

Tutorial de POV-Ray (III)

Por Eric Albrecht

Después de leer los primeros dos tutoriales deberías ser capaz de usar LDView para convertir un archivo LDraw en un archivo POV-Ray y hacer un render sencillo. También deberías tener un entendimiento básico del contenido de ese archivo LDraw. Esta vez, comenzaremos a intentar mejorar la calidad y el realismo de los render.

Después de la propia geometría, la parte más importante de crear un render realista es la iluminación. ¡Después de todo, sin luz incluso la más espectacular geometría sólo resultaría en una pantalla llena de píxeles negros! Todo lo que nuestros ojos ven, tiene el aspecto que tiene gracias a la luz. Ver e interpretar luz nos es tan innato que es fácil pasar por alto lo complejo que las fuentes y resultados de la luz son en realidad. Un solo fotón que impacta en nuestra retina puede haber viajado millones de kilómetros y haber sido reflejado, refractado y filtrado miles de veces. Nuestros cerebros son extremadamente eficientes al interpretar la luz, y son por lo tanto también muy adeptos a darse cuenta de que algo no está bien. Averiguar por qué está mal es la parte complicada. El estudio de la luz evidentemente es un tema demasiado complejo para este artículo, pero hay algunas cosas sencillas que podemos hacer para mejorar un render.

Para ilustrar los resultados usaré un modelo más complejo que solo el brick rojo de anteriores lecciones. He elegido el 8110 Unimog porque ofrece muchas formas y colores a modo de ilustración. Para empezar, vamos a renderizar el modelo tal y como se exporta desde LDView. Este es el resultado:



Observa cómo las sombras son completamente negras. Por ejemplo, el interior de la cabina tiene zonas que son demasiado oscuras. Ahora, mira las sombras en el suelo. Son precisas y distraen. Los lugares donde coinciden dos o más sombras tienen un aspecto extraño. Así que aunque el modelo tiene reflejo y sombra, dándole un mejor aspecto que si lo viéramos en LDView, aún no está bien del todo. Para entender el porqué, miremos dentro del archivo POV-Ray en la sección que define la luz.

```
// Lights
light_source { // Latitude,Longitude: 45,0,LDXRadius*2
  <0*LDXRadius,-1.414214*LDXRadius,-1.414214*LDXRadius> + LDXCenter
  color rgb <1,1,1>}
light_source { // Latitude,Longitude: 30.120,LDXRadius*2
  <1.5*LDXRadius,-1*LDXRadius,0.866026*LDXRadius> + LDXCenter
  color rgb <1,1,1>}
light_source { // Latitude,Longitude: 60,-120,LDXRadius*2
  <-0.866025*LDXRadius,-1.732051*LDXRadius,0.5*LDXRadius> + LDXCenter
  color rgb <1,1,1>}
```

Aquí vemos tres fuentes que son puntos de luz blanca, distribuidos alrededor del modelo. Los puntos de luz son los tipos de fuente de luz más sencillos. Son los más fáciles de definir y los más rápidos de renderizar. El hecho de que la luz proviene de un solo punto hace que las sombras sean tan duras. Sin embargo, en la vida real no hay cosa tal como un punto de luz. Cada fuente de luz, incluso las más pequeñas, se distribuye sobre algún área finita. Algunas luces, como el sol, se distribuyen sobre un área muy amplia. Blanco (definido como RGB 1,1,1) es el color más simple. Pero en la vida real casi ninguna fuente de luz realmente emite cada parte del espectro visible, así que el aspecto de luz blanca no es real. Afortunadamente, ambas cosas se pueden cambiar fácilmente. Vamos a reemplazar la definición de luz con la siguiente:

```
// Lights
//front side light
light_source {
  <0*LDXRadius,-1.414214*LDXRadius,-1.414214*LDXRadius> + LDXCenter
  color rgb <1,1,0.9>*0.7
  area_light 100,100,3,3
  orient
  circular
  adaptive 1}

//back light
light_source {
  <1.5*LDXRadius,-1*LDXRadius,0.866026*LDXRadius> + LDXCenter
  color rgb <1,1,0.9>*0.7
  area_light 100,100,3,3
  orient
  circular
  adaptive 1}

//key light
light_source {
  <-0.866025*LDXRadius,-1.732051*LDXRadius,0.5*LDXRadius> + LDXCenter
  color rgb <1,1,0.9>*0.9
  area_light 100,100,3,3
  orient
  circular
  adaptive 1}
```

La cosa más evidente que he hecho es cambiar los puntos de luz por áreas. Las luces que he definido son circulares con un diámetro de 100 unidades (5 studs). POV-Ray crea luces de área usando una cuadrilla de puntos de luz. En este caso, he definido una cuadrícula de 3x3, lo que significa que cada área de luz consta de 9 puntos de luz formados en círculo. Lo siguiente que he hecho es reducir el azul en el color de la luz en un 10%(RGB es ahora 1,1,0.9). Esto hace que la luz sea un poco más amarilla, como la de la luz solar o de una bombilla incandescente. La mayoría de los fotógrafos profesionales emplean 3 luces que llaman a la luz principal (delante), luz trasera y otra fuera del eje para resaltes. Lo he replicado aquí. También he incluido un factor para cada luz que puedes usar para ajustar la luminosidad relativa de cada una. En este caso hice la luz principal la más brillante (0.9), pero puede que no siempre quieras hacer eso, dependiendo del efecto que quieras conseguir. También puedes ajustar la luminosidad general, subiendo o bajando todos los niveles en la misma cantidad. Veamos cuánta diferencia hace esto (parte superior siguiente página)

Comparado con el último render, este tiene sombras mucho más suaves y realistas. También el color es algo mejor. Pero las zonas oscuras siguen siendo demasiado oscuras.

