

Grafické efekty

Jan Zelený

Senior Engine Programmer



jan.zeleny@bistudio.com

Úvodem

Úvodem

- Bohemia Interactive - Enfusion



Úvodem

- Bohemia Interactive - Enfusion
- Mashinky – Vláčková Strategie



Úvodem

- Bohemia Interactive - Enfusion
- Mashinky – Vláčková Strategie
- 6 let v 2K Czech (Mafia II + III)



Úvodem

- Bohemia Interactive - Enfusion
- Mashinky – Vláčková Strategie
- 6 let v 2K Czech (Mafia II + III)
- První hra 2003 – U Becherů



Úvodem

- Bohemia Interactive - Enfusion
- Mashinky – Vláčková Strategie
- 6 let v 2K Czech (Mafia II + III)
- První hra 2003 – U Becherů

- Specializace
 - Skin / Hair / Eye shader / SSS
 - Wrinkles / GPU Morphs
 - Depth of Field / Motion Blur
 - Water / Deferred decals / Terrain



Grafické efekty ve hrách



- Mnoho změn za pár let. Nové postupy, metody
- Jeden z nejdynamičtější se rozvíjejících oborů v herní tvorbě

Realismus vs stylizace

- Realistická grafika

Realismus vs stylizace

- Realistická grafika
 - Physically Based Rendering



Realismus vs stylizace

- Realistická grafika
 - Physically Based Rendering



- Šíření světla
- Chování materiálů

Realismus vs stylizace

- Stylizovaná grafika

Realismus vs stylizace

- Stylizovaná grafika
 - Mnoho možností, směrů



Realismus vs stylizace

- Stylizovaná grafika
 - Mnoho možností, směrů



Realismus vs stylizace

- Stylizovaná grafika
 - Mnoho možností, směrů



Realismus vs stylizace

- Stylizovaná grafika
 - Mnoho možností, směrů



Realismus vs stylizace

- Stylizovaná grafika
 - Mnoho možností, směrů



Realismus vs stylizace

- Stylizovaná grafika
 - Mnoho možností, směrů
- Stejný herní obsah se dá grafikou výrazně odlišit
 - Cílová skupina
 - Atmosféra hry
 - Hardwarové možnosti

Realismus vs stylizace

- Stylizovaná grafika
 - Mnoho možností, směrů
- Stejný herní obsah se dá grafikou výrazně odlišit
 - **Cílová skupina**
 - **Atmosféra hry**
 - **Hardwarové možnosti**

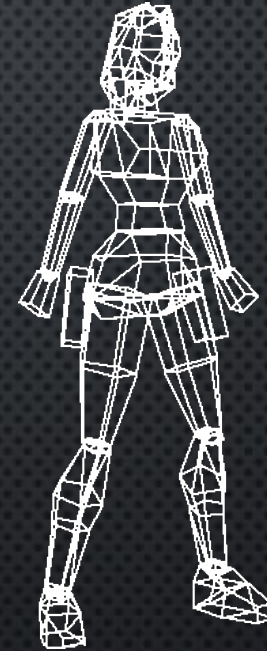


Realistická grafika

- Geometrie objektu
- Přímé nasvětlení
- Stíny
- Nepřímé nasvětlení a odrazy
- Physically based rendering
- Optimalizace
- Zpracování výsledného obrazu

Realistická grafika

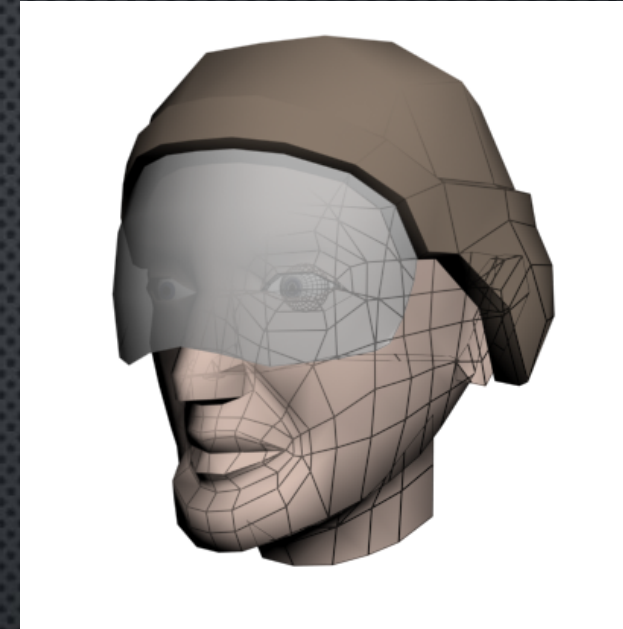
- Geometrie objektu
- Přímé nasvětlení
- Stíny
- Nepřímé nasvětlení a odrazy
- Physically based rendering
- Optimalizace
- Zpracování výsledného obrazu



Geometrie objektu - získání

Geometrie objektu - získání

- Získání modelu s co nejvíce polygony
 - Problém modelovat **konvenční** metodou



Geometrie objektu - získání

- Získání modelu s co nejvíce polygony
 - Problém modelovat **konvenční** metodou
 - **Sculpting** - Zbrush, MudBox
 - + Přirozenější
 - + Adaptivní detaily
 - Nutnost zjednodušení
 - Silnější hardware na práci



Geometrie objektu - získání

- Získání modelu s co nejvíce polygony
 - Problém modelovat **konvenční** metodou
 - **Sculpting** - Zbrush, MudBox
 - **Procedurální** generování
 - + Variabilita
 - + Úspornost
 - Výpočetní náročnost



Geometrie objektu - získání

- Získání modelu s co nejvíce polygony
 - Problém modelovat **konvenční** metodou
 - **Sculpting** - Zbrush, MudBox
 - **Procedurální** generování
 - **Scanování**
 - + Realismus
 - Realismus



Geometrie objektu - získání

- **Fotogrammetrie**
 - Rekonstrukce 3D objektů z fotografií



Geometrie objektu - získání

- **Fotogrammetrie**
 - Rekonstrukce 3D objektů z fotografií
 - + Získání Geometrie i Textur



Geometrie objektu - získání

- **Fotogrammetrie**
 - Rekonstrukce 3D objektů z fotografií
 - + Získání Geometrie i Textur
 - Zapečená světla



Geometrie objektu - získání

- **Fotogrammetrie**
 - Rekonstrukce 3D objektů z fotografií
 - + Získání Geometrie i Textur
 - Zapečená světla
 - Jen to, co je vidět



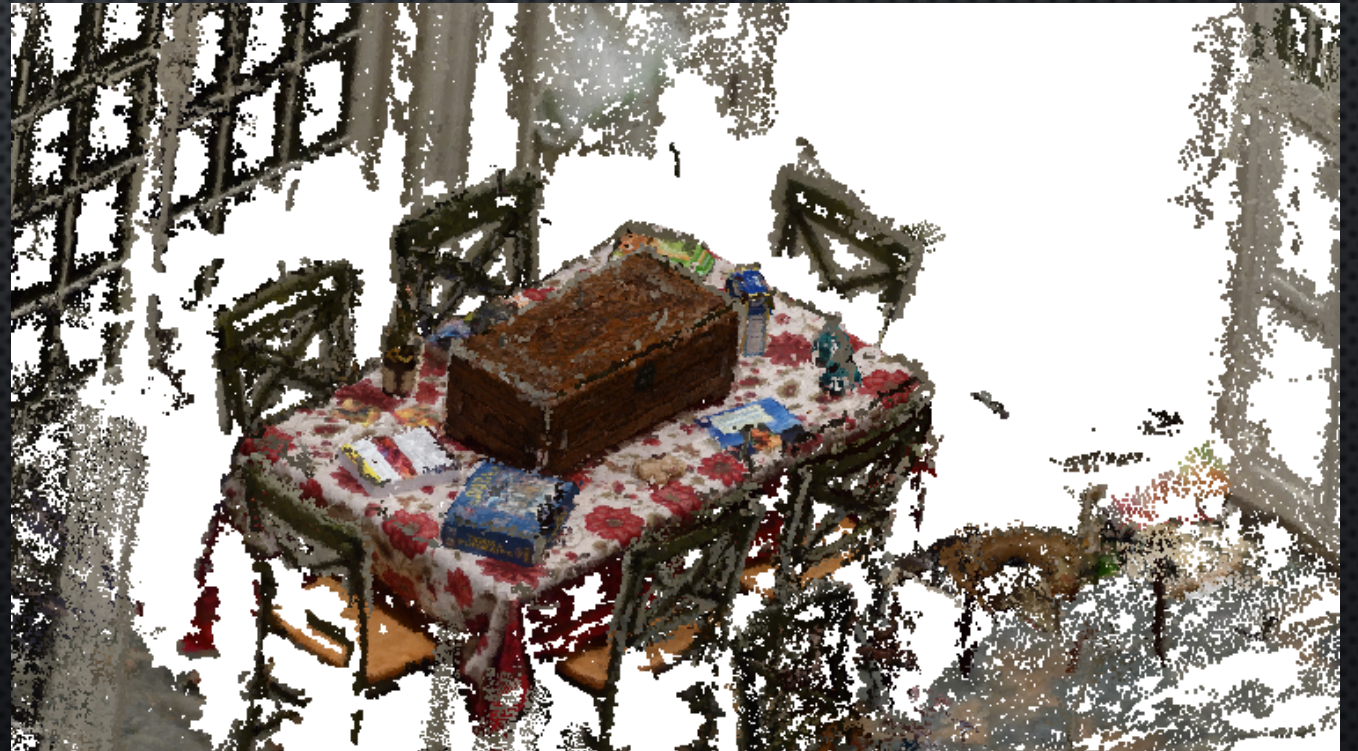
Geometrie objektu - získání

- **Fotogrammetrie**
 - Rekonstrukce 3D objektů z fotografií
 - + Získání Geometrie i Textur
 - Zapečená světla
 - Jen to, co je vidět
 - Objekt musí existovat :-)



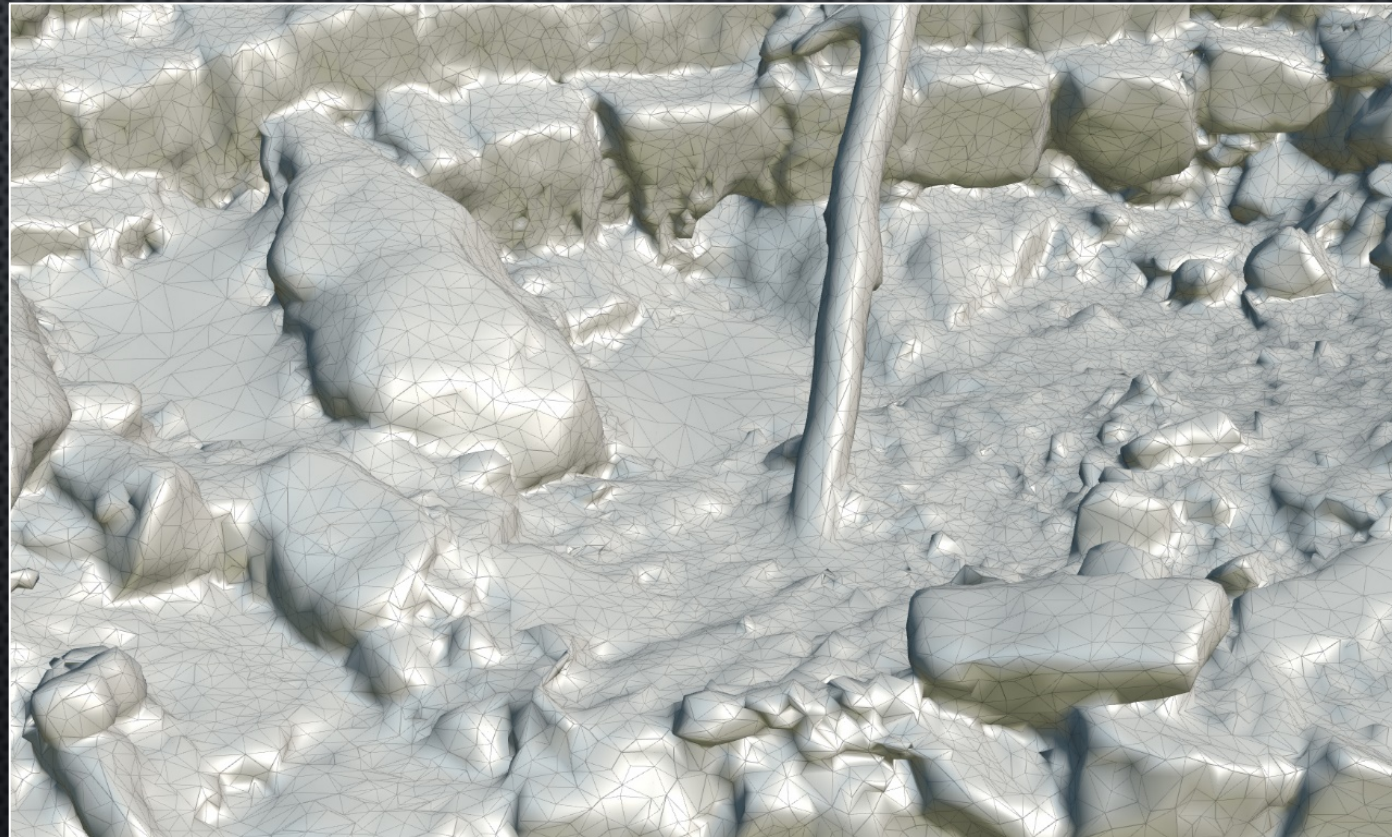
Geometrie objektu - získání

- **Fotogrammetrie**
 - Rekonstrukce 3D objektů z fotografií
 - + Získání Geometrie i Textur
 - Zapečená světla
 - Jen to, co je vidět
 - Objekt musí existovat :-)
 - Náchylné na šum, algor. složité



Geometrie objektu - získání

- Fotogrammetrie



Geometrie objektu - získání

- Fotogrammetrie

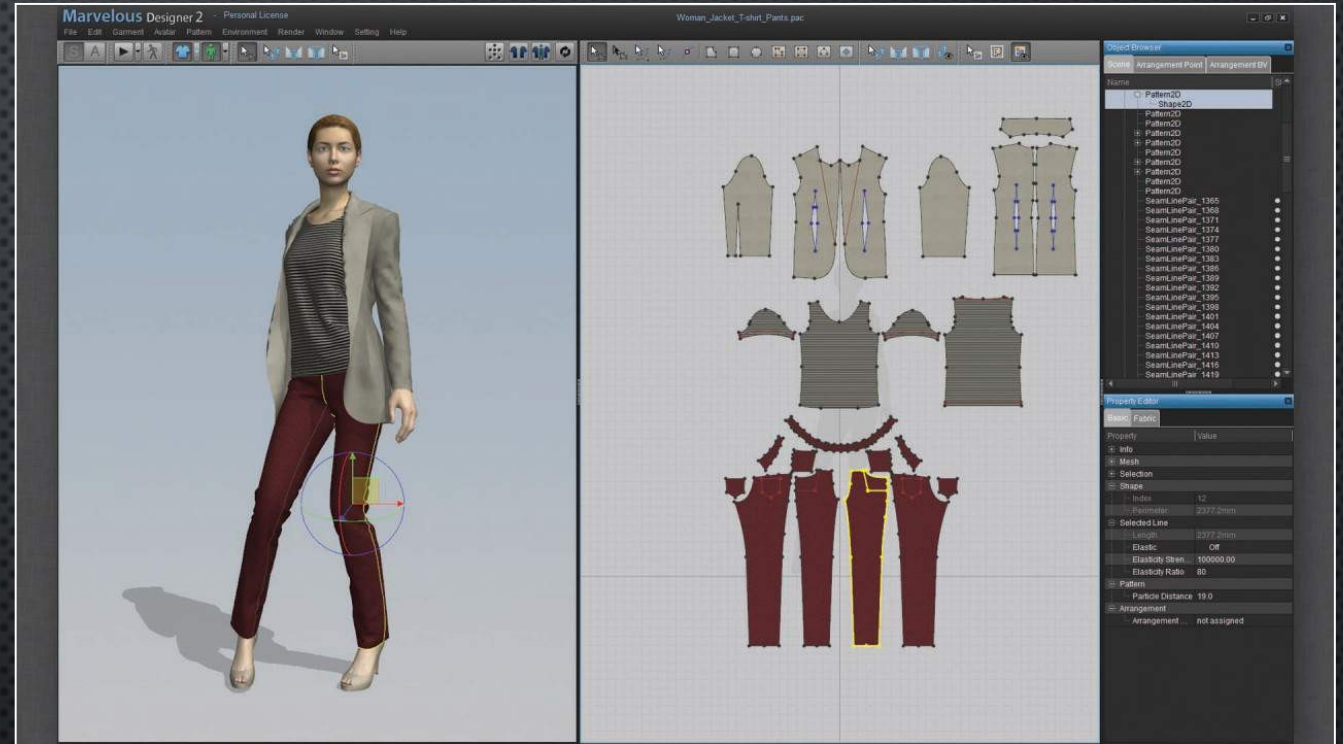


Geometrie objektu - získání

- **Marvelous** Designer

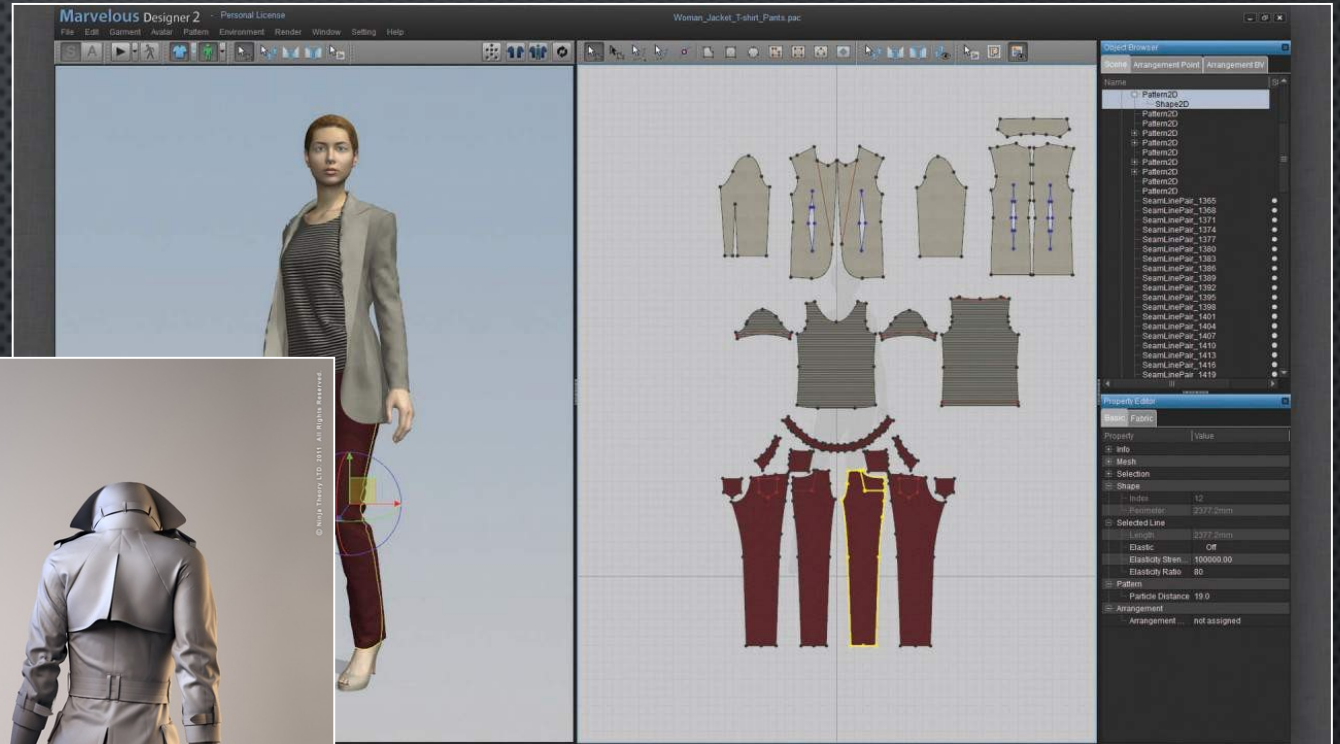
Geometrie objektu - získání

- **Marvelous Designer**
 - Grafik určí „**střih**“ a kde „**sešít**“



Geometrie objektu - získání

- **Marvelous Designer**
 - Grafik určí „**střih**“ a kde „**sešít**“
 - Zbytek udělá **fyzika**

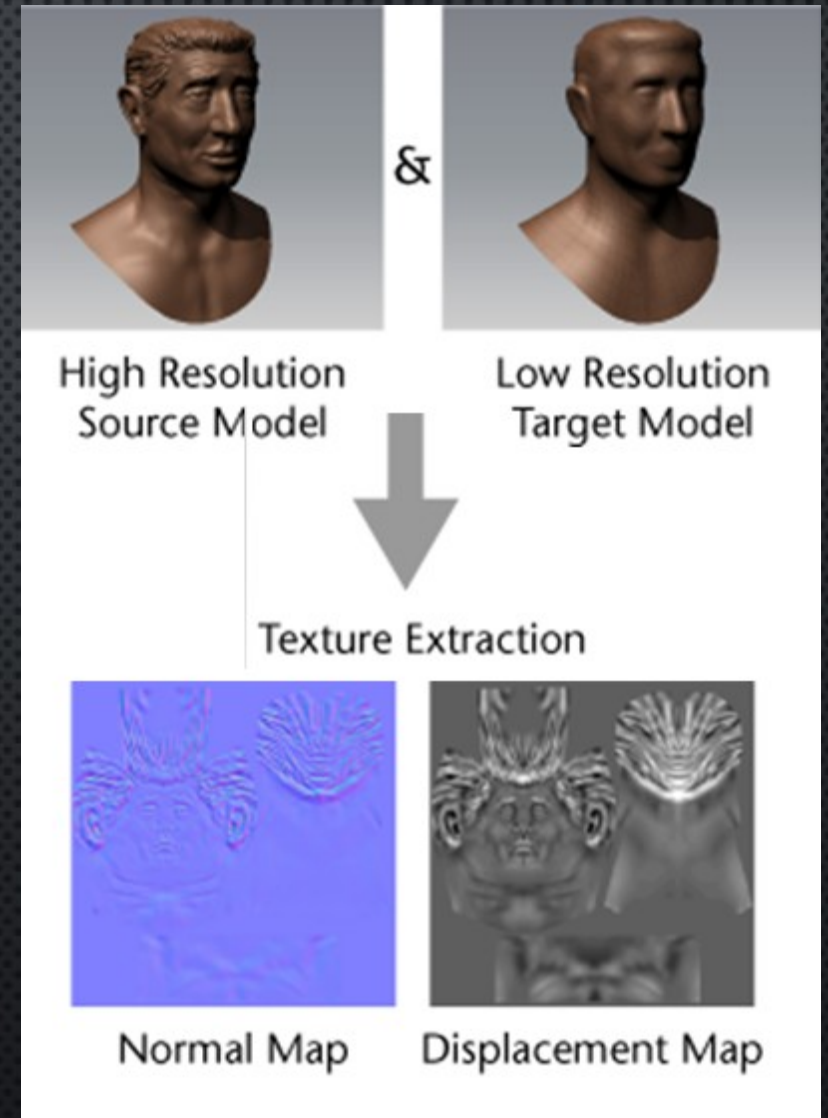


Geometrie objektu - Optimalizace

- Zjednodušení modelu pro vykreslení

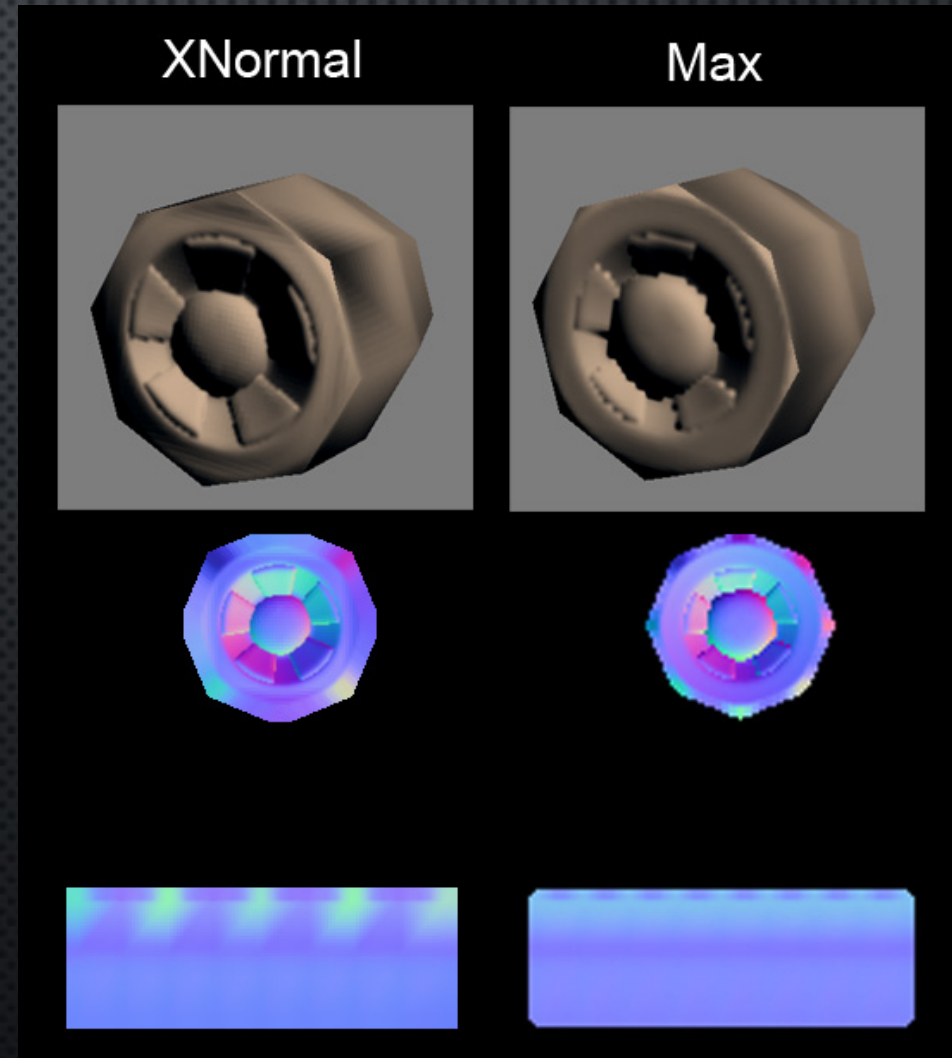
Geometrie objektu - Optimalizace

- Zjednodušení modelu pro vykreslení
 - Příprava zjednodušeného modelu
 - Vyexportování **Normal** map, **Height** map



