Semestrálna práca: Šachy

# Úvod:

Testovaná aplikácia Šachy bola implementovaná ako semestrálna práca pre predmet PJV v letnom semestri 2020/2021 mnou, Timotejom Barbušom. Následne bola aplikácia využitá aj ako semestrálna práca pre predmet TS1, ktorej cieľom je otestovať danú aplikáciu. Aplikácia Šachy je hra implementovaná pomocou programovacieho jazyka Java, sieťová hra a ukladanie hier sú naopak vytvorené pomocou JSON.

# Popis funkcionality aplikácie:

Funkčné požiadavky:

* FP 1: Možnosť výberu typu hry (singpleplayer, multiplayer local, multiplayer network).
* FP 2: Kompletná kontrola pravidiel hry (rošáda, vylepšenie pešiaka, ťahy figúrok...).
* FP 3: Možnosť uloženia a opätovného načítania hry.
* FP 4: Možnosť manuálneho umiestnenia figúrok.
* FP 5: Šachové hodiny počítajúce čas obom hráčom.
* FP 6: Možnosť výberu farby figúrok.
* FP 7: Možnosť nastaviť šachové hodiny na ľubovoľný čas.

Aplikácia dáva užívateľovi možnosť vybrať si z troch rôznych typov hry:

1. Singleplayer

Hra jedného hráča proti počítaču. Počítač náhodne generuje ťahy, pričom nevyužíva žiadnu logiku.

1. Multiplayer local

Hra dvoch hráčov súperiacich proti sebe, pričom obaja hráči ovládajú aplikáciu z jedného spoločného počítača.

1. Multiplayer network

Hra dvoch hráčov súperiacich proti sebe. Na rozdiel od predchádzajúceho typu, tentokrát majú obaja hráči otvorenú aplikáciu na svojich počítačoch, jedná sa teda o sieťovú komunikáciu.

Základom pre užívateľa je teda možnosť hýbať sa s figúrkami na šachovnici, s cieľom poraziť druhého hráča. Hráč ktorý ako prvý dostane šach-mat prehráva hru.

# Prehľad častí aplikácie:

* Enums: Tento package definuje dve veľmi dôležité Enum classy, prvá definuje farbu figúrok a teda aj farbu hráčov, druhá definuje typ figúrok.
* Figures: Package obsahuje hlavnú abstract classu, ktorá ďalej definuje všetkých 6 typov figúrok, pričom siedmim typom je figúrka „Empty“, ktorá definuje prázdne políčko na šachovnici.
* Game: Obsahuje hlavnú classu „Main“ obsluhujúca celý program a takisto classu „Clock“, ktorá počíta čas hráčom, už od začiatku hry.
* Menus: Daný package obsahuje classy definujúce set-up menu a takisto in-game menu.
* Moves: Najdôležitejší package zo všetkých, na základe súradníc figúrok na šachovnici classy v tomto package kontrolujú možné pohyby pre všetkých hráčov.
* NetworkGame: Package obsahujúci classy potrebné pre vytvorenie spojenia pri sieťovej hre a následné prijímanie a posielanie JSON súborov medzi nimi.
* SaveLoad: Podobne ako predchádzajúci package, tento takisto pracuje s JSON súbormi. Na rozdiel ale od predchádzajúceho, tieto JSON súbory obsahujúce súradnice figúrok a všetky potrebné premenné ukladá do počítača. Neskôr je možné tieto súbory znova načítať a takto opätovne pokračovať v už začatej hre šachu.
* View: Tento package obsahuje všetky grafické classy programu, ktoré generujú samotnú šachovnicu alebo menu.

