Sombras: Shadow Mapping

Computación Gráfica 2023

Juan Pablo Sumski

¿Qué son las sombras?

sombra

De *sombrar*.

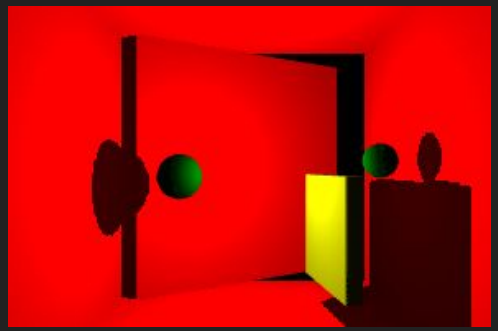
1. f. Oscuridad, falta de luz, más o menos completa. U. m. en pl. *Las sombras de la noche.*

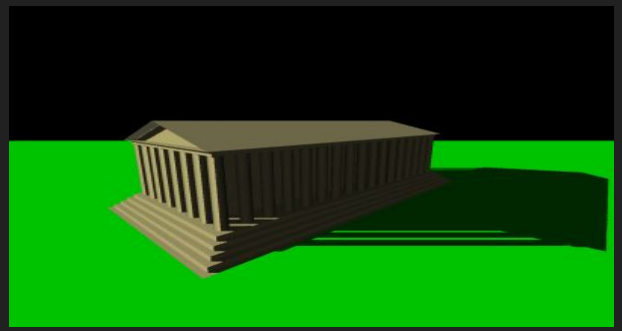
sombra

nombre femenino

1. Imagen oscura que proyecta un cuerpo opaco sobre una superficie al interceptar los rayos de luz.
   "la sombra del árbol cubría casi todo el jardín"
2. Parte de un espacio a la que no llega la luz, especialmente la del sol.
   "se sentó en la sombra para descansar"

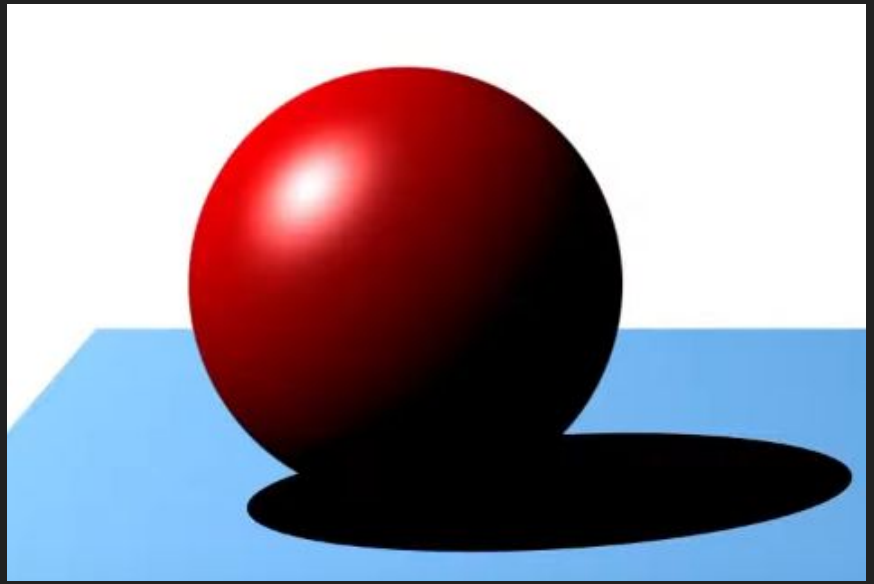
Antónimo: sol

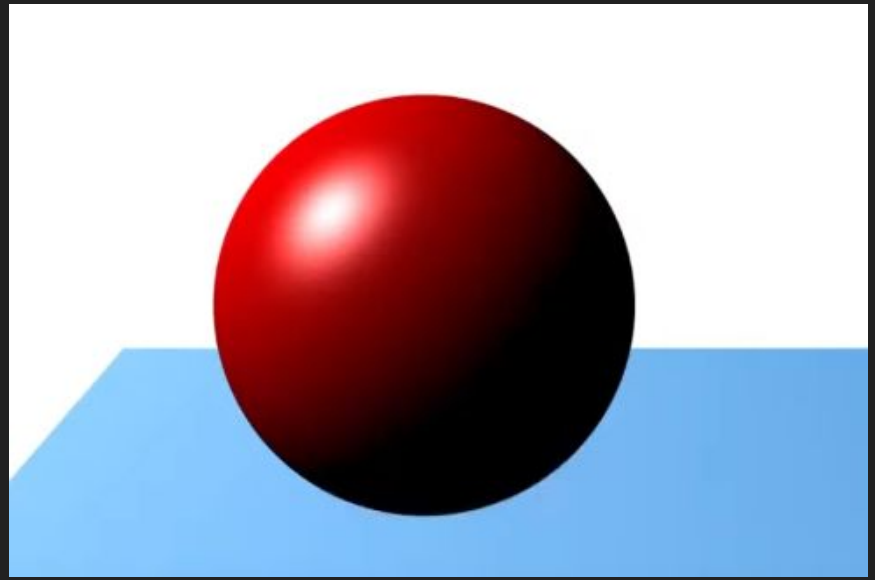




¿Para qué sirven o se utilizan?

