuentre mas a gusto. Si ninguna de las resoluciones funciona bien necesitara crear una nueva usando **advv.exe**. De lo contrario, si encuentras un modeline correcto, presiona [ESC] para volver a la pantalla anterior y selecciona “save for this game” (grabar para este juego) o “save for this game resolution” (grabar para esta resolución del juego). Es normalmente mejor seleccionar “save for this game resolution”, ya que este ajuste valdrá para todos los otros juegos que use esa misma resolución.

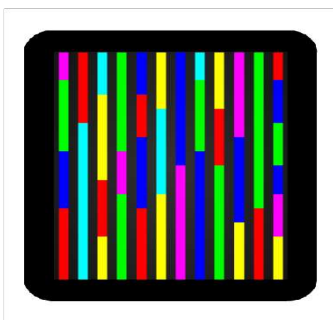
Una vez que ha configurado correctamente el tiempo borrado de su CRT usando **advcfg.exe** y sus controles del monitor, esta preparado para profundizar en la creación de “modelines” y asignaciones. Mientras AdvanceMAME hace un excelente trabajo de creación automática y asignación modos que puede usar, hay ocasiones en la que puede decidir “tweakear” el centrado de un juego, o ajustar un juego para que se ejecute mejor, piense que estos ajustes puede distorsionar la imagen final. Pero para crear y asignar “modelines”, necesitamos explorar alguna un poco mas AdvanceMAME.

La creación y la asignación de “modelines” es una cuestión pura de aritmética y mientras que AdvanceMAME sobresale en la ejecución de tales cálculos, de vez en cuando es agradable saber que puede asumir el control y configurar el mostrado de un juego.



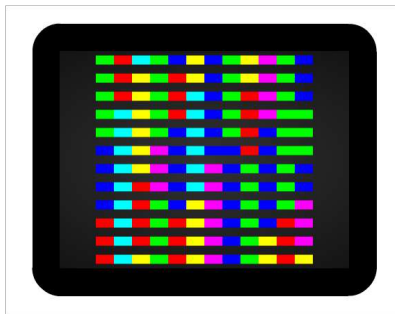
De la imagen de la izquierda, podemos ver cómo AdvanceMAME divide imágenes en columnas y filas. La notación es matemáticamente intuitiva, las columnas serian las X y las líneas se abrevia como Y. mientras no se muestre lo contrario “clock” (reloj) se usa para representar una “Velocidad de Refresco” (refresh rate). Con esta información a nuestra disposición podemos rápidamente pensar a nuestra manera muchos de los parámetros del “display\_adjust” de AdvanceMAME. Pero para hacerlo, vamos volver a nuestro problema de exhibir un juego vertical en un monitor horizontal.

Recordará, que si tomamos un juego vertical y lo exhibimos en una CRT montada horizontalmente, nos encontrábamos con muchos problemas. Y a menos que cambiemos el reloj horizontal o el reloj vertical, nos faltarían algunas líneas. Además, a menos que ajustemos el número de las columnas usadas por el modo, la “Relación de Aspecto” (aspect ratio) de la imagen sera incorrecta. Véalo en la imagen siguiente.



		<i>Horizontal Timing (pixels)</i>				<i>Vertical Timing (lines)</i>			
	pclock	Active	SS	SE	Total	Active	SS	SE	Total

		<i>Horizontal Timing (pixels)</i>				<i>Vertical Timing (lines)</i>			
16x12	5pixels/sec	16	17	18	20	12	13	14	15



		<i>Horizontal Timing (pixels)</i>				<i>Vertical Timing (lines)</i>			
	<b>pclock</b>	<b>Active</b>	<b>SS</b>	<b>SE</b>	<b>Total</b>	<b>Active</b>	<b>SS</b>	<b>SE</b>	<b>Total</b>
16x12	5pixels/sec	16	17	18	20	12	13	14	15

Mientras que antes nosotros asumimos un teórico CRT con soporte múltiple de rangos de escaneo horizontal, ahora asumiremos que no los hay, y que la única frecuencia que el CRT admite es una velocidad de escaneo (scan speed) de 4segundos/linea. Si esto fuera así, desafortunadamente, requerirá de ciertos sacrificios para mostrar nuestros juegos. (En los monitores tradicionales arcade, este escenario es la norma. Debes a menudo decidir entre una imagen perfecta o tener una correcta sincronización vertical o perder parte imagen vertical. Esta sería su decisión y no hay ninguna mejor solución.)

Afortunadamente para nosotros, AdvanceMAME asume estos problemas y nos hace muy fácil movernos en la maraña de opciones para buscar nuestra solución favorita. Y lo mas importante, nosotros no tenemos que usar una solución genérica para todos nuestros juegos. Podemos decidir fácilmente estrechar un juego, usar bordes en otro, cambiar el “vsync” en otro, o perder algunas pocas líneas...

Traducción de David Skywalker (A ka D\_Skywalk)  
 Para Marcianitos.net  
 Vea la versión original en  
 Easy MAMEcab: [Arcade monitor modeline howto](#)