

# Šachy pomocí sledování paprsku

řešitel: Petr Fusek, xfusek08

## Zadání

1. Vytvořte program, který pomocí metody sledování paprsku zobrazí šachovnici s figurami.
2. Navrhněte jednoduchý způsob uložení stavu šachovnice, který by sloužil ke komunikaci se šachovým strojem nebo modulem pro interakci s uživatelem.
3. Demonstrujte kvalitu zobrazení na několika příkladech stavu šachovnice, zaměřte se na jevy, které nejsou jednoduše řešitelné rasterizací na GPU (stíny, odlesky, průhlednost, ...).
4. Viz například <http://rhk.dataslab.com/wiki/rtChess>
5. Důraz je kladen na vizuální krásu výsledku, kvalitu technického řešení, parametrizovatelnost.

## Vlastní úprava

- Vykreslování v reálném čase
- Použití vlastní CSG pro vykreslované modely
- Požití ray-marching (sphere tracing) pro vykreslování na GPU per-fragment
- Použití optimalizační datové struktury pro zrychlení vykreslování a procházení scény

## Nejdůležitější dosažené výsledky

- Vlastní modelované CSG modely figurek a šachovnice.
- Vykreslování CSG scény v reálném čase na fragment shaderu.
- Použití BVH (binární AABB strom), který je používán při vykreslování..

## Ovládání vytvořeného programu

- Program je ovládán pouze myší
- Při stisknutí **levého tlačítka** lze otáčet kamerou kolem středu.
- **Kolečkem** se přibližuje a oddaluje.

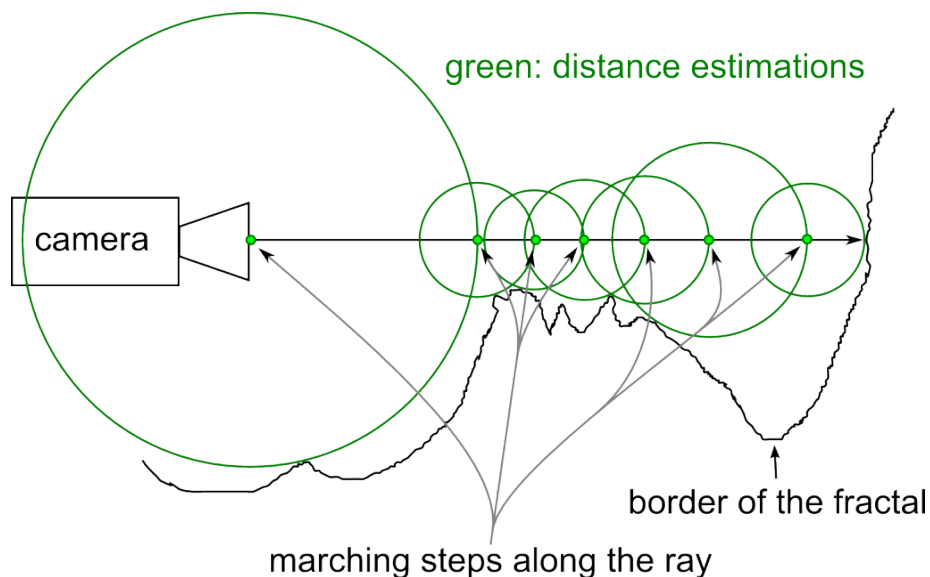
## Zvláštní použité znalosti

Hlavní myšlenka bylo vykreslovat scénu složenou z CSG (constructive solid geometry) modelů pomocí ray marching techniky a užít nad ní optimalizaci hierarchii obalových těles.

### Ray marching

Ray marching je variace na ray tracing, pro 5ípady, kdy nelze přesně nalézt průsečík s primitivem. Scéna je definovaná jako *signed distance function*, která dokáže vrátit vzdálenost bodu v prostoru od nejbližšího povrchu. Provádí kroky ve směru vrženého paprsku do té doby dokud vzdálenost není menší než určitá hranice a tehdy současnou pozici prohlásí za bod průniku s povrchem.

