

# Software Requirements Specification

for

*Super Sudoku*

*Created by Microware Concepts*

*2/10/2011*

Version	Release Date	Responsible Party	Major Changes
1.0	2/10/11		Release for design

## ***Table of Contents***

<b>1.0 Introduction .....</b>	<b>Page 3 of 15</b>
<b>1.1 Purpose.....</b>	<b>Page 3 of 15</b>
<b>1.2 Scope .....</b>	<b>Page 3 of 15</b>
<b>1.3 Definitions, Acronyms, and Abbreviations .....</b>	<b>Page 4 of 15</b>
<b>1.4 References.....</b>	<b>Page 4 of 15</b>
<b>2.0 General Description.....</b>	<b>Page 5 of 15</b>
<b>2.1 Product Perspective.....</b>	<b>Page 5 of 15</b>
<b>2.2 Product Function Summary .....</b>	<b>Page 5 of 15</b>
<b>2.3 User Characteristics .....</b>	<b>Page 5 of 15</b>
<b>2.4 General Constraints .....</b>	<b>Page 6 of 15</b>
<b>2.5 Assumptions and Dependencies.....</b>	<b>Page 6 of 15</b>
<b>3.0 Specific Requirements .....</b>	<b>Page 7 of 15</b>
<b>3.1. Functional Requirements .....</b>	<b>Page 7 of 15</b>
<b>3.2. External Interface Requirements .....</b>	<b>Page 10 of 15</b>
<b>3.3. Use-Case Descriptions .....</b>	<b>Page 11 of 15</b>
<b>3.4. Performance Requirements .....</b>	<b>Page 11 of 15</b>
<b>3.5. Design Constraints .....</b>	<b>Page 11 of 15</b>
<b>3.6. Quality Characteristics .....</b>	<b>Page 11 of 15</b>
<b>3.7. Other Requirements.....</b>	<b>Page 12 of 15</b>
<b>4. Supporting Information .....</b>	<b>Page 13 of 15</b>
<b>Appendix A: Preliminary User's Guide.....</b>	<b>Page 13 of 15</b>

## 1.0 Introduction

The *Super Sudoku* program is a computer game based on the popular paper and pencil Sudoku game.

### 1.1 Purpose

This SRS describes the complete required functionality for *Super Sudoku*, specifying all of the features available to the program user, and all of the processing capabilities which must be implemented in order for the game to work properly. It is intended to be read by both the program specifier (the customer) and the software designers.

This document provides the contractual basis for all work estimates, the requirements specification for all future program design work, and the standard for final program validation. It is the sole, complete, written authority to which the final program will be compared, and the basis for all testing and acceptance of the completed program.

***This document contains various descriptions and explanations of the program functionality to be provided, all of which are to be implemented and verified during program validation. However, for traceability purposes, only the specifications stated in this document which have the format of an indented number followed by a ')', like:***

***1) Some requirement***

***shall be interpreted as the complete set of verifiable requirements. They shall be numbered with the section number, appended with the indented number, so that, for example, the above indent would be numbered as 1.1.1.***

### 1.2 Scope

The program to be developed is called *Super Sudoku For Windows* (abbreviated by the acronym *SSFW* throughout the remainder of this document). It will be a stand-alone program for PCs running Windows XP or later operating systems. The program will offer the user a GUI interface, matching standard Windows program menu standards, and will:

- 1) generate Sudoku games for individual players to play
- 2) optionally provide hints and solutions to the player during the game
- 3) solve Sudoku games entered from other sources using the keyboard
- 4) save and load games for later play

Sudoku is a popular logic game, providing users with varying levels of challenge. *SSFW* will provide a convenient, computer-based version of the game which will allow users to improve their game-play by easily changing their guessed input values, optionally seeing clues and correct answers for the input values, and

keeping a score based on the player's elapsed time to solve the puzzle, and the puzzle level of difficulty. The objective is to provide the user with an intuitive, easy-to-use interface which replicates very closely the basic functioning of the paper and pencil version of the game, but with the convenience of keyboard inputs. The user will have all the same challenges of the original game, with the additional benefits of having hints or answers provided for individual unfilled squares within the game grid.

