

FISICA DE JUEGOS

M.C. ARNULFO ZEPEDA NAVRATIL

CENTRAL TECHNOLOGY GROUP

WARGAMING.NET

a_zepeda@wargaming.net

arnulfo@hotmail.com

COMERCIAL (VIL)

- WARGAMING.NET
 - Empresa establecida en 1998
 - Creadores de
 - World of Tanks
 - World of Warships
 - World of WarPlanes
 - Oficinas de desarrollo en Seattle, Chicago, Baltimore, Austin, Kiev, Minsk, San Petersburgo y Sydney
 - Oficinas de apoyo en San Francisco, Tokyo, Seul, Paris, Berlin, Larnaka
 - 3000 empleados a nivel global

SIMULACIONES

- Simulación de dinámica de molecular
- Simulación de efectos electromagnéticos
- Simulación de elementos finitos (COMSOL)
 - Óptica
 - Termodinámica
 - Acústica
- Simulaciones virtuales con dinámica, generación óptica visual , reacciones interactivas en tiempo real

VIDEOJUEGOS

- Simulación
- Modelando un Mundo Virtual
 - En cuanto a su dinámica
 - En cuanto a su apariencia
 - En cuanto a sus reacciones
- Interactivas
- En Tiempo Real

ESTRUCTURA DE UNA SIMULACIÓN

- Entrada de datos/condiciones iniciales
- Bucle (loop) de simulación por cada ΔT
 - Aplicación de ecuaciones físicas a cada elemento
 - Mecánica / EM / Termodinámica
 - Correcciones por desviaciones de simulación
 - Repetir hasta llegar a condiciones finales
 - Tiempo total de simulación
 - Invariancia de condiciones finales (estabilidad)
 - Obtención de otras variables o estadísticas dependientes
 - Otros

ESTRUCTURA DE UN VIDEOJUEGO

- Entrada de datos/condiciones iniciales
 - Terreno / Geometrias-Objetos / Equipos
- Bucle (loop) de simulación por cada ΔT
 - Obtención de entradas interactivas
 - Aplicación de entradas a objeto(s) controlad(os)
 - Dinámica de animación y física de objetos/actores
 - Respuesta de IA de objetos a nuevas entradas o seguimiento de estrategias
 - Respuesta de Audio
 - Generacion de Imagen
- Repetir hasta llegar a condiciones finales



DIFERENCIA ENTRE SIMULACIONES CIENTIFICAS Y VIDEOJUEGOS

- Exactitud / Comparación con el experimento
- "Suficientemente bueno"
 - Visualmente
 - En comportamiento dinámico
 - En reacción al jugador

AREAS EN DONDE HAY APLICACIÓN FÍSICA EN VIDEOJUEGOS

- Generación de imagen (Rendering)
- Dinámica
- Interacción

GENERACIÓN DE IMÁGEN (RENDERING)

- En tiempo real
 - Videojuegos para mobil / PC / Consolas
 - 30 o más usualmente 60 fps, dicta vel simulación
 - AR/VR requiere más , 120 fps
 - A 30 fps se dispone de 33 ms para todo el paso completo de simulación
 - A 4GHz, 33ms = 13 millones de ciclos

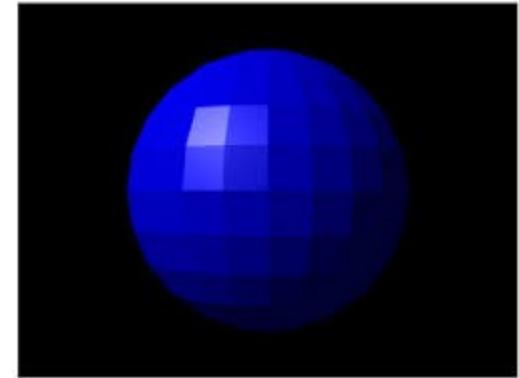
Fuera de linea

- No hay limite de tiempo
- Peliculas animadas, efectos especiales

FLAT/GOURAUD/PHONG/PIXEL SHADING

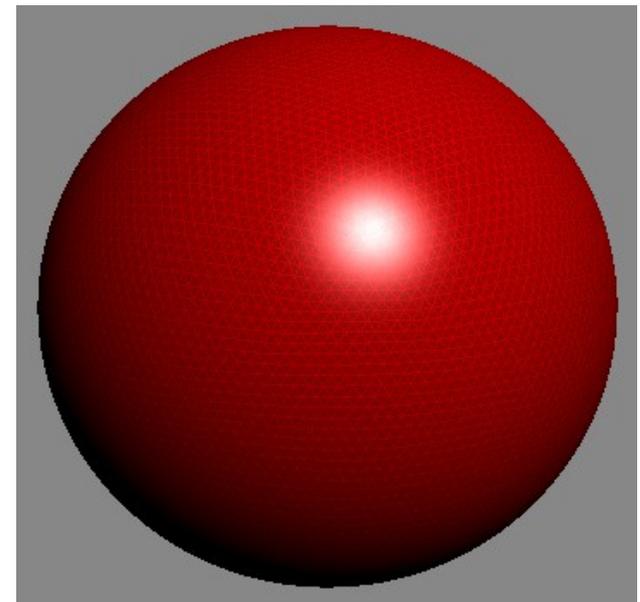
Flat shading

Cada cuadro o triangulo adquiere un solo color individual. Cada color es calculado Usando una ecuacion de iluminaci3n, como la ecuaci3n Phong-Blinn

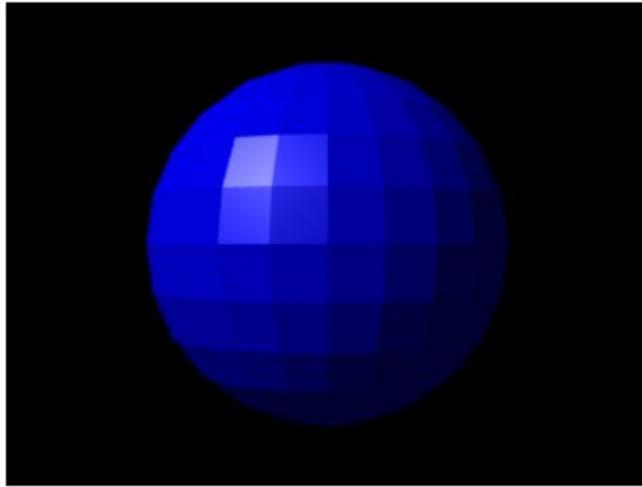


Gouraud shading

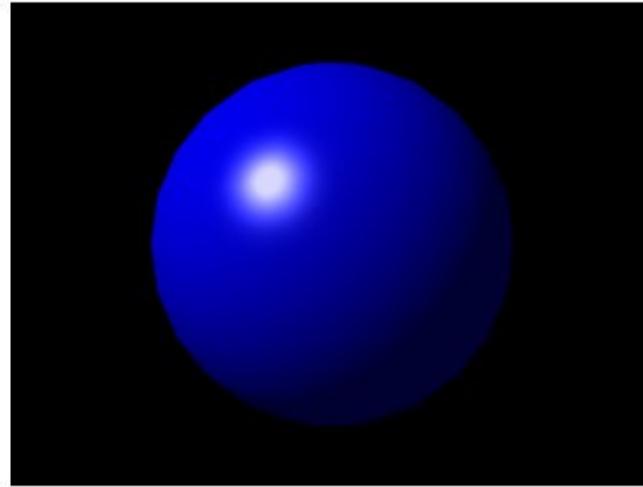
Cada triangulo que conforma la superficie recibe un color individual, y los colores interiores del triangulo se obtienen por interpolacion



