

Šachy



Gymnázium, Praha 6, Arabská 14

Předmět Programování, vyučující Jan Lána

Prohlášení

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené.

Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská 14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V Praze dne ____ dubna 2020

Podpis:

Obsah

1) Zadání projektu.....	1
2) Anotace.....	1
3) Abstract	1
4) Úvod	2
5) Návod k projektu.....	3
6) Informace o fungování programu	4
6.1) Použité technologie	4
6.2) Postup instalace a systémové požadavky	4
6.3) Struktura programu	4
6.4) Datové struktury	4
7) Popis zdrojového kódu	5
7.1) Main.java	5
7.2) Chessboard.java.....	5
7.3) MovesCheck.java	5
7.4) Timer.java.....	5
7.5) ChessboardMP.java	5
7.6) Receiver.java	6
7.7) ServerThread.java.....	6
7.8) ChessServer.java	6
7.9) Statistics.java	6
7.10) Settings.java	6
8) Šachovnice.....	7
8.1) Šachovnice.....	8
8.2) Označení možných tahů	9
8.3) Figurky	10
9) Klíčové problémy	11
9.1) Posílání tahů mezi hráči	11
9.2) Předání hodnot z třídy Settings třídě Chessboard	11
10) Závěr.....	12
11) Zdroje	13

1) Zadání projektu

Šachy

Zadáním je vytvořit hru šachy pro dva hráče v Javě. První částí bude aplikace pro klienta, ve které bude samotná hra. Druhou částí bude server, který oba hráče propojí. Hra bude zaznamenávat statistiku, ve které bude počet výher a proher. Před začátkem hry půjde nastavit celkový čas a čas na jednotlivé tahy. Dále ve hře bude možnost nápovědy, např. označení ohrožených figurek, pozice, kam lze figurku přesunout atd.

2) Anotace

Cílem mého ročníkového projektu je naprogramovat šachy v Javě. V této známé hře proti sobě hrají dva hráči, jeden z nich ovládá figurky bílé barvy, druhý ovládá figurky černé barvy. Každý z nich vždy udělá jeden tah podle pravidel, poté hraje druhý hráč. Cílem je dát soupeřovu králi mat. Moje verze funguje tak, že můžou hrát dva hráči u jednoho počítače proti sobě anebo mohou dva hráči hrát proti sobě s tím, že každý je u svého počítače. U her, kdy oba hráči nejsou u jednoho počítače, se zaznamenává počet výher a proher. Před každou hrou lze nastavit celkový čas a čas na jeden tah.

3) Abstract

The goal of my project is to program chess in Java. In this famous game, two players play against each other, one of them controls white pieces, the other controls black pieces. Each of them always makes one move according to the rules, then the other player plays. The goal is to checkmate your opponent's king. My version works so that two players can play against each other on the same computer, or two players can play against each other with each one at their own computer. For games where both players are not on the same computer, the number of wins and losses is recorded. Before each game, you can set the total time and the time for one move.

4) Úvod

V této práci popíšu, jak jsem postupoval při programování své ročníkové práce Šachy. Tuto hru jsem si vybral proto, že rád hraji šachy a přišlo mi zajímavé je naprogramovat. V této práci popisuji, jak jsem postupoval při jejich programování a jaká je struktura mého projektu. Dále obsahuje návod k použití a popis programu. Také vysvětluje použité algoritmy a grafickou část projektu.

5) Návod k projektu

Hra se zapne spuštěním spustitelného Java souboru ve složce Chess, čímž se otevře okno hry. V něm je menu, tlačítko hra označuje variantu hry, kdy dva hráči spolu hrají u jednoho počítače. Hra přes internet označuje variantu, kdy každý z hráčů je u jiného počítače a pro tuto možnost je potřeba program povolit v bráně firewall. Pokud chcete hrát jinak než jenom v lokální síti, tak je potřeba mít nainstalovaný program Hamachi, ve kterém jeden z hráčů založí „network“, ke kterému se poté oba připojí. Hráč, který hru hostuje, napíše do textového pole pro IP „HOST“. Druhý hráč, pokud jsou oba ve stejné lokální síti, napíše do textového pole IPv4 hostujícího hráče (v příkazové řádce příkaz „ipconfig“). Pokud nejsou ve stejné lokální síti, tak připojující se hráč použije IP hosta v Hamachi. Hra se hraje myší, hráči kliknou na figurku, kterou chtějí táhnout a poté kliknou na políčko, které je označeno žlutým kruhem.

