

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Úvod do počítačových sítí KIV/UPS

Síťová hra Scrabble na protokolu TCP

Plzeň, 2017

David Pivovar, A13B0403P
pivovar@students.zcu.cz

Obsah

1	Zadání	2
2	Komunikační protokol	3
2.1	Komunikace server-klient	3
2.2	Server -> Klient	4
2.3	Klient -> Server	5
3	Implementace	6
3.1	Server	6
3.2	Klient	7
4	Uživatelská dokumentace	8
4.1	Server	8
4.1.1	Překlad a spuštění	8
4.1.2	Ovládání	8
4.2	Klient	9
4.2.1	Překlad a spuštění	9
4.2.2	Ovládání	9
5	Závěr	11

1. Zadání

Naprogramujte síťovou hru (server, klient) na protokolu TCP.

Úlohu naprogramujte v programovacím jazyku C/C++ anebo Java (C#). Pokud se jedná o úlohu server/klient, pak klient bude v Javě (C#) a server v C/C++. Komunikace bude realizována textovým nešifrovaným protokolem nad TCP protokolem. Výstupy serveru budou v alfanumerické podobě, klient může komunikovat i v grafice (není podmínkou). Server řešte pod operačním systémem Linux, klient může běžet pod OS Windows.

Realizujte konkurentní (paralelní) servery. Server musí být schopen obsluhovat požadavky více klientů souběžně. Bude možno hrát více her najednou a hru o větším počtu hráčů (dle pravidel hry). Klient se bude moci znovu připojit do přerušené hry.

Uživatelské rozhraní klienta bude nezávislé na zpoždění síťových operací. Aplikace se bude schopna vyrovnat s problémy na síti.

2. Komunikační protokol

Komunikační protokol pracuje na přenosu ASCII řetězců. Jednotlivé části jsou odděleny znakem ':' a hodnoty znakem ';'. Zpráva je zakončena '\n'.

2.1 Komunikace server-klient

	Server		Klient
Přihlášení		←	NICK
	NICKOK	→	
	NICKERR	→	
Znovu připojení do hry	RETURN	→	
		←	RETURN
		←	NEW
Zahájení hry	GAME	→	
	GAMER	→	
Tahy	TURN	→	
		←	TURN
	TURNOK	→	
	TURNERR	→	
	TURNP	→	
Přeposlání tahu			
Ukončení hry		←	END
Hráč opustil hru/znovu se připojil	DISC	→	
	RECN	→	
Ping		←	PING
	PING	→	
Chyba v přijaté zprávě		←	ERR
	ERR	→	

2.2 Server -> Klient

Klíčové slovo	Předpis	Význam
NICKOK	NICKOK	Nick je přijat
NICKERR	NICKERR:value	Nick je nevalidní: USE - nick je obsazen / CHAR - nick obsahuje nevalidní znaky
GAME	GAME:game_ID:pl_ID,nick; pl_ID,nick;...	Připojení do nové hry, info o hráčích
GAMER	GAMER:game_ID:pl_ID,nick, score;pl_ID,nick,score;...: x,y,char;x,y,char;...	Připojení do existující hry, info o hráčích, info o tazích
TURN	TURN	Tah hráče
TURNP	TURNP:pl_ID:score;x,y,char; x,y,char;...	Tah protihráče
TURNOK	TURNOK	Tah je přijat
TURNERR	TURNERR	Tah nebyl přijat
RETURN	RETURN	Hráč se může znovu připojit do opuštěné hry
DISC	DISC:pl_ID	Hráč opustil hru
RECN	RECN:pl_ID	Hráč se znovu připojil do hry
PING	PING	Odpověď na kontrolní zprávu od klienta
ERR	ERR:value	Chyba v přijaté zprávě

2.3 Klient -> Server

Klíčové slovo	Předpis	Význam
NICK	NICK:nick;n	Připojení se do hry, nick a počet hráčů
TURN	TURN:game_ID:score; x,y,char;x,y,char;...	Tah hráče
RETURN	RETURN	Vracení se do hry
NEW	NEW	Nová hra
END	END	Konec hry
PING	PING	Kontrolní zpráva, jestli je spojení se serverem
ERR	ERR:value	Chyba v přijaté zprávě

3. Implementace

3.1 Server

Server je implementován v jazyce C++.

O síťové spojení se stará třída *Network*, o správu her a přidělování hráčů pak třída *GameManager*. Datová struktura pro hru a její ovládání je ve třídě *Game*, datová struktura jednotlivých hráčů je třída *Player*.

Po spuštění aplikace se vytvoří BSD socket, nabinduje se na zvolený port (defaultně 1993) a spustí se vlákno, které naslouchá příchozím spojením. Také se spustí vlákno čekající na příkazy z konzole.

