DX1219
SHADER OPTIMISATION

SHADER LIGHTING II

Copyright © 2025/2026 Prepared by Lim Chian Song

INVERSE TRANSPOSE MATRIX IN DEPTH

- Non-uniform scaling = normal are scaled differently along different axes. (Skew)
- Inverse Transpose
  - Inverse - Remove the effect of any rotations on the normal
  - Transpose – Undo distortions caused by non-uniform scaling
- Let use a example with values.

$$n = \begin{bmatrix} 0.5 \\ 0.5 \\ 0 \end{bmatrix}, S = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Copyright © 2025/2026 Prepared by Lim Chian Song

INVERSE TRANSPOSE MATRIX IN DEPTH

If we directly apply the scaling matrix, this will happen (Incorrect)

$$n = \begin{bmatrix} 0.5 \\ 0.5 \\ 0 \end{bmatrix}, S = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$n' = S \cdot n = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0.5 \\ 0.5 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1.5 \\ 0 \\ 0 \end{bmatrix}$$

Copyright © 2025/2026 Prepared by Lim Chian Song

INVERSE TRANSPOSE MATRIX IN DEPTH

So if we used the Inverse Transpose Matrix instead to correctly transform the normal.

$$S^{-1} = \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 0.33 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$(S^{-1})^T = \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 0.33 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Copyright © 2025/2026 Prepared by Lim Chian Song

INVERSE TRANSPOSE MATRIX IN DEPTH

Using the Inverse Transpose Matrix to the normal vector. We will normalized it as well.

$$n' = (S^{-1})^T \cdot n = \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 0.33 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0.5 \\ 0.5 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.25 \\ 0.117 \\ 0 \\ 0 \end{bmatrix}$$

