Programming Sudoku

Wei-Meng Lee

Apress®

Programming Sudoku

Copyright © 2006 by Wei-Meng Lee

ISBN-13 (pbk): 978-1-59059-662-3

ISBN-10 (pbk): 1-59059-662-5

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail `orders-ny@springer-sbm.com`, or visit `http://www.springeronline.com`.

For information on translations, please contact Apress directly at 2560 Ninth Street, Suite 219, Berkeley, CA 94710. Phone 510-549-5930, fax 510-549-5939, e-mail `info@apress.com`, or visit `http://www.apress.com`.

The source code for this book is available to readers at `http://www.apress.com` in the Source Code section. You will need to answer questions pertaining to this book in order to successfully download the code.

CHAPTER 2

Creating the Sudoku Application

Now that you have a firm grounding in the basic rules of Sudoku, it is time for us to start the journey into solving Sudoku puzzles using computer programming. For this task, you will build a Windows application that represents a Sudoku puzzle. The application that you build in this chapter will act as a rule enforcer, helping you to make sure that a value inserted into a cell does not violate the rules of Sudoku. We aren't concerned about how to solve a Sudoku puzzle yet; we leave that for the next few chapters.

In this chapter, I walk you through the various steps to construct a Sudoku puzzle board using a Windows application. This is the foundation chapter that all future chapters will build on. While the application that you build in this chapter lacks the intelligence required to solve a Sudoku puzzle, it will provide you with many hours of entertainment. Moreover, it will provide some aid to beginning Sudoku players, because it helps to check for the rules of Sudoku. Your Sudoku application will have the capabilities to do the following:

- Load and save Sudoku puzzles
- Ensure that only valid numbers are allowed to be placed in a cell
- Check whether a Sudoku puzzle has been solved
- Keep track of the time needed to solve a Sudoku puzzle
- Undo and redo previous moves

As in all large software projects, I will be breaking the functionalities of the Sudoku application into various functions and subroutines. The following are the major tasks in this chapter:

- Creating the user interface of the Sudoku application
- Using arrays to represent values in the grid
- Storing the moves using the stack data structure
- Generating the grid dynamically using Label controls
- Handling click events on the Label controls
- Checking whether a move is valid
- Checking whether a puzzle is solved
- Updating the value of a cell
- Undoing and redoing a move
- Saving a game
- Opening a saved game
- Ending the game

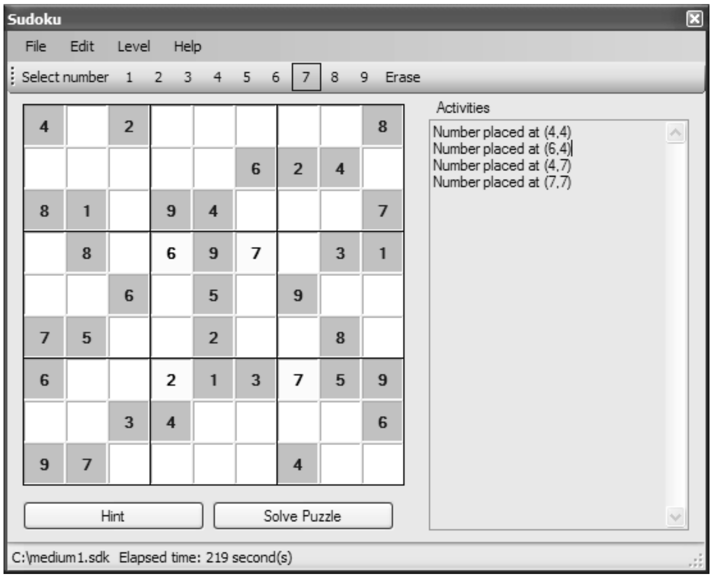
At the end of this chapter, you will have a functional Sudoku application that you can use to solve your Sudoku puzzles!

Creating the Sudoku Project

The application that you will build in this chapter is a Windows application. Figure 2-1 shows how the application will look at the end of this chapter.

Using this application, users will be able to load and save puzzles to disk. The application will act as a rule enforcer, ensuring that the user cannot place a number in a cell that will violate the rules of Sudoku. This is useful for beginners who are learning Sudoku.

■Note The application in this chapter will not have the intelligence to solve a Sudoku puzzle yet. You will begin building the intelligence in Chapter 3.



```
Number placed at (4,4)
Number placed at (6,4)
Number placed at (4,7)
Number placed at (7,7)
```

C:\medium1.sdk Elapsed time: 219 second(s)

Figure 2-1. *The Sudoku application you will build in this chapter*

Creating the User Interface

For the Sudoku application, you will create a Windows application using Microsoft Visual Studio 2005. Launch Visual Studio 2005. Choose File ➤ New Project, select the Windows Application template, and name the project **Sudoku**.

