

Semestrální práce TS1

Popis systému

Testovaná aplikace Šachy byla implementovaná jako semestrální práce pro předmět TS1 v semestru 2020/2021. Autoři : Akylbek Mendibayev a Dmytro Lylo. Následně byla aplikace využita i jako semestrální práce pro předmět PJV. Aplikace implementovaná pomocí programovacího jazyka Java.

Funkční požadavky

FR1: Kontrola pravidel hry

FR2: Správné zobrazení desky (UI)

FR3: Možnost hry dvou hráčů

Přehled částí aplikace

- application:
 - **Program:** Tato třída obsahuje kód, který spouští šachovou aplikaci a je přehráván v celém průběhu hry
 - **UI:** Základní třída která obsahuje kód interaktivního prostředí s uživatelem - User Interface
- boardGame:
 - **Board:** Tato třída působí jako deska v šachové hře a může zahrnovat pozice a postavy
 - **BoardException:** Hlavní výjimky jsou zachyceny v této třídě a poskytují bezchybný proces hry
 - **Piece:** Jedná se o abstraktní třídu, která je vzorem pro všechny postavy na desce v šachové hře
 - **Position:** Třída, která spojuje třídy Board a Piece a slouží jako pozice postav
- chess:
 - **ChessMatch:** Tato třída obsahuje základní logiku hry Šachy a má základní metody interakce postav a hráčů
 - **ChessException:** Třída, která zachycuje všechny chyby během hry
 - **ChessPiece:** Tato třída má kód, který je zodpovědný za postavy na desce ze strany třídy ChessMatch
 - **ChessPosition:** Třída je zodpovědná za pozice postav na desce ze strany třídy ChessMatch
 - **Color:** Enum třída, která je zodpovědná za barvy hráčů

Testovací strategie

Třída rizika		Pravděpodobnost selhání		
		High	Medium	Low
Možné poškození	High	A	B	B
	Medium	B	B	C
	Low	C	C	C

Určení priorit

Proces	Požadavek	Dopad	Vysvětlení možné poškození	Část systému	Pravděpodobnost selhání	Vysvětlení pravděpodobnosti selhání	Třída rizika
Pohyb figur na šachovnici	Kontrola pravidel hry	H	Figury mohou hýbat nekorektně	Pohyby a jejich kontrola	H	Komplikovaná implementace	A
Proměna pěšce	Kontrola pravidel hry	L	proměna pěšce může fungovat nekorektně	Pohyby a jejich kontrola	M	Implementace proměny pěšce je dost	B
Střídání hráčů	Kontrola pravidel hry, Možnost hry dvou hráčů	L	Střídání hráčů může fungovat nekorektně	Pohyby a jejich kontrola	L	Implementace střídání hráčů je jednoduché	C
Zobrazení desky	Správné zobrazení desky	M	Není správně zobrazení desky	Uživatelské rozhraní	L	Implementace je jednoduchá	C
Zobrazení zachycených figur	Kontrola pravidel hry, Správné zobrazení desky	M	Aplikace nebude správně zobrazovat zachycené figury. Chyba neovlivňuje funkčnost aplikace	Pohyby a jejich kontrola	M	Záleží na implementaci pohybu figur	B

Test Levels

Quality characteristic Část systému / funkce	Třída rizika	Revize	Vývojářské testy	Systémové testy	UAT	Test v produkci
Pohyb figur na šachovnici	A	Ano	Střední	Vysoká	Vysoká	Ano
Proměna pěšce	B	Ano	Nízká	Nízká	nízká	Ano
Střídání hráčů	C		Nízká	Nízká	nízká	Ano
Zobrazení desky	C		Nízká	Nízká	nízká	Ano
Zobrazení zachycených figur	B		Nízka	Nízká	nízká	Ano

Třída ekvivalence

Pro analýzu tříd ekvivalence bylo vybráno tyto metody:

public ChessPiece newPiece(String type, Color color)

Tento metoda vrátí novou figuru.

