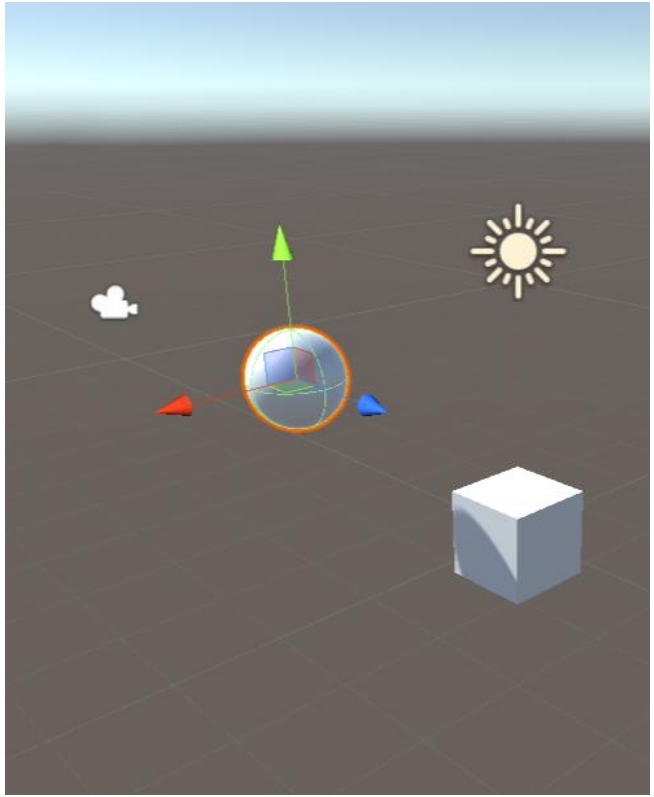
Licht

Lichtquellen,
Beobachter (Kamera),
Beleuchtung

Szene

- Anordnung von
  - Lichtquellen
  - Objekten
  - virtuellen Betrachtern (“Kamera”, Viewport)
  - Kulisse (Skybox)

- Im 3D Raum: Geometrische Modellierung
- 2D Bildsynthese: Rendering Pipeline

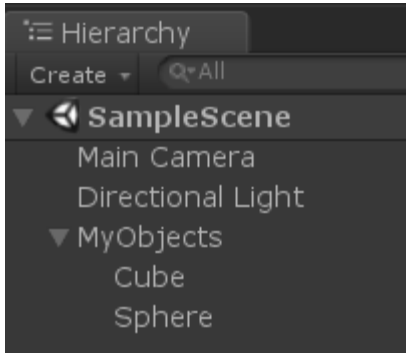


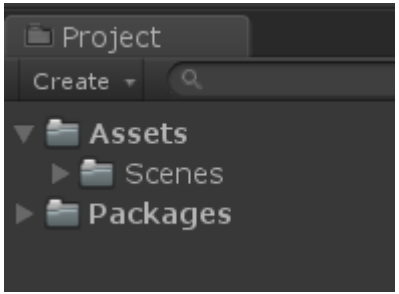
Szenengraph

- Hierarchische Anordnung der Elemente einer Szene (Struktur, Übersichtlichkeit, Pflegbarkeit)

- Child Elemente “übernehmen” Transformationen der Parent Elemente (Objektkoordinatensystem → Gruppenkoordinatensystem)

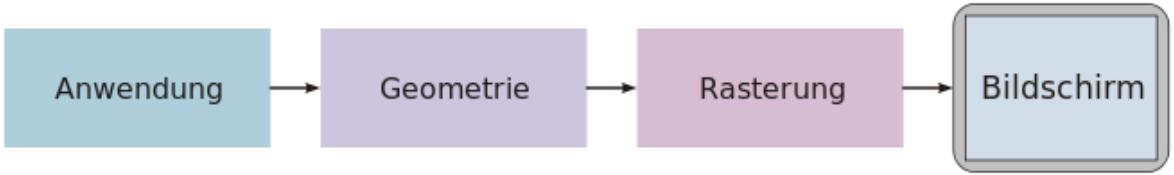
- In Unity:
  - Hierarchy
  - analog: Projektresourcen





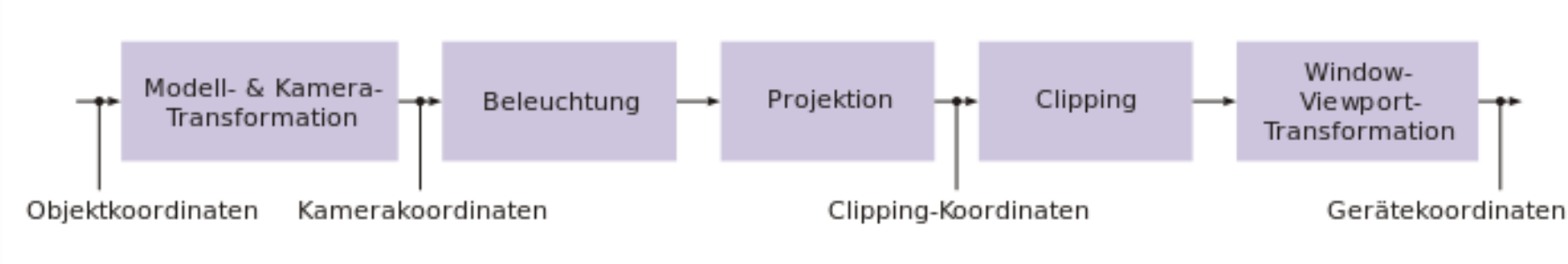
Graphikpipeline

- Anwendung
  - Content
  - Interaktion
- Geometrie
  - Modellbeschreibung
  - Transformation
  - Shader
- Rasterung
  - Post Processing
- Bildschirm
  - Anpassungen Ausgabegerät (Viewport etc.)



(Quelle: Wikipedia)

Geometrie



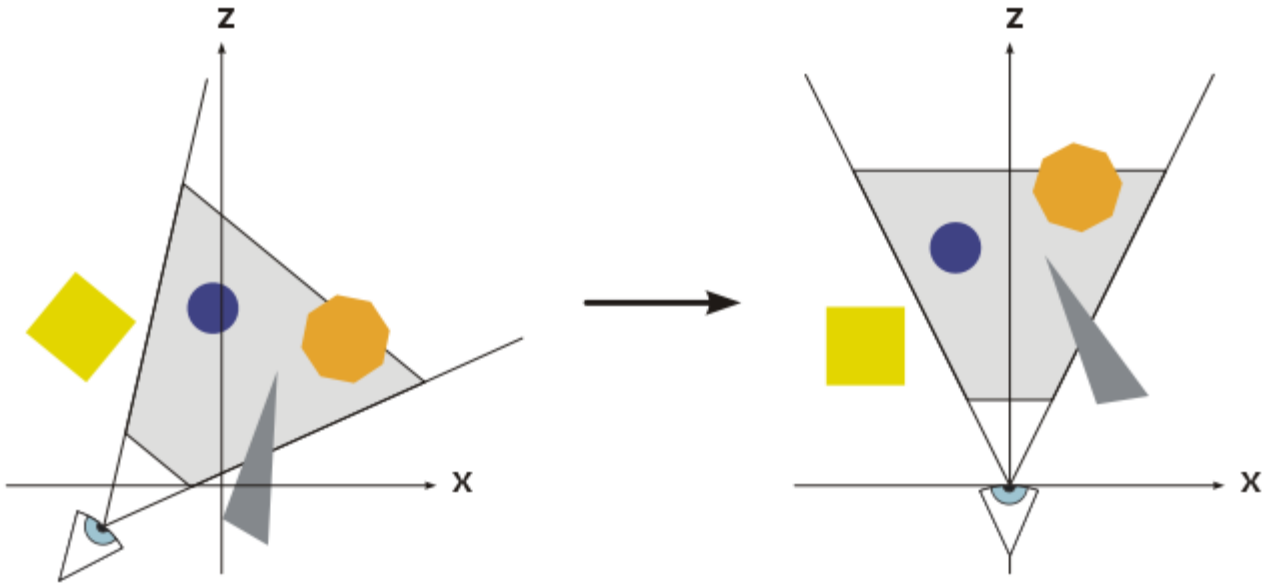
(Quelle: Wikipedia)