Phong Shading



FLAT SHADING



PHONG SHADING

Cada pixel (visible) tiene un color calculado usando una ecuación de iluminación

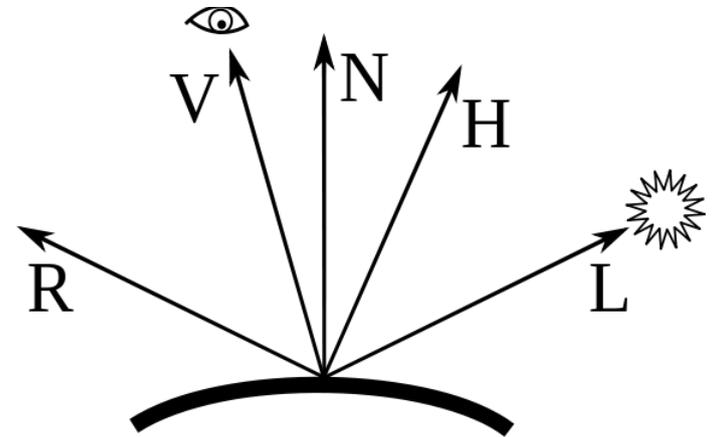
ECUACIÓN DE ILUMINACIÓN BLINN-PHONG

$$I_p = k_a i_a + \sum_{m \in \text{lights}} (k_d (\hat{L}_m \cdot \hat{N}) i_{m,d} + k_s (\hat{R}_m \cdot \hat{V})^\alpha i_{m,s}).$$

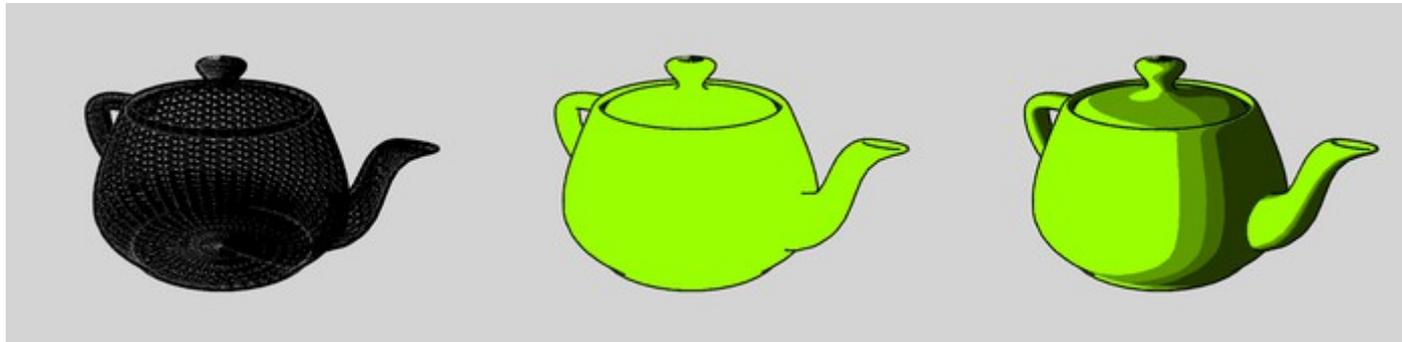
K = propiedades material/superficie
(ambientales[a], difusas[d], especulares[s])

I = propiedades fuente de luz

Alpha = propiedad interacción luz-material



TECNICA DE ILUMINACIÓN CEL SHADING



X



=



ECUACIÓN DE ILUMINACIÓN COOK-TORRANCE

$$r_s = \frac{F \times D \times G}{\pi (\mathbf{n} \cdot \mathbf{l})(\mathbf{n} \cdot \mathbf{v})}$$

$$F_\lambda(u) = f_\lambda + (1 - f_\lambda)(1 - (\mathbf{h} \cdot \mathbf{v}))^5$$

