

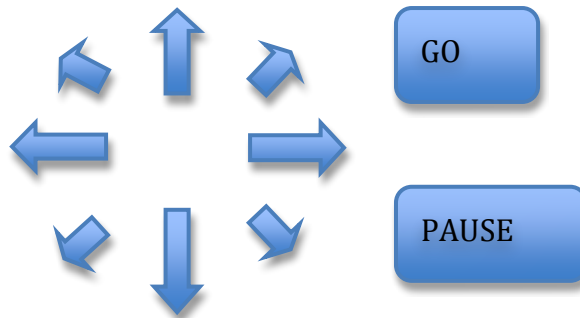
Distributed: 1 February 2011

Due: 15 February 2011

Specification

Position a sprite at a random location on the screen.

Position six buttons as follows (you can change the look as you wish, e.g., plain rounded corner buttons with appropriate labels are fine).



The buttons are used to control the direction of motion of the sprite. The meaning of the buttons are obvious.

The button “GO” will turn on an interval timer. The button “STOP” will turn off the interval timer.

The sprite has an initial velocity of +5 pixels per timer event in the x direction and in the y direction ($dx = +5$, $dy = +5$). The velocity is changed by clicking on a directional button.

Directional buttons



$dx = 0$; $dy = -5$



$dx = +5$; $dy = -5$;



$dx = +5$; $dy = 0$;



$dx = +5; dy = +5;$



$dx = 0; dy = +5;$



$dx = -5; dy = +5;$



$dx = -5; dy = 0;$



$dx = -5; dy = -5;$

The sprite can occupy the same screen location as the buttons.

Hitting a wall When a sprite hits a wall, do wrap-around so that the sprite appears at the opposite wall continuing with unchanged direction. The size of the sprite will be important here so that things don't look 'unnatural'.

Sprite speed is controlled through a combination of (dx, dy) and the rate the interval timer goes off. Choose an interval timer rate so that the sprite moves at a stately, but not lazy, speed.

Turn in:

Hard copy of all code (html, css, javascript)
Screen shot
Demo

Grade based on:

Meeting spec
Satisfying coding standards
Quality of demo
Elegance

Suggestion for code development:

- (1) Position sprite
- (2) Code go
- (3) Modify (2) to code stop
- (4) Code up a single directional change
- (5) Modify code for (4) to all directions.