- Geometrie
  - Modellbeschreibung
  - Transformationen

Kamera

- Kamera Transformation
  - Transformation von Weltkoordinaten nach Kamerakoordinaten
- Perspektive
  - In Unity: Eigenschaft der “Kamera”
  - In Unity: “Perspective”, “Orthographic”
- Clipping
  - Beschränkung der Objekte auf Kamera-Blickwinkel
  - Beschränkung der Objekte auf z-Tiefe (Unity: “near clipping plane”, “far clipping plane”)
- Viewport Transformation
  - 16:9 etc.

Kamera Transformation



(Quelle: Wikipedia)

Wdh: Homogene Koordinaten

2D $\begin{pmatrix} x \\ y \\ z \end{pmatrix}$ $z \neq 0$, beliebig

$z = 1$: x, y = kartesische Koordinaten

Translation

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & & \\ & 1 & \\ t_x & t_y & 1 \end{pmatrix} = \begin{pmatrix} x + t_x \\ y + t_y \\ 1 \end{pmatrix}$$

Wdh: Homogene Koordinaten

Skalierung

$$\begin{pmatrix} x \\ y \\ \underline{1} \end{pmatrix} \cdot \begin{pmatrix} s_x & & \\ & s_y & \\ & & \underline{1} \end{pmatrix} = \begin{pmatrix} s_x \cdot x \\ s_y \cdot y \\ \underline{1} \end{pmatrix}$$

Rotation

$$\begin{pmatrix} x \\ y \\ \underline{1} \end{pmatrix} \cdot \begin{pmatrix} \cos\varphi & \sin\varphi & \\ -\sin\varphi & \cos\varphi & \\ & & \underline{1} \end{pmatrix} = \begin{pmatrix} x\cos\varphi - y\sin\varphi \\ x\sin\varphi + y\cos\varphi \\ \underline{1} \end{pmatrix}$$

Kamera Transformation

① Kamera: $\vec{cam}$ ("Augpunkt")

Weltkoordinaten

Kamera Koordinaten system

Sichtstrahl

$$\{\vec{e}_{hori,cam};\ \vec{e}_{vert,cam};\ \vec{e}_{hori,cam} \times \vec{e}_{vert,cam}\}$$

② Objektschwerpunkt (Pivot-Punkt)

$\vec{osj}$

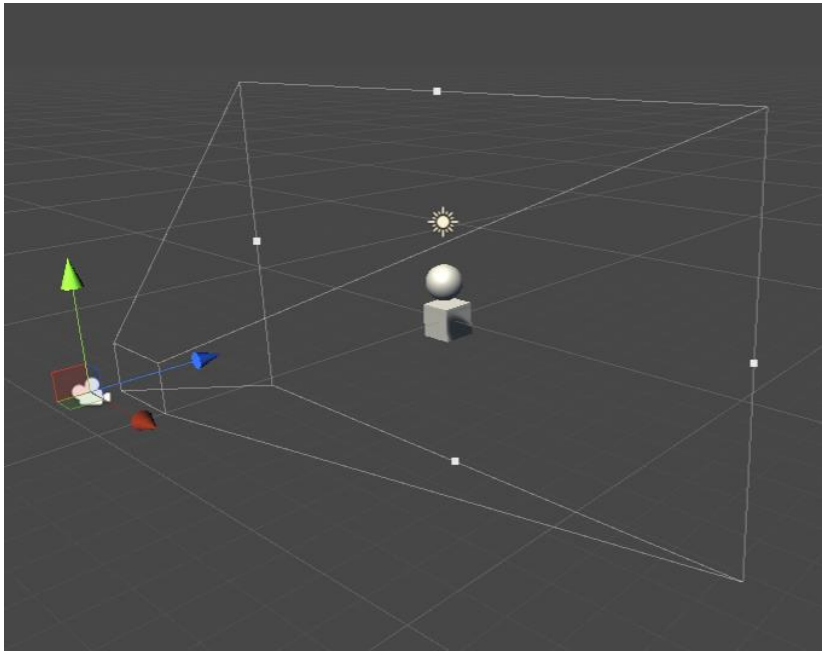
Kamera Transformation

③ Transformation

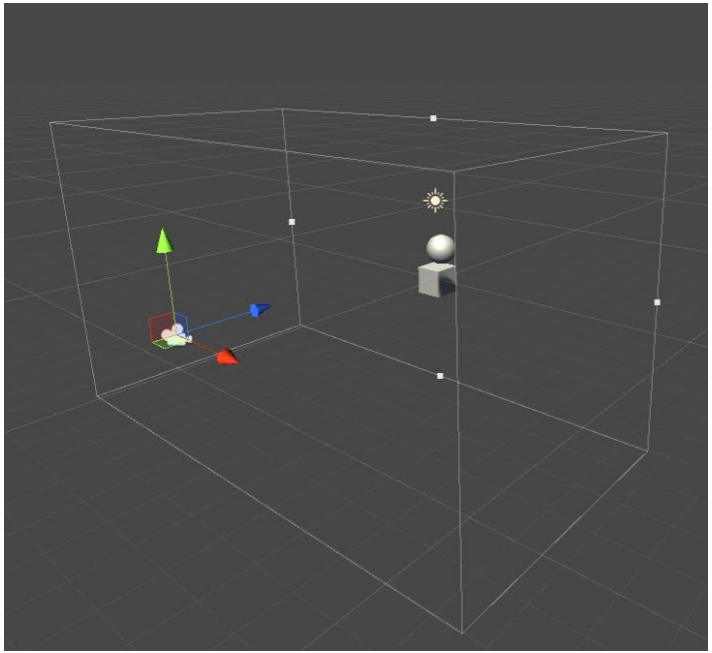
$$
\left.
\begin{aligned}
\vec{e}_z &= \frac{\vec{cam} - \vec{oS_j}}{|\vec{cam} - \vec{oS_j}|} \\
\vec{e}_x &= \frac{\vec{e}_{vert,cam} \times \vec{e}_z}{|\vec{e}_{vert,cam} \times \vec{e}_z|} \\
\vec{e}_y &= \vec{e}_z \times \vec{e}_x
\end{aligned}
\right\}
=
\begin{pmatrix}
\vec{e}_x & \vec{e}_y & \vec{e}_z & 0 \\
0 & 0 & 0 & 0 \\
-(\vec{e}_x \cdot \vec{cam}) & -(\vec{e}_y \cdot \vec{cam}) & -(\vec{e}_z \cdot \vec{cam}) & 1
\end{pmatrix}
$$

Perspektive in Unity

“Perspective”



“Orthographic”



