

GDC®

Level Design Workshop, “Invisible Intuition...” How To Light A Level

Robert Yang

@radiatoryang

www.debacle.us

 NYU GAME CENTER



ABOUT ME

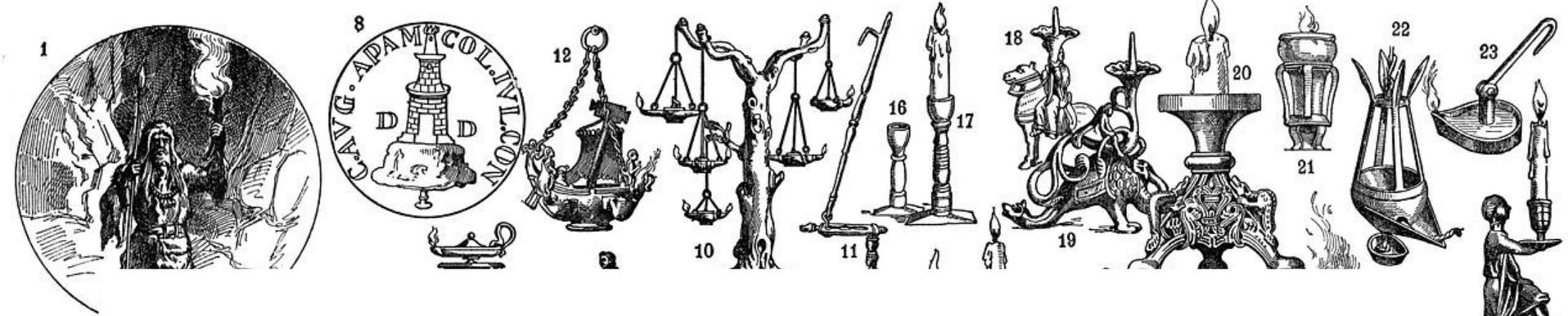




IN THIS TALK:

- 1) a brief history of light
- 2) what is game lighting?
- 3) intro to three-point lighting
- 4) how to light a level





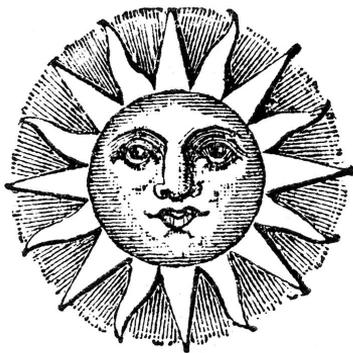
1. a brief history of light





Various light sources through the ages

the sun



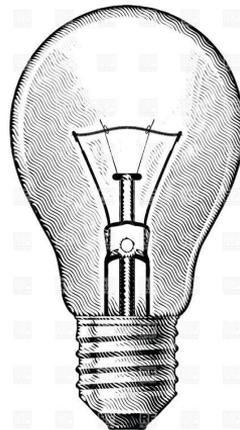
fire



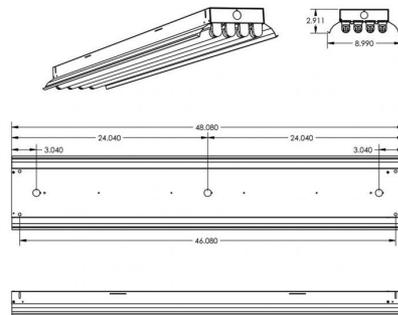
gas light



incandescent



fluorescent



LED





Real-life lighting design today

FL-WS-S¹ Surface Ceiling Light -IP56

Waterproof Fluorescent Light INDOOR LIGHT

Application
Engine room, hall, laundry, store
Chemical processing and industrial areas

Protection degree IP56

Materials
Body : Electro zinc coated steel sheet deep drawn
Diffuser : Clear polycarbonate
Lamp : Stainless steel

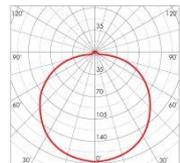
Finish
Body : White epoxy powder coat and stove
enamelled(Munsell No. N9.3)

Equipment
Fluorescent lamp : 18/36W
Light sources : T26
Lamp base : G13
Four cable glands 20x-c (Diameter : Max.16mm cable)

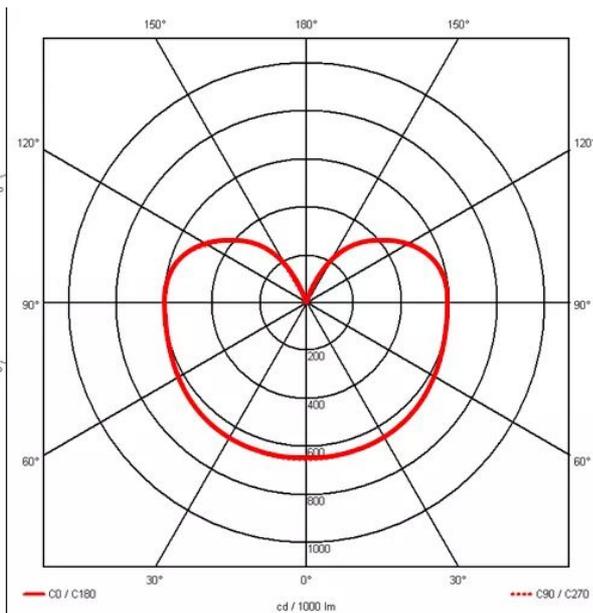
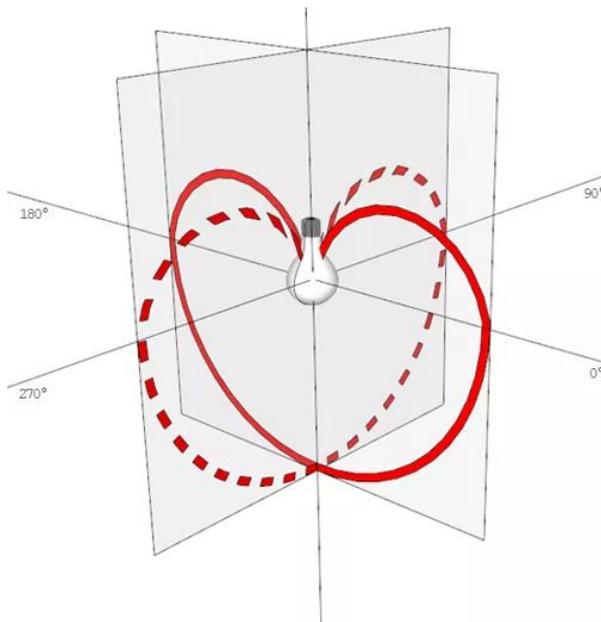
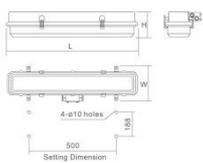
Electrical
AC100-240V, 50/60Hz are available on request
Ballast : Glow, Rapid, Electronic

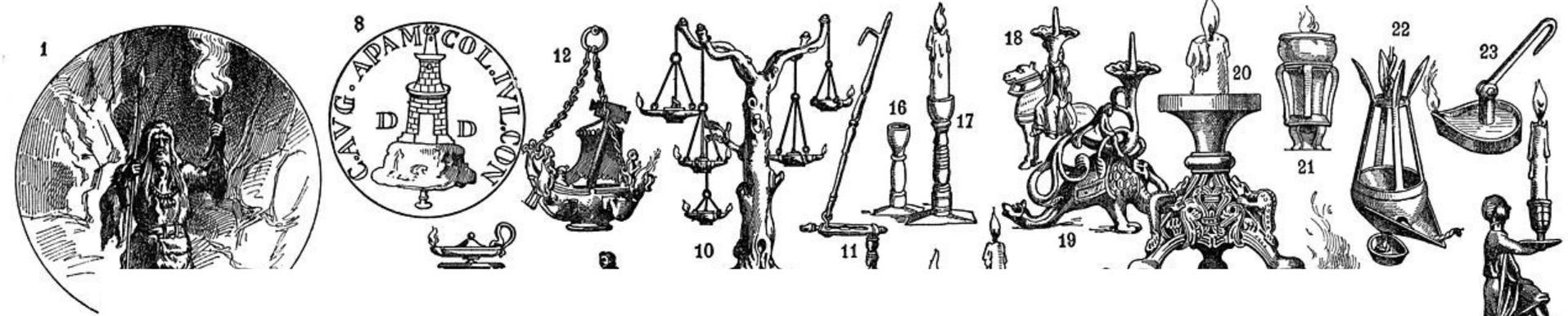


Light Distribution Curve [Cd / 1000lm]



