

Sít'ová hra pro více hráčů – Dots & Boxes

Semestrální práce z předmětu KIV/UPS

23.12.2023

Šárka Dvořáková

A21B0116P

Obsah

1	Popis hry	3
2	Komunikační protokol	3
2.1	<i>Přenášené zprávy</i>	3
2.1.1	Zprávy klient → server	3
2.1.2	Zprávy server → klient	4
2.1.3	Detekce výpadku	5
2.1.4	Poznámka	5
2.2	<i>Stavové diagramy</i>	6
2.2.1	Stavový diagram hráče	6
2.2.2	Stavový diagram hry	6
3	Implementace	7
3.1	<i>Server</i>	7
3.1.1	Použitá verze prostředí	7
3.1.2	Moduly programu	7
3.1.3	Metoda paralelizace	8
3.2	<i>Klient</i>	8
3.2.1	Použitá verze prostředí a knihoven	8
3.2.2	Dekompozice	8
3.2.3	Metoda paralelizace	9
4	Uživatelská příručka	9
4.1	<i>Server</i>	9
4.2	<i>Klient</i>	10
5	Závěr	10

1 Popis hry

Popis hry je možné najít na stránce https://en.wikipedia.org/wiki/Dots_and_boxes.

2 Komunikační protokol

Jako oddělovač je ve zprávách využíván znak pipe |. Konec zprávy je označen jako \n.

2.1 Přenášené zprávy

2.1.1 Zprávy klient → server

- **login** | string *username*

⇒ odpověď:

logged | ok, **logged** | error

Proměnná *username* musí být textový řetězec o délce maximálně 20 znaků. Mezery budou z řetězce odstraněny.

- **queue**

⇒ odpověď:

queued | ok, **queued** | error

- **line_coords** | int *coordinate_1* | int *coordinate_2*

⇒ odpověď:

invalid_coords

draw_coords | int *coordinate_1* | int *coordinate_2*

Obě souřadnice *coordinate_1* a *coordinate_2* musí být nezáporné, nesmí být větší nebo rovny počtu spojovaných teček. Nesmí být už předtím propojené a musí spolu fyzicky sousedit, tj. musí být bezprostředně vedle sebe v grafickém uživatelském rozhraní.

- **leave_queue**

⇒ odpověď:

left_queue | ok, **left_queue** | error

- **leave_game**

⇒ odpověď:

you_left_game | ok, **you_left_game** | error

- jiný příkaz

⇒ odpověď:

unknown_command

2.1.2 Zprávy server → klient

Server tyto zprávy zasílá při nastání určité události. Klient na zprávy neodpovídá.

- **start_game** | int *player_number* | string *opponent_username*
Zasláno oběma hráčům po vytvoření nové hry, když jsou ve frontě alespoň dva hráči. Proměnná *player_number* označuje číslo hráče (nabývá hodnot 1 nebo 2) a *opponent_username* je řetězec s přezdívkou oponenta.
- **play** | int *active_player_number*
Zasláno oběma hráčům, aby věděli, který hráč je právě na tahu. Proměnná *active_player_number* má hodnotu 1 nebo 2.
- **draw_squares** | int *coordinate_1* | int *coordinate_2* | int *coordinate_3* ...
Zasláno oběma hráčům při vytvoření nového čtverce (či čtverců) na herní ploše. Souřadnice označují levý horní bod čtverce a nesmějí obsahovat duplicity. Také se nesmí jednat o souřadnice již vytvořených čtverců. Souřadnice musejí být nezáporné nesmějí být větší nebo rovny počtu bodů na herní ploše. Nesmí jít o souřadnice bodů na krajní pravé hraně či krajní dolní hraně.
- **winner** | int *player_number* | int *score*
Zasláno oběma hráčům při kompletním zaplnění herní plochy čtverci – jedná se o konec hry. Proměnná *player_number* označuje vítěze a může nabývat hodnot 1 a 2 (nebo 0 při remíze). Proměnná *score* obsahuje skóre vítěze. Skóre nesmí být záporné a nesmí být větší než maximální možný počet čtverců. Skóre je rovné 0 při remíze.
- **opponent_left_game**
Zasláno hráči, když se jeho oponent odpojil (to zahrnuje dlouhodobou nedostupnost) nebo odešel ze hry. Hráč bude umístěn znovu do fronty (bude mu tedy poslána ještě zpráva **queued** | ok).
- **resume_game** | int *player_number* | string *opponent_username* |
int *active_player_number* | int *my_score* | int *opponent_score* |
int *my_square_coordinate_1* | int *my_square_coordinate_2* ... |
int *opponent_square_coordinate_1* | int *opponent_square_coordinate_2* ... |
int *line_coordinate_1* | int *line_coordinate_2* ...

