

# 3D In Depth

• • •

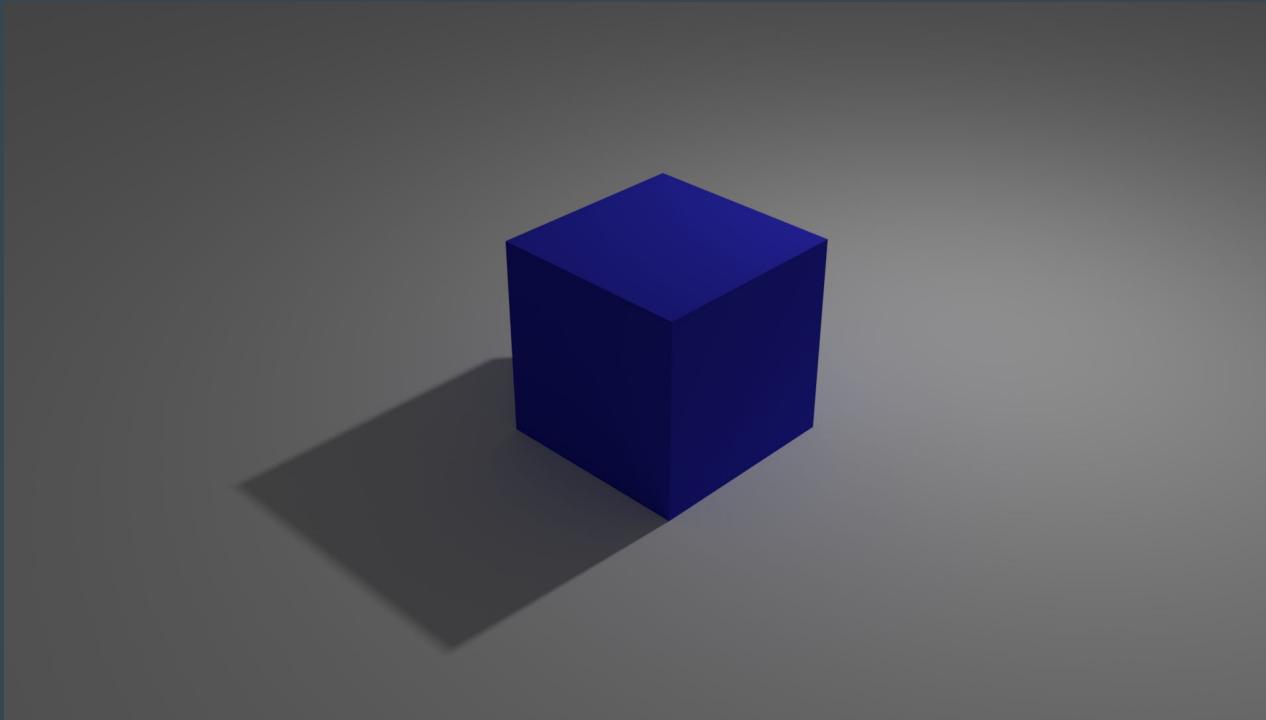
ElectronicArmory.com  
3D Game Development Course

# The Utah Teapot

- Mathematical model
  - Simple shape
  - Solid
  - Cylindrical
  - Convex
- Easy 3D scene with a non-trivial model for a basic scene and light setup
- The Utah teapot was created in 1975 by Martin Newell, computer graphics researcher at the University of Utah



# Hello World



# 3D Concepts

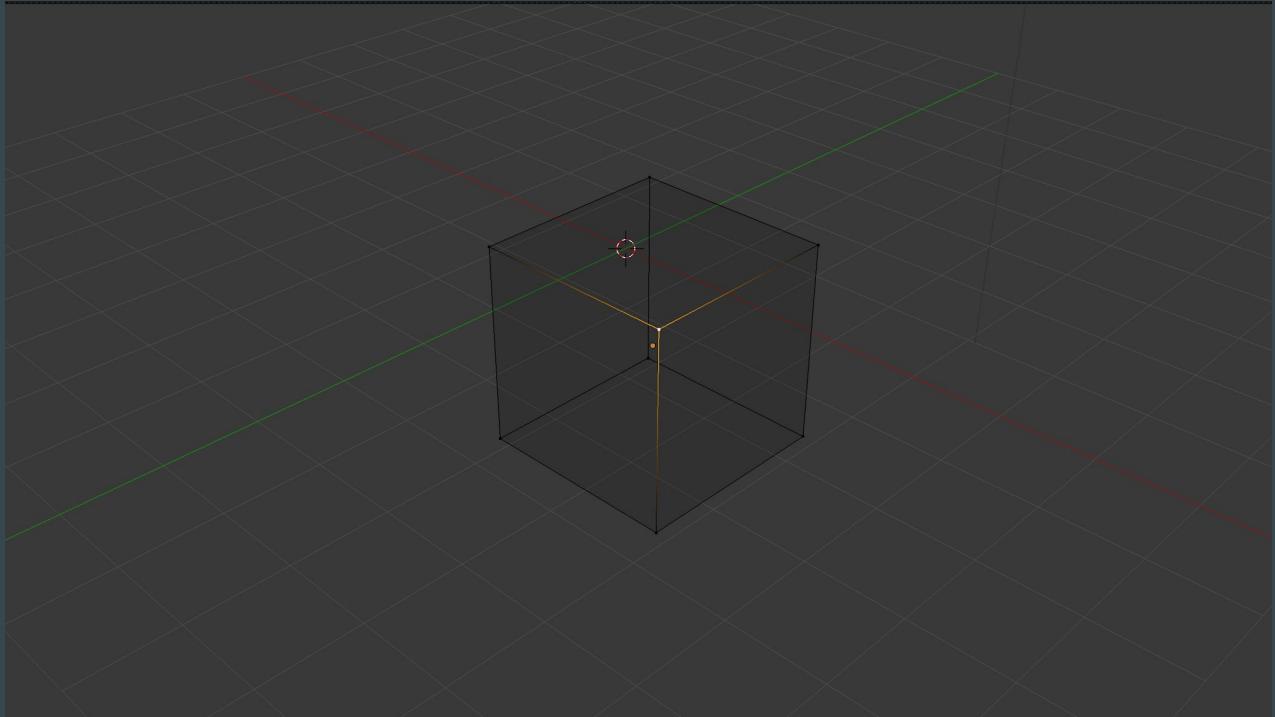
- Modeling Geometry (Meshes)
  - Vertices
  - Edges
  - Faces
- Lighting
- Texture/Material
- Rendering

# Vertices

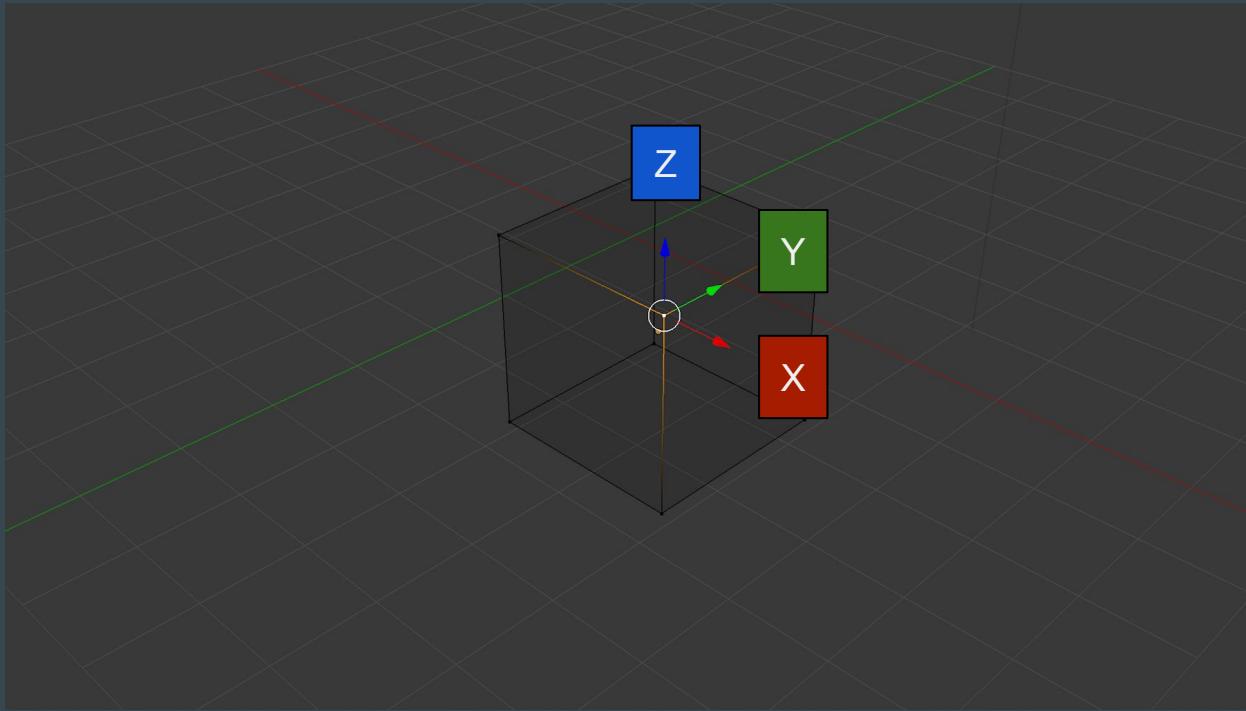
Points that occupy a specific coordinate in space.

In 3D space, vertices have an X, Y and Z coordinate:

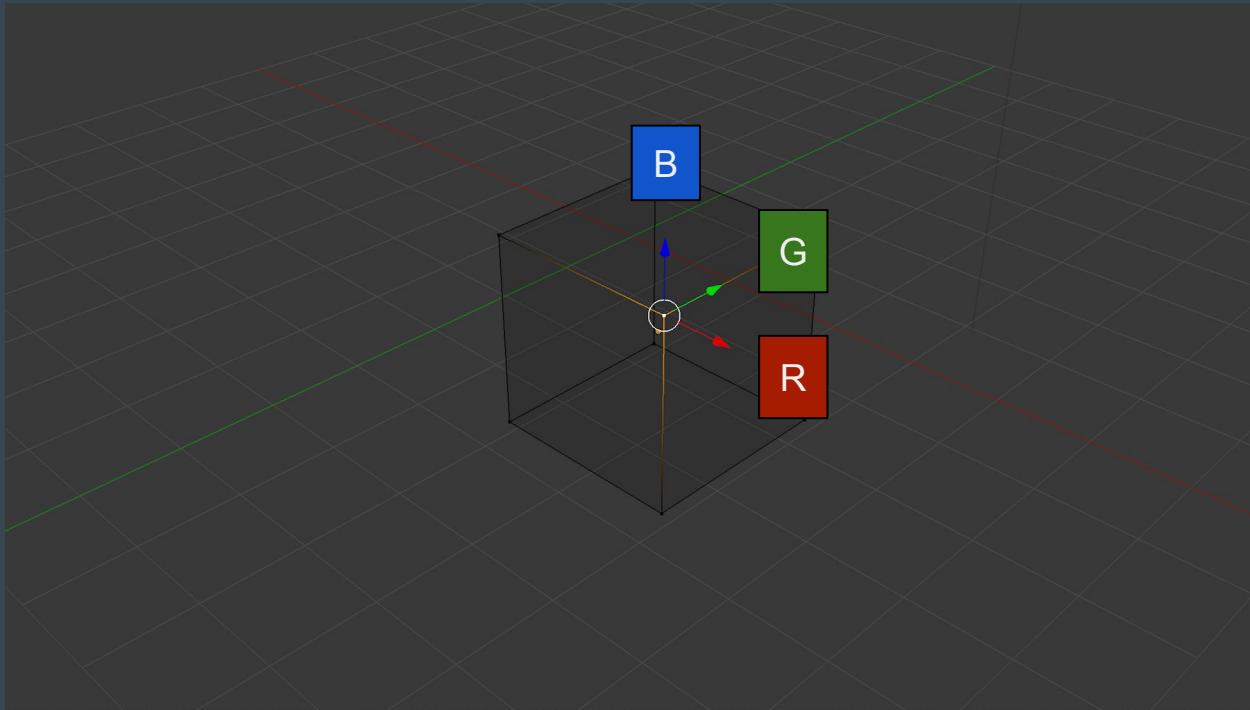
$(3, 4, 19)$



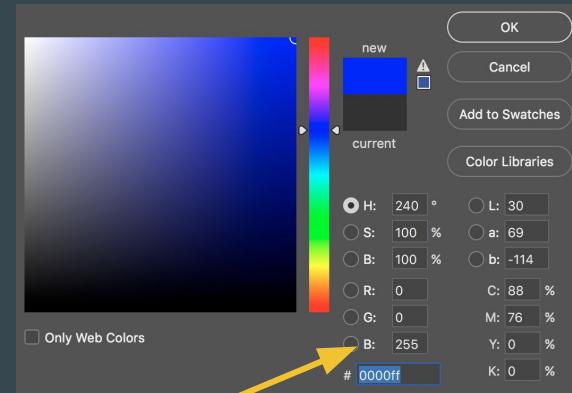
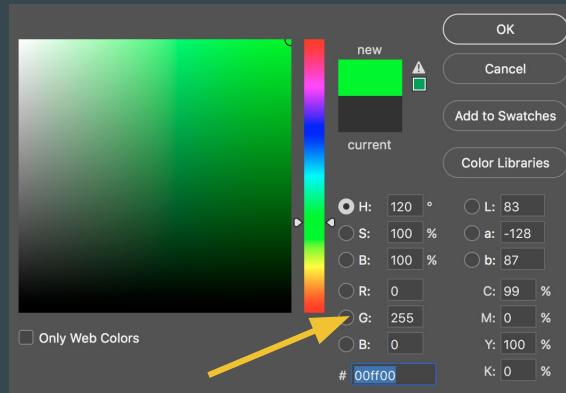
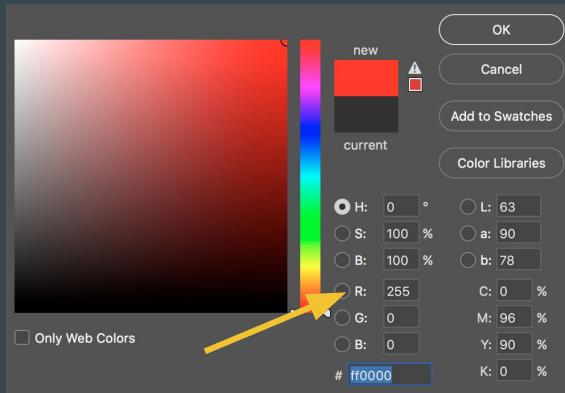
# 3D Axis



