

## Dokumentace k projektu

### 1. Základní popis hry

Jedná se o Piškvorky pro dva hráče. Hráči se připojují k serveru, který umožňuje současné vedení více her. Každý hráč zadává své tahy pomocí souřadnic (x, y) na hrací desce. Hra končí výhrou jednoho hráče, remízou nebo ukončením hry ze strany hráče.

### 2. Popis protokolu

#### Obecná charakteristika

- Typ: Textový, nešifrovaný, běžící nad TCP.
- Komunikace: Klient–Server.
- Kódování znaků: UTF-8.
- Formát: Každá zpráva je ukončena znakem nového řádku (`\n`).

#### Formát zpráv

- Zpráva: `<PŘÍKAZ>[PARAMETR1];[PARAMETR2] ...\n` (odděleny pomocí “;”)

#### Přenášené datové typy

- String: Text bez mezer, víceslovné výrazy nahrazují mezery podtržítkem (`_`).
- Integer: Celé číslo (např. souřadnice, ID hráče).
- Status code: Celé číslo reprezentující stav nebo chybu.

#### Omezení a validace

- Uživatelská jména: Max. délka 14 znaků, pouze alfanumerické znaky a `_``.
- Souřadnice: Hodnoty odpovídající rozsahu herního pole.
- Maximální počet her: Nastavitelný při spuštění serveru.
- Kontrola zpráv: Server i klient validují syntaxi a semantiku.

#### Přehled příkazů

`RECONNECT;<player name>;`

Znovu připojí hráče do hry

`YOUR_TURN;<row number>;<column number>;`

Pošle potvrzení tahu hráče

`OPPONENT_TURN;<row number>;<column number>;`

Pošle potvrzení tahu protihráče

`STATUS;<zpráva>;`

Pošle hráči status zprávu

PING;

Pošle hráči kontrolní ping

NAME;<jméno>;

Přihlásí uživatele pod jménem

WAITING\_FOR\_GAME;

Zařadí hráče jako čekajícího na hru

EXIT;

Odpojí hráče

ACK;

Potvrzení hráče po ping

GAME\_OVER;

Upozornění hráčům, že hra skončila

TURN;<row number>;<column number>;

Tah co hráč pošle na server

REMATCH;

Pošle hráči žádost o novou hru

### 3. Implementace klienta a serveru

#### Moduly a třídy

Server:

- `Server.cpp`: Hlavní logika serveru.
- `Responder.cpp`: Obsluha příchozích zpráv.
- `Game.cpp`: Logika hry.
- `Logger.cpp`: Logování událostí.
- `Main.cpp`: Hlavní třída.
- `GameAdmin.cpp`: Řídí hru.
- `Makefile`: Pro kompilaci serveru.

Klient:

- Složka resources: Jsou v ní JavaFX knihovny.
- Pom.xml: Slouží pro kompilaci klienta
- Složka src: Zdrojové kódy

- Enum: Obsahuje enumy, jako například stavy a zprávy
- Logic: Controller.java, který ovládá klienta
- Main: App.java, hlavní třída
- Network: Obsahuje třídy pro komunikaci se serverem
- Object: Obsahuje objekty, které reprezentují například hráče, nebo souřadnice
- Scene: Obsahuje třídy, které slouží pro vykreslování GUI.

#### Rozvrstvení aplikace

- Server:
  - Vrstvy pro příjem, zpracování a odpovídání na zprávy.
- Klient:
  - Vrstvy pro komunikaci se serverem, zpracování příkazů a uživatelské rozhraní.

#### Použité knihovny

- Server: C++ (gcc), knihovny STL.
- Klient: Java (JDK 11+), Maven pro správu závislostí.

#### Paralelizace

- Server: Použití vláken pro obsluhu více klientů.
- Klient: Sekvenční zpracování událostí.

### 4. Požadavky na překlad a spuštění

- Server: GCC 11+, Makefile pro překlad.
- Klient: Java JDK 11+, Maven.

#### Postup překladu

- Server: Spustíte `make` v adresáři serveru. Následně jde spustit pomocí příkazu:

`./server <ip adresa> <port> <maximální počet her>`

- Klient: Spustí se rovnou v adresáři klienta pomocí příkazu:

`mvn clean compile exec:java -D1="<adresa>" -D2="<port>"`



