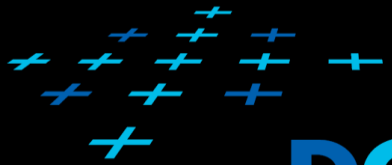


DCGI

KATEDRA POČÍTAČOVÉ GRAFIKY A INTERAKCE

Optimalizační metody pro hry

Jiří Bittner

Obsah přednášky

- Optimalice pro hry — GEA 14.1-14.4, RTR 18-20
  - Optimalizace scény
  - LOD
  - Předpočítání osvětlení
  - Redukování podle viditelnosti

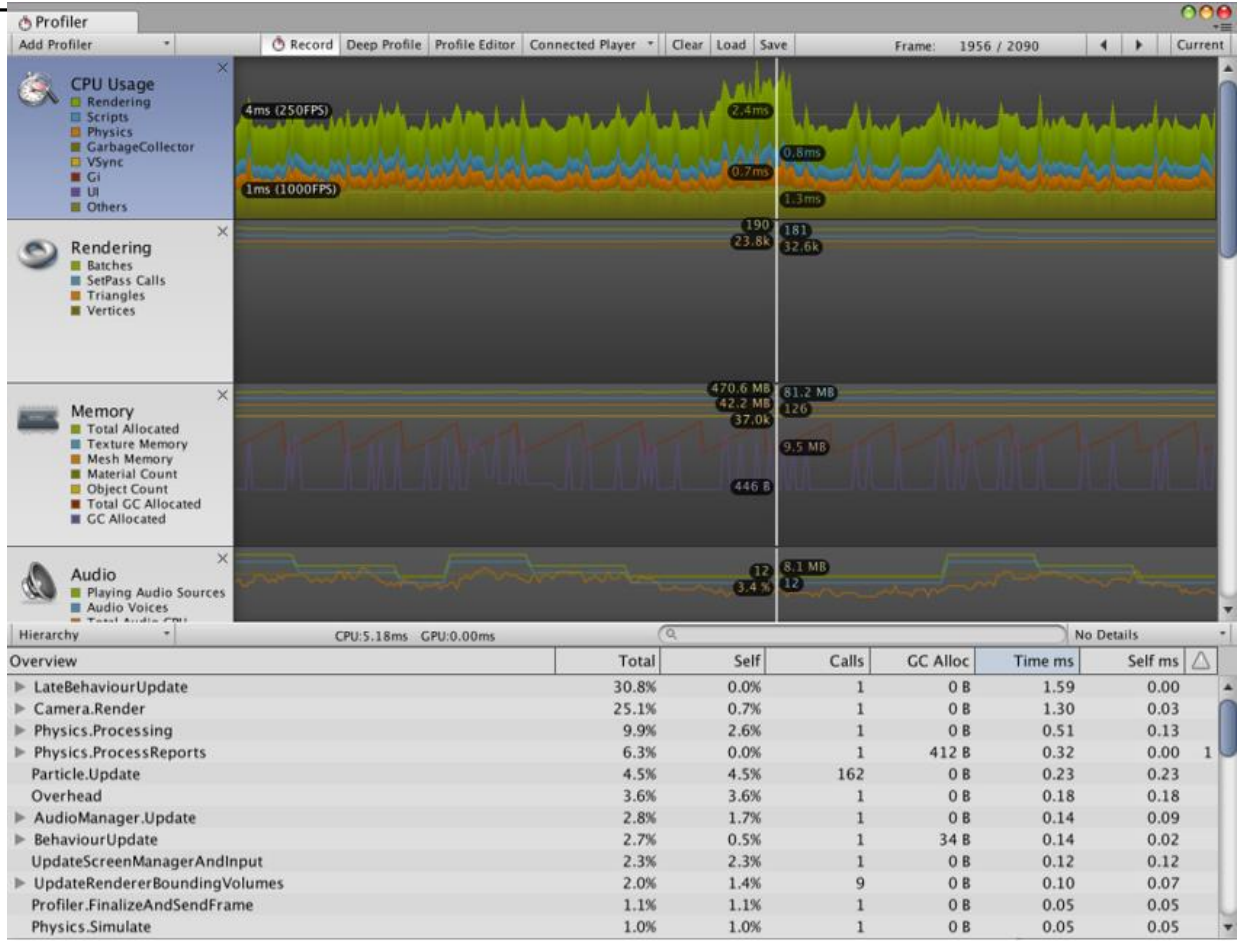
[RTR] Akenine-Moeller, Haines, Hoffman, Real-Time rendering 4th ed., 2018.

Optimalizace pro hry

- Zachování konstantní snímkové frekvence!
- 60FPS ~ 16.7ms / frame
- 90FPS ~ 11.1 ms / frame (VR)
- Nelze ladit pouze na nejnovější GPU
  - Herní konzole – o generaci starší GPU
- Většinu času zabírá rendering!
  - Optimalizovat, optimalizovat, optimalizovat, …
- I jiné části je třeba implementovat efektivně
  - Fyzika / kolize, herní dotazy: prostorové datové struktury
  - Animace: komprimované animační klipy
  - …

Profiling - Detekce úzkého hrdla

- Integrovaný profiler
- C/C++ kompilátor
- Nástroje
  - VTune
  - gprof
  - …



Přehled základních (grafických) optimalizací

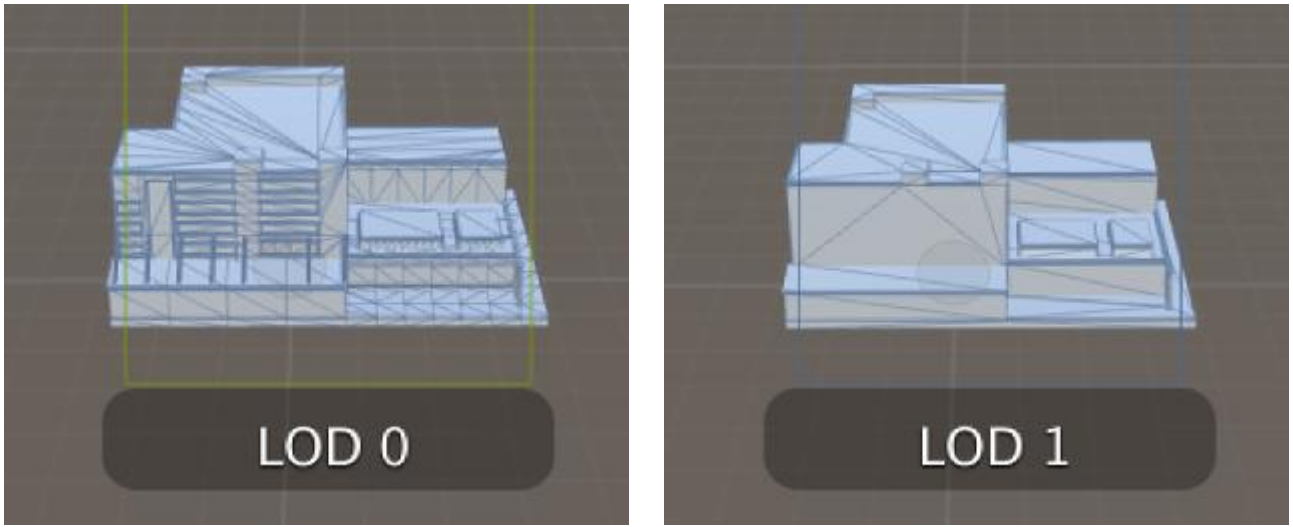
- Optimalizace scény / grafického modelu
- Předpočítání osvětlení
- Redukování podle viditelnosti (visibility culling)

Optimalizace scény

- Cíl: zobrazovat co nejméně dat se zachováním kvality!
- Textury
  - Detaily pomocí textur, bump mapy, normálové mapy, použití MIP-map
  - Atlas textur, vytvoření zobrazovacích dávek (batches)
- Geometrie
  - Efektivní indexace geometrie (indexed triangle list)
- Dělení modelu na bloky / části herního světa
  - Dobré i pro vytváření modelu, spolupráci více modelářů
- Dělení na statickou a dynamickou část modelu
- LOD

LOD (Level-of-Detail)

