

Fakulta aplikovaných věd Západočeské univerzity v Plzni



Dokumentace semestrální práce předmětu KIV/UPS

Plzeň, 12. ledna 2023

Karel Matějovský

Obsah

1	Úvod	2
2	Popis průběhu hry	3
3	Protokol	4
3.1	Formát zpráv	4
3.2	Přenášené datové struktury, typy	4
3.3	Význam přenášených dat a kódů	4
3.4	Omezení vstupních hodnot a validace dat	7
3.5	Návaznost zpráv	7
3.6	Chybové stavy	9
4	Implementace	10
4.1	Server	10
4.1.1	Card	10
4.1.2	Client	10
4.1.3	ClientStates	10
4.1.4	Commands	10
4.1.5	Game	10
4.1.6	Message	10
4.1.7	Parser	10
4.1.8	Queue	10
4.1.9	Server	10
4.1.10	Signals	11
4.1.11	StateMachine	11
4.1.12	Stringop	11
4.2	Klient	11
4.2.1	NetworkDriver	11
4.2.2	UIDriver	11
4.3	Paralelismus	11
5	Požadavky na překlad	12
5.1	Server	12
5.2	Klient	12
5.3	Server	12
6	Závěr	13

1 Úvod

Obsahem této semestrální práce je vytvoření programů serveru a klienta hry Blackjack. Semestrální práce má za úkol seznámit se se základy komunikace v síti a technologiemi použitými pro přenos dat

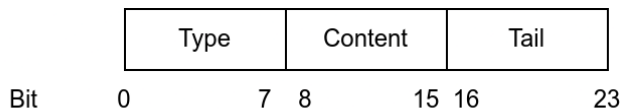
2 Popis průběhu hry

Cílem blackjacku je mít takovou kombinaci vyšší než krupiérův, ale zároveň ne vyšší než 21. Pokud je váš součet vyšší než 21, je to "bust", což znamená, že jste vypadli ze hry. Hru začínají všichni kromě krupiéra uzavřením sázky. Poté krupiér rozdá každému 2 karty. Karty 2 až 10 se hodnotí pomocí jejich bodů. Hodnota karty se počítá podle jejího líce, přičemž spodky, dámy a králové se všechny rovnají 10, esa mají hodnotu 1. Pokud jsou vaše dvě karty v součtu 21, vyhráváte jeden a půl násobek své sázky a pro toto kolo končíte. V opačném případě se vás krupiér zeptá, zda chcete další kartu z horní části balíčku. Pokud ano, zvolíte "hit". Počet karet, které můžete mít, není omezen, ale jakmile váš součet karet v ruce překročí hranici 21, jste vyřazeni a krupiér dostane vaši sázku. Pokud už žádné další karty nechcete, zvolíte "stay". Jakmile krupiér obejde stůl, pokud má hodnotu svých karet 16 nebo méně, musí si vzít další kartu. Pokud je na ní 17 nebo více, musí zůstat. Pokud krupiér překročí hodnotu 21, každý hráč, který je ještě v tomto kole, vyhrává dvojnásobek své sázky. Pokud však krupiér hranici nepřekročí, vyhrávají pouze hráči, jejichž kombinace jsou vyšší než hodnota karet krupiéra - vyhrávají dvojnásobek své sázky. Všichni ostatní svou původní sázku prohrají. Jakmile kolo skončí, všichni hráči vsadí novou sázku a začíná další kolo.

3 Protokol

3.1 Formát zpráv

Veškeré přenášené zprávy mají délku 24 bytů. Zpráva se dělí na tři části, každá o délce 8 bytů. První část 'Type' obsahuje typ zprávy, druhá a třetí část obsahují parametry zprávy. Pokud daná zpráva parametry neobsahuje, zůstávají prázdné. Aby bylo zabráněno chybám při obsahu částí kratší než 8 bytů, jsou veškeré prázdné části zprávy doplněny napravo od hodnoty dané části. Je-li např. v první části obsažena zpráva 'WIN', doplní se zbylé místo části mezerami tak, aby část obsahovala 'WIN_____'



Obrázek 1: Struktura zprávy protokolu

3.2 Přenášené datové struktury, typy

Zprávy jsou přenášeny jako řetězec znaků. Tyto znaky jsou dále interpretovány podle typu zprávy. Typ zprávy je vždy řetězec. Pokud je přenášen jiný datový typ než je řetězec, je převeden na řetězec a po přijetí převeden zpět. Toto se týká převážně číselných údajů.

3.3 Význam přenášených dat a kódů

Význam, směr a parametry přenášených kódů se nacházejí na následující stránce.

