

Semestrální práce KIV/UPS

Hra Mlým

Jméno a příjmení: Jiří Velek

Osobní číslo: A20B0269P

1. Pravidla hry mlým

Každý hráč má devět kamenů, které se pohybují po hrací ploše. Cílem hry je sebrat protihráči všechny kameny, nebo mu zabránit v dalším tahu. Pokud se tři kameny stejné barvy dostanou do řady nebo sloupce, hráč složil tzv. mlým a táhne podruhé tak, že sebere z hrací plochy soupeřův kámen. Hra je rozdělena do třech fází:

1.1. Rozmíst'ovací

Hra začíná s prázdnou hrací plochou, hráči na ní postupně umísťují kameny. Po umístění všech osmnácti kamenů mohou hráči kameny přesouvat.

1.2. Přesouvací

Kameny se přesouvají k nejbližšímu volnému sousednímu poli. Hráči se snaží složit mlým a sebrat spoluhráči kameny, a nebo mu zabránit v dalším tahu.

1.2.1 Skoky

Pokud má hráč už pouze 3 kameny, je mu povoleno s kameny skákat a pohybovat s nimi na jakémkoliv volné pole na hrací ploše.

2. Sít'ový protokol

2.1 Formát zpráv

Každá zpráva je ve tvaru:

PREFIX ; ZPRÁVA ; DATA

PREFIX = počet bytů, kolik je třeba přečíst (bez prefixu)

ZPRÁVA = jedinečný identifikátor zprávy

Maximální délka zprávy je 4kB.

2.2 Zprávy a jejich argumenty

CREATE	USERNAME
JOIN	USERNAME
PUT	INDEX
TAKE	INDEX

MOVE	INDEX odkud	INDEX kam		
JOINED				
READY	STATE hráče	COLOR	BOARD	USERNAME protihráče
OVER				
OK				
NOK	MESSAGE			
PING				
PONG				

2.2.1 Význam argumentů

USERNAME = uživatelské jméno, UTF-8 string

INDEX = index v hrací ploše na dané políčko, 4B unsigned integer

STATE = současný stav hráče ve hře (jako číslo)

COLOR = barva hráče, 4B unsigned integer, hodnoty: 0 (prázdná), 1 (červená), 2 (modrá)

BOARD = 24 byte struktura, obsahující jednu hodnotu barvy (0, 1, 2) na jeden byte

MESSAGE = zpráva od serveru, proč se akci nepovedlo vykonat

2.3 Návaznost zpráv

Uživatel se připojí na server, a pošle CREATE nebo JOIN zprávu. Server odpoví zprávou JOINED. Nyní je hráč v lobby. Po naplnění kapacity lobby (2 hráči) se oboum hráčům pošle zpráva READY. Hra začne a hráči mohou hrát pomocí zpráv PUT, TAKE a MOVE. Po každém pohybu se protihráči zkopíruje zpráva, kterou hráč poslal a hráči je odeslána zpráva OK. Pokud byl tah neplatný dostane zprávu NOK s vysvětlením, proč byl tah neplatný. Po dokončení hry je odeslána zpráva OVER a hra je „zničena“ a klienti odpojeni.

Po dobu komunikace jsou periodicky odesílány zprávy PING, PONG.

2.4 Chybové stavy (zprávy)

Chybový stav aplikace nastane, pokud server pošle zprávu NOK. Většinou nastává, když server odpovídá na zprávy.

CREATE, JOIN – uživatelské jméno již někdo využívá

JOIN – není žádná prázdná lobby

MOVE, PUT, TAKE – Neplatná pozice, Hráč není na tahu

2.5 Seznam herních stavů

InGamePut	2
InGamePutOpponent	3
InGameTake	4
InGameTakeOpponent	5

InGameMove	6
InGameMoveOpponent	7
GameOver	8

2.6 Několik předpřipravených zpráv

11;CREATE;Pepa --- vytvoří hru s hráčem Pepa

7;JOINED --- uživatel se úspěšně připojil

4;PING -- pošle ping

4;PONG --- pošle pong

41;READY;2;1;000000000000000000000000;Zdenek --- pošle hráči potvrzení o připojení oponenta, hráč obdrží jeho stav (2), barvu (1), hrací plochu, a název oponenta

5;PUT;8 -- položí kámen na pozici 8

6;TAKE;8 --- sebere kámen z pozice 8

2;OK

29;NOK;Nepovedlo se uvařit steak

3. Implementace

3.1 Server

Server je napsán v jazyce Rust.

3.1.1 Moduly

Server je rozdělen na několik modulů. Moduly, které pracují v různých vláknech spolu komunikují skrz kanály.

Server

Stará se o připojování nových klientů, přijímání zpráv a odpojování klientů.

Receiver

„Dispatcher“ pro zpracování jednotlivých zpráv.

Message

Stará se o serializaci a deserializaci zpráv.

Machine

Konečný automat. Říká, ve kterém stavu se klient nachází. Zařizuje přechody mezi stavy a obsahuje obsluhu zpráv od klienta.

Game

Obsahuje logiku hry.

Player

Obsahuje informace o hráči a slabou referenci na jeho klienta.

3.1.2 Použité knihovny

once_cell – Využita pro inicializaci statických proměnných.

3.1.3 Překlad a spuštění

K překladu je potřeba mít nainstalovaný Rust compiler (rustc, stabilní verze 1.65.0) a balíčkovací systém cargo. Poté stačí spustit příkaz „cargo build“ pro přeložení programu a „cargo run <argumenty>“ pro spuštění programu.

Argumenty programu

Program bere dva argumenty – IP adresu a port

3.2 Klient

Klient je napsán v jazyce Java.

3.2.1 Třídy

Client

Stará se o připojení k serveru, přijímání a odesílání zpráv.

Machine

Enum obsahující všechny stavy hry.

Message

Třída serializující zprávy.

MessageType

Enum s typem zprávy.

3.2.2 Použité knihovny

JavaFX – využito pro grafické rozhraní aplikace

3.2.3 Překlad a spuštění

K překladu je potřeba mít nainstalovaný Java JDK verze 16, Maven. Pro překlad stačí spustit „mvn package“. Ve složce target vznikne jar soubor client-1.0.jar – tento soubor lze spustit příkazem „java --jar client-1.0.jar“.

4. Závěr

Server i klient fungují na operačních systémech Windows a Linux. Dle mého server využívá moc kódu, který vykonává vedlejší účinky a dal by se přepsat mnohem lépe. Bohužel jsem ale v době psaní semestrálky moc neuměl Rust, ani moc nerozuměl tomu, jak sockety fungují a přepisovat celou práci nyní by bylo moc práce. Zároveň mě práce přesvědčila k tomu, abych už nikdy v životě nepsal nic v Javě.