Testovací strategie

Popis funkcionality aplikace

Aplikace je počítačová verze klasické deskové hry Šachy s kompletními pravidly (en passant, rošáda atd.). Hráč v ní může hrát proti druhému hráči na jednom počítači nebo proti počítači (náhodný generátor tahů). Lze také ukládat a nahrávat hry ve formátu PGN a prohlížet odehrané tahy. Hra informuje hráče o možných tazích a případném konci hry.

Více info o aplikaci lze najít v README.md repozitáře.

Použité testovací nástroje/frameworky: JUnit 5, Mockito, PowerMockito

Přehled částí aplikace

Projekt se skládá z 3 hlavních packagů pod packagem cz.cvut.fel.bouredan.chess

- **game** – zahrnuje game engine a je nezávislý od package **gui**, také zahrnuje části pro načítání a ukládání her v PGN
- **gui** - (view) ukazuje UI uživateli, zobrazuje šachovnicy s figurkami a možné tahy
- **common –** utils používané jak v game tak gui package, obsahuje třídu znázorňující souřadnice na šachovnici

Třída Board a některé další jsou immutable. Třída Board je vždy poslána jako objekt view, který jí zobrazí uživateli.

Prioritizace částí aplikace

Proces	Možné poškození	Vysvětlení možného poškození	Pravděpodobnost selhání	Vysvětlení pravděpodobnosti selhání
Zobrazení šachovnice	střední	Uživateli se nezobrazí šachovnice, aplikace spadne.	malá	Třída Board je vždy inicializována, tudíž by mělo být vždy co zobrazit.
Zobrazení možných tahů	velké	Uživateli se nezobrazí správně možné tahy, uživatel pak nemůže tento tah provést.	střední	V algoritmu na vyhodnocování možných tahů může být chyba.
Provést tah	střední	Uživatel nebude moc táhnout figurkami.	malé	Je spojené se zobrazením možných tahů, samotný tah už je pak triviální
Ukončení hry	střední	Hra nepozná šachmat nebo remízu.	střední	V algoritmu na zjištění konce hry může být chyba nebo hra nechá hráče hrát o po jejím konci.
Uložení hry	střední	Hru se nepovede uložit, nebo se uloží ve špatném formátu.	vysoká	V algoritmu na zápis hry do PGN formátu může být chyba a hra se tak zapíše špatně, že nepůjde opět načíst.
Načtení hry	střední	Hru se nepovede správně načíst z PGN formátu.	střední	V algoritmu na načtení hry může být chyba, PGN soubor může mít trochu jinou notaci.

Test levels

Proces	Revize	Vývojářské testy	UAT
Zobrazení šachovnice	ANO	ANO	NE
Zobrazení možných tahů	ANO	ANO	ANO
Provést tah	ANO	ANO	NE
Ukončení hry	ANO	ANO	ANO
Uložení hry	ANO	ANO	ANO
Načtení hry	ANO	ANO	ANO

