A composite image featuring soldiers in a tank. The scene is overlaid with a white wireframe mesh that outlines the tank's structure and the soldiers' forms. The background is a blurred, green-tinted street scene. The text is centered on a dark, semi-transparent rectangular background.

# Počítačová grafika pro videohry aneb ze života pixelu

Matematické problémy nematematiků



Martin Kahoun

9. 10. 2024



## Martin Kahoun

[martin.kahoun\[at\]bistudio.com](mailto:martin.kahoun@bistudio.com)

- Senior render programátor @ Bohemia Interactive
- Donedávna výuka na MFF UK:
  - Počítačová grafika pro vývoj her (přednášky)
  - Realtime grafika na GPU (cvičení)

THUS, FOR ANY NONDETERMINISTIC TURING MACHINE  $M$  THAT RUNS IN SOME POLYNOMIAL TIME  $p(n)$ , WE CAN DEVISE AN ALGORITHM THAT TAKES AN INPUT  $w$  OF LENGTH  $n$  AND PRODUCES  $E_{M,w}$ . THE RUNNING TIME IS  $O(p^2(n))$  ON A MULTITAPE DETERMINISTIC TURING MACHINE AND...


WTF, MAN. I JUST WANTED TO LEARN HOW TO PROGRAM VIDEO GAMES.

SIPSER CH7  
 $y_{i,j-1,0} \wedge y_{i,j,0} \wedge y_{i,i+1,0} \wedge y_{i,i,j-1}$   
 $y_{i,j-1,0} \wedge y_{i,j,0} \wedge y_{i,i+1,0} \wedge y_{i,i,j}$   
 $N_i = (A_{i,0} \vee B_{i,0}) \wedge (A_{i,1} \vee B_{i,1}) \wedge \dots \wedge$   
 $N = N_0 \wedge N_1$

金吾博

# Obsah

- 1 Úvod
- 2 Rendering 101
- 3 Příprava scény
- 4 Pitva snímku na GPU — GTA V (PC, 2015)
- 5 Závěr



# Rendering 101

# Pohyblivé obrázky

- Kolik snímků za vteřinu pro iluzi pohybu?

# Pohyblivé obrázky

- Kolik snímků za vteřinu pro iluzi pohybu?
- Filmy: 24–25 fps

# Pohyblivé obrázky

- Kolik snímků za vteřinu pro iluzi pohybu?
- Filmy: 24–25 fps
- Hry: 30 fps, i.e., 33 ms / snímek
- Hry: 60 fps, i.e., 16 ms / snímek

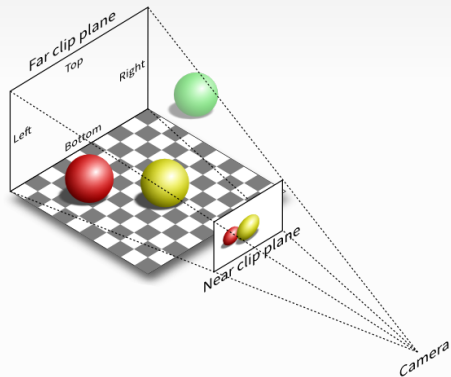


# Pohyblivé obrázky

- Kolik snímků za vteřinu pro iluzi pohybu?
- Filmy: 24–25 fps
- Hry: 30 fps, i.e., 33 ms / snímek
- Hry: 60 fps, i.e., 16 ms / snímek
- VR: min 90 fps pro *jedno oko*, i.e. <11 ms / snímek

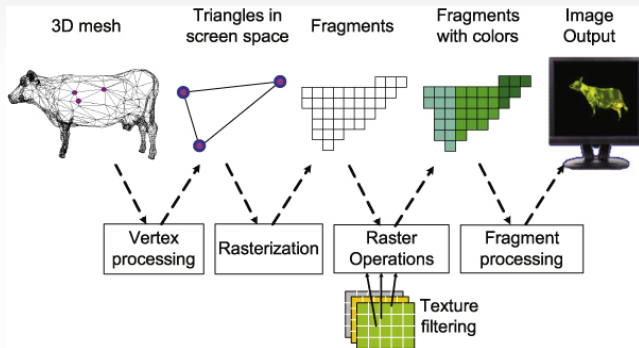


# View frustum a projekce

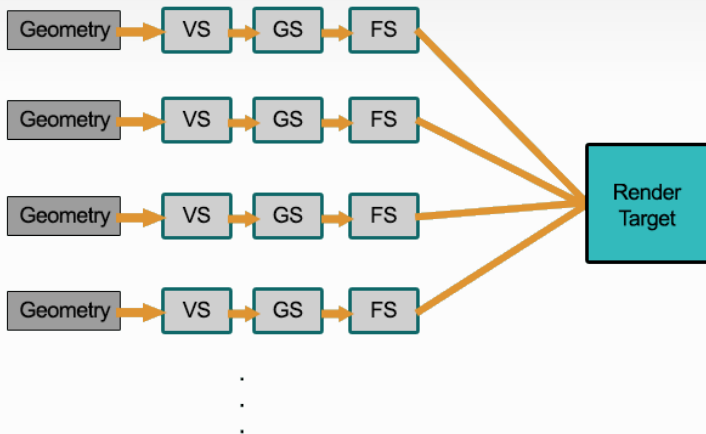


- Komolý jehlan
- Pozice kamery
- Směr pohledu
- 2 ořezové roviny
- FOV (vertikální úhel)
- Typicky 2 matice 4x4

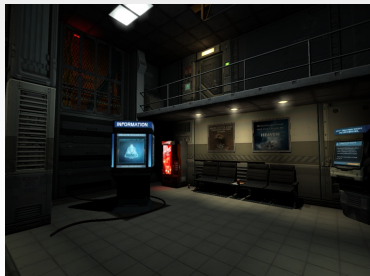
# Ze života pixelu



# Forward rendering



# Forward rendering



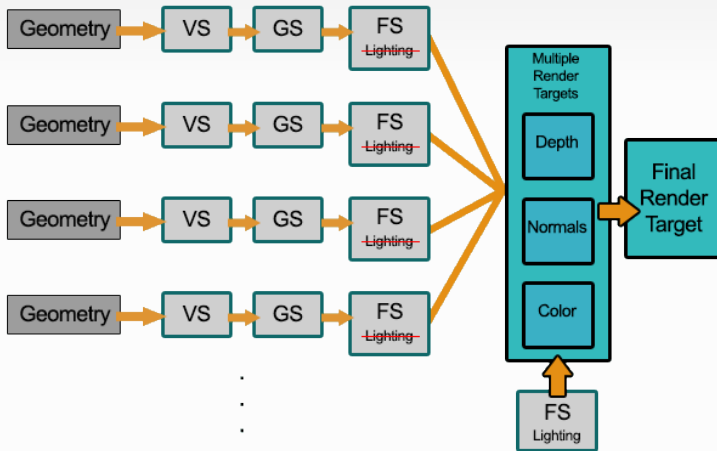
Doom 3 (2004)  
13 passes: depth, 11 lights, ambient



