



**FAKULTA  
APLIKOVANÝCH VĚD  
ZÁPADOČESKÉ  
UNIVERZITY  
V PLZNI**

# **úvod do počítačových sítí hra: dáma (anglická verze)**

**kiv/ups**

pavel třeštík  
a17b0380p

2. února 2021

# Obsah

<b>1</b>	<b>popis hry dáma (anglická verze)</b>	<b>3</b>
1.1	pohyb . . . . .	3
1.2	Braní kamenů protivníka . . . . .	4
1.3	Pohyb . . . . .	4
<b>2</b>	<b>Popis protokolu</b>	<b>6</b>
2.1	Obecný tvar zpráv . . . . .	6
2.2	Tabulka instrukcí a stavový diagram protokolu . . . . .	7
2.3	Popis instrukcí a odpovědí . . . . .	8
2.3.1	Obecné negativní odpovědi serveru . . . . .	8
2.3.2	Instrukce: CONNECT . . . . .	9
2.3.3	Instrukce: CREATE_LOBBY . . . . .	10
2.3.4	Instrukce: JOIN_GAME . . . . .	11
2.3.5	Instrukce: DELETE_LOBBY . . . . .	11
2.3.6	Instrukce: LOBBY . . . . .	12
2.3.7	Instrukce: TURN . . . . .	12
2.3.8	Instrukce: DISCONNECT . . . . .	13
2.3.9	Instrukce: OPPONENT_JOIN . . . . .	14
2.3.10	Instrukce: OPPONENT_TURN . . . . .	14
2.3.11	Instrukce: OPPONENT_DISC . . . . .	15
2.3.12	Instrukce: OPPONENT_RECO . . . . .	15
2.3.13	Instrukce: OPPONENT_LEFT . . . . .	16
<b>3</b>	<b>Programátorská dokumentace</b>	<b>17</b>
3.1	Server . . . . .	17
3.1.1	Struktura serveru . . . . .	17
3.1.2	Obsluha spojení . . . . .	18
3.2	Klient . . . . .	18
3.2.1	Struktura klienta . . . . .	19
<b>4</b>	<b>Uživatelská dokumentace</b>	<b>20</b>
4.1	Server . . . . .	20
4.1.1	Překlad . . . . .	20

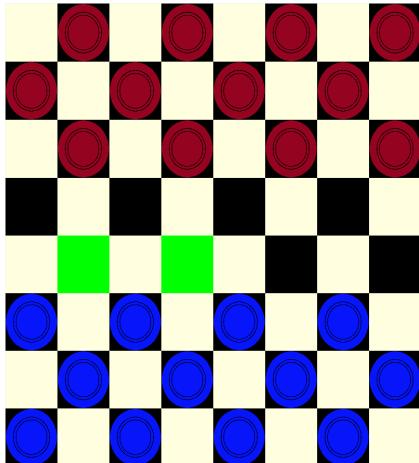
4.1.2	Spuštění . . . . .	20
4.2	Klient . . . . .	21
4.2.1	Překlad . . . . .	21
4.2.2	Spuštění . . . . .	21
<b>5</b>	<b>Závěr</b>	<b>22</b>

# 1 popis hry dáma (anglická verze)

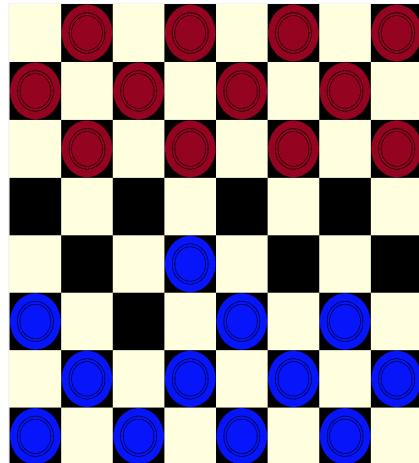
jako téma semestrální práce jsem si zvolil hru dáma. zvolil jsem takzvanou anglickou dámu. tedy hra se hraje na šachovnici s 64mi polí a vyhrává ten hráč, který první sebere všechny kameny protivníka. hráči mají kameny na černých polích šachovnice a smí se pohybovat pouze diagonálně, tedy pouze po černých polích. kamenů jsou dva druhy: pěšák a král. rozdíl mezi klasickou verzí dámy a anglickou démou je, že v anglické verzi král (česky označován jako dáma), smí všemi směry, ale stejně jako pěšák pouze o jedno políčko. mezikdalu v klasické verzi král (dáma) může všemi směry a skočit libovolný počet polí.