Technical Drawing



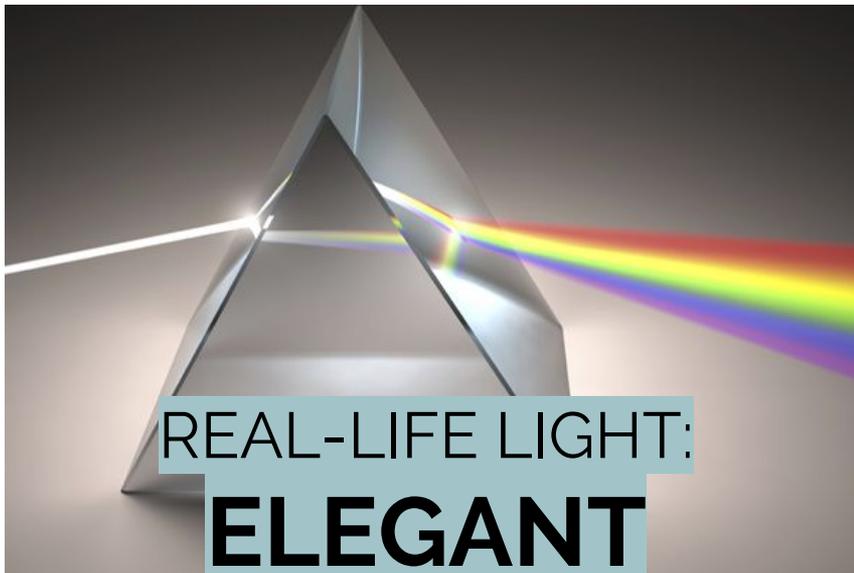


2. what is game lighting?





What is light vs. how games simulate it





Game lighting: also... *cascading shadow maps, refraction shaders, caustics projectors, fog, glow sprites, HDR, SSAO, bloom, SSSSS, SH light probes, light maps, reflection probes, etc...* **IT'S UP TO US TO MAKE THESE HACKS COHERE.**





FOCUS ON THE FUNDAMENTALS!!!

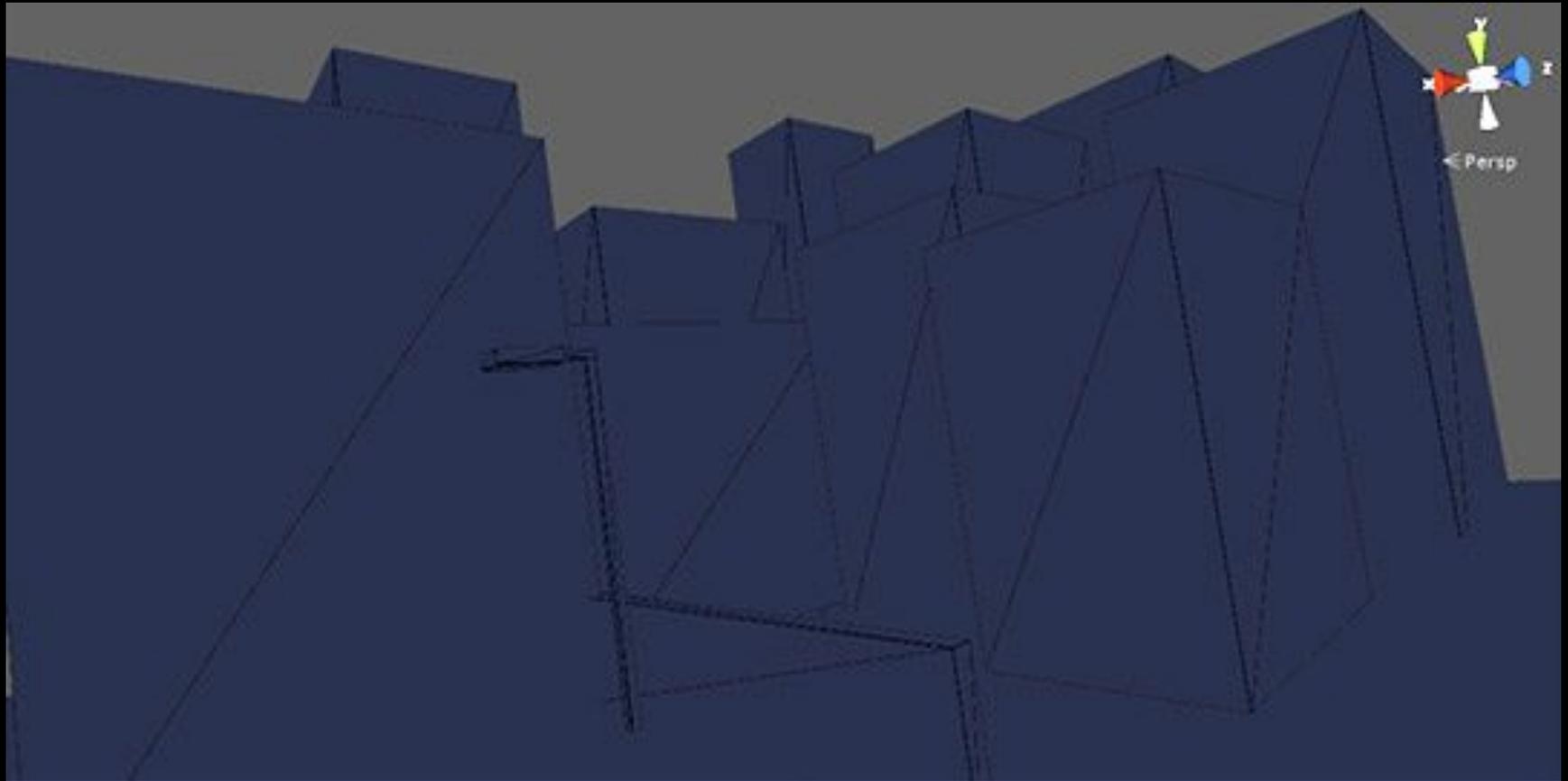
	Directional light	Spotlight
	Ambient light	Point light



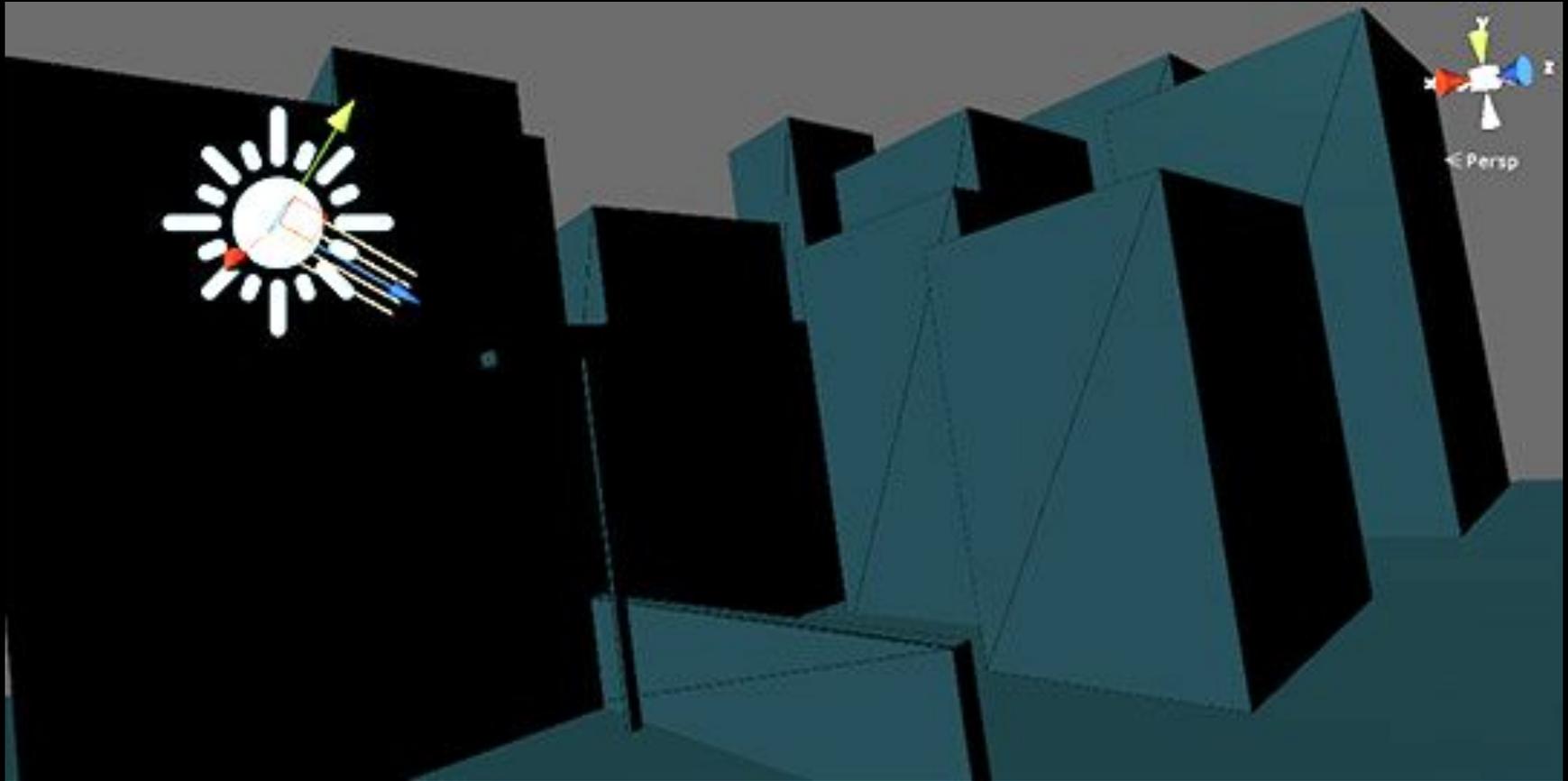
Game lighting and 4 basic light types...



