Dokumentace k semestrální práci z předmětu KIV/UPS

Téma práce: Hra Inverzní lodě

Autor: Milan Vlachovský

Obsah

1. Úvod

Cílem tohoto projektu bylo vytvořit elementární hru pro více hráčů používající síťovou komunikaci se serverovou částí v nízkoúrovňovém programovacím jazyce a klientskou částí v programovacím libovolném jazyce. Autor projektu zvolil vlastní variantu hry Lodě s upravenými pravidly nazvanou *„Inverse Battleships“*. Stejně jako její předloha je hra určena pro 2 hráče, přičemž se hráčí střídají v tazích.

Byl zvolen protokol na bázi TCP. Jako programovací jazyk serveru byl zvolen jazyk Go, díky své rychlosti a nízkoúrovňovému přístupu k síťové komunikaci. Klientská část byla implementována v jazyce Python s využitím knihovny pygame pro správu vykreslování grafického prostředí hry.

2. Popis hry

Hra *Inverse Battleships* je variantou klasické hry Lodě, ve které se hráči snaží najít a zničit všechny lodě protivníka. V této variantě hráči sdílí jedno pole 9x9 a každý dostane na začátku přiřazenou jednu loď. Následně se hráči střídají v tazích, kdy každý hráč se může pokusit vykonat akci na prázdné políčko. Mohou nastat tři situace:

- Hráč zkusí akci na prázdné políčko, ve kterém se nachází nikým nezískaná loď. V tomto případě hráč loď získává a získává body.
- Hráč zkusí akci na prázdné políčko, ve kterém se nachází protivníkova loď. V tomto případě protivník o loď přichází, je zničena, hráč získává body a protivník ztrácí body.

- Hráč zkusí akci na políčko, na kterém se nic nenachází. V tomto případě hráč nezískává nic.

Hra končí, když jeden z hráčů ztratí všechny lodě. Vítězem je přeživší hráč. Body jsou pouze pro statistické účely a nemají vliv na průběh hry.

3. Popis síťového protokolu

4. Architektura systému

Požadavky

Projekt byl vyvíjen s použitím následujících technologií:

- Python 3.12
  - pygame 2.6.0
  - pydantic 2.8.2
  - typing-extensions 4.12.2
  - termcolor 2.5.0
  - pyinstaller 6.11.1
- Go 1.23

> Za použití zmiňovaných technologií by měl být projekt bez problémů spustitelný. Spouštění na starších verzích nebylo testováno a nemusí fungovat správně.

5. Návod na zprovoznění

Pro sestavení celého projektu byly vytvořeny soubory *Makefile* a *Makefile.win*, které obsahují instrukce pro sestavení projektu na Unixových a Windows OS. Pro sestavení projektu na Unixových OS stačí spustit příkaz:

```
make
```

a pro Windows OS stačí spustit příkaz:

```
make -f Makefile.win
```

Předpokládá se, že je nainstalován program `make`; na Windows je možné použít například make z chocolatey, či jiné alternativy.

Skript sestaví spustitelné soubory ve složce *client/bin/* pro klientskou část projektu a ve složce *server/bin/* pro serverovou část projektu. Spustitelné soubory jsou pojmenovány *client* a *server*, případně na Windows *client.exe* a *server.exe*. Stačí pouze z kořenové složky projektu na Unix OS spustit příkaz:

```
make
```

Nebo na Windows OS:

```
make -f Makefile.win
```

> Jelikož je klientská část implementována v jazyce Python, je možné ji spustit i bez sestavení. Stačí spustit soubor *client/src/main.py* v Python virtuálním prostředí s nainstalovanými závislostmi ze souboru *requirements.txt*. Spustitelné soubory pro klientskou část byly vytvořeny pomocí knihovny *pyinstaller* a jejich úspěšnost překladu bývá závislá na operačním systému a verzi Pythonu.

> Na základě standardu PEP 394 počítají soubory *Makefile* a *Makefile.win* s tím, že Python rozkaz pod Unixem je `python3` a pod Windows je `python`. V případě odlišného nastavení je nutné soubory upravit.

5.1 Klientská část

Klientská část je napsaná v jazyce Python, tedy kód je standardně interpretován řádku po řádce pomocí Python interpreteru. Pro spuštění klientské části je tedy nutné mít nainstalovaný Python 3.12 a nainstalované závislosti ze souboru *requirements.txt*.

