

Semestrální práce z předmětu KIV/UPS

Server - klient : hra Senet

Zdeněk Valeš - A13B0458P - valesz@students.zcu.cz

13.12. 2016

1 Zadání

Naprogramuje síťovou verzi hry Senet. Server naprogramuje v jazyce C, klient může být naprogramován v Jave.

Obojí musí být přeložitelné a spustitelné na školních počítačích, za pomoci automatizačních nástrojů (make, scon, ant, maven).

2 Popis hry Senet

Senet je staroegyptská desková hra pro dva hráče. Každý hráč má 5 kamenů, které jsou na přeskáčku rozestavěny na hrací desce (1. hráč má kámen na poli 1). Hází se dřívky, možné hodnoty jsou 1 - 5 (na rozdíl od kostky mají různou pravděpodobnost).

Hrací deska je rozdělena do třech řad po deseti sloupcích, dohromady tedy třicet polí. Pokud kámen dosáhne na 30. pole, může v dalším tahu opustit hrací plochu.

Hráč si během tahu vybere jeden kámen, kterým se posune o hozenou hodnotu. Může se pohnout dopředu i do zadu, případně může tah přeskočit. Pokud je na poli, kam se chce hráč pohnout kámen druhého hráče kameny se vymění. Pokud má druhý hráč za sebou dva a více kamenů, výměna není možná. Cílem hry je dostat všechny kameny z hrací desky.

3 Popis řešení

3.1 Protokol

Protokol je textově-binární a každá zpráva se skládá ze dvou částí: typ zprávy a obsah zprávy. Typ zprávy je řetězec o právě třech znacích, obsah zprávy má různou délku.

Jednotlivé zprávy jsou uvedeny v následující tabulce:

Tabulka 1: Zprávy posílané serverem klientovi

| Název | Popis | Typ | Obsah |
|------------|---|-----|---|
| Error | Chybová zpráva. Používá klient i server. | ERR | Dva znaky s kódem chyby. |
| Start game | Zpráva oznamující klientovi začátek hry. Posílá pouze server. | INF | Řetězec 'START_GAMEnick1,nick2;'. Kde <code>nick1</code> a <code>nick2</code> jsou nicky hráčů. |
| End game | Zpráva oznamující klientovi konec hry a vítěze. Posílá pouze server | INF | Řetězec 'END_GAMEnick;'. Kde <code>nick</code> je nick vítěze. |

| | | | |
|------------|--|-----|---|
| Start turn | Zpráva oznamující klientovi začátek nového tahu. Klient si updatuje tahová slova uložená u sebe podle tahových slov z této zprávy. | CMD | Tahová slova obou hráčů. 1. je tahové slovo 1. hráče. |
|------------|--|-----|---|

Tabulka 2: Zprávy posílané klientem na server

| Název | Popis | Typ | Obsah |
|------------|---|-----|---|
| New player | Zpráva, kterou se klient přihlašuje na server. Server by měl v určitém časovém limitu odpovědět buď OK, nebo ERR. | CMD | Řetězec ve tvaru 'délkanick'. Kde délka je jeden znak (cifra), který určuje délku nicku. |
| Exit | Klient oznamuje serveru, že odchází. Server by měl reagovat vítězstvím druhého hráče. | INF | Řetězec 'EXIT'. |
| End turn | Zpráva oznamující server, že klient ukončil tah. Server by na ni měl v určitém časovém limitu odpovědět OK (pokud je tah validní), nebo ERR (pokud je tah nevalidní). | INF | Tahová slova obou hráčů. 1. je tahové slovo 1. hráče. |

Tabulka 3: Zprávy posílané klientem i server

| Název | Popis | Typ | Obsah |
|----------|---|-----|------------------|
| Is alive | Obecný dotaz na život protějšku. Protějšek by měl do určitého časového limitu (může být jiný u klienta i serveru) odpovědět OK zprávou. | INF | Řetězec 'ALIVE'. |
| OK info | Ok zpráva. Použitá k potvrzování. | INF | Řetězec 'OK'. |

Tabulka 4: Tabulka obsahující chybové kódy

| Kód chyby | Význam |
|-----------|--|
| 50 | Obecná chyba. Pokud je přijatý jakýkoliv jiný chybový kód, měl by být interpretován takto. |

| | |
|----|---|
| 49 | Chyba při přijetí, nebo zpracování zprávy. |
| 48 | Zpráva má chybný typ. |
| 47 | Zpráva má chybný obsah. |
| 46 | Chybný nick (obecná chyba). |
| 45 | Nick už je na ve hře zaregistrovaný. |
| 44 | Nick nesplňuje požadavek na délku. |
| 43 | Server je plný a nemůže přijmout dalšího hráče. |
| 42 | Ted' není můj tah. Server touto chybou odpovídá na téměř všechny zprávy odeslané klientem, který není na tahu. |
| 41 | Hra už je spuštěná. Tato chyba byla použita ve staré verzi. |
| 40 | Tah odeslaný klientem byl vyhodnocen jako neplatný. |
| 39 | Maximální doma pro přijetí zprávy uplynula (timeout). Například pokud se klient přihlásí na server a nepošle nick v daném časovém limitu. |
| 38 | Maximální počet pokusů pro přijetí zprávy dosažen. Například pokud je maximální počet pokusů na zaslání nicku zastaven na tři, bude tato chyba vrácena po přijetí 3. chybné zprávy. |
| 37 | Nečekaná zpráva. Například pokud server očekává nick a klient pošle zprávu o konci tahu (nebo jinou validní zprávu). |

