Žilinská univerzita v Žiline

Fakulta riadenia a informatiky

**PRINCÍPY OPERAČNÝCH SYSTÉMOV**

**Bulanci**

Semestrálna práca





Školský rok Dávid Pavličko, Adam Hrín

2017/2018 5ZY031

## **Bulanci – Sieťová hra pre dvoch hráčov**

Hra **Bulanci** je veľmi populárnou a známou hrou pre viacerých hráčov (2-4). Hlavnou úlohou každého hráča je zostať na žive do konca a poraziť ostatných hráčov.

V našej verzii tejto skvelej hry **Bulanci** sme sa rozhodli urobiť nasledujúce úpravy:

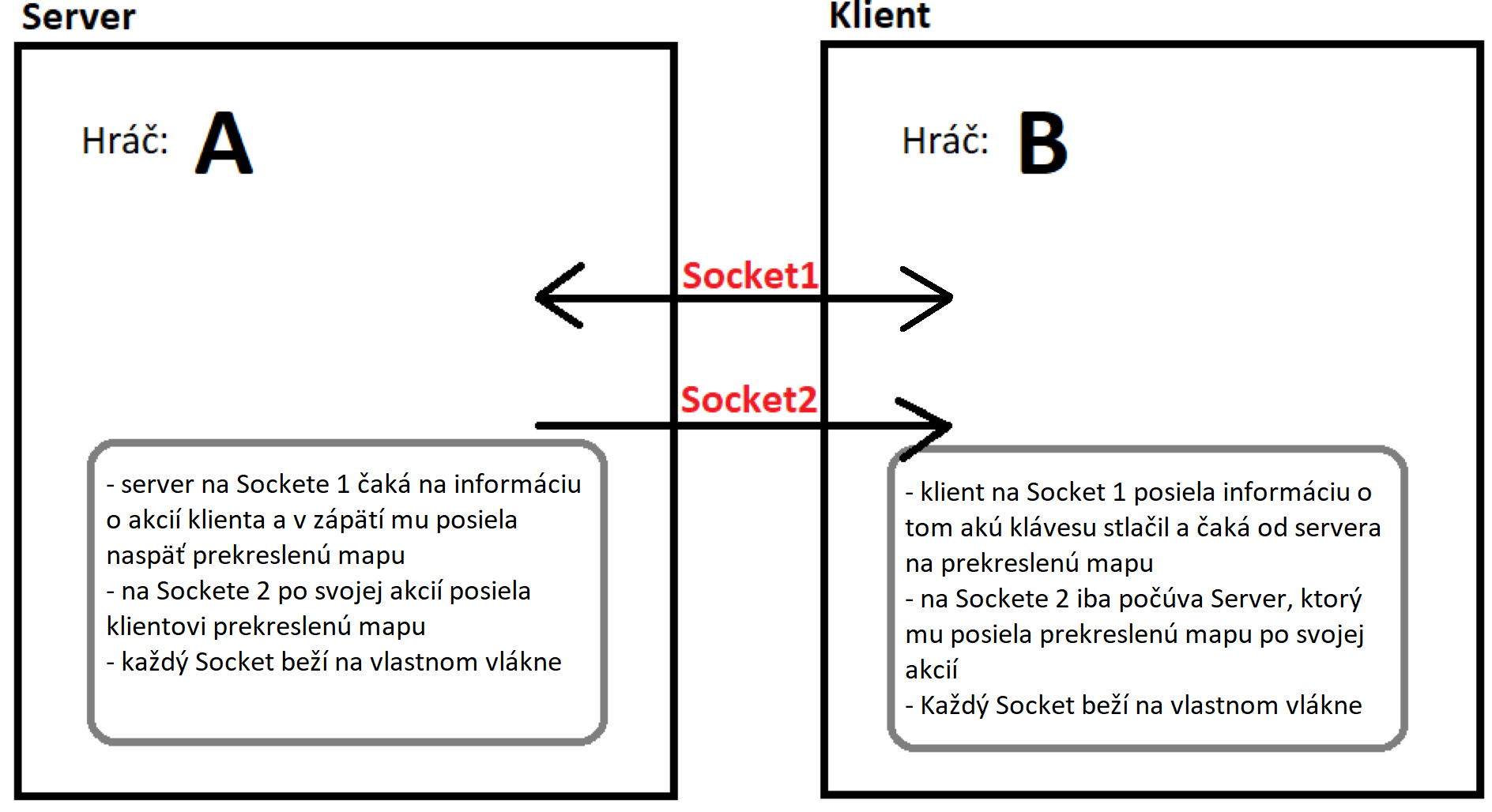
* Textová grafika (použitie znakov na konzole)
* Hra pre dvoch hráčov
* Viac životov
* Rýchli „neviditeľný“ náboj simulujúci realitu

