

# Posudek oponenta diplomové práce

**Kamil Rendl:** Simulátor optických jevů

Diplomová práce se zabývá návrhem a realizací programového simulátoru optických soustav pro vytváření a zobrazování hologramů, s primárně výukovým účelem. Práce má být základem pro další rozvoj simulátoru, důraz tedy byl kladen na dokumentovanost a rozšiřitelnost implementace.

Text práce je formálně dobře strukturován a členěn do celkem 10 kapitol, což je až zbytečně mnoho. Seznam použité literatury obsahuje pouze pět položek, které ale odpovídají povaze práce. Jedna příloha uvádí příklad konfiguračního souboru, chybí uživatelská příručka (ať už jako příloha textu nebo samostatný elektronický dokument). Na přiloženém CD jsou zdrojové kódy a spustitelná verze simulátoru, bohužel ale žádné příklady experimentů; CD nemá popisku. Implementace nástroje v jazyce Java má rozsah cca 4000 řádek kódu a je dobře strukturovaná a komentovaná. V textu i ve zdrojových kódech se vyskytuje poměrně velké množství překlepů (namátkou „package controller.contorller“, „rayTraicing“ či „getSingValue“) a občasné gramatické problémy.

Moje největší výhrada k textu diplomové práce spočívá v tom, že sice formálně vhodně pokrývá analytickou, návrhovou, realizační a ověřovací část (co do struktury a názvů kapitol) ale je příliš rozvěklý a podstatné informace jsou umístěny na jiných místech, než odpovídá formální struktuře. Podstatné informace se proto špatně zjišťují.

Analytická část je minimální, omezuje se na stručné představení dvou podobných aplikací na 2 stranách. Zcela postrádám přehled související fyzikální teorie, potřebné pro pochopení funkčnosti projektovaného programu (některé odpovídající partie jsou až u popisů metod tříd implementace). Zadání není nikde v práci zjevně shrnuto a rozvedeno, jednotlivé požadavky na funkčnost/vlastnosti simulátoru jsou rozptýleny v textu několika kapitol, zejména (a nevhodně) v kapitole 4 Návrh architektury jako “poznatky z předchozí verze”. Popis implementace programu v kap. 6 a 7 nevhodně používá strukturování “po jednotlivých třídách” nikoli metodou shora dolů a neposkytuje proto nadhled, který je potřeba pro pochopení aplikace (např. klíčová informace, že třída Box je de facto hlavní třídou celého simulátoru, je skryta pod nadpisem “Vazby v datovém modelu”). Pochopit koncepci programu proto vyžaduje přečíst v podstatě celý text práce, přestože design implementace není nakonec příliš složitý. Paradoxně je v tomto ohledu nejvíce užitečná kapitola 8, která má ale popisovat (pouze) mechanismus rozšiřování aplikace.

Text až příliš využívá rozvěklého popisného stylu i v místech, která potřebují strukturovaný technický text (popisy algoritmů – např. sledování paprsku str. 37-38, mechanismy použité v designu pro rozšiřitelnost kap. 8, způsob ověření kap. 9); za zcela nadbytečné považuji podrobné popisy jednotlivých tříd a metod v kap. 6 zvláště s ohledem na dobře komentovaný a strukturovaný zdrojový kód.

Ověření funkčnosti, tak jak je dokumentováno v kap. 9, neodpovídá požadavku „realistických i netriviálních umělých optických soustav“. Oddíl 9.1 začíná “Prvním testem bude...” ale nenásleduje žádný další test. Celkově je kap. 9 spíše uživatelskou příručkou než zprávou o testování – chybí mj. ověření, že fyzikální model funguje správně.

Realizovaný program je funkční a i relativně intuitivní co se týče ovládání, pouze práce s 3D zobrazením je dosti nepřehledná. Při použití obsluhou znalou jeho účelu bude aplikace pravděpodobně vhodným nástrojem pro demonstrování principů paprskové optiky a práce s hologramy.

Implementace vhodně využívá architekturu Model-View-Controller, až na jednu přímou závislost z Model do View (třída model.Shape obsahuje seznam objektů gui.workspace.Face). Mechanismus rozšiřitelnosti aplikace je jednoduchý ale zřejmě vyhovující vzhledem k účelu aplikace. Budoucí rozšiřitelnost by nicméně usnadnilo použití rozhraní, která autor v implementaci vůbec nevyužívá.

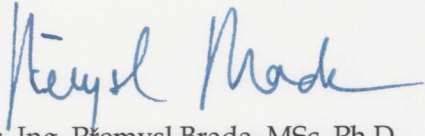
O jisté programátorské nevyzrállosti autora svědčí také to, že se v implementaci občas vyskytují nevhodné konstrukce (opakované přetěžování, mrtvý kód, neužitečné komentáře typu `/** Inicializační metoda. */ void init() {...}`, apod.) a zbytečně pracná „ruční“ implementace standardních činností místo využití. Např. pro persistenci nastavení a objektů by bylo vhodnější použít `java.util.Properties` nebo knihovny pro práci s textovými daty (YAML, JSON). Očekával bych také použití automatizovaných testů pro ověření výpočtů na úrovni fyzikálního modelu.

V souhrnu tedy konstatuji, že text práce má nedostatky vzhledem k jejímu účelu, nicméně jako celek předložená diplomová práce naplňuje zadání a je dobře využitelná pro zamýšlený účel. Práci hodnotím stupněm **velmi dobře** a doporučuji k obhajobě.

Doplňující otázky k obhajobě:

1. Text zásad pro vypracování uvádí, že v *ideálním* případě by měl simulátor fungovat v podobě jak webové tak desktopové aplikace. Proč tento cíl nebyl naplněn?
2. Bylo by pro výpočet sledování paprsku (ray tracing) možné využít nějakou již dostupnou knihovnu, a jaké by to mělo výhody či zápory?
3. Můžete vysvětlit kostru mechanismu spolupráce jednotlivých tříd všech vrstev při zobrazování simulace, např. pro scénář „aktualizace scény při přidání jednoho prvku optické soustavy“?

V Plzni 3.6.2015

  
doc. Ing. Přemysl Brada, MSc. Ph.D.