

SPŠE Ječná
Informační technologie
Ječná 30, Praha 2, 120 00

Šachy
Šachy lokálně 1v1, graficky

Šimon Pavelka
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. System requirements	3
4. Základní struktura	3
5. Testovací data	4
5.1 GameTest	4
5.2 KnightTest	5
5.3 PawnTest	5
6. Uživatelská příručka	5
7. Závěr	5
8. Zdroje	5

1. Cíl práce

Cílem projektu je vytvoření grafických šachů lokálně hráč proti hráči. Hra by měla splňovat všechny pravidla šachů – umožňovat hráči tahy figurkami, vyhrát, remizovat nebo začít novou partii.

2. Popis hry

2.1 Příběh/Algoritmus

Jedná se o klasické šachy – bílý a černý se střídají v tazích, po každém tahu se kontroluje, zda král jednoho z hráčů není v šachu, případně jestli nenastal mat nebo pat. Program obsahuje i speciální tahy, jako je rošáda nebo en passant. Program je graficky zpracovaný – hráči s ním interagují pomocí kliknutí myši.

2.2 Postavy

Postavy jsou král, dáma, věž, střelec, jezdec a pěšec – všechny v bíle a černé variantě. Každá postava má jiné možnosti tahu.

2.3 Mechaniky

Hráči se střídají v jednotlivých tazích, klikají myší na postavy – po kliknutí se jim zobrazí dostupné tahy dané postavy jejich barvy. Pokud hráč dojde pěšcem na druhou stranu šachovnice, zobrazí se okno, ve kterém si hráč může vybrat figurku, na kterou chce pěšce proměnit.

Dále hra obsahuje červené zvýraznění krále, pokud se nachází v šachu. Pokud nastane mat nebo pat, výsledek se vypíše na obrazovku v novém okně. Vpravo se nachází tlačítko pro spuštění nové hry.

3. System requirements

Program vyžaduje pro spuštění minimálně Java Development Kit (JDK) 17. Kromě toho je nutné mít přidanou knihovnu Apache Batik 1.19, která slouží pro vykreslování SVG šachových figurek. Žádné další knihovny nejsou pro běh aplikace nutné.

4. Základní struktura

Program je navržen objektově a rozdělen do několika základních tříd, které společně zajišťují herní logiku, grafické rozhraní a interakci s uživatelem. Výchozím bodem programu je třída Main, která inicializuje hlavní okno (MyFrame) a spouští grafické rozhraní hry.

Grafická část programu je řešena pomocí knihovny Java Swing a je rozčleněna do následujících komponent:

- MyFrame představuje hlavní okno programu

- BoardPanel slouží jako vizuální reprezentace šachovnice a zajišťuje zpracování kliknutí na jednotlivá pole
- SquarePanel reprezentuje jednotlivá pole na šachovnici

Samotnou logiku hry zajišťují hlavně následující třídy:

- Game – třída zajišťující řízení průběhu hry, správu tahu, sledování stavu partie (včetně šachu, matu, remízy) a vyhodnocování koncových podmínek
- ChessBoard – objektová reprezentace šachovnice, která uchovává aktuální rozmístění figur a poskytuje metody pro manipulaci s nimi
- Piece – abstraktní třída reprezentující šachovou figuru, z níž dědí konkrétní třídy jako Pawn, Rook, Knight, Bishop, Queen a King

Každá figurka implementuje metodu `getLegalMoves()`, která vrací seznam legálních tahů z dané pozice podle aktuálního stavu šachovnice. Třída `Move` slouží jako datová struktura pro uchování informace o jednotlivém tahu, včetně specifik jako je rošáda, en passant nebo proměna pěšce.

Komunikace mezi komponentami je zajištěna pomocí metod jako, `isKingInCheck()`, `getLegalMoves()`, `checkEndConditions()` apod., které propojují herní logiku s grafickým rozhraním.

