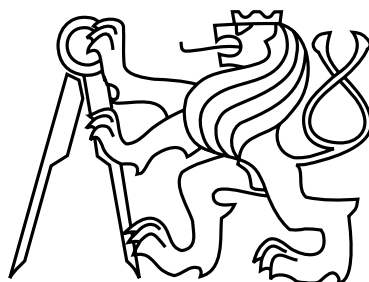


České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra počítačů



Bakalářská práce

Internetová stránka pro porovnávání extenzí

Jan Fiala

Vedoucí práce: Ing. Petr Felkel, Ph.D.

Studijní program: Softwarové technologie a management, Bakalářský

Obor: Web a multimedia

25. května 2011

Prohlášení

Prohlašuji, že jsem práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 25. 5. 2011

.....

Abstract

This work deals with creating a web site containing information about OpenGL extensions. The site should include a current list of all known extensions and all their specification documents. Additional content can be added by users, who will be able to add code examples and upload executable tests for any extension. They will also be able to contribute to discussions.

Abstrakt

Tato práce se zabývá vytvořením webových stránek pro zjišťování informací o extenzích OpenGL. Stránky mají obsahovat aktuální seznam všech známých extenzí, jejich specifikace a další obsah, který budou vytvářet uživatelé. Ti budou moci ke každé extenzi přidávat ukázky kódu, nahrávat soubory spustitelných aplikací, které extenzi testují, a vést o nich diskuse.

Obsah

1	Úvod	1
2	Cíle a požadavky	2
2.1	Funkční požadavky	2
2.2	Nefunkční požadavky	2
2.3	Testování	2
3	Analýza a návrh řešení	3
3.1	Existující implementace	3
3.2	Návrh obsahu stránek	4
3.2.1	Sekce specifikací	4
3.2.2	Sekce ukázek kódu	5
3.2.3	Sekce testů extenze	5
3.2.4	Sekce diskuse	5
3.2.5	Ostatní	5
3.3	Získávání obsahu	5
3.3.1	Seznam extenzí	5
3.3.2	Specifikace extenzí	6
3.4	Uživatelé a jejich pravomoce	6
3.5	Uživatelské rozhraní	7
3.5.1	První návrh uživatelského rozhraní	8
3.5.2	Druhý návrh uživatelského rozhraní	8
3.6	Volba implementačního prostředí	8
3.7	Návrh databáze	9
4	Realizace	11
4.1	Stručný postup tvorby	11
4.2	Vytvoření databáze	12
4.3	Sada funkcí pro manipulaci s databází	13
4.4	Stahování a parsování seznamu extenzí	15
4.5	Stahování a parsování specifikací	15
4.6	Manuální update	17
4.7	Registrace uživatele	17
4.8	Sekce diskusí	17
4.9	Sekce testů extenzí	18
4.10	Uživatelské rozhraní, design	20

4.11	Testování nehotové aplikace	20
4.11.1	Výsledky testování	21
4.11.2	Závěr testování	22
4.12	Vkládání a zobrazování ukázek kódu	22
4.13	Zobrazení výsledků testů extenzí	23
4.14	Nahrávání souborů	25
4.15	Vyhledávání	25
4.16	Aktualizace	26
5	Testování	27
5.1	Příprava pro test s uživateli	27
5.1.1	Cíl testu	27
5.1.2	Screeener	27
5.1.3	Pre-test dotazník	27
5.1.4	Nastavení testu a popis rolí	28
5.1.5	Výchozí stav aplikace	28
5.1.6	Seznam úloh	28
5.1.7	Post-test dotazník	28
5.2	Provedení testu s uživateli	28
5.2.1	Reálné podmínky testu	28
5.2.2	Vyhodnocení testu s uživateli	29
6	Závěr	31
6.1	Shrnutí splnění cílů	31
6.2	Možnosti dalšího vývoje aplikace	31
A	Seznam použitých zkratk	33
B	Instalační příručka	34
C	Obsah přiloženého CD	35
D	Testování: Veřejná část Screeeneru	36
E	Testování: Neveřejná část Screeeneru	38
F	Testování: Pre-test dotazník (ústně)	39
G	Testování: Seznam úloh	40
H	Testování: Post-test dotazník (ústně)	41
I	O dočasném umístění projektu	42

Seznam obrázků

3.1	První návrh rozložení obsahu stránky	7
3.2	Druhý návrh rozložení obsahu stránky	8
3.3	Návrh databáze v programu ER Modeller 4.2	10
4.1	Ukázka výsledku parsování specifikace	16
4.2	Ukázka zobrazení jednoho testu extenze	18
4.3	První verze vzhledu stránek	19
4.4	Druhá verze vzhledu stránek s relativní šířkou	20
4.5	Ukázka využití Syntax Highlighter	23
4.6	Ukázka fungování filtru extenzí	26

Kapitola 1

Úvod

OpenGL je grafická knihovna, navržena firmou SGI jako aplikační programové rozhraní k akcelerovaným grafickým kartám. Jedna z největších výhod OpenGL je, že bylo navrženo jako snadno rozšiřitelné, aby snadno zahrnuje nové hardwarové inovace. Každé takové rozšíření, extenze, je popsáno jako soubor změn oproti určité specifikaci OpenGL. Každý tento soubor změn, který tuto extenzi tvoří, je pak popsán ve specifikačním souboru dané extenze. Specifikační soubory všech známých extenzí udržuje OpenGL ARB na svých stránkách OpenGL Extensions Registry [1].

Tato bakalářská práce byla zadána v souvislosti se zadáním dvou dalších diplomových prací, ve kterých mají studenti tvořit programy, které různými způsoby testují rychlost a efektivnost vybraných extenzí OpenGL, a to jak v závislosti na konfiguraci počítače a typu grafické karty, tak i porovnání podobných extenzí mezi sebou. Původně bylo součástí jejich práce i vytvoření stránek, na kterých by své programy prezentovali a kam by se v budoucnu mohly nahrávat další podobné testy, jak se ale ukázalo, byl by takovýto projekt příliš časově náročný a typově velmi odlišný od jejich hlavního zadání. V konzultaci se zadávajícím pak byla tvorba této aplikace vypsána jako samostatné téma.

Kapitola 2

Cíle a požadavky

2.1 Funkční požadavky

Cílem této práce je vytvoření internetové aplikace, která bude obsahovat databázi extenzí OpenGL. Ke všem extenzím bude v aplikaci existovat jejich specifikace, ne však v originální ASCII textové podobě, která není pro běžné používání příliš přehledná, ale ve formátu HTML, s graficky odlišenými kapitolami a zvýrazněnými odkazy na jiné specifikace. Tyto specifikační dokumenty musí být udržovány aktuální.

Ke každé extenzi musí existovat sekce s příklady užití extenze, které budou přidávat uživatelé. Dále extenze obsahuje i sekci s testy, které se k této extenzi vztahují, a možnost k těmto testům přidávat a prohlížet uživatelské výsledky. Uživatelé budou také moci o každé extenzi vést diskusi.

2.2 Nefunkční požadavky

Aplikace bude kompletně v Angličtině.

Webové stránky se budou korektně zobrazovat v aktuálních verzích všech nejvýznamnějších internetových prohlížečích (v době dokončení aplikace), konkrétně Internet Explorer, Mozilla Firefox, Google Chrome a Opera.

Webové stránky budou v omezené míře fungovat i pro zastaralé verze prohlížečů (Internet Explorer 6) a pro prohlížeče s vypnutým zpracováním JavaScriptu.

Stránky budou validní vůči vybrané specifikaci HTML. Kvůli tématu (grafické knihovny) stránky nemusí nutně splňovat všechna doporučená pravidla, týkající se uzpůsobení webu pro uživatele s těžkým zrakovým postižením.

2.3 Testování

Po vytvoření systému nastane fáze testovací, kde se otestuje použitelnost stránky. Po provedení analýzy výsledů těchto testů je třeba stanovit správnost navržené jednotlivých částí systému, odhalit možné chyby a případně navrhnout změny a vylepšení.

Kapitola 3

Analýza a návrh řešení

3.1 Existující implementace

Jedním z důvodů, proč bylo rozhodnuto vytvořit tuto webovou aplikaci, byla právě absence podobně komplexního projektu. Žádná z nalezených stránek nenabízí jak kompletní databázi extenzí pro OpenGL a jejich specifikace, tak možnost o extenzích diskutovat a zároveň přidávat testy a jednoduše porovnávat výsledky. Tento systém mohla nahrazovat pouze kombinace samotné databáze extenzí a odborného webového fóra.

Protože se mi žádnou takovou kombinaci nalézt nepodařilo, pokusil jsem se zjistit, jestli existují stránky, které by mohly být porovnány alespoň s jednotlivými částmi mého projektu. První částí, na kterou jsem se zaměřil na samotnou databázi extenzí.

V sekci extenzí na oficiálních stránkách OpenGL [1] se vyskytuje pouze očíslovaný seznam extenzí, odkazující na textové specifikace. Texty těchto specifikací jsou mnohdy stovky řádků dlouhé, a protože jsou formátované pouze ASCII, jsou velmi nepřehledné. Před samotným seznamem extenzí je upozornění, že se chystá přechod z formátu textového na modernější XML.

Soudě dle mnoha již neaktivních odkazů, existovala alternativní databáze extenzí i na webových stránkách společnosti SGI, která je tvůrcem OpenGL. Na tento web odkazovaly i stránky, které zároveň odkazovaly na oficiální stránky OpenGL, podle čehož se dá soudit, že zde byl také seznam extenzí a jejich specifikací, a to i těch, které ve své době v seznamu OpenGL Registry nebyly. V současné době však tyto stránky neexistují a pouze přesměrovávají na stránky OpenGL. Podle typů odkazu lze ale jistě říci, že i zde se specifikace vyskytovaly pouze v podobě textové.

Další seznam extenzí je možné nalézt na stránkách společnosti Apple [5]. Tento seznam má sice na jednu stranu nevýhodu v tom, že obsahuje pouze extenze, které podporuje operační systém MacOS, na stranu druhou je zde ale lehčí vyhledávání, a to nejen kvůli redukovanému počtu extenzí, ale i kvůli šikovnému rozdělení extenzí podle funkcí a objektů, se kterými extenze pracují. Po vybrání extenze se uživateli zobrazí její krátký popis. Více informací a konkrétní specifikaci extenze ale zajišťuje pouze odkaz na textovou specifikaci na stránkách OpenGL, nebo v horším případě nefunkční odkaz na specifikaci na výše zmíněných stránkách SGI.

Pouze jeden objevený projekt se zabývá zpřehledněním specifikací [4]. Tento projekt jako jediný obsahuje specifikace extenzí zparsované do HTML, což je i jeden z úkolů, kterým se budu zabývat já. Přehledně jsou označeny všechny odstavce ve specifikaci, v samotném textu jsou pak vždy rozpoznána jména ostatních extenzí a převedena na odkazy, což činí navigaci po specifikacích mnohem snazší. Samotný seznam obsahuje pouze názvy extenzí bez očíslování a bez možnosti vyhledávání. Bohužel jsou ale nyní stránky neaktuální, naposledy byl totiž seznam vygenerován v roce 2005.

Druhou část tvoří stránky, které by řešily použití extenzí a případně by mohly sloužit pro umístění testů extenzí, jejich výsledků a porovnání. Zde jsem objevil pouze relativně novou oficiální Wiki OpenGL[2] a obecná internetová fóra, kde se v konkrétních vláknech řeší pouze specifické problémy a příklady konkrétních extenzí. Pro účely zveřejňování testů extenzí by oficiální OpenGL Wiki použita být mohla, nebyla by ale pravděpodobně dostatečně uživatelsky jednoduchá pro uživatele, kteří by chtěli pouze zveřejnit výsledky těchto testů na své konfiguraci.