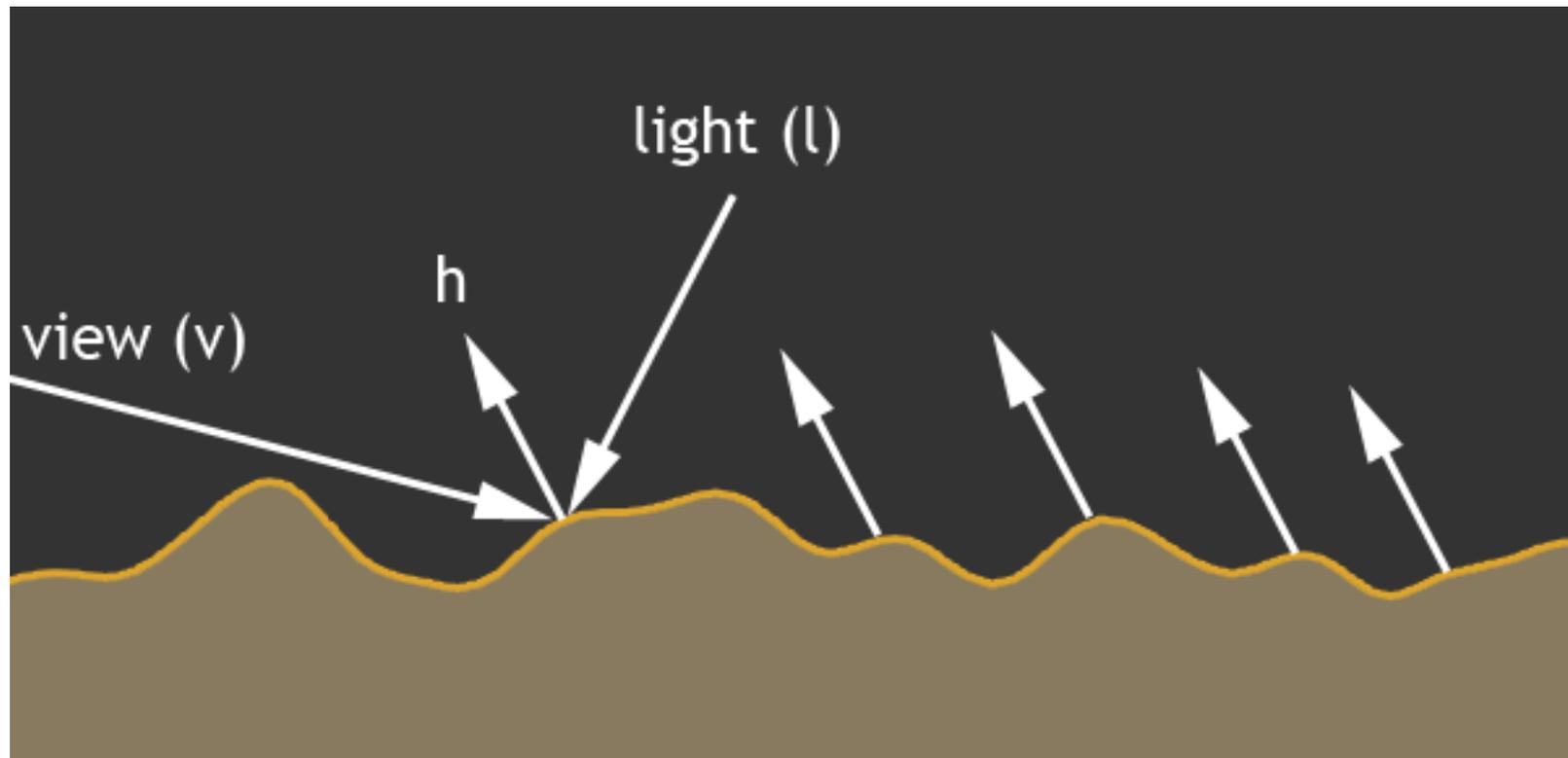
TERMINO DE FRESNEL

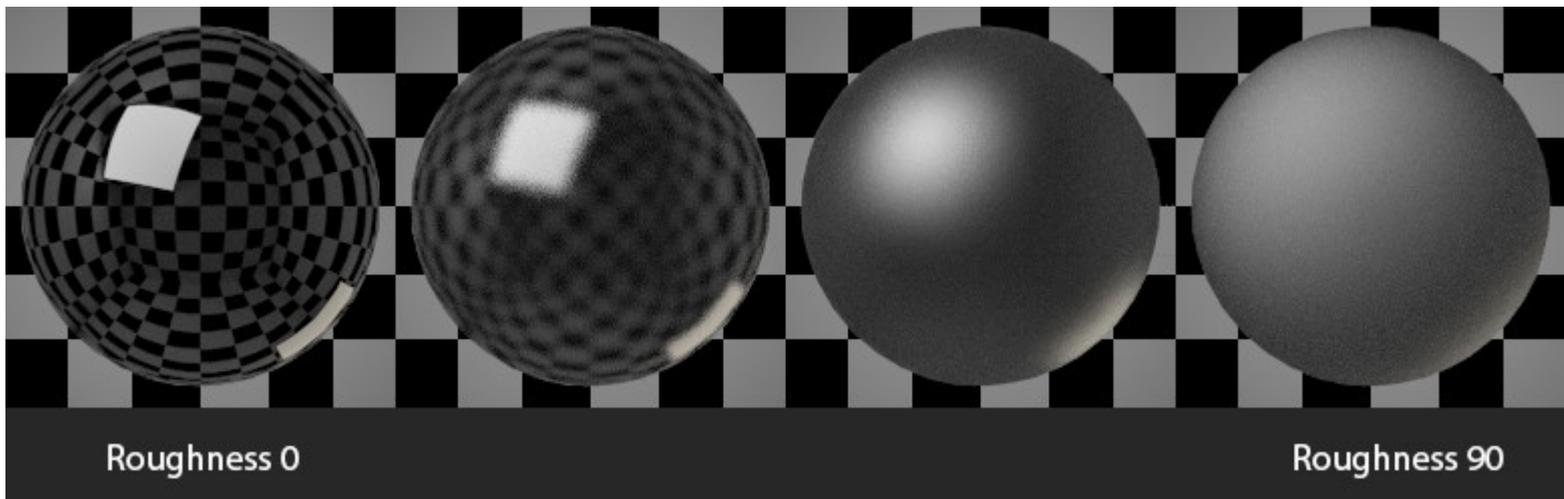
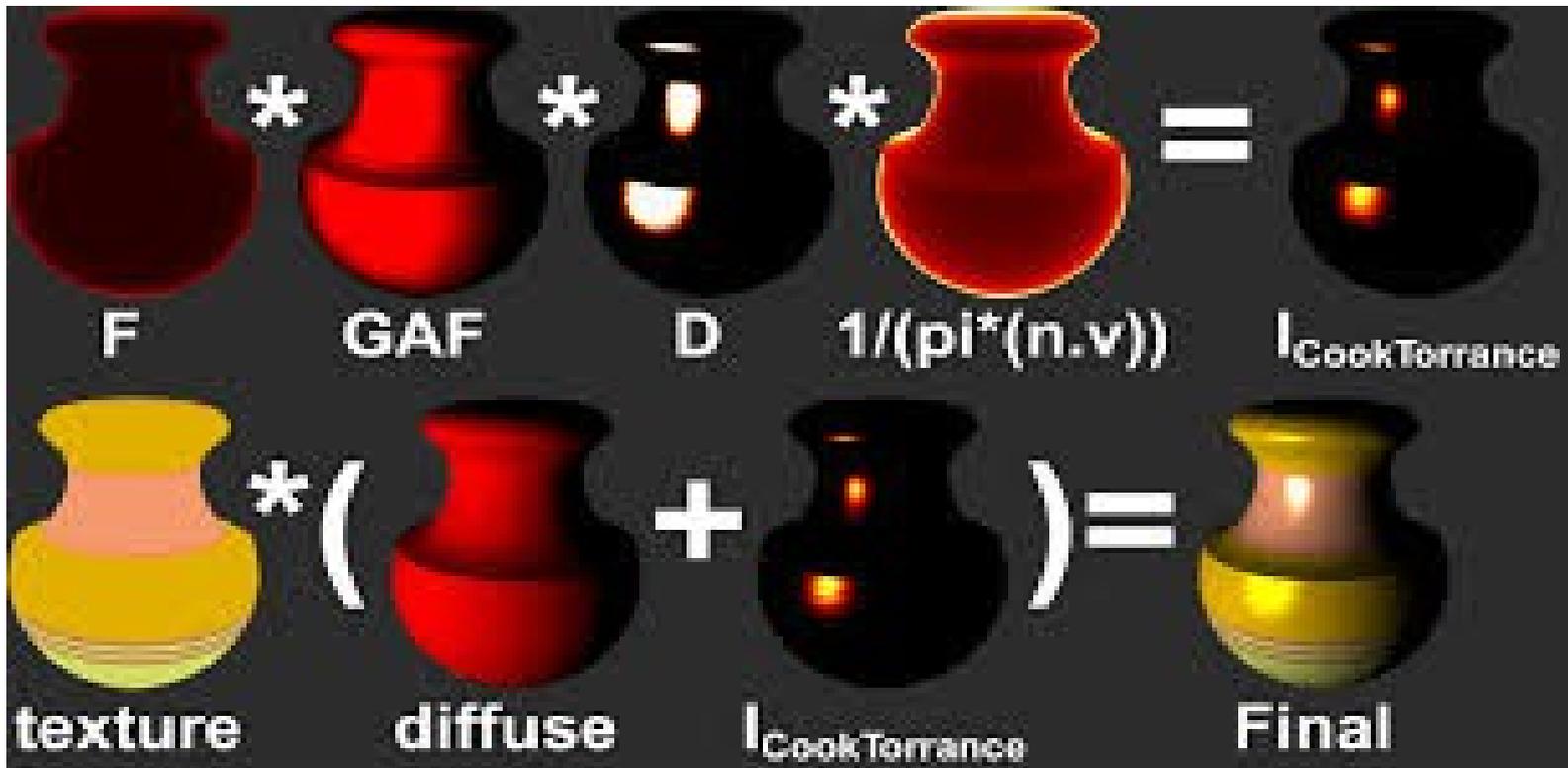
$$D = \frac{1}{\pi m^2 \cos^4 \alpha} e^{-\left(\frac{\tan \alpha}{m}\right)^2} = \frac{1}{\pi m^2 (\mathbf{n} \cdot \mathbf{h})^4} e^{\left(\frac{(\mathbf{n} \cdot \mathbf{h})^2 - 1}{m^2 (\mathbf{n} \cdot \mathbf{h})^2}\right)}$$