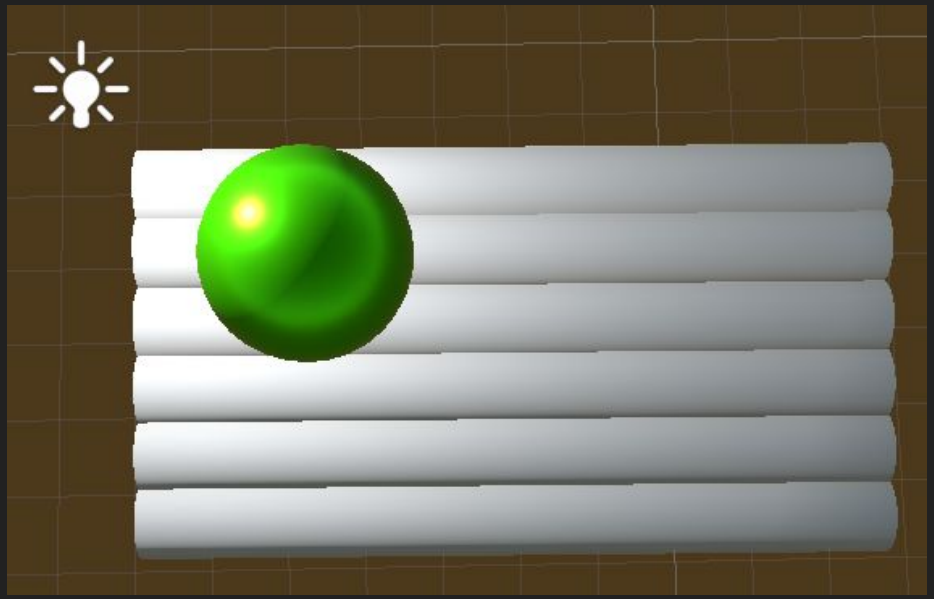
Ubicación espacial de los objetos

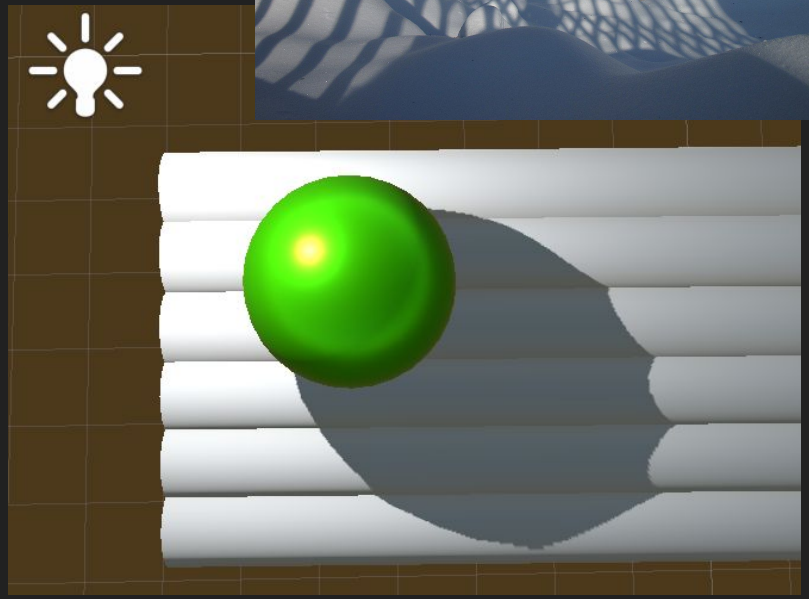




¿Para qué sirven o se utilizan?

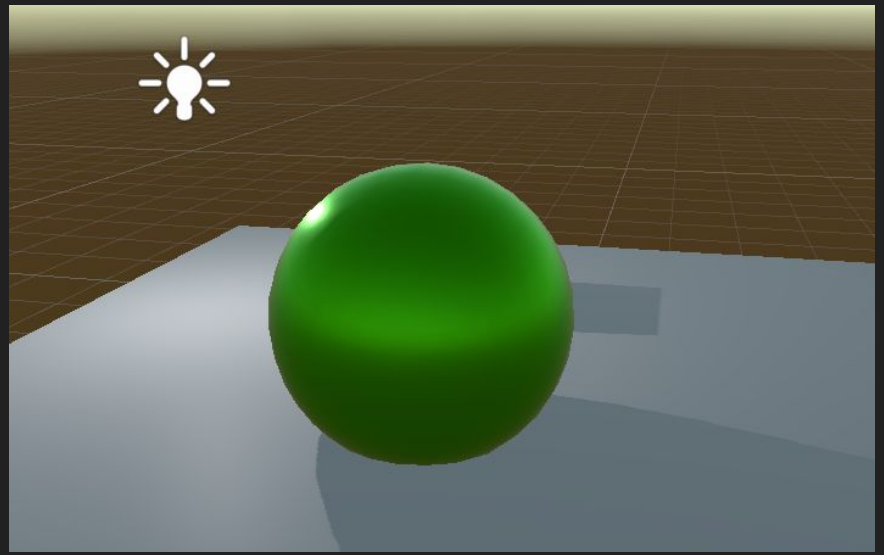
Información de la superficie gracias a la sombra

