

Spše Ječná
Informační Technologie
Ječná 30, 121 36, Praha 2

Šachy
(Lokální šachy 1v1, graficky)

Oliver Janusz Pecka
IT
2025

Obsah

1	Cíl práce	3
2	Popis hry	3
2.1	Příběh/Algoritmus	3
2.2	Postavy	3
2.3	Mechaniky	3
3	Systém requirements	3
4	Základní struktura	4
5	Testovací data	4
6	Uživatelská příručka	4
7	Závěr	4
8	Zdroje	5

1 Cíl práce

Cílem projektu je vytvořit graficky lokální 1V1 (hráč proti hráči) šachovou partii. Uživatelé by měli být schopni zahrát tah, vrátit tah, vyhrát, remizovat nebo prohrát, kouknout se na historii her, smazat historii her.

2 Popis hry

Jedná se o jednoduchou partii šach.

2.1 Příběh/Algoritmus

Hra nemá žádný příběh, je to klasická šachová partie. V programu je zajištěno, že se kontroluje, jestli je tah platný, jestli je šach, mat nebo pat, a jsou tam i speciální pravidla jako rošáda nebo en passant. Dál jsem přidal i kontroly na tři opakování stejné pozice a 50 tahů bez tahu pěšcem nebo braní.

2.2 Postavy

Král, dáma, věž, střelec, jezdec, pěšec. Všechny postavy mají jiné možnosti tahu.

2.3 Mechaniky

Mezi hlavní mechaniky patří:

- Tahy pomocí myši
- Zvýraznění legálních tahů
- Detekce šachu a matu
- Podpora undo/redo s uchováním historie
- Grafické znázornění tahové notace
- Záznam do souboru (historie a statistiky)

3 Systém requirements

-Jazyk: Java (verze 8+)

-JRE: Java Runtime Environment

-OS: Windows, macOS, Linux

Minimální hardware:

- CPU: 1 GHz
- RAM: 512 MB
- Rozlišení: 1024×768

4 Základní struktura

Program je napsaný v jazyce Java a je rozdělený do více tříd, které spolu navzájem spolupracují. Každá třída má svůj účel. Hlavní třída se jmenuje Main a slouží k tomu, aby spustila celou aplikaci. Vytvoří šachovnici, časovač a otevře okno hry.

Třída ChessBoardPanel zobrazuje samotnou šachovnici. Ukládá aktuální stav hry a zajišťuje, že se figurky správně vykreslují. Uživatel kliká myši na figury a šachovnice reaguje podle toho, co hráč udělá. Tato třída spolupracuje s GameController, který se stará o historii tahů, možnost vrátit tah (undo) nebo ho znovu provést (redo) a střídání hráčů.

Třída GameLog ukládá tahy, které hráči provedli, a zobrazuje je v šachové notaci. Díky tomu je možné vidět, jak hra probíhá. Třída SpecialMoves obsahuje pravidla šachu, jako třeba jak se figury mohou pohybovat, kdy je šach, mat, rošáda nebo en passant.

Všechny třídy spolu komunikují pomocí metod, jako jsou makeMove(), calculateLegalMoves() nebo registerMove(). Celkově je program napsaný tak, aby byl přehledný a dal se v budoucnu jednoduše rozšířit třeba o hraní proti počítači.

5 Testovací data

Byly provedeny unit testy:

- Správná detekce šachu
- Správná detekce patu (remízy bez legálního tahu)
- Generování tahů pro různé figury, včetně překážek
- Ověření třikrát opakované pozice (pravidlo o remíze)
- Test toho, jak se pohybuje kůň

Testy se nacházejí v souboru SpecialMovesTests.java.

6 Uživatelská příručka

Program se ovládá prostřednictvím grafického rozhraní (GUI) pomocí myši. Každý hráč střídavě kliká na šachovou figuru a následně na cílové pole, kam chce figuru přesunout. Legální tahy jsou při výběru figury vizuálně zvýrazněny pomocí barevných indikátorů. Hráč tedy nemusí zadávat žádné textové příkazy ani znát notaci tahů.

Program po každém tahu automaticky aktualizuje stav šachovnice, zvýrazní poslední provedený tah a případně zobrazí šach nebo mat. V pravé části okna se průběžně zobrazuje historie tahů v algebraické notaci.

Uživatel má také k dispozici následující funkce:

- Undo** – krok zpět (vrátí předchozí tah)
- Redo** – krok vpřed (znovu provede tah)
- Draw** – nabídka remízy (po dohodě obou hráčů)
- Zobrazování šachu nebo matu** – program vizuálně upozorňuje, pokud je některý král v šachu
- Automatická detekce konce hry** – při matu, patu nebo pravidlech o remíze

Na konci hry se výsledek automaticky uloží do souboru historie a uživatel může zobrazit statistiky nebo začít novou partii z hlavního menu.

7 Závěr

Kód jsem převážně psal koncem května, takže jsem několik dní seděl u počítače velice dlouho. Trošku lituji, že jsem nezačal dříve, ale nakonec se mi to podařilo napsat, takže to snad dobře dopadlo. Při psaní tohoto projektu jsem se velice hodně sblížil s Jpanelem a ostatními podobnými prvky. Také mi to připomnělo, kolik mají šachy vlastně pravidel, na které jsem časem úplně zapoměl. Také jsem se více sblížil s GitHubem a zjistil, jak se dělá z projektu .jar file.

Abych to ukončil, řekl bych, že se mi ten kód docela povedl, ikdyž jsem si velice dlouho drbal hlavu nad metodami jako `redo()`, tak jsem to nakonec dotáhl a vše podle mě funguje tak, jak má.

8 Zdroje

Použity standardní knihovny Javy

JPanel | Java Swing Tutorial for Beginners – Java Code Junkie (Youtube)

Na testy byl použit JUnit 5