Geometrie objektu - Optimalizace

- **Zjednodušení** modelu pro vykreslení
 - Příprava zjednodušeného modelu
 - Vyexportování **Normal** map, **Height** map
- **Nástroje**
 - xNormal – free, vypečení map
 - Simplygon – příprava lowpoly
 - MudBox, Zbrush, 3dsMax - vše

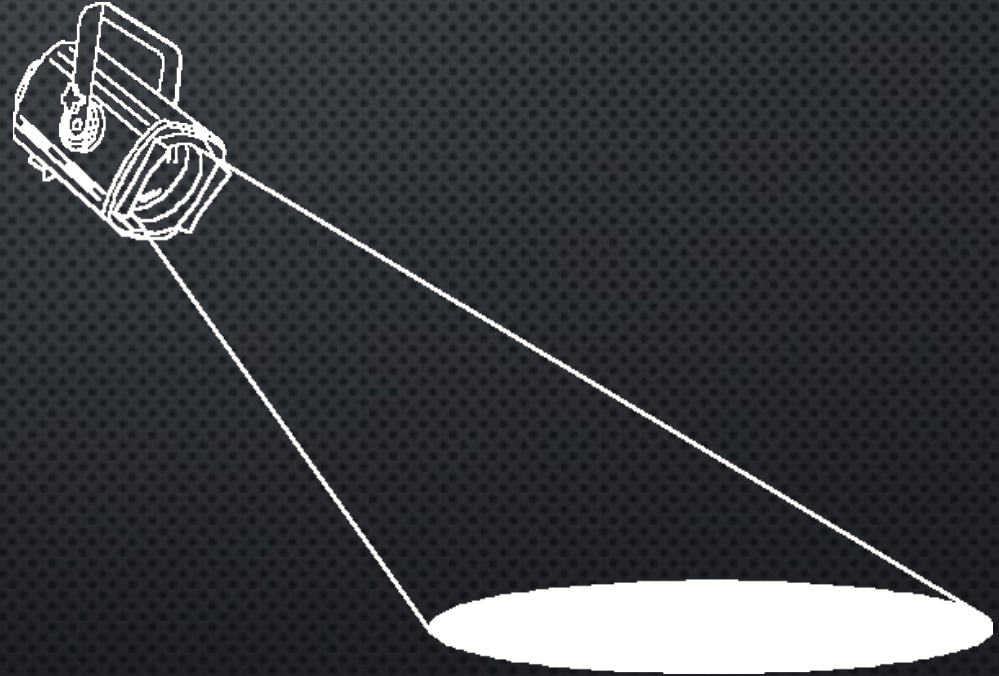


Realistická grafika

- Geometrie objektu
- Přímé nasvětlení
- Stíny
- Nepřímé nasvětlení a odrazy
- Physically based rendering
- Optimalizace
- Zpracování výsledného obrazu

Realistická grafika

- Geometrie objektu
- **Přímé nasvětlení**
- Stíny
- Nepřímé nasvětlení a odrazy
- Physically based rendering
- Optimalzace
- Zpracování výsledného obrazu



Přímé nasvětlení – Struktura povrchu

- Struktura povrchu

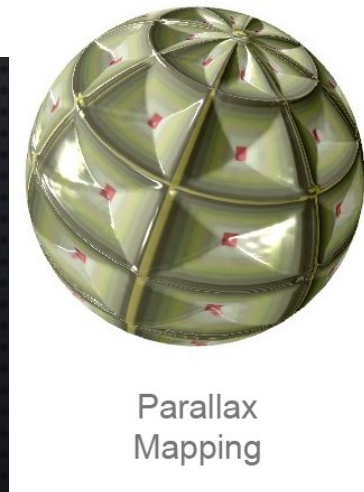
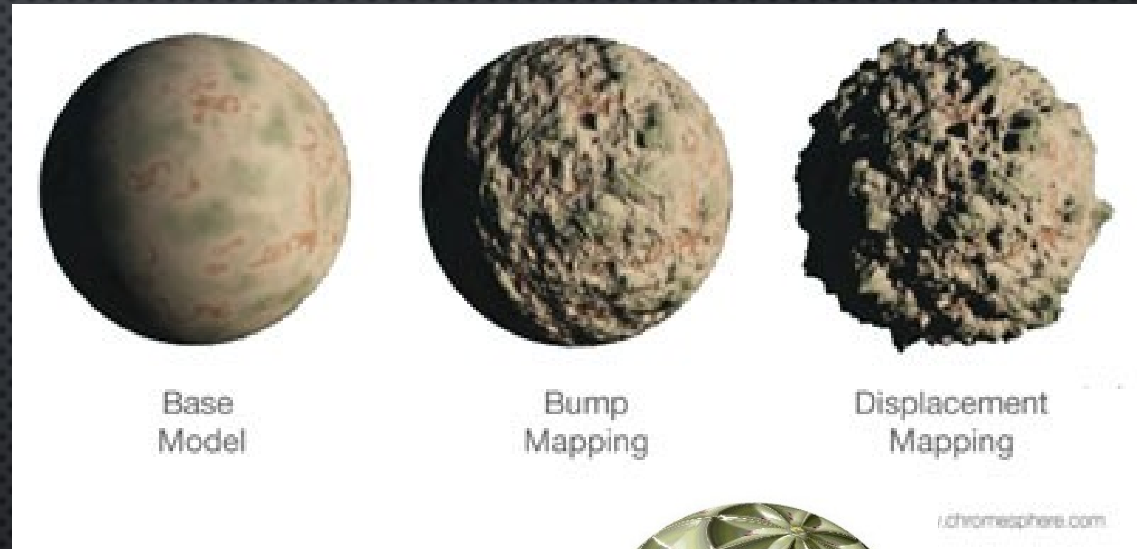
Přímé nasvětlení – Struktura povrchu

- Struktura povrchu
 - Pouze **Diffuse** mapa
 - **Normal** mapa
 - **Height** mapa + **Displacement**



Přímé nasvětlení – Struktura povrchu

- Struktura povrchu
 - Pouze **Diffuse** mapa
 - **Normal** mapa
 - **Height** mapa + **Displacement**
 - **Parallax** mapping
- jednoduchý Raytracing



Přímé nasvětlení – Hodně světel?

- Hodně světel?



Přímé nasvětlení – Hodně světel?

- Hodně světel?
- Dříve **Forward**

na každý objekt se nejprve aplikují všechna světla, pak se jde na další objekt)



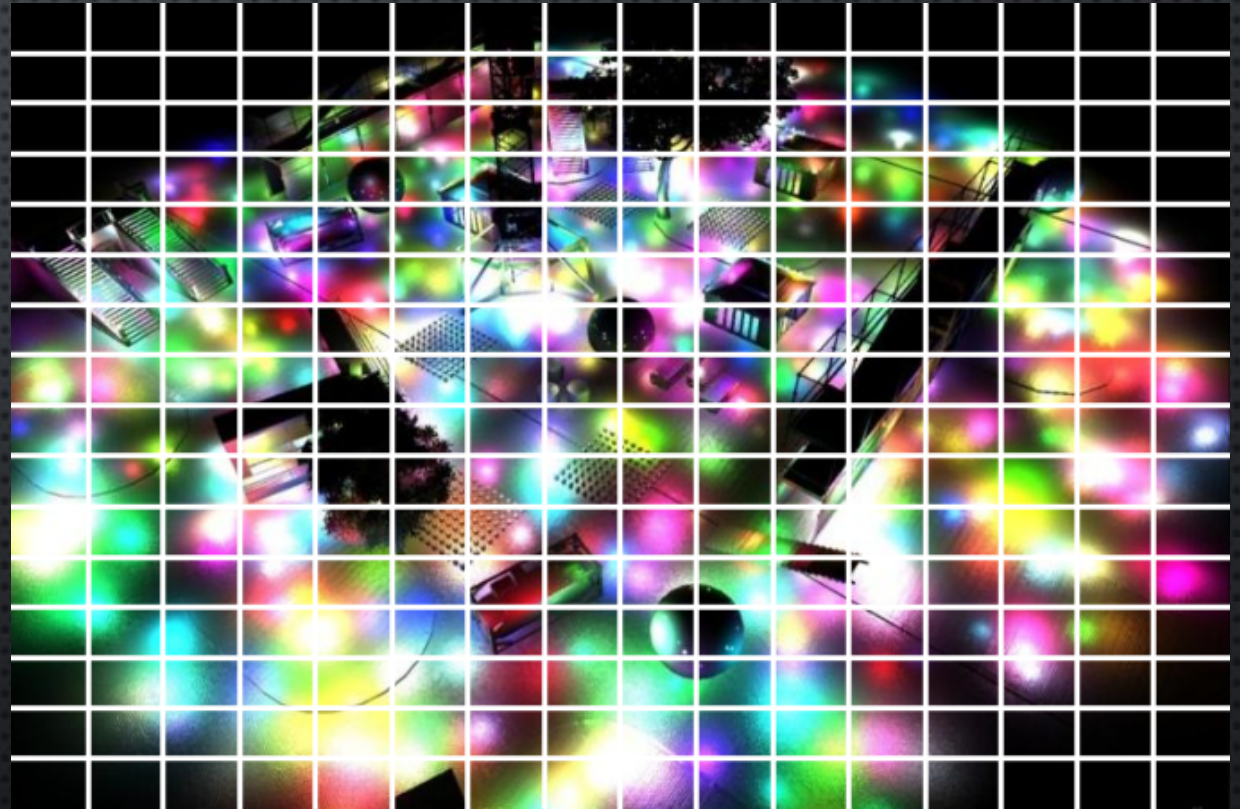
Přímé nasvětlení – Hodně světél?

- Hodně světél?
- Dříve **Forward**
na každý objekt se nejprve aplikují všechna světla, pak se jde na další objekt)
- Později **Deferred Shading**
příprava bufferů obrazovky „**GBuffers**“, poté iteračně všechna světla jen tam, kam dosáhnou)



Přímé nasvětlení – Hodně světel?

- Trendem je využití návrat k jakému si Forward renderingu
Rozdělit obrazovku a tvořit seznam světel pro každý díl
Compute shadery k dalšímu zefektivnění
Tile-based Lighting
alias **Forward plus**



Přímé nasvětlení – Deffered Shading

Přímé nasvětlení – Deffered Shading

- Buffery obrazovky
 - Albedo



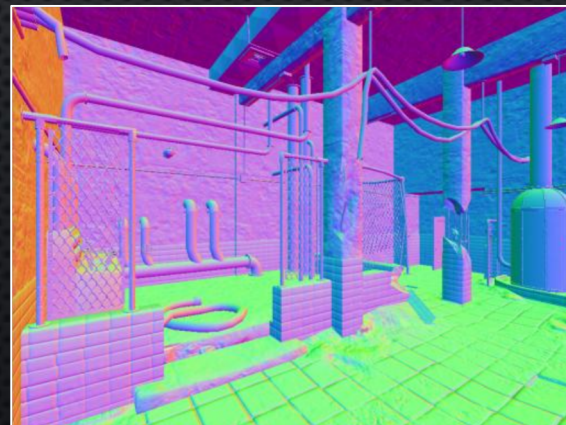
Přímé nasvětlení – Deffered Shading

- Buffery obrazovky
 - Albedo
 - Depth (hloubka scény)



Přímé nasvětlení – Deferred Shading

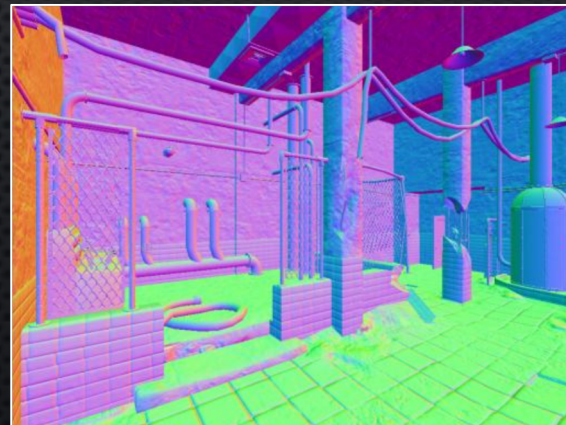
- Buffery obrazovky
 - **Albedo**
 - **Depth** (hloubka scény)
 - **Normal** (směr povrchu)



Přímé nasvětlení – Deffered Shading

- Buffery obrazovky
 - **Albedo**
 - **Depth** (hloubka scény)
 - **Normal** (směr povrchu)
 - **Specular** (velikost / síla odlesku)

PBR - **Gloss** / **Roughness**



Přímé nasvětlení – Deffered Shading

- Zpracování na závěr
 - Aplikace **Diffuse** světla



Přímé nasvětlení – Deffered Shading

- Zpracování na závěr
 - Aplikace **Diffuse** světla
 - **Specular** odlesk
/ **Reflexe**



Přímé nasvětlení – Deffered Shading

- Zpracování na závěr
 - Aplikace **Diffuse** světla
 - **Specular** odlesk
/ **Reflexe**
 - **Výsledný obraz**



Realistická grafika

- Geometrie objektu
- Přímé nasvětlení
- Stíny
- Nepřímé nasvětlení a odrazy
- Physically based rendering
- Optimalizace
- Zpracování výsledného obrazu

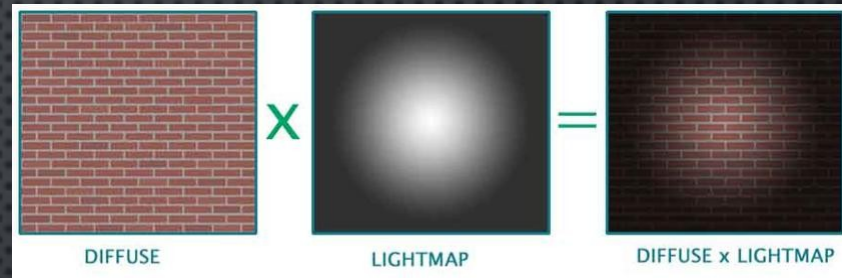
Realistická grafika

- Geometrie objektu
- Přímé nasvětlení
- **Stíny**
- Nepřímé nasvětlení a odrazy
- Physically based rendering
- Optimalizace
- Zpracování výsledného obrazu



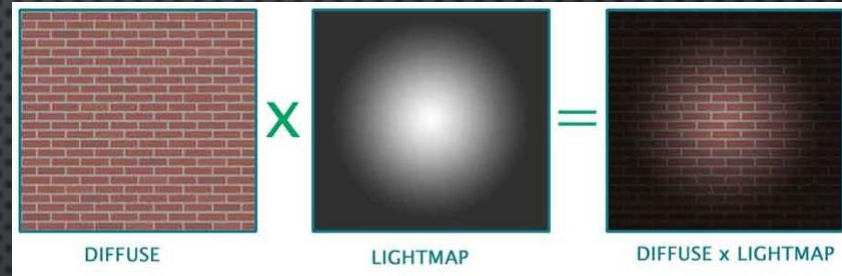
Přímé nasvětlení – Vrhání stínů

- Předpočítané **light mapy**



Přímé nasvětlení – Vrhání stínů

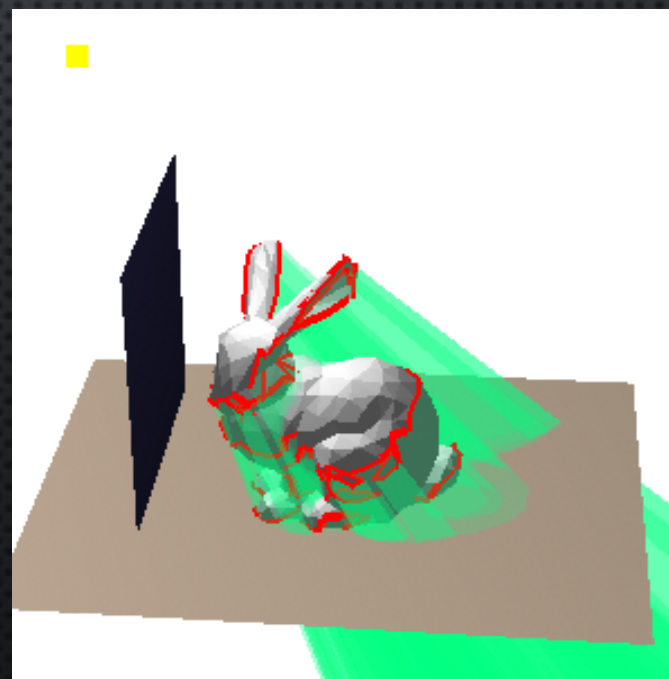
- Předpočítané **light mapy**
- Dynamické lightmapy (vitráže, dým)



Přímé nasvětlení – Vrhání stínů

- Předpočítané **light mapy**
- Dynamické lightmapy (vitráže, dým)
- **Geometrické** metody

Pomocí volume tělesa a **stencil bufferu**



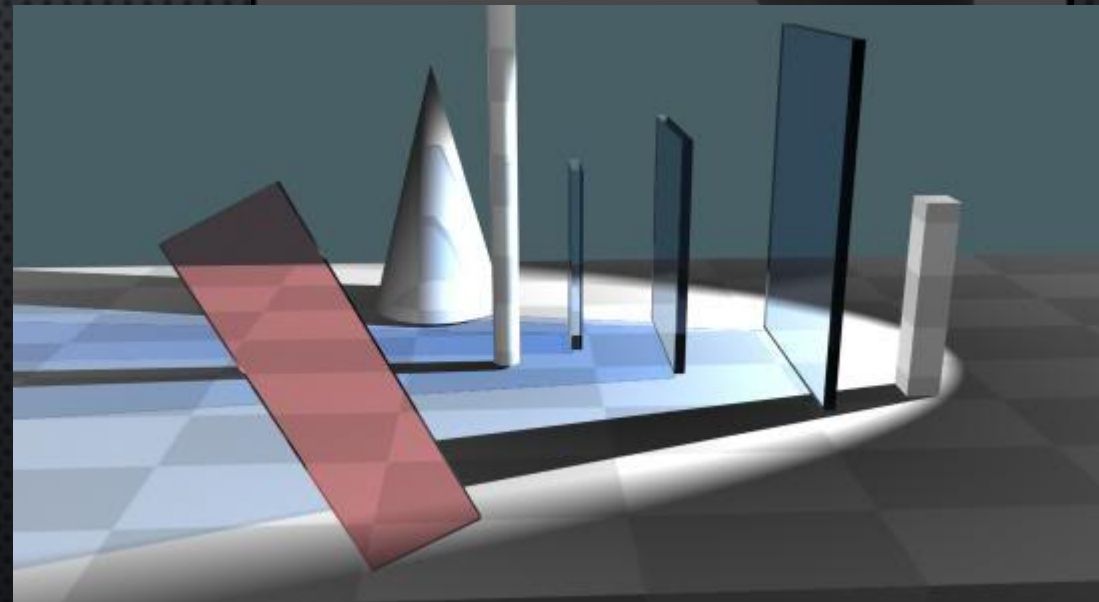
Přímé nasvětlení – Vrhání stínů

- Předpočítané **light mapy**
- Dynamické lightmapy (vitráže, dým)
- **Geometrické** metody
 - Pomocí volume tělesa a stencil bufferu
- **Shadow depth** mapy
 - V mnoha variantách dnes nejrozšířenější



Přímé nasvětlení – Vrhání stínů

- Předpočítané **light mapy**
- Dynamické lightmapy (vitráže, dým)
- **Geometrické** metody
Pomocí volume tělesa a stencil bufferu
- **Shadow depth** mapy
V mnoha variantách dnes nejrozšířenější
- **Raytracing**
Zatím vzácně (screen space, obličej, vlasy, ...)



Přímé nasvětlení – Vrhání stínů

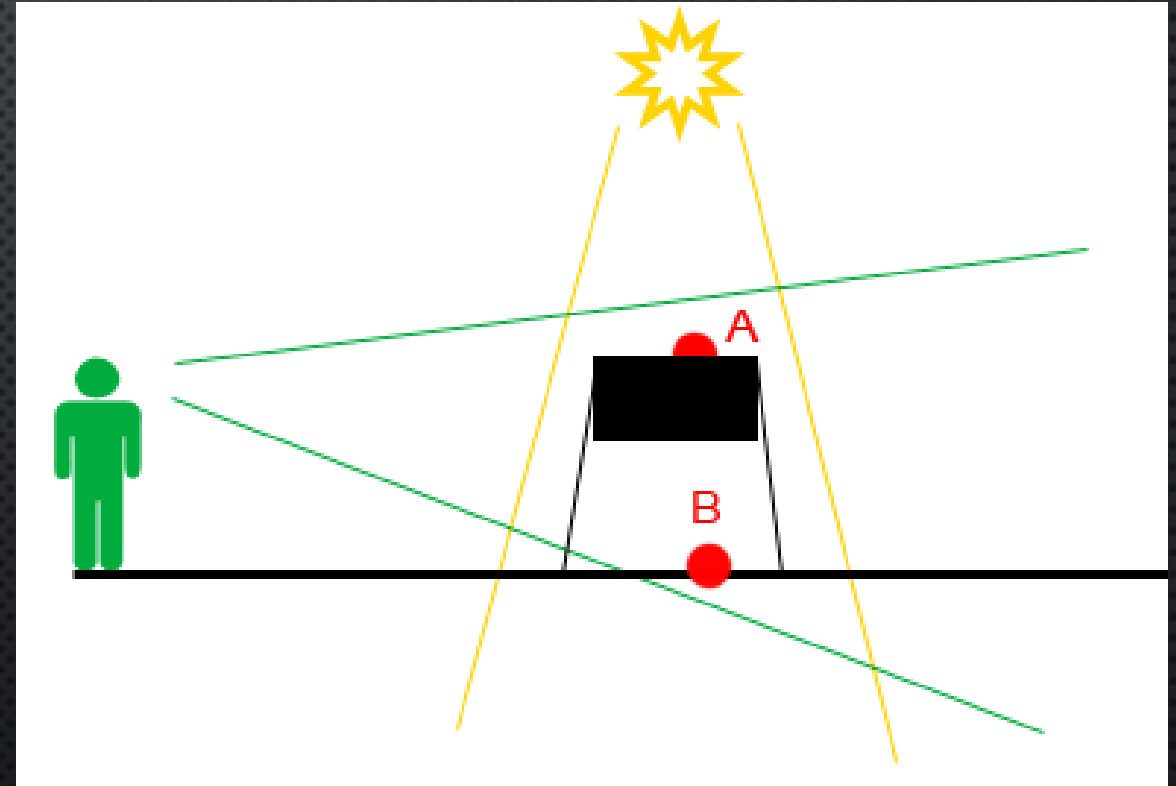
- **Shadow depth map** metody
(textura s hloubkou)

Přímé nasvětlení – Vrhání stínů

- **Shadow depth map** metody
(textura s hloubkou)
 - **CSM, PSSM, Per Object, PSM...**

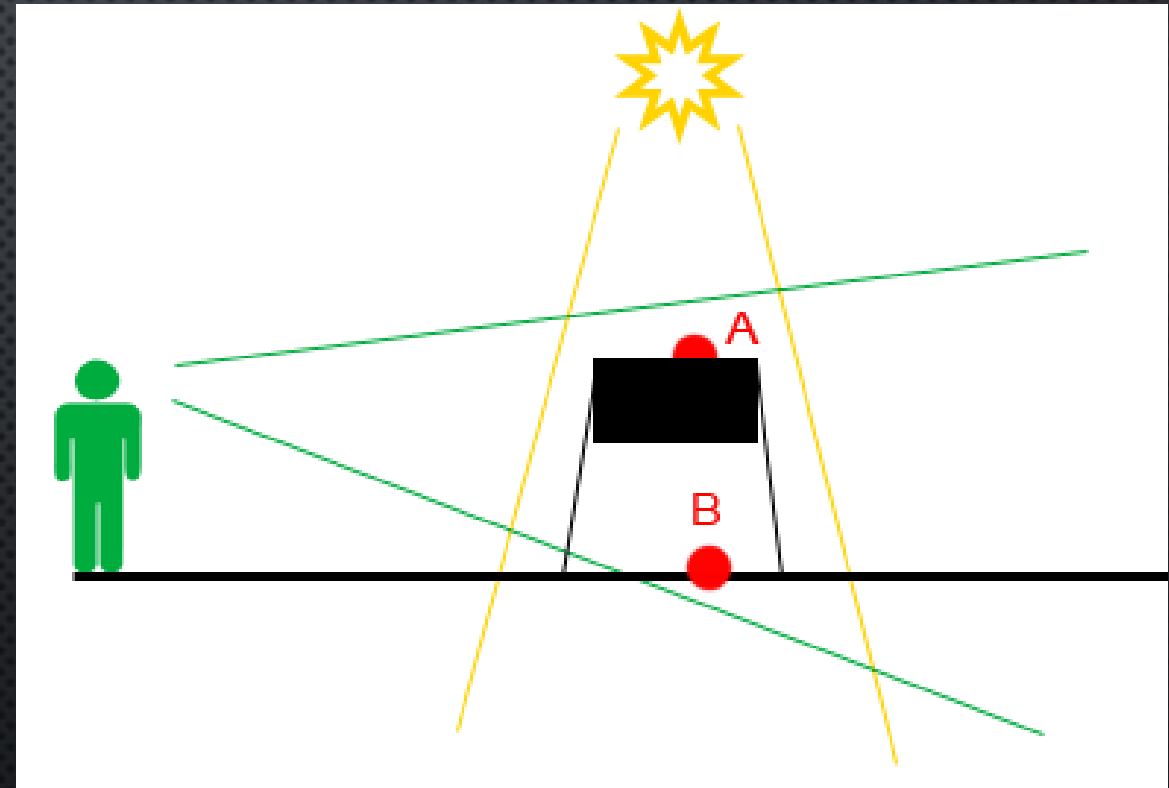
Přímé nasvětlení – Vrhání stínů

- **Shadow depth map** metody
(textura s hloubkou)
 - **CSM, PSSM, Per Object, PSM...**
 - Většina se stejným principem



Přímé nasvětlení – Vrhání stínů

- **Shadow depth map** metody
(textura s hloubkou)
 - **CSM, PSSM, Per Object, PSM...**
 - Většina se stejným principem
- Každá metoda má své výhody i nevýhody, případy použití
- V praxi se využívá **kombinace** k dosažení lepších výsledků



