



SEMESTRÁLNÍ PRÁCE Z PŘEDMĚTU KIV/UPS

# TAHOVÁ MULTIPLAYEROVÁ HRA NA ZPŮSOB KRIS KROS

STANISLAV KRÁL

A17B0260P

ZÁPADOČESKÁ UNIVERZITA V PLZNI  
FAKULTA APLIKOVANÝCH VĚD

# Obsah

<b>1</b>	<b>Popis hry Kris Kros</b>	<b>2</b>
<b>2</b>	<b>Analýza</b>	<b>3</b>
2.1	Návrh protokolu . . . . .	3
2.2	Výběr programovacího jazyku pro server . . . . .	3
<b>3</b>	<b>Popis protokolu</b>	<b>4</b>
3.1	Formát zpráv . . . . .	4
<b>4</b>	<b>Implementace</b>	<b>5</b>
4.1	Typy příchozích zpráv na serveru . . . . .	5
4.1.1	Atributy zpráv . . . . .	5
4.2	Typy příchozích zpráv na klientovi . . . . .	7
4.2.1	Atributy zpráv . . . . .	8
4.3	Přenášené struktury . . . . .	12
4.3.1	Struktura <code>Tile</code> . . . . .	12
4.3.2	Struktura <code>Letter</code> . . . . .	13
4.3.3	Struktura <code>Player</code> . . . . .	13
4.3.4	Struktura <code>User</code> . . . . .	14
4.3.5	Struktura <code>Lobby</code> . . . . .	15
4.4	Chybové kódy . . . . .	15
4.5	Popis serveru . . . . .	16
4.5.1	Ošetření správné posloupnosti zpráv . . . . .	16
4.5.2	Struktura modulů . . . . .	17
4.5.3	<i>Keepalive</i> zprávy . . . . .	18
4.5.4	Nedodržování protokolu . . . . .	18
4.5.5	Použité knihovny . . . . .	18
4.6	Popis klienta . . . . .	18
4.6.1	Struktura modulů . . . . .	19
4.6.2	Nedodržování protokolu . . . . .	19
4.6.3	Použité knihovny . . . . .	19
<b>5</b>	<b>Překlad a spuštění aplikace</b>	<b>20</b>
5.1	Server . . . . .	20
5.2	Klient . . . . .	20
<b>6</b>	<b>Závěr</b>	<b>21</b>

# 1 Popis hry Kris Kros

Jedná se o deskovou hru pro 2-4 hráče inspirovanou hrou Scrabble. Základním principem hry je skládat slova na herní desku z písmenek, která byla hráči rozdána. Hráči pokládají písmena na herní desku, kde buď v horizontálním vertikálním směru vznikají slova. Každé písmenko je ohodnoceno počtem bodů, které hráč za jeho položení dostane. Na konci každého kola se sečtou body písmenek všech slov, která v daném kole vznikla. Některá políčka na desce mohou bonifikovat celkové ohodnocení slova.

## 2 Analýza

### 2.1 Návrh protokolu

Protokol je třeba navrhnout podle charakteru aplikace, ve které bude použit. V případě hry Kris Kros jsou zprávy ve většině případů relativně jednoduché, ale například při aktualizaci herní desky, kdy je třeba poslat všem hráčům seznam ovlivněných políček, by taková datová část zprávy mohla být složitá a vlastní formát dat komplikovaný. To samé platí pro případ regenerace stavu klienta po výpadku připojení. V tomto případě jsou totiž data v datové části zprávy vrstvená a ideálně i popsána nějakým identifikátorem. Návrh takového vlastního formátu by byl zbytečný, jelikož v dnešní době existuje mnoho standardizovaných formátů. V úvahu tedy připadá například XML nebo JSON.

