Semestralní prace TS1

Popis systemu

Testovaná aplikáce Šachy byla implementovaná jako semestrální práce pro předmět TS1 v semestru 2020/2021. Autoři : Akylbek Mendibayev a Dmytro Lylo. Následně byla aplikace využití 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

Proces	Požadavek	Část systému
Pohyb figúr na šachovnici	Kontrola pravidel hry	Pohyby a jejich kontrola
Zobrazení desky	Správné zobrazení desky	Uživatelské rozhraní
Střídání hráčů	Kontrola pravidel hry, Možnost hry dvou hráčů	Pohyby a jejich kontrola
Proměna pěšce	Kontrola pravidel hry	Pohyby a jejich kontrola
Zobrazení zachycenych figur	Kontrola pravidel hry, Správné zobrazení desky	Pohyby a jejich kontrola

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 figúr 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**	**Střídani hraču může fungovat nekorektně**	**Pohyby a jejich kontrola**	**L**	**Implementace střídani hraču je jednoduche**	**C**
Zobrazení desky	**Správné zobrazení desky**	**M**	**Není správně zobrazení desky**	**Uživatelské rozhraní**	**L**	**Implementace je jednoducha**	**C**
Zobrazení zachycenych 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 implementace pohybu figur**	**B**

Test Levels

Quality characteristic Část systému / funkce	Třída rizika	Test levels		
		Unit testy	Procesní testy	Manuální testy
Pohyb figúr na šachovnici	A	***	***	**
Proměna pěšce	B	*	**	**
Střídání hráčů	C	*	*	*
Zobrazení desky	C	*	*	***
Zobrazení zachycenych figur	B	**	**	*

Třída ekvivalence

V aplikace mamé dva typy vstupu a to je **source** a **target**

- Pro source mame takovou TE:

Pozice na desce: [a1 … h8]

Figura na pozice: true, false

Barva figury: Color.WHITE, Color.Black

ThereArePossibleMoves: true, false

- Pro target mame takovou TE:

Pozice na desce: [a1 … h8]

