

Semestrální práce z Úvodu do počítačových sítí

Jan Novák
A11B0442
nowitz@students.zcu.cz



Fakulta aplikovaných věd Západočeské univerzity v Plzni

1) Zadání

Úkolem mé samostatné práce je naprogramovat hru Dots v programovacím jazyce Java a server v programovacím jazyce C s komunikačním protokolem TCP.

Jde o hru, ve které se hráči střídají v umístování “stěn”, přičemž “stěny” lze vytvářet pouze mezi sousedícími body, které neleží vůči sobě úhlopříčně. Když se hráči podaří uzavřít nějaký čtverec, tedy umístit čtvrtou stěnu, dostává bod. Hra končí, když jsou uzavřeny všechny čtverce, respektive umístěny všechny zdi. Ze začátku hráč umísťujete zdi pouze do volného prostoru, ale jakmile se prostor dostatečně zaplní, musí si dávat pozor, neboť jedna jediná chyba může hráči prohrát celou hru.

ZÁSADY VYPRACOVÁNÍ SEMESTRÁLNÍ PRÁCE:

- Úlohu naprogramujte v programovacím jazyku C anebo Java. Pokud se jedná o úlohu server/klient, pak klient bude v Javě a server v C.
- 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 XP. Emulátory typu Cygwin nebudou podporovány.
- Realizujte **bezestavové** a **konkurentní (paralelní)** servery. Server musí být schopen obsluhovat požadavky více klientů souběžně.
- V případě použití nespojovaných služeb (UDP) vyřešte na úrovni aplikačního protokolu problematiku ztráty, příp. duplicity dat (např. číslování, metoda okénka, apod.).
- Součástí programu bude trasování komunikace, dovolující zachytit proces komunikace na úrovni aplikačního protokolu a zápis trasování do souboru.
- Každý program bude doplněn o zpracování statistických údajů (přenesený počet bytů, přenesený počet zpráv, počet navázaných spojení, počet přenosů zrušených pro chybu, doba běhu apod.).
- Zdrojové kódy organizujte tak, aby od sebe byly odděleny části volání komunikačních funkcí, které jste vytvořili na základě zadání, od částí určených k demonstraci funkčnosti vašeho řešení (grafické rozhraní).

2) Řešení

a) Specifikace

Program jsem si specifikoval tak, že klienti přihlašující se po sobě hrají vždy spolu. To znamená, že hraje vždy první hráč s druhým, třetí se čtvrtým, atd. Dále, že je zbytečné si uchovávat přezdívky jednotlivých hráčů, protože každému hráči je vždy přidělena jedna barva, díky které je zřejmé, které tahy udělal jednotlivý hráč. V neposlední řadě jsem se rozhodl, že při výpadku klienta je hra ukončena bez možnosti obnovy stavu hry. U tohoto zadání není potřebná další specifikace. Vše je napsané v úvodu dokumentu v zadání.

b) Analýza

U této úlohy je důležité rozhodnout, jakým způsobem se bude hra ovládat. Zda použít například souřadnicový systém, díky kterému by uživatel zadával hodnoty například ve tvaru 1:1, 1:2. Což by znamenalo, že chce spojit bod na souřadnici $X=1$ $Y=1$ s bodem na souřadnici $X=1$ $Y=2$. Tento způsob je velice nepraktický. Ideálním řešením by byla možnost vybírat body pomocí kliku myši na konkrétní bod.