Perspektive (mathematisch)

Verkleinerung,
abhängig von z Abstand
„x,y rücken ein“

Homogene Transformation nach 2D Bildkoordinaten
Augpunkt (0, 0, d)

Zentralprojektion:

$$\mathbf{P}_{\text{zp}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{d} & 1 \end{pmatrix}, \quad \mathbf{P}_{\text{zp}}\ (x, y, z, 1)^T = (x, y, 0, \tfrac{d-z}{d})^T$$

Orthogonale Projektion:

$$\mathbf{P}_{\text{op}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{P}_{\text{op}}\ (x, y, z, 1)^T = (x, y, 0, 1)^T$$

(Quelle: Wikipedia)

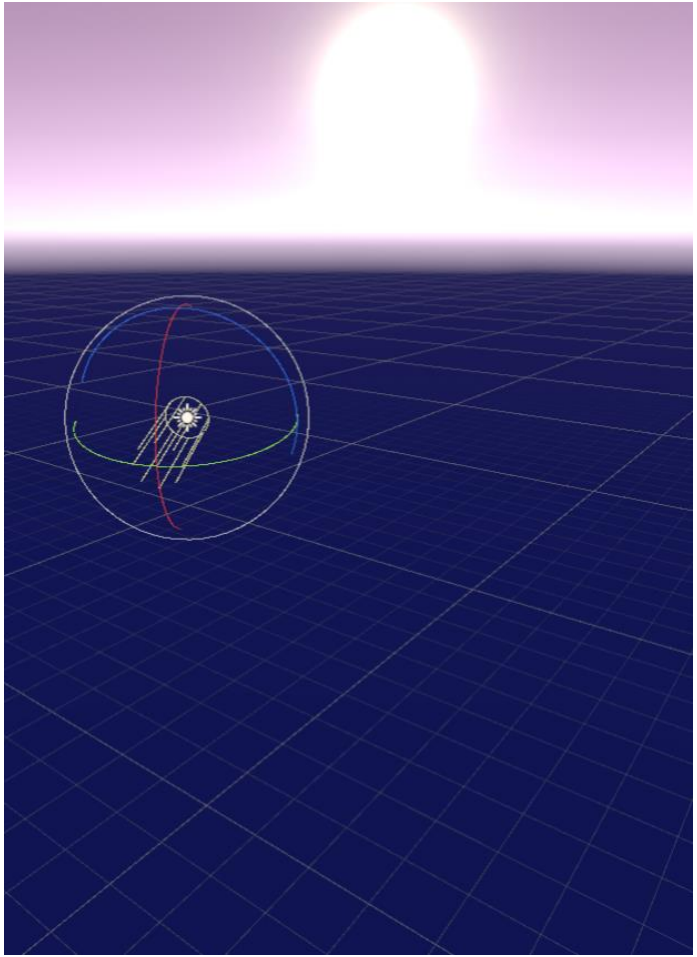
Beleuchtung

Zweck

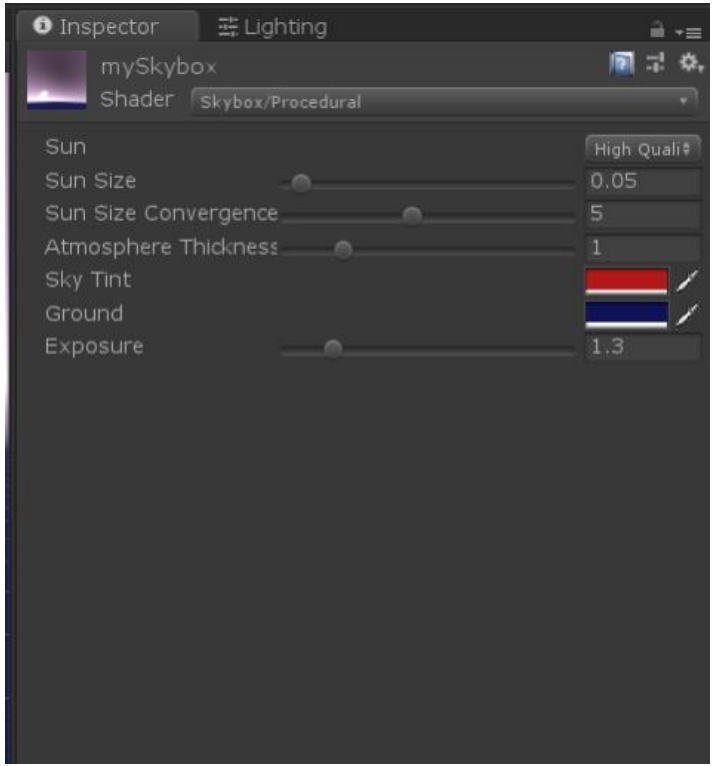
- Sichtbarkeit
- Glaubwürdigkeit der Szene, Stimmung
- Steuerung von Aufmerksamkeit

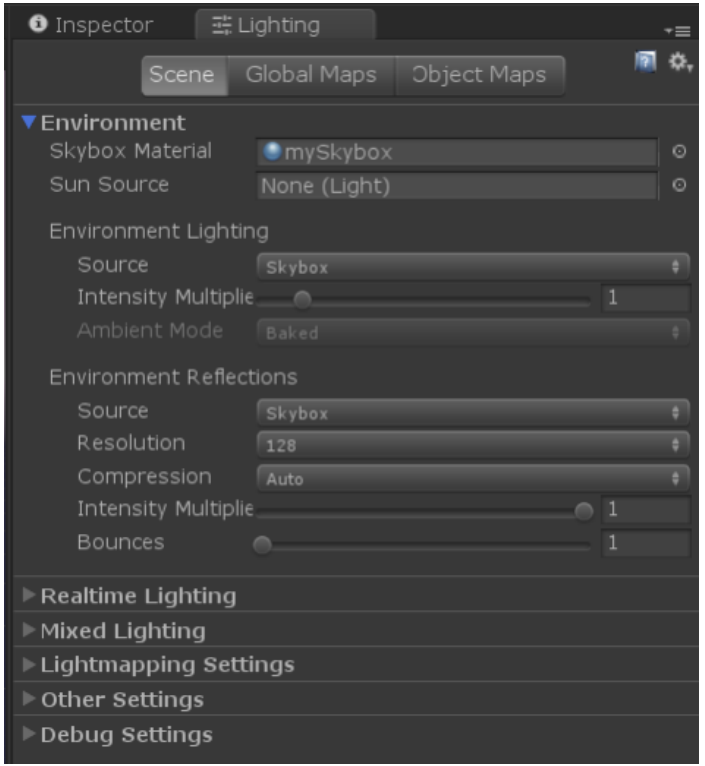
Arten von Lichtquellen

- Directional Light
- Pointlight
- Spotlight
- Area Light (“baked only”)