### 2.2 Výběr programovacího jazyku pro server

Při použití standardizovaného formátu dat v datové části zpráv protokolu je třeba vybírat programovací jazyk tak, aby ve standardní knihovně obsahoval parser takového formátu a zároveň byl nízkoúrovňový, jak je stanoveno v zadání této semestrální práce. V úvahu tedy připadá programovací jazyk **Golang**, který má oproti jiným nízkoúrovňovým jazykům tu výhodu, že při jeho použití není třeba ručně uvolňovat paměť, jelikož pro automatickou správu paměti používá *garbage collector*. Další výhodou tohoto jazyka je jednoduchá paralelizace pomocí konstrukce *goroutine*.

## 3 Popis protokolu

Protokol je **textový** a jeho zprávy mají speciální formát.

### 3.1 Formát zpráv

Každá zpráva musí začínat řídicím znakem \$. Po počátečním znaku následuje délka datové části zprávy v bytech. Délka zprávy je přirozené nenulové číslo. Po délce zprávy následuje oddělovací znak #, za kterým se nachází přirozené číslo reprezentující typ zprávy. Typ zprávy je opět následován oddělovacím znakem #, za kterým je umístěné přirozené číslo reprezentující identifikátor zprávy. Za identifikátorem zprávy se nachází poslední oddělovací znak #, který odděluje identifikátor a datovou část zprávy. Datová část je ve formátu **JSON** a nesmí obsahovat neescapovaný řídicí znak \$. Escapování tohoto řídicího znaku se realizuje tím, že se před znak umístí znak \.

---

```
$19#7#1#{"name" : "standa"}
```

---

Ukázka zprávy 1: Zpráva protokolu, která je typu 7 a je identifikována číslem 1. Její datová část je dlouhá 19 bytů.

---

```
$16#6#4#{"ready" : true}
```

---

Ukázka zprávy 2: Zpráva protokolu, která je typu 6 a je identifikována číslem 4. Její datová část je dlouhá 16 bytů.

## 4 Implementace

### 4.1 Typy příchozích zpráv na serveru

Typ zprávy	Název zprávy	Popis zprávy
2	CreateLobby	Požadavek na vytvoření herní místnosti.
3	GetLobbies	Požadavek na získání aktuálního seznamu herních místností.
4	JoinLobby	Požadavek na vstoupení do herní místnosti.
5	LeaveLobby	Požadavek na opuštění herní místnosti.
6	PlayerReady	Požadavek na změnu připravenosti hráče.
7	UserAuthentication	Požadavek na autentizaci uživatele
8	UserLeaving	Požadavek na odpojení uživatele ze se serveru.
9	StartLobby	Požadavek na spuštění hry.
10	LetterPlaced	Požadavek na položení písmenka na herní desku.
11	LetterRemoved	Požadavek na odebrání písmenka z herní desky.
12	FinishRound	Požadavek na ukončení kola.
13	ApproveWords	Požadavek na potvrzení aktuálních slov na desce.
14	DeclineWords	Požadavek na odmítnutí aktuálních slov na desce.
15	KeepAlive	Požadavek na ověření funkčnosti spojení mezi klientem a serverem.
16	LeaveGame	Požadavek na opuštění hry.

V případě, že je daný požadavek dopadl úspěšně, tak server odpoví zprávou nechybového typu. Typ zprávy pro obecný úspěch je 701.

#### 4.1.1 Atributy zpráv

Níže jsou popsány významy atributů jednotlivých typů zpráv. Pokud se zde nějaká zpráva nevyskytuje znamená to, že nemá žádný atribut.

Tabulka 1: Atributy zprávy JoinLobby

Název atributu	Typ atributu	Význam atributu
lobby_id	int	ID místnosti, do které se chce hráč připojit

Tabulka 2: Atributy zprávy PlayerReadyToggle

Název atributu	Typ atributu	Význam atributu
ready	bool	Booleanská hodnota, která nabývá hodnoty <code>true</code> pokud je hráč připravený a <code>false</code> pokud není připravený.