Dále je důležité rozhodnout, jakým způsobem bude docházet k vyhodnocování čar a čtverců? Zda je možné přidat čáru, zda již přidané čáry tvoří čtverec. Pro ověření čtverců se například nabízí možnost mít předem vytvořené všechny čtverce a při přidání nové čáry ověřit, zda již vytvořené čáry nevytvářejí některý z předdefinovaných čtverců. Nebo také možnost, že po přidání nové čáry budeme zkoumat, zda z těchto bodů, které tvoří tuto čáru, existují další dvě čáry, které jsou vůči ní kolmé. Pokud ano, tak jestli koncové body těchto kolmých čar tvoří čáru, která je rovnoběžná vůči přidané čáře.

c) Implementace

IMPLEMENTACE SERVERU:

Pro správný chod serveru jsou naimplementovány tři datové struktury. Struktura *Hra_s* slouží pro uchování klientských socketů pro hru. Struktura *Cara* slouží pro uchování informací o jednotlivých čarách mezi dvěma body. Každý z nich je definován souřadnicí X a Y . Struktura *Ctverec* je složená ze čtyř čar, dále nese informaci o existenci, barvě a id hodnoty čtverce.

Po spuštění *server.c* dojde k inicializaci proměnných, které jsou potřebné pro správný chod programu. Následně se spustí vlákno, které slouží pro naslouchání na serveru (posílání příkazů pro server). Poté se vytvoří server socket, který se sváže s adresou a dojde k samotnému naslouchání serveru, který čeká na připojení klientů. Po připojení klienta dojde ke spuštění vlákna, které zajistí, že pokud se nepřipojí soupeř do 10 sekund, tak dojde k jeho odpojení. Po každém připojení dvou klientů do 10 sekund dochází ke spuštění vlákna pro samotný běh hry, do kterého jsou předány klientské sokety.

V samotné funkci vlákna dojde nejprve k inicializaci proměnných, které jsou potřebné pro správný chod funkce. Poté dochází k vyzvání klienta, aby odehrál svůj tah. Během této

doby je spuštěno vlákno i pro druhého klienta, které hlídá, zda se klient neodpojil. Přijatá zpráva je zkontrolována, zda se nejedná o nesmyslné hodnoty a následně dochází k rozparsování této zprávy do pole, které je předáno funkci, která ověří, zda je možné požadovanou čáru vytvořit. Tato informace je poslána zpět klientům. Pokud byla čára přidána, dochází k volání funkce, která ověří, zda již nebyl vytvořen některý ze čtverců. Opět tato informace je předána zpět klientům. Následně dochází k vyzvání druhého hráče, aby odehrál svůj tah. Pokud se nepovedlo čáru vytvořit, nebo již existuje, je tato informace poslána zpět klientovi, který je znovu vyzván, aby odehrál svůj tah.

Během celého běhu funkce hry, dochází k zapisování informací o počtu přenesených bytů do souboru.

IMPLEMENTACE KLIENTA:

Program klienta je rozdělen do několika základních tříd.

Třída **Pozice** slouží pro nastavení polohy jednotlivých bodů.

Třída **Bod** slouží pro vytvoření objektů svého typu a následně jejich uchovávání v ArrayListu. Dále tato třída v sobě obsahuje metodu pro vygenerování všech potřebných bodů a jejich správné rozmístění.

Třída **Cara** slouží pro vytváření objektů svého typu a uchovávání jednotlivých čar v ArrayListu. Struktura čáry je definována pomocí barvy čáry a dvou bodů, přičemž každý bod má souřadnici X, Y.

Třída **Ctverec** také opět slouží pro vytváření objektů svého typu a uchovávání jednotlivých čtverců v ArrayListu. Její struktura je definována pomocí barvy čtverce a čtyř čar, které využívají třídu **Cara**.

Třída **MujProgram** slouží pro vytvoření hlavního okna programu a zajišťuje chod celého programu.

Třída **MujPanelHl** slouží pro vytvoření herního okna, ve kterém jsou umístěny dva obrázky, které znázorňují souřadnice jednotlivých bodů.

Třída **MujPanel** slouží už k samotnému vytvoření herní plochy, ve které dochází k vykreslení jednotlivých hodnot z ArrayListu jednotlivých tříd **Bod**, **Cara**, **Ctverec**.

Třída **MojeMenu**, slouží pro vytvoření panelu, ve kterém se vypisují hlášky programu a dále také jednotlivé body hráčů.

