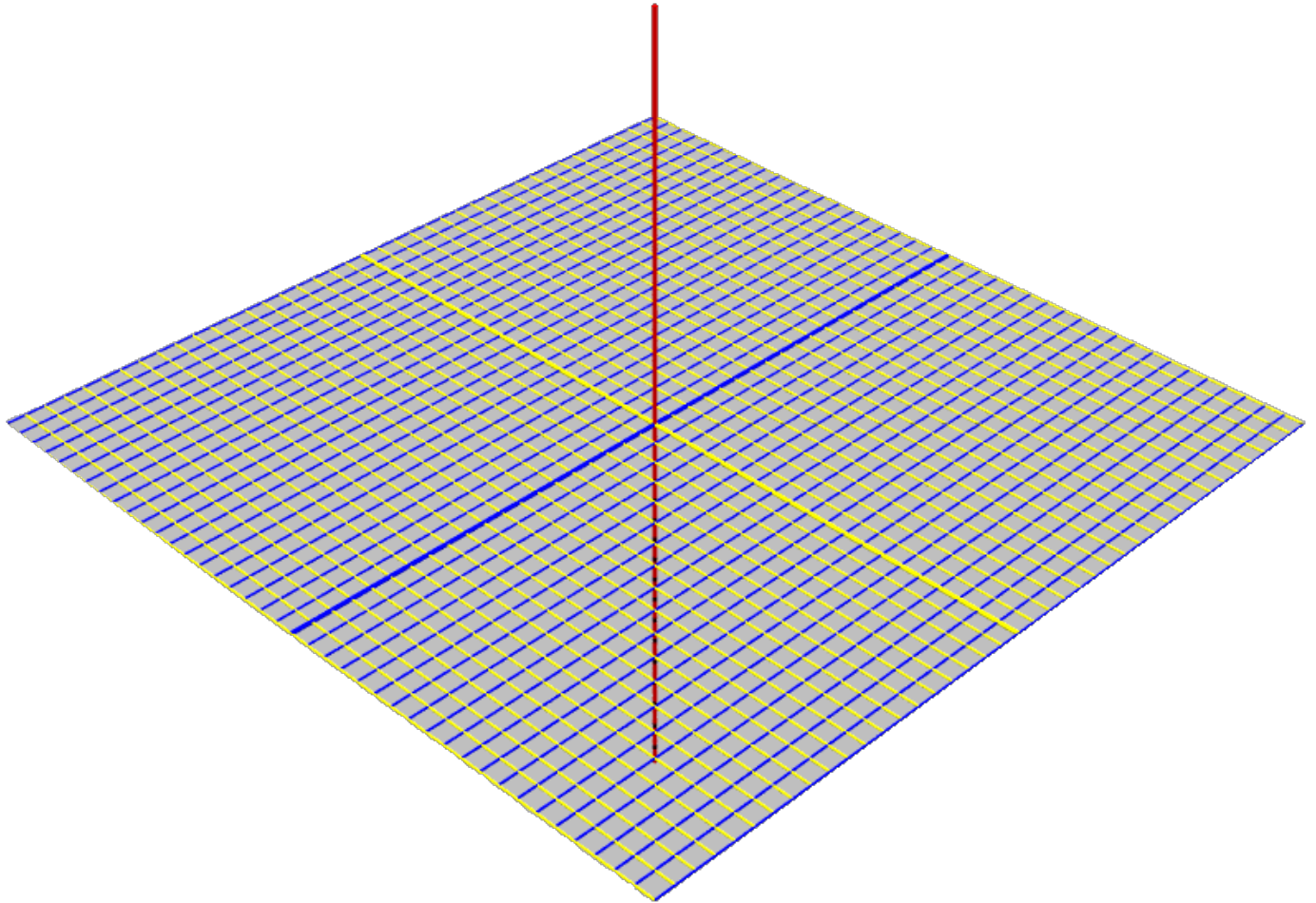


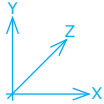
# Eagle 3D

paso a paso



**paso 1**

**INTRODUCCIÓN**



**E**n este manual avanzado que ahora comienza trataremos de ver todas aquellas cuestiones que quedaron pendientes en la parte básica. A diferencia del anterior, en el que se ha procurado seguir un ritmo lógico de aprendizaje, ahora no es necesario, ya que cada capítulo o paso en sí engloba un tema que no dependerá en la mayoría de las ocasiones de haberse aprendido un paso previo. No obstante se da por entendido que se manejan las técnicas vistas en el manual básico y que la configuración en cuanto a carpetas y demás es la misma(1), con la salvedad de que añadiremos una nueva carpeta llamada animaciones. Y recordaros por última vez que a partir de ahora haremos bastante uso de los archivos de configuración de Eagle 3D y hemos de extremar las precauciones. No estará demás hacer una copia a un lugar seguro antes de continuar.

