

C# Zápočtový program - CPU Ray-tracer

Vytváření fotorealistické scény skrze procesor

David Říha

Uživatelská dokumentace

Popis

Program zprostředkovává uživateli vykreslování libovolné scény složené z podporovaných těles a světelných zdrojů. Veškeré detaily výsledného obrázku si nastaví dle potřeby (viz ovládání a konfigurace) a následně spustí program pro vytvoření souboru s obrázkem.

Výstup

Po doběhnutí programu (délka běhu může nabývat netriviálních hodnot v závislosti na konfiguraci) se výsledek uloží do složky s programem do souboru s názvem `outputImage.pfm`. Tento soubor je formátu Portable FloatMap a je třeba jej otevřít skrze nějaký program, který s ním dokáže pracovat (GIMP je doporučován a byl používám při vývoji).

Během běhu programu je otevřena konzole, která slouží k sledování případných chyb, na které program narazí. Mezi takové kritické chyby může patřit například: Chybný nebo neexistující konfigurační soubor, špatný formát konfiguračního souboru nebo chybějící nutná část konfigurace. Mohou se objevit i pouhá varování, která upozorňují uživatele, že v konfiguračním souboru nebyl popsán nějaký element, který sice není nutný k tvorbě obrázku, ale dojde k nastavení na základní hodnotu, a tedy může dávat nečekané výsledky (uživatel pravděpodobně pouze zapomněl element popsat).

Do konzole se na konci běhu programu též vypíše délka běhu programu.

Ovládání a konfigurace

Celé nastavení scény probíhá skrze konfigurační soubor `'config.json'`. Zde se ve formátu JSON nastavují různé aspekty scény, které se následně nahrají do programu a budou využívány při běhu pro vytvoření obrázku.

Nastavují se následující prvky: Kamera, objekty scény, osvětlení scény, parametry výsledného obrázku – pro konkrétní vlastnosti prvků viz JSON Formát.

Jisté položky jsou povinné pro vytvoření scény. Mezi ně patří: Kamera (se všemi vlastnostmi), Barva objektu, Materiál objektu

V případě chybějících informací se program adekvátně brání – pokud je informace nezbytná, tak se ukončí – pokud není nezbytná, použije přednastavenou smysluplnou hodnotu a do konzole napíše varování

JSON Formát

V konfiguraci se u několika částí objevují společné vlastnosti:

position - x (desetinné číslo)

- y (desetinné číslo)

- z (desetinné číslo)

color - r (desetinné číslo v rozmezí 0..1)

- g (desetinné číslo v rozmezí 0..1)

- b (desetinné číslo v rozmezí 0..1)

material - diffuse (difusní koeficient - jak moc je materiál matný (0-1))

- reflection (odrazivý koeficient - jak moc materiál odráží světlo (0-1))

- ambient (ambientní koeficient - jak moc k barvě přispívá okolní světlo (0-1))

- reflectionSize (exponent) (ostrota odrazu (větší exponent = ostřejší světlo) (5-500))

- transparency (průhlednost (0-1))

- refractionIndex (index lomu - určuje jak se má zalomit světlo mezi dvěma materiály (viz tabulky indexů lomů) (index lomu vzduchu = 1.0003, vakuum = 1, led = 1.3))

Obecný Formát vyžadovaný pro správné načtení konfiguračního souboru:

imageParameters

- width (šířka v px)

- height (výška v px)

- rtRecursionDepth (maximální hloubka rekurze ray-tracingu)

- spp (kvalita antialiasingu - samples per px) (doporučeno 2-144)

- aa (true/false - antialiasing toggle)

- backgroundColor (barva vykreslena v paprscích bez průsečíku) (formát *color*)

camera

- position (pozice kamery)

- viewVector (směr pohledu kamery) (formát *position*)

- upVector (normála kamery - určuje rotaci kolem pohledového vektoru) (formát *position*)

shapes (pole všech těles - (tělesa jsou zapisována v hierarchiích))

(jedna položka = 1 hierarchický strom)