$$n'' = \begin{bmatrix} 0.83 \\ 0.55 \\ 0 \end{bmatrix}$$

Copyright © 2025/2026 Prepared by Lim Chian Song

INVERSE TRANSPOSE MATRIX IN DEPTH

$$x \cdot n = x_1 n_1 + x\ _2 n_2 + x_3 n_3$$

$$x \cdot n = 0$$

They are **perpendicular (90°)**

$$x \cdot n = x^T n$$

If we plug in $x^{'} = Ax$

$$x' \cdot n' = (Ax) \cdot n'$$

$$(Ax) \cdot n' = (Ax)Tn' = x^T A^T n'$$

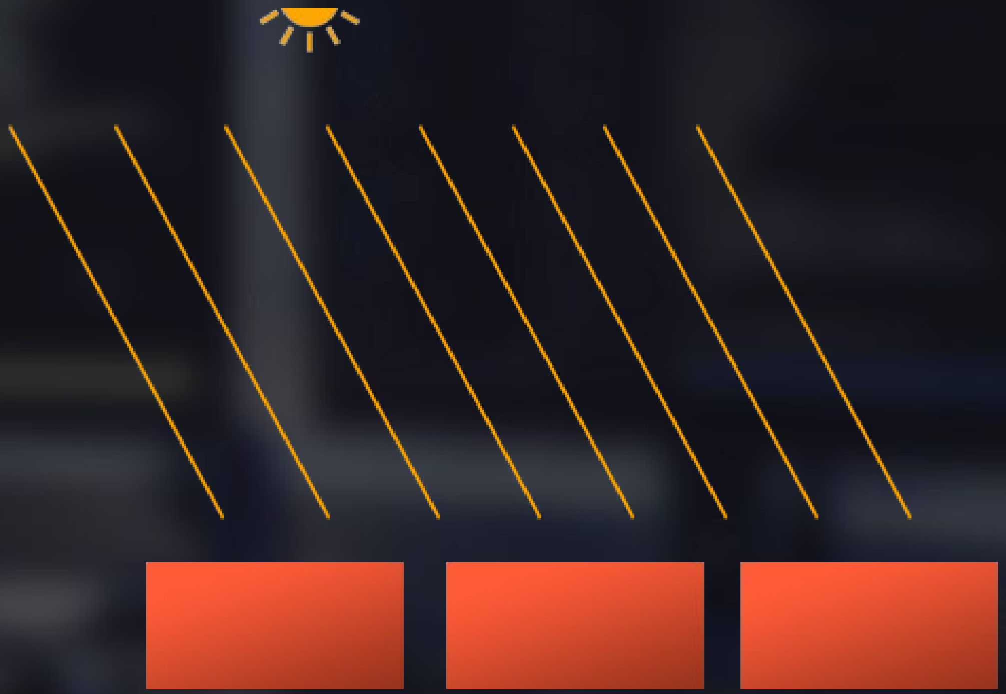
$$A^T n' \propto n$$

$$n' \propto A^{-T} n$$

Copyright © 2025/2026 Prepared by Lim Chian Song

Light Type (Directional)

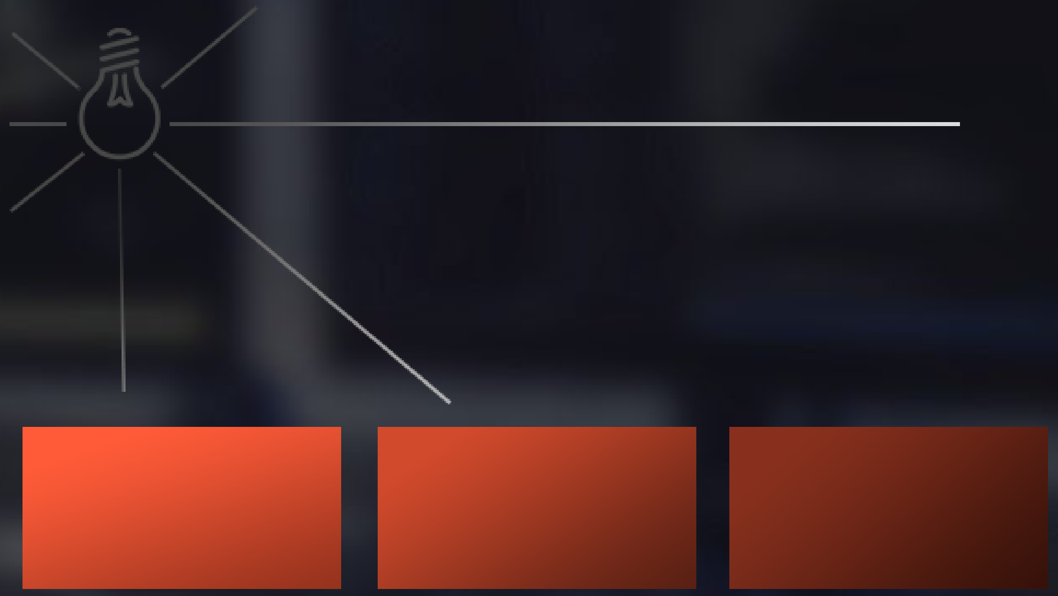
- Light rays come from 1 direction
- Light source infinitely far away
- Has direction but no position
- Shadows are parallel and consistent
- Eg: Sunlight, Moonlight, Outdoor scenes



