



**FAKULTA APLIKOVANÝCH VĚD
ZÁPADOČESKÉ UNIVERZITY
V PLZNI**

KIV/UPS – Semestrální práce

Síťová aplikace hry Kostky(Farkle)

Vypracoval: Martin Schön
Studijní číslo: A22B0144P
Datum: 25. ledna 2025

Obsah

1	Zadání	3
2	Zvolená varianta	3
3	Popis hry	4
4	Protokol	5
4.1	Zprávy	5
4.1.1	Zprávy ze serveru	6
4.1.2	Zprávy z klienta	7
5	Implementace	9
5.1	Server	9
5.1.1	Main	9
5.1.2	Server	9
5.1.3	Messages	10
5.1.4	Data	10
5.1.5	Events	11
5.1.6	Utils	11
5.2	Klient	12
5.2.1	Main	12
5.2.2	GameObjects	12
5.2.3	GUI	12
5.2.4	Server	13
5.2.5	Utils	13
6	Uživatelská příručka	14
6.1	Server	14
6.2	Klient	14
7	Závěr	15

1 Zadání

Hlavní cíle práce:

- Síťová hra pro více hráčů, architektura server-klient (1:N), PC
- Server: C/C++
- Klient: Java, C# (i např. Unity), Kotlin nebo jiný vysokoúrovňový jazyk (musí schválit cvičící)

Varianty zadání:

- tahová hra (prší, mariáš, šachy, piškvorky, ...)
- real-time hra (různé skákačky a střílečky, tanky, "Bulánci", ...)
- pseudo-real-time (Pong, Arkanoid, ...)

Požadavky na protokol:

- textový protokol nad transportním protokolem TCP nebo UDP
 - při shodě zadání v rámci cvičení (max. 2) bude každý student používat jiný
- bez šifrování
- využijte znalostí ze cvičení při návrhu (např. transparentnost při přenosu dat, apod.)
- když se nic neděje (žádný hráč nic nedělá), nic se neposílá
 - výjimkou může být občasná ping zpráva
- na každý požadavek přijde nějaká reakce (byť by šlo pouze o jednoznakové potvrzení, že se operace podařila)

Celé zadání je dostupné zde:

<https://home.zcu.cz/~ublm/files/PozadavkyUPS.pdf>.

2 Zvolená varianta

Server je implementován v jazyce C++ a klient je implementován v jazyce Java. Jako hra byla zvolena hra Kostky (Farkle) inspirovaná minihrou ze hry **Kingdome Come: Deliverance**. Hra je tahová pro dva hráče.

Byl využit textový protokol nad transportním protokolem TCP.

3 Popis hry

Hra Kostky (Farkle) je hra pro dva hráče, kteří se střídají v tazích. Každý tah hráč hodí šest kostek a podle výsledku hodů získává body. Hráč po každém hodu může vybrat, které kostky si nechá a které hodí znovu. Hráč se může rozhodnout bude-li házet znovu anebo si ponechá body z tohoto tahu. Pokud hráč vybral všechny kostky, může hodit znovu všechny šest kostek. Pokud hráč hodí a nezíská žádné body, ztrácí všechny body z tohoto tahu a je na tahu druhý hráč. Hra končí, když jeden z hráčů dosáhne 4000 bodů.

Kombinace kostek, které hráč může získat a libovolně kombinovat. Kombinace s vyšší hodnotou jsou mají vyšší priority.

$\frac{\times}{value}$	1	2	3	4	5	6
1	100	200	1000	2000	4000	8000
2	0	0	200	400	800	1600
3	0	0	300	600	1200	2400
4	0	0	400	800	1600	3200
5	50	100	500	1000	2000	4000
6	0	0	600	1200	2400	4800

Dále jsou zde 3 speciální kombinace:

- **Straight** – 1500 bodů
- **Straight 1 to 5** – 750 bodů
- **Straight 2 to 6** – 1000 bodů

4 Protokol

Protokol je založen na textových zprávách, které jsou posílány mezi serverem a klientem. Každá zpráva má na prvních 8 znacích "**magic**", který obsahuje:

- Dvoupísmenný kód (**dc** a **ds**), který určuje, zda zpráva přichází ze serveru nebo z klienta.
- 4 číslice, které určují délku zprávy.
- 2 středníky pro oddělení kontrolního kódu, délky a zbytku zprávy.

