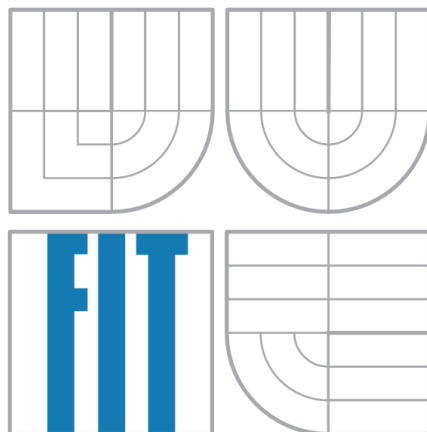


**Vysoké učení technické v Brně
Fakulta informačních technologií**



**Projektová dokumentace projektu
do předmětů IJA a IPP**

tým 27

Řešitelé:

- Nový Ondřej, xnovyo01@stud.fit.vutbr.cz
- Kaláb Jan, xkalab00@stud.fit.vutbr.cz
- Janda Martin, xjanda01@stud.fit.vutbr.cz
- Choleva Ondřej, xchole00@stud.fit.vutbr.cz

Obsah

1. Úvod.....	3
2. Zadání.....	3
3. Analýza a návrh.....	4
3. Implementace.....	5
4. Spolupráce.....	6

1. Úvod

Tato dokumentace se zabývá popisem analýzy, implementace a práce v týmu na projektu do předmětů Principy programovacích jazyků a OOP a Seminář Java.

2. Zadání

Navrhněte a implementujte hru dáma. Aplikace bude typu klient-server. Každý uživatel má své jednoznačné jméno (login) a může využívat libovolného klienta. Nemusíte uvažovat autentizaci uživatelů při registraci/logování (např. pomocí hesla) či zabezpečení komunikace (šifrování apod.)

Server:

- bez grafického uživatelského rozhraní
- zajišťuje registraci nových uživatelů a jejich ověřování
- zajišťuje navazování dvojice hráčů
- zajišťuje komunikaci mezi klienty (uživateli)
- zajišťuje ukládání/načítání rozehraných partií
- seznam uživatelů a rozehrané hry jsou uloženy v XML dokumentech

Klient:

- s grafickým uživatelským rozhraním, obsahuje
 - menu
 - hrací plochu
 - ovládací tlačítka
 - editaci konfiguračního souboru ve formátu XML (načte se při startu, uloží při změně konfigurace)
 - jednoduchý systém nápovědy – HELP; jméno (cesta k) souboru s nápovědou je uloženo v konfiguraci a při instalaci programu bude správně nastaveno (podrobnější informace k nápovědě viz zadání v předmětu IPP)

Akce:

1. bezprostředně po startu klienta se uživatel přihlásí (pokud již existuje) nebo zaregistruje – ověřuje se na serveru
2. po přihlášení uživatele je klient ve stavu off-line, možné aktivity:
 1. uživatel si může otevřít libovolnou partii, kterou si uložil (viz bod 3.3) a off-line si ji přehrávat (krokovat tam i zpět, možnost plynulého spuštění partie s časovou prodlevou mezi kroky)
 2. uživatel se může připojit na server (a tím přepnout do stavu on-line)
3. po připojení na server je klient ve stavu on-line, možné aktivity:
 1. připojením k serveru dává klient najevo, že chce hrát a hledá partnera – současně vidí přihlášené a volné uživatele → vytvoří se dvojice (navzájem se potvrdí)
 2. po utvoření dvojice buď začínají novou hru, nebo (pokud jsou uloženy) pokračují ve vybrané rozehrané partii
 3. na straně klienta je možné průběh hry logovat do XML dokumentu a později si ji přehrát

Průběh hry:

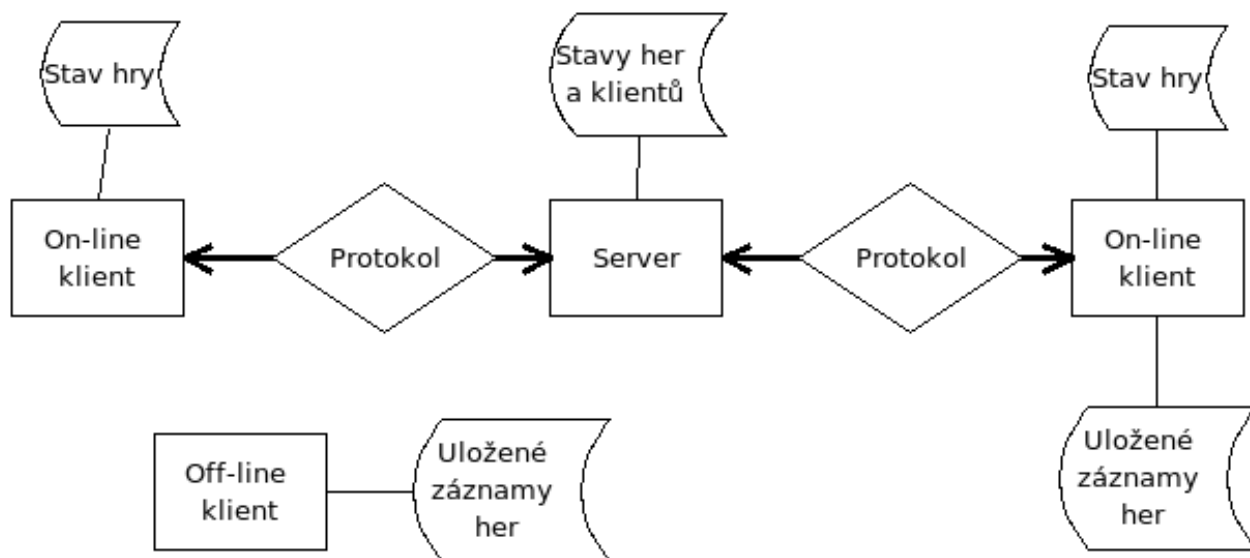
- hráč s bílými kameny začíná
- první táhne, tah se přenese přes server na stranu klienta druhého hráče a první hráč čeká na reakci
- druhý hráč táhne, tah se opět přenese přes server na stranu klienta prvního hráče a druhý hráč čeká na reakci
- kdykoliv v průběhu hry může hráč hru přerušit a nechat uložit její stav na server – druhému hráči se oznámí přerušení hry
- logika hry je implementovaná na straně klienta, tj. klient neumožní provést tah, který není povolen

Součást odevzdání:

- připravte několik ukázek (alespoň dvě) uložených rozehraných partií – tyto soubory se vytvoří/nakopírují do adresáře s přeloženým serverem při překladu aplikace
- připravte několik ukázek (alespoň dvě) logovaných partií – tyto soubory se vytvoří/nakopírují do adresáře s přeloženým klientem při překladu aplikace

3. Analýza a návrh

Na první pohled se projekt skládá ze dvou větších nezávislých celků: klientská aplikace a serverová část.



Obr. č. 1 – Model komunikace

Z pohledu rozdělení úkolů, funkčnosti a objektového návrhu je potřeba problém však rozdělit na menší celky. Při úvahách jsme postupovali následovně:

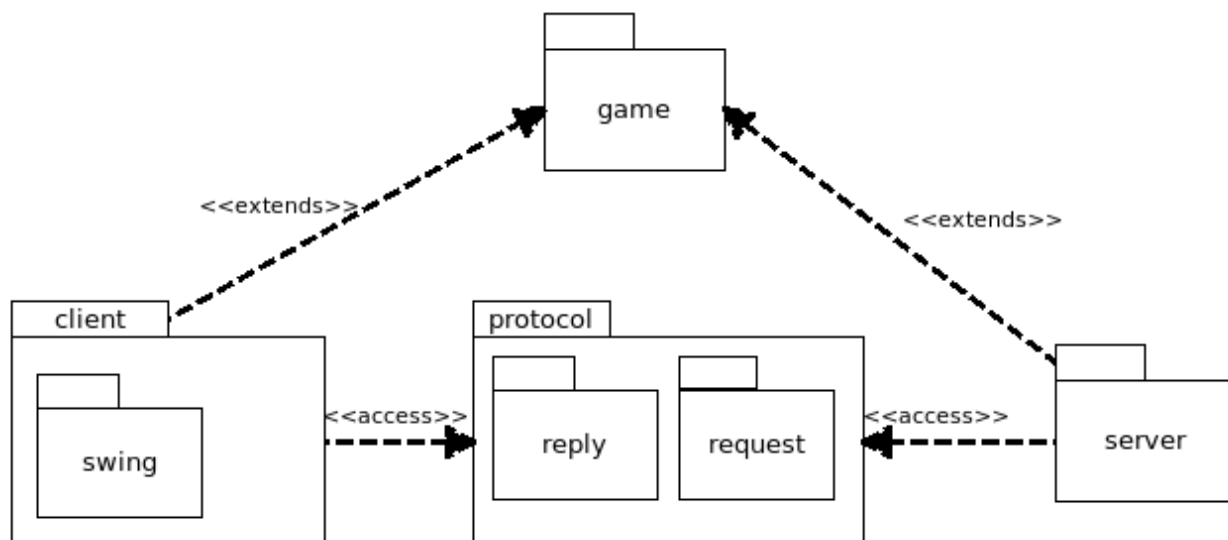
Klientskou aplikaci jsme rozdělili na uživatelské rozhraní, aplikační logiku a herní logiku.