- Různé úrovně detailu modelu
- Přepínané typicky podle vzdálenosti od kamery

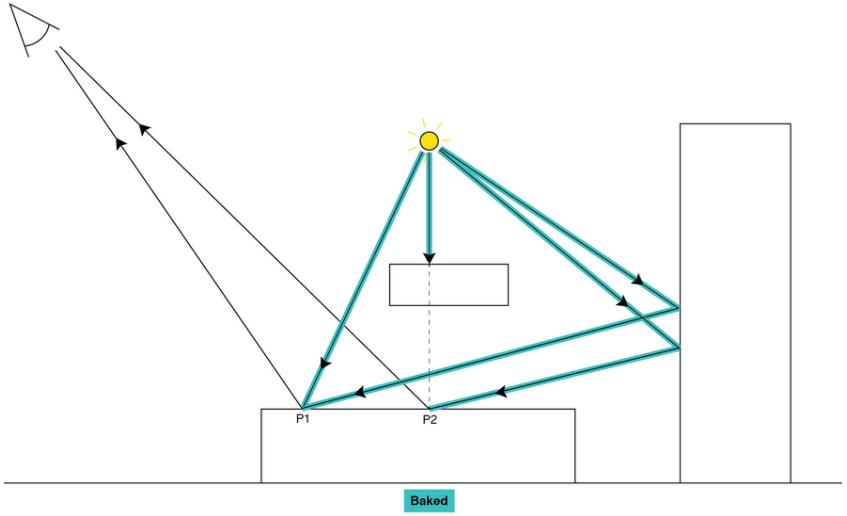


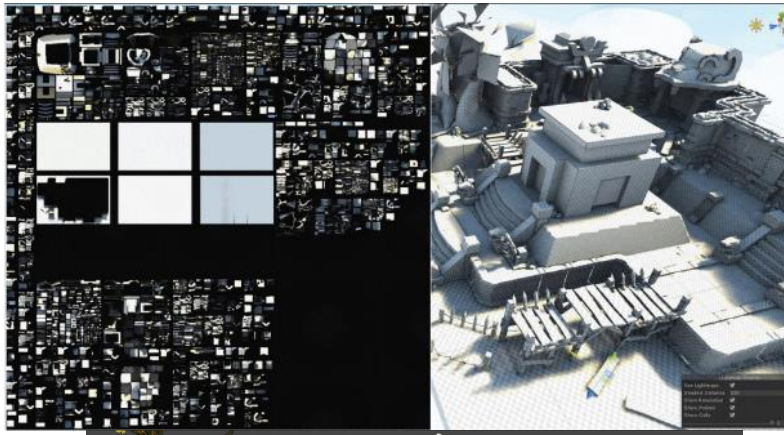
LOD

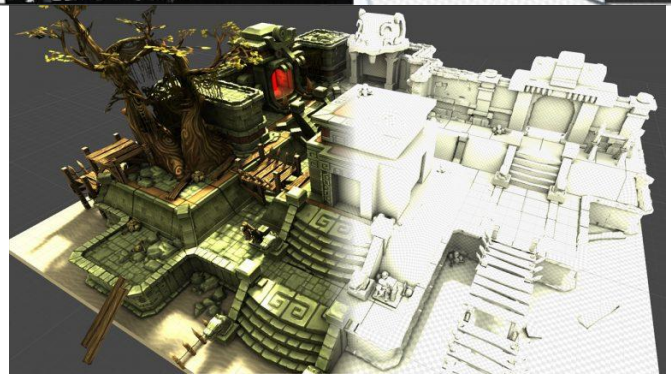
- Diskrétní
  - Několik předpočítaných LOD, které se přepínají
  - Přepínání LOD: skokové, alpha blending, bitová maska průhlednosti
- Spojité
  - Generují se za běhu
  - Plynulá úprava složitosti modelu
  - Výpočetně náročnější
  - Dynamicky generované subdivision surfaces
- Vytváření LOD
  - Eliminace hran, eliminace vrcholů
  - Externí nástroje (MeshLab http://www.meshlab.net/)

Předpočítání osvětlení

- Výpočet globálního osvětlení / statických stínů
- Uložení do map osvětlení (light maps)

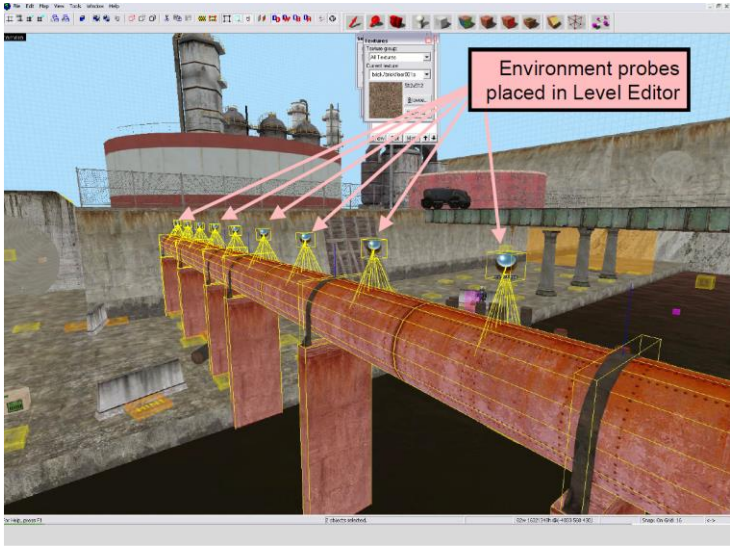






Předpočítání osvětlení

- Do light map není možné uložit osvětlení na lesklých površích!
  - Je pohledově závislé, mění se s pozicí kamery
- Použití předpočítaných (baked) reflection probes



Obsah přednášky

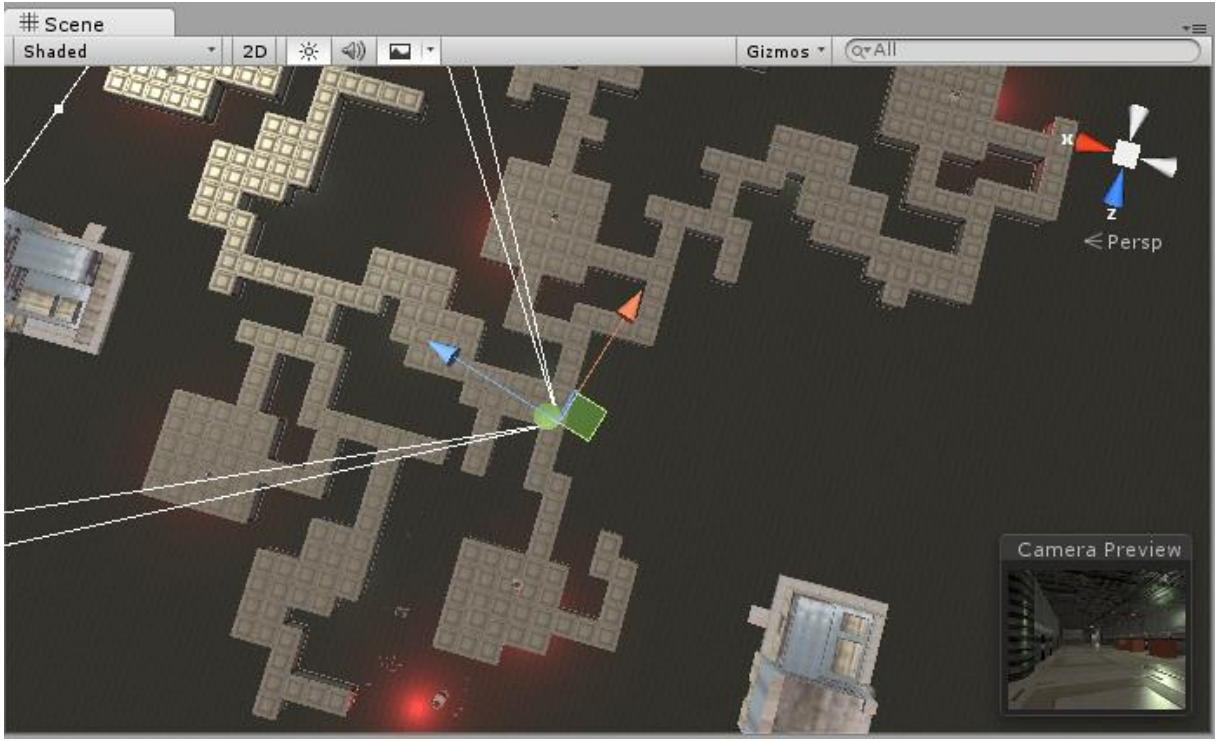
- Optimalice pro hry — GEA 14.1-14.4, RTR 18-20
  - Optimalizace scény
  - LOD
  - Předpočítání osvětlení
  - Redukování podle viditelnosti

[GPP] R. Nystrom. Game Programming Patterns, 2014.

[RTR] Akenine-Moeller, Haines, Hoffman, Real-Time rendering 4th ed., 2018.

Optimalizace zobrazování – Redukování (Culling)

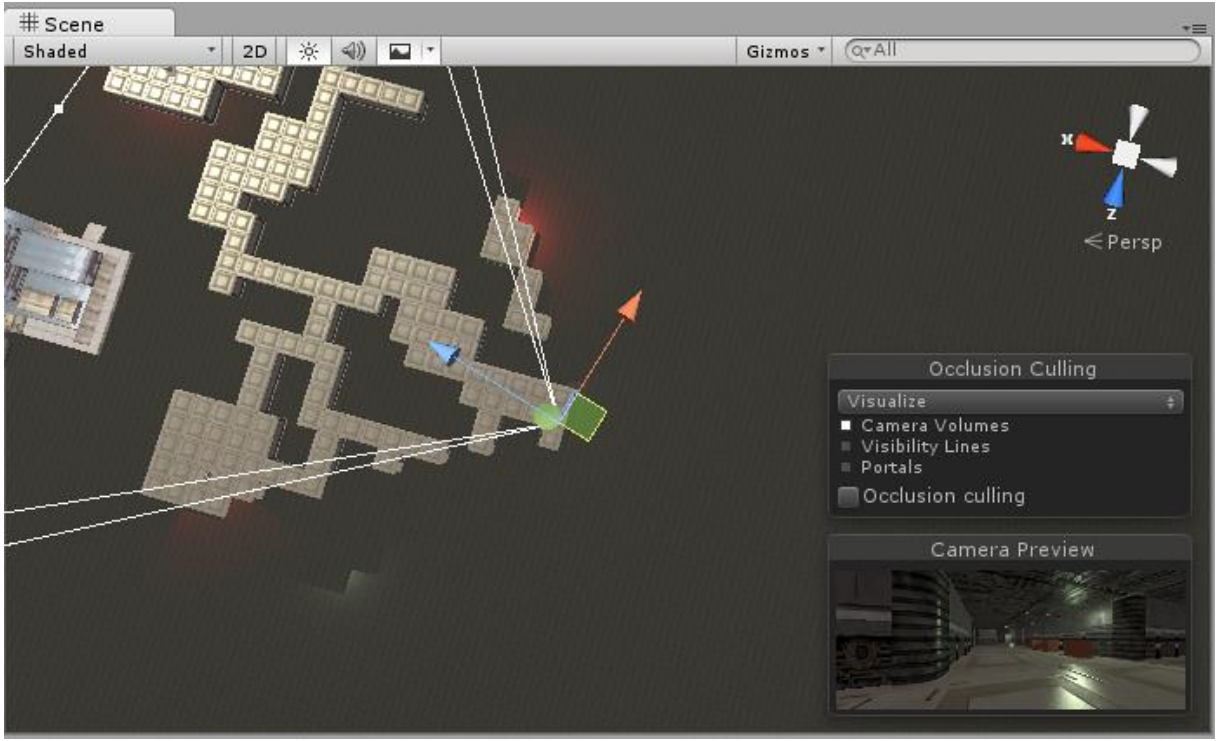
- Zobrazovat jen to co je vidět!



