

GRAFICACIÓN IA7200-T

Profesor

- Luis Eduardo Gamboa Guzman
 - Sitio web: <https://lc.fie.umich.mx/~legg/>
- Comunicación fuera de clase:
 - luis.gamboa@umich.mx
 - Contestaré tan pronto me sea posible

Sobre el curso

- Es una materia del sexto semestre
 - Prerrequisitos: Álgebra Lineal y Programación
 - ¡Recomendable dar un repaso a cálculo!
 - No es un curso de OpenGL, DirectX, etc
- Tres horas teóricas por semana + tres hora prácticas
 - SEIS horas en total
- Horario será sin excepciones el asignado en escolar

Sobre el curso: objetivos

- Dominar las técnicas de renderización ***basadas en física*** para el modelado de fuentes luminosas y la interacción luz-material
- Conocimiento básico sobre el cual se desarrolla la *tecnología de punta* en el área de gráficos por computadora en *motores de renderizado realista*

Sobre el curso: contenido

- Introducción a las imágenes generadas por computadora
 - Aplicaciones, tiempo real vs fuera de línea, ventajas de renderizado basado en física, geometría de escena
- Lanzamiento de rayos
 - Definiciones, transformaciones, primitivas de intersección, estructuras de aceleración de intersección
- Examen

Sobre el curso: contenido (2)

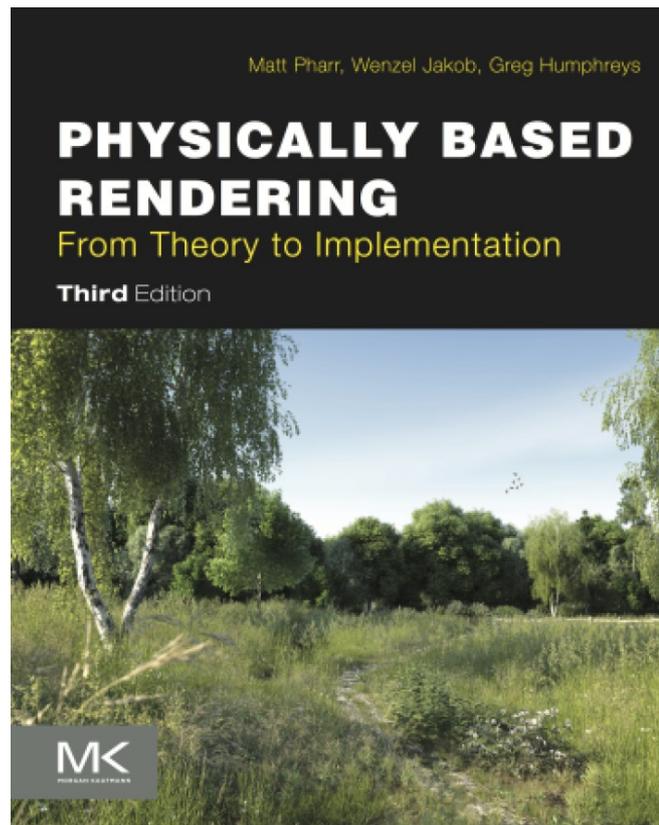
- Iluminación directa
 - Ecuación de renderizado, método Monte Carlo, muestreo de importancia, muestreo de importancia múltiple
- Fuentes luminosas
 - Medidas, fuentes puntuales, fuentes de área, mapas ambientales
- Examen

Sobre el curso: contenido (3)

- Materiales
 - Lambertianos, especular perfecto, dielectricos suaves, conductores y dielectricos ásperos, modelo microfacet
- Iluminación global
 - Ecuación de renderizado, caminos de luz, ray tracing vs path tracing, path tracing implícito y explícito, path tracing bidireccional
- Examen

