Semestrální práce z KIV/UPS

Dots and Boxes

Vladan Trhlík

28. prosince 2024

1 Popis hry

Hra Dots and Boxes je strategická tahová hra pro dva hráče. Herní pole tvoří čtvercová mřížka teček (v tomto případě 4 × 4), a cílem hry je propojit sousední tečky tak, aby vznikl čtverec. Hráč, který uzavře čtverec, získá bod a pokračuje dalším tahem. Hra končí, jakmile jsou všechny čtverce uzavřeny, a vítězem se stává hráč, který získal více bodů.

2 Protokol

2.1 Základní popis

Komunikace mezi klientem a serverem je zajištěna pomocí TCP protokolu. Při posílání zpráv mezi klienty a serverem se jednotlivé části zpráv oddělují znakem '`|`' a ukončovací znak je '`\n`'. Jména hráčů a názvy her jsou omezena na velká a malá písmena a znak '`_`'. Pokud není zpráva ve správném formátu, nebo zadané parametry nedávají v aktuálním kontextu smysl, odpovědí je zpráva `ERR<n>`, kde `n` je číslo jedné z chybových zpráv:

- `1` – invalid
- `2` – name already exists
- `3` – player not on turn
- `4` – player not in game
- `5` – already in game
- `6` – max limit exceeded

2.2 Zprávy

Přihlášení

- **Client:** `LOGIN|<name>`
- **Server:** `OK / ERR<1/2>`
- `<name>`: přihlašovací jméno hráče

Načtení všech her

- **Client:** `LOAD`
- **Server:** `OK|<game1-name>|<game2-name>|...|<gameN-name>`
- `<game-name>`: název I-té hry

Vytvoření hry

- **Client:** `CREATE|<game-name>`
- **Server:** `OK / ERR<1/2>`
- `<game-name>`: název hry

Připojení do hry

- **Client:** `JOIN|<game-name>`
- **Server:** `OK|<op-name> / ERR<1/5>`
- `<game-name>`: název hry, `<op-name>`: jméno oponenta

Oponent se připojil

- **Server:** `OP_JOIN|<op-name>`
- **Client:** `OK / ERR<4/5>`
- `<op-name>`: jméno oponenta

Oponent se odpojil

- **Server:** `OP_LEAVE`
- **Client:** `OK / ERR<4>`

Odpojení od hry

- **Client:** `LEAVE`
- **Server:** `OK / ERR<4>`

Tah

- **Server, Client:** `TURN|<X>|<Y>`
- **Client, Server:** `OK / ERR<1/3>`
- `<X>`, `<Y>`: pozice tahu
- Pokud přijde zpráva ze severu, jedná se o tah oponenta, pokud od klienta, posílají se data jeho oponentovi.

Obsazení čtverce

- **Server:** `(OP_)ACQ|<X>|<Y>(|<X2>|<Y2>)`
- **Client:** `OK`
- `<X>`, `<Y>`, `<X2>`, `<Y2>`: pozice čtverce
- `OP_ACQ` informuje o obsazení oponentem, `ACQ` hráčem, který zprávu přijímá.
- Při propojení dvou teček může dojít k obsazení jednoho nebo dvou čtverců – podle toho se pošle počet pozic čtverců.

