

Tutorial de POV-Ray

Por Eric Albretch

Si has estado siguiendo los tutoriales de Jetro sobre LDraw en la revista HispaBrick Magazine o en TechnicBRICKs, entonces ya conocerás la biblioteca de piezas de LDraw y los editores de LDraw como MLCAD. Ya sabes cómo instalar las piezas no oficiales (piezas que aún no han sido oficialmente publicadas como parte de la biblioteca LDraw), crear modelos virtuales y cómo verlos con visores como LDView. Tal vez incluso sepas generar complejas piezas flexibles con LSynth. Tal vez hayas tratado de hacer tus propias instrucciones con LPub. Ahora que has pasado tanto tiempo haciendo un modelo virtual, el siguiente paso lógico es que sea hermoso. Los editores como MLCAD son eficientes y rápidos mostrando tus modelos, pero no usan iluminación compleja o suavizada, así que obtienes algo con este aspecto.



Visores independientes como LDView usan renderizado en tiempo real y, con las preferencias apropiadas seleccionadas, pueden producir una imagen mucho más llamativa. Esta imagen tiene sombras suaves y bordes suavizados. Se ve bien, pero todavía no se ve real.



Para realmente hacer brillar tus modelos, es necesario utilizar un programa de trazado de rayos. Estos son programas complejos con capacidades muy avanzadas, pero las recompensas de aprender a manejarlo pueden ser imágenes increíblemente realistas. Esta imagen fue producida partiendo del mismo archivo LDraw que se utilizó para la primera imagen, pero convertido, mejorado y renderizado con POV-Ray, un programa gratis (open source) de trazado de rayos.



Este tutorial comienza asumiendo que has utilizado el instalador LDraw All-in-One, por lo que ya tienes el siguiente software necesario:

- Biblioteca LDraw
- MLCAD (u otro editor)
- LDView 4.1 o posterior (PC o Mac)
- POV-Ray 3.6 * (PC o Mac)
- Biblioteca LGEO *

Los elementos marcados con un asterisco (*) no se instalan por defecto, por lo que tendrás que volver e instalar el AIOI de nuevo y asegurarte de seleccionarlos. Si estás usando un Mac, todavía puedes hacer todo lo que aparece en este tutorial, pero tendrás que instalar el software necesario de forma manual porque no hay instalador All-in-One. Además del software anterior, algunas de las características avanzadas de tutoriales posteriores requerirán Mega-POV, un parche de POV-Ray, que puedes elegir instalar ahora. (<http://megapov.inetart.net/>)

El conocimiento de los diferentes tipos de archivos y su funcionamiento será de gran valor en la solución de problemas más adelante. Vamos a empezar por revisar qué son los archivos LDraw realmente, en su núcleo. Los archivos LDraw consisten en nada más que líneas, triángulos y cuadriláteros. Para fines de representación, no nos preocupamos por los bordes, ya que nunca se mostrarán en una imagen real, así que vamos a olvidarnos de ellos. Los triángulos y cuadriláteros

solo pueden crear superficies, no sólidos, así que las piezas de LDraw son huecas. Los triángulos y cuadriláteros tampoco pueden crear curvas, solo se pueden aproximar con facetas. Esto significa que el lenguaje LDraw no es muy bueno para hacer suaves curvaturas complejas.

Las piezas LDraw se almacenan como archivos de texto en una biblioteca. Cada archivo de pieza por lo general hace referencia a "primitivos", que son pequeñas porciones de las piezas que se usan una y otra vez (como los studs). Un archivo de modelo de LDraw también es un archivo de texto que hace referencia a muchas piezas, las ensambla en el espacio, y les asigna colores. La posición y la rotación en el espacio se controlan con una matriz de transformación. No necesitas saber cómo funciona una matriz de transformación para utilizar estas herramientas, pero ayuda. A continuación se muestra una línea de un archivo de modelo de LDraw que posiciona un solo ladrillo rojo 2x4 en el espacio. El primer número, 1, solo significa que esta línea está llamando a una pieza de la biblioteca. El segundo número, 4, es el color rojo. Los siguientes 3 números, 0 0 0, son la posición de la pieza en el espacio XYZ. Los siguientes 9 son una matriz 3x3 de las rotaciones en los ejes XYZ (ésta es la parte más difícil). Por último, 3001.dat es el número de pieza para un ladrillo 2x4.

```
1 4 0 0 0 1 0 0 0 1 0 0 0 1 3001.dat
```

