



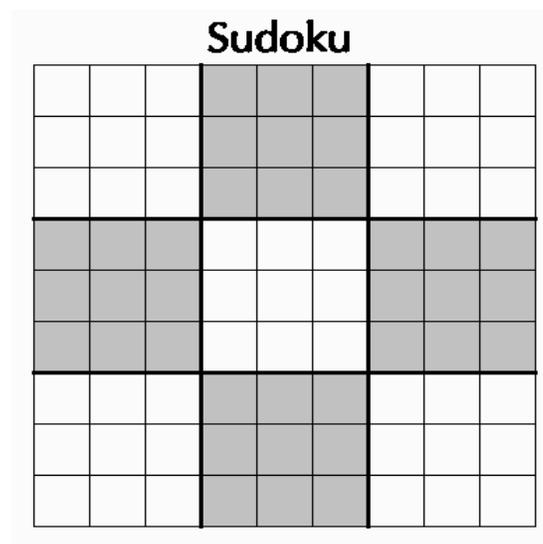
## Homework 11: Building With Functions

**Goal:** The goal of this assignment is to build a Graphic User Interface (GUI, pronounced GOO-ey) for the game of Sudoku by building layers of functions. [You don't need to know anything about the game to do the assignment.]

In Lecture 11, functions were used to define more functions and so on to build a complex program to implement a timer; the Sudoku game is built the same way. All you will build is the picture of the game and its mouse click operation.

### **Sudoku GUI**

The Sudoku board is a 9×9 array of cells. Alternating 3×3 cells are colored gray and strong lines separate all of the 3×3 cells, as follows. (More information at <http://www.websudoku.com/>.)



Function layering can be used to create this GUI as follows.

**Assignment Part 1. Cell Array.** Begin with a function

```
void cell(int x, int y, int s, color tinto)
```

where  $x$  and  $y$  are the positions of the upper left corner of a box,  $s$  is its size and  $tinto$  gives the color of the cell. The function simply draws a box filled with the proper color.

Using `cell( )`, you can build a function

```
void triple(int x, int y, int s, color tinto)
```

where the parameters have the same meaning as before, though now  $x, y$  is the position of the upper left corner of three boxes in a row. Use a **for** loop in this function.

Continue in this way, next building a 3×3 array of cells. The goal is the 9×9 array of cells created by the function `cellarray(x, y, s)`. [Hint: A good set of functions are: `cell( )`, `triple( )`, `block( )`, `row( )`, `cellarray( )`.]

**Assignment Part 2. Sudoku Board.** Write a function to draw the four thick black lines, given the position of the upper left corner of the cell array and the size of the cells as parameters. Package this function and the function from Part 1, into a function called `board` that has the position and size parameters.

**Assignment Part 3. Identify Cell.** We want to move the mouse pointer over the Sudoku board, and identify which cell it is over when the user clicks the mouse. When the click comes, we know the following information:

`mouseX`, `mouseY` the position of the mouse  
`x`, `y` upper left corner of the board, which is where we draw it  
`s` the size of the cells

and from these values you can figure out `cell_x`, `cell_y`, the position of the cell the user clicked on. For example, clicking on the center cell of the top row of squares should produce 4 for `cell_x` and 0 for `cell_y`.

Write a `mousePressed( )` function that figures out the values of `cell_x`, `cell_y`. These are the cell positions (0 through 8) that the mouse is over in the x- and y-directions. Someone else will need these later to write code to play the game of Sudoku. To prove that we've got this right, call your `cellarray( )` function from `setup( )` and then make your `draw( )` function be:

```
void draw ( ) {
  if (mousePressed) {
    fill(255,0,0);
    ellipse(x+s*cell_x+s/2, y+s*cell_y+s/2, s/4, s/4);
  }
}
```

which draws a red dot in the cell the user clicked in.

**Wrap Up:** You used functions to build a Sudoku GUI and included the function that identifies the cell that the mouse is over.

**Turn in:** Submit a copy of our `.pde` file renamed to `<yourname>.pde` to the course drop box.