5. Testovací data

Program je možné manuálně testovat zkoušením různých situací, ve kterých je pravděpodobné, že by se v nich mohla vyskytnout chyba. Mezi takové situace patří třeba správná detekce šachu, zvláště pak v určitých situacích – při proměně pěšce na jinou figurku platí, že takové figurka okamžitě může dát šach. Hráč také nesmí táhnout figurkami, kterými by vlastního krále přivedl do šachu. Dále pak otestování správné funkčnosti rošády – ta nesmí být provedena, pokud již bylo v partii jednou taženo věží nebo králem, dále pak nelze provést v případě, že je král v šachu a nebo by v šachu byl v průběhu rošády.

Dále jsem provedl 5 unit testů, které se nachází ve třídách `GameTest`, `KnightTest` a `PawnTest`.

5.1 GameTest

Ve třídě `GameTest` jsem testoval některé z výše popsaných situací – testoval jsem správnou funkčnost vyhodnocení šachu, dále jsem testoval, že rošádu nelze provést v případě, že je král v šachu a poslední test se zaměřil na funkčnost vyhodnocení matu.

5.2 KnightTest

Ve třídě KnightTest jsem testoval funkčnost pohybu jezdce. První test se zaměřil na otestování počtu možných tahů z prostředku šachovnice, kterých má být 8. Další dva testy se zaměřili na ověření, že jezdec nemůže táhnout na pole obsazená vlastními figurkami a že může sebrat nepřátelské figurky.

5.3 PawnTest

Ve třídě PawnTest jsem se zaměřil na funkčnost speciálního tahu en passant.

6. Uživatelská příručka

Program je graficky zpracovaný a uživatelé s ním interagují klikáním na vlastní figurky, po kterých se jim zobrazí všechny dostupné tahy danou figurkou. Po uskutečnění tahu zůstanou zvýrazněná pole, odkud a kam hráč táhl.

Když nastane šach, král se zvýrazní červeně. Pokud se hráči podaří pěšcem přejít až na druhou stranu šachovnice, zobrazí se mu nové okno, ve kterém se může vybrat, na jakou figurku chce pěšce přeměnit.

Pokud nastane mat nebo pat, otevře se nové okno, ve kterém se vypíše, kdo vyhrál, případně že nastala remíza. Kdykoliv je také možné začít novou hru stisknutím tlačítka new game.

7. Závěr

Jsem rád, že jsem si zvolil zrovna šachy, protože to není zrovna nejnáročnější projekt, ale zároveň je dostatečně těžký na to, aby mi pomohl zlepšit se v programování a používání Swingu.

Největší problém jsem měl asi s načítáním svg souborů. Rozhodl jsem se program totiž udělat responzivní a omezil jen minimální velikost, takže mi přišlo lepší místo png souborů použít svg. Dlouho se mi je nedařilo načíst, ale nakonec se mi to s pomocí asistence od ChatGPT podařilo. I tak jsem z výsledku ale trochu zklamaný, protože jsou i přes použití svg figurky stále trochu rozmazané a pořádně doted' netuším proč.

Na projektu jsem pořádně začal pracovat vlastně až na poslední chvíli, čehož lituji, protože jsem kvůli časovému presu musel zavrhnout i výpis a historii jednotlivých tahů, které jsem měl původně v plánu. Zároveň jsem kvůli tomu nestihl včas udělat Unit testy a dokumentaci. Nepovedlo se mi bohužel vygenerovat funkční jar file – nejde mu načíst svg soubory.

8. Zdroje

[1] Apache Batik 1.19 – The Apache Software Foundation. Oficiální dokumentace knihovny pro práci se SVG v Javě.

<https://xmlgraphics.apache.org/batik/> [cit. 2025-06-03].

[2] JUnit 5 – Oficiální dokumentace k nástroji pro jednotkové testování v jazyce Java. <https://junit.org/junit5/> [cit. 2025-06-03]

[3] ChatGPT – Asistence při řešení problémů se zpracováním SVG
<https://chat.openai.com> [cit. 2025-06-03].

[4] Šachové SVG figurky

<https://github.com/lichess-org/lila/tree/master/public/piece/alpha> [cit. 2025-06-03].