Un modelo de LDraw se compone de cientos o miles de estas piezas, posiblemente divididos en submodelos. A su vez, cada pieza se divide en primitivos, y cada primitivo está formado de líneas, triángulos y cuadriláteros. Todo esto junto forma tu modelo. Sin embargo, este lenguaje es único para LDraw y no puede ser leído por POV-Ray o cualquier otro trazador de rayos. Esto significa que tenemos que convertirlo.

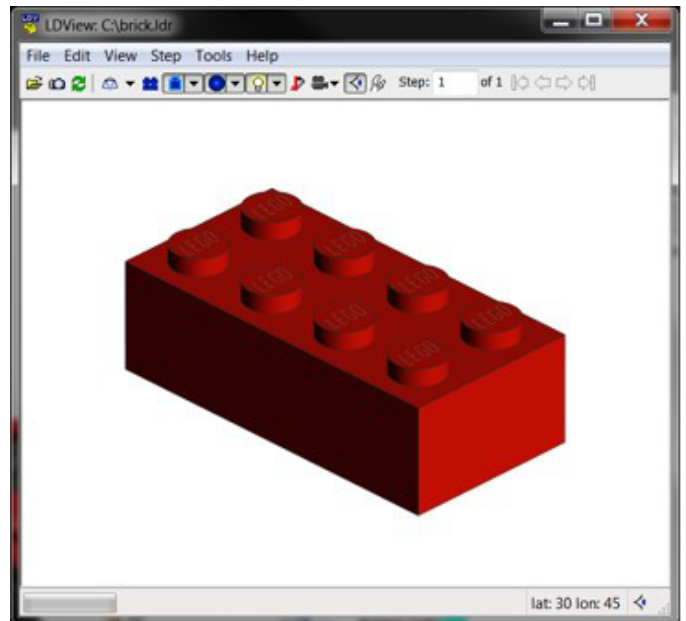
POV-Ray crea formas usando CSG (Geometría Sólida Constructiva). Este método consiste en comenzar con sólidos simples, tales como conos, esferas y cajas y luego combinarlos con operaciones booleanas como la suma y la intersección para crear formas complejas. Obviamente esto es muy diferente a LDraw así que el proceso de convertir de uno a otro no es sencillo. Por suerte para nosotros, ya se ha resuelto este problema por nosotros. También por suerte para nosotros, POV-Ray coloca las piezas en el espacio utilizando el mismo tipo de matriz de transformación que LDraw y también consiste en archivos de texto que hace referencia a bibliotecas de ficheros "incluidos", por lo que hay muchas cosas que parecen familiares.

Ha habido varias herramientas a lo largo de los años que pueden llevar a cabo la conversión de LDraw a POV-Ray, pero mi favorita es LDView. Hace un trabajo excelente y producirá un buen render sin ningún esfuerzo adicional. Con un mínimo adicional de edición del archivo de POV-Ray, los resultados pueden mejorarse aún más. Por último, con una edición avanzada y los cambios en algunas de las bibliotecas principales, se pueden lograr resultados realmente sorprendentes.