Parameter	TE	Validnost TE	Mezní podmínky
type	["B", "N", "R", "Q"]	Validní	Nejsou
	Ostatní vstupy typu String	Nevalidní z business pohledu	
	null	Nevalidní z technického pohledu	
	Jiny datovy typ	Nevalidní z technického pohledu	
color	WHITE, BLACK	Validní	Nejsou
	null	Nevalidní z technického pohledu	
	Jiny datovy typ	Nevalidní z technického pohledu	

Pairwise testing:

	Type	Color
1	B	WHITE
2	B	BLACK
3	N	WHITE
4	N	BLACK
5	R	WHITE
6	R	BLACK

Celá tabulka ve formátu .xml se nachází ve repozitáři projektu ve složce pairwise (newPiecePairwise.xml)

public boolean positionExists(int row, int column)

Tento metoda kontroluje existuje li pozice na desce. Na vstup očekává číslo řádku a sloupce

Parameter	TE	Validnost TE	Mezni podmínky
row	< 0	Nevalidní z business pohledu	-1, 0, 7, 8
	0 až 7	Validní	
	> 7	Nevalidní z business pohledu	
	null	Nevalidní z technického pohledu	
	Jiný datovy typ	Nevalidní z technického pohledu	
column	< 0	Nevalidní z business pohledu	-1, 0, 7, 8
	0 až 7	Validní	
	> 7	Nevalidní z business pohledu	
	null	Nevalidní z technického pohledu	
	Jiný datovy typ	Nevalidní z technického pohledu	

Pairwise testing:

	row	column
1	0	0
2	0	1
3	0	2
4	0	3
5	0	4
6	0	5
7	0	6
8	0	7
9	1	0
10	1	1
11	1	2
12	1	3
13	1	4
14	1	5
15	1	6
16	1	7
17	2	0
18	2	1
19	2	2
20	2	3

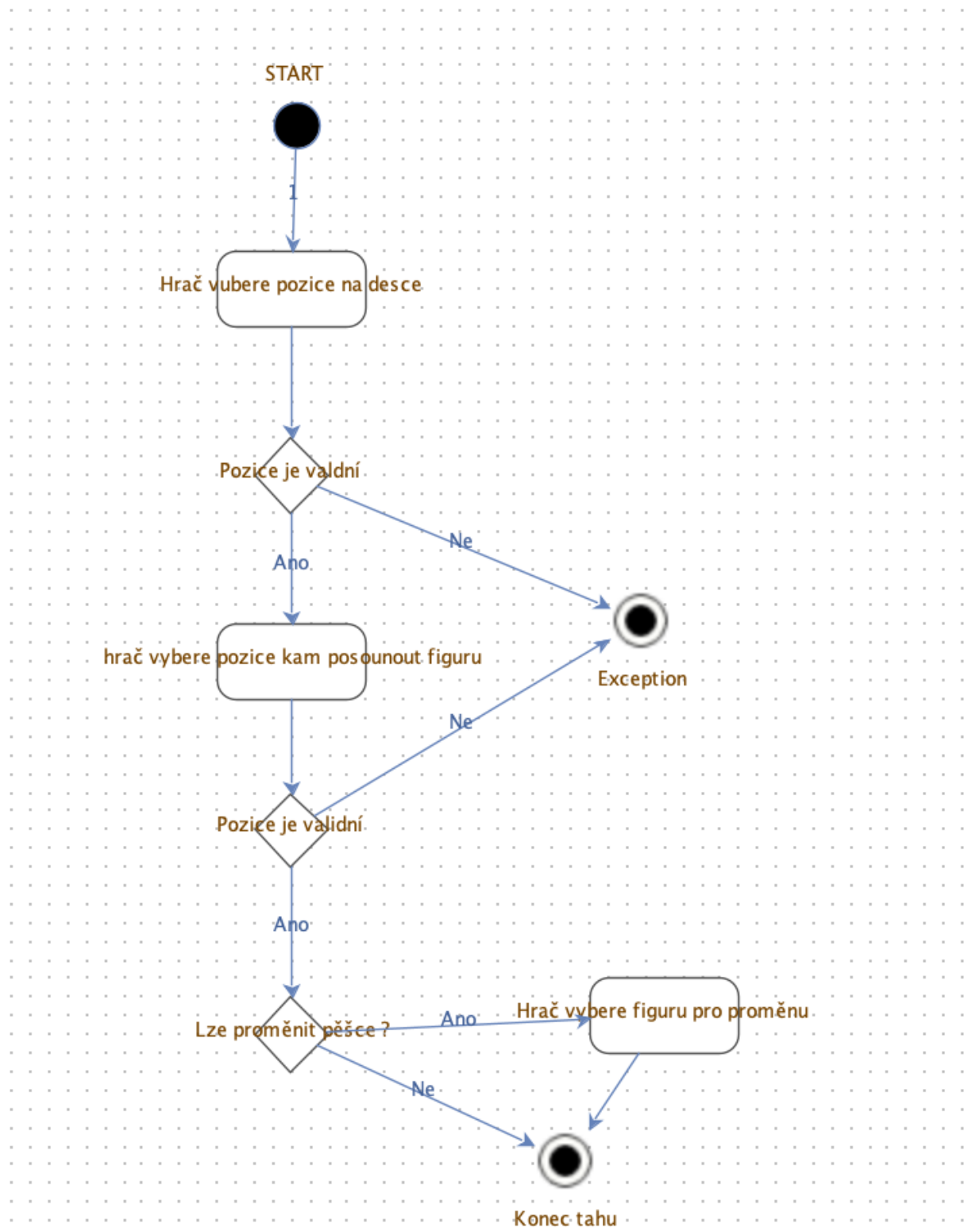
Celá tabulka ve formátu .xml se nachází ve repozitáři projektu ve složce pairwise (positionExistPairwise.xml)