Po připojení hráče se spustí se vlákno, které naslouchá příchozím zprávám od klienta, a vlákno, které kontroluje zda je spojení s klientem aktivní. Zkontroluje se, zda přezdívka již není použita a v případě, že je volná, klient se notifikuje o přijetí a přidá se do seznamu hráčů pro hru o zvoleném počtu hráčů (2, 3, 4). Pokud je přezdívka obsazena nebo nevalidní, je o tom hráč informován a musí použít jinou. Pokud je v seznamu pro danou hru určené množství hráčů, spustí se hra.

Hráči se ve svých tazích střídají. Hráči, je odeslána notifikace, že je na tahu. Hra poté čeká, až od hráče přijme jeho tah. Ten zvaliduje a uloží do dvourozměrného pole. Hráči pošle notifikaci o přijetí (či nepřijetí) tahu a ostatním hráčům tah přepošle. Poté notifikuje dalšího hráče na tahu.

Při odpojení nebo znovu připojení některého z hráčů server notifikuje všechny hrající hráče. Odpojený klient se může znovu připoji a má na výběr, zda se znovu připojí do rozehrané hry nebo chce hrát novou hru. Při znovu připojení se mu odešle aktuální stav hry. Pro hraní hry musí být připojeni minimálně 2 hráči.

Hra se ukončí v okamžiku, kdy se odpojí všichni hráči (jejich přezdívky jsou pak volné). Pokud od klienta dorazí zpráva, která neodpovídá komunikačnímu protokolu, je spojení s klientem ukončeno. Zadáním příkazu *exit* se zavře socket a server se ukončí.

3.2 Klient

Klient je implementován v jazyce C#, uživatelské rozhraní je implementováno knihovnou WinForms.

Aplikace má 2 uživatelská okna, jedno pro připojení ke hře a druhé k ovládání hry. O komunikaci se serverem se stará třída *Network*. Datová struktura pro hru je třída *Game* a datová struktura hráče je třída *Player*.

Pro připojení musí být vyplněna IP adresa a port serveru, přezdívka hráče a zvolen počet hráčů. Poté se vytvoří socket a pokusí se připojit k severu. Následně se na server odešle přezdívka a počet hráčů, spustí se vlákno, které naslouchá příchozím zprávám ze serveru, a vlákno, které kontroluje zda je spojení se serverem aktivní posíláním zprávy *PING*.

Při připojení ke hře se hra a hráči inicializují (nová nebo rozehraná) a vytvoří se okno se hrou. Písmena do zásobníku jsou náhodně generována při startu hry a po odeslání tahu (samohlásky mají dvojnásobnou pravděpodobnost). Při umísťování písmene na hrací plán je kontrolováno, zda je možné ho na danou pozici umístit. Skóre se počítá dle pole s hodnotami jednotlivých písmen (1, 2, 3, 10), případně mohou být vynásobena bonusovou hodnotou hracího pole. Umístěná písmena lze zpětně odebrat z hracího pole zpět do zásobníku. Na server se odesílá řetězec se skóre a pozicemi umístěných písmen.

Při nedostupnosti serveru, odpojením/připojením hráče, obsazenou nebo nevalidní přezdívkou hráče anebo nevalidním tahem je uživatel informován zprávou v dialogovém okně. Pokud od serveru dorazí zpráva, která neodpovídá komunikačnímu protokolu, je spojení ukončeno.

4. Uživatelská dokumentace

4.1 Server

4.1.1 Překlad a spuštění

Linux

Překlad serveru se provede sérií příkazů *cmake .*, který vytvoří soubor Makefile, a příkaz *make* soubory zkompile. Pro vytvoření souboru Makefile je nutný mít nainstalovaný nástroj CMake 3.6.

Pokud není dostupný nástroj CMake, překlad se provede příkazem *g++ -std=c++11 -lpthread *.h *.cpp*

Server se spustí příkazem *./UPS_Scrabble_server [port]*. Program má jeden volitelný parametr, nastavení čísla portu v rozmezí 1025 - 65535. Defaultní port je 1993.

Windows

Překlad serveru se provede sérií příkazů *cmake .*, který vytvoří soubor Makefile, a příkaz *make* soubory zkompile. Pro vytvoření souboru Makefile je nutný mít nainstalovaný nástroj CMake 3.6.

Server se spustí příkazem *UPS_Scrabble_server.exe [port]*. Program má jeden volitelný parametr, nastavení čísla portu v rozmezí 1025 - 65535. Defaultní port je 1993.

4.1.2 Ovládání

Server se po spuštění řídí automaticky, ukončení se provede příkazem *exit*.