# Priorita častí aplikácie:

Pri každej z hlavných funkčných častí aplikácie je uvedená pravdepodobnosť zlyhania a taktiež miera poškodenia, ktorá pri chybe v programe hrozí. Z týchto informácií je následne odvodená celková trieda rizika podľa vzťahov v tejto tabuľke:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tabuľka celkovej triedy rizika (TR) | | Pravdepodobnosť zlyhania (PZ) | | |
| **HIGH** | **MEDIUM** | **LOW** |
| **Možné poškodenie (MP)** | **HIGH** | HIGH | MEDIUM | MEDIUM |
| **MEDIUM** | MEDIUM | MEDIUM | LOW |
| **LOW** | LOW | LOW | LOW |

*Low – L, Medium – M, High – H*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Proces** | **Požiadavka** | **Časť systému** | **MP** | **Vysvetlenie možného poškodenia** | **PZ** | **Vysvetlenie pravdepodobnosti zlyhania** | **TR** |
| Šach-mat | Hráč musí byť v šachu | Pohyby a ich kontrola | H | Chyba pri kontrole platnosti šachu alebo pri kontrole šach-matu znemožní ukončenie hry. | H | Jedná o množstvo komplikovaných funkcií definujúcich pohyby figúrok, ktoré spolu komunikujú. | H |
| Vytvorenie sieťovej hry | IP adresa a socket | Sieťová hra | L | Chyba pri vytváraní sieťovej hry sa prejaví okamžite a vytváranie hry sa týmto resetuje. | M | Chyba v komunikácií medzi počítačmi sa vyskytla niekoľkokrát počas implementácie. | L |
| Načítanie uloženej hry | Zadanie názvu súboru | Save-load | L | Chyba pri načítaní uloženej hry nespôsobí žiadne komplikácie a aktuálna hra môže pokračovať. | L | Zadanie neplatného názvu súboru alebo lokácie. | L |
| Vylepšenie pešiaka | Pešiak sa nachádza na poslednom políčku | Pohyby a ich kontrola | H | Chyba spôsobí, že sa pešiak nezmení na inú „lepšiu“ figúrku. Následkom bude znemožnenie pohybu s daným pešiakom. | L | Jedná sa o pomerne krátku a jednoduchú operáciu. | M |
| Rošáda | Kráľ a veža sa od začiatku hry ešte nepohli | Pohyby a ich kontrola | L | Chyba pri rošáde nespôsobí žiadnu výraznú komplikáciu, obe figúrky budú naďalej funkčné... | M | Pomerne zložitá funkcia spoliehajúca sa na kontrolu rôznych faktov, ktoré môžu byť falošne platné. | L |
| Generovanie AI pohybov | Singleplayer hra | Pohyby a ich kontrola | H | Chyba pri generovaní náhodných AI ťahov spôsobí znemožnenie pokračovania v hre. | M | Jedná sa o pomerne dlhú funkciu, avšak celkovo nie až tak zložitú. | M |

# Test levels:

*Low – L, Medium – M, High – H*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Funkcia** | **TR** | **Revízia** | **Vývojárske testy** | **Systémové testy** | **UAT** | **Test v produkcií** |
| **Bezchybná funkcionalita** | | | | | | |
| Šach (Šach-mat) | H | Áno | Vysoká | Vysoká | Stredná | Áno |
| Vytvorenie sieťovej hry | L |  | Nízka | Nízka | Nízka |  |
| Načítanie uloženej hry | L |  | Nízka | Nízka | Nízka |  |
| Vylepšenie pešiaka | M |  | Stredná | Stredná | Nízka |  |
| Rošáda | L |  | Nízka | Nízka | Nízka |  |
| Generovanie AI pohybov | M |  | Stredná | Stredná | Stredná |  |
| **Výkonnosť** | | | | | | |
| Šach (Šach-mat) | H |  |  | Nízka |  |  |
| Generovanie AI pohybov | M |  |  | Nízka |  |  |
| **Užívateľská prívetivosť** | | | | | | |
| Grafické rozhranie | M |  | Stredná | Stredná | Stredná |  |
| **Bezpečnosť** | | | | | | |
| Vytvorenie sieťovej hry | L |  | Stredná | Nízka | Nízka |  |

# Triedy ekvivalencie (TE):

Pre analýzu tried ekvivalencie bola vybraná funkcia:

**checkIfCheckMate(EnumFiguresColor allyColor, AbstractFigures[][] board, Coordinates kingCoordinates)**

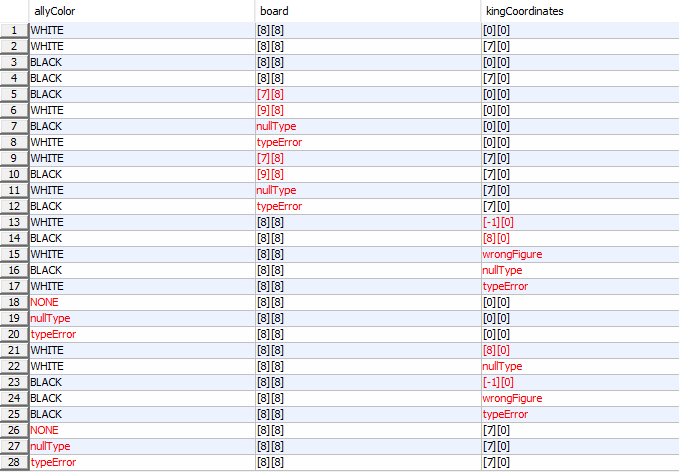
Táto funkcia očakáva na vstupe informácie o farbe hráča, súradnice všetkých figúrok na šachovnici a súradnice kráľa, ktorý bude kontrolovaný, či dostal šach-mat alebo nie. Funkcia vracia True ak daný kráľ dostal šach-mat, False ak nie je šach-mat.

Parametre funkcie sú:

* allyColor – je hodnota podľa typu EnumFiguresColor udávajúca farbu hráča
* board – je hodnota podľa typu AbstractFigures[][] udávajúca zoznam figúrok na šachovnici (obvykle o veľkosti 8x8)
* kingCoordinates – je hodnota podľa typu Coordinates udávajúca súradnice kráľa (A : B)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Názov TE** | **Validita TE** | **Hodnoty TE** | **Medzné podmienky** |
| allyColor | WHITE | Validná TE | WHITE | Neexistujú  (výčtový typ) |
| BLACK | Validná TE | BLACK |
| NONE | Nevalidná TE z business pohľadu | NONE |
| nullType | Nevalidná TE z technického podľahu | null |
| typeError | Nevalidná TE z technického podľahu | iný dátový typ |
| board | Veľkosť šachovnice je presne 8x8 | Validná TE | <8; 8> | AbstractFigures[7][8],  AbstractFigures[8][8],  AbstractFigures[9][8] |
| Veľkosť šachovnice je iná ako 8x8 | Nevalidná TE z business pohľadu | iná veľkosť ako AbstractFigures[8][8] |
| nullType | Nevalidná TE z technckého pohľadu | null |
| typeError | Nevalidná TE z technckého pohľadu | iný dátový typ |
| kingCoordinates | Súradnice o veľkosti max 8x8 & súradnice kráľa | Validná TE | <0; 7> | Čísla -1, 0, 7, 8 |
| Súradnice o veľkosti viac/menej ako 8x8 | Nevalidná TE z business pohľadu | <-2,147,483,648; 0)  <8; 2,147,483,647> |
| Súradnice inej figúrky | Nevalidná TE z business pohľadu | záleží od aktuálneho usporiadania |
| nullType | Nevalidná TE z technckého pohľadu | null |
| typeError | Nevalidná TE z technckého pohľadu | iný dátový typ |

# Pairwise testing:



Exportovaný výstup z aplikácie je priložený v súbore: checkIfCheckMate1-output.csv

# Detailný testovací scenár:

ID testu: CHECKMATE\_001

Názov testu: 3 ťahový šach-mat

Hĺbka testu: stredná

Zhrnutie testu: biely hráč dá šach-mat hráčovi s čiernymi figúrkami do 3 ťahov

Popis testu: Začínajúci užívateľ (biely hráč) sa pohne pešiakom (4; 6) o dve políčka dopredu (4; 4). Na to sa druhý hráč pohne pešiakom (4; 1) rovnako o dve políčka dopredu (4; 3). Hráč s bielymi figúrkami sa pohne kráľovnou z defaultnej pozície (3; 7) na pozíciu 7; 3. Nepriateľský kráľ sa pohne z defaultnej pozície (4; 0) o jedno políčko dopredu na 4; 1. Biely hráč sa následne pohne kráľovnou z pozície 7; 3 na pozíciu 4; 3, čím vyhodí nepriateľského pešiaka na tejto pozícií a zároveň dá šach-mat nepriateľskému kráľovi. Hrá sa skončila.

