

# ARES Slideshow

## Manual de instrucciones V1.3

---

ARES Video ofrece la posibilidad de utilizar presentaciones de diapositivas "inteligentes" de creación propia para el entrenamiento de tiro con láser. En ellas no sólo pueden mostrarse diferentes imágenes, sino que la transición entre ellas puede definirse de muchas maneras e incluso diseñarse de forma interactiva. Esto significa que las presentaciones pueden reaccionar a los disparos. Los pases están escritos en un lenguaje de script sencillo, que no requiere conocimientos de programación, pero que también ofrece la posibilidad de crear secuencias complejas. Para facilitar la creación de los pases de diapositivas, también hay un editor que, entre otras cosas, permite reproducir los pases de diapositivas paso a paso. Además, ofrece la posibilidad de resaltar en color las palabras clave y, si es necesario, muestra mensajes de error durante la ejecución.

### ESTRUCTURA BÁSICA Y FUNCIONAMIENTO DE LOS PASES DE DIAPOSITIVAS

Los archivos de presentación de diapositivas tienen la extensión ".dia". Son simples archivos de texto que pueden abrirse con cualquier editor de texto. Durante la reproducción, el archivo se procesa línea por línea. Las líneas vacías están permitidas y simplemente se omiten. También es posible escribir comentarios. Éstos comienzan con un símbolo de almohadilla (#) que garantiza que se ignore todo el texto hasta el final de la línea correspondiente. Cada línea contiene exactamente un comando, que puede ser de cualquier longitud. Por tanto, el final de la línea marca también el final de un comando. No es necesario un carácter explícito, como es habitual en otros lenguajes de programación. La presentación se detiene automáticamente cuando se ha procesado la última línea. También puede finalizarse en cualquier momento pulsando la tecla escape.

### ELEMENTOS BÁSICOS DE LOS SCRIPTS: COMANDOS, VARIABLES Y LÓGICA

Las capacidades de los scripts de presentación de diapositivas pueden dividirse a grandes rasgos en tres áreas. Hay comandos, variables y lógica.

Los comandos son los que determinan el aspecto real de la presentación de diapositivas. Pueden utilizarse para mostrar imágenes, emitir texto, reproducir sonidos o definir tiempos de espera.

LAS VARIABLES SE utilizan para almacenar valores y calcular con ellos o tomar decisiones en función de los valores. Puede definir y utilizar cualquier número de variables propias. El propio sistema también proporciona mucha información sobre las variables. Por ejemplo, la posición de acierto del último tiro, el tamaño de varios elementos o el tiempo transcurrido desde el inicio.

LA LÓGICA es la parte que determina el flujo de las presentaciones y, por tanto, aporta la "inteligencia". Existen ramas, bucles, saltos y los enlaces lógicos AND y OR. Esto permite recorrer distintas partes de la presentación en función de valores variables.

### RESUMEN DE LOS COMANDOS

Todos los comandos deben colocarse al principio de una línea, ya que provocan una acción pero no pueden ser procesados por otros comandos. La única excepción es el comando RAND, que puede utilizarse dondequiera que pueda colocarse una variable o un número.