Optimalizace zobrazování – Redukování (Culling)

- Zobrazovat jen to co je vidět!

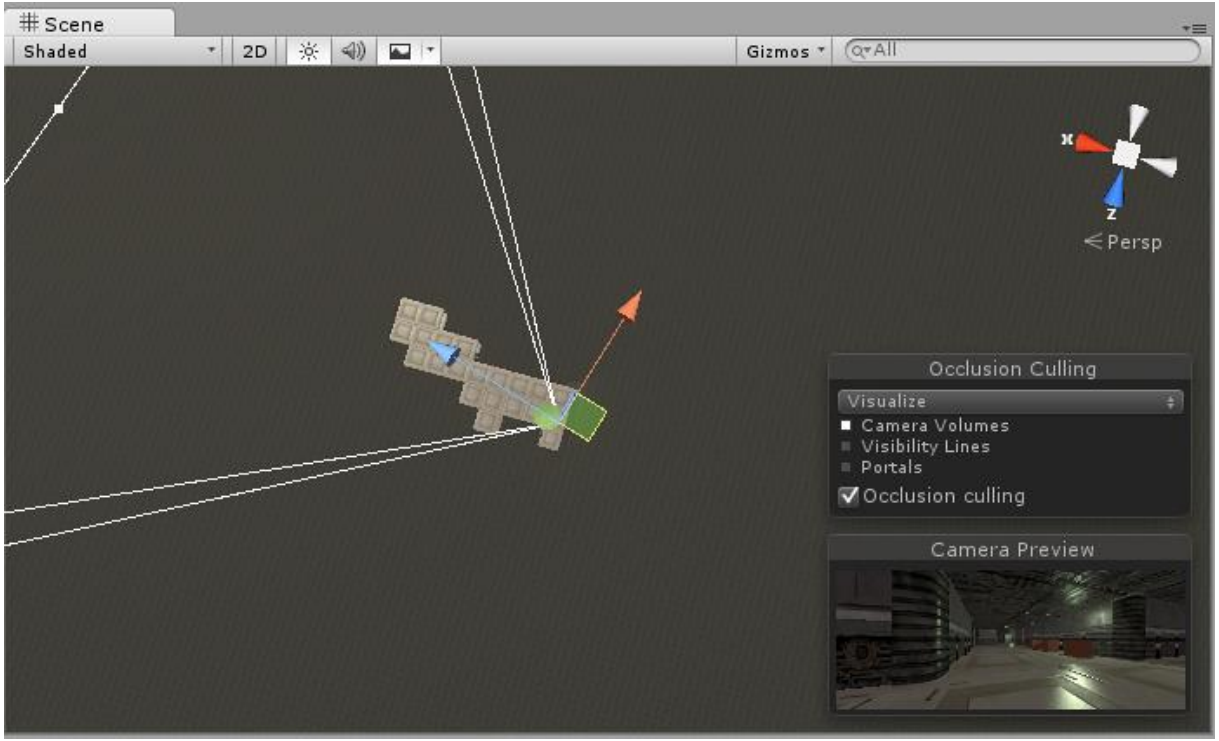
View Frustum Culling



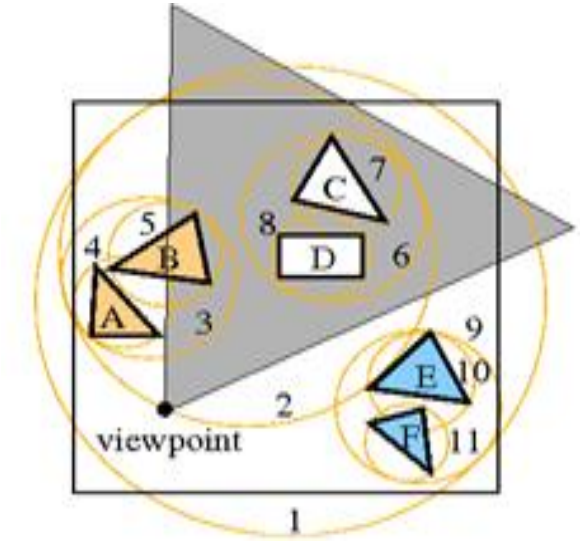
Optimalizace zobrazování – Redukování (Culling)

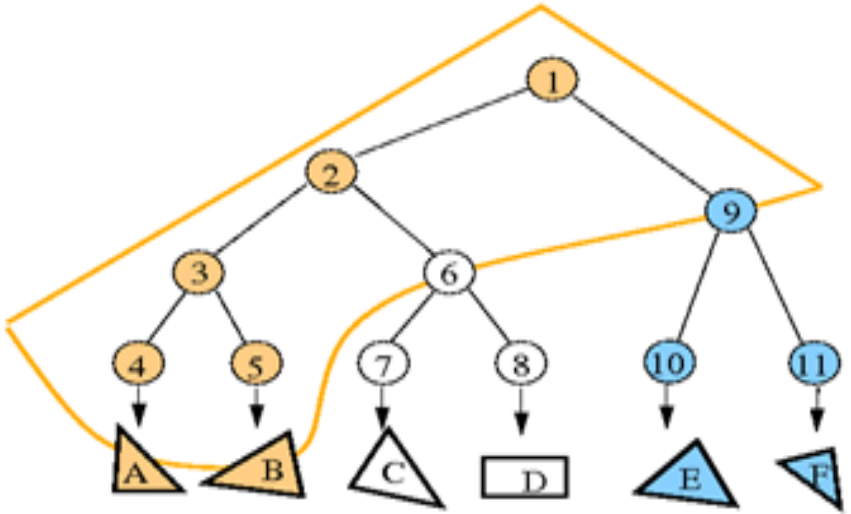
- Zobrazovat jen to co je vidět!

Occlusion Culling



View Frustum Culling – Redukování pohl. jehlanem

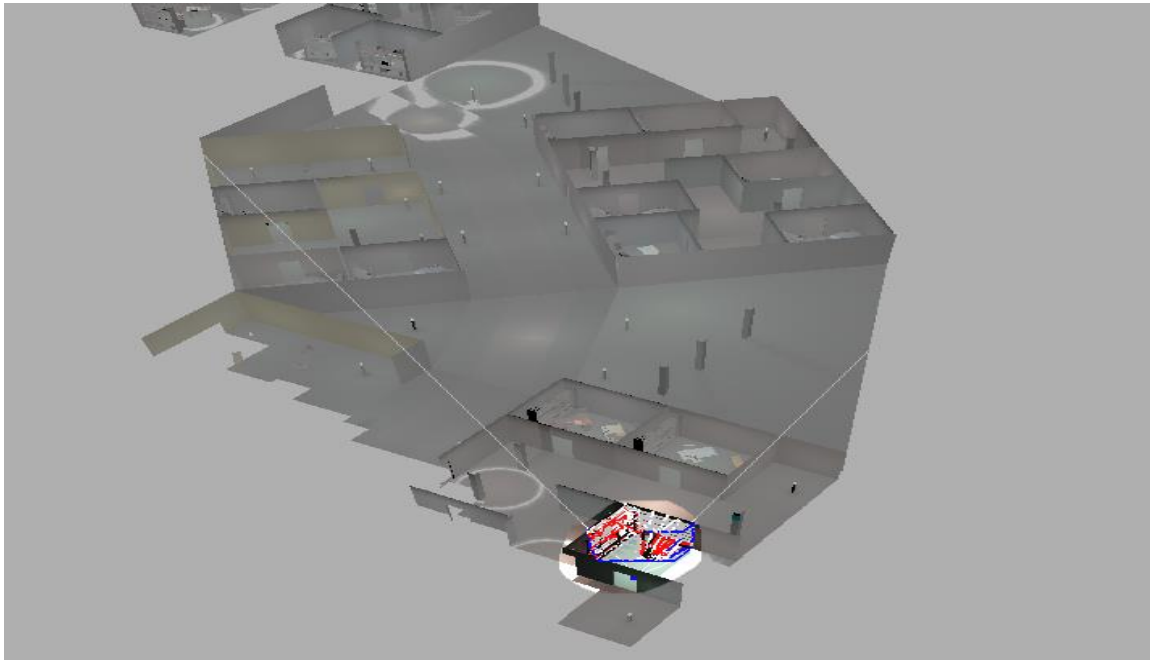




- Výsledek nemusí být přesný
  - Stačí nadmnožina objektů v pohl. jehlanu!
  - Konzervativní algoritmus

