



SEMESTRÁLNÍ PRÁCE Z PŘEDMĚTU KIV/UPS

# REALTIME ONLINE MULTIPLAYER HRA - ASTEROIDY

PATRIK JANOUŠEK

A17B0231P

janopa@students.zcu.cz

ZÁPADOČESKÁ UNIVERZITA V PLZNI  
FAKULTA APLIKOVANÝCH VĚD

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Protokol</b>	<b>2</b>
2.1	Specifikace . . . . .	2
<b>3</b>	<b>Implementace</b>	<b>2</b>
3.1	Popis hry . . . . .	2
3.2	Definice použitých zpráv . . . . .	2
3.2.1	Zprávy zasílané klientem . . . . .	3
3.2.2	Zprávy zasílané serverem . . . . .	7
3.3	Server . . . . .	10
3.3.1	Sít . . . . .	10
3.4	Reakce na zprávy . . . . .	11

# 1 Úvod

## 2 Protokol

### 2.1 Specifikace

Protokol byl navržen s ohledem na jednoduchost implementace, ale aby zároveň splnil všechny požadavky, které by aplikace mohla mít.

Mezi hlavní požadavky na protokol byla identifikace typu zprávy. Tu je možné provádět pomocí trojčiferných číselných identifikátorů. Díky těmto identifikátorům je schopná komunikující protistrana předpokládat strukturu příchozích dat.

Podporuje tedy identifikaci zpráv, kde si klient může určit vlastní identifikátor zprávy, a server tento identifikátor zapíše i do odpovědi, díky čemuž je klient schopen poznat na jakou zprávu server odpovídá. Tento identifikátor má variabilní délku, tudíž je zde i možnost jeho nepoužití.

Tělo zprávy je pak zakódované ve formátu JSON, kde je pro klíče použita konvence snake case.

Popis	Délka	Hodnota
Dělicí znak	1	„ “
Typ zprávy	3	Číslo v ASCII
Dělicí znak	1	„ “
Délka těla zprávy	variabilní	Číslo v ASCII
Dělicí znak	1	„ “
Identifikátor požadavku	variabilní	Řetězec znaků a-z, A-Z a 0-9 v ASCII
Dělicí znak	1	„ “
Tělo zprávy	variabilní	Data zakódovaná v JSONu

Tabulka 1: Specifikace protokolu

## 3 Implementace

### 3.1 Popis hry

### 3.2 Definice použitých zpráv

Jak již bylo uvedeno výše, každý typ zprávy má svůj unikátní trojčiferný identifikátor. Tento identifikátor může být z hlediska protokolu libovolný,

nicméně pro lepší orientaci byla v konkrétní implementaci vytvořena určitá pravidla.

A to taková, že identifikátor 1xx je použit pro obecné zprávy, které nemusí přímo souviset se samotnou aplikací (keep-alive, atd. . . ). 2xx pak slouží pro zprávy zasílané klientem, a 3xx zprávy zasílané serverem. Zde je ještě dodržováno to, že „xx“ je stejné pro klientskou i serverovou zprávu, pokud je serverová zpráva odpovědí na klientský požadavek. Tyto identifikátory jsou použity pro zprávy, které přímo nesouvisí se hrou, ale spíše s její správou (autentizace, založení lobby, získání seznamu hráčů, . . . ). Nakonec jsou pak použity identifikátory zpráv 4xx a 5xx, kde 4xx jsou zprávy zasílané klientem, a 5xx zprávy zasílané serverem. Tyto identifikátory jsou použity pro zprávy přímo související se hrou (pohyb hráče po herní ploše, střelba, . . . ).

Všechny zprávy zasílané serverem, jsou obalovány do struktury, která umožňuje k zasílaným datům přidat status, a případně nějakou dodatečnou zprávu. Přenášená zpráva je pak dostupná pod klíčem „data“.

Klíč	Datový typ
data	object
message	string
status	boolean

Tabulka 2: Obalovací zpráva serveru

### 3.2.1 Zprávy zasílané klientem

#### KeepAlive

Typ zprávy: 100

Tato zpráva je zasílána klientem, aby ověřila funkčnost spojení se serverem, a zároveň schopnost komunikace mezi těmito dvěma stranami.

Klíč	Datový typ
ping	string

Tabulka 3: Specifikace KeepAlive

Klient jako hodnotu pole „ping“ vyplňuje řetězec „pong“. A server na tuto zprávu odpovídá stejným typem zprávy, jen jako hodnotu pole „ping“ vyplňuje řetězec „ping-pong“.

## ActionError

Typ zprávy: 101

Zpráva, která je zasílána serverem jako odpověď na typ zprávy, ke které klient nemá oprávnění, nebo daná zpráva neexistuje. Tato zpráva má prázdné tělo a neposílá žádná dodatečná data.