Vstupné podmienky: multiplayer (hráč proti hráčovi), defaultné rozloženie figúrok, začína hráč s bielymi figúrkami

Očakávaný výsledok: všetky pohyby figúrok prebehnú bez problémov, čierny kráľ dostane šach-mat do 3 ťahov bieleho hráča

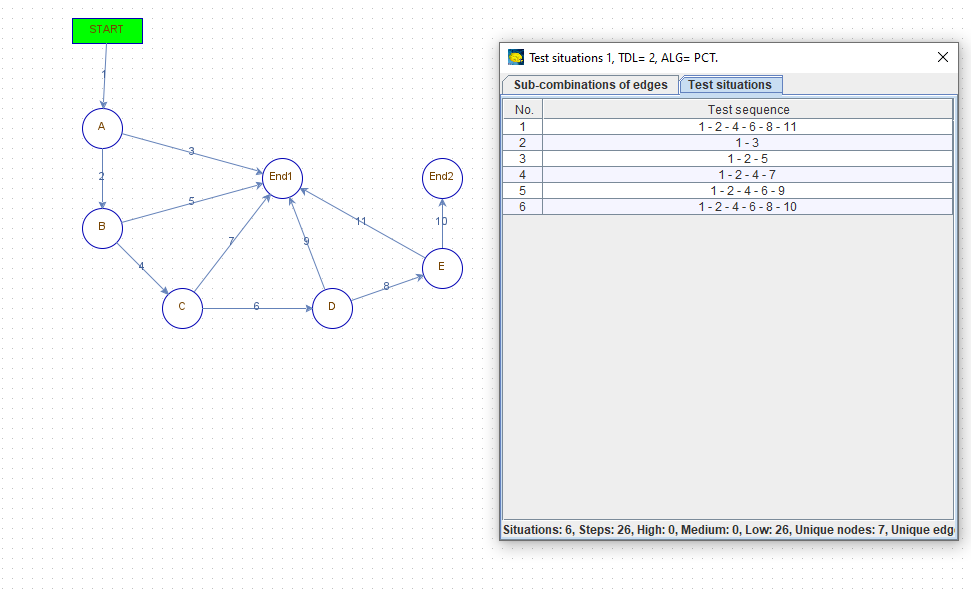
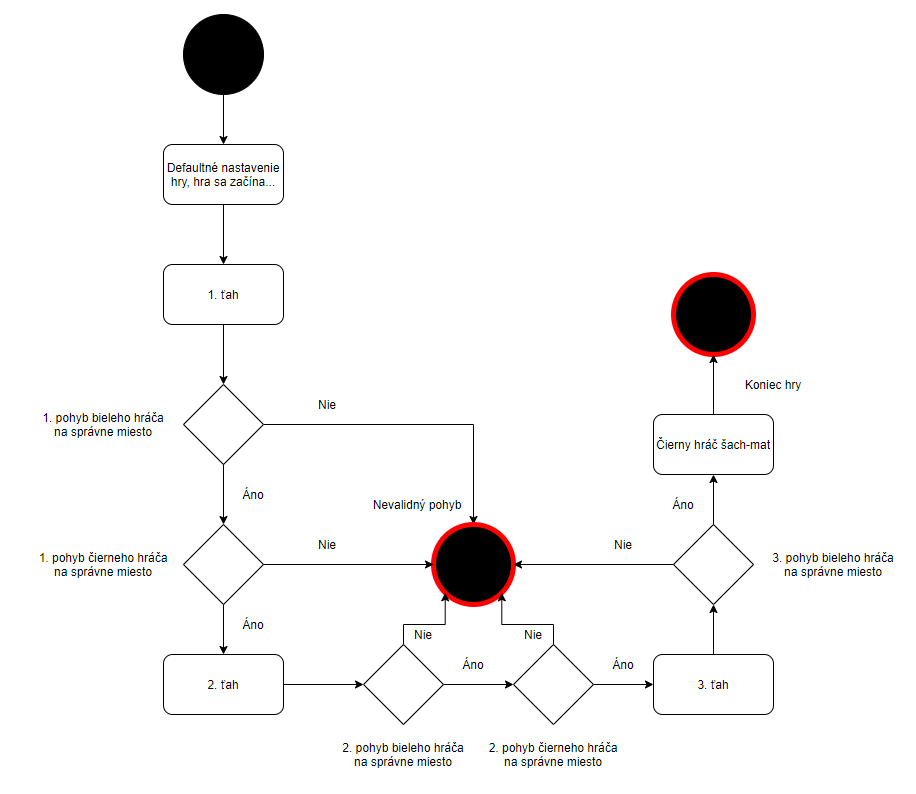
Autor: Timotej Barbuš ([barbutim@fel.cvut.cz](mailto:barbutim@fel.cvut.cz))