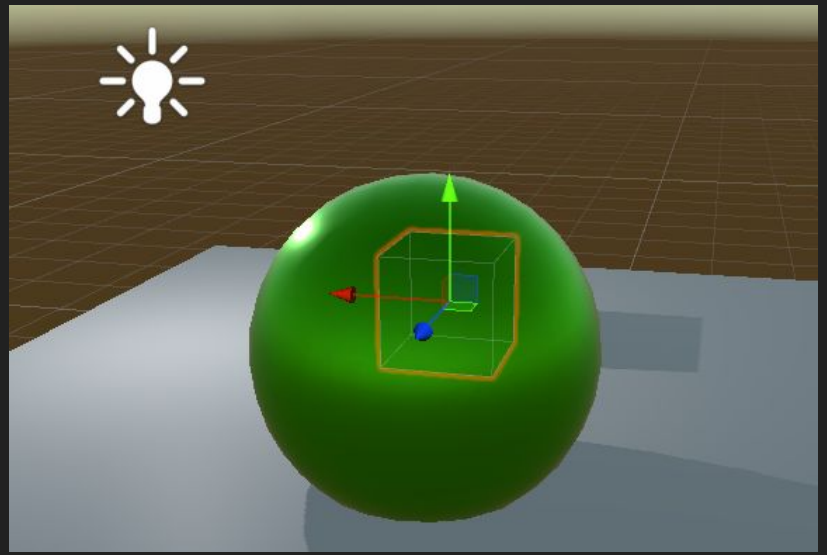




¿Para qué sirven o se utilizan?

Objetos que no vemos proyectan sombras





¿Cómo las representamos en CG?

Casting Curved Shadows on Curved Surfaces

Lance Williams

CASTING CURVED SHADOWS ON CURVED SURFACES

Lance Williams
Computer Graphics Lab
New York Institute of Technology
Old Westbury, New York 11568

Abstract

Shadowing has historically been used to increase the intelligibility of scenes in electron microscopy and aerial survey. Various methods have been published for the determination of shadows in computer synthesized scenes. The display of shadows may make the shape and relative position of objects in such scenes more comprehensible; it is a technique lending vividness and realism to computer animation.

To date, algorithms for the determination of shadows have been restricted to scenes constructed of planar polygons. A simple algorithm is described which utilizes Z-buffer visible surface computation to display shadows cast by objects modelled of smooth surface patches. The method can be applied to all environments, in fact, for which visible surfaces can be computed. The cost of determining the shadows associated with each light source is roughly twice the cost of rendering the scene without shadows, plus a fixed transformation overhead which depends on the image resolution. No extra entities are added to the scene description in

A "frame buffer," in the current computer graphics parlance, is a memory that stores a complete digital picture. It may serve as an intermediary between the computer that produces the picture and a video driver which continuously refreshes a display. Some visible surface algorithms (e.g. [2]) require a frame buffer in order to compute an image. In this case, the frame buffer mediates the display process in a more substantial way.

The Z-buffer is an extension of this mass-memory approach to computer graphics which resolves the visible surfaces in a scene by storing depth (Z) values at each point in the picture. As objects are rendered, their Z values are compared at each point with the stored Z values to determine visibility. Since this determination requires only that a measure exist which orders the surfaces to be displayed, it is not too strong a statement to say that the Z-buffer algorithm provides a discrete solution to all scenes for which visible surfaces can be computed.

Z-buffer visible surface computation is of particular interest because it exhibits limiting-case pro-

1978

Shadow Mapping

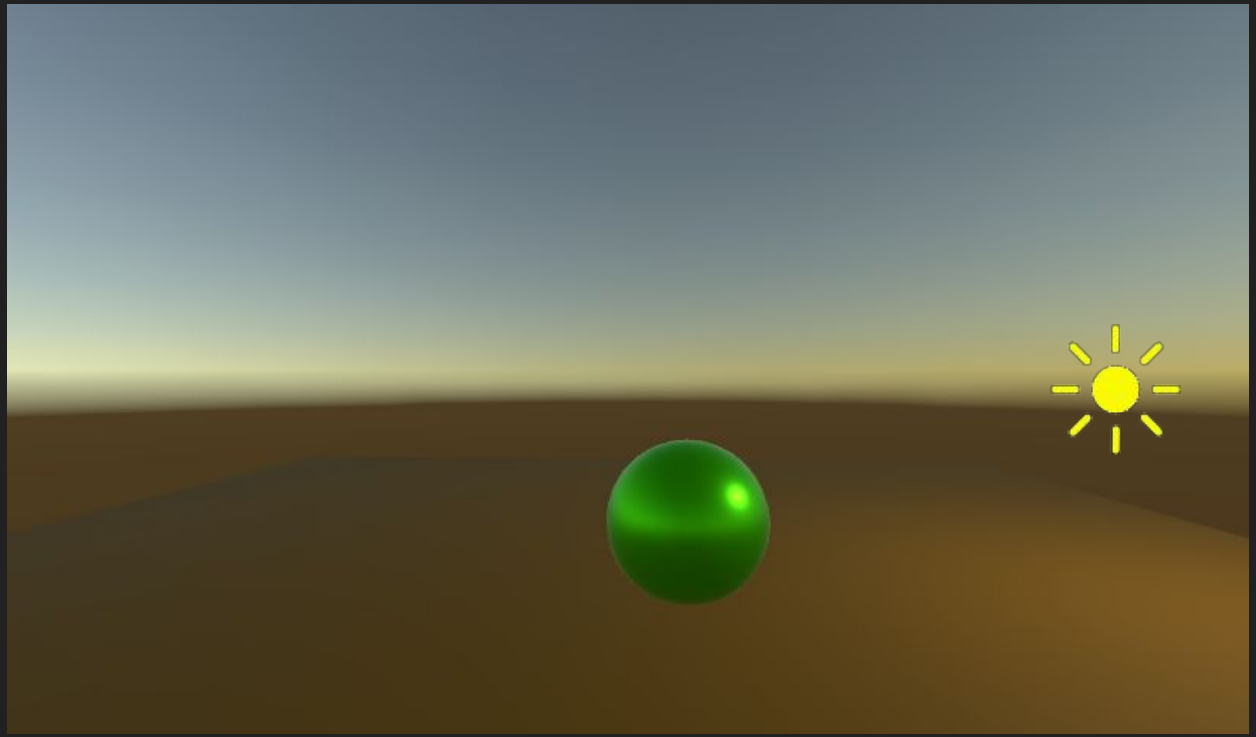
- **PRE-RENDER** o **REAL-TIME**
- **No** requiere **conocimiento de la geometría** de la escena
- Usa **z-buffer** de la vista de la cámara
- **EFICIENTE** pero tiene **LIMITACIONES**
- Llegó a ser estándar: **Pixar's RenderMan**