3.2 Návrh obsahu stránek

Na této webové aplikaci je nejdůležitější seznam extenzí, který bude tvořit hlavní obsah úvodní strany. Uživatel se totiž potřebuje co nejrychleji dostat na stránku extenze, o které potřebuje zjistit informace. Stejný seznam, jako je na oficiálních stránkách OpenGL, se jeví s určitými doplňky a úpravami jako ideální. Kvůli počtu extenzí (téměř 500) bude potřeba zvýšit počet extenzí, který se vejde na jednu stranu a jednoduchý způsob vyhledávání. Seznam extenzí je rozdělen na dvě hlavní a jednu doplňkovou sekci. Dva hlavní seznamy tedy bude (vzhledem k dnešnímu rozšiřování obrazovek do stran) logické umístit vedle sebe. Pro vyhledávání by mohl sloužit jednoduchý filtr, který bude vybírat pouze ty extenze, které vyhovují zadanému výrazu a ostatní skryje. Tento filtr by vyhledával pouze podle jmen extenzí. Složitější vyhledávání by bylo řešeno zvlášť.

Téměř veškerý obsah se bude vázat k jednotlivým extenzím. Návrhem tedy je, aby každá extenze měla svoji vlastní stránku a k ní navázáno několik „karet“. Tyto podsekcce by měly být v základu čtyři. První podsekcce by tvořila samotná zparsovaná specifikace, druhou příklady použití extenze, třetí testy této extenze a nakonec jednoduchá diskuse pro uživatele, vázaná na tuto extenzi.

3.2.1 Sekce specifikací

Tato sekce by, kromě strany hlavní, byla jedinou stránkou, tvořenou a upravovanou pouze systémem. Původně textová specifikace zde bude transformovaná do formátu HTML. Specifikace bude rozdělena na jednotlivé kapitoly, které půjdou pro přehlednost skrývat. V textu budou označené důležité odkazy, jako například odkazy na specifikace verzí OpenGL, nebo například odkazy na jiné extenze v kapitolách Dependencies.

3.2.2 Sekce ukázek kódu

Obsah sekce ukázek použití extenzí bude tvořen pouze uživateli. Každá ukázka kódu bude opatřena jménem a komentářem autora. Uživateli by zde mělo stačit vkládání holého textu, u označené části kódu se pak systém postará o zvýraznění syntaxe. Ke každé ukázce kódu budou moci ostatní uživatelé přidávat vlastní komentáře.

3.2.3 Sekce testů extenze

Sekce testů bude také spravována uživateli. Každý test bude mít právě jednoho autora a tělo zobrazeného testu bude mít několik částí. Tvůrce testu zde už nebude zadávat pouze holý text, jako tomu bude u ukázek kódu. Zde bude možnost text formátovat, vkládat externí obrázky a tvořit tabulky. Důležitá bude možnost nahrávat do systému vlastní soubory. Typy souborů, které budou povoleny, se určí později po konzultaci s prvními tvůrci testů. Předpokládám však obrázky a především spustitelné soubory se samotnými testy.

Ke každému testu půjdou nahrávat výsledky testů (naměřené časy), které se pod testem budou zobrazovat. Výsledky testů budou tvořeny standardizovanými XML soubory. Tyto soubory budou přímými výstupy spustitelných testů a systém bude nastaven tak, aby je převáděl do tabulek HTML.

Ke každému testu bude pak totožným způsobem jako u ukázek kódu přidávat jednoduché textové komentáře.

3.2.4 Sekce diskuse

V sekci diskuse budou uživatelé diskutovat na téma dané extenze. Bude možné zakládat takzvaná „vlákna“. Kořenové komentáře budou zvýrazněné více, než odpovědi na ně. Více vláken se ale bude zobrazovat na jedné stránce, nepůjde tedy o samostatná fóra ve stylu phpBB pro každou extenzi.

3.2.5 Ostatní

Ostatní stránky systému budou tvořit většinou formuláře na přidávání a editaci obsahu a administraci projektu. Vlastnosti a obsahy stránek pro editaci obsahu obyčejnými uživateli vyplývají z předchozích odstavců. Role uživatelů a jejich pravomoce budou jsou určeny níže.

3.3 Získávání obsahu

3.3.1 Seznam extenzí

Jedním z hlavních požadavků je udržovat aktuální seznam všech existujících extenzí a jejich specifikací. Stránky, které aktuální seznam existujících extenzí udržují, jsou v současné době pouze jedny – OpenGL Registry [1]. Jak bylo zmíněno výše, dříve existoval ještě jeden seznam na stránkách společnosti SGI, ale ten je v současné době již nefunkční a pouze odkazuje na

stránky OpenGL. Aplikace tento hlavní seznam tedy bude kompletně a pravidelně přebírat přímo z této stránky.

Tento kompletní seznam se nachází na jediném dokumentu HTML a pravděpodobně bude třeba jména extenzí a odkazy na jejich specifikace získávat přímo hledáním klíčových slov v textu, protože přestože je dokument hlavičkován jako XHTML, validní není a nemůže tedy být jednoduše zpracován jako XML. Dodržuje totiž pouze syntaxi HTML 4.1 nebo nižší. Problémem jsou především atributy tagů `` v číslovaných seznamech extenzí, kterým oproti standardu XHTML chybí uvozovky a v některých případech obsahují jejich hodnoty kromě čísel i písmena, které správně obsahovat nemají.

3.3.2 Specifikace extenzí

Do té doby, než specifikace přejdou na formát XML, jak je plánováno, bude třeba specifikace zpracovávat z textových dokumentů. Formátování těchto textových specifikačních dokumentů má určitá základní pravidla. Těmi hlavními jsou:

- Jsou standardizované základní názvy kapitol (např. Name, Name Strings, Overview)
- Obyčejný text je od levého okraje odsazen o čtyři mezery
- Nadpis kapitoly by měl být jediným druhem odstavce, který od levého okraje není odsazený
- Jednotlivé odstavce a nadpisy jsou od sebe odděleny prázdným řádkem.
- Existuje maximální počet znaků na řádek. Tento počet však mohou překročit ASCII tabulky

Bohužel je mnoho specifikačních souborů, které tato pravidla nedodržují. Porušená jsou v různých souborech téměř všechna pravidla, ať už jde o nesprávná pojmenování odstavců, neodsazování obyčejných textů, neoddělené nadpisy od odstavců a podobně. Parser, který bude tyto specifikace zpracovávat, je musí zpracovávat podle daných standardů, zároveň ale bude muset odhalovat alespoň ty nejčastější formátovací chyby.

Všechny tyto chyby ale obsáhnout nepůjde. Extenzí a jejich specifikačních dokumentů je totiž skoro 500 a počty řádků každé této specifikace se pohybují v řádu několika set až tisíc. Z tohoto počtu extenzí tedy vyberu několik zástupců náhodně a v průběhu vývoje parseru budu zjišťovat a opravovat nejčastější chyby.

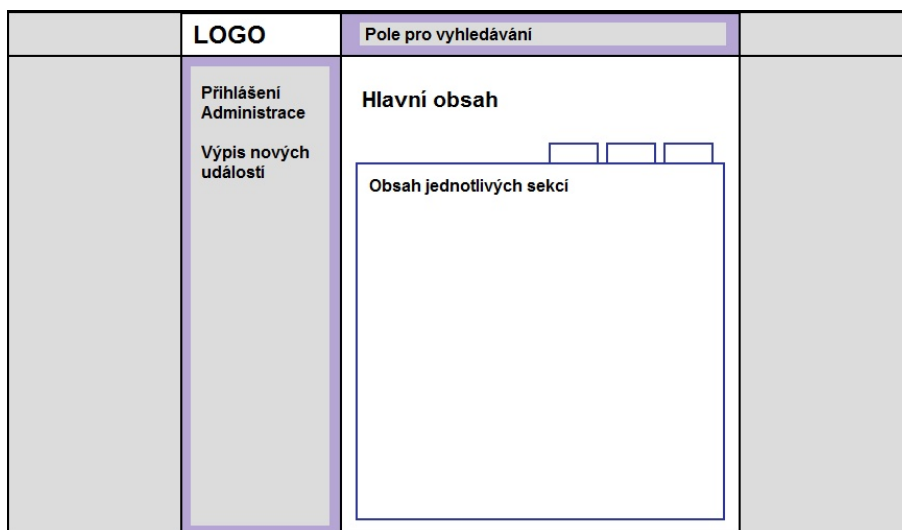
3.4 Uživatelé a jejich pravomoce

Jakým způsobem rozdělit typy uživatelů a jaké jim přiřadit pravomoce, aby byla nejmenší pravděpodobnost, že bude docházet k jejich zneužíváním, je velká otázka, kterou se mi nepovedlo zcela vyřešit. Připravil jsem tedy systém, který mi se mi zdá relativně kontrolovatelný, ale zároveň jsem si jistý, že až se aplikace dostane do ostrého provozu, může se tento systém uživatelských pravomocí projevit jako zcela nedostatečný a bude muset být upraven. Záležet bude kromě jiného na počtu uživatelů.

Systém bude tedy rozlišovat čtyři typy uživatelů:

- Neregistrovaný uživatel bude moci procházet stránky a číst veškerý obsah. Žádný vlastní obsah ale vkládat smět nebude. Bude se smět samozřejmě zaregistrovat.
- Registrovaný uživatel bude moci zakládat vlákna v diskusích, přispívat do nich, vkládat ukázkové příklady a přidávat k nim komentáře, komentovat testy a vkládat své výsledky testů. Registrovaný uživatel může vložit i test, který ale nebude ihned zobrazen. Nejprve bude muset být schválen dvěma důvěryhodnými uživateli. Vzhledem k tomu, že bude smět vkládat tyto skryté testy, bude smět i nahrávat vlastní soubory. Je to sice zneužitelné, ale relativně nutné, takže pokud by nedocházelo k masivnějšímu zneužívání, tuto možnost bych pro registrované uživatele otevřenou nechal.
- Registrovaným-důvěryhodným uživatelem se registrovaný uživatel stane, pokud jsou schváleny alespoň dva jeho vložené testy. Důvěryhodný uživatel smí schvalovat testy, tedy i ty svoje, stále ale potřebuje alespoň jednoho dalšího důvěryhodného uživatele, aby daný test schválil. Důvěryhodný uživatel smí také mazat komentáře obyčejných registrovaných uživatelů.
- Administrátor smí upravovat a mazat veškerý obsah, stejně jako důvěryhodný smí testy schvalovat. Uživatele může mazat a měnit jim role. Má také k dispozici nástroj na manuální aktualizaci seznamu a specifikací extenzí.

3.5 Uživatelské rozhraní



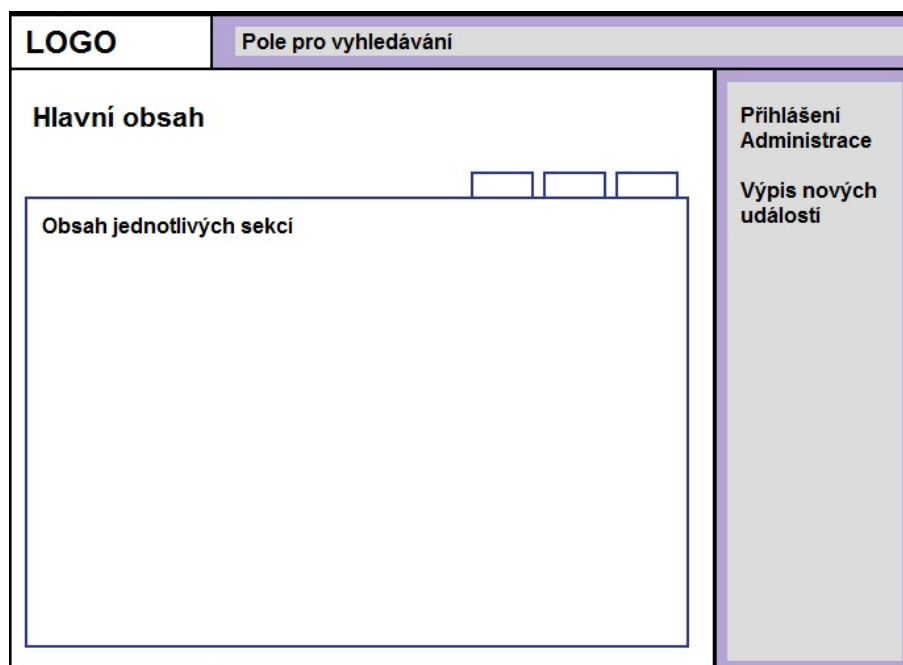
Obrázek 3.1: První návrh rozložení obsahu stránky

Vznikly celkem dvě mírně odlišné verze uživatelského rozhraní. První se při prvním testování ukázala jako méně vhodná, proto se u finální verze se využívá verze druhá (kromě zobrazení webu v prohlížeči Internet Explorer 6, kde druhá šablona kvůli špatnému zpracování CSS nefungovala a zůstalo se proto u verze první). Okolní obrázky zobrazují návrhy obou verzí.