Spuštění krok za krokem

Autor dopořučuje vytvořit si virtuální prostředí a nainstalovat závislosti pomocí následujících příkazů:

> Příkazy jsou pro Unix OS, pro Windows OS je nutné je přizpůsobit.

```
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
cd client/
```

Následně je možné spustit klienta pomocí příkazu:

```
python ./src/main.py
```

Je také možné spustit klienta se specifickým nastavením a nastavením logování pomocí:

```
python ./src/main.py -c ./cfg/debug_cfg.json -l
./cfg/debug_loggers_cfg.json
```

> Standardně se předpokládá spouštění ze složky *client/*

Sestavení spustitelného souboru

Kvůli charakteru jazyka není možné bez externích knihoven vytvořit spustitelný soubor. Autor proto zvolil cestu sestavení pomocí knihovny *pyinstaller*. Pro sestavení spustitelného souboru je nutné mít

nainstalovaný *pyinstaller*.

Pro sestavení spustitelného souboru stačí pouze spustit z kořenové složky na Unix OS příkaz:

```
make client
```

Nebo na Windows OS:

```
make -f Makefile.win client
```

Pro manuální sestavení lze použít kód z Makefile souborů.

5.2 Serverová část

Serverová část je napsaná v jazyce Go. Pro sestavení serverové části je nutné mít nainstalovaný Go 1.23 či novější. Sestavení opět probíhá za pomocí souborů *Makefile* a *Makefile.win*. Na Unix OS stačí spustit příkaz:

```
make server
```

Na Windows OS:

```
make -f Makefile.win server
```

Make sestaví spustilený soubor ve složce *server/bin/* s názvem *server*.

Spuštění serveru

Pro spuštění serveru je nutné zadat jako parametr buď IP adresu, na které bude server naslouchat (parametr *-a*), nebo specifikovat konfigurační soubor (parametr *-c*). Pro spuštění serveru na 127.0.0.1 a na portu 8080 stačí zadat:

> Příkazy jsou pro Unix OS, pro Windows OS je nutné je přizpůsobit.

```
./server/bin/server -a "127.0.0.1:8080"
```

> Při volbě IP adresy a portu je vhodné zjistit, zda je adresa a port dostupný a nejsou blokovány firewallem apod.

6. Struktura projektu

Projekt je rozdělen následovně:

- *Kořenová složka* — Obsahuje soubory pro sestavení projektu, složky *client* a *server*, složku *docs* s dokumentací a soubor *requirements.txt* s definicemi Python závislostí.
    - *client/* — Obsahuje celou klientskou část projektu včetně konfiguračních souborů, použitých textových a obrazových zdrojů a programátorské referenční dokumentace.
    - *server/* — Obsahuje celou serverovou část projektu včetně konfiguračních souborů a programátorské referenční dokumentace.

Kořenová složka

- *Makefile* — Soubor pro sestavení projektu na Unix OS.
- *Makefile.win* — Soubor pro sestavení projektu na Windows OS.
- *client/* — Složka obsahující kód s klientskou částí aplikace.
    - *client/Doxyfile* — Soubor s konfigurací pro Doxygen.
    - *client/cfg/* — Složka obsahující konfigurační soubory klientské části.
        - *client/cfg/debug_cfg.json* — Soubor s konfigurací pro debugování.
        - *client/cfg/debug_loggers_cfg.json* — Soubo s konfigurací logování pro debugování.
        - *client/cfg/debug_loggers_cfg_win.json* — Soubor s konfigurací logování pro debugování na Windows.
        - *client/cfg/default_config.json* — Soubor s výchozí konfigurací.
        - *client/cfg/default_user_config.json* — Soubor s výchozí konfigurací nového uživatele.
        - *client/cfg/loggers_config.json* — Soubor s konfigurací logování.
        - *client/cfg/users/* — Složka obsahující konfigurace uživatelů.
    - *client/docs/* — Složka obsahující dokumentaci kódu klientské části.
    - *client/res/* — Složka obsahující zdroje pro klientskou část.
        - *client/res/colors.json* — Soubor s definicemi barev použitých v GUI klienta.
        - *client/res/img/* — Složka obsahující obrázky použité ve GUI klienta.
        - *client/res/strings.json* — Soubor s definicemi textoých řetězců použitých v GUI klienta.
    - *client/src/* — Složka obsahující zdrojové kódy klientské části.
        - *client/src/const/* — Složka obsahující konstanty.
            - *client/src/const/exit_codes.py* — Soubor s konstantami pro návratové kódy.
            - *client/src/const/loggers.py* — Soubor s konstantami pro logování.
            - *client/src/const/paths.py* — Soubor s cestovými konstantami.
            - *client/src/const/typedefs.py* — Soubo s definicemi požívaných objektů v kódu klienta.
        - *client/src/game/* — Složka zastřešující kód pro správu hry.
            - *client/src/game/connection_manager.py* — Soubor s kódem pro správu spojení se serverem.
            - *client/src/game/ib_game.py* — Soubor s kódem spravujícím hru.
            - *client/src/game/ib_game_state.py* — Soubor s kódem pro stav hry.

