Západočeská univerzita v Plzni

Úvod do počítačových sítí

KIV/UPS

Dokumentace semestrální práce - Dáma

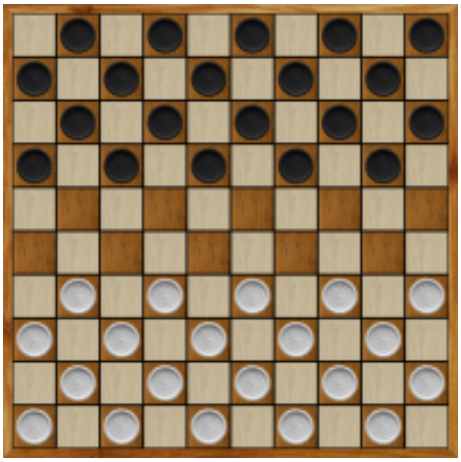
Vojtěch Danišík
A16B0019P
danisik@students.zcu.cz

19. prosince 2018

Obsah

1 Zadání

Vytvořte program realizující vybranou hru. Vytvořte server, který bude obsluhovat více hráčů i her současně a bude schopen uložit stav hry. Klient bude po opětovném přihlášení pokračovat tam, kde přestal. Přenosový protokol je TCP. Jazyk použitý pro server: C. Jazyk použitý pro klienta: Java. Vybraná hra je **Dáma** .



Obrázek 1: Hrací pole dámy

2 Programátorská dokumentace

2.1 Server

Server je naprogramován v jazyce **C**.

2.1.1 Datové struktury

Server používá pro svůj běh tyto datové struktury:

- STATES (enum) - Konstantní proměnné použité jako stavy pro jednotlivé hráče, ve kterých se mohou nacházet
- client - Struktura pro uložení hráče
  - (char*) name - Jméno hráče zvolené při přihlášení
  - (int) socket_ID - Socket ID hráče získané při připojení na server

  - (char*) color - Barva hráče, kterou mu server přiřadí při spuštění hry
  - (STATES) state - Stav, ve kterém se hráč nachází

- clients - Struktura pro uložení všech klientů do pole
  - (int) clients_count - Počet přihlášených hráčů
  - (client**) clients - Pole klientů

- piece - Hrací figurka
  - (char*) color - Barva figurky (black/white)
  - (char*) type - Typ figurky (man/king)

- field - Políčko hrací plochy
  - (int) row - Řádek políčka na hrací ploše
  - (int) col - Sloupec políčka na hrací ploše
  - (char*) color - Barva políčka
  - (piece*) piece - Figurka na políčku (políčko může obsahovat figurku)

- fields - Struktura pro uložení všech políček
  - (int) size - Velikost hrací plochy
  - (field***) all_fields - Pole všech políček na hrací ploše
  - (int) count_pieces - Počet políček na hrací ploše

- wanna_play - Struktura pro zjištění, kteří hráči chtějí hrát hru
  - (int) size - Počet hráčů, kteří chtějí hrát hru
  - (int*) socket_IDs - Pole ID socketů hráčů

- game - Struktura reprezentující hru
  - (char*) name_1 - Jméno prvního hráče
  - (char*) name_2 - Jméno druhého hráče
  - (char*) now_playing - Aktuálně hrající hráč
  - (int) game_ID - ID hry
  - (fields*) fields - Hrací plocha

- games - Struktura pro uložení všech her
  - (int) games_count - Počet rozehraných her
  - (game**) games - Pole všech rozehraných her
- log_info - Struktura pro uložení logovacích informací
  - (int) count_bytes - Počet přenesených bytů
  - (int) count_messages - Počet přenesených zpráv
  - (int) count_connections- Počet připojení na server
  - (int) count_bad_transmissions - Počet špatných přenosů
  - (int) server_running_minutes - Čas běhu serveru (v minutách)