Zasláno znovu se připojujícímu již existujícímu hráči (zjištěno podle přezdívky). Klient dostane všechny informace potřebné k vykreslení herní plochy. Proměnná *player_number* je číslo hráče (1 nebo 2), *opponent_username* je přezdívka oponenta, *active_player_number* je číslo hráče na tahu (1 nebo 2), následují skóre hráče *my_score* a *opponent_score* (skóre oponenta). Musí se jednat o nezáporná čísla velká maximálně jako největší možný počet čtverců. Potom zpráva obsahuje souřadnice čtverců hráče *my_square_coordinate* a poté oponenta *opponent_square_coordinate*. Počet souřadnic musí odpovídat skóre. Poslední jsou dvojice spojených bodů *line_coordinate*. Pro všechny souřadnice musí platit již výše zmíněná pravidla.

- **disconnected_suspicious_activity**

Zasláno klientovi, když byl odpojen za použití zprávy nevyhovující protokolu.

- **disconnected_inactivity**

Zasláno klientovi, když byl odpojen za 12 vynechání odpovědí na zprávu **ping**.

- **server_shutdown**

Server tuto zprávu pošle všem připojeným klientům před svým vypnutím.

2.1.3 Detekce výpadku

Server každých n (v této implementaci $n = 5$) sekund posílá zprávu **ping**, na kterou od klienta požaduje odpověď **pong**. Pokud klient neodpovídá, je krátkodobě nedostupný. Pokud neodpoví 12x za sebou, je dlouhodobě nedostupný a je odstraněn ze seznamu hráčů. Zároveň klient podle **ping** zpráv detekuje dostupnost serveru. Pokud již dlouho nepřišla **ping** zpráva, klient se pokouší k serveru znovu připojit.

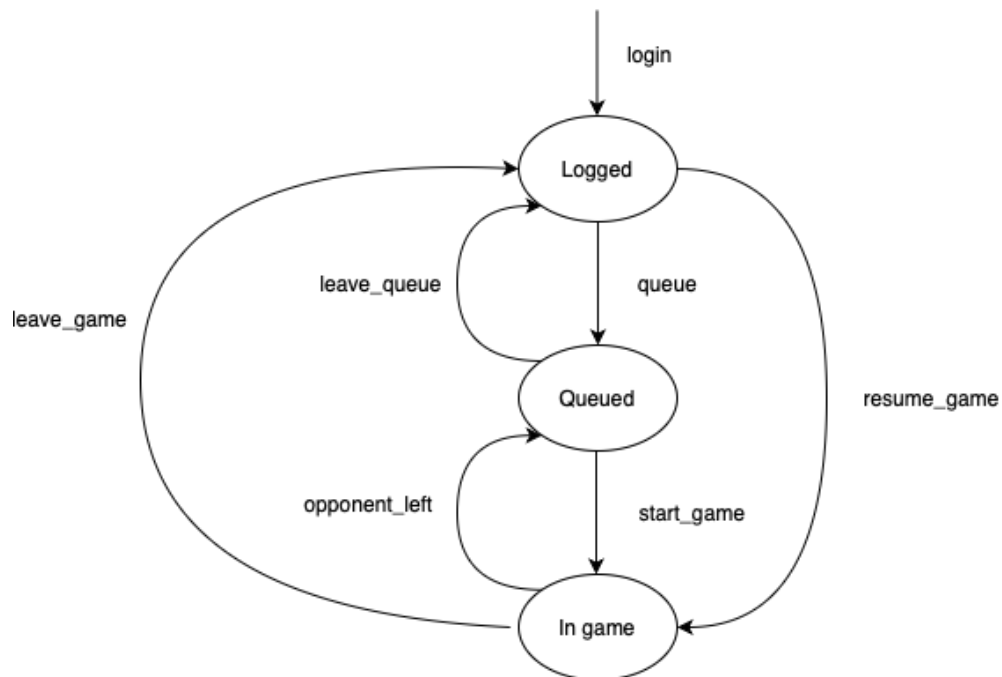
Server po poslání **ping** zprávy zjistí, jestli na předchozí zprávu klient odpověděl. Podle toho pošle oponentovi status připojení zprávou **opponent_connection** | int *state*, aby každý klient znal status svého protihráče. Proměnná *state* může nabývat hodnot 0 (připojení v pořádku, odpovídá na **ping**), 1 (odpovídá na **ping**, ale už dlouho neprovedl tah – v této implementaci 30 sekund) a 2 (krátkodobě nedostupný).