Instalaremos dos nuevos programas en este manual, dedicados ambos a crear un clip de vídeo con la posibilidad de optar por varios formatos de salida. Que puede ser un GIF animado, FLASH e incluso AVI con diferentes niveles de compresión y calidad

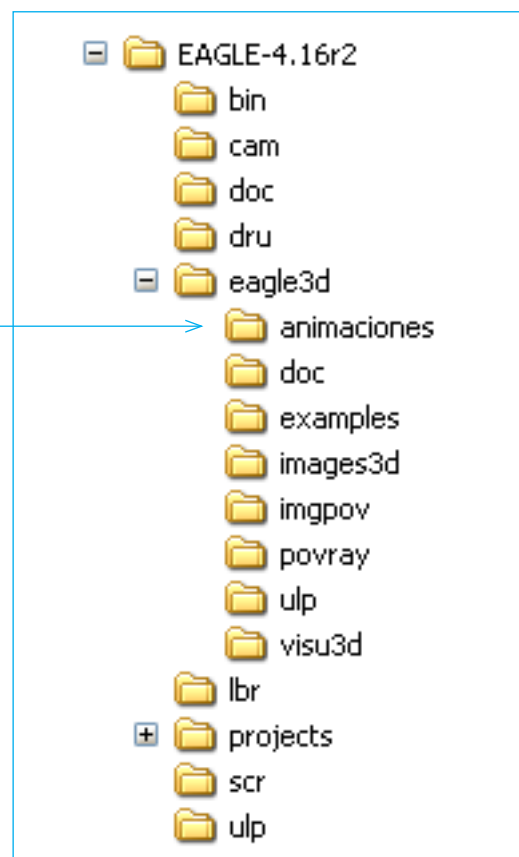
**VirtualDub** - <http://www.virtualdub.org/index>

**Easy GIF Animator** - <http://www.blumentals.net/>

El hecho de orientarme en esta ocasión por un programa de pago como es Easy GIF Animator fue por considerar que sus prestaciones y su bajo precio lo convertían en la mejor de las opciones vistas. Y en cuanto a VirtualDub es también una excelente alternativa y es gratuito. No hay nada a resaltar en cuanto a la instalación, y siempre se puede optar por otras alternativas de vuestra preferencia

En la Figura podemos ver como deben de quedar ahora nuestras carpetas

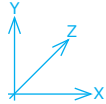
Nueva carpeta añadida



(1) El manual básico de Eagle 3D se puede descargar de este enlace: <http://eagle.pcbpics.es/3dbasico.pdf>

**paso 2**

**ANIMACIONES**

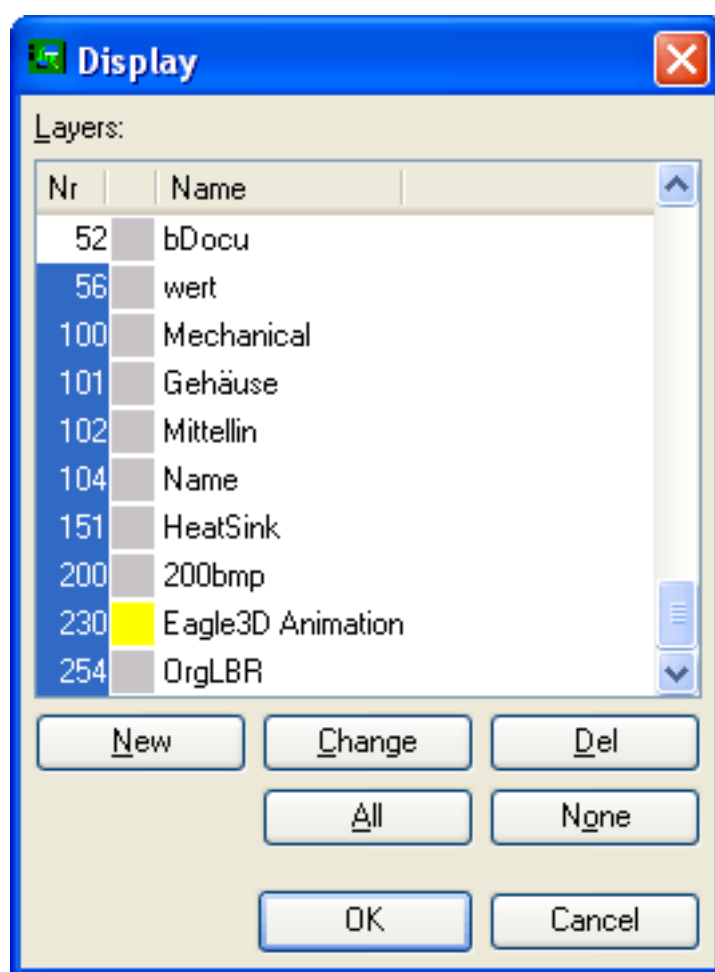


Adentrándonos ya en las animaciones de Eagle 3D se puede decir que es posible generar dos tipos de animaciones con nuestra imagen, y que en ambos casos necesitamos de un archivo .ini en el cual estableceremos los parámetros necesarios para que ésta, la animación, se genere.