Tabulka 3: Atributy zprávy UserAuthenticated

Název atributu	Typ atributu	Význam atributu
name	string	Jméno hráče, který se chce autentizovat.
reconnecting	bool	Booleanská hodnota, která nabývá hodnoty <code>true</code> pokud se hráč snaží o znovupřipojení a <code>false</code> pokud ne.

Tabulka 4: Atributy zprávy LetterPlaced

Název atributu	Typ atributu	Význam atributu
letter	struct letter	Písmeno, které chce hráč položit.
row	int	Číslo řádku, na který má být písmenko položeno.
column	int	Číslo sloupce, na který má být písmenko položeno.

Tabulka 5: Atributy zprávy LetterRemoved

Název atributu	Typ atributu	Význam atributu
row	int	Číslo řádku, ze kterého má být písmenko odebráno.
column	int	Číslo sloupce, ze kterého má být písmenko odebráno.

## 4.2 Typy příchozích zpráv na klientovi

Typ zprávy	Název zprávy	Popis zprávy
101	GetLobbies	Zpráva obsahující seznam herních místností na serveru.
103	LobbyUpdated	Zpráva oznamující změnu herní místnosti, která obsahuje aktuální stav místnosti.
105	LobbyDestroyed	Zpráva oznamující zrušení aktuální herní místnosti.
106	LobbyJoined	Zpráva oznamující úspěšné připojení do herní místnosti.
107	UserAuthenticated	Zpráva oznamující výsledek autentizace uživatele.
108	LobbyStarted	Zpráva oznamující spuštění herní místnosti.
109	GameStarted	Zpráva oznamující spuštění hry.
111	TilesUpdated	Zpráva obsahující informace o změněných políčkách herní desky a bodový stav hráče, který je momentálně na tahu.
112	RoundFinished	Zpráva oznamující ukončení aktuálního kola.
113	PlayerAcceptedRound	Zpráva oznamující to, že některý ze hráčů schválil slovíčka na herní desce.
114	NewRound	Zpráva oznamující počátek nového kola.
115	YourNewRound	Zpráva oznamující, že začalo nové kolo a uživatel je nyní na tahu
116	PlayerDeclined	Zpráva oznamující to, že některý ze hráčů zamítl slovíčka na herní desce
117	GameEnded	Zpráva oznamující konec hry, která obsahuje výslednou tabulku bodů jednotlivých hráčů.



118	AcceptResultedInNewRound	Zpráva oznamující to, že schválení slovíček uživatelem vyústilo v nové kolo.
119	PlayerConnectionChanged	Zpráva oznamující změnu připojení některého z hráčů.
120	GameStateRegeneration	Zpráva obsahující aktuální stav hry po znovupřipojení k serveru.
121	KeepAlive	Zpráva potvrzující spojení mezi klientem a serverem.
122	UserStateRegeneration	Zpráva oznamující aktuální stav uživatele po znovupřipojení k serveru.
123	FinishResultedInNextRound	Zpráva oznamující, že potvrzení kola uživatele vyústilo v nové kolo.
701	PlainSuccess	Vznešený požadavek dopadl úspěšně.

Všechny tyto zprávy se vztahují ke kontextu uživatele, kterému přišly.

#### 4.2.1 Atributy zpráv

Níže jsou popsány významy atributů jednotlivých typů zpráv. Pokud se zde nějaká zpráva nevyskytuje znamená to, že nemá žádný atribut.

Tabulka 6: Atributy zprávy PlainSuccess

Název atributu	Typ atributu	Význam atributu
content	string	Popis úspěchu akce.

Tabulka 7: Atributy zprávy GetLobbies

Název atributu	Typ atributu	Význam atributu
lobbies	[] struct Lobby	Seznam herních místností na serveru.

Tabulka 8: Atributy zprávy LobbyUpdated

Název atributu	Typ atributu	Význam atributu
lobby	struct Lobby	Aktualizovaná herní místnost.

Tabulka 9: Atributy zprávy LobbyJoined

Název atributu	Typ atributu	Význam atributu
lobby	struct Lobby	Aktualizovaná herní místnost.