Recordemos esta definición:

"oscuridad proyectada por un objeto por la intercepción de la luz"

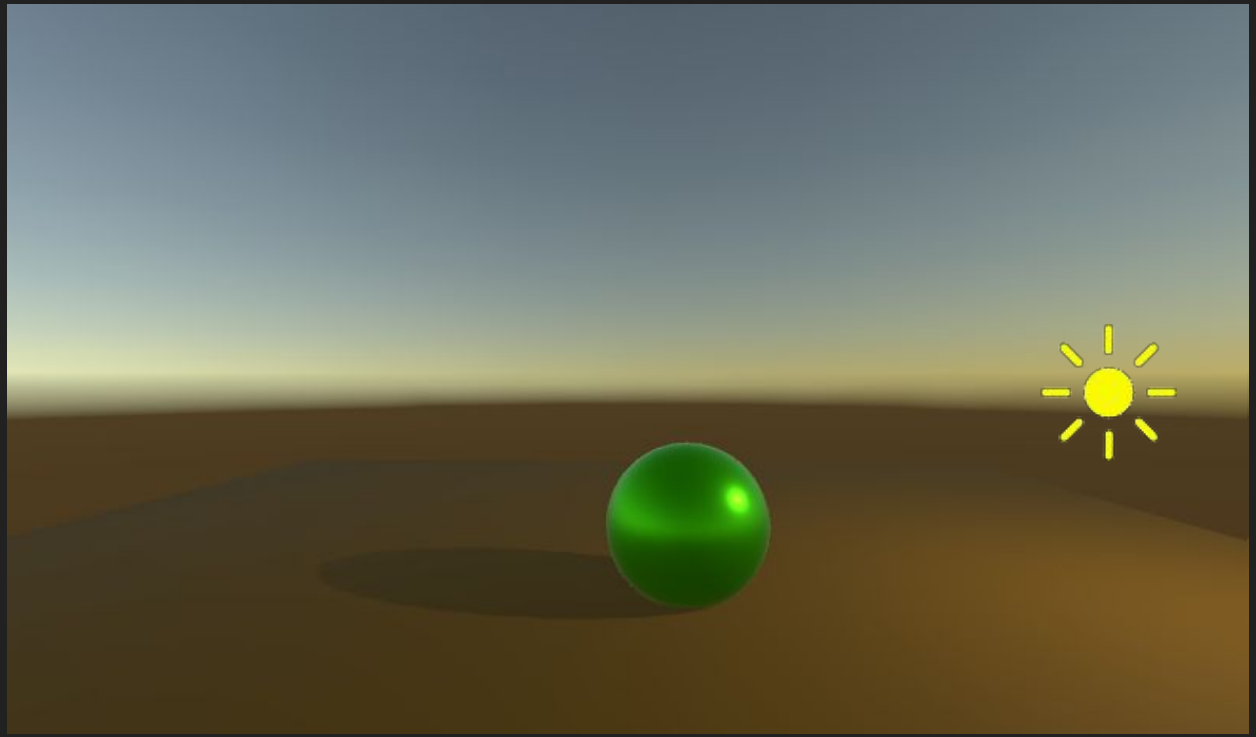
Shadow Mapping

“oscuridad proyectada por un objeto por la intercepción de la luz”

