**Софийски университет „Св. Климент Охридски“**

*Факултет по математика и информатика*

*Специалност: “Информационни системи”*

*Курс: 2, Група: 4*

*Проект No. 10 Обектно Ориентирано Програмиране*

*Судоку*

***Изработил:***

Иф Николаев Антонов (Фак. №71875)

***Проверител:***

Професор, д-р Евгений Кръстев

Съдържание:

2. Упътване за ползване на приложението...........................................................Страница 3

3. Структура........................................................... Страница 4-5

4. Алгоритми......................................................... Страница 6-7

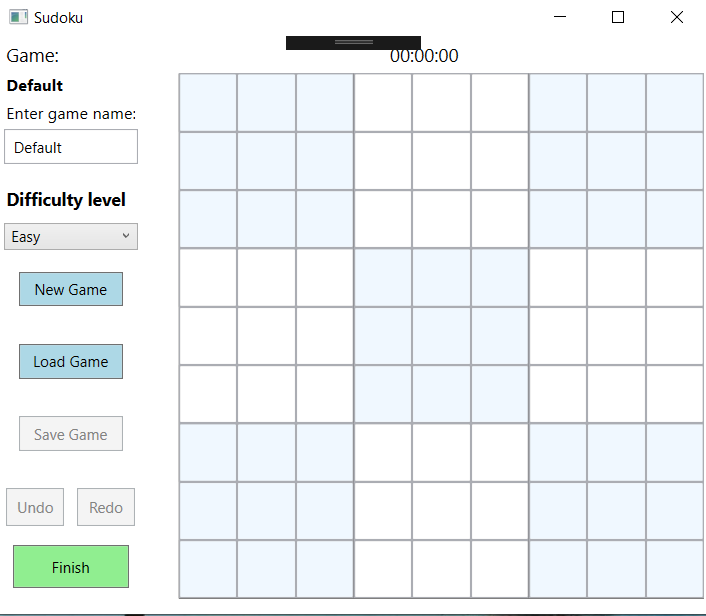
5. Използвани материали................................................................ Страница 8

6.Допълнение...........................................................Страница 8

Упътване

UI

Таймер за отчина на времето



Бутони за ход назад или напред

Можете да въведете трудност на играта.

Бутон за проверка за край на играта.

Бутони за Нова игра, зареждане или запаметяване на игра. Проверете името по-горе дали съответства на играта, която искате да запазите или заредите.

Можете да въведете името на играта, а то ще се появи по-горе

Структура

Приложението е направено от един WPF проект. UI-ът е във файл - MainWindow.xaml. Има бутони за стартиране на нова игра, запазване или зареждане на игра, както и бутони за ход назад и напред. Text box и textBlocks за име на играта. ComboBox за трудността. TextBlock за отчитане на времето. Основната част е таблицата от клетки за судокуто. Направена е с 9 Borders за 9-те кутийки. Вътре има Rectangles за разграничаване на клетките във всяка кутийка. Във всяка клетка има TextBox за въвеждане на число. Binding се използва, за да се изобразяват промените направени в кода директно на UI-a. Той е в TwoWay метод, за да може и промените от UI-а да се отразят и в кода. Кодът зад файла MainWindow.xaml е във файл: MainWindow.xaml.cs. Той имплементира функционалностите, когато натиснем бутон, сложим времето за отчитане всяка секунда и т.н. Също има методи за генериране на судоку. В Проекта има PositionValue.cs файл, който е клас, който държи стойността и координатите на една клетка в матрицата. Използва се при Undo и Redo. Основната част е файлът Sudoku.cs. Притежава клас Судоку, който има:

- Матрица от стрингове

- Два Стака, които са всъщност контейнери за Undo и Redo функионалностите. Те са от тип PositionValue

-Две булеви стойности: едната за проверка ако едно судоку е решено, а другата за добавяне в стека Undo

-Събитие OnPropertyChanged когато се сменя дадена клетка

Properties за всяка клетка в UI-а, за да може да се осъществява binding. Всяка има валидация в сет метода. Undo стекът се пълни от там. Всяка клетка има еднакъв код освен нейните координати

Методи (Алгоритмите са описани по-долу):

-Проверка за валидна позиция

-За попълване на матрицата

-Проверка за решено судоку

-За запазване на игра във файл

-За зареждане на игра от файл

Алгоритми

Генериране на судоку – специални алгоритми, които построяват валидно судоку и после го разбъркват, за да бъде различно всеки път.

Трудност – 3 нива. Трудността е всъщност колко клетки ще се премахнат след построяването на судокуто. За по-бързо махане – всяка итерация две клетки се премахват. Изборът на клетка е случаен. За проверка дали дадена клетка вече е премахната се използва булева матрица със стойност false ако може да се премахне и със стойност true ако вече е премахнат елемент от там.

За запазване или зареждане на игра се използват файлове. Когато се запазва – името от UI-a се взима и файлът се създава с това име. Ако вече съществува такова име – се създава нов файл с име Default + число (Пример Default – Default1, Default2 и т.н.). Взима матрицата и записва стойностите ѝ във файла. След това взима таймера и го записва и него вътре.

При зареждане – ако файлът не е намерен – съобщение се изписва на екрана. Ако е намерен – чете данните за матрицата и ги налива в нея. След това взима данните за часовника и го задейства отново като затваря минал таймер ако е бил в действие досега.

За проверка ако позиция е валидна – проверява дали числото вече го има в кутийката 3x3, на същия ред или същата колона. Ако го има – числото не се записва.

За проверка дали судокуто е решено – ако матрицата е пълна с числа – то е готово, защото други символи не са допустими при решаването. А валидацията се грици те да са на правилното място.

Използвани материали

* Data Binding лекции.
* <https://gist.github.com/fabiosoft/b41067106bebf1498399f4eb9826e4de> – Алгоритми за генериране на судоку
* WPF лекции.
* <https://www.wpf-tutorial.com/misc/dispatchertimer/> - Таймер в UI-а

Допълнение:

Има два тестови файла, които ползвах в Sudoku/SudokuForm/SudokuForm/bin/Debug

При запазване на файл – се запазват в тази папка, защото не е зададен път. Ако се тестват Save и Load – файловете се търсят на пътя подаден по-горе.