Tabulka 10: Atributy zprávy UserAuthenticated

Název atributu	Typ atributu	Význam atributu
user	struct User	Uživatel, který byl vytvořen procesem autentizace.

Tabulka 11: Atributy zprávy GameStarted

Název atributu	Typ atributu	Význam atributu
players	[] struct Player	Seznam hráčů ve hře.
letters	[] struct Letter	Seznam písmenek, které má hráč k dispozici.

Tabulka 12: Atributy zprávy TilesUpdated

Název atributu	Typ atributu	Význam atributu
tiles	[] struct Tile	Seznam aktualizovaných políček herní desky.
current_player_points	int	Body hráče v aktuálním kole, který je na tahu.
current_player_total_points	int	Celkové body hráče, který je na tahu.

Tabulka 13: Atributy zprávy PlayerAcceptedRound

Název atributu	Typ atributu	Význam atributu
player_id	int	ID hráče, který schválil aktuální slova na desce.

Tabulka 14: Atributy zprávy `NewRoundResponse`

Název atributu	Typ atributu	Význam atributu
<code>active_player_id</code>	<code>int</code>	ID hráče, který je v novém kole na tahu.

Tabulka 15: Atributy zprávy `YourNewRoundResponse`

Název atributu	Typ atributu	Význam atributu
<code>letters</code>	<code>[] struct Letter</code>	Seznam písmen, která byla hráči v tomto novém kole přidělena.

Tabulka 16: Atributy zprávy `PlayerDeclinedWords`

Název atributu	Typ atributu	Význam atributu
<code>player_id</code>	<code>int</code>	ID hráče, který zamítl aktuální slova na desce.
<code>player_name</code>	<code>string</code>	Jméno hráče, který zamítl aktuální slova na desce.

Tabulka 17: Atributy zprávy `GameEndedResponse`

Název atributu	Typ atributu	Význam atributu
<code>player_points</code>	<code>map int:struct Player</code>	Mapa, kde klíčem je počet bodů a hodnotou je hráč, který tohoto zisku dosáhl.

Tabulka 18: Atributy zprávy `PlayerConnectionChanged`

Název atributu	Typ atributu	Význam atributu
<code>player_id</code>	<code>int</code>	ID hráče, jehož stav připojení se změnil.
<code>disconnected</code>	<code>bool</code>	Booleanská hodnota, která nabývá hodnoty <code>true</code> pokud se daný hráč odpojil nebo hodnoty <code>false</code> pokud se odpojil.

Tabulka 19: Atributy zprávy `GameStateRegeneration`

Název atributu	Typ atributu	Význam atributu
<code>user</code>	<code>struct User</code>	Informaci o uživateli, ke kterému se uživatel znovu připojil.
<code>players</code>	<code>[] struct Players</code>	Seznam hráčů ve hře.
<code>tiles</code>	<code>[] struct Tile</code>	Seznam políček, které jsou na herní desce obsazené a obsahují nějaké písmeno.
<code>active_player_id</code>	<code>int</code>	ID hráče, který je v aktuálním kole na tahu.
<code>player_points</code>	<code>map int:struct Player</code>	Mapa, kde klíčem je počet bodů a hodnotou je hráč, který tohoto zisku ve stávající hře dosáhl.
<code>current_player_points</code>	<code>int</code>	Počet bodů, které hráč, který je nyní na tahu, získal v aktuálním kole.

<code>player_ids_that_accepted</code>	<code>[] int</code>	Seznam ID hráčů, kteří schválili aktuální slova na herní desce.
<code>letters</code>	<code>[] struct Letter</code>	Seznam písmenek, která má hráč toto kolo k dispozici.

