**Project Assignment: HTML Canvas Web Animation**

**Course:** CSW 38 - Introduction to HTML Canvas Web Animation

**Instructor:** Professor Pacheco

**Project:** Interactive Bouncing Particle System

**Due Dates:** \* **Stage 1:** Due Oct. 27 (10 pts.) (extra credit 5pts, use separate files, css, js, and htm)

* **Stage 2:** Due Nov. 17 (15 pts.) (extra credit 5pts, Add 2 page of content, your choice, continue using separate files. Each page should have a link to the other)
* **Stage 3:** Due Dec. 5 (20 pts.) (extra credit 5pts, add different animation to 2nd page of content, your choice, continue using separate files. Each page should have a link to the other)
* **FYI: Extra Credit 2nd page** also needs to be responsive and have same type of color scheme and fonts as main assignment.

**Project Overview**

Welcome to the CSW 38 Course Project! Over the remainder of the course, you will build an interactive particle animation from scratch using the HTML <canvas> element and JavaScript. This project is divided into three stages, each building upon the last. You will start by creating a single animated particle, then build a system of multiple particles, and finally add user interaction. All files are to be PUBLISHED to a web host for online access.

This project will solidify your understanding of core animation concepts, including the animation loop, object-oriented programming in JavaScript, and event handling.

**Stage 1: The Lone Particle – Due Oct. 27 (10 pts.)**

**Objective:** Create a single particle (a circle or other object) that moves across the canvas and bounces off the edges.

**Requirements:**

1. **Canvas Setup:** Create an HTML file with a <canvas> element that fills most of the browser window.
2. **Particle Object:** In your JavaScript, create a Particle object or class. This object should have properties for:
   * x and y position
   * radius
   * color
   * dx and dy for velocity (speed in the x and y directions)
3. **Draw the Particle:** Create a function that draws the particle on the canvas at its current x and y position.
4. **Animation Loop:** Implement an animation loop (requestAnimationFrame) that does the following on each frame:
   * Clears the entire canvas.
   * Updates the particle's x and y position based on its velocity (dx, dy).
   * Draws the particle at its new position.
5. **Edge Detection:** Add logic to check if the particle has hit any of the four edges of the canvas. If it has, reverse the corresponding velocity (dx or dy) to make it "bounce."

**What to Submit:** Your single HTML file containing all your HTML, CSS, and JavaScript code.  
 Name the file in the following format, First Iniital.Last Name.part\_1.htm (i.e. T.Pacheco.part\_1.htm) Submit the URL to your published HTML file. Use Canvas Assignment link.

**Stage 2: A Swarm of Particles - Due Nov. 17 (15 pts.)**

**Objective:** Expand your program to create and animate a collection of particles, each with its own unique properties.

**Requirements:**

1. **Particle Array:** Create an array to hold multiple Particle objects.
2. **Particle Generation:** Write a function that creates a specified number of particles (e.g., 100) and adds them to your array.
3. **Randomization:** When creating each particle, randomize its initial properties:
   * Starting x and y position (within the canvas bounds).
   * radius (within a reasonable range, e.g., 5 to 15 pixels).
   * color.
   * dx and dy velocity (can be positive or negative).
4. **Update the Animation Loop:** Modify your animation loop to iterate through the entire particle array. In each frame, it should update the position of *every* particle and draw it to the canvas. The edge detection logic from Stage 1 should apply to all particles.

**What to Submit:** Your updated HTML file- named Same format as before.part\_2.htm. Submit the URL to your published HTML file. Use Canvas Assignment link – part 2.  
**Note**: If creating 2nd page for E.C., be sure each page contains a link to the other.

**Stage 3: User Interaction and Polish - Due Dec. 5 (20 pts.)**

**Objective:** Make the particle system interactive by having the particles react to the user's mouse.

**Requirements:**

1. **Mouse Tracking:** Add a mouse event listener (mousemove) to the canvas to track the x and y coordinates of the user's cursor. Store these coordinates in a global mouse object.
2. **Interaction Logic:** In your Particle object's update function, add logic to check the distance between the particle and the mouse cursor.
3. **Enlarge on Proximity:** If a particle is within a certain distance of the mouse (e.g., 50 pixels), make it grow larger (increase its radius). If it's outside that distance, it should return to its original size.
   * *Hint:* You may want to store the particle's original radius in a separate property so it can shrink back.
4. **Smooth Sizing (Optional but Recommended):** Instead of instantly changing the radius, try to animate the growth and shrinking smoothly for a more polished effect.
5. **Window Resizing:** Add an event listener for the browser resize event. When the window is resized, update the canvas's width and height attributes to match the new window dimensions. This will prevent your animation from becoming distorted or clipped.

**What to Submit:** Your final, completed HTML file. – named, same format as before.part\_3.htm. Submit the URL to your published HTML file. Use Canvas Assignment link – part 3.

Good luck, and have fun!