4.1 Zprávy

Zprávy jsou rozděleny do dvou tabulek podle toho, zda zpráva přichází ze serveru nebo z klienta.

Úplná zpráva obsahuje dodatečné informace, které jsou oddělovány středníkem. Konkrétně se jedná o tag a o data.

Formát zprávy:

- Zpráva ze serveru: **ds** ; <délka> ; <tag> ; <data>
- Zpráva z klienta: **dc** ; <délka> ; <tag> ; <data>

4.1.1 Zprávy ze serveru

Id	Tag	Data	Cíl
S1S	log:in:	SUCCESS	žádost o C1
S1E	log:in:	ERROR	odpojení
S2S	log:nameset:	SUCCESS	přesunutí do Queue
S2E	log:nameset:	ERROR	odpojení
S3	game:create:	InitGameData	přesun do hry
S41	game:throw:	GameData	zaslání všech změn po hodu kostkami
S42	game:select:	GameData	zaslání všech změn po výběru kostky
S43	game:nextT:	GameData	zaslání všech změn po dalším hodu
S44	game:endT:	GameData	zaslání všech změn po konci tahu
S5	alive:	xxx	dotaz pro klienta zda žije
S6	connection:recConn:	FullGameData	kompletní synchronizace hry
S7	queue:rejoin:	xxx	přesunutí ze hry do Queue
S8	game:playerLeft:	Name	oponent se odpojil
S9	game:playerJoined:	Name	oponent se vrátil
S10	game:playerDisconnected:	Name	oponent ztratil připojení
S11	ping:	xxx	ping-pong
S12	terminating:	xxx	porušení protokolu → odpojení

Legenda:

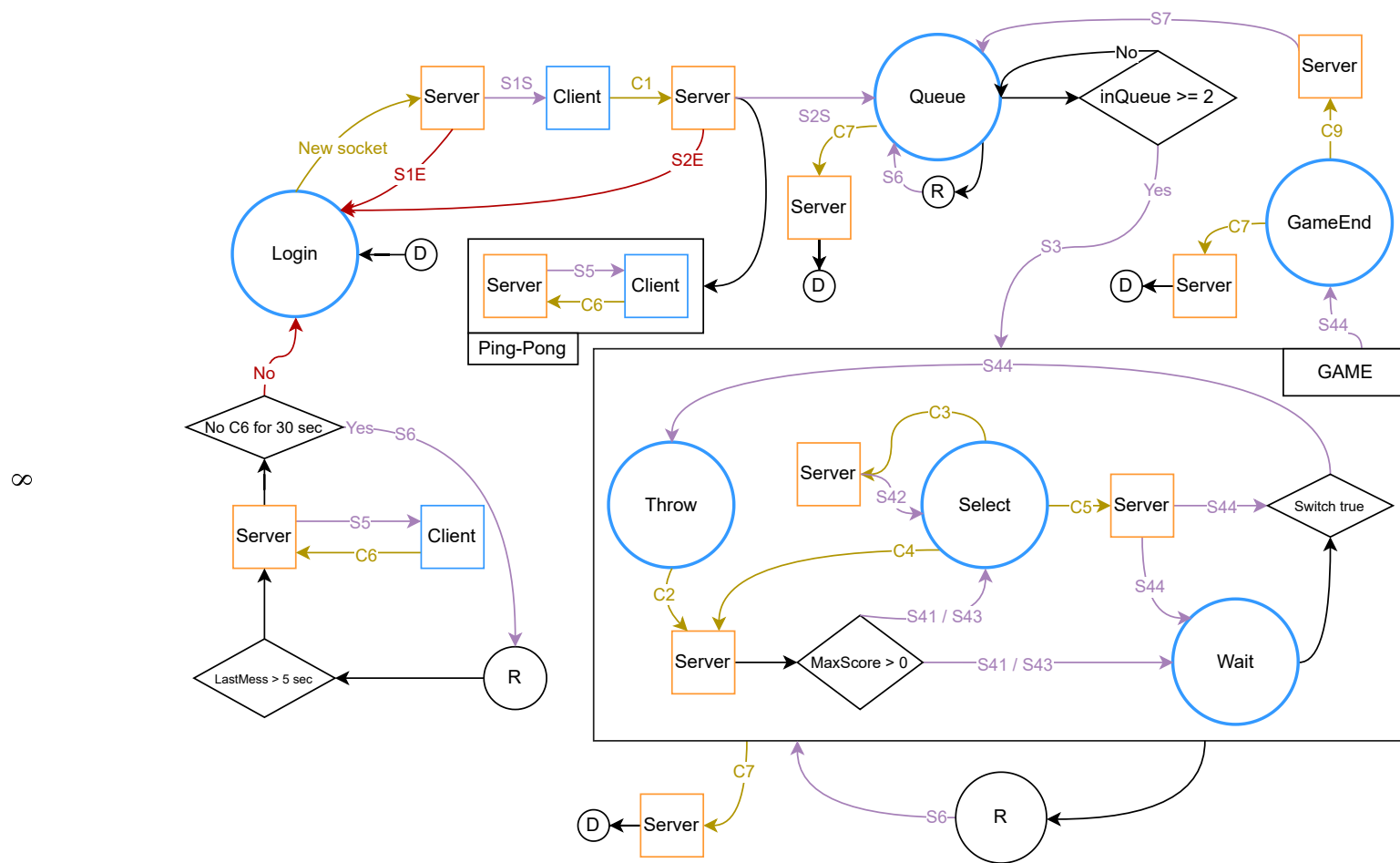
- **InitGameData** – Počáteční data pro hru.
- **GameData** – Data obsahující změny ve hře.
- **FullGameData** – Kompletní data pro hru.
- **Name** – Jméno hráče, který se odpojil.
- **xxx** – Žádná data.
- **SUCCESS** – Zpráva o úspěchu.
- **ERROR** – Zpráva o chybě.

4.1.2 Zprávy z klienta

Id	Tag	Data	Cíl
C1	log:nameset:	Name	zaslání jména na server
C2	game:throw:	xxx	zaslání příkazu hod kostkami
C3	game:select:	DiceId	zaslání vybrané kostky
C4	game:nextT:	xxx	zaslání příkazu o hod znovu
C5	game:endT:	xxx	zaslání příkazu o ukončení kola
C6	connection:recConn:	xxx	žádost o celá herní data
C7	log:out:	xxx	zaslání zprávy pro odpojení
C8	pong:	xxx	ping-pong
C9	queue:rejoin:	xxx	žádost o opětovné připojení do Queue

Legenda:

- **Name** – Jméno hráče.
- **DiceId** – Id kostky.
- **xxx** – Žádná data.



Obrázek 1: Diagram protokolu

5 Implementace

5.1 Server

Server je implementován v jazyce C++ s využitím BDS socketů pro komunikaci s klientem a standardní knihovny pro zpracování zpráv. Celý server je rozdělen do několika modulů:

- Main – Hlavní modul.
- Server – Modul pro komunikaci s klientem.
- Messages – Modul pro zpracování zpráv a generací odpovědí.
- Data – Modul pro ukládání dat.
- Events – Modul pro zpracování událostí.
- Utils – Modul pro pomocné funkce.