Odesílatel	Příkaz	Parametr	Očekávaná odpověď	Očekávaná odpověď
C	CONN		OK	Navázání spojení se serverem
C	SENDNCKN	Přezdívka hráče	LADDROOM <číslo místností>	Zaslání přezdívky hráče
S	SETLBY		OK	Přepnutí pohledu na lobby
S	SETGAME		OK	Přepnutí pohledu na hru
S	LADDROOM	Číslo místnosti	OK	Přidání místnosti do lobby
C	JOINROOM	Číslo místnosti	OK	Připojení se do místnosti
C	GAMERDY		SHBETUI	Klient je připraven ke hře
S	PLRRDY	Přezdívka hráče	OK	Zpráva o připraveném klientovi ostatním
S	PLRNRDY	Přezdívka hráče	OK	Zpráva o nepřipraveném klientovi ostatním
C	PLACEBET	Hodnota sázky	UPDBET, CLRCARDS, 2x DEALCARD <názvy karet>	Položení sázky
S	UPDCRDT	Hodnota kreditu	OK	Aktualizace kreditu klienta
S	UPDBET	Hodnota sázky	OK	Aktualizace sázky klienta
S	CLRCARDS		OK	Ostranění viditelných karet klienta
S	DEALCARD	Název karty	YOUTURN nebo PLRTURNS	Přidání karty klientovi
S	PAUSE		OK	Pozastavení hry
S	UNPAUSE		OK	Zotavení hry
C	HIT		OK	Akce "hit"
C	STAY		OK	Akce "stay"

Odesílatel	Příkaz	Parametr	Očekávaná odpověď	Očekávaná odpověď
S	LBYBTN		OK	Zobrazení tlačítka pro návrat do lobby
C	LEAVE		OK	Zobrazení tlačítka pro opuštění hry
S	SHBTNRDY		GAMERDY	Zobrazení tlačítka pro potvrzení připraveného hráče
S	SHBETUI		PLACEBET	Zobrazení nabídky sázení
S	BUST		OK	Zpráva o události "bust"
S	WIN		OK	Zpráva o výhře
S	LOSE		OK	Zpráva o prohře
S	HGRDLR		OK	Zpráva o částce vyšší než deale-rově
S	DLRBUSTS		OK	Zpráva o deale-rově přesažení hraniční hodnoty karet
S	YOUTURN		HIT / STAY	Zpráva, že je daný hráč na tahu
S	PLRTURNS	Přezdívka hráče	OK	Zpráva, že je jiný hráč na tahu
S	PLRDSCN	Přezdívka hráče	OK	Zpráva o odpo-jení jiného hráče
S	PLRRCN	Přezdívka hráče	OK	Zpráva o zno-vupřipojení jiného hráče
S/C	OK			Odpověď na zaslanou validní zprávu

3.4 Omezení vstupních hodnot a validace dat

Omezení vstupních hodnot se zde vyskytuje jen zřídka, jelikož se hra skládá především z jednoduchých tahů. Jsou však ošetřeny vstupy jako nevalidní číslo místnosti nebo sázení moc velké nebo naopak záporné částky. Je validován i převod z řetězce na čísla v případech, kdy je to zapotřebí.

3.5 Návaznost zpráv

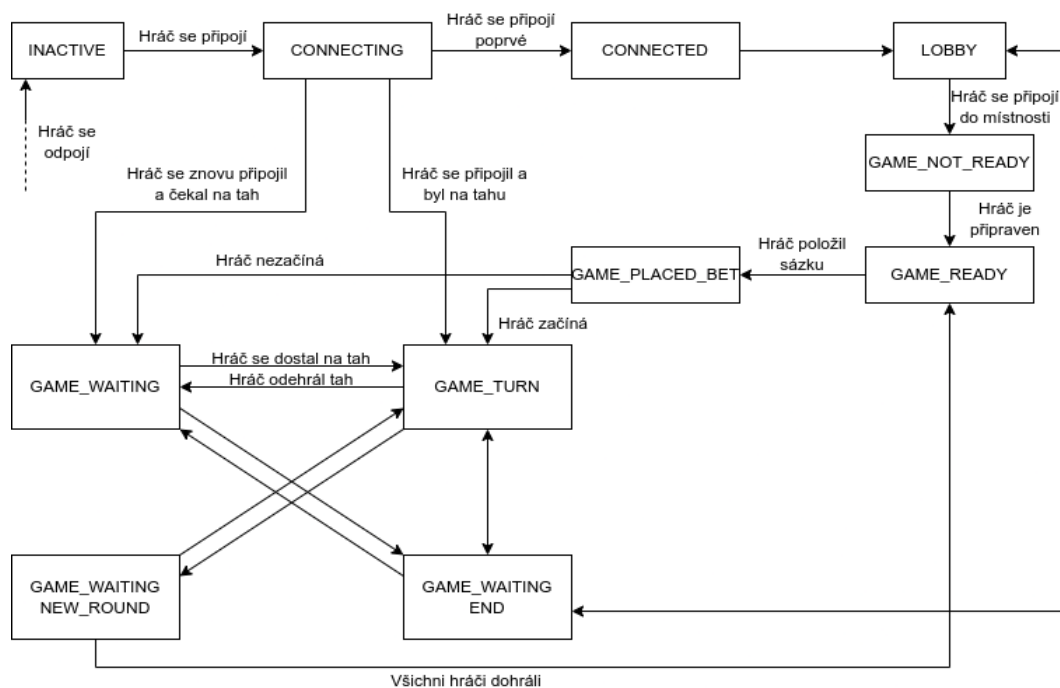
Ošetření návaznosti zpráv probíhá pomocí stavového diagramu. Hráči jsou stavy měněny za pomocí funkce 'changeState', která v závislosti na zadané tabulce vyhodnotí, zda-li je daný tah validní. Veškeré možné stavy klientů jsou uvedeny v následující tabulce