Comando	Función
<b>OPEN</b> "%Archivo%"	<p>Abre un archivo de imagen y lo muestra en la pantalla. El archivo debe estar en la misma carpeta que la presentación. Sólo se admiten los siguientes formatos (jpg, png, bmp). El nombre del archivo se escribe entre comillas. El nombre del fichero también puede generarse a partir de una combinación de texto y variables.</p> <p>Ejemplo:  <b>OPEN "Imagen" + \$Var + ".jpg"</b>  Si la variable \$Var tiene el valor 1, se abre el archivo de imagen "Imagen1.jpg", si tiene el valor 2, se abre el archivo "Imagen2.jpg".</p>
<b>OPEN_SCORE</b> "%Archivo%"	<p>Sin embargo, de forma similar a OPEN, el archivo de imagen no se muestra, sino que se abre en segundo plano como una "plantilla de puntos". Lo ideal es que la plantilla tenga el mismo tamaño que la imagen visualizada. Tras un disparo, el color de la plantilla de puntos está disponible en el punto de impacto a través de 3 variables. Por lo tanto, se pueden definir zonas de impacto, pero no tienen que ser visibles para el tirador.</p>
<b>OPEN_DECAL</b> "%Archivo%"	<p>Selecciona el archivo para el objeto calcomanía. Se trata de una imagen que puede mostrarse sobre la presentación de diapositivas, cuya posición y tamaño pueden ajustarse libremente mediante variables. También se pueden realizar movimientos cambiando continuamente la posición.</p>
<b>PRINT</b> "%Text%"	<p>Establece el texto del objeto de texto en el texto especificado. El objeto de texto puede utilizarse, por ejemplo, para mostrar mensajes, pero también puntuaciones y tiempos. Al igual que con los comandos ABRIR, el texto puede estar compuesto por una combinación de fragmentos de texto y valores.</p> <p>Ejemplo  <b>PRINT "Tiempo: " + \$RUNTIME + " segundos".</b>  Establece el texto a "Tiempo: 31.415 segundos" donde el número es el valor actual de la variable \$RUNTIME, que contiene el tiempo de ejecución de la presentación hasta el momento.</p>
<b>SOUND</b> "%Archivo.wav%"	<p>Reproduce un archivo de sonido en formato wav. Este comando no debe utilizarse demasiado seguido, sino esperar a que se haya reproducido el sonido anterior antes de reproducir el siguiente.</p>
<b>ESPERA</b> %TIEMPO	<p>Espera el tiempo especificado en segundos antes de ejecutar la siguiente línea.  Ejemplo:  <b>ESPERA 1,5</b></p>
<b>RAND</b> %Número%	<p>Genera un número aleatorio entre 0 y el número introducido menos 1. Sólo se generan números enteros sin parte de coma. Si desea números decimales, deberá dividir el resultado según corresponda.  Ejemplo:  <b>RAND 5</b>  Devuelve un número aleatorio que puede ser 0,1,2,3 o 4.  <b>(EDGE 100) /10</b>  Devuelve un número aleatorio entre 0,0 y 9,9</p>

### VARIABLES Y OPERACIONES ARITMÉTICAS

Una variable tiene un nombre y se le puede asignar un valor. El nombre de la variable debe empezar por el símbolo del dólar (\$) y puede estar formado por cualquier número de letras mayúsculas, minúsculas o números. Las diéresis no están permitidas y el único carácter especial es el guión bajo (\_). Cada variable se crea automáticamente en cuanto aparece su nombre por primera vez. Si aún no se le ha asignado un valor, se asume el valor 0. El valor es un número de coma fija con 3 decimales. El valor más pequeño que 0 es 0,001. El valor más grande posible es 2.000.000 (2 millones). Si a una variable se le asigna un valor mayor, se limita automáticamente a estos 2 millones. Los valores

negativos también son posibles, de modo que todo el rango de valores va de -2.000.000 a 2.000.000. Para los valores, tanto el punto como la coma están permitidos como separadores decimales.

Una variable conserva su valor hasta el final de la presentación, a menos que se le asigne explícitamente un nuevo valor. La asignación de un valor se realiza mediante un simple signo igual (=), por el que el valor situado a la derecha del signo igual se asigna a la variable situada a su izquierda. El valor de la derecha también puede ser una variable. Las asignaciones en cadena también funcionan.

```
$varA = 5           #La variable $varA tiene ahora el valor 5
$varB = $varA       #La variable $varB tiene ahora el valor 5
$varA = $varB = 7   #Variable $varA y $varB ambas tienen el valor 7
```

Las cuatro operaciones aritméticas básicas pueden realizarse con variables y valores fijos. Los resultados de estos cálculos pueden utilizarse como entrada para comandos o pueden asignarse a variables para guardarlos. En estos cálculos se respeta la regla del punto antes del guión, aunque también pueden utilizarse corchetes para influir en el orden de procesamiento.

```
$varA = 2 + 1.5      #Variable $varA tiene el valor 3.5
$varB = $varA * 2    #Variable $varB tiene el valor 7
$varC = $varB/3      #Variable $varC tiene el valor 2.333
$varD = ($varA - $varC)*2 #La variable $varD tiene el valor
```

A menudo ocurre que se desea calcular el valor de una variable con otro valor y asignar de nuevo el resultado a la misma variable. Para ello, existe una notación abreviada en la que el símbolo de la operación aritmética va seguido de un signo igual.

```
$varA = $varA + 5    #Variable $varA se incrementa en 5
$varA += 5           #La variable $varA también se incrementa en 5
$varB -= $varA       #$varB se reduce en el valor de $varA
```

## LÓGICA

Los comandos lógicos pueden utilizarse para influir en el curso de la presentación de diapositivas vinculando la ejecución de comandos con condiciones.

