



Projekt do předmětů IPP a ICP

Dáma 2008

Projektová dokumentace

Autoři:

Černý Lukáš	< xcerny37@stud.fit.vutbr.cz >
Dvořák Miroslav	< xdvora11@stud.fit.vutbr.cz >
Mrázek Petr	< xmraze03@stud.fit.vutbr.cz >
Oujeský Miroslav	< xoujes00@stud.fit.vutbr.cz >

1 Úvod

Tento dokument si klade za cíl Vás seznámit s pojetím a průběhem řešení projektu do předmětů IPP a ICP - „Jednoduchá síťová hra – dáma“.

Projekt je rozdělen na dvě části – klient s grafickým rozhraním pro hraní hry, případně přehrávání zaznamenaných her a server pro řízení komunikace mezi hráči.

Projekt byl implementován v jazyce C++ za použití technologie XML pro ukládání a výměnu dat. Součástí projektu je i skript pro zjišťování statistik zdrojového kódu implementovaný v jazyce Python a programátorská dokumentace popisující jednotlivé třídy a jejich rozhraní.

2 Serverová část aplikace

2.1 Návrh

Server je navržen tak, aby zvládl zpracovávat několik klientů najednou. V původním návrhu se počítalo s tím, že se využijí zprávy (events) pro obsluhování klientů. Bohužel vzhledem k problémům s použitím zpráv ve wxWidgets konzolové aplikaci bylo nutno tento návrh přepracovat. Konečný návrh řešení serveru využívá pro obsluhu klientů vlákna. Je to řešení poněkud složitější, než kdyby se využívalo zpráv, ale stejně dobré po stránce výkonnostní (možná dokonce lepší).

Celý proces komunikace klientů se serverem funguje následovně: server naslouchá na portu a čeká na příchozí spojení, jakmile je spojení navázáno je předáno do nového vlákna kde se následně zpracovává.

Komunikace s klientem je řešena pomocí krátkých XML požadavků, které se na serveru rozparsují. Formát XML je použit proto, že jak klient, tak i server používají na ukládání souborů XML parser a tudíž by bylo nemístné přemýšlet nad nějakým vlastním složitým parserem požadavků.

2.2 Protokol

Jak již bylo uvedeno výše komunikační protokol je implementován formou XML. Při navrhování protokolu bylo dbáno hlavně na jednoduchost a stručnost. Úplný výpis protokolu je v programátorské dokumentaci. Zde bych chtěl uvést příklad jak funguje například vyzvání hráče ke hře.

Hráč 1 sestaví požadavek

```
<duel player="hrac1" with="hrac2" />
```

tento požadavek následně odešle na server. Server požadavek vezme a přepośle ho hráči 2. Hráč 2 potom co obdrží požadavek se rozhodne zda výzvu přijme, či nikoliv. Pokud výzvu nepřijme tak pošle na server zprávu

```
<duel player="hrac1" with="hrac2" accept="false" />
```

a server zprávu pouze přepośle hráči 1. Jestliže však hráč 2 výzvu přijme, pošle

```
<duel player="hrac1" with="hrac2" accept="true" />
```

a server si uloží novou hru a následně pošle požadavek hráči 1.

Jak je vidět komunikace mezi klienty je vskutku jednoduchá jak na pochopení, tak na implementaci.

2.3 Implementační detaily

Vzhledem k tomu, že server ve finální verzi používá vlákna, bylo třeba zajistit jejich vzájemnou komunikaci a synchronizaci. Synchronizace je řešena způsobem globálních seznamů do nichž se řídí přístup přes takzvané mutexy (z ang. mutual exclusion), které zabezpečí, že v jednu chvíli má do proměnné uzavřené mutexem přístup pouze vlákno které ji uzavřelo.

Jelikož server potřebuje pro svou činnost také shromažďovat data ukládá si potřebná data do XML

souborů. Jsou to tyto soubory:

players.xml – obsahuje seznam registrovaných uživatelů

savedgames.xml – obsahuje seznam uložených her

game*.xml – jsou soubory s jednotlivými hrami

3 Klientská část aplikace

3.1 Návrh

Návrh klienta byl z větší části určen vzhledem grafického rozhraní, protože právě třídy oken jsou největší částí klienta. Zpočátku byly generovány pomocí nástroje wxGlade, ten však velmi rychle přestal vyhovovat požadavkům na návrh aplikace. Logika hry byla tvořena a testována zvlášť - je vyjádřena pomocí tříd `Deska`, `Kamen` a `Dama` – tyto jsou využívány i na serveru a řídí průběh a chování celé hry. Síťová komunikace je řešena tak, že klient má Singleton (`clSocket`), který zapouzdřuje implementační detaily síťové komunikace. Jednotlivá okna, která chtějí přijímat zprávy ze sítě musí být potomky třídy `Receiver` a zaregistrovat se pro jejich odebrání. Rozesílání zpráv oknům je pak řízeno časovačem (`wxTimer`).