Note Throughout this book, I will use Visual Basic 2005 as the programming language. C# programmers should not have any major problem understanding/translating the code.

The project contains a default Windows form named `Form1`. Set the properties of `Form1` as shown in Table 2-1. To change the property of a control in Visual Studio 2005, right-click the control and select Properties to open the Properties window.

Table 2-1. *Properties of Form1*

Property	Value
FormBorderStyle	FixedToolWindow
Size	551, 445
Text	Sudoku

Figure 2-2 shows how `Form1` will look like after applying the properties listed in Table 2-1. Essentially, you are creating a fixed-size window.

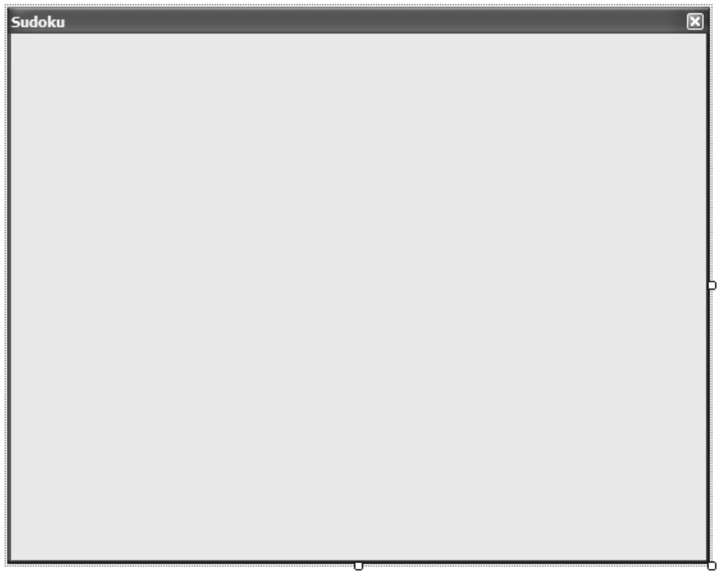


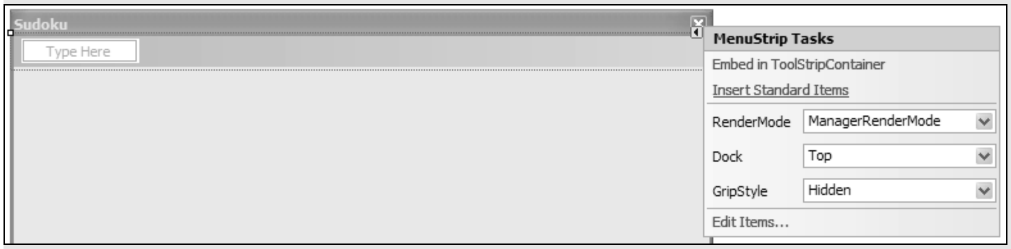
Figure 2-2. *Modifying Form1*

Adding a MenuStrip Control

In the Toolbox, double-click the MenuStrip control located on the Menus & Toolbars tab to add a menu to `Form1`. In the MenuStrip Tasks menu (also known as a Smart Tag), click Insert Standard Items to insert a list of standard menu items.

SMART TAGS IN VISUAL STUDIO 2005

A Smart Tag is a panel that is displayed next to a control (by clicking the arrow icon at the top-right corner of the control), containing a list of commonly used properties. By saving you a trip to the Properties window for some of the more common properties you need to set, Smart Tags can improve development productivity. Smart Tags are a new feature in Visual Studio 2005.



RenderMode ManagerRenderMode
Dock Top
GripStyle Hidden

Once the standard menu items are inserted, you can customize the menu by removing menu items that are not relevant (use the Delete key to remove menu items) and inserting new items. Figure 2-3 shows the different menu items that you will add for this application.

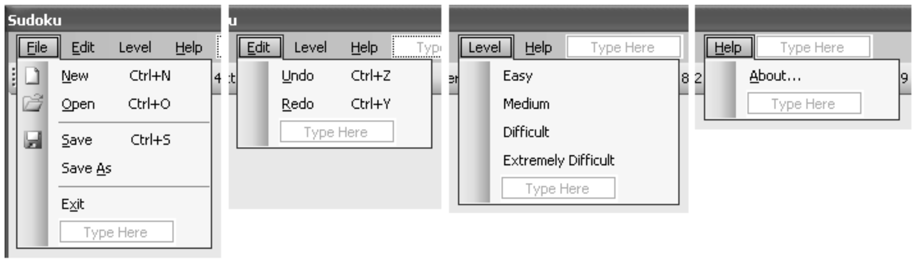


Figure 2-3. *The menu items for the Sudoku application*

Tip The standard menus by default include a Tools menu rather than a Level menu. You can simply replace the Tools menu with the Level menu. In addition, you can change the menu items to Easy, Medium, Difficult, and Extremely Difficult. For the File, Edit, and Help menus, if you want to delete any menu items, simply select the unwanted item and press the Delete key.