## 1.1 pohyb

pěšák smí pouze o jedno políčko v před. král (dáma) smí o jedno políčko všemi směry. tah hráče končí po posunutí kamene.



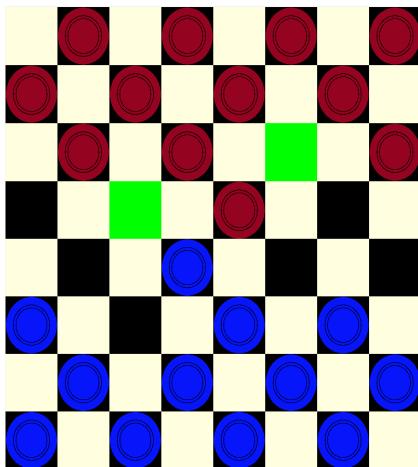
Obrázek 1.1: Zvýraznění tahu



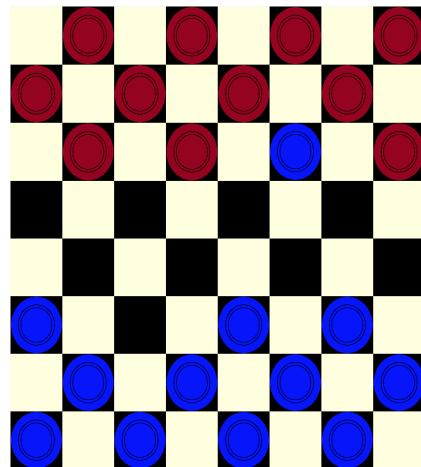
Obrázek 1.2: Po tahu

## 1.2 Braní kamenů protivníka

Pokud v cestě ku předu stojí kámen protivníka a je za ním prázdné políčko, tak jak král, tak pěšák přeskočí kámen protivníka na políčko za něj a kámen protivníka odstraní. Stále platí, že pěšák smí brát pouze ku před, zatímco král může brát všemi směry.



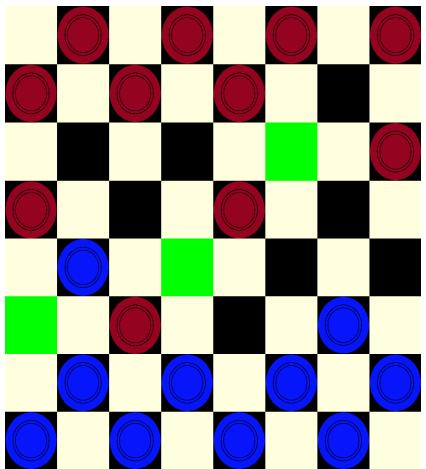
Obrázek 1.3: Sebrání protivníkova kamene



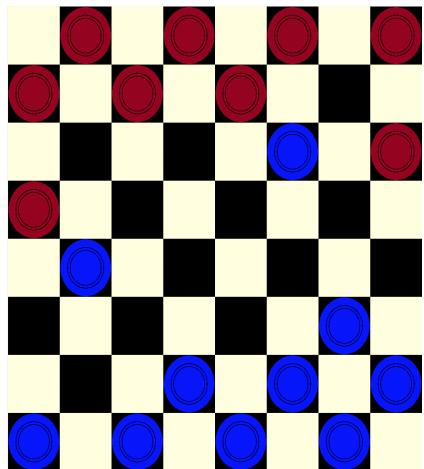
Obrázek 1.4: Po provedení tahu

## 1.3 Pohyb

Pokud po tom, co hráč vzal protivníkův kámen, je v cestě další protivníkův kámen, za kterým je volné místo, hráč táhne znovu a vezme tím tak 2 protivníkovy kameny v jednom tahu. Toto je možné opakovat dokud je splněna ta podmínka, že v blízkosti kamene je protivníkův kámen a za ním je volné políčko. Tedy hráč může vykonat další tah, pokud během dalšího tahu sebere kámen protivníka.