3.2 Implementační detaily

Konfigurace

Konfigurace klienta je uložena v XML souboru `cl_config.xml` a obsahuje kromě cesty k nápovědě i naposledy zadané uživatelské jméno a adresu serveru. Pro přístup ke konfiguraci byla použita třída `ClientConfig` která využívá návrhový vzor „Singleton“ - pro celou aplikaci existuje jedinečná instance této třídy dostupná ze všech částí aplikace.

Záznamy her

Při spuštění hry je uživatel dotázán, zda si přeje zaznamenat průběh hry. Pokud ano, je každý tah uložen do XML souboru (pro který jsme zvolili příponu `.d2008`). V prohlížeči záznamů je potom možno tyto soubory otevřít a procházet hru po jednotlivých tazích.

Pravidla hry

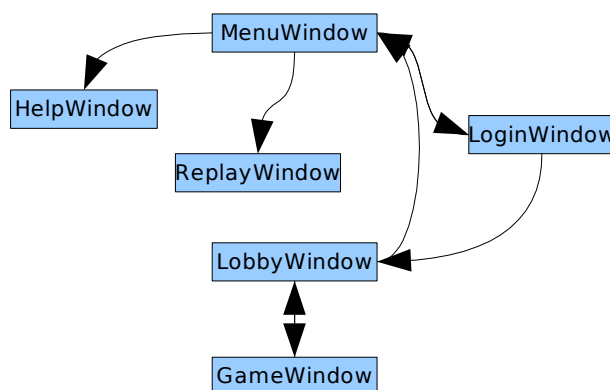
Tak jako u každé hry s delší tradicí, tak i u dámy existuje více variant pravidel. My jsme si vzali jako referenční pravidla stanovená Českou federací dámy, která se nacházejí na <http://www.damweb.cz/pravidla/cdfull.html>

3.3 Grafické rozhraní aplikace

GUI aplikace bylo navrženo v programu wxGlade, který umožňuje vizuální návrh aplikace a nastavení důležitých vlastností jednotlivých prvků aplikace. Pro takto navržená okna je poté vygenerován C++ kód, který je možno okamžitě přeložit.

Diagram oken aplikace a jejich vzájemných vztahů je zobrazen na obrázku

Po spuštění aplikace je zobrazeno okno `MenuWindow` které obsahuje tlačítka pro spuštění jednotlivých hlavních částí klientské aplikace.



Ilustrace 1: Diagram oken aplikace

Okno `HelpWindow` zobrazuje nápovědu ve formátu HTML, jejíž umístění je dáno v konfiguračním souboru aplikace.

Okno `ReplayWindow` slouží k přehrávání dříve uložených záznamů her. Obsahuje šachovnici, na které se zobrazuje situace v daném tahu a ovládací prvky pro přehrávání tahů. Je možné přehrávat jak ručně po jednom tahu, nebo automaticky s nastavitelnou prodlevou mezi jednotlivými tahy.

Okno `LoginWindow` slouží k přihlášení uživatele na herní server. Obsahuje editační pole pro adresu serveru, uživatelské jméno a heslo. Pokud uživatel ještě není na serveru zaregistrován, lze jej v tomto okně zaregistrovat. Obsah polí pro adresu serveru a uživatelské jméno je ukládán do konfiguračního souboru aplikace, čímž odpadá nutnost tyto informace pokaždé znovu zadávat. Po úspěšném přihlášení se toto okno zavře a otevře se okno `LobbyWindow`.

V okně `LobbyWindow` je seznam uživatelů, kteří jsou online, seznam rozehraných her aktuálního uživatele a chat. Obsah tohoto okna je periodicky aktualizován, je možná i ruční aktualizace.

Po vybrání hráče v seznamu online hráčů je možné jej vyzvat ke hře. Vyzvanému hráči se zobrazí okno s dotazem, zda výzvu přijímá či ne. Pokud výzvu přijme, je vytvořena nová hra. O rozhodnutí je následně informován vyzyvatel.