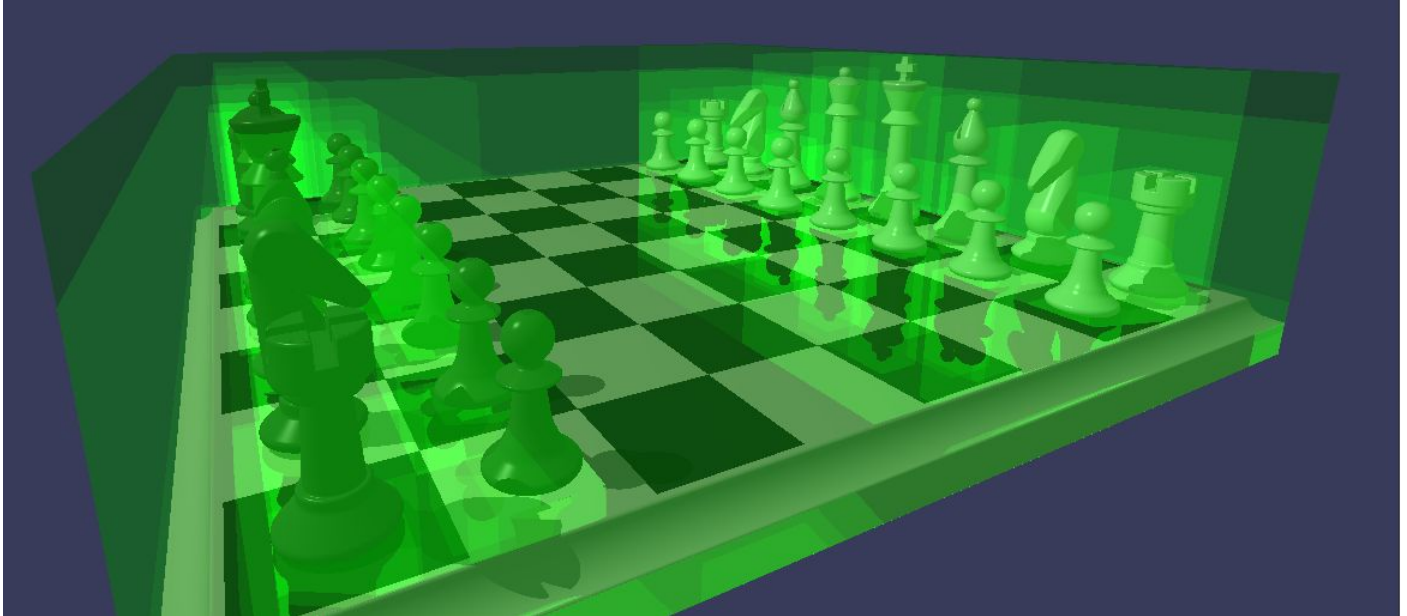
Implementovaná byla populární variace *Sphere Tracing*, která mění velikost následujícího kroku podle aktuální vzdálenosti od scény. Viz. obrázek:



## BVH - AABB Strom

Problém této techniky je, že modely ve scéně jsou složeny ze seznamu primitiv a ten je třeba projít a spočítat pro každé vzdálenost. Toto je příliš náročné dělat pro každý pixel pro všechny modely ve scéně, kde se nachází šachovnice a 32 figurek, které obsahují cca. 8 primitiv, každá.

Proto byla implementována optimalizační datová struktura hierarchie obalových těles. V tomto případě binární strom osově zarovnaných kvádrů. Tento strom je sestaven na CPU při načtení scény ze souboru a uložen v grafické paměti. Listy stromu odpovídají jednotlivým modelům (krátký seznam primitiv a operací nad nimi).

