

ZÁPADOČESKÁ UNIVERZITA V PLZNI  
FAKULTA APLIKOVANÝCH VĚD

DOKUMENTACE K  
SEMESTRÁLNÍ PRÁCI  
KIV/UPS  
SÍŤOVÁ HRA "SEDMA"

Martin Červenka A14B0239P

13. prosince 2016

# Contents

<b>1</b>	<b>Zadání</b>	<b>2</b>
<b>2</b>	<b>Překlad systému</b>	<b>3</b>
2.1	Automatická instalace . . . . .	3
2.2	Překlad serveru . . . . .	3
2.3	Průběh klienta . . . . .	3
<b>3</b>	<b>Spuštění aplikace</b>	<b>4</b>
3.1	Spuštění pomocí skriptu . . . . .	4
3.2	Spuštění serveru . . . . .	4
3.3	Spuštění klienta . . . . .	5
<b>4</b>	<b>Protokol</b>	<b>6</b>
4.1	Žádosti . . . . .	6
4.2	Odpovědi . . . . .	7

## 1 Zadání

## 2 Překlad systému

Pro to, abychom mohli spustit systém je nejprve důležité systém přeložit. V následujících podsekcích je postup tohoto překladu.

### 2.1 Automatická instalace

Program lze automaticky nainstalovat pomocí jednoho skriptu. Tento skript se nachází v kořenové složce projektu a jmenuje se **run**. Nepotřebujeme ani mít celý projekt stažený, protože se stáhne sám automaticky. K stažení celého projektu včetně závislostí pak stačí spustit příkaz **run** s parametrem **i**:

- `cd cesta_do_slozky_s_projektem`
- `./run i`

Po dokončení příkazu **run** máme celý projekt stažený a přeložený, můžeme tedy přejít do sekce Spuštění aplikace.

### 2.2 Překlad serveru

V případě, že z nějakého důvodu nelze automatikou instalaci použít, nebo instalace selže, je možné projekt snadno přeložit ručně.

Pro překlad serverové části systému je potřeba mít nainstalovaný program **CMake** a program **make**. Překlad poté provedeme následující sekvencí příkazů:

- `cd cesta_do_slozky_s_projektem`
- `mkdir build`
- `cd build`
- `cmake ..`
- `make`

Ve složce **build** vznikne přeložený program s názvem **server**.

### 2.3 Překlad klienta

Na překlad klienta potřebujeme javu v minimální verzi 8. Překlad pak provedeme sekvencí příkazů:

- `cd cesta_do_slozky_s_projektem/client/src`
- `javac cards/client/*.java -Xdiags:verbose -Xlint:unchecked -d ../bin/`

## 3 Spuštění aplikace

Po překladu aplikace je nyní možné aplikaci spustit. Postup pro spuštění jednotlivých částí systému je uveden níže.

### 3.1 Spuštění pomocí skriptu

Je možné jak server tak i klienta spustit pomocí skriptu **run** (viz sekce Překlad systému). Server se spustí s parametrem **s**, klient s parametrem **c**.

Příklad spuštění klientské části:

- `cd cesta_do_slozky_s_projektem`
- `./run c`

### 3.2 Spuštění serveru

Po překladu se program nachází ve složce `build`, proto do této složky přejdeme a pak spustíme server. Defaultní port pro aplikaci je 43158:

- `cd cesta_do_slozky_s_projektem/build`
- `./server 43158`

### 3.3 Spuštění klienta

Přeložený klient se nachází ve složce **client/bin**. spuštění provedeme příkazy:

- `cd cesta_do_slozky_s_projektem_/client/bin`
- `java cards.client.Main`

Po provedení příkazů se zobrazí okno podobné jako na obr. 1. Pokud známe IP adresu a port serveru, můžeme se připojit. Server oěří, zda-li existuje hra, kde by daný hráč mohl hrát. Pokud ano, zobrazí se okno hry (viz obr. 3, jinak se zobrazí okno vytvoření hry (viz obr. 2).