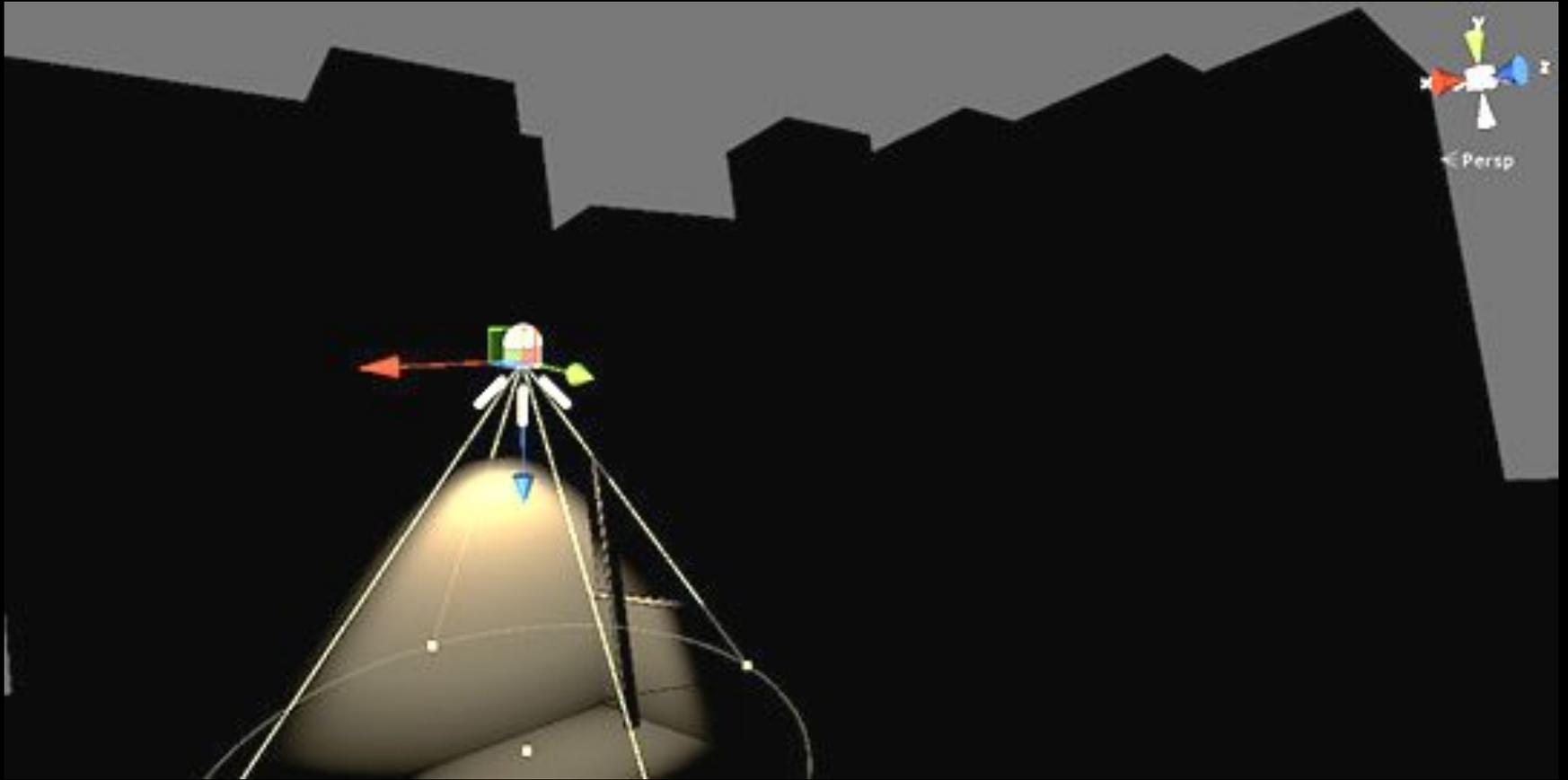
Game lighting: ambient light



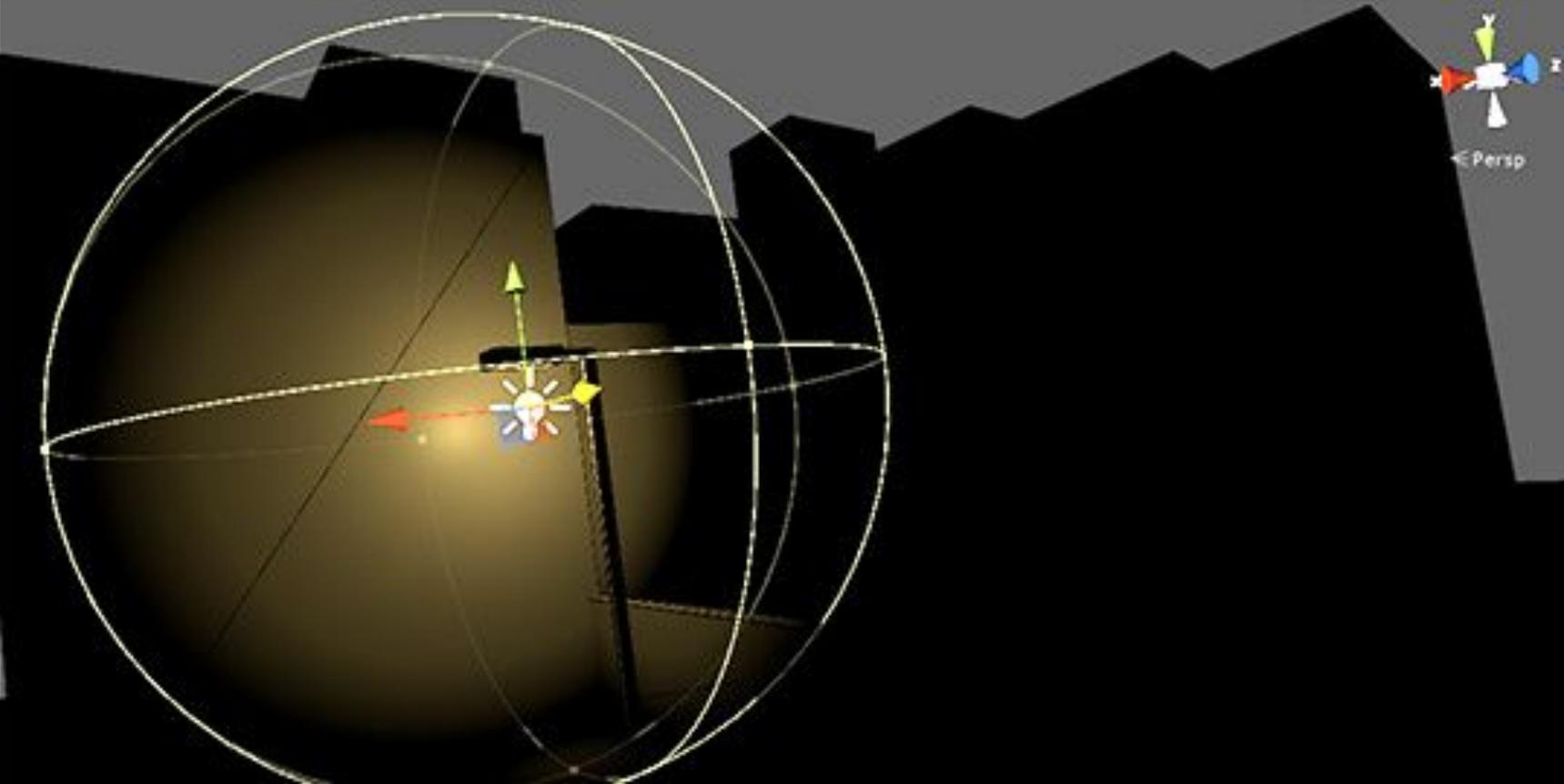
Game lighting: directional light



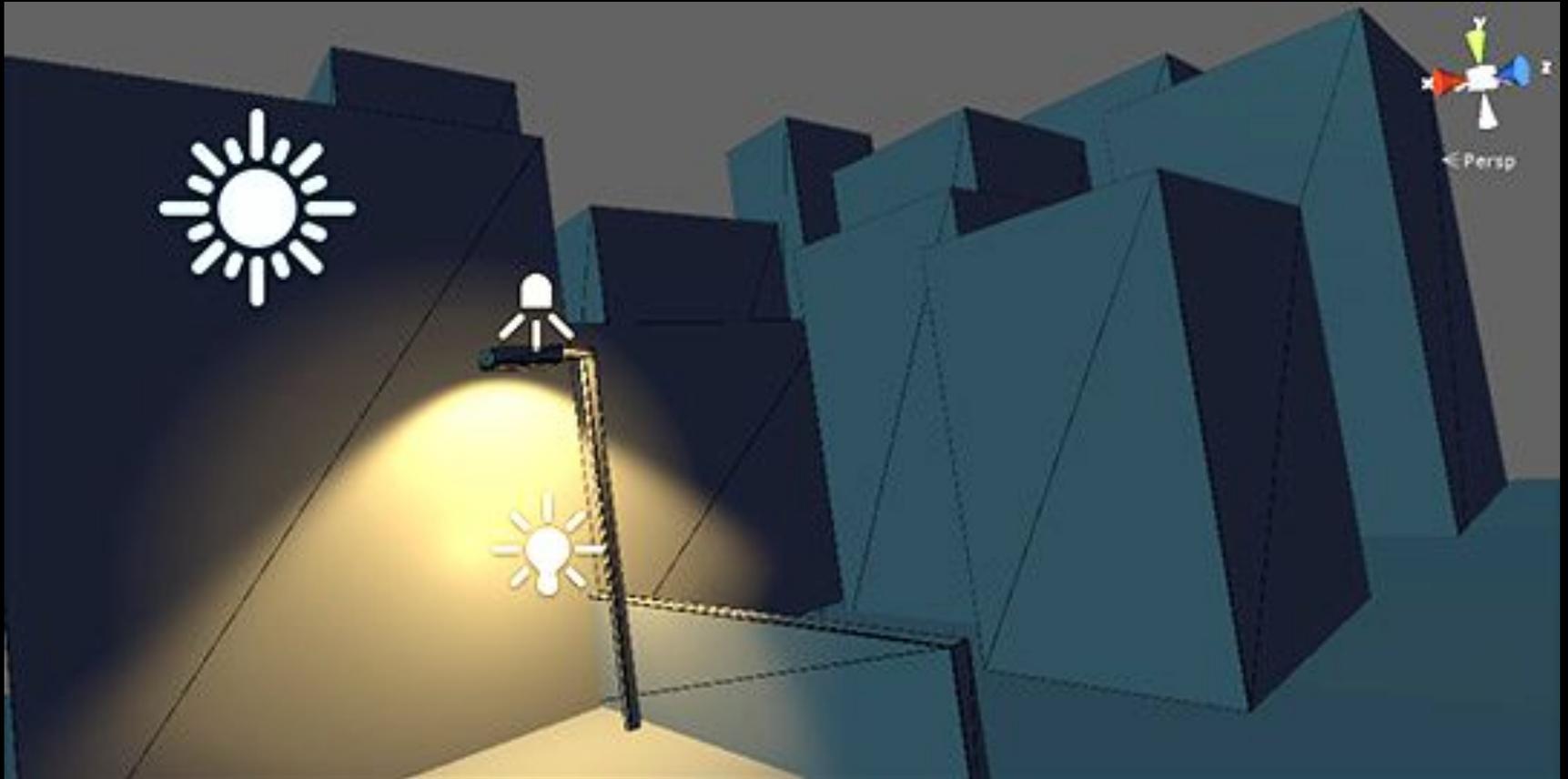
Game lighting: spotlight



Game lighting: point light (“omnidirectional”)