Třída **Spojeni** slouží pro navázání komunikace mezi serverem a klientem.

Třída **Prijem** slouží pro vyhodnocování přijatých zpráv od serveru, ve které je i metoda pro výpočet bodů jednotlivých klientů.

Po spuštění souboru *MujProgram.java* dojde k vytvoření a zobrazení klientské aplikace, následně k navázání spojení se serverem a spuštění vlákna pro příjem zpráv od serveru. Pokud při navazování spojení dojde k chybě, klient je automaticky odpojen. Po připojení dvou hráčů, dochází ke spuštění hry, přičemž jeden z klientů je vyzván, aby odehrál svůj tah. Pro odehrání tahu je naimplementován `MouseListener`, který nám zajistí výběr bodů po kliknutí myši na konkrétní body, které chceme spojit. Následně je tato zpráva odeslána serveru, který ji vyhodnotí a tuto informaci pošle zpět klientům.

Vždy po přidání čáry či čtverce dochází k překreslování panelu, aby došlo k jejich zobrazení.

FORMÁTY VYMĚŇOVANÝCH ZPRÁV:

Zprava se posílá jako pole znaků, přičemž jednotlivé položky jsou oddělované znakem ;(středník). Počet položek zprávy je proměnný podle typu zprávy.

Pokud se jedná o zprávu ze serveru pro klienta, tak vždy je první znak zprávy brán jako řídicí, díky kterému klient ví, jakou funkci má vykonat. Následující položky zprávy jsou jednotlivé hodnoty, díky kterým může klient vykonat požadovanou funkci.

Pokud se jedná o zprávu z klienta na server, tak dochází k poslání souřadnic bodů, které chceme spojit, například: bod na souřadnici X=1,Y=1 s bodem na souřadnici X=2, Y=1.

Jednotlivé typy zpráv jsou popsány níže.

Č.	Odkud > Kam	Popis zprávy	Formát zprávy
1	Klient > Server	Žádost o spojení dvou bodů, na konkrétních pozicích.	"X;Y;X;Y;"
2	Klient > Server	Žádost o ukončení hry.	"k;"
3	Server > Klient	Žádost o přidání čáry na konkrétních souřadnicích.	"c;X;Y;X;Y;"
4	Server > Klient	Zpráva, že čáru nelze přidat.	"f;"
5	Server > Klient	Zpráva, že čára již existuje.	"d;"
6	Server > Klient	Žádost o přidání čtverce na konkrétních souřadnicích, s určitou barvou.	"s;1.vodorovná čára;2.vodorovná čára;1.svislá čára;2.svislá čára;barva;"
7	Server > Klient	Žádost o přidání dvou čtverců na konkrétních souřadnicích, s určitou barvou.	"h;1.vodorovná čára;2.vodorovná čára;1.svislá čára;2.svislá čára;barva;1.vodorovná čára;2.vodorovná čára;1.svislá čára;2.svislá čára;"
8	Server > Klient	Zpráva klientovy, že vypadl jeho soupeř.	"v;"
9	Server > Klient	Zpráva klientovy, aby odehrál svůj tah.	"a;"
10	Server > Klient	Zpráva klientovy, která slouží k jeho odpojení z důvodu nepřipojení soupeře.	"z;"
11	Server > Klient	Zpráva klientovy, která slouží k jeho odpojení z důvodu chyby na straně soupeře.	"x;"