Tabulka 20: Atributy zprávy `UserStateRegeneration`

Název atributu	Typ atributu	Význam atributu
<code>state</code>	<code>int</code>	Stav uživatele po znovupřipojení, kdy hodnota 0 znamená, že se server restartoval, hodnota 1 znamená, že dané uživatelské jméno se již na serveru vyskytuje, hodnota 2 znamená, že uživatel byl přesunut na výběr místností a hodnota 3 znamená, že po znovupřipojení nedošlo k žádné změně.
<code>player_name</code>	<code>string</code>	Jméno hráče, který zamítl aktuální slova na desce.

## 4.3 Přenášené struktury

### 4.3.1 Struktura `Tile`

Tato struktura reprezentuje políčko herní desky, které může obsahovat položené písmenko. Každé políčko je nějakého typu, kdy typ 0 představuje obyčejné políčko, 1 představuje políčko, které zdvojnásobí bodové ohodnocení slova, kterého je součástí, 2 představuje políčko, které ztrojnásobí bodové ohodnocení slova, kterého je součástí, 3 představuje políčko, které zdvojnásobí hodnotu písmena, které je na tomto políčku položeno, 4 představuje políčko, které ztrojnásobí hodnotu písmena, které je na tomto políčku položeno. V každém se mění hodnota atributu `highlighted` dle toho, zdali je políčko součástí nějakého slova, které v tomto kole hráč, jenž je na tahu, vytvořil.

Tabulka 21: Struktura Tile

Název atributu	Typ atributu	Význam atributu
row	int	Řádek, na kterém se políčko vyskytuje.
column	int	Sloupec, na kterém se políčko vyskytuje.
set	boolean	Booleanská hodnota, která reprezentuje, zdali se na políčku nachází nějaké písmeno.
highlighted	boolean	Booleanská hodnota, která reprezentuje, zdali má být políčko zvýrazněné či nikoliv.
type	int	Přirozené číslo, které představuje typ políčka. Nabývá hodnot 0 až 4 včetně.
letter	struct Letter	Struktura představující písmeno, které je na políčku položené.

#### 4.3.2 Struktura Letter

Tato struktura představuje písmeno, které může být položeno na herní desku. Obsahuje samotné písmeno, bodové ohodnocení písmena a ID hráče, který jej položil.

Tabulka 22: Struktura Letter

Název atributu	Typ atributu	Význam atributu
value	string	Samotné písmeno.
points	int	Bodové ohodnocení písmena.
PlayerID	boolean	ID hráče, který písmeno položil na herní desku.

#### 4.3.3 Struktura Player

Tato struktura představuje hráče hry, který má nějaké jméno a identifikátor. Dále tato struktura obsahuje booleanské hodnoty `ready` a `disconnected`.

Tabulka 23: Struktura Player

Název atributu	Typ atributu	Význam atributu
name	string	Jméno hráče.
id	int	Přirozené číslo, které představuje identifikátor hráče.
ready	boolean	Booleanská hodnota, která nabývá hodnoty <code>true</code> , pokud je hráč připravený ke startu hry a <code>false</code> pokud připravený není.
disconnected	boolean	Booleanská hodnota, která nabývá hodnoty <code>true</code> , pokud je hráč odpojený ze hry a <code>false</code> pokud odpojený není.

#### 4.3.4 Struktura User

Tato struktura představuje uživatele, který je připojen k serveru. Každý uživatel má nějaké jméno a identifikátor.

Tabulka 24: Struktura Player

Název atributu	Typ atributu	Význam atributu
name	string	Jméno hráče.
id	int	Přirozené číslo, které představuje identifikátor hráče.

### 4.3.5 Struktura Lobby

Tato struktura představuje místnost, která slouží pro vytvoření seznamu hráčů, kteří spolu budou hrát hru. Každá místnost má unikátní číselný identifikátor, seznam hráčů a vlastníka místnosti.

Tabulka 25: Struktura **Letter**

Název atributu	Typ atributu	Význam atributu
id	int	Přirozené číslo, které představuje identifikátor místnosti.
players	int	Seznam hráčů, kteří se momentálně nachází v místnosti.
owner	boolean	Hráč, který založil tuto místnost.

