

Modelamiento

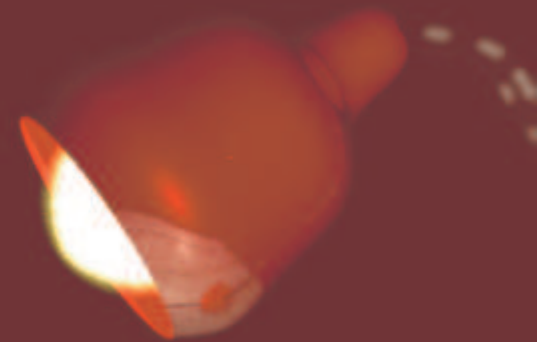
de

fuego



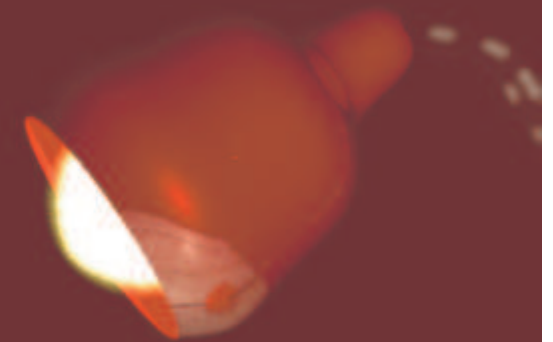
- El fuego es algo que siempre ha llamado la atención.

- Desde su descubrimiento se ha tratado de entender su naturaleza.



- ¿Cómo representar al fuego en una computadora?

- Inicios: Sólo dibujos 2D



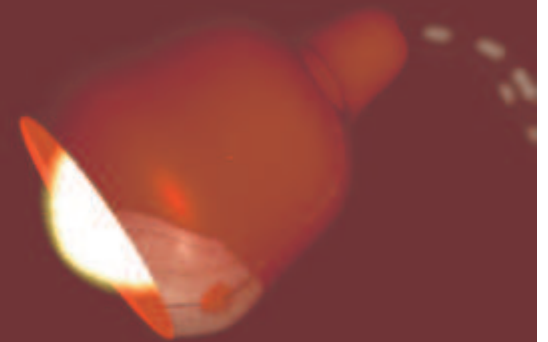
- Siguiente paso: sistemas de partículas.

- Un sistema de partículas permite modelar fenómenos complejos, "borrosos", abstractos, difíciles de producir con las técnicas convencionales de Rendering.



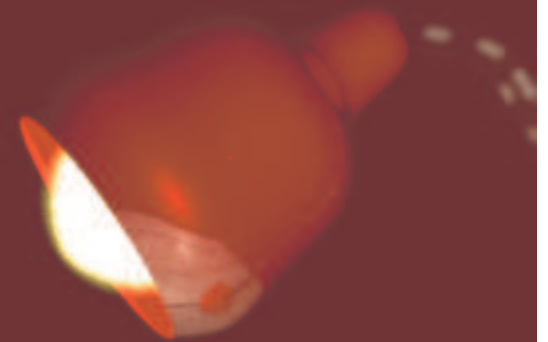
- Algunos ejemplos de usos de sistemas de partículas:

Modelar nieve



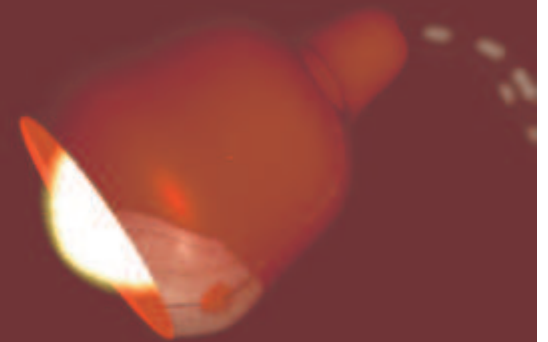
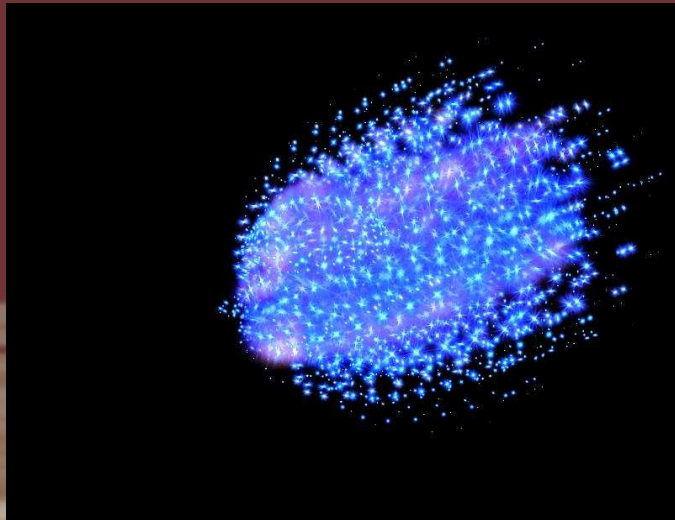
- Algunos ejemplos de usos de sistemas de partículas:

Modelar cabello o pelaje



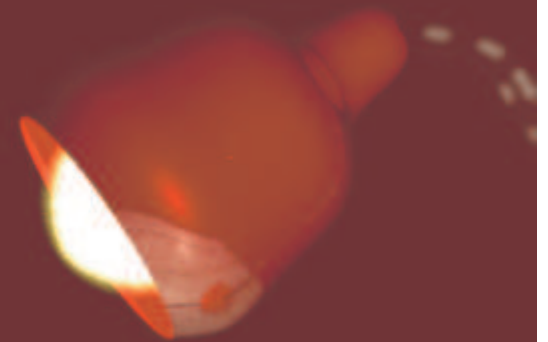
- Algunos ejemplos de usos de sistemas de partículas:

Modelar fenómenos abstractos



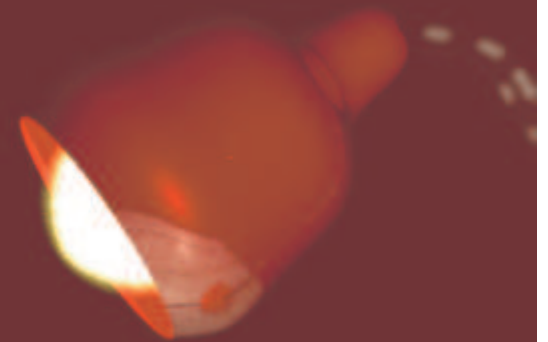
- ¿Cómo funciona un sistema de partículas?

- Fuente emisora: De alguna manera se genera una fuente de emisión de partículas.



- Se emiten partículas con propiedades como color, textura, velocidad, etc.

- Se define una cantidad de partículas que se emiten de la fuente por unidad de tiempo.



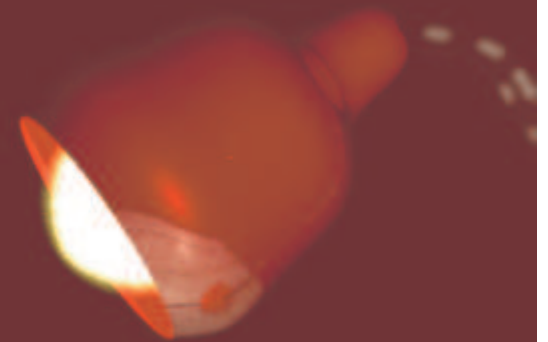
- Por cada cuadro de animación se hace una actualización del estado del sistema de partículas.

- Cada actualización por cuadro se divide en dos fases: Actualización de parámetros y renderización de la escena.



• Actualización de parámetros:

- Cantidad de partículas a crear.
- Posición desde donde aparecen.
- Inicialización de parámetros de las partículas: color, velocidad, alpha, etc.



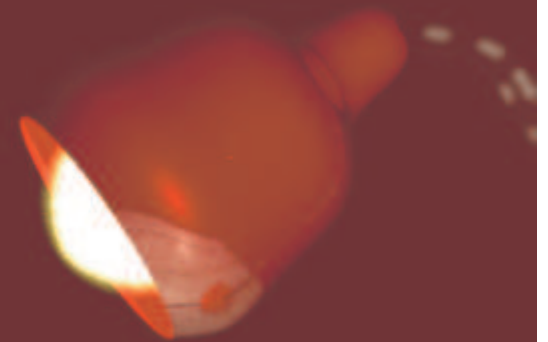
• Actualización de parámetros:

- Puede aplicársele alguna función a las partículas para hacer más realista la simulación.
- Puede ocuparse detección de colisión con objetos para algunos casos especiales.