## Animaciones con Eagle 3D

Las animaciones con Eagle 3D se basan en establecer una ruta de cámara en una capa del programa Eagle mediante una serie de cadenas de texto. Por defecto, y tal como se puede apreciar en la pestaña Varios del Panel de Control de Eagle 3D será la capa 230. Veámoslo paso a paso

Ejecutamos Eagle y con la herramienta Display creamos una nueva capa mediante el botón **New**. Le damos como número el 230 y en texto puede ser **Eagle3D Animation** u otro que se asocie bien con las animaciones, seleccionamos un color que contraste bien con el fondo y activamos la opción **Displayed**.



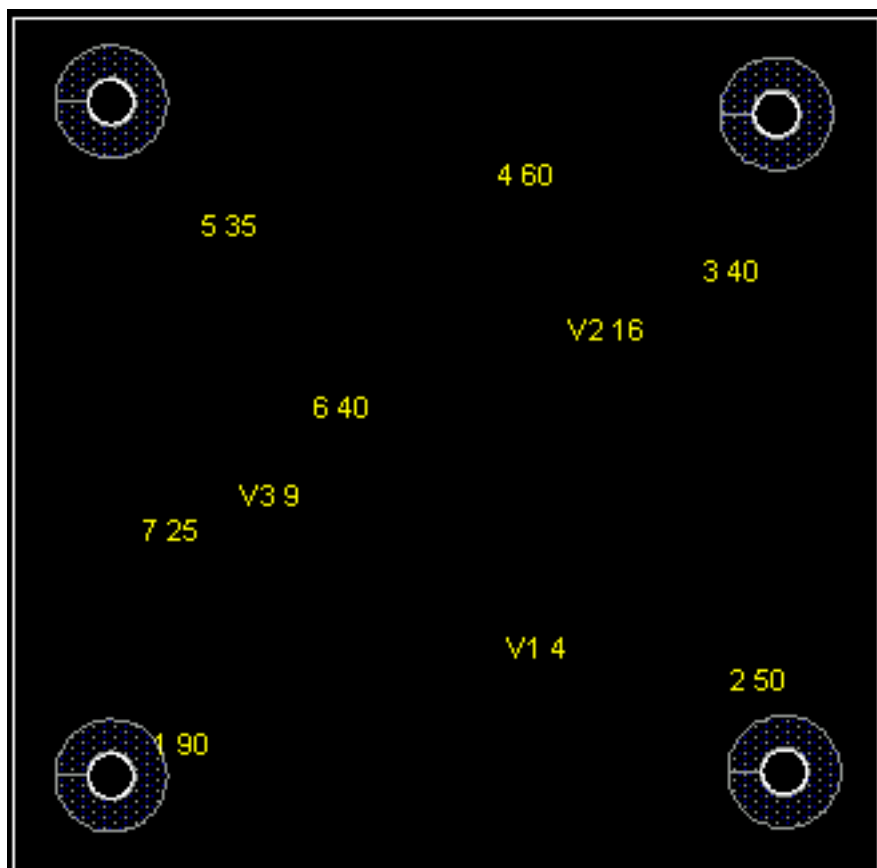
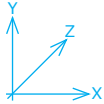
A continuación, sobre esta misma capa iremos creando el camino a seguir por la cámara. Este camino está representado por una función matemática llamada 'spline'. Son necesarios un mínimo de tres puntos de control, de lo contrario dará un error. El camino se define mediante textos que han de tener un formato especial. Este formato consiste en dos números.

El primer número es la información de la secuencia. Esto significa que en esta secuencia los puntos se usan para crear el spline. Ha de ser un número entero.

El segundo número representa la altura del punto sobre la superficie de la PCB. Este número puede ser un número real.

A modo de ejemplo: 4 6.7 significa que es el cuarto punto en el camino de la animación y está a 6.7 mm sobre la superficie de la PCB. El punto de vista de la cámara está normalmente localizado antes de la dirección de vuelo de la cámara. Es también posible añadir un path como punto de vista adicional. Para hacer esto simplemente se añaden puntos de control adicionales con una V como primer carácter: p.ej. V1 3.4.

Povray necesita un archivo .ini para crear una animación. Este archivo es generado por Eagle 3D cuando se suministran un mínimo de tres puntos de animación. El archivo tendrá el mismo nombre que el archivo Povray, pero con la extensión .ini en vez de .pov.



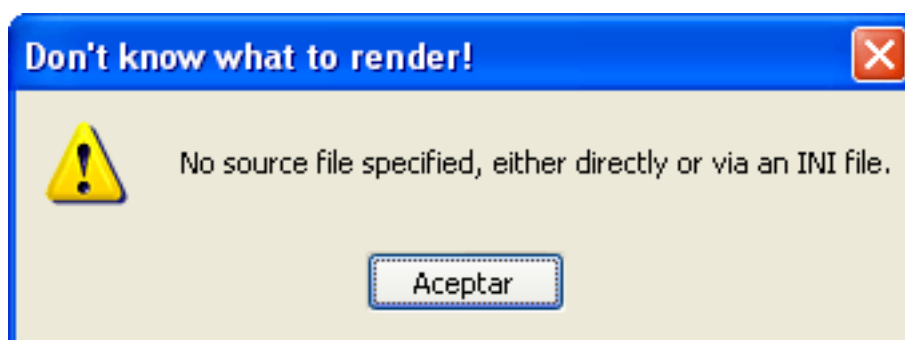
En este ejemplo se han dejado únicamente visibles la capa 20 y la capa 230 para mayor claridad, aunque mostrar los componentes ayuda mucho a la hora de establecer el camino de la cámara y puntos de vista. Los valores variarán en función del tamaño, forma de la PCB, etcétera. Es algo que se aprenderá con la práctica. En este caso los valores son ficticios, puestos únicamente a modo de ejemplo.