6) Informace o fungování programu

6.1) Použité technologie

Jako vývojové prostředí jsem použil NetBeans IDE 8.2. Hra byla naprogramována v JavaFX. K vytvoření obrázků figurek ve hře byl využit program GIMP.

6.2) Postup instalace a systémové požadavky

Ke spuštění hry je potřeba operační systém Windows pro počítače a nainstalovaná Java. Hra se spustí pomocí souboru Chess.

6.3) Struktura programu

Program se dělí na několik částí. První z nich je Main.java, která se stará o přepínání mezi scénami. Další částí jsou třídy, v nichž jsou naprogramovány jednotlivé scény, což jsou Chessboard, ChessboardMP, Settings, Statistics. Další částí je třída MovesCheck, která se stará o to, aby se prováděly platné tahy. Další částí je třída Timer, která se stará o aktualizaci času. Další částí je třída Receiver, která přijímá informace od serveru o provedených tazích a realizuje je. Poslední částí je server, který zajišťuje posílání tahů mezi hráči.

6.4) Datové struktury

Z datových struktur je v mém programu nejvíce zastoupen int pro uložení jednoduchých informací. Dále je využíván ArrayList pro uložení jednotlivých figurek, tlačítek a obrázků figurek. Také je použit String, např. pro uložení změn, které se poté pošlou přes server druhému hráči.

7) Popis zdrojového kódu

7.1) Main.java

V tomto souboru se volají konstruktory tříd, ve kterých jsou definovány scény hry. Také je zde zajištěno jejich přepínání. Zde se také vytváří scéna pro menu, takže se zde vytváří tlačítka, nastaví se jejich styl akce (změna aktuální scény) a přidají se do správce zobrazení VBox. Dále se v tomto souboru nastavují akce tlačítek, která jsou v jiných scénách než jen menu, ale je jejich účelem změnit scénu, např. tlačítka zpět, potvrdit.

7.2) Chessboard.java

Tato třída má za úkol vytvořit scénu pro samotnou hru. Načtou se v ní obrázky, vytvoří se šachovnice, na ni se umístí načtené obrázky reprezentující šachové figurky a označení možných tahů, nastaví se jim akce, co se má stát, když se na ně klikne. Také se nastaví, na které lze kliknout a které jsou neaktivní. Když hráč klikne na šachovou figurku, tak se zavolá metoda třídy MovesCheck, která vrátí pozice, kam je možné s danou figurkou táhnout (kromě rošády, která se řeší samostatně). Tyto pozice se vyznačí žlutými kruhy. Pokud hráč klikne znovu na jinou figurku, tak se zobrazí možné tahy té jiné figurky. Pokud poté klikne na žlutý kruh, figurka se tam přesune a poté se figurky stejné barvy zablokují a odblokují se figurky druhé barvy. Takto se hráči střídají, dokud hra neskončí. Po každém tahu se kontroluje, jestli nenastal pat, nebo mat. Hra také může skončit, když jednomu z hráčů dojde čas. Hry, které se odehrávají přes tuto třídu, se neukládají do statistiky.

7.3) MovesCheck.java

Tato třída má metodu, která vrací možné tahy. Metoda dostane jako parametr ArrayList, ve kterém jsou uloženy pozice figurek a index, který označuje figurku, se kterou by hráč chtěl táhnout. Nejprve se rozhodne, jaká figurka to je a jakou má barvu a podle toho se zavolá příslušná metoda, které se jako parametr opět dá ArrayList s pozicemi figurek a index zkoumané figurky. Například figurka je černý střelec, takže se zavolá metoda pro černého střelce, která zjistí, kam se může úhlopříčným pohybem dostat. Poté se zavolá metoda, která dostane jako parametr index zkoumané figurky, ArrayList s figurkami a ArrayList s pozicemi, na které se může zkoumaná figurka přesunout. Tato metoda poté provede jednotlivé tahy a vyzkouší, že se tím vlastní král nedostane do šachu. To se testuje tak, že se zkusí za krále postupně dosadit různé figurky, a když tato figurka může na pozici, která je obsazena stejnou figurkou jiné barvy, tak to znamená, že král by byl v šachu a tento tah se vyloučí.