## Authenticate

Typ zprávy: 200

Autentizační zpráva musí být klientem zaslána jako první zpráva po připojení k serveru. V případě neposlání této zprávy nemá klient žádné pravomoce, mimo zasílání *KeepAlive*.

Klíč	Datový typ	Popis
name	string	Jméno hráče

Tabulka 4: Specifikace Authenticate

## CreateLobby

Typ zprávy: 201

Zpráva určená k založení lobby. Hráč, který lobby vytváří, je automaticky do lobby připojen a není již potřeba posílat *JoinLobby*.

Klíč	Datový typ	Popis
name	string	Název lobby
players_limit	integer	Limit počtu hráčů v lobby

Tabulka 5: Specifikace Authenticate

## DeleteLobby

Typ zprávy: 202

Zpráva určená ke smazání lobby. Ačkoliv zpráva přijímá jako parametr název lobby ke smazání, tak hráč má právo smazat lobby jenom za předpokladu, že je jejím vlastníkem.

Klíč	Datový typ	Popis
name	string	Název lobby ke smazání

Tabulka 6: Specifikace DeleteLobby

### ListLobbies

Typ zprávy: 203

Prázdná zpráva sloužící k získání seznamu všech dostupných lobby.

### JoinLobby

Typ zprávy: 204

Zpráva sloužící k připojení do lobby. Připojení je možné pouze za předpokladu, že je v lobby volný slot pro hráče. Tato skutečnost je ověřována serverem, který vyplní patřičný status zprávy.

Klíč	Datový typ	Popis
name	string	Název lobby k připojení

Tabulka 7: Specifikace JoinLobby

### LeaveLobby

Typ zprávy: 214

Prázdná zpráva, kterou se hráč odpojí z lobby.

### ListLobbyPlayers

Typ zprávy: 206

Prázdná zpráva sloužící k získání seznamu hráčů připojených do lobby.

### StartGame

Typ zprávy: 207

Prázdná zpráva pro spuštění hry. Tuto zprávu může poslat libovolný hráč z lobby. Od této chvíle je na serveru vytvořená instance hry, a všichni hráči jsou neustále informováni o jejím stavu.

## GameReconnectAvailable

Typ zprávy: 211

Touto prázdnou zprávou si může klient ověřit, zda připojený hráč nepatří do nějaké hry, a tudíž se do ní může opět připojit, nebo se z ní odpojit.

## Reconnect

Typ zprávy: 212

Zasláním této prázdné zprávy se hráč připojí zpět do probíhající hry, kterou předtím opustil.

## LeaveGame

Typ zprávy: 213

Zasláním této prázdné zprávy hráč definitivně opustí probíhající hru, ze které odešel, a již se do ní nemůže vrátit.

## PlayerMove

Typ zprávy: 400

Zpráva, která informuje server o změně atributů hráče. Tuto zprávu by měl klient posílat při změně své pozice, nebo libovolného atributu svého pohybu (vektor pohybu, rotace, ...). Zaslané údaje jsou na serveru verifikovány a v případě chybně zaslaných údajů je klient vyžádán k synchronizaci údajů se serverem (viz. *UpdateState*).

Klíč	Datový typ	Popis
pos_x	double	Pozice na ose X
pos_y	double	Pozice na ose Y
velocity_x	double	Složka X vektoru pohybu
velocity_y	double	Složka Y vektoru pohybu
rotation	double	Rotace hráče v radiánech

Tabulka 8: Specifikace PlayerMove

## ShootProjectile

Typ zprávy: 401

Prázdná zpráva informující server o vystřelení projektilu

### 3.2.2 Zprávy zasílané serverem

#### CreatedLobbyResponse

Typ zprávy: 301

Odpověď na zprávu: *CreatedLobbyResponse*

Prázdná zpráva, kterou je klient informován o založení lobby.

#### DeleteLobbyResponse

Typ zprávy: 302

Prázdná zpráva, kterou je klient informován o smazání lobby.

#### ListLobbiesResponse

Typ zprávy: 303

Odpověď na zprávu: *ListLobbies*

Tuto zprávu server používá pro zaslání seznamu všech dostupných lobby. V kořenu zprávy je jedinný klíč „lobbies“, který má následující strukturu:

Klíč	Datový typ	Popis
name	string	Název lobby
connected_players	integer	Počet hráčů připojených do lobby
players_limit	integer	Maximální počet připojených hráčů

Tabulka 9: Specifikace ListLobbiesResponse

#### JoinLobbyResponse

Typ zprávy: 304

Odpověď na zprávu: *JoinLobby*

Prázdná zpráva informující klienta o stavu připojení do lobby.