Game lighting: use all (4) types at once!

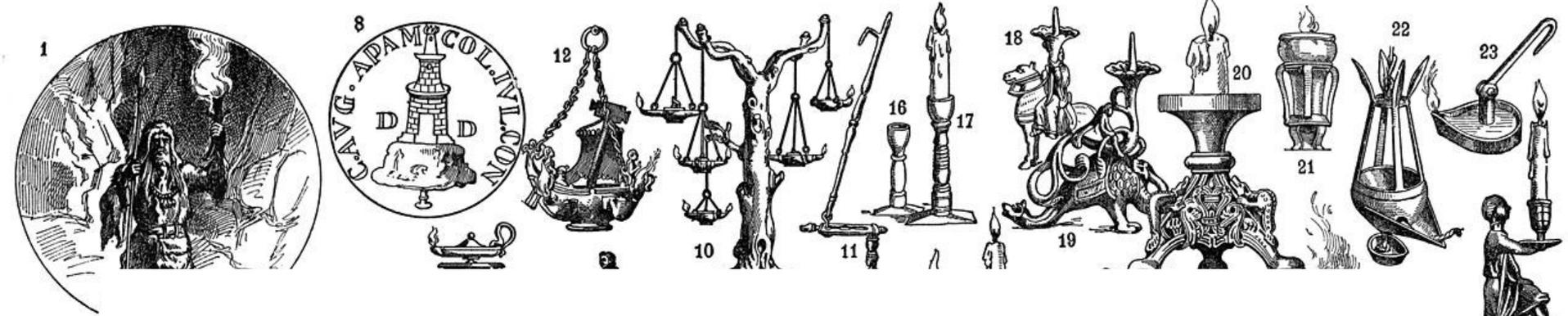




Game lighting: the fundamentals

	Global, affects everything	Local, affects nearby things
Shines in one direction	Directional light	Spotlight
Shines in all directions	Ambient light	Point light





