



Hangman klient a server

Semestrální práce KIV/UPS

Mikuláš Mach
mimach@students.zcu.cz
A21B0202P

14. ledna 2024

Obsah

1	Princip hry	3
1.1	Obecná pravidla hry	3
1.2	Implementace pravidel hry	3
2	Komunikační protokol	4
2.1	Popis komunikačního protokolu	4
2.2	Klientovy zprávy	4
2.2.1	LOGIN	4
2.2.2	JOIN	5
2.2.3	START	5
2.2.4	QUESS	6
2.3	Zprávy od serveru	6
2.3.1	PING	6
2.3.2	LOBBYUPDATE	6
2.3.3	GAMEUPDATE	7
2.3.4	QUESSED	7
2.3.5	WINNER	7
2.4	Shrnutí zpráv	8
3	Implementace serveru	10
3.1	Úvod	10
3.2	Třída server	10
3.3	Třída ClientManager	10
3.4	Třída GameManager	10
3.5	Třída MessageHandlerer	11
3.6	Třída Game	11
3.7	Třída Client	11
3.8	Třída Responder	11
3.9	main.cpp	11
4	Implementace klienta	12
4.1	Úvod	12

4.2	Třída MyApp	12
4.3	Třída CommunicationHandlerer	12
4.4	Třída Game	12
5	Uživatelská příručka	13
5.1	Přeložení serveru	13
5.2	Spuštění serveru	13
5.3	Spuštění klienta	13
6	Závěr	14

Kapitola 1

Princip hry

1.1 Obecná pravidla hry

Hangman neboli šibenice je hra, ve které je úkolem uhodnout předem určené slovo. Hádá se po jednotlivých písmenech a při každém špatném pokusu o uhodnutí písmena je hráči odebrán život. Hráč vyhrává, když uhodne celé slovo před tím než přijde o všechny životy

1.2 Implementace pravidel hry

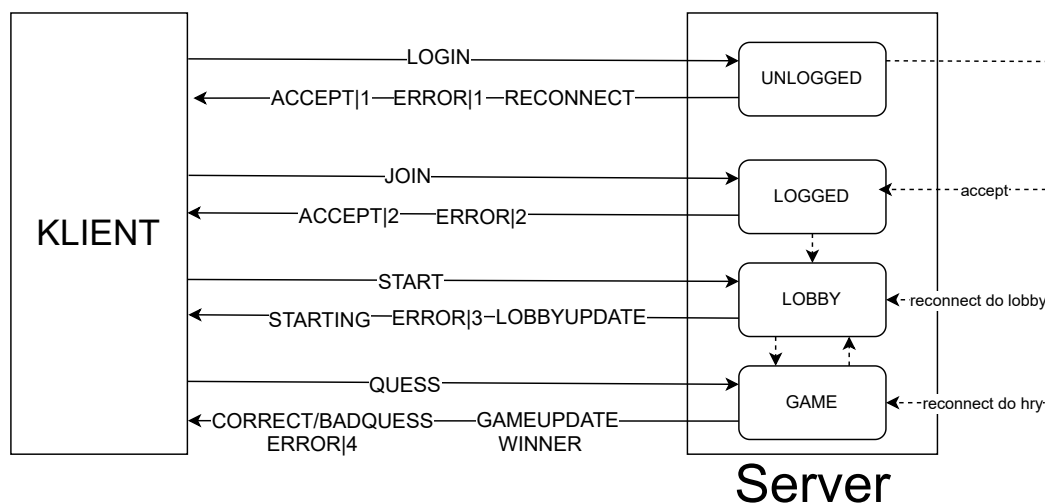
Hra je implementována tak, aby šla hrát ve 2 až 4 hráčích. Hráč vyhrává, když jako první uhádne poslední dosud neuhodnuté písmeno z hádaného slova nebo když zůstane jako poslední naživu.

Kapitola 2

Komunikační protokol

2.1 Popis komunikačního protokolu

Dle požadavků je protokol transparentní, tudíž nešifrovaný. Klient posílá vždy zprávu s daným formátem. Zpráva vždy začíná klíčovým slovem. Na každou zprávu od klienta existuje odpověď od serveru. Na obrázku 2.1 je ukázána komunikace klienta se serverem a jak se mění stav klientského objektu na serveru.



Obrázek 2.1: Diagram komunikace

2.2 Klientovy zprávy

2.2.1 LOGIN

- Formát: **LOGIN|<jméno>**

- Server tuto zprávu přijme pouze když objekt klienta na serveru je ve stavu UNLOGGED
- **Odpověď** při správném připojení: **ACCEPT|1|<ping interval>|<ping hranice>**
 - <ping interval> - v jakém intervalu bude server zasílat zprávu PING
 - <ping hranice> - po kolika neodpovězených zprávách PING bude klient úplně odpojen
- **Odpověď** při reconnectu: **RECONNECT|<stav>|<ping interval>|<ping hranice>**
 - <stav> - do jakého stavu se klient připojí, buď LOBBY nebo GAME
- **Odpověď** při odmítnutí: **ERROR|1**
- **Odpověď** při duplicitě jména: **ERROR|1|1**