Přímé nasvětlení – Vrhání stínů

- Pokročilejší metody & problémy

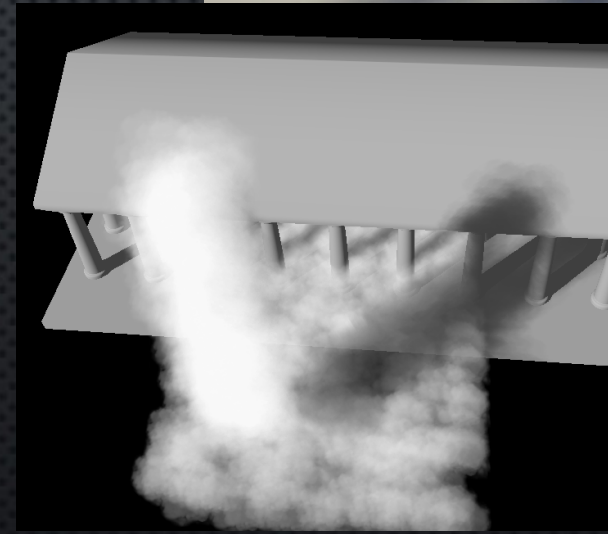
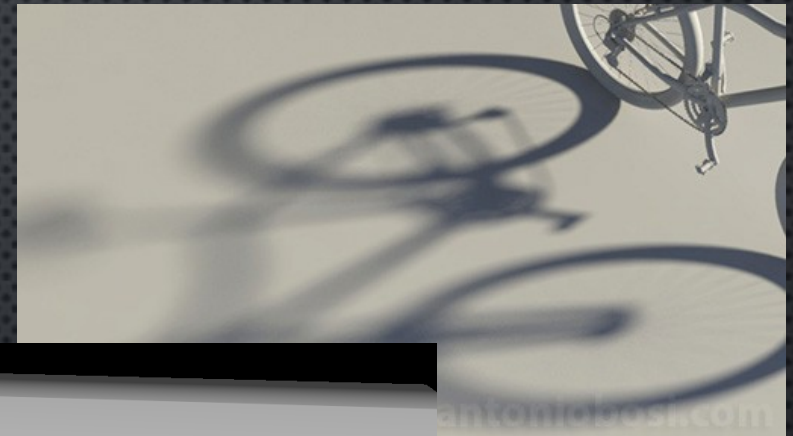
Přímé nasvětlení – Vrhání stínů

- Pokročilejší metody & problémy
 - **Area shadows** (penumbra) – soft shadows



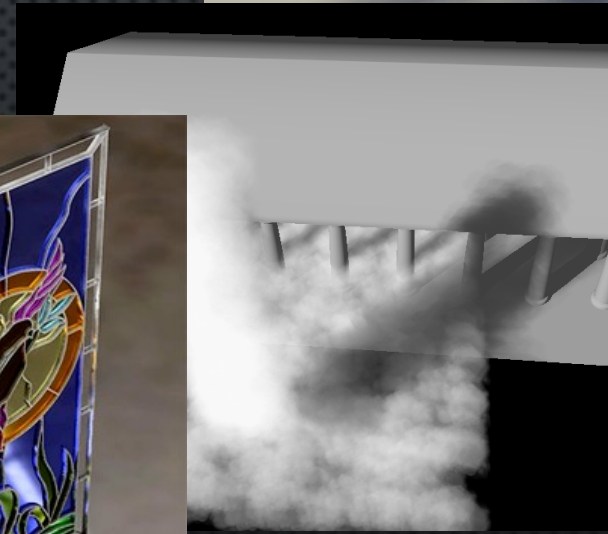
Přímé nasvětlení – Vrhání stínů

- Pokročilejší metody & problémy
 - **Area shadows** (penumbra) – soft shadows
 - **Transparentní** objekty (kouř) – depth opacity maps



Přímé nasvětlení – Vrhání stínů

- Pokročilejší metody & problémy
 - **Area shadows** (penumbra) – soft shadows
 - **Transparentní** objekty (kouř) – depth opacity maps
 - **Barevné** stíny (vitráže) – light maps, raytrace



Realistická grafika

- Geometrie objektu
- Přímé nasvětlení
- Stíny
- Nepřímé nasvětlení a odrazy
- Physically based rendering
- Optimalizace
- Zpracování výsledného obrazu

Realistická grafika

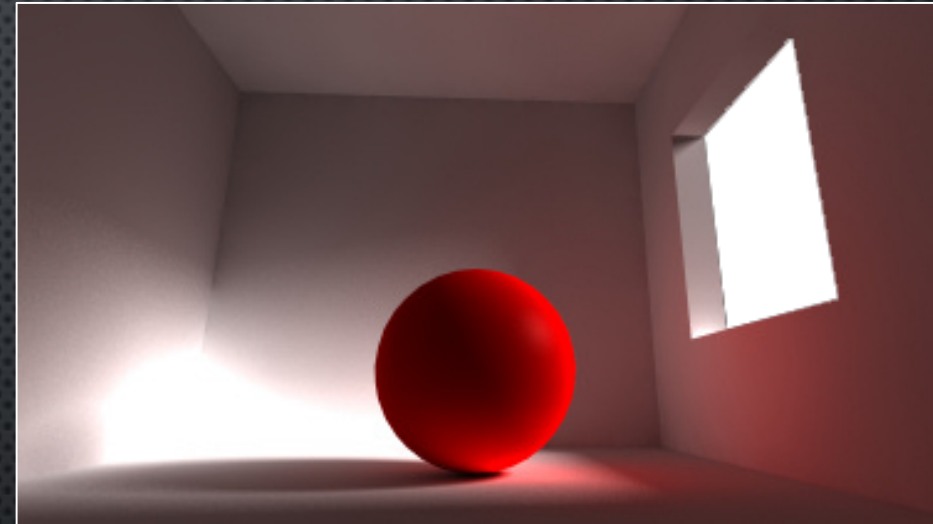
- Geometrie objektu
- Přímé nasvětlení
- Stíny
- **Nepřímé nasvětlení a odrazy**
- Physically based rendering
- Optimalizace
- Zpracování výsledného obrazu



Nepřímé nasvětlení a odrazy - Radiozita

Nepřímé nasvětlení a odrazy - Radiozita

- Každá nasvětlená plocha se stává **zdrojem světla**



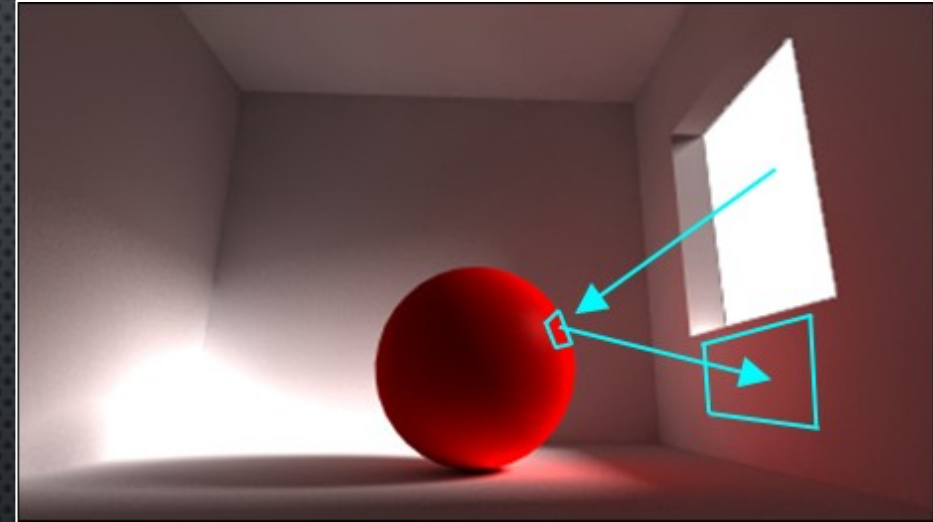
Nepřímé nasvětlení a odrazy - Radiozita

- Každá nasvětlená plocha se stává **zdrojem světla**

- Metoda **předpočítaných vztahů**

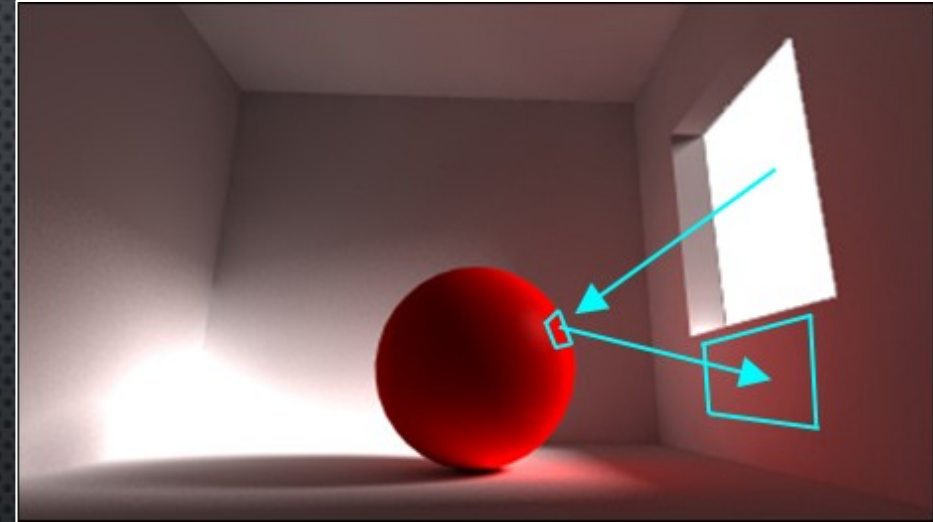
který polygon ovlivňuje který polygon z kolika procent)

Jakmile se plocha nasvítí, propaguje energii dalším



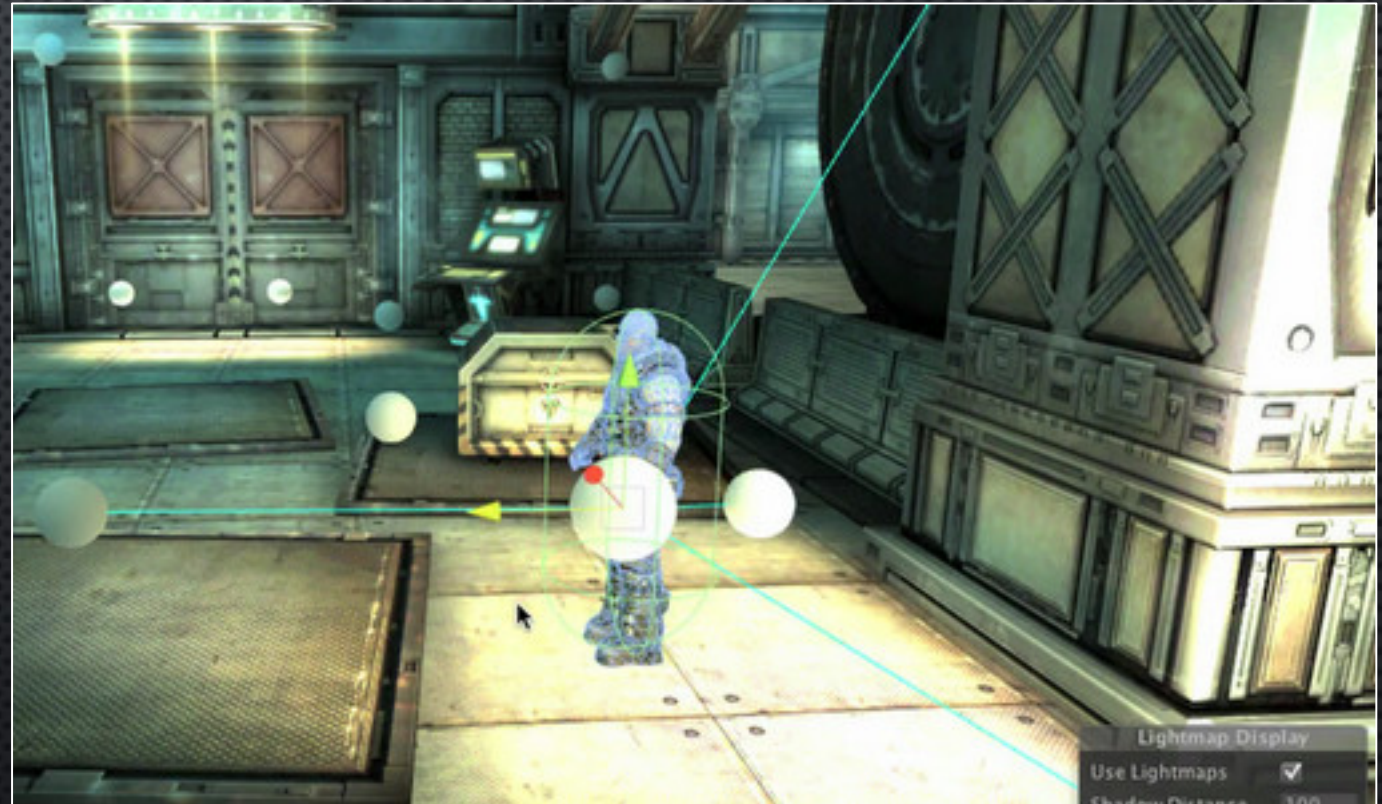
Nepřímé nasvětlení a odrazy - Radiozita

- Každá nasvětlená plocha se stává **zdrojem světla**
- Metoda **předpočítaných vztahů**
který polygon ovlivňuje který polygon
z kolika procent)
Jakmile se plocha nasvítí, propaguje
energii dalším
- V komerčních hrách se používá např.
Enlighten. Své řešení ale obsahuje
většina enginů včetně **Unity**.



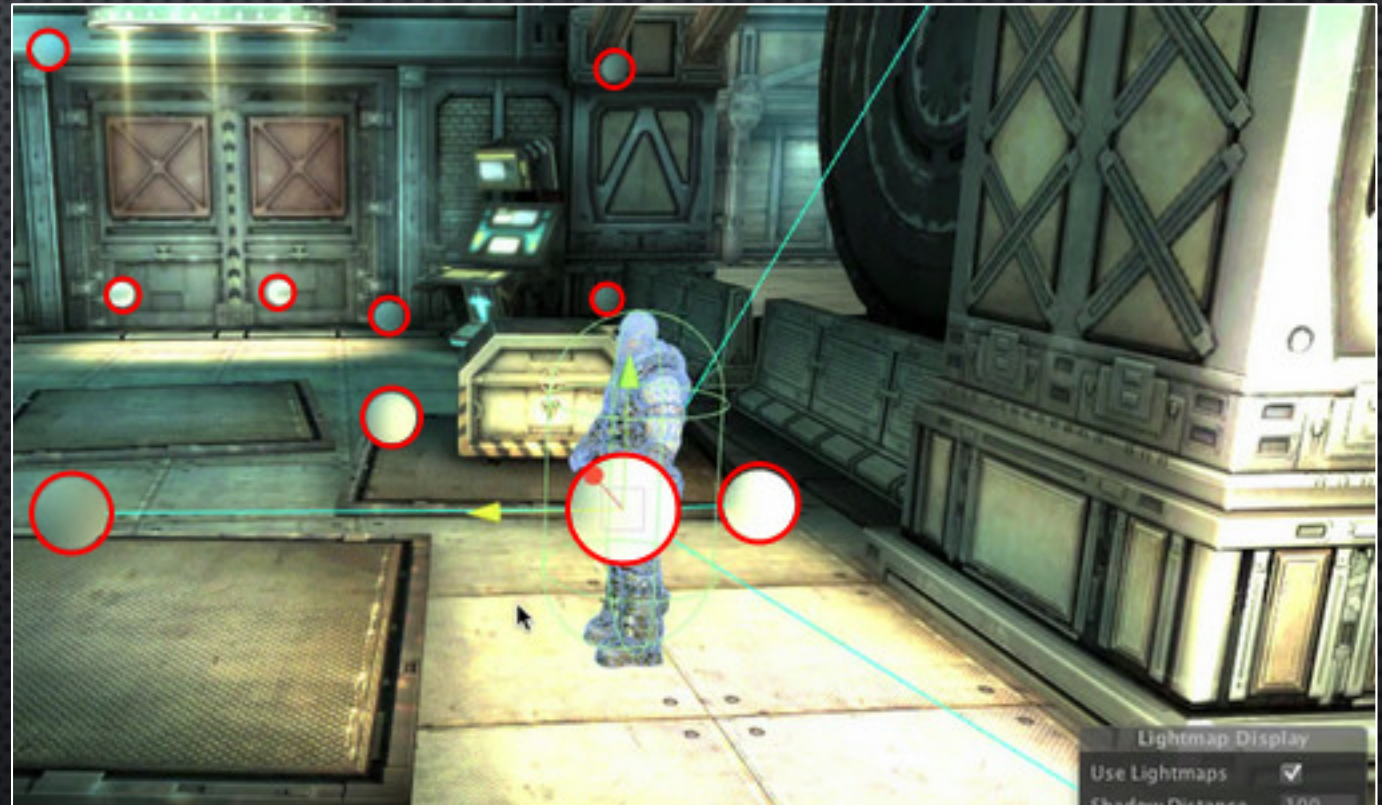
Nepřímé nasvětlení a odrazy - Radiozita

- **Dynamické objekty neodrážejí!**



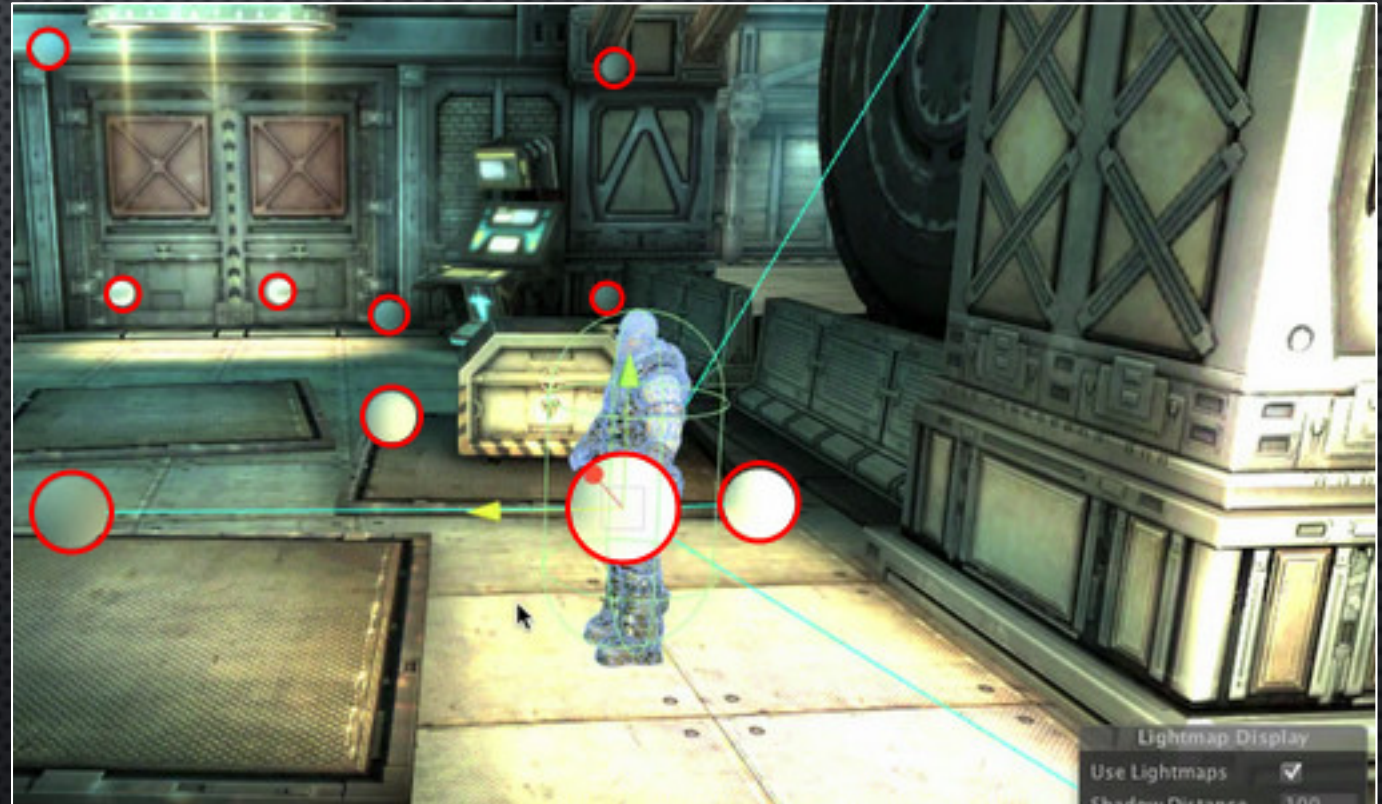
Nepřímé nasvětlení a odrazy - Radiozita

- **Dynamické objekty neodrážejí!**
- Umístí se „**Light probes**“



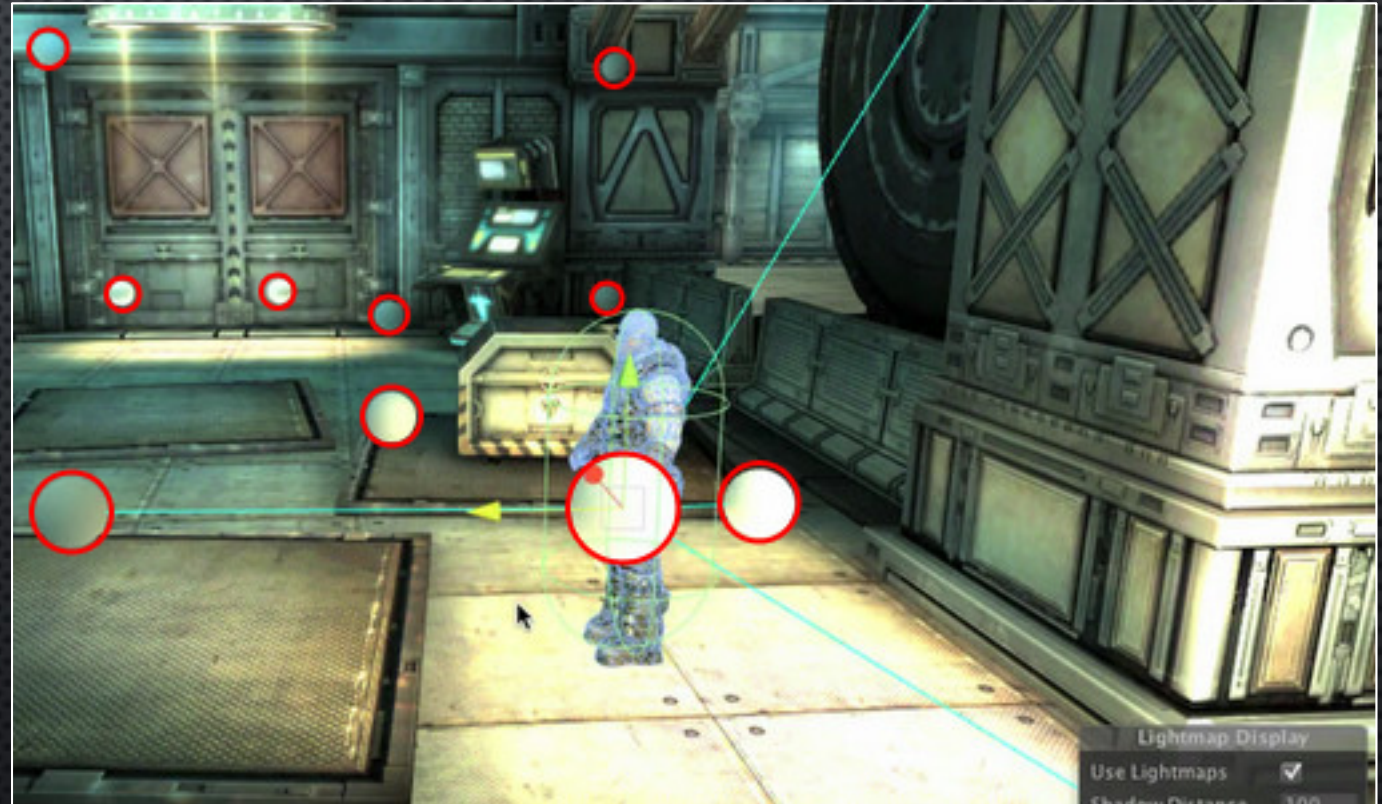
Nepřímé nasvětlení a odrazy - Radiozita

- **Dynamické** objekty **neodrážejí!**
- Umístí se „**Light probes**“
- V nich se **vypočítá** odkud jde kolik **světla** (low freq)



Nepřímé nasvětlení a odrazy - Radiozita

- **Dynamické** objekty **neodrážejí!**
- Umístí se „**Light probes**“
- V nich se **vypočítá** odkud jde kolik **světla** (low freq)
- Ukládání do **Spherical harmonics**



Nepřímé nasvětlení a odrazy - Radiozita

- **Dynamické objekty neodrážejí!**
- Umístí se „**Light probes**“
- V nich se **vypočítá** odkud jde kolik **světla** (low freq)
- Ukládání do **Spherical harmonics**
- Dynamické objekty se nasvětlují **podle nejbližších práb.**



Nepřímé nasvětlení a odrazy - Radiozita

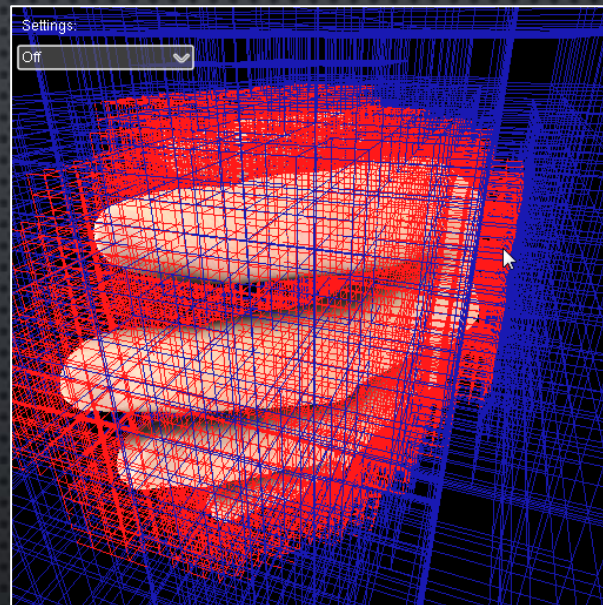
- Metoda 3D Voxelizace

Nepřímé nasvětlení a odrazy - Radiozita

- Metoda **3D Voxelizace**

Převod scény do 3D mřížky

Každý voxel šíří energii do
sousedních, příp. ji odrazí,
rozptýlí



Nepřímé nasvětlení a odrazy - Radiozita

- Metoda **3D Voxelizace**

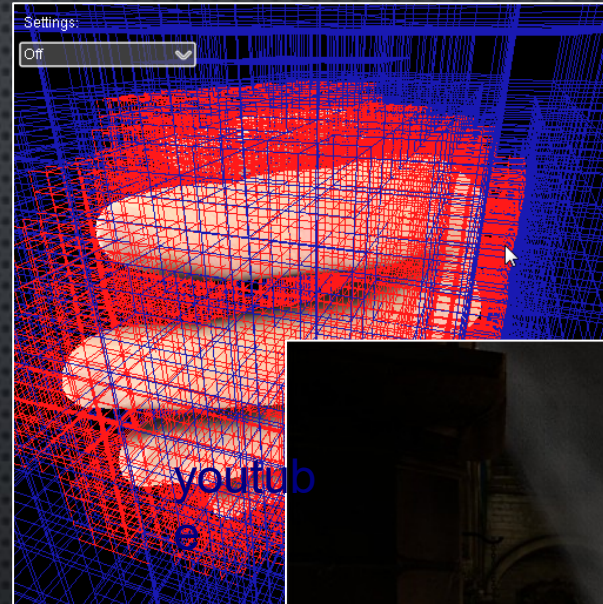
Převod scény do 3D mřížky

Každý voxel šíří energii do sousedních, příp. ji odrazí, rozptýlí

- V komerční sféře

vytvořeno od nVidia pro UDK

„**Voxel Cone Tracing**“



Realistická grafika

- Geometrie objektu
- Přímé nasvětlení
- Stíny
- Nepřímé nasvětlení a odrazy
- Physically based rendering
- Optimalizace
- Zpracování výsledného obrazu

