České vysoké učení technické Semestrální práce

Testování Softwaru

Volha Kramko

30.5.2024

# Úvod

## Cíl projektu

Úkolem je navrhnout testy a základně otestovat již vzniklou aplikaci v jazyce Java.

Výsledkem práce je úspěšně absolvovat předmět Testování Softwaru 1.

## Cíl dokumentu

Dokument má sloužit k podrobné dokumentaci první části semestrální práce. Popsat a rozebrat testovanou aplikaci a navrhnout testovací strategii se scénáři. Tyto činnosti budou vypracované pomocí technik probraných v semestru.

## Autor

Na tvorbě této semestrální práce se podílí právě jeden člověk a to autor této dokumentace. Z důvodu nedostatečných zkušeností s programováním práce na samotné aplikaci zabrala autorovi spousta času. Proto se předem omlouvá za vzniklé překlepy a možné chyby z nepozornosti.

## Výběr aplikace

Pro testování jsem si vybral aplikaci, kterou jsem vymyslela, navrhl, i vlastnoručně napsala pro semestrální práci do předmětu Programování v Jave. Jedná se o hru sudoku, kde existují 2 režimy: Sudoku Classic a Sudoku Killer.

Zvláštností hry je generování úrovní a jejich zobrazení.

Uživatel si může vybrat herní režim a úroveň. Aby úroveň prošel, musí vyplnit všechny buňky. Kromě toho může uživatel využívat nastavení hry.

# Popis aplikace

## Hlavní menu

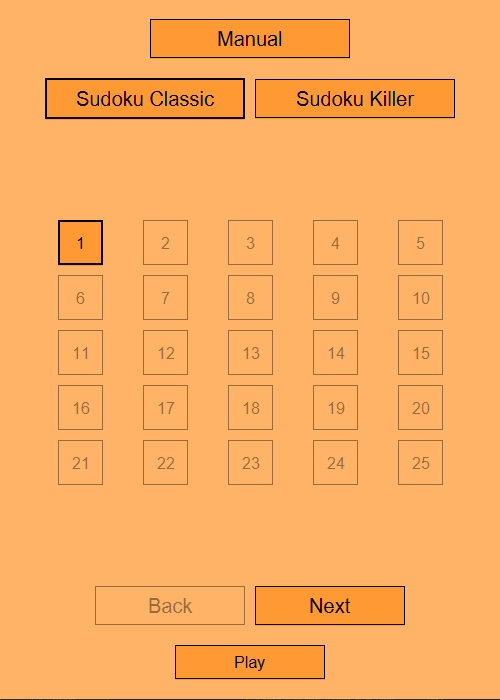
Nahoře je výběr herního režimu (Sudoku Classic nebo Sudoku Killer). ¨

V prostřední části je seznam dostupných úrovní.

Dole jsou tlačítka Back, Next pro prohlidku úrovní a tlačítko “Play” pro pro spuštění hry.

Funkce:

1. Výběr herního režimu (Sudoku Classic nebo Sudoku Killer)
2. Zobrazení dostupných úrovní pro každou herní režim
3. Zobrazení herního pole v souladu s vybraným herním režimem a úrovní



## Herní pole

Nahoře je název režimu, číslo úrovně, časovač a tlačítko nastavení.

V prostřední části je herní pole 9x9.

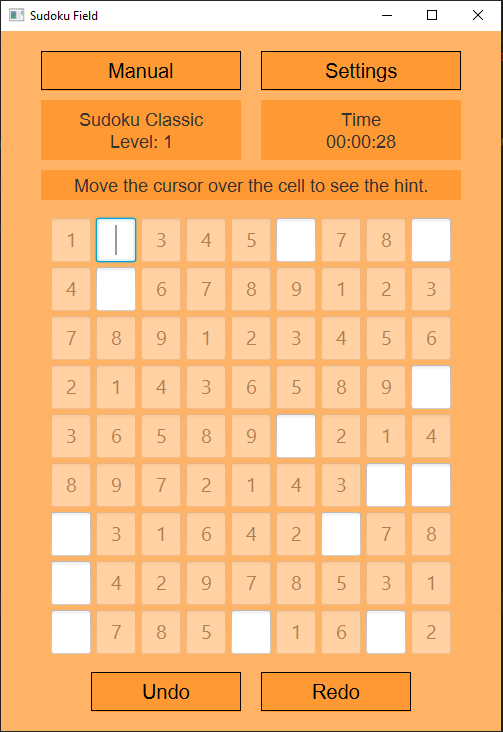
Dole jsou tlačítka Undo a Redo pro zrušení nebo opakování tahu.