Stav prípravy: finálny

Kroky testovacieho scenára:

* Názov kroku testu: 1. pohyb bieleho hráča
* Popis kroku testu: Biely hráč sa pohne pešiakom z pozície 4; 6 na pozíciu 4; 4.
* Očakávaný výsledok: Figúrka sa premiestni na dané políčko. Zmení sa aktuálny hráč – na ťahu je čierny hráč.
* Názov kroku testu: 1. pohyb čierneho hráča
* Popis kroku testu: Čierny hráč sa pohne pešiakom z pozície 4; 1 na pozíciu 4; 3.
* Očakávaný výsledok: Figúrka sa premiestni na dané políčko. Zmení sa aktuálny hráč – na ťahu je biely hráč.
* Názov kroku testu: 2. pohyb bieleho hráča
* Popis kroku testu: Biely hráč sa pohne kráľovnou z pozície 3; 7 na pozíciu 7; 3.
* Očakávaný výsledok: Figúrka sa premiestni na dané políčko. Zmení sa aktuálny hráč – na ťahu je čierny hráč.
* Názov kroku testu: 2. pohyb čierneho hráča
* Popis kroku testu: Čierny hráč sa pohne kráľom z pozície 4; 0 na pozíciu 4; 1.
* Očakávaný výsledok: Figúrka sa premiestni na dané políčko. Zmení sa aktuálny hráč – na ťahu je biely hráč.
* Názov kroku testu: 3. pohyb bieleho hráča
* Popis kroku testu: Biely hráč sa pohne kráľovnou z pozície 7; 3 na pozíciu 4; 3.
* Očakávaný výsledok: Figúrka sa premiestni na dané políčko. Nepriateľský pešiak na danej pozícií bude vyhodený. Nepriateľský kráľ dostane šach, a po ďalšej kontrole aj šach-mat.

# Diagram pre procesný test:



Exportovaný výstup z aplikácie je priložený v súbore: Test situations 1, TDL= 2, ALG= PCT..csv

Testovací scenár 1:

1. Defaultné nastavenie hry, hra sa začína...
2. 1. ťah.
3. Biely hráč sa nepohne pešiakom na správne miesto t.j. iné ako 4; 6 -> 4; 4.
4. 1. pohyb bieleho hráča na správne miesto – „Nie“.
5. Nevalidný pohyb pre pokračovanie v 3 ťahovom šachu.

Testovací scenár 2:

1. Defaultné nastavenie hry, hra sa začína...
2. 1. ťah.
3. Biely hráč sa pohne pešiakom na správne miesto t.j. 4; 6 -> 4; 4.
4. 1. pohyb bieleho hráča na správne miesto – „Áno“.
5. Čierny hráč sa nepohne pešiakom na správne miesto t.j. iné ako 4; 1 -> 4; 3.
6. 1. pohyb čierneho hráča na správne miesto – „Nie“.
7. Nevalidný pohyb pre pokračovanie v 3 ťahovom šachu.