3. three point lighting



Standard three point



COMBINED



Key light



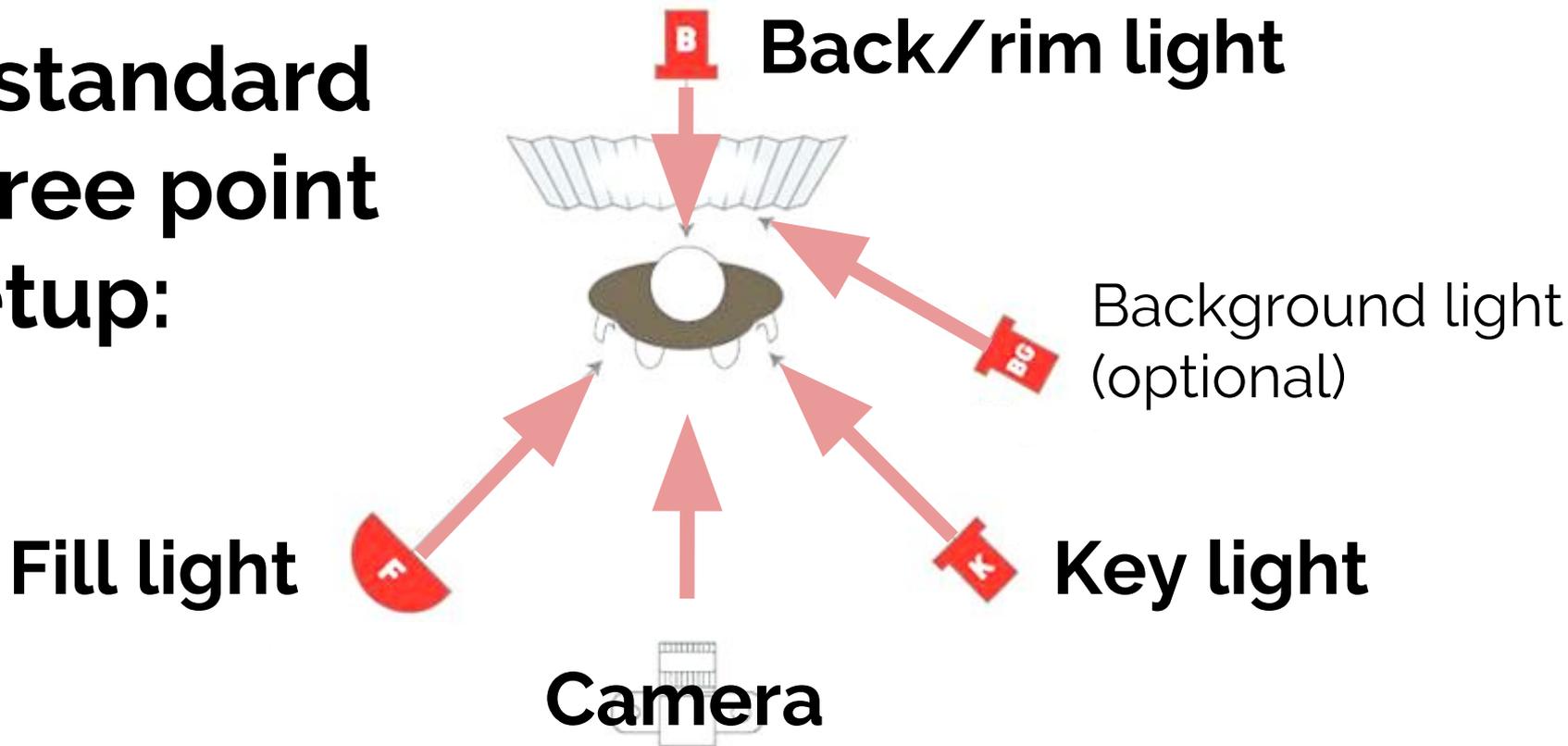
Fill light



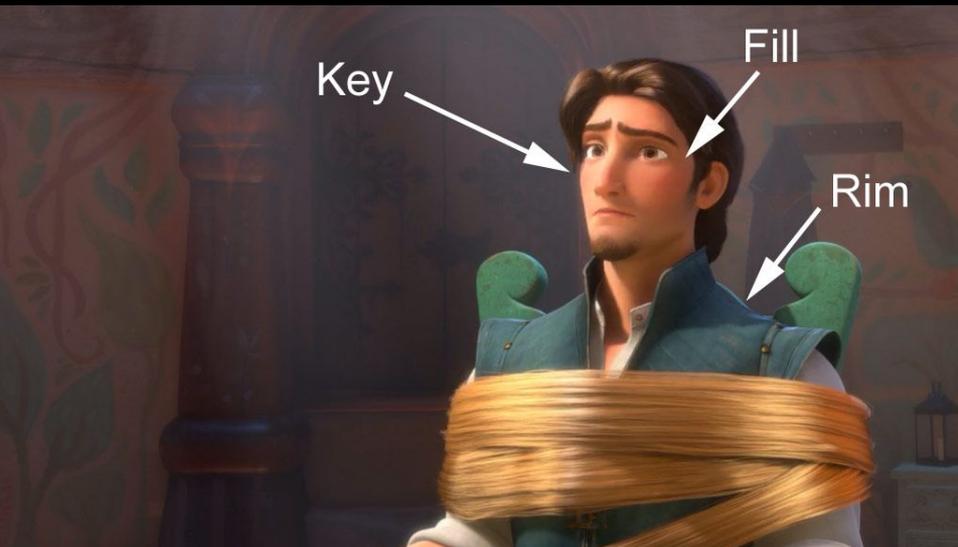
Rim light



A standard three point setup:

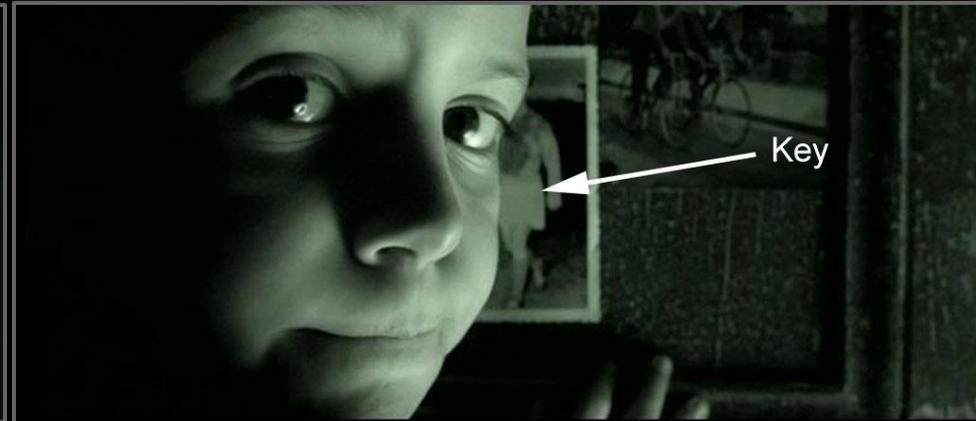


Three point in film



from: <http://lightingpixels.blogspot.com/2013/01/tutorials-does-three-point-lighting-suck.html>

3 point = 3 types, not always 3 lights



from: <http://lightingpixels.blogspot.com/2013/01/tutorials-does-three-point-lighting-suck.html>

Three point example (in Unity)



**TO FILL OR NOT
TO FILL?**

← no fill light is too dark! When in doubt, fill it.

How most games light people:

Key: Directional light

Fill: Ambient / SH probes

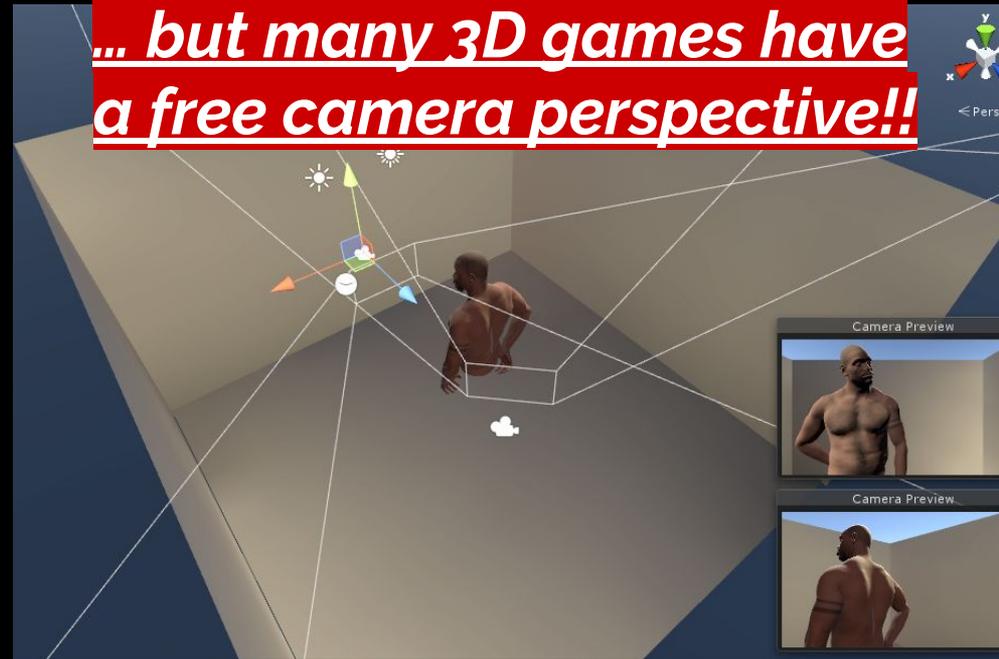
Rim: Fresnel / reflect probes

Where three point theory fails for games

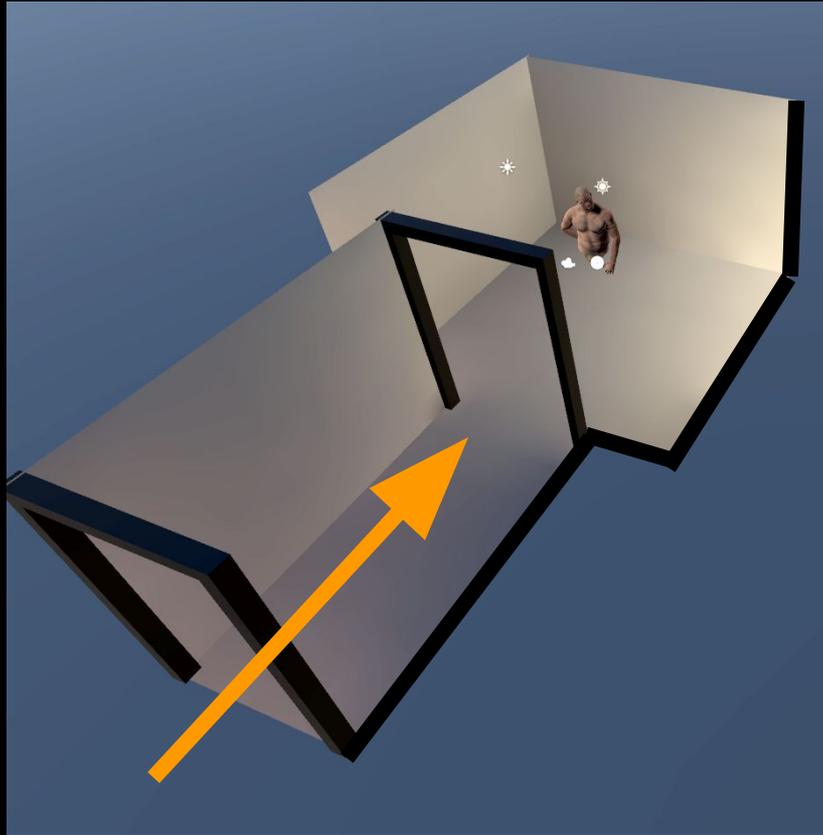


Three point depends on fixed camera perspective...

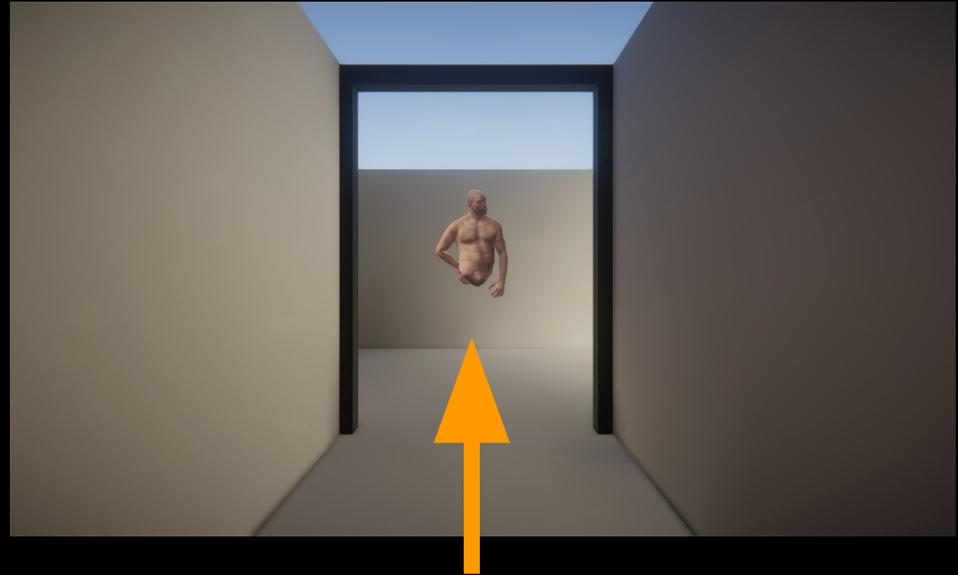
... but many 3D games have a free camera perspective!!

