VTL25s4.warmer Spin the wheel

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Vocabulary Technique Spinner</title>

<!-- Tailwind CSS CDN -->

<script src="https://cdn.tailwindcss.com"></script>

<!-- Google Font - Inter -->

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600;700&display=swap" rel="stylesheet">

<style>

body {

font-family: 'Inter', sans-serif;

background-color: #f3f4f6;

display: flex;

justify-content: center;

align-items: center;

min-height: 100vh;

}

.wheel {

width: 25rem;

height: 25rem;

border-radius: 50%;

border: 8px solid #4b5563;

position: relative;

overflow: hidden;

transition: transform 5s cubic-bezier(0.25, 0.1, 0.25, 1);

}

.wheel-section {

position: absolute;

top: 0;

right: 0;

bottom: 0;

left: 0;

transform-origin: 50% 50%;

display: flex;

justify-content: center;

align-items: center;

font-weight: bold;

font-size: 1.125rem;

text-align: center;

padding-top: 1rem;

color: #1f2937;

clip-path: polygon(50% 50%, 100% 0, 100% 50%, 50% 100%);

}

.marker {

width: 0;

height: 0;

border-left: 15px solid transparent;

border-right: 15px solid transparent;

border-top: 25px solid #ef4444;

position: absolute;

top: 0;

left: 50%;

transform: translateX(-50%);

z-index: 10;

}

.center-circle {

position: absolute;

width: 6rem;

height: 6rem;

background-color: #f3f4f6;

border-radius: 50%;

border: 6px solid #4b5563;

display: flex;

justify-content: center;

align-items: center;

z-index: 5;

}

</style>

</head>

<body class="p-4 sm:p-8">

<!-- Game Container -->

<div class="w-full max-w-4xl bg-white rounded-3xl shadow-2xl p-6 sm:p-8 md:p-10 transform transition-all duration-300">

<!-- Game Title -->

<h1 class="text-3xl sm:text-4xl font-extrabold text-center text-purple-800 mb-6">

<span class="text-yellow-500">✨</span> The Vocab Spinner <span class="text-yellow-500">✨</span>

</h1>

<p class="text-center text-gray-600 mb-8 font-semibold">Spin the wheel to get a vocabulary practice technique!</p>

<!-- Spinner & Game Area -->

<div class="flex flex-col items-center justify-center mb-8">

<div class="relative flex items-center justify-center">

<div id="wheel" class="wheel">

<!-- Wheel sections will be dynamically added here -->

</div>

<div class="marker"></div>

<div class="center-circle">

<span class="text-xl font-bold text-gray-800">Spin!</span>

</div>

</div>

<button id="spin-button" class="mt-8 bg-purple-600 text-white font-bold py-3 px-8 rounded-xl shadow-lg hover:bg-purple-700 transition duration-300">

Spin the Wheel 🎡

</button>

</div>

<!-- Result & Instructions Card -->

<div id="result-card" class="bg-purple-100 rounded-2xl p-6 shadow-inner hidden">

<h2 id="technique-title" class="text-2xl font-bold text-gray-800 mb-2 text-center"></h2>

<p id="technique-description" class="text-lg text-gray-700 text-center"></p>

</div>

</div>

<script>

const spinButton = document.getElementById('spin-button');

const wheel = document.getElementById('wheel');

const resultCard = document.getElementById('result-card');

const techniqueTitle = document.getElementById('technique-title');

const techniqueDescription = document.getElementById('technique-description');

const techniques = [

{ name: "TPR (Simon Says)", desc: "The teacher volunteers quickly runs a 1-minute mini-activity with the group as if they are very young learners. Use physical actions to demonstrate a word (e.g., 'jump' or 'dance')." },

{ name: "Flashcard reveal & repeat", desc: "The teacher reveals a flashcard with a new word and image. The whole group must repeat the word loudly and clearly as if they are young learners." },

{ name: "Chant/song", desc: "The teacher makes a rhythm with claps, and everyone repeats a vocabulary word in rhythm. The whole group must participate fully as VYLs." },

{ name: "Drawing & guessing", desc: "The teacher draws a simple object on a board or paper. The 'children' (the other teachers) shout the word as a team." },

{ name: "Odd one out", desc: "The teacher presents three words, two of which are related (e.g., 'cat', 'dog', 'table'). The group must identify the word that doesn't belong and explain why." },

{ name: "Story with target words", desc: "The teacher tells a very short, simple story that includes the new vocabulary words. The group listens and then retells the story using the new words." },

{ name: "Matching pairs", desc: "The teacher displays two sets of cards, one with words and one with corresponding pictures. The group must work together to match the pairs as quickly as possible." },

{ name: "Hot seat", desc: "One teacher sits in a 'hot seat' with their back to the board. The rest of the group silently acts out or describes a word on the board, and the teacher in the 'hot seat' has to guess it." }

];

function createWheel() {

const numSections = techniques.length;

const angle = 360 / numSections;

const colors = ['#fde68a', '#d9f991', '#a7f3d0', '#bfdbfe', '#e9d5ff', '#fecaca', '#fcd5b8', '#d0e8f2'];

techniques.forEach((technique, index) => {

const section = document.createElement('div');

section.classList.add('wheel-section');

section.textContent = technique.name;

const rotation = index \* angle;

section.style.transform = `rotate(${rotation}deg)`;

const backgroundAngle = `conic-gradient(from ${rotation}deg, ${colors[index]} 0deg, ${colors[index]} ${angle}deg, transparent ${angle}deg)`;

section.style.background = backgroundAngle;

// Adjust text rotation to be upright

const textRotation = 90 + angle / 2 - rotation;

section.style.transform = `rotate(${rotation}deg) translate(0%, -50%) rotate(${textRotation}deg)`;

section.style.transformOrigin = `50% 100%`;

// Add a polygon clip-path

const halfAngle = angle / 2;

const path = `polygon(50% 50%, 100% 0, 100% 50%, 50% 100%, 0% 50%, 0% 0)`;

section.style.clipPath = path;

wheel.appendChild(section);

});

}

function spinWheel() {

spinButton.disabled = true;

resultCard.classList.add('hidden');

const randomIndex = Math.floor(Math.random() \* techniques.length);

const selectedTechnique = techniques[randomIndex];

// Calculate a random rotation angle that lands on the correct section

const totalRotations = 5 \* 360; // Spin it around a few times

const baseAngle = 360 / techniques.length;

const landingAngle = totalRotations - (randomIndex \* baseAngle) - (baseAngle / 2);

wheel.style.transform = `rotate(${landingAngle}deg)`;

// Wait for the animation to finish before showing the result

setTimeout(() => {

techniqueTitle.textContent = selectedTechnique.name;

techniqueDescription.textContent = selectedTechnique.desc;

resultCard.classList.remove('hidden');

spinButton.disabled = false;

}, 5000); // Match the CSS transition duration

}

spinButton.addEventListener('click', spinWheel);

createWheel();

</script>

</body>

</html>