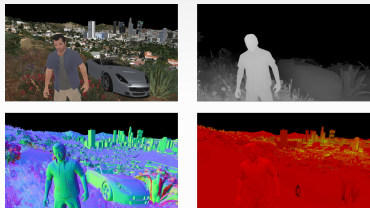
Doom (2016)

- Pro každý pixel spočítej světlo (FS)
- Přímočará implementace
- Jednoduše řešená průhlednost
- Jednoduchý anti-aliasing (MSAA)
- Problémy:
  - Velké množství světel (→ *clustering*)
  - Mnoho průhledné geometrie
  - Složitější postprocesy

# Deferred rendering



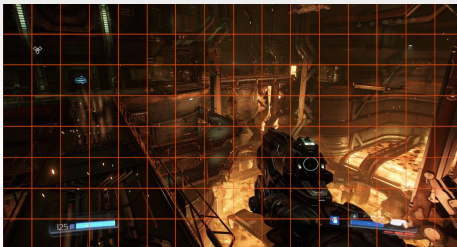
# Deferred rendering



GTA V (PC, 2015)  
G-buffer & final image

- Oddělení geometrie od nasvětlení
- Nejprve kreslí scénu do G-bufferu
- Fullscreen pass pro výpočet osvětlení
- Kreslení jednotlivých světel / *clustering*
- Problémy:
  - Propustnost paměti grafické karty
  - Netriviální průhledná geometrie (→ *forward*)
  - Anti-aliasing vyžaduje zvláštní péči

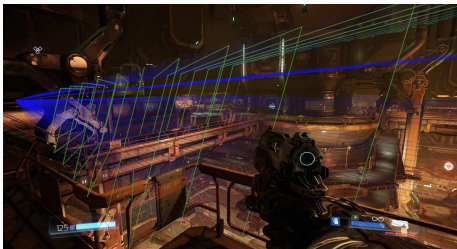
# Clustering



16x8 dílků

Původní myšlenka pro deferred:

- Rozděl obraz na 16x8 dílků
- Ulož světla pro každý dílek
- Během renderu iteruj tato světla



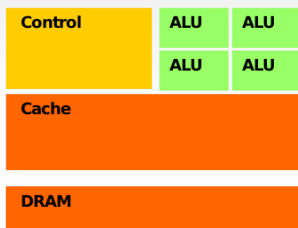
24 logaritických řezů hloubky

Rozšíření (nejen) pro forward:

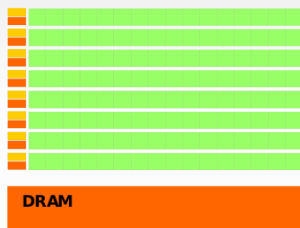
- Voxelizuj pohledový jehlan
- → 16x8x24 froxelů (*clusters*)
- Pro každý generuj seznam světel
- Během renderu iteruj tato světla
- Známé též jako *Clustered Forward+*



# CPU vs. GPU



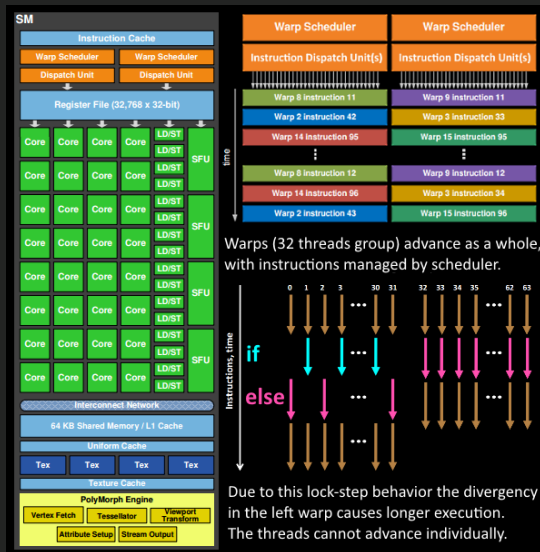
**CPU**



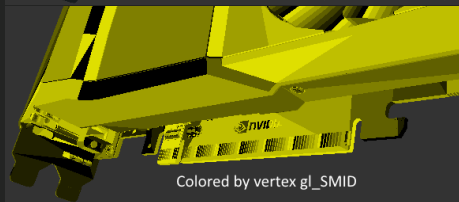
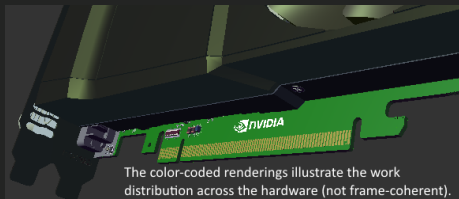
**GPU**

- GPU zpracuje miliony polygonů za frame
- GPU uzpůsobené pro paralelní výpočty (SIMD)
- Drahé větvení v kódu nebo změny stavů (RT, shader, ...)
- Asynchronní vůči CPU i běžícím výpočtům (klasický zdroj chyb)

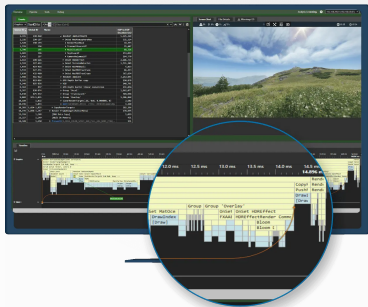
# Architektura NVIDIA Maxwell (GTX 9xx)



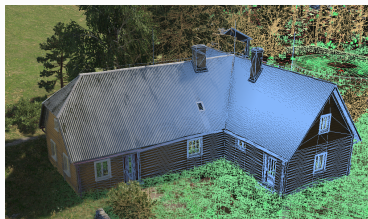
# Architektura NVIDIA Maxwell (GTX 9xx)



# Klíčové oblasti pro výkon



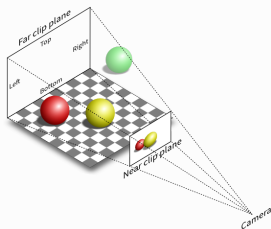
- Kreslit jen to, co je vidět
- Kreslit jen rozumně velké polygony
- Minimalizovat tzv. *pixel overdraw*
- Neblokovat GPU (např. čtení zpět na CPU)
- Optimalizovat využití paměti
- Minimalizovat změny stavu GPU
- Minimalizovat počet *shaderů*
- Minimalizovat větvení v *shaderech*
- Minimalizovat počet instrukcí v *shaderech*



A scenic landscape featuring a large mountain range in the background under a blue sky with scattered white clouds. In the middle ground, there is a farm with several buildings, including a large white house with a dark roof and a smaller wooden structure. A wooden fence runs across the foreground, and a grassy field is visible. The overall scene is bright and clear.