2.1.2 Formáty zpráv

Předpis	**Význam**
login_ok;	Uživatel přihlášen
login_false;error_ID;	Uživatel nepřihlášen + číslo erroru
start_game;player_color;game_ID;	Nová hra vytvořena a startnuta
correct_move;move_type;cp_row; cp_col;dp_row;dp_col;	Správný tah
end_move;	Konec tahu aktuálního hráče
wrong_move;error_ID;	Špatný tah
update_game_ID;game_ID;	Update game ID
play_next_player;	Hraje druhý hráč
end_game;player_state;	Korektní konec hry (žádný error)
end_game_left;error_ID;	Nekorektní konec hry, oponent ukončil hru
end_game_timeout;error_ID;	Nekorektní konec hry, oponent se nepřipojil
opponent_connection_lost;	Oponent ztratil připojení k internetu
connection_restored;	Oponent se vrátil do hry
promote;dp_row;dp_col;	Vylepšení figurky muže na krále
board;name;player_color;game_ID; count_pieces;who_play;piece_x;piece_y; piece_color;piece_type;....;	Aktuální hrací plocha hry

Legenda
error_ID - ID erroru, podle kterého se vypíše určitá hláška u klienta
player_color - Barva hráče, za kterou bude hrát
game_ID - ID hry
move_type - Typ pohybu (2 - klasický pohyb, 3 - pohyb s přeskokem oponenta)
cp_row - Řádek figurky, ze které skáču
cp_col - Sloupec figurky, ze které skáču
dp_row - Řádek figurky, na kterou skáču
dp_col - Sloupec figurky, na kterou skáču
player_state - Hráčův stav po konci hry (win - výhra, draw - remíza, lose - prohra)
count_pieces - Aktuální počet figurek ve hře
who_play - Kdo je momentálně na tahu ve hře
piece_x - Řádek figurky
piece_y - Sloupec figurky
piece_color - Barva figurky (white - bílá, black - černá)
piece_type - Typ figurky (man - muž, king - král)
.... - Opakování parametrů od piece_x do piece_type

2.2 Klient

Klient je naprogramován v jazyce **Java**.

2.2.1 Datové struktury

Klient obsahuje tyto balíčky s třídami:

- connection - Třídy pracující s připojením k serveru, čtením a odesíláním zpráv. Je zde klient reprezentující hráče, čtecí vlákno, které na základě přijmuté zprávy vykoná činnost (zobrazí nové okno).

- constants - Konstanty aplikace (Velikosti okna, zobrazované texty, názvy komponent, ...)
- enums - Výčtové typy (Barvy použité v aplikaci - pro políčka, zvýraznění textu, ...; Posílané a přijímané zprávy;Typy figurek; ID chybových hlášení a jejich převod na text)
- field - Třídy generující políčka do hry
- main - Hlavní spouštěcí třída
- messages - Třídy reprezentující přijímané a odesílané zprávy na server
- piece - Třídy reprezentující figurky
- windows - Hlavní vykreslovací třída
- client - Struktura pro uložení hráče

2.2.2 Formáty zpráv

Předpis	**Význam**
login;name;	Uživatel se chce přihlásit
play;	Uživatel chce hrát hru
client_move;game_ID;cp_row;cp_col; dp_row;dp_col;piece_color;piece_type;	Uživatel chce pohnout figurkou
new_game_no;	Uživatel nechce hrát novou hru

Legenda

name - Jméno hráče
game_ID - ID hry
cp_row - Řádek figurky, ze které skáču
cp_col - Sloupec figurky, ze které skáču
dp_row - Řádek figurky, na kterou skáču
dp_col - Sloupec figurky, na kterou skáču
piece_color - Barva figurky (white - bílá, black - černá)
piece_type - Typ figurky (man - muž, king - král)

2.3 Implementace

2.3.1 Start serveru

Server se po startu snaží obsadit port zadaný při startu (pokud není zadán, použije se defaultní port 10000), pokud se mu to podaří, vypíše server do konzole *Bind OK*
Přijde-li příchozí spojení (socket), server vytvoří sadu deskriptorů pomocí metody *select()* a začne jednomu z tří desktriptorů naslouchat (jeden slouží pro read, druhý pro write, třetí pro chybové hlášení).