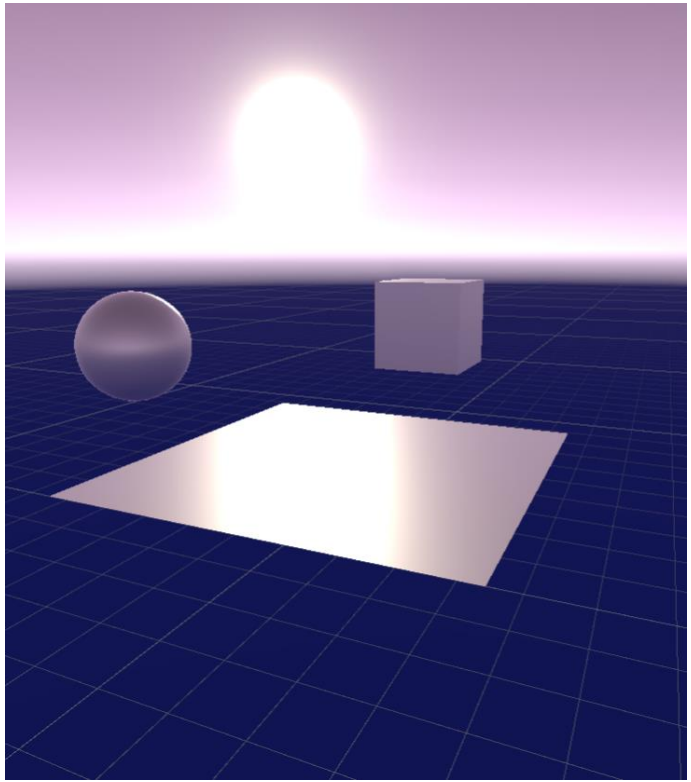
Übung: Directional Light + Skybox

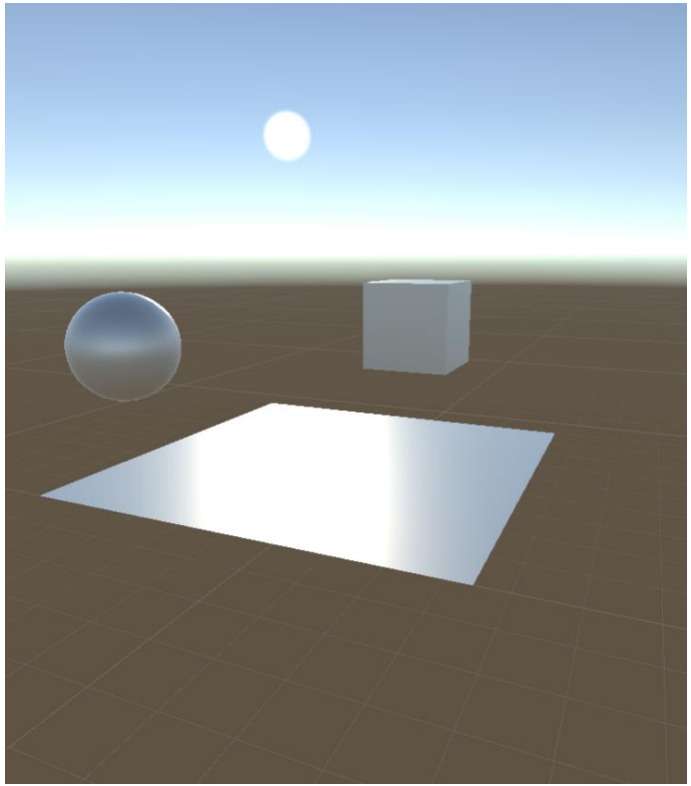




Ambient Light

Stimmung





Unity: Skybox als Lichtquelle

- Lighting / Environment Lighting
- Lighting / Environment Reflexions

- Beispiel: Unity HDRI Cubemap als Lichtquelle

- Photorealistische Beleuchtung

Space Robot Kyle

default skybox
cubemap skybox
directional light

procedural skybox

Cornell Box

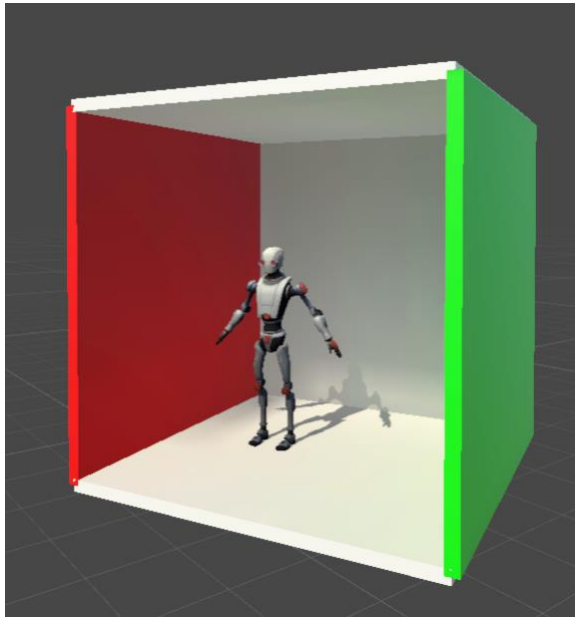
- Directional Light
  - Unendlich entfernt, Parallele Lichtstrahlen
  - In Unity: Positionierung egal, Parameter = Rotation
  - Culling Mask: Möglichkeit der Einschränkung, welche Objekte ausgeleuchtet werden

- Shadow Type
  - Soft / Hard Shadows: Übergang an Kanten
  - Bias / Normal Bias:
    Entfernung: Schattenwurf zu Objekt
    Erodieren der Schatten um Lücken zu vermeiden

Normal Bias

Übung

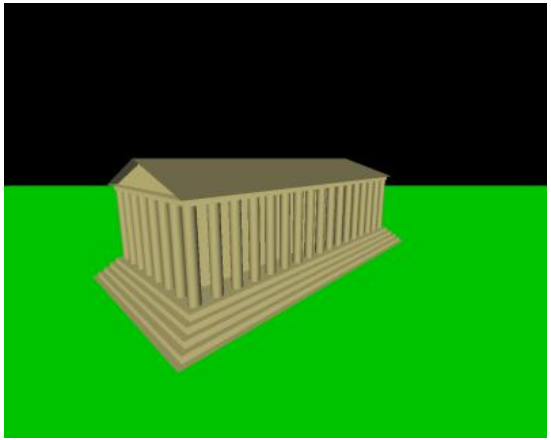
- Cornell Box erstellen
- Lichtquellen und Objekte hinzufügen
- Experimentieren



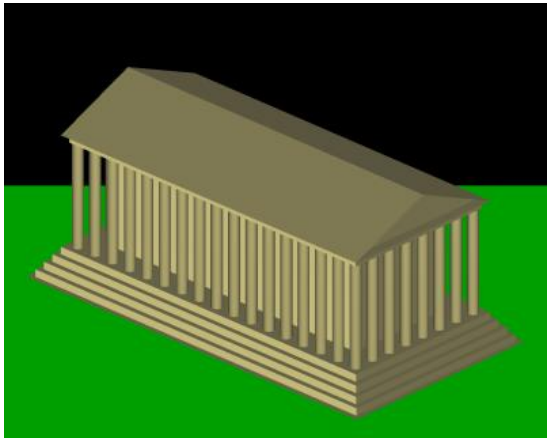
Shadow Mapping

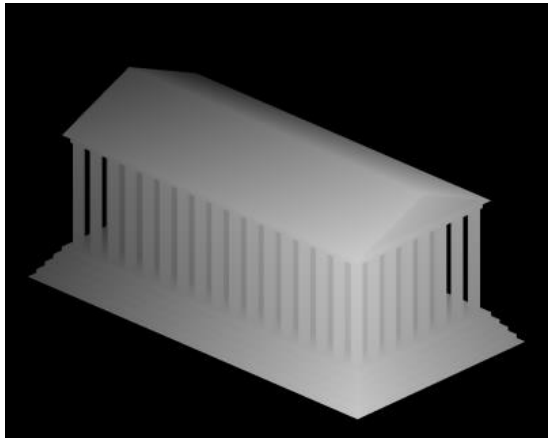
Kamera-Ansicht

Tiefeninformation



Ansicht: Lichtquelle

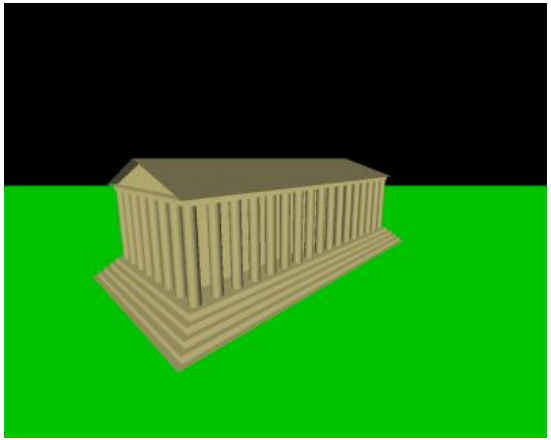




(Quelle: Wikipedia)