Obrázek 1.5: Řetězové braní dvou kamenů



Obrázek 1.6: Po provedení tahu

## 2 Popis protokolu

Protokol je posílán jako nešifrovaný text. Jednotlivé části jsou oddělené znakem '|'. Zpráva je ukončena znakem '\n'.

### 2.1 Obecný tvar zpráv

Obecný tvar zprávy je:

- ID\_hráče|INSTRUKCE|parametr|parametr...

Obecný tvar odpovědi serveru je:

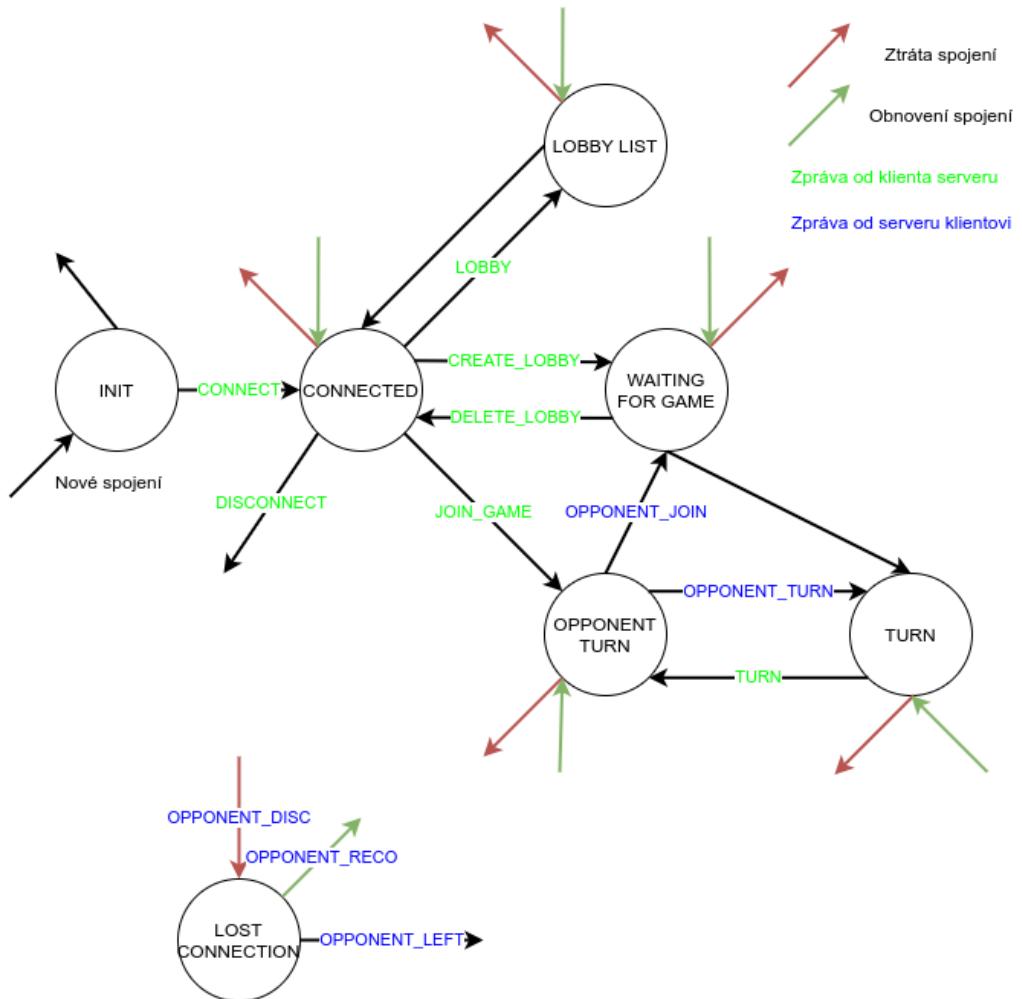
- ID\_hráče|VYSLEDEK|kód\_výsledku|zpráva|parametr|parametr...

VYSLEDEK nabývá hodnot "OK" a "ERROR", lze ho považovat za typ instrukce.

