Timers, Loops, Lists

This assignment covers the various techniques of repetition in Javascript: loops, timers, lists.

Working Directory Do your work in a directory called hw11 in your website.

Starting Files Change into the hw11 directory and get the starting files for this assignment by typing:

```
type: cp /comp/10IDI/files/hw11/* .
```

Problem → Procedure → Code: Do not just start writing code. These problems are not tricky, but they do require some thought and planning. Start by writing the HTML to design the layout of the elements. Then, write out in English an outline of the algorithm. It is an excellent idea to put this outline in your directory for reference and even as comments at the start of your Javascript code.

Problem 1: Reminder

This problem builds on the countup.html program shown in class, with code included on the class handout, and code also in the Code section of that class on the website. The purpose of this program is to remind you of some event after a specified number of seconds. To draw your attention to the reminder, the table flashes alternately between red and green.

Write a page called reminder.html that has a table that looks like:

```
  -------------------
  | REMINDER       |
  | START | STOP |
  | Time Left: 12  |
  -------------------
```

When the user clicks the START button, the program uses a prompt pop-up to ask the user "Display what message?", then uses a prompt pop-up to ask the user "After how many seconds?". The program then fills in the text box next to 'Time Left' with the initial number of seconds then begins counting down to zero.

When the counter reaches zero, the program puts the reminder message in the text box at the bottom of the table and begins flashing the table background color between red and green, changing once each second.

If the user presses the STOP button, the counter stops counting or the box stops flashing, depending on what the program is doing at that time.

HINTS: To flash the table, give the table an id attribute. Then you can use getElementById() to select the table, and then set the background color component of its style to 'red' or to 'green', changing each second. The countup.html code is a place to start.

Problem 2: Elephant March

Imagine a row of elephants marching across a plain.

Five across, they march steadily in pursuit of food, friends, adventure. How can we use our knowledge of repetition in Javascript and arrays in HTML to simulate this dramatic action?

Animation Basics

When something moves, we see it at one spot, then a little while later, we see it at another spot, and then a little bit later, we see it at yet another spot. To produce animation on the computer, we just need to know how to draw something at one spot, then how to draw it somewhere else and undraw it from its previous spot. Continuing in this manner, we can simulate motion. For this lab, you will build up to creating a row of marching elephants. This technique can even prove helpful for a homework assignment.

The Idea

The webpage has a table of 5 rows, each with 9 columns. This grid of 45 images is a list of 45 img tags organized into an HTML table. The cells are numbered so:

```
<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>44</td>
</tr>
</tbody>
</table>
```

The numbering starts at 0 and goes to 44. The first two rows of cell numbers are shown on the diagram.
Part 2.0: Experiment with the Controls

The page, jumbowalk.html, has some buttons and functions already included. Try the following sequence of actions:

a. Put an elephant at cell 27
b. Put an elephant at cell 28
c. Clear cell 27
d. Put an elephant at cell 29
e. Clear cell 28

Note: In this page, there are the table has a background image of the entire scene. The file blank.gif is transparent; it fills the cell but, like a piece of glass, allows the underlying image to show through.

Part 2.1: Manual Automation

Once you get a sense for how animation works as drawing the next instance and clearing the previous instance, it is time to write code for that procedure. The page already has buttons for moving the elephant left and right, but they do not do anything yet.

(a) Write the move_elephant() function. This function takes an argument that tells how many cells to move.

First: Write your procedure in outline form. Write on paper, or type the procedure into a comment at the start of your html file.

The function must move the elephant that many cells and erase the previous elephant image. HINT: You need a global variable to keep track of where the elephant is currently displayed. Luckily, the file already has a variable defined.

(b) Make sure your function works at the upper left and lower right corners of the grid. When the elephant moves forward from the lower right, it should appear in the upper left, and when it moves backwards from the upper left, it should appear at the lower right.

Part 2.2: Timed Automation

Elephants do not need a button-push to move. Once started, they can keep moving on their own. The walk button calls a function that does not yet exist.

(c) Write the walk_elephant() function. This function must work as follows: When the button is pressed, the elephant moves one cell forward each second. If the user presses the ← button, the elephant moves backwards one cell each second. If the user presses the → button, the elephant starts moving forward. HINT: You will need a global variable that keeps track of whether the elephant is moving forward or backward. Each time you press an arrow button, this variable needs to be set to the correct value. What values should you use to indicate forward and backward? Why?

(d) Add a stop button that causes the elephant to stop moving.

Part 2.3: Columns of Elephants

Now that you can place one elephant and get it to walk across the plain, you can extend the program to create a column of elephants crossing the plain.

(e) Write a new button marked place herd that asks for a cell number in the range 0..8. The program then draws a column of elephants extending downward from that cell. For example, if the user specifies cell 2, the program draws five elephants at cells 2, 11, 20, etc.

(f) Write new buttons for Herd controls that have two arrow buttons and a walk and stop button. These buttons should do for the herd (of five elephants) what the buttons do for the single elephant.

(g) Have your herd stop walking when the herd hits the right side or the left side of the screen.

Part 2.4: No Grid

For super-realism, edit the style section for td that specifies a border-width of 1px. Change that to 0px. Then reload the page. Amazing?

Part 2.5: Rotation

Consider a bunch of people playing musical chairs. The people might be seated as:

Ann  Barry  Carol  Drew  Ellen

The music plays; they rotate one seat to the right:

Ellen  Ann  Barry  Carol  Drew

The basic idea is to do something like:

document.images[43].src = document.images[42].src
...

but there is one small wrinkle: what happens to the image in cell 44? The solution is not too tricky.

Add a new button marked rotate right that rotates all 45 cells right one place and takes the last cell and rotates it into the first cell. Then put a single elephant or a herd of elephants, or even a bunch of elephants and herds on the plain. Then press rotate.