Testovací scenár 3:

1. Defaultné nastavenie hry, hra sa začína...
2. 1. ťah.
3. Biely hráč sa pohne pešiakom na správne miesto t.j. 4; 6 -> 4; 4.
4. 1. pohyb bieleho hráča na správne miesto – „Áno“.
5. Čierny hráč sa pohne pešiakom na správne miesto t.j. 4; 1 -> 4; 3.
6. 1. pohyb čierneho hráča na správne miesto – „Áno“.
7. 2. ťah.
8. Biely hráč sa nepohne kráľovnou na správne miesto t.j .iné ako 3; 7 -> 7; 3.
9. 2. pohyb bieleho hráča na správne miesto – „Nie“.
10. Nevalidný pohyb pre pokračovanie v 3 ťahovom šachu.

Testovací scenár 4:

1. Defaultné nastavenie hry, hra sa začína...
2. 1. ťah.
3. Biely hráč sa pohne pešiakom na správne miesto t.j. 4; 6 -> 4; 4.
4. 1. pohyb bieleho hráča na správne miesto – „Áno“.
5. Čierny hráč sa pohne pešiakom na správne miesto t.j. 4; 1 -> 4; 3.
6. 1. pohyb čierneho hráča na správne miesto – „Áno“.
7. 2. ťah.
8. Biely hráč sa pohne kráľovnou na správne miesto t.j. 3; 7 -> 7; 3.
9. 2. pohyb bieleho hráča na správne miesto – „Áno“.
10. Čierny hráč sa nepohne kráľom na správne miesto t.j. iné ako 4; 0 -> 4; 1.
11. 2. pohyb čierneho hráča na správne miesto – „Nie“.
12. Nevalidný pohyb pre pokračovanie v 3 ťahovom šachu.

Testovací scenár 5:

1. Defaultné nastavenie hry, hra sa začína...
2. 1. ťah.
3. Biely hráč sa pohne pešiakom na správne miesto t.j. 4; 6 -> 4; 4.
4. 1. pohyb bieleho hráča na správne miesto – „Áno“.
5. Čierny hráč sa pohne pešiakom na správne miesto t.j. 4; 1 -> 4; 3.
6. 1. pohyb čierneho hráča na správne miesto – „Áno“.
7. 2. ťah.
8. Biely hráč sa pohne kráľovnou na správne miesto t.j. 3; 7 -> 7; 3.
9. 2. pohyb bieleho hráča na správne miesto – „Áno“.
10. Čierny hráč sa pohne kráľom na správne miesto t.j. 4; 0 -> 4; 1.
11. 2. pohyb čierneho hráča na správne miesto – „Áno“.
12. 3. ťah.
13. Biely hráč sa nepohne kráľovnou na správne miesto t.j. iné ako 7; 3 -> 4;3.
14. 3. pohyb bieleho hráča na správne miesto – „Nie“.
15. Nevalidný pohyb pre pokračovanie v 3 ťahovom šachu.

Testovací scenár 6:

1. Defaultné nastavenie hry, hra sa začína...
2. 1. ťah.
3. Biely hráč sa pohne pešiakom na správne miesto t.j. 4; 6 -> 4; 4.
4. 1. pohyb bieleho hráča na správne miesto – „Áno“.
5. Čierny hráč sa pohne pešiakom na správne miesto t.j. 4; 1 -> 4; 3.
6. 1. pohyb čierneho hráča na správne miesto – „Áno“.
7. 2. ťah.
8. Biely hráč sa pohne kráľovnou na správne miesto t.j. 3; 7 -> 7; 3.
9. 2. pohyb bieleho hráča na správne miesto – „Áno“.
10. Čierny hráč sa pohne kráľom na správne miesto t.j. 4; 0 -> 4; 1.
11. 2. pohyb čierneho hráča na správne miesto – „Áno“.
12. 3. ťah.
13. Biely hráč sa pohne kráľovnou na správne miesto t.j. 7; 3 -> 4;3.
14. 3. pohyb bieleho hráča na správne miesto – „Áno“.
15. Čierny hráč šach-mat.
16. Koniec hry.

# Unit testy:



# Mockito testy:



# Procesné testy:

