**Документација**

1. **Опис на апликацијата**

Апликацијата што ја развиваме е играта Sudoku и притоа застапуваме 2 варијанти на играта: Classic Sudoku и Squiggly Sudoku.

Sudoku(Classic Sudoku) е игра каде што има 9 региони( димензии 3 X 3) кои што треба да се пополнат со броевите од 1 до 9 за да се победи играта и притоа:

-Не смее да се повторуваат два или повеќе исти броја во една колона

-Не смее да се повторуваат два или повеќе исти броја во еден ред

-Не смее да се повторуваат два или повеќе исти броја во едно поле.

Squiggly Sudoku е варијанта на Classic Sudoku при тоа што ги има истите правила од Classic Sudoku со една варијација – регионите имаат различни димензии.

Во апликацијата постои можност за селектирање на ниво на тежина (easy , medium , hard). Овозможено е да се започне нова игра како и да се продолжи веќе започната игра. Кога ќе се победи една игра, резултатот од играчот се зачувува во листа на рекорди (High scores).

1. **Решение на проблемот**

Проблемот е поделен во класи кои претставуваат определени единки од играта:

PuzzleGrid е класата што ја претставува формата на Classic Sudoku. Таа во неа содржи Матрица од Integer вредности што всушност се полињата како и константа Max што го ја означува максималната големина и е секогаш 9.

Во PuzzleGrid ги имаме функциите:

* public PuzzleGrid()
* public int InitSetCell(int rowA, int columnB, int value)
* public int UserSetCell(int rowA, int columnB, int value)
* public void Finish()
* public object Clone()
* public bool IsBlank()

PuzzleSolver е класата што чува информации за точното решение ??? Таа содржи објект од класата PuzzleGrid

Во PuzzleSolver ги имаме функциите:

* public PuzzleSolver()
* public bool IsSolved(PuzzleGrid grid)
* public int FirstTrue()
* public int PickOneTrue()
* public bool IsInRow(PuzzleGrid grid, int row, int value)
* public bool IsInRow(PuzzleGrid grid, int row, int value,out int ii, out int jj)
* public bool IsInCol(PuzzleGrid grid, int col, int value)
* public bool IsInCol(PuzzleGrid grid, int col, int value,out int ii, out int jj)
* public int GroupNum(int rc)
* public bool IsIn3X3(PuzzleGrid g, int row, int col, int value)
* public bool IsIn3X3(PuzzleGrid g, int row, int col, int value,out int ii, out int jj)
* public bool IsPossible(PuzzleGrid g, int row, int col, int value)
* public int ListPossible(int row, int col, PuzzleGrid g)
* public void FillSingleChoices(PuzzleGrid grid)

public bool FindFewestChoices(PuzzleGrid grid, out int r, out int c,

* out int numChoices)
* public void ClearList()
* public bool SolveGrid(PuzzleGrid g, bool checkUnique)

SquigglyGrid е класата што ја претставува формата на Classic Sudoku. Таа во неа содржи Матрица од Integer вредности што всушност се полињата како и константа Max што го ја означува максималната големина и е секогаш 9.

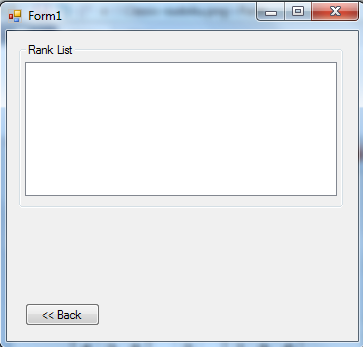
SquigglySolver e класата што чува јавен интерфејс SquigglyGrid , така што го зема Grid-ot како влез, пробува да го реши, и враќа вредност boolean во зависност од ота дали може да го реши или не. Int [] GridRow го чува моменталниот ред од Grid-от. bool [] List ги чува сите можни вредности во листата. SquigglyGrid final чува низа од пронајдени решенија. Int NumSolns го чува бројот на решенија. Bool stoplooking кажува дали да продолжи со барање решенија. Int recursions го дава бројот на рекурзии. const int MaxDepth ја дава горната граница на рекурзии. int [,] scheme ја дава...?

1. **Упатство за користење**

****

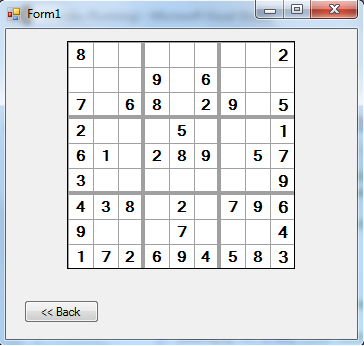
**Слика 1.**

На почетниот прозорец(слика 1) при стартување на апликацијата имаме можност да започнеме нова игра(New game), да продолжиме веќе постоечка игра(Load game) како и да видиме листа со рекорди(High scores)(слика 2).

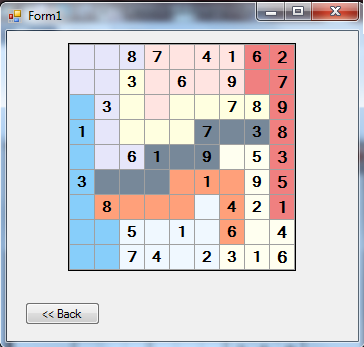


**Слика 2.**

Доколку сакаме да започнеме нова игра најпрво се селектира тип на судоку што сакаме да решаваме. Има две понудени опции Classic Sudoku-предефинирано(слика 2) и Squiggly Sudoku(слика 3). Доколку сакаме селектираме тежина(easy,medum и hard) различна од предифинираната и притискаме на копчето New Game.



**Слика 3.**

****

**Слика 4.**

Играта се игра така што селектираме поле по наш избор. Доколку селектираме поле со предефинирана вредност со некој од броевите од 1 до 9, ќе засветат останатите исти вредности од матрицата но притоа не сме во можност да внесеме нова вредност.

Доколку сакаме да внесеме или избришеме вредност која не е предефинирана, го селектираме полето што сакаме да го измениме и притоа ја пишуваме новата вредност. Полињата се пополнуваат се додека во играта не се полни сите полиња со точните вредности. Кога ќе се внесат сите точни вредности излегува прозорче за внесување на името на играчот во листата со рекорди.