## 2.2 Tabulka instrukcí a stavový diagram protokolu

Instrukce	Popis
<b>CONNECT</b>	Požadavek na vytvoření hráče. Očekávána jako první instrukce komunikace. Server odpojí kterékoliv připojení, které jako první instrukci nepošle CONNECT
<b>LOBBY</b>	Požadavek pro získání seznamu hracích místností, které jsou dostupné k připojení.
<b>CREATE_LOBBY</b>	Požadavek na vytvoření nové herní místnosti.
<b>DELETE_LOBBY</b>	Požadavek o zrušení existující herní místnosti.
<b>JOIN_GAME</b>	Požadavek na připojení k zvolené herní místnosti.
<b>TURN</b>	Požadavek na provedení tahu.
<b>DISCONNECT</b>	Požadavek na odpojení spojení.
<b>PING</b>	Ping slouží k udržení stávající komunikace. Server na PING také odpovídá instrukcí PING.
<b>OPPONENT_JOIN</b>	Instrukce serveru informující protivníka hráče, který zaslal požadavek o připojení protihráče.
<b>OPPONENT_DISC</b>	Instrukce serveru informující protivníka hráče o protivníkově ztrátě připojení.
<b>OPPONENT_TURN</b>	Instrukce serveru informující protivníka hráče o tahu provedeném hráčem.
<b>OPPONENT_RECO</b>	Instrukce informující protivníka hráče o úspěšné znovu připojení hráče do hry.
<b>OPPONENT_LEFT</b>	Instrukce informující hráče o protivníkově permanentní ztrátě připojení a ukončení probíhající hry.

Tabulka 2.1: Tabulka instrukcí a popisů



Obrázek 2.1: Diagram komunikace protokolem

## 2.3 Popis instrukcí a odpovědí

### 2.3.1 Obecné negativní odpovědi serveru

- 400 - Sockets don't match
- 400 - Unknown connection
- 400 - Verification failed

- 401 - Instruction got too many parameters
- 401 - Unrecognized instruction
- 401 - TURN needs at least 2 parameters
- 401 - Too many parameters for TURN

### 2.3.2 Instrukce: CONNECT

#### Popis

Připojí hráče k serveru. Pokud první zpráva od nového připojení není CONNECT, tak server spojení odpojí. Při zadání ID hráče slouží k znovu připojení k serveru. Při znovu připojování k serveru, server posílá dodatečné informace nutné k obnovení klienta do stavu před ztrátou připojení.

#### Přesné formáty

- 0|CONNECT|username - hráč se připojuje poprvé
  - username - jméno hráče
- ID\_hráče|CONNECT|username - hráč se pokouší znovu připojit
  - username - jméno hráče

#### Kódy a zprávy odpovědí

- Pozitivní
  - 201 - Connection success
  - 202 - Reconnection success|state
  - 202 - Reconnection success|state|board|on\_top|opponent\_name
    - \* state - stav hráče na serveru (connected, turn, opponent\_turn)
    - \* board - hrací pole (pouze černé pole - délka 32 znaků)
    - \* on\_top - jestli má hráč červené/ modré kameny (0 = modré, 1 = červené)
    - \* opponent\_name - jméno protivníka

- Negativní
  - 402 - Username is empty
  - 403 - Username too long
  - 404 - Username already in use
  - 405 - Server failed to add player
  - 406 - Player with this ID doesn't exist
  - 407 - Is this an attack attempt
  - 408 - Failed to attach gameboard
  - 409 - Maximum number of connections reached
  - 410 - This socket is already connected

### 2.3.3 Instrukce: CREATE\_LOBBY

#### Popis

Vytvoří nové lobby a přidá hráče, který ho vytvořil, jako hráč 1. Hráč 1 má vždy bílé (v mé verzi modré) kameny.