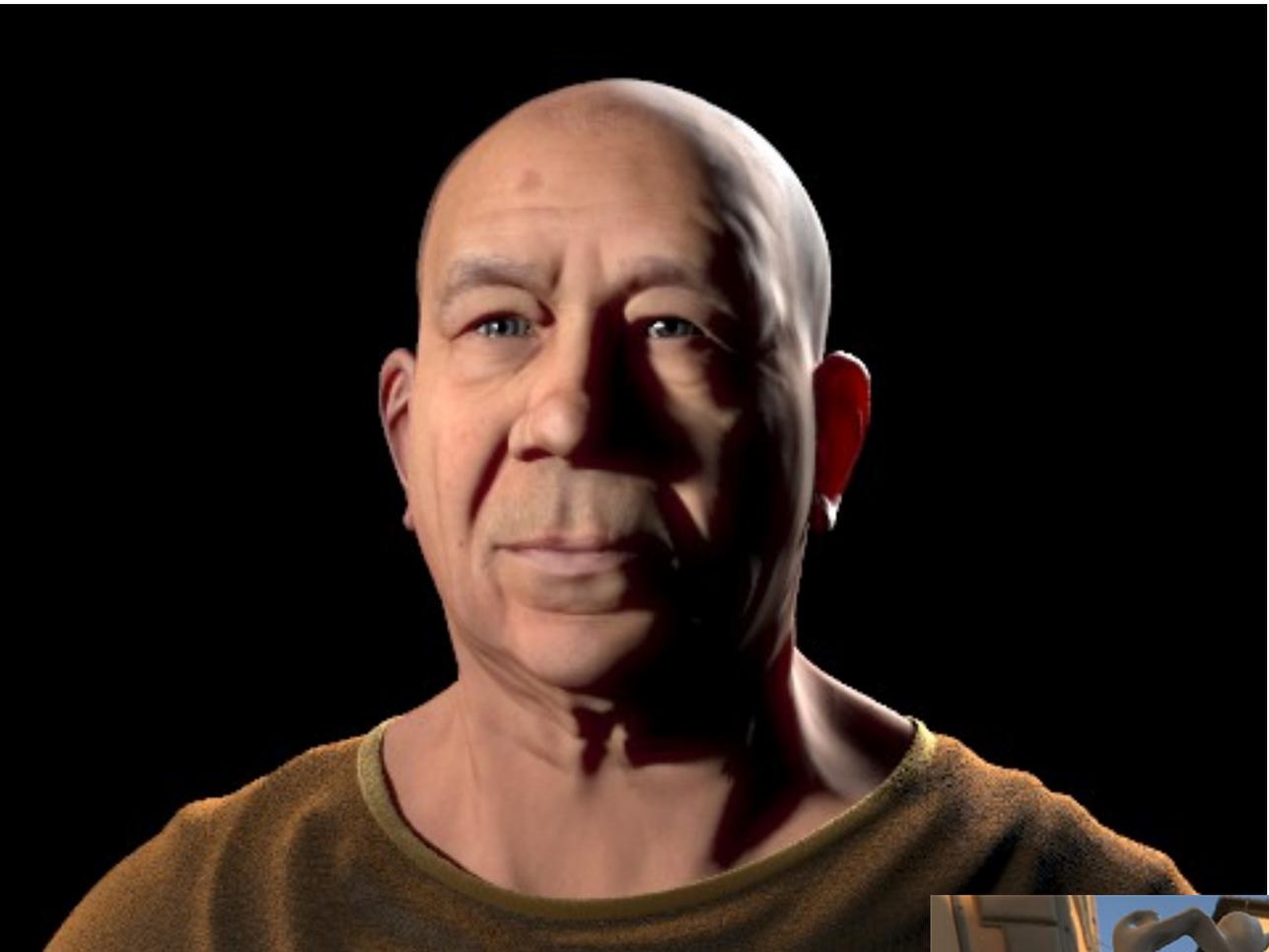
DISTRIBUCION DE
MICROFACETAS

$$G = \min\left(1, \frac{2(\mathbf{n} \cdot \mathbf{h})(\mathbf{n} \cdot \mathbf{v})}{\mathbf{v} \cdot \mathbf{h}}, \frac{2(\mathbf{n} \cdot \mathbf{h})(\mathbf{n} \cdot \mathbf{l})}{\mathbf{l} \cdot \mathbf{h}}\right)$$