2.1.4 Poznámka

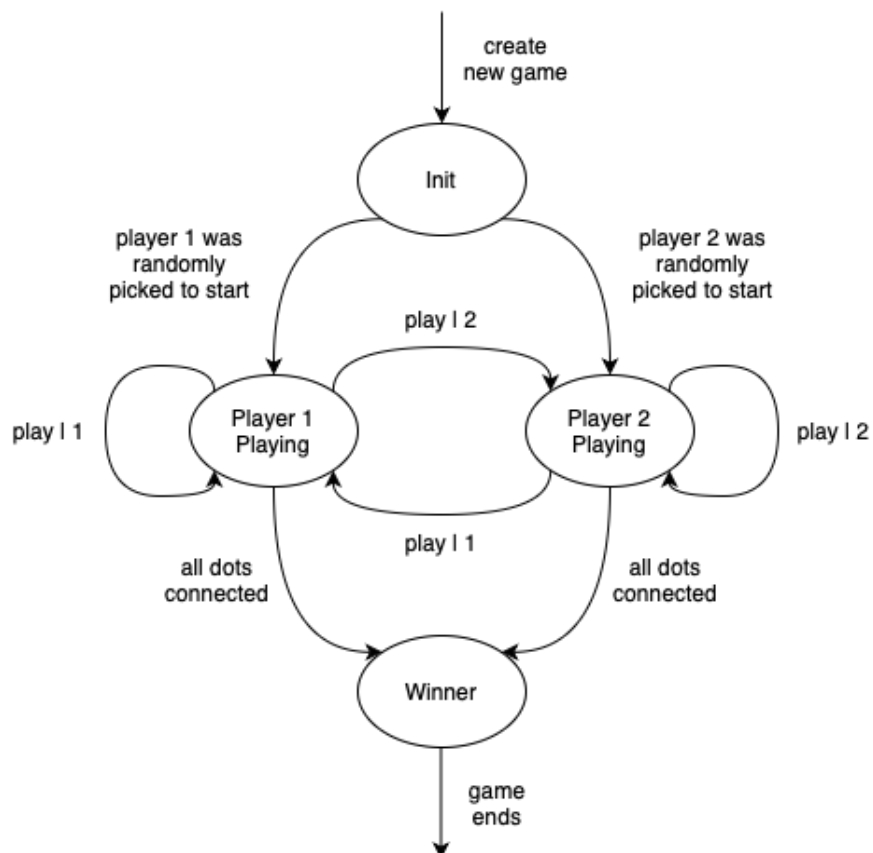
Klientská aplikace neimplementuje reakce na zprávy od serveru **unknown_command**, **invalid_coords**, **disconnected_suspicious_activity** a **disconnected_inactivity**. Tyto zprávy od serveru není možné v grafickém uživatelském rozhraní dostat, byly vytvořeny spíše pro předání informace v příkazové řádce.

2.2 Stavové diagramy

2.2.1 Stavový diagram hráče



2.2.2 Stavový diagram hry



3 Implementace

3.1 Server

3.1.1 Použitá verze prostředí

Server je z větší části napsán v programovacím jazyce C, přičemž ale využívá některé funkce a datové struktury jazyka C++. Je použit standard C++ 11.

3.1.2 Moduly programu

- **server**

Vstupní modul programu, připraví server na příjem zpráv pomocí funkce *select()*. Zajišťuje volání funkcí *parseru* pro přípravu zpráv a předání příkazu a parametrů funkci z modulu *commands*. Zároveň obstarává rozesílání zprávy ping, *opponent_connection*, volání funkce pro zařazení hráčů z fronty do hry a odpojování dlouhodobě nedostupných klientů.

- **commands**

Modul pro vykonávání příkazů podle příchozích klientských zpráv. Příkaz a parametry ve zprávě jsou předány funkci *try_running_command()*, která se postará o předání parametrů správné funkci podle zvoleného příkazu.

- **game_structures**

Modul obsahující struktury *game* a *player*. Struktura *game* má členy pro stav hry (enumerátor), dva ukazatele na hráče, dva ukazatele na pole integerů představující dokončené čtverce každého hráče, skóre obou hráčů a čas posledního tahu. Struktura *player* obsahuje členy pro stav hráče (enumerátor), socket, přezdívku, ukazatel na hru, logickou proměnnou reakce klienta na poslední zprávu ping a počet zpráv ping, na které klient v kuse neodpověděl.

- **state_machine**

Modul obsahující enumerátory pro stav hry a hráče i události hry a hráče. Také se zde nachází funkce převádějící daný stav a událost na nový stav.