Copyright © 2025/2026 Prepared by Lim Chian Song

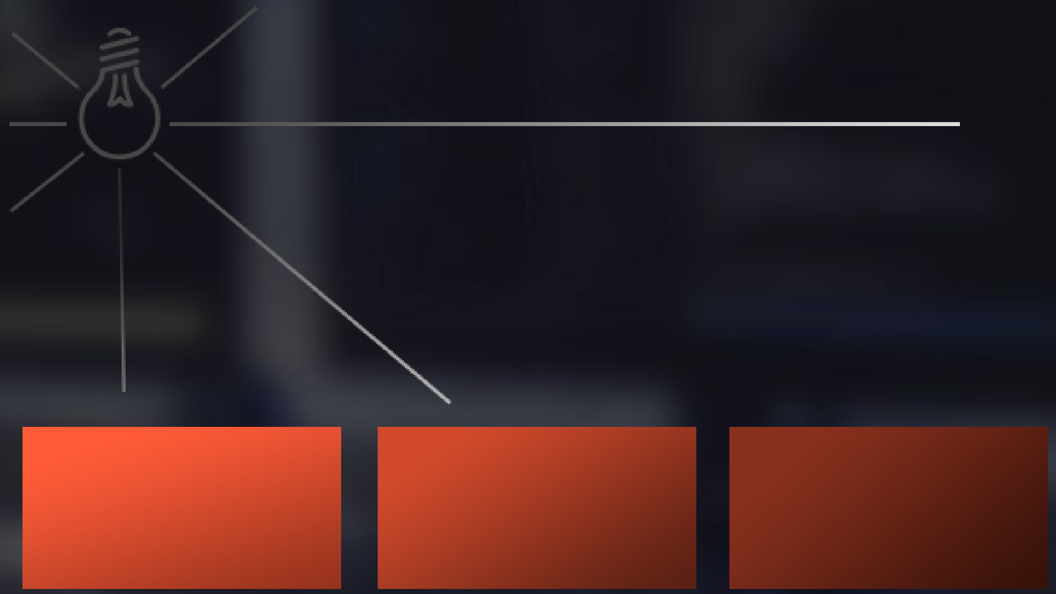
Light Type (Point)

- Omni Light
- Light emitted equally in all directions from a single point
- Has position and radius
- Has attenuation
- Eg: Lamps, Candles, Magic Orbs



Light Type (Spot)

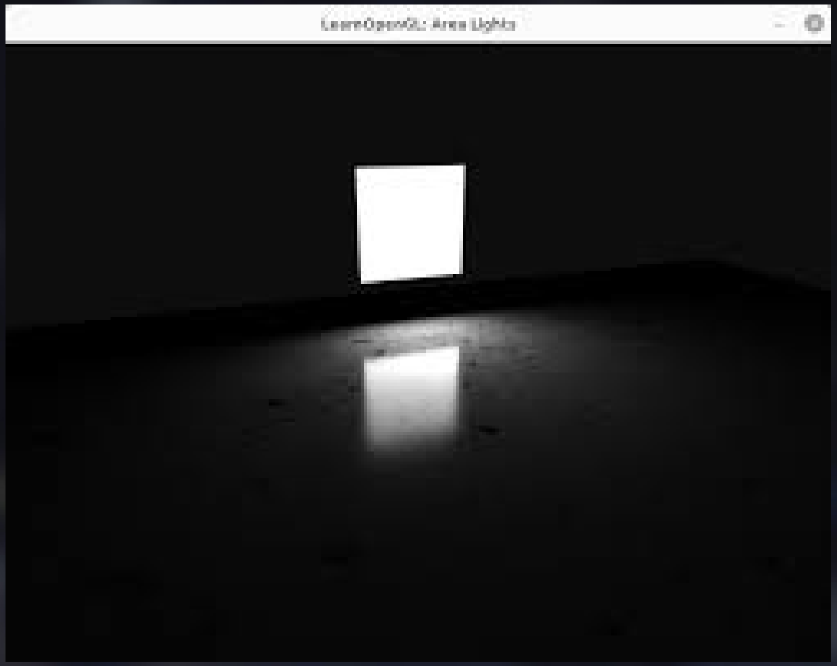
- Cone-shaped light
- Has
  - Position
  - Direction
  - Inner/Outer cone angle
  - Soft or Hard edge
  - Range
- Eg: Flash light, Car light



Copyright © 2025/2026 Prepared by Lim Chian Song

Light Type (Area)

