КУРСОВ ПРОЕКТ

на тема:

СУДОКУ

Иван Цветанов Младенов, фн 61950

1. Цел

Целта на настоящия курсов проект е да се направи .NET WPF приложение, реализиращо популярната игра Судоку.

1. Имплементация

Судоку представлява игра, в която има таблица с 9 реда и 9 колони. В настоящото приложение тази таблица е реализирана като 9 таблици, в които има по 9 текстови полета, обединени в User control. В основния прозорец има добавеният User Control, 6 бутона, таймер и combo box меню.

* Combo box менюто предлага избор на ниво. (Beginner/Easy/ Medium/ Hard/Expert)
* Бутонът “New Game” зарежа нова игра в зависимост от избраното ниво на трудност.
* Бутонът “Load” зарежда игра, която е била записана във файл.
* Бутонът “Save” запазва текущата игра във файл.
* Бутонът “Check” проверява настоящото Судоку и в зависимост от резултата, показва различно съобщение.
* Бутонът “Clear” изчиства дъската от попълнените полета.
* Бутонът “Exit” служи за изход от приложението.
* Таймерът представлява текстово поле, чието съдържание се променя на всяка секунда.

MainWindow има пет private полета (DispatcherTimer, Stopwatch, колекция от всички текстови полета на Sudoku User control-a, обект от тип SudokuGame и GameGenerator, който създава игри).

Как е реализиран класът SudokuGame?

Класът Sudoku има поле board, което представлява двумерен масив от тип byte с размери 9х9. Това поле си има публични get и set методи. Има също и поле от тип TimeSpan, в което се запазва изминалото време. В конструктора на класа се инициализира масивът. Класът е обявен като Serializable, за да може да се сериализира и записва във файлове.

Класът GameGenerator има следните методи:

* Public int[,] GetGame(Difficulty difficulty) – публичен метод, който генерира двумерен масив според зададено Difficulty.
* Private int GetNumberOfCellsFilled(Difficulty difficulty) – метод, който генерира произволно число, според трудност (има горна и долна граница на произволното число), което ще представлява броя на попълнени клетки
* Private bool IsLegalMove(int x, int y, int current) – метод, който проверява дали е възможно да се постави числото на текущата позиция
* Private bool GetNextCell(int x, int y) – метод, който попълва полето поетапно
* Private int[,] CopyBoard() – метод, който прави копие на полето
* Private void MakeHoles(int holesToMake) – метод, който „дупчи“ полето, според трудността

Класът MainWindow има следните методи:

* Конструктор, който инициализира компонента, извиква метода за инициализацията на таймерите, както и попълва колекцията от текстови полета на компонента
* private void ButtonBase\_Click(object sender, RoutedEventArgs e) – метод, който изключва текущото приложение.
* private void BtnCheck\_OnClick(object sender, RoutedEventArgs e) – метод, който спира таймера, записва стойностите на всички текстови полета в board и проверява дали судокуто е решено правилно.
* private void BtnClear\_OnClick(object sender, RoutedEventArgs e) – метод, който изчиства текстовите
* private void BtnLoad\_OnClick(object sender, RoutedEventArgs e) – метод, който отваря нов диалогов прозорец за отваряне на файл и зарежда указаното судоку.
* private void BtnSave\_OnClick(object sender, RoutedEventArgs e) – метод, който спира таймера, отваря нов диалогов прозорец за записване и записва текущото судоку в указаната директория.
* void Dispatcher\_Tick(object sender, EventArgs e) – метод, който взима текущо изминалото време и го присвоява на текстовото поле, което показва времето.
* private void FillBoard() – метод, който запълва текстовите полета на User Control-a със съответните им стойности в board.
* Private void FillTable() – метод, който запълва полетата на SudokuGame инстанцията на прозореца
* private void IsFull() – метод, който обхожда всички текстови полета на User Control-a и проверява дали цялото поле е попълнено.