### **1.3 Definitions, Acronyms, and Abbreviations**

#### **Definitions:**

*Sudoku* is the name of a popular number-based logic game which challenges the player to complete a 9 x 9 grid of squares, each containing either the digits one through nine or a blank, by filling in the blank squares. The object of the game is to place numbers into the empty squares so that each row, each column and each of the nine 3 x 3 sub-grids within the larger 9 x 9 grid contains the numbers one through nine once.

#### **Acronyms:**

*SSFW* is the acronym for the full name of the game program, *Super Sudoku For Windows*,

#### **Abbreviations:**

None identified.

### **1.4 References**

This document is self-contained, and does not refer directly to any other documents. Further related information on the game of Sudoku, including rules, history, and variations can be found through the wikipedia web page located at:  
<http://en.wikipedia.org/wiki/Sudoku>

Further internet searches will easily find additional game hints and examples.

## **2.0 General Description**

*SSFW* is an entertainment game program for individual PC users. It provides a fun and easy to play alternative to the original paper-based Sudoku game, with additional features which provide hints and solutions to the game grid during game play.

### **2.1 Product Perspective**

*SSFW* is intended to be a single, self-contained, stand-alone product, not related to any other software products. It does not function as a component of any larger system. It is possible that in the future, additional games could be developed which would, where possible, use the same look-and-feel of the final *SSFW* user interface that is developed; however, that future potential is not relevant to current development for the purposes of this document.

This game is intended to be marketed as an inexpensive, easy to use product distributed as a download via a company website (details to be determined at a later time), or via a single CD distribution disk readily available through software vendors, electronics specialty stores, and large-box variety store discount retailers.

### **2.2 Product Function Summary**

*SSFW* is a challenging logic game providing the following major functions:

- 1) Displays Sudoku puzzles on the computer screen for interactive solution by the user using keyboard and mouse inputs.
- 2) Randomly generates virtually any possible Sudoku puzzle, rather than storing a finite set of pre-generated puzzles.
- 3) Allows the user to save any puzzle with a Windows file name, and later reload and continue playing the previously saved puzzle.
- 4) Solves any valid (solvable) Sudoku puzzle which is in play.
- 5) Provides the ability to enter manually the grid values for a new puzzle to be played (such as from a printed puzzle), verifying that the puzzle is solvable (ensuring it has been entered correctly).

### **2.3 User Characteristics**

The *SSFW* user will require only minimal computer literacy skills, and enough familiarity with the Windows operating system to allow the basic skills of mouse usage, keyboard usage, installing a program, starting and exiting a Windows program, using a Windows program main menu, and selecting and naming files using a standard file dialog box for loading and saving game files. Familiarity with the basic rules and play of Sudoku is helpful, but the program will provide a few-sentence summary of the rules for the puzzle via the Help button.

## 2.4 General Constraints

The *SSFW* program imposes minimal constraints upon its design and implementation. There are no known regulatory policies, nor any security- nor encryption-related algorithms which could in any way restrict the program's export to other countries. The program does not interface to any other applications. There are no hardware interfaces, beyond the standard PC keyboard and mouse equipment commonly present. The software has no critical functionality, and there are no health, safety, or security issues to be taken into consideration.

The following are the only constraints specified:

- 1) *SSFW* shall execute on all standard PC platforms running Windows versions XP through Windows 7.
- 2) Hardware requirements shall include a keyboard, full mouse functionality, and a standard SVGA display operating in 600x800 or higher resolution.
- 3) The program shall allow multiple instances of itself to run without interference or any over-writing of program-critical information.
- 4) The program shall be written in a high-order language such as C, C++, C#, or any other language for which a Windows-compliant compiler exists.
- 5) The program shall be distributed as a single EXE file which can be started by double-clicking its icon in standard Windows fashion.

## 2.5 Assumptions and Dependencies

This SRS assumes that the program to be developed:

- 1) Shall fully comply with all stated requirements contained herein
- 2) Shall function with the same reliability and functionality on all the listed operating systems in **Requirement 2.4.1**.