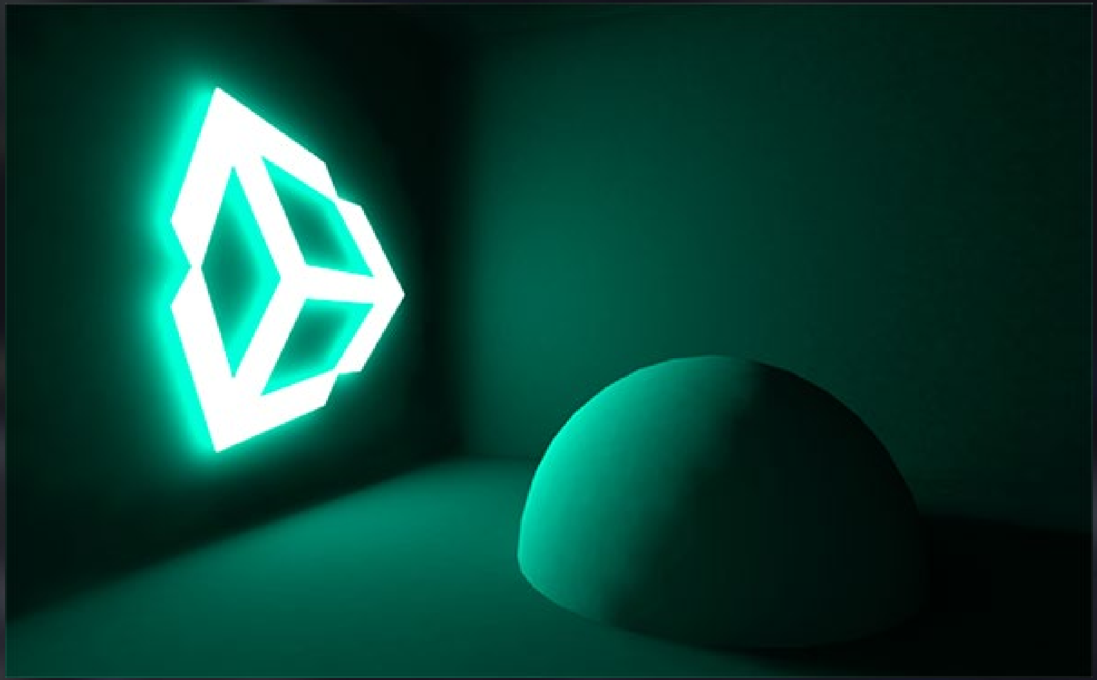
- Light comes from a surface, not a point
- Shapes like rectangle, disk, line
- Soft realistic shadows
- Use offline renderers for realism
- Use approximated in real time
- Eg: Fluorescent panel, window light



Copyright © 2025/2026 Prepared by Lim Chian Song

Light Type (Emissive)