Realistická grafika

- Geometrie objektu
- Přímé nasvětlení
- Stíny
- Nepřímé nasvětlení a odrazy
- **Physically based rendering**
- Optimalizace
- Zpracování výsledného obrazu

$$\Phi_e = \frac{L}{4\pi r^2} S$$

$$U_{ef} = \frac{U_m}{\sqrt{2}} \quad E = \hbar\omega \quad U = W_{AB} = |E_{PA} - E_{PB}| = |V_A - V_B| \quad X_L = \frac{U_m}{I_m} = \omega L = 2\pi f L \quad F_g = \frac{m_1 m_2}{r^2} g$$

$$k = \frac{1}{4\pi \epsilon_0 \epsilon_r} \frac{q_1 q_2}{r^2} \quad T = \frac{4 n_1 n_2}{(n_2 + n_1)^2} \quad R_m = \frac{c}{T} k = \pm \sqrt{\frac{2m}{\hbar^2} (E - V)}$$

$$k = \frac{p}{\hbar} \quad v = \frac{\omega}{k} \quad \varphi = \frac{E - E_0}{\hbar} = k \frac{\hbar}{m} \varphi \quad m = N \cdot m_0 = \frac{M_m}{N_A} \quad E = \frac{E_c}{a} \int_{-a/2}^{+a/2} \sin(\omega t + \phi) dy$$

$$K = \frac{p^2}{2m} \quad m_0 = \frac{M_m}{N_A} = \frac{M_r \cdot 10^{-3}}{N_A} \quad l_t = l_0(1 + d \Delta t) \quad I = \frac{U_e}{R + R_i} \quad \omega = 2\pi f$$

$$\lambda = \frac{h}{\sqrt{2eUm_e}} \quad R = \rho \frac{l}{S} \quad E = mc^2 \quad \sin \alpha = \frac{v_1}{v_2} = \frac{nv_2}{nv_1} \quad v = \frac{1}{\sqrt{\epsilon \cdot \mu}}$$

$$\psi(x) = \sqrt{2/L} \sin \frac{n\pi x}{L} \quad E = \frac{1}{2} \hbar |k|/m \quad \beta = \frac{\Delta I c}{\Delta t} \quad \phi_e = \frac{\Delta E}{\Delta t} \quad \frac{m_1}{x} + \frac{m_2}{x'} = \frac{m_2 - m_1}{r}$$

$$\oint \vec{B} \cdot d\vec{l} = \mu_0 \iint_S \vec{J} \cdot d\vec{S} \quad \vec{S} = \frac{1}{\mu_0} (\vec{E} \times \vec{B}) \quad E_c = \frac{\hbar^2}{8m[L^2 \cdot \hbar^2]} \quad \iint_S \vec{J} \cdot d\vec{S} = Q^*$$

$$\sqrt{\frac{3kT N_A}{M_m}} = \sqrt{\frac{3R_m T}{M_r \cdot 10^{-3}}} \quad E = \frac{\hbar^2 k^2}{2m} \quad 1 \text{ pc} = \frac{1 \text{ AU}}{r} \quad S = R = \frac{U}{I} \quad W_2 = U_e I t \quad F_n = \int \frac{F_n}{R}$$

$$\lambda = \frac{h n_2}{T} \quad F_h = S h \rho g \quad f_0 = \frac{1}{2\pi \sqrt{LC}} \quad M = \int F \cos \alpha$$

Physically based rendering - Trendy

Physically based rendering - Trendy

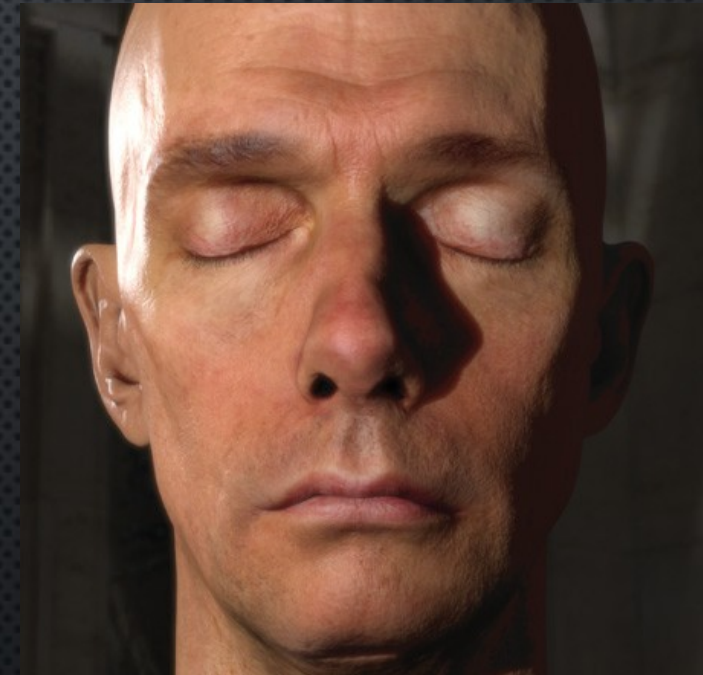
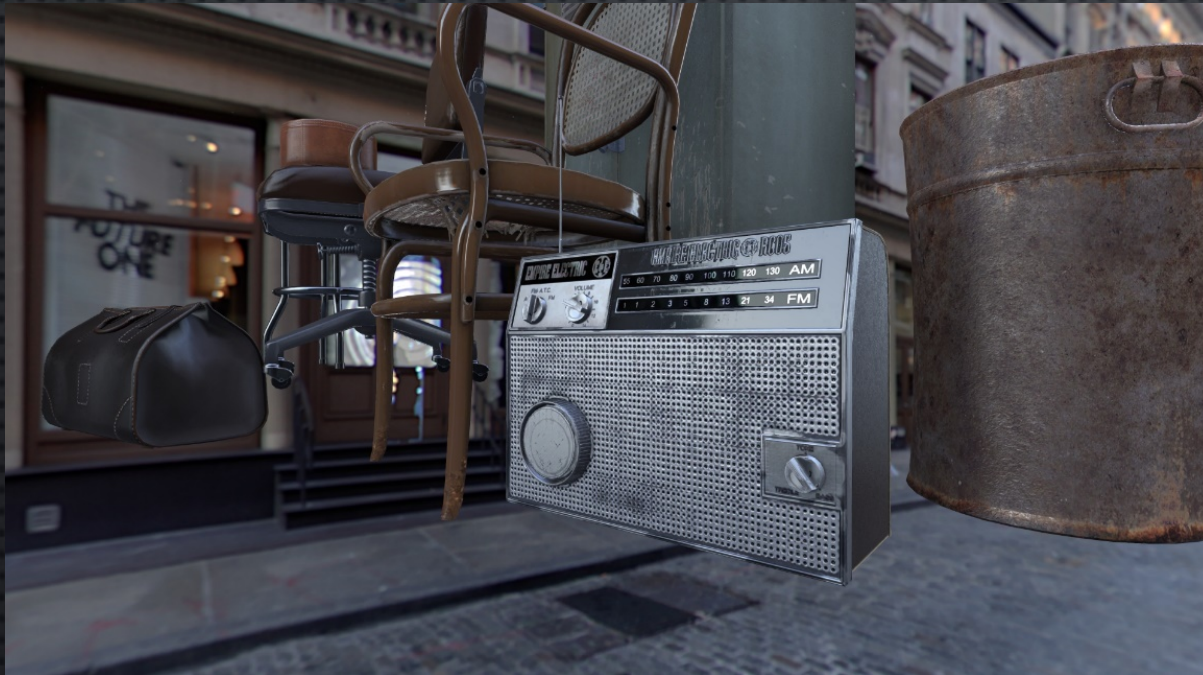
- Trendem je co možná nejvíce **informací** pro render **získat** na základě alespoň **zjednodušených fyzikálních vlastností a jevů**.

Physically based rendering - Trendy

- Trendem je co možná nejvíce **informací** pro render **získat** na základě alespoň **zjednodušených fyzikálních vlastností a jevů**.
- Grafik vytvářející jednotlivé 3D modely čím dál **méně ovlivňuje** finální vzhled hry **v konkrétních situacích**. Výsledný **vzhled určuje engine** a grafik nastavující **světelné podmínky, vlastnosti materiálu a postprocesy**.

Physically based rendering - Materiály

- Komplexní shadery na každém objektu



Physically based rendering - Materiály

- Komplexní shadery na každém objektu
 - Potřeba **konzistence** při různém nasvětlení



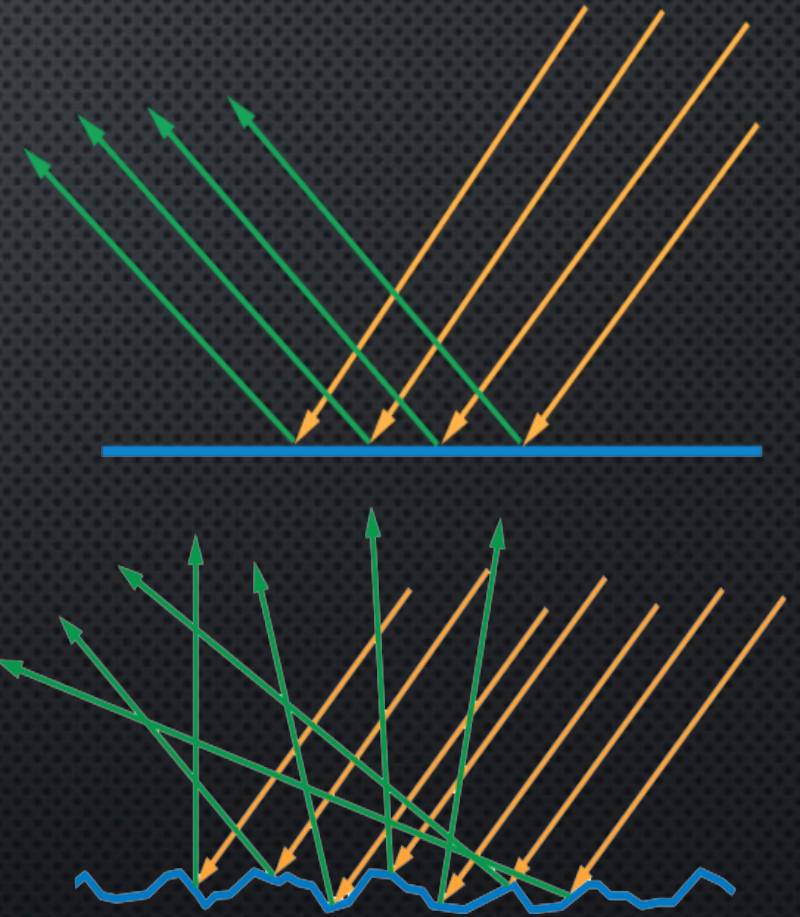
Physically based rendering - Microfacets

Physically based rendering - Microfacets

- Teorie **Microfacets** - mikro-plošky (Naty Hoffman)

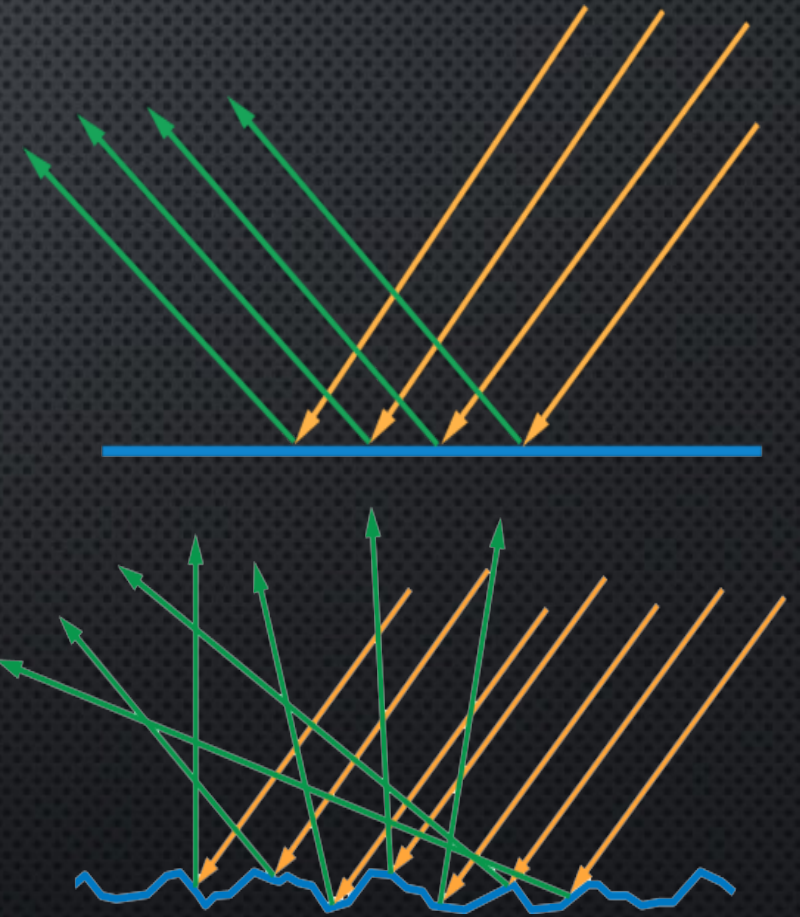
Physically based rendering - Microfacets

- Teorie **Microfacets** - mikro-plošky (Naty Hoffman)
 - Drobné plošky odrážející světlo



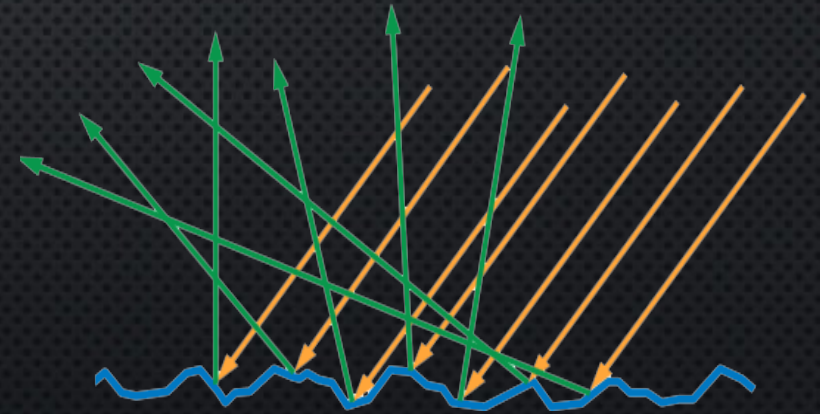
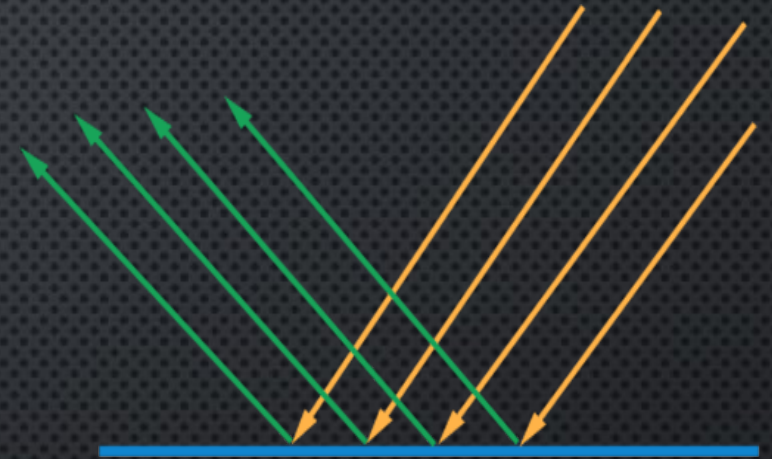
Physically based rendering - Microfacets

- Teorie **Microfacets** - mikro-plošky (Naty Hoffman)
 - Drobné plošky odrážející světlo
 - **Statistické** řešení



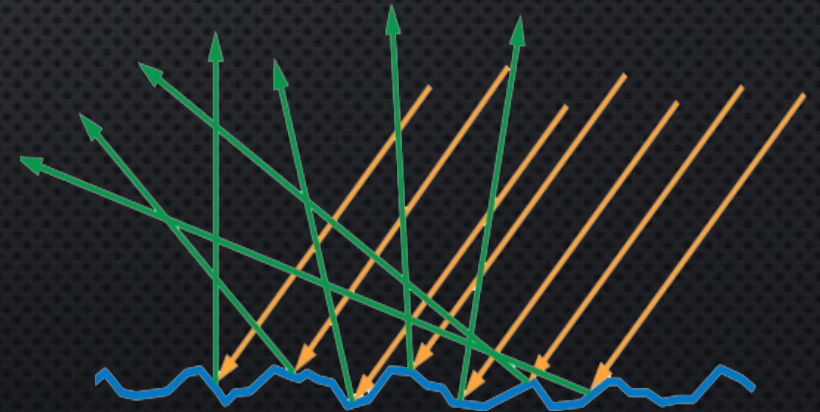
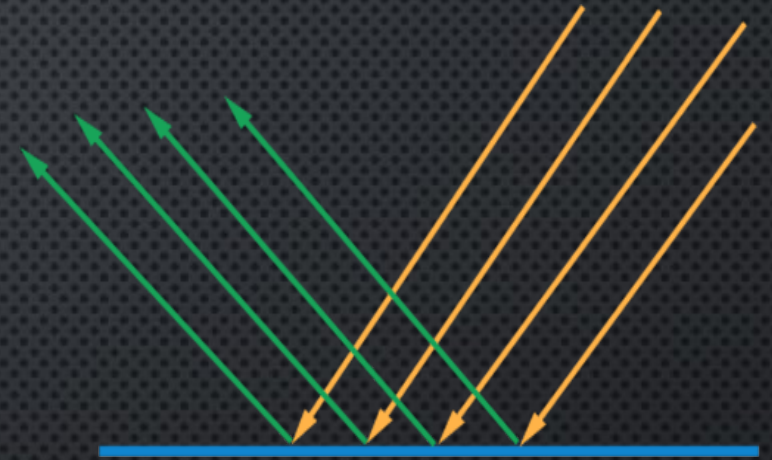
Physically based rendering - Microfacets

- Teorie **Microfacets** - mikro-plošky (Naty Hoffman)
 - Drobné plošky odrážející světlo
 - **Statistické** řešení
 - Určuje lesk / mat povrchu
 - „**Glossiness / Roughness**“



Physically based rendering - Microfacets

- Teorie **Microfacets** - mikro-plošky (Naty Hoffman)
 - Drobné plošky odrážející světlo
 - **Statistické** řešení
 - Určuje lesk / mat povrchu
 - „**Glossiness / Roughness**“
 - Zachování energie

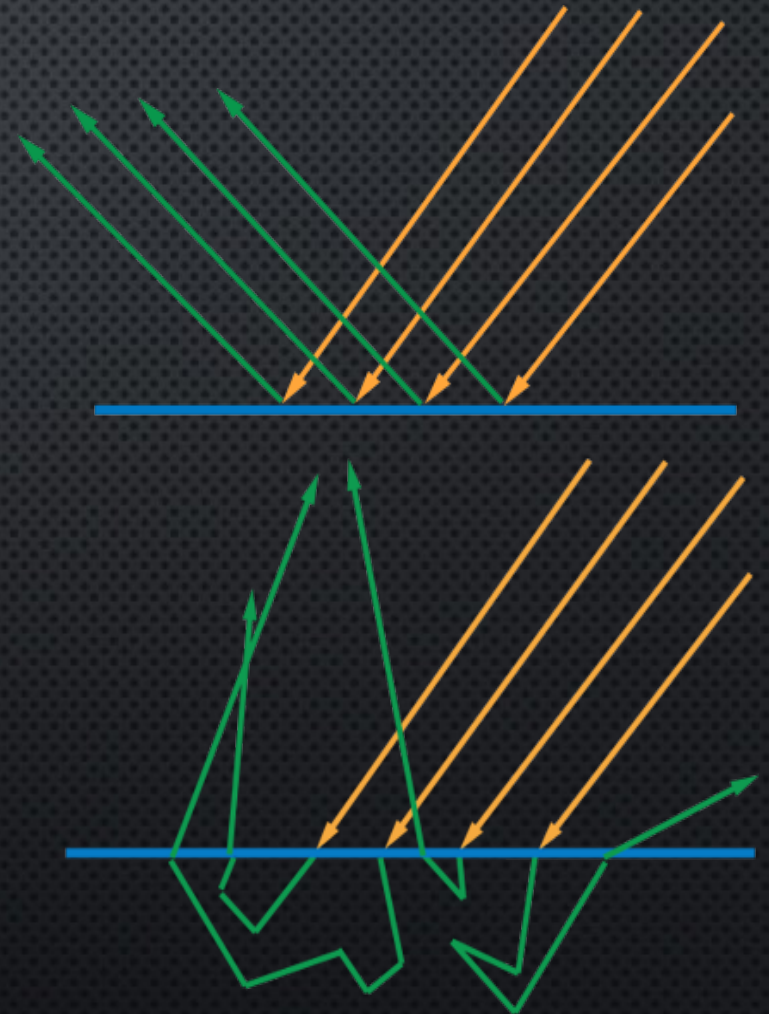


Physically based rendering - Microfacets

- Difúzní vs Reflexní barva (kovy)

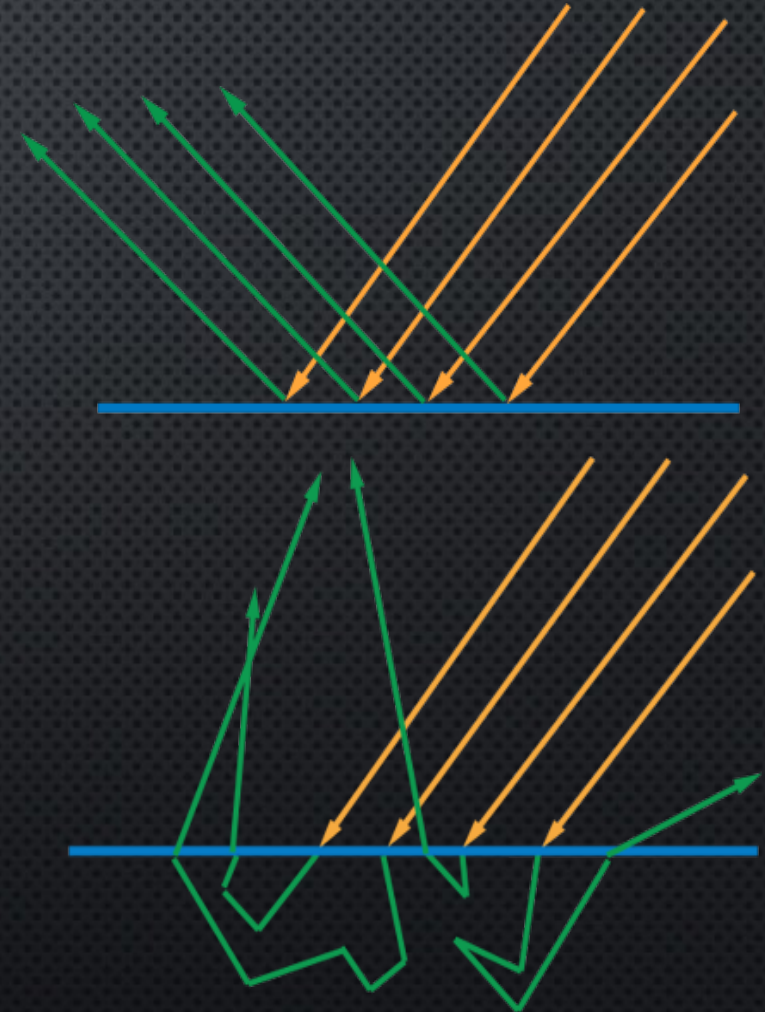
Physically based rendering - Microfacets

- **Difúzní** vs **Reflexní** barva (kovy)
 - U téměř všech materiálů - **krom kovů** - vstupuje světlo pod povrch a „obarvuje se“



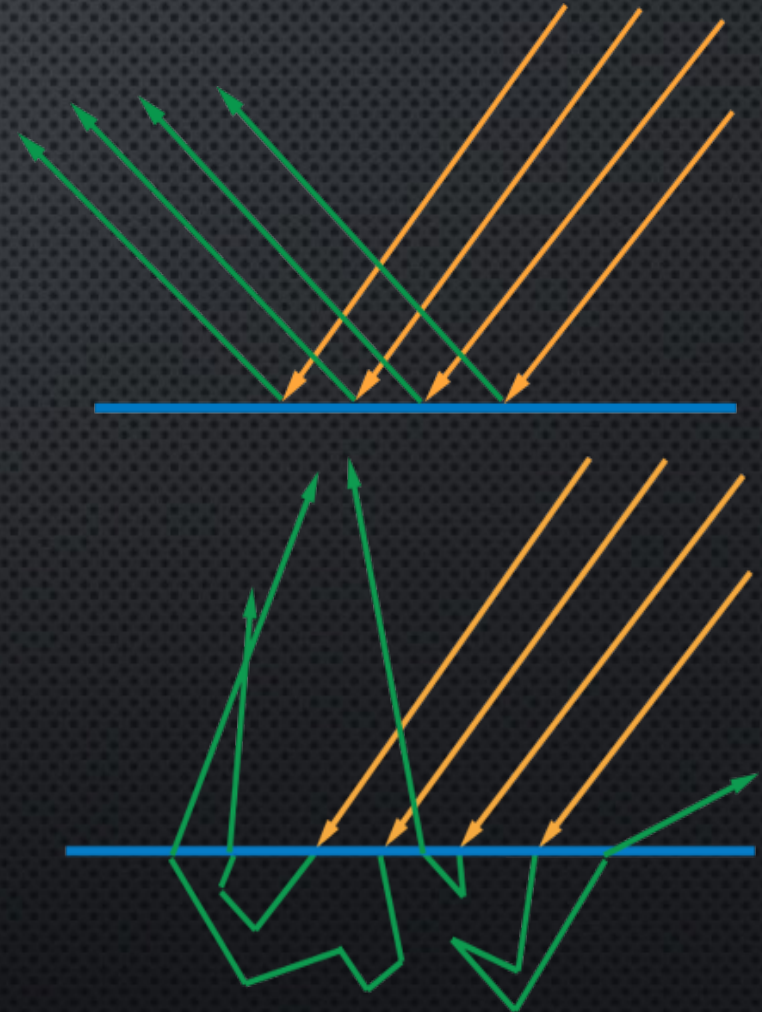
Physically based rendering - Microfacets