En la carpeta examples está el archivo *MoDsMega.brd* que es un buen ejemplo de lo que se puede hacer con esta técnica.

(Ver la capa 230 de este archivo para tener una orientación de partida).

El archivo .ini generado tendrá el mismo nombre que el archivo .pov, tan solo cambiará su extensión. Ahora solamente nos queda cargar ambos archivos con Povray, seleccionar una resolución de salida para los archivos de imagen en el archivo .pov y a continuación, seleccionar el archivo .ini y pulsar sobre el botón **RUN**.

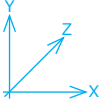
Pero parece que algo no va bien, y aparece un mensaje de error:



El mensaje nos indica que no se ha especificado un archivo de origen. Si observamos el archivo .ini generado quizá nos dé alguna pista:

```
Initial_Frame = 1  
Final_Frame = 20  
Initial_Clock = 0  
Final_Clock = 0.9  
Cyclic_Animation=off
```

Este es el texto genérico que siempre crea Eagle 3D y que como se puede apreciar no hace referencia a ningún archivo .pov de origen, de ahí el error.



Para solucionarlo le hemos de indicar el nombre del archivo y su ruta (path), de forma que quedaría algo así:

```
; Archivo .pov de origen(2)
Input_File_Name=c:\archivos de programa\eagle-4.16r2\eagle3d\imgpov\miejemp.pov
Initial_Frame = 1
Final_Frame = 20
Initial_Clock = 0
Final_Clock = 0.9
Cyclic_Animation=off
```

Pero eso no es todo, también se han de cambiar un par de parámetros del archivo .pov para que la animación se efectúe:

```
//Animation
#declare global_anim = off;
#local global_anim_showcampath = no;
```

Han de convertirse en:

```
//Animation
#declare global_anim = on;
#local global_anim_showcampath = off;
```

Y ahora ya podemos pulsar el botón **RUN** teniendo seleccionado el archivo .ini... y nuestra animación se creará. Bien, hace falta quizá aclarar una cosa, y es que a pesar de que se mencione una y otra vez la palabra animación, en realidad lo que se genera es un número n de fotogramas (determinado por Initial\_Frame y Final\_Frame) que posteriormente se convertirán en una animación con un programa externo.

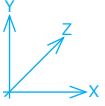
Este sistema tiene un 'pequeño' inconveniente, y es que cada vez que tengamos que hacer un cambio en el camino de la cámara en Eagle, y esto está prácticamente asegurado, nuestros archivos serán sobrescritos, perdiendo los cambios efectuados anteriormente. Esto en el caso del archivo .pov no tiene mayor importancia porque se trata tan solo de modificar un par de valores, pero en el caso del archivo .ini, en el cual como veremos se pueden establecer muchos parámetros además de los que hemos visto, sería un auténtico inconveniente.

Para solucionarlo crearemos un archivo .ini 'genérico' que será el que usaremos para todos los archivos .pov, cambiando únicamente el nombre de éste. Son muchos los parámetros que se pueden establecer para crear animaciones, describirlos todos se escapa un poco de la intención de este manual, pero veremos los que considero más habituales o interesantes. Ciertos parámetros aumentan considerablemente el tiempo de renderizado de cada imagen. En la ayuda del programa Povray tenéis la descripción de todos y cada uno de ellos.

Conviene comenzar siempre con resoluciones pequeñas y un número reducido de fotogramas, ya que es habitual tener que hacer cambios, y una vez satisfechos con el resultado aumentar el número de fotogramas, y posteriormente cambiar a una resolución mayor.

Veamos a continuación algunos de los parámetros configurables en nuestro archivo .ini. Esta información ha ido extraída del manual de Povray

(2) En los archivos .ini los comentarios comienzan por el signo del punto y coma (;), a diferencia de los archivos .pov que comienzan con dos barras (//)



### clock (reloj)

El identificador predefinido clock se usa para controlar las animaciones en POV-Ray. Al contrario que otros paquetes de animación, la acción en las escenas animadas con POV-Ray no depende del número (entero) de fotogramas. Debes diseñar tus escenas basándote en el valor de coma flotante clock. Para escenas fijas, de un solo fotograma, el valor por defecto es 0, pero puedes cambiarlo a cualquier valor de coma flotante usando la opción `Clock=n.n` en el archivo INI

Se pueden usar otras opciones INI para animar escenas, haciendo bucles automáticos durante el renderizado de los fotogramas, usando diferentes valores del reloj. Por defecto, el valor del reloj es 0 para el fotograma inicial y 1 para el fotograma final. Para el resto de fotogramas se interpola entre estos dos valores.

Por ejemplo, si tu objeto debe rotar un giro completo durante el transcurso de la animación, podrias especificar `rotate 360*clock*y`. Así, tal como el reloj va desde 0 hasta 1, el objeto rotará alrededor del eje y de 0 a 360 grados.