### 3.0 Specific Requirements

The detailed requirements for *SSFW* are contained in this section, broken into the categories of:

- (a) Functional Requirements
- (b) External Interface Requirements
- (c) Use-Case Descriptions
- (d) Performance Requirements
- (e) Design Constraints
- (f) Quality Characteristics
- (g) Other Requirements

#### 3.1. Functional Requirements

The following requirements define the operational capabilities to be provided by *SSFW*. *SSFW* shall meet the following requirements:

- 1) Be installed onto the PC so that it can be started either by double-clicking its desktop icon or by selecting **Start-> Programs -> Super Sudoku** to open the program.
- 2) Provide a main menu with a **File-> New Game** option.
- 3) Prompt for the option to save the game when a New Game is selected, if another game is in progress, and if selected, use a standard Windows File Save dialog to obtain a file name for the user.
- 4) Save and load files having the extension '.SUD'.
- 5) Save and load files in an ASCII format which is readable by the developer in Notepad, in a simple, logical format of the designer's choosing.
- 6) Generate unique, randomly produced games, varying both in number ordering as well as selection of blanked out values.
- 7) When the New Game option is chosen, ask the user to select the difficulty level, then generate a game with one of 3 different difficulty levels, which determine the total number of blank squares that are left for the user to fill in.
- 8) Display the familiar Sudoku 9x9 grid in a Windows screen, with a square (not a distorted rectangle) shape, along with the 81 identical, square input fields.
- 9) Highlight the 9 sub-grids within the 9x9 grid.
- 10) Highlight the fixed, non-editable initial puzzle values within the input squares by setting their backgrounds to a grey shade, easily distinguishable from the editable, white background input squares into which the user will enter values.
- 11) Allow only the following key presses to affect the input squares: **0,1,2,3,4,5,6,7,8,9, <DELETE>, <BACKSPACE>, and <SPACEBAR> keys.** g  
**<DELETE>, <BACKSPACE>, or <SPACEBAR>** will cause the current value in the input square to be blanked out.
- 12) Determine all the allowable legal values which could be placed into the currently focused input square.

- 13) Optionally display the allowable legal values for the current input square in a labeled status at the bottom of the screen display area.
- 14) Allow Hint display to be turned on through a main menu selection labeled **Options-> Always Show Hints** as a toggled menu item which displays a check mark when activated, and activates the continual display of all possible numbers for the currently focused input square in the Hint display area on the screen. This is to be updated immediately when the input square focus changes.
- 15) Greyed out initial value input squares should not be focusable.
- 16) The arrow keys and tab keys will advance the focused input square location to the next editable input square, with wrap-around at the edges of the 9x9 square. Tab will operate as a **Right Arrow** key, but advance to the next lower row when the rightmost edge of the grid is reached, providing the expected wrap-around behavior familiar to users tabbing through input fields.
- 17) The Hint status area will be blanked when the **Always Show Hints** option is de-selected.
- 18) A context popup menu will be associated with the editable input squares, and will be activated by a standard right-click of the mouse.
- 19) The context popup menu will contain a **Show Hint** item. Selecting this item will cause the hint for the current square to be displayed in the Hint bar until a value is entered or the current square is exited, at which time the Hint bar will be blanked.
- 20) Right-clicking an input square will popup a context menu containing the item **Fill In Answer**. Selecting this item will place the correct value into the input square.
- 21) The main menu will have a **File -> Load** option to quit the current game and load a previously saved game into the program. If a game is currently being played, the user will be given the option of saving that game before loading the new one.
- 22) When performing the **Load Game** operation, a standard Windows file dialog box will appear allowing the user to select a previously saved game file from any file location accessible to the machine. Only files having the **'.SUD'** extension will be shown in the dialog as selectable files.
- 23) Saved games will be loaded at the same point of play as it was last saved.
- 24) The main menu will have two options available, **File-> Save Game** and **File -> Save Game Unsolved**.
- 25) Selecting either **Save Game** or **Save Game Unsolved** from the main menu option will popup a **Save Game** dialog box, allowing the user to navigate to any file location accessible to the machine, and select a file name for the current game. The game extension is not selectable by the user, and will always have the extension of **'.SUD'**, appending to the file name if necessary.
- 26) **Save Game** will save the entire current game contents, including the pre-determined non-editable puzzle values as well as the editable user-entered values, as the puzzle currently exists, including erroneous user values if they exist. In other words, the game will be saved at the current point of game play,



27) **Save Game Unsolved** will save only the game's original, unplayed state, with none of the user-entered data retained, only the initial puzzle values. Loading this game will start the play with no user-entered values in the puzzle.