Funkce:

1. Vytváření herního pole o velikosti 9x9, rozděleného na 9 čtverců po 3x3.
2. Vytváření časovače.
3. Zadávání hodnot do prázdných buněk.
4. Ověření správnosti vyplnění pole.
5. Zrušení nebo opakování herního tahu.
6. Automatické ukládání, když pole bylo vyplněno spravně.
7. Zobrazení nápovědy: dostupné hodnoty pro každou buňku; pokud uživatel zadal správnou nebo nesprávnou hodnotu.
8. Možnost Otevřít nastavení a uložit časovač.
9. Automatické ukončení hry a uložení časovače, když pole nemá přázdné bunky.

### Herní režim Sudoku Classic

Generování náhodných, avšak vyvážených Sudoku, kde každý řádek, sloupec a čtverec obsahuje všechna čísla od 1 do 9 bez opakování.

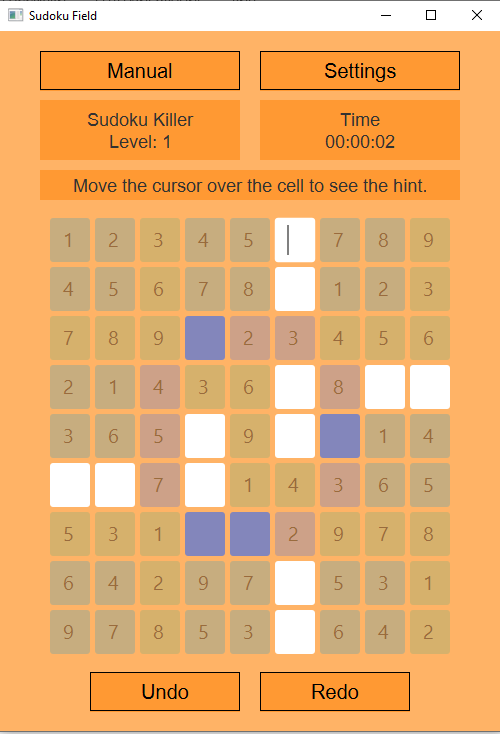


### Herní režim Sudoku Killer

1. Generování náhodných, avšak vyvážených vražedných Sudoku, kde každý řádek, sloupec a čtverec obsahuje všechna čísla od 1 do 9 bez opakování.
2. Dodržení součtu čísel v oblastech:

oblasti jsou 4 typů: 2, 3, 4, 5 buněk;

pro pohodlnost jsou zvýrazněny různými barvami. 2 - bílá, 3 - fialová, 4 - modrá, 5 - zelená.





## Nastavení

Nahoře je název režimu, číslo úrovně, časovač a tlačítko nastavení.