Aunque el valor de clock cambia de fotograma a fotograma, nunca cambia durante el análisis de la escena.

### initial\_clock

Este identificador lee el valor inicial del reloj que se asignó mediante `Initial_Clock=n.n`

### final\_clock

Este identificador lee el valor final del reloj asignado mediante `Final_Clock=n.n`

### initial\_frame

Este identificador lee el valor de fotograma inicial asignado mediante `Initial_Frame=n`

### final\_frame

Este identificador lee el valor de fotograma final asignado mediante `Final_Frame=n`

`Output_File_Type=x` Establece el tipo de archivo de salida a x

El tipo de archivo de imagen por defecto depende de la plataforma que estés usando. En Windows, por defecto es de tipo "sys", que corresponde a un BMP de 24 bits. Puedes seleccionar uno de los diferentes tipos de archivo usando `Output_File_Type=x`, donde x es uno de los siguientes:

C Formato Targa de 24 bits Comprimido (RLE, run length encoded)

N Formato PNG (portable network graphics)

P Formato PPM de Unix

S Específico del Sistema, como el Pict de Mac o el BMP de Windows

T Formato Targa-24 sin comprimir

El PNG es un formato de imagen diseñado no sólo para remplazar al GIF, si no también para mejorarlo. El formato PNG ofrece la mayor compresión disponible sin pérdidas, para aplicaciones de alta calidad, como el trazado de rayos. El formato específico del sistema depende de la plataforma escogida y se explica en la documentación específica del sistema apropiado.

`Input_File_Name=file` Establece el nombre de archivo de entrada file

Lo más normal es utilizar siempre esta opción. Si no especificas la extensión, se asumirá .pov. También se puede especificar la trayectoria completa. Además de especificar el nombre del archivo de entrada, también establece el nombre de la escena.

`Output_File_Name=file` Nombra el fichero de Salida como file

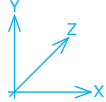
El nombre por defecto del archivo de salida se genera a partir del nombre de la escena y no necesita ser especificado. El nombre de la escena es el nombre del archivo de entrada, después de quitarle la letra de unidad, la trayectoria y la extensión.

`Height=n` Establece una altura de imagen de n píxeles

`Width=n` Establece una anchura de imagen de n píxeles

Establecen la altura y anchura de la imagen en píxeles, y especifican el tamaño del fichero de salida. La previsualización, si está activada, intenta normalmente escoger un modo de vídeo que pueda acomodar este tamaño, aunque ello no afecta en absoluto al fichero de salida resultante.





**Pause\_When\_Done**=bool Activa o desactiva la pausa al finalizar.

**Verbose**=bool Activa o desactiva la visualización de mensajes.

En algunos sistemas, cuando la imagen está acabada, la visualización de gráficos se borra y Povray cambia otra vez al modo de texto, para presentar las estadísticas finales y salir. Normalmente, cuando la visualización está activada, es porque queremos ver la imagen durante un momento antes de continuar. Usando la opción **Pause\_When\_Done=on** podemos hacer que Povray haga una pausa en modo gráfico hasta que pulsemos una tecla para continuar. Por defecto, la pausa no está activada.

Cuando no se usa la visualización de gráficos, es a menudo deseable ir viendo el progreso del trazado. La opción **Verbose=on** activa la visualización del indicador de progreso del trazado. Éste informa del número de la línea que está siendo trazada, el tiempo transcurrido hasta el momento para el fotograma actual y alguna otra información. En algunos sistemas, esta información textual puede entrar en conflicto con la visualización gráfica. Puedes necesitar desactivar el indicador de progreso cuando la visualización está activada. Por defecto, está desactivada.

**Display**=bool Activa o desactiva la visualización gráfica.

La opción **Display=on** activa la visualización gráfica de la imagen que está siendo trazada. Incluso en algunos sistemas sin gráficos, Povray puede visualizar una representación ASCII de la imagen en 80x24 caracteres. Cuando está disponible, la visualización puede ser completa, con color verdadero de 24 bits. Al establecer la opción **Display=off** desactivaremos la visualización de gráficos por defecto.

En la versión Windows, el valor por defecto es **Display=on**. Al desactivar la visualización hará que Povray simplemente no abra la ventana de salida en la que normalmente muestra la imagen.

**Antialias**=bool Activa o desactiva el anti-alias

**Antialias\_Threshold**=n.n Establece el umbral del anti-alias

**Antialias\_Depth**=n Establece la profundidad del anti-alias ( $1 \leq n \leq 9$ )

El proceso de trazado de rayos es en realidad un muestreo digital (discreto) de la imagen, normalmente con una muestra por píxel. Este muestreo puede introducir diferentes errores, como líneas curvas o inclinadas con apariencia escalonada, líneas finas rotas o discontinuas, efectos de tipo muaré (moirè), patrones de interferencia y pérdida de detalle o incluso de objetos que son tan pequeños que residen entre dos píxeles adyacentes. El efecto responsable de estos errores se llama alias.

