

Shader 02 – Simulating Diffraction

En el siguiente link, encontraran la implementación de un algoritmo que ayuda a simular el efecto de color de los CD o DVD, que se conoce como Diffraction.

<http://developer.nvidia.com/node/103>

El proyecto consiste en simular este efecto en RenderMonkey. Comentarios acerca de la implementación:

1. Aquí si tienen que usar el H, no obstante NO la normalicen, si normalizan el H los resultados son diferentes.
2. El código de la pagina esta en el vertex shader, ustedes deberán hacerlo en el fragment shader.
3. La siguiente línea `Mul((Float3x3)ModelViewMatrixIT, tangent);` es equivalente a `normalize(gl_NormalMatrix * rm_Tangent);`
4. Las únicas variables para crear el efecto que el usuario debe dar son d, r y hiliteColor.
5. La línea de código `Float4 anis = hiliteColor * Float4(c.x, c.y, c.z, 1);` pueden suplantarlos por `Float4 anis = hiliteColor * Float4(c, c, c, 1);` o `Float4 anis = hiliteColor * c.` Esto es porque "c" es float, no se por qué en el código en CG, usan "c" como si fuera un vector puede que sea un error.
6. En vez de $n < 8$, usen $n < 4$, obtuve mejores resultados visuales de esa manera, tambien pueden usar $n < 3$.

Además de implementar el código de la pagina web, deberán solucionar los siguientes problemas:

1. Con este proyecto se les da un archivo .obj y una máscara MaskCD.jpg, el archivo obj deberán colocarlo (antes de abrir el rendermonkey) en la siguiente dirección `C:\Program Files (x86)\AMD\RenderMonkey 1.82\Examples\Media\Models`, y la textura en `\Media\Texture`
2. La textura mascara les va ayudar para aislar la zona central y borde que no lleva diffraction.



La parte negra, es el área que debe tener el efecto en cuestión.

3. Deberán implementar la reflexión, esta reflexión también solo debe suceder en el área negra de la imagen.

Los datos que el usuario puede cambiar:

1. Posición o dirección de la luz (deberán implementar el sistema de iluminación en el Fragment Shader)
2. Color Ambiental
3. Color Difuso
4. Color de la luz
5. Intensidad de la luz
6. iR, índice de reflexión
7. iD, índice de diffraction (esta variable no está en el código), este controla la intensidad del brillo del efecto.
8. r,d datos para manipular el efecto de Diffraction.
9. hiliteColor, color utilizado en el shader.
10. 2 variables del tipo booleano, una que determina si hay o no reflexión, y otra que determina si hay o no diffraction.

Entrega

1. **El proyecto es individual para ser entregado el domingo 20 de mayo.**

Ejemplos para q tengan un punto de comparación.