Shadow Mapping

Kamera-Ansicht



Projektion
Tiefeninformation

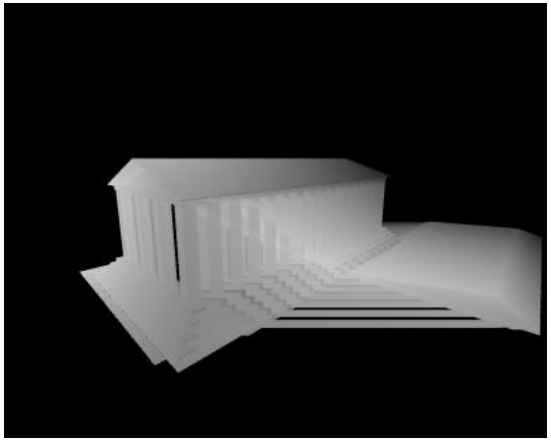
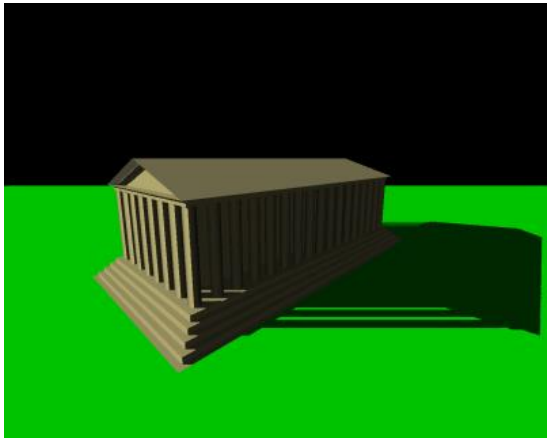


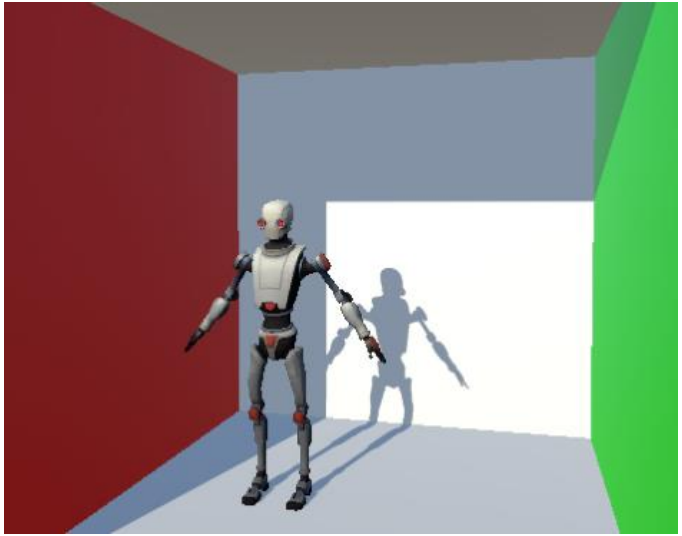
Bild mit Schatten
(Shader Pass)

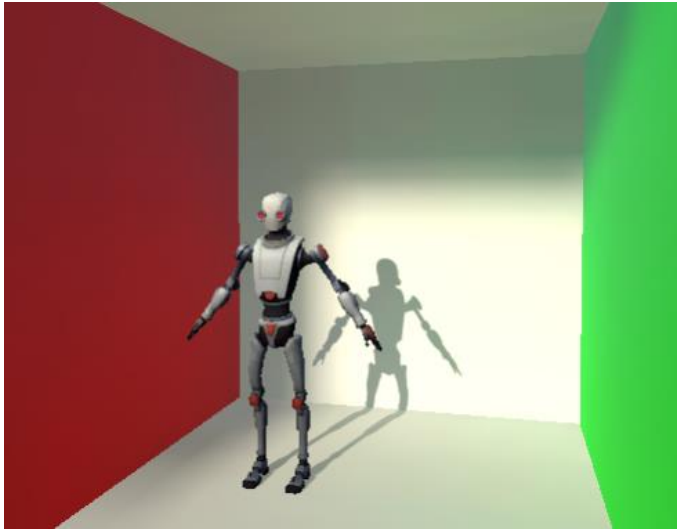


(Quelle: Wikipedia)

Global Illumination

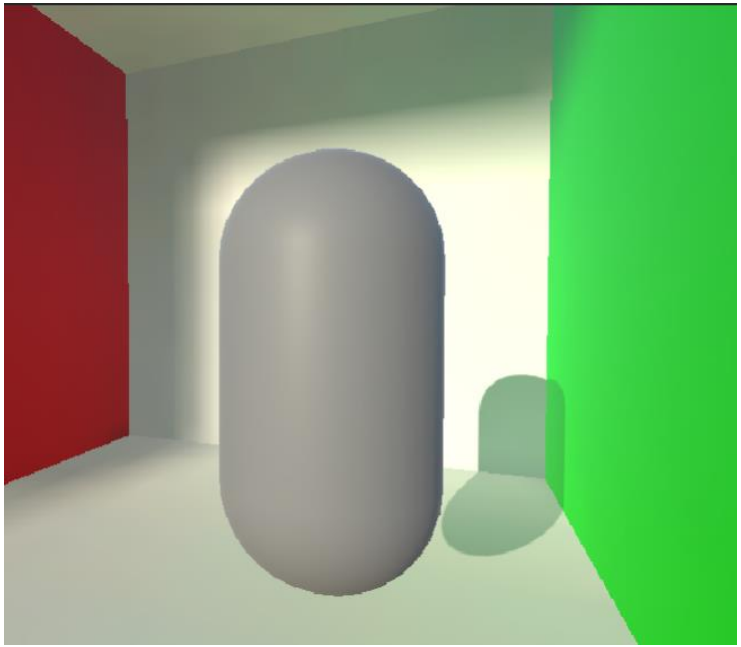
- Direct Lighting
  - von Lichtquellen
- Indirect Lighting
  - Reflektionen von Objekten
  - nur möglich für statische Objekte (vorab berechnet)