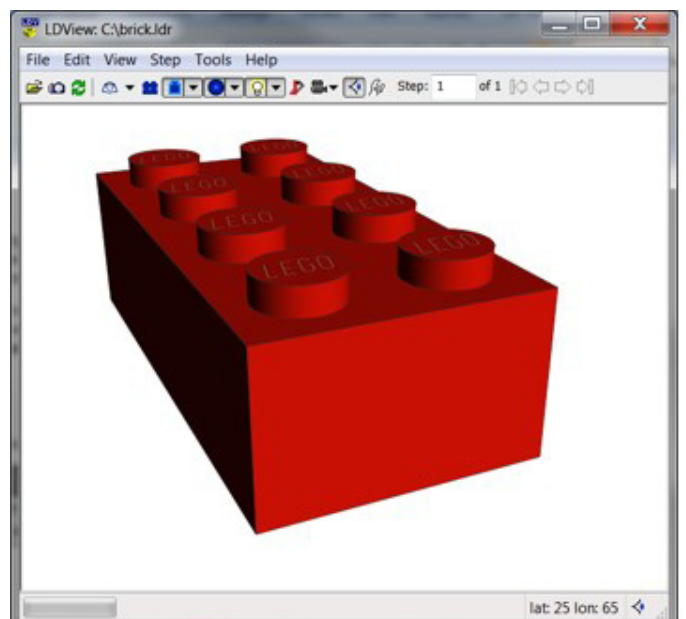
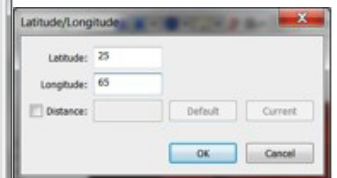
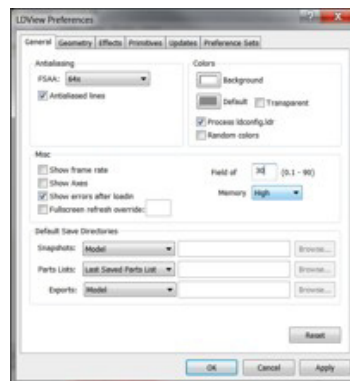
Vamos a trabajar con un ejemplo de principio a fin utilizando los ajustes más sencillos. Vamos a empezar con el modelo descrito arriba: un archivo que contiene un solo ladrillo rojo 2x4. Crea un archivo denominado brick.ldr que se parezca a esto:

```
0 Brick Render
1 4 0 0 0 1 0 0 0 1 0 0 0 1 3001.dat
```

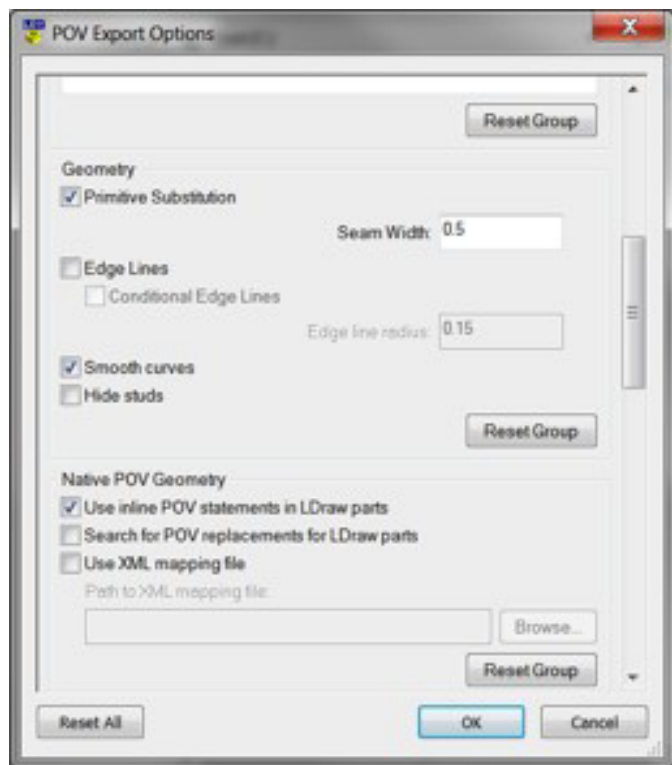
Si lo abres en LDView, deberías ver esto:



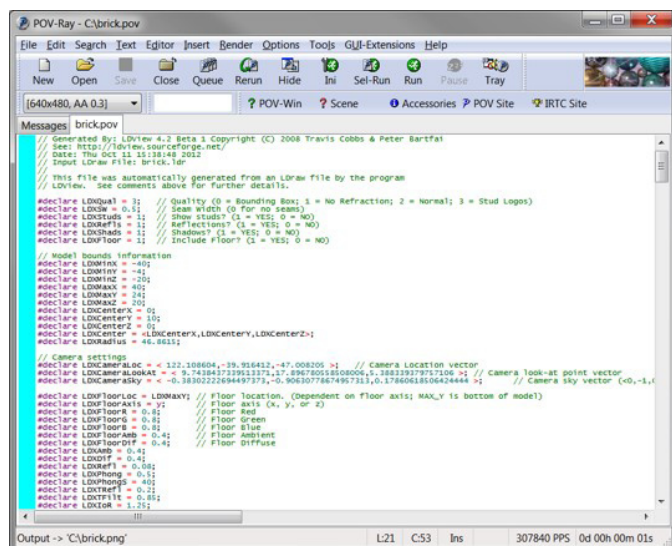
Al exportar desde LDView, también exportará la información de la vista, incluyendo la posición de la cámara. Vamos a hacer esto un poco más interesante cambiando la perspectiva y la rotación de la pieza un poco. Abre el panel de preferencias y cambia el campo de visión a 30 grados. Ahora haz que aparezca la ventana de vista con CTRL-9 para entrar una latitud y una longitud de 25 y 65. Después de eso, tu modelo debe tener este aspecto.



Ahora es el momento de exportar a POV-Ray. Puedes utilizar CTRL-E o seleccionar Exportar en el menú Archivo. Con ello se abre un cuadro de diálogo que le solicitará que des nombre al archivo. Vamos a llamarlo "brick". La extensión ". pov" se añadirá automáticamente. El "Type" debe ser "POV-Ray Scene File". Si haces clic en el botón "Options", podrás ver un montón de cosas que se pueden configurar sobre la conversión POV. Vamos a empezar con todos los valores predeterminados, solo asegúrate de cambiar "Quality" a "Include Stud Logos" y desplázate hacia abajo y desactiva la opción "Use XML Mapping File". Ahora puedes seleccionar Aceptar y guardar el archivo.



Ahora ejecuta POV-Ray 3.6. Si abres "brick.pov", deberías ver esto. Se verá un poco diferente en Mac.

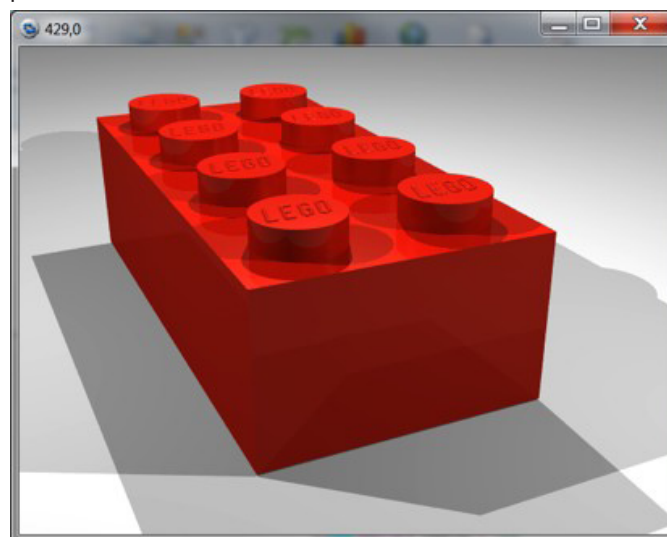


POV-Ray necesita saber el tamaño de la imagen que intentas crear, y en Windows esto es controlado en un archivo ini. Por suerte, solo tendrás que hacerte cargo de esto una vez y luego POV-Ray recordará tus ajustes. Queremos intentar un render a 640x480 con anti-aliasing. Para ello, seleccione "Edit resolution ini file" en el menú "Tools". Aparecerá un editor de

texto con un archivo llamado "QuickRes", y puedes agregar esto al archivo:

```
[640x480, AA 0.3]
Width=640
Height=480
Antialias=On
Antialias_Threshold=0,3
Output_File_Type=N
Output_Alpha=On
```

Guarda el archivo y sal del editor de texto. Esto agregará una opción llamada "640x480, AA 0.3" a un menú desplegable que se puede ver en la parte superior izquierda de la imagen anterior. (Ten en cuenta que puede que tengas que salir y reiniciar POV-Ray para que aparezca la nueva opción). Esta opción establece el alto y el ancho de la imagen, ajusta la configuración anti-aliasing (que suavizará los bordes), y genera un archivo PNG. No te preocupes por los detalles de la configuración anti-aliasing, solo tendremos que dejarlos siempre igual. Ahora todo lo que tienes que hacer es presionar el botón "Run", y unos segundos más tarde deberías tener tu primer render.