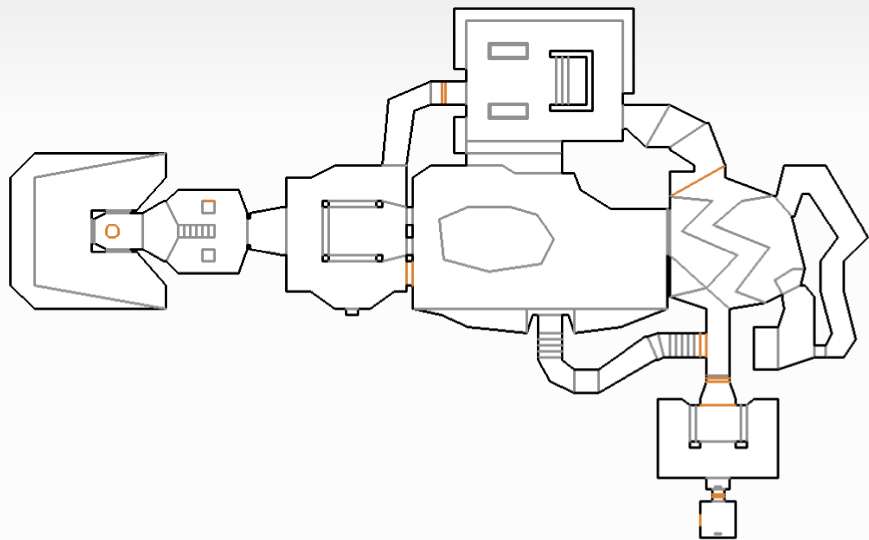
## Příprava scény

# Ořezávání scény

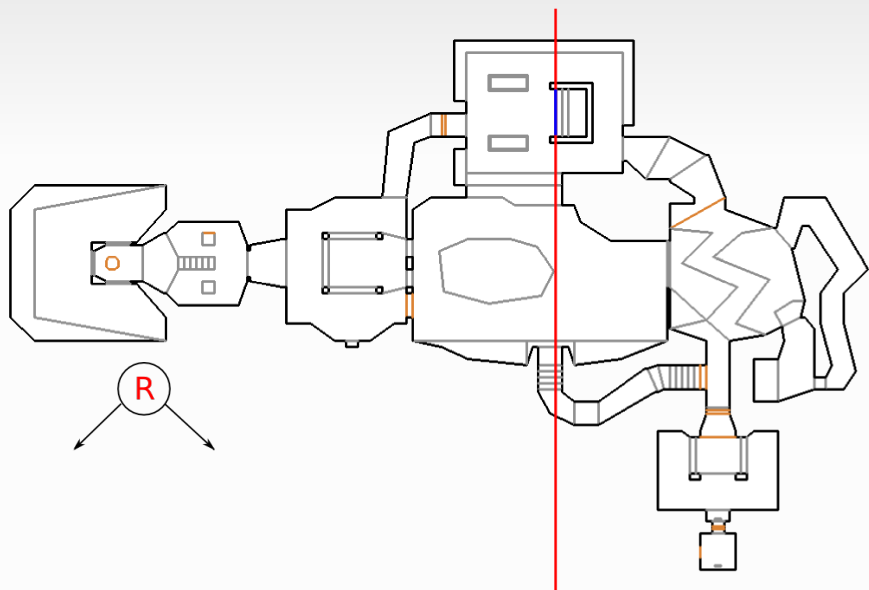


- Test průniku jehlanu s koulí nebo kvádrem
- Scény mají sta tisíce až miliony objektů
- Everon v *Arma Reforger*: jen stromů cca 400 000
- Hierarchické struktury nutné:
  - Hierarchie obálek
  - BSP / kD stromy
  - Kvadrantní / oktalové stromy
  - Alespoň pravidelná mřížka
- Další možnosti:
  - Okluze terénem
  - Okluze rovinou / boxem
  - Okluze počítaná na GPU