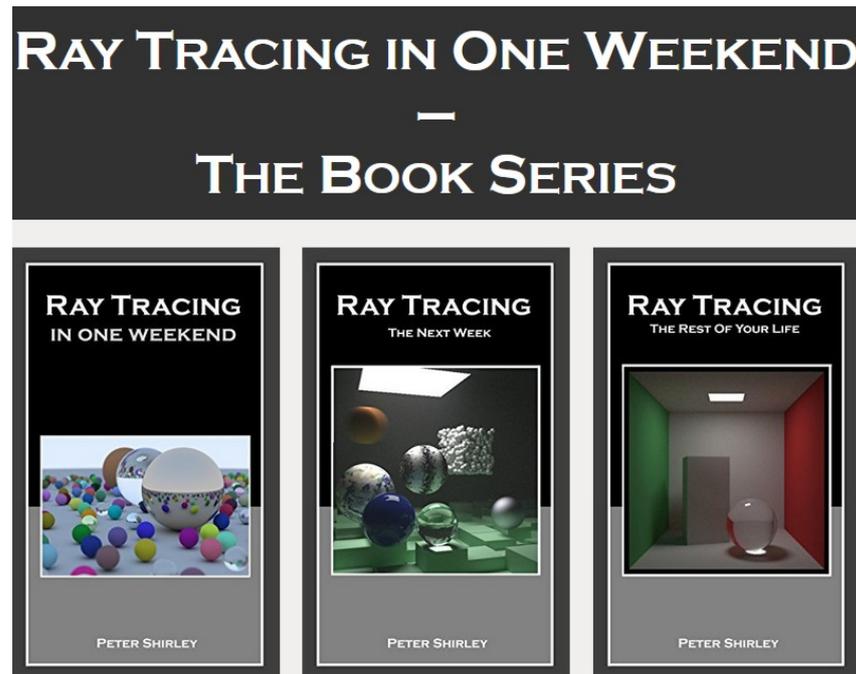
Material de la clase

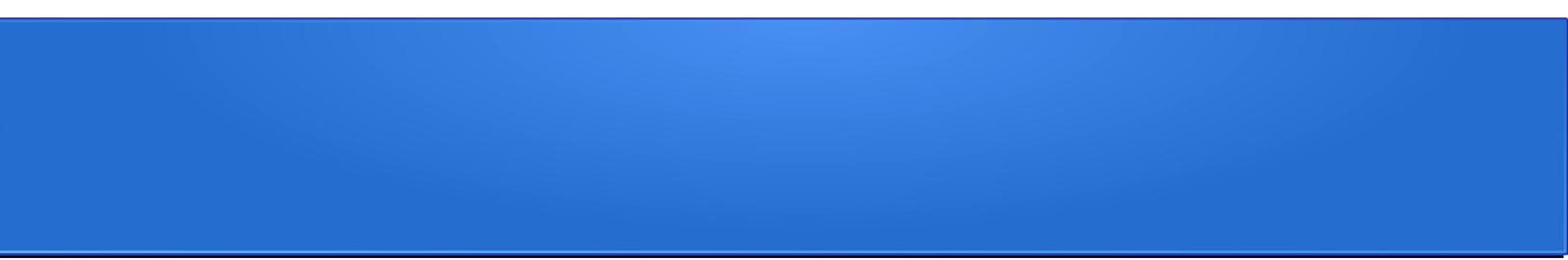
- ¡Diapositivas en español!
- Libro, “el libro”:
 - *Physically Based Rendering: From Theory To Implementation*
 - <https://www.pbr-book.org/>
 - *Autores ganadores de un Oscar: Scientific and Technical Achievement*



Material de la clase (2)

- Bibliografía complementaria:
 - *Ray Tracing in One Weekend*
 - <https://raytracing.github.io/>
 - *Peter Shirley es un distinguido investigador en el área*
 - *Casi 20,000 citas*
 - *h-index 56*

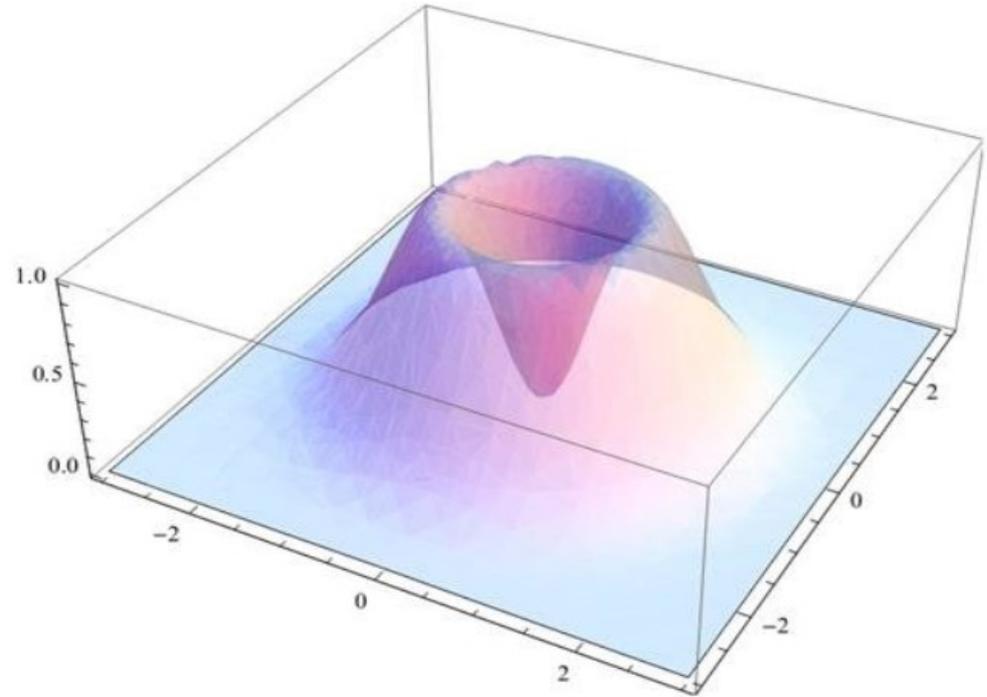
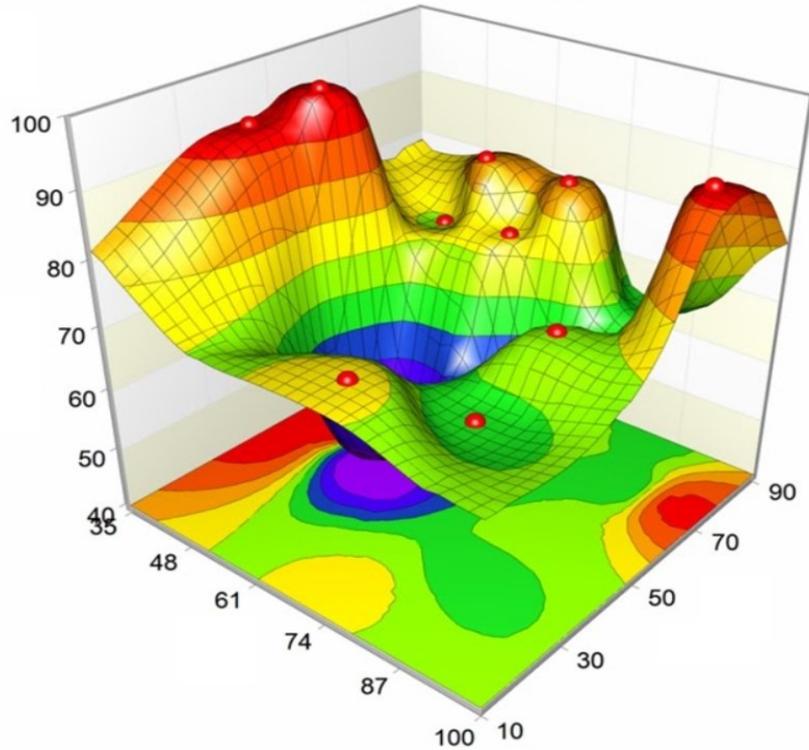