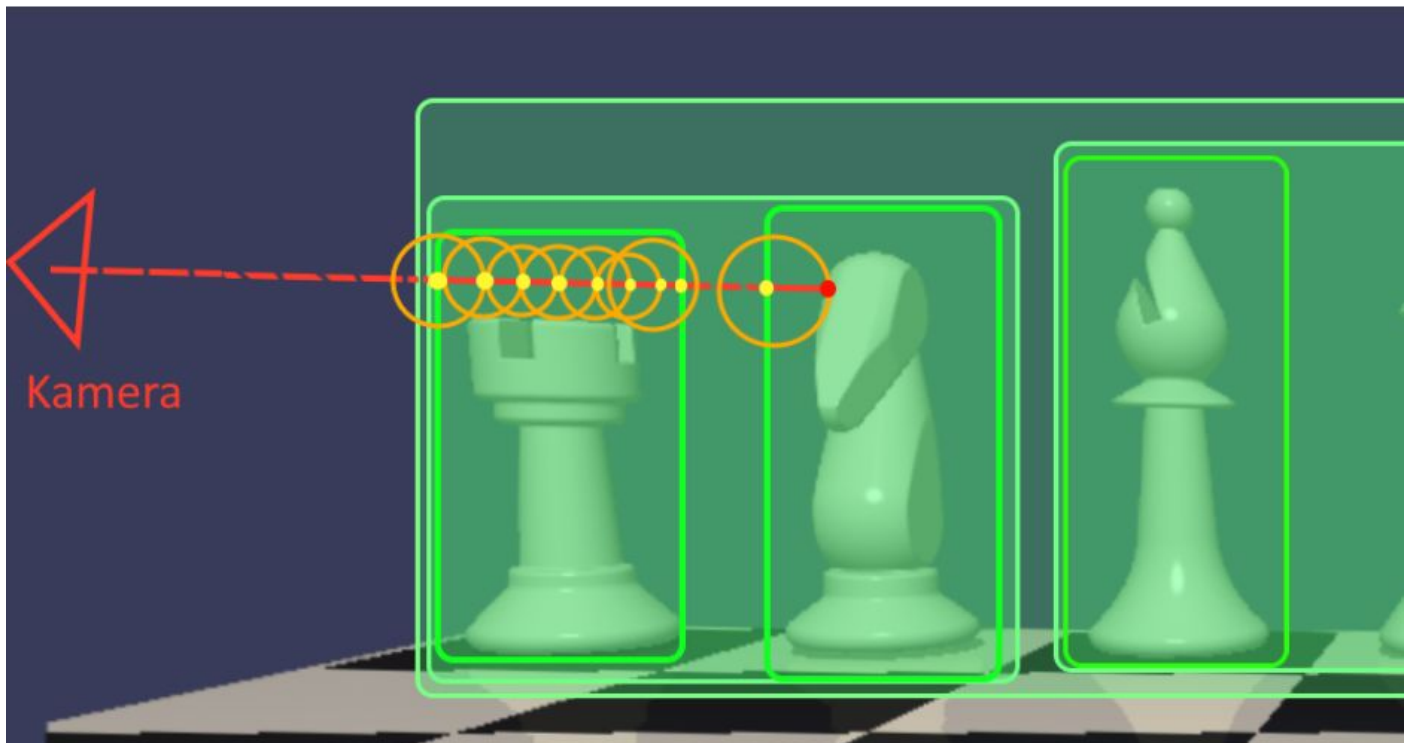


Obrázek ukazující skutečnou hierarchii obalových kvádrů.

## Vykreslování

Pro každý vržený paprsek z pixelu se BVH projde *left-first search* způsobem a je nalezen seznam listových uzlů, které paprsek protíná. Z něj se najde ten nejbližší a provede se **ray-marching** jenom v prostoru daného obalového tělesa a pokud paprsek minul, opakuje se process pro druhý nejbližší uzel ze seznamu.

Obrázek toto ilustruje:



## Použité technologie

- C++ 17
- CMake 3.18
- grafická knihovna GLM <http://glm.g-truc.net/0.9.8/>
- OpenGL 4.5 s GLAD loaderem <https://glad.dav1d.de/> (součást projektu)
- SDL2 <https://www.libsdl.org/>
- Správa verzí - Git  
Projekt: <https://github.com/xfusek08/PRGChess>  
(využívá gitmodules je třeba použít git clone --recursive)
- Json for modern cpp - hlavičková knihovna: <https://github.com/nlohmann/json> (součást projektu)

## Použité zdroje

### Ray marching + obrázky:

- <https://computergraphics.stackexchange.com/questions/161/what-is-ray-marching-is-sphere-tracing-the-same-thing/163>
- <http://celarek.at/>
- Obrázek: [https://en.wikipedia.org/wiki/Bounding\\_volume\\_hierarchy](https://en.wikipedia.org/wiki/Bounding_volume_hierarchy)

### Kusy kódu:

- <https://stackoverflow.com/questions/2333728/stdmap-default-value>
- <https://stackoverflow.com/questions/8975773/stackless-pre-order-traversal-in-a-binary-tree>
- Funkce vzdálenosti pro jednotlivá primitiva a plynulé maximum a minimum: <https://iquilezles.org/www/articles/distfunctions/distfunctions.htm>
- AABB průniky: <https://medium.com/@bromanz/another-view-on-the-classic-ray-aabb-intersection-algorithm-for-bvh-traversal-41125138b525>

## Co bylo nejpracnější

Ze začátku byl velký problém rozchodit samotnou aplikaci, nakonfigurovat cmake, správně naimportovat všechny knihovny a dostat se do bodu, kdy je zobrazené okno, aktualizací smyčka, a funkční načtené funkce z OpenGL 4.5.