3.5.1 První návrh uživatelského rozhraní

Uživatelské rozhraní bude navrženo a vytvářeno především pro zobrazení v prohlížečích na stolních a přenosných počítačích. Dílčí úpravy pro zobrazení na mobilních přístrojích budou provedeny dodatečně. Návrh první verze počítal s pevnou šířkou hlavního okna stránky, nastavenou na 1000 pixelů. Důvod byl ten, že i malé netbooky v současné době mají šířku rozlišení alespoň 1024 pixelů a stránka se tak na ně vždy vejde, na větších monitorech pak pevná šířka zabrání přílišnému roztažení textu, protože příliš široké odstavce textu se nečtou velmi dobře.

Horní pruh je vyhrazen logu projektu a vyhledávací liště. Místo pod horním pruhem vyplní z části pruh levý o velikosti 300px. V tomto pruhu leží přihlašování a funkce související s rolí uživatele, pod ní je pak seznam změn na stránkách. Zbylé místo tvoří hlavní obsah, jehož organizace je proměnlivá v závislosti na sekci, ve které se uživatel vyskytuje.



Obrázek 3.2: Druhý návrh rozložení obsahu stránky

3.5.2 Druhý návrh uživatelského rozhraní

Druhý návrh je téměř totožný s prvním, pouze přesunuje levou lištu na pravou stranu a zmenšuje ji. Zbylý obsah má také místo pevně dané šířky velikost relativní, využije tak lépe celý prostor okna prohlížeče.

3.6 Volba implementačního prostředí

Tento systém má být webovou aplikací, proto je možné ho implementovat v mnoha různých jazycích, od PHP, přes Ruby on Rails až po jazyk Java. Z těchto možností jsem si vybral

právě jazyk PHP, který budu na klientské straně doplňovat jazykem JavaScript. Největší výhodou jazyka PHP je jeho přenositelnost a rozšířenost.

Podle předpokládaného návrhu UI budou všechny stránky zasazené do jedné jednoduché šablony, proto není dle mého názoru nutné používat framework a jednoduchý systém oddělení aplikační logiky a prezentace vypracuji sám.

Jako databázi jsem zvolil MySQL a pro přístup do ní budu pro mnohé výhody využívat PDO [8], které je součástí PHP od verze 5.1.

Jako WYSIWYG editor HTML pro vkládání a editaci doprovodného textu k uživatelským testům budu využívat JS aplikaci TinyMCE [7], pro snadnou přizpůsobitelnost a předchozí kladné zkušenosti.

Pro některé moderní efekty přechodů a animací na stránce využiji JavaScriptové knihovny Script.aculo.us, a pro zvýrazňování syntaxe mnoha programovacích jazyků aplikaci Syntax Highlighter [9].

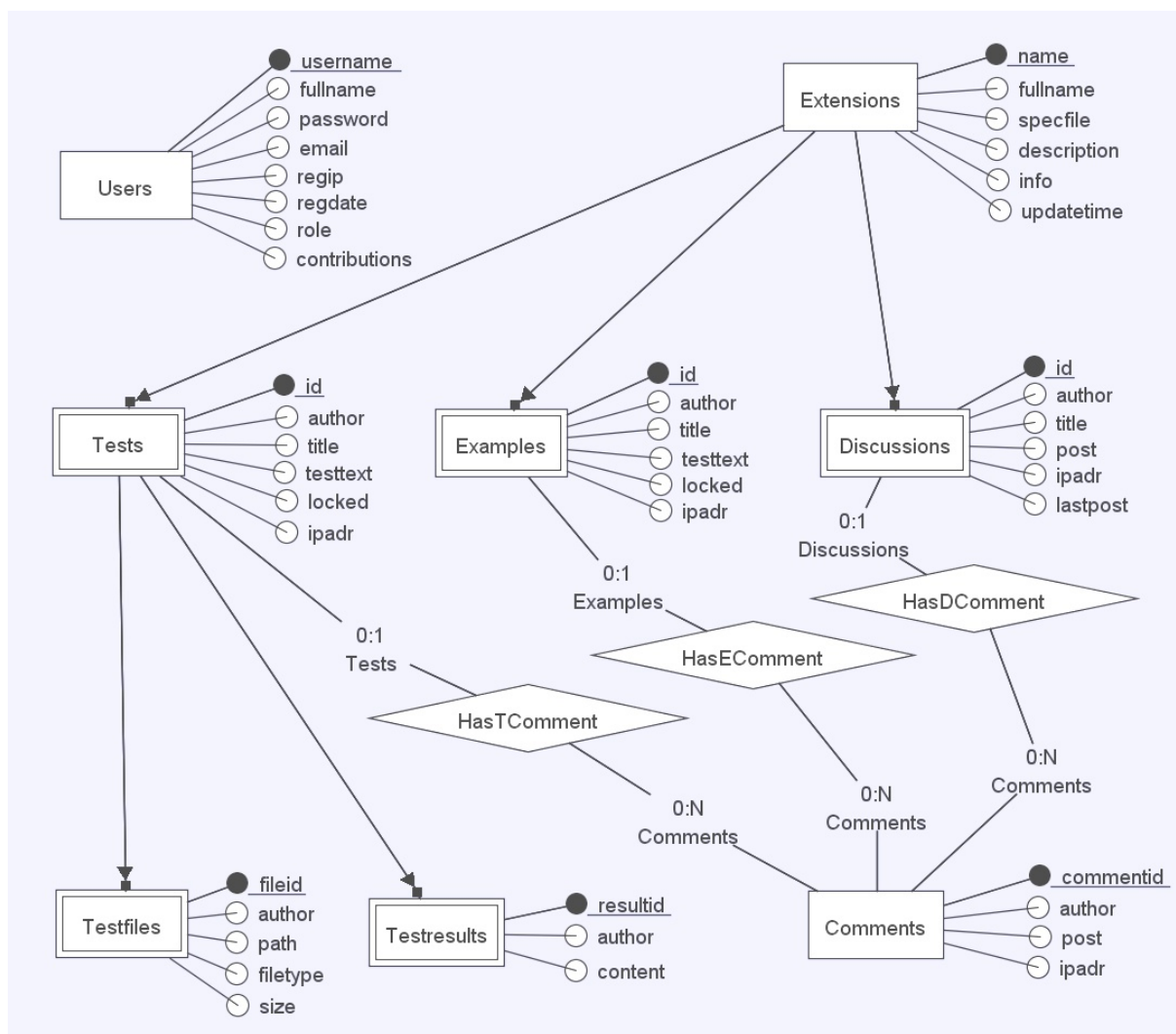
Jako verzi HTML jsem zvolil XHTML 1.0 Transitional, zejména kvůli tomu, že verze Strict zakazuje tagy „iframe“, které jsou podle mne užitečné a ve specifických případech nenahraditelné (i proto jsou v novém HTML 5 tagy iframe opět povoleny) – teoreticky by mohly být nahrazovány tagem „object“, ale podpora většiny prohlížečů pro takové řešení je nedostatečná. Dále bývá „iframe“ nahrazován AJAXem, avšak přes AJAX nelze od klienta na server nahrávat soubory, což je právě jedna z funkcí, kvůli které plánuji „iframe“ využít.

U kaskádových stylů se budu držet převážně verze CSS 2, občas možná použiji pro přehlednější zobrazení některé novější značky. Změny to ale budou jen „kosmetické“ a v prohlížečích, které CSS 3 nepodporují, by si člověk neměl všimnout výraznějších rozdílů.

3.7 Návrh databáze

Pro návrh databáze jsem využil program ER Modeller 4.2.

Hlavní obsah databáze závisí na tabulce extenzí. Každá extenze může mít libovolný počet testů, ukázek kódu a vláken diskuse, jejichž struktura je velmi podobná. Všechny tři tabulky mohou mít libovolný počet komentářů. Tabulce testů jsou navíc přiřazeny i tabulky s informacemi o nahraných souborech a výsledcích testu. Sloupec "author" u většiny tabulek odkazuje na uživatelské jméno "username" v tabulce uživatelů. Pro zachování přehlednosti návrhu není tento vztah nijak vyznačen.



Obrázek 3.3: Návrh databáze v programu ER Modeller 4.2

Kapitola 4

Realizace

4.1 Stručný postup tvorby

Nejprve jsem podle návrhu vytvořil seznam MySQL příkazů pro vytvoření databáze. Poté jsem vytvořil sadu funkcí v PHP s předpřipravenými příkazy pro manipulaci s vytvořenou databází. Dále bylo nutné databázi naplnit údaji o extenzích, následovalo tedy vytvoření několika parserů, které zpracovávají informace z oficiálních stránek OpenGL.

Hlavním byl parser hlavní strany OpenGL Registry [1], který v HTML kódu nalezne všechny extenze a odkazy na jejich specifikační soubory. Druhý parser zpracovává samostatné specifikace extenzí, kdy rozděluje obsah na kapitoly, převádí obsah do formátu HTML. Pro otestování funkčnosti těchto funkcí vznikla sekce „Manuální aktualizace“, využívající JavaScript a AJAX, kterou později bude moci používat administrátor.

Dále byl vytvořen systém pro registraci a přihlašování uživatelů, který pro registraci využívá nástroj od firmy Google – ReCaptcha. Lehce upravena musela být databáze a předpřipravené příkazy. Následoval vznik sekce diskuse a uživatelských testů. V první podobě vzniklo také nahrávání uživatelských výsledků testů.

Na konci první fáze byl vytvořen design pomocí upravení kostry HTML a sepsání kaskádových stylů CSS. Pro sekce specifikace byly vytvořeny ovládací prvky používající JavaScript a JavaScriptové knihovny Script.aculo.us [10].

Představení první podoby stránek studentům, vytvářejícím obsah stránek – testy extenzí, proběhlo formou jednoduchých uživatelských testů, díky kterým jsem si mohl vytvořit lepší pohled na použitelnost webové aplikace. Z těchto testů vyplynulo mnoho problémů, které bylo potřeba vyřešit, a které ovlivnily další průběh tvorby aplikace. Jednalo se především o práva a kategorie uživatelů a formát výsledků uživatelských testů.

Po tomto prvním testování jsem nejprve dokončil sekci s uživatelskými ukázkami kódu extenze, kde jsem zvolil podobný formát jako u uživatelských testů. Pro zvýraznění syntaxe jsem zde použil JavaScriptovou aplikaci SyntaxHighlighter [9].

Na konzultaci se studenty, kteří budou tvořit testy extenzí, jsme po mnoha zvažovaných podobách zvolili nový formát výsledků testů - XML. Tento návrh studenti zapracovali do svých aplikací a dodali mi své výstupní XML. Pro tento soubor jsem vytvořil validační XML Schema a PHP parser, převádějící tyto výsledky do tabulek HTML. Opět zde byla nutná úprava databáze.

Přepřeváděna byla poté také šablona CSS a celkový vzhled, pro lepší zobrazení na širších i menších monitorech při odlišných typech a velikostech písma.

Pro sekci testů extenzí jsem dále vytvořil jednoduchý systém pro nahrávání souborů.

Pro vyhledávání jsem vytvořil vlastní sekci, která zobrazuje výsledky Google Custom Search [3]. Vzhledem dlouhé době zaindexování jsem tuto funkci ale nemohl vyzkoušet.

Na konci samotné implementace jsem vytvořil systém automatické aktualizace extenzí a jejich specifikací v PHP, založený na „Visitor Based webcron solution“ návrhu. Tedy aktualizaci založenou na návštěvách stránek.

Použitelnost hotové aplikace jsem nejprve krátce otestoval sám, pomocí metody Kognitivního průchodu. Poté jsem provedl testy použitelnosti s uživateli. Nalezené nedostatky jsem se podle míry závažnosti a rozsahu pokusil opravit, nebo je určil jako další možnosti vývoje aplikace.

4.2 Vytvoření databáze

Pro manipulaci s databází a její počáteční úpravy jsem zvolil prostředí phpMyAdmin. Následující MySQL příkazy, které vytvářejí potřebné tabulky, může být proto při použití jiných způsobů zadávání nutně lehce upravit – použitím jiných typů uvozovek. Tabulky přibližně odpovídají mému návrhu, zde je vypsáno pouze několik tabulek, ostatní je možno prohlédnout na přiloženém CD.

```
CREATE TABLE 'Extensions' (  
  'name' VARCHAR( 60 ) NOT NULL ,  
  'fullname' VARCHAR( 60 ) NOT NULL ,  
  'specfile' VARCHAR( 60 ) NOT NULL ,  
  'description' VARCHAR( 700 ) NULL ,  
  'info' VARCHAR( 100 ) NULL ,  
PRIMARY KEY ( 'name' )  
) ENGINE = MYISAM ;  
  
CREATE TABLE 'Users' (  
  'username' VARCHAR( 20 ) NOT NULL ,  
  'fullname' VARCHAR( 40 ) NULL ,  
  'password' VARCHAR( 40 ) NOT NULL ,  
  'email' VARCHAR( 30 ) NOT NULL ,  
  'regip' VARCHAR( 30 ) NULL ,  
  'regdate' VARCHAR( 30 ) NULL ,  
  'role' TINYINT NULL ,  
  'contributions' SMALLINT NOT NULL DEFAULT '0',  
PRIMARY KEY ( 'username' )  
) ENGINE = MYISAM ;  
  
CREATE TABLE 'Discussion' (  
  extension VARCHAR(40) NOT NULL,
```

```

    id INT NOT NULL AUTO_INCREMENT,
    author VARCHAR(20) NOT NULL,
    title VARCHAR(50) NULL,
    post VARCHAR(2000) NULL,
    ipadr VARCHAR(20) NULL,
    lastpost VARCHAR( 32 ) NULL,
    PRIMARY KEY (extension,id)
) ENGINE = MyISAM;

```

Nestandardně jsem vyřešil komentáře, které mohou být vázány na příklady, testy, nebo vlákna diskuse. Enumerace určuje, na který typ je vázán. Existenci objektu, ke kterému se komentář váže, musí v tomto případě kontrolovat aplikace.

```

CREATE TABLE 'Comments' (
    extension VARCHAR(40) NOT NULL,
    id INT NOT NULL,
    typ ENUM('example','test','discussion') NOT NULL,
    commentid INT NOT NULL AUTO_INCREMENT,
    author VARCHAR(20) NOT NULL,
    post VARCHAR(200) NULL,
    ipadr VARCHAR(20) NULL,
    PRIMARY KEY (extension, id, typ, commentid)
) ENGINE = MyISAM;

```

4.3 Sada funkcí pro manipulaci s databází

Soubory s funkcemi pro manipulaci s databází se nachází v adresáři „database“. Soubor „dbinfo.php“ obsahuje proměnné s informacemi potřebnými pro připojení k databázi. Tento soubor je „includován“ (zahrnut, funkce require) souborem „dbconn.php“, který obsahuje dvě funkce - „dbconn()“ pro vytvoření nepersistentního připojení a „dbconnpers()“ pro připojení persistentní. Tyto funkce po zavolání zkontrolují, jestli už spojení je vytvořeno, a pokud není, vytvoří nové.

Ostatní metody tvoří funkce s předpřipravenými SQL dotazy. Funkce jsou rozděleny do pěti souborů podle oblasti, které se týkají: „examplesDBC.php“, „discussionDBC.php“, „userDBC.php“, „extensionDBC.php“ a „testingDBC.php“. Funkce vždy zkontrolují, zda existuje spojení (funkcemi z „dbconn.php“) a poté provedou příkaz. Buďto se jedná o příkaz, který z databáze čte - funkce v tom případě vrací konkrétní hodnotu nebo pole hodnot, nebo jde o příkaz, který do databáze vkládá – v tom případě vrací hodnotou boolean informaci o úspěšnosti vkládání.

Většina těchto funkcí pouze provádí příkazy a neobsahuje žádnou další logiku, výjimku tvoří pouze metody ukládání a kontroly hesel uživatelů, kdy tato metoda heslo, které dostane, nejprve zašifruje. Metoda heslu přidá „sůl“ a takto vzniklý řetězec zahašuje algoritmem MD5. Pro připojení k databázi jsem využíval PDO [8]:

```
// Funkce, která vytvoří nepersistentní spojení
function dbconn(){

    global $pdo;

    if($pdo==false){
    try{

        require '../database/dbinfo.php';
        $pdo = new PDO('mysql:host=' . $dbhost . ';dbname=' . $dbname ,
        $dbuser , $dbpasswd ,
        array(
        PDO::ATTR_PERSISTENT => false
        ));
        $pdo->beginTransaction();

    } catch (Exception $e) {
        $pdo->rollBack();
        echo "Failed: " . $e->getMessage();
    }
    }
}
```

Příklad předpřipraveného dotazu:

```
//Funkce kontrolující dostupnost uživatelského jména
function uncheck($username){

    global $pdo;
    dbconn();

    $username = htmlspecialchars(strtolower(trim($username)));

    try{

        /* Dotaz, zda jméno již není použito. */
        $sql = "SELECT username FROM Users WHERE username=:jmeno";
        $stmt = $pdo->prepare($sql);
        $stmt->bindParam(":jmeno",$username);
        $stmt->execute();
        $sql_result = $stmt->fetchAll();

        /* Pokud už uživatel existuje, vrátí funkce TRUE */
        if($sql_result[0]["username"]== $username)return true;
```

```
} catch (Exception $e) {
    echo "Failed: " . $e->getMessage();
    $pdo->rollBack();
    echo "Failed: " . $e->getMessage();
}
return false;
}
```

4.4 Stahování a parsování seznamu extenzí

Seznam extenzí se nachází na hlavní stránce OpenGL Extension Registry [1]. Jak už bylo zmíněno v analýze, tato stránka není validní oproti standardu XHTML, který má definovaný v hlavičce, a není ho tedy možné zpracovávat jako XML. Zpracovávám ho tedy jako obyčejný text. Stáhnutí a zpracování obstarává funkce „downloadList()“ ze souboru „downloadList.php“ adresáře „specification“.

Nejprve načtu obsah hlavní stránky OpenGL Registry pomocí PHP metody `file_get_contents()`, tento obsah pak na vlastní server uložím do textového souboru. Zde je nutné, aby obě tyto akce byly povoleny v konfiguračním souboru PHP „php.ini“.

Uložený textový soubor je pak čten po jednotlivých řádcích. Jakmile se metoda dostane na řádek obsahující '`<h6> `', začne zpracovávat všechny následující položky HTML seznamu. Každý řádek, který obsahuje jméno extenze a odkaz na její specifikaci, zpracuje metoda dvakrát. Nejprve upraví odkaz tak, aby odkazoval na budoucí umístění specifikace v mé aplikaci (relativní cesta), a tento odkaz přidá do seznamu, který se poté bude zobrazovat na hlavní stránce. Poté je odkaz upraven tak, aby odkazoval na stránku mé aplikace, která daný odkaz stáhne a zpracuje. Seznam těchto odkazů pak metoda vrátí.

Tato metoda závisí na stabilním formátování zdrojové stránky. Pokud se v blízké době struktura stránek OpenGL změní, tento script přestane fungovat a bude potřeba ho napsat znovu.

4.5 Stahování a parsování specifikací

O stažení a zpracování konkrétní specifikace se stará funkce `downloadParseSave()` ze stejnojmenného souboru v adresáři „specification“. Tato funkce má dva vstupní atributy: jméno textového souboru specifikace a cesta k němu. Kvůli efektivitě tato metoda danou specifikaci parsuje přímo do formátu HTML, v případě budoucích změn ale lze výstup jednoduše změnit na jiný formát.

Podobně jako při stahování seznamu extenzí je i zde načten obsah stránky pomocí PHP metody „`file_get_contents()`“ a tento obsah je pak na serveru uložen v dočasném textovém souboru. Tento soubor je čten po řádcích a řádky jsou spojovány do odstavců. Tyto odstavce jsou rozpoznávány podle prázdných řádků mezi nimi. Každý odstavec je pak zpracován samostatně.

The screenshot shows the OpenGL Extensions Database interface. At the top, there is a search bar with the text "Google Custom Search" and a "Search" button. The main content area is titled "ARB_wgl_extensions_string" and includes a link to the "Original specification file". Below this, it lists "Name strings: WGL_ARB_extensions_string" and provides an "Overview" stating that this extension allows applications to determine which WGL extensions are supported by a device. There are tabs for "Specification", "Examples", "Testing", and "Discussion". A sidebar on the right contains a "List of all extensions" link, a login form with fields for "Username" (containing "asdf") and "Password" (containing "****"), and buttons for "Login" and "Register". The main content area has a table with the following sections: "Name", "Name Strings", "Contact" (listing Pat Brown, NVIDIA and Paula Womack), "Status", "Version", "Number", "Dependencies", and "Overview".

Obrázek 4.1: Ukázka výsledku parsování specifikace

Nejprve se určí, zda je odstavec obyčejný, nebo zda se jedná o nadpis. Nadpis je standardně určen tím, že není zleva odsazený, a jeho první znak je malé nebo velké písmeno. Zároveň nadpis nesmí obsahovat opakující se znaky „- - -“, neboť v tomto případě se s nejvyšší pravděpodobností jedná o neodsazenou, ASCII formátovanou tabulku.

Pokud je odstavec obyčejný, pouze použijeme funkci „htmlspecialchars()“, protože specifikační soubor často obsahuje speciální znaky HTML, a označíme klíčová slova odkazy. Tento text bude později v HTML obalen tagy `<pre>`, aby si zachoval své ASCII formátování. Důvod je ten, že specifikační text používá ASCII formátování ve velké míře pro nejružnější odsazení a tabulky. Všechny tyto případy pak není možné zjistit a nelze proto připravit všechny výjimky při případném převádění ASCII do HTML.

V případě odstavce postupujeme jinak. Nejprve přidáme uzavírací tagy `</pre>` a `</div>` pro uzavření předchozí kapitoly, poté vložíme text nadpisu, obalený tagy `<h2>`. Této kapitole přiřadíme jednoznačné ID, spočítané jako hash MD5 řetězce nadpisu. Připravíme také onclick JavaScript event pro nadpis, který bude odkazovat na JavaScriptovou funkci, která jako vstupní parametr dostane toto ID. Po odstavci zapíšeme otevírací tagy `<div>` a `<pre>`. Tagu `<div>` přitom přiřadíme ID kapitoly, aby s ním posléze mohl JavaScript snáze pracovat.

Pro specifické kapitoly se chování parseru odlišuje. Jedná se o kapitoly, jejichž název začíná na „Additions to“ a „Dependencies on“. Těchto kapitol je ve většině specifikačních souborů mnoho, a proto je parser změněn na podkapitoly a přiřadí je pod kapitolu „Additions“ resp. „Dependencies“.

Po provedení transformace metoda zapíše výsledný HTML kód do textového souboru. Z

názvu zdrojového souboru odvodí jméno extenze a zavolá databázovou funkci, která zkontroluje, zda je extenze už v databázi a případně ji vytvoří. Zároveň jsou do databáze zvlášť zapsány pole NameStrings a Overview ze specifikačního souboru a čas poslední aktualizace.

4.6 Manuální update

Pro otestování předchozích parsovacích metod vznikla menší sekce „Manuální update“. Ač byla vytvořena pouze pro testování, může se hodit i například administrátorovi a nechal jsem ji proto přístupnou. Tato sekce při spuštění zavolá první zmíněnou parsovací funkci, čímž zaktualizuje seznam extenzí, a tento seznam pak vypíše. Funkcemi, které jsou napsané v jazyce JavaScript, je možné zaktualizovat konkrétní specifikační soubory, nebo zapnout zaktualizování všech těchto extenzí. V případě aktualizace všech specifikací funkce prochází kompletní seznam a postupně volá aktualizací metodu. Interval mezi aktualizacemi jednotlivých specifikačních souborů je nastaven na 5 sekund. Aktualizační funkce využívá AJAX.