Po vybrání hry v seznamu her je možné tuto hru spustit. Po spuštění hry se zobrazí okno `GameWindow`. To obsahuje herní plochu, zobrazuje který hráč je právě na tahu a statistiky o počtu zbývajících kamenů každého z hráčů. Pokud je hráč na tahu, jsou barevně zvýrazněny ty kameny, kterými může táhnout. Po kliknutí na jeden z těchto kamenů je tento aktivován a již není možno vybrat pro tento tah jiný kámen. Po kliknutí na validní prázdné pole je tah dokončen a hraje oponent. Hru je možné v kterékoliv fázi přerušit a později se k ní opět vrátit.

4 Organizace a nástroje

4.1 Organizace

V našem 4-členném týmu jsme si rozdělili práci následujícím způsobem:

- Tvorba serverové části
- Tvorba klientské části
- Tvorba dokumentace a nápovědy
- Tvorba skriptu pro statistiky zdrojových kódů a výpomoc s klientskou částí

4.2 Nástroje

TinyXML a TinyXML++

Pro práci s XML jsme využili knihovnu `TinyXML` a její nadstavbu `TinyXML++`.

`TinyXML` (<http://www.grinninglizard.com/tinyxml>) je jednoduchá C++ knihovna pro zpracovávání XML souborů.

`TinyXML++` (<http://code.google.com/p/ticpp>) je nadstavba nad knihovnou `TinyXML`, která obaluje její jednotlivé části a přidává jim vlastnosti které více podporují objektově orientovaný návrh, jako jsou např. výjimky.

wxGlade

wxGlade (<http://wxglade.sourceforge.net>) slouží k vizuálnímu návrhu grafického rozhraní aplikace. Pokužili jsme jej k vygenerování příslušných tříd pro okna aplikace. Tento nástroj nám velmi usnadnil a urychlil tvorbu GUI – ušetřil spoustu času, který bychom jinak strávili studiem dokumentace wxWidgets a hledáním možností nastavení jednotlivých komponent.

Subversion

Pro správu a sdílení projektu mezi členy týmu jsme si vybrali systém Subversion (<http://subversion.tigris.org>), se kterým máme již z dřívějších projektů dobré zkušenosti.

Pro hostování našeho projektu jsme si vybrali online službu Assembla (<http://www.assembla.com>), která poskytuje zdarma Subversion repozitář a další možnosti pro správu projektu (úložiště souborů, wiki, atd.)

Doxygen

Pro tvorbu programátorské dokumentace jsme využili systém Doxygen (<http://www.doxygen.org>). Doxygen prochází definované soubory se zdrojovým kódem a na základě specifických komentářů sestavuje interaktivní programátorskou dokumentaci ve formátu HTML.

4.3 Týmová spolupráce

Jako hlavní komunikační kanály mezi členy týmu byly v průběhu řešení projektu používány instant-messagingové služby a IRC chat.

Důležité organizační záležitosti a prvotní plánování projektu jsme probrali na několika osobních schůzkách. Zde proběhlo ustanovení vedoucího týmu, rozdělení jednotlivých činností a rámcový návrh řešení.

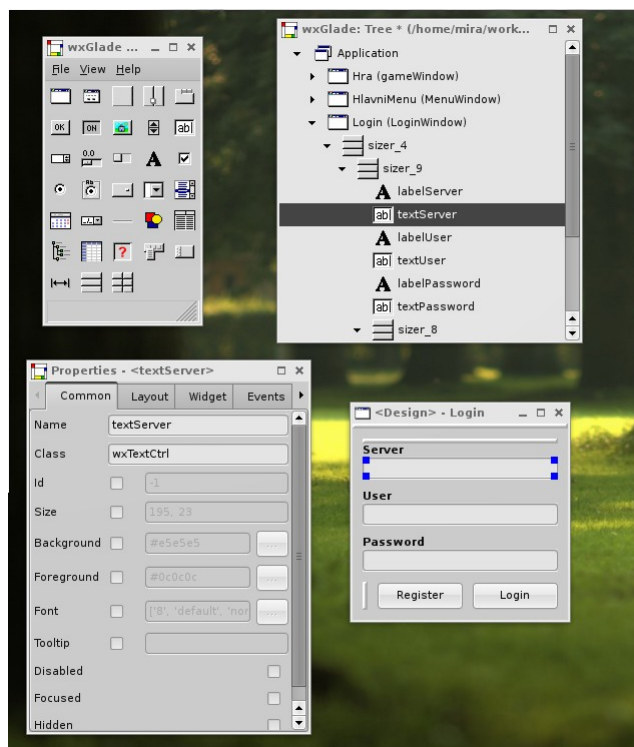
Při implementaci projektu jsme se spoléhali na náš systém pro správu verzí Subversion a interní programátorskou dokumentaci.