Occlusion Culling – Redukování zastíněním

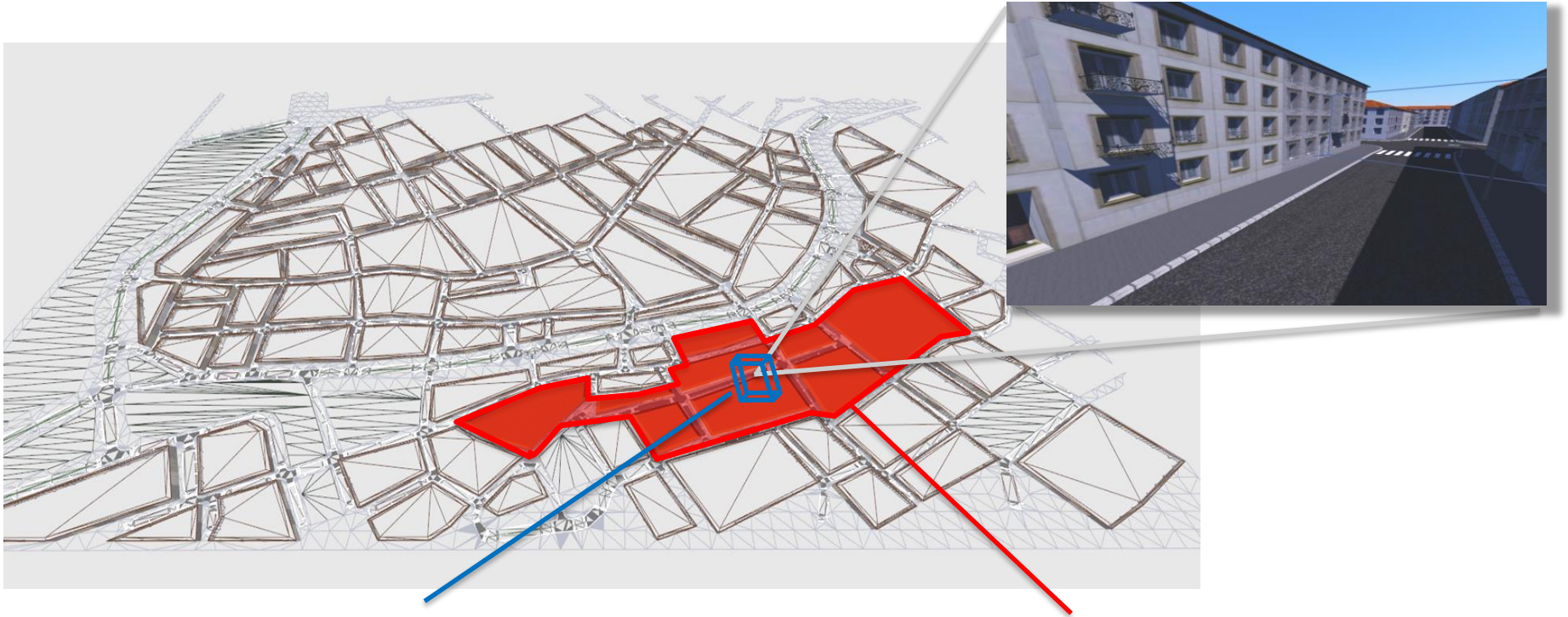
- Offline (předpočítaná viditelnost pro statickou část scény)
- Online (počítáno v reálném čase pro každou pozici kamery)



Offline Occlusion Culling

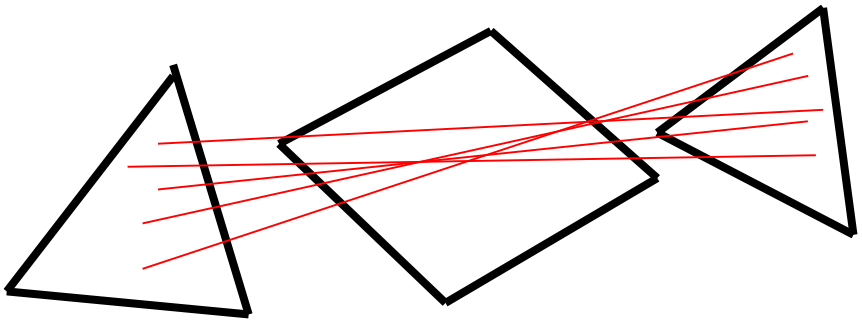
- Předzpracování
  - Rozděl pohledový prostor na pohledové buňky (Unity: occlusion area)
  - Pro každou buňku vypočti potenciálně viditelnou množinu objektů (PVS)
  - Vyřeší viditelnost “offline” pro všechny možné pozice kamery
- Použití
  1. Najdi pohledovou buňku (lokace bodu)
  2. Zobraz asociovanou PVS

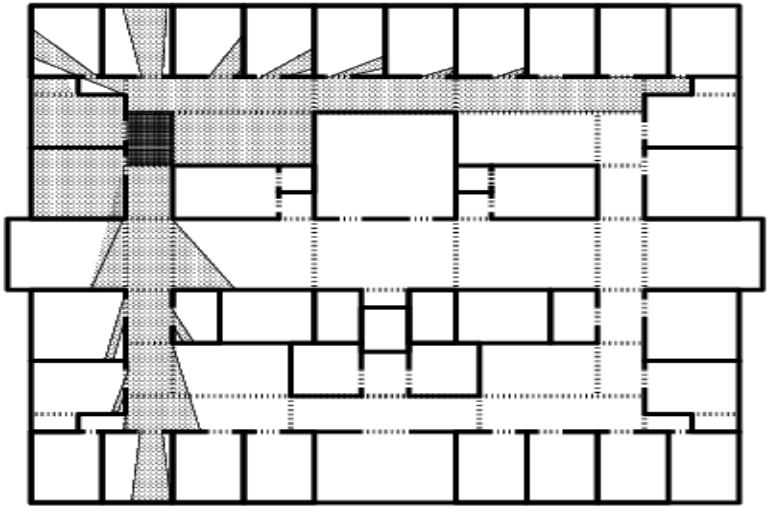
Příklad



Interiéry – Speciální algoritmy pro výpočet PVS

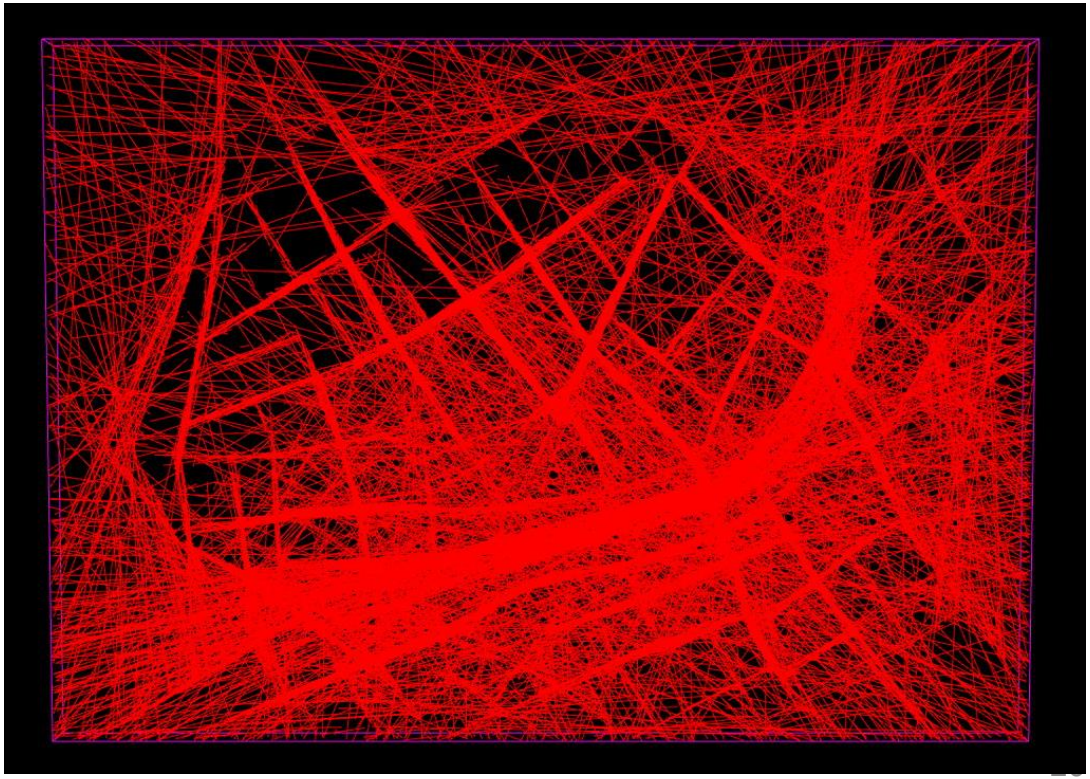
- Rozděl na buňky a portály
- Vytvoř graf sousednosti
- Omezené prohledávání grafu
  - Viditelnost skrz sekvenci portálů
  - Vzorkování i analytické algoritmy





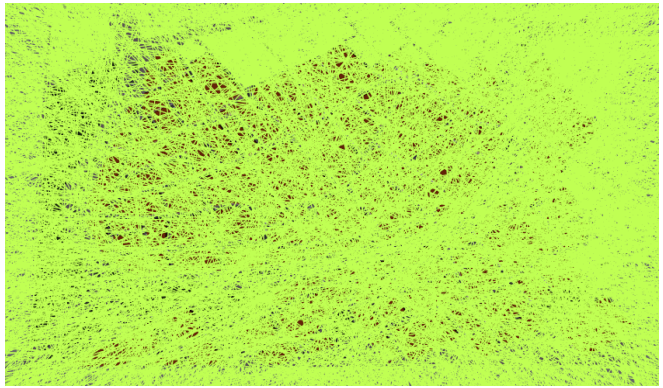
Obecné scény

- Výpočet PVS je velmi náročný
  - Efektivní vzorkování
  - Analytické řešení



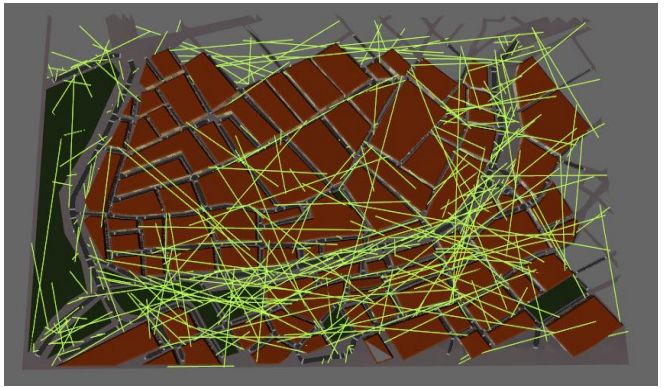
Adaptivní vzorkování pro výpočet PVS

stationary distribution



many samples
no new PVS entries

adaptive sampling

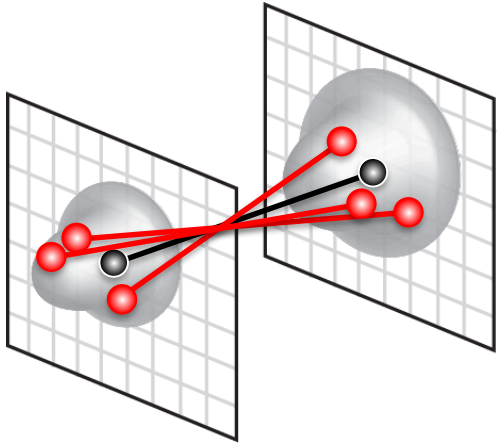


a few samples
new PVS entries!

