



Dokumentace k projektu pro předmět ITU

Varianta zadání:  
**Raytracer GUI**

11. prosince 2012

Autoři: Matej Kendra, [xkendr00@stud.fit.vutbr.cz](mailto:xkendr00@stud.fit.vutbr.cz)  
Josef Kyloušek, [xkylou00@stud.fit.vutbr.cz](mailto:xkylou00@stud.fit.vutbr.cz)  
Petr Dvořáček, [xdvora0n@stud.fit.vutbr.cz](mailto:xdvora0n@stud.fit.vutbr.cz)

Fakulta Informačních Technologií  
Vysoké Učení Technické v Brně

## 1. Úvod

Cílem tohoto projektu bylo navrhnout, vytvořit a otestovat uživatelské prostředí pro jednoduchý raytracer. Jako základ jsme použili rozložení ovládacích prvků typické pro programy tohoto typu (práce s grafikou). Jedná se o boční panely. Jako vlastní nápad jsme přidali ještě panel při dolním okraji okna. Ten je určen pro provádění často opakovaných akcí, jako je například vkládání objektů do scény, což se v jiných programech musí často provést průchodem přes několik nabídek. Tlačítka v tomto panelu by měla být uživatelsky definovatelná, aby si každý uživatel nastavil často používané operace podle vlastních preferencí. Taktéž boční panely mají určitou možnost uživatelské úpravy. Především je to možnost skrýt či zobrazit pokročilejší nástroje.

Program je určen hlavně začátečníkům s počítačovou grafikou, proto je snaha o jeho jednoduché a intuitivní ovládání pomocí několika málo příkazů. Jelikož tato aplikace slouží pouze jako jednoduchá ukázka práce raytraceru, žádné další složitosti nejsou ani žádoucí. Cílovými zařízeními pro používání tohoto programu jsou především moderní platformy typu mini-notebook, smartphony, tablety, atp., jejichž společná vlastnost je nevelký displej. Proto jsou oba boční panely skrývatelné na šířku několika pixelů, aby v tuto chvíli byla umožněna práce se scénou s co nejmenšími omezeními překážejícími panely. Aby uživatel nemusel pro každé skrytí a zobrazení panelu klikat, mají panely navíc režim automatického skrývání, při kterém jsou skryty a zobrazují se najetím myši nad jejich plochu. Uživatel provede požadovanou operaci a po přesunutí myši mimo panel, je tento opět skryt. Toto automatické chování ušetří čas strávený zobrazováním a skrýváním panelu. Pro uživatelskou přívětivost je též nastavitelná šířka těchto panelů.

## 2. Implementace

Využili jsme knihovnu Qt pro zobrazení oken, menu, panelů a dalších prvků uživatelského prostředí a dále knihovnu OpenSceneGraph pro práci se scénou a modely.

Principy knihovny Qt jsou obecně známy a byly i náplní přednášek a nebudeme je tedy dále rozebírat.

Knihovna OpenSceneGraph je výkonná open source 3D knihovna, používaná vývojáři na poli vizuálních simulací, her, virtuálních realit, vědeckých vizualizací a modelování. Knihovna je celá napsána ve standardním jazyce C++ a OpenGL. Běží na všech platformách od Windows, OSX, GNU/Linux, IRIX, Solaris, HP-Ux, AIX a FreeBSD. V dnešních dnech je vedoucí světovou zobrazovací technologií, používaná ve vizuálních simulacích, simulacích vesmíru, vědeckých, a palivových simulacích, hrách a virtuálních realitách.

Použití knihovny OpenSceneGraph.

V našem projektu je knihovna použita převážně ve třídě osgWidget. Ta implementuje hlavní widget, který zobrazuje, otevírá, ukládá a pracuje se scénou Raytraceru. Komunikace mezi ostatními třídami případně hlavním oknem, probíhá v rámci knihovny Qt za pomoci signálu. Takovýmto způsobem se posílají názvy souboru k načtení či uložení. Ve třídě se pak tyto soubory otvírají/ukládají podle přijatého signálu.

Zobrazování je prováděno OpenGL rendererem. Ten má při svém otevření prázdnou scénu. Přesouváním objektů z panelu, implementovány byly pouze krychle a koule, se do scény vloží primitiva podle daného objektu. Další možností je vložit již uložený podél funkcí Open v Menu (otevřít se dají formáty .osg a .osgt).

Hlavními problémy bylo propojit krajní dokované widgety s třídou `osgWidget`. Za pomoci signálu sice totiž lze předávat data, ale každý z přidáných objektů má své vlastní atributy - pozici, barvu, název objektu apod. Z toho důvody jsme implementovali pouze změnu barvy, kdy si barvu pamatovat nemusíme, pouze ji přepíšeme novou hodnotou. Například u pozice objektu bychom si museli pamatovat pozici ve všech souřadných osách (x,y,z).

Oříškem také bylo jakým způsobem rozlišovat objekty. Nejjednodušší se zdálo textově, podle názvu vkládaného do stromu scény dokovací widget `treeWidget`), do kterého se vkládají názvy až skrze třídu `osgWidget`. Avšak v knihovně `OpenSceneGraph`, nelze přímo nastavovat jméno objektu. Důvtipným vložením názvu objektu funkcí `addDescription(<název>)`, jsme nakonec tento neduh eliminovali.

Nevýhodou této knihovny, ale ovšem je dle našeho názoru, transformace os displeje do os 3D scény. Ani po pár dnech usilovného hledání, jsme nenašli způsob, jak získat souřadnice středu scény. Proto také je vkládání čistě "osové", kdy se transformují pouze osy x a y z displej na padesátkrát menší velikost x a y scény (velké implicitní skoky, rozlišení displeje), kdy osa z je pro naše účely vždy 0.

### 3. Testování

Testování probíhalo na 2 skupinách uživatelů. V první skupině byli lidé bez velkých znalostí počítačové grafiky (18 osob). Aplikace byla konstruována především pro tuto skupinu. Dále jsme ještě testovali na vzorku lidí s pokročilejšími znalostmi poč. grafiky (21 osob). Testování probíhalo formou jednoduchých scénářů. Např.: „Umísti do scény 3 koule, obarvi jednu na zeleno, druhou na modro a třetí na červeno.“ Uživatelé byli při práci sledováni, ale dělali jsme vše pro to, aby je tato skutečnost nerozrušovala. Testování probíhalo v neformálních prostorách a na účastníky nebyl vyvíjen žádný tlak, aby nebyly výsledky tímto zkreslené. Jako měřenou veličinu jsme zvolili čas, který testeři potřebují pro provedení jednotlivých scénářů. Abychom mohli výsledky interpretovat, změřili jsme dobu provádění scénářů i nám, tvůrcům, a zjištěné hodnoty porovnali s těmi získanými od uživatelů. Skupině začátečníků trvalo provedení scénáře průměrně o 23 % déle, pokročilým uživatelům o 16 % déle. Tyto výsledky považujeme za dostatečné, tedy že uživatelské prostředí je dostatečně intuitivní. Dále jsme se ještě uživatelů zeptali na několik jednoduchých otázek. Získali jsme tyto výsledky:

85 % uživatelů by tento program po jeho úplném dokončení využilo.

90 % uživatelů označilo rozhraní za intuitivní.

100 % uživatelů splnilo všechny požadované scénáře.

### 4. Závěr

Některé části programu nejsou ještě v době odevzdání programu funkční, ale všechny základní prvky, na které jsme se v projektu zaměřili, fungují podle našich představ a jsou i otestované s výbornými výsledky.