4.7 Registrace uživatele

Přidávat jakýkoliv obsah na stránky může jen registrovaný a přihlášený uživatel. Registrace není nijak omezená. V registračním formuláři musí uživatel vyplnit své uživatelské jméno, skutečné jméno, heslo, email a kontrolní součet nebo Captcha kód pro omezení registrace tzv. botů. Po odeslání formuláře script zkontroluje, zda je vybrané uživatelské jméno volné. Pokud je volné, zadané údaje jsou spolu s jeho IP adresou uloženy do databáze a uživatel je registrován. Ošetření speciálních znaků a zahašování hesla provede funkce pro komunikaci s databází ještě před vložením.

Pro omezení registrace robotů, kteří vkládají „spam“, jsem se rozhodl použít službu reCAPTCHA od firmy Google, která by měla být dostatečně spolehlivá. ReCAPTCHA nabízí PHP knihovnu a jednoduchý návod, jak ji používat. Pro použití je ale třeba mít účet na Googlu, teprve poté si můžete nechat vygenerovat dvojici klíčů (privátní a veřejný), se kterou můžete službu využívat.

Protože reCAPTCHA při některém nastavení serveru nefunguje, což byl i případ serveru freehostingové firmy IC.cz, na kterém jsem svoji aplikaci testoval, rozhodl jsem se přidat jednoduchou vlastní alternativu v podobě nutnosti spočítání jednoduché rovnice.

4.8 Sekce diskusí

Do diskuse mohou přispívat všichni registrovaní uživatelé. Každá diskuse je vázána k jedné konkrétní extenzi. Každá diskuse se dělí na vlákna. Vlákna jsou tvořena kořenovým příspěvkem a reakcemi na něj. Na jedné straně diskuse se zobrazuje více vláken, protože se nemá jednat o mohutné samostatné fórum ke každé extenzi. Pouze kořenový příspěvek je odlišný od ostatních komentářů, všechny zbylé jsou stejné. Není zde reagování na konkrétní komentáře.

Pro založení nového vlákna existuje samostatný formulář „addnewForm.php“ v adresáři „discussion“. Je možné vkládat pouze text, bez odkazů a obrázků. Při zpracování formuláře je

vstup ošetřen funkcí „htmlspecialchars()“. Stejná pravidla platí i pro komentáře kořenového příspěvku. Minimální délka textu komentáře jsou 4 znaky, kratší příspěvky přijaty nejsou.

Pod kořenovým příspěvkem se vždy zobrazují maximálně tři příspěvky. Pokud je jich více, zůstávají ty starší skryté a zobrazí se až po kliknutí na odkaz pro zobrazení. Skryté jsou pouze pomocí kaskádových stylů a jejich zobrazení obstarává JavaScript, v tomto případě se ale nejedná o AJAX. Pod komentáři se také zobrazuje formulář pro přidání komentáře, pokud je uživatel přihlášený.

4.9 Sekce testů extenzí

Každý test extenze obsahuje název (titul), jméno autora, obsah (popis), související soubory (spustitelný test ke stažení, obrázky), uživatelské výsledky testů a komentáře. Test může nahrát jakýkoliv registrovaný uživatel, ale na výpisu testů u extenze se objeví až po tom, co je schválen dvěma uživateli.

Prvni test (author: asdf) | Show/hide other test results | Show/hide comments | Edit test |

asfdasfdasf
asfdasfdasd
fasdfasdfads
fasdfasdfasdf

Attached files:
../data/voucher-dd25a5.pdf [84828 B , application/pdf]

Testresult number: 1 | Show all results | Next Testresult |

Test suite: GL_ARB_draw_elements_base_vertex OpenGL version: 3.3.0 OpenGL renderer: GeForce 9500M GS/PCI/SSE2		CPU			GPU		
		min	max	average	min	max	average
2 little models (1 index), each drawn 100 times.	Time spent with two index arrays (switching)	3.83248e-006	2.78026e-005	8.55465e-007	1496	1664	1184
	Time spent with one index array and manual offset	1.70238e-006	2.56639e-005	8.55465e-007	1232	1536	1152

Comments
Show all comments (12)

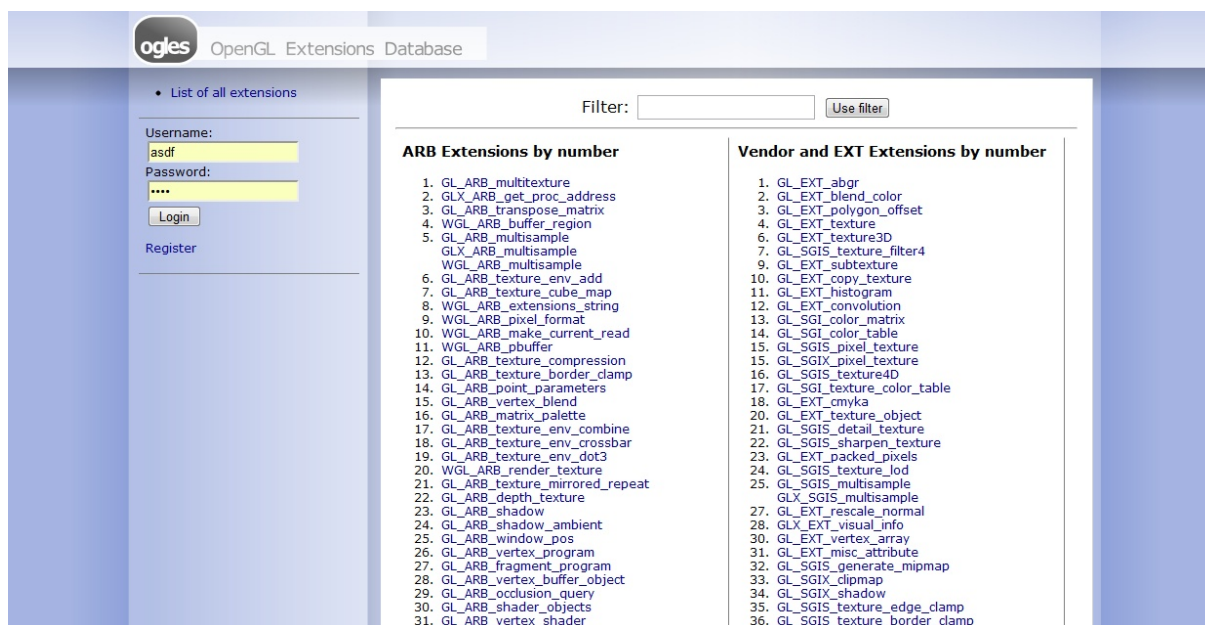
From: **asdf**
ihihhi

Obrázek 4.2: Ukázka zobrazení jednoho testu extenze

Soubory, které smí uživatel nahrávat, jsou v databázi spojeny vždy s jedním testem. Proto již před prvním uložením nového testu uživatelem musí být tento test v databázi zaznamenán, avšak označen jako prázdný. Po kliknutí na odkaz pro vytvoření nového testu je tedy test nejdříve vytvořen a poté je uživatel přesměrován rovnou na editaci tohoto testu. S tím ovšem vznikl problém, jak se vypořádat s testy, které uživatelé nedokončí a zůstanou prázdné. Při vytvoření testu tento prázdný záznam dostane zápis do sloupce „locked“ (původně určený pro jiný účel) zápis „6“, což je náhodně zvolené číslo, které indikuje, že je test prázdný, neupravený a seznamy testů ho nemají zobrazovat. Teprve po první editaci a uložení se toto číslo změní. Zároveň při vytváření nových testů se nejprve zkontroluje, zda náhodou neexistují tyto prázdné testy starší dvou dnů, a záznamy takových testů jsou vymazány.

Při vytváření doprovodných textů vlastních testů by uživatel mohl potřebovat více, než jen samotný text. Proto jsem se rozhodl využít JavaScriptový WYSIWYG Editor TinyMCE [7], který uživatelské vstupy (formátování textu, přidávání tabulek a obrázků) převádí do HTML. Tyto záznamy v databázi pak ale nesmí být ošetřovány funkcemi „htmlspecialchars()“ a podobnými. Nechat uživateli úplnou volnost ale nemůžeme, proto je nutné ošetřit alespoň různé typy vkládání škodlivého kódu. Musíme tedy stanovit seznam zakázaných slov a znaků, jejichž přítomnost budeme při pokusu o uložení testu na serverové straně kontrolovat a v případě nálezu uživateli nedovolíme takový test uložit. Klíčová slova se týkají hlavně HTML tagů vložených rámu <iframe>, <embed>, <object> a také všech možností vkládání scriptovacích jazyků, pro JavaScript například klíčová slova „onload“, „onclick“, „onMouseOver“, „javascript:“ a tagy <script>.

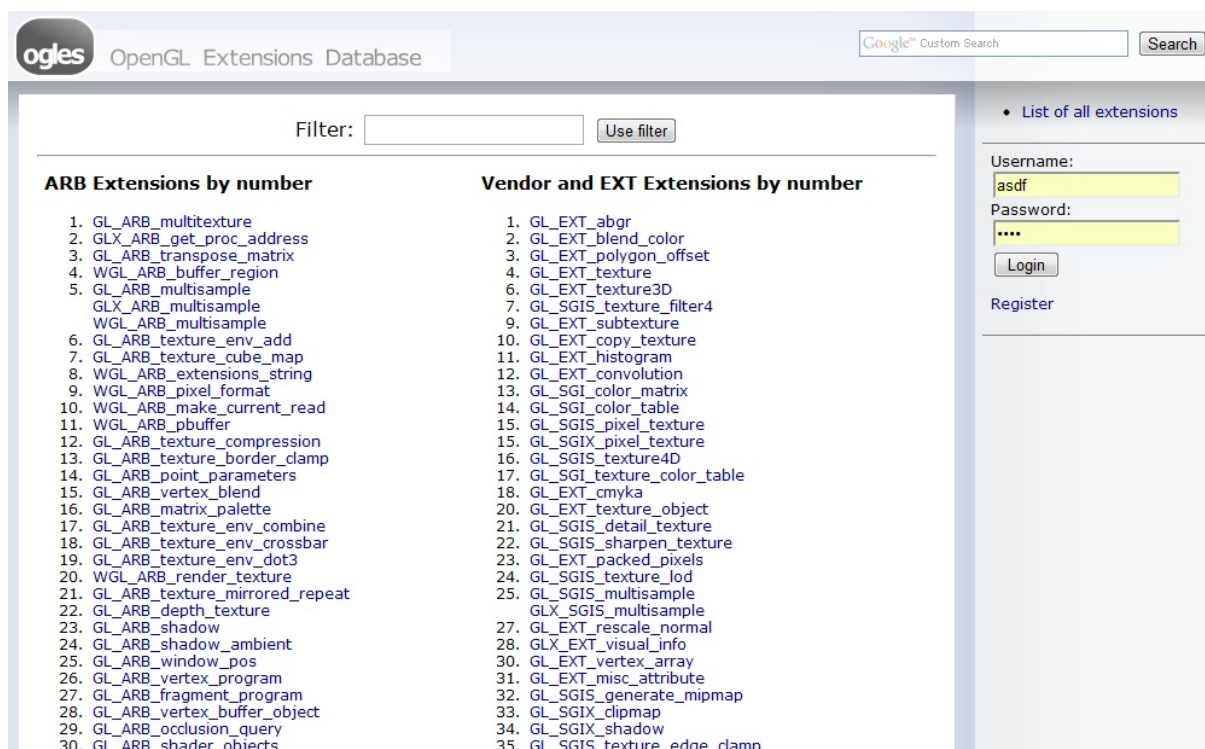
Formát výsledků testů byl předpokládán jako dvě textová pole, kde do prvního uživatel zadá konfiguraci své počítačové sestavy, do druhého potom samotné výsledky. Část komentářů je pak totožná s komentáři kořenových příspěvků v sekci diskuse.