7.4) Timer.java

Tato třída funguje jako vlákno, které se stará o aktualizaci zbývajících času jednotlivých hráčů. Při každé hře jsou vytvořeny dva Timery, jeden se stará o aktualizaci času bílého a druhý o aktualizaci černého hráče. Vždy, když je daný hráč na tahu, tak daný Timer je aktivní a každou vteřinu aktualizuje čas, když není na tahu, tak čeká a nic nedělá.

7.5) ChessboardMP.java

Tato třída je velmi podobná třídě Chessboard. Liší se zejména v tom, že na začátku buďto spustí vlákno se serverem a připojí se k němu (pokud v nastavení bylo do IP napsáno

„HOST“), nebo se k serveru pouze připojí a přijme od něj nastavení (celkový čas jednotlivých hráčů, čas na tah). Dále se liší v tom, že každá změna se zapíše do Stringu, který se po konci tahu vždy odešle serveru, který tuto změnu předá druhému hráči. Také se po konci každého tahu odešle další String, ve kterém je uložen stav hry (jestli hra běží, již skončila matem, patem). Také odblokování figurek probíhá jinak. Poté, co se tah hráče odešle, se mu zablokují figurky, ale odblokuje mu je Receiver poté, co přijme tah od druhého hráče. Hry, které probíhají přes tuto třídu se ukládají do statistiky.

7.6) Receiver.java

Tato třída funguje jako vlákno. Vytvoří se a spustí v konstruktoru ChessboardMP, jako parametr kromě jiného dostane BufferedReader, ze kterého získává informace od serveru. Konkrétně přijímá tahy provedené druhým hráčem od serveru a realizuje je (posunuje figurky). V prvním Stringu, který dostane je provedený tah a ve druhém je stav hry, např. jestli pořad ještě pokračuje, nebo jestli už skončila matem, nebo patem.

7.7) ServerThread.java

Tato třída funguje jako vlákno, které vytvoří instanci třídy ChessServer, aby běžela v samostatném vlákne. Toto vlákno se vytvoří a spustí v konstruktoru třídy ChessboardMP, pokud daný hráč zrovna hostuje hru.

7.8) ChessServer.java

Tato třída má za úkol přeposílat tahy a informace o stavu hry mezi hráči. Její instance se vytvoří ve vlákne ServerThread. Server poslouchá na portu 8080. Nejprve se připojí hráč, který hru hostuje a pro komunikaci s ním si server vytvoří socket a přes něj BufferedReader a BufferedWriter. Poté se připojí druhý hráč a celé se to opakuje. Potom hostující hráč pošle serveru nastavení hry, které poté server přepošle druhému hráči. Poté, dokud hra neskončí matem, nebo patem, tak server vždy od jednoho hráče přijme String s provedeným tahem a String se stavem hry a přepošle to druhému hráči.