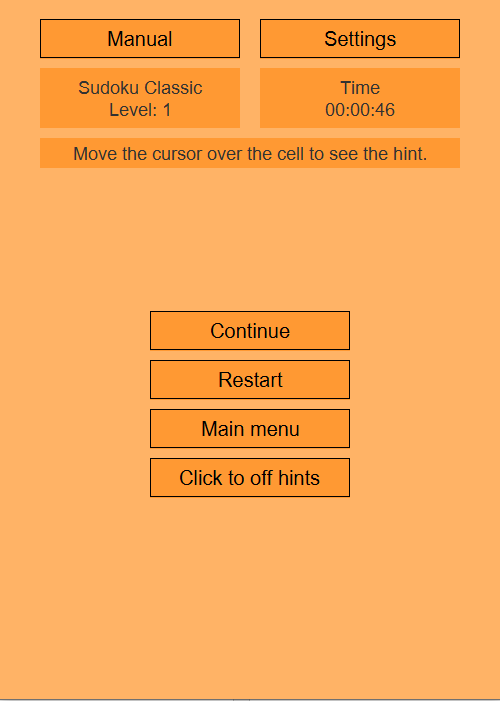
V prostřední části jsou

1. Tlačítko “Continue” pro pokračování úrovně s posledním uloženým výsledkem.
2. Tlačítko “Restart” pro restartování hry, tj. vymazání výsledků úrovně.
3. Tlačítko “Main Menu” pro návracení do hlavního menu. Při dalším otevření se úroveň načte s posledním uloženým výsledkem.
4. Tlačítko z textem “Click to off hints” nebo ”Click to on hints” pro zapnutí a vypnutí nápovědy.

Dole je nic.

Funkce:

1. Zavření nastavení a pokračování úrovně s posledním uloženým výsledkem.
2. Restartování hry, tj. vymazání výsledků úrovně.
3. Návrat do hlavního menu. Při dalším otevření se úroveň načte s posledním uloženým výsledkem.
4. Zapnutí a vypnutí nápovědy.



## Hra skončila

Nahoře je název režimu, číslo úrovně, časovač a tlačítko nastavení.

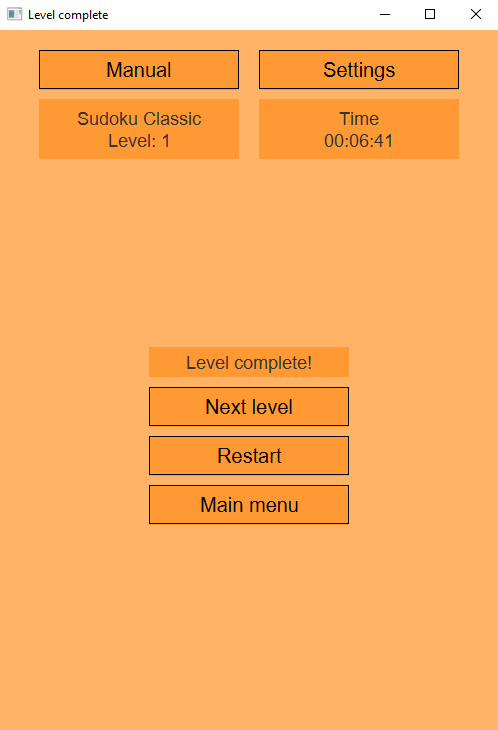
V prostřední části jsou

1. Text “Level complete” aby uživatel pochopil, co se děje.
2. Tlačítko “Next level” pro přechod na další úroveň.
3. Tlačítko “Restart” pro restartování hry, tj. vymazání výsledků úrovně.
4. Tlačítko “Main Menu” pro návracení do hlavního menu. Při dalším otevření se úroveň načte s posledním uloženým výsledkem.

Dole je nic.

