Posudek oponenta diplomové práce

Michal HALAŠA: Visualizátor struktury webu

Cílem této práce bylo navrhnout a implementovat metodu vizualizace rozsáhlého orientovaného grafu. Typicky grafu webu. Program by měl dle zvolených kritérií zobrazovat jen určité odkazy nebo webové stránky. Zahrnuta měla být podpora shlukování skupin stránek pro zjednodušení struktury.

Práce je odevzdávaná již potřetí. Autorovi se podařilo vcelku dobře shrnout analytickou a specifikační část práce. V nich jsou popsány důvody pro potřebu vizualizace grafů, existující možná řešení, jejich výhody a nevýhody. Autor se – na základě studia vlastností prezentovaných algoritmů – rozhodl pro implementaci algoritmu H3.

Bohužel, designu vlastního programu se autor v práci věnuje velmi okrajově. V sekci 5.7 na straně 34 je možné se dočíst, že „pro reprezentaci vrcholů, hran grafu a jejich vlastností (jako například barva, viditelnost, souřadnice v Euklidovském a hyperbolickém prostoru) byla použitá pole s přesně definovaným pořadím prvků v nich". Jak ta pole vypadají, jaké je to pořadí a jak se z těchto dat připraví linie a trojúhelníky pro vykreslení pomocí OpenGL už ale práce nezmiňuje. Jen s obtížemi si představuji, že se někdo bude v budoucnosti prokousávat zdrojovými texty (které komentáři rovněž neoplývají) a namáhavě zjišťovat, jak program vlastně vnitřně funguje, aby jej mohl upravit. Programátorská příručka se sestává prakticky z jediné stránky, popisující strukturu adresářů se zdrojovými kódy. Lepšího efektu a úplnější „příručky" by šlo dosáhnout pomocí JavaDoc, nejlépe pokud by si autor dal práci a program pořádně okomentoval, včetně všech tříd, metod a parametrů. Design práce se měl věnovat modulární struktuře programu, pokud nějaká existuje, popisu datových struktur ve vnitřní paměti a podobně.

Jako grafickou knihovnu pro komunikaci s 3D akcelerátorem byla zvolená Java3D komponenta. To se ukázalo jako poměrně komplikované kvůli chybám v samotné knihovně, případně komunikaci mezi Javou a knihovnou psanou v nativním jazyce cílové platformy. Je otázka, zda by se nenašla nějaká stabilnější implementace OpenGL pro Javu, do které by vykreslovací část programu mohla být přepsána. Pokud je program dobře navržen, nemělo by být zase až tak těžké vypočítanou geometrii do HW pomocí jiného OpenGL wrapperu protlačit. Zvláště, když Java3D s velkými daty (které byly důvodem zvolení H3 algoritmu) vcelku pravidelně padá. Například při pokusu o změnu velikosti okna, ale i v řadě dalších situací.

Nebylo mi jasné, proč – jak se uvádí na straně 35 – nebylo možné z licenčních důvodů umístit na CD data z Egothoru, když se jedná o robota s otevřeným zdrojovým kódem, a navíc data neobsahují texty stránek, pouze informace o tom, že stránka X odkazuje na stránku Y, což se dá na webu snadno zjistit. Navíc, když už by chtěl autor práce tuto (veřejnou) informaci utajit, mohl URL nějak překódovat.

Obrázek „Texty odkazů na stránkách" na straně 36 jsou kopií předchozího obrázku. Také by mne zajímalo, jak je to s možností shlukování uzlů pro zjednodušení zobrazované struktury.

Celkově mám z práce rozporuplné pocity. Na jedné straně autor ukázal svoji schopnost programování netriviálních algoritmů s využitím jistě netriviálních datových struktur. Na straně druhé nedokázal v práci samotné srozumitelně popsat, jak tyto struktury vypadají a jak program vnitřně funguje, ani jakým způsobem se řeší požadované filtrování zobrazení nebo shlukování, čímž podstatně zkomplikoval možnosti dalšího využití odevzdaného softwarového díla.

Přes tyto výtky doporučuji práci k obhajobě.

V Praze dne 25. 1. 2010

Poznámka: Na CD bych uvítal Javu ve verzi, na které byl jako Java3D knihovnu.