7.9) Statistics.java

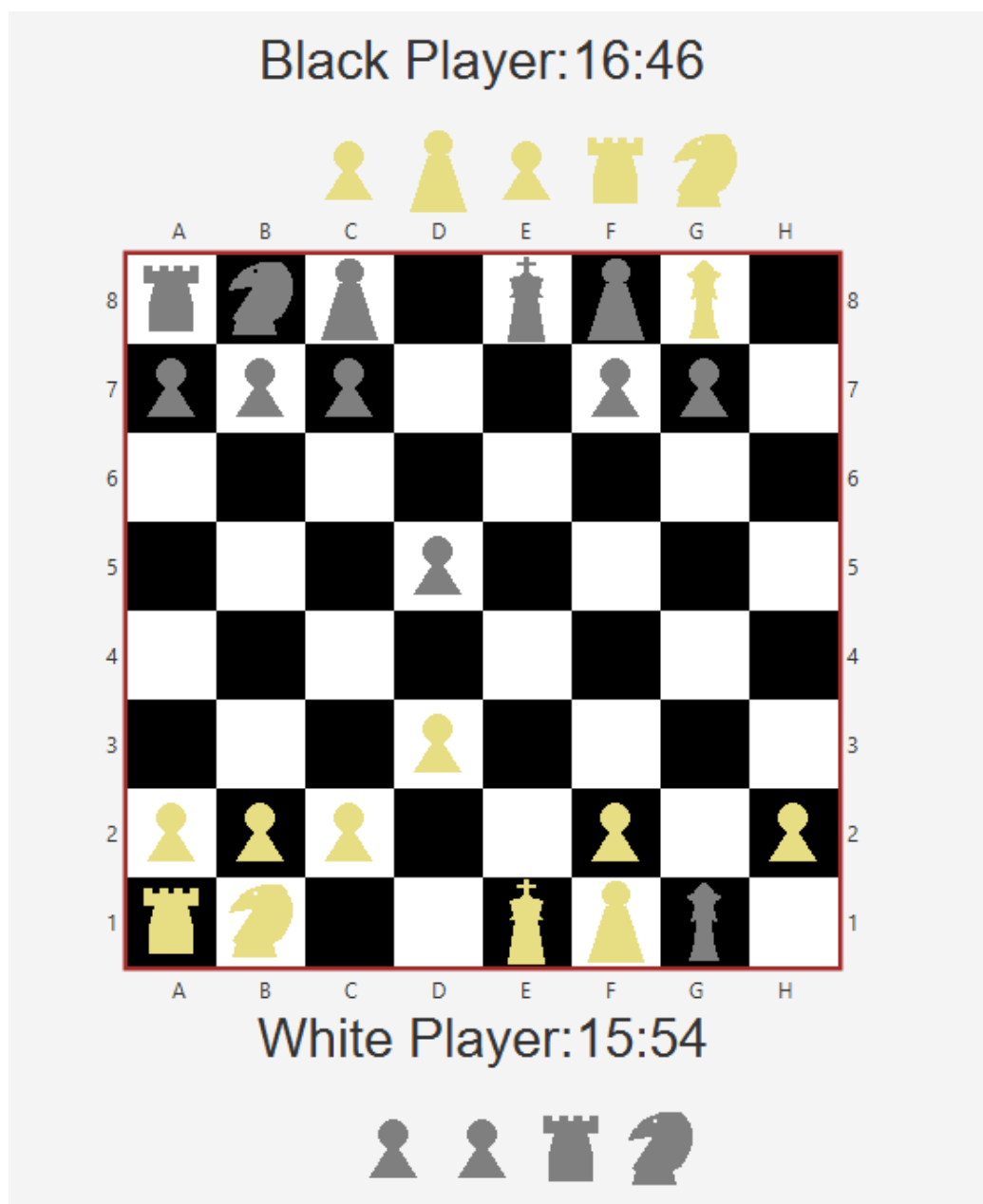
Tato třída má za úkol vytvořit scénu pro statistiku. Ze souboru „Statistika“, pokud existuje, se načte počet výher, počet proher a poměr výher a proher. Pokud soubor neexistuje, tak se vytvoří nový a uloží se do něj tři nuly a poté se načte. Sloupce jsou udělány přes dva správce zobrazení VBox, do kterých jsou vloženy třídy Label, ve kterých jsou načtená čísla a popisy k nim.

7.10) Settings.java

Tato třída má za úkol vytvořit scénu pro nastavení hry. V levém sloupci je popis toho, co se nastavuje v políčkách v pravém sloupci. Sloupce jsou udělány přes dva VBoxy a jsou spojeny HBoxem. V pravém sloupci byly pro zadávání pouze celočíselných hodnot použity instance třídy Spinner. Jako parametr se jim při vytváření zadala minimální, maximální a počáteční hodnota a o kolik se hodnota mění. Pokud se jedná o nastavování hry přes internet, tak se ještě zobrazí textové pole pro zadání IP.