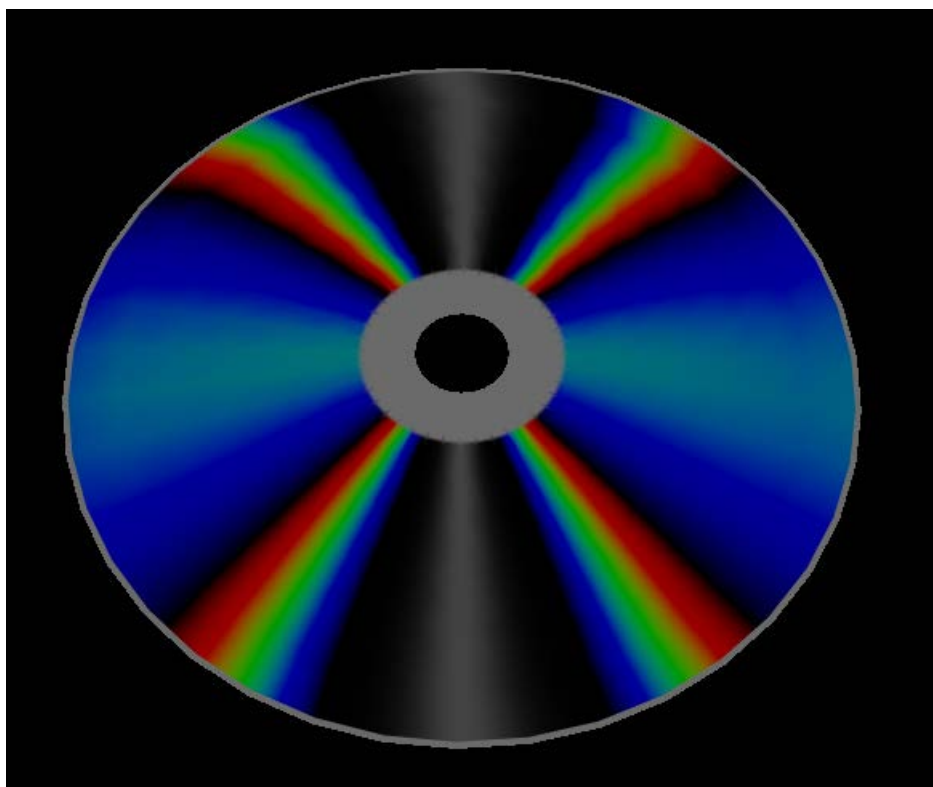
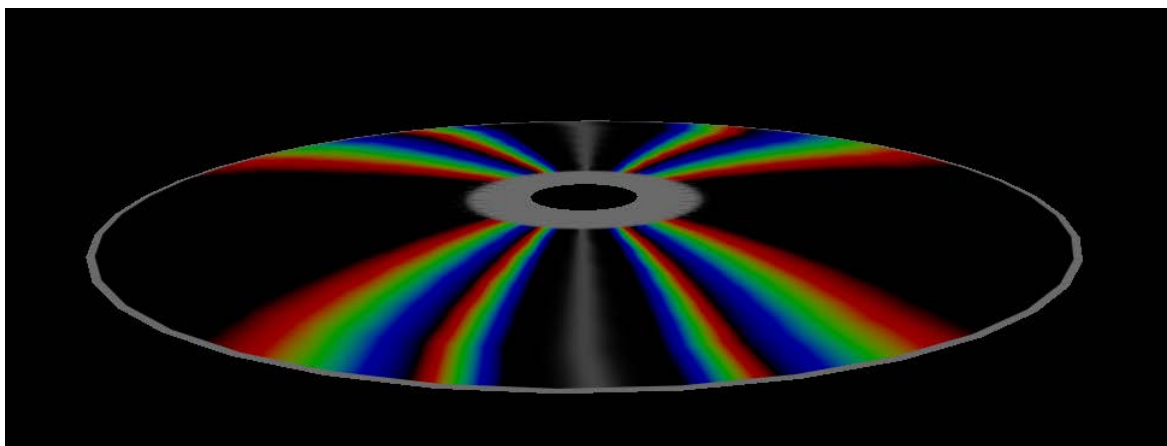
Luz puntual (0,100,0,1.0)

Color Difuso (Negro), Color Ambiental (Negro)

d = 2.55, r = 10, iD = 0.6, iR = 0.0

El n del algoritmo lo fije en $n < 3$.

HiliteColor (0.5,0.5,0.5).