**Ukážka z hry:** Hráč A, Hráč B



## **Analýza & Návrh**

Chod hry bude riadiť dvojica vlákien. Jedno vlákno bude reprezentovať každého hráča zvlášť a oba budú vykonávané serverovou časťou (viď obrázok). Hra bude využívať dva sockety, ktoré bežia na týchto vláknach (každý na jednom), kvôli problému reálneho zobrazovania pre obidvoch hráčov na rôznych monitoroch. Server bude vedieť o všetkom, klient bude poslúchať server. Klient aj Server budú dva samostatné programy, ktoré budú spolu spolupracovať na základe nižšie uvedeného náčrtu.





## **Programátorská príručka**

**Trieda Hráč:**

Konštruktor hráča sa vykonáva na základe troch parametrov, pozícia x, pozícia y a znak, ktorý bude reprezentovať hráča po grafickej stránke.

Trieda Hráč vie vrátiť hráča, pozíciu x, pozíciu y, vyhodnotiť čo sa nachádza pred hráčom. Hráč odchytáva vstupné klávesy zadané užívateľom, povolené klávesy sú šípky pre pohyb a medzerník pre akciu streľba. Ostatné klávesy sú zahodené a ďalej sa nevyhodnocujú.  
Hráč vie vyhodnotiť do akého smeru sa má nasmerovať, vie sa pohybovať a vie vystreliť v danom smere aký je uložený v príslušnom atribúte triedy Hráč.

**Trieda Mapa:**

Mapa vie o triede Hráč a riadi celé vykresľovanie daných hráčov na pozíciách x a y, načítava textový súbor map.txt v ktorej je uložená hracia plocha.

**Trieda Server:**

Server vie o Mape a teda aj o Hráčoch. Server vytvorí dva sockety a vlákna pre každého hráča v metóde createServer(), kde čaká na klienta kým sa nepripojí do hry a neprebehne celý proces create(), bind(), listen(), accept()...

Následne sa spustia obidva vlákna a každé beží na svojej metóde clientPlay() a serverPlay(), kde si posielajú medzi sebou sockety o vykresľovaní a klávesových vstupov.

**Trieda Client:**

Trieda Client je len „hlúpy hráč“, ktorý číta prichádzajúce buffery, zachytáva klávesové vstupy od hráča a následne ich posiela v prostredníctvom buffera naspäť na server, ktorých ho spracuje. Spracovaný buffer odosiela klientovi aby si vykreslil zmeny v hre.  
Pripája sa na sockety príkazom connect().s

**Trieda Hra:**

Hra používa triedy ako Map, Client & Server. Obsahuje jedinú metódu play() v ktorej vytvára samostatne bežiaci program, ktorý rozhodne či ma vytvoriť serverovú časť, alebo klientsku s daným portom.

**Trieda main:**

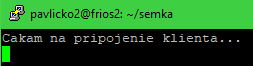
Trieda main vytvára samostatne bežiacu aplikáciu/hru.  
Po spustení hráč musí zadať dva vstupné parametre stranu -> či sa jedná o klienta alebo server **(server, client)** a číslo portu na ktorom bude hra bežať. Čísla portov musia byť rovnaké aby sa v rámci localhostu server a klient navzájom prepojili a mohli si posielať buffre prostredníctvom socketov. Číslo portu musí byť väčšie ako **1024!**

## **Vytvorenie a Spustenie Hry**

Ako prvý sa musí vytvárať serverová časť, teda hráč, ktorý bude hrať na strane servera. Docielime toho nasledujúcim príkazom.