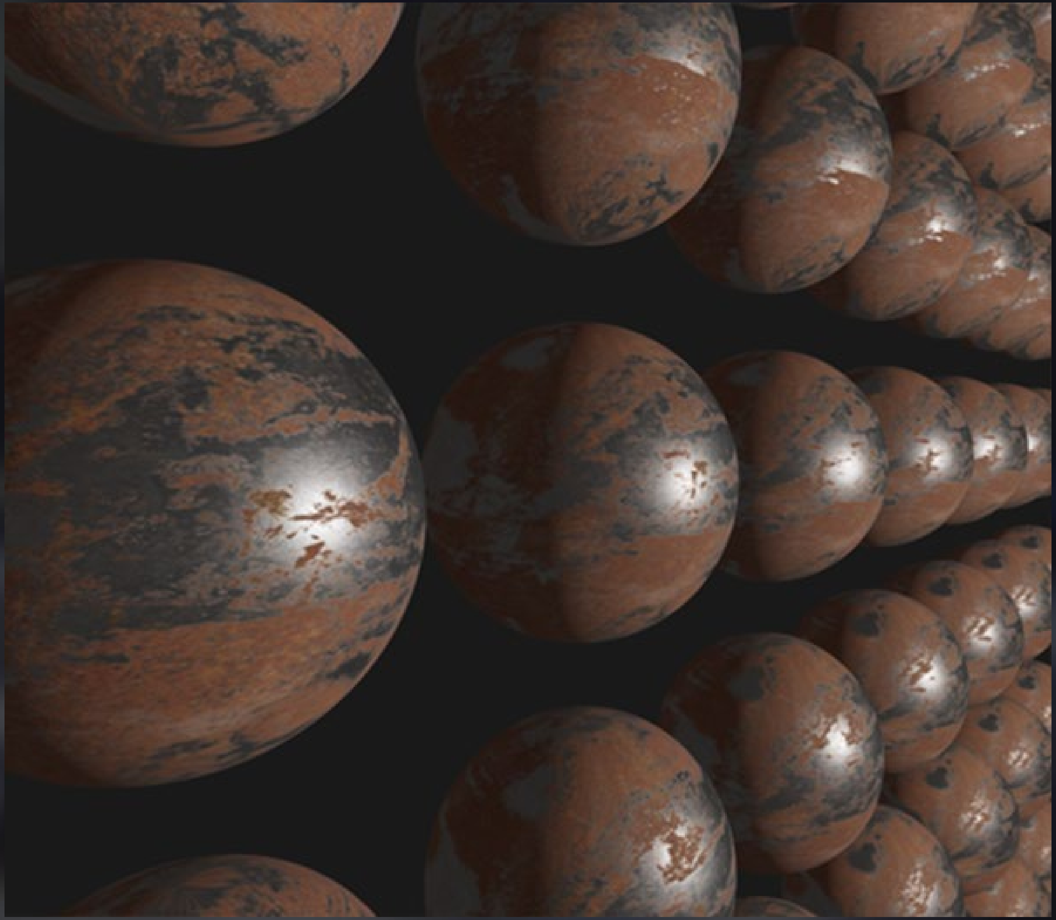
- Objects that glow and cast light themselves
- Emissive material
- Make the object itself bright
- Eg. Neon Sign, Glowing Logo



Copyright © 2025/2026 Prepared by Lim Chian Song

Physically Based Rendering

- Lighting based on physical model of light
- Usually uses albedo + metallic + roughness maps
- More realistic under different lighting condition