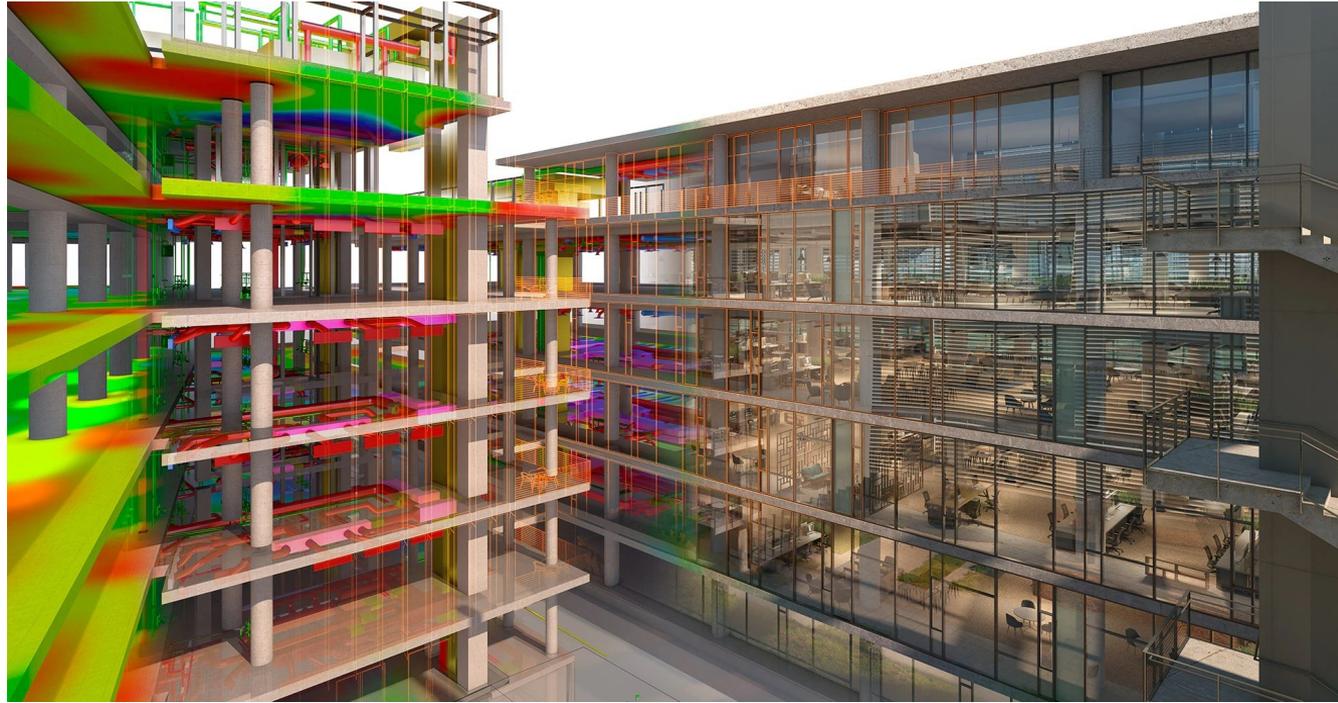
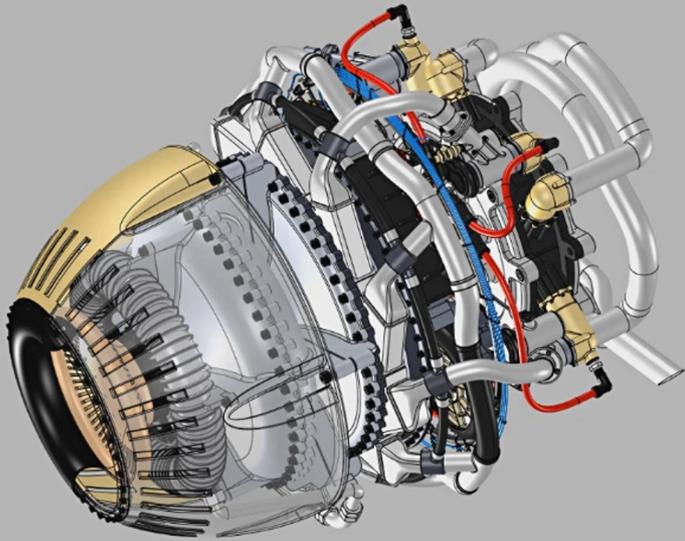
Tema 1

*Introducción a las imágenes
generadas por computadora*

Aplicaciones



Aplicaciones (2)



Diseño asistido por computadora (CAD)

Aplicaciones (3)

2D



3D



4D

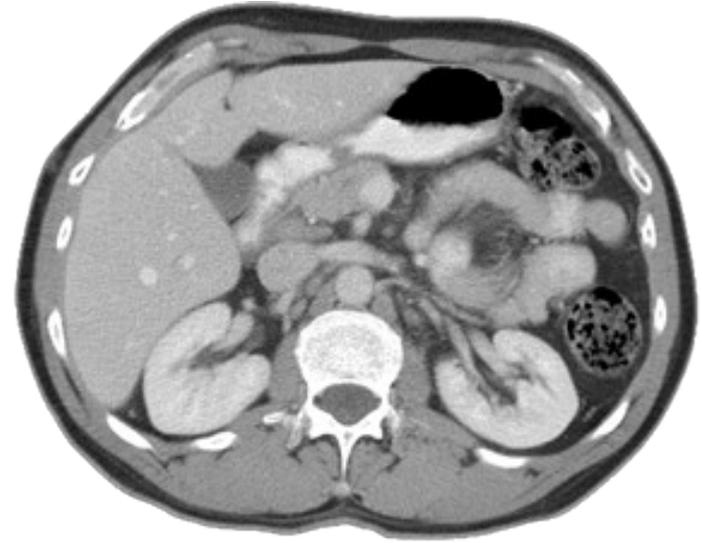


HD



Ultrasonido

Aplicaciones (4)



Tomografía / RM computarizada

Aplicaciones (5)



Unreal Engine 5 Reveal Demo (Epic Games)

Aplicaciones (6)



Toy Story 4 (Disney/Pixar 2019)

Aplicaciones (7)



The Hobbit: An Unexpected Journey (Warner Bros. Pictures 2012)

Impacto económico

- Industria con un valor global de ~ 240,000 mdd (2020)
- Cine: se destina 20-25% en efectos especiales
- Películas de animación 20-300 mdd
- Videojuegos > cine + deportes
- La investigación es priorizada en la industria