Č.	Příklad	Min. znaků	Max. znaků
1	"1;1;2;1;"	8	8
3	"c;30;30;90;30;"	14	18
6	"s;30;30;90;30;30;90;90;90;30;30;30;90;90;30;90;90;1;"	52	68
7	"h;30;30;90;30;30;90;90;90;30;30;30;90;90;30;90;90;1;30;90;90;90;30;150;90;150;30;90;30;150;90;90;90;150;"	100	132

d) Design

Server je naimplementovaný v programovacím jazyce C a klient v programovacím jazyce Java. Všechny využívané knihovny jsou součástí těchto jazyků. Kód klienta je přenositelný na jakýkoli počítač za předpokladu, že je na daném počítači nainstalovaný jazyk JAVA. Kód serveru je přenositelný na všechny počítače s operačním systémem Linux. Programování této samostatné práce proběhlo pod operačním systémem Linux-Debian 7.1.0. Při práci jsem používal programy Gedit a Eclipse 3.8. Program je spustitelný pomocí příkazové řádky a kód je dále rozšiřitelný.

3) Uživatelská dokumentace

PRAVIDLA HRY:

Jde o hru, ve které se hráči střídají v umístování "stěn", přičemž "stěny" lze vytvářet pouze mezi sousedícími body, které neleží vůči sobě úhlopříčně. Když se hráči podaří uzavřít nějaký čtverec, tedy umístit čtvrtou stěnu, dostává bod. Hra končí, jestliže jsou uzavřeny všechny čtverce respektive umístěny všechny zdi. Ze začátku hráč pouze umísťuje zdi do volného prostoru, ale jakmile se prostor dostatečně zaplní, musí si dávat pozor, neboť jedna jediná chyba může způsobit konec - prohru celé hry.

OVLÁDÁNÍ HRY:

Hra se ovládá pomocí myši, přičemž čáru, tj. stěnu, hráč vždy vytvoří postupným kliknutím na dva různé body, které spolu sousedí a zároveň neleží vůči sobě v úhlopříčné

poloze. O vyhodnocování čtverců a připisování bodů se program stará sám. Informace o stavu hry jsou zobrazeny v TextBoxu a hned pod ním jsou umístěny ukazatele s body jednotlivých hráčů.

SPUŠTĚNÍ HRY:

Před spuštěním hry je nejprve důležité provést předklad serveru a klienta. Veškeré zdrojové soubory jsou ve složce `c_src` a `java_src`.

Pro překlad serveru je připraven soubor *Makefile*. Překlad probíhá pomocí překladače GCC a pro bezproblémový překlad je nutné mít zdrojové soubory (*.c) v jedné složce se souborem *Makefile*. Překlad je pak řízen z příkazové řádky pomocí příkazu **make**, ve které musí být nastavená cesta na místo, kde se nalézají zdrojové soubory a *Makefile*. Pokud vše proběhlo v pořádku, vytvoří se spustitelný soubor *server* v adresáři `c_bin`.

Také pro překlad klienta je připraven soubor *Makefile*. Překlad probíhá pomocí překladače JAVAC a pro bezproblémový překlad je nutné mít zdrojové soubory (*.java) v jedné složce se souborem *Makefile*. Překlad je pak řízen z příkazové řádky pomocí příkazu **make**, ve které musí být nastavená cesta na místo, kde se nalézají zdrojové soubory a *Makefile*. Pokud vše proběhlo v pořádku, vytvoří se spustitelná soubor *klient.jar* v adresáři *java_bin*.

