**SUDOKU DOKUMENTACE**

**ÚVOD**

Program Sudoku je generátor a zároveň solver her sudoku. To zamená, že umožňuje uživateli vygenerovat náhodné hry sudoku, s různou obtížností, které je možné v programu i řešit a následně ověřit správnost řešení. Zároveň program funguje jako řešička, tedy je možné nahrát libovolnou sudoku hru, a nechat ji programem správně vyřešit. V programu je možno pracovat jak s klasickou tabulkou sukodu, čili 9x9 buněk, tak i s jinými rozměry sudoku tabulek.

**UŽIVATELSKÁ DOKUMENTACE**

V této sekci se pokusím zaměřit na co nejlepší popis jednotlivých funkcionalit programu tak, aby vše bylo jasné nezasvěcenému uživateli. Po spuštění programu se zobrazí okno, které lze vidět na následujícím obrázku.

A screenshot of a game

Description automatically generated

V základu se po spuštění načte klasická sudoku tabulka, čili tabulka o rozměru 9x9 buněk.

**OVLÁDÁCÍ PRVKY:**

1. **Kombinované pole „Sudoku Size“** – při stisknutí šipky v levém rohu kombinovaného pole se rozbalí nabídka s výběrem různých rozměrů tabulek sudoku. Pro vybrání určitého typu sudoku stačí kliknout na požadovaný rozměr. Rozbalení nabídky kombinovaného pole je vidět na následujícím obrázku.

A screenshot of a computer

Description automatically generated

1. **Textové pole „Difficulty“ a tlačítko „Generate!“**– pro vygenerování nové hry sudoku je možné zadat počet klíčů, se kterými chce uživatel hrát. Tento počet se zadává do bílého pole po levé straně nápisu „Difficulty“. Čím méně klíčů je v tabulce po vygenerování zobrazeno, tím těžší daná hra je. Naopak při zobrazení moc velkého počtu klíčů je řešení triviální. Proto je pro každý rozměr tabulky sudoku zadáno rozpětí počtu klíčů, které je možno pro hraní zobrazit. Poté je třeba kliknout na tlačítko „Generate!“, což spustí akci generování a vygeneruje se nová tabulka s požadovanými rozměry a počty klíčů.

Rozpětí počtu klíčů:

* Sudoku 4x4: minimalní počet klíčů = 4, maximální počet klíčů = 6
* Sudoku 6x6: minimalní počet klíčů = 10, maximální počet klíčů = 16
* Sudoku 8x8: minimalní počet klíčů = 16, maximální počet klíčů = 32
* Sudoku 9x9: minimalní počet klíčů = 20, maximální počet klíčů = 45
* Sudoku 10x10: minimalní počet klíčů = 45, maximální počet klíčů = 70
* Sudoku 12x12: minimalní počet klíčů = 72, maximální počet klíčů = 95
* Sudoku 16x16: minimalní počet klíčů = 130, maximální počet klíčů = 180

Pokud uživatel zadá špatný počet klíčů, program vypíše chybovou hlášku. Konkrétně pro klasickou sudoku 9x9, jsou chybové hlášky o moc nízkém a vysokém počtu klíčů vidět na následujícím obrázku.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Pokud se po kliknutí na tlačítko „Generate!“ neobjeví žádná chybová hláška (počet klíčů byl zadán ve správném rozmezí), tak se v tabulce objeví požadovaný počet klíčů a je tak možné začít hrát a vyplňovat prázdné buňky. Klíče jsou v tabulce zobrazeny čistě černou barvou a čísla, která vyplňuje uživatel, se zobrazují šedou barvou, pro snadné rozeznání toho, která čísla uživatel doplnil a která jsou klíči. Na následujícím obrázku je ukázka správně vygenerované tabulky s 40 klíči a 2 čísly doplněnými uživatelem.

A grid with numbers in it

Description automatically generated

1. **Tlačítko „Verify!“** – jak bylo zmíněno ve druhém bodě, po vygenerování tabulky s klíči může uživatel začít hrát a tedy vyplňovat prázdné buňky. Po vyplnění celé tabulky si uživatel může ověřit správnost svého řešení kliknutím na tlačítko „Verify!“. Pokud je uživatelem vyplněné řešení správně, program vypíše hlášku o výhře, která je vidět na následujícím obrázku.

A screenshot of a computer

Description automatically generated

Pokud se v uživatelském řešení nachází chyby, program počet chyb spočítá, špatně vyplněná čísla zvýrazní červeně a vypíše hlášku o chybném řešení, která je vidět na následujícím obrázku.

A screenshot of a game

Description automatically generated

1. **Tlačítko „Solve!“** – jak bylo zmíněno v úvodu, program funguje také jako řešička. Tudíž pokud si uživatel neví rady s vygenerovanou hrou, stačí kliknout na tlačítko „Solve!“ a řešení dané hry se objeví v tabulce. Čísla vyplněná programem se zobrazí bíle, tak jako je vidět na následujícím obrázku.