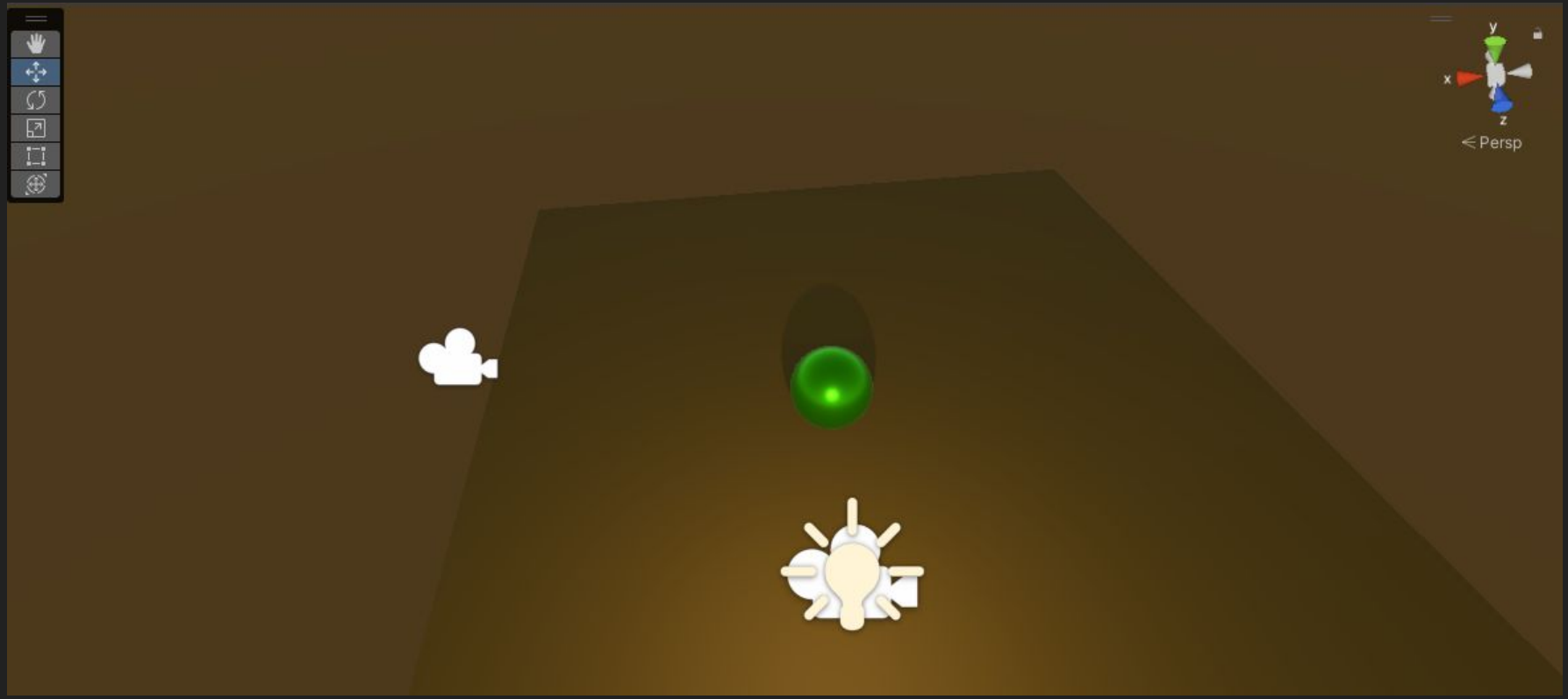


Shadow Mapping

“oscuridad proyectada por un objeto por la intercepción de la luz”