## Příklad BSP stromu: offline konstrukce

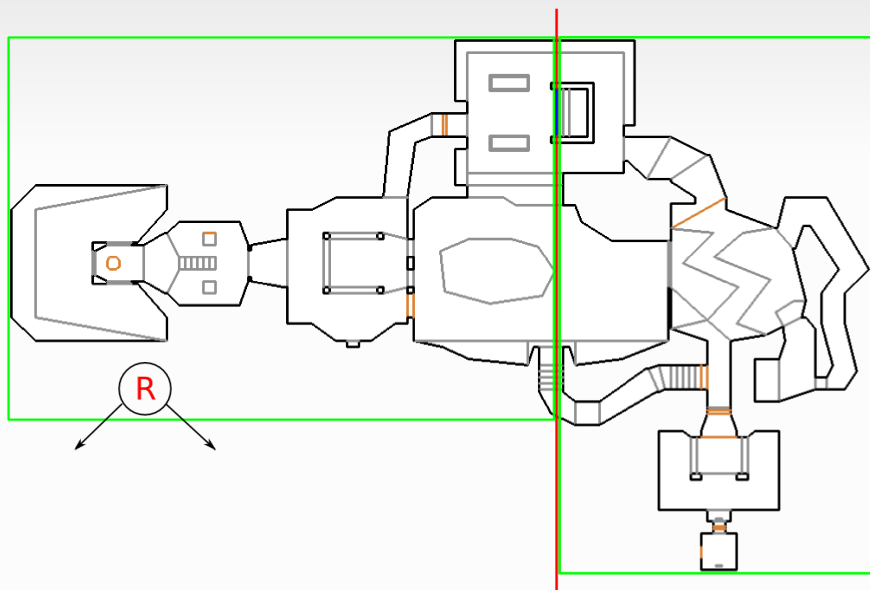


## Příklad BSP stromu: offline konstrukce

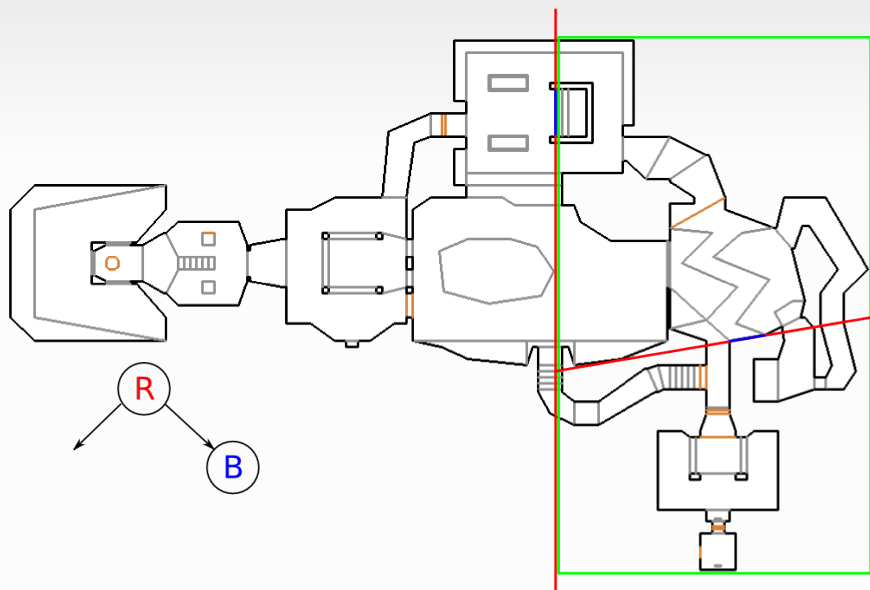




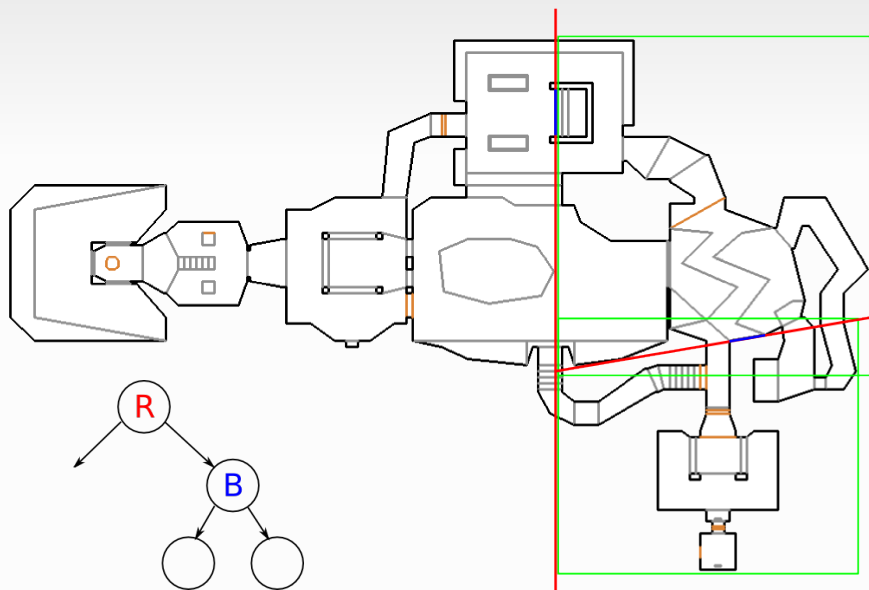
## Příklad BSP stromu: offline konstrukce



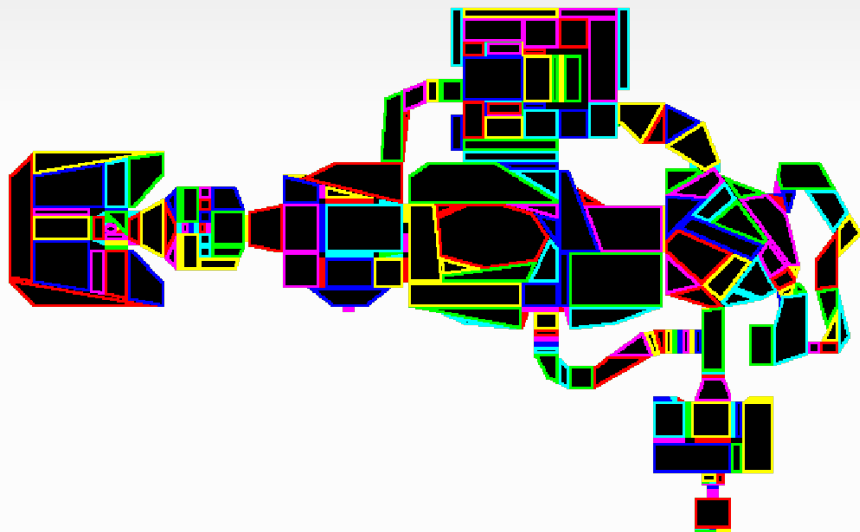
## Příklad BSP stromu: offline konstrukce



## Příklad BSP stromu: offline konstrukce



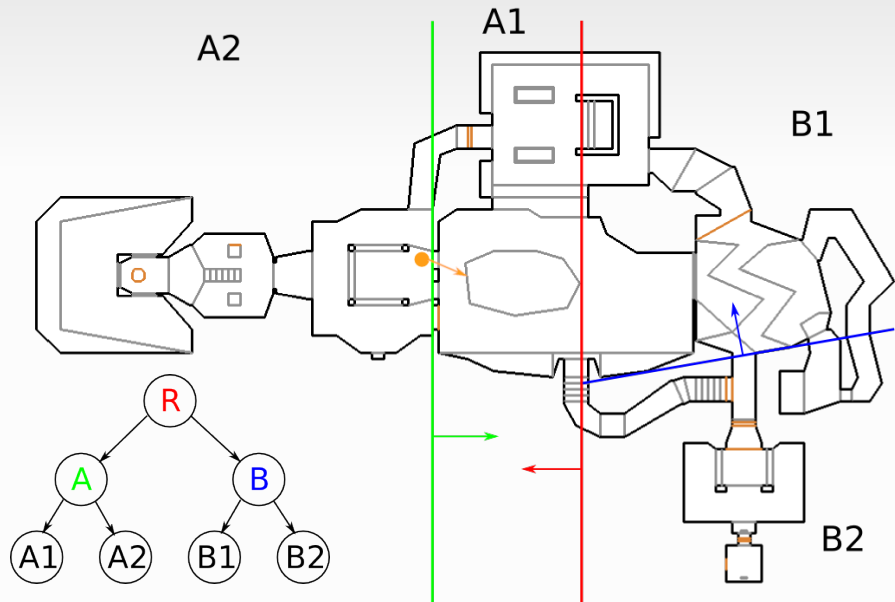
## Příklad BSP stromu: offline konstrukce