- Difúzní vs Reflexní barva (kovy)
 - U téměř všech materiálů - krom kovů - vstupuje světlo pod povrch a „obarvuje se“
 - „Metalness“



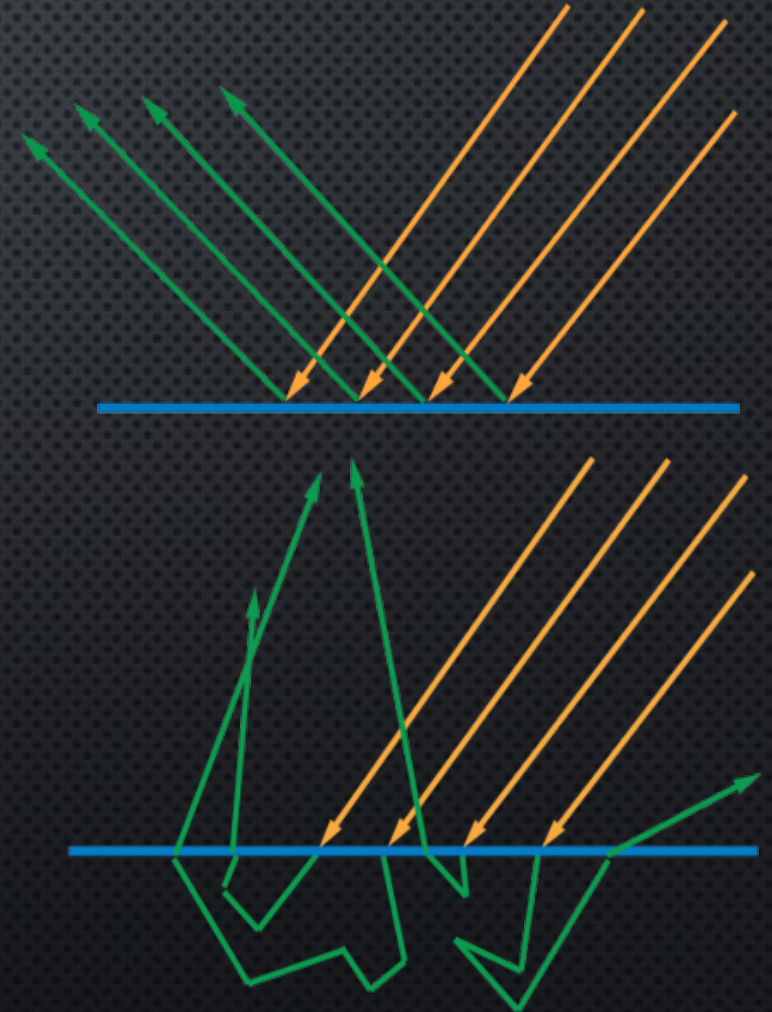
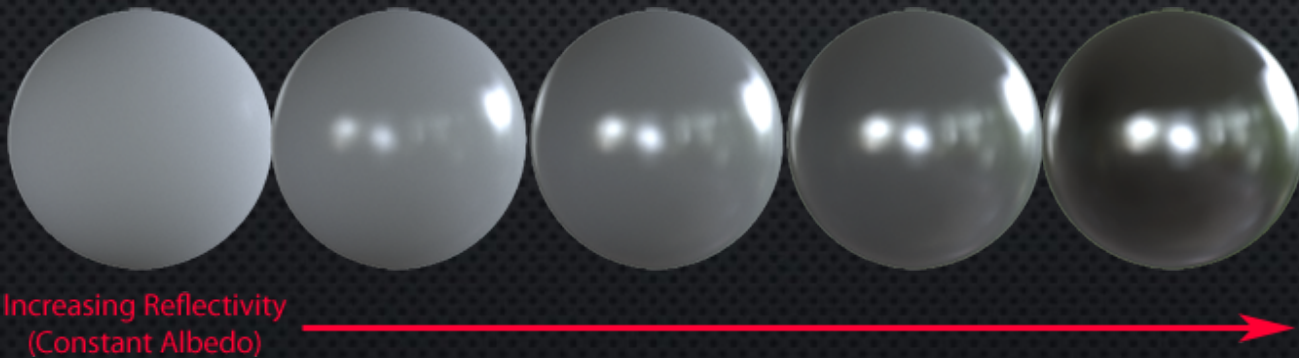
Physically based rendering - Microfacets

- **Difúzní vs Reflexní** barva (kovy)
 - U téměř všech materiálů - **krom kovů** - vstupuje světlo pod povrch a „obarvuje se“
 - „**Metalness**“
 - Vystupující světlo je pak více všesměrové



Physically based rendering - Microfacets

- **Difúzní** vs **Reflexní** barva (kovy)
 - U téměř všech materiálů - **krom kovů** - vstupuje světlo pod povrch a „obarvuje se“
 - „**Metalness**“
 - Vystupující světlo je pak více všesměrové
 - Zachování energie

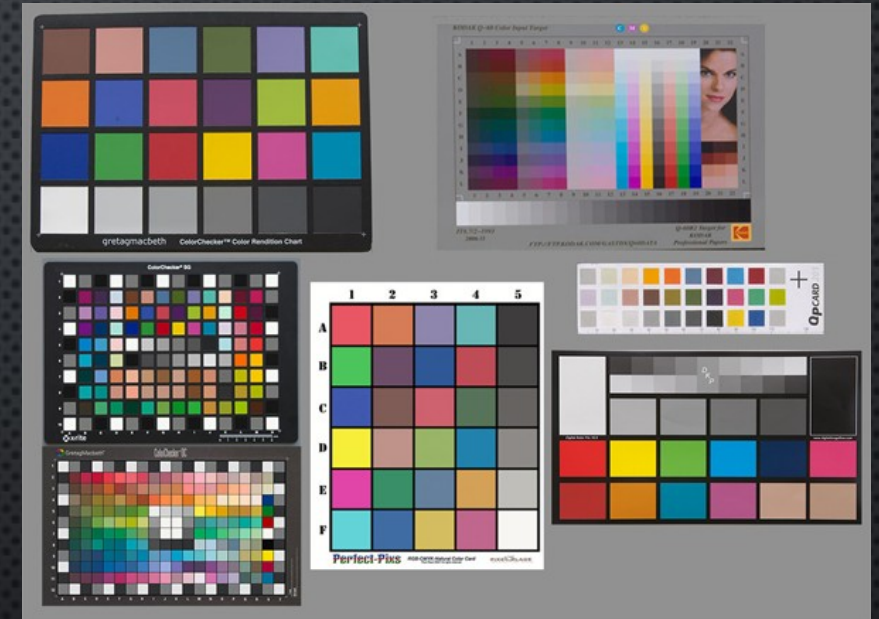


Physically based rendering - Textury

- Kalibrování textur

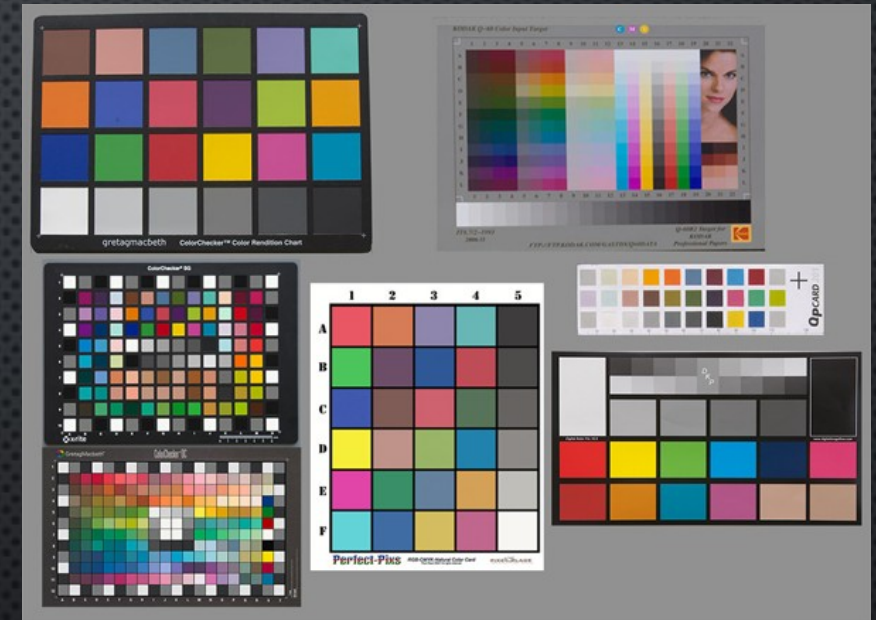
Physically based rendering - Textury

- Kalibrování textur pomocí kalibrační tabulky



Physically based rendering - Textury

- Kalibrování textur pomocí kalibrační tabulky
- Konzistentní při různých podmínkách focení



Physically based rendering - Textury

- Online databáze

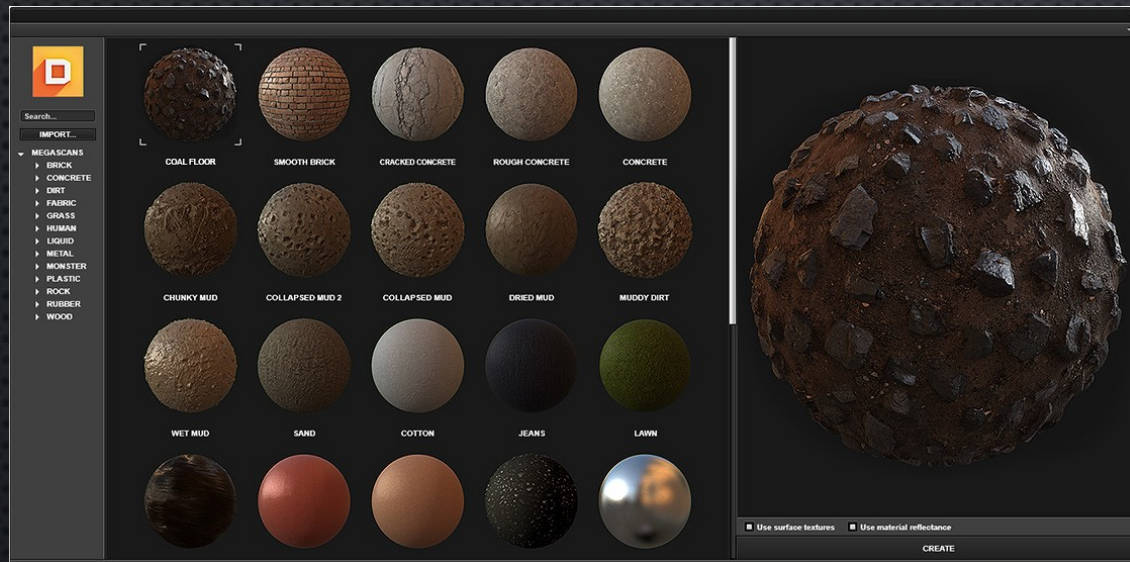
Physically based rendering - Textury

- Online databáze
 - Fyzikálně korektní materiály



Physically based rendering - Textury

- Online databáze
 - Fyzikálně korektní materiály
 - Nasvětlování, HDR, optimalizace pro mobily



Realistická grafika

- Geometrie objektu
- Přímé nasvětlení
- Stíny
- Nepřímé nasvětlení a odrazy
- Physically based rendering
- Optimalizace
- Zpracování výsledného obrazu

Realistická grafika

- Geometrie objektu
- Přímé nasvětlení
- Stíny
- Nepřímé nasvětlení a odrazy
- Physically based rendering
- **Optimalizace**
- Zpracování výsledného obrazu



Optimalizace - Zjednodušení

Optimalizace - Zjednodušení

Urychlení vykreslování

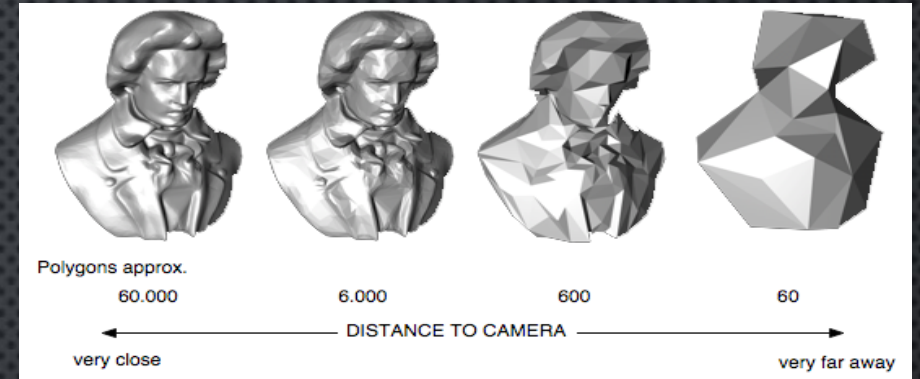
- **LOD** – geometrie
- **MipMap** – textura



Optimalizace - Zjednodušení

Urychlení vykreslování

- **LOD** – geometrie
- **MipMap** – textura
- **Shader**



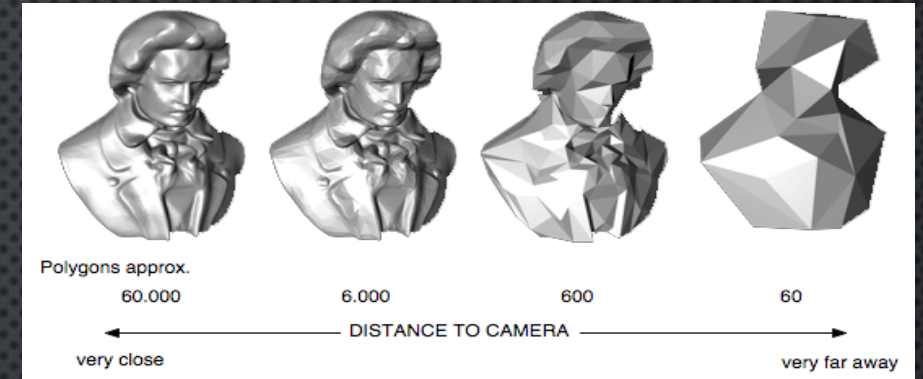
Optimalizace - Zjednodušení

Urychlení vykreslování

- **LOD** – geometrie
- **MipMap** – textura
- **Shader**

V **produkční** fázi podle platformy

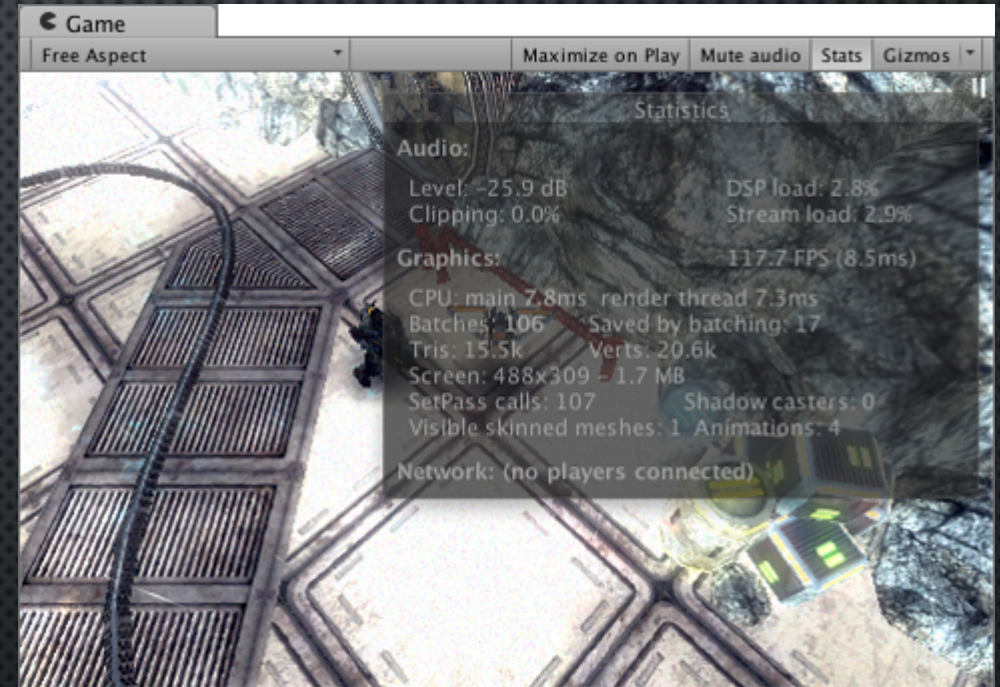
Za **běhu** podle vzdálenosti / nastavení



Optimalizace – Identifikace problémů v Unity

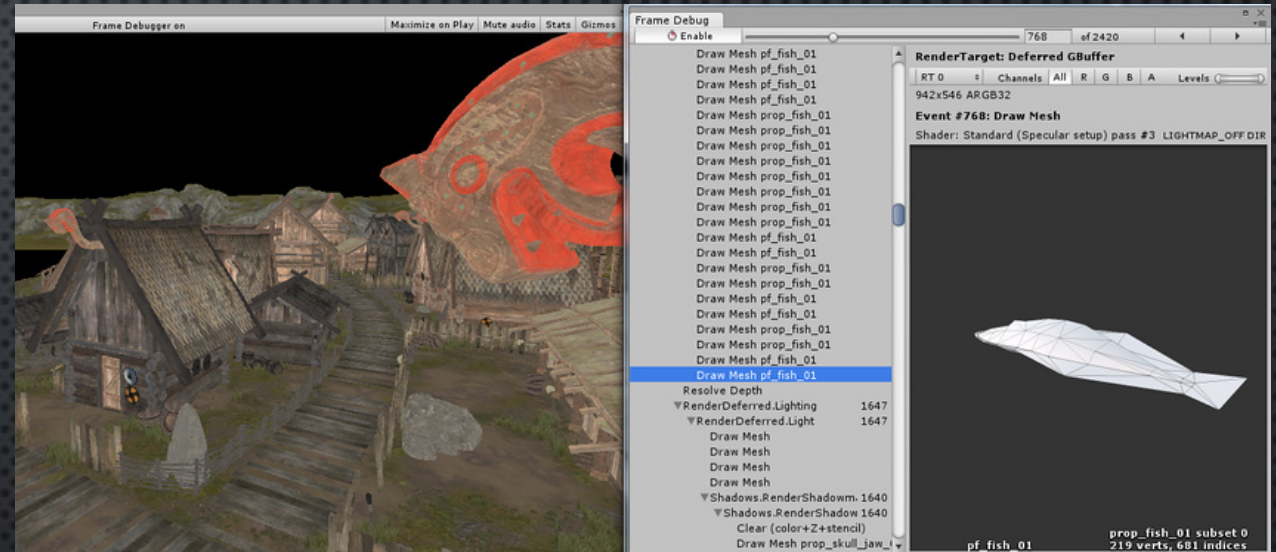
Optimalizace – Identifikace problémů v Unity

- **Rendering Statistics Window**
 - Tris & Verts, Batches



Optimalizace – Identifikace problémů v Unity

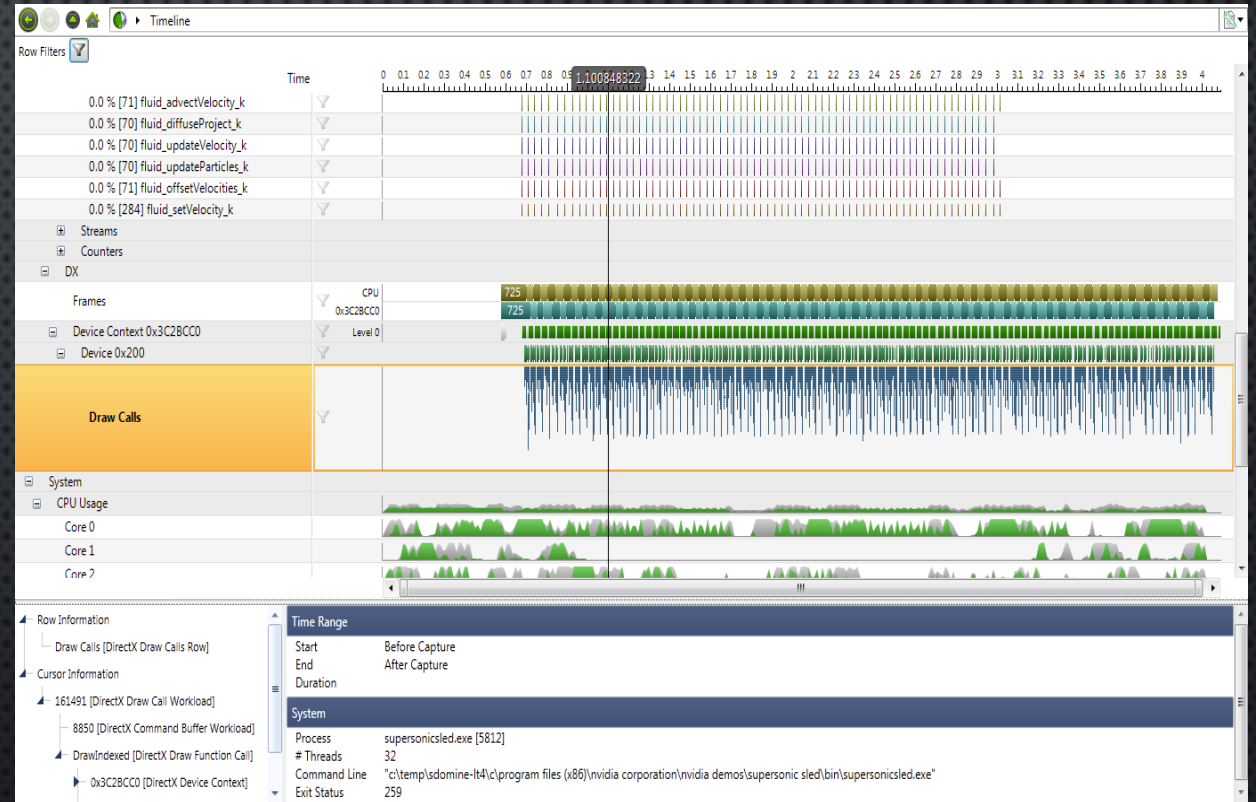
- Rendering Statistics Window
 - Tris & Verts, Batches
- Frame Debugger
 - Draw Calls, playback



Optimalizace – Identifikace problémů **Obecně**

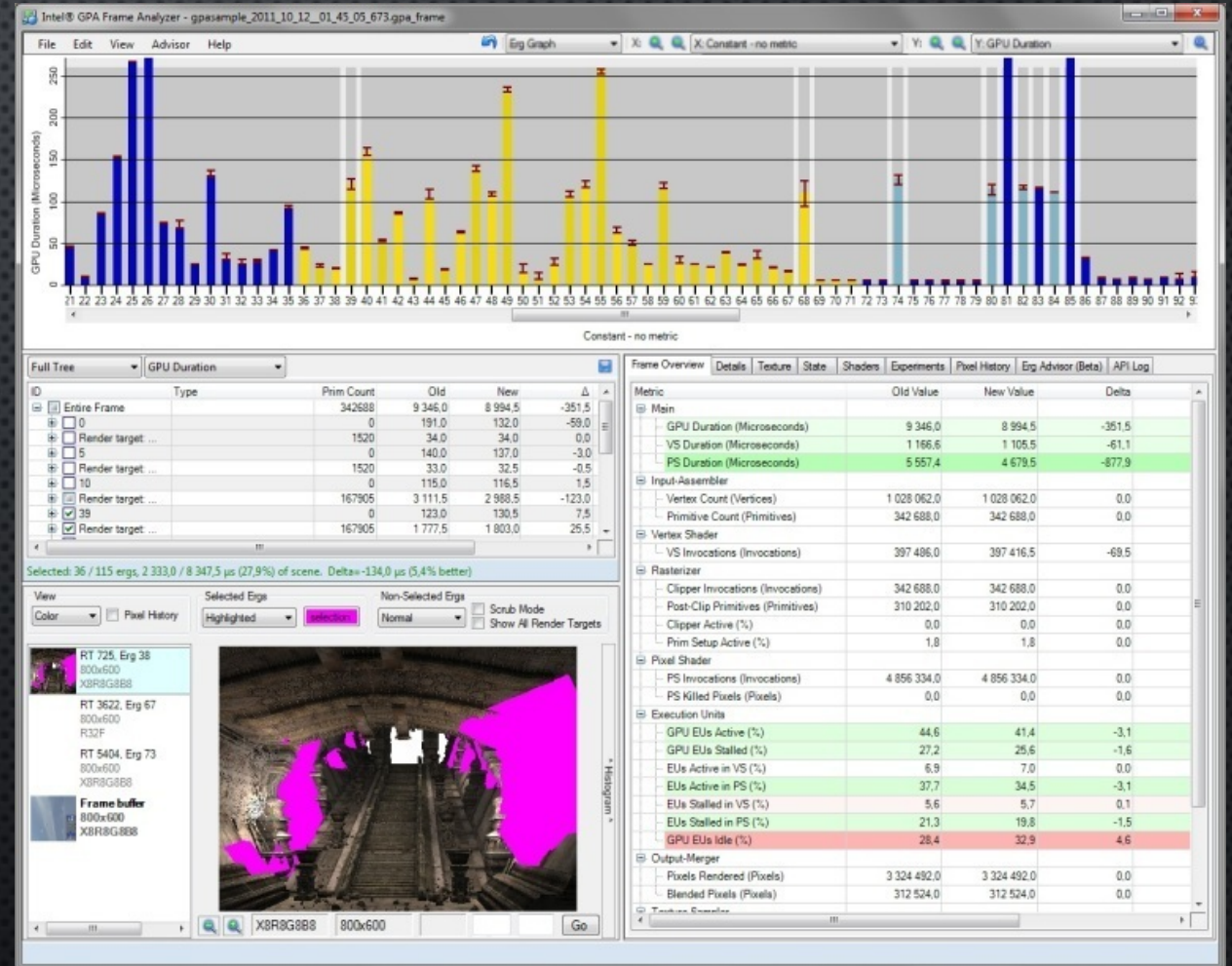
Optimalizace – Identifikace problémů Obecně

- **nVidia Nsight** (nástupce PerfHud)
 - Timing CPU&GPU, Bottlenecks,
 - Memory, Buffers, Shaders, CUDA
 - . . .
 - Integrace do Visual Studia



Optimalizace – Identifikace problémů Obecně

- nVidia Nsight (nástupce PerfHud)
 - Timing CPU&GPU, Bottlenecks,
 - Memory, Buffers, Shaders, CUDA
 - ...
- Integrace do Visual Studio
- Intel Graphics Performance Analyzer
 - Podobné, jen pro Intel. Jinak PS/VS time



Optimalizace – Identifikace problémů Obecně

- nVidia Nsight (nástupce PerfHud)
 - Timing CPU&GPU, Bottlenecks,
 - Memory, Buffers, Shaders, CUDA
 - Integrace do Visual Studia
- Intel Graphics Performance Analyzer
 - Podobné, jen pro Intel. Jinak PS/VS time
- GPU Perf Studio – ATI
 - Podobné nSightu, jen AMD

The screenshot displays the NVIDIA Nsight Visual Studio Edition interface. On the left, a pipeline view shows stages: IA, VS, HS, DS, GS, PS (selected), OM, and CS. The main window shows the PS[0] shader code, which is a pixel shader for shadow calculation. The code includes comments and operations for calculating light space coordinates, applying texture coordinates, and performing 2x2 percentage closer filtering. A table at the bottom of the code window shows the bounding box for a constant buffer (cb[0]).