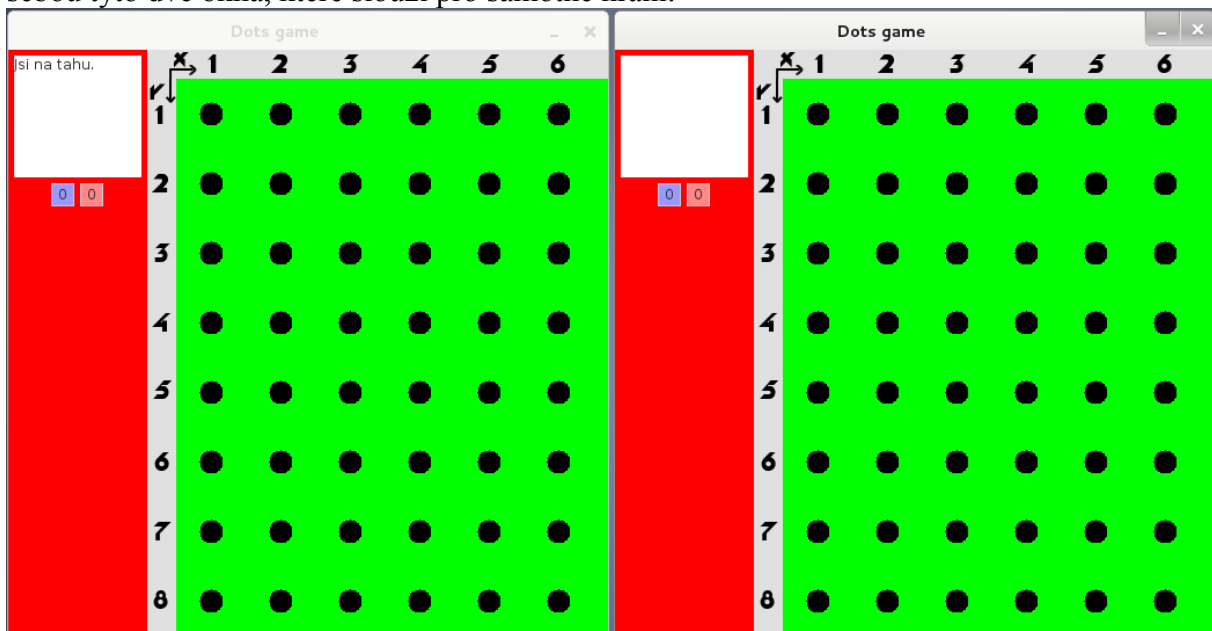
Klientský program je vygenerovaný a plně funkční v souboru *klient.jar*, spustit ho lze pomocí příkazové řádky. Cesta v příkazové řádce musí být nastavená na místo, kde se nalézá spustitelný soubor *klient.jar*. Dále v této složce musí být umístěna složka *img* s potřebnými obrázky pro správný chod programu. Při spuštění klienta je nutné zadat IP adresu serveru a číslo portu, jako vstupní parametry klientského programu. Jinak je klient defaultně nastaven na IP adresu 127.0.0.1 s portem 10000.

```
nowitz@nowitz:~/hra-FINIS/java_bin$ java -jar klient.jar 127.0.0.1 20000
```

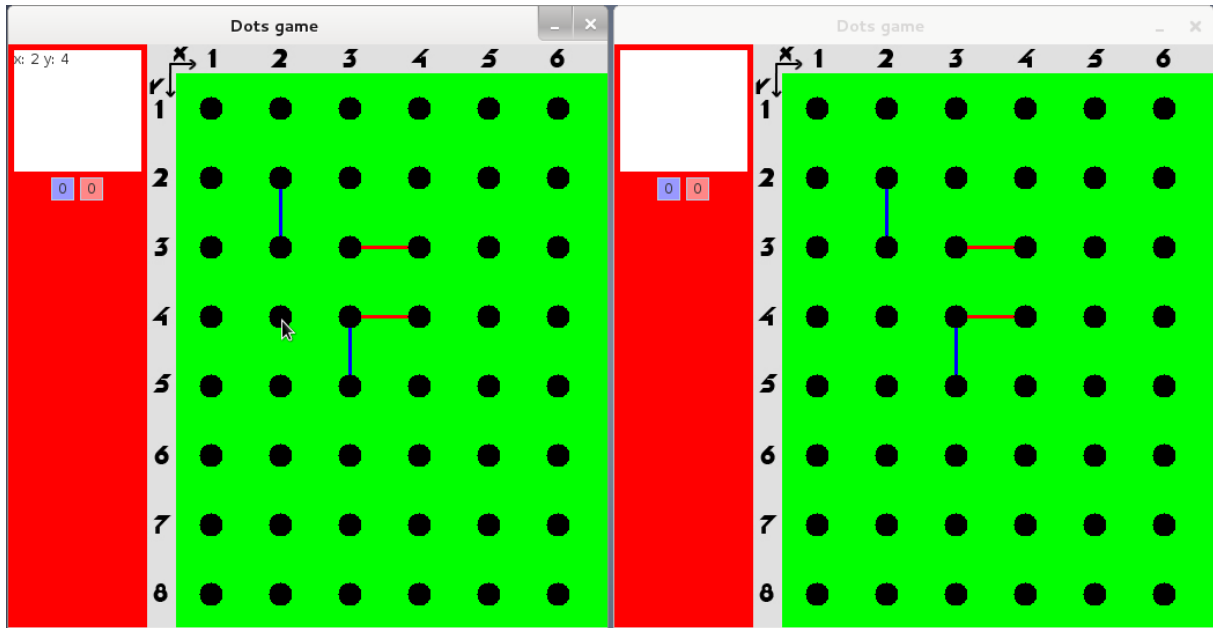
Při spouštění serveru je možnost zadat jako vstupní parametr číslo portu, na kterém má server naslouchat. Pokud je server spuštěn bez parametru, je port defaultně nastaven na 10000. Samozřejmě je nutné, aby byl server spuštěn před klientem.

```
nowitz@nowitz:~/hra-FINIS/c_bin$ ./server 20000
```