Obrázek 4.3: První verze vzhledu stránek

4.10 Uživatelské rozhraní, design

Rozložení ovládacích prvků přesně kopíruje oba návrhy. První rozhraní mělo pevně určenou šířku 1000 pixelů a sloupec s uživatelskou sekcí a seznamem novinek byl po levé straně. Po problémech zjištěných po prvním testování (viz níže), souvisejících se zjištěním nepraktičnosti při prohlížení zmenšeného okna, byl použit druhý návrh s relativní šířkou a užším sloupcem, přesunutým na pravou stranu.



Obrázek 4.4: Druhá verze vzhledu stránek s relativní šířkou

4.11 Testování nehotové aplikace

Seznámení studentů, kteří vytvářeli testy extenzí, s první částečně funkční verzí aplikace proběhlo formou krátkého testování 23.3. 2011. Oba účastníci testu do té doby s aplikací nesetkali. Testování proběhlo na jejich vlastních přenosných počítačích, dočasná adresa projektu byla <http://score.phorum.cz/ogl/>. V době testování nebyla ještě funkční sekce ukázek kódu, vyhledávání extenzí a kvůli chybě v kódu nebylo funkční nahrávání souborů.

Účastníci dostávali postupně ústně jednotlivé úkoly a otázky. O dosud neimplementovaných součástech aplikace jsme vedli diskusi. Problematické části testu jsem zaznamenával ručně na poznámkový papír. Připomínek a návrhů na vylepšení, vycházejících z testování, bylo více, velký problém se objevil pouze jeden, který se týkal výsledků uživatelských testů. Všechny zjištěné poznámky a nedostatky jsou v následujícím přehledu, rozděleném podle kategorií.

4.11.1 Výsledky testování

- Příliš mnoho potvrzování

Účastníkům se zdálo, že u všech formulářů je by nemělo být třeba potvrzovat téměř každou akci, je třeba to nechat potvrzování jen tam, kde to dává smysl.

- Vyhledávání

Při testování byl na hlavní straně pouze výpis extenzí totožný se seznamem na stránkách OpenGL. K tomuto seznamu bylo v plánu přidat jednoduchý filtr (JavaScript). Jako hlavní vyhledávání pak použít "Vlastní vyhledávání" od společnosti Google [3]. Účastníci testu tuto kombinaci hodnotili jako dostatečnou.

- Práva uživatelů

Mnou navrhovaný systém rozdělení uživatelů do kategorií byl: neregistrovaný, registrovaný, registrovaný důvěryhodný, admin. Přičemž i neregistrovaný mohl přispívat do diskusí. Tento systém práv se účastníkům nejevil jako dostatečný.

- Rychlost rolování

Rychlost rozbalování odstavců specifikace se zdá být pomalá. V době testování sekce specifikací používala animaci rozbalování pomocí JavaScriptové knihovny Script.aculo.us. Jako řešení bylo navrženo rozbalování pouze instantně (bez animace) případně přepínač pro vypnutí a zapnutí provádění animace.

- Prázdné odstavce

Prázdné kategorie specifikací by podle druhého účastníka testu měly být označené a pokud možno nerozbalovací. Zde osobně příliš nesouhlasím s tím, že by měly být nerozbalovací. Bylo by to podle mne matoucí. Označené by ale jistě být mohly - například zašedlé.

- Návrh: Parser kodu

Kategorie Examples zatím nebyla v provozu, ale dle mého plánu měla být totožná se sekci Testování (kromě uživatelských výsledků testů). Protože se zde ale bude vyskytovat ukázkový kód, hodilo by se zde podle uživatelů testu zvýraznění syntaxe - zazněl tedy návrh zjistit, zda existuje nějaký jednoduchý parser s volnou licencí.

- Výsledky testů extenzí

Protože jsem při tvorbě sekce uživatelských výsledků testů neměl jasnou představu o tom, jak budou výstupy testů vypadat, navrhl jsem podle účastníků testu sekci špatně. Podobu bude ještě třeba prodiskutovat vzlašť po skončení testování. Jeden z návrhů byl, aby sám tvůrce testu vybíral typ a počet položek, které bude muset uživatel, který test vyzkouší, vyplnit.

- Co když se test týká více extenzí najednou?

V době testování byl jeden test v systému vázán na jednu konkrétní extenzi. Mělo by být vymyšleno, jak řešit situaci, kdy se jeden test týká více extenzí.

- Položka „title“ při tvorbě testu není příliš viditelná.

Při vytváření testů je třeba vyplnit nadpis (title) testu a jeho text. Protože pole pro vyplňování textu je velmi výrazné kvůli použití editoru TinyMCE, uživatel si nevšimne a nevyplní název testu. Jako řešení bylo navrženo zobrazení pole pro vyplnění textu testu až po zadání názvu (title).

- Nahrávání souborů - obrázků

Kvůli jednoduššímu přidávání obrázků do textu padl návrh zjistit, zda lze provést drobné změny v používaném editoru TinyMCE.

- Datum a čas zprávy

Diskuse se účastníkům testu zdála funkční a dostatečná, postrádali pouze zobrazení data a času přidání příspěvku.

4.11.2 Závěr testování

Opravu nalezených chyb a jednodušší navrhovaná vylepšení provedu, o následujících tématech bude ještě třeba provést diskusi:

- Práva a kategorie uživatelů
- Formát uživatelských výsledků testů
- Test pro více extenzí

Poznámka: Diskuse byla provedena na schůzce se zadavatelem projektu. Výsledky diskuse jsou zahrnuty v následujících částech popisu realizace.

4.12 Vkládání a zobrazování ukázek kódu

Realizace sekce ukázek kódu použitím extenze je velmi podobná sekci testů extenzí. Oproti testům zde však chybí nahrávání vlastních souborů a samozřejmě sekce výsledků testů. Ukázky kódu také nepoužívají editor TinyMCE a tudíž nemají povolené vkládání HTML kódu. Ukázky kódu může vkládat každý registrovaný uživatel.

Po návrhu z prvního testování jsem našel zvýrazňovač kódu, napsaný v jazyce JavaScript, SyntaxHighlighter (verze 3.0.83) [9], který jsem do svého projektu zahrnul. Pokud chce uživatel v textu zvýraznit kód, obalí ho do tagů `[code]` a `[/code]`.

Protože je nutné u sekce ukázek kódů uvádět nadpis („title“) stejně jako u sekce testů, kde si této nutnosti uživatelé testů nevšimli, vyřešil jsem tento problém pomocí CSS, kdy jsem použil žlutou barvu pro zvýraznění všech důležitých částí formuláře.

The screenshot shows a web-based code editor interface. At the top, there are tabs for 'Specification', 'Examples', 'Testing', and 'Discussion'. Below the tabs, the text 'Code examples for ARB_transpose_matrix' is displayed. A dark blue header bar contains the text 'Example from view-source:http://read.pudn.com/down (author: Sirall)'. To the right of this bar are links for 'Show/hide comments' and 'Edit example'. The main area contains a code editor with a light blue background and a vertical scrollbar on the right. The code is C++ and uses syntax highlighting: keywords are blue, comments are green, and strings are red. The code is as follows:

```

1 void CMySDOpenGLView::DrawFKinematics(void)
2 {
3     glPushMatrix();
4
5     glTranslatef(vectorp.x,vectorp.y,vectorp.z);
6
7
8
9
10
11
12
13 if((fabs(vectorx.x)!=0.0)||fabs(vectorx.y)!=0.0)||fabs(vectorx.z)!=0.0)
14 { //     glPushMatrix();
15     glRotatef(A1,vectorx.x,vectorx.y,vectorx.z);
16 //     glPopMatrix();
17 }
18
19 if((fabs(vectory.x)!=0.0)||fabs(vectory.y)!=0.0)||fabs(vectory.z)!=0.0)
20 { //     glPushMatrix();glMultMatrix
21 //GetFloatv(TRANSCOPE_MODELVIEW_MATRIX_ARB,m);
22     glRotatef(A2,vectory.x,vectory.y,vectory.z);
23 //     glPopMatrix();
24 }
25
26
27
28
29
30
31
32
33
34
35

```

Obrázek 4.5: Ukázka využití Syntax Highlighter

4.13 Zobrazení výsledků testů extenzí

Jak bylo zjištěno při testování, dosavadní koncept, počítající se dvěma textovými řetězci pro každý výsledek, byl nedostatečný. Další diskusí bylo rozhodnuto, že pro pohodlí uživatelů testů, budou výsledky testovacích programů ve formátu XML. Uživatel tento soubor jednoduše nahraje k danému testu a nebude muset vyplňovat žádné další informace.

Po dohodě přibližné struktury studenti upravili své programy a poskytli mi ukázkový výsledkový XML soubor. Pro validaci jsem podle tohoto souboru vytvořil validační XML Schema. Ke kontrole validnosti XML souboru podle Schematu v PHP využívám PHP DOM [6]. I to je důvod, proč jsem pro validaci volil XML Schema místo DTD. DTD soubor by totiž musel být definován v souboru s výsledky.

XML soubory s výsledky jsou po nahrání na server a po validaci uloženy do databáze (typ BLOB) v originální podobě. Transformace do tabulek HTML se provádí až při zobrazování. Tato transformace mohla být prováděna pomocí XSLT, ale kvůli tomu, že by v budoucnu tyto XML soubory mohly podle návrhů tvůrců testů obsahovat i obrázky, jsem zvolil převod do HTML pomocí PHP DOM [6].

Ukázka kódu:

```

function parseXMLresult($testresult){
    $dom = new DOMDocument;
    if($dom->loadXML($testresult)) {

        $sections = $dom->getElementsByTagName('test-section');
        $toutcome = $toutcome . "<table><tbody>";
        $toutcome = $toutcome . "<tr><td colspan='2' rowspan='2' class='head'>";

        ( . . . )

        foreach ($sections as $section) {

            ( . . . )

                $timerdatas = $section->getElementsByTagName('timer-data');
                foreach ($timerdatas as $timerdata){
                    ( . . . )

                        $toutcome = $toutcome . "<td>"
                        . $timerdata->getElementsByTagName('cpu-timer-data')->
                        item(0)->getElementsByTagName('average')->item(0)->
                        nodeValue . $cpuunits . "</td><td>" . $timerdata->
                        getElementsByTagName('cpu-timer-data')->
                        item(0)->getElementsByTagName('highest')->item(0)->
                        nodeValue . $cpuunits . "</td><td>" . $timerdata->
                        getElementsByTagName('cpu-timer-data')->item(0)->
                        getElementsByTagName('lowest')->item(0)->nodeValue
                        . $cpuunits . "</td>";

                    ( . . . )

                }

            }

            $toutcome = $toutcome . "</tbody></table>";

            return $toutcome;
        }else {
            return "Error loading xml testresults file.";
        }
    }
}

```

4.14 Nahrávání souborů

Nahrávání vlastních souborů je možné v sekci testů extenzí. Záznam o každém souboru je v databázi vázán na konkrétní test, určený jménem extenze a identifikačním číslem. O souboru je však do databáze ukládán pouze záznam, samotný soubor se neukládá do databáze, ale na samotný server. Proto je nutné povolit nahrávání jen některých bezpečných typů souborů. Jako povolené MIME typy jsem zvolil různé typy obrázků, soubory PDF a archivy ZIP.

4.15 Vyhledávání

Když vyhledávání v mé aplikaci ještě neexistovalo, vyhledávali účastníci testu potřebnou extenzi pomocí internetových prohlížečů a jejich funkce Hledat, kdy prohlížeč na stránce barevně zvýrazní všechna místa výskytu zadaného řetězce. To mě přivedlo na myšlenku podobnou – jednoduchý filtr by po zadání řetězce zobrazoval pouze extenze, ve kterých se tento řetězec vyskytuje.

Tento filtr jsem implementoval pomocí jazyka JavaScript. Script se volá při každém stisknutí klávesy a funkce nejprve načte seznam všech extenzí, poté přečte zadaný řetězec a porovná ho s jednotlivými položkami seznamu. Pokud se řetězec v položce seznamu vyskytuje, funkce ji pomocí CSS zviditelní, pokud ne, tak ji skryje.