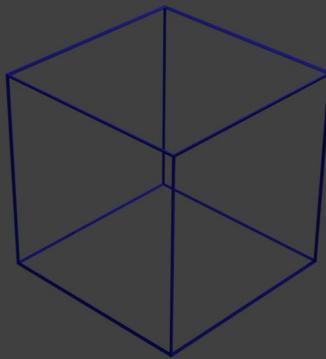
# 3D Axis - RGB



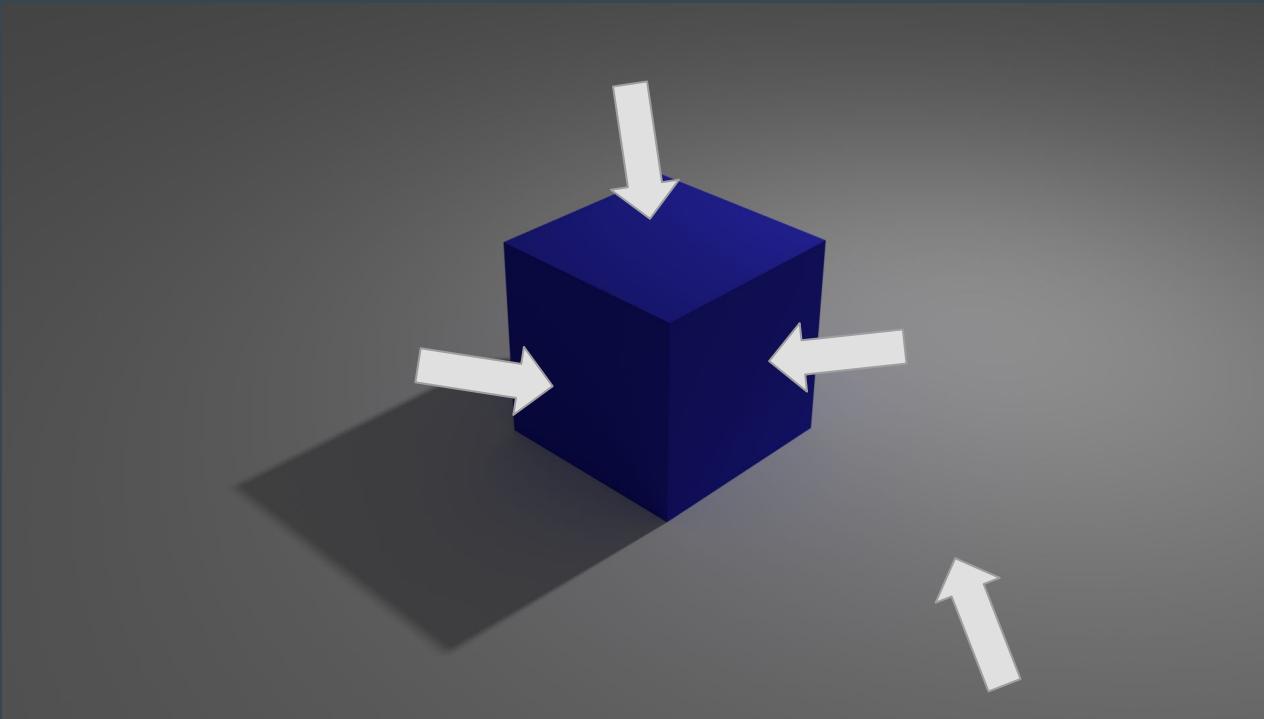
# RGB in Color Pickers



# Edges

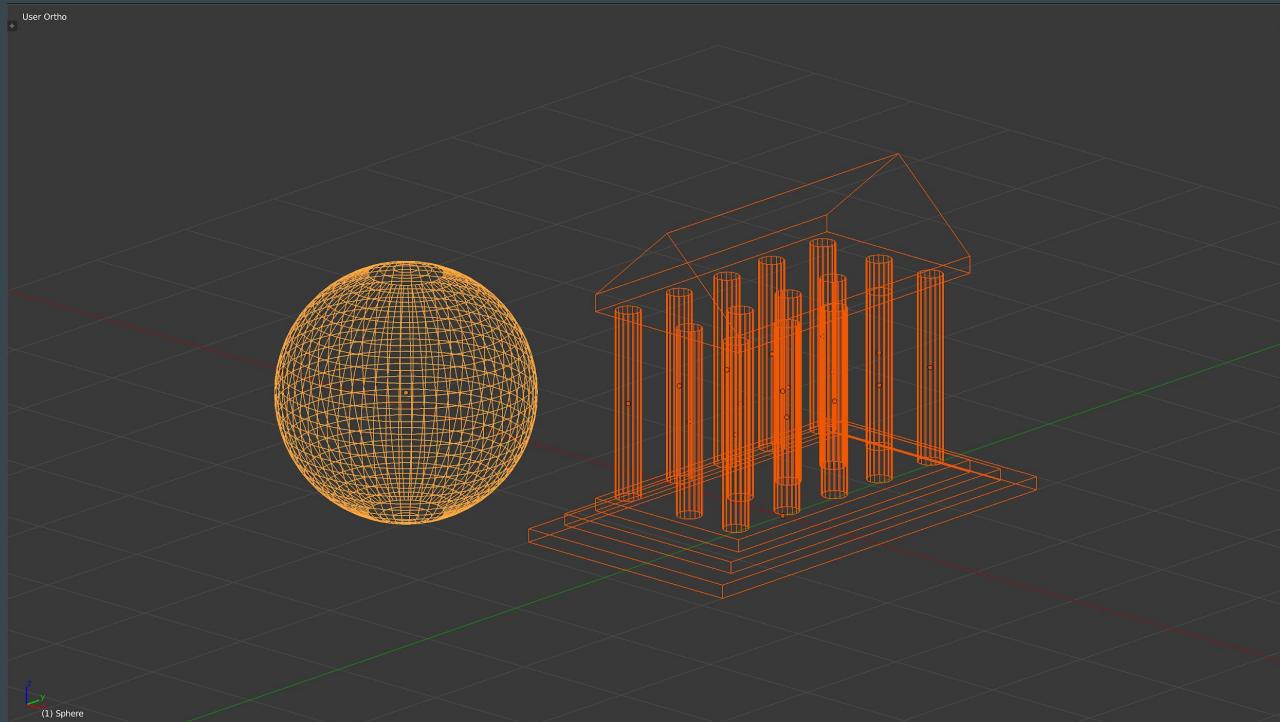


# Faces

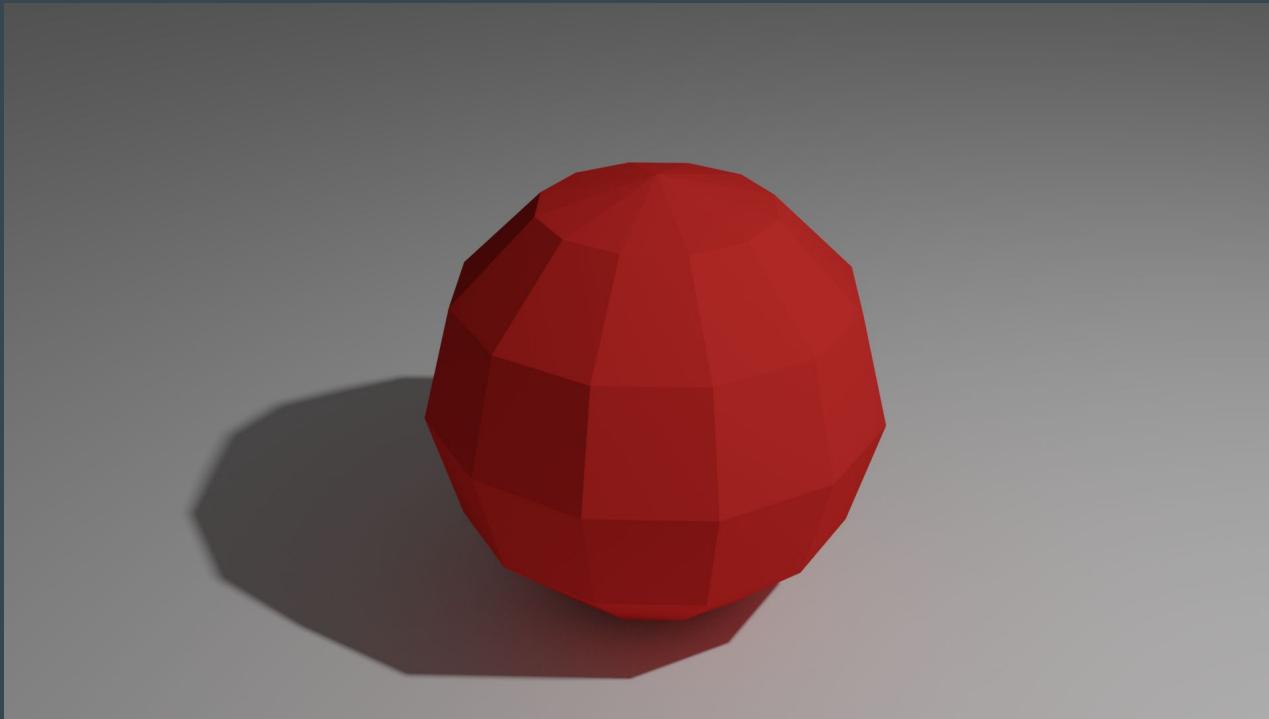


# Modeling

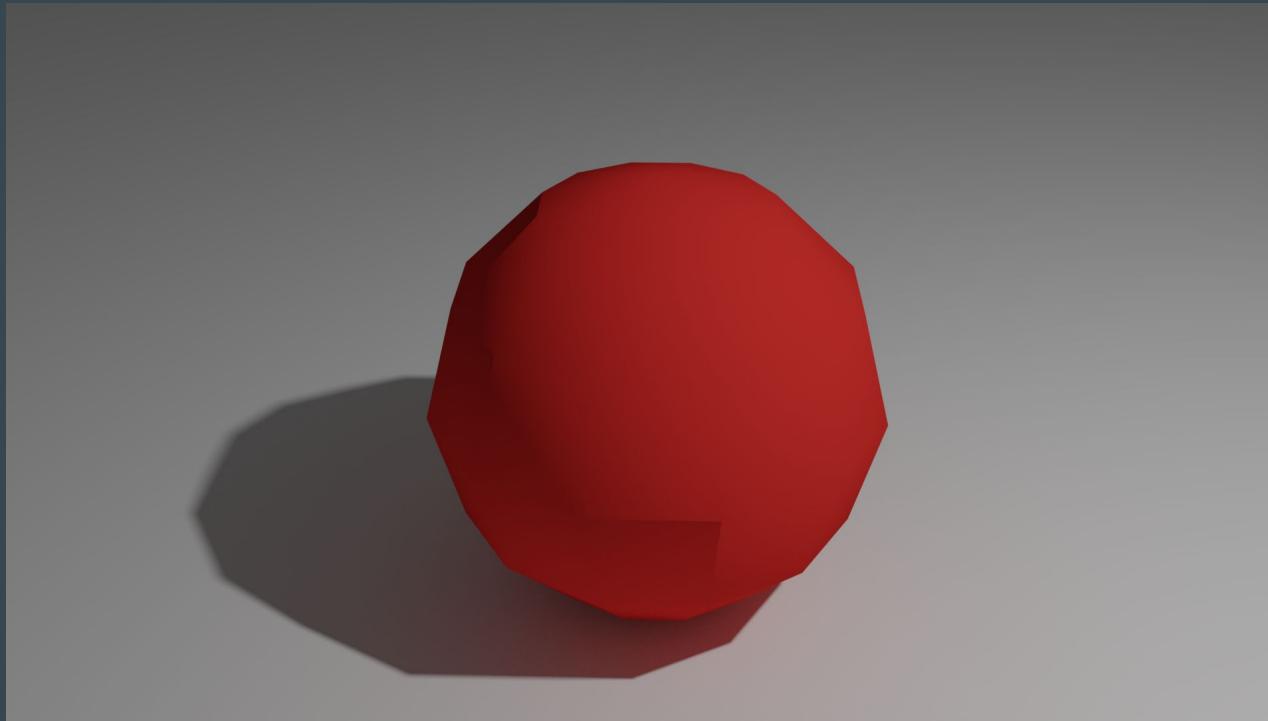
# Modeling



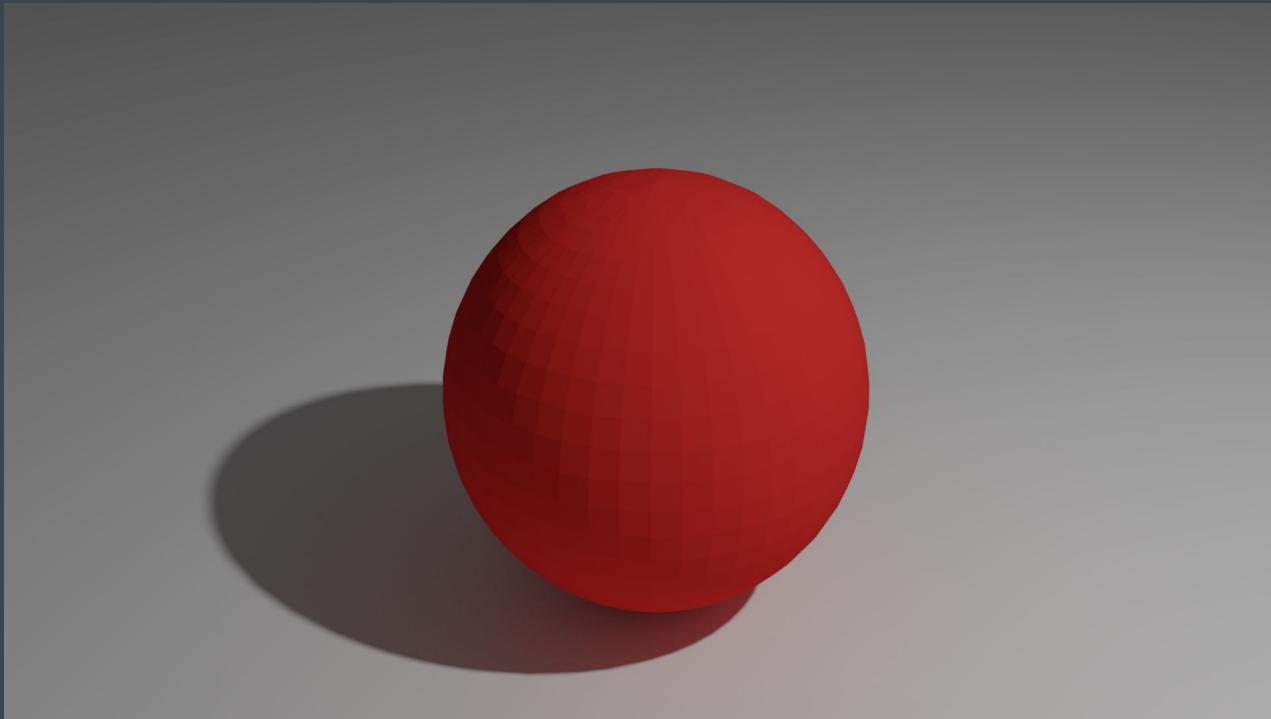
# Low Poly - Flat Shaded



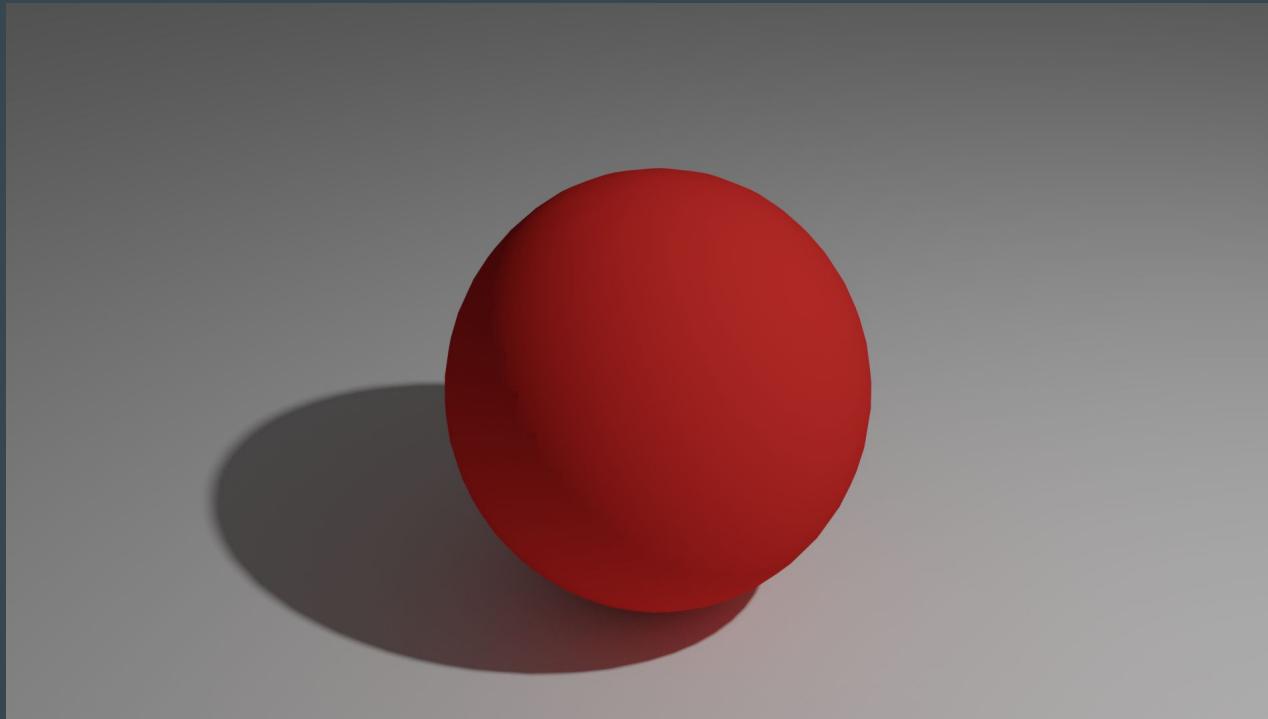
# Low Poly - Smooth Shaded



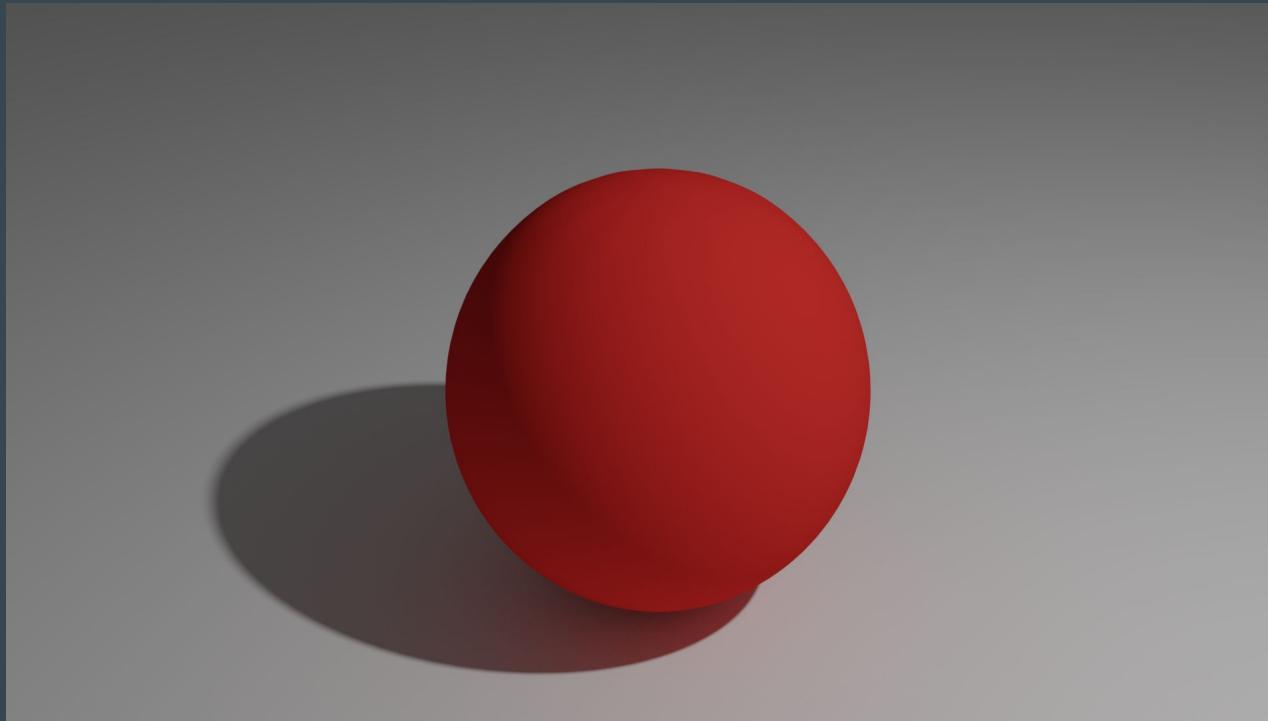
# High Poly - Flat Shaded



# High Poly - Smooth Shaded



# Super High Poly - Smooth Shaded



# Level of Detail (LODs)

For efficiency, objects that are farther away can have few polygons since they lack the detail needed on objects that are closer to the camera.

Game engines allow for different LODs on meshes for efficiency.