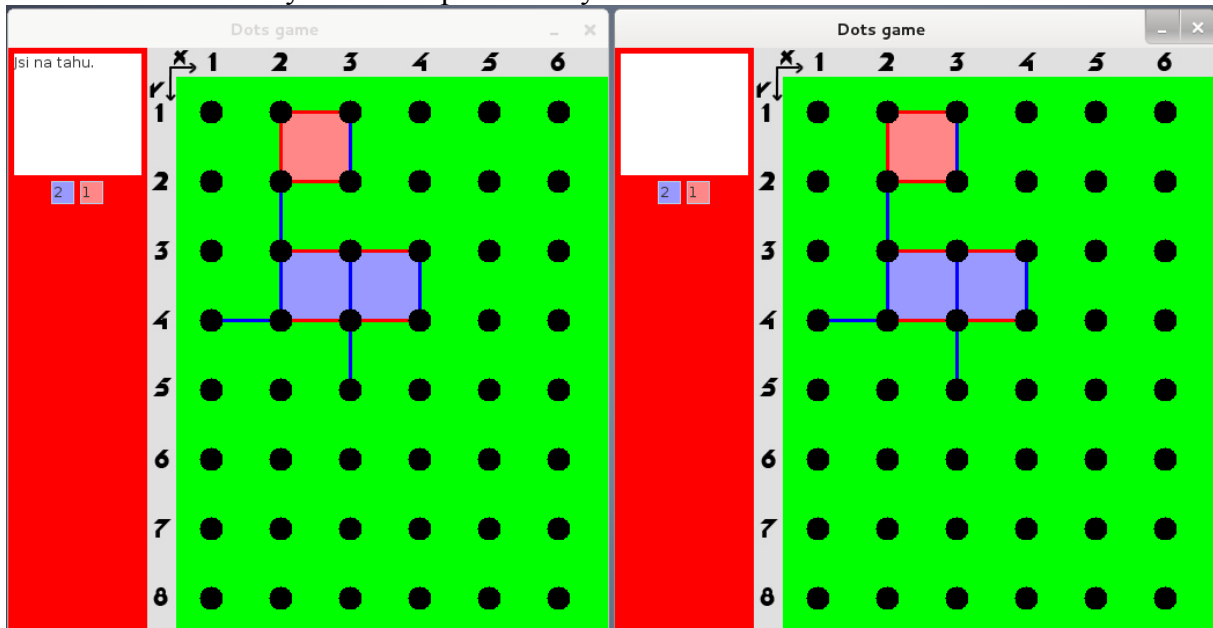
Pokud se vše zdárně zdařilo a hráč hraje hru na jednom počítači, měl by vidět před sebou tyto dvě okna, které slouží pro samotné hraní.



