

Dokumentace – PIG

Autor: Filip Doležal (A22B0288P)

Předmět: KIV/UPS

Popis hry

Pig (Kostková hra)

Pig je jednoduchá kostková hra určená pro dva hráče. Cílem hry je jako první dosáhnout hranice 30 bodů (tuto hodnotu lze konfigurovat).

Pravidla

Hráči se střídají v tazích. Když je hráč na tahu, má dvě možnosti:

- **ROLL (hodit):** Hráč hodí kostkou (hodnoty 1-6). Pokud padne jednička, ztrácí všechny body nasbírané v aktuálním tahu a jeho tah končí. Pokud padne jakékoliv jiné číslo (2-6), hodnota se přičte k dočasným bodům za tento tah a hráč se může rozhodnout házet znovu.
- **HOLD (zastavit):** Hráč ukončí svůj tah a body, které v něm nasbíral, se bezpečně přičtou k jeho celkovému skóre.

Vítězí ten, kdo jako první dosáhne 30 bodů.

Herní mechaniky

Hra počítá s možností výpadku spojení. Pokud hráč ztratí spojení, má 20 sekund na to, aby se znovu připojil. Zároveň je zde ochrana proti nečinnosti – pokud hráč neprovede žádnou akci déle než 20 sekund, server ho odpojí. Aby spojení zůstalo aktivní, klient musí posílat signál PING alespoň jednou za 10 sekund.

Protokol

Formát zpráv

Komunikace probíhá pomocí textových zpráv, které jsou vždy zakončeny znakem nového řádku (\n).

Struktura zprávy vypadá takto: PŘÍKAZ | klíč1: hodnota1 | klíč2: hodnota2

Příkazy se píší velkými písmeny. Jednotlivé parametry jsou odděleny svíslítkem a zapisují se jako pár klíč: hodnota.

Příklad zprávy o stavu hry:

GAME_STATE|my_score:10|opp_score:5|turn_score:3|roll:4|your_turn:1

Příkazy od klienta na server

Příkaz	Parametry	Popis
LOGIN	nick:<přezdívka>	Přihlášení hráče
RESUME	-	Obnovení pozastavené hry po reconnectu
LIST_ROOMS	-	Získat seznam místností
JOIN_ROOM	room:<id>	Připojit se do místnosti
LEAVE_ROOM	-	Opustit místnost (pouze v čekání)
ROLL	-	Hodit kostkou
HOLD	-	Ukončit tah
QUIT	-	Vzdát hru
EXIT	-	Odpojit se ze serveru
PING	-	Heartbeat
GAME_STATE_REQUEST	-	Vyžádat aktuální stav hry

Odpovědi serveru klientovi

Odpověď	Parametry	Popis
WELCOME	players:<max>, rooms:<max>	Uvítání po připojení
OK	cmd:<příkaz>, [další]	Potvrzení úspěšného příkazu
ERROR	msg:<chyba>, cmd:<příkaz>	Chybová odpověď
ROOM_INFO	room:<id>, count:<počet>, state:<stav>	Info o místnosti
GAME_START	opp_nick:<přezdívka>, your_turn:<0 1>	Začátek hry
GAME_STATE	my_score, opp_score, turn_score, roll, your_turn	Stav hry
GAME_WIN	msg:<zpráva> (volitelné)	Výhra
GAME_LOSE	msg:<zpráva> (volitelné)	Prohra
GAME_PAUSED	-	Hra pozastavena (reconnect flow)
OPPONENT_DISCONNECTED	-	Soupeř se odpojil
OPPONENT_RECONNECTED	-	Soupeř se znovu připojil
DISCONNECTED	-	Odpojení ze serveru

Klíče a parametry

V protokolu se používají různé klíče. Například "cmd" označuje typ příkazu, "msg" nese textovou zprávu nebo popis chyby. Hráči jsou identifikováni pomocí "nick", místnosti pomocí "room". Stav hry a skóre se přenáší přes klíče jako "my_score", "opp_score" nebo "roll" pro hodnotu na kostce.