8) Šachovnice

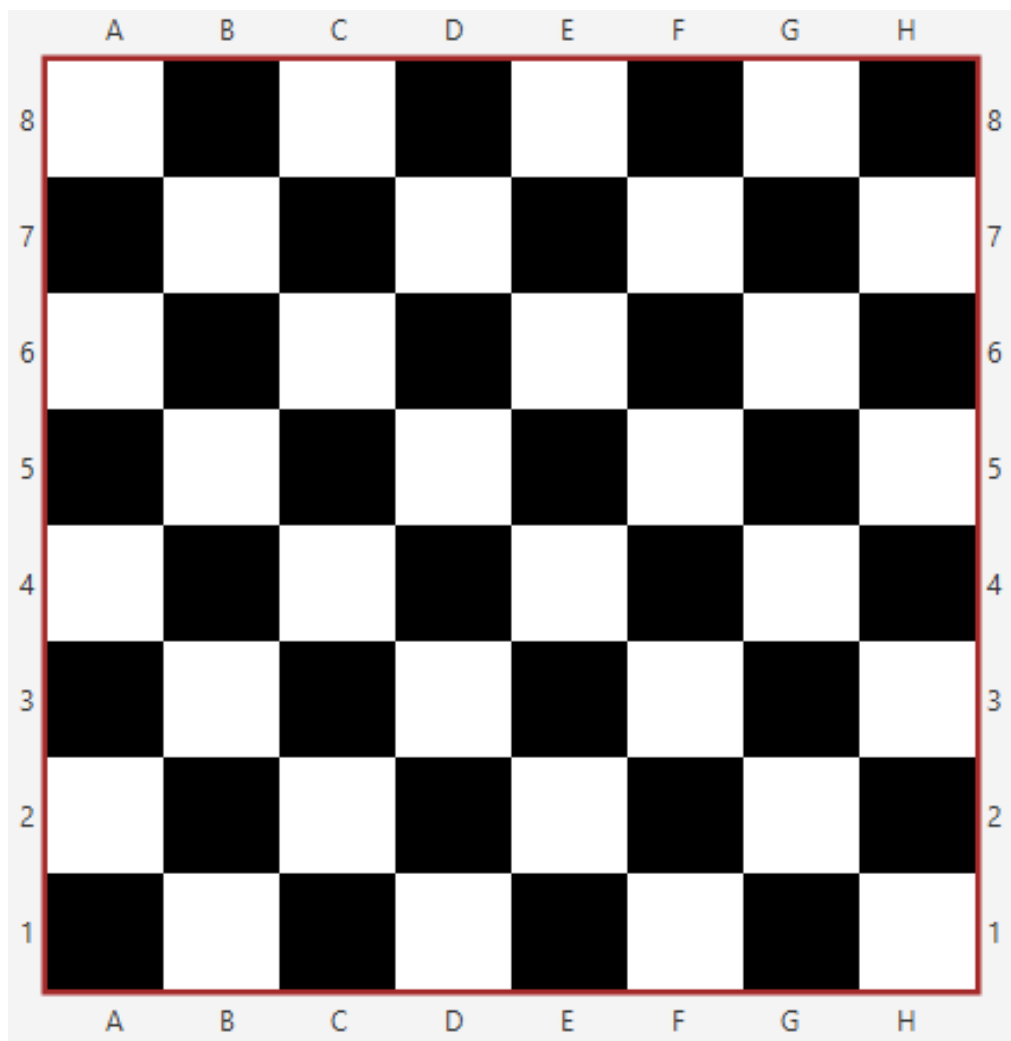
Šachovnice s figurkami je složena ze tří vrstev, které se skládají z více prvků a jsou k nim poté ještě přidány další prvky. Skládá se ze samotné šachovnice s označenými sloupci a řádky, ze čtverce 8x8 složeného ze žlutých kruhů označujících možné tahy a ze čtverce 8x8 složeného z šachových figurek a žlutých kruhů. Dále je k šachovnici pomocí VBoxu připojeno zobrazení časů a vyhozených figurek. Také jsou k šachovnici pomocí HBoxu připojena tlačítka pro povýšení pěšce, která se zobrazí tehdy, když se nějaký pěšec dostane na druhou stranu šachovnice. Obrázky figurek a označení pro možný tah jsou uloženy ve formátu png ve složce images a byly vytvořeny v programu GIMP.