        - *client/src/graphics/* — Složka obsahující kód GUI klienta.
            - *client/src/graphics/game_session.py* — Soubor s kódem pro GUI herní session.
            - *client/src/graphics/menus/* — Složka obsahující kód GUI menu.
                - *client/src/graphics/menus/info_screen.py* — Soubor s kódem GUI informační obrazovku.
                - *client/src/graphics/menus/input_menu.py* — Soubor s kódem GUI vstupní menu.
                - *client/src/graphics/menus/lobby_select.py* — Soubor s kódem GUI výběr lobby.
                - *client/src/graphics/menus/primitives.py* — Soubor s kódem pro primitiva GUI.
                - *client/src/graphics/menus/select_menu.py* — Soubor s kódem pro výběrové menu.
                - *client/src/graphics/menus/settings_menu.py* — Soubor s kódem pro nastavení.
            - *client/src/graphics/viewport.py* — Soubor s kódem představujícím viewport pro zobrazování libovolného GUI.
        - *client/src/main.py* — Soubor s kódem pro spuštění klienta.
        - *client/src/util/* — Složka obsahující pomocné metody.
            - *client/src/util/assets_loader.py* — Soubor s kódem pro načítání zdrojů (obrázky, zvuky, ...).
            - *client/src/util/etc.py* — Soubor s vedlejšími pomocnými metodami.
            - *client/src/util/file.py* — Soubor s pomocnými metodami pro práci se soubory.
            - *client/src/util/generic_client.py* — Soubor s kódem představující generického klienta (založeném na socketech).
            - *client/src/util/graphics.py* — Soubor s pomocnými metodami pro práci s grafikou.
            - *client/src/util/init_setup.py* — Soubor s kódem pro inicializaci klienta.
            - *client/src/util/input_validators.py* — Soubor s validátory vstupu.
            - *client/src/util/loggers.py* — Soubor s kódem pro příspůsobené logování.
            - *client/src/util/path.py* — Soubor s pomocnými metodami pro práci s cestami.
- *docs/* — Složka obsahující dokumentaci.
    - *docs/doc.md* a *docs/doc.pdf* — Tento dokument ve formátu Markdown a PDF.
    - *docs/client_ref.html* — Odkaz na dokumentaci klientské části.
    - *docs/server_ref.html* — Odkaz na dokumentaci serverové části.
- *requirements.txt* — Soubor s definicemi Python závislostí.
- *server/* — Složka obsahující kód s serverovou částí aplikace.
    - *server/cfg/* — Složka obsahující konfigurační soubory serverové části.
    - *server/docs/* — Složka obsahující dokumentaci kódu serverové části.

- *server/src/* — Složka obsahující zdrojové kódy serverové části.
    - *server/src/const/* — Složka obsahující konstanty.
        - *server/src/const/const_file/* — Složka obsahující konstanty pro práci se soubory.
            - *server/src/const/const_file/const_file.go* — Soubor s konstantami pro práci se soubory.
        - *server/src/const/custom_errors/* — Složka obsahující definice chyb.
            - *server/src/const/custom_errors/custom_errors.go* — Soubor s definicemi chyb.
        - *server/src/const/exit_codes/* — Složka obsahující konstanty pro návratové kódy.
            - *server/src/const/exit_codes/exit_codes.go* — Soubor s konstantami pro návratové kódy.
        - *server/src/const/msg/* — Složka obsahující definice uživatelských zpráv.
            - *server/src/const/msg/msg.go* — Soubor s definicemi uživatelských zpráv.
        - *server/src/const/protocol/* — Složka obsahující definice síťového protokolu.
            - *server/src/const/protocol/server_communication.go* — Soubor s definicemi síťového protokolu.
    - *server/src/go.mod* — Soubor s definicí modulů Go.
    - *server/src/logging/* — Složka obsahující kód pro logování.
        - *server/src/logging/logging.go* — Soubor s kódem pro logování.
    - *server/src/main.go* — Soubor s kódem pro spuštění serveru.
    - *server/src/server/* — Složka obsahující kód pro správu serveru.
        - *server/src/server/connection_manager.go* — Soubor s kódem pro správu spojení.
        - *server/src/server/client_manager.go* — Soubor s kódem pro správu klientů.
    - *server/src/util/* — Složka obsahující pomocné funkce.
        - *server/src/util/arg_parser/* — Složka obsahující kód pro parsování argumentů.
            - *server/src/util/arg_parser/arg_parser.go* — Soubor s kódem pro parsování argumentů.
        - *server/src/util/cmd_validator/* — Složka obsahující kód pro validaci síťových příkazů.
            - *server/src/util/cmd_validator/cmd_validator.go* — Soubor s kódem pro validaci síťových příkazů.
        - *server/src/util/msg_parser/* — Složka obsahující kód pro parsování zpráv.
            - *server/src/util/msg_parser/msg_parser.go* — Soubor s kódem pro parsování zpráv.
        - *server/src/util/util.go* — Soubor s pomocnými funkcemi.