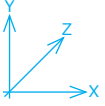
El anti alias es cualquier técnica que se use para intentar eliminar estos errores, o para reducir su efecto negativo en la imagen. En general, el anti alias hace que la imagen trazada aparezca suavizada. La opción **Antialias=on** activa el sistema de anti alias de Povray.

Cuando se activa el anti alias, Povray intenta reducir los errores enviando más de un rayo de visualización por cada píxel, y calculando la media entre los resultados para determinar el color aparente del píxel. Esta técnica se llama supermuestreo, y puede mejorar la apariencia de la imagen final, pero también incrementa drásticamente el tiempo requerido para trazar una escena, ya que deben realizarse muchos más cálculos.

El umbral por defecto es 0.3, pero puede cambiarse con la opción **Antialias\_Threshold=n.n**. El cálculo de comparación con el umbral se detalla a continuación. Si  $r_1$ ,  $g_1$ ,  $b_1$  y  $r_2$ ,  $g_2$ ,  $b_2$  son los componentes rgb de dos píxeles, la diferencia entre ambos píxeles se calcula con la fórmula:

$$\text{dif} = \text{abs}(r_1 - r_2) + \text{abs}(g_1 - g_2) + \text{abs}(b_1 - b_2)$$

Si esta diferencia es mayor que el umbral, ambos píxeles se supermuestran. Los valores rgb están en el rango de 0.0 a 1.0, así que lo más que pueden diferir es 3.0. Si el umbral del anti alias es 0.0, se supermuestran todos los píxeles. Si el umbral es de 3.0, no se usa el anti alias.



Los valores bajos del umbral implican mayor nivel de anti alias, pero menor velocidad de trazado. Usa el anti alias para la versión final de tu imagen, no para los bocetos. Cuanto más bajo es el contraste, menor debería ser el umbral. Imágenes con alto contraste pueden salvarse con valores más altos del umbral. Los valores buenos parecen oscilar entre 0.2 y 0.4.

Cuando se usa el método no recursivo, el número por defecto de supermuestras por píxel es de nueve, localizadas en una rejilla de 3x3. La opción `Antialias_Depth=n` controla el número de filas y columnas de muestras que se tomará por cada píxel supermuestreado. Por ejemplo, `Antialias_Depth=4` nos daría  $4 \times 4 = 16$  muestras por píxel.

Si no se usa el anti-alias, sólo se tomará una muestra por píxel, sin importar el método de muestreo especificado. Para un mayor información véase la ayuda de Povray

`Quality=n` Ajusta la calidad de trazado a n ( $0 \leq n \leq 11$ )

La opción `Quality=n`, permite especificar la calidad de trazado de la imagen. Puedes desear bajar la calidad en trazados de prueba y subirla para el trazado final. Los ajustes de la calidad suelen hacerse por eliminación de algunos de los cálculos que normalmente se hacen. Por ejemplo, los ajustes por debajo de 4 no trazan las sombras. Por debajo de 8 no se usan la reflexión ni la refracción. Los valores duplicados se reservan para futuras expansiones. Éstos son los valores para los diferentes niveles de calidad:

0, 1 Representa sólo colores rápidos. Sólo se usa luz de ambiente completa. Los colores rápidos se usan hasta el nivel 5.

2, 3 Muestra la luz difusa y de ambiente especificada en las texturas.

4 Traza sombras, pero no luces extendidas.

5 Traza sombras, incluyendo luces extendidas.

6, 7 Procesa los patrones de textura, activa el uso de fotones

8 Calcula los rayos reflejados, refractados y transmitidos.

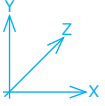
9, 10, 11 Calcula el medio y la radiosidad

El valor por defecto, si no se especifica otro, es 9.

`Cyclic_Animation=bool` Establece la animación cíclica a on/off

Muchas secuencias de animación generadas por ordenador están diseñadas para que sean vista en un bucle continuo. Supongamos que tenemos un objeto que rota exactamente 360 grados a lo largo de una animación y que pusimos `rotate 360*clock*y` para hacerlo. Ambos, el primero y último fotograma serían idénticos. Durante la reproducción de la animación habría una breve sacudida de un fotograma. Para eliminar este problema se necesita ajustar la variable `clock` de modo que el último fotograma no coincida con el primero. Por ejemplo una animación cíclica no debería usar la variable `clock` desde 0.0 a 1.0. Debería ir desde 0.0 hasta 0.9 en incrementos de 0.1. Sin embargo si se cambia a 20 fotogramas debería ir desde 0.0 hasta 0.95 en incrementos de 0.05. Esto complica las cosas porque hay que cambiar el valor final para `clock` cada vez que se cambia `Final_Frame`. Establecer `Cyclic_Animation=on` hará que Povray ajuste automáticamente el valor final para `clock` en las animaciones cíclicas sin importar el número total de fotogramas. El valor por defecto para esta opción es off.

Crearemos a continuación dos archivo .ini genéricos para usarlos como punto de partida en nuestras animaciones: *inicial.ini* y *final.ini*.