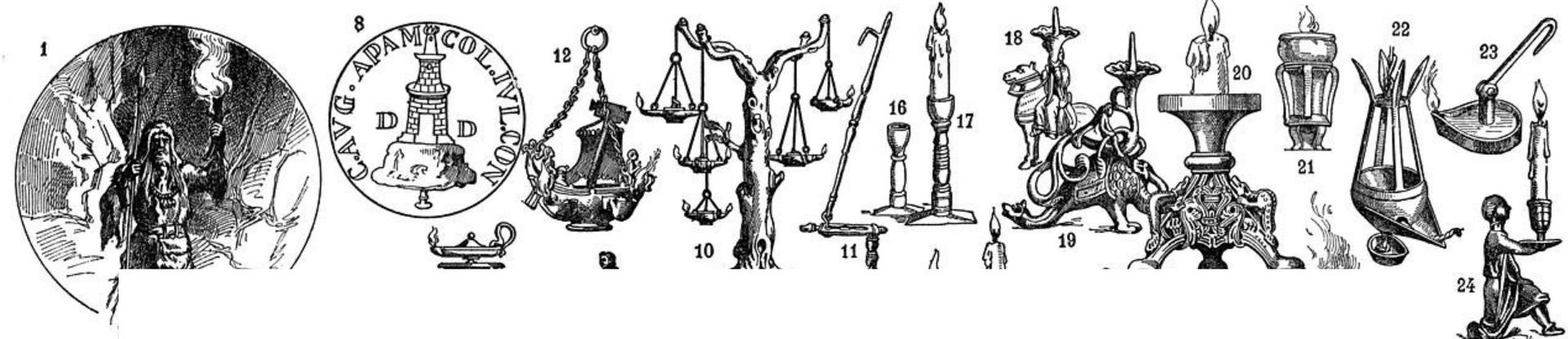


Your level layout is a lighting tool



You can suggest a view /
frame a composition if you
know where the player can go

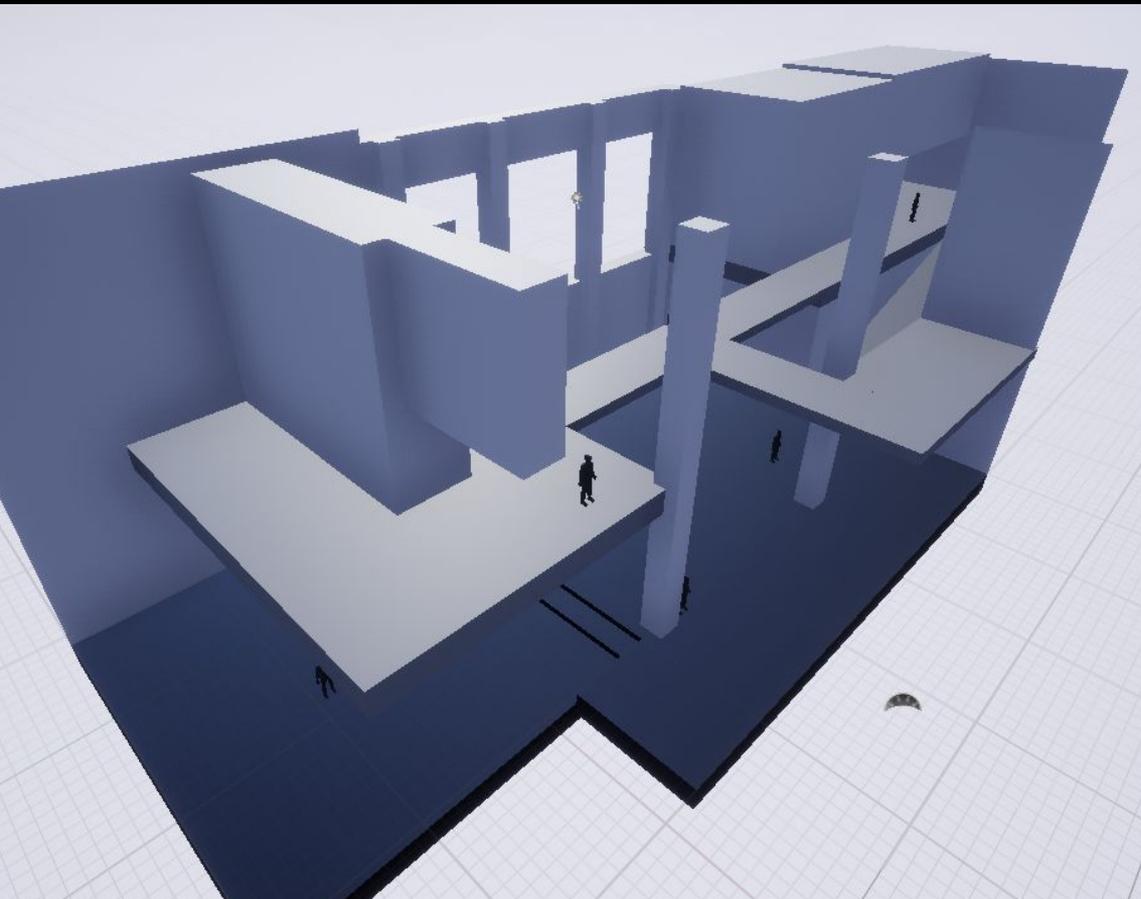




4. how to light a level



STEP 1 **blockout with basic realtime light**



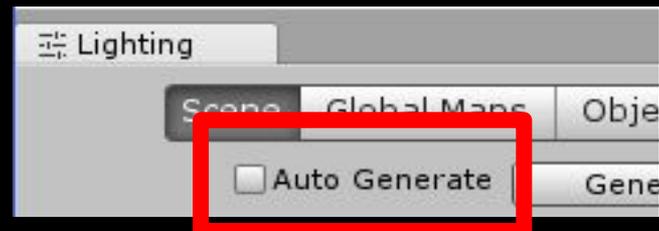
Unreal:

Use temporary “Stationary” lights
“Force No Precomputed Lighting”

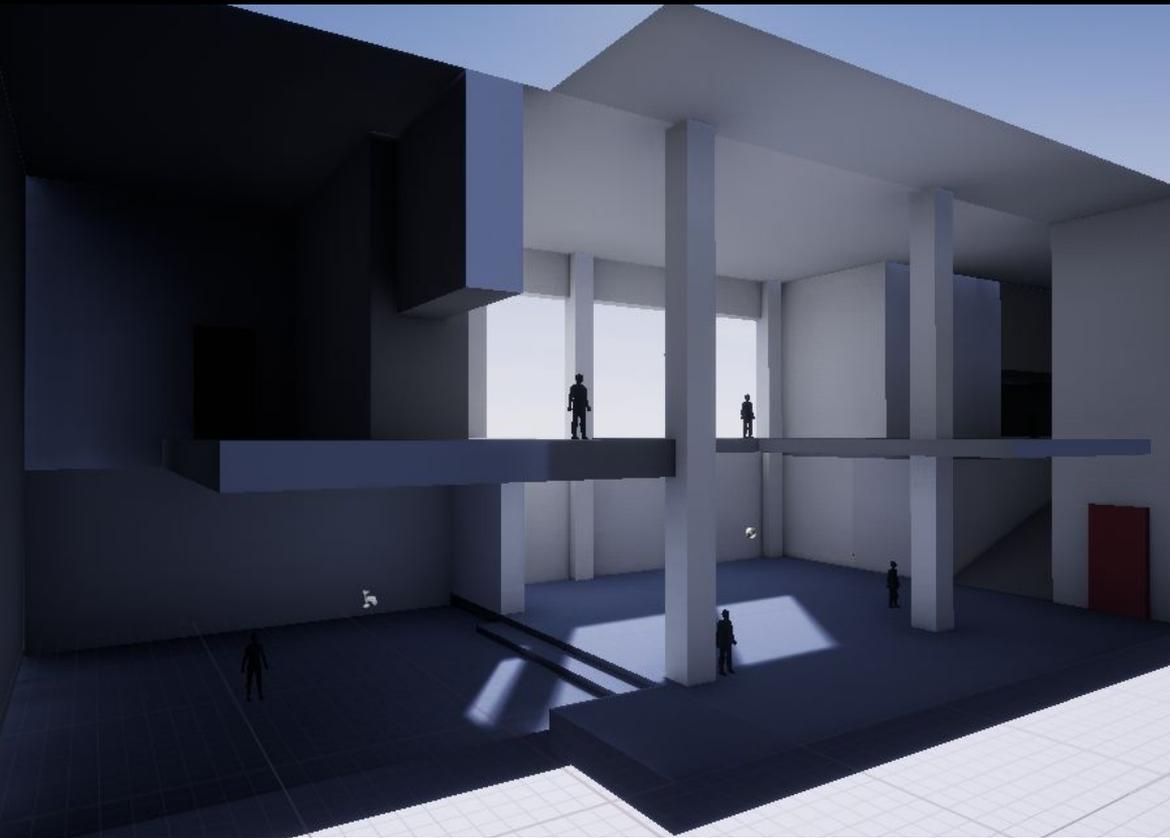


Unity:

Use temporary “Realtime” lights
Disable “Auto Generate” Lighting



STEP 2 **big mood? (ambient, directional)**



Big mood: safe, scary?

KEY LIGHT:

Directional light

FILL LIGHT:

Ambient / light bounce

Unity: check Light Settings

Unreal: check World Settings
("Diffuse Boost") and SkyLight

No GI or bake? Hand-place
faint point lights to fill rooms.