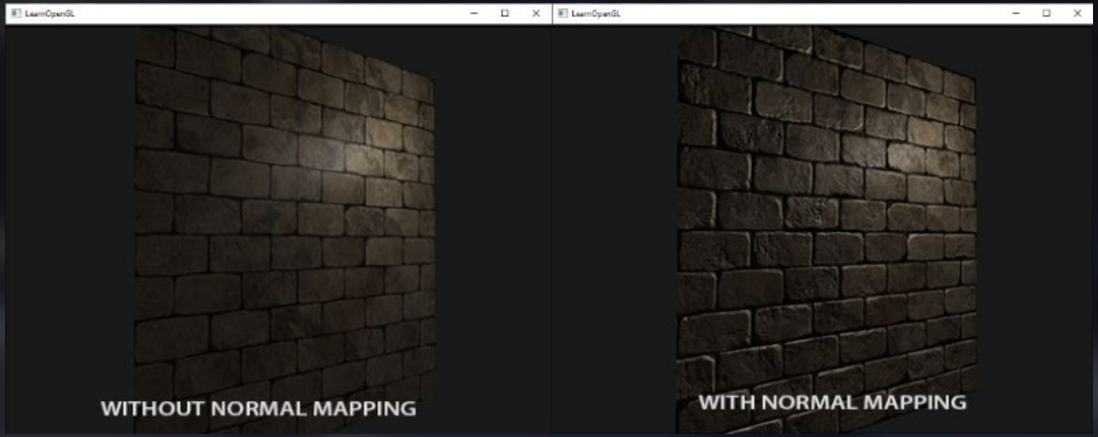


NANYANG THE INNOVATIVE POLYTECHNIC | SCHOOL OF DESIGN & MEDIA

Copyright © 2025/2026 Prepared by Lim Chian Song

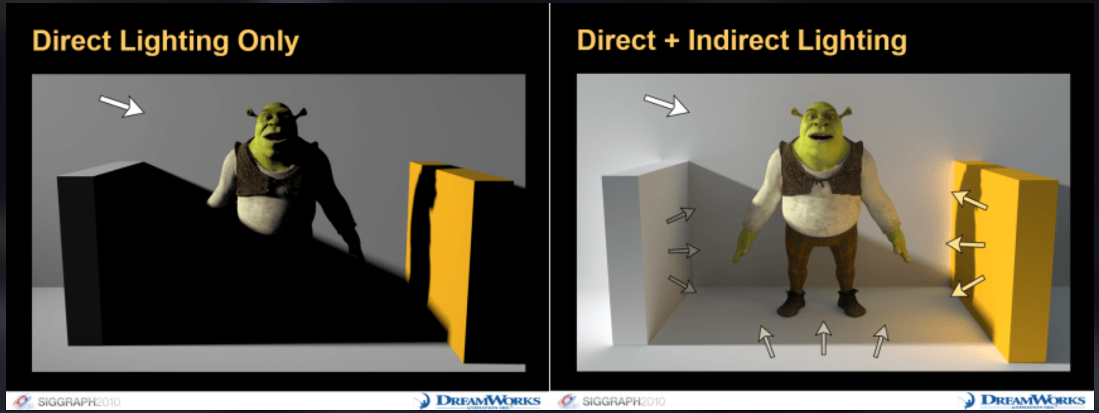
Normal Mapping

- Use a normal map (Height map) to fake small surface details without adding extra geometry
- High-detail look with low polycount
- Example: Scratches, Bricks



Global Illumination (GI)

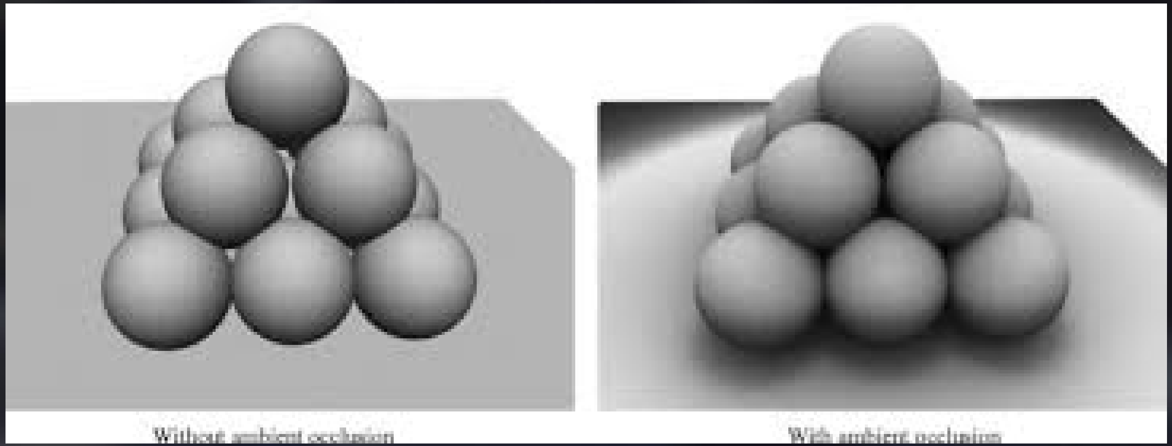
- Light bouncing off surfaces to other surfaces (not just direct light)
- Natural contrast
- Less flat game lighting
- Baked (Lightmaps, light probes)
- Real – Time (Ray tracing)



