

Concordia University  
Department of Computer Science and Software Engineering  
SOEN 341 — Software Process  
Winter 2007 — Section S  
Project Deliverable 3 Description

## Third Incremental Code Build

You have to deliver and demonstrate some code that implements a second version for each of the three programs — Game Player, Game Advisor, and Game Generator — associated with Sudoku on a 9-by-9 grid.

The rules of the game are simple: (1) Each row must contain precisely the numbers 1 to 9 without repetition. (2) Each column must contain precisely the numbers 1 to 9 without repetition. (3) Each 3-by-3 subgrid must contain precisely the numbers 1 to 9 without repetition.

The Game Player must allow a user to play a game by entering numbers into squares. The Game Advisor should give a hint as to what number to enter in a square in any given situation. The Game Generator must return a random game of the required difficulty.

The third version integrates the three programs, and adds any additional features that you planned to add to each program. For example, the Game Player should allow a user to make two child copies of the current state of the game, and be able to play each child game individually. For example, the Game Advisor should implement more heuristics such as the “pigeonhole principle”. In Sudoku the pigeonhole principle arises when a collection of  $n$  cells (in the same row, column or subgrid) can only hold  $n$  different numbers from 0..9. Then these  $n$  numbers must occupy those cells, and cannot occur in any other cell (of the same row, column or subgrid). Usually the pigeonhole principle is applied for  $n=2$ , but cases when  $n=3$  do arise too. For example, the Game Generator should implement a backtracking approach to completing a give Sudoku game: see Wikipedia “Algorithmics of Sudoku”.

Come prepared to the presentation. Construct various and appropriate test cases that will demonstrate that your code is effectively achieving its duty. You have to proceed with a demonstration of your build. The procedure for the reservation of a time slot for your demonstration will be available on the web page. The goal of the demonstration is to effectively demonstrate that you have three programs as described above. The grading scheme of the demonstration is as follows:

Effectiveness and level of preparation of the demonstration	/2
Effective and complete demonstration that the 3 programs are in fact working	/4
Compliance with the above mentioned design constraints	/4
Total	/10

## Project Testing and Delivery Document

You have to deliver this document following the template provided on the course web page (the Wiki web page, same as for Fall 2006). A detailed grading scheme and instructions are provided in the template.

## **Evaluation**

As stated in the course outline, the code build (1) is worth 10% and the document (2) is worth 5%, for a total of 15% of the final numeric grade.

## **Assignment Submission**

All project assignments are to be handed in using the ENCS Electronic Assignment Submission system. A link to this system is available on the course web page. On the day of your build demonstration, you have to submit a zip file containing your document (see section 2 above), as well as your source code for this build. It has to be submitted by midnight on the due date of the assignment.