## 4.4 Chybové kódy

Pokud nějaký požadavek uživatele skončí chybou, tak server vrátí zprávu o typu některého z následujících kódů. Detailnější popis chyby se nachází v atributu zprávy `content`.

- **401** - `MarshalError` - datová část je ve chybném formátu JSON
- **407** - `OperationCannotBePerformed` - odeslanou zprávu není možné v daném kontextu uživatele zpracovat
- **420** - `PlayerAlreadyCreatedLobby` - hráč se pokusil založit místnost pokud již jednu založil
- **421** - `LobbyDoesNotExist` - hráč se pokusil připojit do místnosti pomocí neexistujícího ID místnosti
- **422** - `CouldNotLeaveLobby` - hráč se pokusil opustit lobby pokud se v žádné momentálně nenachází
- **423** - `CouldNotFindSuchUserInLobby` - hráč se pokusil změnit svůj stav v lobby, když se v žádné ve skutečnosti nenachází
- **424** - `PlayerNameAlreadyTaken` - nelze se autorizovat dle daného jména, protože je již zabrané



- **425** - `PlayerNameAlreadyTaken` - nelze se autorizovat dle daného jména, protože je již zabrané
- **426** - `GeneralError` - při obsluze požadavku vznikla blíže nespecifikovaná chyba
- **427** - `LobbyPlayerLimitExceeded` - místnost byla zcela naplněna, již do ní nelze vstoupit
- **428** - `GameNotFoundByPlayerID` - dle hráčovo identifikátoru nebyla nalezena hra
- **429** - `NotPlayersTurn` - hráč není na tahu
- **430** - `PlayerNotFound` - neexistující hráč se pokusil schválit slova na herní desce
- **431** - `PlayerCannotAcceptHisOwnWords` - hráč, který je momentálně na tahu, se pokusil schválit vlastní písmena
- **432** - `LetterCannotBePlaced` - písmenko nelze položit na herní desku

## 4.5 Popis serveru

Server je napsaný v jazyce **Golang** a je jednovláknový díky tomu, že z jednotlivých socketů čte pomocí konstrukce *SELECT*. Pro periodické kontrolování stáří *keep alive* zpráv je vytvořena jedna *goroutina*, jenž obsluhuje časovač.

### 4.5.1 Ošetření správné posloupnosti zpráv

K dosažení toho, že server reaguje jedině na zprávy uživatelů, které dávají smysl pouze v jeho aktuálním kontextu, server implementuje jednoduchý stavový automat, který pro každý stav akceptuje pouze předem definovaný výčet zpráv. Automat dle typů zpráv přepíná mezi možnými stavy, avšak některé obsluhy zpráv provádí manuálně přechod mezi stavy (např. akce jednoho hráče přesune druhého hráče do jiného stavu). Pokud se uživatel pokusí poslat serveru zprávu mimo jeho kontext, server odpoví zprávou s chybovým typem 407.



- **networking** - v tomto modulu je implementován TCP server využívající konstrukci *SELECT*. Obsah z jednotlivých socketů předává nezávisle další vrstvě.
- **protocol** - v tomto modulu je implementován navržený protokol. Čtení zpráv je rozděleno do tří nezávislých vrstev, kdy první vrstva čte hlavičky zpráv a druhá parsuje datovou část zprávy. Třetí vrstva jednotlivé zprávy obsluhuje. V tomto modulu jsou definovány i jednotlivé zprávy a odpovědi.
- **test** - obsahuje testy, které testují funkčnost parserů zpráv protokolu. Dále se zde také testují pomocné funkce, jako například `IsNextByteEscaped`, která kontroluje, zdali by byl další byte escapovaný.

#### 4.5.3 *Keepalive* zprávy

Server vyžaduje, aby od každého klienta byly přijímány každé dvě sekundy zprávy typu `KeepAlive`. Pokud během dvou sekund nepřijde tato zpráva potvrzující funkčnost klienta, tak je spojení mezi klientem a serverem považováno za nevyhovující a klient je odpojen.