Copyright © 2025/2026 Prepared by Lim Chian Song

Ambient Occlusion (SSAO)

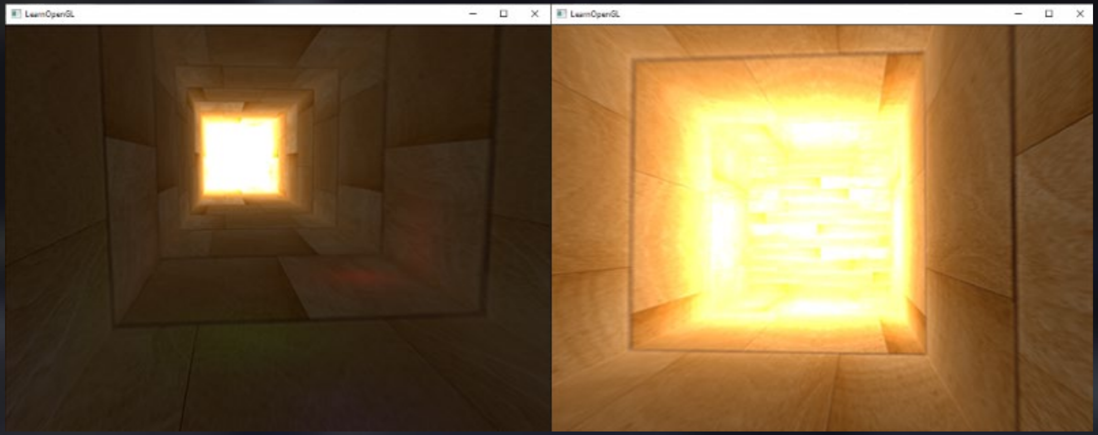
- Extra darkening in creases / corners (screen-space)
- Add depth and contact between objects
- Make details pop
- Visual improvement for relatively low cost



Copyright © 2025/2026 Prepared by Lim Chian Song

HDR Lighting + Tone Mapping + Bloom

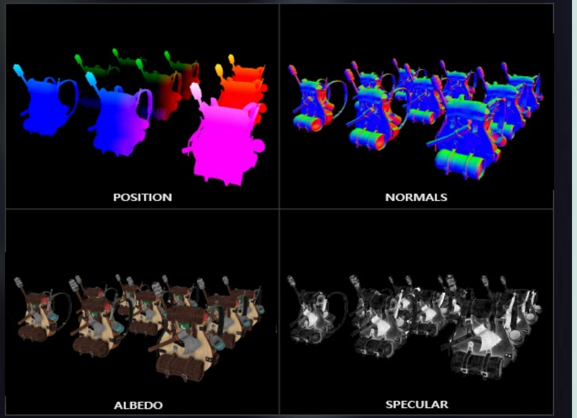
- **HDR**: Allow lighting value > 1.0 internally
- **Tone mapping**: Process of optimizing and compressing brightness and contrast levels in HDR content
- **Bloom:** glow around very bright areas.