To assign shortcuts to the different levels of difficulty, click each of the Level menu items and enter the values as shown in Table 2-2 (see also Figure 2-4).

Table 2-2. *Values to Set for the Level Menu and Its Menu Items*

Menu/Item	Value
Level	&Level
Easy	&Easy
Medium	&Medium
Difficult	&Difficult
Extremely Difficult	Ex&tremely Difficult

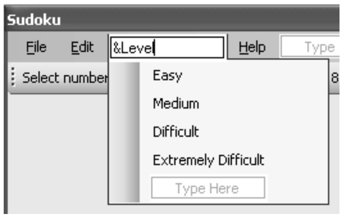


Figure 2-4. *Setting the values for the Level menu and its menu items*

After you set the values, the Level menu looks like Figure 2-5.

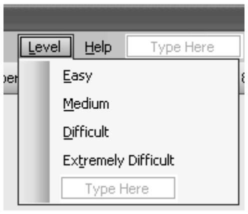


Figure 2-5. *The Level menu and its menu items*

Adding a ToolStrip Control

You will now add a ToolStrip control to the Windows form so that users can choose a number to insert into the cells. In the Toolbox, double-click the ToolStrip control (also located on the Menus & Toolbars tab) to add it onto `Form1`. You need to add Label and Button controls to the ToolStrip control; Figure 2-6 shows how to add controls to a ToolStrip control.

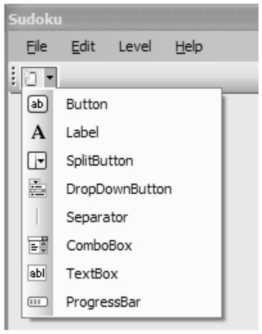


Figure 2-6. *Adding controls to the ToolStrip control*

Add a Label control to the ToolStrip control, and set the `Text` property of the Label control to **Select number**.

Next, add ten Button controls to the ToolStrip control. Set the `DisplayStyle` property of each Button control to Text. Set the `Text` property of the ten Button controls to **1**, **2**, **3**, **4**, **5**, **6**, **7**, **8**, **9**, and **Erase**, respectively, as shown in Figure 2-7, which depicts how the finished ToolStrip control should look.

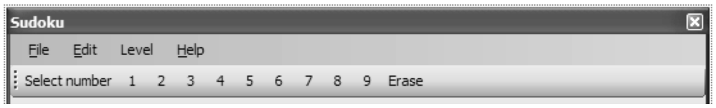


Figure 2-7. *The finished ToolStrip control*

Adding a StatusStrip Control

You will also add to the bottom of the form a StatusStrip control (also located on the Menus & Toolbars tab). Click the StatusStrip control on the form and insert two StatusLabel controls (see Figure 2-8).

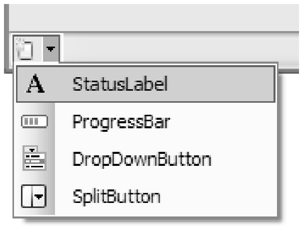


Figure 2-8. *Populating the StatusStrip control*

Figure 2-9 shows the form at this stage.

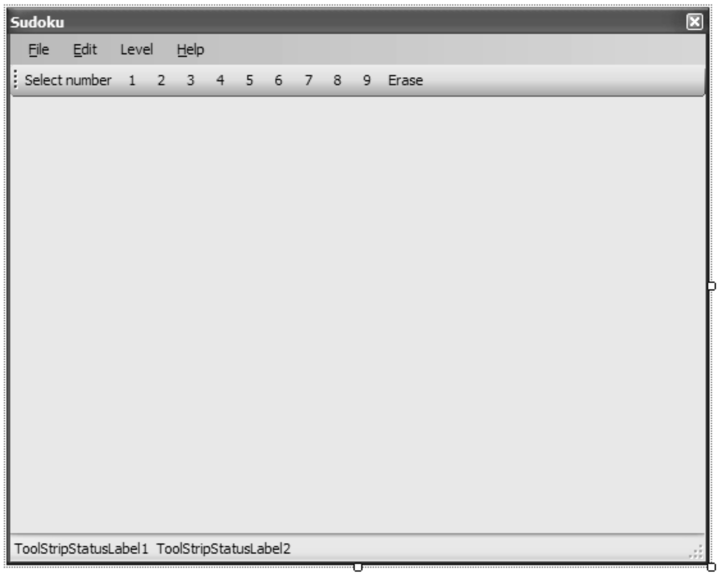

Sudoku
File Edit Level Help
Select number 1 2 3 4 5 6 7 8 9 Erase
ToolStripStatusLabel1 ToolStripStatusLabel2

Figure 2-9. *The form with the various menu controls*