



Semestrální práce z předmětu
Úvod do počítačových sítí

Piškvorky

22. prosince 2023

Autor:

Václav Tran

A21B0299P

`nuva@students.zcu.cz`

Obsah

1	Zadání	2
2	Programátorská dokumentace	3
2.1	Popis řešení	3
2.1.1	Server	3
2.1.2	Klient	4
2.2	Formáty vyměňovaných zpráv	4
2.3	Operace	4
3	Uživatelská příručka	6
3.1	Přeložení a sestavení programu	6
3.2	Použití programu	6
4	Použité zdroje	7
4.1	Zdroje	7
	Literatura	8

Kapitola 1

Zadání

Popis zadání viz [1].

Kapitola 2

Programátorská dokumentace

2.1 Popis řešení

2.1.1 Server

Server byl implementován v programovacím jazyce Go (verze 1.15.15).

Nejdůležitějšími funkcemi byly `disconnectHandler`, `ConnectionCloseHandler` a `ProcessClient`. Přihlášený klient byl uložen na serveru pod svým jménem, které bylo unikátní po určitou dobu a byly také uloženy veškeré informace potřebné k určení stavu klienta. Pokud se klient odpojil (x-krát se nedostal ping na server), `disconnectHandler` o tomto informoval protějšiho hráče a aktualizoval stav klienta, hráč a jeho informace byly ale stále uloženy na serveru.

`ConnectionCloseHandler` odpojil připojení a vymazal klienta ze serveru, pokud uplynula určitá doba. Pro každé nové připojení se vytvořilo nové vlákno nad funkcí `ProcessClient` a tato funkce zpracovávala a vyhodnocovala příkazy klienta.

Datové struktury

`Player` datová struktura reprezentuje klienta s unikátním jménem. Společně s `TicTacToeGame` obsahuje veškeré informace potřebné k obnovení stavu.

`TicTacToeGame` datová struktura reprezentuje hru piškvorek, ale reprezentují i lobby, do kterého se můžou hráči připojit, dát ready a podobně. Obsahuje metody ohledně hry jako tah, kontrola stavu hry (výhra, prohra, remíza).

2.1.2 Klient

Klient byl implementován v programovacím jazyce Python. Pro grafické rozhraní byla využita knihovna tkinter. Nejdůležitější třídy byly `Dispatcher`, `Pinger` a `TCPCClient`. `TCPCClient` spravuje připojení a posílání či přijímání dat. Pokud data nevyhovují protokolu, klient se uzavře.

`Dispatcher` třída zpracovává veškeré přijaté data od serveru. Kontroluje kam patří a pošle data dál do určené třídy, například ping zpráva se předá třídě `Pinger`.

`Pinger` třída periodicky posílá ping zprávu serveru a pokud nedostane od serveru odpověď po x pingech (hodnota x se dá nastavit v souboru `const.py`), tak začne u hry obnovovací proces (opakovaně se zkouší o připojení na server a chce od serveru dostat informace k obnovení svého stavu).

Datové struktury

`IDispatchReceiver` představuje rozhraní, které určuje, kdo může od `Dispatcher` přijímat data.

2.2 Formáty vyměňovaných zpráv

Formáty zpráv vypadají takto (znak ; je zde jenom k přehlednění, v komunikaci se posílá zpráva bez daného znaku):

`Magic;Opcode;Data`

Kde **Magic** představuje hlavičku, který ve zprávě vždycky musí být, jinak zpracování na klientovi či serveru povede k ukončení připojení.

Opcode představuje číslo operace.

Data představují data, server do dat vkládá i stav operace (jestli prošla nebo nastala chyba) a data mohou být oddělená oddělovačem.

2.3 Operace

Zde jsou stručně popsány operace (opcode). Čísla operací se dají nastavit v `const.go` (server) a `const.py` (client).

`MsgLoginOpcode` - Operace přihlášení (Opcode: "001")

`MsgJoinOpcode` - Operace připojení do lobby (hry) (Opcode: "002")

`MsgMoveOpcode` - Operace tahu (Opcode: "003")

`MsgPlayAgainOpcode` - Operace znovu hraní (hráč chce hrát znova po hře) (Opcode: "004")

`MsgGameStartedOpcode` - Hra začala (Opcode: "005")

`MsgReturnToStartOpcode` - Návrat na začátek (zpátky do hlavního menu) (Opcode: "006")

`MsgGameOverOpcode` - Konec hry (Opcode: "007")

`MsgYourTurnOpcode` - Představuje zprávu, která říká, že hráč je na řadě a může provádět tah (Opcode: "010")

`MsgPingOpcode` - Operace ping (Opcode: "011")

`MsgRecoveryOpcode` - Operace obnovení, data jsou v závislosti na stavu hráče a hry (deska, vítěz, atd.) (Opcode: "012")

`MsgPauseOpcode` - Pauza, server posílá klientovi zprávu o pozastavení hry (Opcode: "013")

`MsgContinueOpcode` - Pokračování, server posílá klientovi zprávu o pokračování hry (Opcode: "014")

`MsgStatusOpcode` - Server předává textovou informaci o stavu hry (Opcode: "015")

Kapitola 3

Uživatelská příručka

3.1 Přeložení a sestavení programu

Server byl vyvinut v programovacím jazyce Go verze 1.15.15. Je tudíž potřeba nainstalovat tento jazyk na počítač viz <https://go.dev/>. Následovně je potřeba nainstalovat starší verzi Go (1.15.15) viz <https://go.dev/doc/manage-install> .

Klient byl vyvinut v programovacím jazyce Python verze 3.10. Je tudíž potřeba nainstalovat tento jazyk na počítač viz <https://www.python.org/downloads/>. Následovně je někdy potřeba nainstalovat i tkinter pokud je využit Linux OS. Na Debian a Ubuntu pro toto slouží příkaz:

```
sudo apt-get install python3-tk
```

3.2 Použití programu

Server by se měl spouštět příkazem:

```
go1.15.15 run .
```

Klient by se měl spouštět příkazem:

```
python3 main.py
```

Na Windows slouží k spuštění klienta příkaz:

```
py main.py
```

Kapitola 4

Použité zdroje

4.1 Zdroje

1. Golang dokumentace (<https://go.dev/doc/>)
2. Tkinter dokumentace (<https://docs.python.org/3/library/tk.html>)

Literatura

- [1] Martin Úbl. Zadání semestrální práce z předmětu KIV/UPS. <https://home.zcu.cz/~ublm/files/PozadavkyUPS.pdf>, 2023. [Online].