28) A **Solve Now** button will be displayed near the Sudoku grid, enabled only when a puzzle can be solved. For puzzles determined to be invalid, the button will be disabled.

29) Clicking **Solve Now** causes a warning message asking 'Are you sure you want the computer to solve the current game?' will appear. Clicking 'Yes' will cause the completed puzzle to be displayed, while clicking 'No' will continue the current game in progress or game being entered.

30) Puzzles may be entered manually by selecting the main menu **File-> Enter Puzzle** item.

31) If a game is currently in progress when **File -> Enter Puzzle** is selected, the user will be given the opportunity to save it.

32) Selecting **File -> Enter Puzzle** causes a new, completely blank puzzle screen will be presented for input, with all the input squares accessible.

33) The puzzle entry process is finished when the user clicks the **Entry Complete** button, which is visible once only during puzzle entry mode. Clicking it will offer the opportunity to save the puzzle, then lock in the currently entered values as fixed, greying out their input squares so that they are no longer editable.

34) After **Entry Complete** is clicked, the current puzzle is automatically in 'play' mode, and will allow the user to enter values normally in the white input squares, just as if the game had been started through the **New Game** menu option.

35) **Entry Complete** will immediately attempt to solve the entered puzzle without displaying the computed answers, and if, for some reason, such as an incorrect input, the game cannot be solved, a dialog box will popup, noting that there seems to be an error in the game, and asking whether the user you would like to re-enter the **Enter Puzzle** mode to correct any errors. Clicking 'Yes' will return to the puzzle entry process, and display the active **Entry Complete** button. Clicking 'No' will exit the entry mode, leaving the invalid puzzle available for normal play; see requirement 36 below.

36) Whenever it is determined that a puzzle cannot be solved as entered, the message 'This puzzle cannot be solved' will appear in the Status Bar area of the screen, and remain displayed until the puzzle is edited and made solvable, or a new game is started.

37) The main menu shall have an **Options** main menu item.

38) Clicking the **Options** main menu item opens the submenu, allowing the listed options found there to be toggled on or off as desired. When an option is selected as ON, it will display a checkmark next to it; when OFF, no checkmark will appear.

39) Options available under the **Options** menu item are"  
**Always Show Hints** and

### Show Errors

40) Toggling the **Always Show Hints** option selects whether the allowable numbers for the current entry square are constantly displayed or not. See requirement (14) in this section.

41) Toggling the **Show Errors** option ON will cause incorrect input entries to be displayed in red as soon as they are entered, indicating instantly when an error has occurred.

42) A game is considered completed when (a) there are no errors, and (b) all of the input squares contain values, or (c) when a user has selected **File-> Quit Game**. If there are errors when the last blank square has finally been given a value, the message 'There are errors, would you like to see them?' pops up. Clicking 'Yes' will cause the **Show Errors** option to be activated, and the grid display to show all erroneous values in red; clicking 'No' will return the game to normal play. Play will then continue until all errors and all blank squares have been removed.

43) When all the squares have been filled and the **Show Errors** option is already on, the 'Would you like to see them' part of the Errors message (see requirement 42 in this section) is **not** displayed.

44) A main menu **Help** item will be provided, with a standard **About** option to display program identification information (developer and version number), as well as a **Game Rules** option, which will popup a dialog box with essentially the text contained herein at section **1.3 Definitions: Sudoku**.

45) When a game has been successfully completed with no errors and no blanks, a dialog will popup to reveal the total elapsed time in minutes and seconds for solving the game. This total time will not extend beyond the current run of the program and the start of play, so no cumulative time must be retained between runs of the program.

46) Provide shortcut key combinations for all the main menu options, using CTRL- plus the first letter of the labeled menu subitem wherever possible.

### 3.2. External Interface Requirements

SSFW does not interface with any other programs or hardware systems, and utilizes only the standard PC hardware as specified in **Requirement 2.4.2**.

Regarding the human interface:

- 1) Standard windows appearance will be maintained.
- 2) Shortcut keys for menu items shall be provided and listed in the submenu.
- 3) A file **Exit** option, with confirmation by the user that they want to quit the program, shall be provided.
- 4) The interface must be intuitive and easy for a beginning user to operate.

