



Semestrální práce z KIV/UPS

Síťová hra - Othello

Daniel Stuš
A17B0354P
stusd@students.zcu.cz

14. 02. 2021

Obsah

1	Zadání	2
2	Popis hry	1
2.1	Pravidla hry	1
3	Popis protokolu	3
3.1	Formát zpráv	3
3.2	Zprávy posílané klientem	3
3.3	Zprávy posílané serverem	5
3.4	Omezení a validace hodnot ve zprávách	7
4	Implementace	8
4.1	Klient	8
4.2	Server	9
5	Uživatelská příručka	10
5.1	Překlad	10
5.1.1	Server	10
5.1.2	Client	10
5.2	Průběh hry	11
5.2.1	Hlavní menu	11
5.2.2	Připojení na server	12
5.2.3	Lobby	12
5.2.4	Hra	14
6	Závěr	15

1 Zadání

Cílem úlohy je vytvořit síťovou aplikaci představující hru, pro nejméně dva hráče a to za použití TCP protokolu. Serverová část má být psána v nízkoúrovňovém jazyce typu `C/C++`, a klientská aplikace naopak ve vysokoúrovňovém jazyce jakými jsou například `Java` nebo `C`. Klient by také měl implementovat jednoduché a přehledné grafické uživatelské rozhraní (GUI), které bude vizualizovat kompletní průběh hry.

Aplikace musí být robustní a zvládat obsluhovat několik uživatelů v jednu chvíli. Měla by umět reagovat na neočekávané situace a vypisovat podrobně svůj stav. Obě části aplikace (klient-server) mají respektovat protokol, který si student sám navrhne.

V neposlední řadě, klientskou i serverovou část je nutné kompilovat pomocí standardních kompilačních nástrojů jakými jsou `make` pro `C/C++`, a `Ant` nebo `Mave`. Vyjimkou jsou programy zhotovené v herním enginu `Unity`.

Kompletní, detailní podobu zadání lze nalézt na adrese: <http://home.zcu.cz/~ublm/vyuka/ups/Pozadavky2019.pdf>.

2 Popis hry

Othello, někdy též nazývána Reversi, je tahová logická hra pro dva hráče. Hraje se na šachovnici o velikosti 8x8 polí. Hráči střídavě na šachovnici pokládají kameny, které jsou z jedné strany bílé a z druhé černé, a to tak, aby mezi právě položený kámen a jiný kámen své barvy uzavřeli souvislou řadu soupeřových kamenů. Uzavřené kameny se následně otočí a stávají se kameny druhého hráče. Cílem hry je takto nasbírat více kamenů, než protivník.

2.1 Pravidla hry

Hraje se na desce 8x8 polí, jež je nazývána otheliera. Deska vypadá obdobně jako šachovnice, ačkoli na ní nejsou rozlišena bílá a černá pole. Při hře se používají kameny, které jsou z jedné strany bílé a z druhé černé.

Na počátku hry jsou rozloženy čtyři kameny ve čtyřech středových polích (d4, e4, d5, e5). Tyto počáteční kameny se střídají tak, že vždy přímo z boku sousedícím kamenem je kámen protihráče. Na políčku d4 je umístěn kámen bílé barvy.

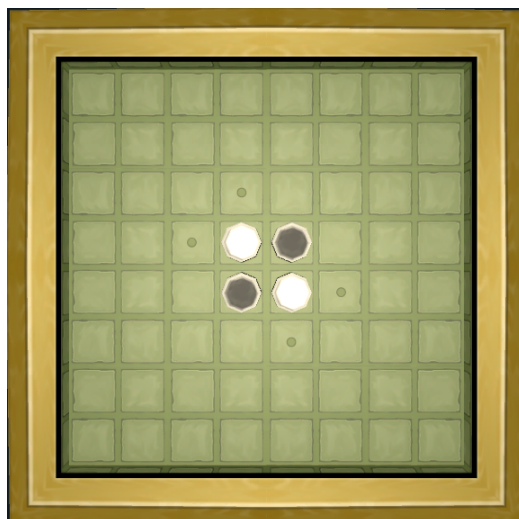
Hru oproti běžnému úzu jako například v šachovém turnaji, zahajuje v Othellu hráč s černými kameny. Hráči se následně v tazích střídají.