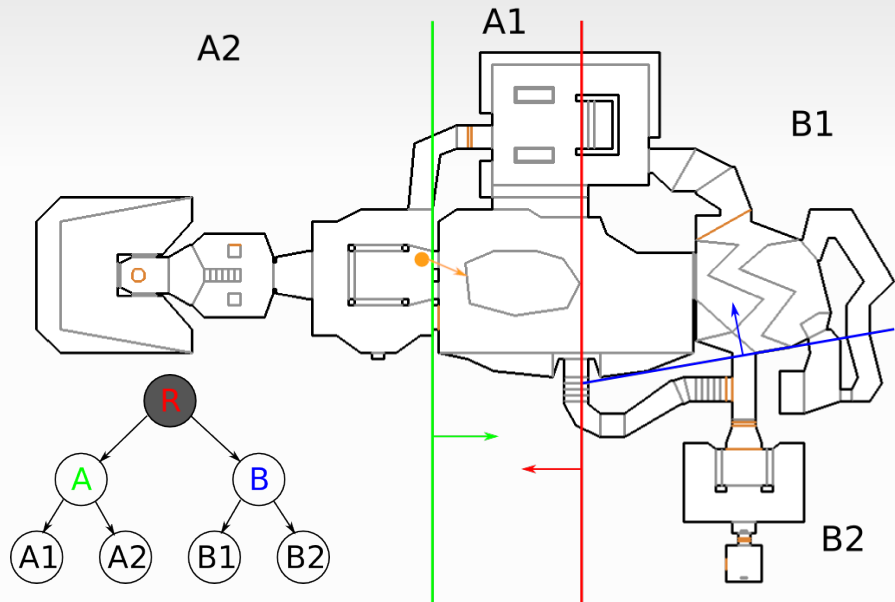
## Příklad BSP stromu: runtime průchod



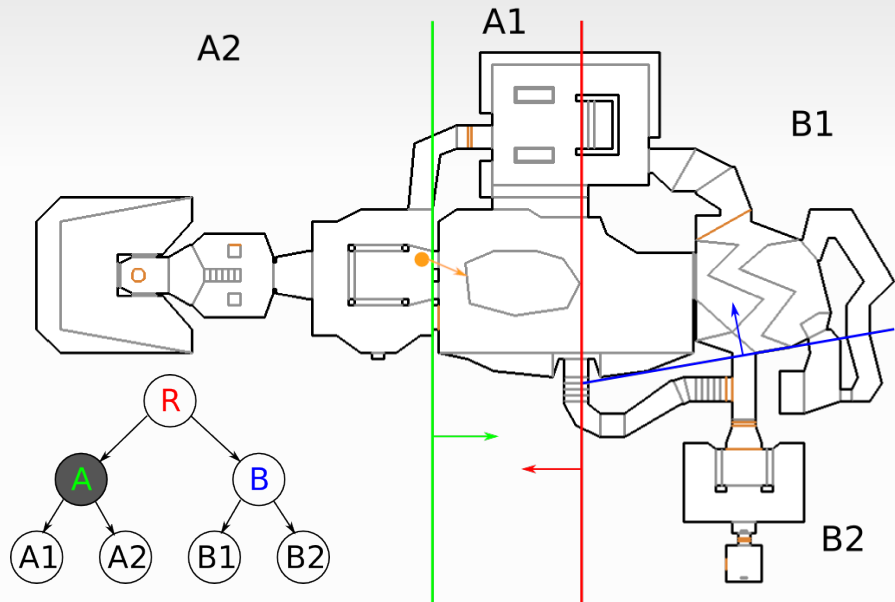
# Příklad BSP stromu: runtime průchod



# Příklad BSP stromu: runtime průchod

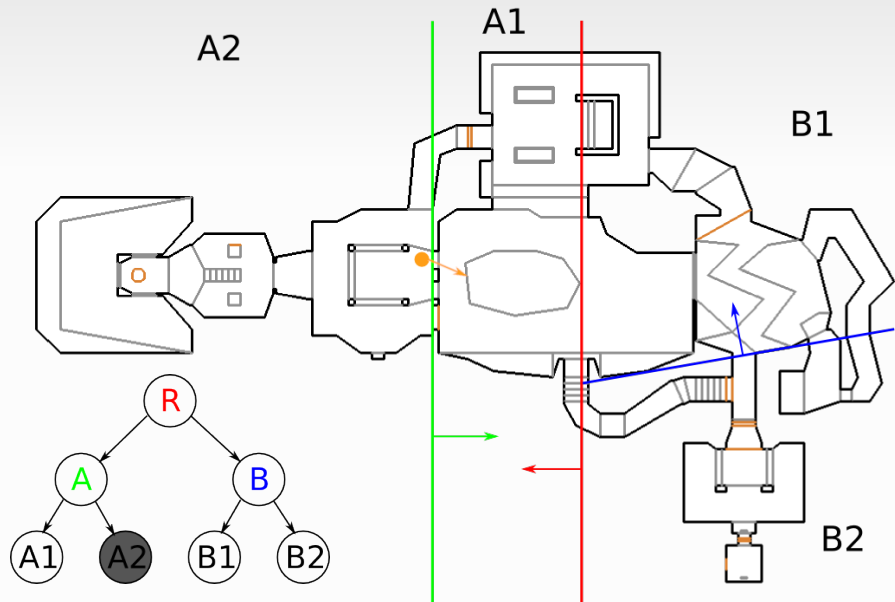


# Příklad BSP stromu: runtime průchod

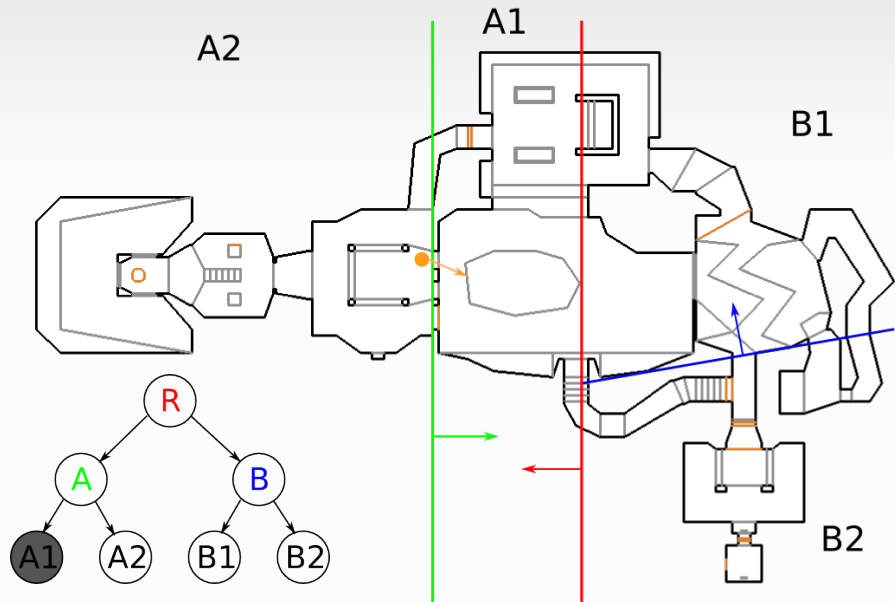




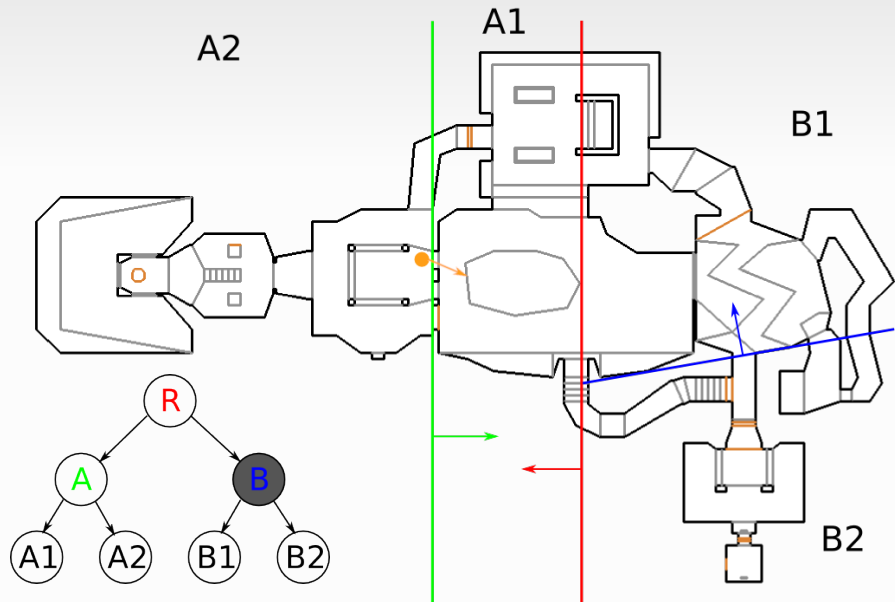