FACTOR DE
ATENUACION
GEOMETRICA







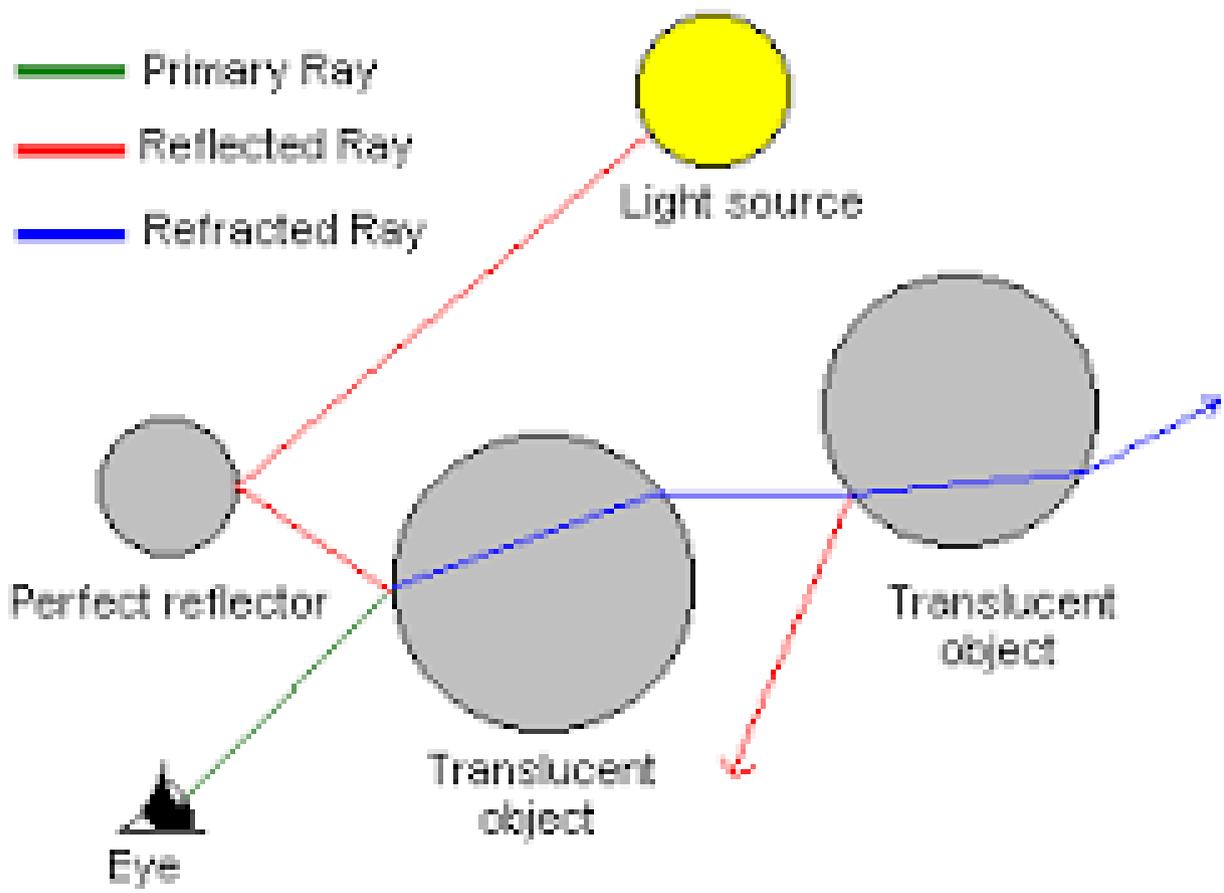
IMPACTO DE GPUS

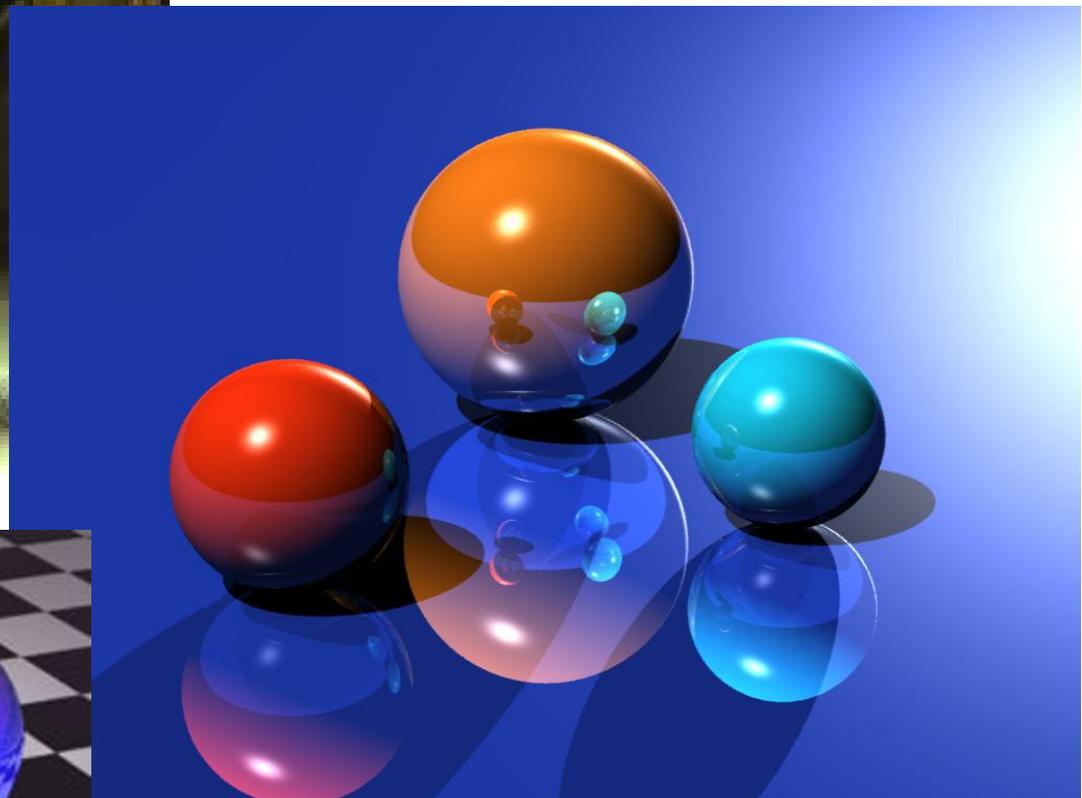
- Antes de los GPUS : Todo en el CPU
- Interpolación dentro del triangulo : por sw
- Ahora :
 - Interpolación automática
 - Shaders de vertices
 - Shaders de pixeles
- Nvidia GTX 1080 = 2560 unidades paralelas
- Nota adicional : GPUs son tan poderosos como para hacer compúto científico hoy en dia

RAY TRACING – TRAZO DE RAYOS/FOTONES

- Técnica de iluminación global
- Sin embargo hay mucho trabajo en progreso para usarlo como técnica de tiempo real
- Idealmente, deberíamos de tomar a cada fuente de luz (fotones), calcular cuantos/cuales son emitidos y en que angulos (las fuentes son siempre homogeneas), en que frecuencias y luego seguir a cada foton mientras interactua con todos los objetos presentes, reflejandose y refractandose, hasta llegar a los ojos del observador

- Problema : La mayoría (99.99%) de los fotones nunca llegarán a nuestros ojos.
- Consecuencia : La simulación sera costosísima
- Alternativa (primera optimización) :
 - Hacer el cálculo inverso
 - O sea, asumir que el fotón ya llego al ojo desde cierta dirección. Ahora, calcular de donde debe haber venido.
 - En cada superficie que se encuentre en dicha busqueda, calcular fotones reflejados y refractados





- Segunda optimización :
 - Calcular bonches (bundles) de fotones que provienen de aproximadamente la misma dirección.
 - Todos ellos rebotarán (hasta cierto grado) de las mismas superficies, permitiendonos hacer calculos que se pueden compartir/aprovechar entre varios de ellos
 - En cierto momento los bonches necesitan ser separados para seguir distintos caminos

RADIOSITY

- Técnica de iluminación global
- No aplicada usualmente a tiempo real
- Calcula transmisión de energía (EM) entre superficies
- La radiosidad es la energía por unidad de área que sale de una superficie por unidad discreta de tiempo y es la combinación de energía emitida y reflejada

$$B(x) dA = E(x) dA + \rho(x) dA \int_S B(x') \frac{1}{\pi r^2} \cos \theta_x \cos \theta_{x'} \cdot \text{Vis}(x, x') dA'$$

donde:

donde:

$B(x) dA$ es la energía total que sale de una pequeña área alrededor del punto x

$E(x) dA$ es la energía emitida

$\rho(x)$ es la reflectividad del punto, dando energía reflejada por unidad de área

, multiplicando la energía incidente por unidad de área

(energía total que llega de otros pedazos de área)

S denota que la variable de integración x' va sobre todas las superficies en la escena