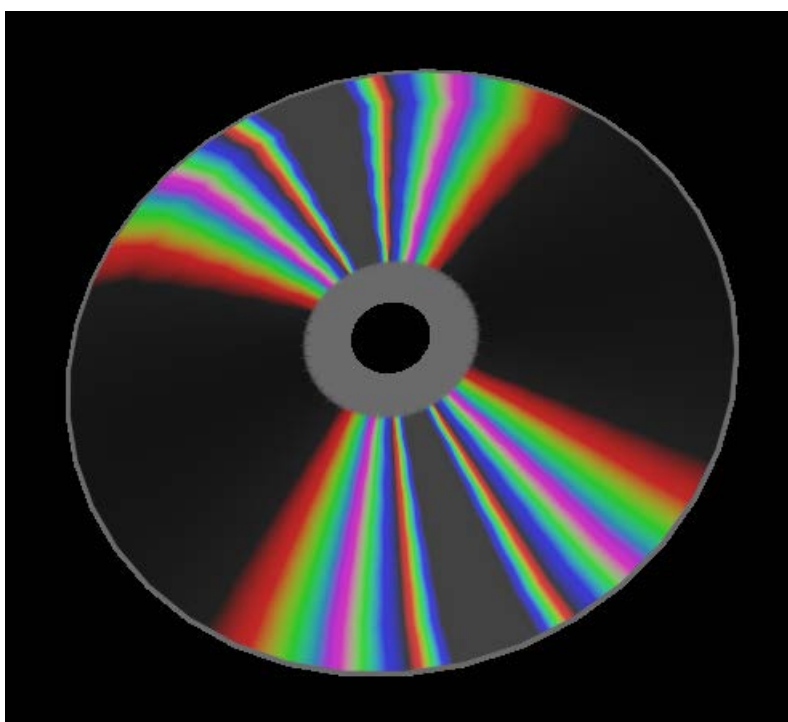
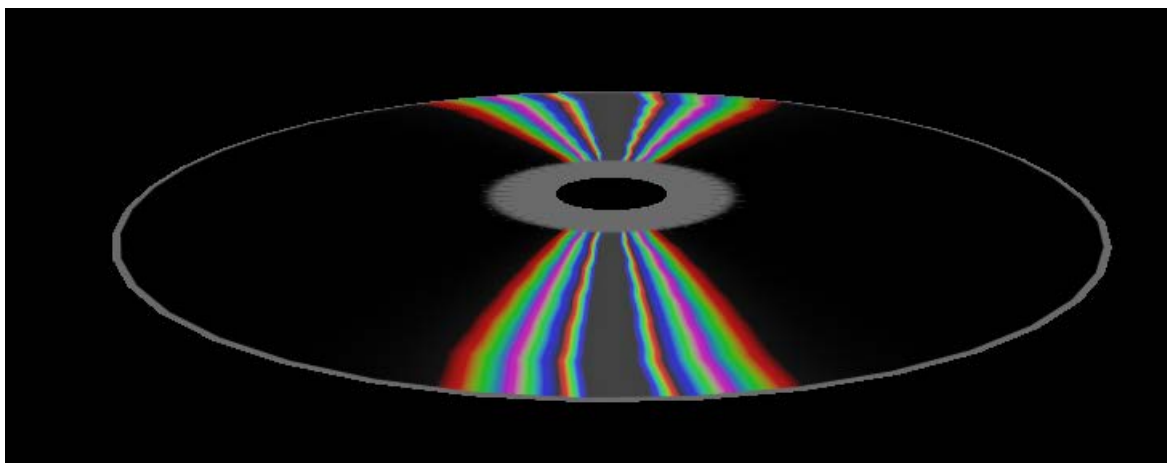
Luz puntual (0,100,0,1.0)

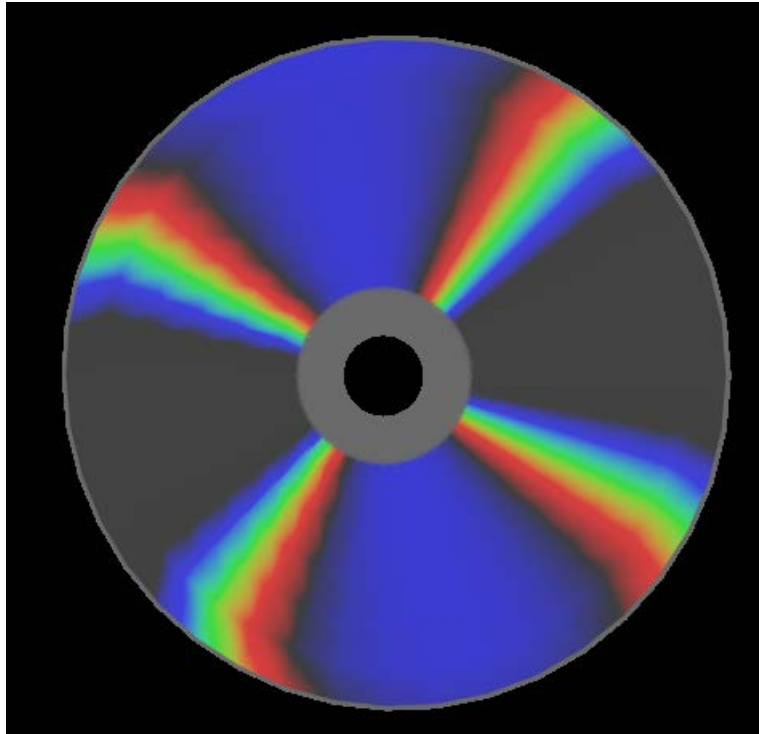
Color Difuso (Negro), Color Ambiental (Negro)