INACTIVE
CONNECTING
CONNECTED
LOBBY
GAME_NOT_READY
GAME_READY
GAME_PLACED_BET
GAME_WAITING
GAME_TURN
GAME_WAITING_NEW_ROUND
GAME_WAITING_END

Tabulka programu s přechody jednotlivých stavů vypadá následovně:

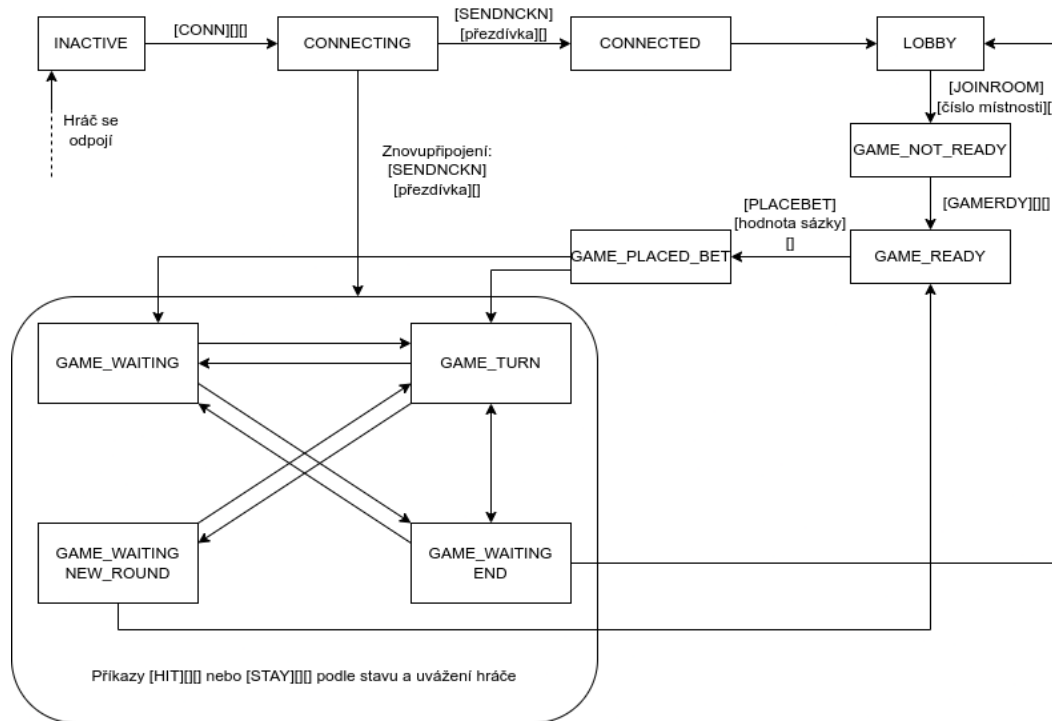
1	1	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	0	0	0
1	0	0	0	1	1	0	0	0	0	0
1	0	0	0	0	1	1	0	0	0	0
1	0	0	0	0	0	0	1	1	0	0
1	0	0	0	0	0	0	1	1	0	0
1	0	0	0	0	0	0	1	1	1	1
1	0	0	0	0	1	0	1	1	1	0
1	0	0	1	0	1	0	0	0	0	1

Následující obrázek ilustruje přechody stavového diagramu klienta. Přechody ze všech stavů do stavu INACTIVE nejsou zakresleny kvůli přehlednosti.



Obrázek 2: Stavový diagram klienta

Následující obrázek ilustruje přechody stavového diagramu klienta stejně jako předchozí, pouze značí, které příkazy převádějí diagram do kterých stavů. Některé stavy jsou měněny programem serveru a hráč jejich změnu neovlivní. Přechody ze všech stavů do stavu INACTIVE nejsou zakresleny kvůli přehlednosti stejně tak jako přechody činnostmi HIT a STAY mezi čtyřmi stavy rozehrané hry.