Každý hráč ve svém tahu smí položit kámen svou barvu nahoru tak, aby mezi tímto kamenem a jiným kamenem stejné barvy byla ve stejné řadě, sloupci či uhlopříčce přítomna souvislá nepřerušovaná řada soupeřových kamenů. Všechny tyto kameny, které takto hráč uzavřel se následně otočí, a stávající se z nich tedy jeho kameny. Pokud se hráči povede takto uzavřít více řad najednou, otočí se všechny kameny. Po otočení se dále kameny už neotáčejí a to i v případě, že nově otočené kameny uzavírají další nepřerušovanou řadu.

Hráč musí táhnout tak, aby otočil alespoň jeden soupeřův kámen. Není-li žádný tah možný, jeho tah je přeskóčen a pokračuje soupeř. Pokud hráč může táhnout alespoň na jednu pozici, musí tak učinit, i za předpokladu, že je to pro něj nevýhodné.

Hra končí, když je zaplněno všech 64 polí nebo když ani jeden z hráčů nemůže táhnout. Vyhrává ten z hráčů, který na desce má více kamenů své barvy. Pokud mají hráči stejný počet kamenů, hra končí remízou. Pokud hra skončila tím, že oba hráči nemohou táhnout a na desce ještě zbylo volné

místo, jsou často tyto volná místa připočítávána ke skóre vítěze tak, aby součet bodů byl vždy roven 64.



Obrázek 2.1: Počáteční pozice hry

3 Popis protokolu

3.1 Formát zpráv

Zprávy posílané mezi klientem a serverem mají jednoduchý formát, kdy na prvním místě je vždy prefix, odkud zpráva pochází. Prefix C označuje zprávy zasílené klientem a prefix S naopak zprávy zasílané serverem. Po prefixu následuje konkrétní typ zprávy a následně její obsah. Zpráva je vždy ukončena znakem nové řáky ("\n"). Jednotlivé části zprávy jsou odděleny vlnovkou ("~").

Podoba zprávy: <Prefix><Typ>~<Obsah>\n

Př. CNAME~Daniel\n

3.2 Zprávy posílané klientem

- CNAME~<jméno>
 - Slouží pro přihlášení uživatele
 - <jméno> = jméno uživatele, který se chce přihlásit
- CPING
 - Jednoduchá zpráva dotazující se na dostupnost serveru. Na základě odpovědi klient ví zda ztratil spojení se serverem
- CMOV~<x>~<y>
 - Slouží pro provedení tahu (MOVE)
 - <x> - x-ová souřadnice tahu
 - <y> - y-ová souřadnice tahu

- **CREQ~<jméno__protihráče>**
 - Umožňuje vyzvat protihráče ke hře
 - <jméno__protihráče> = jméno protihráče, kterému chceme poslat pozvánku ke hře
- **CRES~<jméno__hráče>~<ACCEPT/DECLINE>**
 - Slouží pro přijmutí, či zamítnutí pozvánky ke hře
 - <jméno__hráče> = jméno hráče, kterému odpovídáme na pozvánku
 - <YES/NO> = odpověď YES = přijmout, NO = odmítnout pozvánku
- **CREQC~<jméno__hráče>**
 - Zruší odeslanou pozvánku ke hře
 - <jméno__hráče> = jméno hráče, kterému byla pozvánka odeslána
- **CENDGAME**
 - Slouží pro předčasné ukončení hry
- **CEXIT**
 - Slouží pro ukončení spojení hráče se serverem (hráč ukončil program)

3.3 Zprávy posílané serverem

- **SADDP~<jméno_hráče>**
 - Slouží pro přidání klienta do lobby
 - <jméno_hráče> = jméno uživatele, který se má přidat do lobby
- **SDELP~<jméno_hráče>**
 - Slouží pro odstranění hráče ze seznamu v lobby
 - <jméno_hráče> = jméno uživatele, který se má odstranit z lobby
- **SUPS~<jméno_hráče>~<READY/BUSY>**
 - Slouží pro změnu stavu hráče v lobby
 - <jméno_hráče> = jméno uživatele u kterého chceme změnit stav
 - <READY/BUSY> = stav, který chceme přiřadit hráči; READY = je v lobby a volný, je možné jej pozvat do hry. BUSY = obdržel pozvánku nebo aktuálně hraje. Takového hráče není možné pozvat.