#### Přesný formát

- ID\_hráče|CREATE\_LOBBY|lobby\_name
  - lobby\_name - jméno vytvářené místo

#### Kódy a zprávy odpovědí

- Pozitivní
  - 201 - Successfully created lobby
- Negativní
  - 402 - Server failed to create lobby
  - 403 - Lobby name is too long
  - 404 - Lobby name already exists
  - 405 - This cannot be done in current state

- 406 - Failed to add player to game
- 407 - Failed to add game

### 2.3.4 Instrukce: JOIN\_GAME

#### Popis

Připojí hráče do existující místnosti. Připojeného hráče nastaví jako hráč 2, který má černé (v mé verzi červené) kameny. Místnost, do které se hráč připojil je odstartována.

#### Přesný formát

- ID\_hráče|JOIN\_GAME|lobby\_name
  - lobby\_name - jméno místnosti pro připojení

#### Kódy a zprávy odpovědí

- Pozitivní
  - 201 - Successfully joined game
- Negativní
  - 402 - Lobby name is too long
  - 403 - This cannot be done in current state
  - 404 - Failed to find game lobby
  - 405 - Failed to contact opponent
  - 406 - Server lost game
  - 407 - Failed to add game

### 2.3.5 Instrukce: DELETE\_LOBBY

#### Popis

Zruší vytvořené lobby, pokud v něm je pouze zakladatel lobby (teoreticky nemůže existovat lobby s více hráči než 1, protože hra je automaticky odstartována po připojení druhého hráče).

### Přesný formát

- ID\_hráče|DELETE\_LOBBY

### Kódy a zprávy odpovědí

- Pozitivní
  - 201 - Lobby deleted
- Negativní
  - 402 - This cannot be done in current state
  - 402 - You don't have lobby
  - 402 - No game found

## 2.3.6 Instrukce: LOBBY

### Popis

Požadavek uživatele na získání místonosti dostupných k připojení.

### Přesný formát

- ID\_hráče|LOBBY

### Kódy a zprávy odpovědí

- Pozitivní
  - 201 - Available lobbies|lobby\_name\_1|...
- Negativní
  - 402 - Failed to fetch game
  - 403 - This cannot be done in current state

## 2.3.7 Instrukce: TURN

### Popis

Provede tah nebo sekvenci tahů. Kontaktuje protivníka o hráčovo tazích.

### Přesný formát

- ID\_hráče|TURN|parametr\_1|parametr\_2|...|parametr\_30
  - parametr\_1 - index (pozice) kamene
  - parametr\_x - index (pozice) cíle tahu - index nabývá hodnot 0-63

### Kódy a zprávy odpovědí

- Pozitivní
  - 202 - Turn Successful
  - 203 - You won!
  - 204 - You lost!
- Negativní
  - 402 - This cannot be done in current state
  - 403 - Failed to find game
  - 404 - It is not your turn
  - 405 - Need starting position
  - 406 - Too few parameters
  - 407 - Parameter isn't number
  - 408 - Failed to validate move
  - 409 - Failed to find opponent
  - 410 - Failed to contact opponent
  - 411 - Opponent message error

## 2.3.8 Instrukce: DISCONNECT

### Popis

Použito když se hráč odpojuje od serveru. Vymaže hráče ze seznamu hráčů. Zavře spojení z kterého přišel požadavek.

### Přesný formát

- ID\_hráče|DISCONNECT

## Kódy a zprávy odpovědí

- Pozitivní
  - 201 - You were disconnected
- Negativní
  - nemá negativní odpovědi. Vždy zavře připojení, z kterého přišel požadavek.

### 2.3.9 Instrukce: OPPONENT\_JOIN

#### Popis

Instrukce, kterou posílá server klientovi (protivníkovi hráče, který poslal instrukci JOIN\_GAME), informující ho o připojení protivníka.

#### Přesný formát

- ID\_protivníka|OPPONENT\_JOIN|kód|zpráva|username
  - kód - jedná se o zprávu serveru, takže klient očekává kód operace
  - zpráva - podobně jako kód je očekávána klientem
  - username - jméno hráče volající JOIN\_GAME

#### Kódy a zprávy

- Pozitivní
  - 201 - Successfully joined game
- Negativní
  - 401 - Server lost your lobby

### 2.3.10 Instrukce: OPPONENT\_TURN

#### Popis

Pošle protivníkovi klientův tah.