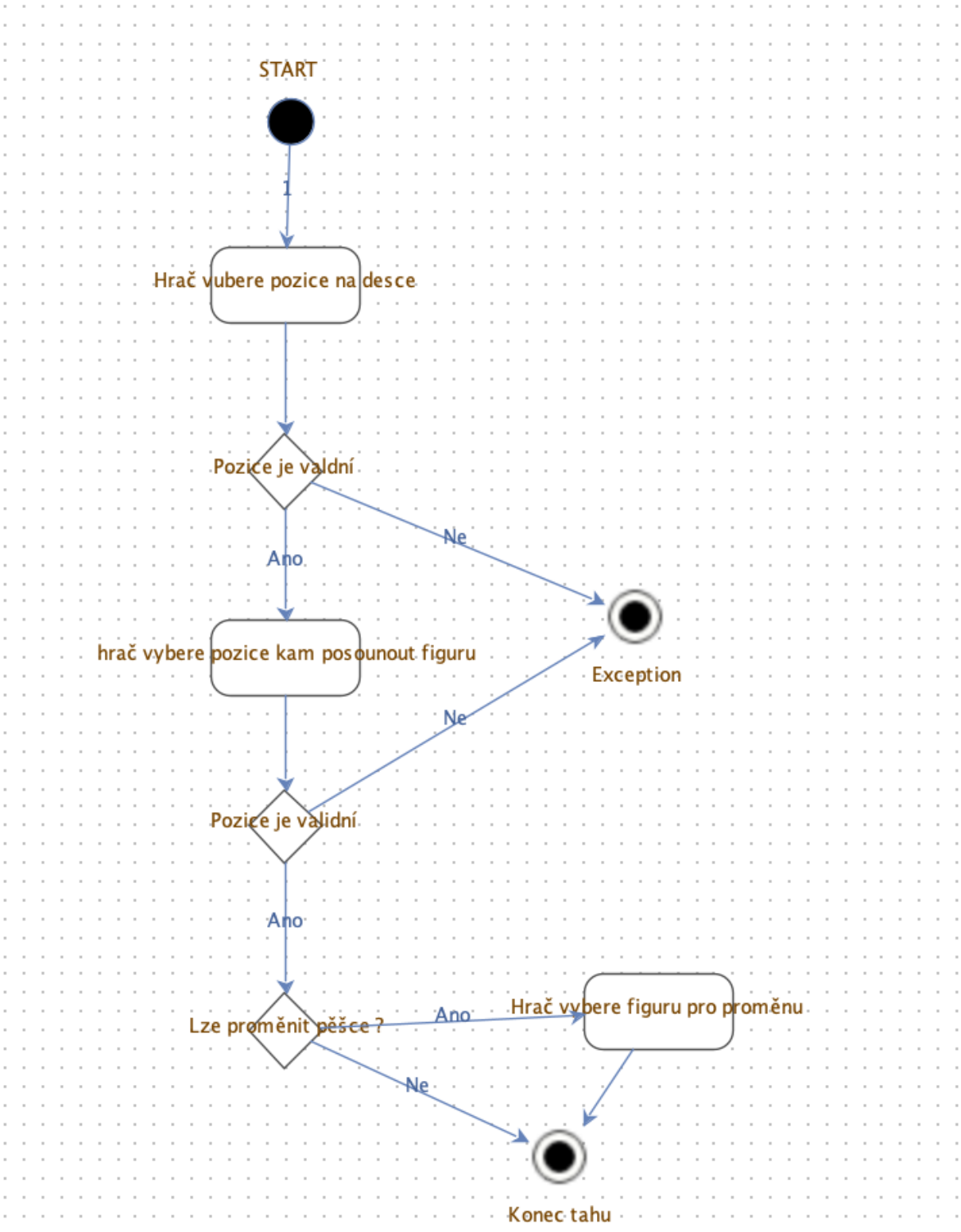
ThereIsPossibleMove: true,false

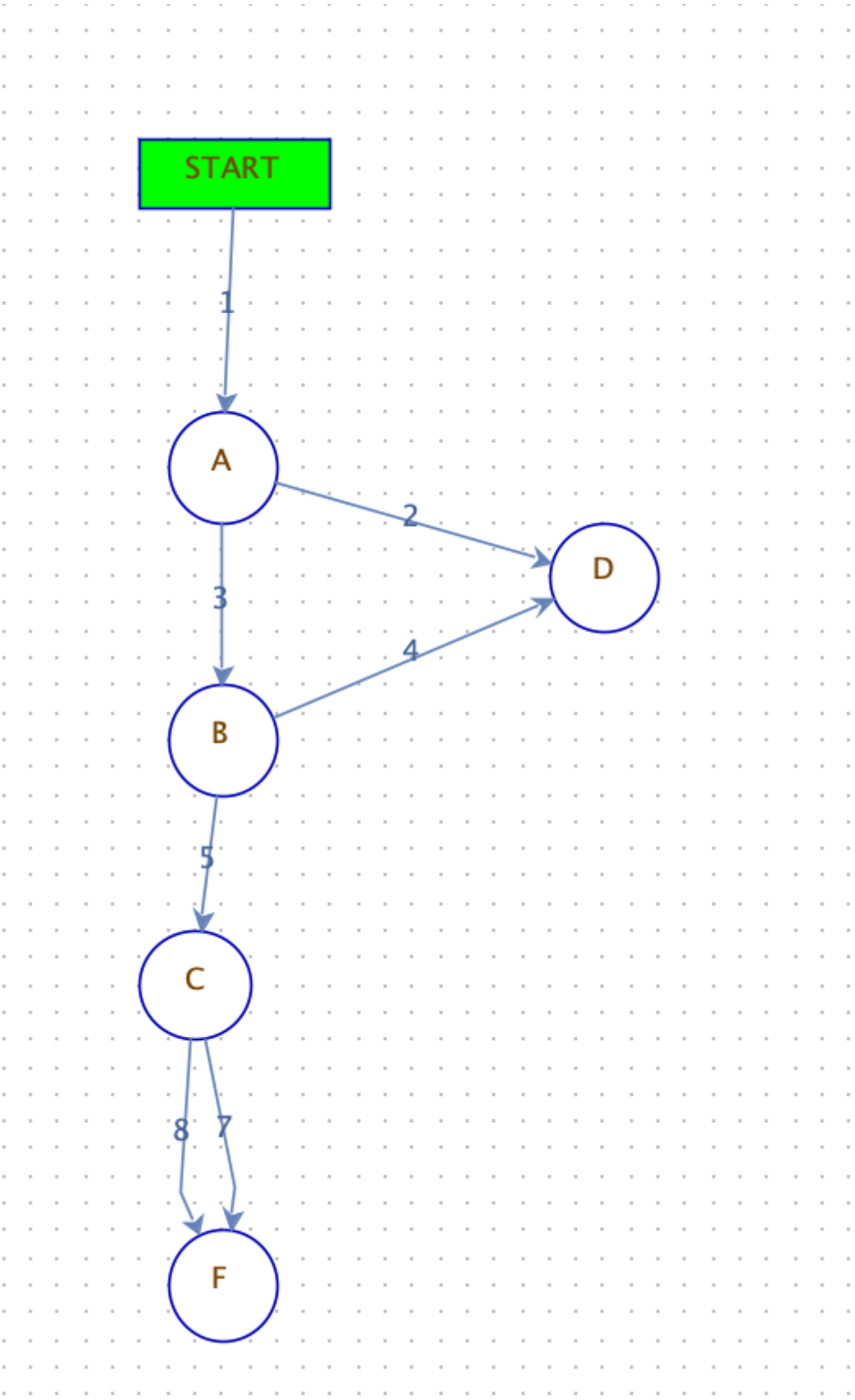
a1,true,BLACK,false
a1,false,WHITE,true
a2,true,WHITE,false
a2,false,BLACK,true
a3,true,WHITE,true
a3,false,BLACK,false
a4,true,WHITE,true
a4,false,BLACK,false
a5,true,WHITE,true

Pairwise kombinace pro SourcePosition. Celou tabulku lze najít ve repozitáře ve složce pairwise.

Testy průchodů

Pohyb figury




START
1
A
2
D
3
4
B
5
C
8
7
F

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

Detailní testovací scénáře

ID testu	01_checkmate
Nazev testu	Mat bláznů. Černý hráč vyhra
Autor	Dmytro Lylo
Hloubka testu	střední
Popis testu	Černý hráč vyhra černého hráče za 2 tahy 1. f3 e5 2. g4 Qh4#
Vstupní podminky	standardní rozložení figur
Očekávaný vysledek	Všichni pohyby figur provedou normálně, bily hráč vyhra

ID testu	02_improvement
Nazev testu	Zlepšení pěšáka. Pěšák se zlepší
Autor	Akylbek Mendibayev
Hloubka testu	střední
Popis testu	Přesuneme pěšáka na poslední klec, dochází ke zlepšění
Vstupní podminky	Před pěšákem nejsou jiné figurky
Očekávaný vysledek	Pěšák se stává královnou

Implementované testy

Unit testy

Nazev testu
public void BoardPositionExist_Mocked(String row, String column){}
public void performChessMove_SourcePositionIsEmpty_ThrowException(){}
public void performChessMove_SourcePositionIsNotYours_ThrowException(){}
public void performChessMove_SourcePositionDoesntHavePossibleMoves_ThrowException(){}
public void performChessMove_CantMoveToTargetPosition_ThrowException(){}
public void removePiece_Mocked() {}
public void pawnPossibleMoves_Mocked() {}
public void KnightPossibleMoves_Mocked() {}
ThereIsAPiece_Mocked(){}
PlaceNewFigureOnTheSamePlace_ThrowException(){}

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(){}