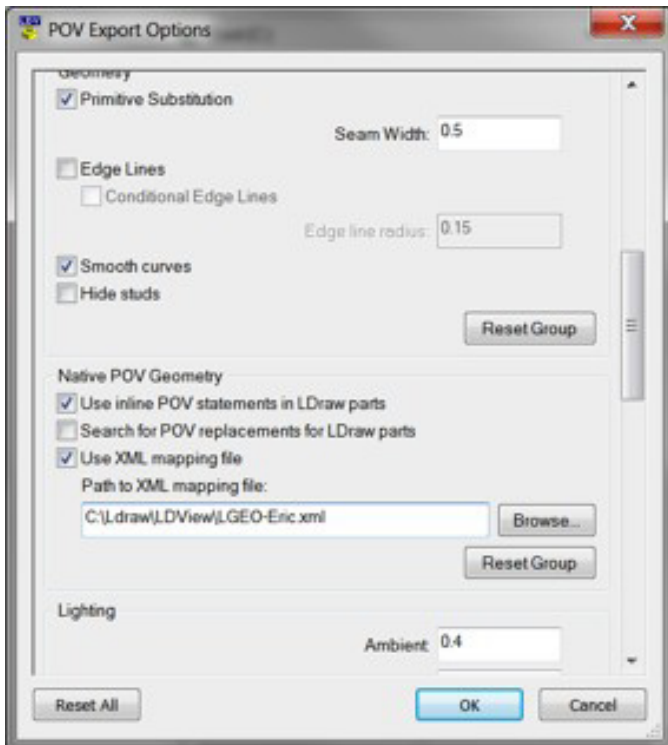
¡Es una gran mejora! Pero tiene un largo camino por recorrer. Probablemente te darás cuenta de que hay un montón de sombras superpuestas. También te darás cuenta de que, al tratarse de una conversión directa desde LDraw, los bordes son completamente afilados y cuadrados. LDView fue lo suficientemente inteligente como para hacer que los studs parezcan redondos por ti, en vez de poligonales y también agregó los logotipos del stud, pero aún no parece una pieza real. La siguiente mejora importante que podemos hacer es usar la biblioteca LGEO.

LGEO es una biblioteca de piezas de LEGO® hechas por Lutz Uhlmann usando la lengua nativa CSG de POV-Ray. Esto significa que no estaba limitado a solo triángulos y cuadriláteros, sino que podía hacer casi cualquier forma que él quisiera. El modeló las piezas con mucho más detalle incluyendo bordes suavemente curvados. La biblioteca LGEO de piezas POV-Ray también debería haber sido instalada por el instalador All-in-One. Para utilizarlo, es necesario asegurarnos de que un par de cosas están en su lugar. De nuevo, esto es algo que solo tendrás que hacer una vez. POV-Ray tiene que saber dónde encontrar la biblioteca LGEO. Vete a POV-Ray y selecciona "Edit master povray.ini" en el menú "Tools". Verás un archivo que tiene algunas rutas en la parte inferior. Asegúrate de que el instalador All-in-One ha añadido la ruta a la biblioteca LGEO aquí. Debe ser algo como esto. Si no está allí, es posible que tengas que agregarla manualmente