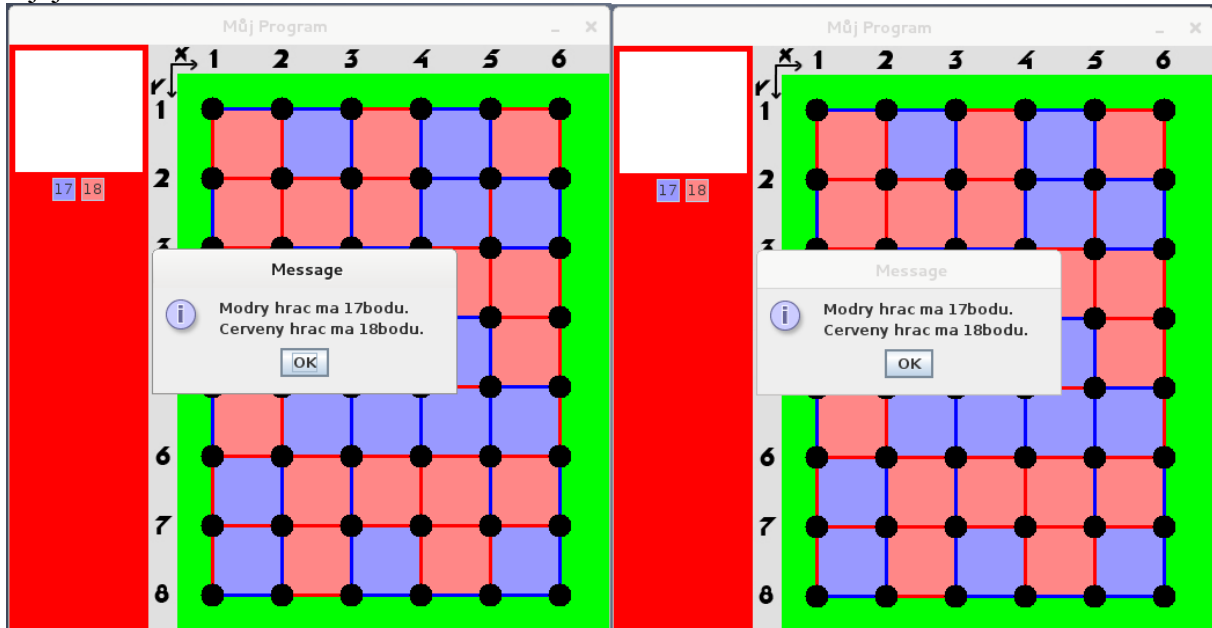
V horním levém rohu je umístěn Textbox, který informuje hráče o stavu hry, a hned pod ním jsou umístěni ukazatelé počtu bodů jednotlivých hráčů. Na dalším obrázku je vidět již několik odehraných kol, přičemž je patrné z Textboxu, že hráč označil bod na souřadnicích 2:4.



Další ukázkový obrázek z průběhu hry.



Po zaplnění veškerých čtverců dojde k samotnému vyhodnocení hry a následně k jejímu ukončení.



4) Závěr

Zadání jsem splnil v plném rozsahu. Práce na této semestrální práci byla pro mě přínosná. Během vývoje programu jsem se seznámil se sokety v Jave a tvorbou serverových aplikací. Také jsem se dozvěděl více informací ohledně tvorby vláken v Jave programovacím jazyku C.

Můj subjektivní závěr je takový, že to byla práce zajímavá a dala mi nejen nové zkušenosti v programovacím jazyce C a Java, ale i větší celkový rozhled o tom, jak funguje protokol TCP, který využívám pro komunikaci mezi klientem-serverem. Se svojí prací jsem spokojen.