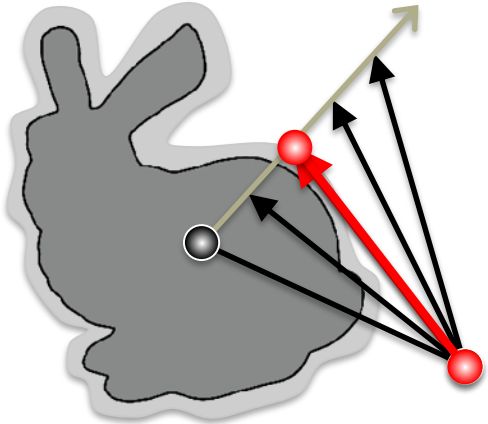
[Adaptive Global Visibility Sampling (2009)]

Adaptivní vzorkování pro výpočet PVS

- Směs adaptivních vzorkovacích distribucí



two point mutation



silhouette mutation

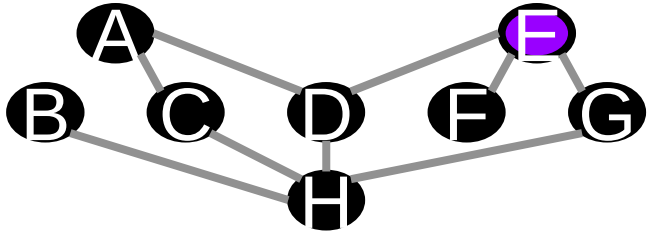
- **100x urychlení** ve srovnání s rovnoměrným vzorkováním

Online occlusion culling

- Výpočet PVS pro každý snímek
- Jednodušší
  - Známe pozici kamery
  - Ale musí být velmi rychlé!
- Rychlá SW rasterizace
  - Malé rozlišení, použití předpřipravených objektů (occluders)
- Interiérové scény
  - Cells - portals
- GPU metody
  - HW dotazy zastínění
  - Hierarchický z-buffer
  - Warping hloubkového bufferu z minulého snímku

Interierové scény: Buňky a portály

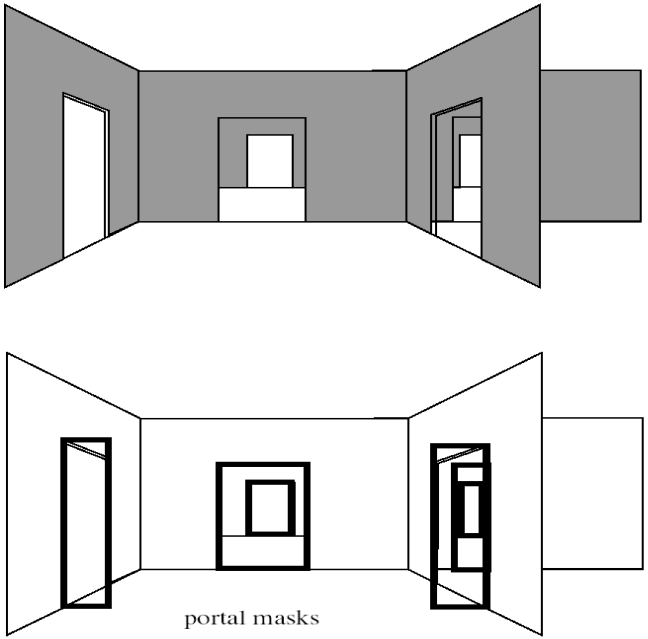
- Graf sousednosti
  - Buňky ~ místnosti
  - Portály ~ dveře & okna

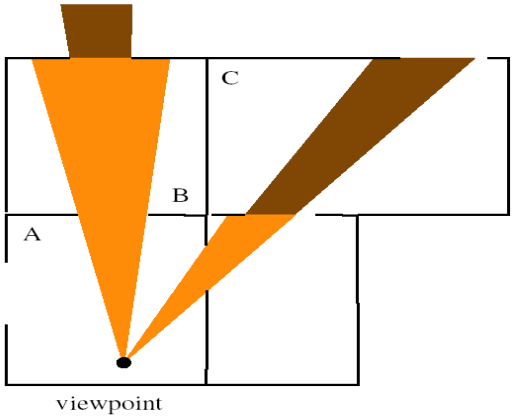


- Omezené prohledávání
  - Test viditelnosti portálů [Luebke 96]