# Low Poly Scene

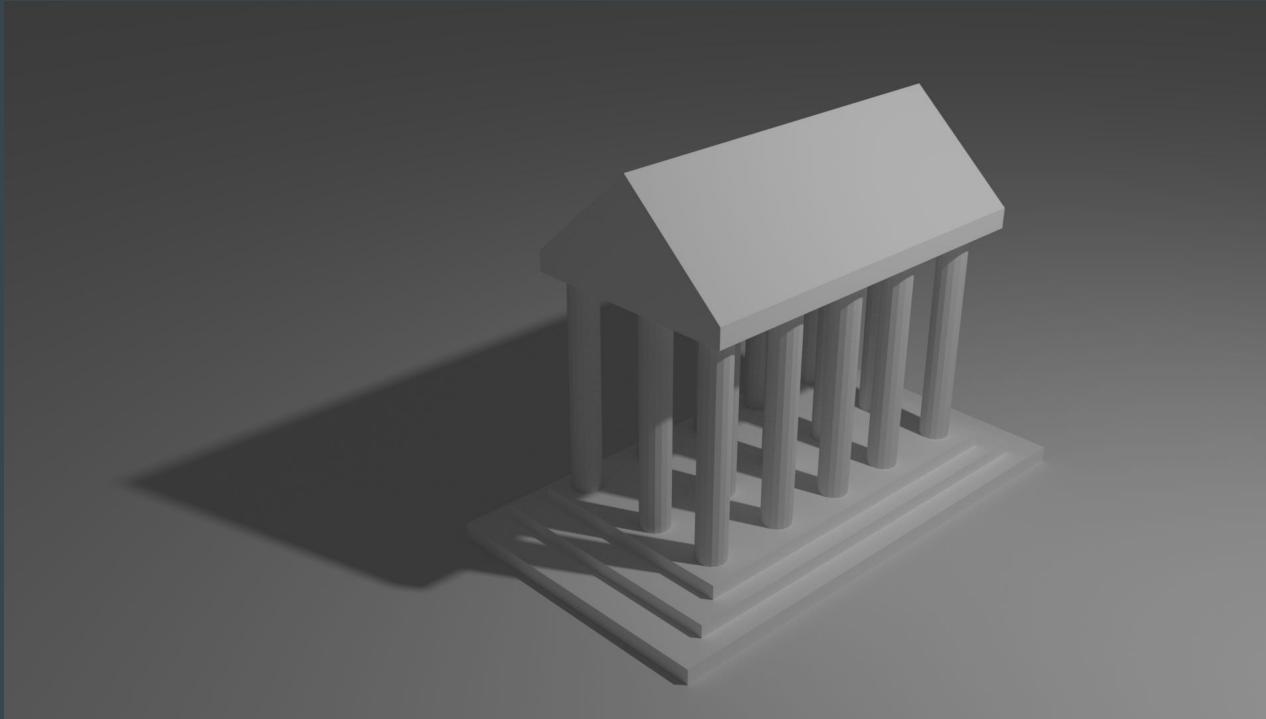


# Low Poly Scene

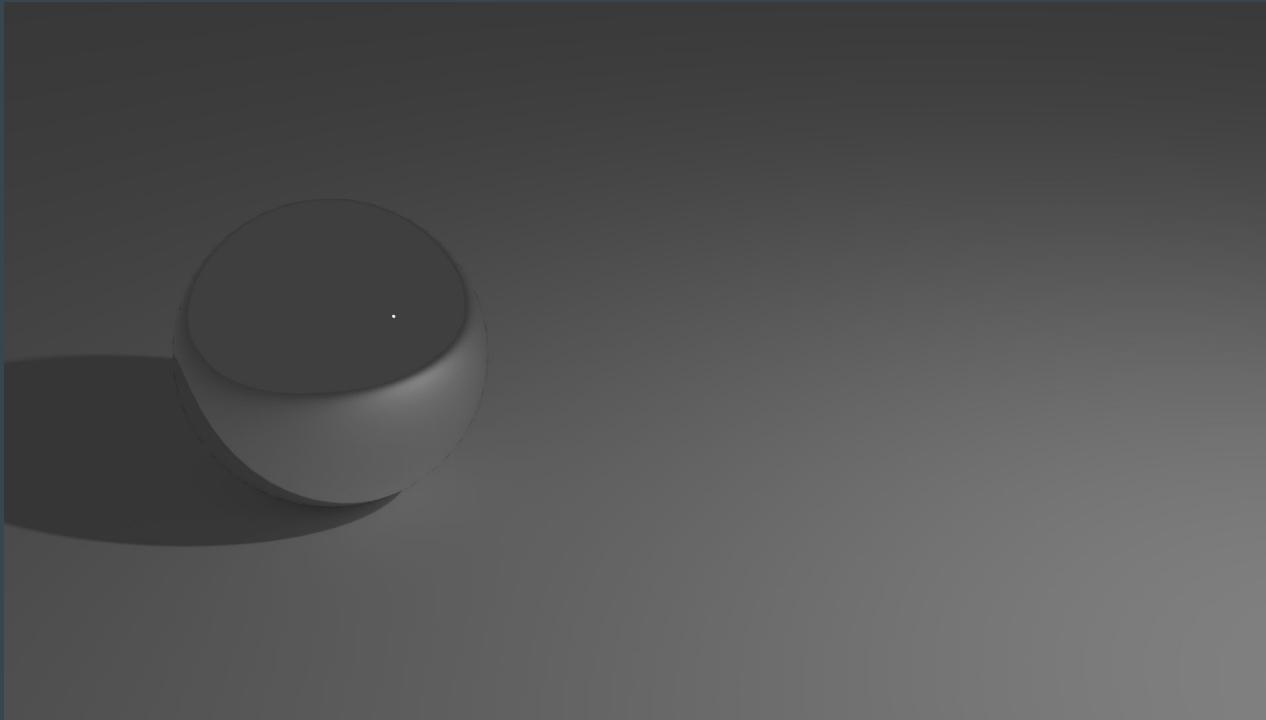


# Textures and Materials

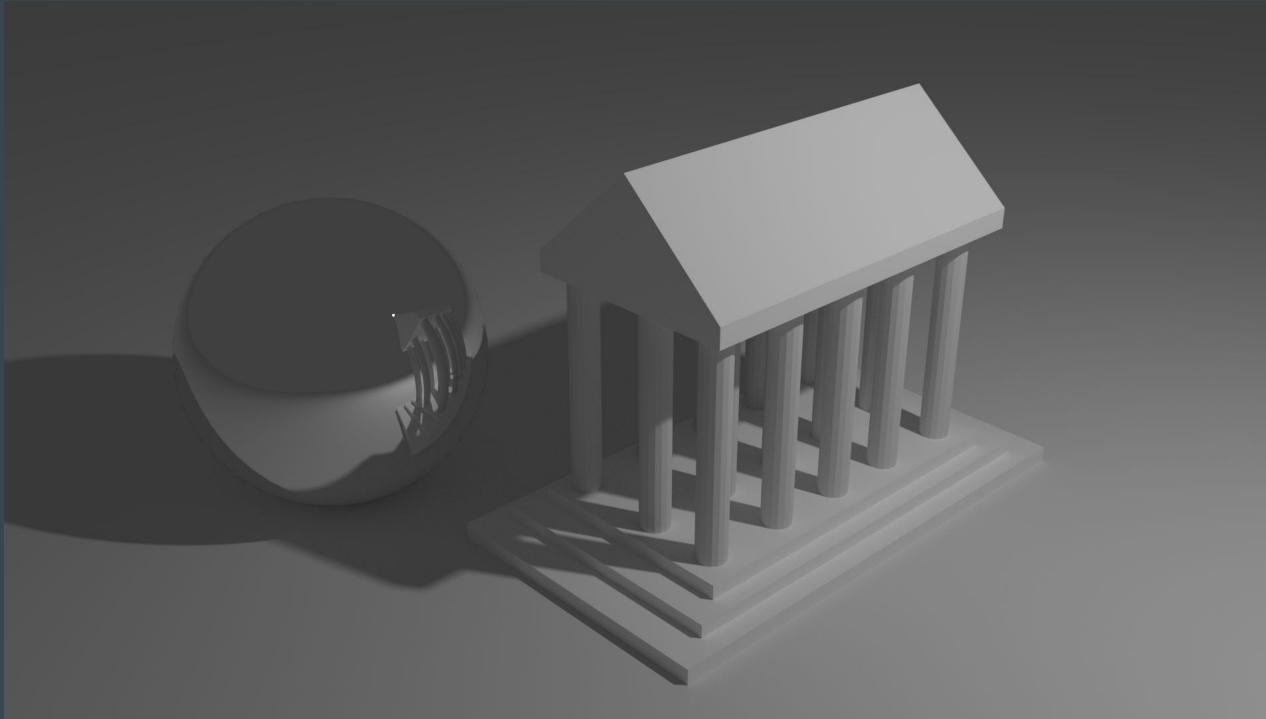
# Diffuse Material



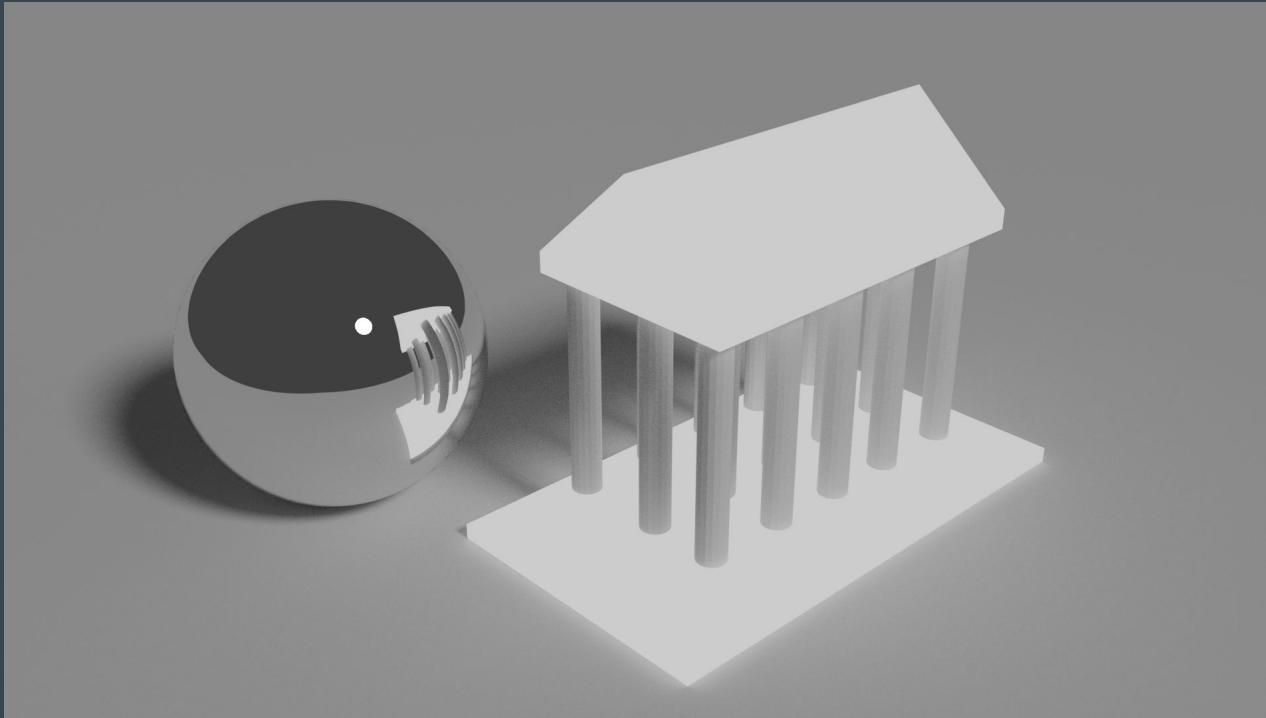
# Gloss



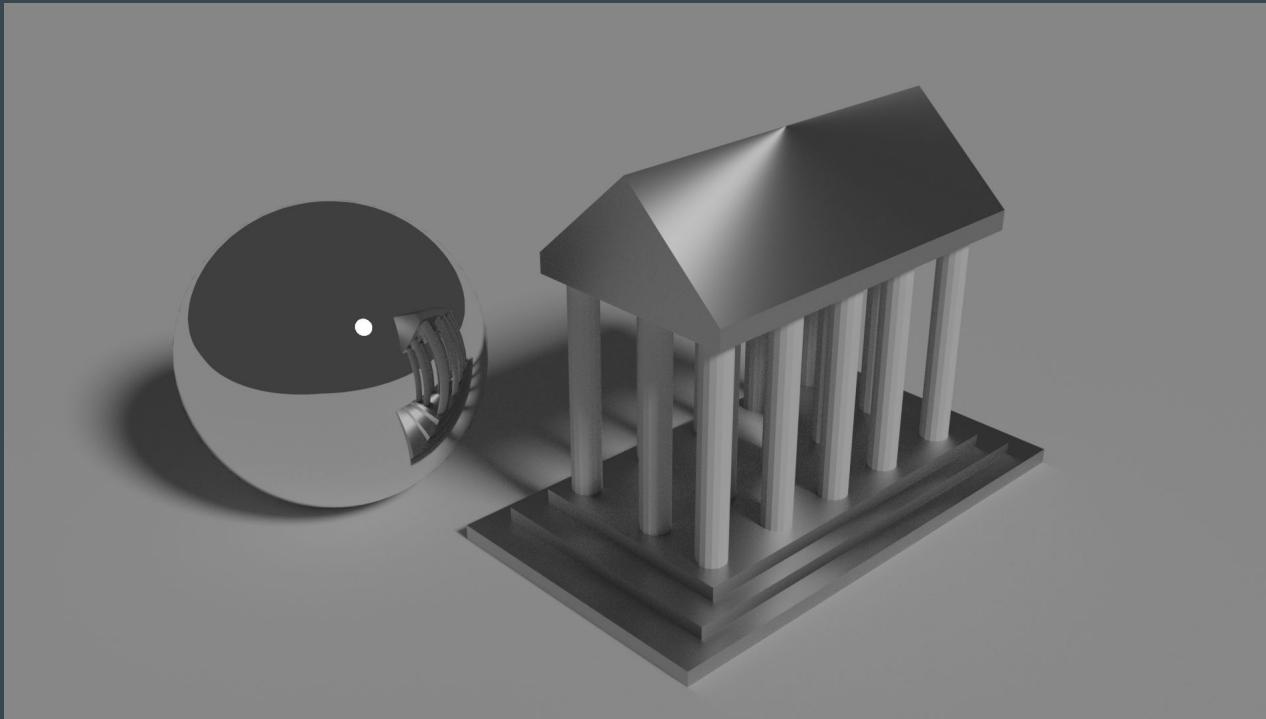
# Gloss and Diffuse Shading



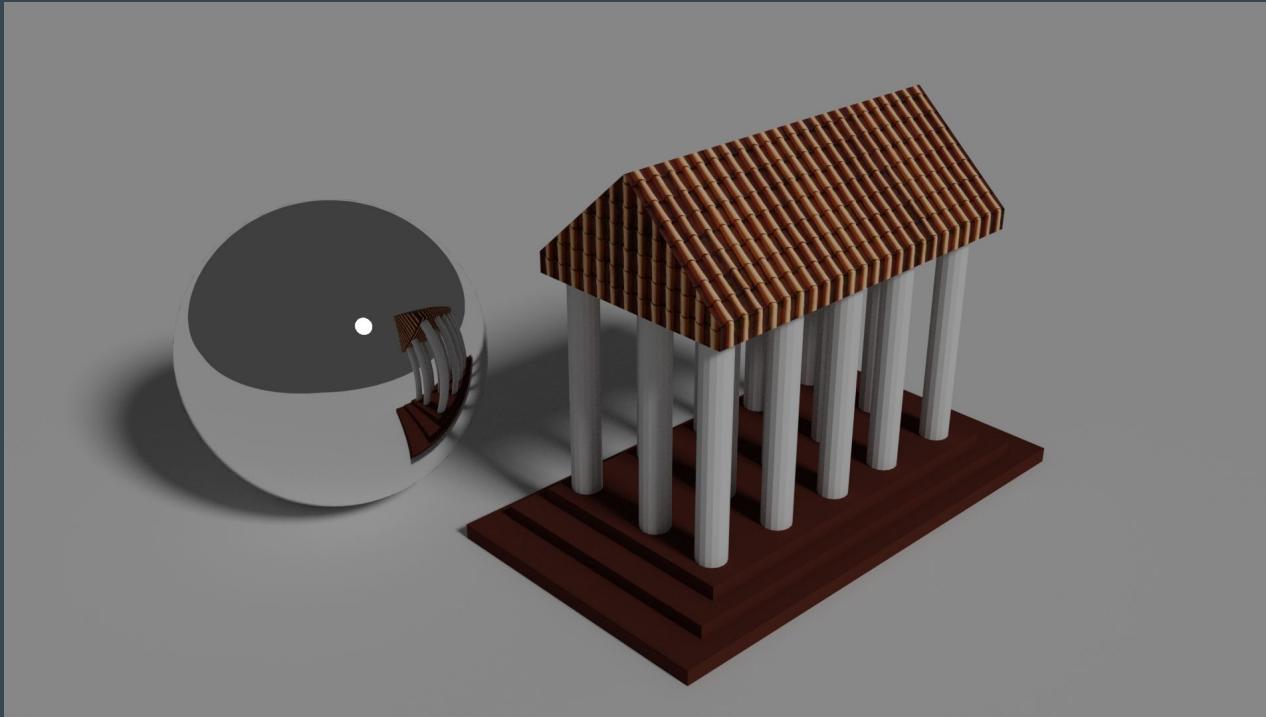