- **SREQC~<jméno_hráče>**
 - Slouží pro zrušení pozvánky do hry
 - <jméno_hráče> - jméno hráče, pro kterého cheme zrušit pozvánku
- **SPONG**
 - Opozvěď na uživatelský dotaz CPING, díky tomu server i klient ví, že mezi sebou mají stabilní spojení
- **SSTARTG~<jméno_hráče>**
 - Informuje hráče o spuštění hry a tím začíná hru samotnou
 - <jméno_hráče> - jméno protihráče
- **SGAMEEC~GameHasBeenCanceled**
 - Slouží pro předčasné ukončení hry. Ať už v případě odchodu hráče nebo na popud hráčů samotných
- **SGAMEI~<text>**
 - Slouží pro zaslání informace do hry klienta
 - Využíváno např. při oznámení o ztrátě spojení serveru s protihráčem a pod.
 - <text> - Text, který se má uživateli ukázat

- **SRECOV~<pole_stavu>~<hráč_na_tahu>~<hostitel/host>**
 - Slouží pro obnovu uživatelského stavu do hry po znovunapojení
 - pole_stavu - textová reprezentace stavu hry
 - <hráč_na_tahu> - 1, pokud je bílý hráč na tahu; 0, pokud černý
 - /item <hostitel/host> - 1, pokud je uživatelem hostitelem hry (vyzivatelem); 0, pokud byl příjemcem pozvánky

3.4 Omezení a validace hodnot ve zprávách

Server validuje prefix zpráv. Pokud je prefix nerozpoznaný, server dále zprávu nezpracovává. V případě, že na server dorazí zpráva v nekorektním formátu či v nesprávném stavu, server klienta odpojí.

4 Implementace

4.1 Klient

Klient je implementován v jazyce C# v multiplatformním herním enginu Unity (verze 2020.2.1f1). Grafické prvky UI jsou součástí standardní výbavy enginu. Herní prvky, včetně kamenů a šachovnice byli modelováni v 3D programu Blender. Klient pracuje na základě dvou vláken. Hlavní vlákno se stará o obsluhu programu, přijímání a odesílání zpráv a zobrazování grafických komponent a druhé vlákno slouží pro kontrolu dostupnosti serveru zasíláním průběžných ping zpráv. Klient je rozdělen do několika scriptů.

- **GameManager** - Balík uchovává informace o scénách. Script je postaven jako jedináček, který je zachovávan i v průběhu přepínání scén, které sám provádí. GameManager dále uchovává informace o připojených hráčích v lobby.
- **Client** - Jednoduchý TCP client pro navázání spojení a příjem a obsluhu zpráv. Script rovněž obsahuje samostatné vlákno pro periodické ping zprávy.
- **Othello** - Balík obsahuje vnitřní logiku pro hráče a hru. Třída Othello uchovává informace o hře a umožňuje její obsluhu pomocí funkcí.
- **MenuController** - Přepíná mezi jednotlivé části menu. Stará se o GUI.
- **Board** - Script obsahující logiku hry, využívá scriptu Token a umožňuje pomocí funkcí hrát hru. Rovněž uchovává odkazy na všechny prvky v herní scéně.

Pokud se klient v průběhu hry odpojí je mu umožněno se znovu připojit po dobu 60 sekund. Pro připojení do hry stačí po znovu spuštění aplikace zadat jméno uživatele, který se chce znovu připojit do probíhající hry. Server pak automaticky nalezne ve které hře se hráč nachází a v jakém stavu a klienta znovu okamžitě připojí. Pokud tak hráč neučiní do 60 sekund, bude hra ukončena stejně, jako by se jednalo o přirozený průběh hry, tedy sečte se skóre a vyhodnotí výherce a zbývajícím hráč ve hře bude přenesen zpět do lobby. Pokud se během času na připojení odpojí i druhý hráč, je hra kompletně ukončena a odstraněna.