### Přesný formát

- ID\_hráče|OPPONENT\_TURN|kód|zpráva|parametr\_1|parametr\_2|...
  - kód, zpráva - viz předchozí
  - parametr\_x - stejně jako u TURN, index v intervalu 0-63

### Kódy a zprávy

- Pozitivní
  - 201 - Opponent moved
  - 203 - You won!
  - 204 - You lost!

## 2.3.11 Instrukce: OPPONENT\_DISC

### Popis

Informuje hráče o protivníkově odpojení.

### Přesný formát

- ID\_hráče|OPPONENT\_DISC|kód|zpráva
  - kód, zpráva - viz instrukce OPPONENT\_JOIN

### Kódy a zprávy

- Pozitivní
  - 201 - Opponent disconnected

## 2.3.12 Instrukce: OPPONENT\_RECO

### Popis

Informuje hráče o protivníkově znovu připojení.

### Přesný formát

- ID\_hráče|OPPONENT\_RECO|kód|zpráva
  - kód, zpráva - viz instrukce OPPONENT\_JOIN

### Kódy a zprávy

- Pozitivní
  - 201 - Opponent reconnected

## 2.3.13 Instrukce: OPPONENT\_LEFT

### Popis

Slouží k informování protivníka o permanentním ukončení protivníkovo spojení.

### Přesný formát

- ID\_hráče|OPPONENT\_RECO|kód|zpráva
  - kód, zpráva - viz instrukce OPPONENT\_JOIN

### Kódy a zprávy

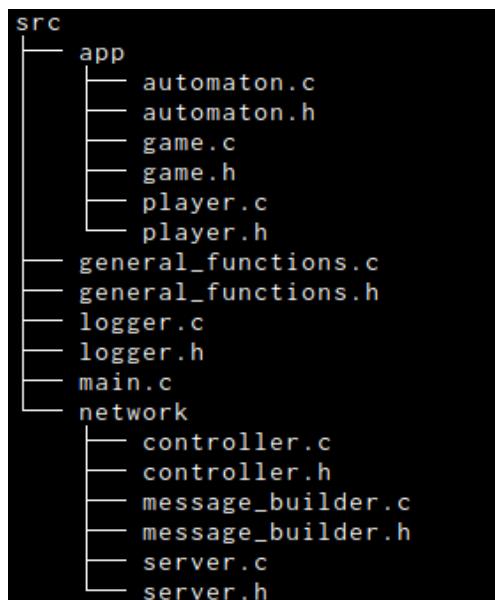
- Pozitivní
  - 201 - Terminating game. Opponent left.

# 3 Programátorská dokumentace

## 3.1 Server

Je napsán v jazyce C a překládán překladačem gcc pomocí poskytnutého makefile.

### 3.1.1 Struktura serveru



Obrázek 3.1: Struktura src adresáře serveru

#### Adresář app

V adresáři app se nacházejí moduly `automaton`, `game` a `player`, ve kterých se nachází struktury hry, hráče a automatu. Dále se zde nachází logika hry a obsluhující funkce.

## **general\_functions**

Tento modul poskytuje ”generickou” strukturu spojového seznamu. Vzhledem k tomu, že data jsou v tomto seznamu uložené pomocí void pointeru, umožňuje tento modul pouze přidávat prvky. Práci nad seznamy už si implementuje každý modul využívající seznam sám. V této práci jsou využity spojové seznamy pro hráče a hry.

## **logger**

Poskytuje funkci zápisu zprávy do souboru.

## **main**

Hlavní modul programu obstarávající spuštěcí parametry.

## **Adresář network**

Obsahuje moduly **server**, **controller**, **message\_builder**. Server se stará o spuštění serverového socketu a následných připojení a komunikaci klientů. Komunikaci klientů obsluhuje controller, který využívá message\_builder pro snadnější vytvoření odpovědi klientovi.

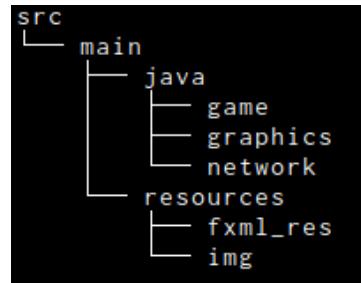
### **3.1.2 Obsluha spojení**

Jak již bylo zmíněno spojení/ připojení a komunikaci obsluhuje modul **server**. Obsluha více socketů je realizována použitím **fd\_set** a funkce **select**. Program tedy obsluhuje klienty postupně pouze jedním vláknem.