Ukázka ze hry

8.1) Šachovnice

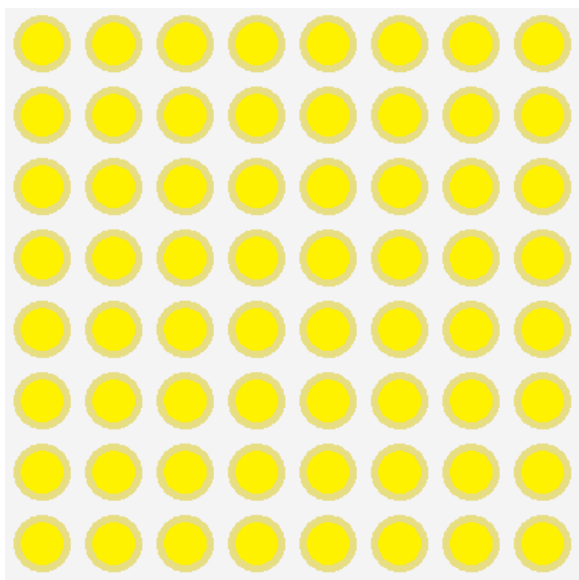
Šachová políčka jsou vykreslena a obtažena pomocí Canvasu. Pomocí HBoxů a VBoxů se přidalo označení šachovnice.



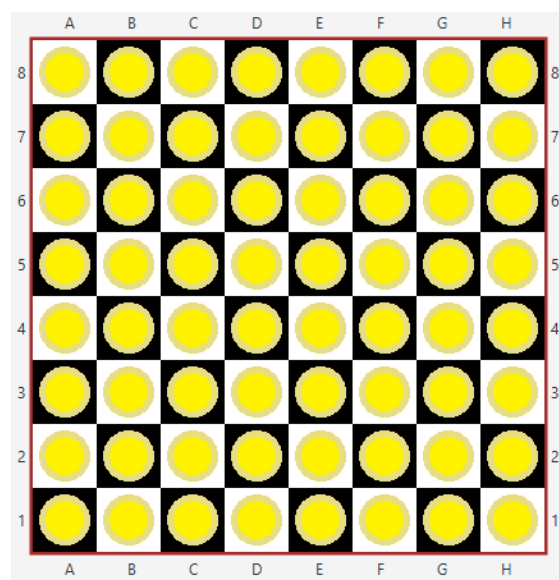
Šachovnice vykreslená pomocí Canvasu

8.2) Označení možných tahů

Tato vrstva se skládá pouze ze 64 žlutých kruhů, každý z nich se nachází na jednom políčku. Žluté kruhy jsou zobrazeny přes třídu ImageView. Normálně jsou nastavena tak, aby nebyla vidět, ale když hráč označí nějakou figurku, tak se zobrazí. Tato vrstva je tu proto, aby nemusely u každé figurky existovat dvě varianty textury (jedna standardní a druhá, když lze táhnout na její pozici), jelikož takto se pod figurkou zobrazí žlutý kruh.



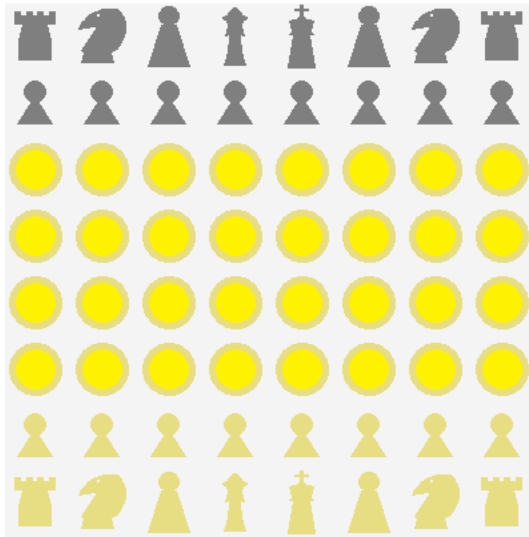
Vrstva sloužící k zobrazení možných tahů