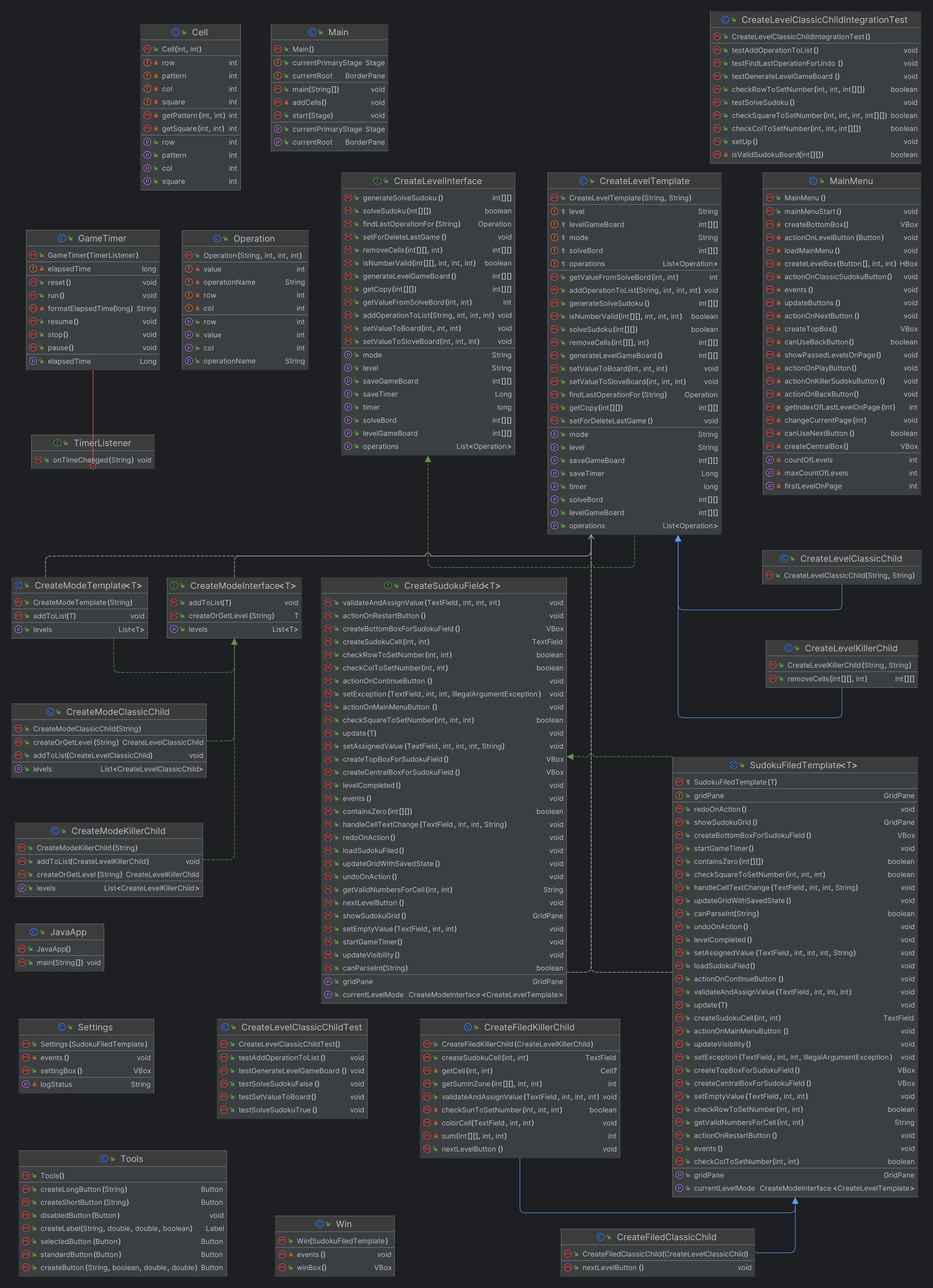
Funkce:

1. Přechod na další úroveň.
2. Návrat do hlavního menu. Při opětovném otevření této úrovně nebude úroveň uložena.



# Testovací strategie

## Objektový model aplikace



## Priorita částí aplikace

## Z modelu je možné si všimnout, že nejvytíženějších tříd je jen několik. To je způsobené jejich nadřazeností vůči ostatním třídám. Mezi tyto třídy patří:

* Main (hlavní třída aplikace, která koordinuje všechny části)
* MainMenu (spravuje hlavní menu aplikace a interakce uživatele)
* CreateLevelTemplate<T> (spravuje vytváření a inicializaci úrovní)
* CreateModeTemplate <T> (spravuje vytváření a inicializaci seznamů úrovní)
* CreateSudokuFieldTemplate<T> (zajišťuje všechny operace spojené s hracím polem Sudoku)

## Přehled rizik

Rizik, která mohou nastat v programu obecně nebo konkrétně v mé aplikaci, je velké množství a mnoha druhů. Pro ilustraci jsem zde nějaké z nejrůznějších odvětví vybrala.