Name	X	Y	Z	W
cb[0]	1.000000	1.000000	1.000000	1.000000
cb[1][0]	0.722057	0.000000	2.303706	0.000026
cb[1][1]	-1.638008	1.697578	0.513405	-594.152466
cb[1][2]	-0.731971	-0.775671	0.229424	815.989624
cb[1][3]	-0.670974	-0.711032	0.210305	847.990479
cb[1][4]	402.000000	776.000000	-125.999992	1.000000

On the right, a performance table shows the state bucket, draw call count, GPU time, and GPU busy percentage for various state buckets.

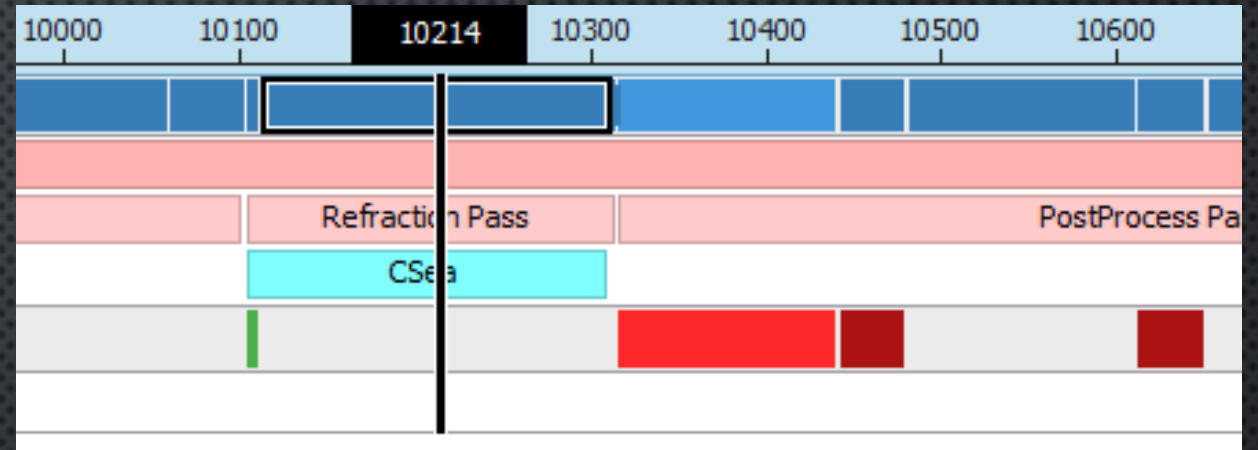
State Bucket	Draw Call #	GPU Time (ms)	GPU Busy (%)
SB 0	4138	11.209	97.139
ClearDepthStencilView	3	0.318	99.993
DrawIndexed	4087	0.107	99.985
DrawIndexed	1006	0.102	99.910
DrawIndexed	1007	0.080	99.895
DrawIndexed	3152	0.066	100
DrawIndexed	71	0.052	1.047
DrawIndexed	3085	0.051	100
DrawIndexed	682	0.048	99.988
DrawIndexed	4	0.041	99.963
DrawIndexed	3957	0.039	99.996
DrawIndexed	876	0.033	100
DrawIndexed	1903	0.032	100
DrawIndexed	2931	0.032	100
DrawIndexed	681	0.031	100
DrawIndexed	3939	0.031	99.997
DrawIndexed	304	0.030	99.781
DrawIndexed	4090	0.029	99.960
DrawIndexed	1009	0.021	99.724
DrawIndexed	3061	0.021	99.949
DrawIndexed	3295	0.021	99.993
DrawIndexed	3933	0.020	0.337
DrawIndexed	1002	0.018	99.701
DrawIndexed	2126	0.018	99.996
DrawIndexed	2737	0.017	99.993
DrawIndexed	1709	0.017	99.996
DrawIndexed	2736	0.017	99.993
DrawIndexed	3762	0.017	99.992
DrawIndexed	1708	0.017	99.996
DrawIndexed	3763	0.017	99.996
DrawIndexed	1504	0.016	99.993
DrawIndexed	477	0.016	99.992
DrawIndexed	3558	0.016	99.992
DrawIndexed	1031	0.016	100
DrawIndexed	2532	0.016	99.993
DrawIndexed	2059	0.016	0.379
DrawIndexed	858	0.016	99.665
DrawIndexed	1012	0.015	99.658
DrawIndexed	1013	0.015	99.667
ClearDepthStencilView	2	0.014	99.883
DrawIndexed	4086	0.013	100
DrawIndexed	272	0.013	99.632
ClearRenderTargetView	1	0.013	99.883
DrawIndexed	642	0.013	99.631

Optimalizace – Identifikace problémů

- Co hledáme?

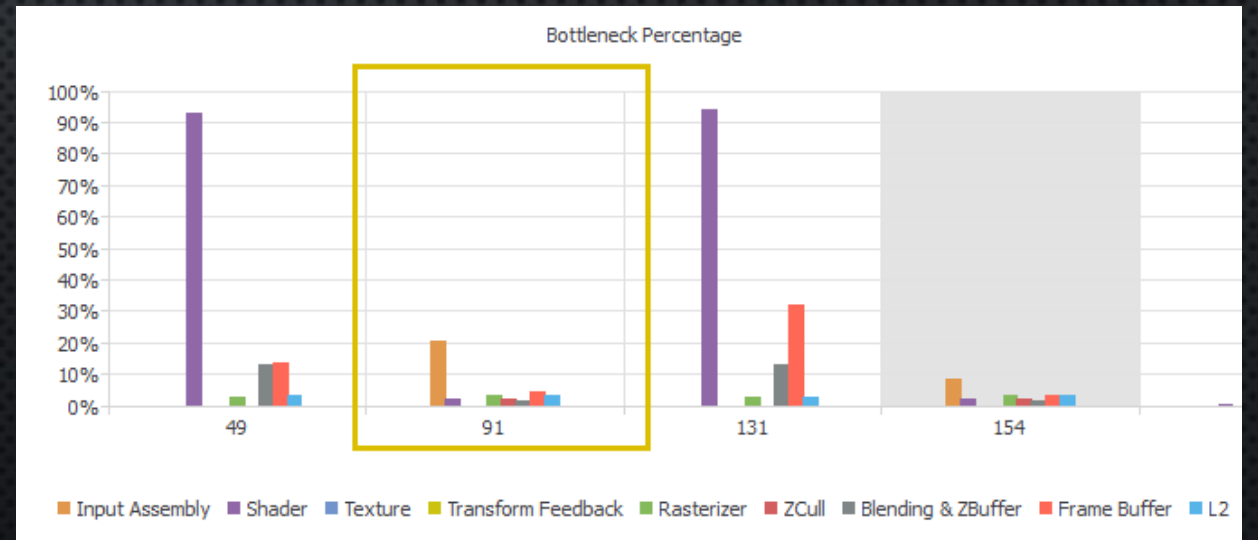
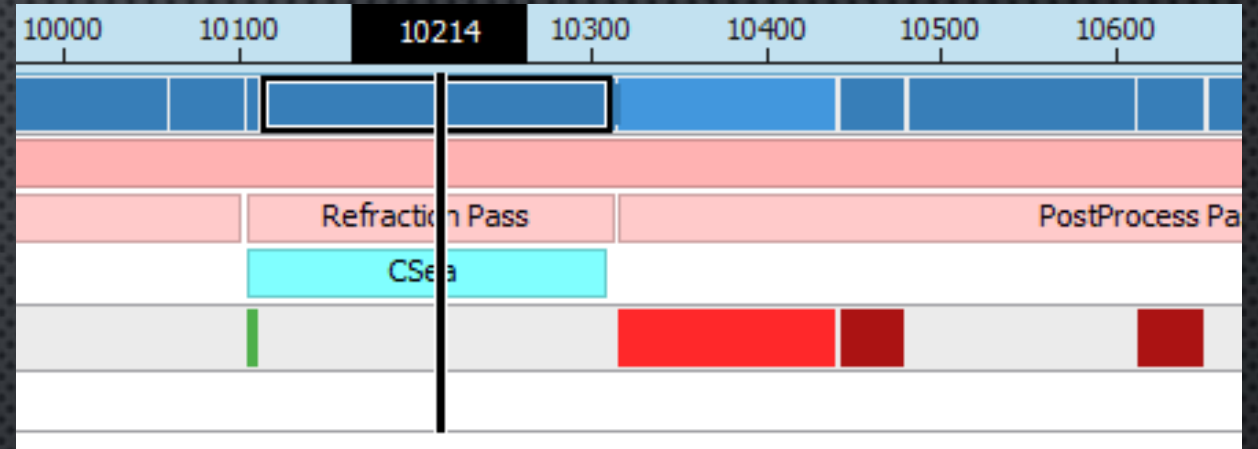
Optimalizace – Identifikace problémů

- Co hledáme?
 - Příliš dlouhé **Draw Calls** (>1ms)



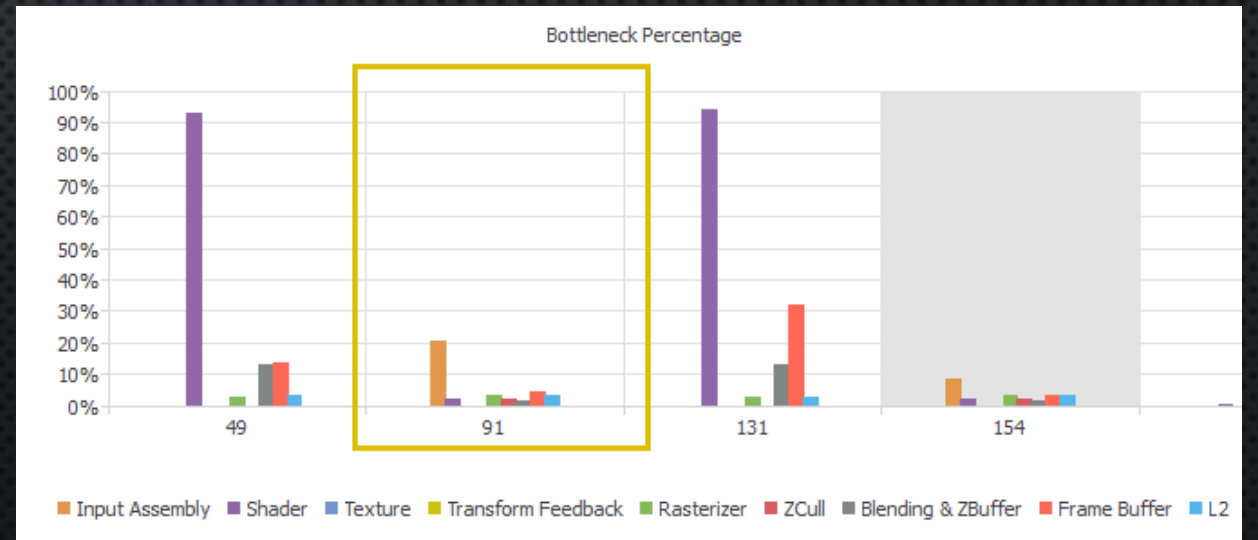
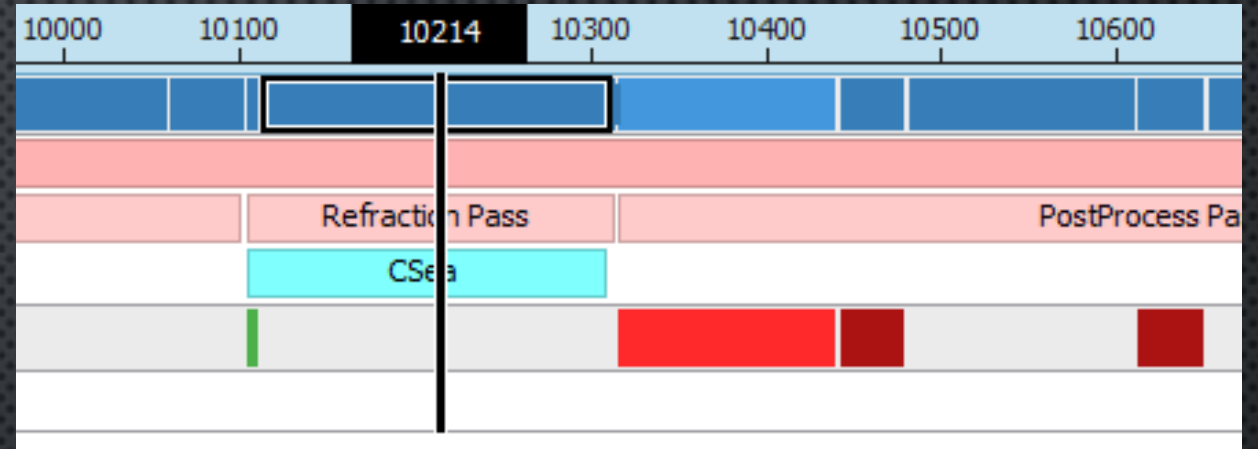
Optimalizace – Identifikace problémů

- Co hledáme?
 - Příliš dlouhé Draw Calls (>1ms)
 - Zjistíme v čem je problém – Shader logic, Frame buffer, L2 cache, ... ?



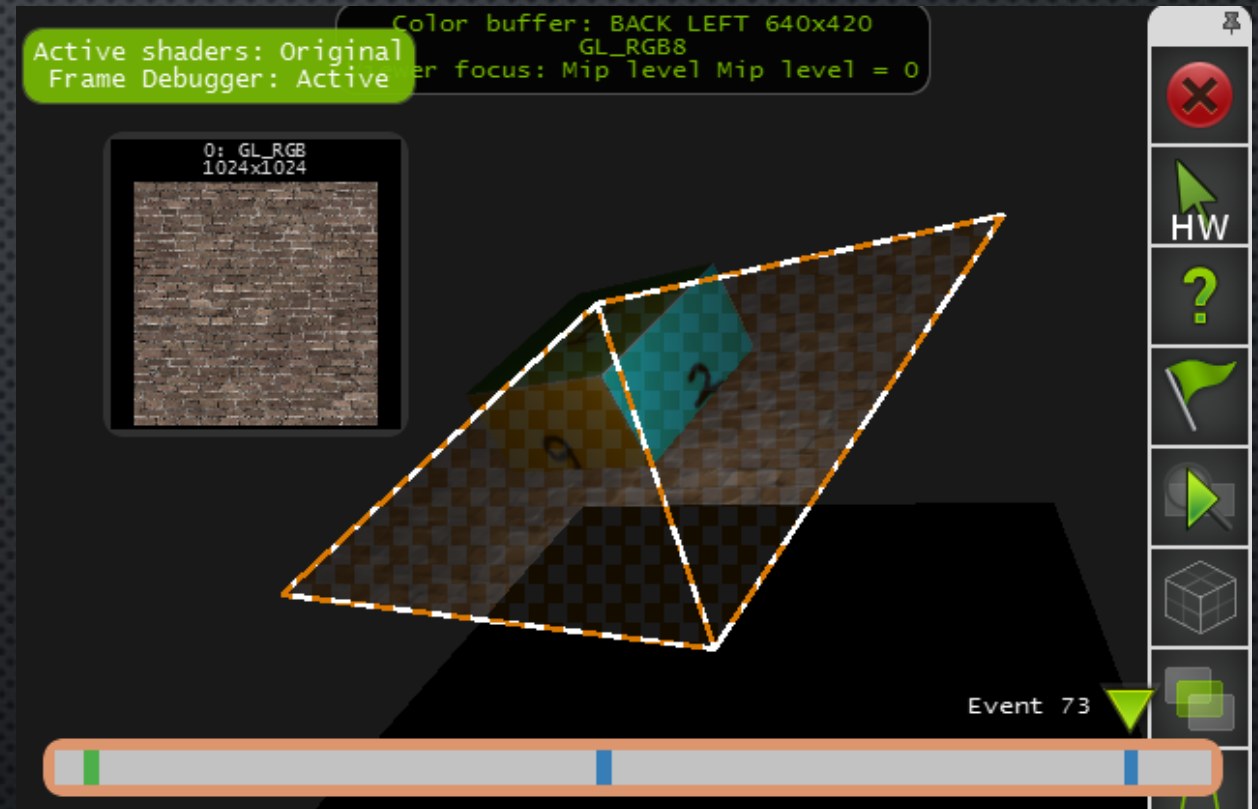
Optimalizace – Identifikace problémů

- Co hledáme?
 - Příliš dlouhé Draw Calls (>1ms)
 - Zjistíme v čem je problém – Shader logic, Frame buffer, L2 cache,...
 - **Závislosti** – Následuje efekt, který využívá výsledek hned předchozí operace?



Optimalizace – Identifikace problémů

- Co hledáme?
 - Příliš dlouhé Draw Calls (>1ms)
 - Zjistíme v čem je problém – Shader logic, Frame buffer, L2 cache,...
 - Závislosti – Následuje efekt, který využívá výsledek hned předchozí operace?
- **Problémy**
 - Stejně nástroje i při hledání chyb, glitchů, apod.



Optimalizace – Zásady

Optimalizace – Zásady

- Naučit se s těmito nástroji

Optimalizace – Zásady

- Naučit se s těmito nástroji
- Nevykreslovat co není nutné (zahazovat na CPU, využívat předchozí snímky, ...)

Optimalizace – Zásady

- Naučit se s těmito nástroji
- Nevykreslovat co není nutné (zahazovat na CPU, využívat předchozí snímky, ...)
- Minimalizovat Překreslování (Fillrate limit GPU, Depth pass?, Rozlišení mezivýsledků)

Optimalizace – Zásady

- Naučit se s těmito nástroji
- Nevykreslovat co není nutné (zahazovat na CPU, využívat předchozí snímky, ...)
- Minimalizovat Překreslování (Fillrate limit GPU, Depth pass?, Rozlišení mezivýsledků)
- LODovat (Vertex processing limit, počet per Vertex parametrů)

Optimalizace – Zásady

- Naučit se s těmito nástroji
- Nevykreslovat co není nutné (zahazovat na CPU, využívat předchozí snímky, ...)
- Minimalizovat Překreslování (Fillrate limit GPU, Depth pass?, Rozlišení mezivýsledků)
- LODovat (Vertex processing limit, počet per Vertex parametrů)
- Batche (unity) – shlukování geometrie dohromady, vykreslování naráz (static/dynamic)

Optimalizace – Zásady

- Naučit se s těmito nástroji
- Nevykreslovat co není nutné (zahazovat na CPU, využívat předchozí snímky, ...)
- Minimalizovat Překreslování (Fillrate limit GPU, Depth pass?, Rozlišení mezivýsledků)
- LODovat (Vertex processing limit, počet per Vertex parametrů)
- Batche (unity) – shlukování geometrie dohromady, vykreslování naráz (static/dynamic)
- Instancing – vykreslování více instancí jednoho objektu naráz (stromy např.)

Optimalizace – Zásady

- Naučit se s těmito nástroji
- Nevykreslovat co není nutné (zahazovat na CPU, využívat předchozí snímky, ...)
- Minimalizovat Překreslování (Fillrate limit GPU, Depth pass?, Rozlišení mezivýsledků)
- LODovat (Vertex processing limit, počet per Vertex parametrů)
- Batche (unity) – shlukování geometrie dohromady, vykreslování naráz (static/dynamic)
- Instancing – vykreslování více instancí jednoho objektu naráz (stromy např.)
- Co nejméně materiálů (spojování textur do jedné)

Optimalizace – Zásady

- Naučit se s těmito nástroji
- Nevykreslovat co není nutné (zahazovat na CPU, využívat předchozí snímky, ...)
- Minimalizovat Překreslování (Fillrate limit GPU, Depth pass?, Rozlišení mezivýsledků)
- LODovat (Vertex processing limit, počet per Vertex parametrů)
- Batche (unity) – shlukování geometrie dohromady, vykreslování naráz (static/dynamic)
- Instancing – vykreslování více instancí jednoho objektu naráz (stromy např.)
- Co nejméně materiálů (spojování textur do jedné)
- Vypékání všeho nedynamického (nasvětlení levelu do textur / vertex color ?)

Realistická grafika

- Geometrie objektu
- Přímé nasvětlení
- Stíny
- Nepřímé nasvětlení a odrazy
- Physically based rendering
- Optimalizace
- Zpracování výsledného obrazu

Realistická grafika

- Geometrie objektu
- Přímé nasvětlení
- Stíny
- Nepřímé nasvětlení a odrazy
- Physically based rendering
- Optimalizace
- **Zpracování výsledného obrazu**



Zpracování obrazu - PostProcesy

- Nejjednodušší postFx je
tónování obrazu

Zpracování obrazu - PostProcesy

- Nejjednodušší postFx je tónování obrazu
- Mafia 1 – obarvení jen pomocí **barvy světla**



Zpracování obrazu - PostProcesy

- Nejjednodušší postFx je tónování obrazu
- Mafia 1 – obarvení jen pomocí **barvy světla**
- Mafia 2 – jednoduché **obarvení post-processem**



Zpracování obrazu - PostProcesy

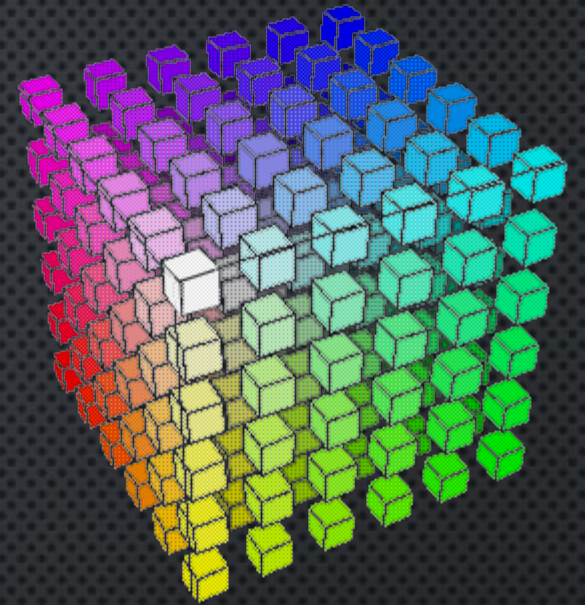
- Nejjednodušší postFx je tónování obrazu
- Mafia 1 – obarvení jen pomocí **barvy světla**
- Mafia 2 – jednoduché **obarvení post-processem**
- Grafici by ale chtěli:



Zpracování obrazu - Color grading

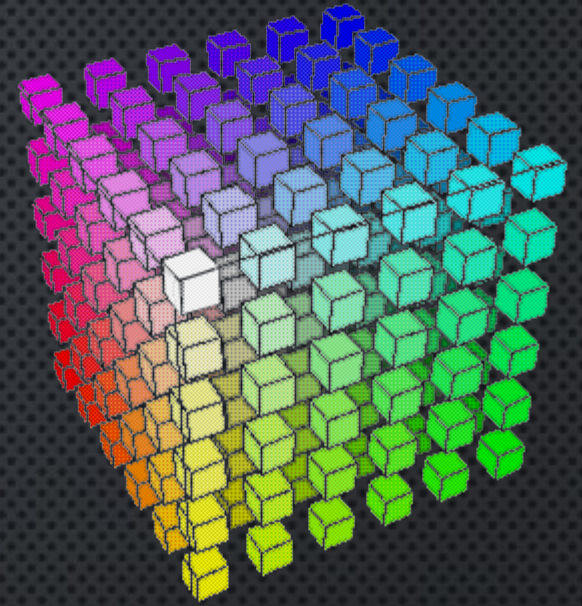
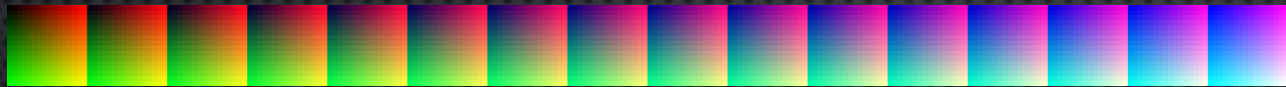
Zpracování obrazu - Color grading

- Překladová tabulka (3D textura) kde $XYZ \sim RGB$



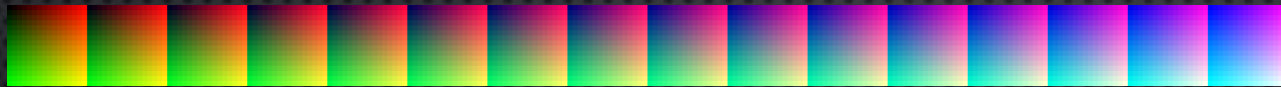
Zpracování obrazu - Color grading

- Překladová tabulka (3D textura) kde $XYZ \sim RGB$
- Tabulka se rozbalí do 2D a ukryje do screenshotu ze hry

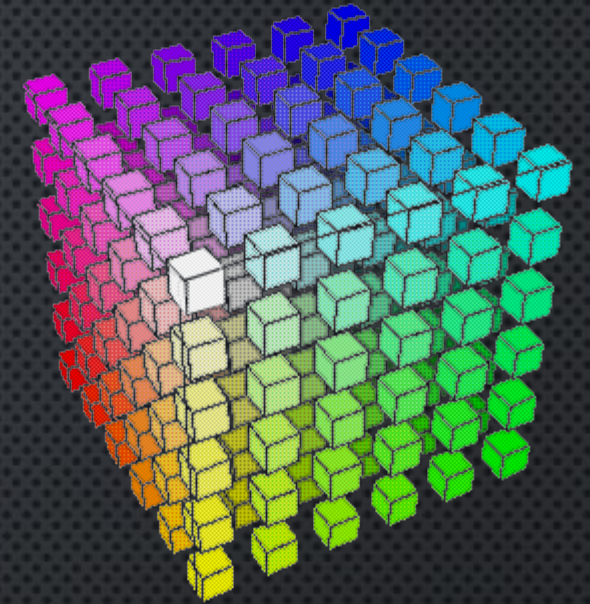


Zpracování obrazu - Color grading

- Překladová tabulka (3D textura) kde $XYZ \sim RGB$
- Tabulka se rozbalí do 2D a ukryje do screenshotu ze hry

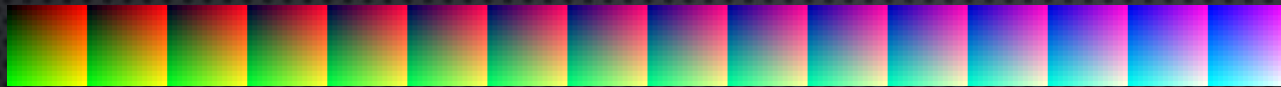


- Screenshot se upraví třeba ve photoshopu.



