TIMER ANIMATION

Preparatory Remarks

- Make a new directory in your class projects directory for this homework assignment.
- Download the starting template code (H06 Starting Template) from the session resources.
- The layout template, directory structure, libraries, and starting code will all be familiar to you.

In this assignment, we will be animating a ball that bounces around in it's window, leaving a trail behind as it moves.

Assignment Instructions

1. Create a black rectangle that fits on the Raphael paper.
2. Use Raphael to create a green disk (circle) at the center of the paper with a radius of 20, and assign it to a variable named 'disk'.
3. Make a function named 'draw' that keeps track of how many times it's called and prints the count to the console each time.
    - Hint: you'll need to remember 'count' outside of the 'draw' function
4. Use setInterval to call draw once per second.
    - Print something out to the console window to make sure it is operating properly.
5. Add properties to disk object to keep track of its xpos and ypos (initialized to the point at the center of the paper).
    - Hint: Just use the JavaScript object 'dot' notation to name a property and assign a value to it - the property will be created if it does not already exist.
6. Add properties xrate and yrate properties to disk. We'll use these numbers to update the disk position each time we draw it. Notice that if these rates were negative, it would just mean moving in the opposite direction.
    - Initialize xrate and yrate to 10.
7. Update disk.xpos and disk.ypos by adding disk.xrate and disk.yrate to them each time in 'draw'. That is, the disk.xrate and disk.yrate numbers represent the number of pixels in each dimension that we will move the disk each time draw() is called.
    - Print the disk.xpos and disk.ypos values to the console to see if they make sense and are updating on each call to draw as you expect.
    - Note: xpos, ypos, xrate, and yrate are just numbers we use for computation - they don't actually move the graphical object on the webpage.

Assignment instruction continues

TIMER ANIMATION

Assignment Instructions Continues

8. Use disk.xpos and disk.ypos to update the Raphael attribute values cx and cy (which actually determine where the Raphael graphical objects are drawn) in the 'draw' function.
    - Load the page in the browser. Is the disk moving as you'd expect?
    - What happens if you wait for long enough?
    - How would you fix the problem?
    - What are the 'if' conditions that you could check for to catch the 'problem'?
        - Hint: there are 4 of them
9. Check each of the 4 conditions in your draw routine, and reverse the direction (change the appropriate rate variable) of disks when they hit walls.
    - Now what does your animation do?
10. Next, make your animation run smoother/faster.
    - Hint: there are two separate numbers in your code for doing this.
11. Create a new white circle *inside* the draw routine with a var statement at the beginning of draw, and name the variable "nd" (short for "new disk"). Give it the same xpos, ypos, xrate, and yrate, properties you gave to the green disk. You can initialize its position to anywhere.
    - Change the one line of code that updates the cx and cy attrs of your green disk to instead update the cx and cy attrs of your new white circle, nd.
    - Run it. Can you explain what is happening? Important to understand: what difference does it make where we create the circle- inside or outside the draw function?
    - What happens if you let it run a long time? Why? How might we fix that (besides reloading and restarting the program)?
    - Change the size of your browser window and then reload to see different patterns.
12. Following the call to set the cx and cy attrs in 'draw', make a call to animate the fill of your circle to some color over 1 second in a linear fashion. Now your animated disk with "leave trails" behind as it moves.
    - To solve the problem encountered in 11b above, check out the optional last argument to Raphael's animate function to remove the circle from the paper - use an anonymous function as the argument. Can you see why removing the object helps?
    - Are we tripping yet?