|  |  |
| --- | --- |
| **Riziko** | **Návrh řešení** |
| Nesprávná generace vyřešeného sudoku | Junit testy |
| Nesprávný počet odstraněných buněk při generování hracího pole | Junit testy |
| Nesprávná posloupnost odstraňování buněk v režimu Sudoku Killer | Junit testy |
| Nesprávné přidání úrovně do levelsInModeClassic a levelsInModeKiller | Junit testy |
| Nesprávné přidání dokončené operace | Junit testy |
| Nesprávná generace vyřešeného sudoku | Junit testy |
| Nesprávný počet odstraněných buněk při generování hracího pole | Junit testy |
| Nenalezen urovní v seznamu levelsInModeClassic a levelsInModeKiller | Procesní testy |
| Nenastavení hodnoty pro režim Classic | Procesní testy |
| Nenastavení hodnoty pro režim Killer | Procesní testy |
| Nepřidání operace do seznamu operací | Procesní testy |

## Určení priority



*Obrázek 1: z přednášky*

## Charakteristika – Funkcionality

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Proces** | **Podproces** | **Požadavek** | **Možné poškození** | **Vysvětlení poškození** | **Část systému** | **Prav. selhání** | **vysvětlení prav. Selhání** | **Třída rizika** |
| Generování Sudoku | Generování vyřešeného Sudoku | Nesprávná generace vyřešeného Sudoku | M | Chyba při generování může vést k nesprávně vyřešenému Sudoku | Generátor | H | Postup generování je jednoduchý a stabilní | B |
| Generování Sudoku | Generování hracího pole | Nesprávný počet odstraněných buněk při generování hracího pole | M | Může dojít k nesprávnému obtížnosti hracího pole | Generátor | M | Generování závisí na více proměnných | B |
| Generování Sudoku | Generování hracího pole | Nesprávná posloupnost odstraňování buněk v režimu Sudoku Killer | M | Může dojít k nesprávnému vytvoření Sudoku Killer | Generátor | M | Náročný algoritmus pro Sudoku Killer | B |
| Správa úrovní | Přidání úrovní | Nesprávné přidání úrovně do levelsInModeClassic a levelsInModeKiller | M | Může vést k nesprávné správě úrovní | Správce úrovní | M | Operace závisí na správné inicializaci dat | B |
| Správa operací | Přidání operace | Nesprávné přidání dokončené operace | M | Dokončená operace nemusí být správně uložena | Správce operací | L | Jednoduchý proces přidání operace | C |
| Správa úrovní | Vyhledávání úrovní | Nenalezen urovní v seznamu levelsInModeClassic a levelsInModeKiller | H | Uživatel nebude moci pokračovat ve hře | Správce úrovní | M | Záleží na správném nastavení seznamu | H |
| Správa režimů | Nastavení režimu | Neastavení hodnoty pro režim Classic | H | Hra nebude fungovat správně v Classic režimu | Správce režimů | M | Jednoduché nastavení hodnoty | B |

## Test levels

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Třída | Proces | Podproces | Rizika | Revize | Vývojářské testy | Systémové testy |
| B | Generování Sudoku | Generování vyřešeného Sudoku | Nesprávná generace vyřešeného Sudoku | Ano | Ano | Ano |
| B | Generování Sudoku | Generování hracího pole | Nesprávný počet odstraněných buněk | Ano | Ano | Ano |
| B | Generování Sudoku | Generování hracího pole | Nesprávná posloupnost odstraňování buněk | Ano | Ano | Ano |
| B | Správa úrovní | Přidání úrovní | Nesprávné přidání úrovně | Ano | Ano | Ano |
| C | Správa operací | Přidání operace | Nesprávné přidání dokončené operace | Ano | Ano | Ano |
| H | Správa úrovní | Vyhledávání úrovní | Nenalezen úrovní v seznamu | Ano | Ano | Ano |
| B | Správa režimů | Nastavení režimu | Neastavení hodnoty pro režim Classic | Ano | Ano | Ano |