# Emission



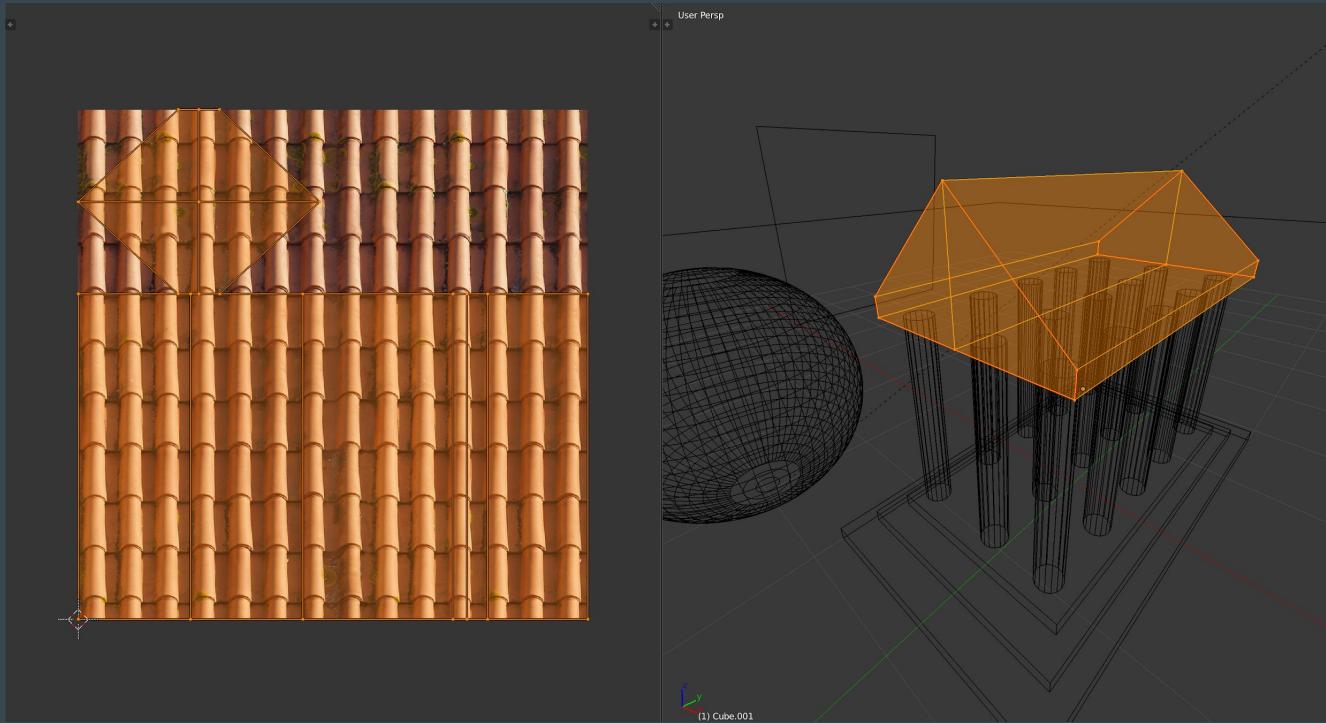
# Anisotropic



# Textures



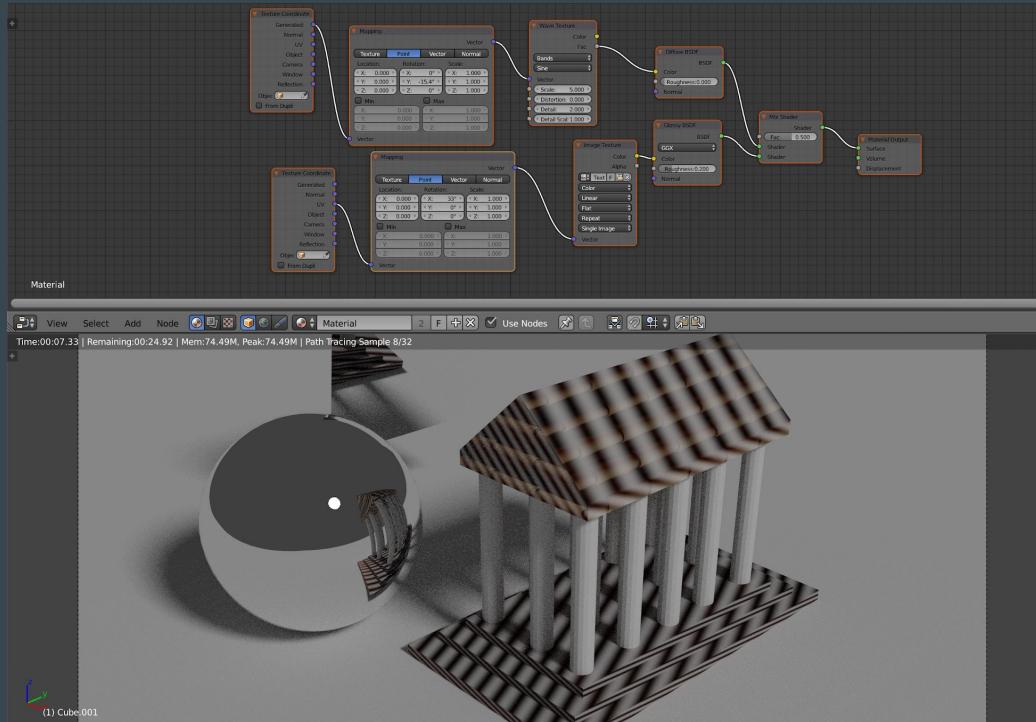
# UV Mapping



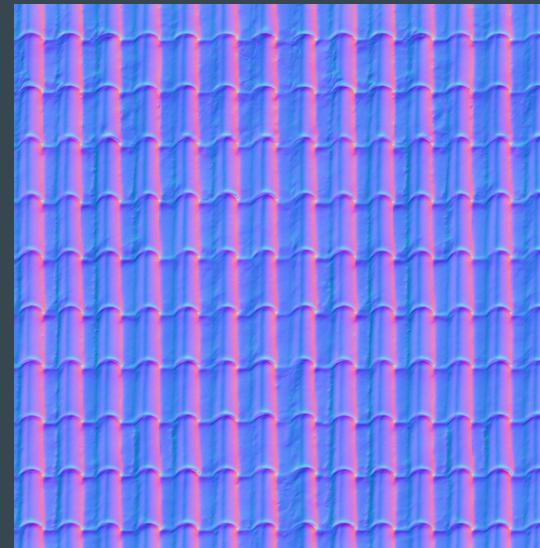
# Image textures



# Procedural



# Normal Maps



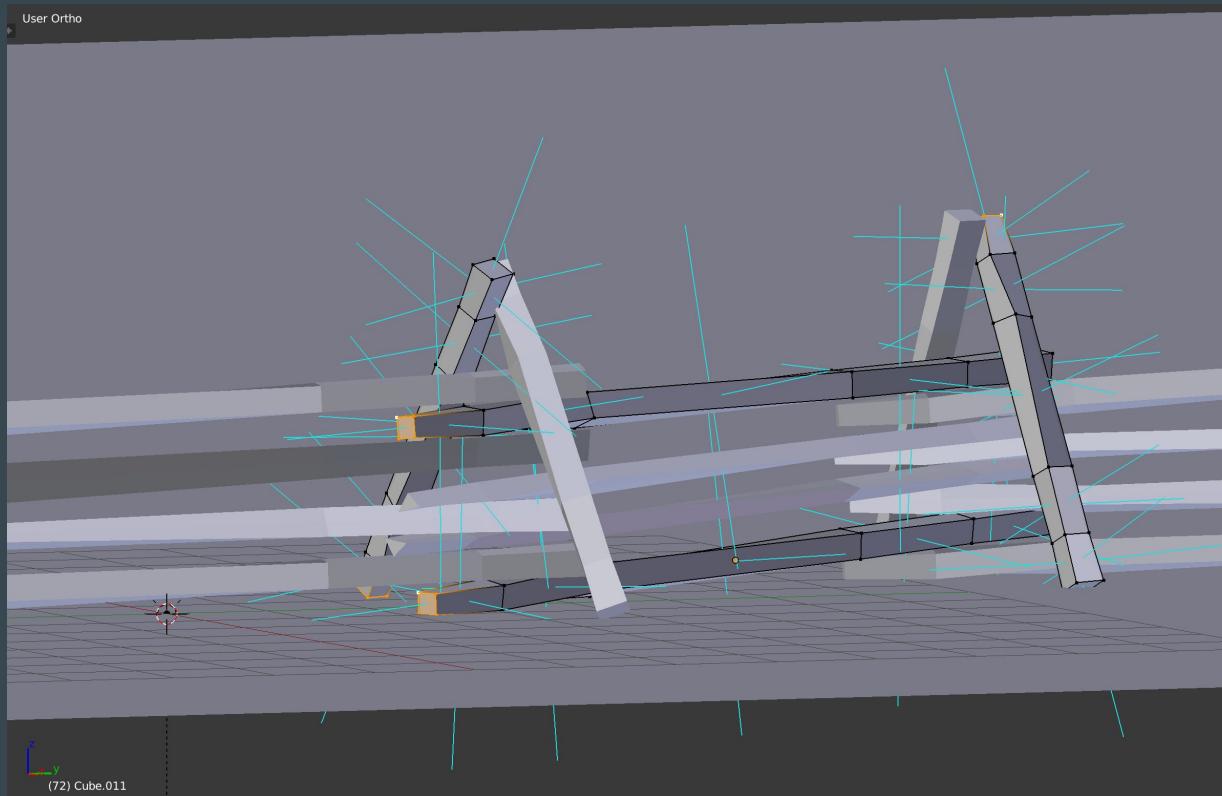
# Normal Maps

Tells the renderer to bounce light off the surface differently than the geometry would normally.