Poté bylo nejvíce práce se samotným fragment shaderem, který byl mnohokrát přepsán. Je v něm implementován ray marching, vzdálenostní funkce pro jednotlivá primitiva, a nejtěžší ze všeho: Průchod binárního stromu a správné pořadí zpracovávání listů s průsečíkem.

## Zkušenosti získané řešením projektu

Popište, co jste se řešením projektu naučili. Zahrňte dovednosti obecně programátorské, věci z oblasti počítačové grafiky, ale i spolupráci v týmu, hospodaření s časem, atd.

Rozsah: formulujte stručně, uchopte cca 3-5 věcí

- Dobrá znalost principů **ray-marching** a možnosti jak lze využít.
- Získaná znalost iniciačních prací na grafickém projektu (potřebné knihovny, CMake)
- Principy osvětlování, odlesků a stínů pro sledování paprsku.
- Vlastnosti a úskalí užití hierarchie obalových těles.

## Autoevaluace

Ohodnoťte vaše řešení v jednotlivých kategoriích (0 – nic neuděláno, zoufalství, 100% – dokonalost sama). Projekt, který ve finále obdrží plný počet bodů, může mít složky hodnocené i hodně nízko. Uvedení hodnot blízkých 100% ve všech nebo mnoha kategoriích může ukazovat na nepochopení problematiky nebo na snahu kamuflovat slabé stránky projektu. Bodově hodnocena bude i schopnost vnímat silné a slabé stránky svého řešení.

**Technický návrh: 60%** (analýza, dekompozice problému, volba vhodných prostředků, ...)

Stručně (1-2 řádky) komentujte hodnocení.

Projekt je spíše prototyp, velice nedotažený, odlesky a stíny jsou velice základní, a vznikají zde artefakty spojené s BVH průchodem.

Obecně se jedná o slibnou metodu, která s dostatkem času a ladění může produkovat zajímavé výsledky.

**Programování: 80%** (kvalita a čitelnost kódu, spolehlivost běhu, obecnost řešení, znovupoužitelnost, ...)

Jako součást projektu byl vyvíjen git-modul, který aplikace používá a který obaluje všechny nutné kroky k tomu aby uživatel měl k dispozici okno s funkční OpenGL 4.5 a poskytuje zjednodušené API pro práci s Okny a OpenGL.

Tento projekt lze použít v budoucnu na jakýkoliv další OpenGL projekt a měl by být nezávislý na okenní knihovny.

**Vzhled vytvořeného řešení: 65%** (uvěřitelnost zobrazení, estetická kvalita, vzhled GUI, ...)

Osvětlovací model je celkem základní a chybí možnost textur a pokročilejších materiálů.

Co se povedlo jsou modely figurek, které jsem vytvořil sám pomocí jednotlivých primitiv a lze je za běhu aplikace měnit v konfiguračním souboru scény.

**Využití zdrojů: 70%** (využití existujícího kódu a dat, využití literatury, ...)

Věřím, že všechny kusy kódu zkopírované, či inspirované jsou v řešení okomentované zdrojem.

**Hospodaření s časem: 30%** (rovnoměrné dotažení částí projektu, míra spěchu, chybějící části řešení, ...)

Příliš času zabrala implementace optimalizační struktury a poněkud komplikovaného fragment shaderu.

Program umí pouze jednu scénu a proti zadání nezbyl čas na integraci s šachovým strojem.

Zcela chybí průhlednosti a jiné pokročilé aspekty sledování paprsků.

Zbývá dodělat integraci světla do scény + ovládání pomocí GUI.

**Celkový dojem: 50%** (pracnost, získané dovednosti, užitečnost, volba zadání, cokoliv, ...)

Jedná se spíše o experiment, jestli dokážu implementovat ray-marching složitější scény v reálném čase na úkor vizuální kvality a uživatelské přívětivosti. To co bylo původně zamýšleno jako příprava na práci s grafickými efekty scény se ukázalo jako hlavní náplň projektu.

## Doporučení pro budoucí zadávání projektů

Na rozsah zadání projektu bylo v rámci semestru málo času, i tak jsem tomu věnoval více než 40 hodin což je nejvíce ze všech ostatních předmětů jelikož toto je pro mě nejdůležitější předmět.