## CONDICIONES

son expresiones en las que se comparan variables o valores entre sí. El resultado de esta comparación puede ser verdadero o falso. Son posibles las siguientes comparaciones:

<code>\$varA == \$varB</code>	Igualdad
<code>\$varA != \$varB</code>	Desigualdad
<code>\$varA &lt; \$varB</code>	Pequeño
<code>\$varA &gt; \$varB</code>	Más grande
<code>\$varA &lt;= \$varB</code>	Menor o igual
<code>\$varA &gt;= \$varB</code>	Mayor o igual

Tenga en cuenta que debe utilizarse un signo igual doble para comparar la igualdad. El signo igual simple se utiliza para la asignación.

Los valores o variables también pueden ser una condición en sí mismos. Si sólo se especifica una variable en la posición donde debe estar una condición, ésta se evalúa como no cumplida si esta variable es exactamente 0, en caso contrario se considera cumplida.

Se pueden enlazar varias comparaciones individuales con las palabras clave **AND** y **OR**, pudiéndose utilizar también corchetes para determinar el orden de procesamiento (de dentro a fuera).

```
IF $varA < 5 OR $varA == 8
#...
ENDIF

IF ($varA >= 10 AND $varB != 7) OR $varC == $varD
#...
ENDIF
```

### IF [CONDICIÓN]...ELSE...ENDIF

Con la **construcción IF**, se pueden ejecutar partes del script dependiendo de si se cumple una determinada condición. La **parte** opcional **ELSE** también puede utilizarse para ejecutar otra parte si no se cumple la condición. Para una mejor legibilidad, las partes en las ramas individuales están sangradas en el ejemplo. Esto es permisible y recomendable, pero no necesario.

```
#Ejemplo IF sin rama ELSE
IF $varA < 5
    PRINT "A es menor que 5"           #Sólo ocurre si A es menor que 5
ENDIF

#Ejemplo IF con rama ELSE
IF $varB == 3
    PRINT "B es 3"
ELSE
    PRINT "B no es 3"
ENDIF
```

### WHILE [CONDICIÓN]...ENDWHILE

Un bucle While se utiliza para recorrer una sección determinada una y otra vez mientras se cumpla una condición. Cuando el flujo del programa llega a la línea con el **WHILE**, se comprueba si se cumple la condición. En caso afirmativo, se ejecuta la siguiente parte del programa; en caso contrario, el flujo del programa salta a la línea situada después del **comando ENDWHILE** correspondiente. Cuando el programa llega a la línea con el **ENDWHILE** después de ejecutar los comandos, se comprueba si se sigue cumpliendo la condición. En caso afirmativo, el programa vuelve a saltar y repite toda la parte. Cada vez que la presentación de diapositivas vuelve al principio de un bucle, hace una pausa automática de 0,01 segundos. Esto hace que sea perfectamente seguro crear bucles infinitos, es decir, bucles cuya condición se cumple siempre. En muchos otros "lenguajes de programación", esos bucles infinitos darían lugar a errores graves.

```
#Salida de una cuenta atrás
$Countdown = 3
WHILE $Cuenta Atrás > 0 #repetir hasta que el número llegue a 0
  PRINT $Cuenta Atrás #Salida del número actual
  $Countdown -= 1 #Reduce la cuenta en 1
  ESPERA 1 #1 segundo de espera
ENDWHILE
PRINT "" #Texto vacío
```

## FRENO

Brake no es una construcción propia, sino un comando que permite salir de un bucle **WHILE** incluso si se sigue cumpliendo la condición tras su comando **WHILE**. Si el flujo del programa dentro de un bucle **WHILE** encuentra un **BREAK**, salta inmediatamente a la línea después del **ENDWHILE** asociado. De esta forma, se pueden crear bucles con varias condiciones de terminación. Por ejemplo, se puede crear un bucle infinito y aún así salir de él. Dependiendo del estilo de escritura preferido, esto puede dar lugar a programas cortos y claros.

```
WHILE 1 #Bucle sin fin: "1" se considera siempre
  cumplido
  SI $RUNTIME > 10
    BREAK #Tras 10 segundos se sale del bucle
  ENDIF
```