Técnicas en tiempo real vs fuera de línea

- Tiempo real
 - 30 fotogramas por segundo
 - Presupuesto de ~33ms por fotograma
 - Todavía menos si el objetivo es 60fps o 120fps
 - Resolución 1920x1080 --> 2,073,600 píxeles
 - Rápidamente se hace costoso
 - Hay que introducir optimizaciones, atajos o hacks

Los atajos a lo largo del tiempo

- Wolfenstein 3D
 - 1992



Los atajos a lo largo del tiempo

- Doom
 - 1993



Los atajos a lo largo del tiempo

- Doom II
 - 1994



Los atajos a lo largo del tiempo

- Quake
 - 1996



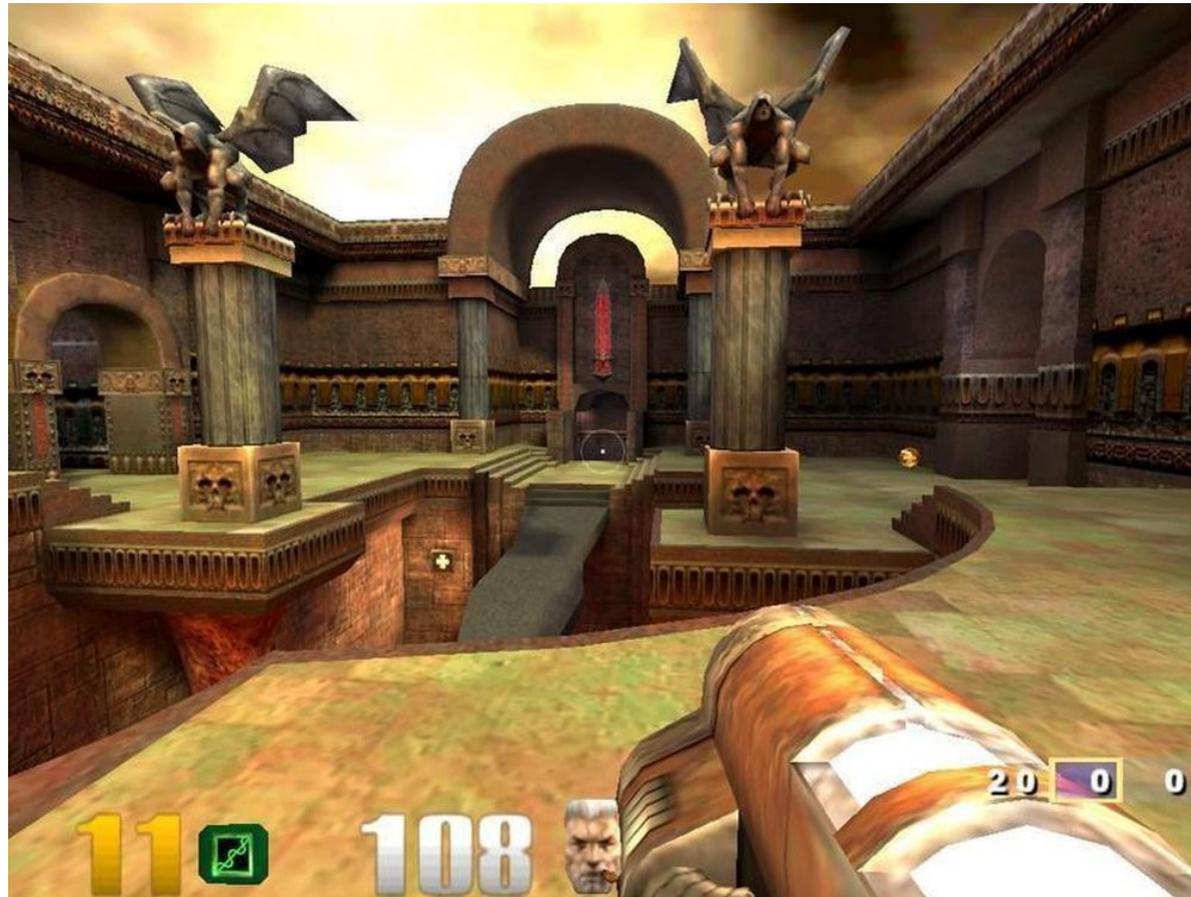
Los atajos a lo largo del tiempo

- Quake 2
 - 1997



Los atajos a lo largo del tiempo

- Quake 3