# Příklad BSP stromu: runtime průchod



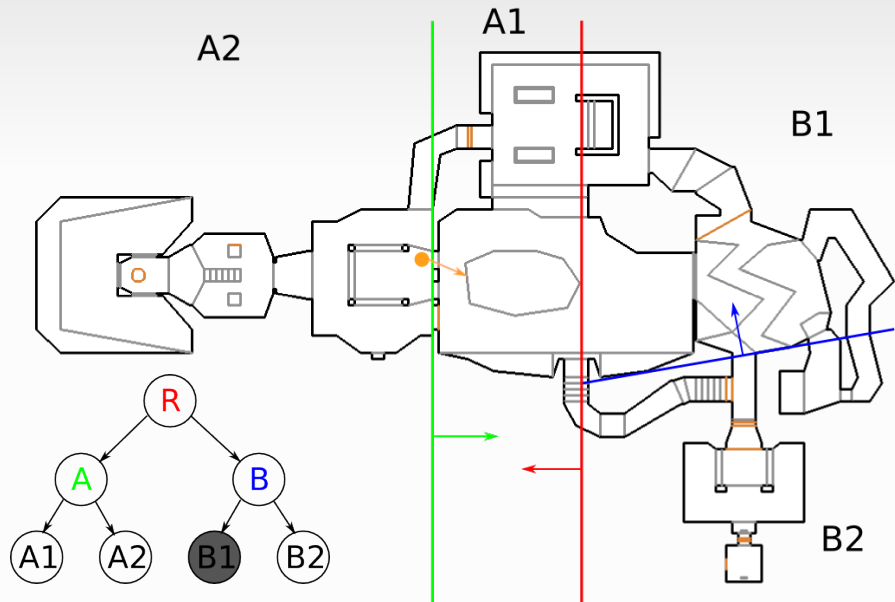
# Příklad BSP stromu: runtime průchod



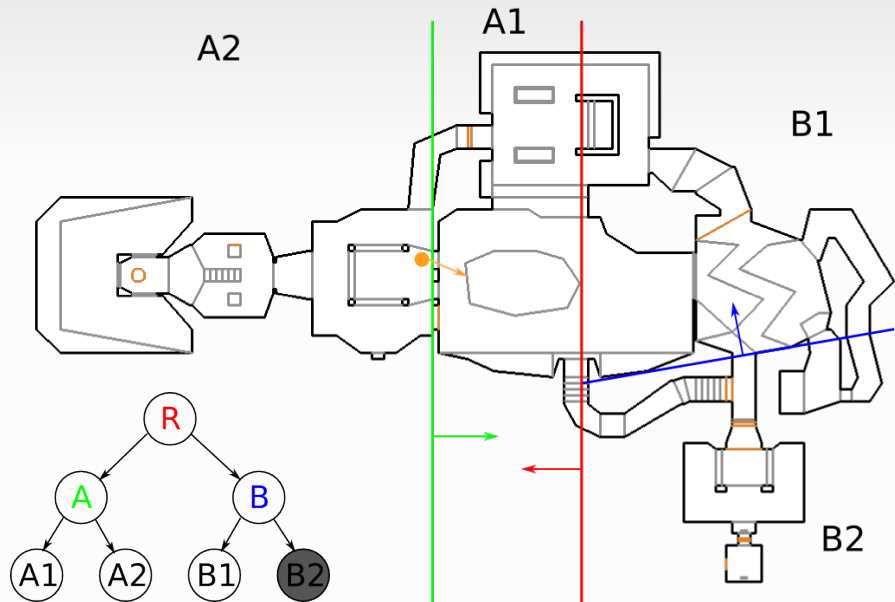
## Příklad BSP stromu: runtime průchod



# Příklad BSP stromu: runtime průchod



## Příklad BSP stromu: runtime průchod



## Příklad BSP stromu: runtime průchod



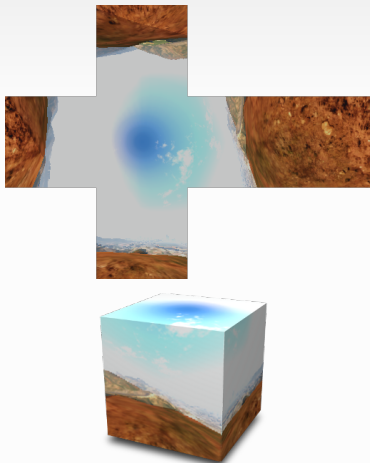


Pitva snímku na GPU — GTA V (PC, 2015)