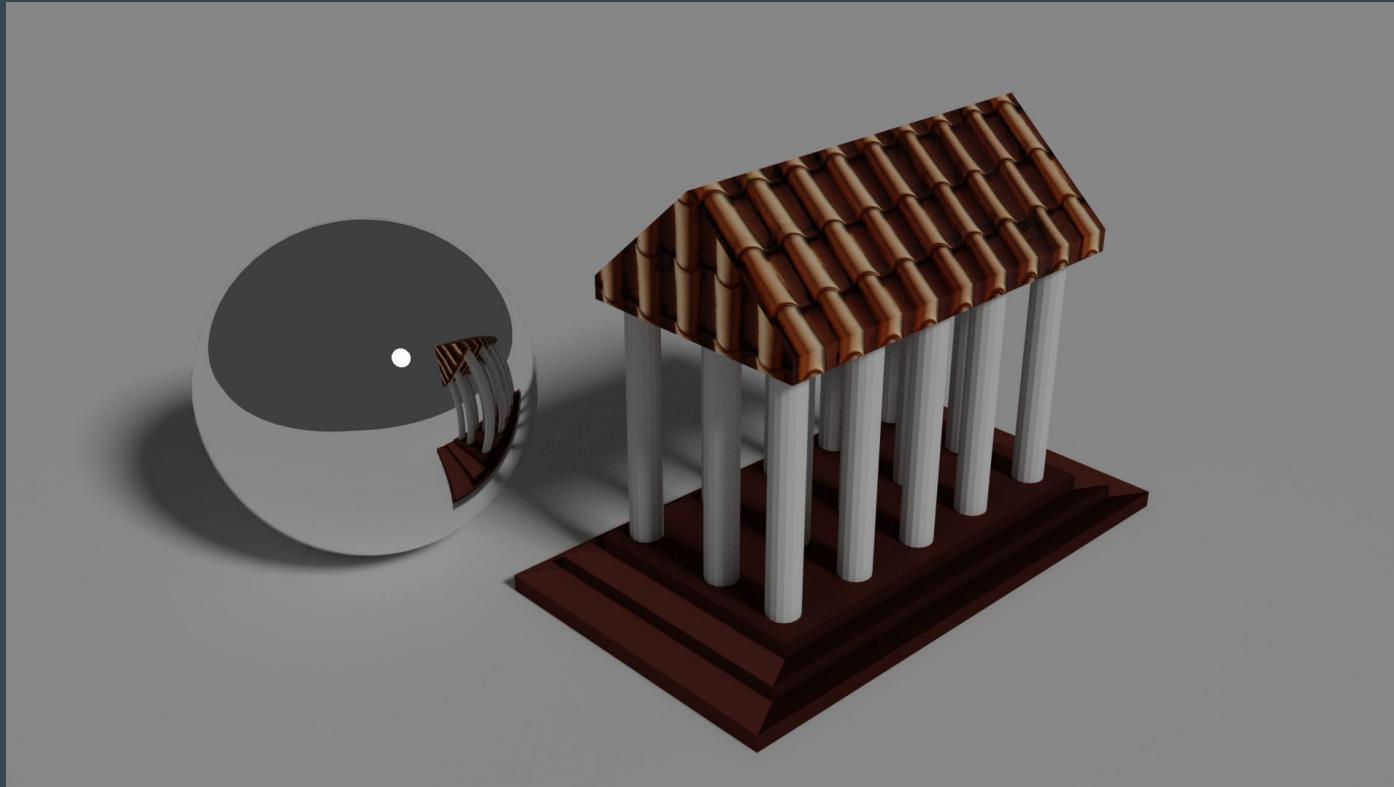
“Normals” refers to the line, perpendicular to the surface plane.

In almost all game engines, the back face of a plain is not rendered. This is the face opposite to the direction of the plain’s normal.

# Normal Maps

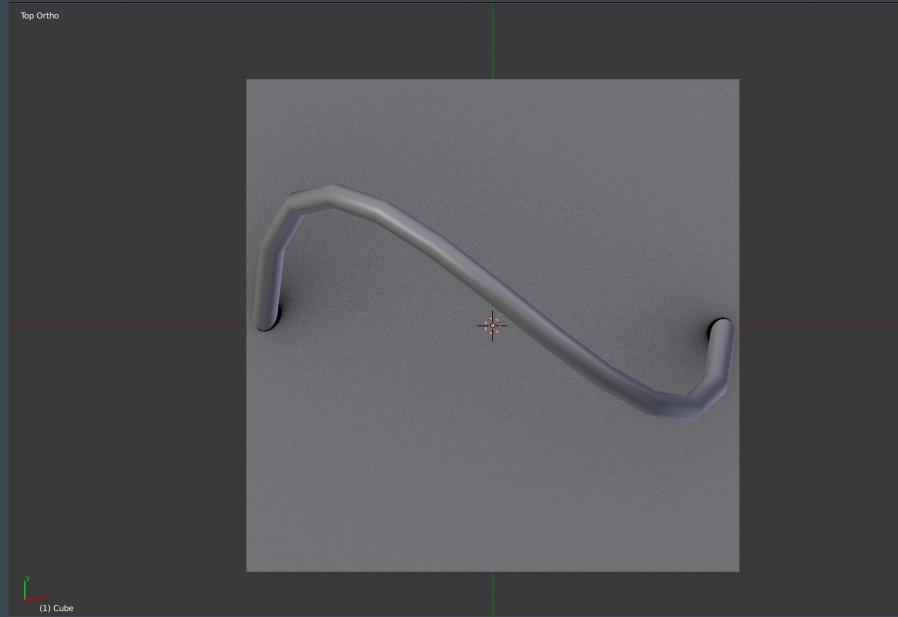


# Normal Maps

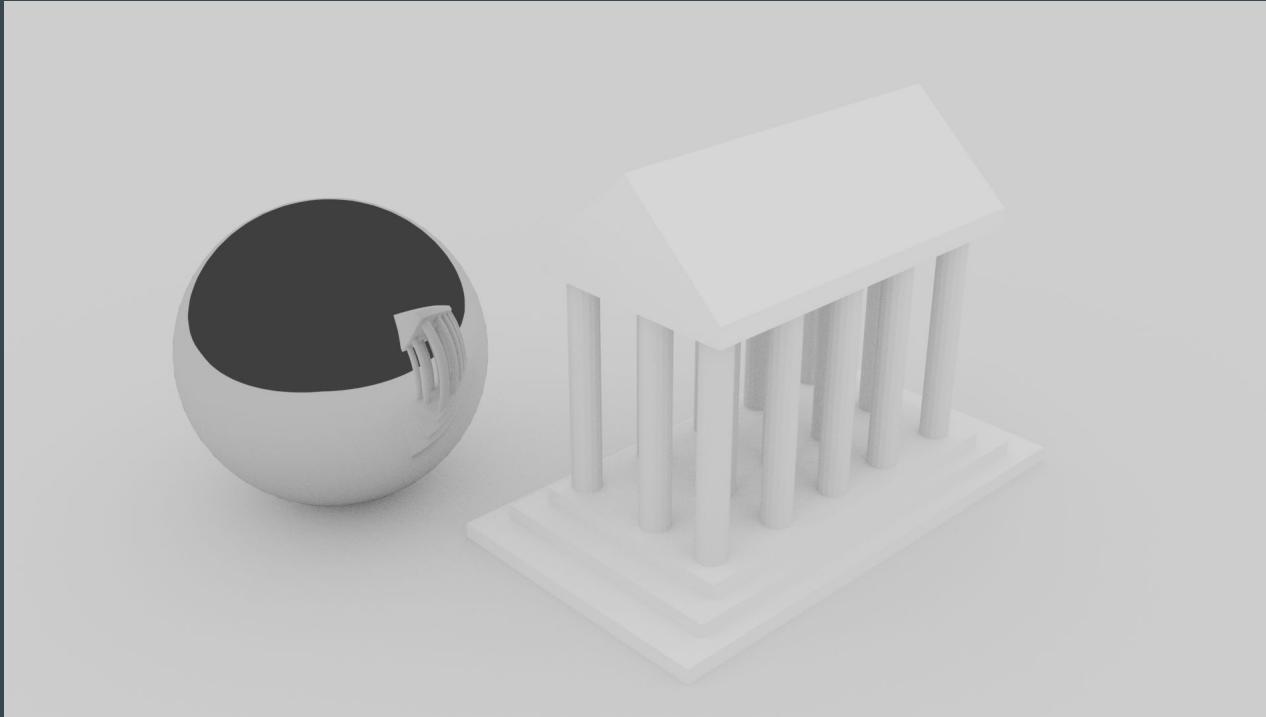


# Ambient Occlusion (AC)

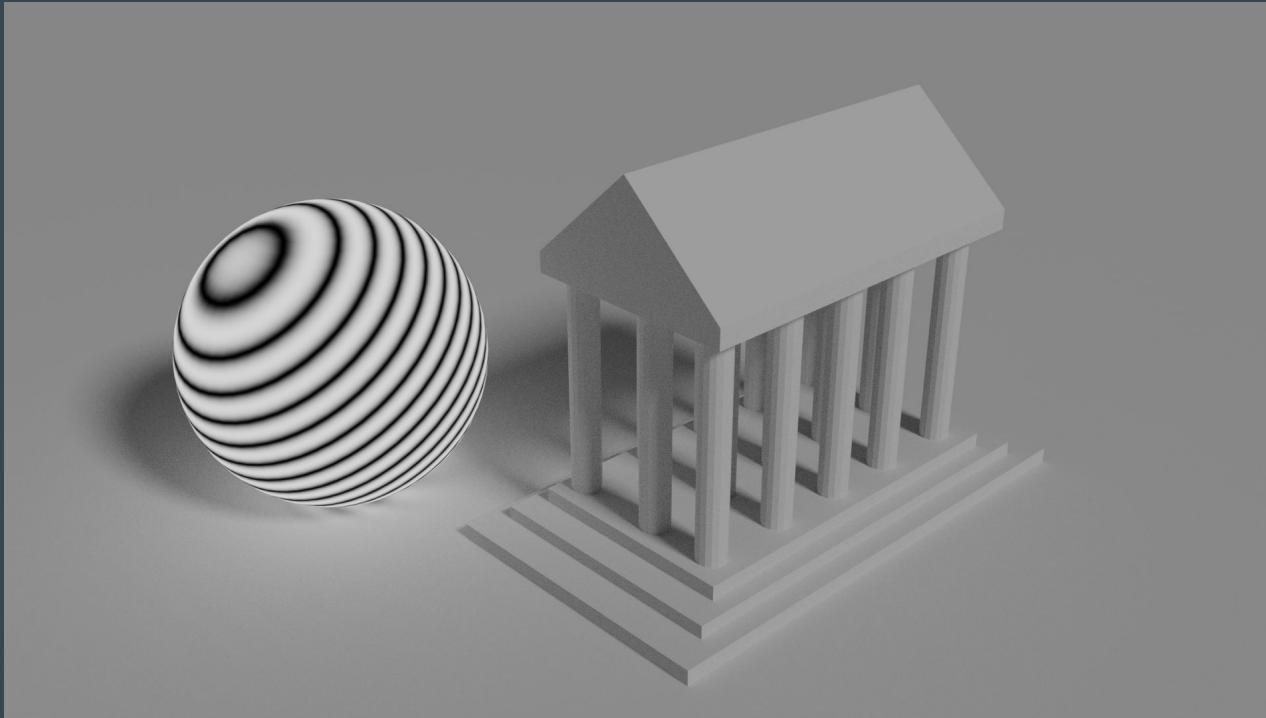
Ambient occlusion is a technique to approximate how light shines on different parts of an object's surface.