- **parser**

Modul pro zpracovávání zpráv od klientů. Obsahuje funkce pro rozdělení zpráv podle znaku pro konec zpráv, a také funkce pro extrakci příkazu a parametrů ze zprávy podle znaku rozdělovače.

- **graph**
Modul datové struktury graf, která umožňuje propojování bodů (přidávání hran) a detekci smyček délky 4 (vznikají, když je dokončen čtverec).
- **config**
Modul datové struktury obsahující konfiguraci programu (především IP adresa a port, ale také počet vrcholů grafu, jak často posílat zprávu ping v sekundách, kolik zpráv ping smí klient nezodpovědět a jak dlouho má hráč na tah).
- **logger**
Modul pro zápis logů do souboru.

3.1.3 Metoda paralelizace

Server pracuje pseudoparalelně, tj. využívá funkci *select()*. Tato funkce umožňuje monitorovat množinu file deskriptorů klientských socketů a procházet pouze ty sockety, na kterých se nachází data ke čtení.

3.2 Klient

3.2.1 Použitá verze prostředí a knihoven

Klient je naprogramován v programovacím jazyce Python verze 3.10 a využívá knihovnu pro grafické uživatelské rozhraní *tkinter* 8.6.

3.2.2 Dekompozice

- **main.py**
Vstupní modul programu, odstartuje hlavní vlákno pro aktualizaci GUI a vykonávání příkazů, i další vlákno s funkcí *select()* pro příjem zpráv od serveru.
- **controller.py**
Modul pro zpracovávání zpráv od serveru. Příkaz a parametry ze zprávy jsou předány metodě *run_command()*, která vybere správnou metodu podle názvu příkazu, a dále jí předá parametry. Přepíná mezi jednotlivými grafickými rámci, aby ten aktivní byl vždy nahoře.
- **playground.py**
Modul pro správu herní plochy, obsahuje canvas pro vykreslování tlačítek (bodů), úsečkami mezi nimi a vytvořenými čtverci.
- **graph.py**
Modul s datovou strukturou graf, která spravuje propojení tlačítek (bodů).

- **login_frame.py**
Modul s rámcem pro připojení na server. Zadává se zde IP adresa, port a uživatelské jméno.
- **enqueue_frame.py**
Modul s rámcem pro zařazení do fronty.
- **inqueue_frame.py**
Modul s rámcem, ve kterém je možné odejít z fronty.
- **game_frame.py**
Modul s rámcem, v němž je hrána samotná hra. Obsahuje instanci třídy *playground*.
- **message.py**
Modul obsahující třídy *MessageDecoder* (dekódovaná zpráva rozložená na příkaz a parametry) a *MessageEncoder* (zakódovaná zpráva vytvořená z příkazu a parametrů).
- **button.py**
Modul se třídou pro speciální propojovaná tlačítka, představující body.
- **constants.py**
Modul obsahující konstanty pro zprávy a dále např. velikosti bufferů či barvy použité v GUI.

3.2.3 Metoda paralelizace

Klient běží na dvou vláknech. Hlavní vlákno vykonává příkazy ve frontě a aktualizuje grafické uživatelské rozhraní. Druhé vlákno využívá funkci *select()*, stejně jako server. Zde přijímá zprávy od serveru a dává je do fronty pro hlavní vlákno. Pokud se jedná o zprávu ping, rovnou na ni odpovídá bez zařazení do fronty.

4 Uživatelská příručka

4.1 Server

Pro přeložení souborů serveru je potřebný nástroj *make* a kompilátor *g++*. Stačí pouze v adresáři se soubory programu spustit příkaz *make* a následně spustit vytvořený binární soubor s názvem *server*. IP adresu a port je možné nastavit buď jako parametry při spouštění programu (v tomto pořadí), nebo v konfiguračním souboru *config.txt*, kde na první řádce bude IP adresa a na druhé číslo portu.

4.2 Klient

Pro běh klienta je potřebné mít nainstalovaný Python verze 3.10 a *tkinter* verze 8.6. Klient se spustí příkazem *python3 main.py* ve složce se soubory programu.

5 Závěr

Byla vytvořena tahová síťová hra pro více hráčů nazvaná Dots & Boxes. Vytvořený klient i server jsou stabilní a není nutné je mezi hrami restartovat. Obě aplikace umožňují zotavení po výpadku protistrany a ošetřují nevalidní zprávy. Server je schopen paralelně obsluhovat více klientů. Klient je určen pro operační systémy GNU/Linux a Windows, server pracuje pouze na GNU/Linux.