2.2.2 JOIN

- Formát: **JOIN**
- Touto zprávou klient žádá server, aby byl připojen do prvního volného lobby
- Server tuto zprávu přijme pouze když objekt klienta na serveru je ve stavu LOGGED
- **Odpověď**: **ACCEPT|2**
- **Odpověď** při odmítnutí: **ERROR|2**

2.2.3 START

- Formát: **START**
- Touto zprávou klient žádá server o nastartování hry
- Server tuto zprávu přijme pouze když objekt klienta na serveru je ve stavu LOBBY a v daném lobby jsou aspoň 2 klienti
- **Odpověď**: **STARTING|<počáteční životy>|<délka slova>|<hráč na tahu>**
 - <počáteční životy> - životy klientů na počátku hry
 - <délka slova> - délka slova, které je v dané hře hádáno
 - <hráč na tahu> - jméno hráče, který je jako první na tahu
- **Odpověď** při odmítnutí: **ERROR|3**

2.2.4 QUESS

- Formát: **QUESS|<písmeno>**
- Klient se pokouší uhodnout jedno písmeno z hádaného slova
- Server tuto zprávu přijme pouze když objekt klienta na serveru je ve stavu ON_TURN
 - <písmeno> - jedno velké písmeno A-Z (bez háčeků a čárek)
- **Odpověď** při správném uhodnutí: **CORRECTQUESS|<písmeno>;<index>**
 - <index> - index uhodnutého písmena ve slově
 - jedno písmeno může mít více indexů, v tom případě budou indexy za sebou odděleny znakem ;
 - tato odpověď je zaslána všem v dané hře
- **Odpověď** při neuhodnutí: **BADQUESS|<jméno>;<životy>|<nový hráč na tahu>**
 - <jméno> - jméno hráče, který neúspěšně hádal
 - <životy> - aktuální stav životů po neuhodnutí
 - <nový hráč na tahu> - jméno hráče, který je jako další na tahu
 - tato odpověď je zaslána všem v dané hře
- **Odpověď** při odmítnutí: **ERROR|4**

2.3 Zprávy od serveru

2.3.1 PING

- Formát: **PING**
- Odpověď od klienta: **PONG**

2.3.2 LOBBYUPDATE

- Formát: **LOBBYUPDATE|<jméno>;<stav připojení>|.....**
 - <stav připojení> 0 - hráč je připojen, 1 - hráč je dočasně nedostupný
 - každá zpráva LOBBYUPDATE má 1 až 4 dvojice <jméno>;<stav připojení>, každá dvojice je oddělena pomocí |
 - tato zpráva je zaslána všem klientům, kteří se nacházejí v lobby, ve kterém došlo k nějaké změně (např. odpojení nebo připojení hráče)

2.3.3 GAMEUPDATE

- Formát: **GAMEUPDATE**|<jméno>;<počet životů>;<stav připojení>|...!<hráč na tahu>
 - <hráč na tahu> - jméno hráče, který je aktuálně na tahu
 - Stejný princip jako u zprávy LOBBYUPDATE, akorát tato zpráva navíc zasílá aktuální životy daného uživatele

2.3.4 QUESSED

- Formát: **QUESSED**|<délka slova>|<písmeno>;<index>;...|...
 - Tato zpráva je zaslána klientovi, který se po výpadku opět připojil do hry, aby dostal aktuální stav uhodnutých písmen v hádaném slově
 - Každé písmeno může mít více indexů a ve zprávě může být víc písmen
 - příklad: QUESSED|<délka slova>|<písmeno>;<index>;<index>|<písmeno>;<index>
 - písmena pouze A-Z (bez háček a čárek)

2.3.5 WINNER

- Formát: **WINNER**|<jméno>|<slovo>
 - Zpráva je zaslána všem v dané hře
 - zpráva obsahuje jméno vítěze a slovo, které bylo hádáno

2.4 Shrnutí zpráv

Klient posílá	Server odpovídá (jedna zpráva z bloku)
LOGIN <jméno>	ACCEPT 1 <ping interval> <ping hranice> RECONNECT LOBBY <ping interval> <ping hranice> RECONNECT GAME <ping interval> <ping hranice> ERROR 1 ERROR 1 1
JOIN	ACCEPT 2 ERROR 2
START	STARTING <počáteční životy> <délka slova> <hráč na tahu> ERROR 3
QUESS <písmeno>	CORRECTQUESS <písmeno>;<index>... BADQUESS <jméno>;<životy> <nový hráč na tahu> ERROR 4

Tabulka 2.1: Zprávy co posílá klient.

Server posílá	Klient odpovídá
PING	PONG
LOBBYUPDATE <jméno>;<stav připojení>...	
GAMEUPDATE <jméno>;<počet životů>;<stav připojení>...!<hráč na tahu>	
QUESSED <délka slova> <písmeno>;<index>... ..	
WINNER <jméno> <slovo>	

Tabulka 2.2: Zprávy co posílá server.