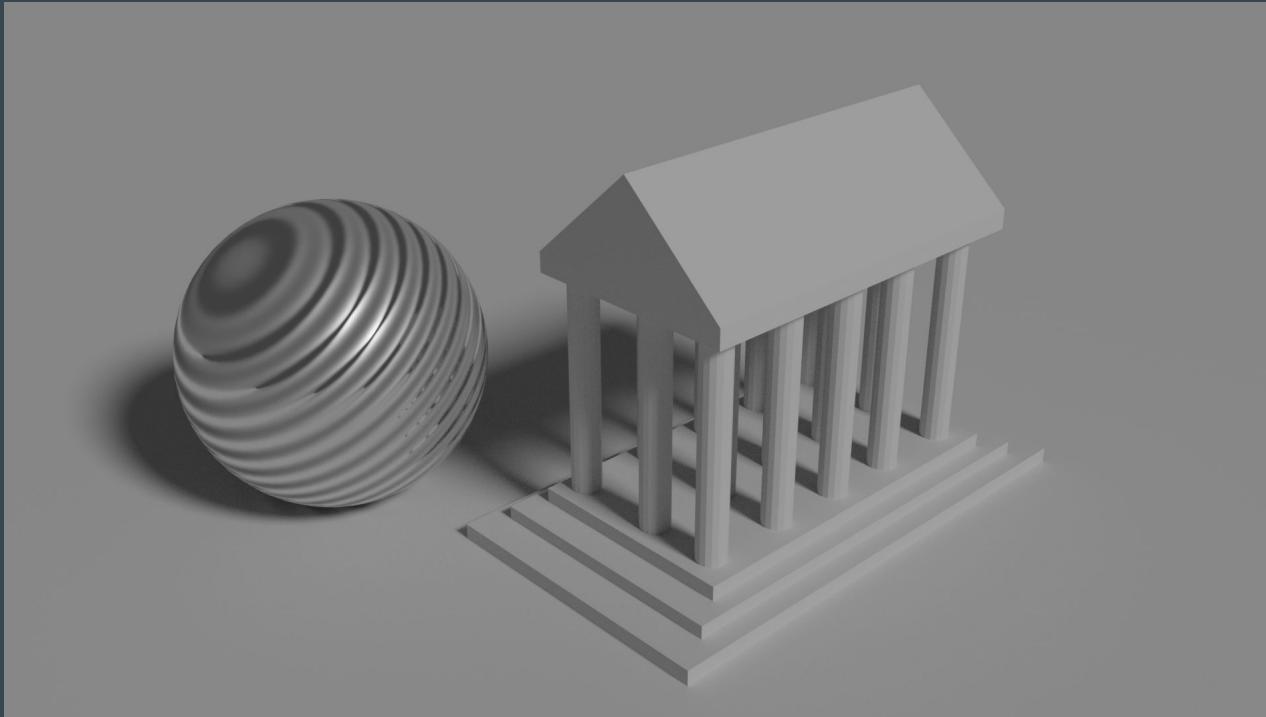
# Ambient Occlusion



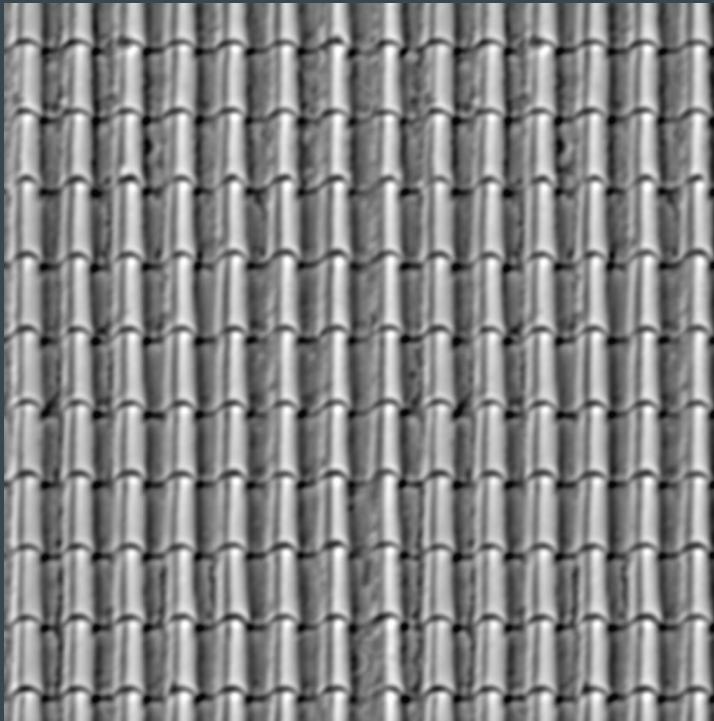
# Roughness



# Roughness



# Height Maps

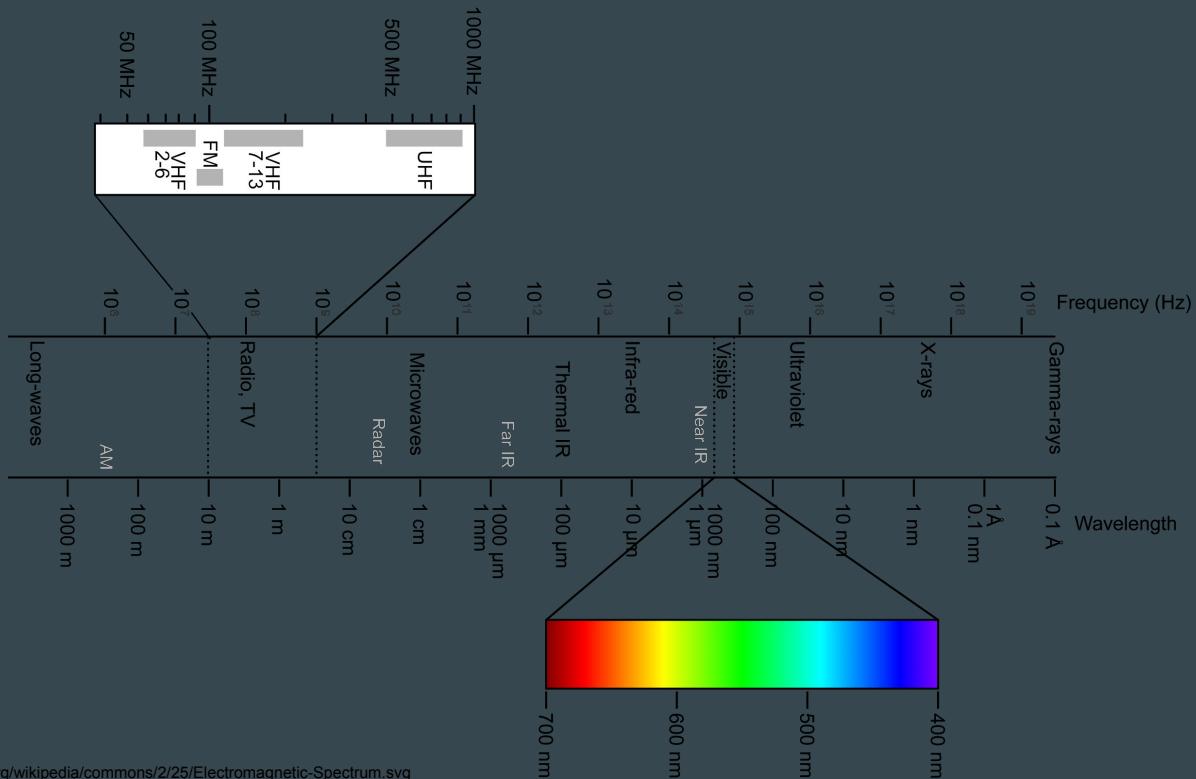


# PBR Textures

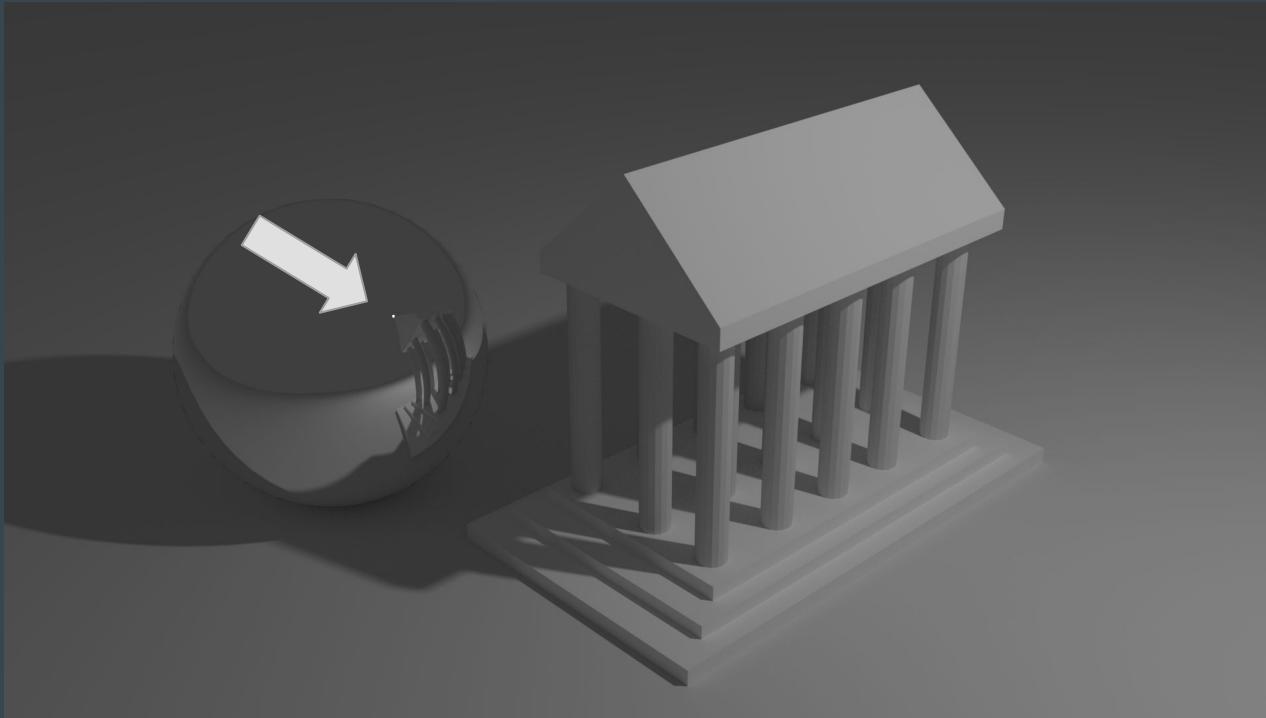
- Base Color
- Roughness
- Metallic
- Specular