Testy průchodů

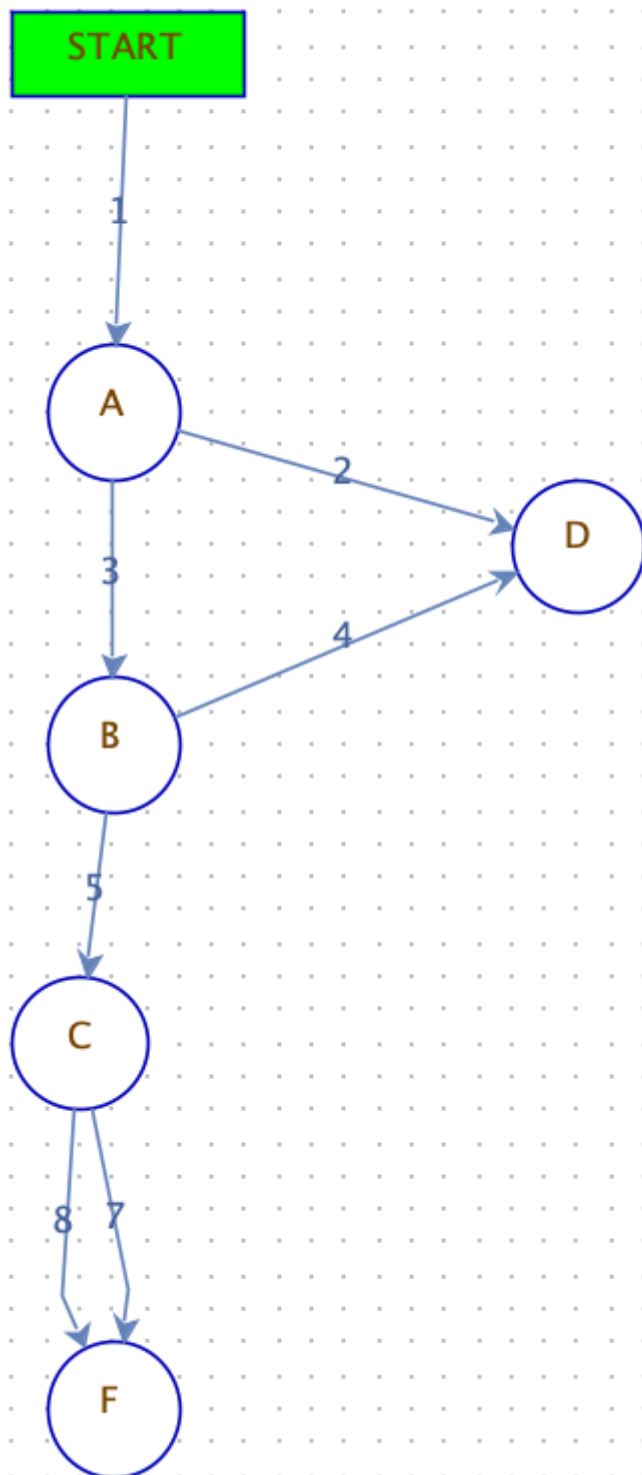
Výstupy z aplikace se nachází ve repozitáři projektu ve složce process-tests