Klíč	Datový typ	Popis
cmd	string	Název příkazu
msg	string	Textová zpráva/chyba
nick	string	Přezdívka hráče
room	int	ID místnosti
state	enum	Stav místnosti: WAITING, IN_PROGRESS, PAUSED, ABORTED
count	int	Počet hráčů v místnosti
opp_nick	string	Přezdívka soupeře
your_turn	int (0 nebo 1) (bool)	1 = je váš tah
my_score	int	Vaše skóre
opp_score	int	Skóre soupeře
turn_score	int	Body v aktuálním tahu
roll	int (0-6)	Výsledek hodu kostkou. Nula reprezentuje HOLD
players	int	Max počet hráčů na serveru
rooms	int	Max počet místností

Chybové stavy

Server může vrátit chybu v několika případech. Například pokud pošlete neplatný příkaz, zvolíte si již obsazenou přezdívku nebo se snažíte připojit do plné místnosti. Hra také nedovolí opustit místnost, pokud už zápas běží.

Chyba	Kdy nastane
INVALID_COMMAND	Neplatný nebo nerozpoznaný příkaz
INVALID_NICKNAME	Prázdná nebo příliš dlouhá přezdívka
SERVER_FULL	Server dosáhl limitu hráčů
ROOM_FULL	Místnost je plná
GAME_IN_PROGRESS	Nelze opustit místnost - hra běží
CANNOT_JOIN	Nelze se připojit do místnosti
NICKNAME_IN_USE	Přezdívka je již používána

Stavový diagram a průběh

Životní cyklus hráče začíná připojením a přihlášením (LOGIN), čímž se dostane do lobby. Odtud se může připojit do místnosti (WAITING). Jakmile jsou v místnosti dva hráči, hra začíná (IN_GAME).

Pokud během hry někdo vypadne, hra se pozastaví (GAME_PAUSED). Pokud se hráč nestihne vrátit, hra končí. Standardní konec nastává výhrou nebo prohrou, následně se hráči vrací do lobby.

Omezení

Zprávy nesmí být delší než 256 bytů. Přezdívka má limit 31 znaků. Hra hlídá časové limity pro nečinnost i pro opětovné připojení (obojí 20 sekund).

Implementace serveru

Serverová část je napsaná v jazyce C (standard C11).

Struktura kódu

Kód je rozdělen do několika modulů pro lepší přehlednost. Složka include obsahuje hlavičkové soubory s konfigurací, definicemi pro lobby, herní logiku, protokol a logování. Zdrojové kódy ve složce src implementují samotnou logiku:

```
server/
├── include/
│   ├── config.h      # Konfigurační konstanty
│   ├── server.h      # Hlavní serverové funkce
│   ├── lobby.h       # Správa hráčů a místností
│   ├── game.h        # Herní logika (Pig)
│   ├── protocol.h    # Serializace zpráv, TCP buffering
│   ├── parser.h      # Parsování příkazů
│   └── logger.h      # Logování
└── src/
    ├── main.c        # Entry point, argument parsing
    ├── server.c       # Accept loop, klientská a herní vlákna
    ├── lobby.c        # Správa hráčů, místností, reconnect
    ├── game.c         # Pravidla hry Pig
    ├── protocol.c     # Odesílání/příjem zpráv
    ├── parser.c       # Tokenizace příkazů
    └── logger.c       # Thread-safe logování
```

Fungování aplikace

Server běží na hlavním vlákne, které přijímá nová spojení. Pro každého klienta se vytváří vlastní vlákno, které vyřizuje jeho požadavky v lobby. Pro každou běžící hru pak vzniká samostatné herní vlákno, které řídí průběh zápasu.

Pro synchronizaci dat (jako seznam hráčů nebo stav místnosti) se používají mutexy. Síťová komunikace využívá funkci select pro neblokující čtení, což zajišťuje plynulý chod i při více hráčích.