Los bucles también pueden anidarse, por lo que un comando **BREAK** sólo abandona el más interno de los bucles activos en ese momento.

```
#Bucles anidados. Se carga una imagen DECAL y esta
#luego se mueve línea por línea de izquierda a derecha. Al final de
#cada línea bájalo un poco
OPEN_DECAL "Objetivo.png"
$DECAL_X = $DECAL_Y = 0
WHILE $DECAL_Y < 100
  WHILE $DECAL_X < 100
    $DECAL_X += 10
    ESPERA 0,02
  ENDWHILE
  $DECAL_X = 0
  $DECAL_Y += 10
ENDWHILE
```

## GOTO :MARCADOR

Con un comando **GOTO**, la secuencia del programa salta directamente a la línea con el marcador especificado. Este marcador debe existir y ser único. Un marcador debe estar solo en una línea. Se le aplican las mismas reglas de nomenclatura que a las variables, pero no empieza por el signo de dólar, sino por dos puntos.

```
GOTO :END #Salta al marcador ":END"
$varA = 7 #Esta línea no se ejecuta
FIN
```

No hay límites para la posición del marcador de destino. Un **comando GOTO** puede utilizarse para salir de bucles o entrar en ellos en cualquier punto. Se recomienda utilizar el comando **GOTO** con mucha moderación, ya que puede hacer que el programa se ejecute de forma muy confusa.

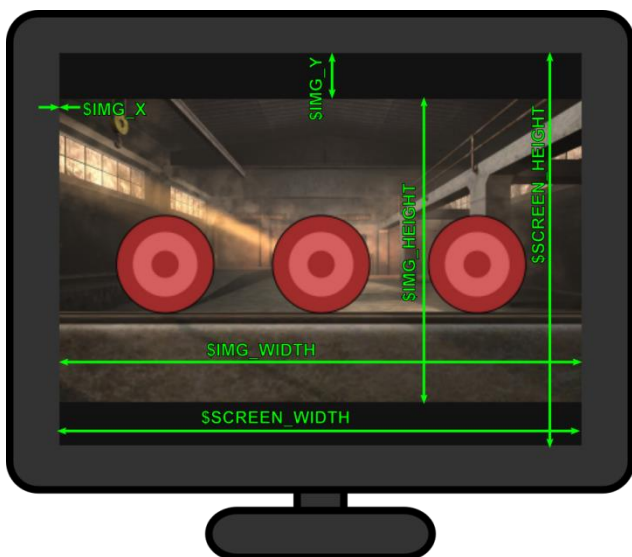
### VARIABLES DEL SISTEMA, CALCOMANÍAS Y TEXTO

Las variables de sistema son variables a través de las cuales el sistema proporciona información o a través de las cuales se pueden manipular objetos. Se pueden utilizar como variables normales, pero no conservan necesariamente su valor de forma permanente, sino que se rellenan siempre con los valores actuales sin intervención del script. Pueden dividirse a grandes rasgos en 5 categorías. Cada una de estas categorías puede mostrarse u ocultarse en el editor de diapositivas para mantener clara la lista de variables mostradas.

LAS VARIABLES GENERALES DEL SISTEMA contienen información sobre el tamaño de la pantalla y la hora

Variable	Descripción
<b>\$RUNTIME</b>	Tiempo transcurrido desde el inicio de la presentación de diapositivas También puede modificarse en caso necesario, tras lo cual el tiempo se cuenta a partir del nuevo valor.
<b>ANCHURA_PANTALLA</b>	Anchura del área de visualización en píxeles (ancho de pantalla)
<b>ALTURA_PANTALLA</b>	Altura del área de visualización en píxeles (altura de la pantalla)
<b>\$IMG_WIDTH</b>	Anchura de la imagen mostrada en píxeles
<b>\$IMG_HEIGHT</b>	Altura de la imagen mostrada en píxeles
<b>\$IMG_X</b>	Posición X de la imagen visualizada en píxeles en relación con el borde izquierdo de la pantalla.
<b>\$IMG_Y</b>	Posición Y de la imagen visualizada en píxeles en relación con el borde superior de la pantalla.