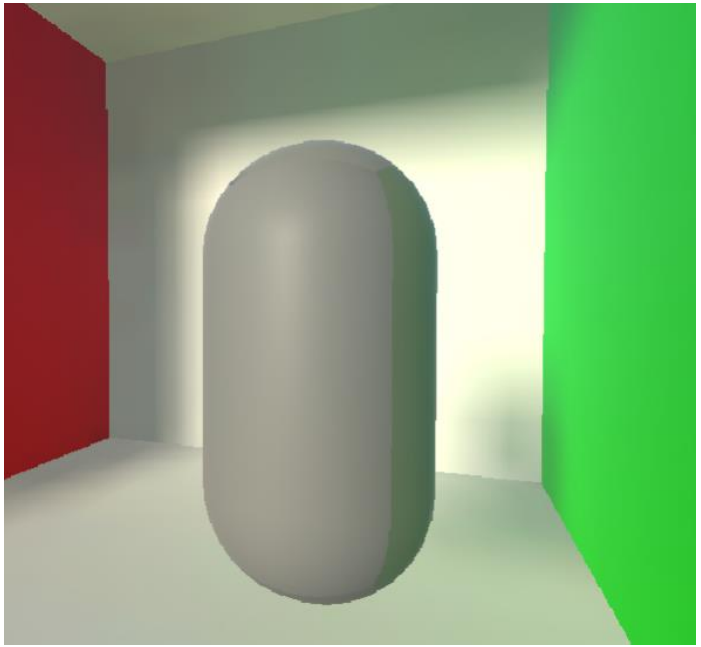




Global Illumination

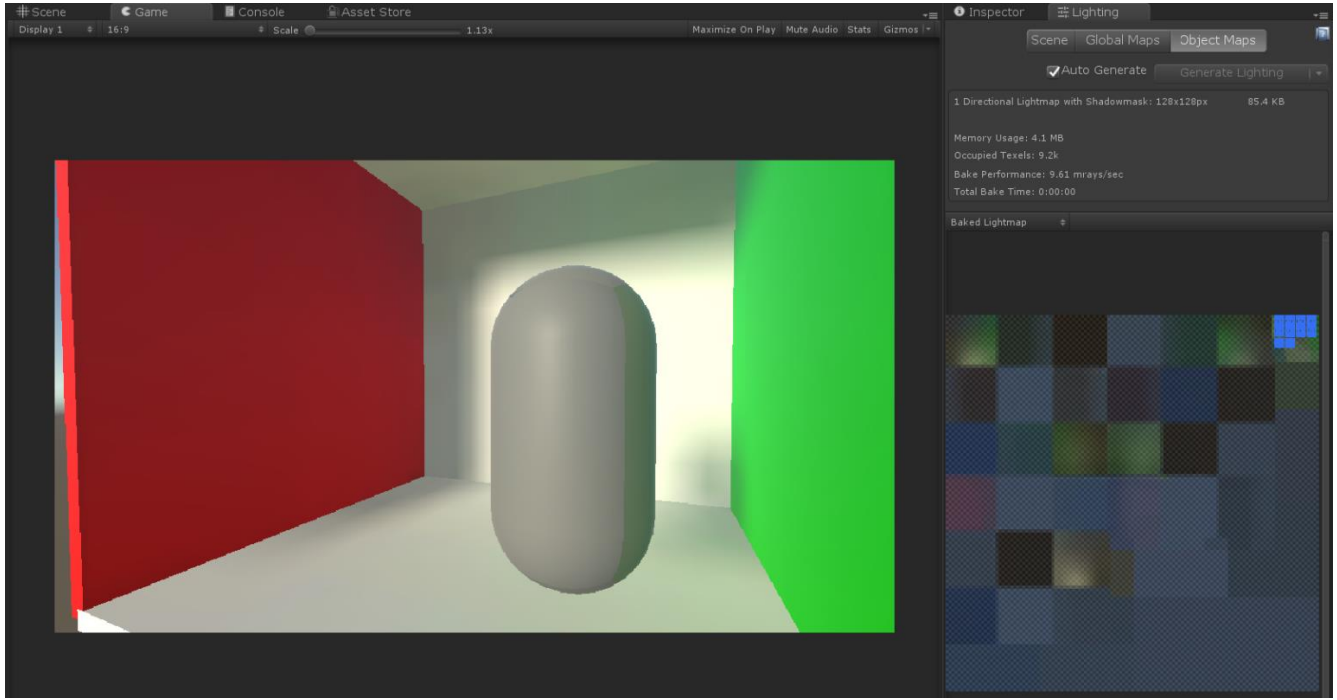
Non-static Capsule vs. Static Capsule



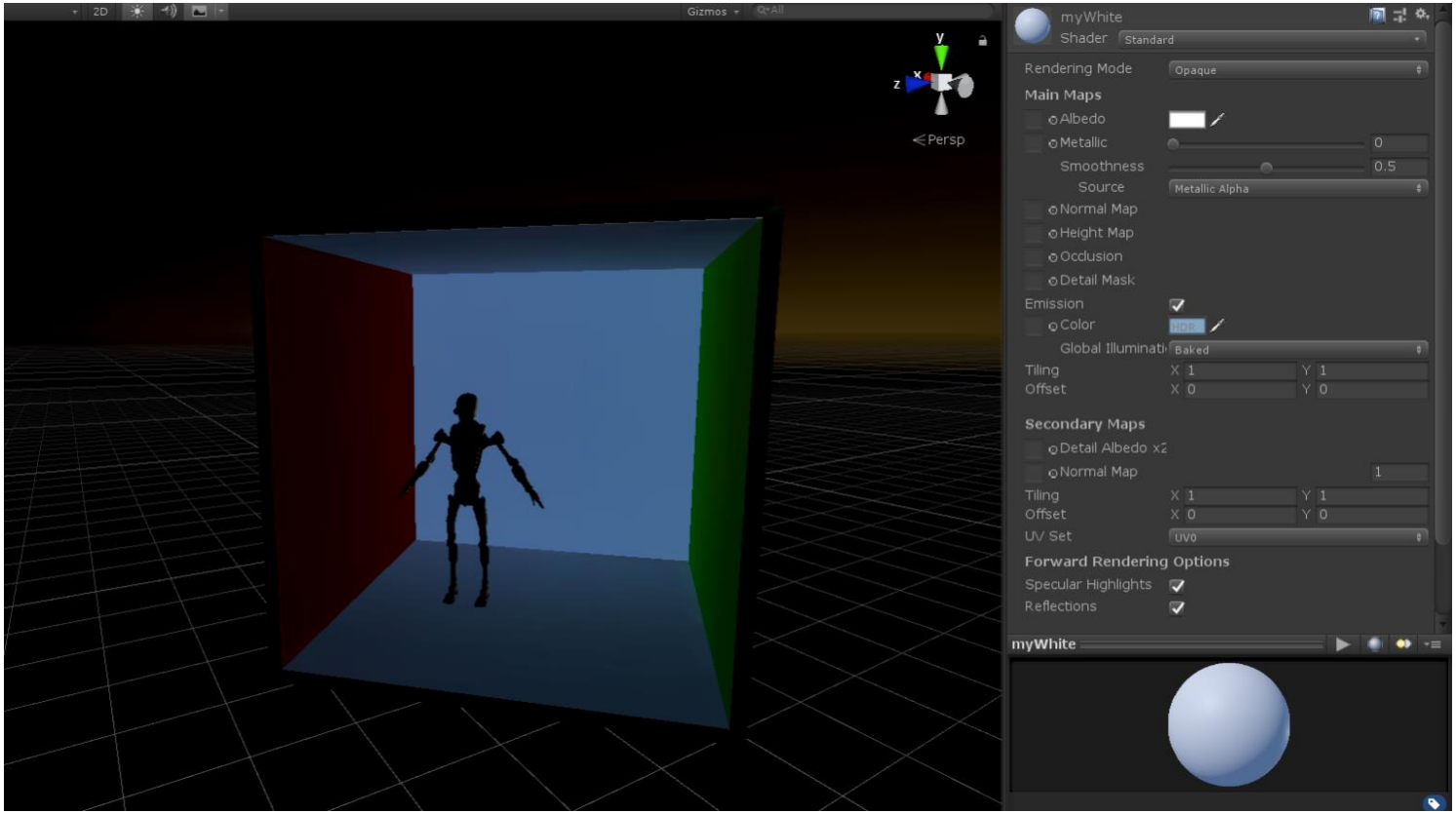


Lightmap

Unity erzeugt für statische Objekte eine Lightmap Texture (“Light baking”)