2.3.2 Zpracování příchozích zpráv

Každý klient má svůj vlastní desktriptor, který mu naslouchá a stará se o příjem. Po přijmutí zprávy jej rozparsuje a vyhodnotí. Povoleny jsou jen zprávy se správným formátem, viz 2.1.2

2.3.3 Vytvoření nové hry

Klient po přihlášení na server má možnost se připojit do fronty hráčů, kteří chtějí hrát hru. Pokud jsou ve frontě alespoň 2 hráči, server vybere posledního hráče, který chce hrát a náhodně z fronty druhého hráče a spojí je do jedné hry. Každému klientovy bude náhodně přiřazena barva a odeslána zpráva s jejich barvou a ID hrou.

2.3.4 Konec hry

Když hra skončí korektně (jeden hráč sebere všechny figurky tomu druhému), tak je na serveru vyhodnoceno kdo vyhrál/-prohrál nebo zda je remíza a jednotlivé výsledky jsou rozeslány hráčům. Ti mají možnost hrát znovu či nikoliv. Pokud by se jeden z hráčů během hry odpojil ze hry tím, že ukončí klienta i přes to, že mu spojení se serverem funguje, je hra ukončena,

druhý hráč informován o tom, že oponent odešel ze hry a automaticky vyhrává.

2.3.5 Chybové řízení

Kontrola příchozích dat: Server po přijmutí dat je zkontroluje, zda obsahují alespoň jeden středník (oddělovač informací v datech). Pokud je středník na indexu 1 - 20, data jsou správná a dále s nimi pracujeme, jinak data označíme za nevalidní a dále s nimi nepracujeme.
Jakmile na server přijdou validní data (ve správném formátu, lze rozdělit text na několik částí pomocí znaku *;*), server zkontroluje první část dat, zda obsahuje jedno z klíčových slov viz 2.1.2. Pokud ano, tak server dále kontroluje stav klienta ve kterém se nachází a zda tuto instrukci může provést (př. Klient je v rozehrané hře a pošle zprávu, která má klienta přihlásit. Bez patřičných ošetření se klient znovu přihlásí i přes to, že už se jednou přihlašoval.). Pokud je klient ve stavu, který neumožňuje vykonat zadanou instrukci, tak zprávu označíme za nevalidní a dále s ní nepracujeme. Tento způsob je aplikován i na straně klienta.
Ztráta spojení: Pokud klient ztratí spojení ser serverem (metoda **recv()** vrátí hodnotu 0), přejde do stavu *disconnect* a spustí se časovač, který je nastavený na **5 minut**. Pokud se do té doby klient pod stejným jménem přihlásí, bude vpuštěn do rozehrané hry, ve které byl před tím, než ztratil spojení. Server tomuto klientovy odešle ID hry, počet figurek a jejich pozice, barvu a typ, aby se klient mohl ihned zapojit do hry.
Jestliže se klient do 5 minut nepřipojí (pomocí návratové hodnoty metody **select()**, která je rovna 0 v případě timeoutu), server ho vymaže z pole připojených hráčů, případně ukončí rozehranou hru, jestliže se klient nacházel uprostřed rozehrané hry a uvolní klientovo dříve zadané jméno.

3 Uživatelská dokumentace

3.1 Překlad

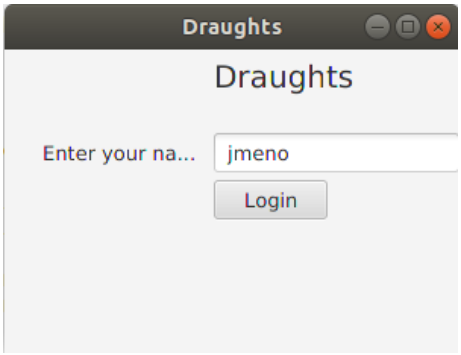
Přeložení zdrojových souborů se provádí zadáním příkazu *make* v terminálu v kořenovém adresáři staženého souboru. Zdrojové soubory pro klienta najdeme ve složce **java_src** a pro server ve složce **c_src**.

3.2 Spuštění serveru

Server se spouští pomocí příkazu *./server -address [IPv4] -port [port_ID]*, kde *IPv4* je validní IPv4 adresa, které bude server naslouchat (pokud je potřeba naslouchat všem, tak tento parametr se nezadává) a *port_ID* je číslo portu v rozsahu 1024 - 65535, na kterém bude server naslouchat (defaultní port je 10000).