Hráč je na tahu

- **Server:** `ON_TURN`
- **Client:** `OK`

Oponent je na tahu

- **Server:** `OP_TURN`
- **Client:** `OK`

Konec hry

- **Server:** `END|WIN/LOSE`
- **Client:** `OK`

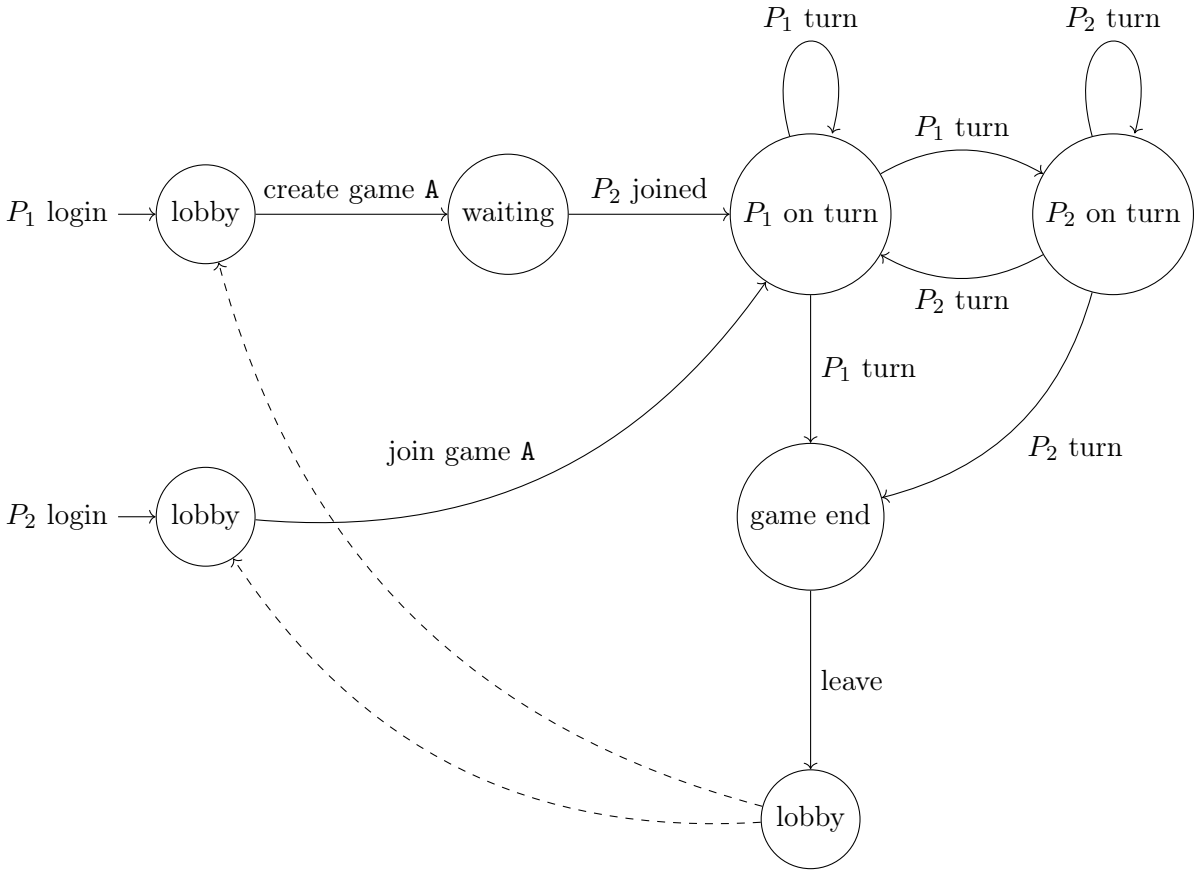
Ping

- **Client:** `PING`
- **Server:** `PONG`

Synchronizace hry

- **Client:** `SYNC`
- **Server:** `OK|<w>|<h>|<stick-data>|<square-data> / ERR<4>`
- `<w>`, `<h>`: velikost hracího pole, `<stick-data>`: data o propojených tečkách, `<square-data>`: data o obsazených čtvercích
- data o propojených tečkách a obsazených čtvercích jsou řetězce z čísel 0,1,2
  - 0 = nepropojeno / neobsazeno
  - 1 = hráč který žádá o `SYNC`
  - 2 = oponent

3 Diagram



4 Popis implementace

4.1 Server

Server je napsán jazyce C za použití základních knihoven. Je rozdělen do několika modulů:

- `game` – funkce pro vytváření hry, ověřování správnosti tahů apod.
- `handlers` – obslužné funkce pro jednotlivé typy zpráv
- `server` – funkce pro spuštění serveru a obsluhy klientů
- `utils` – užitečné funkce pro odesílání zpráv a ověřování přechodů mezi stavy klientů
- `main` – vstupní bod programu

Všechny struktury jsou definovány ve `structs.h` – obsahují strukturu `Game`, `Server` a `Player`, které slouží k uložení dat o jednlivých hráčích a hrách. Kromě toho se zde nachází i typy zpráv, výčty možných stavů a událostí, které jsou využity v přechodovém automatu (v `utils.c`).

Celý program běži jako jeden proces a k obsluze klientů využívá `fd_set`.

Pokud server detekuje zprávy ve špatném formátu, zvedne se čítač nevalidních zpráv u daného hráče. Pokud překročí 10 nevalidních zpráv, je hráč odpojen.

Při přerušení spojení se hráč může připojit zpět do hry po přihlášení pod stejným jménem.

Pro konfiguraci serveru lze použít soubor `config.txt`, který obsahuje data o maximálních počtech hráčů a her, IP adrese a portu.

4.2 Client

Client je napsán v jazyce Python za použití základních knihoven, grafické knihovny `pygame` a GUI knihovny `pygame_gui`. Program je rozdělen do scén, mezi kterými se přepíná. Při implemetaci byla snaha se držet MVC architektury, ale ta byla kvůli jednoduchosti většiny scén zachována jen v samotné scéně hry.

Každá scéna dědí třídu `Scene`, která obsahuje základní metody na zpracování vstupu od uživatele a správného vykreslení všech komponent. Měnění mezi scénenami zajišťuje jednoduchá třída `SceneManager`.

Pro komunikaci se serverem je zde třída `Socket`, která za využitím knihovny `socket` zajišťuje komunikaci se serverem, frontu přijatých zpráv a metody zajišťující opětovné připojení při přerušení spojení. Tato třída běži v druhém vlákně pro zajištění plynulého chodu uživatelského rozhraní.