5.1.1 Main

Hlavní modul obsahuje inicializaci serveru a jeho běh. Načítá IP a port z příkazové řádky a spouští server. Dodatečně spouští vlákno pro zpracování příkazu nad serverem. Příkazy:

- **exit** – Ukončení serveru.
- **help** – Vypsání nápovědy.
- **players** – Vypsání seznamu připojených klientů.
- **games** – Vypsání seznamu her.
- **all** – Vypsání všech dat.

5.1.2 Server

Modul pro komunikaci s klientem. Zde se server zinicizuje, naslouchá na zadaném portu a přijímá nové klienty. Pro zpracování byly využity Select a BDS socket. Každý klient tedy má svoje fd neboli file descriptor. Všichni klienti jsou zpracováni v jednom vlákně. Po kontrole všech fd se zpracují speciální události (nová hra, smazání hry, odpojení klienta, update, znovu připojení klientů do hry).

Dále zde běží vlákno pro zjištění živosti klienta. Klient každé 3 sekundy pošle pong a server odpoví ping. Pokud klient nepošle ping do 5 sekund, server se zeptá, jestli klient žije. Pokud klient neodpoví do 30 sekund je odpojen.

Čtení zprávy má dvě fáze: Načtení hlavičky a načtení zbytku zprávy. Po načtení hlavičky se zkontroluje, zda je zpráva validní. Pokud ano načte se zbytek zprávy a zpracuje se. Pokud ne, je klient odpojen.

Zpracování zpráv je delegováno do modulu Messages.

5.1.3 Messages

Modul pro zpracování zpráv a generování odpovědí. Z přečtené zprávy se zjistí tag a data. Tag se vyhledá v unordered mapě a zavolá se příslušná funkce. Pokud tag není nalezen, je zaslána zpráva, která klienta odpojí. Zprávy jsou do jednotlivých modulů, pokud se jedná o Login, Logout, Reconnect, RejoinQ a GameM (Všechny zprávy týkající se hry).

5.1.4 Data

Modul pro ukládání dat. Zde jsou uloženy všechny informace o hrách, hráčích a herních objektech.

Objekt **hráč** obsahuje:

- Jméno hráče.
- File descriptor.
- Stav hráče určen číslem:
 - > -2 – Hráč bez jména.
 - > -1 – Odpojený hráč s možností se vrátit.
 - > 0 – Hráč v Queue.
 - > 1 – Hráč ve hře.
- Poslední aktivita (pro kontrolu živosti).

Objekt **hra** obsahuje:

- Vector hráčů.
- Vector jmen hráčů.
- onMove – Jméno hráče na tahu.
- Vector Vektorů kostek.
- Vector Vektorů skóre.
- bool end – Konec hry.
- bool stop – Zastavení hry.

Tento objekt má dále metody pro funkce hry.

Objekt **kostka** obsahuje:

- Id kostky.
- Hodnotu kostky.
- Vybraná kostka.
- Držená kostka.

Dále má metody pro zpracování kostek (Roll, Select, Hold).

Objekt **DataVectors** obsahuje 2 vektory pro ukládání všech her a hráčů.

5.1.5 Events

Modul pro zpracování událostí. Jedná se o události:

- Nová hra – Vytvoření nové hry.
- Reconnect – Připojení odpojeného hráče do hry.
- Oznámení, zda byl klient odpojen nebo připojen.

5.1.6 Utils

Modul pro pomocné funkce. Nachází se zde funkce pro počítání skóre a zpracování příkazů nad serverem. Dále je zde soubor **Consts.h** s konstantami pro server.

5.2 Klient

Klient je implementován v jazyce Java s využitím knihovny Swing pro grafické rozhraní. Celý klient je rozdělen do několika modulů:

- Main – Hlavní modul.
- GameObjects – Modul pro herní objekty.
- GUI – Modul pro grafické rozhraní a zpracování příchozích zpráv.
- Server – Modul pro komunikaci se serverem.
- Utils – Modul pro pomocné funkce.