# Přehled testovacích technik

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Proces | Podproces | Třída rizika | Revize | Vývojářské testy | Systémové testy | Technika |
| Generování Sudoku | Generování vyřešeného Sudoku | B | Ano | vysoká | vysoká | Procesní testy hloubky 3 |
| Generování Sudoku | Generování hracího pole | B | Ano | vysoká | střední | JUnit, Procesní test 3, Pairwise |
| Generování Sudoku | Generování hracího pole | C | Ano | nízká | nízká | Procesní test hloubky 2 s parametrem |
| Správa úrovní | Přidání úrovní | C | Ano | střední | nízká | JUnit, Procesní test hloubky 2 |
| Správa operací | Přidání operace | A | Ano | střední | střední | Procesní test hloubky 2 |
| Správa úrovní | Vyhledávání úrovní | B | Ano | střední | střední | CRUD extended, procesní testy hloubky 2 |
| Správa režimů | Nastavení režimu | C | Ano | střední | vysoká | Pairwise |

# Testovací situace

## MCC – Multiple Condition

MCC (Multiple Condition Coverage) je testovací technika, která zajišťuje, že všechny možné kombinace logických podmínek jsou testovány alespoň jednou. Tato technika se používá k ověření, že všechny jednotlivé podmínky ve složeném logickém výrazu přispívají k výsledku.

A: číslo je již ve stejném řádku

B: číslo je již ve stejném sloupci

C: číslo je již na stejném čtverci.

R: zda je číslo platné v dané pozici.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test | A (Row) | B (Column) | C (Square) | R(isNumberValid) |
| 1 | 1 | 0 | 0 | 0 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 0 | 1 | 0 |
| 4 | 0 | 0 | 0 | 1 |
| 5 | 1 | 1 | 0 | 0 |
| 6 | 1 | 0 | 1 | 0 |
| 7 | 0 | 1 | 1 | 0 |
| 8 | 1 | 1 | 1 | 0 |

## MC/DC – Multiple Condition/Decision Coverage

MC/DC (Multiple Condition/Decision Coverage) je rozšířením MCC, které zahrnuje nejen pokrytí všech možných kombinací podmínek, ale také ověření, že každá podmínka může nezávisle ovlivnit výsledek rozhodnutí.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test | Rozsah | Sloupec | Řádek | Čtverec |
| 1 | 1 | 1 | 1 | 1 |
| 2 | 0 | - | - | - |
| 3 | 1 | 0 | 1 | 1 |
| 4 | 1 | 1 | 0 | 1 |
| 5 | 1 | 1 | 1 | 0 |
| 6 | 1 | 1 | 1 | 1 |

## Pairwise testing

Pairwise testing je metoda testování softwaru, která se zaměřuje na kombinace párů vstupních hodnot. Cílem této metody je snížit počet potřebných testů při zachování efektivity testování. Pairwise testing vychází z předpokladu, že většina chyb v softwaru je způsobena interakcemi mezi dvěma parametry. Proto testování všech možných párů hodnot vstupních parametrů může odhalit většinu potenciálních chyb.