Pohyb figury

Diagram aktivity



Graf pro vytvoření procesních testu

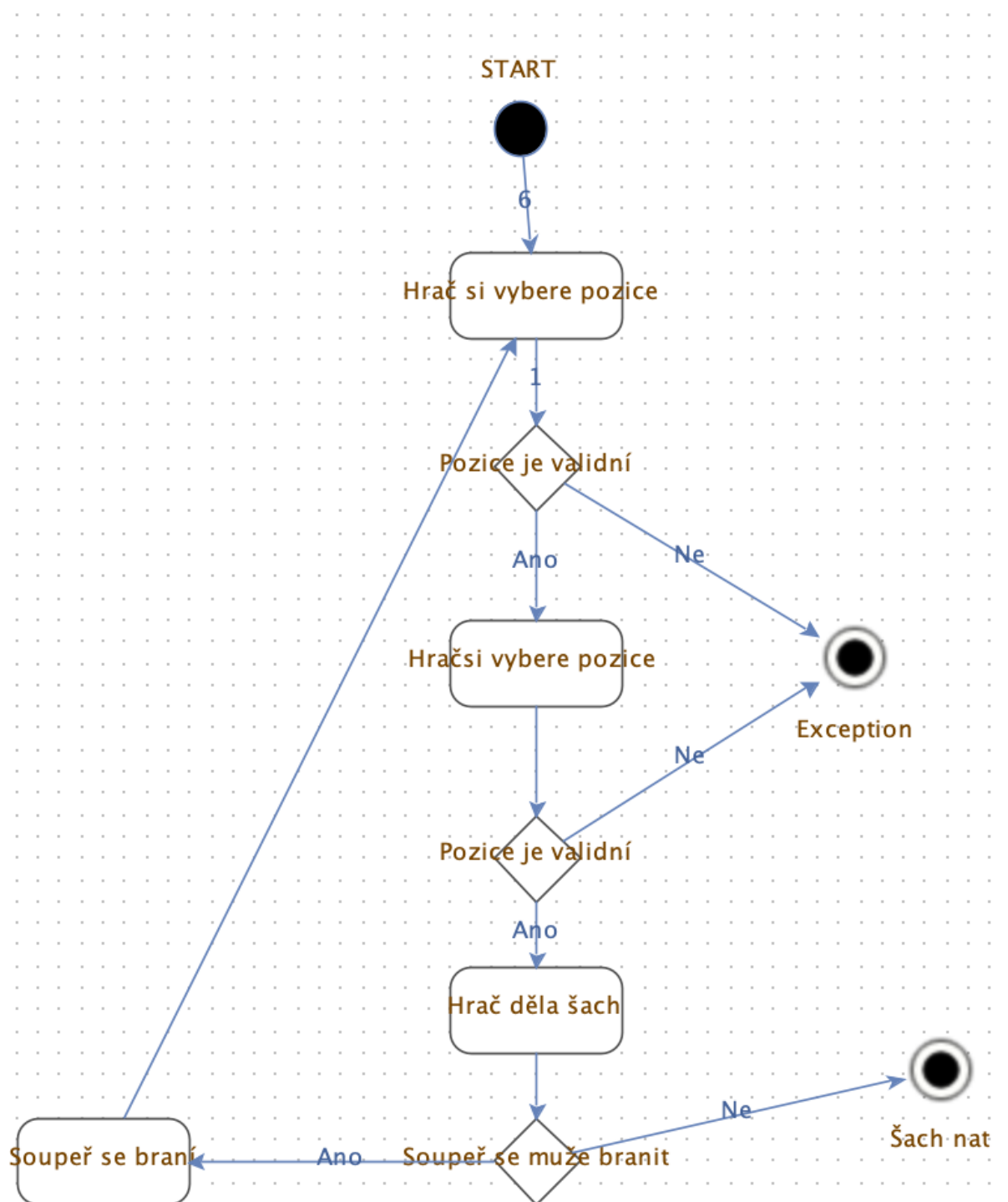


Vygenerované test cases (TDL-2)