### 3.3. Use-Case Descriptions

There are a small number of different usage scenarios for the *SSFW* program. The user's possible actions are limited to:

- starting a new game
- playing a game
- saving or loading a game
- entering a game manually
- solving a game
- setting game options

Refer to **Appendix A: Preliminary User's Guide** for a description of how these various functions will be accessed from the user's perspective.

### 3.4. Performance Requirements

Performance requirements for this program are not critical, but must fall within the guidelines specified below:

- 1) The program will load and save game files within 10 seconds after the user has selected a file name.
- 2) The program will solve any Sudoku puzzle that is currently in play mode within 25 seconds.
- 3) The program will update the display of hints and erroneous entries within 3 seconds of focusing on an input square.

### 3.5. Design Constraints

The design of this program must conform to the following constraints.

- 1) Stored game files will be stored in an ASCII text format which can be read by a simple Notepad program during development, so that file contents can be validated during testing.
- 2) The design will be object-oriented as completely as is practical.

### 3.6. Quality Characteristics

The *SSFW* program will be designed and implemented according to the following quality standards"

- 1) Correctness- the program will solve any correctly defined Sudoku puzzle, and identify as 'unsolvable' any puzzles which are incorrectly specified. This is critical to the credibility of the program from the user's perspective, and hence to its desirability as a tool for solving any Sudoku puzzle, as it will be advertised. This characteristic will be verified through testing of the solution algorithm during development, and through testing the final program by solving several highest-level difficulty puzzles from outside sources.
- 2) Maintainability- the program will be designed to be easily repairable should errors be found, so that individual modules or objects demonstrate a high

degree of cohesion and decoupling according to software engineering standards. This will be verified through design reviews to demonstrate that the design is logical and minimizes the possibility of ripple-effects should changes be required.

3) Usability- the program will utilize a simple, clear, easily understood user interface, with a feel that is intuitive to typical Sudoku players having experience with having solved a few Sudoku games previously. The level of effort to learn the game will be measured by having new users who have played Sudoku start the program, and attempt to solve a generated puzzle, and asking them to evaluate the degree of difficulty they had in using the program.

### **3.7. Other Requirements**

No other requirements have been identified.

#### 4. Supporting Information

The attached **Appendix A** section specifies and clarifies additional requirements by describing the program functionality from the user's point of view.

#### **Appendix A: Preliminary User's Guide**

*Super Sudoku* is a fun, easy-to-use version of the popular Sudoku game. It can generate millions of different puzzles, and uses simple, intuitive controls for nearly endless Sudoku fun! Please read through this guide thoroughly. By following the instructions below, you will be able to take full advantage of the various features that have been included.

##### **1. Start the program:**

*Super Sudoku* is started by double-clicking on its desktop icon, or by opening the Windows **Start-> Programs -> Super Sudoku** menu to open the program.

##### **2. Start a new game:**

Select the **File-> New Game** option from the main menu. If another game is in progress, you will be asked if you would like to save the current game, and if so, asked to enter the file name under which the current game will be saved. *Super Sudoku* can generate millions of unique games so that it is likely that no two games are identical. You will be prompted to select the desired game difficulty level: easy, medium, or difficult, and a puzzle of that level will be generated and displayed.

When the new game is started, you will be presented with the familiar 9x9 Sudoku number grid. The game squares containing the initial game numbers, which cannot be changed, will be highlighted with a greyed background.

##### **3. Entering values into the game:**

The game input squares where you will enter numbers to play the game will always have a white background, and will allow you to enter the value of your choice. Squares having a greyed background represent the fixed puzzle values which are part of the original puzzle, and cannot be changed. Only the **0,1,2,3,4,5,6,7,8,9, <DELETE>, <BACKSPACE>, and <SPACEBAR> keys** will have any effect on the input squares; pressing any other key will not affect the game. Pressing **<DELETE>, <BACKSPACE>, or <SPACEBAR>** will cause the current input square to be blanked out.

##### **4. Receiving hints:**

In order to make the game play easier, a 'hint' option allows all the possible legal numbers which could be entered into the current input square to be displayed in the 'Hint bar' at the bottom of the screen. At any time during the game, there are two different ways to activate the **Hint** option.