Formát	Význam
ERROR 1 1	zpráva LOGIN zaslána ve špatném klientském stavu
ERROR 1 2	špatný formát zprávy JOIN
ERROR 1 3	neplatné uživatelské jméno
ERROR 1 4	uživatelské jméno je již zabráno
ERROR 2 1	zpráva JOIN zaslána ve špatném klientském stavu
ERROR 2 2	špatný formát zprávy JOIN
ERROR 3 1	zpráva START zaslána ve špatném klientském stavu
ERROR 3 2	špatný formát zprávy START
ERROR 3 3	v lobby je málo klientů pro nastartování hry
ERROR 4 1	zpráva QUESS zaslána ve špatném klientském stavu
ERROR 4 2	špatný formát zprávy QUESS

Tabulka 2.3: Číselník errorů.

Kapitola 3

Implementace serveru

3.1 Úvod

Pro tvorbu aplikace serveru byl použit programovací jazyk C++, zejména kvůli podpoře již implementovaných datových struktur, jako třeba string, hash mapa, vector, queue, list, atd...

Obsluha více klientů byla implementována pomocí pseudoparalelizace použitím funkce **select()**. Díky tomuto postupu se program nemusí starat o kritické sekce, protože běží sekvenčně. Kostra komunikace byla použita z ukázky na cvičení.

Server je členěn do tříd.

3.2 Třída server

Tato třída zajišťuje spuštění serveru. Dále přijímá příchozí zprávy od klientů pomocí funkce **select()** a kontroluje zda jsou všichni klienti dostupní. Obsahuje metodu pro dočasné a úplné odpojení hráče.

3.3 Třída ClientManager

Tato třída má v sobě uložené všechny klienty aktuálně připojené k serveru. Obsahuje metody pro přihlášení uživatele, smazání uživatele a pro znovupřipojení uživatele.

3.4 Třída GameManager

Tato třída má v sobě uloženy všechny započaté i nezapočaté hry. Stará se o připojování hráčů do her, kontrolu jejich pokusů o hádání, aktualizaci stavu hry a o vyhodnocení vítěze.

3.5 Třída MessageHandler

Tato třída přebírá příchozí komunikaci od třídy server a vyhodnocuje zda je validní. Pokud ano, reaguje na ní volání metod z ostatních tříd.

3.6 Třída Game

Tato třída uchovává jací klienti jsou v dané hře a stav hry.

3.7 Třída Client

Uchovává všechny potřebné data o klientovy. Například jeho jméno, socket, jeho stav nebo hru ve které právě je.

3.8 Třída Responder

Pomocí metody pošle zprávu danému klientovi.

3.9 main.cpp

Obsahuje spuštění aplikace a ověření parametrů příkazové řádky

Kapitola 4

Implementace klienta

4.1 Úvod

Pro implementaci klienta byl zvolen programovací jazyk Python, zejména kvůli jeho jednoduchosti a kvůli tomu že je multiplatformní.

Pro implementaci grafického rozhraní byla zvolena knihovna Tkinter, která je součástí pythonu.

Klientská aplikace běží ve dvou vláknech. Jedno obsluhuje uživatelské rozhraní a druhé obsluhuje všechnu komunikaci se serverem. V komunikačním vláknu je použita funkce `select()`.

4.2 Třída MyApp

Obsahuje metody pro vytvoření jednotlivých scén, jejich zobrazení, přepínání a aktualizaci. Dále obsluhuje tlačítka a odesílá instrukce komunikačnímu vláknu.

4.3 Třída CommunicationHandler

Komunikuje se serverem, ověřuje validitu zpráv a kontroluje zda je server dostupný, pokud ne, pokouší se o znovupřipojení. Volá metody ze třídy MyApp pro aktualizaci stavu uživatelského rozhraní.

4.4 Třída Game

Uchovává informace o hře, ve které se klient nachází.

Kapitola 5

Uživatelská příručka

5.1 Přeložení serveru

Pro překlad je využit software **cmake**, **makefile** a **g++**. V kořenovém adresáři serveru použijeme příkaz **cmake**, který z CMakeLists.txt vytvoří **makefile**. Následně pomocí příkazu **make** vytvoříme spustitelný soubor `hangman_server`.

5.2 Spuštění serveru

Popisované spouštění je pro platformu linux. V adresáři kde máme přeložený spustitelný soubor použijeme v terminálu příkaz: `./hangman_server <ip_adresa> <port>`

- `<ip_adresa>` - ip adresa serveru
- `<port>` - port serveru

5.3 Spuštění klienta

V adresáři kde se nachází soubor **main.py** spustíme pomocí konzole/terminálu příkaz: **python3 main.py**, aby tento postup fungoval, musí být na daném počítači nainstalován python3

Kapitola 6

Závěr

Cílem práce bylo vytvořit síťovou hru pro více hráčů, která bude dostatečně stabilní a bude správně reagovat na výpadky. Toto zadání se podařilo naplnit v plném rozsahu a výsledkem je realizace hry šibenice. Aplikace nepadají, nezasekávají se a reagují na požadavky hráčů. Server je určen pro operační systém Linux a aplikace klienta byla vytvořena pro operační systémy Windows 10 a Linux.