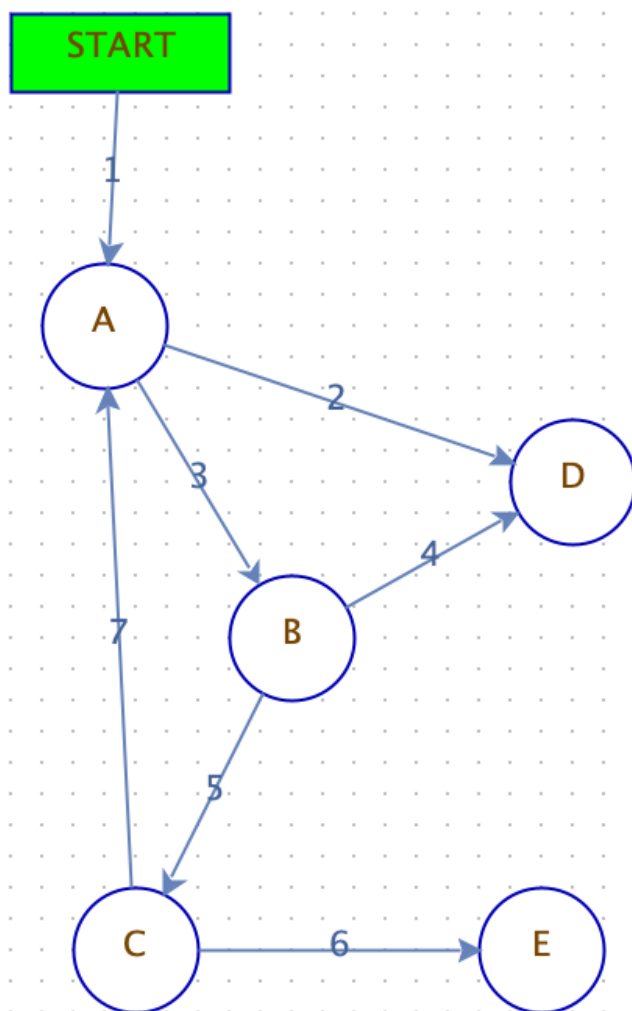
No.	Test sequence
1	1 - 2
2	1 - 3 - 4
3	1 - 3 - 5 - 7
4	1 - 3 - 5 - 8

Šach Mat

Diagram aktivity



Graf pro vytvoření procesních testů



Vygenerované test cases (TDL-2)

No.	Test sequence
1	1 - 2
2	1 - 3 - 4
3	1 - 3 - 5 - 7 - 3 - 5 - 7 - 2
4	1 - 3 - 5 - 6

Detailní testovací scénáře

ID testu	01_checkmate
Nazev testu	Mat bláznů
Autor	Dmytro Lylo
Hloubka testu	střední
Popis testu	Černý hráč vyhra bílého hráče za 2 tahy
Vstupní podmínky	standardní rozložení figur
Očekávaný výsledek	Všichni pohyby figur provedou normálně, černý hráč vyhra

Kroky testu:

- Nazev kroku testu:** Bílý hráč posune figuru_1
Bílý hráč posune z pozice 'f2' na pozice 'f3'
Očekávaný výsledek: Figura se posune na pozice 'f3'. Změní se aktuální hráč na černého.
- Nazev kroku testu:** Černý hráč posune figuru_1
Černý hráč posune z pozice 'e7' na pozice 'e5'
Očekávaný výsledek: Figura se posune na pozice 'e5'. Změní se aktuální hráč na bílého.
- Nazev kroku testu:** Bílý hráč posune figuru_2
Bílý hráč posune z pozice 'g2' na pozice 'g4'
Očekávaný výsledek: Figura se posune na pozice 'g4'. Změní se aktuální hráč na černého.
- Nazev kroku testu:** Černý hráč posune figuru a vyhra
Černý hráč posune z pozice 'd8' na pozice 'h4'
Očekávaný výsledek: Figura se posune na pozice 'h4'. Bílý kral dostane CHECKMATE, černý hráč vyhra.