STEP 3 highlight exits and critical path

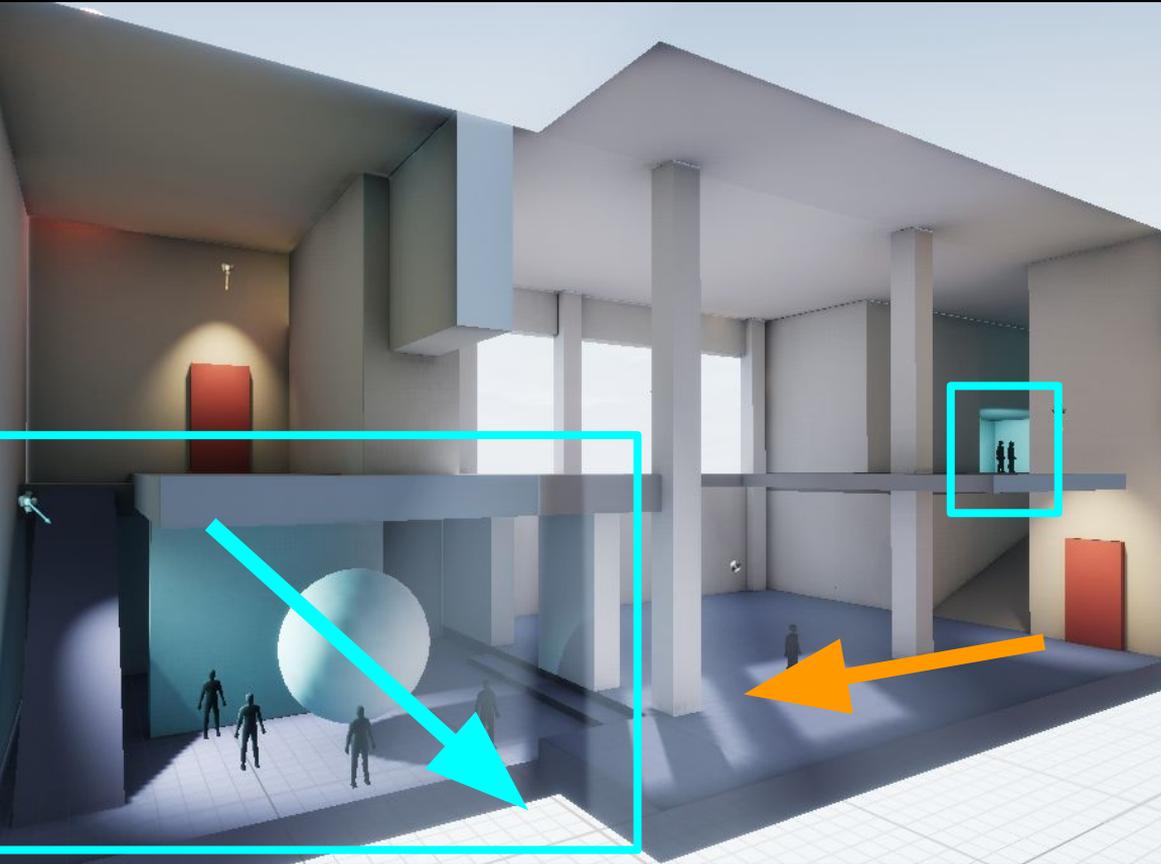


Spotlights act as **key lights** to draw attention to exits

Make a **hierarchy**

- Important exits get more light
- Take layout into account too

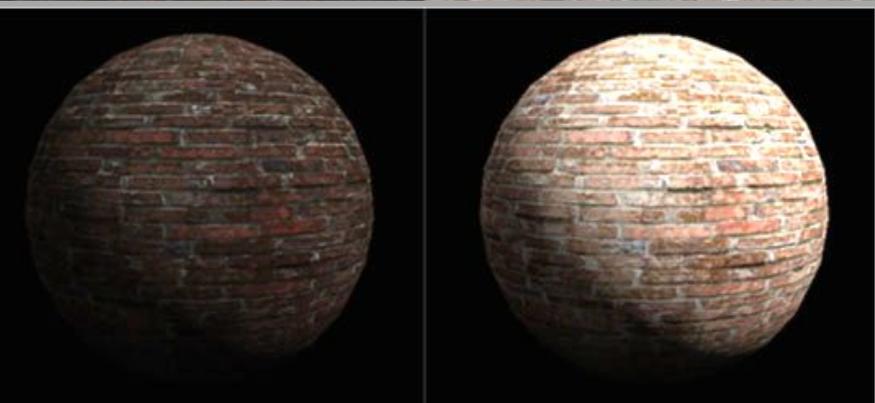
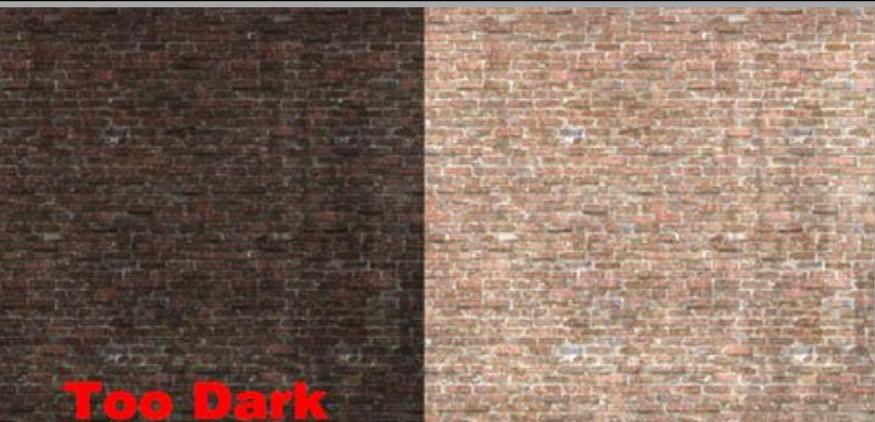
STEP 4 highlight NPCs and setpieces



Spotlights act as **rim/bg lights**, pick out silhouettes

Blue-green lights run perpendicular to critical path, highlight areas for encounters

STEP 5 texture pass; use lighter albedos!



Levels cluster on upper midtones



Mean: 135.95	Level: 186...186
Std Dev: 37.91	Count: 5952
Median: 142	Percentile: 0.76
Pixels: 262144	Cache Level: 1

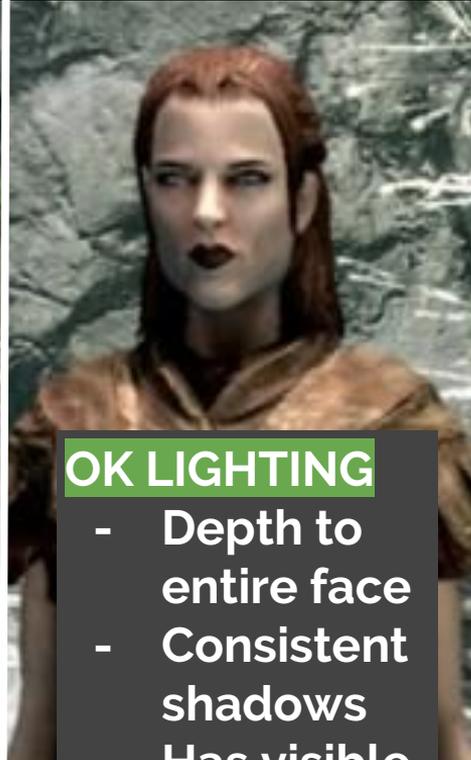
from:
<https://docs.unrealengine.com/udk/Three/TexturingGuidelines.html>

BTW: LIGHT IS POLITICAL!!!



BAD LIGHTING

- Flattening
- Very dark shadows, no jawline?
- Dim key?
- Zero fill?



OK LIGHTING

- Depth to entire face
- Consistent shadows
- Has visible neck

Skyrim example from
“**Black Skin Is Still A Radical Concept in Video Games**” by Yussef Cole and Tanya DePass

https://waypoint.vice.com/en_us/article/78qpxd/black-skin-is-still-a-radical-concept-in-video-games

STEP 6 **iterate, but know when to move on**



Keep iterating and
tweaking lights...
explore variations!!!

But also recognize
when it's good
enough, and **move
on with your life!!**

"Sci-Fi Bunk Lighting" variations, Guillaume Lauer
<https://www.artstation.com/artwork/yrGlR>



to review....



To review:

1. Lighting sets mood, evokes culture
2. Directional, ambient, point, spot
3. 3P: key light, fill light, rim/back light
4. Light globally, then for gameplay
5. Know when to stop and move on





THANKS FOR LISTENING.

Robert Yang

my twitter: [@radiatoryang](https://twitter.com/radiatoryang)

my games / writing: debacle.us

I work / teach at: gamecenter.nyu.edu





bonus slides



Don't take my word for it

Magnar Jenssen (Valve), "Functional Lighting"

http://magnari.net/article_funlight.html





Random lighting tips

- Don't rely on blue and orange lights ("blorange")... there are more than 2 colors! try googling "bisexual lighting"
- Fewer lights is usually better, creatively and technically
- But use more than one light, for god's sake, I'm begging
- Avoid overlapping 4+ lights in Unity or Unreal
- You don't always need a light fixture or visible light source... but still, think about the plausibility of the light!





Read this

Architecture: Form, Space, and Order by Francis Ching
intro to architectural thinking / breaking down shapes

Moving Frostbite to Physically Based Rendering 3.0 (course notes, SIGGRAPH 2014) by Sebastien Lagarde
technical primer, you won't understand most of it unless you're a graphics programmer, but try to read it anyway





Tips for lighting in Unreal 4.xx

- “Mobility” matters A LOT... Stationary is *VERY* different vs Static
<https://docs.unrealengine.com/en-us/Engine/Rendering/LightingAndShadows/LightMobility>
- To reduce lightmap / shadow bleeding:
 - Test build lighting on “Production” quality, make sure it’s a real problem
 - Redo the UV unwrap, allow for proper padding between UV islands
<https://docs.unrealengine.com/en-us/Engine/Content/Types/StaticMeshes/LightmapUnwrapping>
- To get sharper lightmaps:
 - Adjust LM resolution on static mesh (*see Static Mesh Editor*)
 - Change global ratio to a bigger number (higher res is any # > 100%)
(*see Build > Lighting Info > Lightmap Resolution Adjustment*)
- Read “Lighting Troubleshooting” guide on Unreal wiki
<https://wiki.unrealengine.com/LightingTroubleshootingGuide>





Tips for lighting in Unity 201x.xx

- Mark level geometry (walls, floors) as "Static" (especially as Lightmap Static)
- Use "Progressive Lightmapper" (in Light Settings)... I'm sure there are some masochists who like Enlighten, but the rest of us think Enlighten is too slow and gives worse results. <https://docs.unity3d.com/Manual/ProgressiveLightmapper.html>
- Disable "Auto-Generate" (in Light Settings)... it's often distracting when Unity constantly rebakes the lighting. <https://docs.unity3d.com/Manual/GlobalIllumination.html>
- To reduce lightmap bleeding / seam problems:
 - Bake with higher Lightmap Resolution, make sure it's a real problem
 - Redo the UV unwrap, allow for proper padding between UV islands
<https://docs.unrealengine.com/en-us/Engine/Content/Types/StaticMeshes/LightmapUnwrapping>