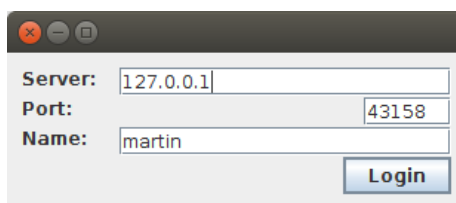


Figure 1: Přihlašovací okno

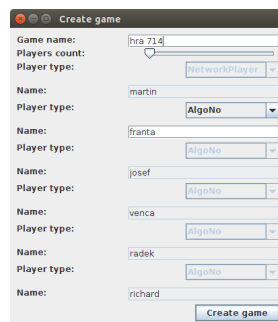


Figure 2: Okno vytvoření hry

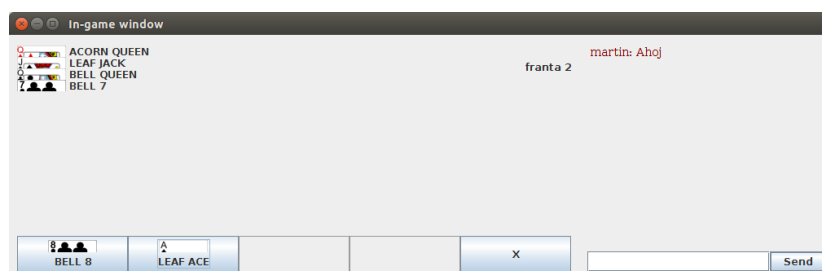


Figure 3: Herní okno

## 4 Protokol

Server a klient musejí používat protokol, aby se spolu domluvili. V této sekci popíšeme zprávy, které si mezi sebou posílají. V tabulce 4 se nachází společná část zpráv.

one_word	"a..zA..Z" "a..zA..Z"
card	("HEART"   "LEAF"   "BELL"   "ACORN" ) " " ("7"   "8"   "9"   "X"   "JACK"   "QUEEN"   "KING"   "ACE")
game_params	TODO

Figure 4: Tabulka obecných položek

### 4.1 Žádosti

V této sekci jsou vypsané zprávy, které odesílá klient na server. Nejprve jsou uvedeny bezparametrické zprávy, které očekávají odpověď od serveru.

Pokud se v pravé poloviční buňce tabulky nachází nějaký odkaz, tak to znamená, že se jedná o reakci na danou událost, která od serveru přišla (viz sekce Odpovědi). Jinak se jedná o asynchronní zprávy, které nejsou nijak vynucovány serverem.

disconnect	
"DISCONNECT"	
what_are_my_cards	
"WHAT ARE MY CARDS"	
what_are_card_counts	
"WHAT ARE CARD COUNTS"	
which_cards_are_on_table	
"WHICH CARDS ARE ON TABLE"	
login	
"LOGIN AS " one_word	
play	play
"PLAY " card	
create_game	
"CREATE GAME " oneword " " game_params	
send_message	
"SEND MESSAGE" oneword { " " oneword }	
vote	vote
"VOTE " ( "YES"   "NO" )	

Figure 5: Tabulka žádostí klienta k serveru

## 4.2 Odpovědi

V této sekci jsou uvedeny zprávy které přicházejí ke klientovi ze serveru. Jako v předchozí sekci pokud se v pravé poloviční buňce nachází nějaký text, jedná se o odkaz na zprávu, která přišla od klienta a vynucuje danou odpověď.

bad_command	jakákoliv akce
"BAD COMMAND" { " " oneword }	
card_played	
"CARD PLAYED" card one_word	
your_cards_are	what_are_my_cards
"YOUR CARDS ARE" [ " " card ] [ " " card ] [ " " card ] [ " " card ]	
game_exists	login
"GAME EXISTS"	
game_created	create_game
"GAME CREATED"	
player_exists	login
"THIS PLAYER IS ALREADY CONNECTED"	
card_counts	what_are_card_counts
"CARD COUNTS FOLLOWS" { " " one_word "0..4" }	
welcome	
"WELCOME"	
play	
"PLAY"	
message_from	send_message
"MESSAGE FROM " one_word " " one_word { " " one_word }	
bye	disconnect
"BYE"	
vote	
"VOTE FOR REPEAT"	
on_table_is	which_cards_are_on_table
"ON TABLE IS" { " " card }	

Figure 6: Tabulka odpovědí serveru klientovi