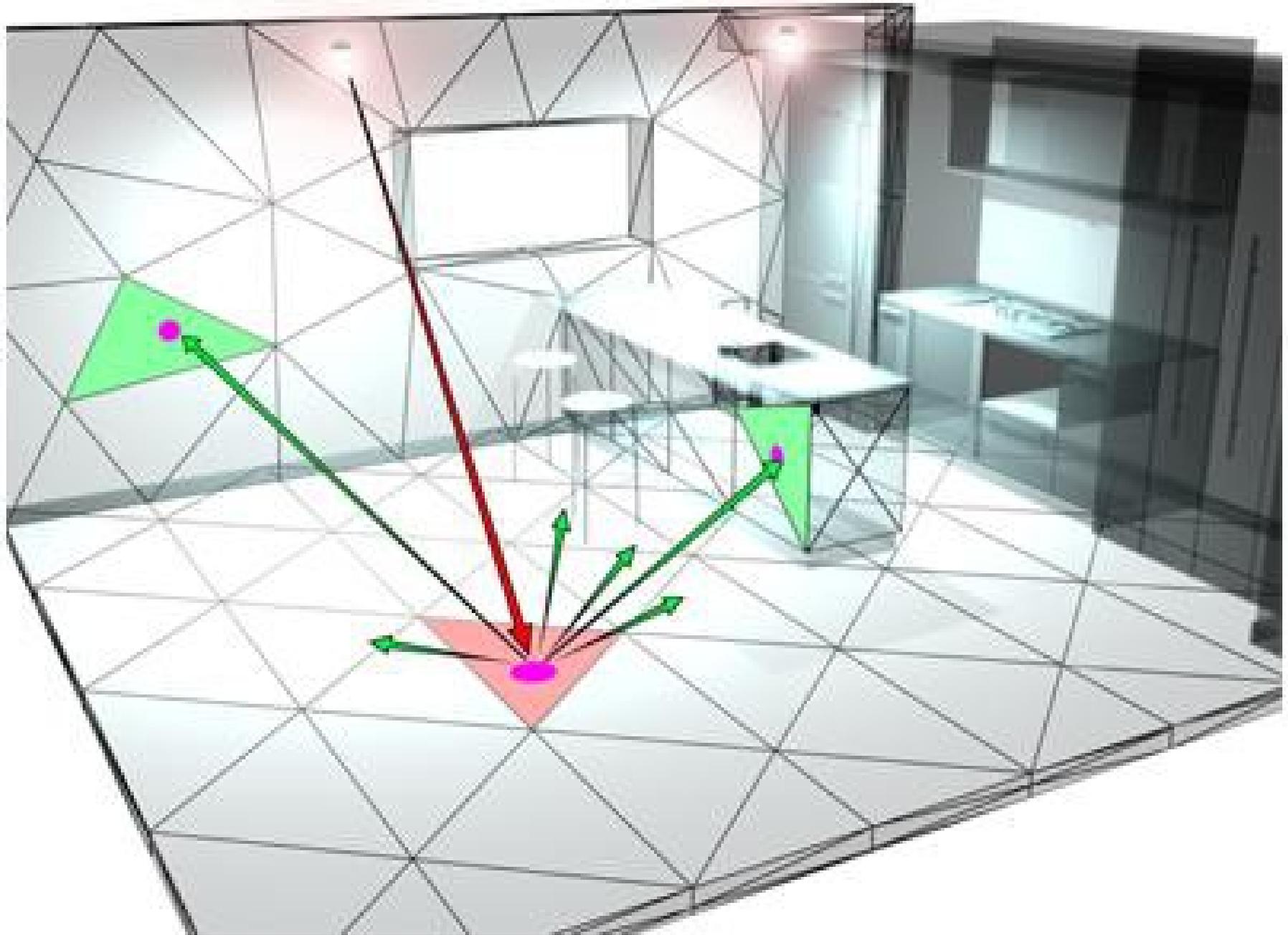
r es la distancia entre x y x'

θ_x y $\theta_{x'}$ son los ángulos entre la línea que corre de x a x'

, y los vectores normales a la superficie entre x y x' respectivamente.

$\text{Vis}(x, x')$ es una función de visibilidad, definida como 1 si los dos puntos x y x'

son visibles entre ellos, y 0 si no lo son.



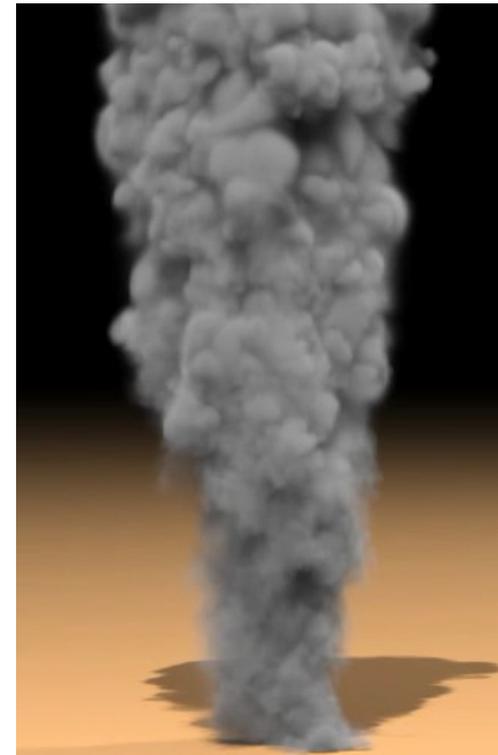


DINAMICA

- En general, dinámica mecánica
- Colisiones
- Sistemas compuestos
- Sistemas con restricciones
- Destrucción/separación

EFFECTOS DE PARTICULAS

- Lluvia
- Fuego
- Humo
- Explosiones



DINAMICA DE PARTICULAS

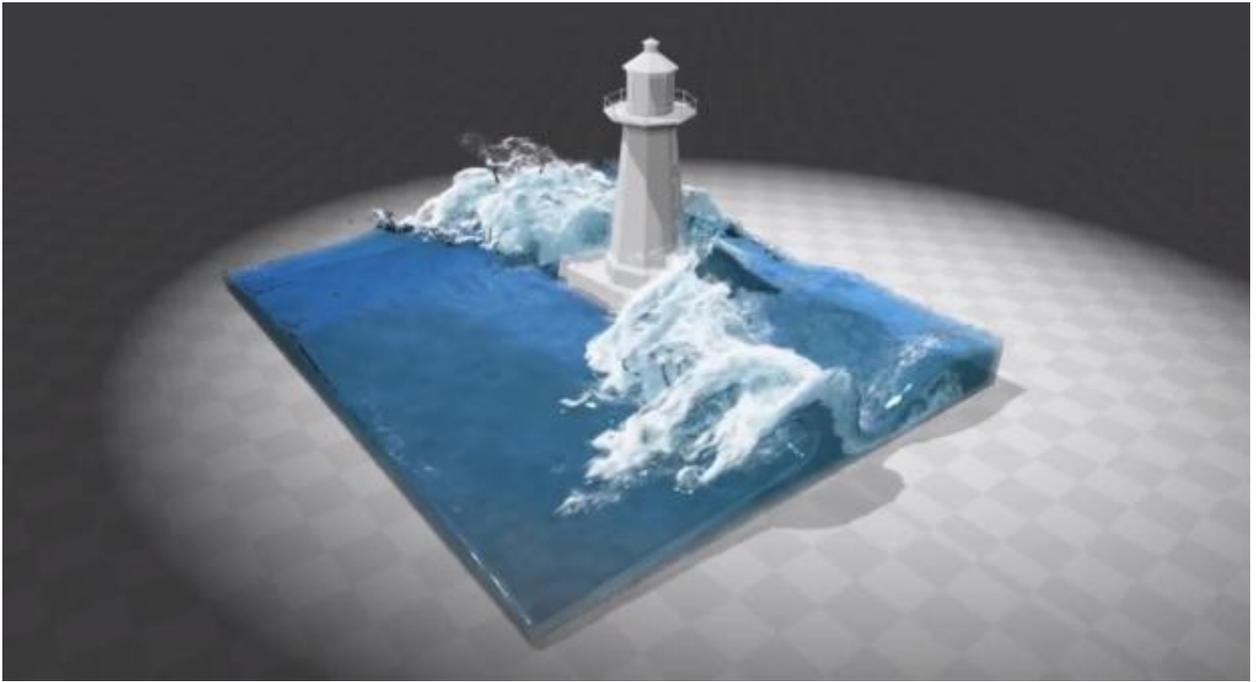
- Creación de una partícula con posición, velocidad, propiedades iniciales. Color inicial, textura (imagen) inicial
- Cada ΔT
 - Avanzar la dinámica de la partícula
 - Afectar posición, velocidad
 - Afectar color, textura
 - Determinar longevidad de la partícula