Pokud se hráč znovu připojí a korektně se autorizuje, tak je jeho stav obnoven a hráč obdrží příslušnou zprávu, pomocí které zrekonstruuje aktuální stav hry.

#### 4.5.4 Nedodržování protokolu

Server vyžaduje, aby od každého klienta byly přijímány pouze zprávy splňující definovaný protokol. Pokud po sobě server přijme od nějakého klienta 5 nevalidních bytů, tak server klienta odpojí.

#### 4.5.5 Použité knihovny

Pro logování byla použita knihovna `logrus` - <https://github.com/Sirupsen/logrus>

### 4.6 Popis klienta

Klient je napsaný v jazyce **Kotlin** a pro zobrazování GUI používá knihovnu `TornadoFX`, jenž obaluje knihovnu `JavaFX`. Pro čtení ze socketu a periodické odesílání *keepalive* zpráv jsou vytvořena speciální vlákna. Klient je tedy vícevláknový.

#### 4.6.1 Struktura modulů

- **model** - definice použitých modelů
- **networking** - implementace definovaného protokolu, čtení ze socketů a definice zpráv implementace protokolu. Stejně jako na serveru, je čtení příchozích zpráv rozděleno do tří vrstev. Třída **Network** se stará o volání připojování k serveru a poskytuje metody, které slouží k registraci reakcí na příchozí zprávy.
- **screens** - definice jednotlivých obrazovek aplikace. Ke každé obrazovce je ve zvláštním souboru definován **Controller** obsluhující události aplikace.
- **test** - obsahuje testy, které testují funkčnost parserů zpráv protokolu. Dále se zde také testují pomocné metody, jako například **IsNextByteEscaped**, která kontroluje, zdali by byl další byte escapovaný.

#### 4.6.2 Nedodržování protokolu

Klient vyžaduje, aby od serveru byly přijímány pouze zprávy splňující definovaný protokol. Pokud po sobě klient přijme od serveru 5 nevalidních bytů, tak klient přeruší se serverem spojení.

#### 4.6.3 Použité knihovny

- Pro logování byla použita knihovna **kotlin-logging** s využitím knihovny **logback** - <https://github.com/MicroUtils/kotlin-logging>.
- Pro parsování JSON zpráv byla použita knihovna **klaxon** - <https://github.com/cbeust/klaxon>.
- Pro testování byla využita knihovna **JUnit 5.5.2**

## 5 Překlad a spuštění aplikace

### 5.1 Server

Pro překlad a spuštění serveru je vyžadován následující software:

- go1.13.1 či novější

V kořenu projektu serveru přeložte server pomocí příkazu `go build -o kral_ups`.  
Poté server spusťte příkazem `./kral_ups`.

V následujícím pořadí lze přidat argumenty při spuštění serveru:

- IPv4 adresa
- port
- limit připojených klientů
- limit místností

### 5.2 Klient

Pro překlad a spuštění klienta je vyžadován následující software:

- OpenJDK 11.0.2
- Gradle 5.6-rc-1 (wrapper je distribuován spolu s aplikací)

V kořenu projektu klienta přeložte a spusťte klienta pomocí příkazu `./gradlew run`.

## 6 Závěr

V rámci této semestrální práce byl vytvořena dvojice programů - server a klient napodující deskovou hru Kris Kros. Důkladný a pečlivý návrh protokolu se osvědčil a v průběhu vývoje neprošel žádnými změnami. Avšak co se několikrát měnilo, byl způsob čtení zpráv, kdy nakonec jsou zprávy čteny spolehlivým automatem. Pokud bych automat navrhl hned na začátku, mohl jsem si ušetřit spoustu času laděním a upravováním parseru zpráv. Aplikace i server jsou však nyní funkční a hra je hratelná. Pokud je spojení s některým hráčem dočasně nedostupné nebo je klient násilně vypnut, tak po autorizaci je hráčovo stav korektně obnoven.