## **3.2 Klient**

Klient je psán v jazyce Java verze 11 s použitím GUI knihovny JavaFX, která do Java 1.8 byla součástí Java JDK. Ovšem pro verzi Java 11 už je JavaFX samostatnou knihovnou.

### 3.2.1 Struktura klienta



Obrázek 3.2: Adresářová struktura src adresáře klienta

V adresáři `java` se nachází podadresáře `game`, `graphics`, `network`. V adresáři `game` se nachází logika hry a funkce obsluhují komunikaci se serverem. Adresář `network` obsahuje třídy starající se o navázání spojení a posílání/přijímání dat a jejich parsování. Adresář `graphics` obsahuje grafické ovladače k souborům s příponou `.fxml`, které se nacházejí v `resources/fxml_res`.

# 4 Uživatelská dokumentace

## 4.1 Server

U serveru je předpokládáno, že bude spuštěn na platformě Linux. Server je konzolová aplikace. Překládá a spouští se z terminálu.

### 4.1.1 Překlad

Server se překládá pomocí poskytnutého **makefile**. V adresáři serveru, kde se nachází **makefile** stačí pouze zavolat příkaz **make** a zdrojové soubory jsou přeloženy překladačem **gcc** a vytvoří se spustitelný soubor **run\_server**.

### 4.1.2 Spuštění

Server se pouští voláním **run\_server**. Při spuštění je možné použít několik volitelných parametrů.

- -h - vypíše nápovědu k parametrům
- -a IP - pokusí se vytvořit a spustit socket s adresou IP.
- -p PORT - pokusí se vytvořit a spustit socket s portem PORT.
- -c NUM - povoluje maximálně NUM počet současných spojení na serveru.

Tyto parametry jsou na sobě nezávislé a mohou být použity v libovolném pořadí. Pokud některý parametr není použit, je použita jeho výchozí hodnota. Výchozí hodnoty:

- IP - 127.0.0.1
- PORT - 61116
- NUM - 50

## 4.2 Klient

Klienta je možné spustit na platformách Linux a Windows. Je psán v Java 11 a překládán pomocí gradle.

### 4.2.1 Překlad

Aplikace k přeložení používá gradle wrapper, který má závislost na Java. Aplikace je také psána v jazyce Java a proto je po uživateli požadováno nainstalovat JDK. Doporučuji nainstalovat **Oracle SE Development Kit 11.0.10**, který je dostupný pro Windows i Linux.

K aplikaci je poskytnutý gradle wrapper, který umožňuje téměř automatické přeložení aplikace. Uživateli stačí pouze v příkazovém řádku zavolat příkaz **gradlew build** v kořenovém adresáři klienta (kde se nachází soubor **gradlew** a **gradlew.bat**).

### 4.2.2 Spuštění

Po příkazu **gradlew build** se v kořenovém adresáři klienta vytvoří adresář **build**, v něm podadresář **libs** a v tomto podadresáři soubor **client.jar** (výsledná cesta tedy je **build/libs/client.jar**). Tento soubor je možné spustit z příkazové řádky pomocí **java -jar jméno\_souboru**. A nebo na systému Windows dvojitým poklepáním na tento soubor, za předpokladu, že systém má nastavenou systémovou proměnnou pro spuštění Java 11.

## 5 Závěr

Obě aplikace splňují hlavní body zadání a jsou stabilní. Není tedy možné je shodit nesprávnou komunikací. Ovšem stále jsou věci, které se dají vylepšit.

Například na straně serveru, jsou použité spojové seznamy, pro uchovávání probíhajících her a připojených hráčů. Tyto struktury, ale nejsou velmi efektivní při hledání jednoho konkrétního prvku v seznamu, což je akce, kterou server dělá velmi často. Není tedy od věci zvážit, jak by si výkonnostně vedla například struktura Binary Search Tree.

U klienta by bylo možné vylepšit celkový vzhled grafického rozhraní, který má výchozí vzhled Java FX komponent.