Testovací scénáře

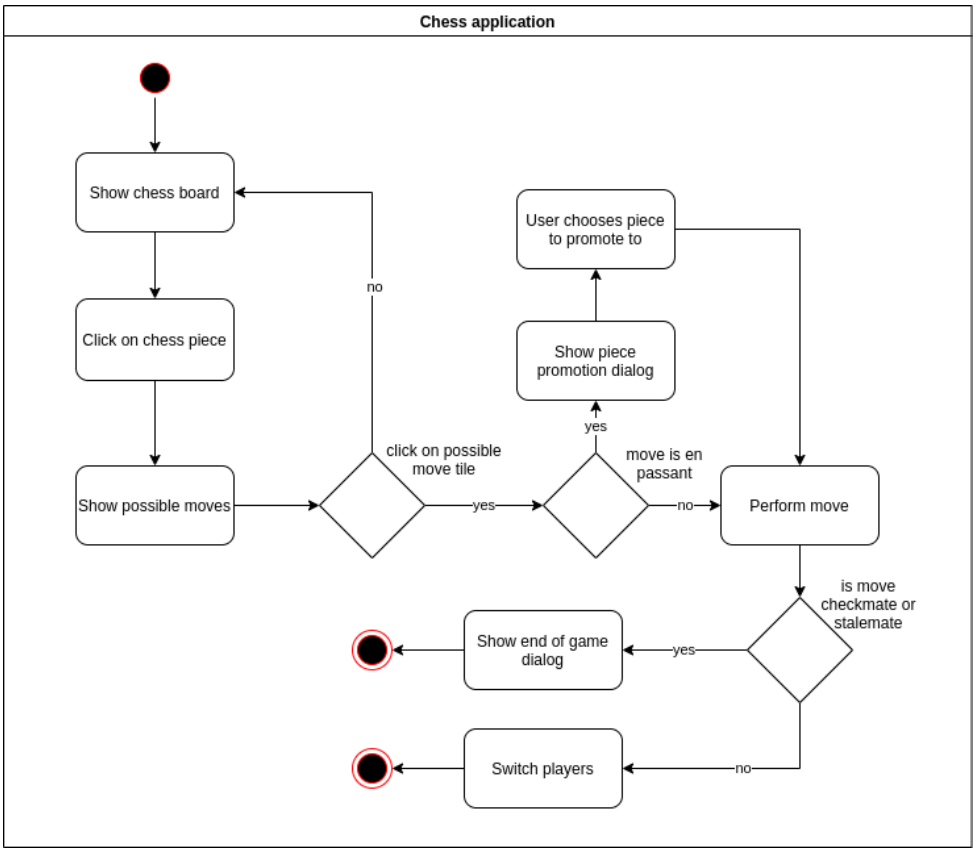
Testy vstupů – třídy ekvivalence

Typ třídy ekvivalence	Co znamená pro aplikaci	Podmínky
PGN formát	Hra musí být zapsaná ve standardním PGN formátu, aby šla načíst.	• Soubor musí být v PGN formátu popsaném zde • Event tagy zatím nejsou podporovány. • Nadbytečné whitespaces jsou ignorovány • Velikost souboru je limitována pouze pamětí

Detailní testovací scénáře

Položka	Obsah
ID testu	MOVE_PIECE_01
Název testu	Standardní tah
Popis testu	Uživatel klikne na figurku a vybere pole na které se s ní dá táhnout. Hra provede tah figurkou na dané pole.
Hloubka detailu	střední
Vstupní podmínky	Hra běží a existuje nějaká figurka se kterou se dá standardně táhnout.
Testovací data	náhodně vygenerované platné tahy
Očekávaný výsledek	hra provede tah
Autor	Daniel Bourek

Diagram procesu tahu



Průchody scénářem

#	Průchod scénáře,
1	Kliknutí na figurku -> Kliknutí na pole, které není možný tah
2	Kliknutí na figurku -> Kliknutí na pole možného tahu -> Tah je en passant -> Tah není šachmat nebo pat
3	Kliknutí na figurku -> Kliknutí na pole možného tahu -> Tah není en passant -> Tah není šachmat nebo pat
4	Kliknutí na figurku -> Kliknutí na pole možného tahu -> Tah je en passant -> Tah je šachmat nebo pat
5	Kliknutí na figurku -> Kliknutí na pole možného tahu -> Tah není en passant -> Tah je šachmat nebo pat

Seznam JUnit testů

Unit testy

#	Test class	Test method
1	PositionUnitTest	positionNotationConstructor_constructingPositionFromNotation_correctPositionConstructed
2	BoardUnitTest	boardConstructor_createNewBoardFromTiles_boardsAreEqual
3	BoardUnitTest	getKingPosition_getWhiteKingPositionOnStartingBoard_correctKingPositionReturned
4	BoardUnitTest	isEnPassantMovePossible_enPassantMoveIsPossible_returnsTrue

Integrační testy

#	Test class	Test method
1	GameIntegrationTest	playMove_movingPieces_pieceMoved
2	GameIntegrationTest	makeMove_makeSame_oveTwice_moveIsNull
3	GameIntegrationTest	loadGame_makePromotionWithCheckMateMove_gameWon
4	GameIntegrationTest	getMove_getMoveFromHistory_correctMoveReturned
5	GameIntegrationTest	createMove_moveIsNotInPossible_moveIsNull
6	GameIntegrationTest	newGame_piecesPositions_piecesPlacedCorrectly
7	PgnIoIntegrationTest	isEnPassantMovePossible_enPassantMoveIsPossible_returnsTrue
8	PgnIoIntegrationTest	loadGame_loadEmptyGame_emptyGameLoaded
9	PgnIoIntegrationTest	loadGame_loadWonGame_gameStateWon
10	PgnIoIntegrationTest	loadGame_loadGameWithLastMoveCastling_assertRookPosition
11	PgnIoIntegrationTest	saveGame_loadGameAndThenSaveTheGame_filesMatch