Následující funkce je volána při každém stisknutí klávesy. Vstupním parametrem je aktuální hodnota pole filtru. Funkce nevyužívá AJAX, manipuluje pouze se styly. Díky její jednoduchosti a efektivnosti funguje rychle a spolehlivě například i v internetových prohlížečích mobilních přístrojů.

```
function efilter(word){
    extlist = document.getElementById('panel').getElementsByTagName('li');

    word = word.toLowerCase();
    word = word.replace(' ', '_');

    var len = extlist.length;

    for(var i=0; i < len ; i++){

        if(word==''){
            extlist[i].style.display = "list-item";
        }else{

            wordpos = extlist[i].innerHTML.toLowerCase().indexOf(word);

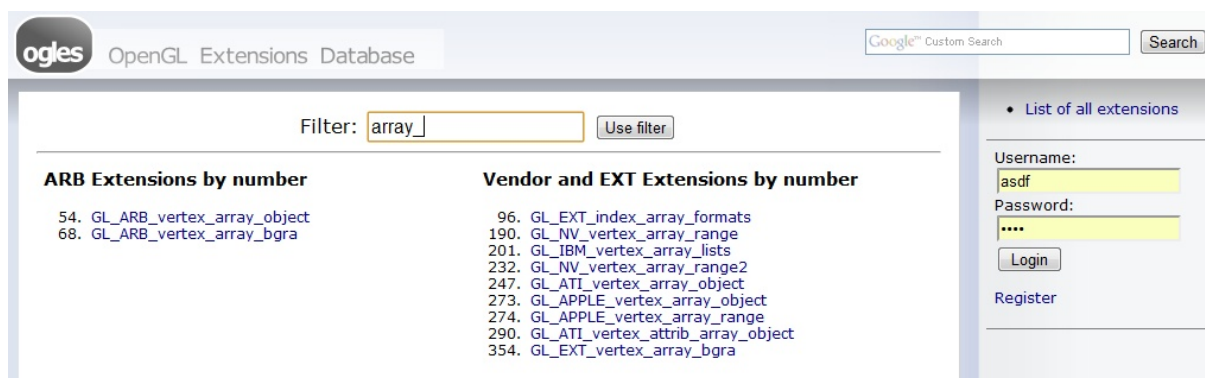
            if( wordpos == -1){
                extlist[i].style.display = "none";
            }else{
                extlist[i].style.display = "list-item";
            }
        }
    }
}
```

```

    }
  }
}

```

Pro složitější vyhledávání jsem zvolil použití Google Custom Search Engine [3] .



Obrázek 4.6: Ukázka fungování filtru extenzí

4.16 Aktualizace

Aktualizování seznamu a specifikací extenzí jsem se nakonec rozhodl implementovat pomocí PHP, řešením Visitor Based Webcron. To spočívá v tom, že je provedení naplánovaných operací spouštěno návštěvou stránky, standardně je script volán skrytým elementem "img" v kódu stránky.

V tomto konkrétním případě je při dotazu na zobrazení stránky nejprve zkontrolováno, kdy byla specifikace naposledy aktualizována. Pokud nebyla specifikace (či seznam) aktualizována déle jak tři dny, je do stránky vložen právě zmíněný skrytý element "img", který volá aktualizovací script. Taková aktualizace se tedy provede až po zobrazení stránky uživateli. Pokud nebyla stránka aktualizovaná déle než dva týdny, je větší šance, že je obsah zastaralý a aktualizace se provede ještě před zobrazením stránky uživateli. To může mít sice za následek lehce delší trvání načítání stránky, při větší návštěvnosti stránek by se ale tento případ stával jen výjimečně.

Výhodou tohoto způsobu aktualizování je kvůli implementaci v PHP opět přenositelnost.

Kapitola 5

Testování

5.1 Příprava pro test s uživateli

5.1.1 Cíl testu

Cílem tohoto testu s uživateli bude zjistit použitelnost vytvořené aplikace. Aplikace má sloužit vývojářům a studentům, kteří hledají informace o extenzích OpenGL. Budeme sledovat, jakou mírou se v praxi projeví chyby již objevené, a pokusíme se zjistit i chyby další. To zjistíme pomocí sledování účastníků testu a doplňujícími dotazy.

5.1.2 Screener

Tzv. Screener je dotazník sloužící k vybrání účastníků testu tak, aby co nejpřesněji splňovali námi zadaná kritéria. Cílem bude vybrat čtyři účastníky. Ti budou vybráni z řad studentů Fakulty Elektrotechnické ČVUT. Tyto čtyři účastníky budeme vybírat tak, aby tuto skupinu tvořili dva zkušení uživatelé extenzí OpenGL a dva s minimálními znalostmi OpenGL extenzí. Screener má dvě části – veřejnou a neveřejnou. Veřejnou část tvoří samotný dotazník a neveřejnou preferované odpovědi. Screener je anonymní a každý bude mít tedy přiřazeno vlastní identifikační číslo.

Veřejná část Screeneru – viz příloha D

Neveřejná část Screeneru – viz příloha E

5.1.3 Pre-test dotazník

Pre-test dotazník bude veden formou strukturovaného rozhovoru. Tento způsob je osobnější - lépe lze tedy poznat účastníka a případně ještě odfiltrovat účastníky nehodící se pro testování. Dotazník bude nahráván na diktafon pro případné pozdější zpracování.

Pre-test dotazník – viz příloha F

5.1.4 Nastavení testu a popis rolí

Testování proběhne v přirozeném prostředí účastníka – tedy např. v kanceláři nebo doma, podmínkou je pouze to, aby bylo prostředí tiché a klidné. Aplikace je stejně jako při prvním testování umístěna na dočasné adrese www.score.phorum.cz/ogl. Bude ji tedy možno testovat na osobních počítačích účastníků testů. V případě nutnosti lze použít můj notebook s Windows Vista. Notebook je vybaven TouchPadem a v průběhu testování k němu bude připojena i usb myš. Webové rozhraní bude spuštěné v prohlížeči Firefox v modu Fullscreen při rozlišení 1280x800.

Jako startovní stránka bude účastníkovi testu nastavena hlavní strana s výpisem extenzí. Na svém notebooku budu mít nastavené nahrávání zvuku pro pozdější zpracování. Úlohu zadavatele, moderátora i zapisovatele budu plnit já sám. Vzhledem k rozsahu testu by se nemělo jednat o problém a všechny úlohy bych měl zvládnout bez větších potíží.

5.1.5 Výchozí stav aplikace

Uživatel si zaregistruje vlastní účet, pro další testování ale využije účet jiný, předem připravený. Tento předem připravený profil má nastavena administrátorská práva („admin“, heslo: „admin“). Aplikace bude předem naplněna náhodnými daty („Lorem Ipsum“). Několik extenzí bude mít příspěvky v sekcích Diskuse, Ukázky kódu i Testování.

5.1.6 Seznam úloh

Seznam úloh obsahuje krátké seznámení s aplikací a samotný seznam úkolů podaných co nejobecněji tak, aby samotné zadání nedávalo účastníkovi testu žádnou nápovědu jak postupovat.

Seznam úloh – viz příloha G

5.1.7 Post-test dotazník

V post-test dotazníků se uživatele ptáme na průběh testu, konkrétní obtížnost úkolů a názor na celou aplikaci. Necháme i prostor pro jeho vlastní připomínky. Post-test dotazník bude stejně jako pre-test dotazník veden ústně.

Post-test dotazník – viz příloha H

5.2 Provedení testu s uživateli

5.2.1 Reálné podmínky testu

Přesně podle plánu byla aplikace testována na čtyřech uživateli. Dva lidé s velkými zkušenostmi s OpenGL a dva studenti se znalostmi minimálními. Testování probíhalo z části v domácím prostředí účastníků v jimi zvolené místnosti a z části v prostorách Fakulty Elektrotechnické v Dejvicích. Testování nebylo vyrušováno žádnými rušivými elementy. Jedinou

odchylku od předpokládaného průběhu testování tvoří změna způsobu nahrávání - místo pouhého nahrávání zvuku byl pořizován i obrazový záznam mobilním telefonem Sony Ericsson Xperia X2 (rozlišení videa 800x480). Každé testování trvalo i s dotazníky vždy okolo šedesáti minut. Z toho práce s aplikací tvořila průměrně čtyřicet pět minut. První účastník stránku testoval na svém vlastním počítači s operačním systémem Windows 7, rozlišením obrazu 1920x1080 a prohlížečem Google Chrome. Druhý stránku testoval na tabletu Iconia Tab s Windows 7 a prohlížečem Internet Explorer 9. Třetí účastník použil také vlastní notebook s Windows XP a prohlížečem Internet Explorer 8. Poslední testoval stránky na mém počítači, jehož specifikace jsou uvedeny výše.

5.2.2 Vyhodnocení testu s uživateli

Chyby a připomínky všech účastníků testu jsem sjednotil do následujícího seznamu. Ke každému bodu jsem zároveň připojil svůj komentář a možné řešení.

- Nejasnosti se objevily ohledně pojmenování sekce „Testing“. Jeden z účastníků nevěděl, co se pod tímto názvem má představit. Po vysvětlení smyslu sekce navrhuje alternativní název „Performace“, což by podle něj lépe vystihovalo obsah.

Řešení: S návrhem souhlasím a sekci přejmenuji.

- Několika účastníkům testu by více vyhovovalo, kdyby byly všechny kapitoly při otevření sekce Specifikace již rozbalené. Hlavně kvůli zvyku vyhledávat v dlouhých textových specifikačních dokumentech. Pokud by měly zůstat zabalené, bylo by prý alepoň vhodné vyměnit pořadí dvou ovládacích prvků nad specifikací – na prvním místě by mělo být právě tlačítko pro rozbalení všech kapitol.

Řešení: Pořadí ovládacích prvků změním

- U specifikací dále některým uživatelům nevyhovovaly vnořené posunovací lišty a omezení maximální výšky jednotlivých kapitol, kterými jsem se snažil obsah zpřehlednit. Tyto lišty jdou sice odstranit funkcí „Force linewrap“, uživatelům ale vadí umístění jejího tlačítka pouze na začátek stránky. Jedna možnost je prý neomezovat maximální výšku vůbec, druhá možnost je pak umístit toto tlačítko na začátek každé kapitoly. Tlačítko je prý také nejasně pojmenováno, navržena byla alternativa - „Remove Scrollbars“.

Řešení: Přidám tlačítka „Remove Scrollbars“ do každé kapitoly.

- Jeden ze zkušených uživatelů OpenGL navrhl další funkci, kterou by podle něj aplikace mohla mít. Mohla by se prý rozšířit o extenze OpenGL ES. Mobilní OpenGL totiž v poslední získává stále větší význam a stránky by tak byly atraktivnější pro větší skupinu vývojářů.

Řešení: Chápu jako návrh pro další rozvoj aplikace.

- V diskusi uživatelům chyběla možnost vkládat obrázky a hlavně zvýraznění syntaxe kódu.

Řešení: V diskusi umožním vkládání kódu.

- Z některých diskusních fór jsou někteří uživatelé zvyklí mít možnost editovat své vlastní příspěvky v diskusi a přáli by si mít možnost měnit své komentáře i zde.

Řešení: Podnět k diskusi.

- V diskusi i dalších sekcích systém nepřijme vložení obsahu kratšího než čtyři znaky. Tento důvod odmítnutí jejich příspěvku ale nezobrazí.

Řešení: Přidám zobrazení důvodu odmítnutí vložení obsahu.

- V sekci výsledků testu extenze je možné přepínat mezi jednotlivými výsledky testu a je zde i možnost zobrazit všechny výsledky najednou. Po zobrazení všech zde ale neexistuje možnost návratu na výpis jednotlivých výsledků, což jednoho z účastníků testu prý zmátlo.

Řešení: Přidám tlačítko pro návrat.

- U nahrávání vlastních souborů účastníkovi testu chybí živý odkaz na nahraný soubor.

Řešení: Přidám živý odkaz.

- V sekci ukázek kódu extenze se objevily různé drobné problémy. Jeden z účastníků testu si například nejprve ne všiml toho, že je potřeba kód obalit speciálními tagy `[code]` a kód vkládal rovnou. Jiný mě upozornil, že ukázky kódu mohou být ve více různých jazycích, takže by mělo jít přidat více různých tagů pro různé druhy jazyků. Další účastník měl podobný problém jako první a jako řešení navrhl script, který by kontroloval přítomnost tagů v textu. Pokud by se v textu při pokusu o odeslání žádný tag nevyskytoval, script by odesílání zastavil a uživatele upozornil, že kód nemá označen.

Řešení: Mohu přidat zmíněnou detekci a tagy pro více jazyků, které Syntax Highlighter podporuje.

- Problém související s prohlížečem Internet Explorer 8 (a pravděpodobně i nižší verze) se objevil u podsekcí komentářů všech sekcí. Pokud je komentářů více jak tři, zobrazí se pouze tři nejnovější a ostatní zůstanou skryté. Zobrazí se až po spuštění JavaScriptové funkce, která manipuluje se styly. Použit je zde bohužel výraz, kterým Internet Explorer nerozumí, bude tedy nutná drobná úprava.

Řešení: Internet Explorer nezná JS funkci "setProperty". Opravím na kompatibilní řešení.

- Poslední nalezený problém se týká použitého opensourcového WYSIWYG editoru TinyMCE, který má aplikace využívá v sekci přidávání testů extenzí. Při testování byla používána neaktuální verze, která korektně nefungovala s prohlížečem Internet Explorer 9.

Řešení: Použití nejaktuálnější verze TinyMCE by mělo tento problém vyřešit.

Kapitola 6

Závěr

6.1 Shrnutí splnění cílů

Podařilo se mi vytvořit funkční stránku pro porovnávání extenzí OpenGL. Aplikace pravidelně přebírá obsah oficiálních stránek OpenGL Registry a transformuje tento obsah do přehlednější a použitelnější podoby. Uživatelé mají i lepší nástroje pro vyhledávání extenzí.

Návštěvníci mohou po registraci přidávat ukázky kódu použití extenzí, nahrávat aplikace testující výkonnost extenzí a vést diskuse. Standardizování výstupů testů extenzí zaručuje uživateli snadné nahrávání výsledků a jejich přehledné zobrazení.

Provedenými testy použitelnosti jsem zjistil některé nedostatky. Většinu z nich jsem odstranil. Účastníci testů se shodují, že aplikace je užitečná a mohli by ji v budoucnosti používat, důležitá pro ně ale bude hlavně komunita těchto stránek.

6.2 Možnosti dalšího vývoje aplikace

V době vypracování projektu byla aplikace ještě stále umístěná na dočasné adrese. Stránky jsou funkční, po přesunu na adresu trvalou a po nárůstu počtu uživatelů bude možná potřeba revidovat uživatelské pravomoce. Vzhledem k možnosti zneužití, kvůli povolení nahrávání souborů na server, bude záležet na poměru počtu problémových a dohlížejících uživatelů.

Ve spolupráci s tvůrci testů extenzí bude také třeba vytvořit a umístit na stránky návod a šablonu pro vytváření dalších testů.

Části aplikace, stahující seznam a specifikace extenzí, budou muset být upraveny v případě, že se změní struktura dat na oficiálních stránkách OpenGL Registry. V případě přechodu z textového formátu specifikací na slibovaný formát XML pak budou muset být tyto parsovací metody přepsány úplně.

Nakonec zmíním jako možnost dalšího rozvoje aplikace návrh jednoho z účastníků testu. Seznam extenzí OpenGL by se mohl rozšířit o extenze OpenGL ES, což je OpenGL pro embedded systémy, například mobilní telefony. V případě realizace tohoto návrhu by bylo třeba rozhodnout míru rozdělení těchto dvou seznamů a vymyslet řešení kolizí, kdy se jedna extenze může objevit v obou seznámech.

Literatura

- [1] ARB. OpenGL Registry, .
<http://www.opengl.org/registry/>, stav z 22. 5. 2011.
- [2] ARB. OpenGL Wiki, .
http://www.opengl.org/wiki/Main_Page, stav z 22. 5. 2011.
- [3] GOOGLE. Custom Search Engine.
<http://www.google.com/cse/?hl=cs>, stav z 22. 5. 2011.
- [4] MARCHESIN, S. The OpenGL extension cross reference.
<http://people.freedesktop.org/~marcheu/extensions/>, stav z 22. 5. 2011.
- [5] web:. Apple OpenGL Extensions Guide, .
<http://developer.apple.com/./reference.html>, stav z 22. 5. 2011.
- [6] web:. PHP: DOM - Manual, .
<http://php.net/manual/en/book.dom.php>, stav z 22. 5. 2011.
- [7] web:. TinyMCE Editor, .
<http://tinymce.moxiecode.com/>, stav z 22. 5. 2011.
- [8] web:. PHP: PDO - Manual, .
<http://php.net/manual/en/book.pdo.php>, stav z 22. 5. 2011.
- [9] web:. SyntaxHighlighter, .
<http://alexgorbatchev.com/SyntaxHighlighter/>, stav z 22. 5. 2011.
- [10] web:. script.aculo.us Documentation, .
<http://madrobby.github.com/scriptaculous/>, stav z 22. 5. 2011.

Příloha A

Seznam použitých zkratk

AJAX Asynchronous JavaScript and XML

CSS Cascading Style Sheets

DOM Document Object Model

DTD Document Type Definition

HTML HyperText Markup Language

JS JavaScript

PHP PHP: Hypertext Preprocessor

UI User Interface

XHTML Extensible HyperText Markup Language

XML Extensible Markup Language

Příloha B

Instalační příručka

Stručný návod pro instalaci projektu, detailnější můžete nalézt v adresáři projektu.

- Projekt je napsaný v jazyce PHP a používá databázi MySQL.
- Nejprve založte databázi MySQL a vytvořte potřebné tabulky. SQL příkazy naleznete v adresáři projektu pod názvem "createDB.sql". Podle programu, který pro správu databáze používáte, bude možná třeba upravit uvozovky příkazů.
- Do cílového adresáře zkopírujte veškerý obsah adresáře projektu. Server musí mít nainstalované PHP ve verzi 5.1 a vyšší. Nesmí být zapnutý Safe Mode.
- Všem souborům v projektovém adresáři "specification" nastavte maximální práva (chmod 777)
- Upravte soubor "dbinfo.php" v adresáři "database", zadejte do něj údaje pro připojení k Vaší databázi.
- Nyní stránky spusťte a přihlašte se jako administrátor (jméno: admin, heslo: admin, heslo si po přihlášení změňte). Otevřete sekci "Manual update" a pro počáteční naplnění databáze stiskněte tlačítko "Download and parse all". Po celou dobu průběhu prvotního stahování dat stránku nezavírejte.
- Pro nastavení služeb společnosti Google, jako je vyhledávání a reCAPTCHA, je nutné se zaregistrovat na stránce <https://www.google.com/accounts/NewAccount> a aktivovat si pro svůj účet služby Google Custom Search a reCAPTCHA. Vygenerované klíče a kódy zapište do zdrojových souborů v adresáři "google". Podrobnosti naleznete v souboru "howto.txt" ve stejném adresáři.

Příloha C

Obsah příloženého CD

- index.html —> Výchozí stránka projektu
- readme.txt —> Popis obsahu CD
- install.txt —> Popis instalace projektu
- project.zip —> Projekt v archivu ZIP
- text/ —> Adresář s textem BP
 - BP.pdf —> Text bakalářské práce
 - BP.zip —> Archiv se zdrojovými soubory BP (LATEX)
- src/ —> Zdrojové kódy projektu
 - project/ —> Adresář projektu
 - * data/
 - * databace/
 - * discussion/
 - * examples/
 - * google/
 - * graphics/
 - * jscripts/
 - * main/
 - * menu/
 - * search/
 - * specification/
 - * tests/

Příloha D

Testování: Veřejná část Screeneru

identifikační číslo:

- 1. Jaká je Vaše znalost angličtiny? (Zakroužkujte variantu, která Vás nejlépe popisuje)
 - a) Žádná
 - b) Začátečník
 - c) Pokročilý
 - d) Nevím / Nechci odpovědět

- 2. Jaká je Vaše věková skupina?
 - a) Mladší než 8 let
 - b) 8-12 let
 - c) 13-18 let
 - d) 19-26 let
 - e) 27-37 let
 - f) 38-60 let
 - g) 61-80 let
 - h) Starší než 80 let
 - i) Nechci odpovědět
 - f) Nevím / Nechci odpovědět

- 3. Jaká je Vaše znalost OpenGL?
 - a) Výborná
 - b) Dobrá
 - c) Základní
 - d) Žádná
 - e) Nevím / Nechci odpovědět

- 4. Vyhledával(a) jste si někdy informace o extenzích OpenGL?
 - a) Ano
 - b) Ne
 - c) Nevím / Nechci odpovědět

Příloha E

Testování: Neveřejná část Screeneru

- 1. Jaká je Vaše znalost angličtiny? (Zakroužkujte variantu, která Vás nejlépe popisuje)
Alespoň základní znalost angličtiny vyžadujeme, vybíráme tedy účastníky B) a C).
- 2. Jaká je Vaše věková skupina?
Budeme pracovat pouze s dospělými, vybíráme tedy účastníky D) až H).
- 3. Jaká je Vaše znalost OpenGL?
Vybíráme dva účastníky C) a dva účastníky B) nebo A).
- 4. Vyhledával(a) jste si někdy informace o extenzích OpenGL?
Pokud účastník u předchozí otázky vyplnil B) nebo A), vyžadujeme u této otázky odpověď A).

Příloha F

Testování: Pre-test dotazník (ústně)

- 1. Jak se dnes cítíte?
- 2. Už jste někdy něco testoval? Kdy?
- 3. Pokud jste si někdy hledal informace o extenzích OpenGL, kde?
- 4. Co od dnešního testování očekáváte?

Příloha G

Testování: Seznam úloh

Vítejte na dnešním testování!

Naším dnešním cílem je otestovat webovou aplikaci pro porovnávání extenzí OpenGL. Aplikace obsahuje aktuální seznam všech extenzí a jejich specifikace. Uživatelům umožňuje přidávat vlastní ukázky kódu a výkonnostní testy extenzí. Stránka uživateli také dává možnost vést o každé extenzi diskusi.

Veškerý průběh testu prosím nahlas komentujte.

Příjemné testování.

- 1. Pokuste se nalézt extenzi "SGI_texture_color_table"
- 2. Zjistěte její závislosti na jiných extenzích
- 3. Nalezněte extenzi "ARB_get_proc_address"
- 4. Podívejte se na ukázky kódu použití této extenze
- 5. Podívejte se na testy této extenze a jejich výsledky
- 6. Zaregistrujte se, přihlašte se a přispějte do diskus u této extenze
- 7. Odhlašte se a znovu se přihlašte jako administrátor (jméno "admin", heslo "admin")
- 8. Přidejte ukázkou kódu k libovolné extenzi
- 9. Přidejte test k libovolné extenzi a nahrajte k němu vlastní soubor
- 10. Odhlašte se

A to je prozatím vše. Děkujeme.

Příloha H

Testování: Post-test dotazník (ústně)

- 1. Jaký máte z dnešního testování pocit?
- 2. Co se Vám na aplikaci nejvíce líbilo?
- 3. Jaký úkol Vám dělal největší problémy?
- 4. Co byste na aplikaci nejraději změnil?
- 5. Dovedete si představit, že byste tuto aplikaci někdy v budoucnu využíval?
- 6. Máte nějakou vlastní připomínku? Chtěl byste něco dodat?

Příloha I

O dočasném umístění projektu

V době odevzdávání práce existují dvě dočasná umístění projektu.

Obě jsou na serverech společnosti Internet Centrum. Tento hosting je zdarma, server ale do kódu přidává reklamní lištu a stránky občas nejsou dostupné. Limituje také počet připojení do databáze na jednoho uživatele. Tyto verze vznikly kvůli testování. První má databázi prázdnou, druhá verze v seznamu je zaplněna testovacími příspěvky typu "Lorem Ipsum".

- <http://ogles.hu.cz>
- <http://score.phorum.cz/ogl>