$d = 8$, $r = 0.54$, $iD = 0.6$, $iR = 0.0$

El n del algoritmo lo fije en $n < 4$.

HiliteColor (0.5,0.5,0.5).





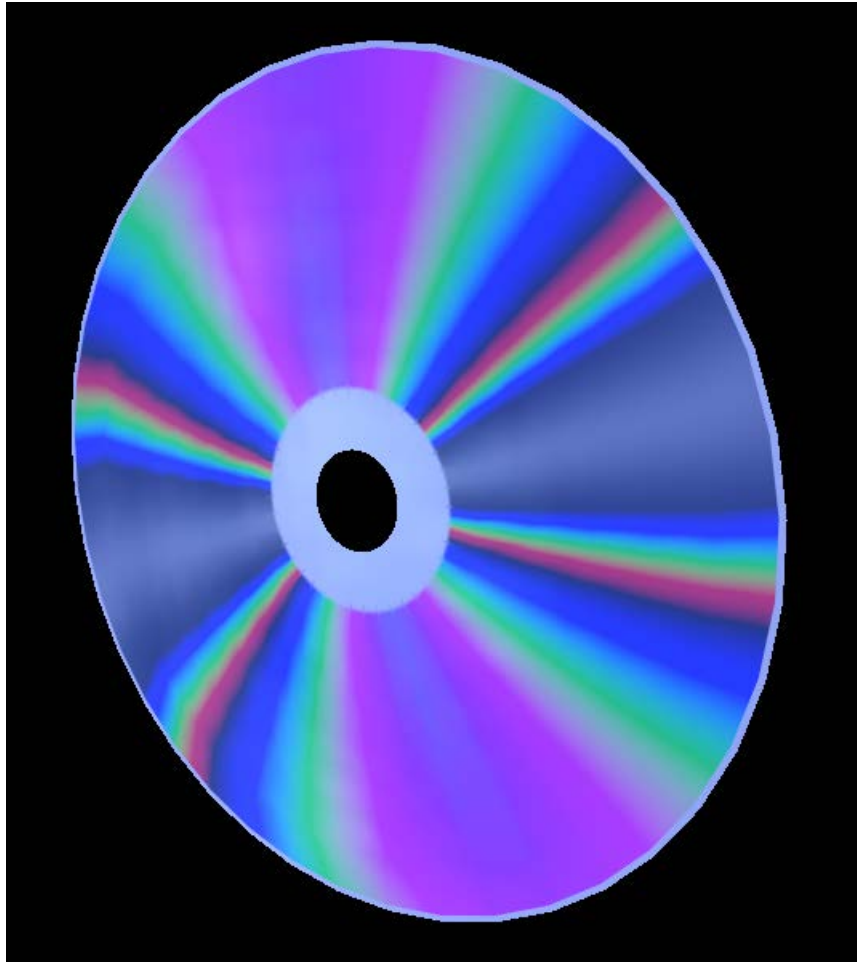
Luz puntual (0,100,0,1.0)

Color Difuso (Azul claro), Color Ambiental (Negro)

$d = 3$, $r = 5$, $iD = 0.5$, $iR = 0.4$

El n del algoritmo lo fije en $n < 4$.

HiliteColor (0.5,0.5,0.5).



Como verán el algoritmo es bien sensible a los valores de r y d y a la posición de la cámara. Fíjense que ni la parte central ni el borde tienen el efecto de difracción, para lograr esto deben usar la máscara de la imagen MaskCD.jpg.