



**FAKULTA  
APLIKOVANÝCH VĚD**  
ZÁPADOČESKÉ  
UNIVERZITY  
V PLZNI

# Semestrální práce KIV/UPS

Hra Čim čong

Jan Pelikán

5.1.2022

## Obsah

|   |   |
|---|---|
| Hra Čim čong .....                          | 2 |
| Popis protokolu .....                       | 2 |
| Bližší popis některých zpráv: .....         | 4 |
| Příkazy pro připojení a odpojení hráče..... | 4 |
| Programátorská dokumentace.....             | 5 |
| Server .....                                | 5 |
| Klient.....                                 | 5 |
| Požadavky pro spuštění.....                 | 6 |
| Závěr .....                                 | 6 |

## Hra Čim čong

Hra nepotřebuje žádné speciální potřeby/náčiní pro hraní. Stačí mít dvě ruce a na obou mít palec. Hra spočívá v tom, že v každém kole, hráč, který je na tahu řekne „čim čong“ a číslo, hráči v ten moment zvednou žádný, jeden nebo oba své palce, podle toho, zda číslo, které řekl hráč na tahu odpovídá počtu zvednutých palců buďto kolo vyhrál nebo prohrál. Při výhře, si může ubrat jeden palec (jednu ruku) ze hry. Cílem hry je vyhrát dvě kola, tudíž nemít žádnou ruku ve hře.

Při mé implementaci jsem omezil hru pro 2 až 4 hráče. Kliknutím na palec se zvedá příslušný palec a je zde okno pro zadání čísla. Po každém kole je 5 sekund pauza, aby hráči stihli zvednout své palce.

## Popis protokolu

Rozdělovač zpráv je „\n“ (enter), rozdělovač parametrů ve zprávě je „|“ a bez mezer, ty jsem přidal kvůli přehlednosti

### Klient:

**Login:** (Pokud přijde pouze LOGIN, doplní se automaticky „Trouba“, při dlouhém jméně se uživatel odpojí)

LOGIN | uživatelem zadané jméno

**Guess:** (tip je v rozmezí 0 - maximální počet prstů, ověřeno v GUI, při zadání nečíselné hodnot = 0, při zadání větší se nic neděje)

GUESS | tip na počet zvednutých palců

**Fingers:** (prsty v rozmezí 0 – 2 (1) ověřeno v GUI, pokud nečíselná hodnota vrátím 0, pokud jiná nevalidní hodnota, ukončím klienta)

FINGERS | počet zvednutých palců (0-2)

Pokud se pošle požadavek LOGIN, hráč se přiřadí do fronty hráčů, co chtějí hrát.

### Server:

**Start:**

START | počet hráčů | Jména hráčů (2-4)

**Reconnect:** Hráči, co se vrací do hry pošle všechny údaje o hře, ostatním hráčům pošle jenom zprávu o tom, že se někdo napojil zpět do hry.

RECONNECT | jméno hráče, co se vrací

RECONNECT | počet hráčů | jméno hráče + počet zbývajících prstů | hráč, který je aktuálně na tahu

**GiveGuess:** (Pro všechny hráče, když přijde od hráče na tahu příkaz Guess, server pošle tento příkaz, aby mi uživatelé odeslali příkaz FINGERS)

GIVEGUESS

**EndOfRound:** (Pro všechny hráče) poslední dva parametry pro výpis, výsledek se řeší už na serveru

ENDOFROUND | jméno hráče co je další na tahu | true/false (podle toho, zda byl GUESS správný) | celkový počet zvednutých prstů | počet prstů, který bych v příkazu guess od hráče