EJEMPLO PARTICULAS: LLUVIA VERTICAL

- Posicion inicial = AlAzar (Area de lluvia)
- Velocidad inicial = (0,0,0)
- Dinámica cada ΔT :
 - $v\text{-nuevo}(z) = v(z) + g * \Delta T$
 - $x\text{-nuevo}(z) = x(z) + v * \Delta T$
 - Hasta que $x(z) < x(\text{del-suelo})$
 - Efecto adicional : dibujar la lluvia como una linea y alargarla cada ΔT , en lugar de ser un punto
 - Efecto adicional #2 : Cambiar color de azul oscuro a blanquecino

SISTEMAS DE FLUIDOS

- Diversos metodos
 - Sistemas de particulas
 - Generacion de superficies equipotenciales
 - Induccion de movimiento harmonico
 - CFD (Computational Fluid Dynamics)
 - Simulación completa de las ecuaciones de Navier-Stokes



DINAMICA DE OBJETOS RIGIDOS

PASO DE VERLET(1)

- Método de Newton (o integrador de Euler) para calcular siguiente posición de un objeto sujeto a fuerzas externas causadas por un campo o por otros medios
 - Inestable , se desvia rápidamente de la superficie de energía constante y además requiere mantener la posición y velocidad de todas las partículas o actores
 - Simple : a cada paso se calcula la fuerza, se adiciona la velocidad * ΔT dividida por m y esta se adiciona a la posición * ΔT

PASO DE VERLET(2)

$$\frac{\Delta^2 \vec{x}_n}{\Delta t^2} = \frac{\frac{\vec{x}_{n+1} - \vec{x}_n}{\Delta t} - \frac{\vec{x}_n - \vec{x}_{n-1}}{\Delta t}}{\Delta t} = \frac{\vec{x}_{n+1} - 2\vec{x}_n + \vec{x}_{n-1}}{\Delta t^2} = \vec{a}_n = A(\vec{x}_n)$$

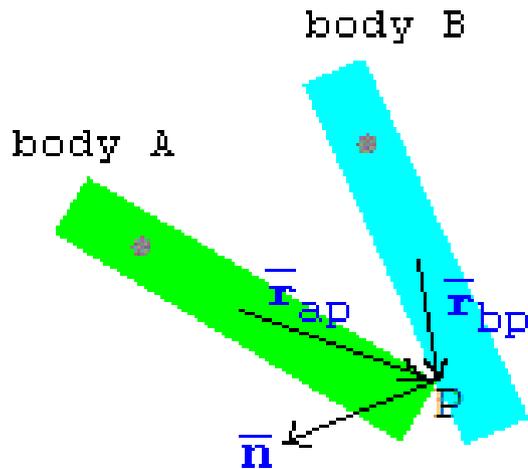
$$\vec{x}_{n+1} = 2\vec{x}_n - \vec{x}_{n-1} + \vec{a}_n \Delta t^2, \quad \vec{a}_n = A(\vec{x}_n).$$

Mas estable

Ocupa menos espacio de almacenamiento

Otros metodos de integracion : Runge-Kutta

ECUACION DE TRANSMISION DE IMPULSO

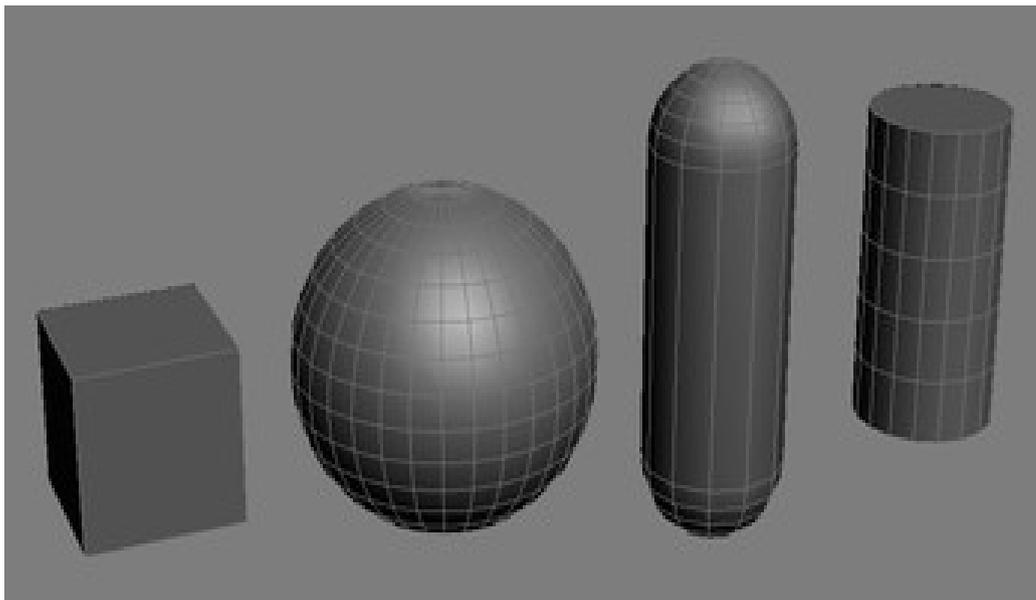
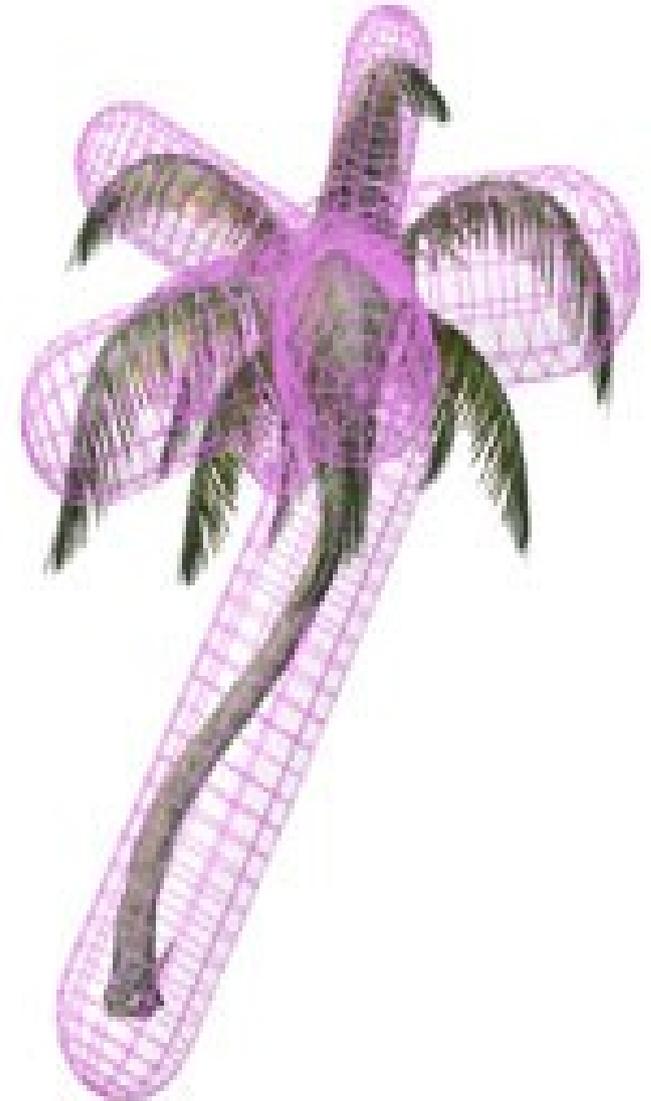
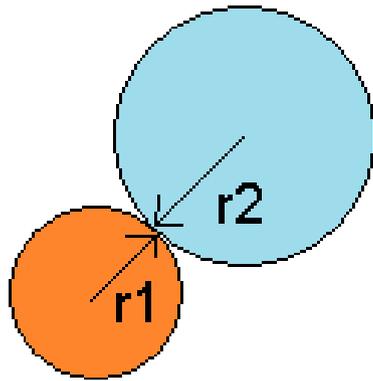