Test viditelnosti portálů

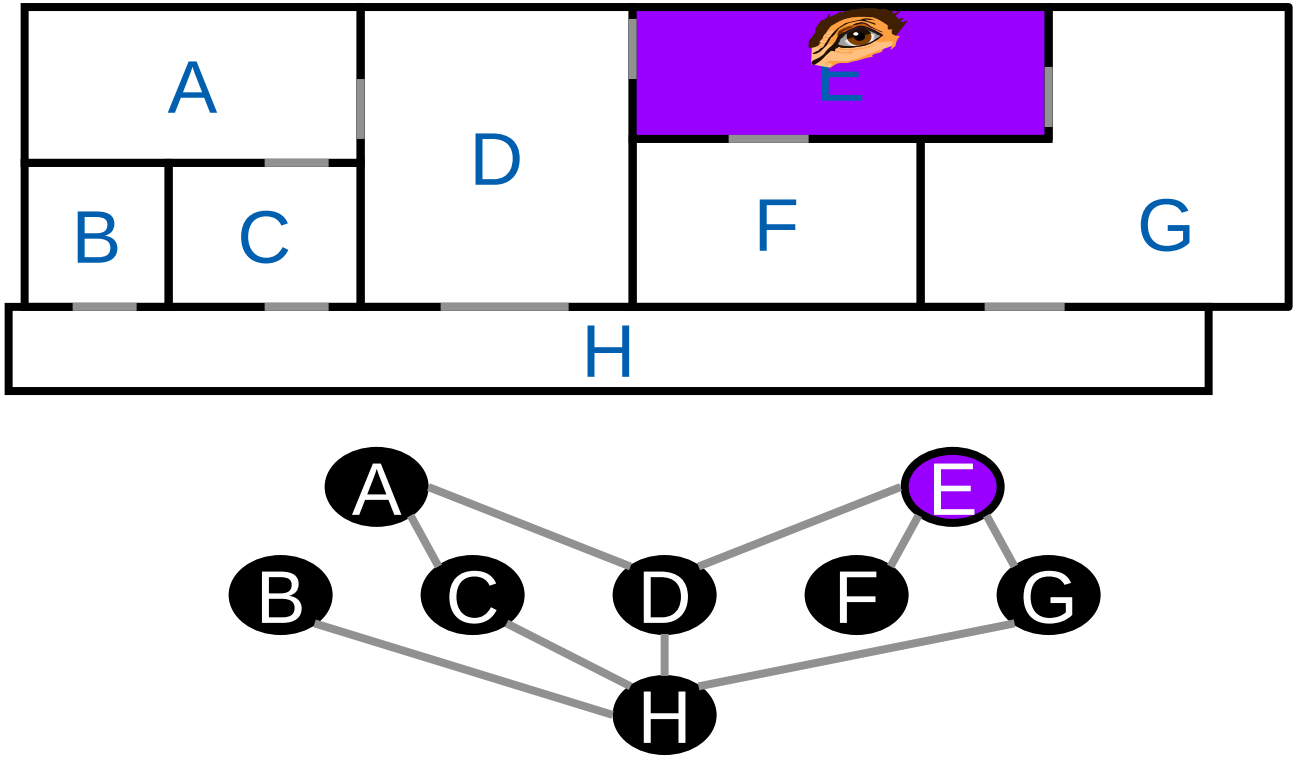
- Průnik obalových obdélníků portálů





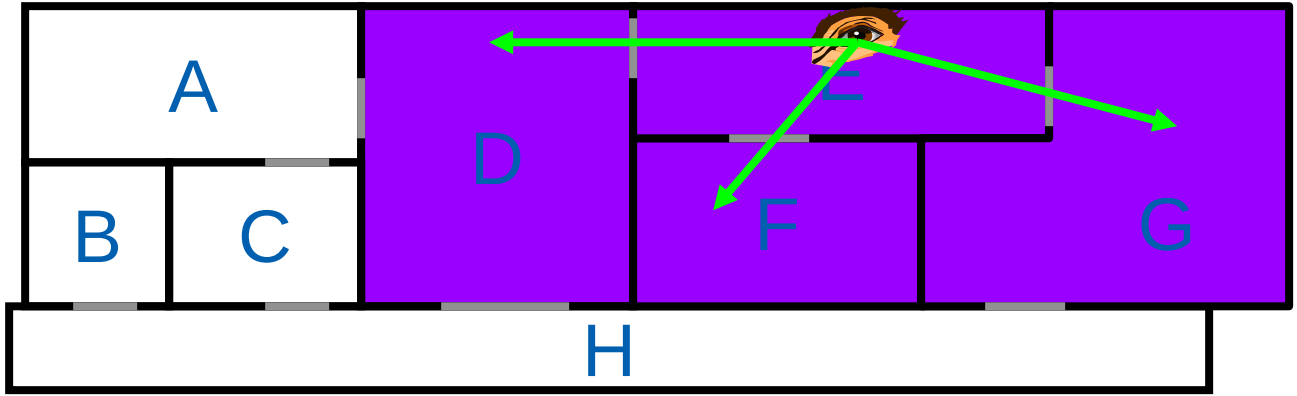
Cells and Portals Example

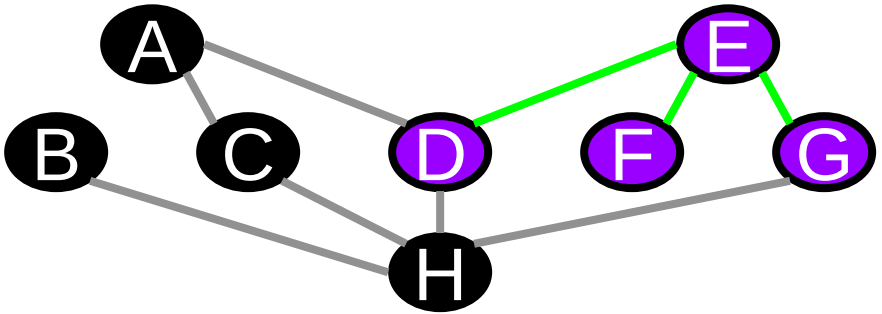
- Viewpoint in cell E



Cells and Portals - Example

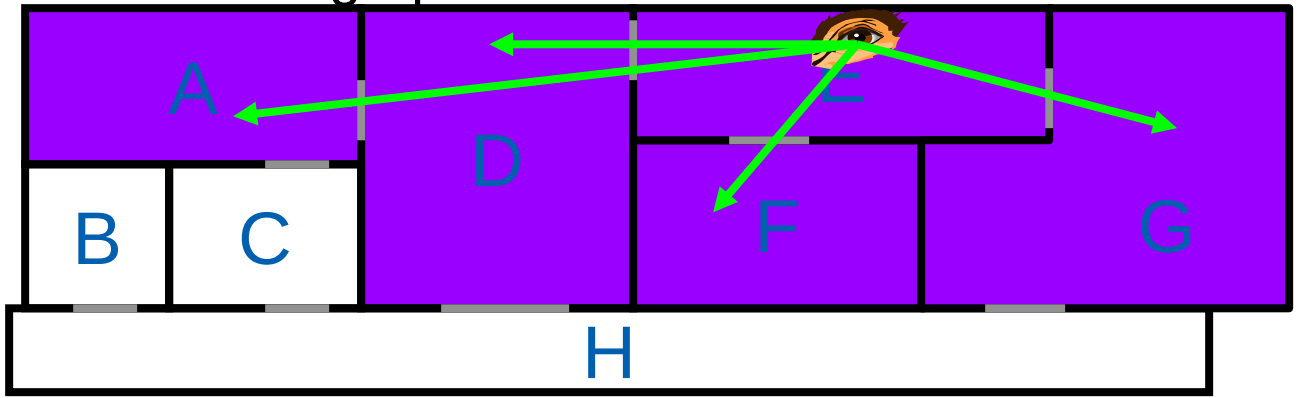
- Adjacent cells DFG

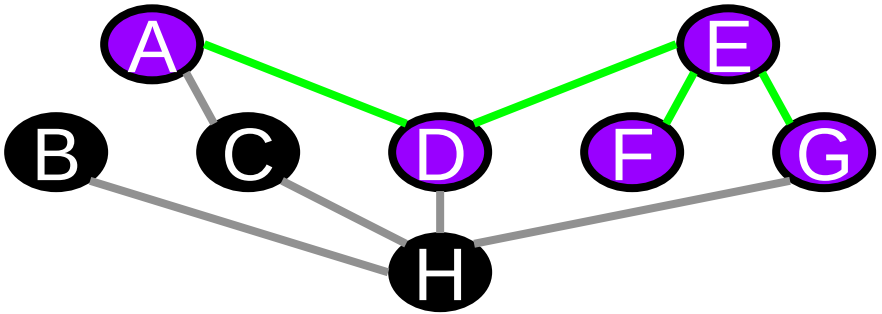




Cells and Portals - Example

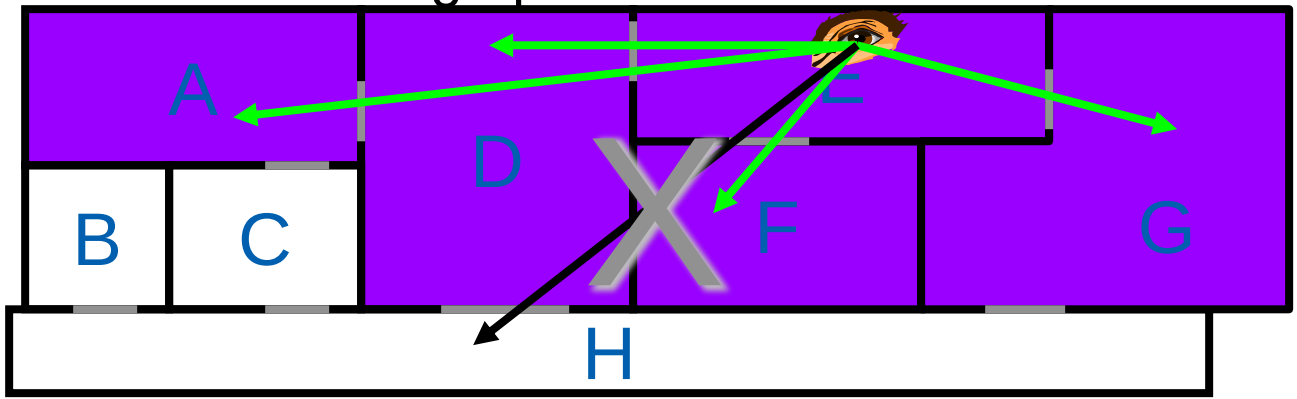
- Cell A visible through portals E/D+D/A

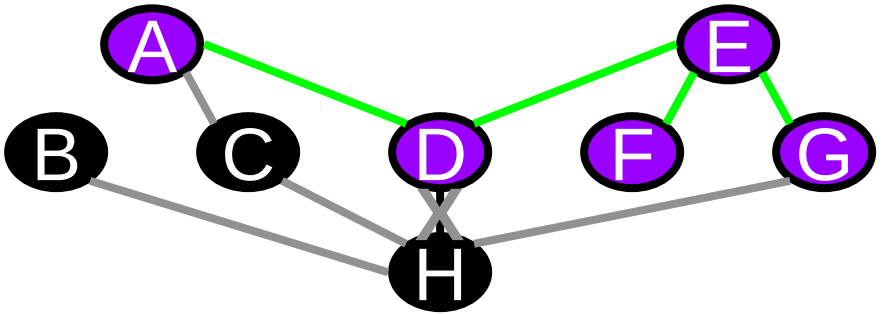




Cells and Portals - Example

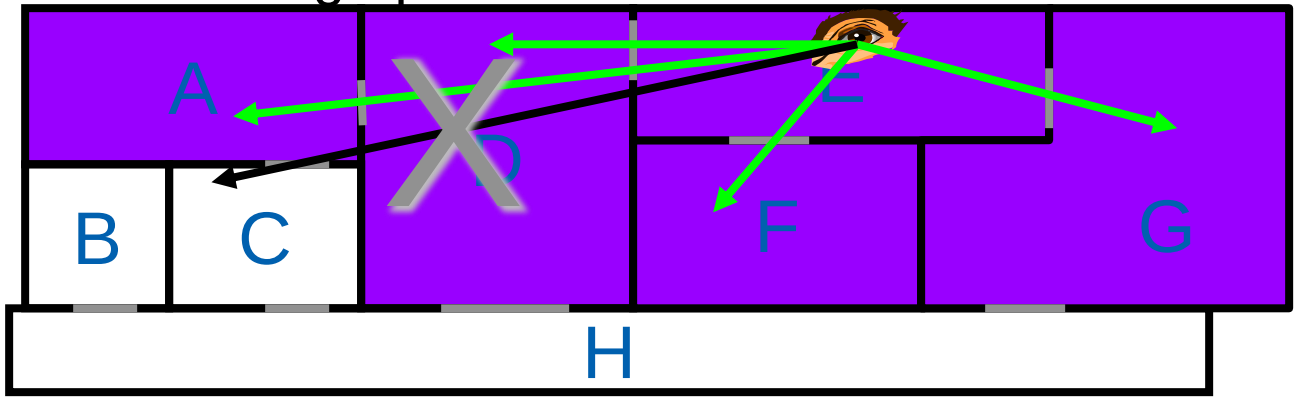
- Cell H not visible through portals E/D+D/H

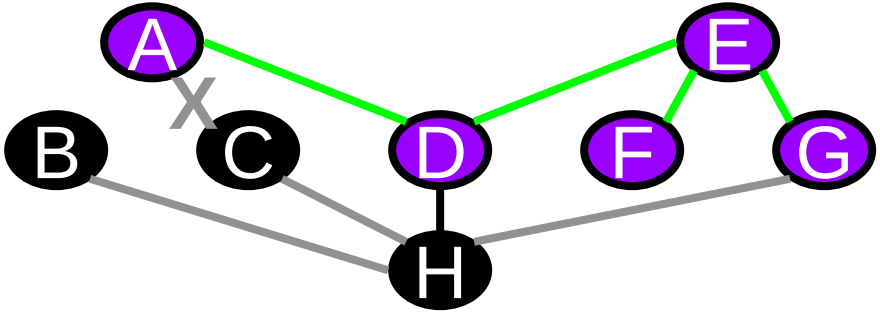




Cells and Portals - Example

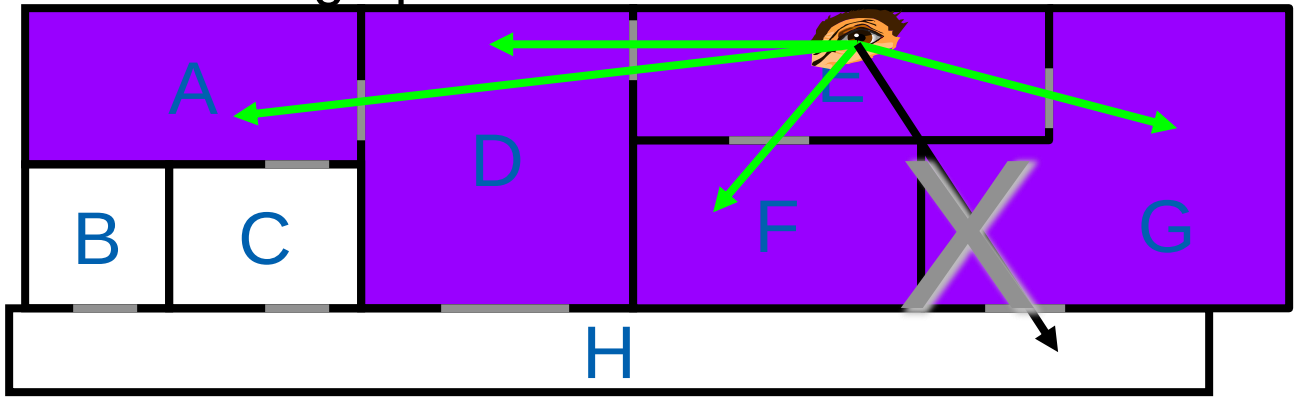
- C not visible through portals E/D+D/A+A/C