A screenshot of a game

Description automatically generated

Řešičku je možné použít i tak, že uživatel manuálně zadá libovolná čísla, například z jiné hry sudoku, kterou chce vyřešit, a klikne na tlačítko „Solve!“. V tomto případě program zkotroluje, zda je hra řešitelná. Pokud ano, tak ji vyřeší a vypíše jako na obrázku nahoře. Pokud hra není řešitelná, vypíše chybovou hlášku, která je vidět na následujícím obrázku.

A screenshot of a game

Description automatically generated

1. **Tlačítko „Clear!“** – kliknutí na tlačítko „Clear!“ smaže všechna čísla z tabulky. To se může hodit například v případě, že by uživatel pracoval na nějaké hře a poté by chtěl zadat nějakou jinou hru manuálně. Díky tlačítku „Clear!“ není potřeba každou buňku mazat zvlášť.
2. **Tlačítko „Save!“** – tlačítko „Save!“ slouží k uložení tabulky do textového souboru. Uživatel si tak může rozehranou tabulku uložit a vrátit se k jejímu řešení později. Tabulka se uloží jako textový soubor se jménem „sudokuX“, kde „X“ značí číslo, které se zvyšuje podle toho, kolik tabulek si již uživatel do svého počítače uložil. Čísla v tabulce se uloží ve formátu, který je vidět na následujícím obrázku (podtržítko značí prázdnou buňku).

A screenshot of a computer

Description automatically generated

1. **Tlačítko „Load!“** – tlačítko „Load!“ slouží k nahrání tabulky sudoku z textového souboru. Po kliknutí na tlačítko se uživateli otevře průzkumník souborů, ve kterém může najít a vybrat textový soubor, který chce načíst. Textový soubor však musí vypadat typově přesně jako na obrázku výše a zároveň být korektní tabulkou sudoku. Pokud se v při nahrávání v souboru vyskytují čísla, která nejsou v rozsahu 1-9, nebo písmena, která nejsou v rozsahu A-G (používají se u sudoku 10x10 a větší), nebo pokud nesedí počet řádků či sloupců, program vypíše chybovou hlášku, kterou lze vidět na obrázku níže.

A screenshot of a computer error message

Description automatically generated

**PROGRAMÁTORSKÁ DOKUMENTACE**

V této sekci se budu snažit zaměřit na popis implementace jednotlivých logických celků kódu. Rozdělení těchto celků se přibližně řídí objektovým návrhem, který jsem vytvořila, který je vidět na obrázku níže. Detailní dokumentace je pak dostupná přímo ve zdrojovém kódu, ve formě komentářů.

A diagram of software components

Description automatically generated

Základním algoritmem pro generování a řešení her sudoku je algoritmus backtracking (DFS) s ořezáváním neperspektivních řešení pomocí seznamů v buňkách tabulky, kvůli velikosti stavového prostoru. Vhodnou optimalizací je vyplňování diagonálních čtverců tabulky nezávisle na sobě náhodně příslušnými čísly bez opakování. Po vyplnění diagonálních čtverců začneme vyplňovat zbytek tabulky právě algoritmem backtracking. Pokaždé když nějaké buňce přiřadíme číslo, zavoláme metodu, která smaže toto číslo ze seznamu použitelných čísel všech buněk ve stejném řádku, sloupci a čtverci. V případě, že v průběhu výpočtu nějaké buňce nezbyde žádné možné číslo k přiřazení, se vrátíme zpět do posledního bezpečného stavu a zavoláme metodu, která naopak přidá zpátky do seznamů buněk číslo, které bylo smazáno.