Zpracování obrazu - Color grading

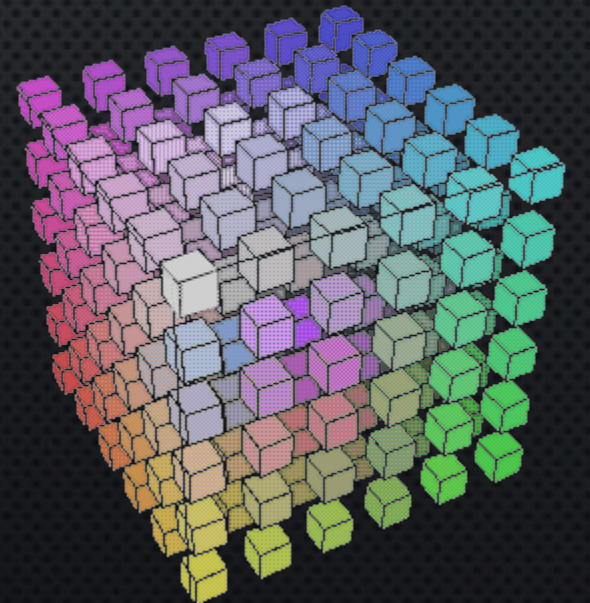
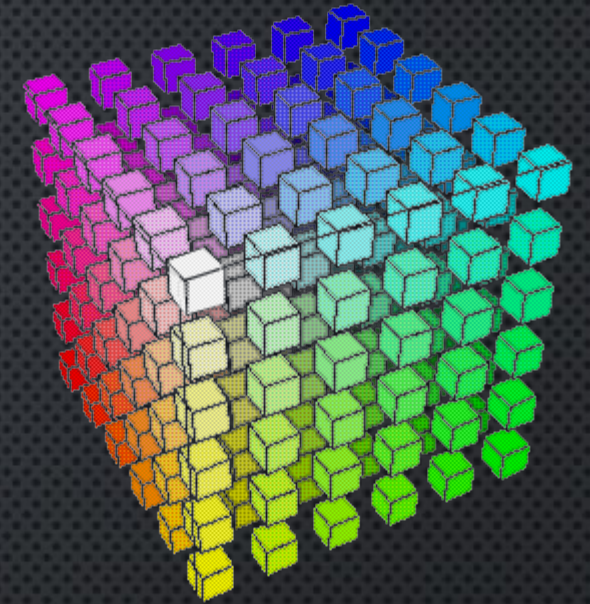
- Překladová tabulka (3D textura) kde $XYZ \sim RGB$
- Tabulka se rozbalí do 2D a ukryje do screenshotu ze hry



- Screenshot se upraví třeba ve photoshopu.



- Výsledek se vyjme a použije v shaderu jako překladová tabulka



Zpracování obrazu - Color grading

- Přečte se barva obrazu, použije se jako souřadnice do překladové tabulky [červená, zelená, modrá] → [x,y,z] a přečtená barva v tabulce se pak aplikuje.



Zpracování obrazu - HDR

- High Dynamic Range + Tonemapping



Zpracování obrazu - HDR

- Formát Backbufferu namísto R8G8B8X8 -> R16G16B16A16F

Zpracování obrazu - HDR

- **Formát Backbufferu** namísto **R8G8B8X8** -> **R16G16B16A16F**
- **HDR** data -> **Tone mapping** (blue shift) -> **LDR** buffer

Zpracování obrazu - HDR

- **Formát Backbufferu** namísto R8G8B8X8 -> R16G16B16A16F
- **HDR** data -> **Tone mapping** (blue shift) -> **LDR** buffer
- **Dynamické** přizpůsobování **expozice** – problémy s kolísáním, histogramem

Zpracování obrazu - HDR

- **Formát Backbufferu** namísto R8G8B8X8 -> R16G16B16A16F
- **HDR data** -> **Tone mapping** (blue shift) -> **LDR buffer**
- **Dynamické** přizpůsobování **expoziční** – problémy s kolísáním, histogramem
- Data pro **Bloom efekt**



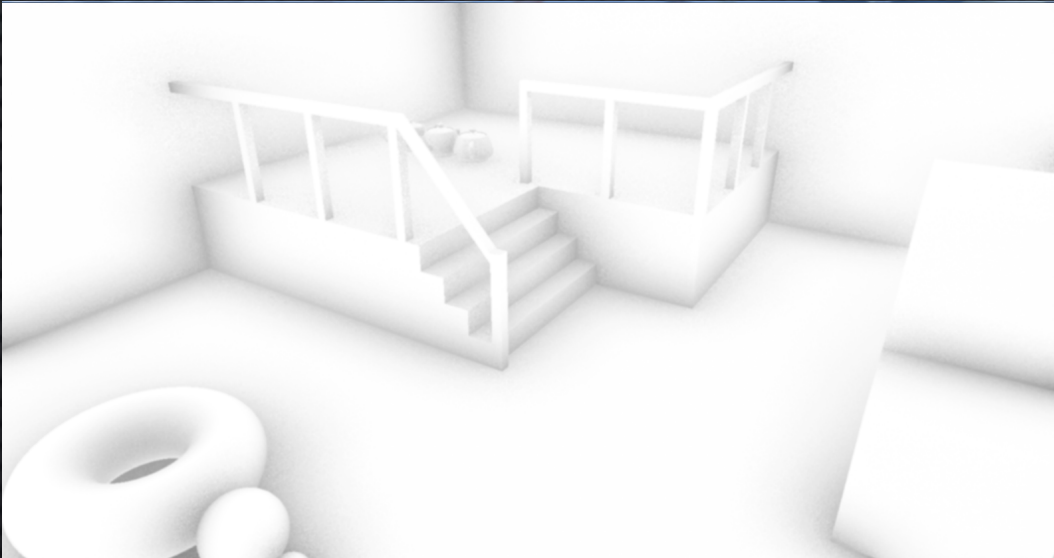
Zpracování obrazu - HDR

- **Formát Backbufferu** namísto R8G8B8X8 -> R16G16B16A16F
- **HDR data** -> **Tone mapping** (blue shift) -> **LDR buffer**
- **Dynamické** přizpůsobování **expoze** – problémy s kolísáním, histogramem
- Data pro **Bloom efekt**
- Složitější nastavení scény



Zpracování obrazu - SSAO

- Screen Space Ambient Occlusion
 - Ztmavení koutů. Snaha o napodobení měkkého světla oblohy



Zpracování obrazu - SSAO

- Bez



Zpracování obrazu - SSAO

- S



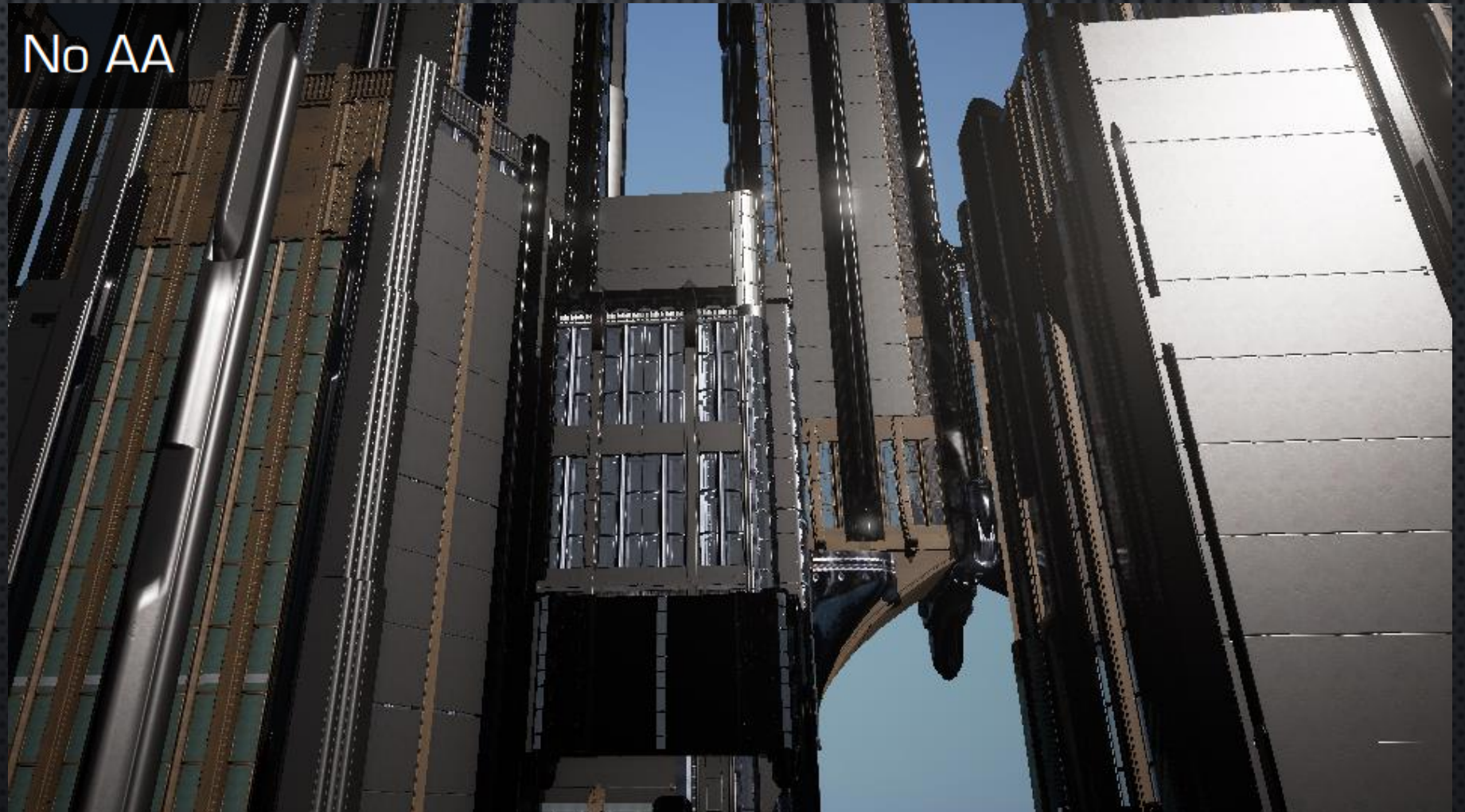
Zpracování obrazu - SSAO

- SSAO



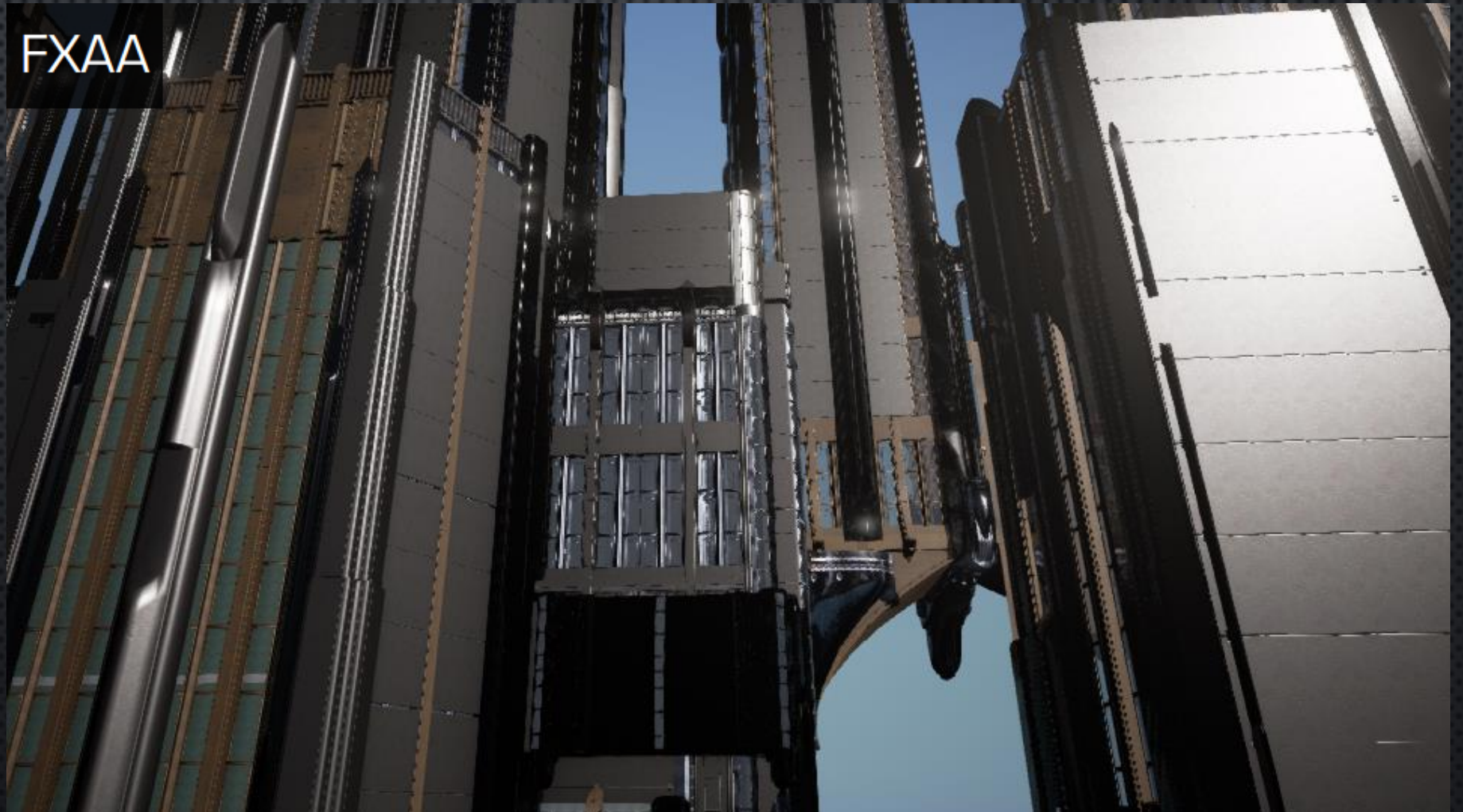
Zpracování obrazu – Vyhlazení (Anti Aliasing)

- Bez



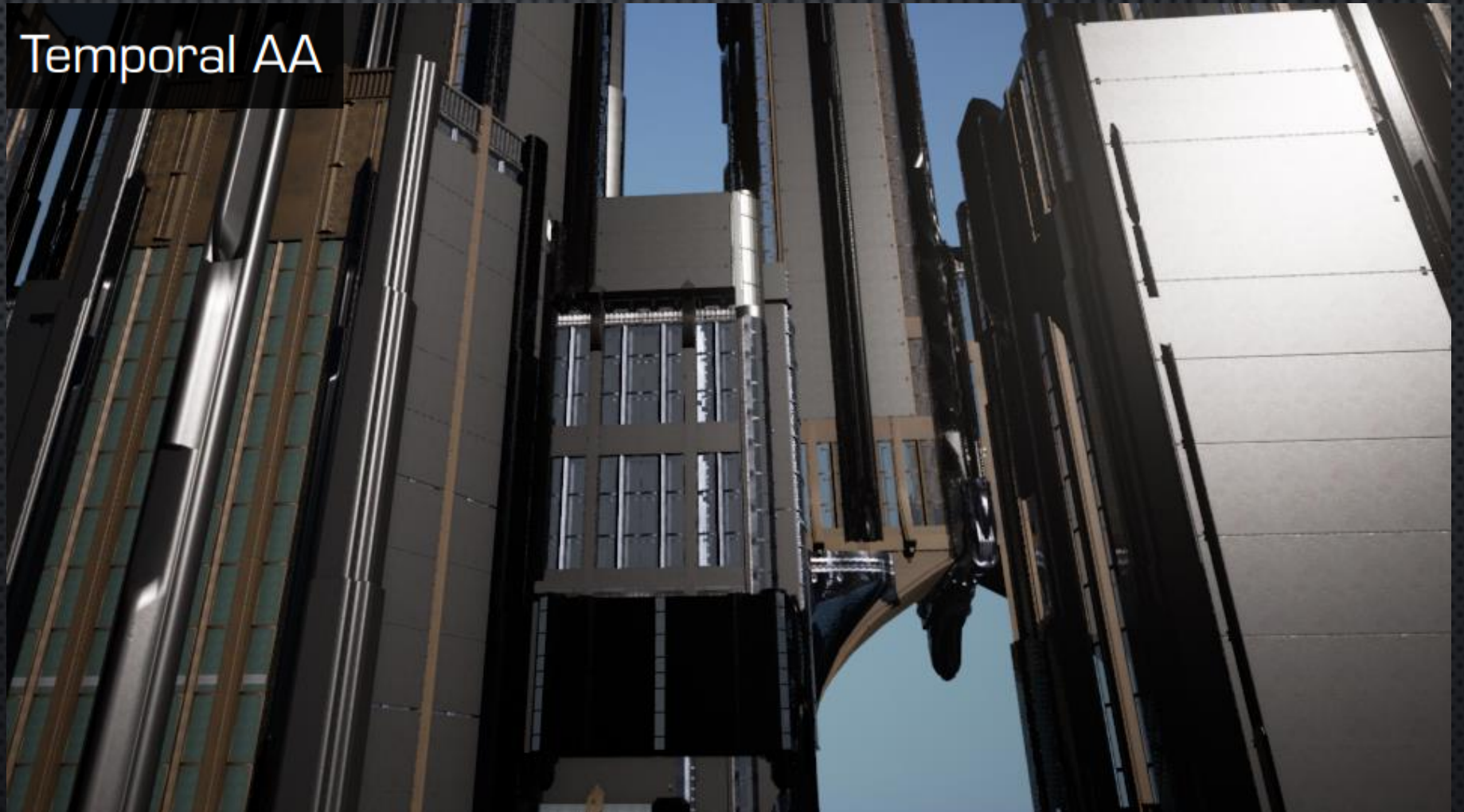
Zpracování obrazu – Vyhlazení (Anti Aliasing)

- Bez
 - FXAA / SMAA
- Post process



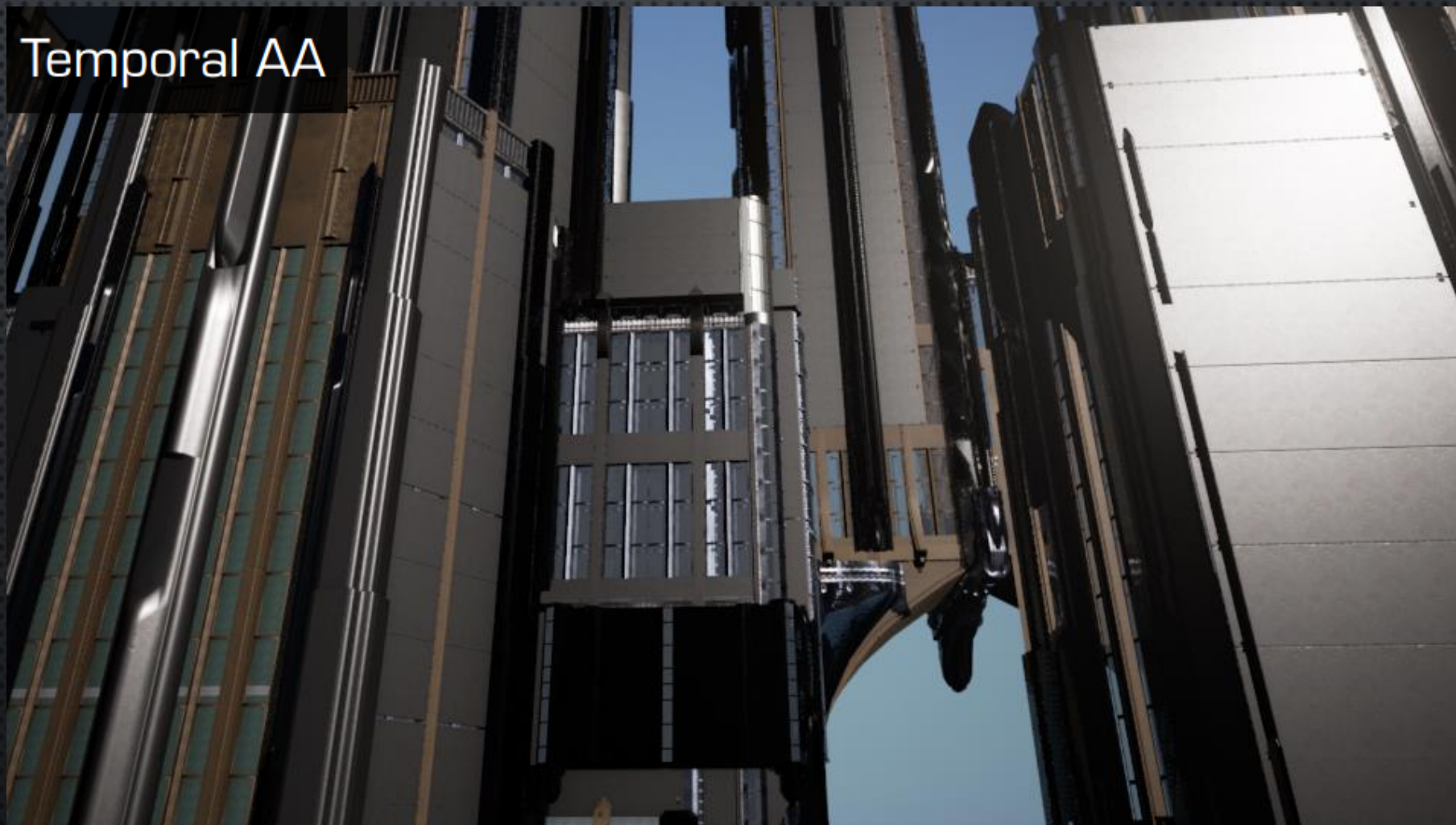
Zpracování obrazu – Vyhlazení (Anti Aliasing)

- Bez
- FXAA / SMAA
Post process
- **TXAA**
Kombinuje
předchozí
snímky

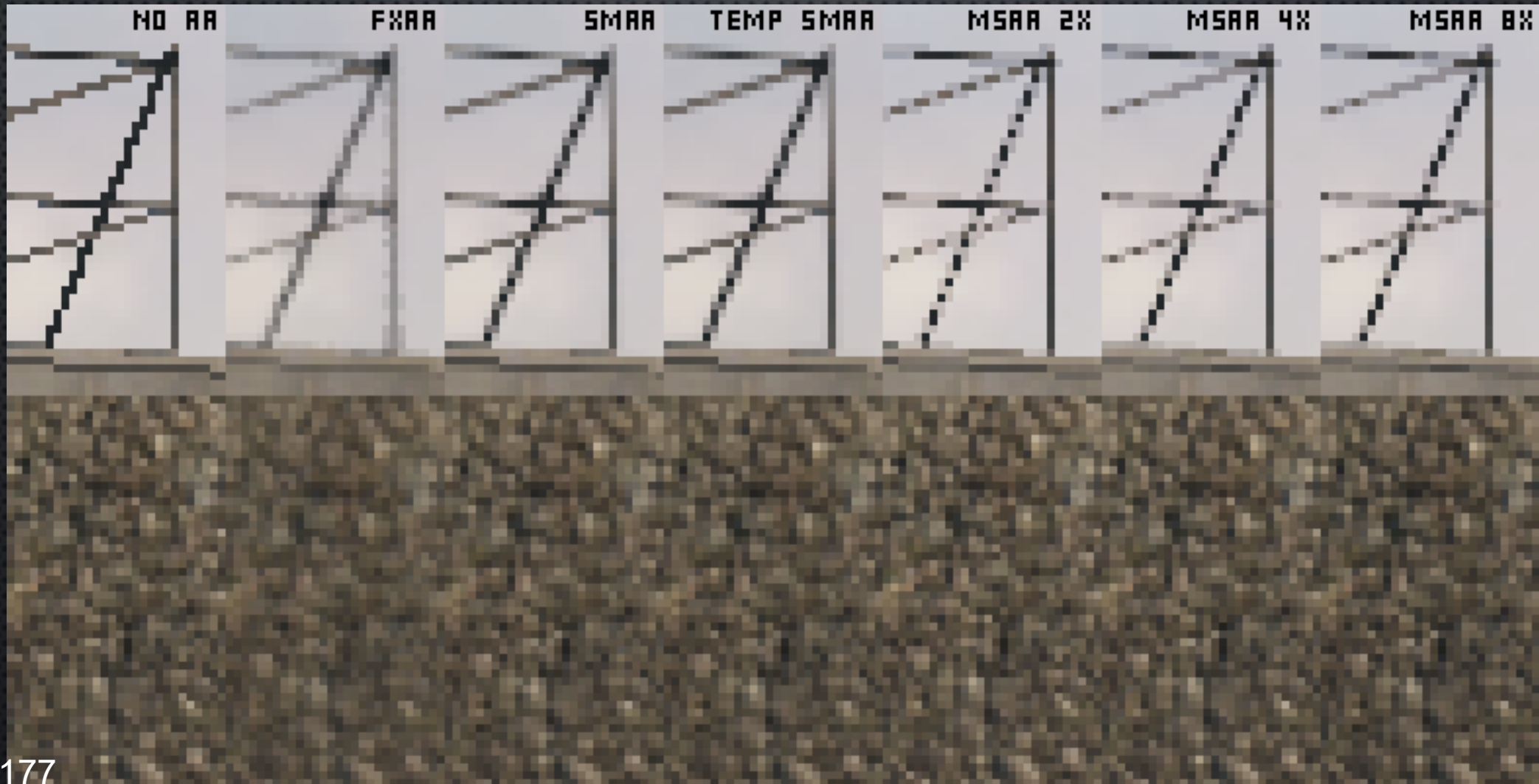


Zpracování obrazu – Vyhlazení (Anti Aliasing)

- Bez
- FXAA / SMAA
Post process
- TXAA
Kombinuje
předchozí
snímky
- **MSAA / SSAA**
Sub pixel



Zpracování obrazu – Vyhlazení (Anti Aliasing)



Zpracování obrazu – Depth Of Field



Zpracování obrazu – Depth Of Field

- Fyzikálně korektní **Bokeh**
- F-Stop, Full Frame, Ohnisková vzdálenost, Počet lamel, ...