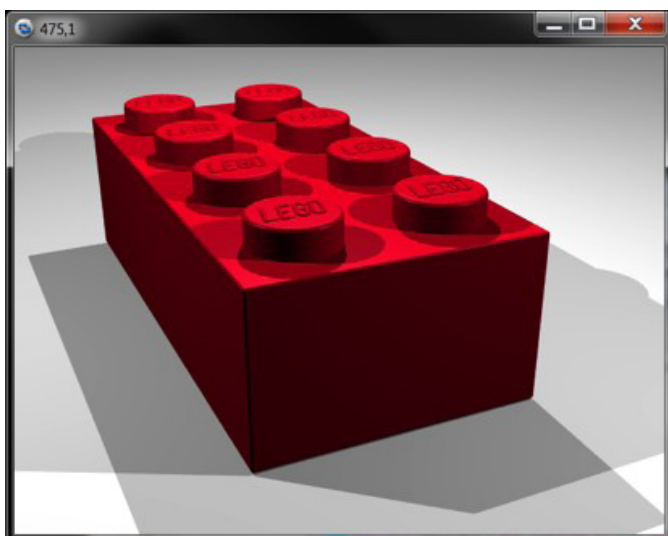
(la ubicación real de este archivo depende de dónde la has instalado en tu sistema).

Library_Path ="C: \ LDraw \ LGEO \ lg"

¿Recuerdas cuando exportamos desde LDView y anulamos la selección de "Use XML mapping file"? Ahora queremos seleccionarla de nuevo. LDView viene con un archivo XML que asigna cada pieza LDraw a su equivalente LGEO y realiza escalado o rotaciones para que estén alineadas. Todo lo que necesitas hacer es seleccionar el archivo de asignación que se incluye con LDView. Debe estar en tu directorio LDView. (Ten en cuenta que yo tengo mi propio archivo personalizado que hace algunos cambios y adiciones al archivo predeterminado, pero el archivo por defecto funciona bien.)

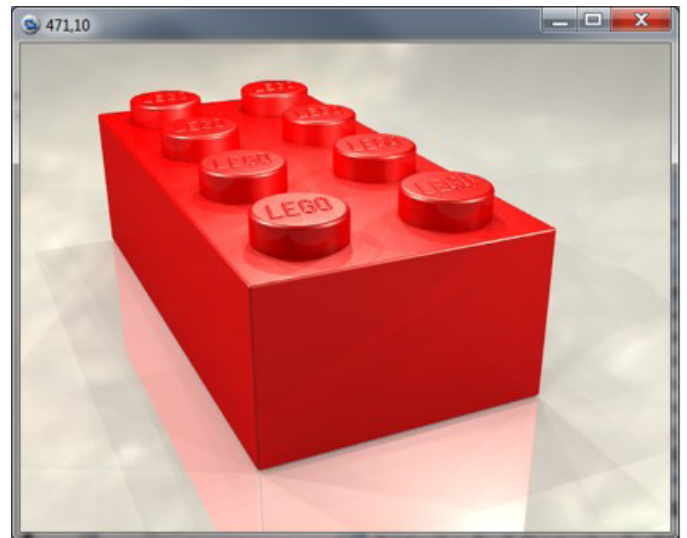


Esta vez vamos a llamar al archivo "brick-lgeo". Si has configurado todo correctamente, este archivo debe trabajar en POV-Ray, sin más cambios. Solo tienes que abrir y hacer clic en "Run". Ahora debes obtener esto.



Puedes ver los bordes redondeados, el color diferente, y el logo más claro. La biblioteca LGEO no contiene cada pieza de la biblioteca LDraw, por lo que en la práctica cuando exportas un modelo grande algunas de las piezas serán reemplazadas por sus equivalentes LGEO y algunas de ellas no lo serán. Por lo general, esto no es un problema.

Con las habilidades que has aprendido aquí, deberías ser capaz de hacer renders básicos. Aunque este tutorial haya podido parecer complicado, una vez que tienes todo configurado el proceso es muy rápido. Puedo crear el archivo de ejemplo que contiene el ladrillo, abrirlo en LDView, exportarlo, y renderizar la imagen de arriba, todo en menos de 30 segundos. En futuros tutoriales vamos a ampliar lo que has aprendido para que puedas entender el contenido del archivo de POV-Ray y modificarlo para incluir fuentes de luz más sofisticadas, utilizar otras características como la radiosidad y la iluminación de alto rango dinámico, añadir fondos, e incluso crear animaciones. Después de la próxima lección, deberías ser capaz de reemplazar la última imagen por esto.



Con el tiempo, deberías ser capaz de hacer esto.



¡Feliz renderizado!
#