Obrázek 3: Stavový diagram klienta s kódy pro přechod

3.6 Chybové stavy

Každá ze zpráv je zpracována tak, aby při správném průchodu vracela hodnotu 0. Pokud při zpracování nastane jakákoli chyba, vrátí příkaz zpracování nenulovou hodnotu podle čísla chyby. Tyto chybové stavy jsou zaznamenávány pro každého klienta a pokud počet chybných zpráv přesáhne danou hranici, hráč bude automaticky odpojen.

4 Implementace

4.1 Server

Server je implementován v programovacím jazyce C. Program je rozdělen do následujících modulů:

4.1.1 Card

Tento modul obsahuje strukturu karty, která se používá ve hře. Uchovává hodnoty názvu karty a její hodnoty.

4.1.2 Client

Tento modul obsahuje strukturu klienta a odpojeného klienta. Obě struktury jsou si velmi podobné, odpojený klient pouze nezahrnuje komunikační buffer a socket. Dále modul obsahuje funkce pro práci s klientem.

4.1.3 ClientStates

Tento modul obsahuje pouze výčet jednotlivých stavů klienta.

4.1.4 Commands

Tento modul obsahuje výčet příkazů, jejich název v protokolu a funkce, které tyto příkazy posílají danému klientovi.

4.1.5 Game

Tento modul je jeden z hlavních modulů programu. Obsahuje veškeré funkce pro manipulaci se všemi hrami a hráči.

4.1.6 Message

Tento modul obsahuje strukturu zprávy, struktura obsahuje pouze řetězec o délce zprávy protokolu.

4.1.7 Parser

Tento modul obsluhuje přijaté zprávy a volá příslušné funkce programu.

4.1.8 Queue

Tento modul obsahuje implementaci fronty. Fronta byla v programu použita, aby se zamezilo ztrátě zpráv při použití příkazu `select()`.

4.1.9 Server

Tento modul obsahuje funkce pro čtení a posílání dat, přečtená data dále předává modulu Parser.

4.1.10 Signals

Tento modul obsahuje obsluhu signálů programu, aby bylo možné server ukončit klávesovou zkratkou, nebo předáním signálu.

4.1.11 StateMachine

Tento modul obsahuje funkci pro obsluhu stavů klientů

4.1.12 Stringop

Tento modul obsahuje funkce pro práci s řetězci.

4.2 Klient

K vytvoření programu klienta byl využit herní engine Unity. Veškeré skriptování je implementováno v jazyce C#. Implementace klienta je rozdělena do dvou hlavních tříd.

4.2.1 NetworkDriver

Tato třída obsluhuje síťový provoz mezi klientem a serverem. Poslouchání příchozích dat běží v odděleném vlákně, což vytváří problém při práci s UI klienta. Unity není ve většině případů thread-safe, proto přijaté zprávy po kontrole předává třída NetworkDriver třídě UIDriver.

4.2.2 UIDriver

Tato třída se stará o veškerou manipulaci s UI. Zpracovává přijaté příkazy, které obdrží od třídy NetworkDriver ve frontě příkazů. Ty pak postupně zpracovává v hlavním vlákně programu

4.3 Paralelismus

Pro paralelní zpracování na straně serveru je použit příkaz `select()`. Ten umožňuje v jednom vlákně spravovat více připojení. Veškeré příchozí zprávy jsou zapsány do fronty, odkud jsou pak postupně vybírány a zpracovávány. Každá instance hry na serveru je uchovávána v jedné struktuře, její stav se mění v závislosti na příchozích zprávách. Každý klient má zapsanou hru, ve které hraje, podle té lze identifikovat, která instance hry bude ovlivněna.

5 Požadavky na překlad

5.1 Server

5.2 Klient

Spustitelný soubor klienta je zapotřebí sestavit v herním enginu Unity verze 2021.3 a výše. Spustitelný soubor se vygeneruje pomocí menu File-Build and run. Tato činnost sestaví aktuální verzi programu a spustí jej. Případné další možnosti se dají nastavit ve File-Build settings.

5.3 Server

K sestavení serveru je zapotřebí nástroje make a kompilátoru gcc verze 12.2.0 (GCC). Sestavení programu je provedeno zavoláním příkazu make v adresáři programu (Na úrovni složky src). Po zavolání příkazu make se v podsložce bin vytvoří spustitelný soubor serveru.

6 Závěr

Nejvíce byla tato práce přínosná pro zdokonalení a hlubšímu pochopení programování v jazyce C, což považuji za velmi přínosné. Dále tato práce zdokonalila mé chápání síťového provozu a práce se sockety. Bylo pouze velmi stresující snažit se ošetřit veškeré chybné stavy tak, aby program běžel bez problémů, jelikož počet takovýchto stavů není nejmenší.