Cuando se abre un archivo de imagen, se escala automáticamente para que llene el área de visualización tanto como sea posible, pero no se distorsiona su relación de aspecto. Así, si la imagen es exactamente tan grande como la pantalla o, al menos, tiene la misma relación de aspecto, se muestra a pantalla completa. La altura y anchura de la imagen son entonces iguales a la altura y anchura de la pantalla, respectivamente, y las posiciones de la imagen son 0. Si la relación de aspecto no se ajusta, aparecen barras negras a izquierda y derecha o arriba y abajo. Los dos valores de posición indican entonces la posición de la esquina superior izquierda de la imagen con respecto a la esquina superior izquierda de la pantalla. Las variables que empiezan por "\$IMG\_" se ajustan automáticamente a los valores que se muestran en la siguiente imagen con cada **comando ABRIR**.



Estas variables son especialmente útiles si desea escribir una presentación de diapositivas de modo que funcione en cualquier resolución de pantalla. En este caso, la posición de los objetos no debe basarse en valores absolutos, sino en función de la resolución de la pantalla. Así, un objeto cuya posición X sea la mitad de **\$ANCHURA\_PANTALLA** y cuya posición Y sea la mitad de **\$ALTURA\_PANTALLA** **estará** siempre centrado en el centro de la pantalla.

**LAS VARIABLES DE ENTRADA** contienen el estado actual de las teclas del teclado. Las variables tienen un valor de 1 si la tecla está pulsada en ese momento y un valor de 0 si no lo está. Esto permite influir en la presentación de diapositivas mediante entradas de teclado. Están disponibles las teclas de letras (sin caracteres especiales), las teclas numéricas 0 a 9, las teclas F F1 a F12 y la tecla Intro. El programa ya ha asignado funciones especiales a las teclas de escape y espacio. Al igual que en la reproducción de vídeos, la presentación de diapositivas puede detenerse en cualquier momento con la barra espaciadora y finalizarse con la tecla Escape.

Variable	Descripción
<b>\$CLAVE_0 . . .</b> <b>\$CLAVE_9</b>	Estado de la tecla numérica correspondiente (1 = pulsada) / (0 = no pulsada)
<b>\$CLAVE_A . . .</b> <b>\$CLAVE_Z</b>	Estado de la tecla de letra correspondiente (1 = pulsada) / (0 = no pulsada)
<b>\$CLAVE_F1 . . .</b> <b>\$CLAVE_F12</b>	Estado de la tecla F correspondiente (1 = pulsada) / (0 = no pulsada)
<b>\$KEY_ENTER</b>	Estado de la tecla Intro (1 = pulsada) / (0 = no pulsada)

**LAS VARIABLES DE IMPACTO** se actualizan con cada nuevo disparo y contienen información sobre el mismo. Se cuentan tanto el número de disparos efectuados como la posición del último disparo. Además, se lee la información del color del píxel impactado, tanto en la imagen normal mostrada como en la imagen de puntuación. Esta última ofrece así la posibilidad de evaluar las zonas de impacto.

Variable	Descripción
<b>\$SHOT_COUNT</b>	Número de disparos realizados hasta el momento. También puede modificarse en caso necesario, tras lo cual el tiempo se cuenta a partir del nuevo valor.
<b>\$HIT_TIME</b>	Anchura del área de visualización en píxeles (ancho de pantalla)
<b>\$HIT_X</b>	Posición X del golpe en píxeles respecto al borde izquierdo de la pantalla
<b>\$HIT_Y</b>	Posición Y del golpe en píxeles respecto al borde superior de la pantalla
<b>\$HIT_RED</b> <b>\$HIT_GREEN</b> <b>\$HIT_BLUE</b>	Valores de color del píxel alcanzado en la imagen previamente abierta mediante el comando "OPEN". Las 3 variables representan los 3 canales de color rojo, verde y azul. Si la imagen no ha sido alcanzada o no se ha disparado todavía, los 3 valores son 0.
<b>\$HIT_SCORE_RED</b> <b>\$PUNTUACION_VERDE</b> <b>\$PUNTU_AZUL</b>	Valores de color del píxel alcanzado en la imagen previamente abierta mediante el comando "OPEN_SCORE". Las 3 variables representan los 3 canales de color rojo, verde y azul. Si la imagen no ha sido golpeada o no se ha realizado ningún disparo, los 3 valores son 0.