Následne sa vytvorí server/serverová časť pre prvého hráča, ktorý čaká na pripojenie ďalšieho hráča do hry.



Druhy hráč sa pripojí do hry prostredníctvom nasledujúceho príkazu.

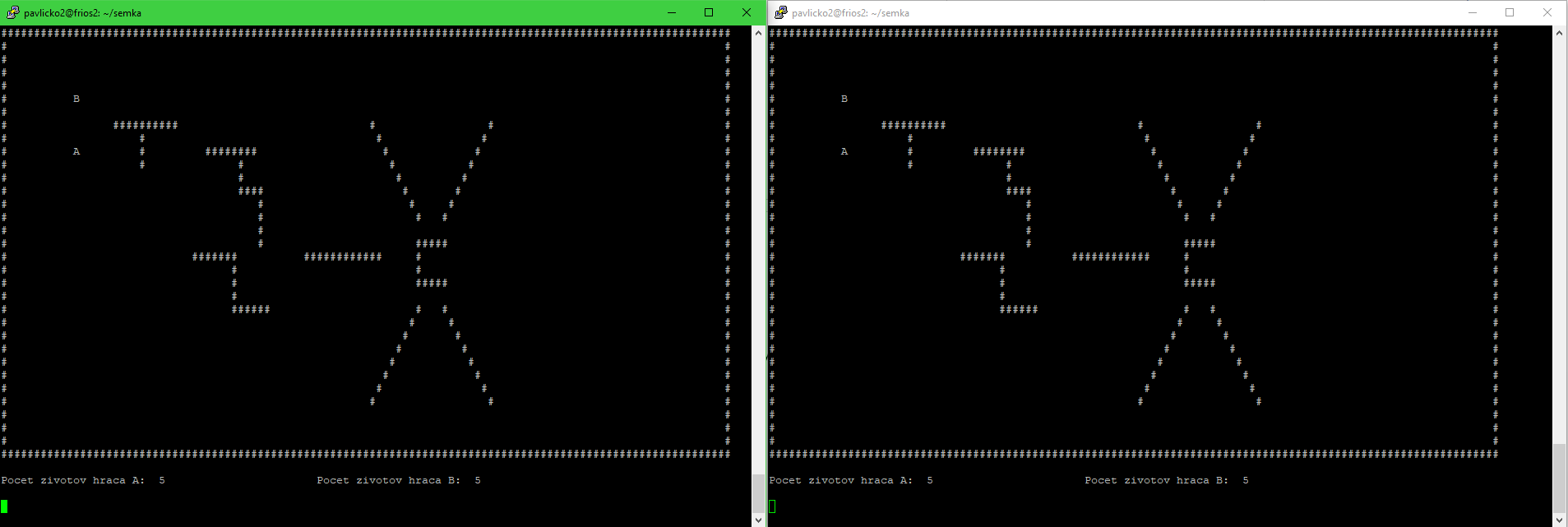


**Pozor:** Je potrebné zadať port väčší ako **1024!**

## **Ukážka & Ovládanie Hry**

Po úspešnom pripojení oboch hráčov do hry sa každému hráčovi vykreslí jeho samostatné okno.

**SERVER KLIENT**



Okraje hracej plochy a nepriechodné steny sú reprezentované znakom „#“. Cez tieto znaky sa nedá prejsť ani strieľať. Hráč má niekoľko životov a sú vypisované na spodku, pod hracou plochou. Rýchlosť strely je instantná, simulovaná realite.

**Ovládanie Hráča:**

Hráč môže vykonať dve akcie:

* Pohyb – pomocou šípok na klávesnici ( ←, ↑, →, ↓ )
* Streľba – pomocou klávesy

## **Sockety**

Sockety sú vytvorené vrámci triedy Server, konkrétne v jeho metóde createServer(), a klient sa na ne pripája vo svojej metóde createClient().