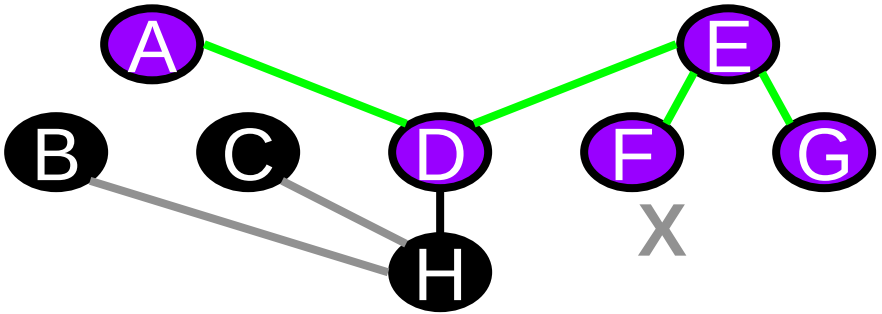




Cells and Portals - Example

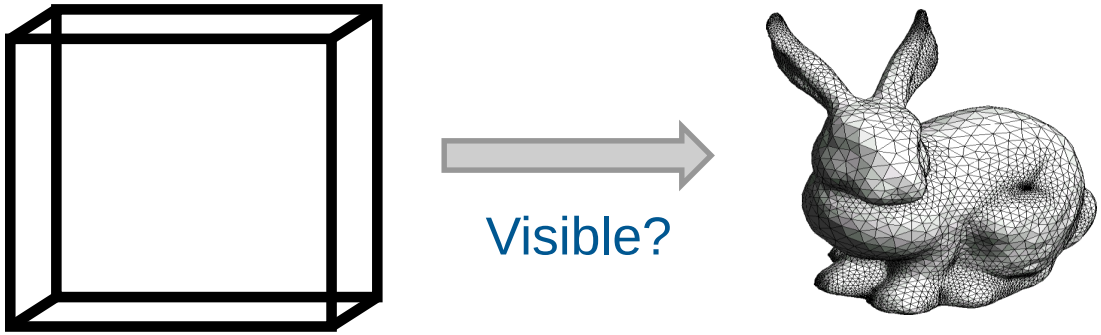
- H not visible through portals E/G+G/H





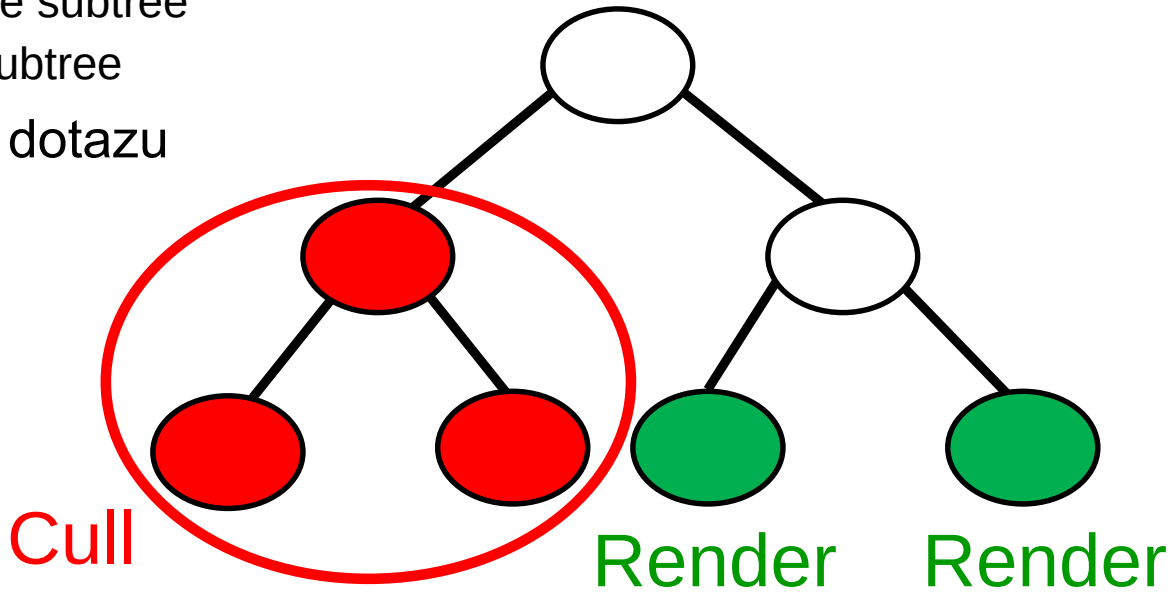
HW dotazy zastínění

- ARB_occlusion_query, NV_occlusion_query
- Vrací počet fragmentů, které prošly hloubkovým testem
- + libovolný typ scény, žádné předzpracování
- - Zpoždění, dotaz stojí čas

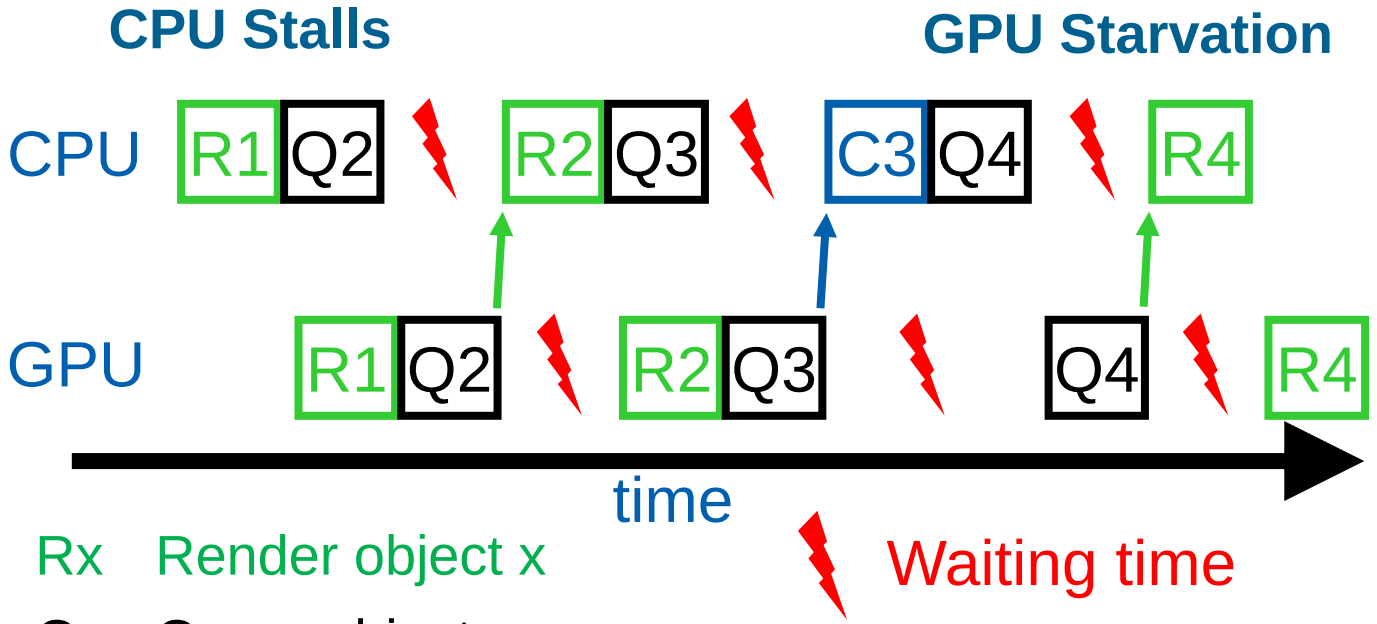


Naivní metoda: Hierarchický Stop & Wait

- Pro každý uzel: Aktivuj dotaz
  - Visible → traverse subtree
  - Invisible → cull subtree
- Problém: Zpoždění dotazu
  - CPU stalls
  - GPU starvation