**LOGICKÉ CELKY PROGRAMU**

1. **UI VIEW** – viz uživatelská dokumentace
2. **VIEW CONTROLER** – v tomto případě představuje kód Form1.cs. Při inicializaci se v základu vytvoří tabulka klasické Sudoku, tedy o rozměru 9x9. View controler obsahuje všechny metody, které reagují na akce provedené uživatelem (event handlery). Tedy například metodu pro změnu rozměru tabulky vybraném v ComboBoxu, nebo metodu pro spuštění generování nové hry se zadaným počtem klíčů po kliknutí na Button „Generate!“. Zároveň uchovává aktuální instanci sudoku.
3. **SUDOKU** – představuje kód Sudoku.cs. Zde jsou uchovány všechny vlastnosti potřebné k vytvoření libovolné tabulky sudoku, čili velikost tabulky (rozměr), velikost čtverců, do kterých je sudoku členěna, minimální a maximální počet klíčů, který může být uživateli zobrazen, velikost buněk a velikost písma (velikost tabulky je potřeba dynamicky měnit, jelikož nejmenší sudoku 4x4 má pouze 16 buněk, zatímco sudoku 16x16 má buněk 256). Také obsahuje metody pro smazání čísel z tabulky, obnovení tabulky do původního stavu a metody pro uložení a načtení textových souborů.
4. **SUDOKU CELL** – představuje kód Cell.cs, tato třída dědí od třídy „Button“. Zde jsou uchovány všechny vlastnosti potřebné k vytvoření jedné buňky. Tabulka sudoku je pak tvořena právě z těchto buněk. K základním vlastnostem patří velikost, umístění, barva písma a ohraničení, hodnota, která je v buňce uchována a seznam hodnot možných do buňky přiřadit. Dále obsahuje metody pro smazání textu buňky, uvedení buňky do původního stavu (nastavení hodnoty na nulu a obnovení seznamu možných hodnot), nebo také metody, které převádí písmena A-G na čísla 10-16 a obráceně, což se hodí při velikostech tabulky 10x10 a větší, kdy se při vyplňování tabulky lépe pracuje s číselnými hodnotami.
5. **GRID GENERATOR** – představuje kód GridGenerator.cs, který zodpovídá za vytvoření tabulky na základě zadaných argumentů. Nastavuje počet buněk v tabulce, rozložení barev do logických čtverců tabulky a také možnost uživateli do tabulky psát a mazat text.
6. **GENERATOR** – představuje kód Generator.cs a SudokuCore.cs. Třída SudokuCore.cs je společná jak generátoru tak solveru sudoku, proto je vyčleněný zvlášť. Generator zodpovídá za vygenerování nové hry sudoku a to následovně. Prvním krokem je vyplnit diagonální čtverce náhodným výberem čísel ze seznamů možných čísel buněk. Jak je popsáno výše u zvoleného algoritmu, pokaždé když přiřadíme nějaké buňce číslo, tak toto číslo smažeme ze seznamů možných čísel všech buněk ve stejném řádku, sloupci a čtverci. Druhým krokem je vyplnit zbytek tabulky algoritmem backtracking, který je implementovaný právě v třídě Sudokucore.cs, jelikož je společný generátoru i solveru. Algoritmus prochází postupně celou tabulku od první buňky a posouvá se po sloupích. Podívá se do seznamu možných hodnot právě zpracovávané buňky a z čísel které seznam obsahuje vybere náhodně (pomocí random funkce) jednu hodnotu, kterou buňce přiřadí, zanoří se do rekurze a posune se na další buňku. Pokud nastane problém, že nějaké buňce nezbyde žádné možné číslo k přiřazení, se vrátíme zpět do posledního bezpečného stavu a zavoláme metodu, která naopak přidá zpátky do seznamů buněk číslo, které bylo smazáno. Třetím krokem je pak podle zadaného počtu klíčů zviditelnit uživateli jen určitý počet čísel. Pro tento krok je implementována metoda, která opět pracuje s random funkcí, aby byla zviditelněná čísla rozložená po celé tabulce. Pro ověření výsledku pak pouze porovnáme uživatelem vyplněná čísla s vygenerevanou tabulkou.
7. **SOLVER** – představuje kód Solver.cs a SudokuCore.cs. V této třídě je implementována metoda, která z tabulky načte čísla zadané uživatelem, ta si uloží a zavolá ze SudokuCore.cs metodu pro vyplnění tabulky algoritmem backtracking. Pro případ, že uživatel zadá neřešitelné hodnoty, jsou připraveny chybové hlášky. Dále obsahuje metodu, která po vyřešení tabulky zviditelní všechna čísla uživateli jako správné řešení.
8. **I/O HANDLER** – představuje kód Sudoku.cs a Form1.cs. Ve třídě Sudoku.cs jsou implementovány metody pro uložení tabulky sudoku do textového souboru a naopak pro načtení tabulky z textového souboru. Ukládání tabulky je provedeno pomocí třídy „StreamWriter“, kdy se jeden řádek sudoku zapisuje do jednoho řádku textového souboru. Pro přiřazení unikátního jména souboru procházíme soubory v počítači uživatele pomocí třídy „File“a kontrolujeme jaké soubory sudoku již má uložené. Pro načtení textového souboru do sudoku tabulky je implementována metoda, která přijímá soubory s příponou \*.txt. Na základě počtu řádků vybere rozměr sudoku tabulky a po řádcích souboru načítá čísla do sudoku tabulky. Tyto metody jsou pak volány ze třídy Form1.cs při kliknutí na příslušná tlačítka.

**SHRNUTÍ**

Původním cílem tohoto projektu bylo vytvořit aplikaci, která bude generovat a řešit hry klasické Sudoku, o rozměrech 9x9 buněk. V průběhu vypracování projektu jsem se rozhodla tento koncept rozšířit na více rozměrů tabulek. Pro sudoku 16x16 je generování a řešení tabulek pomalejší, kvůli velikosti prohledávaného stavového prostoru.

Závěrem, jelikož aplikace obsahuje funkční generátor i solver pro všechny popsané typy sudoku tabulek, věřím že cíl projektu a původní specifikace byl splněn.