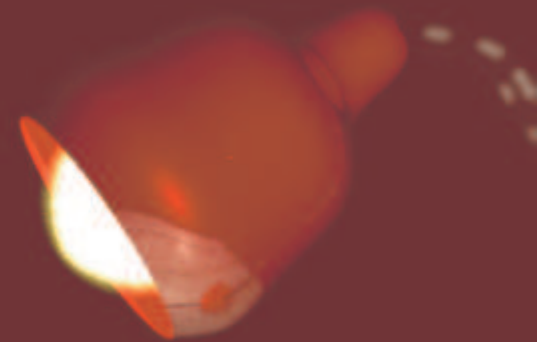
- Actualización de parámetros:

- Detección de colisiones entre partículas del sistema: Caro e inútil para efectos prácticos de la simulación.



•Renderización de la escena:

- Estado del sistema bien definido.
- Se puede aplicar alguna textura plana a cada partícula.
- Cada partícula puede renderizarse como un pixel, entre otras cosas.

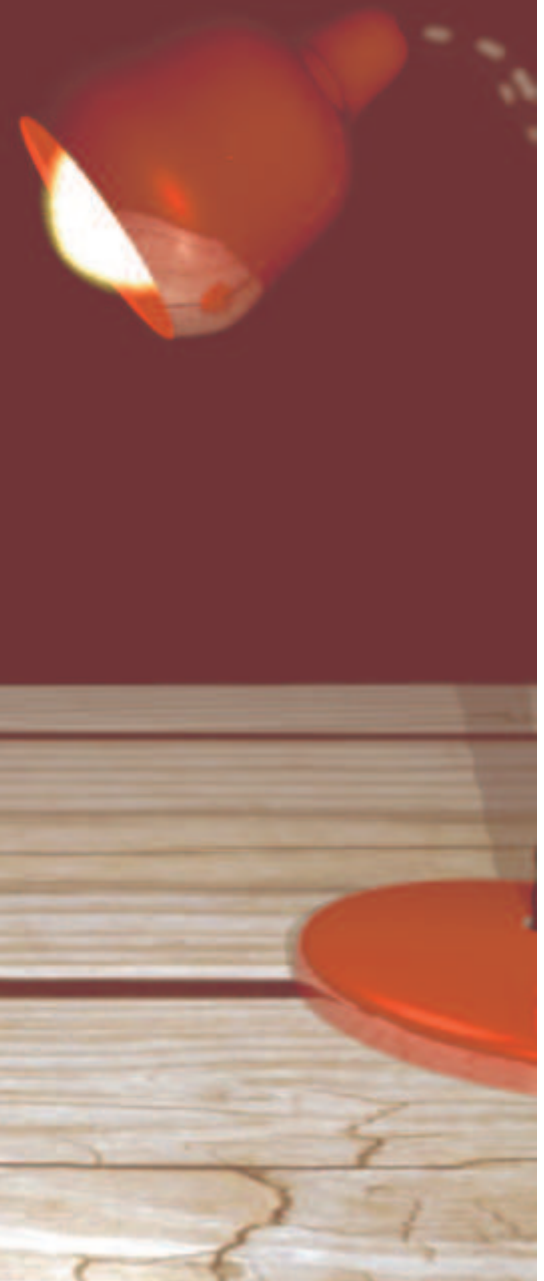


- Fuego en el Demo "Vulcan"

- Criatura inspirada en Balrog, de *Lord of the Ring*



©2003 NVIDIA Corporation. All rights reserved.