La posición en la imagen de puntuación se calcula a partir de la posición en la imagen visualizada. Si la posición en la imagen visualizada es, por ejemplo, un tercio de la anchura de la imagen desde el borde izquierdo y un cuarto desde el borde superior, esto también se aplica a la posición evaluada en la imagen de la partitura. Esto significa que la imagen visualizada y la imagen de puntuación no tienen por qué tener el mismo tamaño ni la misma relación de aspecto. Sin embargo, se recomienda utilizar imágenes del mismo tamaño, ya que esto facilita las cosas. Las imágenes siguientes muestran cómo puede funcionar la evaluación de la puntuación con una imagen de puntuación. Los tres círculos de colores muestran no sólo si se ha alcanzado un objetivo, sino también cuál.

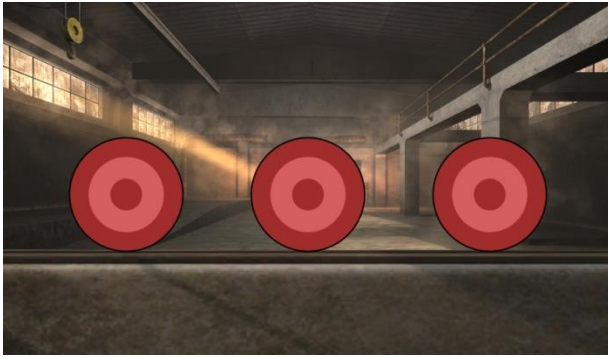


imagen mostrada

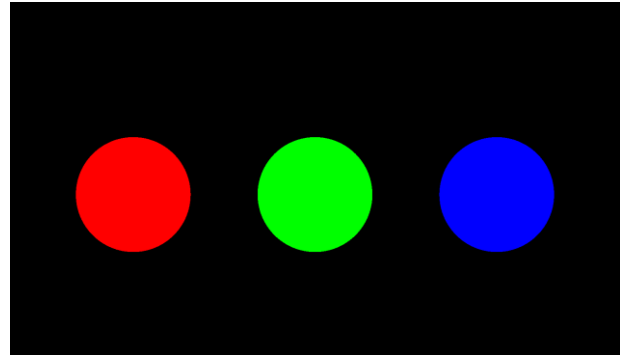


Imagen de puntuación

Si desea evaluar las zonas de impacto mediante áreas coloreadas, debe tener cuidado de que no se produzcan efectos de antialiasing no deseados al crear las imágenes. El antialiasing es una técnica que utilizan muchos programas de pintura para que los bordes parezcan más suaves. En lugar de dibujar una transición dura, se crea una transición de color suave en el borde. Por supuesto, esto resulta molesto si se quiere decidir claramente, basándose en el color, si un golpe está dentro o fuera de la superficie. Por lo tanto, hay que ser consciente del efecto y averiguar cómo desactivarlo en el programa de pintura correspondiente o utilizar un programa adecuado.