5.2.1 Main

Hlavní modul obsahuje inicializaci klienta a jeho běh.

5.2.2 GameObjects

Modul pro herní objekty. Třídy Board Dice a PlayerStats obsahuje potřebné funkce pro vykreslení herního pole. Zatím co Board je jen vykreslení Dice a PlayerStats obsahují i pár proměnných pro zpracování dat.

5.2.3 GUI

Modul pro grafické rozhraní a zpracování příchozích zpráv. Nachází se zde 4 panely:

- LoginPanel – Panel pro přihlášení.
- QueuePanel – Panel pro čekání ve frontě.
- GamePanel – Panel pro hru.
- HelpPanel – Panel pro nápovědu.

Všechny panely, kromě GamePanelu používají GridBagLayout pro rozložení prvků. GamePanel používá funkčnost Graphics2D pro vykreslení hry. Ovládání je pomocí klávesnice s využitím InputMap a ActionMap.

5.2.4 Server

Modul pro komunikaci se serverem. Obsahuje třídu `Connection`, která je navržena jako `Singleton`. Po vyplnění údajů v `LoginPanelu` se otevře socket a začne komunikace se serverem. Pokaždé, co je třeba poslat zprávu se zavolá metoda `makeContact`, která sestaví zprávu a odešle ji. Příchozí zprávy jsou čteny v samostatném vlákně. Pokud je zpráva validní zpráva projde dál do GUI pro zpracování. Dále zde běží pingovací vlákno pro životnost klienta. Po 5 sekundách co nepřišla zpráva se registruje odpojení a po 30 sekundách se klient vrátí na `Login`. Dále je zde třída `Messages`, která obsahuje všechny zprávy, které mohou být poslány ze serveru nebo klienta.

5.2.5 Utils

Modul pro pomocné funkce. Nachází se zde soubor `Consts.java` s konstantami pro klienta. Dále `Colours.java` s barvami pro GUI a objekty. Třída `Materials.java` obsahuje načítání textur pro objekty

6 Uživatelská příručka

6.1 Server

Server může být spuštěn pouze na Linuxu. Pro přeložení a spuštění serveru je třeba mít nainstalovaný `g++` a `make`.

- **Přeložení:** `make`
- **Spuštění:** `./Dice_Server`
- Po spuštění zadat IP a port do příkazové řádky.

6.2 Klient

Klient může být spuštěn na Windows, Linuxu a MacOS. Pro přeložení je třeba mít nainstalovaný JDK 17 a Apache Maven.

- **Přeložení:** `mvn package`
- **Spuštění:** `java -jar target/Dice_Client-1.0-SNAPSHOT.jar`

Průchod hrou:

- Po spuštění zadat Jméno, IP a port do GUI (výchozí je 127.0.0.1 a 10000).
- Po připojení se zobrazí Queue, kde čekáte na hru.
- Po připojení druhého hráče se zobrazí Game.
- Ovládání je pomocí klávesnice.
 - **Space** – Hod kostkami.
 - **A** – Pohyb mezi kostkami doprava.
 - **D** – Pohyb mezi kostkami doleva.
 - **E** – Vybrání kostky.
 - **F** – Další hod.
 - **Q** – Ukončení kola.
 - **T** – Náповěda.
- Po skončení hry se zobrazí výsledek, kdo vyhrál a můžete se vrátit do Queue nebo se odpojit.
- Po odpojení se zobrazí Login.
- Pokud se oponent odpojí můžete se vrátit do Queue a čekat na oponentův návrat.

7 Závěr

Cíle práce byly splněny. Byla vytvořena síťová hra Kostky (Farkle) pro dva hráče. Pokud jste teď dohráli hru Kingdome Come: Deliverance, můžete si zahrát tuto hru s přáteli. Ovšem není to přesná kopie, ale spíše inspirace. Bylo by možné přidat možnosti pro více typů kostek (např. lichá a sudá kostka).