- Fuego en el Demo "Vulcan"

- Se creó para el lanzamiento de GeForce FX 5900 Ultra



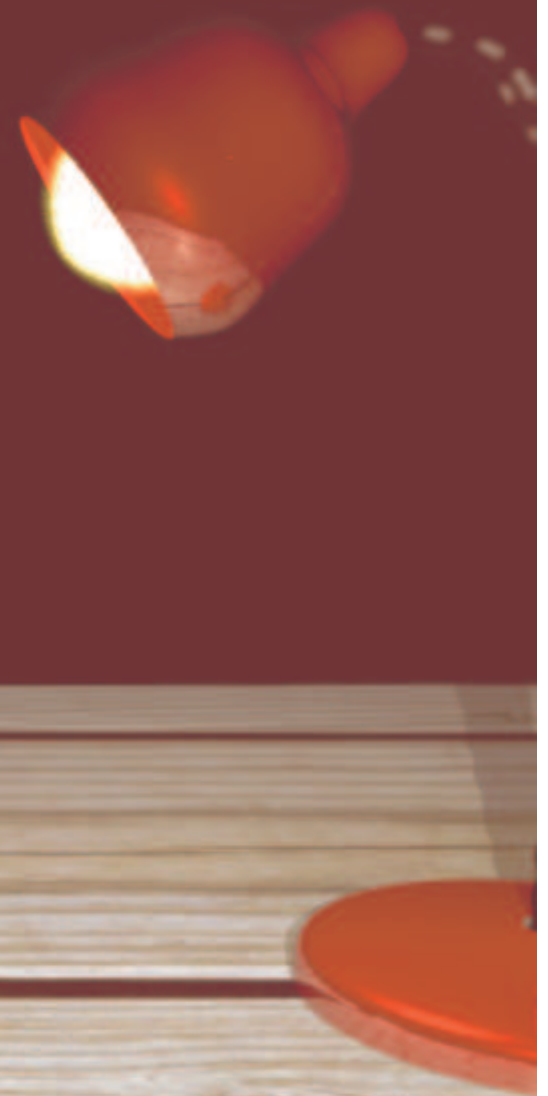
•Fuego en el Demo "Vulcan"

- Dos posibles soluciones para el modelamiento de el fuego:

 - Fully procedural flames*

 - Screen-space 2D distortion-based flames*

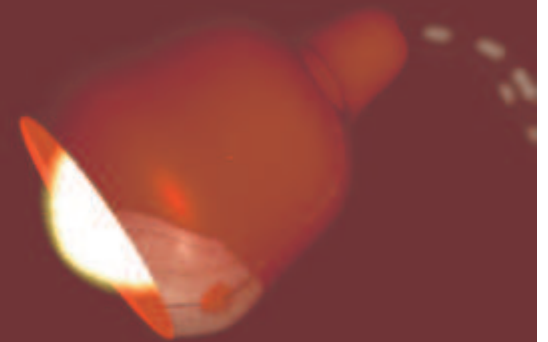
- Fully procedural flames: Para producir llamas bien definidas se necesitarían cientos de partículas y procesar todos esos vértices y pixeles cargan mucho el uso de la CPU y GPU.



•Fuego en el Demo "Vulcan"