Následné vlastnosti se mohou vyskytovat kdekoli v hierarchii, v jakémkoliv vrcholu:

color (barva daná vrcholem – přepíše barvu rodičovského vrcholu)

material (materiál daný vrcholem – přepíše materiál rodičovského vrcholu)

transform (popisuje transformaci těles relativní k rodičovskému vrcholu)

- translateX, translateY, translateZ (posun k dané ose (float))
- rotateX, rotateY, rotateZ (rotace okolo dané osy - dáno v stupních (float))

sphere (popis koule v aktuálním vrcholu) (v jednom vrcholu popisovat jen jedno těleso)

- position (pozice koule – ve finále změněno transformacemi)
- radius (poloměr koule (float))
- color (barva koule)
- material (materiál této koule)

plane (popis roviny v tomto vrcholu)(ideálně v jednom vrcholu popisovat jen jedno těleso)

- position (pozice roviny – ve finále změněno transformacemi)
- normal (normála roviny)(formát *position*)
- color (barva roviny)
- material (materiál této roviny)

childNodes (pole všech synů)(1 položka = 1 vrchol)

V hierarchii platí pravidla:

- Pokud je popsáno těleso, musí být alespoň v jednom z vrcholů k němu vedoucích nebo přímo v něm popsána barva a materiál
- Pokud má těleso vlastní barvu/materiál, platí pro něj absolutně, ale nepřepisují dědičnou barvu/materiál
- Transformace se sčítají (pokud je popsáno těleso, platí pro něj všechny dosavadní transformace ze stromu)

lights (popisuje veškeré osvětlení scény)

- ambientLight (ambientní světlo přítomné ve všech bodech scény)

- intensity (intenzita světla (float))(doporučeno 0.1-0.5))

-lightSources (pole všech zdrojů světla - 1 položka = 1 zdroj)(následuje popis jedné položky)

- position (pozice zdroje)
- color (barva světla)
- intensity (intenzita světla (float))(doporučeno 0.1-0.5))

Schopnosti programu

Program využívá ray-tracing pro získávání barvy do jednotlivých pixelů obrázku. Jednoduše 'vystřelí' paprsek z kamery směrem odpovídajícím danému pixelu a sleduje s čím koliduje. Při naražení na překážku se pokusí získat barvu. Zde se paprsek rozdělí na další paprsky, které počítají odraz, lom a příspěvek světla. Nakonec je díky těmto technikám možné získat v obrázku tělesa s různými vlastnostmi, jako například zrcadla nebo průhledné sklo, která vypadají důvěryhodně.

Pro eliminaci nekvalitních hran je v programu zabudován anti-aliasing, který umožní dosažení pocitově kvalitnějšího obrázku na úkor rychlosti výpočtu.

Počítání jednotlivých paprsků probíhá paralelně pro zrychlení výpočtu.

Programátorská dokumentace

Obsah jednotlivých zdrojových souborů

AntiAlias.cs – Výpočet paprsku s využitím anti-aliasingu

Camera.cs – Informace o kameře a paprsků k ní vztažených

Constants.cs – Důležité globální konstanty

FloatImage.cs – Zpracování obrázku do formátu PFM (vnější zdroj)

ImageCreation.cs – Tvorba obrázku

InputHandling.cs – Deserializace konfiguračního souboru a jiných vstupů

Light.cs – Veškeré osvětlení scény + důležité funkce GetColor – výpočet barvy na jistém místě

Materials.cs – Třída materiál a její vlastnosti

MyExceptions.cs – Výjimky vytvořené konkrétně pro tento program

Program.cs – Hlavní běh programu – obsahuje main()

Rays.cs – Informace o paprscích

RayTrace.cs – Funkce RayTrace, která provádí sledování paprsku a výpočet barvy

Scene.cs – Třída Scene se všemi informacemi o vyobrazované scéně

Solids.cs – Tělesa a výpočty s nimi spojené

Transformations.cs – Transformace a jejich výpočet

Utility.cs – Pomocné třídy a objekty

VectorCalculations.cs – Výpočty spjaté s vektory

Konkrétní funkce a vlastnosti jsou zdokumentovány přímo v kódu