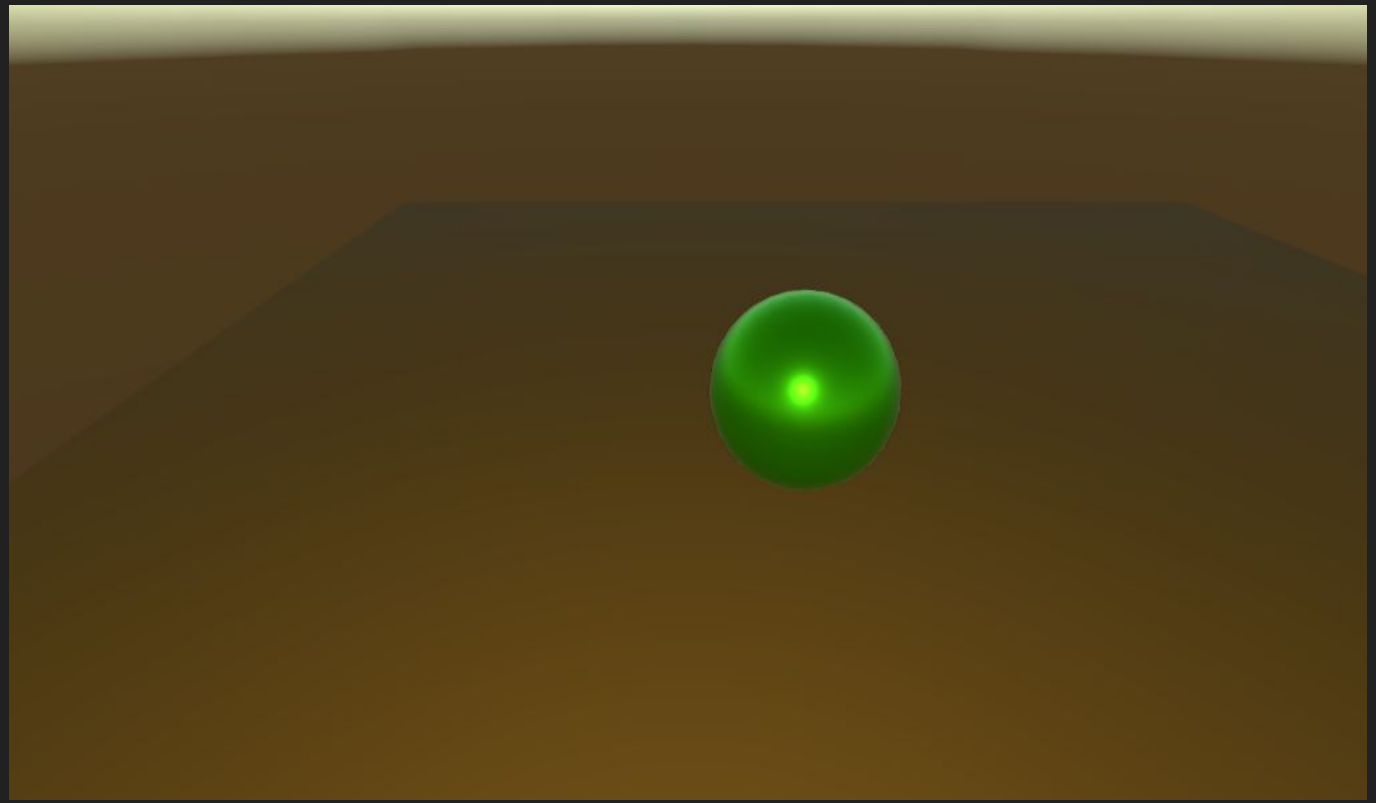


Shadow Mapping

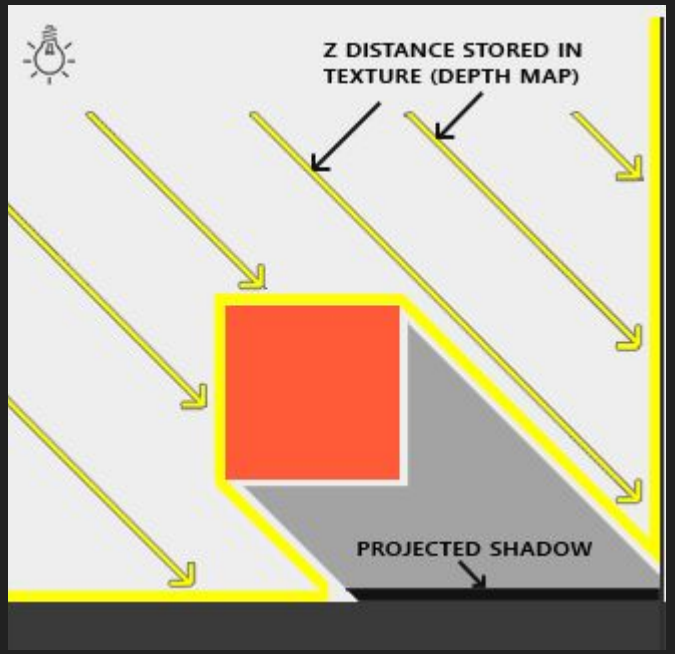


Shadow Mapping

Cámara



¿Cómo se implementa?

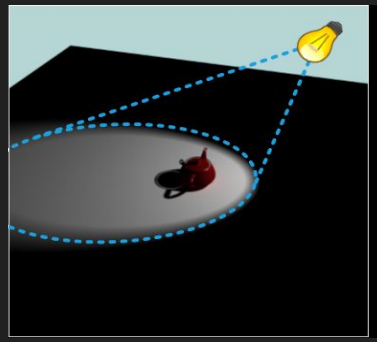


Primera Pasada

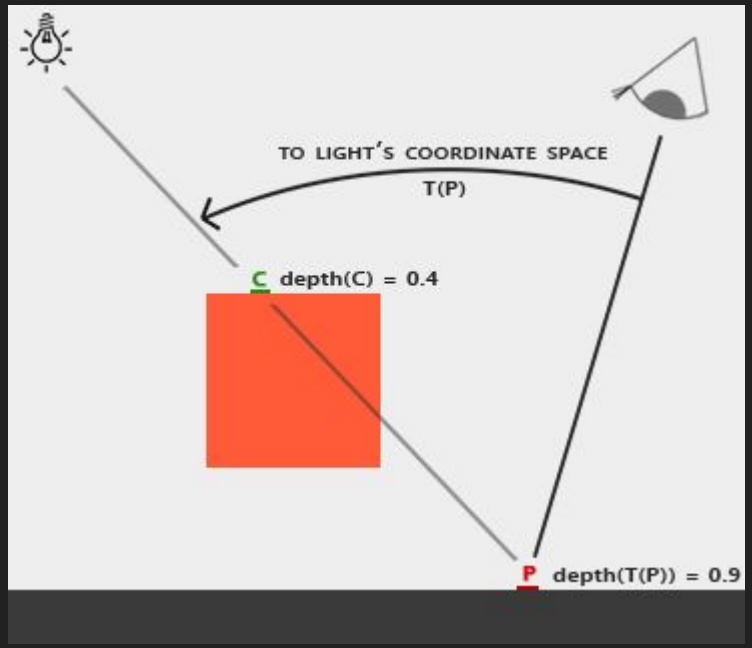
- **CÁMARA** DESDE **LUZ**
- **MAPEA PROFUNDIDAD**

  para cada pixel visto desde

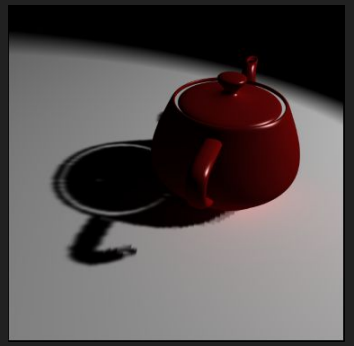
  LUZ



¿Cómo se implementa?