# Lighting

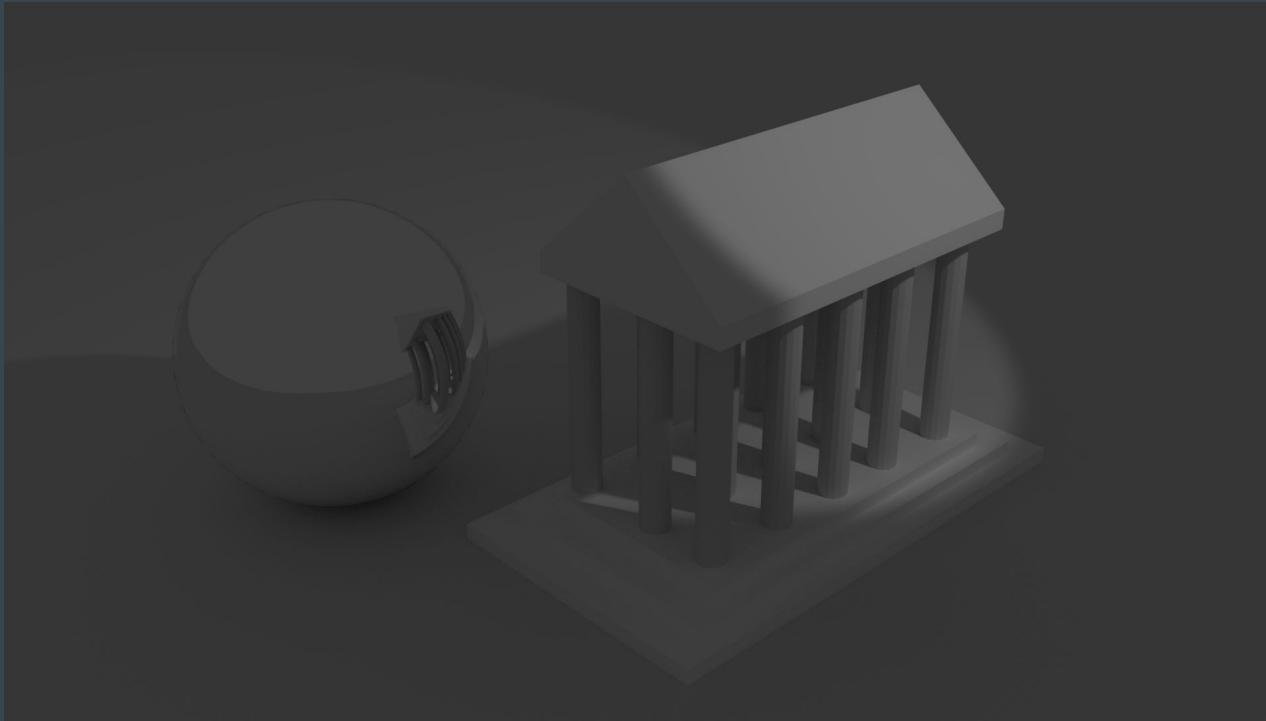
# Visible Light Spectrum



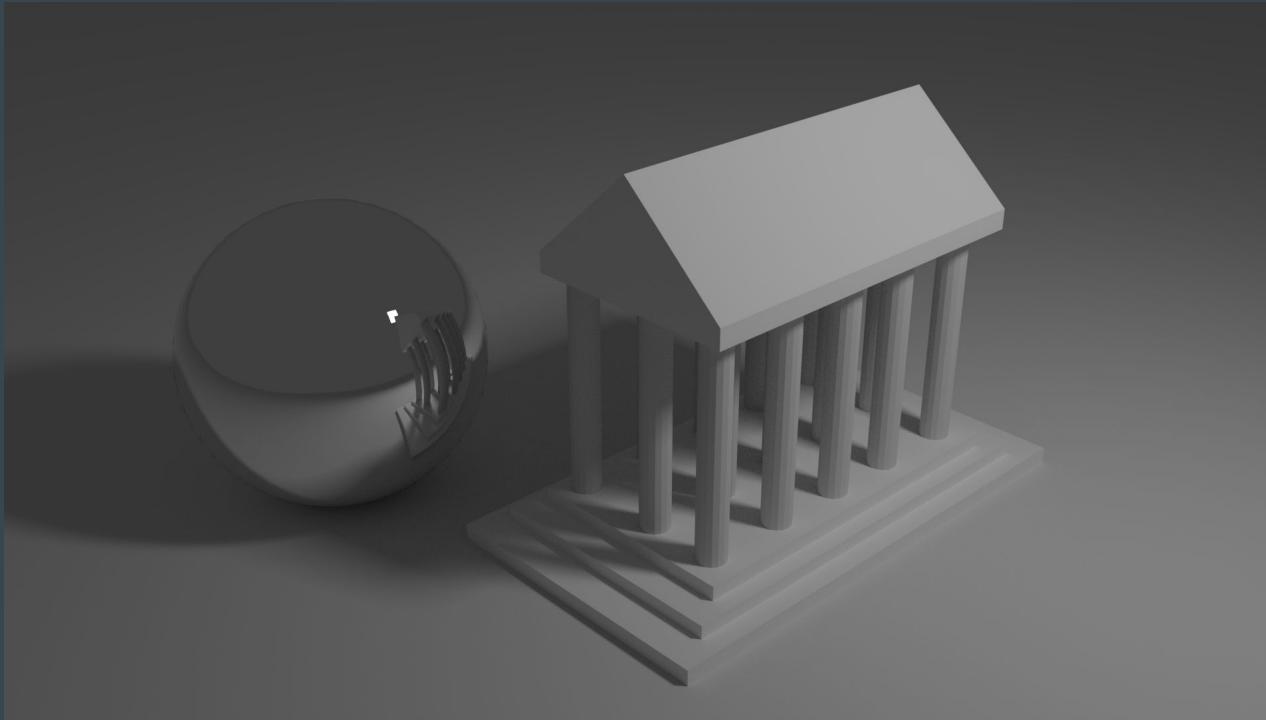
# Point Light



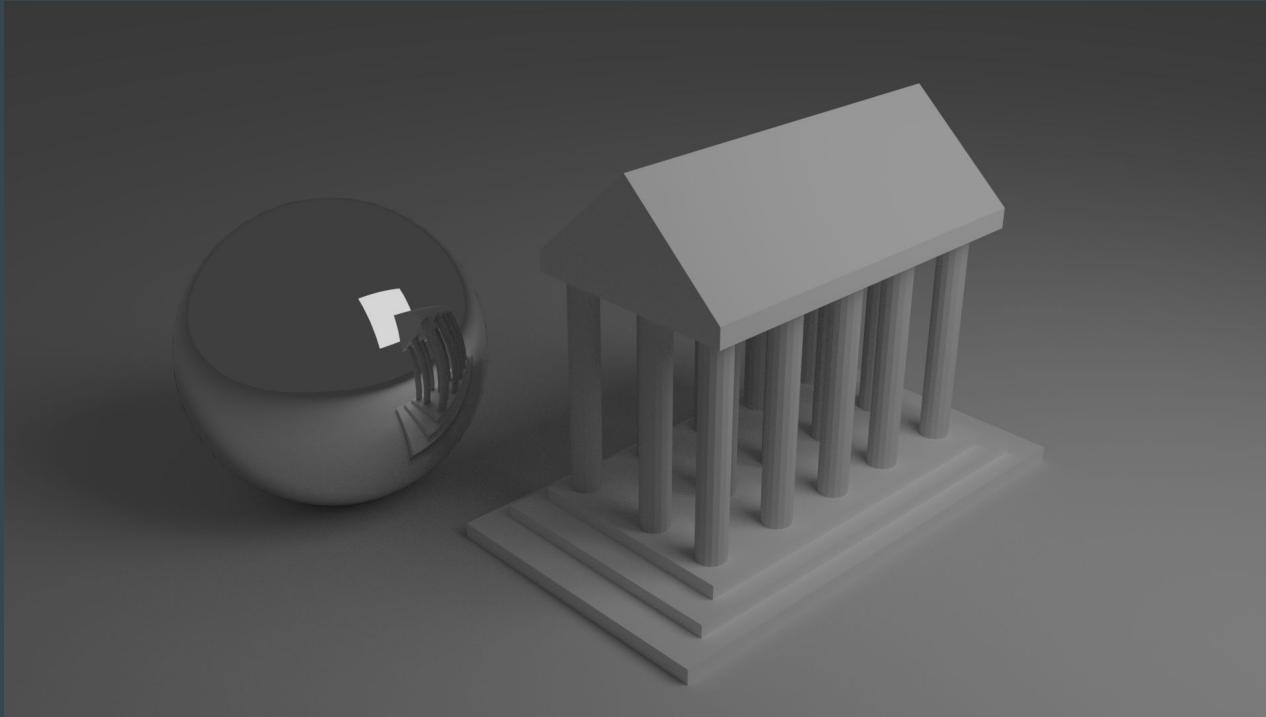
# Spot Light



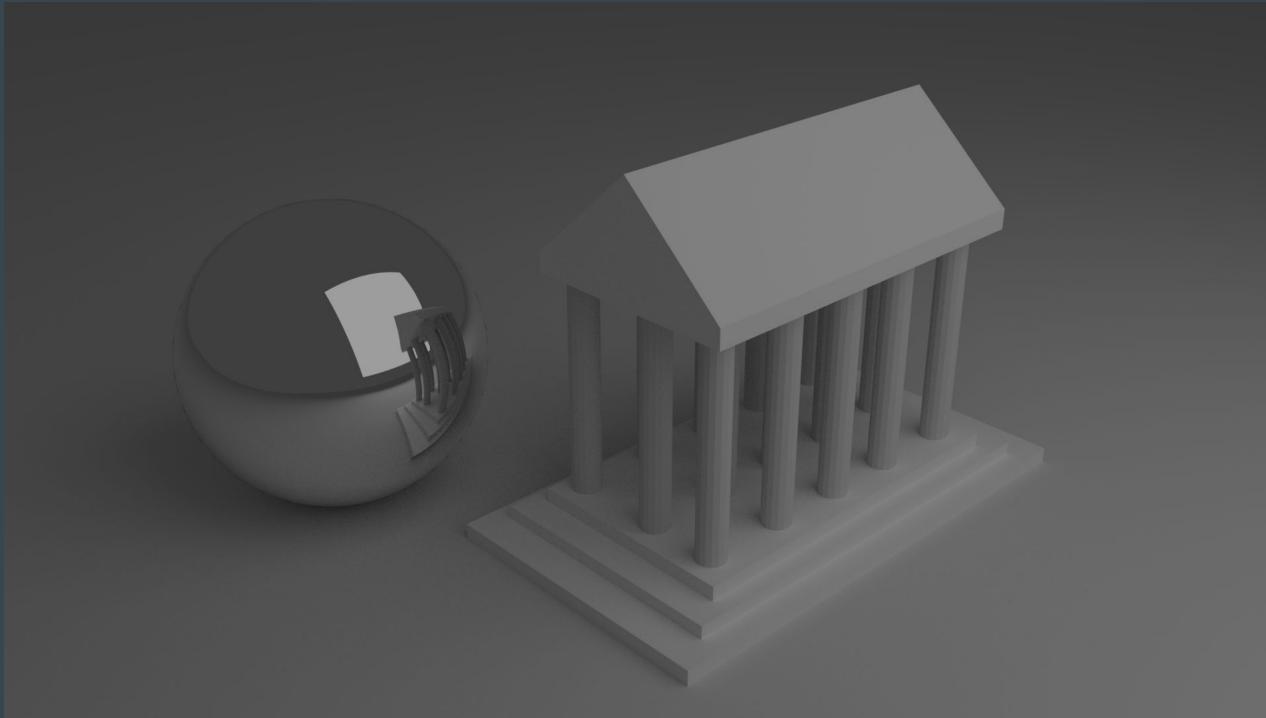
# Area Light - Small



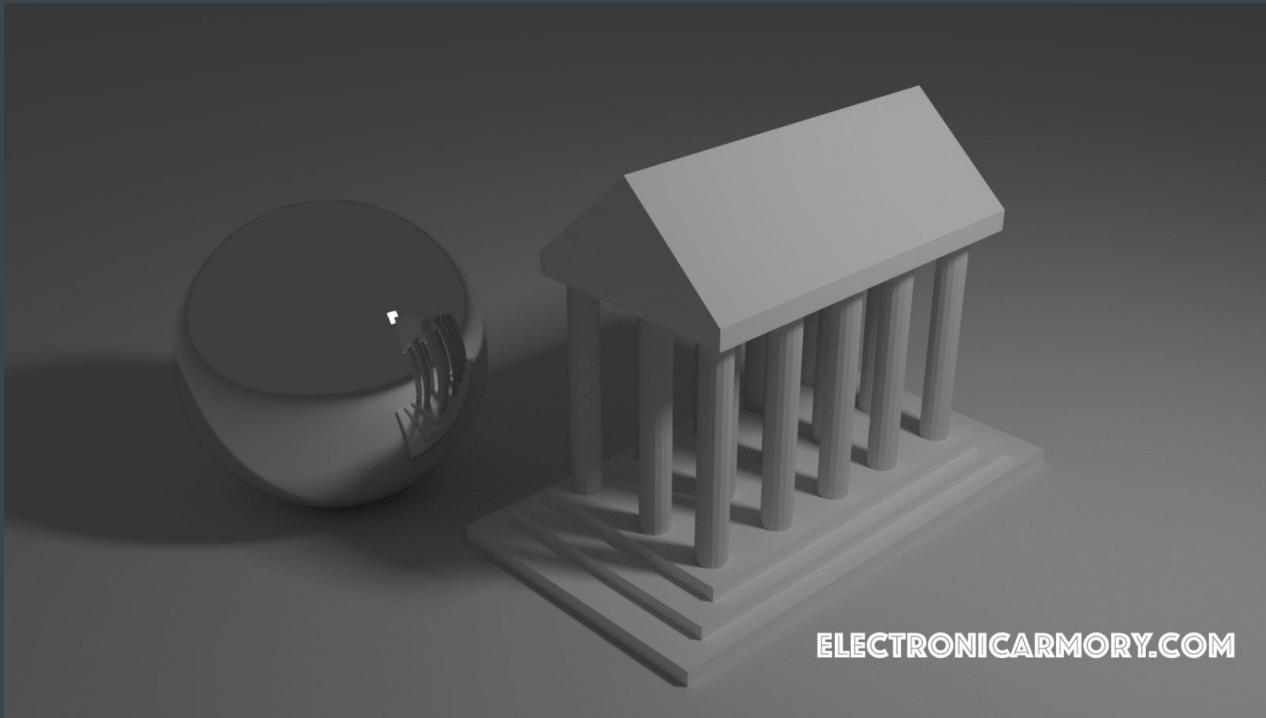
# Area Light - Medium



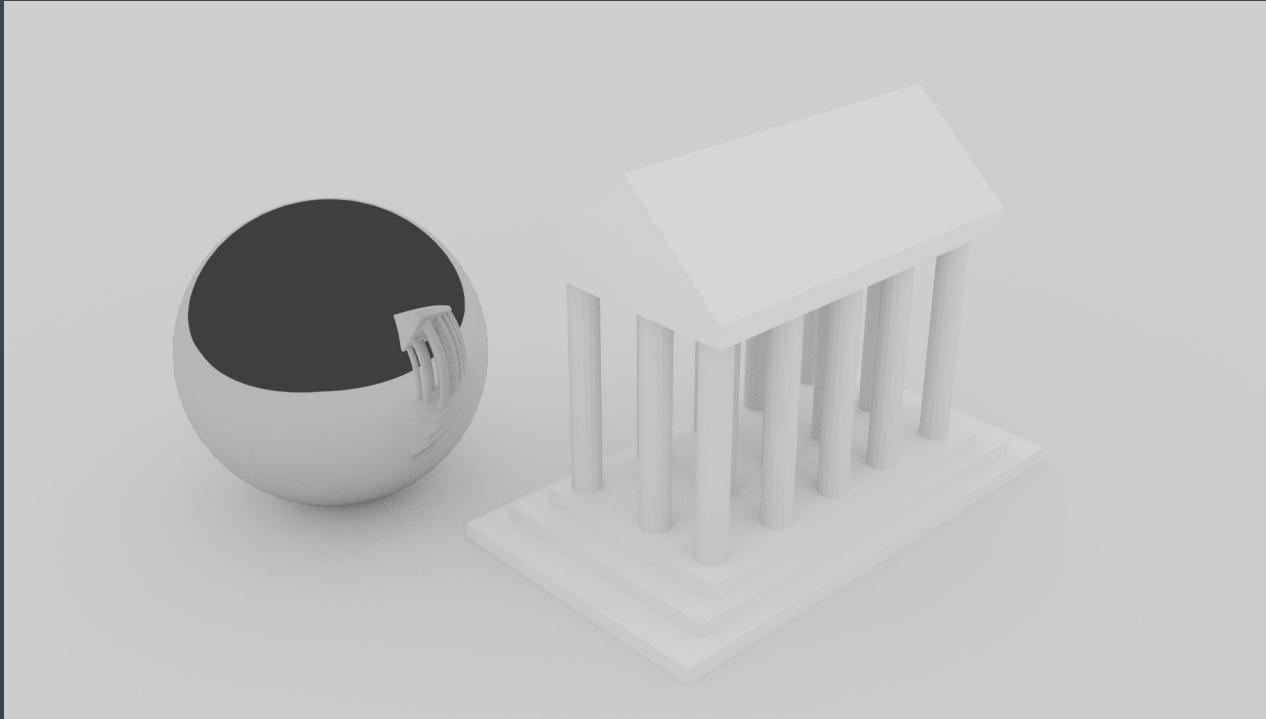
# Area Light - Large



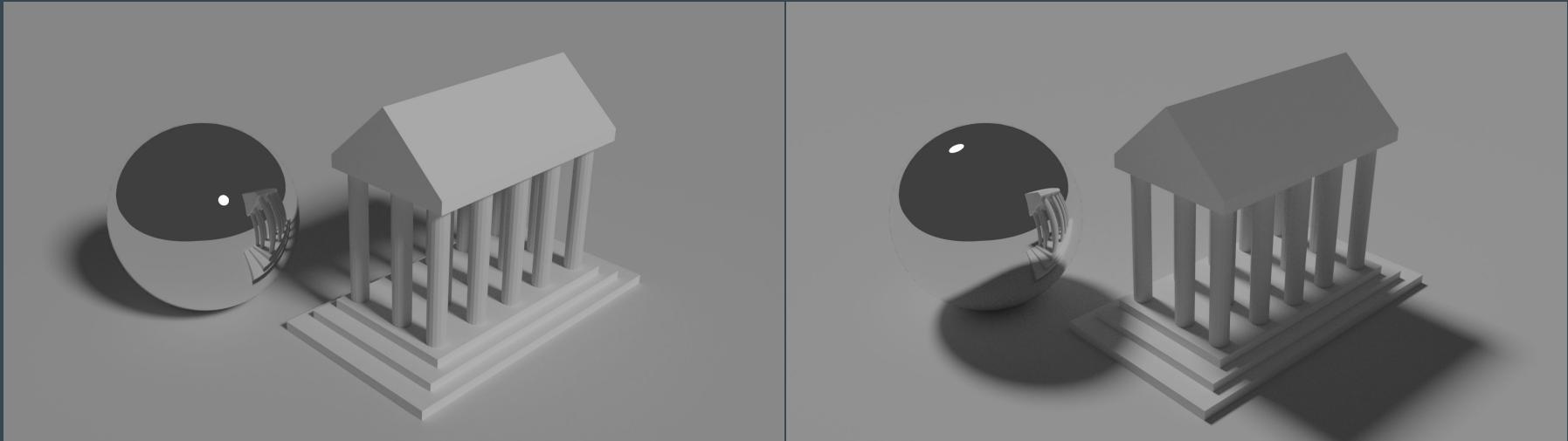
# Area Light - Animation!



# Global Illumination

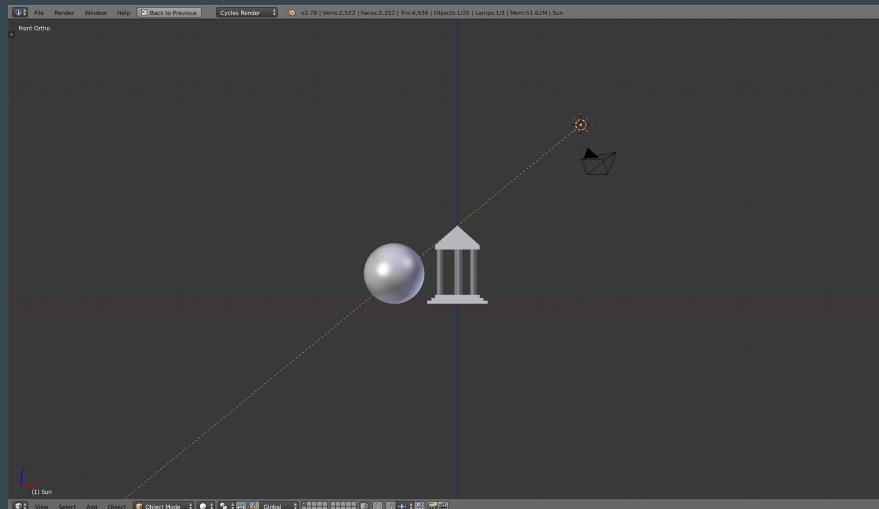


# Sun



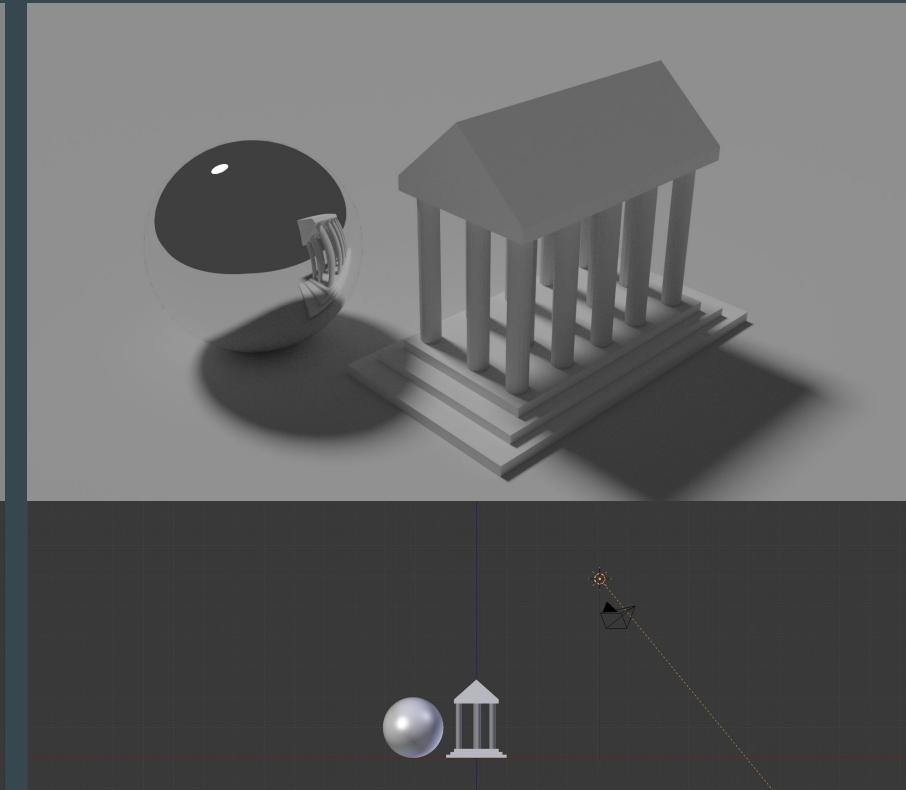
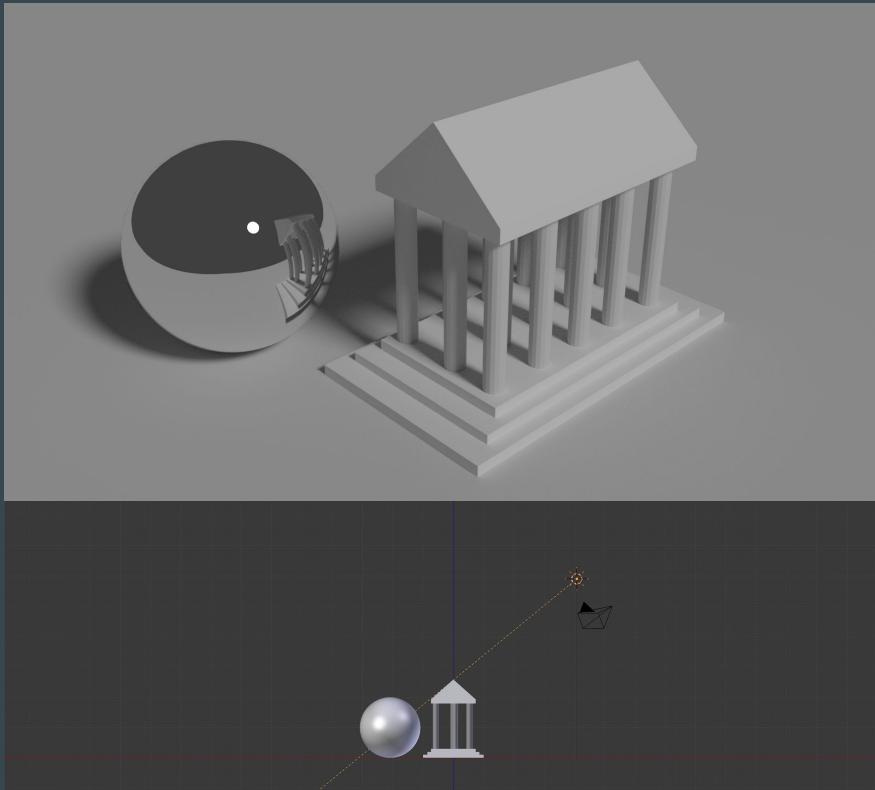
Sun lamps shine from all points in the same direction, independent of where the sun lamp is physically located. Image on the left is shining towards the left, while the image on the right is rotated 90s and shining to the right.