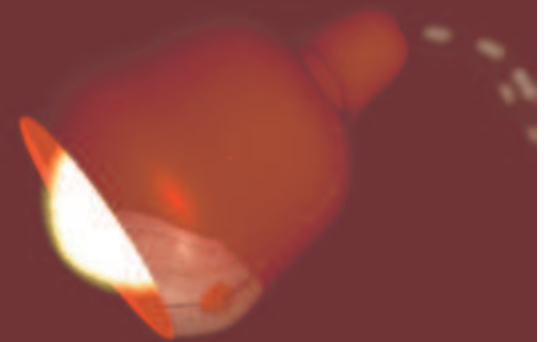
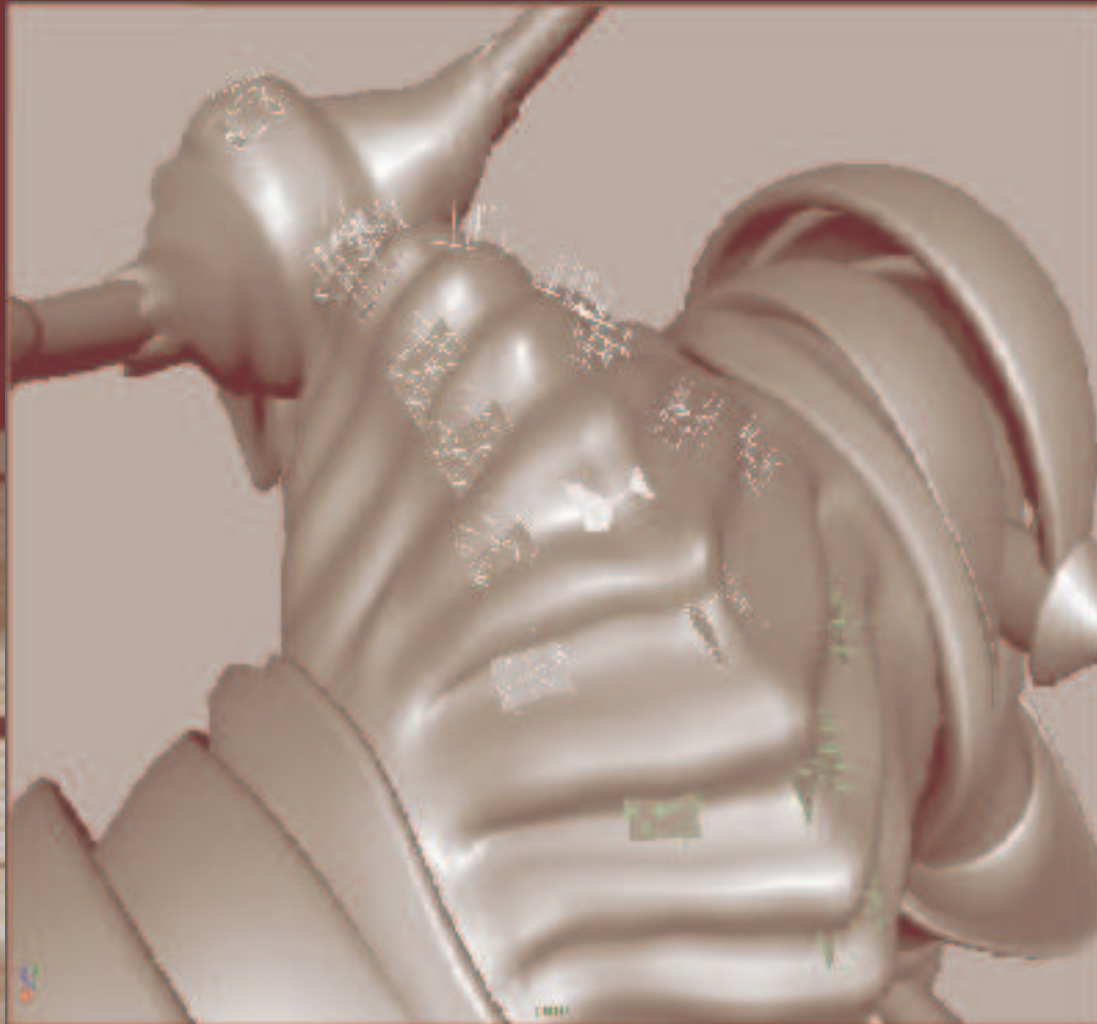
•*Screen-space 2D distortion-based flames* : utilizan una función de perturbación generada por la GPU que alteran la forma de una llama de modo de entregar movimiento casi real.

•Pese a lo buena que son ambas técnicas, falta algo de realismo: Se optó por usar Sprites video-texturizados, de modo de hacer el fuego más real.



- Fuego en el Demo "Vulcan"

- Se ubican los emisores en la espalda y cabeza de la criatura.