Kým aspoň jeden hráč žije, prvý socket na serveri počúva akciu klienta a spracované dáta mu pošle naspäť. Klient po stlačení klávesy zapisuje vstup do buffera a posiela ho na server, ten tento vstup spracuje, vykoná príslušnú akciu, vykreslí mapu na sebe a pošle ju na klienta aby si ju tiež vykreslil.

Druhý socket slúži na posielanie prekreslenej mapy po udalostiach na serveri klientovi aby si vykreslil všetky nové zmeny. Po akcii servera sa na buffer uloží aktuálna mapa a cez socket sa pošle na klienta a prekreslí dáta.

## **Vlákna**

Bulanci bežia na dvoch vláknach. Jedno vlákno je vytvorené pre hráča na serverovej strane a druhé je vytvorené pre hráča na klientskej strane. Obidve vlákna a celá logika je spracovávaná na strane servera a potrebné zmeny sú odosielané na klienta.  
Vlákna sú vytvárané v metóde createServer() triedy Server. Taktiež dve vlákna sú ešte v triede Client, vytvárajú sa v metóde createClient(). Na obidvoch stranách na jednom vlákne sa spracúva akcia servera a na druhom akcia klienta.

Vlákno pre hráča na serveri je spracovávané v metóde serverPlay(), kde v neustále bežiacom cykle pokiaľ sú obidvaja hráči nažive prebieha načítanie prostredníctvom metódy getch() vstupov z klávesnice. Vstupy sa vyhodnotia vykoná daná udalosť a zmeny vrámci vykresľovania sú odosielané na klienta pre druhého hráča. Taktiež na klientovi je na toto vyhradené jedno vlákno (v opačnom smere).

Vlákno pre druhého hráča je spracovávané v metóde clientPlay() triedy Server, ktoré je vykonávané veľmi podobne ako vlákno hráča bežiacom na serveri. Zmenou je, že vlákno musí najskôr prečítať poslaný buffer z klientskej časti, ktoré zachytáva vstupné klávesy až potom sa udalosť vykoná a odošlú zmeny späť na klienta. Taktiež na toto je na klientovi vyhradené jedno vlákno (v opačnom smere).

## **Synchronizačné Problémy**

**Problém reálneho zobrazovania priebehu hry pre obidvoch hráčov:**

Preto aby hráč na serveri ako aj hráč na klientovi videli vždy všetky zmeny v rámci mapy a vykresľovali sa im vždy aktuálne informácie bolo potrebné pridanie druhého socketu.  
Prvý socket sme použili na spojenie servera s klientom, kde pomocou bufferu server zasiela všetky zmeny, ktoré hráč na strane servera vykoná pre hráča na klientskej strane, aby si ich vykreslil aj on.  
Druhý socket sme použili pre klienta, ktorý zachytáva vstupy z klávesnice, ktoré potom odosiela na server. Pretože klient je „hlúpym“ a server vykonáva všetku logiku a chod hry tak v príslušnom vlákne buffer od klienta prečíta a zmeny na vykreslenie odosiela naspäť klientovi.

**Všetko prebieha prostredníctvom vlákien, ktoré sa mutexujú ak klient, alebo server pristupuje na spoločné miesta v pamäti.**

**Problém pohybu a strieľania jednotlivých hráčov:**

Nato aby aplikácia plynule bežala na dvoch rôznych obrazovkách bolo potrebné aby sme vytvorili vlákna zvlášť pre hráča na serveri a zvlášť pre hráča ktorého reprezentuje klient.  
Akonáhle sa zachytí server, alebo klient akciu jeho vlákno sa mutexne a pristupuje k spoločnej premennej mapa, kde si pýta ako aj mení údaje a posiela výsledný výpis pre klienta. Celú logiku hry spracováva server v triede Server, konkrétne v metódach serverPlay() a clientPlay(), kde všetky udalosti hráčov sú ošetrené aby zabezpečili plynulý chod hry.