## PlayerLobbyJoined

Typ zprávy: 305

Zpráva informující hráče, kteří jsou již připojení v lobby, o připojení nového hráče.

Klíč	Datový typ	Popis
name	string	Jméno hráče

Tabulka 10: Specifikace ListLobbiesResponse

## ListLobbyPlayersResponse

Typ zprávy: 306

Odpověď na zprávu: *ListLobbyPlayers*

Zpráva, kterou server klientovi vrací seznam hráčů připojených do lobby.

Klíč	Datový typ	Popis
players	string[]	Pole jmen hráčů

Tabulka 11: Specifikace ListLobbyPlayersResponse

## StartGameResponse

Typ zprávy: 307

Odpověď na zprávu: *StartGame*

Prázdná zpráva, která se zasílá všem hráčům při startu hry.

## LobbyPlayerConnected

Typ zprávy: 308

Zpráva informující ostatní hráče v lobby o připojení nového hráče.

Klíč	Datový typ	Popis
name	string	Jméno příchozího hráče

Tabulka 12: Specifikace LobbyPlayerConnected

## LobbyPlayerDisconnected

Typ zprávy: 309

Zpráva informující ostatní hráče v lobby o odpojení hráče.

Klíč	Datový typ	Popis
name	string	Jméno odpojeného hráče

Tabulka 13: Specifikace LobbyPlayerConnected

## GameEnd

Typ zprávy: 310

Tato zpráva je zaslána všem hráčům hry po jejím skončení. V kořenu zprávy obsahuje jediný klíč „score\_summary“, který obsahuje pole s objekty následující struktury:

Klíč	Datový typ	Popis
player_name	string	Jméno hráče
score	integer	Dosažené skóre
winner	boolean	Informaci o tom, zda je daný hráč výherce

Tabulka 14: Specifikace LobbyPlayerConnected

## GameReconnectAvailableResponse

Typ zprávy: 311

Odpověď na zprávu: GameReconnectAvailable

Zpráva informující o tom, zda je možné se připojit zpět do probíhající hry, ze které se hráč odpojil. Obsahuje pouze jednu hodnotu:

Klíč	Datový typ	Popis
available	boolean	Dostupnost znovupřipojení se do probíhající hry

Tabulka 15: Specifikace LobbyPlayerConnected

## **ReconnectResponse**

Typ zprávy: 312

Odpověď na zprávu: Reconnect

Prázdná zpráva informující klienta o úspěšném znovupřipojení zpět do hry.

## **LeaveGameResponse**

Typ zprávy: 313

Odpověď na zprávu: LeaveGame

Prázdná zpráva informující klienta o úspěšném opuštění probíhající hry.

## **LeaveLobbyResponse**

Typ zprávy: 314

Odpověď na zprávu: LeaveLobby

Prázdná zpráva informující klienta o úspěšném opuštění lobby.

## **3.3 Server**

Pro implementaci serveru byl zvolen jazyk go. Jednak z důvodu vyzkoušení nové technologie, ale i z důvodu jeho technologických výhod. Zejména z důvodu jednoduchosti paralelizace a možnosti jednoduché a bezpečné komunikace mezi jednotlivými vlákny (resp. gorutinami), ale i automatické správy paměti.

### **3.3.1 Síť**

Síť je v serveru implementována s důrazem na co největší oddělení od zbytku aplikace, a je vedle aplikace v samostatném balíku „net“, který je pojmenován po vzoru stejnojmenného balíku ze standardní knihovny jazyka go. Tento balík obsahuje implementaci jak samotného TCP serveru, tak i protokolu.

Protokol je plně oddělen od TCP serveru, se kterým komunikuje pomocí rour. Díky tomu je protokol schopen komunikovat s libovolným binárním streamem dat, který nutně nemusí přijít od TCP serveru. Tato vlastnost ho dělá i samostatně testovatelným. Nevýhodou je, že je díky tomu komunikace s protokolem blokující, a každý klient musí mít pro protokol vlastní gorutinu.

Pokud tedy přijde zpráva, tak jí TCP server přečte ze socketu, a zapíše jí do patřičné roury připojeného klienta, odkud zprávu začne přečte protokol, a začne jí zpracovávat. Po tom, co protokol přečte celou zprávu, tak ji naplní do struktury `ProtoMessage`, a pomocí kanálu ji předá zpět TCP serveru, který tuto zprávu vezme, přidá k ní odesílatele, a skrz další kanál jí předá master serveru, který může zprávu na základě jejího obsahu a typu dále zpracovávat.

### **3.4 Reakce na zprávy**