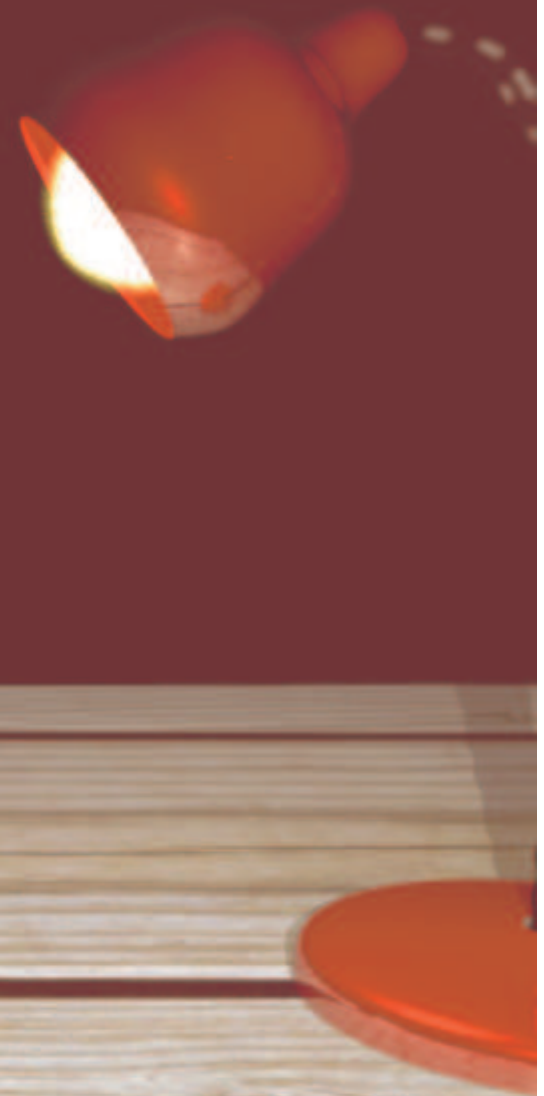
•Fuego en el Demo "Vulcan"

•Para animar el fuego, se utilizan 2 animaciones para fuego y una animación para el humo. Cada animación consta de 64 frames, ocupando entre las tres un total de 192 frames. Se dejan otras 64 frames libres para otros efectos, como chispas, etc.



•Fuego en el Demo "Vulcan"

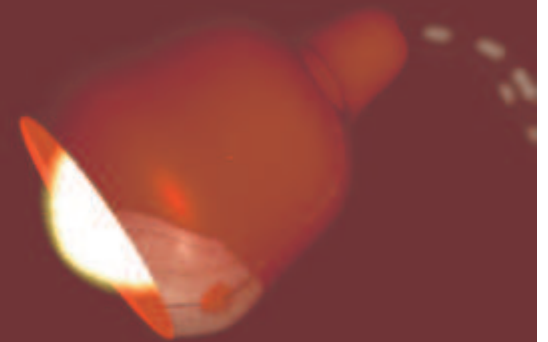
•Una textura 3D requiere que el numero de slices (frames en este caso) sea potencia de dos, por lo que el número de frames asignadas (4 animaciones de 64 frames) satisface dicha condición, ya que son 256 frames. Se había pensado en dar más detalle, pero tendrían que ser necesariamente 512 frames lo que sería muy costoso.



•Fuego en el Demo "Vulcan"

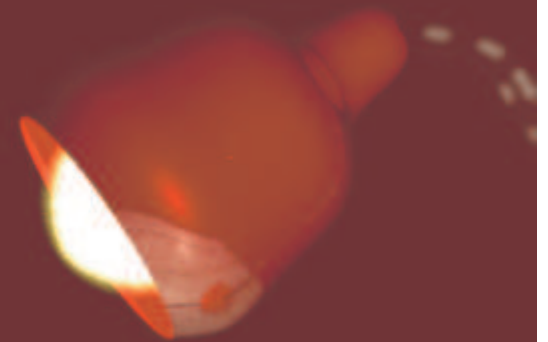
•Para el modelado del humo se pensó en usar la misma técnica para el fuego (sprites animados), pero es difícil encontrar características para modelar el humo, como la luz, el color, etc.

•Se decidió darle a las partículas de humo una rotación en el eje x, de modo que pareciera que el humo rotara en si. Muy barato de hacer.



•Fuego en el Demo "Vulcan"

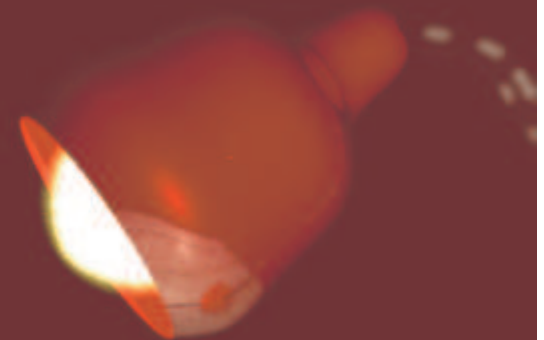
•Para animar el fuego, se utilizan 2 animaciones para fuego y una animación para el humo. Cada animación consta de 64 frames, ocupando entre las tres un total de 192 frames. Se dejan otras 64 frames libres para otros efectos, como chispas, etc.