 - 1999



Los atajos a lo largo del tiempo

- Doom 3
 - 2006



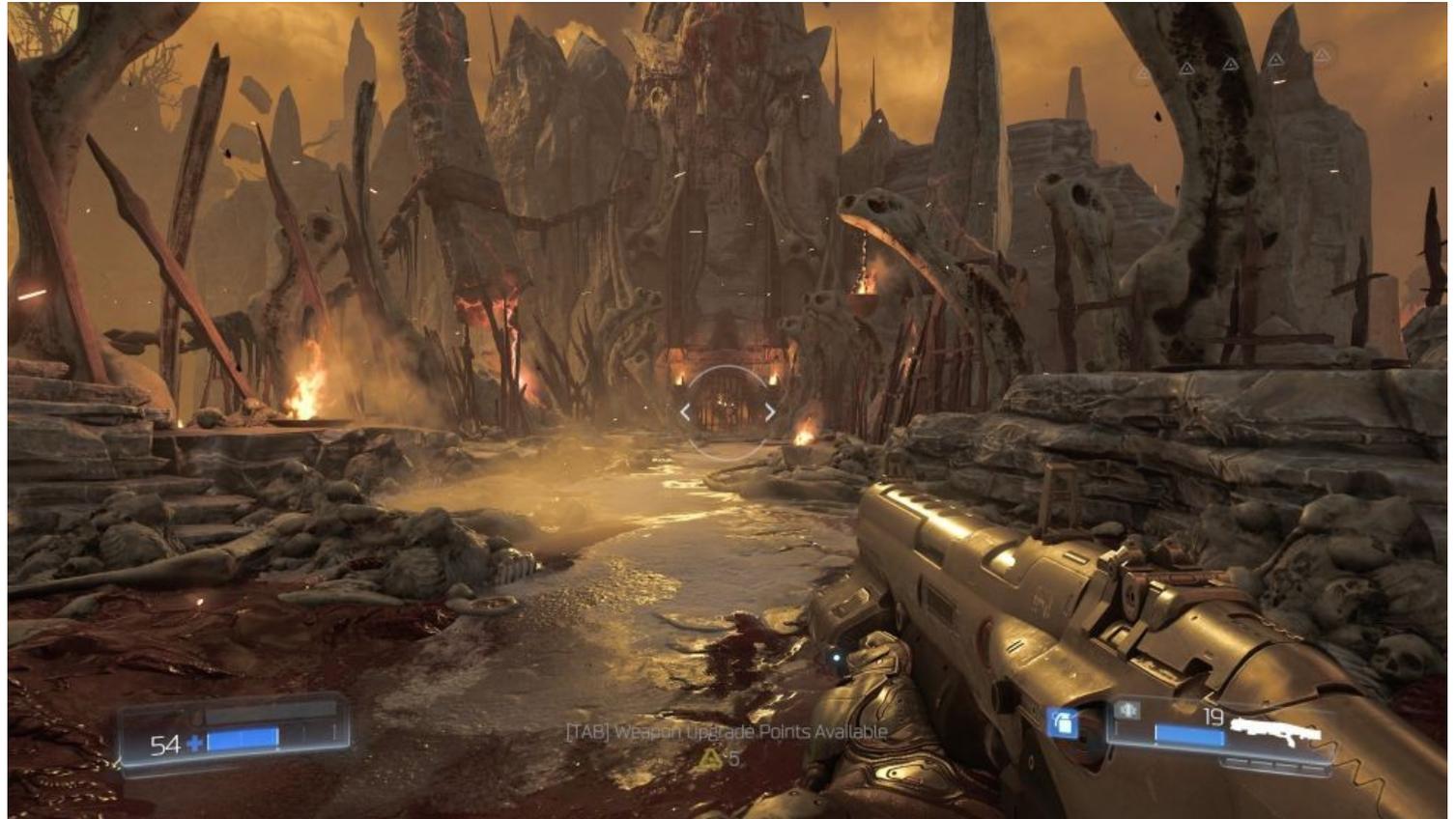
Los atajos a lo largo del tiempo

- Killzone 2
 - 2009



Los atajos a lo largo del tiempo

- Doom
 - 2016



Los atajos a lo largo del tiempo

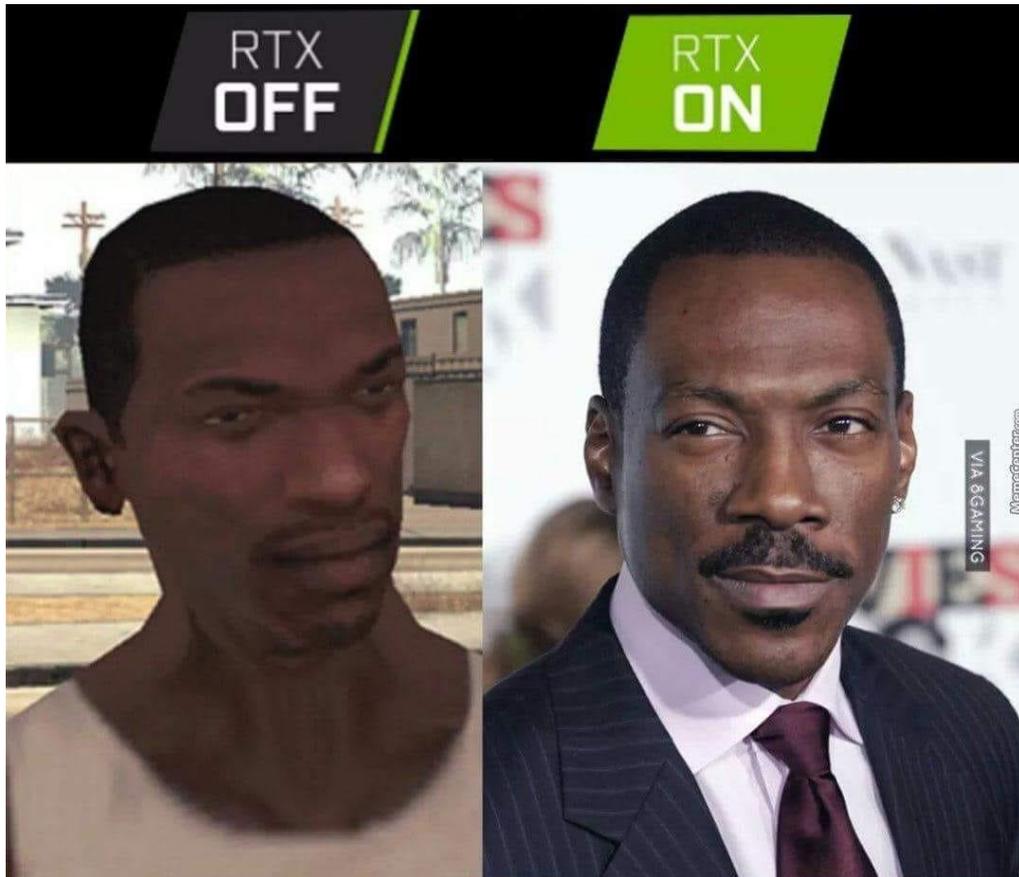
- Doom
Eternal
– 2020



Renderizado en Tiempo Real

- Evitan hacer las cosas como se deben hacer
 - No simula el transporte de luz
 - Modelos e implementaciones fenomenológicas
- Los resultados han mejorado mucho, pero invariablemente la tendencia es el renderizado basado en física

Renderizado en Tiempo Real



Ray Tracing en tiempo real?



Ray Tracing en tiempo real?