### Vstupní hodnoty

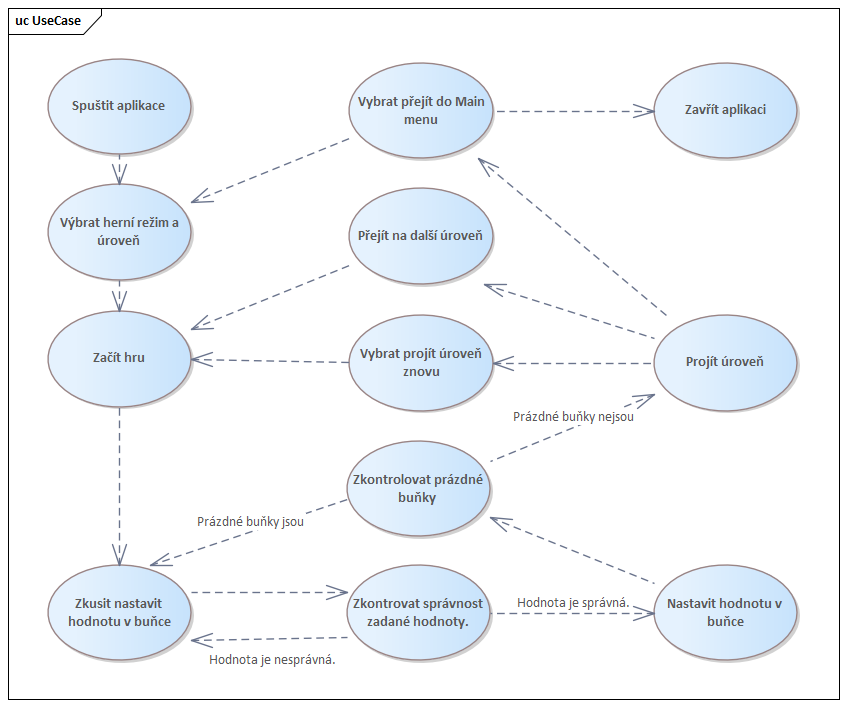
|  |  |  |
| --- | --- | --- |
| Parameter | Třida Ekvivalence | Priority |
| Sloupec | SloupecInd >= 0 | A |
| mezi | B |
| SloupecInd <= 8 | A |
| Řádek | ŘádekInd >= 0 | A |
| mezi | B |
| Řádek Ind <= 8 | A |
| Hodnota | Hodnota >= 1 | A |
| mezi | B |
| Hodnota <= 9 | A |

### Testy

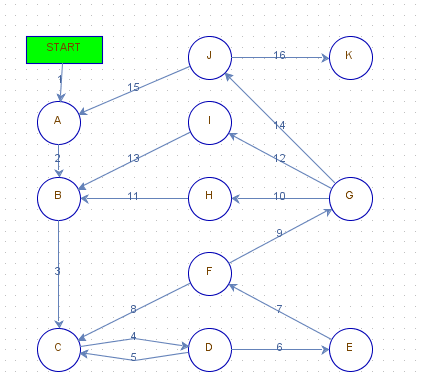
|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | SloupecInd | ŘádekInd | Hodnota |
| 1 | 0 | 0 | 1 |
| 2 | 0 | 4 | 5 |
| 3 | 0 | 8 | 9 |
| 4 | 4 | 0 | 5 |
| 5 | 4 | 4 | 1 |
| 6 | 4 | 8 | 5 |
| 7 | 8 | 0 | 9 |
| 8 | 8 | 4 | 5 |
| 9 | 8 | 8 | 1 |

# Analýza vstupů

# Testování průchodu



## Graf



## Hrany

|  |  |  |
| --- | --- | --- |
| Uzly | Vstupove Hrany | Výstupní hrany |
| A | 1, 15 | 2 |
| B | 2, 11, 13 | 3 |
| C | 3, 5, 8 | 4 |
| D | 4 | 5, 6 |
| E | 6 | 7 |
| F | 7 | 8, 9 |
| G | 9 | 10, 12, 14 |
| H | 10 | 11 |
| I | 12 | 13 |
| J | 14 | 15, 16 |
| K | 16 | - |

## 

## Hloubka pokrytí 1

### Seznam kombinací

|  |  |
| --- | --- |
| Uzly | Kombinace pro testy |
| A | 2 |
| B | 3 |
| C | 4 |
| D | 5, 6 |
| E | 7 |
| F | 8, 9 |
| G | 10, 12, 14 |
| H | 11 |
| I | 13 |
| J | 15, 16 |