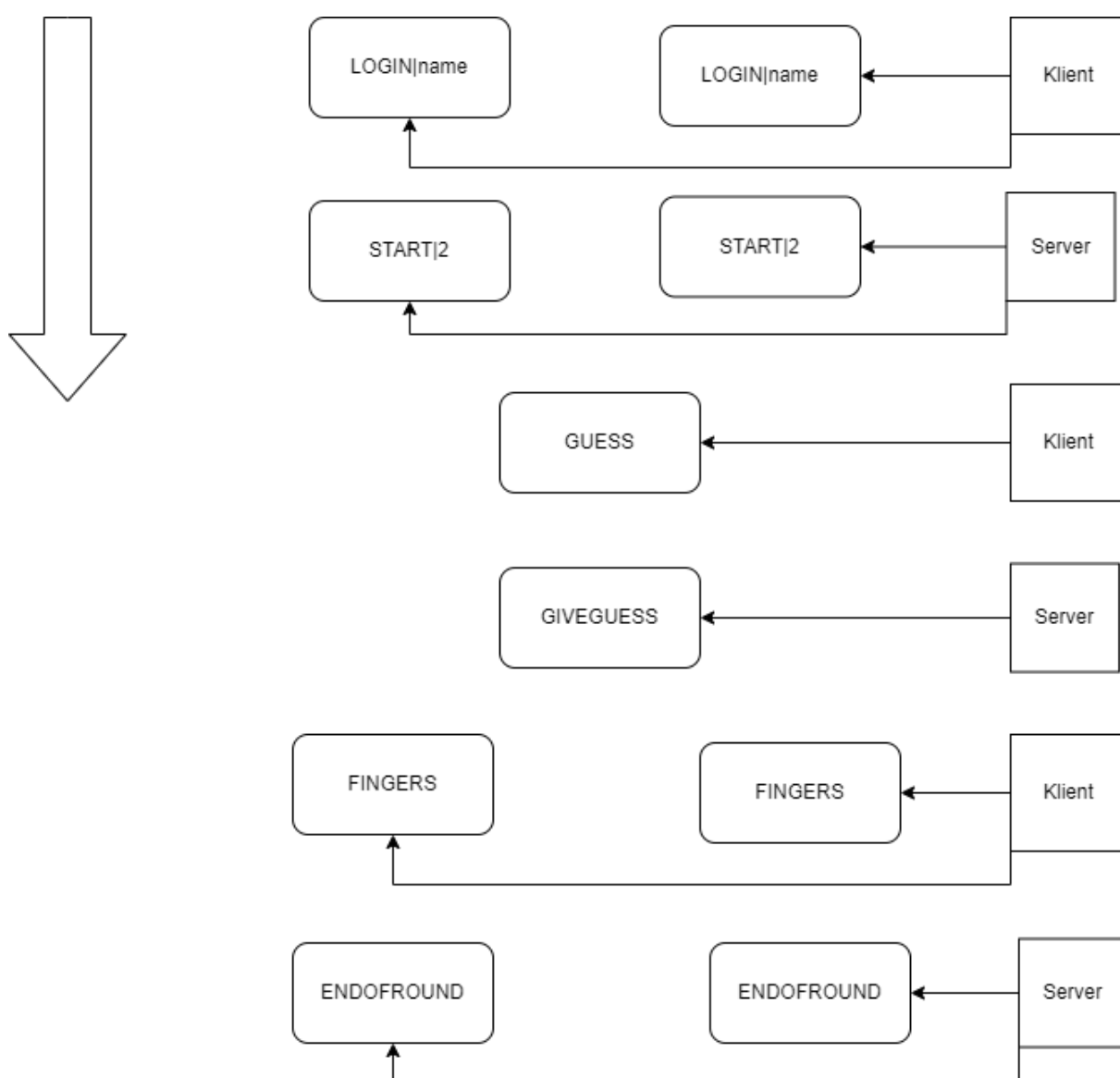
**EndGame:**

ENDGAME | hráč co vyhrál

**Logout:**

LOGOUT | jméno hráče, co se odpojil | jméno hráče, který je aktuálně na tahu

### Diagram průběhu hry



### Bližší popis některých zpráv:

Guess (klient) – tip na počet zvednutých palců je omezen na 0 - maximální aktuální počet prstů ve hře. Je to ošetřeno hned při zadávání hodnoty do pole v GUI, zároveň není umožněno zadat jinou hodnotu než číslici.