CONTROL

REMEDY 



RTX
OFF

CONTROL

REMEDY 



RTX
ON

Renderizado fuera de línea

- No hay un presupuesto de tiempo tan pequeño
 - Menos atajos o hacks
- Hay oportunidad de hacer las cosas “bien”
 - Pero sigue siendo costoso
 - No todo lo que es fuera de línea es perfecto

Renderizado fuera de línea (2)

- Money for nothing
– 1985



Renderizado fuera de línea (3)

- A pesar de no haber restricciones en tiempo
 - La capacidad de cómputo es limitada
- Nuevos modelos todavía se están investigando
- Existen problemas abiertos

Renderizado fuera de línea (4)



Ventajas del renderizado basado en física

- Se hace una simulación de lo que sucede en realidad
- Un rayo de luz es emitido por una fuente luminosa
- La luz “rebota” sobre las superficies
- Llega finalmente a un sensor (cámara u ojos).

Ventajas del renderizado basado en física (2)

- El realismo obtenido es extremadamente alto
 - Pero...
- Esta simulación puede ser demasiado costosa
 - Una sola imagen de 1920x1080 necesitaría más de **2,000 millones de caminos de luz** si se consideran 1,024 muestras por píxel.

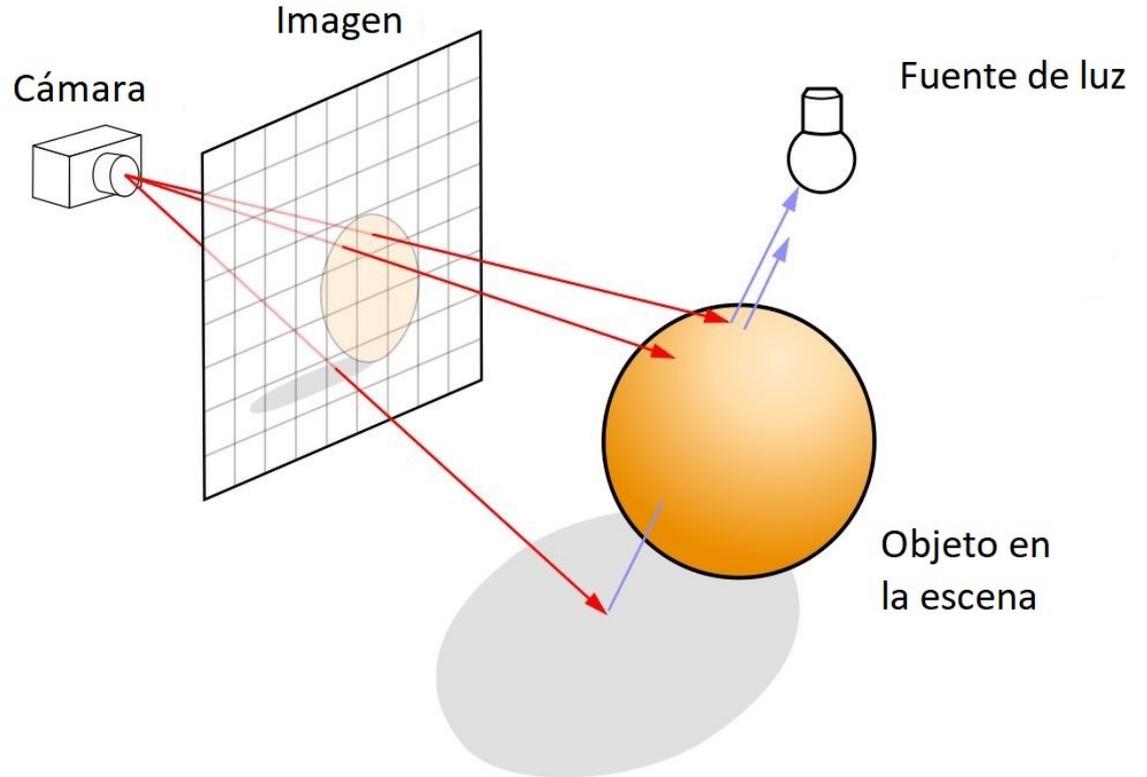
Ventajas del renderizado basado en física (3)

- Modelo:
 - 1.8 millones de vértices
 - 6 fuentes de luz
- 1280 spp
- 73 horas en 4 núcleos