# Sun



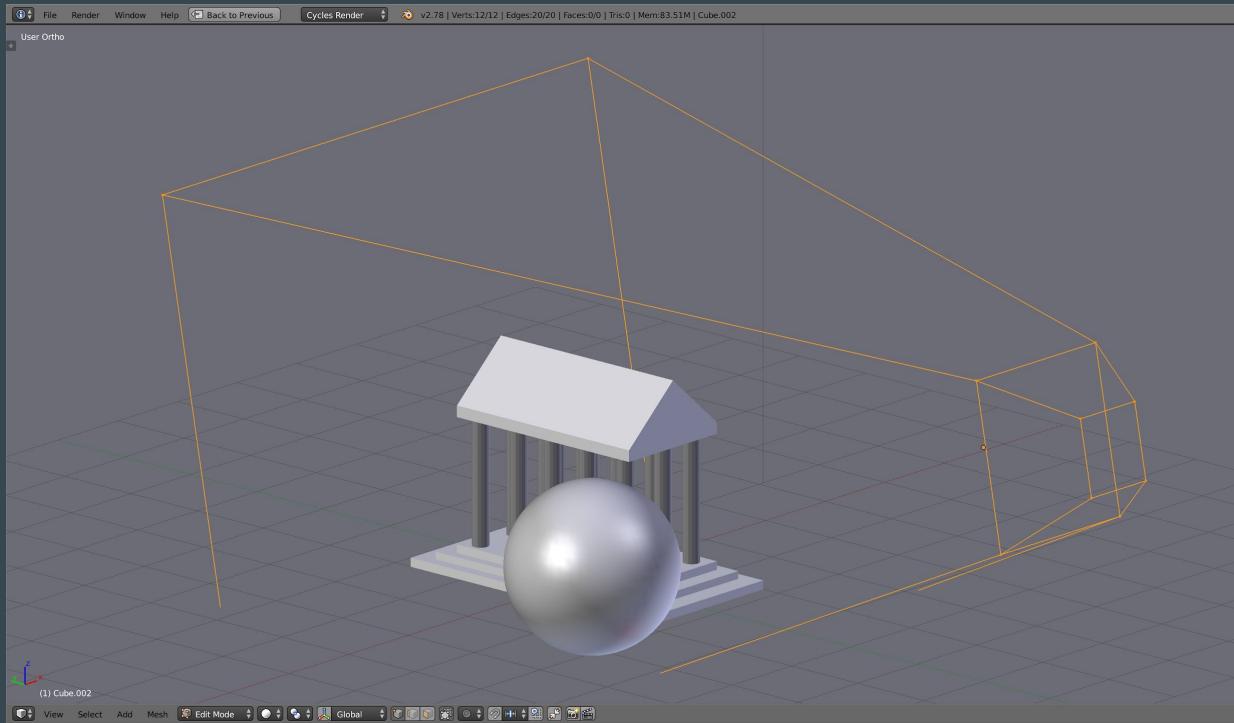
Sun lamps shine from all points in the same direction, independent of where the sun lamp is physically located. Image on the left is shining towards the left, while the image on the right is rotated 90s and shining to the right.

# Sun

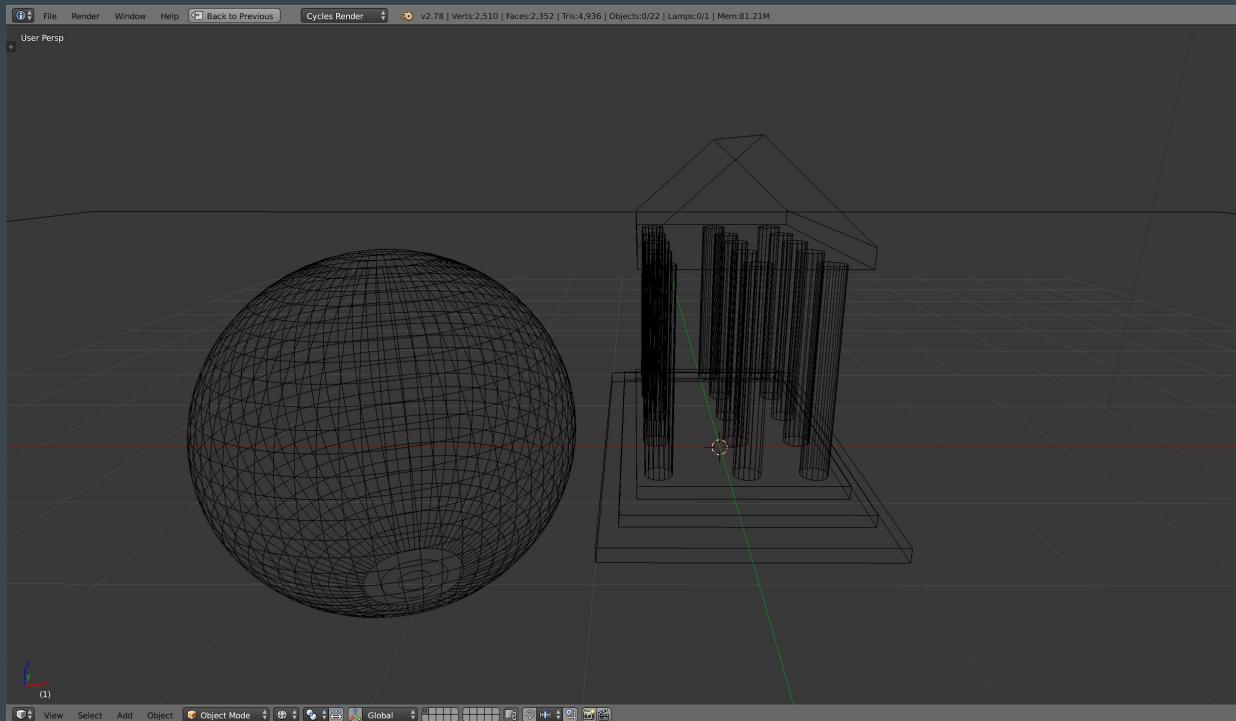


# Rendering

# Camera

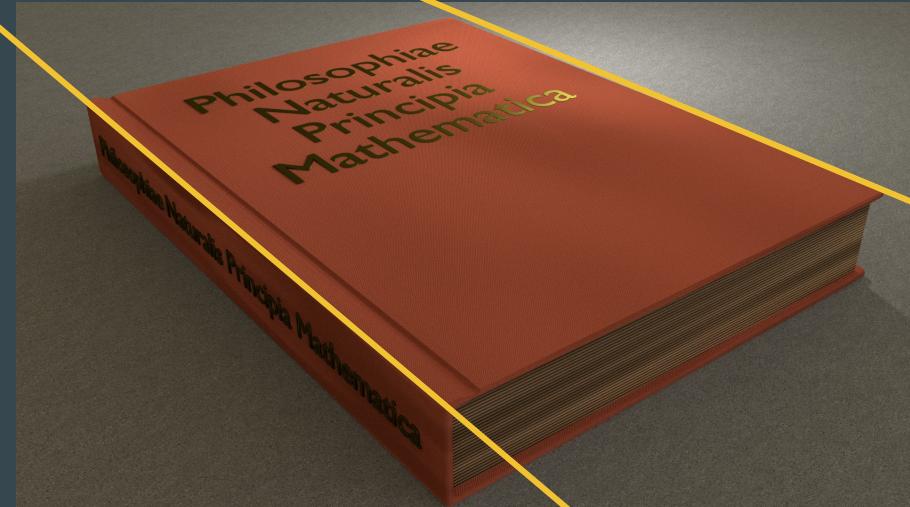


# Perspective

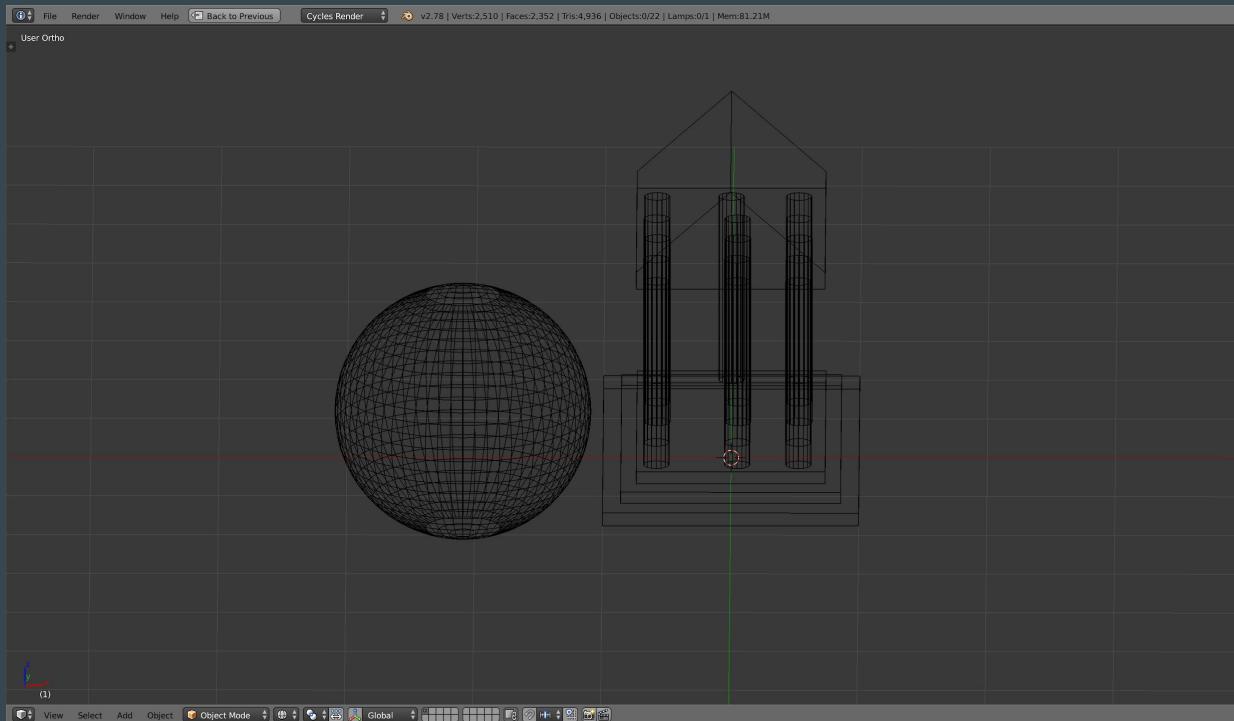


# Perspective - Example

Lines converge

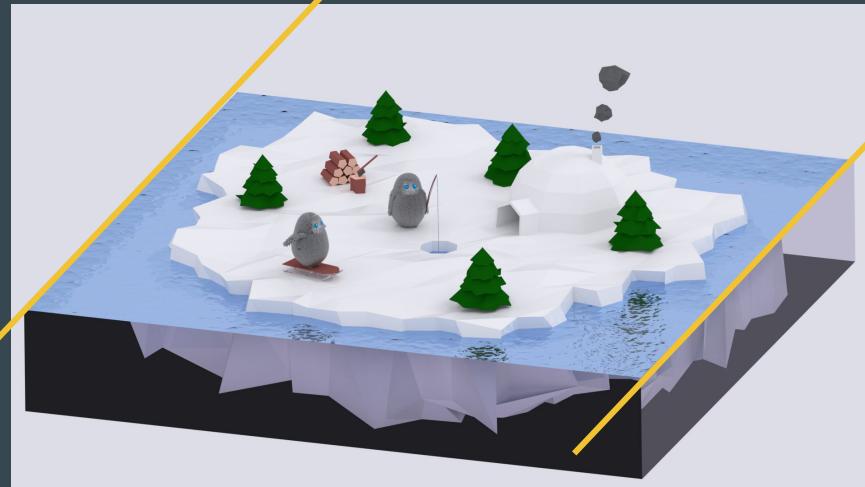


# Orthographic



# Orthographic - Example

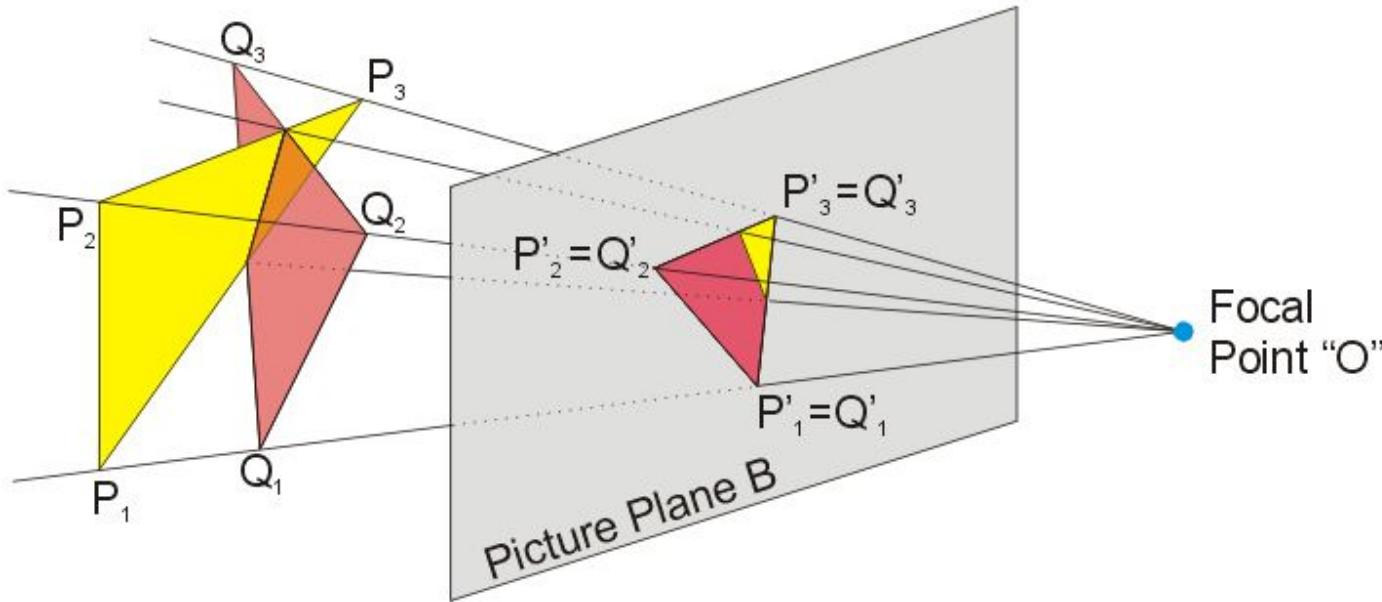
Lines never converge



# Graphics Pipeline

- Vertex processing
  - Processes every vertex independently to determine location and color
- Clipping and primitive assembly
  - Eliminates shapes that are not seen by the camera because they're hidden
- Rasterization
  - Primitives that make it through the clipper get translated to pixels
- Fragment processing
  - Combining the rasterized fragments into a final image

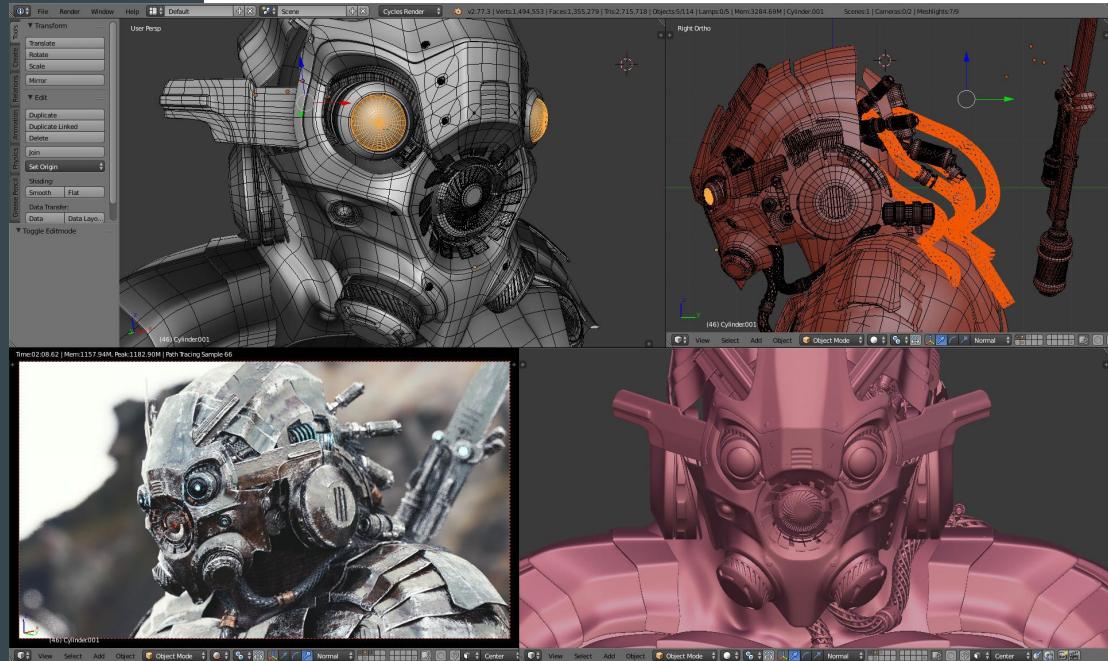
# Fragment Processing



# Object Culling



Source: Horizon Zero Dawn – The making of the game (2017) by vpro documentary <https://www.youtube.com/watch?v=A0eaGRcdwpo>



---

<https://www.blender.org>