Selbstleuchtende Objekte

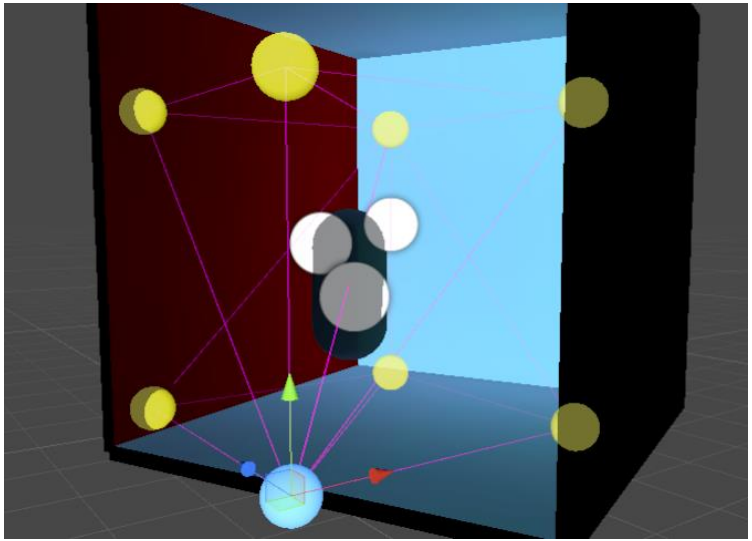


Global Illumination

Indirekte Beleuchtung nicht-statischer Objekte

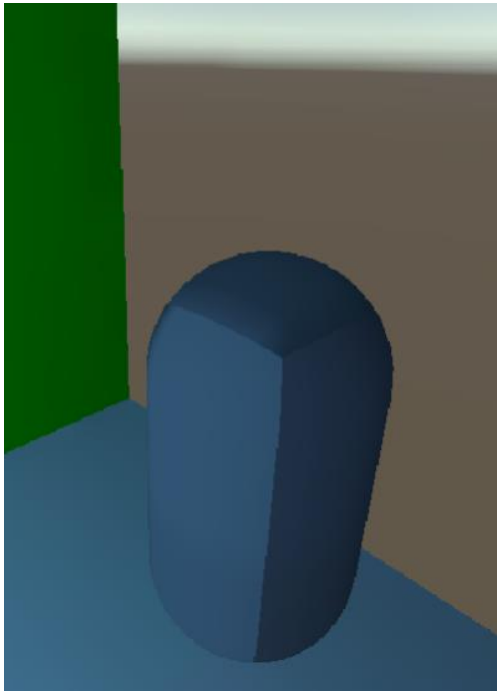
- In Unity: Lightprobe Group
  - Punktweise Erfassen der Beleuchtungsverhältnisse
  - Interpolation des Zwischenraums

- Interessante Bereiche können mit höherer Anzahl Lightprobes beschrieben werden (Add, Duplicate)