Knihovny: Používají se standardní POSIX knihovny pro vlákna (pthread) a síťování (sys/socket).

Implementace klienta

Klient je napsaný v Javě (verze 14 a vyšší).

Struktura

Aplikace je navržena podle architektury MVC (Model-View-Controller).

- View: Stará se o grafické rozhraní. Využívá Swing. Najdete zde třídy pro přihlášení, lobby i samotnou hru.
- Controller: Propojuje logiku s grafikou a sítí.
- Net/Model: Zajišťuje komunikaci se serverem, posílání zpráv a udržování stavu.

```

client/src/
├── Main.java                # Entry point
├── controller/
│   ├── ViewController.java  # Řízení UI
│   ├── NetworkController.java # Síťová komunikace
│   ├── ViewToNetworkInterface.java
│   └── NetworkToViewInterface.java
├── model/
│   └── room/
│       └── GameRoomStatus.java # Enum stavů místnosti
├── net/
│   ├── Client.java          # TCP klient, heartbeat
│   ├── Protocol.java        # Definice protokolu
│   ├── ServerMessage.java   # Parsování odpovědí
│   └── msg/
│       ├── ClientMessage.java # Základní třída zpráv
│       ├── MsgLogin.java      # LOGIN zpráva
│       ├── MsgJoinRoom.java   # JOIN_ROOM zpráva
│       ├── MsgRoll.java       # ROLL zpráva
│       ├── MsgHold.java       # HOLD zpráva
│       ├── MsgPing.java       # PING zpráva
│       └── ...
└── view/
    ├── MainFrame.java        # Hlavní okno
    ├── LoginView.java        # Přihlašovací obrazovka
    ├── LobbyView.java        # Lobby (seznam místností)
    └── GameView.java          # Herní obrazovka

```

Použité technologie

Grafické rozhraní je postaveno na Swing. Komunikace probíhá přes standardní Java Sockets.

Překlad a spuštění

Požadavky

Server: GCC, CMake, Linux/Unix.

Klient: Java JDK 14+, CMake (nebo ruční kompilace).

Jak přeložit server

V adresáři serveru vytvořte build složku, spusťte v ní cmake a následně kompilaci. Výsledkem bude spustitelný soubor server.

```
cd server

# Vytvoření build adresáře a konfigurace
cmake -B build -S .

# Kompilace
cmake --build build

# Spustitelný soubor: build/server
```

Jak přeložit klienta

V adresáři klienta lze využít CMake, který vygeneruje .jar soubor. Alternativně lze použít přiložený Makefile nebo kód zkompilevat ručně přes javac a zabalit do jar archivu.

```
cd client
cmake -S . -B build
cmake --build build --target jar
# Výsledný JAR: build/sp-client.jar
```

Spuštění

Server se spouští z příkazové řádky. Můžete mu nastavit port, maximální počet hráčů nebo místností.

```
./server [OPTIONS] [PORT]

Volby:
  -a ADDRESS      IP adresa pro binding (default: 0.0.0.0)
  -p MAX_PLAYERS  Max počet hráčů (default: 10)
  -r MAX_ROOMS    Max počet místností (default: 5)
  -l LOGDIR       Adresář pro logy (default: logs/)
```

Klient se spouští jako běžná Java aplikace (java -jar sp-client.jar). Po spuštění se objeví okno pro zadání IP adresy serveru a přezdívký.

Závěr

Výsledná aplikace je plně funkční síťová hra pro dva hráče. Protokol je navržen tak, aby byl robustní a zvládal i výpadky spojení díky funkci reconnect. Server dokáže obsloužit více her najednou díky využití vláken a klient nabízí uživatelsky přívětivé grafické rozhraní.

Aplikace úspěšně implementuje:

- Funkční hru Pig
- Robustní protokol s validací dat a ošetřením chyb.
- Reconnect mechanismus
- Heartbeat systém – detekce nečinných/odpojených hráčů.
- Thread-safe server – zvládá více současných her.
- Grafický klient