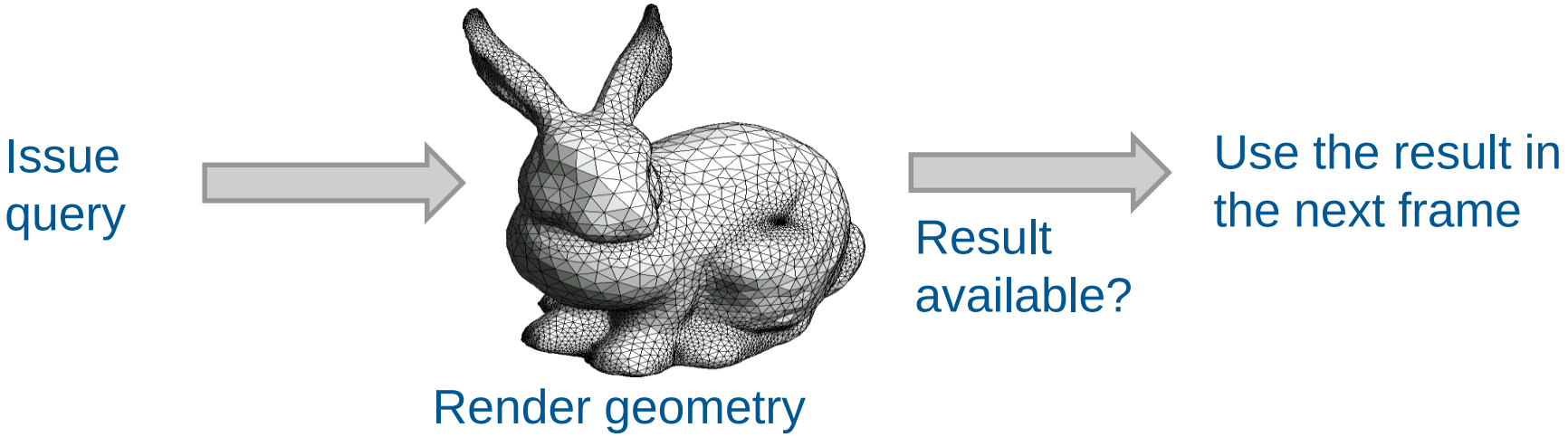


Hierarchický Stop & Wait



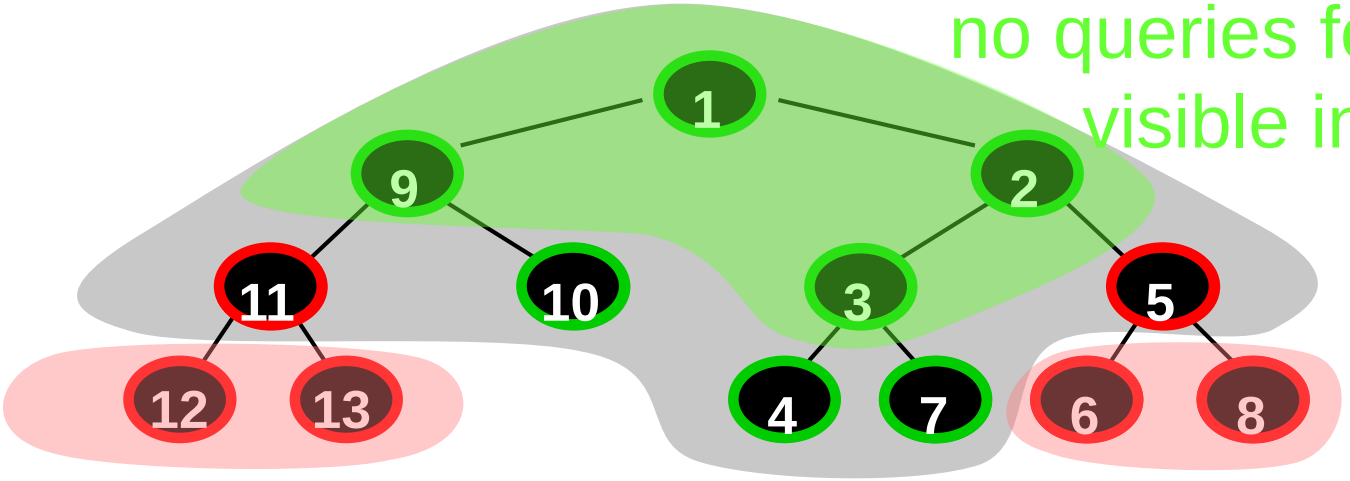
Coherent Hierarchical Culling (CHC)

- Při čekání na výsledky → traverzuj a zobrazuj
- Využij koherenci, předpokládej, že uzly nemění viditelnost
- Pro viditelné uzly z minulého snímky
  - Nečekej na výsledky dotazů



Coherent Hierarchical Culling

- Prokládání dotazů a zobrazování
- Plánuj pořadí dotazů na základě časové koherence

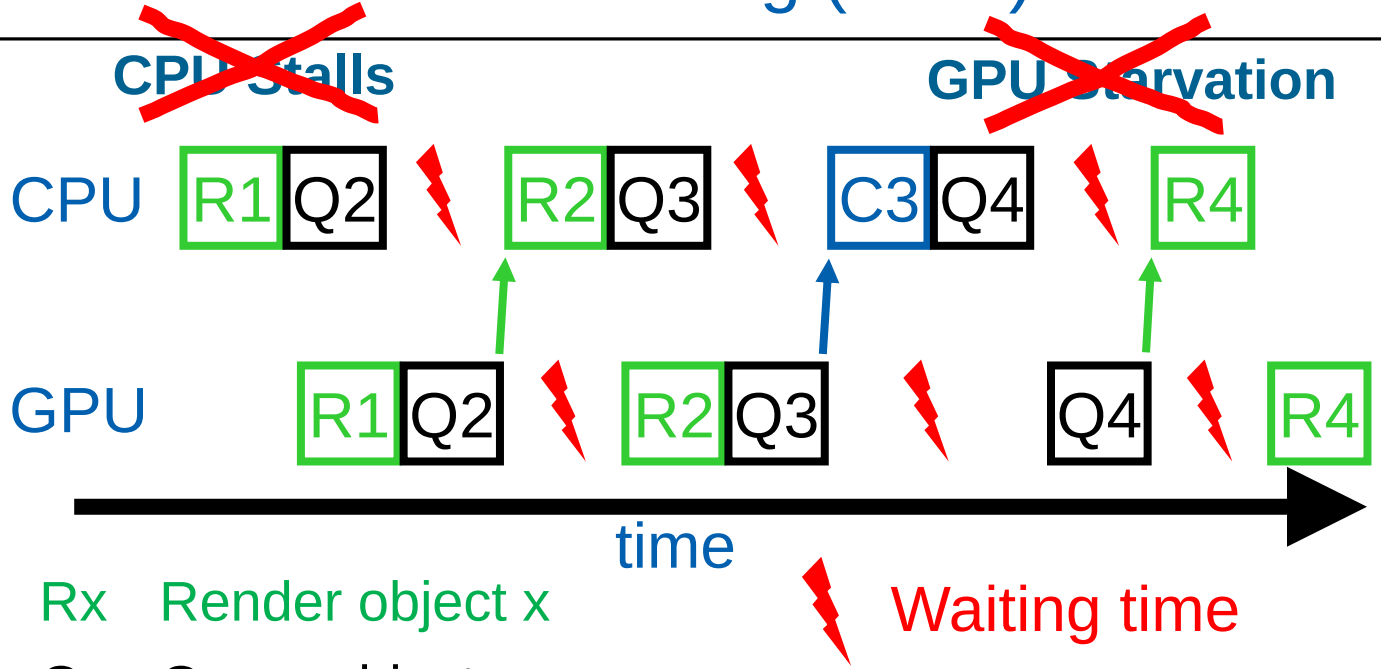


no queries for previously visible interior nodes

query prev. invisible nodes + leaves

Prev. invisible nodes:
queries depend on parents

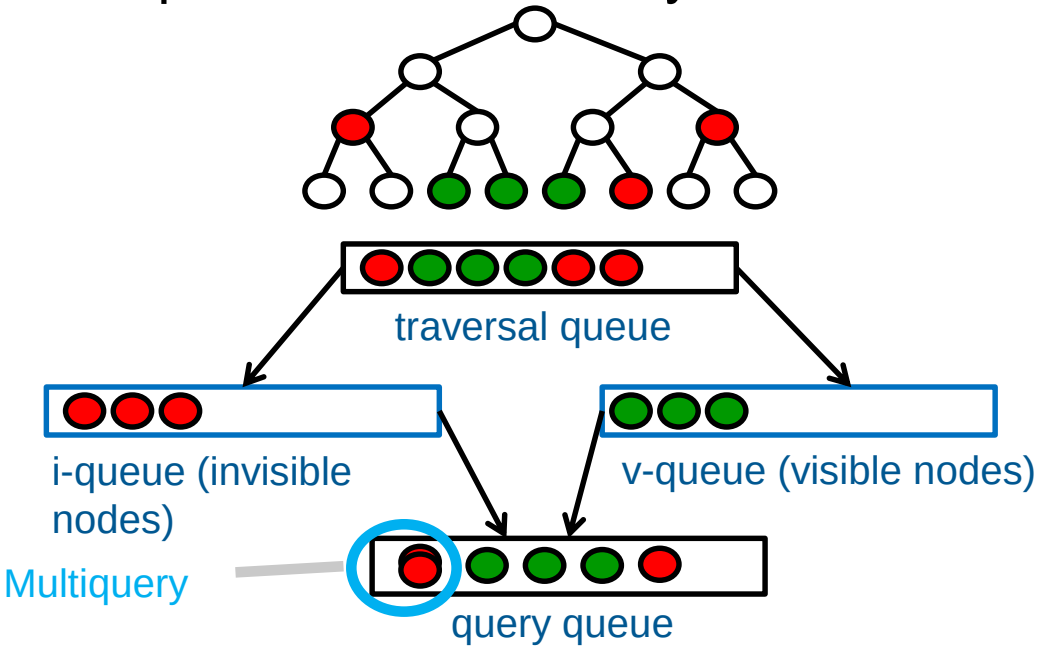
Coherent Hierarchical Culling (CHC)



[Coherent Hierarchical Culling: Hardware Occlusion Queries Made Usefull (2004)]

CHC++

- Minimalizace stavových změn a počtu dotazů
- Batching, multi-queries, těsné obálky



[CHC++: Coherent Hierarchical Culling Revisited (2008)]

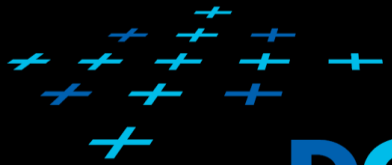
Obsah přednášky

- Optimalice pro hry — GEA 14.1-14.4, RTR 18-20
  - Optimalizace scény
  - LOD
  - Předpočítání osvětlení
  - Redukování podle viditelnosti

[RTR] Akenine-Moeller, Haines, Hoffman, Real-Time rendering 4th ed., 2018.

Ohlédnutí za přednáškami

- Úvod do herního vývoje – základy herního designu
- Komponenty herního enginu
- Geometrie, transformace (→ PGR)
- Reprezentace scény (→ VGO, *KMA*, *MVR)*
- Animace (→ VGO, *MVR)*
- Fyzika / Kolize
- Hudba a Audio
- AI (→ ZUI)
- Shadery, textury (→ PGR)
- *GUI (IUR)*



DCGI

KATEDRA POČÍTAČOVÉ GRAFIKY A INTERAKCE

Otázky?