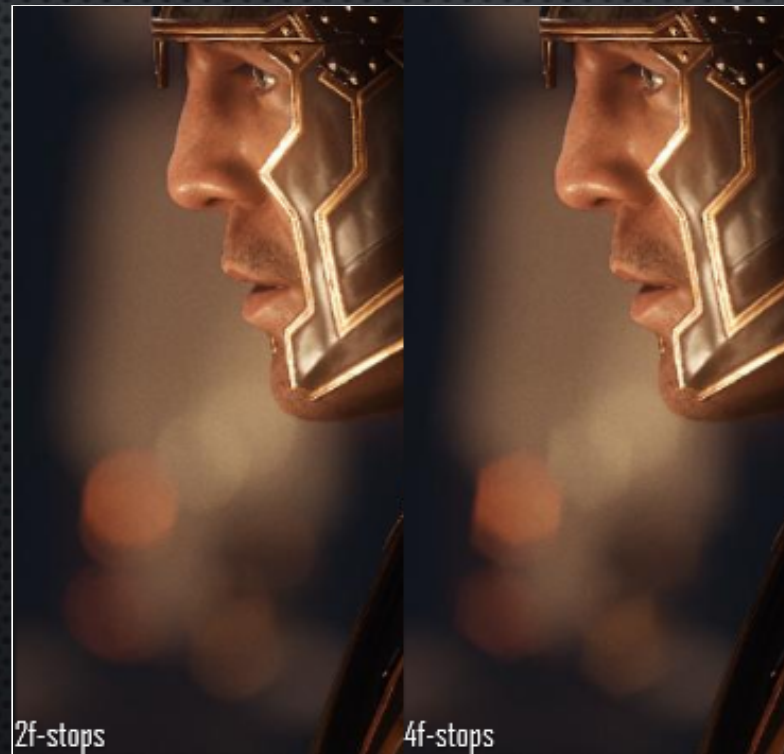
Zpracování obrazu – Depth Of Field

- Váhování pro potlačení přetékání do z popředí do pozadí
- Více průchodů



Zpracování obrazu – Depth Of Field

- Váhování pro potlačení přetékání do z popředí do pozadí
- Fyzikální vlastnosti clony

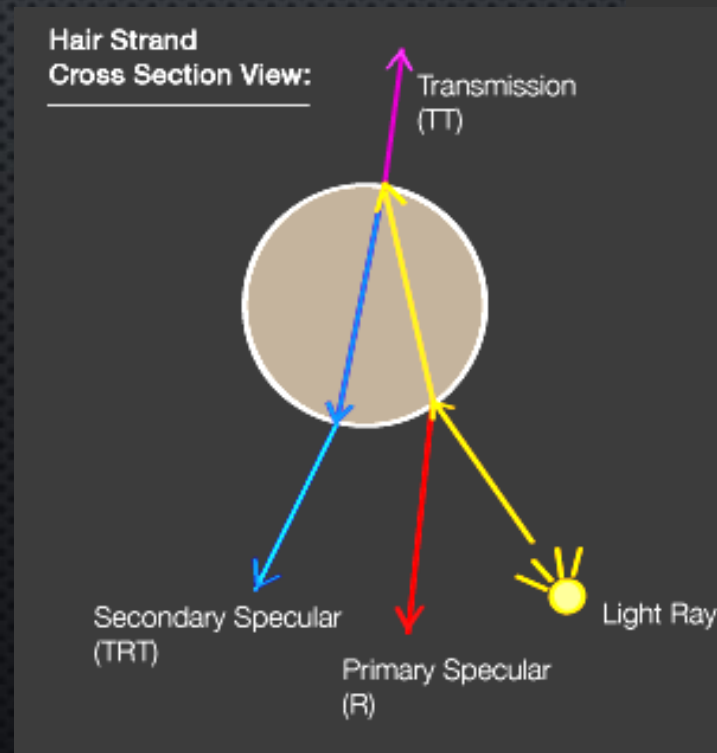
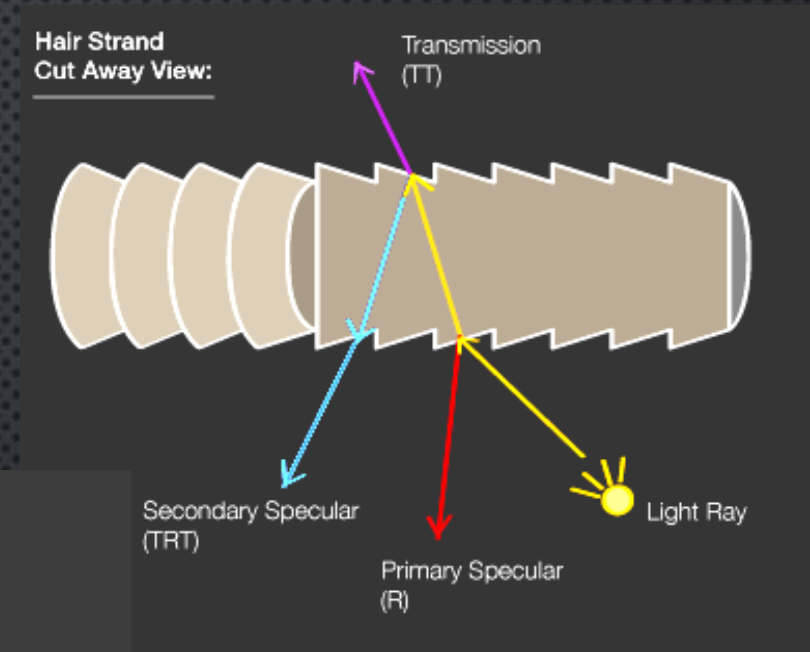


Zpracování obrazu – Depth Of Field - M3



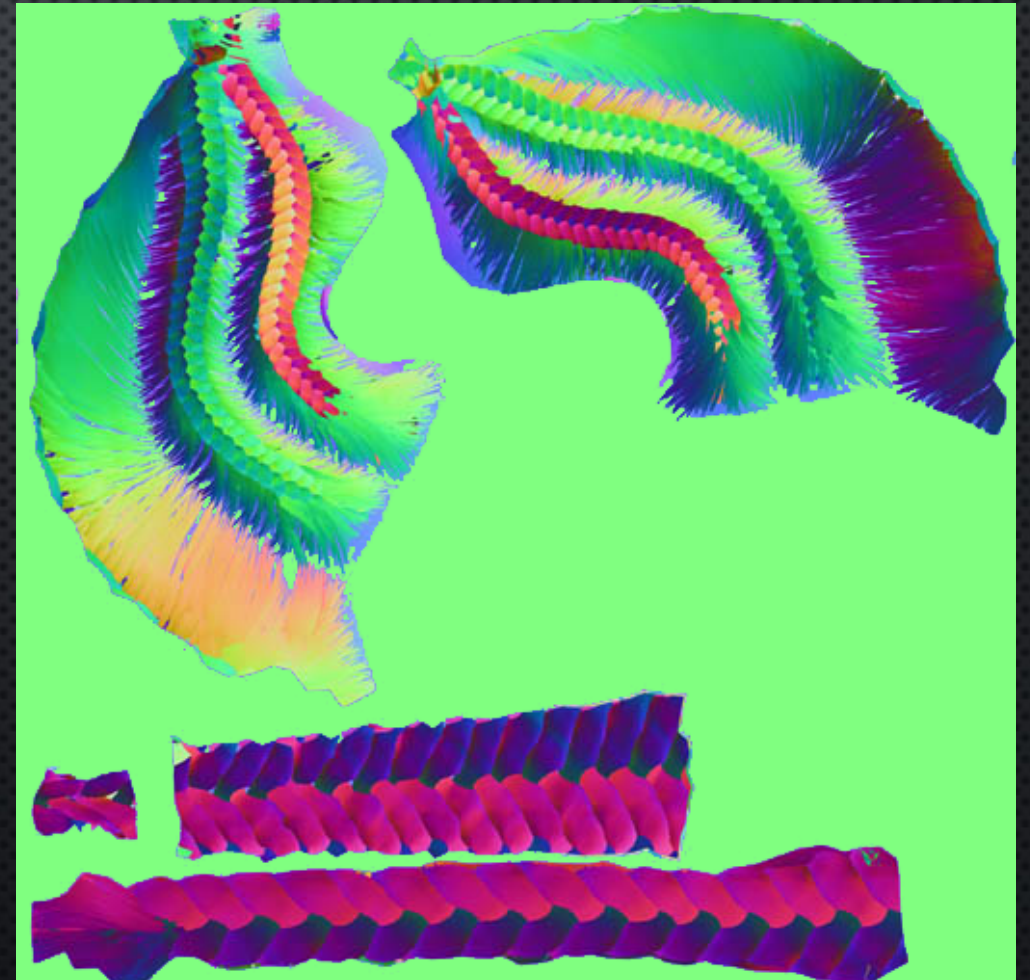
Zpracování obrazu – Hair shader

- Marschnerovi rovnice
 - Light scattering from Human Hair Fibers
 - Statistická metoda
- Tři složky:
 - **R** – Odraz
 - **TT** – Průchod skrz
 - **TRT** – Vnitřní odraz



Zpracování obrazu – Hair shader

- **Marschnerovi rovnice**
 - Light scattering from Human Hair Fibers
 - Statistická metoda
 - Tři složky:
 - **R** – **Odraz**
 - **TT** – **Průchod skrz**
 - **TRT** – **Vnitřní odraz**
- **Tangent** mapa místo Normal map

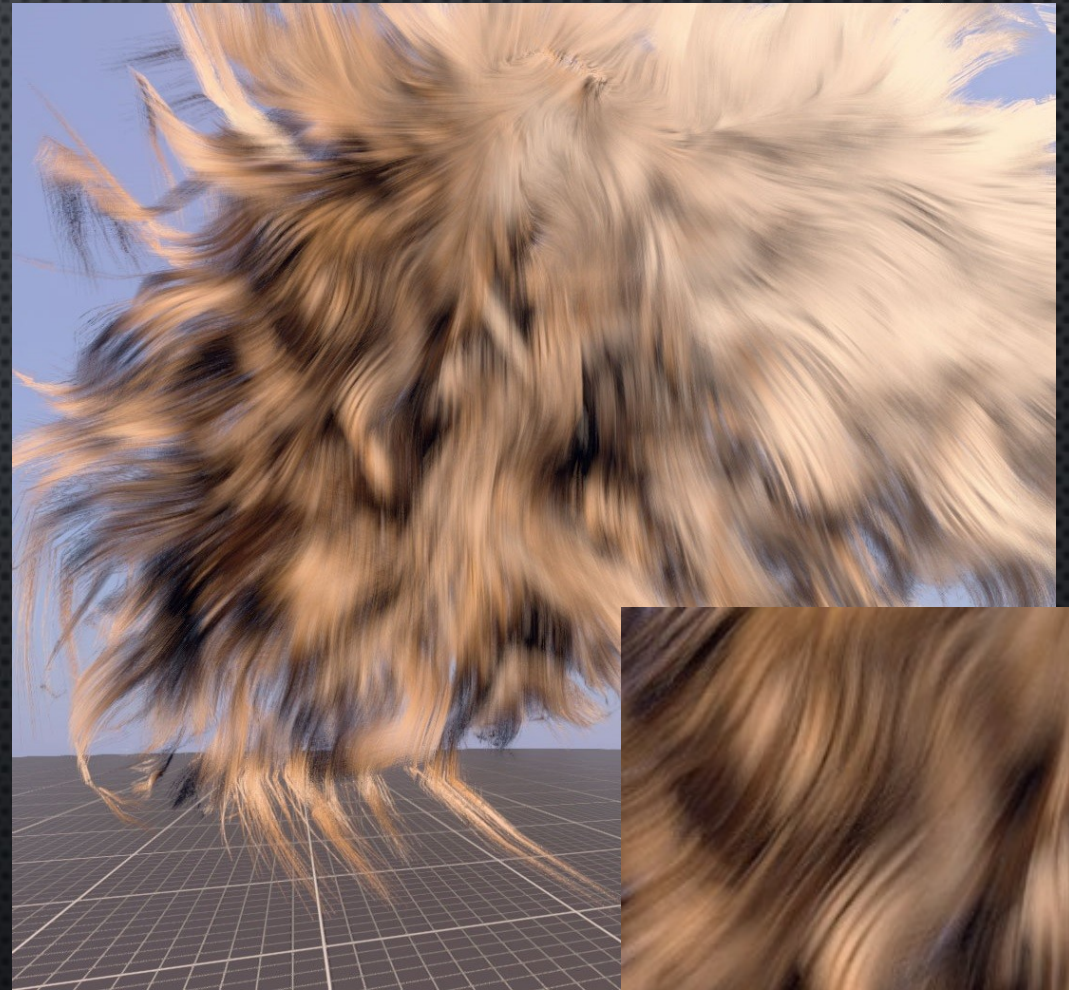
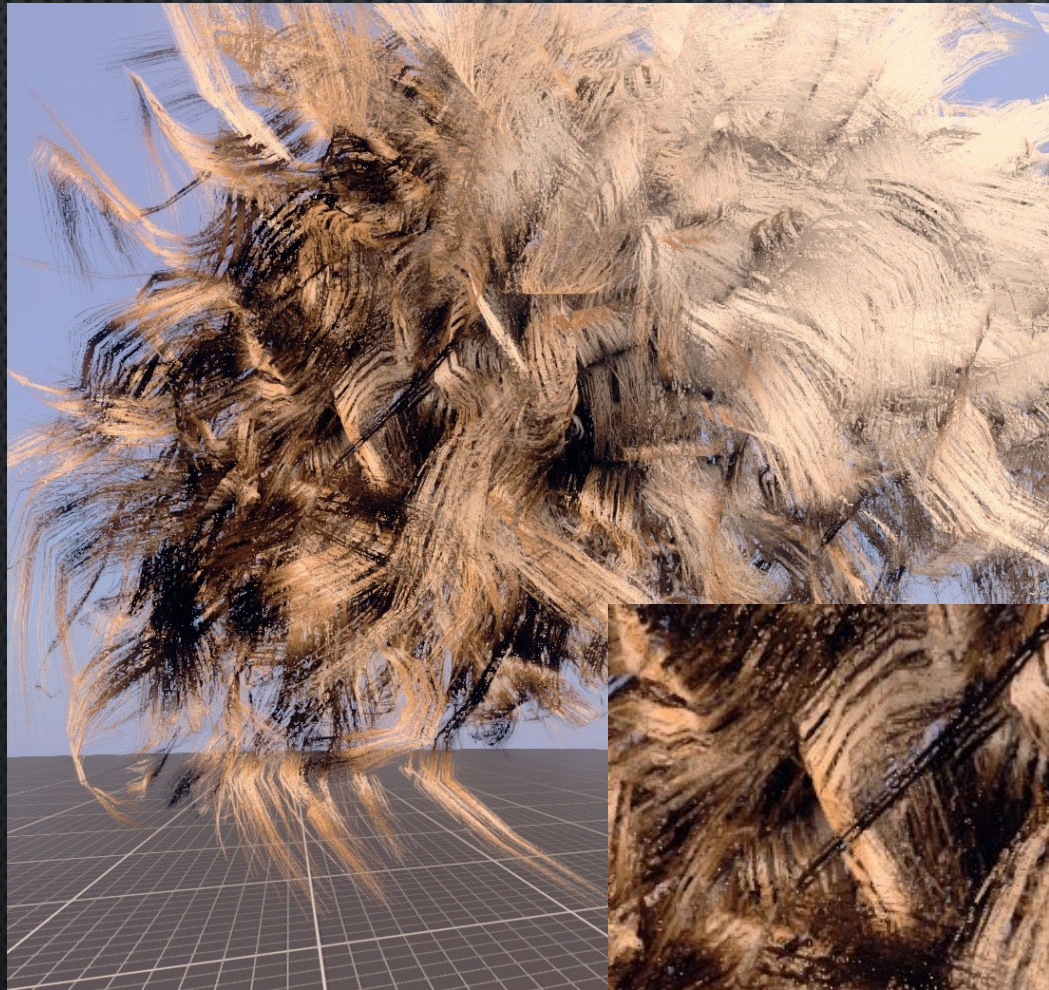


Zpracování obrazu – Hair shader

- **Marschnerovi rovnice**
 - Light scattering from Human Hair Fibers

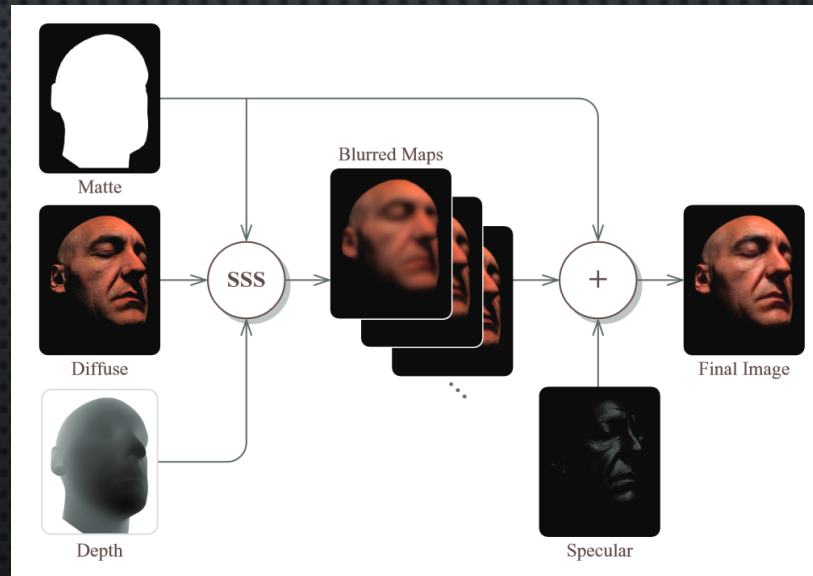


Zpracování obrazu – Hair shader



Zpracování obrazu – Skin shader

- Stíny pomocí **fake Sub Surface Scatteringu** + **PBR** materiál
- Metodou v **Image space**: **Stíny do textury** → **Rozostřit** → **Aplikovat zpět** změkčené stíny spolu s barevným posunem

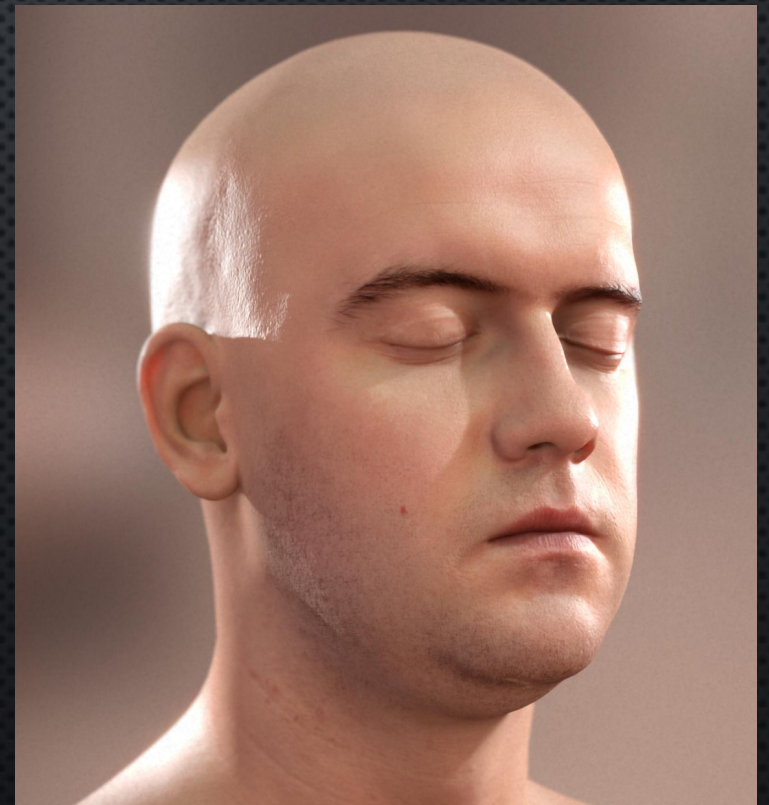
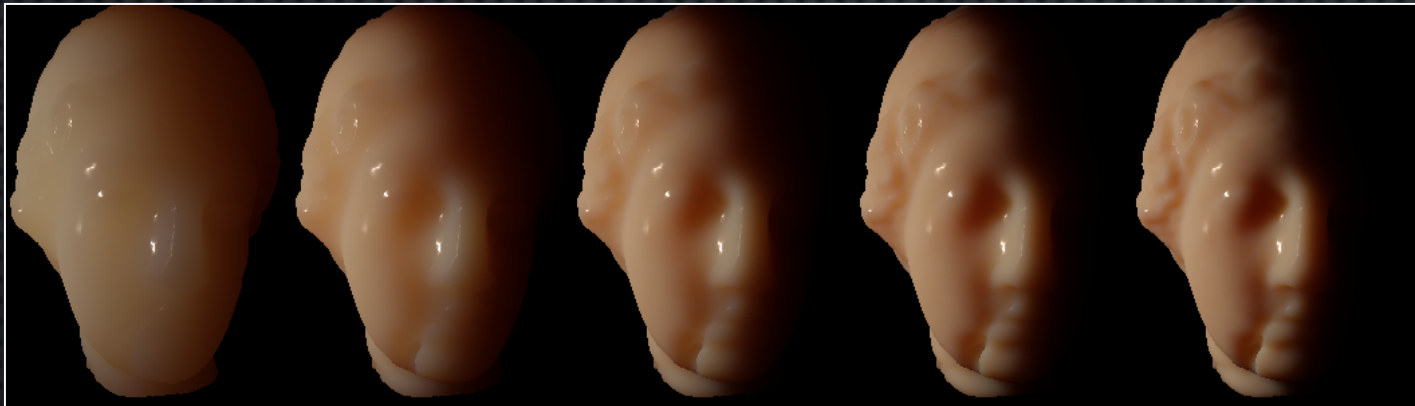


Jorge Jimenez 2009 – Screen-Space Perceptual Rendering of Human Skin

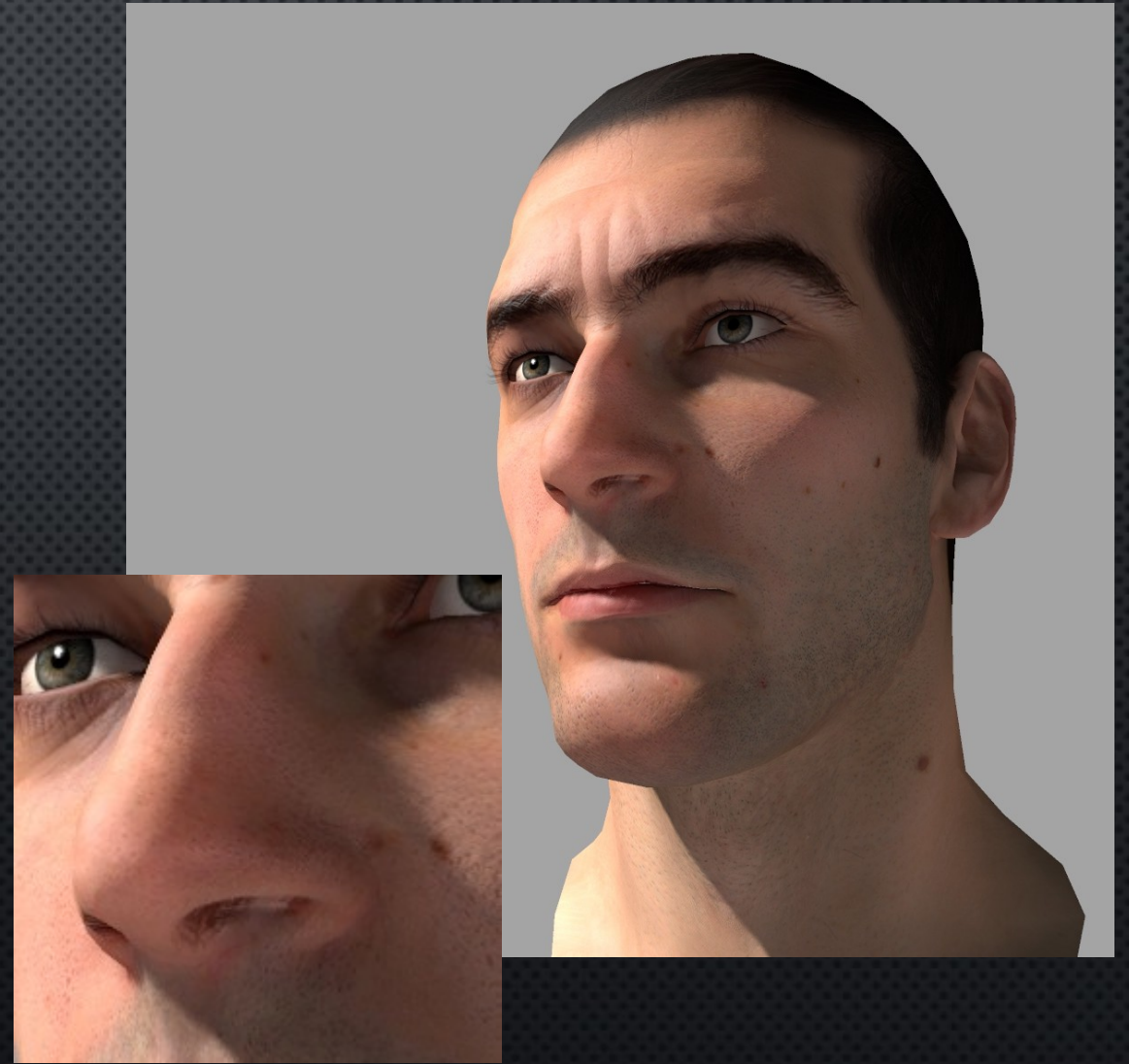
Zpracování obrazu – Skin shader

- Využití přesnějšího **Sub Surface Scatteringu** – Prosvětlení materiálu
- Metodou pomocí **Shadow Depth map**:

Porovnání **vzdálenosti** mezi pozorovaným bodem a vzdáleností od světla v shadow mapě



Zpracování obrazu – Sub Surface Scattering



Na závěr

- **Mafia III** Facial Animation Reel
 - Face anims
 - Depth of Field
 - Hair shader
 - Skin shader

<https://vimeo.com/188343452>



Díky ! Dotazy ?

jan.zeleny@bistudio.com