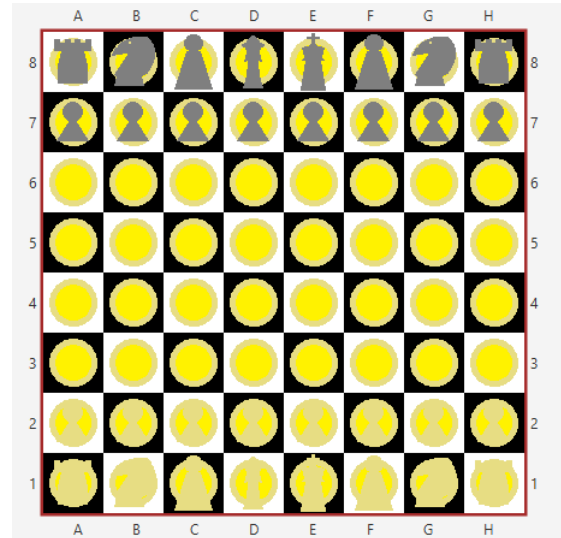
Šachovnice s vrstvou zobrazující možné tahy

8.3) Figurky

Tato vrstva se skládá z šachových figurek a žlutých kruhů, které jsou zobrazeny pomocí třídy `ImageView`. Na začátku jsou zobrazeny pouze šachové figurky, ale když poté hráč nějakou označí, tak se zobrazí žluté kruhy na políčkách, kam lze táhnout (kromě těch, kde stojí figurka opačné barvy, na tom místě se pod ní zobrazí žlutý kruh z předchozí vrstvy).



Zobrazení vrstvy obsahující figurky a označení možných tahů



Šachovnice s vrstvou možných tahů a vrstvou figurek

9) Klíčové problémy

9.1) Posílání tahu mezi hráči

Tahy mezi hráči se přeposílají přes server v podobě Stringu. Odesílání se odehrává v ChessboardMP, konkrétně v akci kliknutí na figurku nebo na pozici, kam s ní lze táhnout. Všechny změny se postupně přidávají do Stringu, který se poté odešle a jeho obsah se vymaže. Když se jedná o standardní tah, kdy se pohne pouze jedna figurka, tak je String složen ze dvou částí. První část je index políčka, ze kterého se figurka přesunula, a druhá část je index políčka, kam se přesunula. Obě části jsou odděleny mezerou. Když provedeným tahem byla rošáda, tak kromě první a druhé části má String třetí a čtvrtou část. V prvních dvou částech je popsán přesun krále a ve třetí a čtvrté části je popsán přesun věže. Pokud po provedeném tahu došlo ještě k povýšení pěšáka, tak má String ještě třetí a čtvrtou část. Ve třetí části jsou souřadnice povýšené figurky a ve čtvrté je, na jakou figurku byla povýšena. String je Receiverem rozložen a tahy v něm popsány jsou realizovány.

9.2) Předání hodnot z třídy Settings třídě Chessboard

Hodnoty z instance třídy Settings se instancí třídy Chessboard předávají přes Menu. Funguje to tak, že akce tlačítka vytvořeného ve třídě Settings se nastavuje v Menu. Dále tlačítko potvrdit, instance Spinner třídy a textového pole jsou nastaveny jako proměnné instance třídy Settings. Takže když hráč zadá nastavení a poté klikne na tlačítko potvrdit, tak se provede akce definovaná v Menu. Vytvoří se nová instance třídy Chessboard, nebo ChessboardMP a jako parametry se dají jejímu konstruktoru proměnné třídy Settings.

10) Závěr

Zadání jsem z velké části splnil. Dva hráči můžou hrát proti sobě u jednoho počítače, nebo každý může být u svého, což funguje nejenom v lokální síti, ale s pomocným programem Hamachi i přes internet. Před začátkem hry lze nastavit celkový čas a čas na jednotlivé tahy. Dále se zaznamenává statistika výher a proher. Také se označují pozice, kam lze s figurkou táhnout. Při vývoji hry jsem se naučil nové věci o vláknech a soketech a naučil se používat třídu Spinner.

11) Zdroje

Při tvorbě této práce jsem vycházel pouze z dokumentace Javy a JavaFX.

Dokumentace Java:

Není uveden, z 28. 4. 2020, Java documentation [online] dostupné z

< <https://docs.oracle.com/en/java/> >