7. Popis implementace

7.1 Klientská část

Klientská část aplikace byla implementována v jazyce Python s využití knihovny **pygame** pro grafické rozhraní. Hlavní součásti klienta jsou rozděleny do modulů podle jejich funkce.

> Veškerý kód se nachází pod složkou *client/src/*.

Moduly klientské části

1. **game**
    - *connection_manager.py*: Spravuje připojení klienta k serveru pomocí socketů.
    - *ib_game.py*: Obsahuje hlavní logiku hry, jako je zpracování tahů a synchronizace herního stavu se serverem.
    - *ib_game_state.py*: Reprezentuje stav hry a poskytuje API pro manipulaci s herními daty.
2. **graphics**
    - *game_session.py*: Zajišťuje vykreslování herního prostředí a interakci uživatele s hrou.
    - *menus*: Obsahuje moduly pro tvorbu a správu herních menu, jako jsou vstupní obrazovky, lobby nebo nastavení.
3. **util**
    - *assets_loader.py*: Zajišťuje načítání grafických a zvukových souborů.
    - *loggers.py*: Implementuje vlastní logování pro snadněší diagnostiku a ladění.
    - *init_setup.py*: Zajišťuje inicializaci klientské aplikace.

Rozvrstvení klientské aplikace

Klientská část je navržena jako modulární aplikace s jasným rozdělením na:

- **Prezentaci (UI)**: Moduly v *graphics*.
- **Logiku hry**: Moduly v *game*.
- **Podpůrné funkce**: Moduly v *util*.

Použité knihovny klientské části

- **pygame 2.6.0**: Pro vykreslování grafického rozhraní.
- **pydantic 2.8.2**: Pro validaci dat.
- **termcolor 2.5.0**: Pro barevné logování v terminálu.
- **pyinstaller 6.11.1**: Pro sestavení spustitelných souborů.

Paralelizace klienta

Klientská část využívá moduly knihovny **asyncio** pro správu asynchronní komunikace se serverem, což umožňuje zpracovávat zprávy od serveru souběžně s vykreslováním a zpracováním vstupů uživatele.

7.2 Serverová část

Serverová část byla implementována v jazyce Go pro svou rychlost a efektivní práci s paralelizací. Server je navržen jako modulární aplikace s těmito hlavními komponentami:

Moduly serverové části

1. **server**
    - *connection_manager.go*: Spravuje připojení klientů a jejich komunikaci se serverem.
    - *client_manager.go*: Uchovává informace o připojených klientech a jejich stavech.
2. **util**
    - *arg_parser.go*: Zajišťuje parsování argumentů při spouštění serveru.
    - *cmd_validator.go*: Validuje příkazy přijaté od klientů.
    - *msg_parser.go*: Zajišťuje správné formátování zpráv při odesílání a příjmu.
3. **const**
    - *protocol/server_communication.go*: Obsahuje definice protokolu a formátu zpráv mezi klientem a serverem.

Rozvrstvení serverové aplikace

Server je rozdělen do těchto vrstev:

- **Komunikační vrstva**: Moduly *connection_manager* a *client_manager*.
- **Aplikační logika**: Validace a zpracování příkazů v *util*.
- **Konfigurace**: Moduly v *const*.

Použité knihovny serverové části

- **Go 1.23**: Základní runtime.
- Standardní knihovna Go pro práci s net (TCP komunikace).

Paralelizace

Server využívá **gorutiny** pro souběžné zpracování klientských požadavků. Pro synchronizaci dat jsou používány **mutexy** a kanály (*channels*).

Tento popis poskytuje přehled o implementaci obou částí aplikace a může být dle potřeby rozšířen o další detaily.

8. Závěr