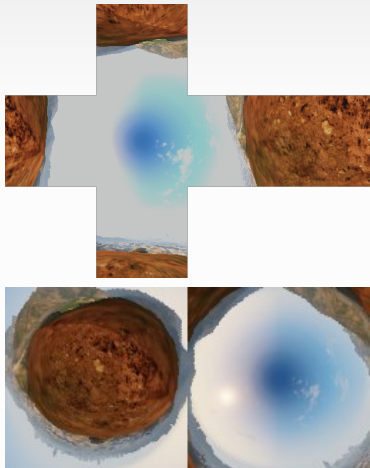


# 1. Odraz okolí



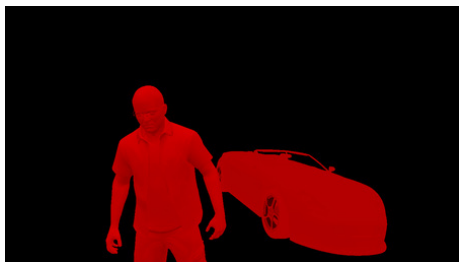
- 128x128 HDR kubická mapa
- 6x *forward* pass s nízkým rozlišením geometrie
- cca 30 kreslení na každou stěnu krychle
- Terén, obloha, významné stavby

# 1. Odraz okolí



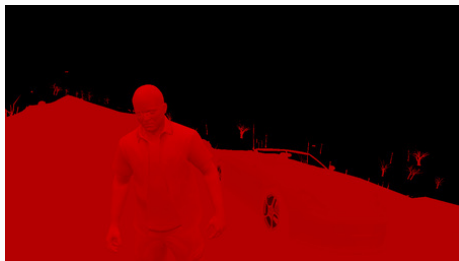
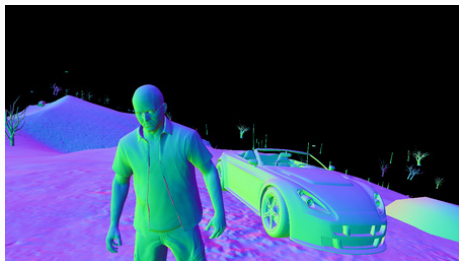
- 128x128 HDR kubická mapa
- 6x *forward* pass s nízkým rozlišením geometrie
- cca 30 kreslení na každou stěnu krychle
- Terén, obloha, významné stavby
- Konverze do dualní paraboloid mapy
- 2x128x128 → méně čtení z textury
- Odstraňuje artefakty na hranách krychle
- Rychlejší přístup do paměti při čtení
- Zkreslení odrazů do stran (málo četné)

## 2. Generování G-bufferů



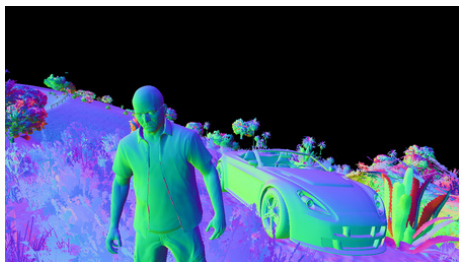
Zleva doprava: barva, lesklá složka osvětlení, normály, osvětlení

## 2. Generování G-bufferů



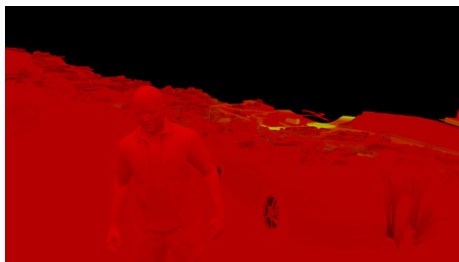
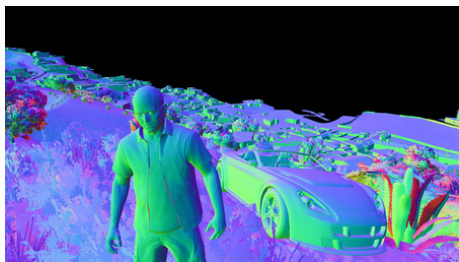
Zleva doprava: barva, lesklá složka osvětlení, normály, osvětlení

## 2. Generování G-bufferů



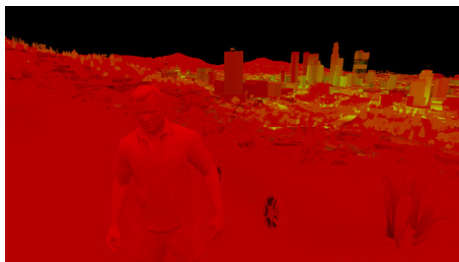
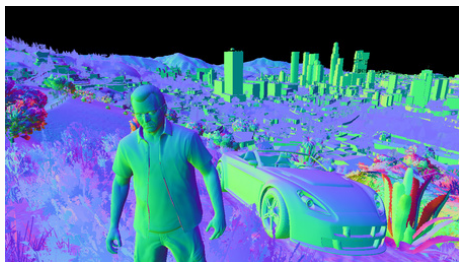
Zleva doprava: barva, lesklá složka osvětlení, normály, osvětlení

## 2. Generování G-bufferů



Zleva doprava: barva, lesklá složka osvětlení, normály, osvětlení

## 2. Generování G-bufferů



Zleva doprava: barva, lesklá složka osvětlení, normály, osvětlení

## 2. Generování G-bufferů



Nahoře depth buffer, dole stencil buffer

- Kreslení neprůhledných objektů
- Zepředu dozadu → *early-Z* test
- Zhruba 1900 kreslení
- 4 buffery: RGBA, 8 bitů / kanál
- 32 bit hloubka, 8 bit stencil
- Hloubka reverzní 1/z kvůli přesnosti
- Stencil buffer pro identifikaci modelů