V případě že hra očekává od klienta příkaz GUESS, obratem pošle všem klientům příkaz GIVEGUESS. Pokud klient přijme tento příkaz, obratem na server pošle aktuální počet zvednutých palců (0-2). Jedná se o příkaz FINGERS. V moment, kdy dojdou od všech aktivních hráčů „FINGERS“. Server vyhodnotí aktuální kolo a v závislosti na stavu hry odešle buďto ENDOFROUND nebo ENDGAME.

EndOfRound (server) – Posílá všechny důležité informace po konci kola všem hráčům hry. První parametr je jméno hráče, který je další na tahu. Druhý parametr je výsledek kola, zda hráč správně / špatně tipnul výsledek a třetí a čtvrtý parametr jsou pro výpis statistiky hráčům. Počet prstů, kolik bylo zvednuto a počet prstů, kolik hráč tipnul že se zvedne.

Příkaz EndGame pouze oznamuje jméno hráče, který vyhrál aktuální hru. Rozpošle se všem hráčům.

### Příkazy pro připojení a odpojení hráče

Reconnect (server) – Tento příkaz má dva druhy. Jeden příkaz je pro aktivní hráče ve hře, a pouze jim posílá informaci o tom, že se hráč znovu připojil do hry. Tudiž posílá pouze jméno hráče, co se připojil. Druhý druh je pro hráče, který se chce připojit, musí dostat informaci o tom, jak daná hra vypadá v momentě, kdy se připojuje. První parametr říká, kolik je hráčů ve hře, druhý parametr spojuje jméno a zbývající počet prstů každého hráče, tudiž těch jmen je 2-4, podle prvního parametru. Poslední parametr je jméno hráče, který je aktuálně na tahu.

Logout (server) – Příkaz oznamuje všem hráčům hry, že se některý hráč odpojil, v prvním parametru posílá jméno hráče, co se odpojil. Druhý parametr je jméno hráče, který je aktuálně na tahu. Pro případ, že se odpojil hráč, co byl na tahu.

## Programátorská dokumentace

### Server

Server je programovaný v jazyce C. Je rozdělen do 5 zdrojových souborů (.c) a do 2 hlavičkových (.h) souborů.

Hlavičkový soubor game.h definuje struktury pro hru a hráče a popisuje jejich obsah, respektive datové typy. Hlavičkový soubor stateMachine.h popisuje matici stavů. Definuje všechny stavy a možné události, které ve hře nastanou.

Soubor queue.c se stará o implementaci fronty. Obsahuje metody jako přidání do fronty, odebrání z fronty a zjištění počtu prvků ve frontě. Další soubor stateMachine.c vytváří matici stavů, doplňuje možné přechody v matici.

Parser.c obsahuje funkce, pro vytváření a zpracovávání příkazů. Game.c obsahuje soubor funkcí pro obsluhu hry, odpojení hráče, vypočtení výsledku kola ....

Poslední a hlavní je soubor main.c. Ve funkci main se nejdříve zajistí, aby se server správně připojil na zadaný port. Poté v nekonečném cyklus iteruje přes všechny file deskriptory, pokud se na nějakém něco změnilo, zkontroluje, co se stalo a zachová se podle toho. Pokud přišla zpráva, zkontroluje, zda je zpráva validní a potom ji vykoná, pokud se uživatel odpojil nebo zpráva nedává smysl, server uživatele odpojí.

Pro přijímání zpráv, jsem využil metodu funkce select a ioctl. Funkce select udržuje file deskriptory a při nějaké změně aktivuje ioctl, které mi přiřadí do proměnné počet bajtů, které mi přišli. Pokud nepřišel žádný bajt, znamená to, že se klient odpojil. Vytvářím jedno nové vlákno v moment, kdy se do fronty připojí druhý hráč, toto vlákno má za úkol spát 25 vteřin a poté založit a spustit novou hru. Vlákno je potřeba, aby na server mohli chodit další požadavky.