$$\bar{\mathbf{v}}_{a2} = \bar{\mathbf{v}}_{a1} + j \bar{\mathbf{n}} / m_a$$

$$\bar{\mathbf{v}}_{b2} = \bar{\mathbf{v}}_{b1} - j \bar{\mathbf{n}} / m_b$$

$$j = \frac{-(1 + e) \bar{\mathbf{v}}_{ab1} \cdot \bar{\mathbf{n}}}{\frac{1}{m_a} + \frac{1}{m_b} + (\bar{\mathbf{r}}_{ap} \times \bar{\mathbf{n}})^2 / I_a + (\bar{\mathbf{r}}_{bp} \times \bar{\mathbf{n}})^2 / I_b}$$

COLISIONES



PROBLEMAS

- No conservacion de energia
- Colisiones “tardias” : hay que separar los objetos para evitar penetracion
- Multiples colisiones : separacion simple no es efectiva

DESTRUCCION

- “Destrucción” de objetos compuestos de triangulos en objetos mas simples/menores debidos a una colision



VEHICULOS

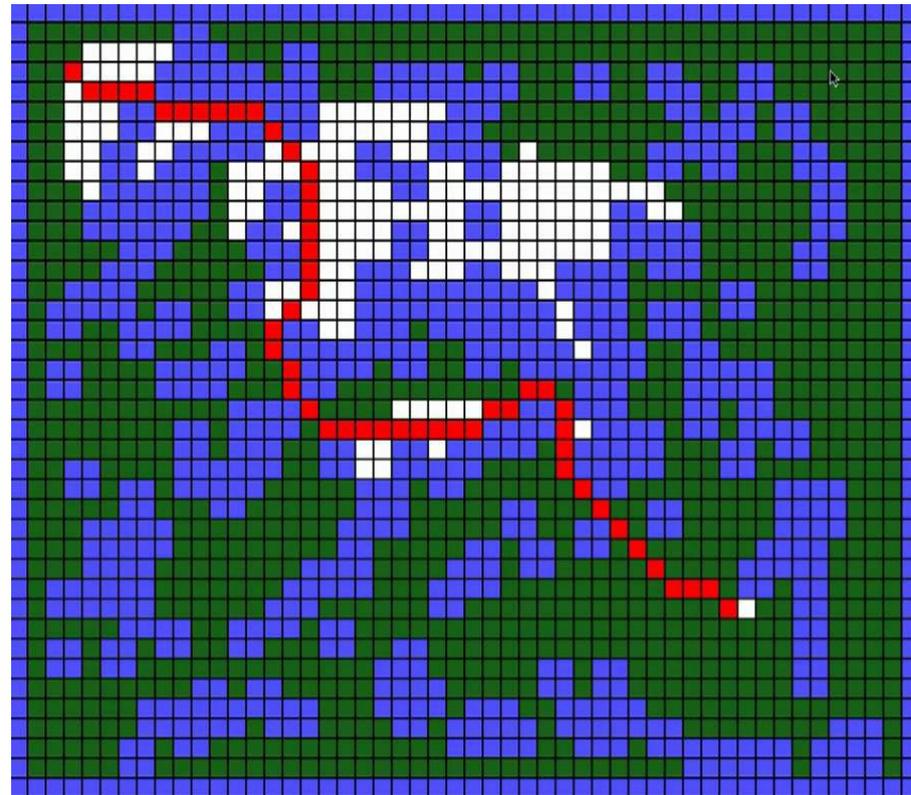
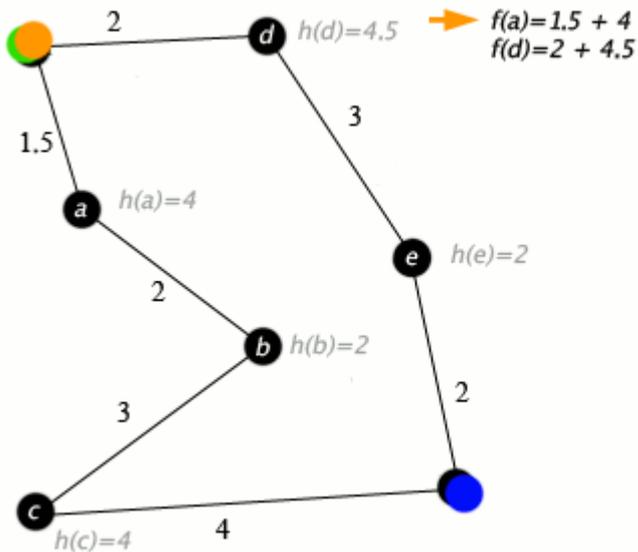
- Dirección (hidráulica o no)
- Suspensión
- Inyección de gasolina
- Respuesta del motor (hp vs rpm)
- Respuesta de la transmisión
- Potencia a las llantas
- Llantas vs terreno

PAQUETES COMERCIALES Y ABIERTOS

- HAVOK (INTEL)
- BULLET
- PHYSX
- EUPHORIA
- OPEN DYNAMICS ENGINE

REACCIÓN

- "INTELIGENCIA ARTIFICIAL"
- ALGORITMO A*
 - HEURISTICA DE "COSTO" DE UN CAMINO



CONCLUSIONES

- La física, o al menos el modelaje físico, tiene muchas aplicaciones en los juegos
- Hay muchas similitudes entre las aplicaciones de simulación científica y la simulación para juegos
- La experiencia en este campo es útil para ambos

REFERENCIAS

- Physically Based Rendering, Third Edition: From Theory to Implementation 3rd Edition by Matt Pharr , Wenzel Jakob , Greg Humphreys
- Foundations of Physically Based Modeling and Animation by Donald H. House, John C. Keyser
- Game Physics Engine Development: How to Build a Robust Commercial-Grade Physics Engine for your Game Paperback by Ian Millington

VIDEOS

- SIGGRAPH University - Introduction to "Physically Based Shading in Theory and Practice"
<https://www.youtube.com/watch?v=j-A0mwsJRmk>
- Physically Based Shading in Theory and Practice
<https://www.youtube.com/watch?v=zs0oYjwjNEo>
- Intro to Game Physics
<https://www.youtube.com/watch?v=wPKzwSxyhTI>
- 10 Games Which Had The Most Incredible Physics Ever