sin antialiasing



con antialiasing

## EL OBJETO DECAL

Es posible mostrar un único archivo de imagen libremente posicionable y escalable delante del fondo. Esto es posible con el llamado objeto calcomanía (calcomanía = pegatina/decal). Mientras no se haya cargado ningún archivo de imagen, la calcomanía no será visible. La carga del archivo de imagen se realiza con el comando "OPEN\_DECAL" y funciona según las mismas reglas que los otros dos comandos Abrir. La imagen debe estar en la misma carpeta que la presentación y puede tener los formatos \*.png, \*.jpg o \*.bmp. Con las imágenes PNG, también se tiene en cuenta el canal de transparencia, por lo que son el medio de elección si desea mostrar calcomanías no rectangulares. El objeto Calcomanía se configura con una serie de variables. Las variables \$DECAL\_X y \$DECAL\_Y determinan la posición relativa a la esquina superior izquierda de la pantalla y las variables \$DECAL\_WIDTH y \$DECAL\_HEIGHT **determinan el** tamaño. Sin embargo, estas dos últimas también son modificadas por el sistema. Después de cada comando OPEN\_DECAL, contienen la altura y la anchura del archivo de imagen cargado, que por lo tanto se muestra sin distorsiones en su tamaño original. Sin embargo, si estas variables son modificadas por el script después de abrir un archivo de imagen, esto asegura que la imagen de la calcomanía sea escalada al tamaño correspondiente. En el proceso, la relación de aspecto también se puede cambiar, lo que distorsiona la imagen. Por supuesto, los valores también pueden calcularse en función de la resolución de la pantalla, para que la calcomanía tenga siempre el mismo aspecto. Al posicionar la calcomanía, la esquina superior izquierda de la calcomanía es siempre el punto de referencia que se posiciona estableciendo \$DECAL\_X y \$DECAL\_Y. Para hacer el objeto calcomanía invisible de nuevo, puede simplemente llamar al comando OPEN\_DECAL con un nombre de archivo inexistente (o vacío).



Variable	Descripción
\$DECAL_X	Posición X del objeto DECAL en píxeles respecto al borde izquierdo de la pantalla.
\$DECAL_Y	Posición Y del objeto DECAL en píxeles respecto al borde superior de la pantalla.
\$DECAL_WIDTH \$DECAL_HEIGHT	Anchura/altura de la calcomanía. Se ajusta a la anchura del archivo de imagen con el comando <b>OPEN_DECAL</b> y puede modificarse para escalar la calcomanía. Si se utilizan valores negativos, la calcomanía se refleja alrededor del eje correspondiente.

## EL OBJETO DE TEXTO

Además de la calcomanía, también es posible crear un objeto de texto y posicionarlo libremente. El texto se especifica mediante el comando **PRINT**. Al igual que con los demás comandos, la palabra clave **PRINT** puede ir seguida simplemente de un texto entre comillas o de una combinación de varios fragmentos de texto y variables que se unen mediante signos más. Los valores de las variables se insertan en el texto. Se aplican las siguientes reglas:

- Si un valor no tiene decimales, sólo se emite el valor puro antes del punto decimal (por ejemplo, "3").
- Si un valor tiene decimales, siempre se muestra con los 3 decimales (por ejemplo, "12,300").
- El separador es un punto
- Si un valor es negativo, va precedido de un signo menos. Los valores positivos no tienen signo.

También está permitido escribir el comando **PRINT** exclusivamente con una variable o un valor fijo sin vincularlo a un fragmento de texto.

Para hacer el texto completamente invisible, simplemente ejecute el comando **PRINT** con un texto vacío.

Al cambiar el tamaño del texto o de la fuente, cambia el tamaño del objeto de texto. Por lo tanto, las variables correspondientes se actualizan en el proceso.

El tipo de letra es Arial y no puede cambiarse, pero el color del texto puede ajustarse mediante 3 variables.

Para crear textos de varias líneas, se puede crear un salto de línea con la combinación de caracteres "\n". No obstante, todo el comando **PRINT** debe estar en una sola línea del script.

Variable	Descripción
\$SHOT_COUNT	Tiempo transcurrido desde el inicio de la proyección de diapositivas También puede modificarse en caso necesario, tras lo cual el tiempo se cuenta a partir del nuevo valor.
\$TEXT_X	Posición X del objeto de texto en píxeles en relación con el borde izquierdo de la pantalla.
\$TEXT_Y	Posición Y del objeto de texto en píxeles en relación con el borde superior de la pantalla.
\$TEXT_SIZE	Tamaño de la fuente del objeto de texto. La altura de la fuente es aproximadamente el valor de esta variable en píxeles
\$TEXT_WIDTH \$TEXT_HEIGHT	Anchura y altura del objeto de texto. Se actualizan automáticamente cuando se modifica el tamaño del texto o de la fuente. Los valores pueden utilizarse, por ejemplo, para centrar el objeto de texto en un lugar determinado.
\$TEXT_RED \$TEXT_GREEN \$TEXT_BLUE	Determina conjuntamente el color del objeto de texto. Sólo deben utilizarse valores entre 0 y 255 para cada canal de color.

```

$TEXT_RED = $TEXT_BLUE = $TEXT_GREEN = 255 #texto color blanco
$TEXT_SIZE = $SCREEN_HEIGHT*0.06 #Tamaño del texto
PRINT $RUNTIME + " Sec.\N" + $SHOT_COUNT + " Tir"
$TEXT_X = ($SCREEN_WIDTH-$TEXT_WIDTH)/2 #Centrar verticalmente
$TEXT_Y = $SCREEN_HEIGHT*0.75 #horizontal 75%

#El código anterior escribe un texto blanco que se divide en dos
#Las líneas representan la hora y los disparos efectuados
#
# 12.865 seg.
# 14 disparo