1) Turn on hints for all squares: The menu selection under the **Options-> Always Show Hints** menu item will toggle this feature on and off; when a

checkmark is displayed next to the item, the Hint bar will always display all possible numbers for the currently focused input square. Clicking on a different input square will immediately show all possible values for that square. Re-selecting the menu item will erase the checkmark, turn off the hint feature, and blank the Hint bar.

2) See a hint for the current square: Right-clicking an input square will popup a context menu containing the item **Show Hint**. Selecting this item will cause the hint for the current square to be displayed in the Hint bar until a value is entered or the current square is exited, at which time the Hint bar will be blanked.

3) See the correct answer for the current square: Right-clicking an input square will popup a context menu containing the item **Fill In Answer**. Selecting this item will place the correct value into the input square.

## 5. Load or Save a game:

Select the main menu **File -> Load** option to quit the current game and load a previously saved game into the program. If a game is currently being played, you will be given the option of saving that game before loading the new one. A **Load Game** file dialog box will appear allowing you to select a previously saved game file from any file location accessible to your machine, and the game will be loaded at the same point of play as it was last saved.

Selecting the **File-> Save Game** or **Save Game Unsolved** main menu option will popup a **Save Game** dialog box, allowing you to navigate to any file location accessible to your machine, and select a file name for the current game. If **Save Game** is selected, the game will be saved at the current point of game play, with your existing entered values saved as well as the initial puzzle values. Optionally, if **Save Game Unsolved** was selected, the game is saved in its original, unplayed state, with none of the user-entered data retained, so that the game can be played fresh from the start.

## 6. Solve a Puzzle:

To solve a game at any point, the **Solve Now** button can be clicked in the upper left corner of the screen. A warning message asking 'Are you sure you want the computer to solve the current game?' will appear. Clicking 'Yes' will cause the completed puzzle to be displayed, while clicking 'No' will continue the game in progress or being entered (see 7. Entering a Puzzle).

## 7. Entering a Puzzle:

*Super Sudoku* can be used to play printed Sudoku puzzles from other sources by allowing you to manually enter a puzzle in its unsolved state. From the main menu, select **File-> Enter Puzzle**. If a game is currently in progress, you will be given the opportunity to save it. Then, a new, completely blank puzzle screen will be presented for input, with all the input squares accessible. When all of the puzzle values have been entered correctly, click the **Entry Complete** button,

which will offer the opportunity to save the puzzle, then lock in the currently entered values as fixed, greying out their input squares so that they are no longer editable. At this point the puzzle is automatically in 'play' mode, and will allow you to enter values normally in the white input squares, just as if the game had been started through the **New Game** menu option. If, for some reason, such as an incorrect input, the game cannot be solved, clicking **Entry Complete** will cause a dialog box to popup, noting that there seems to be an error in the game, and asking whether you would like to re-enter the **Enter Puzzle** mode to correct any errors. Clicking 'Yes' will return to the puzzle entry process, clicking 'No' will exit the entry mode, leaving the invalid puzzle available for normal play, although the message 'This puzzle cannot be solved' will appear in the Status Bar area of the screen.

## 8. Setting Options:

The game options are set by clicking the **Options** main menu item, and toggling the listed options found there on or off as desired. When an option is selected as ON, it will display a checkmark next to it; when OFF, no checkmark will appear.

The options available under this menu item are:

**Always Show Hints-** Selects whether the allowable numbers for the current entry square are constantly displayed or not. See **4. Receiving hints** above for a complete explanation.

**Show Errors-** When checked, this will cause incorrect entries to be displayed in red as soon as they are entered, indicating instantly when an error has occurred.

## 9. Finishing a Game:

A game ends when there are no errors, and all of the input squares contain values. If there are errors when the last blank square has finally been given a value, the message 'There are errors, would you like to see them?' pops up. Clicking 'Yes' will cause the **Show Errors** option to be activated, and the grid display to show all erroneous values in red; clicking 'No' will return the game to normal play. Note that if the **Show Errors** option is already on, the 'Would you like to see them' part of the Errors message is **not** displayed. Play will then continue until all errors and all blank squares have been removed.