### Klient

Klient s grafickým rozhraním je naprogramovaný v Javě za pomoci JavaFX knihovny pro grafiku.

Program je rozdělen do 8 tříd, z toho dvě třídy jsou enum. CommandsServer enum obsahuje příkazy serveru. CommandsKlient enum obsahuje příkazy klienta, implementují metodu toString().

Třída ImageLoader se stará o načítání obrázků hráčů, implementuje metodu pro načtení obrázku a následně nastaví správné velikosti všem obrázkům. Třída Gamer, vytváří objekt hráče, obsahuje jméno a počet prstů.

Třída Message má za úkol navázat spojení se serverem a potom číst zprávy ze serveru, respektive posílat zprávy na server.

Třída Intro vykresluje úvodní obrazovku pro nepřihlášeného uživatele. Čeká na to, než uživatel zadá své uživatelské jméno, při stisku tlačítka nejdříve klient naváže spojení se serverem, při úspěšném navázání spojení pošle na server příkaz LOGIN se svým uživatelským jménem, jinak ukončí klienta, s tím, že se nepodařilo navázat spojení. Pokud vše proběhne v pořádku, dostane správu ze serveru o úspěšném přihlášení do fronty. Pokud budou připraveni již 3 další hráči, hra okamžitě začne. Pokud ne, dostane zprávu s tím, že má čekat, než se přihlásí dostatek hráčů pro začátek hry. Začátek hry začne, když na klienta přijde příkaz START, potom klient vytvoří novou instanci třídy Game, podle parametrů, které přijdou v příkaze.

Třída Game představuje už herní okno pro hráče, okno je rozděleno na horní+středovou, dolní, pravou a levou část. V každé jedné části je jeden hráč, pokud hraje méně než 4 hráči, strany se vyplňují

postupně podle hráčů. Každý hráč má zobrazený počet prstů, které má aktuálně ve hře (1-2), vlastní obrázek (neměnitelný) a své jméno. Každý hráč má pod svým obrázkem ještě textové pole, kde se mu zobrazuje, kolik má aktuálně zvednutých prstů. Uprostřed je pole pro tipování zvednutých prstů, odeslat se dá pouze, pokud je na vás řada. Při každé události, která by mohla uživatele zajímat (konec kola, odpojení hráče, konec hry atd.) se zobrazí vyskakovací upozornění, které nastalou událost oznámí hráči. Na konci hry, se uživatel vrátí zpět do úvodní obrazovky a je vyzván, aby znovu zadal své jméno.

Třída Main obsahuje pouze metodu run, která je nutná pro spuštění JavaFX aplikace a kontrolu vstupních parametrů. Tato třída obsahuje metodu main, která spouští celou aplikaci.

## Požadavky pro spuštění

Verze Javy 11 a potřeba knihovna JavaFX. Kompilace, vytvoření .jar souboru probíhá pomocí Gradle. Na Windowsech „gradlew.bat jar“ a na Linuxu „./gradlew jar“. JavaFX je v .jar souboru, aby se dal spustit. Následné spuštění .jar souboru je příkazem „java -jar“. Aplikace má dva vstupní parametry, a to je adresa server, a port, kde server poslouchá.

Server je dodržuje verzi C99. Příkazem make, se vygeneruje spustitelný soubor „server“, který se spouští standardně a potřebuje jeden parametr, to je číslo portu.

## Závěr

Zadání jsem dle mého názoru splnil. Server i klient jsou dvě samostatně fungující aplikace. Navrhl jsem protokol, podle kterého obě aplikace komunikují mezi sebou. Splňují všechny kritické body zadání.

Možným vylepšením do budoucna je vylepšení grafického rozhraní, či možnost přidat do hry lobby.