4.2 Server

Server byl implementován v multiparadigmatickém programovacím jazyce `c++` a je rozdělen do pěti modulů a jejich hlavičkových souborů:

- **Client** - Modul obsahuje definice a funkce pro vytváření a správu klientů (uživatelů).
- **ErrorMessages** - Modul obsahuje definice chybových zpráv, které jsou využívány modulem **Server**.
- **Othello** - Modul **Othello** obsahuje kompletní herní logiku, která je stínovou verzí logiky v klientské části. Uchovává si informace o hráčích, hře, stavu a kontroluje možné tahy.
- **Server** - Modul zajišťuje vytvoření socketu, jeho obsluhu a obsluhu spojení jednotlivých klientů. Umožňuje komunikovat s klientem pomocí protokolu a ověřuje správnost zpráv.
- **start_server** - Hlavní modul aplikace. Stará se o parsování vstupů z příkazové řádky, naplnění portu a množství hráčů, obsluhuje celý program spouštěním příslušných vláken

Paralelizace bylo dosaženo za pomoci vláken (pro každého připojeného uživatele se vytvoří separátní vlákno). Pro práci s připojením a sockety byly použity knihovny `sys/socket.h`, `sys/select.h`, `netinet/in.h` a `sys/time.h` respektive.

5 Uživatelská příručka

5.1 Překlad

5.1.1 Server

Server se nachází ve složce `server` a lze jej přeložit v terminálu v linuxovém prostředí pomocí příkazu `make`.

Počítač, na kterém je server překládán musí mít nainstalován kompilátor `gcc`. Přeložením zdrojových souborů vznikne složka `bin`, obsahující dílčí soubory s příponou `*.o` a v kořenovém adresáři pak také následně nový soubor `server`.

Server se spouští příkazem `./server` bez parametrů.

Po spuštění serveru je klient proveden úvodním nastavením, kde je postupně dotázán na zadání portu, pod kterým chce aby byl server spuštěn a na maximální počet klientů, který může být na serveru v jednu chvíli připojen. V obou případech může uživatel ponechat výchozí hodnoty prostým stisknutím klávesy `enter` nad prázdným vstupem.

5.1.2 Client

Spouštěcí soubory klienta byli vygenerovány v herním enginu Unity, a to jak pro windows (`Reversi.exe`), tak pro linuxové distribuce (`Reversi.x86_64`). Není tedy potřeba provádět speciální překládající operace.

Pro spuštění klienta na operačním systému windows stačí dvakrát kliknout na ikonu `Reversi.exe`

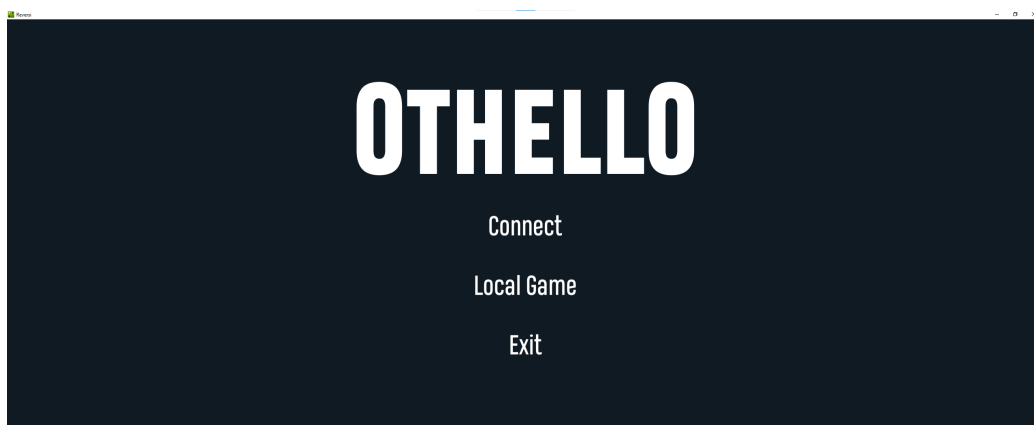
V linuxové distribuci je potřeba klienta spustit v terminálu pomocí příkazu `./Reversi.x86_64`