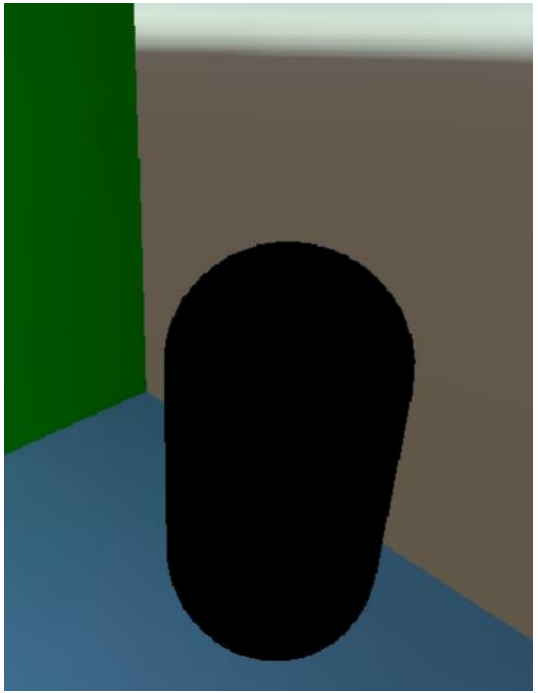


Global Illumination

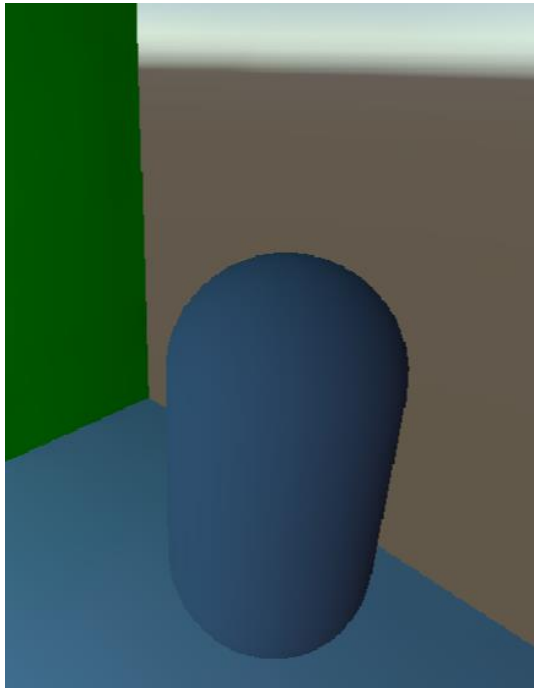
Static Object



Dynamic Object

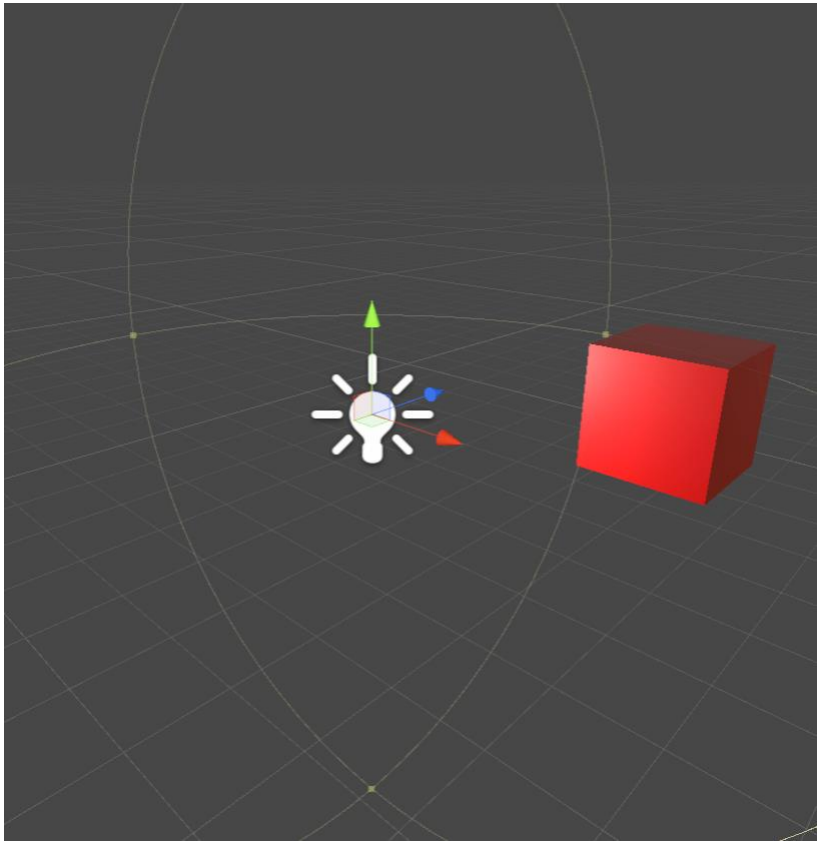


Lightprobes



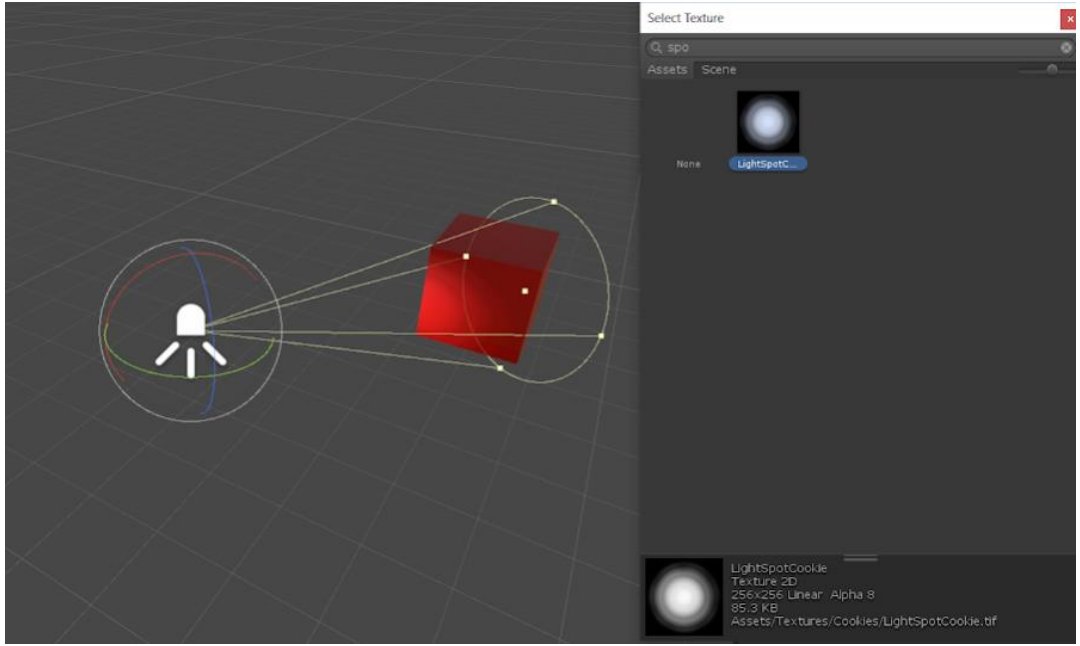
Pointlights

- 4 Pi
- Quadratisches Abklingen
- = 0 für größer “Range”



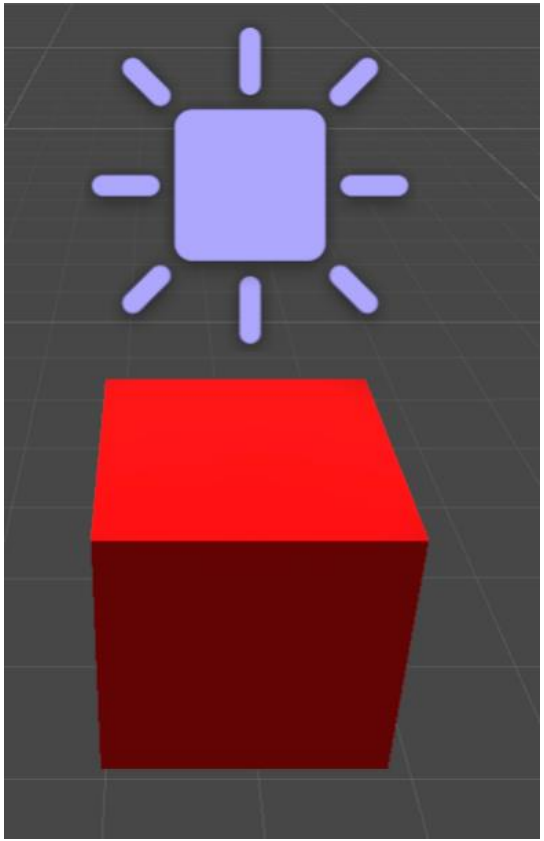
Spotlight

- Gerichtet
- Kegelförmig, Abklingverhalten je nach Unity Version unterschiedlich
- Cookies



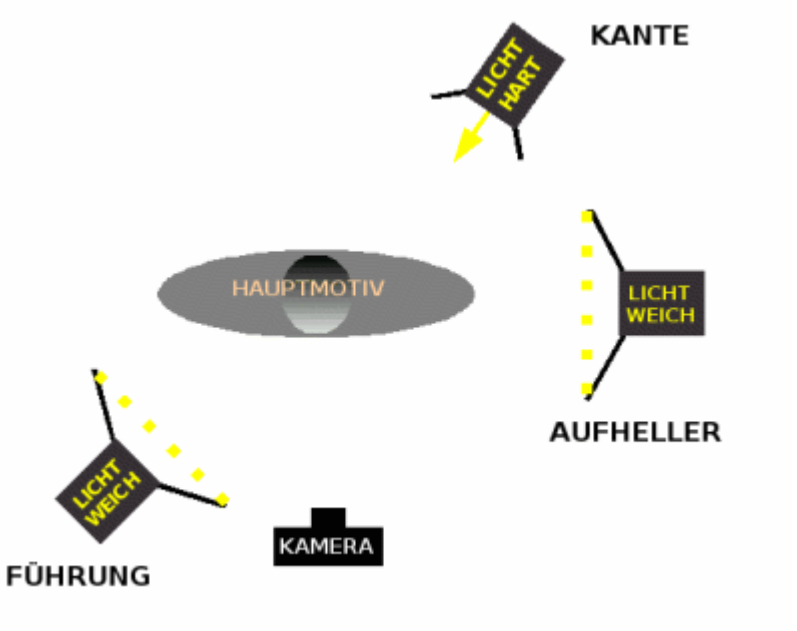
Area Light

- “flat”
- “baked” → nur für statische Objekte



3-Punkt-Beleuchtung

- Führungslicht (key light)
  - höchste Intensität
  - 45° zu Kamera
- Aufhellungslicht (fill light)
  - schräg gegenüber Führungslicht
- Kantenlicht (bump light)
  - von oben, Kopfpartie
  - Kantenaufhellung
- Optional: Hintergrundausleuchtung



(Quelle: Wikipedia)

Übung

“Barbarian Warrior”