Serverová aplikace obstarává komunikaci s klienty a stará se o stavy hry a připojených klientů. Rovněž i zde je potřeba herní logika pro kontrolu stavů hry. Kdybychom zde herní logiku neměli, mohli by si uživatelé vytvářet falešné klienty a ve hře podvádět.

Poslední komponentou je způsob komunikace klientů se serverem, tedy protokol. Ten opět budou server i klient mít totožný. Protokol definuje typy zpráv o stavu hry a klienta a jejich předávání.

3. Implementace

Z analýzy nám vyplynulo určité rozdělení funkčnosti a dat, které jsme využili pro návrh následujících balíčků tříd.



Obr. č. 2 – Diagram balíčků tříd

client

Balíček *client* reprezentuje klientskou aplikaci. Především jde o aplikační logiku. Dědí z třídy *game* herní logiku. Volá třídy z balíčku *protocol* pro komunikaci se serverem.

client.swing

Podbalíček *swing* zahrnuje třídy starající se o vykreslování a práci s okny, hrací desku a komunikaci s uživatelem prostřednictvím grafického rozhraní.

game

Představuje herní logiku a herní data. Kontroluje tahy a operace nad partií. Uchovává historii partií. Dále umožňuje ukládání hry do XML formátu.

protocol (podbalíčky reply a request)

Třídy zajišťují komunikaci klientů se serverem a naopak.

server

Zahrnuje třídy pro práci serveru. Představují aplikační logiku. Dědí metody ze tříd z balíčku *game* a využívá tříd z balíčku *protocol* pro komunikaci.

Popis komunikačního protokolu

Komunikace mezi klienty a serverem probíhá ve formě čistého textu s následujícími pravidly:

- Jeden řádek představuje jeden příkaz.
- První část řádku je typ příkazu: REQ (request = požadavek) nebo REP (reply = odpověď).
- Druhá část řádku je příkaz.
- Třetí (a další) části řádku jsou parametry příkazu.

Příklad komunikace pak vypadá následovně:

REQ LOGIN Franta

REP LOGIN_OK

4. Spolupráce

Ve všech fázích práce na projektu byla zásadní komunikace a výměna názorů. K tomu sloužily především nepravidelné schůzky. Dále pak komunikace přes Jabber a e-mail.

Na začátku byl stanoven jazyk komentářů (čeština), styl psaní kódu (odsazení, komentáře...) a pojmenování identifikátorů.

Vývoj probíhal jak na platformě GNU/Linux, tak i Microsoft Windows. Jako vývojové prostředí byly použity NetBeans a Eclipse.

Pro pohodlné sdílení a zálohování souborů projektu byl použit systém SVN. Během vývoje jsme si vystačili s jednou větví vývojového procesu.

Programová dokumentace byla vygenerována pomocí systému javadoc.

Podíl a úkoly jednotlivých členů týmu byly:

Ondřej Nový jako vedoucí týmu se zasadil největší měrou o analýzu, implementaci serverové části aplikace a herní logiky. Dále se usilovně podílel na všech ostatních částech projektu.

Kaláb Jan měl na starosti uživatelské rozhraní, grafické zpracování a klientskou část aplikace.

Janda Martin dostal na starosti skript stats.pl a dokumentaci.

Choleva Ondřej pracoval na herní logice a práci s XML soubory .