•Fuego en el Demo "Vulcan"

•Para las animaciones, se encontró útil almacenar las llamas en una "textura 3D". Se requiere que cada lado sea una potencia de 2. En este caso se utilizó una textura comprimida 256x256x256 B8 G8 R8 A8 (16 MB).

•Sepuede optimizar, en caso de no poseer tanta memoria como 64x64x256



•Fuego en el Demo "Vulcan"

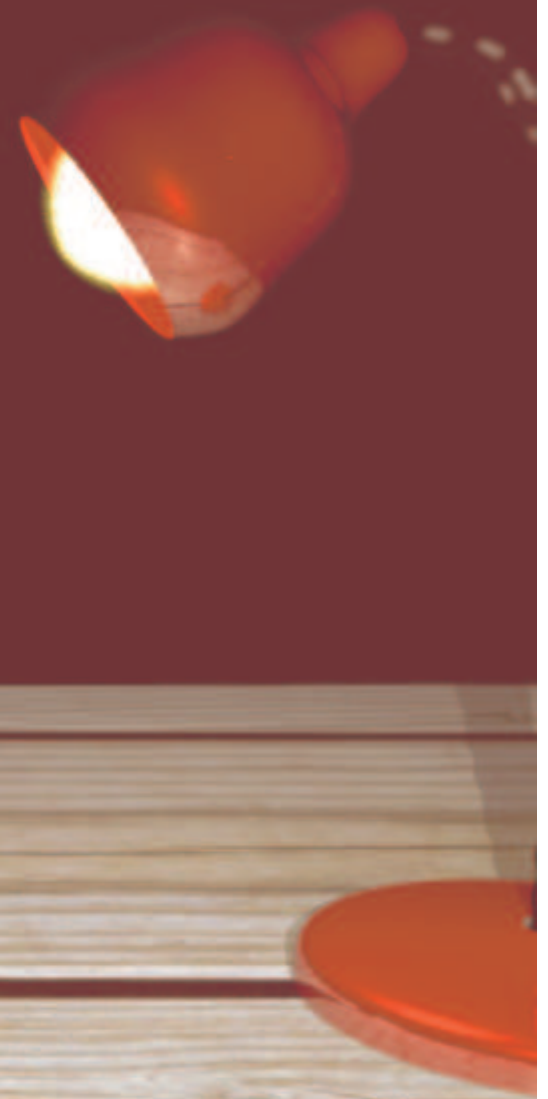
•Para mezclar el fuego con el humo, se le aplicó *additive blending* a las llamas. (una especie de saturación que permite crear brillo).

Sin embargo se veía muy sobrecargado y se aplicó el uso del canal alpha quedando un efecto bastante realista.



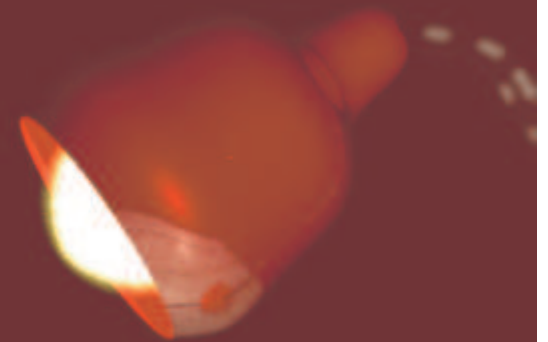
•Fuego en el Demo "Vulcan"

•Para el movimiento de las partículas, se asignó un "peso" a las partículas, de modo que si se encuentran cerca de la fuente emisora, están fuertemente ligadas a ella. En caso contrario, si las partículas están lejanas a la fuente, éstas se ven débilmente influenciadas y se mueven casi libremente por el aire, creando un efecto aún mas realista.



•Fuego en el Demo "Vulcan"

•Finalmente se aplican efectos de Blur en tiempo real, pero esto va más allá de lo que se quiere mostrar en esta presentación...



•Bibliografía

1. GPU Gems, capítulo 6: "Fire in the "Vulcan" demo".
2. http://en.wikipedia.org/wiki/Particle_system