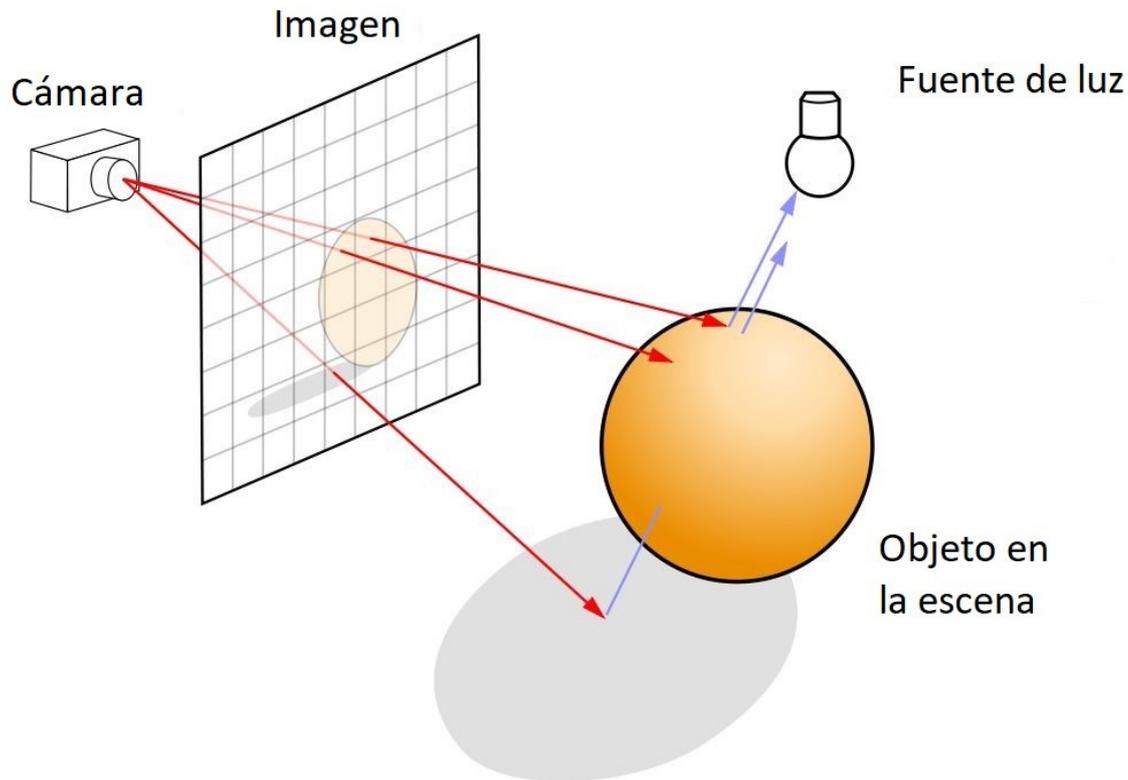
Geometría de escena

- Un sensor o cámara observa la escena
- Se lanzan “rayos” desde la cámara



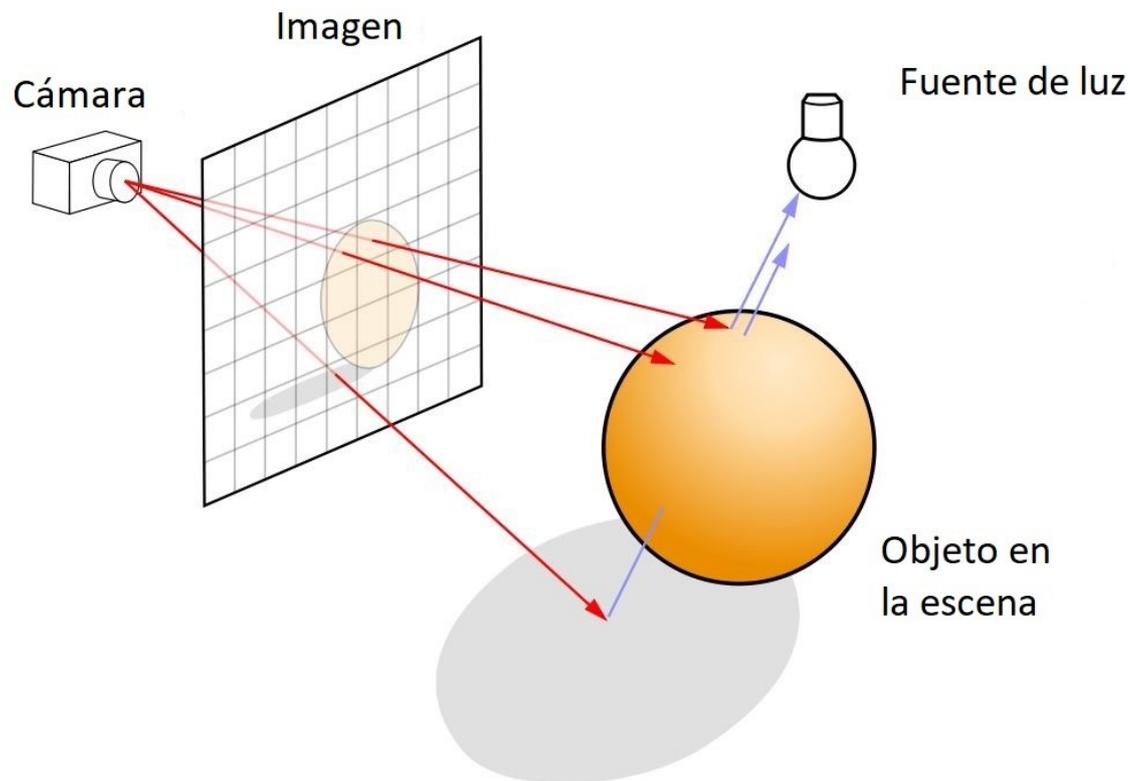
Geometría de escena (2)

- Los rayos atraviesan un plano colocado frente a la cámara
- La posición del rayo en la imagen es el pixel
 - Determinar su color



Geometría de escena (3)

- El color está dado por
 - radiancia saliente en esa posición y dirección
- Aquí está nuestro primer reto



Imagen

- Es una matriz de píxeles
- El color de cada píxel está dado por una representación del espectro visible (380nm - 750nm):
 - Un vector de valores Rojo-Verde-Azul (RGB)
 - Un vector de medidas de distintas bandas espectrales
- El espectro visible:

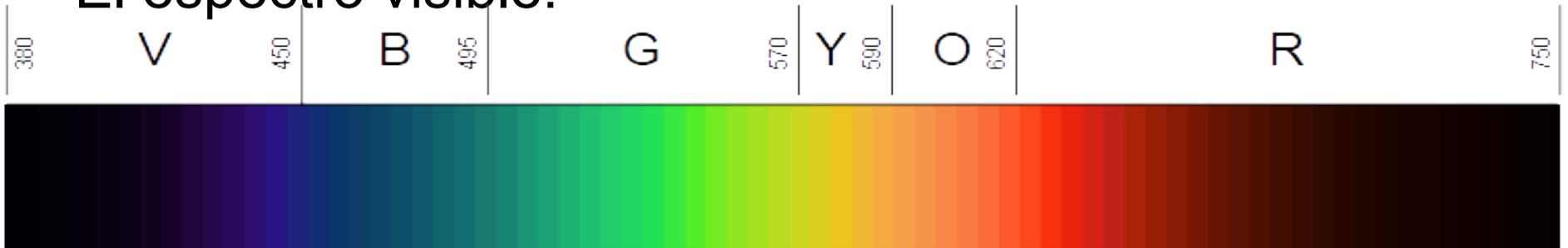
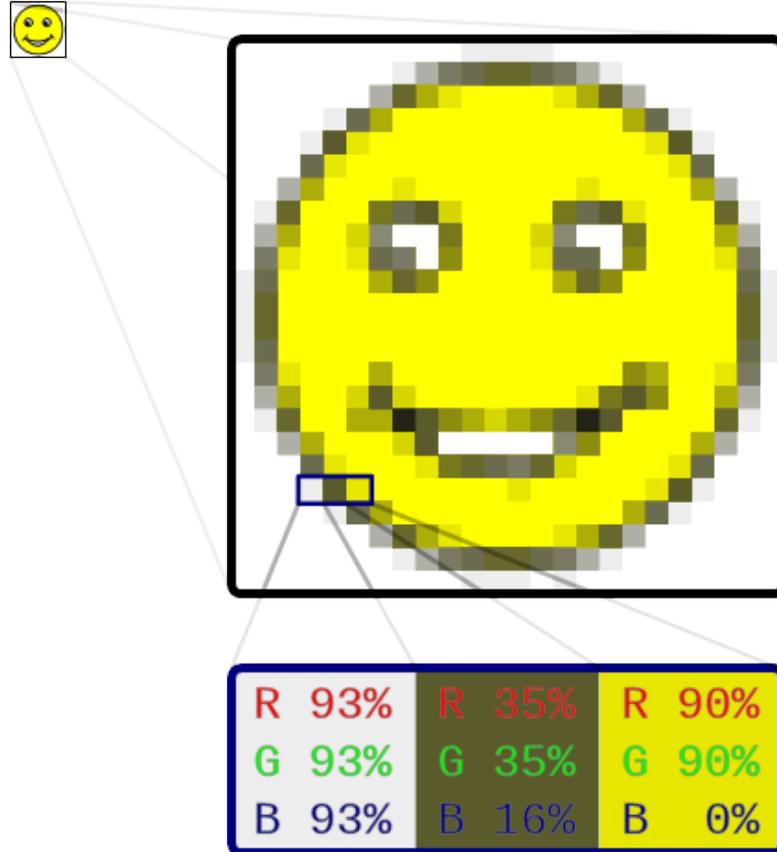