5.2 Průběh hry

5.2.1 Hlavní menu

Po spuštění klienta je uživateli zobrazena úvodní obrazovka ze které vybráním jedné z možností postupuje dále do aplikace.

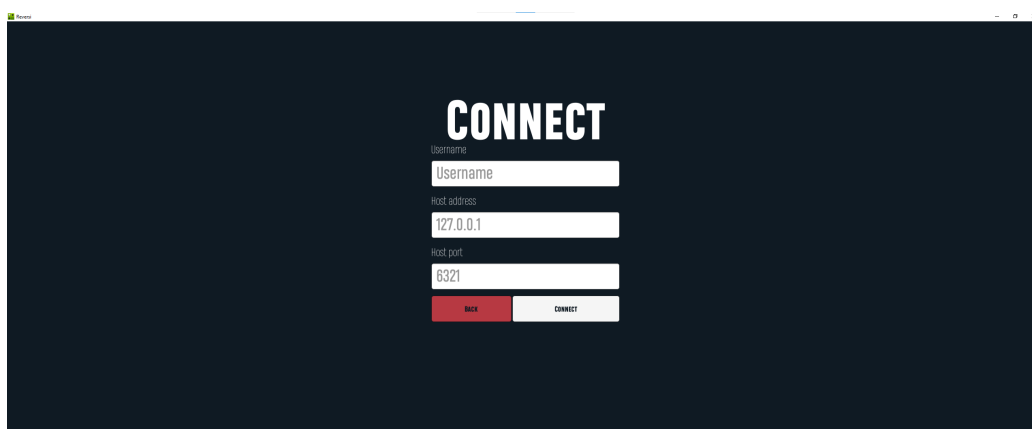
- Tlačítkem **Connect** je uživatel přenesen do sub-menu, kde se může připojit k serveru pro online hru.
- Tlačítkem **Local Game** spustí uživatel hru na lokálním stroji bez nutnosti připojit se k serveru.
- Tlačítkem **Exit** uživatel ukončí klienta hry.



Obrázek 5.1: Hlavní menu hry

5.2.2 Připojení na server

Po stisknutí tlačítka **Connect** v hlavním menu se uživatel ocitne v sub-menu připojení. Zde uživatel vyplní příslušné údaje a stiskem tlačítka **Connect** se připojí k serveru. Uživatel musí vyplnit své jméno v kolonce **Username**. Jméno nesmí být prázdné a musí se jednat o jedno slovo. Do pole **Host address** uživatel zadává IPv4 adresu serveru v běžném formátu čtyř desítkových čísel oddělených tečkou. Pole **Port** slouží pro vyplnění portu, na kterém serverová část aplikace běží. Všechny vstupní pole jsou po stisknutí tlačítka **Connect** kontrolována a o případné chybě je uživatel informován vyskakovacím oknem.