Con sólo 3 luces simplemente no hay manera de iluminar esos recovecos. Un truco realmente sencillo es añadir reflectividad al suelo. Esto permitirá que algo de la luz refleje en las zonas oscuras. La definición del suelo se encuentra al final del archivo POV-Ray. Vayamos allí para añadir un 40% de reflectividad al suelo de esta manera:

```
// Floor
object {plane { LDXFloorAxis, LDXFloorLoc hollow }
  texture {pigment { color rgb <LDXFloorR,LDXFloorG,LDXFloorB> }
    finish { ambient LDXFloorAmb diffuse LDXFloorDif reflection 0.4}}}
```



La última parte que está en cursiva es lo que he añadido yo. Veamos cuánta diferencia hace esto.



¡Menuda diferencia! Todo es mucho más luminoso y la parte inferior tiene menos sombras. Por ejemplo se ven mejor los huecos de la ruedas. Pero aún hay lugar para mejorar. Algunas zonas siguen siendo muy oscuras. En el mundo real, luz difusa nos rodea por todas partes y se refleja en los objetos cercanos, causando cambios en el color filtrado de la luz localmente. POV-Ray simula esto con una herramienta llamada radiosity. Para implementarla, primero crearé una esfera de cielo alrededor de toda la escena. Una esfera de cielo es un esfera muy grande (radio = 10000) alrededor de la imagen. Por ahora nuestra esfera de cielo tendrá el mismo color que nuestras luces, pero en futuras lecciones lo haremos más complejo para replicar el entorno del mundo real de forma más realista. Una vez tengamos una esfera del color de fondo, añadiremos radiosity. Añade esto al inicio de tu archivo POV-Ray:

```
#include "rad_def.inc"
global_settings {
    assumed_gamma 1.4
    max_trace_level 10
    adc_bailout 0.01/2
    radiosity {Rad_Settings(3, off, off)}
    sky_sphere {pigment {color rgb <1,1,0.9>}}
}
```

Radiosity dispone de multitud de parámetros. Afortunadamente, el archivo que hemos incluido (rad_def) viene con POV-Ray y contiene muchos ajustes por defecto que nos quitan casi todo el trabajo. Si haces un clic derecho en "rad_def-inc" puedes abrirlo y ver lo que hay dentro. Lo principal es que hay 10 conjuntos de parámetros diferentes (o niveles de calidad) definidos, numerados del 1 al 10. Habitualmente usaremos los niveles 1, 3 o 5. El nivel 1 es de mala calidad pero renderiza deprisa. Utilízalo para tener una idea del aspecto que dan tus niveles de iluminación al ajustar los niveles de luminosidad de tus otras luces. El nivel 3 es el que usarás habitualmente y tiene muy buen aspecto. El nivel 5 es una locura, pero es fantástico. Se describe como "para los frikis de calidad pacientes con ordenadores rápidos y a punto de irse de vacaciones" y no es broma. Este nivel tardará una eternidad en renderizar, pero merece la pena para imágenes muy especiales. Puedes elegir qué ajuste usas cambiando el primer número después de Rad_Settings en el texto que te di para incluir en tu archivo. Lo he puesto en cursiva para que puedas ver que puse el nivel 3. Veamos qué resultado da.



¡Eso ya está mejor! La iluminación es mucho mejor. Las zonas oscuras han desaparecido. Todo tiene al menos algo de luz. Las piezas transparentes del techo tienen mejor aspecto. Las sombras en el suelo son menos definidas porque la luz difusa las esconde. Un problema nuevo es que el color plano de la esfera del cielo ha hecho que toda la iluminación parezca un poco plana. Eso lo trataremos la próxima vez con la iluminación High Dynamic Range (HDR) después de lo cual tendrá este aspecto:



Tengo que confesar que he hecho trampa. La primera vez que añadí radiosity, el resultado era demasiado brillante así que bajé la calidad a 1 y experimenté bajando la luminosidad de las otras luces hasta obtener un resultado que me gustaba. Entonces volví a subir la calidad a 3 e hice el render que ves. Esta es una buena manera de experimentar sin que tome mucho tiempo. Probablemente no consigas el efecto deseado la primera vez. Pero de eso trata esta lección: la iluminación hace toda la diferencia y tienes que experimentar hasta conseguir que esté bien.

Toda esta calidad viene con una penalización. Curiosamente el tamaño del modelo no tiene mucho efecto en el tiempo de renderizado. Si tu modelo tiene 1 o 1000 piezas, de cualquier manera el ordenador debe calcular el color de cada pixel individualmente. Un modelo más grande requiere más tiempo de análisis (parse time) y requiere más RAM, pero no mucho más tiempo de renderizado. Por otro lado, la iluminación requiere mucho tiempo de CPU. La primera imagen con sólo 3 puntos de luz tomó 2,6 minutos en un ordenador muy rápido a 1280x960 pixeles. La segunda con luces de área tomó 14,5 minutos. Esto no es sorprendente ya que en efecto tiene 9 veces, más fuentes de luz (cada área de luz es de 3x3). La imagen final con radiosity a nivel 3 tomó 45 minutos. Si hubiera usado calidad 5 hubieran sido muchas horas. El modelo más grande que jamás he renderizado tenía casi medio millón de piezas (el Klingon Bird of Prey, abajo). El análisis tomó 15GB de RAM, pero el render se hizo en unas pocas horas. Por otro lado, el render más largo que he hecho jamás (el Tow Truck, abajo) estuvo procesando durante 48 días, debido a las muchas fuentes de luz y especialmente la niebla baja. Todo esto significa que tienes que encontrar el equilibrio entre tu paciencia y la calidad que quieres conseguir.

¡Feliz renderizado!

#