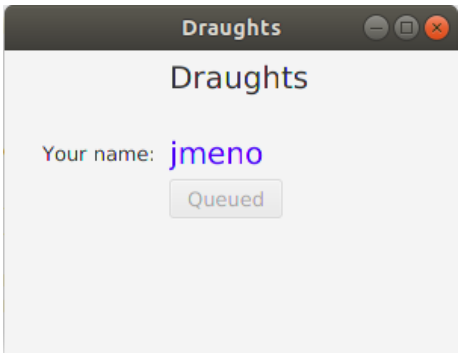
3.3 Spuštění klienta

Klient se spouští pomocí příkazu *java -jar client.jar -address [address] -port [port_ID]*, kde *address* je IPv4 adresa serveru a *port_ID* je číslo portu v rozsahu 1024 - 65535, na kterém bude klient naslouchat.
Po spuštění .jar souboru se otevře přihlašovací okno, kde se zadá jméno, pod kterým bude uživatel hrát. Pokud je toto jméno volné, server vás přihlásí. Pokud bude zabrané, server pošle chybové hlášení.



Obrázek 2: Přihlašování

Po přihlášení se dostaneme do lobby okna, ve kterém po kliknutí na tlačítko *Play* budete přidáni do fronty pro hráče čekající na hru. Pokud ve frontě budou alespoň 2 hráči, budete přiřazeni do hry.

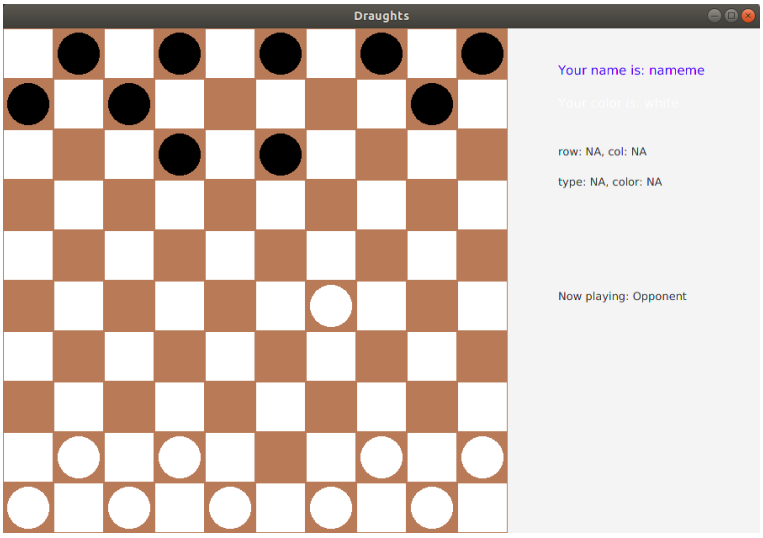


Obrázek 3: Lobby

Po úspěšném přiřazení do hry vám bude přiřazena náhodně serverem barva, za kterou budete hrát. Tuto barvu uvidíte v pravém horním rohu (bílá/černá). Hraní dámy funguje následovně:

- Kliknutí na svojí figurku a posunout jí kliknutím na novou pozici vzhledem k pravidlům dámy. Po každém kliknutí lze vidět pozici políčka, na které jste kliknuli a typ a barva figurky, pokud se nachází na políčku.
- Pokud je krok nevalidní, server pošle chybovou hlášku, která

se zobrazí právě hrajícímu klientovy s textem, jaký nevalidní krok klient provedl. Pokud je krok validní, pouze se v pravé části okna zobrazí *Now playing: Opponent*.



Obrázek 4: Hra dáma

4 Závěr

Práce splňuje zadání, server umí odbavit požadavky několika klientů najednou a zároveň je natolik stabilní, aby jej výpadek nebo chyba jednoho z klientů neukončila chybou. Klient lze spustit na systému Linux i Windows (testováno na Windows 10), server byl testován na systému Linux – konkrétně na distribuci Ubuntu 18.04.1 LTS.