### Testy

2 – 3 – 4 – 6 – 7 – 9 – 10 – 11

2 – 3 – 4 – 6 – 7 – 9 – 12 – 13

2 – 3 – 4 – 6 – 7 – 9 – 14 – 15

2 – 3 – 4 – 6 – 7 – 9 – 14 – 16

## Hloubka pokrytí 2

### Seznam kombinací

|  |  |
| --- | --- |
| Uzly | Kombinace pro testy |
| A | 1-2, 15-2 |
| B | 2-3, 11-3, 13-3 |
| C | 3-4, 5-4, 8-4 |
| D | 4-5, 4-6, |
| E | 6-7 |
| F | 7-8, 7-9 |
| G | 9-10, 9-12, 9-14 |
| H | 10-11 |
| I | 12-13 |
| J | 14-15, 14-16 |

### Testy

# Testování konzistence

Pro testování konzistence jsme použili CRUD matice, kterou jsme aplikovali i zde. Testování konzistence se bude dělat s ohledem na dva režimy hry Sudoku: klasický režim a režim Killer Sudoku.

## CRUD matice

|  |  |  |
| --- | --- | --- |
| Akce | Režim | ID |
| Vytvoření nového hracího pole pro vybraný režim  (klasický nebo Killer Sudoku) | C | 1 |
| Zobrazení hracího pole. | R | 2 |
| Výpis seznamu možných čísel pro buňku. | R | 3 |
| Pohyb po hracím poli (vkládání čísel) | U | 4 |
| Kontrola hodnot v buňce | U | 5 |
| Kontrola hodnot pro konec hry | U | 6 |
| Smazání (konec hry) | D | 7 |

## Testy

Klasický režim Sudoku:

1 – 2 – 3 – 4 – 5 – 6 – [7]

Režim Killer Sudoku:

1 – 2 – 3 – 4 – 5 – 6 – [7]

## 

## Testy Testovací scénáře

Cíl scénáře: spustit hru a zkusit si ji zahrát. Zkontrolovat správnost vyplnění buněk.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Častka aplikace | Popis kroku | Očekávaný výsledek |
| 1 | Main Menu | Spustit aplikaci. | 1. Scénář pro testování hry sudoku, režim "Classic": 2. Zobrazení hlavního menu 3. Tlačítko „Sudoku Classic“ je zvýrazněné 4. Tlačítko první úrovně je zvýrazněné 5. Zobrazení tlačítek „Back“ a „Next“ |
| 2 | Main Menu | Stisknout tlačítko "Play". | 1. Zobrazení herního pole 9x9 2. Přítomnost 5 volných buněk 3. Ostatní hodnoty jsou neaktivní 4. Zobrazení názvu režimu a čísla úrovně 5. Časovač je spuštěn 6. Zobrazení tlačítek „Undo“ a „Redo“ |
| 3 | Field „Sudoku Classic“ | Kliknout myší na prázdnou buňku. | 1. Zobrazí se nápověda „Valid values …“ 2. Buňka je vyčištěna |
| 4 | Field „Sudoku Classic“ | Přepnout fokus na jinou prázdnou buňku pomocí klávesy tabulátoru. | 1. Zobrazí se nápověda „Valid values …“ 2. Buňka je vyčištěna |
| 5 | Field „Sudoku Classic“ | Vložit hodnotu, která již existuje ve sloupci. | 1. Zobrazí se nápověda „The number is in a column“ 2. Buňka je vyčištěna |
| 6 | Field „Sudoku Classic“ | Vložit hodnotu 1-9, která již existuje v řádku. | 1. Zobrazí se nápověda „The number is in a row“ 2. Buňka je vyčištěna |
| 7 | Field „Sudoku Classic“ | Vložit hodnotu 1-9, která již existuje ve čtverci. | 1. Zobrazí se nápověda „The number is in a square“ 2. Buňka je vyčištěna |
| 8 | Field „Sudoku Classic“ | Vložit hodnotu 0. | 1. Zobrazí se nápověda „This is not a number“ 2. Buňka je vyčištěna |
| 9 | Field „Sudoku Classic“ | Vložit hodnotu větší než 9. | 1. Zobrazí se nápověda „This is not a number“ 2. Buňka je vyčištěna |
| 10 | Field „Sudoku Classic“ | Vložit spravnou hodnotu. | 1. Zobrazí se nápověda «The number is assigned» 2. vyplněna |