Imagen (2)

- Normalmente en RGB se utilizan 8 bits por canal
 - Valores discretos en $[0,255]$
- Formatos HDR (ej. EXR) generalmente usan float (32 bits por canal)



Objetos en la escena

- Representación geométrica en mallas de triangulos



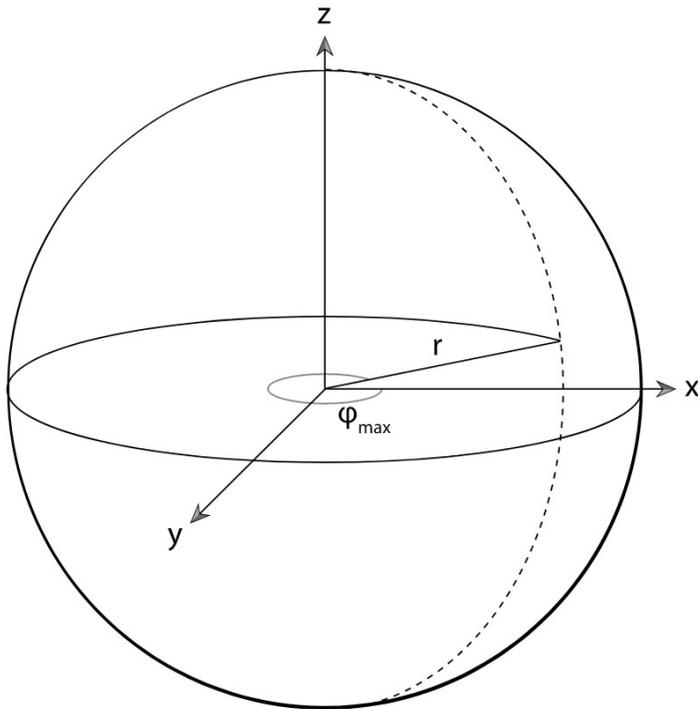
Malla de triángulos



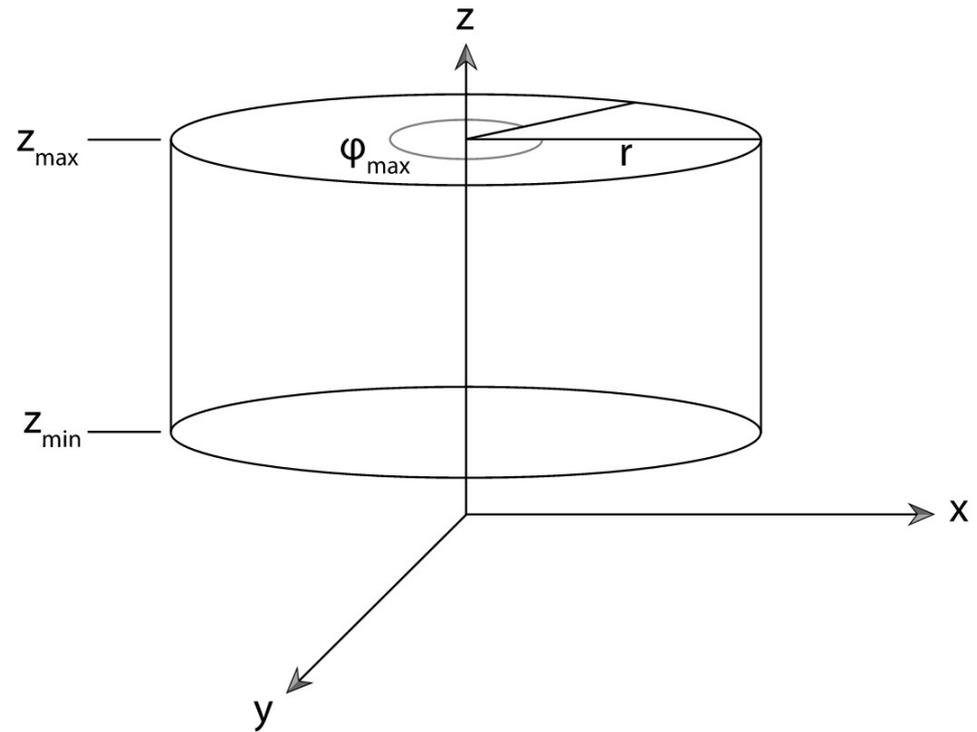
Subdivisión

Objetos en la escena (2)

- Representación geométrica exacta



Esfera (centro y radio)



Cilindro (centro de la base, altura y radio)