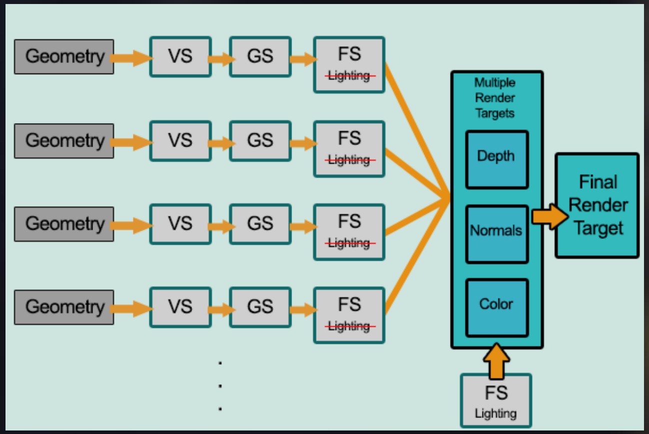


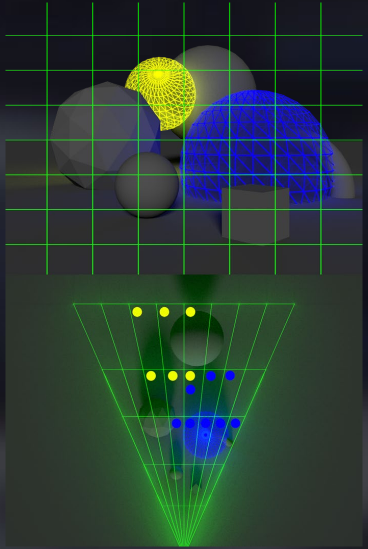
Copyright © 2025/2026 Prepared by Lim Chian Song

Deferred / Forward+ Rendering

- **Deferred: R**ender geometry data first (positions, normals, albedo), then do lighting in a full-screen pass.
- **Forward+ :** smarter forward rendering that partitions space into tiles/clusters to manage many lights.
- For handling lots of light



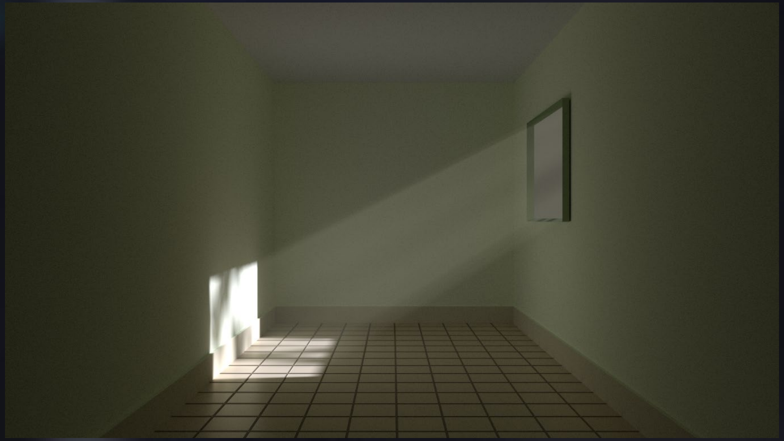


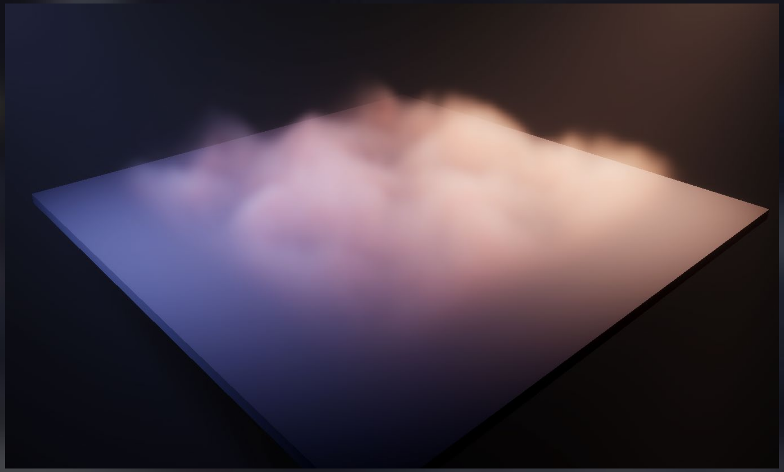


Copyright © 2025/2026 Prepared by Lim Chian Song

Volumetric Lighting / Fog

- Simulate light interacting with **fog, dust, smoke** in the air.
- Strong **mood / atmosphere**
- Helps convey depth and scale.





Copyright © 2025/2026 Prepared by Lim Chian Song

THE END

Any Question ?

Copyright © 2025/2026 Prepared by Lim Chian Song