3.2 Tahové slovo

Tahové slovo uchovává informaci o tahu. Jedná se o pole 5 čísel, kde každé představuje současnou pozici hráčova kamene. V protokolu je tahové slovo implementováno polem 10 znaků, kde každé dva znaky představují číslo v dvojciferné podobě (1 je tedy '0','1').

Příklad tahového slova:

| Kámen | 1 | 2 | 3 | 4 | 5 | Tahové slovo |
|----------------|---|---|---|---|----|--------------|
| Pozice hráče 1 | 1 | 3 | 5 | 7 | 9 | 0103050709 |
| Pozice hráče 2 | 2 | 4 | 6 | 8 | 10 | 0204060810 |

Hodnoty na jednotlivých pozicích musí být v rozsahu 1 až 31, kde 31 značí, že kámen již opustil hrací desku. Tahová slova hráčů se zároveň nesmí překrývat - na jednom poli může být maximálně 1 kámen. Výjimku tvoří pole 31, které na hrací desce reálně neexistuje a v programu značí pouze opuštění hrací plochy.

3.3 Server

3.3.1 Hra

Hra je na serveru naimplementována strukturou `Game_struct` a příslušnými funkcemi v souboru `game.h` (`game.c`). Tento soubor neobsahuje žádný výkonný kód, pouze herní

data a funkce, která tyto herní data podle pravidel mění. Obslužné vlákno hráče pak tyto funkce volá.

Na serveru je pět herních slotů, každý po dvou hráčích. Pokud je všech 5 slotů plně obsazených, server po validaci nicku odešle chybovou zprávu(**ERR43**).

Systém přidělování volných herních slotů funguje na jednoduchém principu - herní slot je volný, pokud má hra nastaven příznak **FREE** a alespoň jeden hráč ještě není inicializován. Po přihlášení druhého hráče je příznak **FREE** vynulován. Pokud hra skončí a opouštějící hráč je poslední, hra si opět nastaví příznak **FREE** a je možné ji znovu přidělit.

3.3.2 Obslužné vlákno a herní smyčka

Obslužné vlákno a herní smyčka z pohledu server jsou popsány následujícím diagramem:

Popis stavů

3.4 Klient

3.4.1 Architektura

Klient je řešen architekturou MVC. Kontrolery se nachází v balíku `org.valesz.ups.controller`, UI v balíku `org.valesz.ups.ui`. Zbytek aplikace je tvořen modelovými třídami a pomocnými třídami (pro komunikaci přes tcp, konstanty...).

3.4.2 Zprávy

Zprávy jsou implementovány třídami v balíku `org.valesz.ups.common.message`. Implementace je rozdělena mezi příchozí a odchozí zprávy. Různé příchozí zprávy jsou odděleny od třídy `AbstractReceivedMessage`, odchozí zprávy jsou tvořeny pouze třídou `Message`. Obě mají podobnou strukturu (typ + obsah). `AbstractReceivedMessage` má ale obsah genericky typovaný - kvůli lepšímu pozdějšímu zpracování v programu. `Message` má obsah pevně typovaný na string.

Možné typy zpráv jsou v enumu `MessageType`. Tento enum je používán oběma typy zpráv.

3.4.3 Hra

Hra je naimplementována v balíku `org.valesz.ups.model.game`. Třída `Game` obsahuje herní principy (posun kamene, hod dřívky...), které jsou volané herním kontrolerem (třída `GameController`). Herní kontroler také odpovídá za update pozic kamenů hráčů (vždy po obdržení zprávy o novém tahu) a přepínání tahů.

Třída sama o sobě tedy pouze obsahuje herní data a metody, které je mohou měnit, ale sama nic nevykonává.

3.4.4 Herní smyčka

Herní smyčka z pohledu klienta je zobrazena na následujícím diagramu:

Popis stavů

4 Postup na sestavení a spuštění

4.1 Server

Pro úspěšný překlad serveru je nutná knihovna `pthread.h`. Překlad a sestavení lze provést pomocí příkazů `scons`, nebo `cmake` v adresáři `code/server`. Spustitelný soubor `server` je v adresáři `code/server/build`.

4.2 Klient

Klient lze přeložit a vyexportovat do spustitelného jar pomocí příkazu:

```
mvn clean compile assembly:single
```

v adresáři `code/client`. Vyexportovaný jar pak lze spustit příkazem

```
java -jar target/*.jar
```

v adresáři `code/client`. Seznam závislostí se nachází v souboru `code/client/pom.xml`.

5 Závěr