```

## TRABAJAR CON EL EDITOR

The screenshot shows the Ares editor interface. On the left is the script editor with the following code:

```

1 #alle Ziele Aufstellen
2 $Target1 = 1
3 $Target2 = 2
4 $Target3 = 4
5 $TEXT_RED = $TEXT_BLUE = $TEXT_GREEN = 255 #WEISS
6 $TEXT_SIZE = $SCREEN_HEIGHT * 0.15
7
8 OPEN "Background0.jpg"
9
10 PRINT "READY"
11 $TEXT_X = ($SCREEN_WIDTH-$TEXT_WIDTH)/2
12 $TEXT_Y = ($SCREEN_HEIGHT-$TEXT_HEIGHT)/2
13 WAIT ((RAND 2000)/1000)+1
14 PRINT ""
15
16 $starttime = $RUNTIME
17
18 #Frühstarkerkennung
19 IF $SHOT_COUNT > 0
20 PRINT "Frühstart"
21 SOUND "Horn.wav"
22 GOTO :END
23 ENDIF
24
25 $Targets = $Target1 + $Target2 + $Target3
26 OPEN "Background" + $Targets + ".jpg"
27 OPEN_SCORE "BGScore.png"
28
29 SOUND "Peep.wav"
30

```

On the right is the preview window showing a dark industrial scene with the word "READY" in large white letters. Below the preview is the 'Variables' panel with a table of target scores:

	1	2
1 \$Target1	1	
2 \$Target2		2
3 \$Target3		4

Below the table are buttons for 'Play', 'Step', and 'Stop'.

El editor ayuda a escribir los guiones de la presentación de diapositivas. Los archivos .dia pueden cargarse y guardarse a través del menú Archivo. También puedes arrastrarlos y soltarlos en la ventana del editor. Con el botón "Reproducir/Pausa" se puede reproducir la presentación de diapositivas en la ventana de vista previa. Al hacer clic en la ventana de vista previa se simulan tomas. Además de la imagen de fondo visible para el jugador, que se carga con un comando OPEN, también se puede mostrar la plantilla de puntuación (OPEN-SCORE). También es posible trabajar a través de la presentación de diapositivas línea por línea. En este caso, un marcador amarillo indica la siguiente línea a ejecutar. Durante la ejecución, todas las variables utilizadas y sus valores actuales se muestran en la ventana de variables; las variables del sistema de las distintas categorías pueden mostrarse u ocultarse.

Si se produce un error durante la ejecución, la presentación se detiene y se resalta la línea correspondiente. En la pestaña "Errores" también se muestra una breve descripción para ayudar a encontrar la causa del problema.

Mensaje de error	Descripción
<b>Ficha(s) desconocida(s): "xyz"</b>	La parte "xyz" de la línea no puede asignarse a una variable, un comando o cualquier otra palabra clave.
<b>Número de soportes</b>	El número de paréntesis de apertura y cierre de la línea no coincide
<b>Tipos de parámetros erróneos: "xyz"</b>	En la parte "xyz", se intenta realizar una operación con componentes que no coinciden. Por ejemplo, asignar un texto a una variable ( <b>\$VAR = "texto"</b> )
<b>OPEN - Archivo no encontrado: "xyz"</b> <b>OPEN_SCORE - Archivo no encontrado: "xyz"</b>	No se ha encontrado el archivo "xyz" que debería cargarse con un comando OPEN o OPEN_SCORE. Con este error, la presentación de diapositivas no se detiene, pero carga una imagen estándar en su lugar.