# Barevný buffer



- Difusní barva povrchu (albedo)
- Alpha kanál — informace pro *dithering*

# Barevný buffer

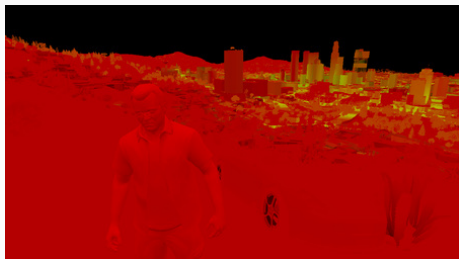


# Barevný buffer



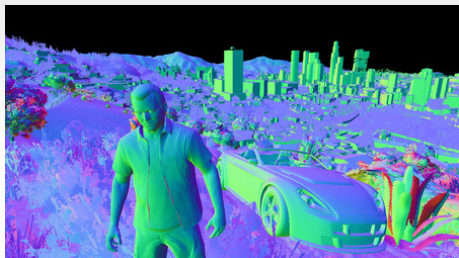
- *Stippled alpha*
- Levná průhlednost vzdálených objektů
- 2x2 šachovnice
- Opraveno později v postprocesu

# Osvětlení



- R — osvětlení sluncem (irradiance)
- G — okolní osvětlení (pravděpodobně)
- B — zářivé povrchy
- A — značí kůži / vegetaci

# Normály povrchu & lesklá složka osvětlení



## Normály

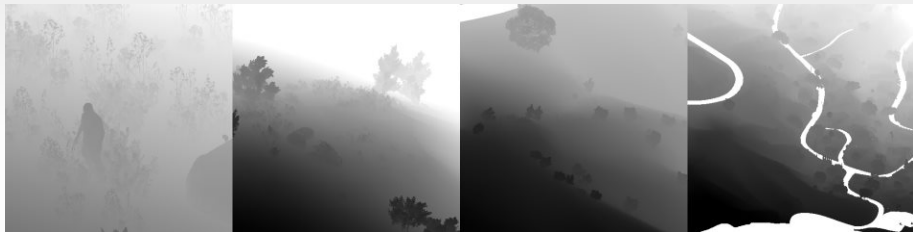
- RGB — normály povrchu
- Alpha — binární maska vegetace



## Lesklá složka

- R — síla lesklého odrazu
- G — lesklost povrchu (hrubost)
- B — Fresnelovy koef. lomu světla
- Alpha — stíny

### 3. Stíny



- 4 stínové mapy: 1024x4096
- Cca 1000 kreslení + stíny mraků
- Určuje zastínění z pohledu slunce
- Konverze do screenspace + rozmazání
- Bílá = osvětleno, černá = stín

## 4. SSAO — aproximace zastínění rozptýleného světla

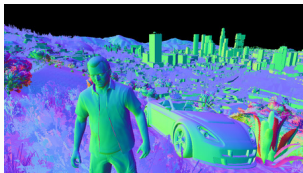
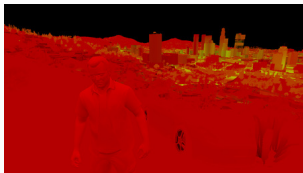


## 4. SSAO — aproximace zastínění rozptýleného světla





## 5. Kombinace G-bufferů



## 5. Kombinace G-bufferů



## 6. Simulace rozptylu světla pod kůží



Před



Po

## 7. Voda



## 7. Voda



## 8. Atmosféra



## 8. Atmosféra



## 8. Atmosféra





## 8. Atmosféra



## 9. Průhledné objekty



## 9. Průhledné objekty



# 10. Postprocesy

Odstranění *ditheringu*



# 10. Postprocesy

Odstranění *ditheringu*



# 10. Postprocesy

HDR Tonemapping, bloom, & FXAA



# Finální snímek

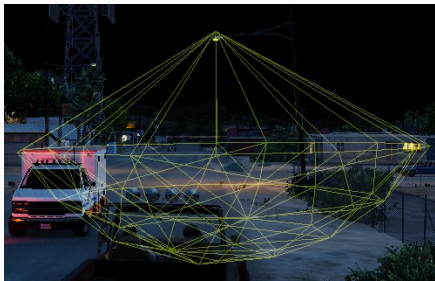


# Shrnutí & vynechané věci

- 4155 kreslení (*draw calls*)
- 1113 textur
- 88 render targetů
- Co jsme vynechali:
  - Kreslení UI (vektORIZOVANÁ GPS mapa)
  - Správa LODů (méně kvalitní objekty v dálce)
  - Kreslení světel
  - Kreslení vody (odraz & lom)
  - Další postprocesy finálního snímku



# Kreslení světla



- Děje se po zkombinování G-bufferů
- Kreslí se geometrie šíření světla
- Koule pro bodová světla
- Kužel pro lampy
- Stínují se jen zasažené pixely
- *Early-Z* test redukuje překreslování

# Kreslení světél

Žádná světla



# Kreslení světél

50% světél



# Kreslení světél

80% světél



# Kreslení světél

100% světél



# Kreslení světel

Finální snímek



# Vzdálená světla

## Chybějící světla



# Vzdálená světla

Kreslení pomocí instancovaných sprajtů





# Vzdálená světla

Test hloubky



# Vzdálená světla

Finální snímek



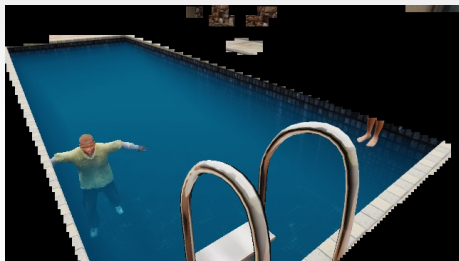
# Kreslení vody



# Kreslení vody



# Kreslení vody



- Vykresli scénu vzhůru nohama
- 240x120 textura odrazů
- Textura pro lom světla
- Na lom aplikuj zamlžení
- Zkombinuj dohromady

# Kreslení vody





Závěr

# Poděkování

## Adrian Courrèges

- <https://www.adriancourreges.com>
- Pitva *GTA V* a *Doom (2016)*

## Fabien Sanglard

- <https://fabiensanglard.net>
- Pitvy *Doom*, *Doom 3*, ...
- Autor knih o enginech *Wolfenstein 3d* a *Doom*



# Odkazy

- <https://careers.bohemia.net/>
- <https://enfusionengine.com/>
- <https://www.bohemia.net/games/armareforger>
- **How do Video Game Graphics Work?**  
<https://www.youtube.com/watch?v=C8YtdC8mxTU>

A large, bright blue moon dominates the background of the slide. In the foreground, the silhouettes of evergreen trees and a person in a vehicle are visible against the moon's glow. The person appears to be holding a long object, possibly a telescope or a tool. The overall scene is a surreal, night-time landscape.

Díky za pozornost

No pixels were harmed in the making of this presentation.