ID testu	02_CheckmateWithPawnPromotion
Nazev testu	Zlepšení pěšáka. Pěšák se zlepší
Autor	Akylbek Mendibayev
Hloubka testu	střední
Popis testu	Bílý hráč vyhra, po proměně pěšce
Vstupní podmínky	standardní rozložení figur
Očekávaný výsledek	Všichni pohyby figur provedou normálně, pěšák se stává královnou, bílý hráč vyhra

Kroky testu:

- Nazev kroku testu:** Bílý hráč posune figuru_1
 Bílý hráč posune z pozice 'e2' na pozice 'e4'
Očekavany výsledek: Figura se posune na pozice 'e4'. Změní se aktuální hráč na černého.
- Nazev kroku testu:** Černý hráč posune figuru_1
 Černý hráč posune z pozice 'd7' na pozice 'd5'
Očekavany výsledek: Figura se posune na pozice 'd5'. Změní se aktuální hráč na bílého.
- Nazev kroku testu:** Bílý hráč posune figuru_2
 Bílý hráč posune z pozice 'e4' na pozice 'd5'
Očekavany výsledek: Figura se posune na pozice 'd5'. Změní se aktuální hráč na černého.
- Nazev kroku testu:** Černý hráč posune figuru_2
 Černý hráč posune z pozice 'g8' na pozice 'f6'
Očekavany výsledek: Figura se posune na pozice 'f6'. Změní se aktuální hráč na bílého.
- Nazev kroku testu:** Bílý hráč posune figuru_3
 Bílý hráč posune z pozice 'f1' na pozice 'b5'
Očekavany výsledek: Figura se posune na pozice 'b5'. Změní se aktuální hráč na černého.
- Nazev kroku testu:** Černý hráč posune figuru_3
 Černý hráč posune z pozice 'c7' na pozice 'c6'
Očekavany výsledek: Figura se posune na pozice 'c6'. Změní se aktuální hráč na bílého.
- Nazev kroku testu:** Bílý hráč posune figuru_4
 Bílý hráč posune z pozice 'd5' na pozice 'c6'
Očekavany výsledek: Figura se posune na pozice 'c6'. Změní se aktuální hráč na černého.
- Nazev kroku testu:** Černý hráč posune figuru_4
 Černý hráč posune z pozice 'd8' na pozice 'b6'
Očekavany výsledek: Figura se posune na pozice 'b6'. Změní se aktuální hráč na bílého.

9. **Nazev kroku testu:** Bílý hráč posune figuru_5
Bílý hráč posune z pozice 'c6' na pozice 'b7'
Ocekavany vysledek: Figura se posune na pozice 'b7'. Změní se aktuální hráč na bílého.
10. **Nazev kroku testu:** Černý hráč posune figuru_5
Černý hráč posune z pozice 'b6' na pozice 'b5'
Ocekavany vysledek: Figura se posune na pozice 'b5'. Změní se aktuální hráč na černého.
11. **Nazev kroku testu:** Bílý hráč posune figuru_6
Bílý hráč posune z pozice 'b7' na pozice 'c8'
Ocekavany vysledek: Figura se posune na pozice 'c8'. Aplikace požaduje vybrat figuru pro proměnu pěšce
12. **Nazev kroku testu:** Bílý hráč změni pěšáka na damu a vyhru hru
Bílý hráč vybere damu pro proměnu pěšce
Očekávaný vysledek: Pěšák se promění na damu pak černý hráč dostane CHECKMATE, bílý hráč vyhru.

Implementované testy

Unit testy

Nazev testu	Používa mockito	Je parametrizovaný
positionExist_validCoordinates_ReturnTrue_Mocked	Ano	Ano
positionExist_NotValidCoordinates_ReturnFalse_Mocked	Ano	Ano
ThereIsAPiece_PositionIsEmpty_ReturnFalse_Mocked	Ano	Ano
ThereIsAPiece_PositionIsNotEmpty_ReturnFalse_Mocked	Ano	Ano
opponentCheckTest	Ne	Ne
readChessPositionString	Ne	Ano
RemovePiece_RemoveWhiteKing_ReturnKing_Mocked	Ano	Ne
NewPiece_ReturnQueen	Ne	Ne
getCurrentPlayer_newMatch_ReturnColorWhite	Ne	Ne
possibleMove_PawnOnPosition60_CanMoveToPosition50 and40_Mocked	Ano	Ano

Procesní testy

Nazev testu
public void SourcePositionIsNotValid(){}
public void TargetPositionIsNotValid() {}
public void MoveFigure(){}
public void checkMateTest(){}
public void TestingTurnFunctionality(){}
public void getMovedPieceTest(){}
public void changeTurnFromWhiteToBlack(){}
public void gettingCapturedPieces(){}