5 Závěr

Celý projekt byl velice cennou zkušeností, jak z hlediska programátorských dovedností, tak týmové spolupráce. Rozšířili jsme si naše znalosti o základy práce s nástroji wxWidgets a také o některé poznatky ze samotného vývoje GUI. Nelehkou částí byla zejména komunikace serverové části s klientskou částí projektu.

Návrh projektu se z nemalé části odehrával až za samotného vývoje, což se místy projevilo jako poměrně brzdicí a matoucí faktor, proto budeme určitě při vývoji dalších projektů klást mnohem větší důraz na formální návrh a zkoumání hlubších závislostí před započatím samotné implementace.

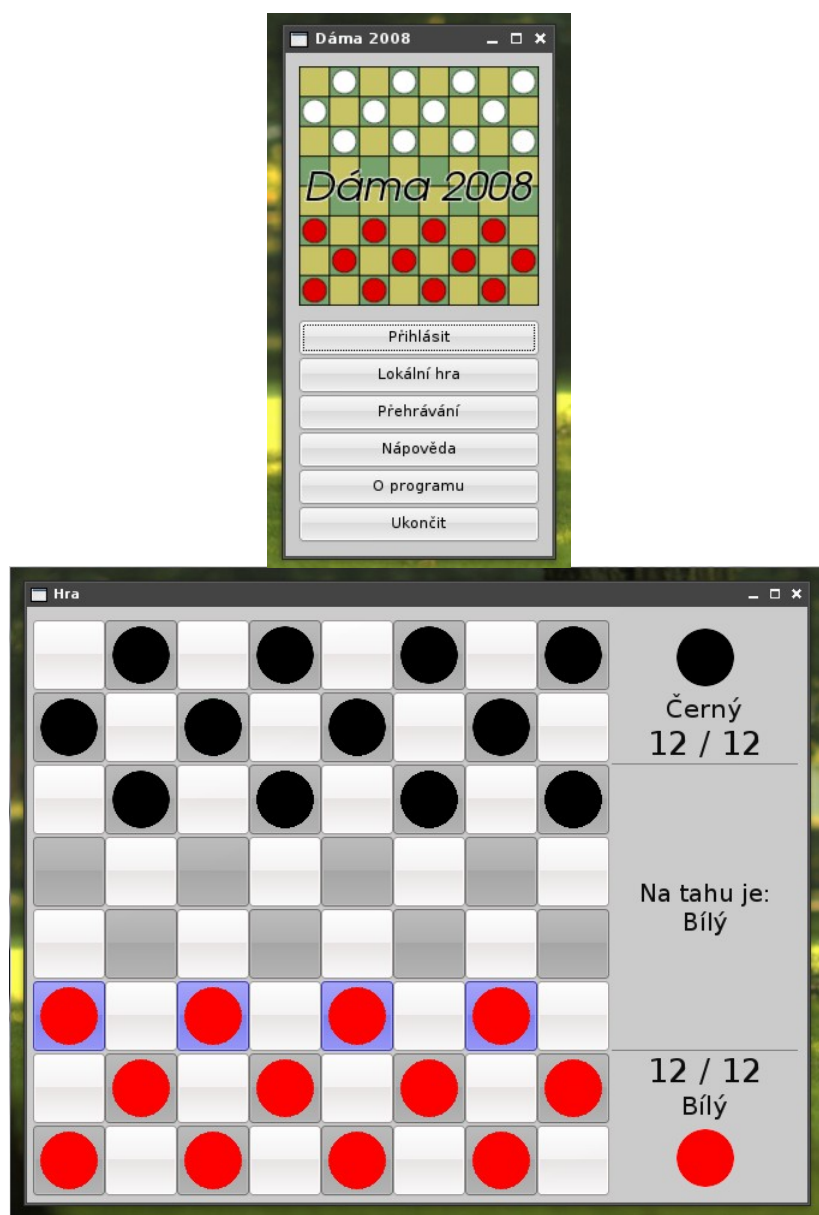
Náplň projektu byla velmi zajímavá a motivující, stejně tak testování hotové aplikace bylo příjemnou odměnou po odvedené práci.



Ilustrace 2: prostředí wxGlade

6 Přílohy

A Náhledy grafického rozhraní



Ilustrace 4: Herní plocha