## ARCHIVO *INICIAL.INI*

```
; Cambiar las siguientes líneas en el archivo .pov:
; #declare_global_anim = on;
; #local_global_anim_showcpath = off;

; Alto y ancho de la imagen
Height=120
Width=160

; Para un procesado más rápido de la imagen
Antialias=off

; Formato del archivo de imagen
Output_File_Type=S
; N para PNG
; S para formatos de archivo de sistema (Windows BMP, Mac Pict ...)
; C para Targa (RLE) comprimido
; T para Targa sin compresión
; P para Unix PPM

; Archivo .pov de origen
Input_File_Name=c:\archivos de programa\eagle-4.16r2\eagle3d\imgpov\miejemp

; Archivo de destino. La extensión está determinada por Output_File_Type
Output_File_Name=c:\archivos de programa\eagle-4.16r2\eagle3d\animaciones\miejemp

; Fotograma de inicio
Initial_Frame=1

; Fotograma final
Final_Frame=15

; El reloj comienza con un valor inicial nulo
Initial_Clock=0

; El reloj termina con un valor 1
Final_Clock=1

; Animación cíclica
Cyclic_Animation=off
```

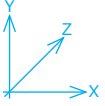
A continuación modificamos **miejemp** por el nombre del archivo .pov cargado. No es necesario indicar la extensión de entrada, y la de salida está determinada por Output\_File\_Type. Teniendo ambos archivos cargados, ejecutamos el archivo .ini y la sorpresa viene nuevamente en forma de error de permisos... ¿Recordáis cuando se hablaba de los errores de permisos en Povray?(3) y se decía que se volvería a este tema porque existe una opción que consiste en definir los permisos 'a mano' tanto de escritura como de lectura. Bien, pues ese momento ha llegado. En Povray menú Tools seleccionamos **Edit PVENGINE.INI** y añadimos las siguientes líneas:(4)

```
[Permitted Input Paths]
3=c:\archivos de programa\eagle-4.16r2\eagle3d\imgpov\

[Permitted Output Paths]
3=c:\archivos de programa\eagle-4.16r2\eagle3d\animaciones\
4=c:\archivos de programa\eagle-4.16r2\eagle3d\imgpov\
```

(3) Página 11 del manual básico de Eagle 3D.

(4) Los números pueden ser diferentes a los aquí expuestos, en todo caso serán los siguientes a los que contenga el archivo



De esta forma estamos dándole los permisos necesarios para que utilice la carpeta 'imgpov' como entrada/salida y la carpeta 'animaciones' como salida. Ahora ya podemos ejecutar mediante el botón **RUN** nuestro archivo .ini y se crearán los archivos de imagen.

Cuando tengamos la animación a nuestro gusto podemos crear el archivo .ini para un resultado final ya más elaborado:

### ARCHIVO FINAL.INI

```
; Cambiar las siguientes líneas en el archivo .pov:
; #declare_global_anim = on;
; #local_global_anim_showcpath = off;

Antialias=on
Antialias_Threshold=0.1
Antialias_Depth=2

; Establecemos el alto y el ancho para las imágenes (según nuestro criterio), por ejemplo:
Height=480
Width=640

; Formato del archivo de imagen
Output_File_Type=N
; N para PNG
; S para formatos de archivo de sistema (Windows BMP, Mac Pict ...)
; C para Targa (RLE) comprimido
; T para Targa sin compresión
; P para Unix PPM

; Archivo .pov de origen
Input_File_Name=c:\archivos de programa\eagle-4.16r2\eagle3d\imgpov\miejemp

; Archivo de destino. La extensión está determinada por Output_File_Type
Output_File_Name=c:\archivos de programa\eagle-4.16r2\eagle3d\animaciones\miejemp

; Fotograma de inicio
Initial_Frame=1

; Fotograma final -- determinar el número deseado
Final_Frame=60

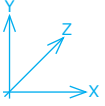
; El reloj comienza con un valor inicial nulo
Initial_Clock=0

; El reloj termina con un valor 1
Final_Clock=1

; Animación cíclica
Cyclic_Animation=off

; Pausa una vez terminado
Pause_when_Done=off
```

En todo caso, crear una animación no es difícil, crear una 'buena' animación es algo que pasa por un proceso de experimentación y práctica antes de llegar a resultados satisfactorios.



## Animaciones básicas en Povray

Si las animaciones en Eagle 3D se pueden calificar de sencillez y fácil aprendizaje (son básicamente un tema de experimentación y práctica, tal como se dijo), las animaciones en Povray lo son aún más.

No es necesario haber trazado una ruta de cámara en la PCB ya que solo vamos a actuar sobre la rotación de los ejes para crearla.

Podemos utilizar los archivos .ini creados con un pequeño cambio en ellos:

```
; En caso de un archivo .pov sin ruta trazada en Eagle  
; Cambiar en el archivo .pov el eje de rotación, por ejemplo:  
; #local pcb_rotate_y = 360*clock;
```

```
; En caso de un archivo .pov con ruta trazada en Eagle  
; Cambiar las siguientes líneas en el archivo .pov:  
; #declare_global_anim = on;  
; #local global_anim_showcpath = off;
```

De esta forma nos recordará siempre lo que debemos hacer en el archivo .pov. En caso de tratarse de una PCB en la cual se ha trazado un camino de cámara, se han de cambiar las líneas:

```
//Animation  
#declare global_anim = on;  
#local global_anim_showcpath = off;
```

Manteniendo las demás sin cambios, y en caso de tratarse de la otra opción, cambiar únicamente esta línea:

```
#local pcb_rotate_y = 360*clock;
```

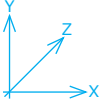
No es necesario nada más para crear una animación que gire 360° sobre el eje y, creando para ello un número de fotogramas determinado en el archivo .ini.

