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 (define a function by calling other functions). You don’t need to know anything about the game to do the assignment. 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.

Define and initialize your global variables x_global, y_global and s_global, where x_global and y_global are positions of the upper left corner of the whole Sudoku board, s_global is the size of each cell.

We begin by drawing one cell of the Sudoku board. Define a function:

```c
void cell(int x, int y, int s, color tinto) {
    fill(tinto);
    rect(x, y, s, s);
}
```

where x and y are the positions of the upper left corner of a cell, s is its size and tinto gives the color of the cell. The function simply draws a cell filled with the proper color. You can test whether this function is working properly by making a function call cell(x_global, y_global, s_global, color(255)); in your setup() function.

Using cell( ), you can build a function
void triple(int x, int y, int s, color tinto) {
    fill(tinto);
    for (int i=0;i<3;i++) {
        cell(x+i*s, y, s, tinto);
    }
}

where the parameters have the same meaning as before, though now \(x, y\) is the position of the upper left corner of three cells in a row. Notice that a \texttt{for} loop is used in this function. Refer to the textbook to find out how a \texttt{for} loop works. Similarly, you can test whether this function is working properly by making a function call \texttt{triple(x\_global, y\_global, s\_global, color(255))}; in your \texttt{setup()} function.

Continue in this way, next building a \(3\times 3\) array of cells. The goal is the \(9\times 9\) array of cells created by the function \texttt{cellarray(x, y, s)}. A good set of functions are: \texttt{cell()}, \texttt{triple()}, \texttt{block()}, \texttt{row()}, \texttt{cellarray()} as shown in the following table. In function \texttt{row()}, you might need to refer to how you changed the color of Blinky in Homework 6.

<table>
<thead>
<tr>
<th>Function</th>
<th>What it draws</th>
</tr>
</thead>
<tbody>
<tr>
<td>void cell(int x, int y, int s, color tinto){...}</td>
<td></td>
</tr>
<tr>
<td>void triple(int x, int y, int s, color tinto) {...}</td>
<td></td>
</tr>
<tr>
<td>void block(int x, int y, int s, color tinto){...}</td>
<td></td>
</tr>
<tr>
<td>void row(int x, int y, int s){...}</td>
<td></td>
</tr>
<tr>
<td>void cellarray(int x, int y, int s){...}</td>
<td></td>
</tr>
</tbody>
</table>

**Assignment Part 2. Sudoku Board.**

Write a function, \texttt{drawLines()} to draw the \textbf{four thick black lines}, given the position of the upper left corner of the cell array and the size of the cells as \textbf{parameters}. Call this function and the function \texttt{cellarray()} from Part 1, from a function called “\texttt{board}” that has the position and size as \textbf{parameters}. You should now have a complete board that matches the image at the beginning of this spec.

**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 (system variables)
- `x_global, y_global` upper left corner of the board, which is where we draw it
- `s_global` the size of the cells

And from these values you can figure out `cell_x, cell_y`, the indicies 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` (the 4th cell along x coordinate and the 1st cell along y coordinate). One function to get the values of `cell_x, cell_y` is as follow.

```java
void mousePressed() {
    cell_x=(mouseX-x_global)/s_global;
    cell_y=(mouseY-y_global)/s_global;
}
```

Remember that `cell_x, cell_y` 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 demonstrate that we’ve got this right, call your board( ) function from setup( ) and then make your draw( ) function be:

```java
void draw() {
    if (mousePressed) {
        fill(255, 0, 0);
        ellipse(x_global+s_global*cell_x+s_global/2,
                y_global+s_global*cell_y+s_global/2,
                s_global/4, s_global/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 clicks on.

Turn in: Submit a copy of our .pde file renamed to hw9.pde to eCommons through the “Assignments” link on the left.