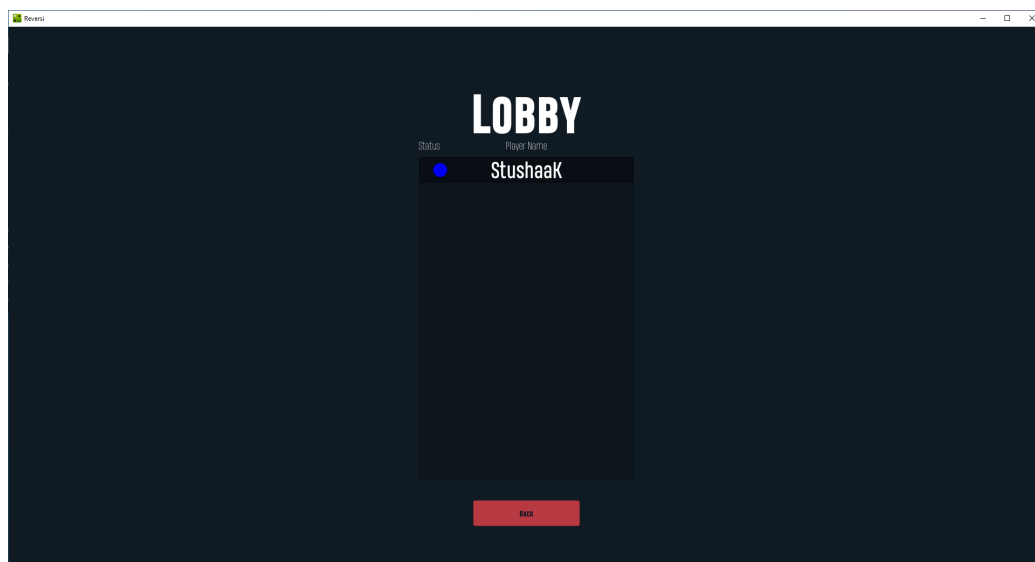


Obrázek 5.2: Sub-menu připojení

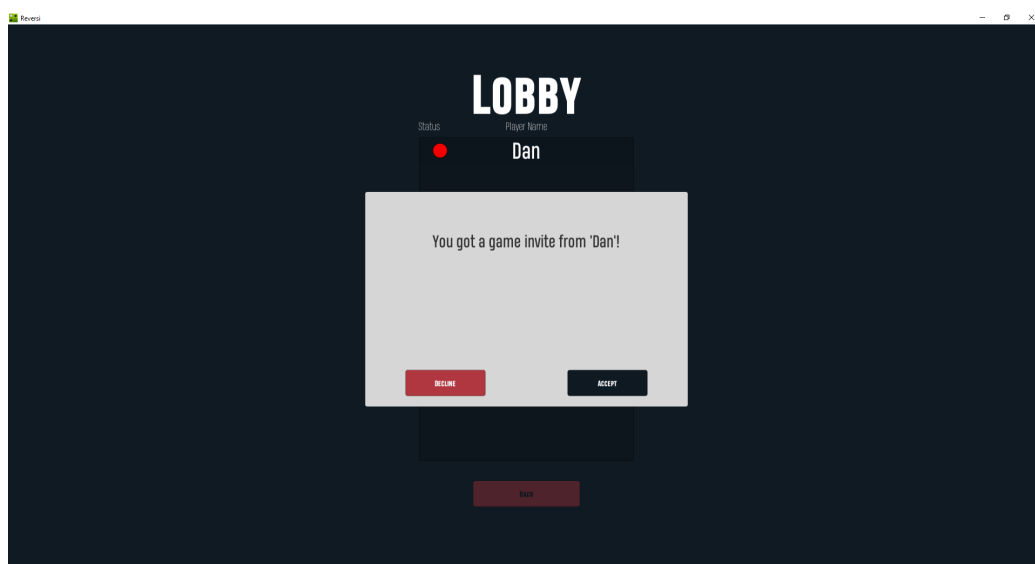
5.2.3 Lobby

Po úspěšném připojení na server se uživatel nachází v lobby. V něm vidí uživatel seznam všech hráčů připojených na server a také jejich stav. Červené kolečko vedle jména hráče značí, že hraje hru, nebo mu již byla zaslána pozvánka a nelze jej tedy momentálně pozvat. Modré značí, že je připraven ke hře.

Kliknutím na jméno kteréhokoliv připraveného hráče může uživatel odeslat pozvánku a tím vyzvat protihráče ke hře. Následně čeká na jeho odpověď. Pokud protihráč pozvánku přijal, jsou oba hráči přeneseni do hry. V opačném případě zůstávají v lobby. Z lobby může kdykoliv uživatel odejít, a tím se i odpojit od serveru, stisknutím tlačítka **Back** v dolní části obrazovky.