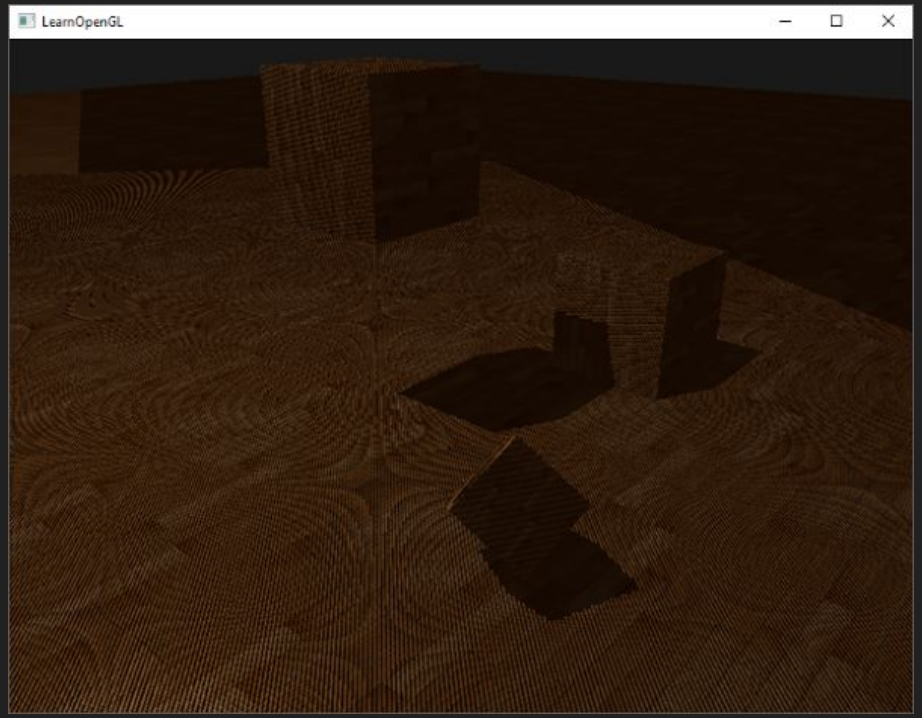
Segunda Pasada

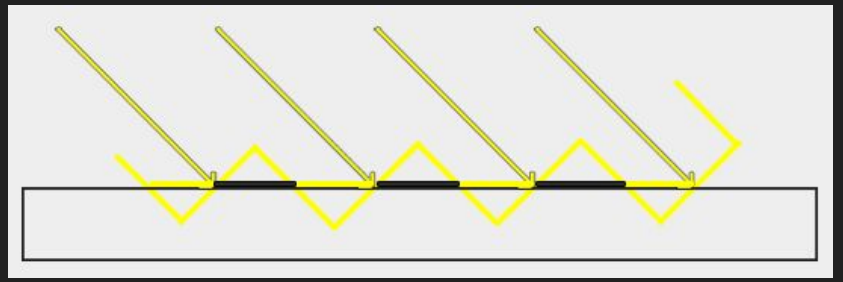


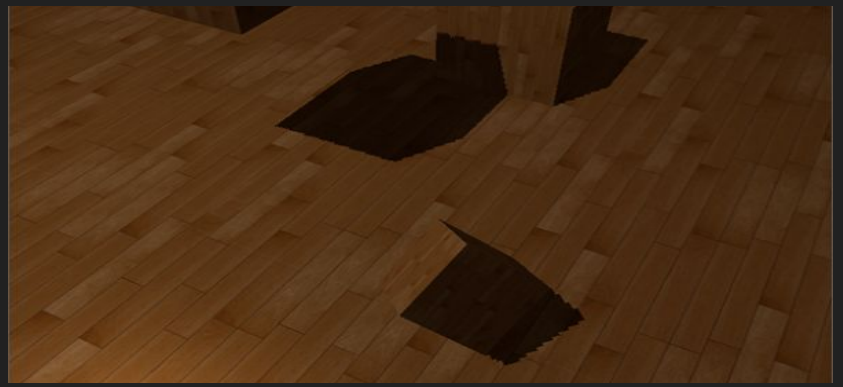
- **CÁMARA**
- **RENDER NORMAL**
- **Para cada píxel:**
  - Se pasa a **COORDENADAS DE LUZ**
  - Si **z de la luz (C) < z vista (P)**
    - Hay **SOMBRA**
    - Si no, **NO**

