const object = document.getElementById('inflating-object');

const button = document.getElementById('hold-button');

// Game parameters

const MAX\_TIME\_MS = 3000; // Object pops after 3 seconds of continuous inflation

const INITIAL\_SIZE = 50; // In pixels

const MAX\_SIZE = 200; // In pixels

let startTime = 0; // Tracks when the button was pressed

let inflationInterval; // Used to continuously update the object's size

/\*\*

\* START INFLATION - Triggered on mousedown/touchstart

\*/

function startInflation() {

// Prevent starting if it's already popped (optional: for restarting the game)

if (object.classList.contains('popped')) return;

startTime = Date.now();

// Start a continuous update loop

inflationInterval = setInterval(() => {

const elapsedTime = Date.now() - startTime;

if (elapsedTime >= MAX\_TIME\_MS) {

// Predetermined pop time reached!

popObject();

stopInflation(); // Stop the interval

} else {

// Calculate a new size based on the elapsed time

const progress = elapsedTime / MAX\_TIME\_MS; // Value from 0.0 to 1.0

const newSize = INITIAL\_SIZE + (MAX\_SIZE - INITIAL\_SIZE) \* progress;

// Apply the new size

object.style.width = `${newSize}px`;

object.style.height = `${newSize}px`;

}

}, 50); // Update the size every 50 milliseconds (for smooth animation)

button.textContent = 'INFLATING...';

}

/\*\*

\* STOP INFLATION - Triggered on mouseup/touchend

\*/

function stopInflation() {

clearInterval(inflationInterval); // Stop the size update loop

startTime = 0;

// Resume text only if it didn't just pop

if (!object.classList.contains('popped')) {

button.textContent = 'HOLD TO INFLATE';

}

}

/\*\*

\* POP OBJECT - Triggered when MAX\_TIME\_MS is reached

\*/

function popObject() {

// Add the 'popped' class for the visual effect

object.classList.add('popped');

button.textContent = 'POP! Start Over?';

// Optional: Reset the object after a short delay

setTimeout(resetGame, 1000);

}

function resetGame() {

// Remove the pop effect class and reset the size

object.classList.remove('popped');

object.style.width = `${INITIAL\_SIZE}px`;

object.style.height = `${INITIAL\_SIZE}px`;

button.textContent = 'HOLD TO INFLATE';

}

// --- EVENT LISTENERS (Handles both mouse and touch for phone/web) ---

// Start Inflation

button.addEventListener('mousedown', startInflation);

button.addEventListener('touchstart', startInflation);

// Stop Inflation

button.addEventListener('mouseup', stopInflation);

button.addEventListener('touchend', stopInflation);

// Handle the case where the mouse leaves the button while held

button.addEventListener('mouseleave', stopInflation);