Una vez creada la secuencia de imágenes final pasamos a la segunda parte del proceso, que consiste en convertir esas imágenes en una animación. Es importante la elección del formato de imagen porque hay diferencias sustanciales en cuanto a su tamaño. Como muestra, renderizar el archivo *demo3.pov* a una resolución de 640\*480 nos da los tamaños de archivo que a continuación se muestran:

```
demo3.bmp = 901KB  
demo3.png = 335KB  
demo3.tga = 817KB (Targa Comprimido)  
demo3.tga = 901KB (Targa sin comprimir)
```

Y esto es muy importante si tenemos que renderizar un número considerable de imágenes, y también lo es cuando tratemos esas imágenes en el programa que usemos para crear la animación, porque de ello dependerá en gran medida el tamaño final. No todos los programas aceptan los tres formatos, aunque es muy probable que si acepten el .bmp y el .png.

También dependerá mucho el tamaño final de la animación del formato de vídeo elegido. No es lo mismo .avi sin comprimir que .mpg comprimido, gif animado o flash.



## VirtualDub

Ejecutamos el programa y en Menú File seleccionamos **Open video file...**, seleccionamos la primera de las imágenes (basta con seleccionar la primera) y pulsamos el botón **Abrir**. Nuestras imágenes estarán ahora preparadas para ser convertidas a un clip de vídeo. Podemos visualizar la secuencia con los botones situados en el margen inferior, además tenemos botones para control de keyframes, controles de escena y controles mark-in y mark-out .



Para guardar el clip pulsamos F7 o seleccionamos en el Menú File la opción **Save as AVI...** de esta forma se grabará al formato AVI sin comprimir y a una secuencia de 29.970 fps, lo cual posiblemente no es lo que deseamos. Para cambiarlo, en el Menú Video seleccionamos **Frame Rate...** y ahí en Source rate adjustment lo cambiamos al valor deseado. Este ajuste afectará a todos los modos de grabación. También en el Menú Video podemos cambiar con la opción **Compression...** el formato de vídeo para el clip como DivX, MPEG, etcétera. En todo caso lo hemos de grabar como AVI.

Para crear un GIF animado, en el Menú File seleccionamos **Export Animated GIF** e indicamos nombre de archivo y looping para el clip.

Dispone de más opciones pero no se consideran imprescindibles en este caso. Leerse la ayuda para más información.

## Easy GIF Animator

La primera vez que ejecutamos Easy GIF Animator nos muestra una pantalla desde la que es posible optar mediante botones a las opciones necesarias para crear la animación de una forma muy sencilla. Una vez que conozcamos un poco el programa esta ventana se puede evitar, para ello seleccionamos 'Don't show this Welcome Screen on startup'. Pero como es la primera vez, pulsamos el botón **Create New Animation**, y después seguimos el proceso pulsando el botón **Add Image...** y seleccionamos todas la imágenes que conformarán la animación (a diferencia de VirtualDub en el que solamente se seleccionaba la primera). Pulsamos el botón **Next** y pasamos a configurar el Loop and Delay, donde le indicaremos el tiempo entre fotogramas y el número de veces que se repetirá nuestro clip. A continuación pasamos a un nuevo cuadro de diálogo donde hemos de indicarle la posición de la imagen(Image Position). Dice: El tamaño de la animación se calcula en base a la imagen de mayor tamaño. Es necesario que las imágenes pequeñas estén dentro de los límites del fotograma. ¿Donde se debería situar la imagen?

Continuamos el proceso hasta el final y tendremos ya la aplicación con todas las imágenes cargadas.



En la aplicación se distinguen tres partes. En la parte superior los menús y la barra de herramientas, a la izquierda los fotogramas y una segunda barra de herramientas que nos permite añadir o eliminar fotogramas, desplazarlos, editar, añadir efectos o insertar textos.

La parte derecha se divide a su vez en dos. Tenemos dos pestañas en la parte superior: Edit, con su propia barra de herramientas, para la edición o creación de nuevos fotogramas y Preview, que nos permite ver el clip de vídeo.

En la parte inferior tenemos también un par de pestañas, Animation Properties en la que se puede establecer el número de veces que se repetirá la animación, el color de fondo, e incluso cambiar su tamaño. Y la pestaña Frame Properties en la que se puede establecer la demora entre fotogramas.

En el Menú File encontraremos las opciones para grabar nuestro clip una vez concluido: **Export to AVI...**, **Export to Flash...** y **Save As...** que nos permite exportar como GIF Animado.

Y poco más podemos decir del programa, ya que es muy sencillo de usar. Al igual que en el caso de VirtualDub, recurrir a la ayuda para obtener más información.