Limitaciones

Shadow Acne

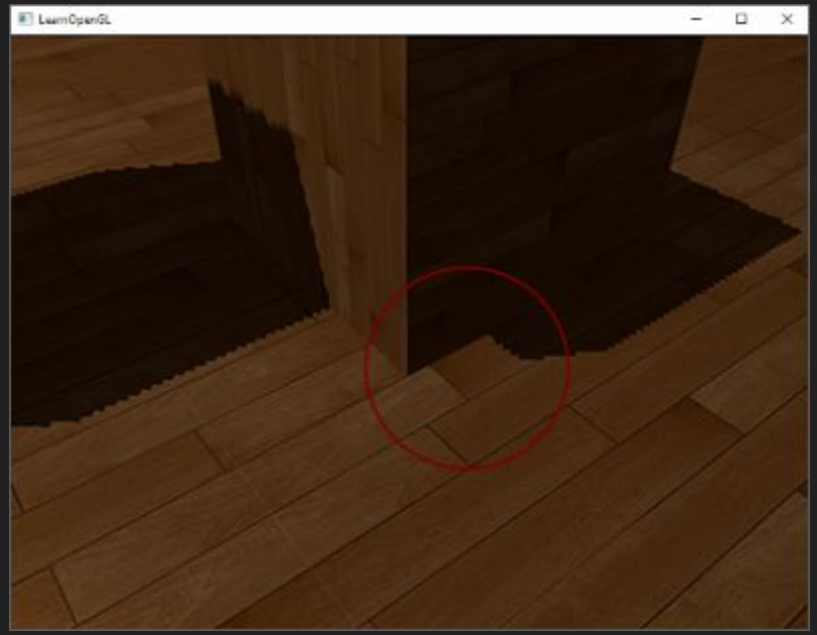




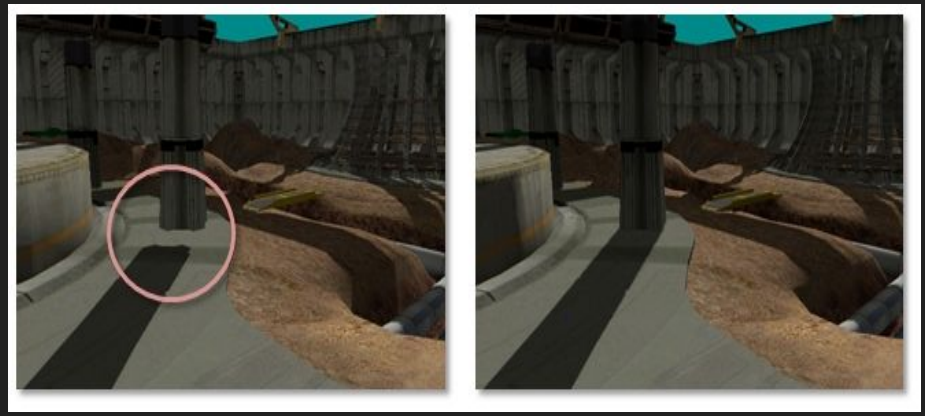


Limitaciones

Peter-Panning



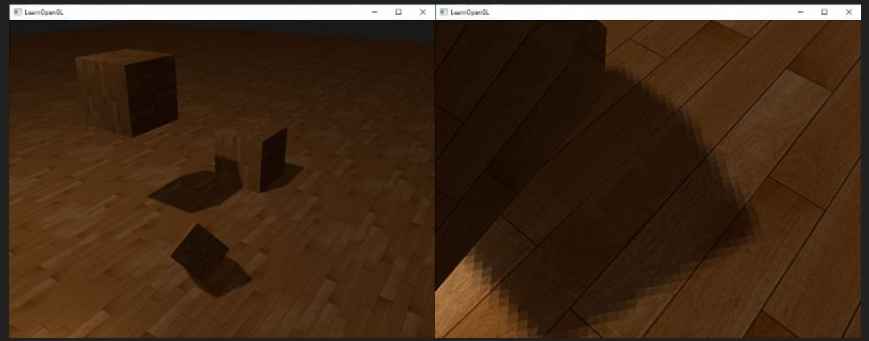
- **CORRECCIÓN** de Shadow Acné ➡️ **Peter-Panning**
- Solución**: ajustar el bias**



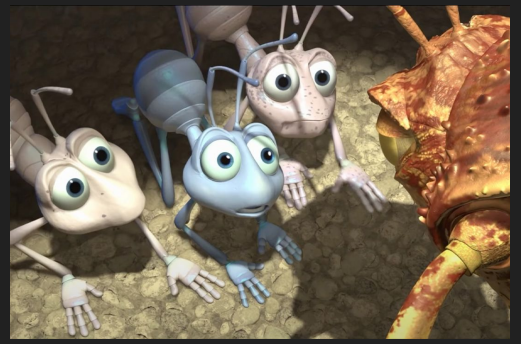
Limitaciones

Aliasing

- **Texels** del **Shadow Map no corresponden 1 a 1** con la **pixels** de la **vista**
- Shadow-Map tiene una resolución determinada
- **Percentage-Closer Filtering**



Pixar RenderMan



Hasta la **versión 20 (2015/16)** se podía seguir utilizando

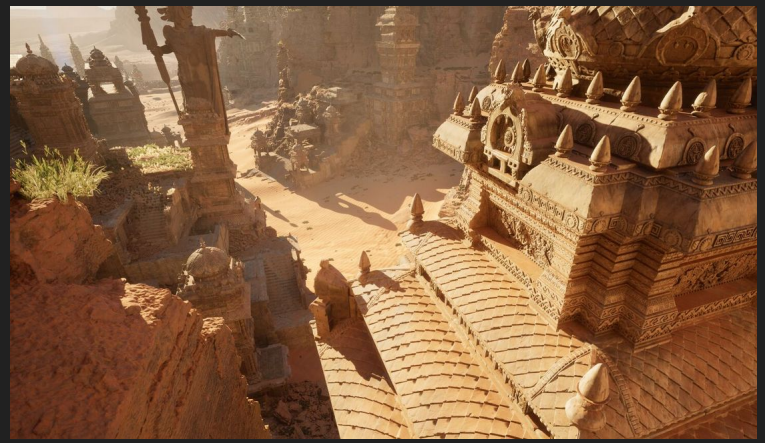
Se usó en películas como ***Toy Story, Bichos, Monster Inc, etc..***

Unity

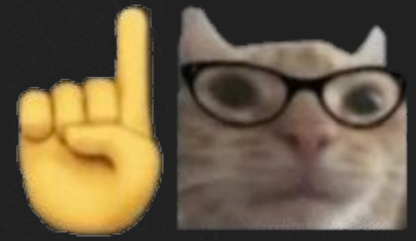
- Para **sombras en tiempo real** usa shadow-maps
- Tiene **4 niveles de calidad**, multiplicador de la resolución del mapa
- Se puede usar un tamaño de mapa custom
- Para **luz puntual**, calcula **6 mapas**

Luz	Máximo Tamaño de Mapa
Luz Direccional	4096x4096
Luz Spot	2048x2048
Luz Puntual	1024x1024

Unreal



- Técnica **por defecto** de Sombras en la **última versión**
- **VSM (Virtual Shadow Maps)**
  - Mapa de **16000x16000 pixeles**
  - Divididos en **sectores de 128x128**
  - Cargados/cacheados **según sea necesario**



¿Preguntas?

Gracias

Bibliografía

Diapositivas de Clase

https://en.wikipedia.org/wiki/Shadow_mapping

Williams, "Casting Curved Shadows on Curved Surfaces", SIGGRAPH, 1978

Learn OpenGL https://learnopengl.com/Advanced-Lighting/Shadows/Shadow-Mapping

https://graphics.cs.utah.edu/courses/cs4600/fall2020/

RenderMan Documentation
https://renderman.pixar.com/resources/RenderMan_20/softShadows.html

Unity Documentation https://docs.unity3d.com/Manual/shadow-mapping.html