Obrázek 5.3: Lobby hry



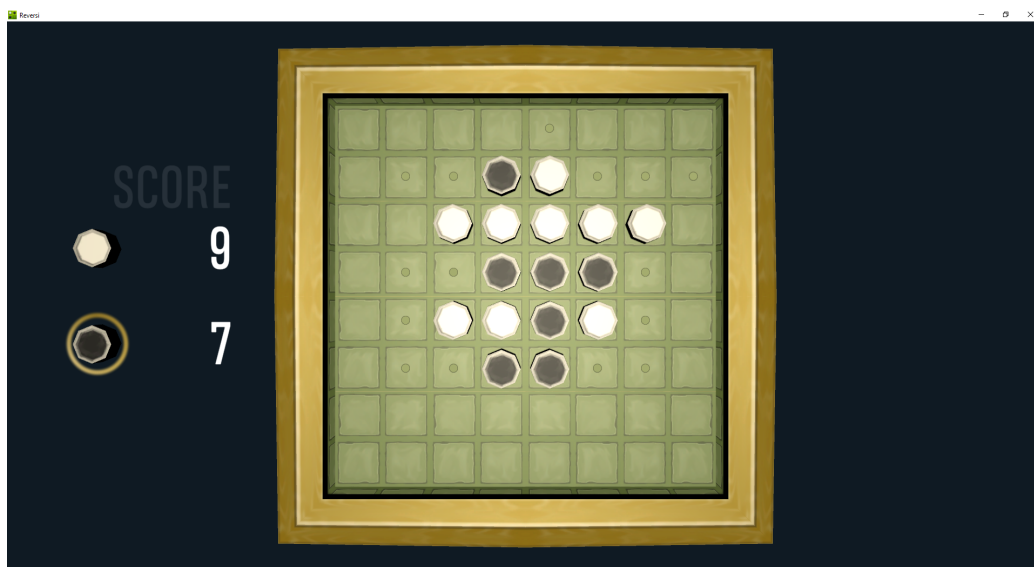
Obrázek 5.4: Přijatá pozvánka od hráče Dan

5.2.4 Hra

Pokud hráč odešle pozvánku a protihráč ji přijme nebo pokud v hlavním menu zvolil uživatel možnost **Local Game**, nachází se klient na obrazovce hry. Hra probíhá dle pravidel popsaných v kapitole 2.1. Hráči se postupně střídají v pokládání kamenů na hrací plochu. Možné tahy hráče jsou znázorněny tečkami na příslušných polích šachovnice. Aktuální hráč na tahu je znázorněn zářícím kruhem okolo kamene v barvě hráče na tahu v levé části obrazovky, kde je rovněž uvedeno skóre jednotlivých hráčů.

V průběhu online hry může hráč ztratit spojení se serverem. V takovém případě je protihráč o tomto stavu informován a hráč který ztratil spojení má 60 sekund na to se znovu připojit. Klient se pokusí hráče který ztratil spojení připojit automaticky. Pokud však hráč ukončil klienta kompletně, může se znovu v rozmezí 60 sekund připojit do hry zadáním stejné přezdívky, kterou si původně vybral při připojování.

Hráči jsou v průběhu a na konci hry informováni o aktuálním stavu (přeskočení tahů apod.). Na konci hry, po oznámení vítěze, jsou hráči přeneseni zpět do lobby, kde mohou opět vyzvat kteréhokoliv připojeného hráče k další hře.



Obrázek 5.5: Průběh hry

6 Závěr

Aplikace běží dle očekávání a v rámci pravidel určených hrou Othello. Obě části aplikace (klient i server) zvládají bezproblémový běh i při příjmu nevalidních hodnot, nepadají a není třeba je restartovat. Aplikace tak představuje pevný základ, který by se dal obohatit o další funkcionality či zdokonalit již užívané algoritmy.

Jako možné zlepšení se nabízí na straně serveru užití paralelizace, a to sice užitím `selectu` nad vlákny, namísto vytváření nového vlákna pro každého připojeného uživatele.

Na straně klienta by pak bylo možné zdokonalit celkové GUI, animace, herní modely a nasvícení herní scény.