4.2 Klient

4.2.1 Překlad a spuštění

Linux

Pro překlad na operačním systému Linux je třeba mít nainstalovanou sadu nástrojů *mono* (viz <http://www.mono-project.com/>). Překlad se provede příkazem `mcs -out:UPS_Scrabble_client -pkg:dotnet -lib:/usr/lib/mono/2.0 *.cs`.

Spuštění se provede příkazem `mono UPS_Scrabble_client`.

Windows

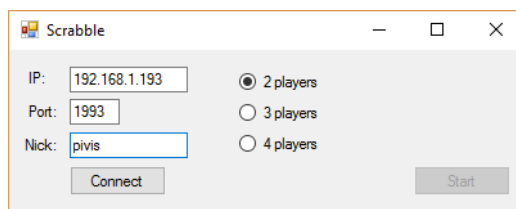
Pro překlad na operačním systému Windows je třeba mít nainstalovanou a nastavenou sadu *Microsoft .NET Framework SDK*. Překlad se provede pomocí kompilátoru `csc.exe` příkazem `csc /out:UPS_Scrabble_client.exe *.cs`.

Spuštění se provede příkazem `UPS_Scrabble_client.exe`.

4.2.2 Ovládání

Připojení do hry

V okně pro připojení (viz obr. 4.1) se zadá IP adresa serveru, port (defaultně 1993), přezdívka a vybere se počet hráčů. K serveru se připojí kliknutím na tlačítko *Connect*. Pokud je možný návrat do předchozí hry, zobrazí se dialog s výběrem připojení do předchozí hry nebo nové hry.



Obrázek 4.1: Přihlašovací okno

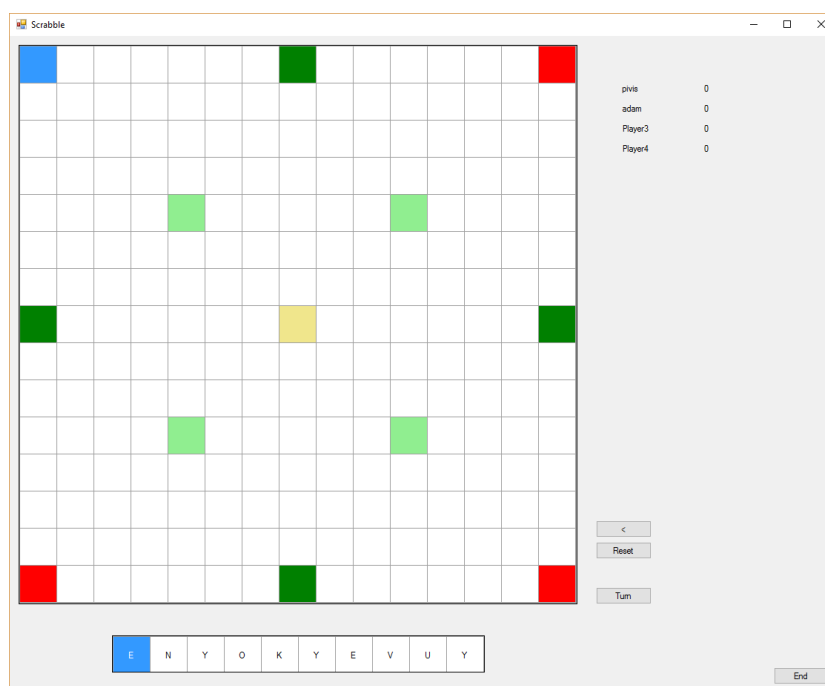
Po připojení se do hry se kliknutím na tlačítko *Start* zobrazí okno hry.

Odpojení se provede kliknutím na tlačítko *Disconnect*.

Ovládání hry

Na hrací plán 15x15 (viz obr. 4.2) se umísťují písmena ze zásobníku. První písmeno musí být umístěno na středové pole. Samohlásky mají bodovou hodnotu 1, souhlásky 2 nebo 3 a písmeno Q, X nebo W hodnotu 10. Na hracím plánu jsou pole, která hodnotu písmene zdvojí (světle zelené), ztrojí (tmavě zelená) nebo zčtyřnásobí (červené).

Po umístění všech písmen se tah ukončí kliknutím na tlačítko *Turn*. Chybně umístěná písmena se dají vrátit zpět do zásobníku. Tlačítko *<* vrátí poslední umístěné písmeno, tlačítko *Reset* se vrátí všechna písmena umístěná v daném tahu.



Obrázek 4.2: Okno hry

5. Závěr

Práce splňuje zadání. Server umí obsluhovat více klientů současně a umožňuje hrát více her najednou. Také umožňuje znovu připojení hráče do probíhající hry. Klient je nezávislý na zpoždění síťových operací. Server i klient jsou schopni vyrovnat se s chybami a výpadky na síti.

Server i klienta lze spustit na operačním systému Linux a Windows.