!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Classroom Bingo</title>

<style>

body {

font-family: 'Comic Sans MS', cursive, sans-serif;

background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);

margin: 0;

padding: 20px;

min-height: 100vh;

color: #333;

}

.container {

max-width: 800px;

margin: 0 auto;

background: rgba(255, 255, 255, 0.95);

border-radius: 20px;

padding: 30px;

box-shadow: 0 20px 40px rgba(0, 0, 0, 0.1);

}

h1 {

text-align: center;

color: #4a5568;

font-size: 2.5em;

margin-bottom: 10px;

text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.1);

}

.subtitle {

text-align: center;

color: #718096;

font-size: 1.2em;

margin-bottom: 30px;

}

.controls {

text-align: center;

margin-bottom: 30px;

}

button {

background: linear-gradient(45deg, #ff6b6b, #feca57);

border: none;

color: white;

padding: 12px 24px;

font-size: 16px;

border-radius: 25px;

cursor: pointer;

margin: 0 10px;

transition: all 0.3s ease;

box-shadow: 0 4px 15px rgba(0, 0, 0, 0.2);

}

button:hover {

transform: translateY(-2px);

box-shadow: 0 6px 20px rgba(0, 0, 0, 0.3);

}

.bingo-board {

display: grid;

grid-template-columns: repeat(5, 1fr);

gap: 8px;

max-width: 600px;

margin: 0 auto;

background: #f7fafc;

padding: 15px;

border-radius: 15px;

box-shadow: inset 0 2px 10px rgba(0, 0, 0, 0.1);

}

.bingo-cell {

aspect-ratio: 1;

background: linear-gradient(135deg, #a8e6cf, #dcedc1);

border: 3px solid #68d391;

border-radius: 12px;

display: flex;

align-items: center;

justify-content: center;

text-align: center;

font-size: 12px;

font-weight: bold;

cursor: pointer;

transition: all 0.3s ease;

position: relative;

overflow: hidden;

}

.bingo-cell:hover {

transform: scale(1.05);

box-shadow: 0 8px 20px rgba(0, 0, 0, 0.2);

}

.bingo-cell.marked {

background: linear-gradient(135deg, #ff6b6b, #feca57);

border-color: #e53e3e;

color: white;

text-shadow: 1px 1px 2px rgba(0, 0, 0, 0.3);

}

.bingo-cell.marked::after {

content: 'â';

position: absolute;

top: 5px;

right: 5px;

font-size: 16px;

color: #fff;

text-shadow: 1px 1px 2px rgba(0, 0, 0, 0.5);

}

.free-space {

background: linear-gradient(135deg, #4299e1, #3182ce) !important;

border-color: #2b6cb0 !important;

color: white !important;

font-size: 14px;

font-weight: bold;

}

.winner-message {

position: fixed;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

background: linear-gradient(45deg, #ff6b6b, #feca57);

color: white;

padding: 30px 50px;

border-radius: 20px;

font-size: 24px;

font-weight: bold;

text-align: center;

box-shadow: 0 20px 40px rgba(0, 0, 0, 0.3);

z-index: 1000;

animation: bounce 0.6s ease-in-out;

}

@keyframes bounce {

0%, 20%, 50%, 80%, 100% { transform: translate(-50%, -50%) translateY(0); }

40% { transform: translate(-50%, -50%) translateY(-20px); }

60% { transform: translate(-50%, -50%) translateY(-10px); }

}

.overlay {

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

background: rgba(0, 0, 0, 0.5);

z-index: 999;

}

.stats {

text-align: center;

margin-top: 20px;

font-size: 18px;

color: #4a5568;

}

@media (max-width: 600px) {

.bingo-cell {

font-size: 10px;

}

h1 {

font-size: 2em;

}

.container {

padding: 20px;

margin: 10px;

}

}

</style>

</head>

<body>

<div class="container">

<h1>ð¯ Classroom Bingo ð¯</h1>

<p class="subtitle">Mark off classroom events as they happen!</p>

<div class="controls">

<button onclick="newGame()">New Game</button>

<button onclick="clearBoard()">Clear Board</button>

</div>

<div class="bingo-board" id="bingoBoard"></div>

<div class="stats">

<div id="markedCount">Marked: 0/24</div>

</div>

</div>

<script>

const classroomEvents = [

"Search for projector remote",

"Pass arrives for an unknown student",

"Package arrives",

"Phone rings",

"Unknown machine beeps",

"Search for keys",

"Search for phone",

"Door opens/closes",

"Someone knocks on unused door",

"Someone enters/exits through unused door",

"Someone swears",

"Someone asks for their password/login",

"Someone can’t find the button",

"Photoshop crashes",

"Someone saves to the “T” drive",

"Principal enters room",

"Another teacher enters room",

"Missing mouse",

"Missing keyboard",

"Recycle bin collections",

"Plants get watered",

"Someone drops something”,

"Someone loses something",

"Announcement over PA",

"Someone asks for bathroom pass",

"Music blasts from a computer",

"A pass arrives for someone",

"Fire drill/tornado drill",

"Student falls out of chair",

"Someone says 'Is this for a grade?'",

"Someone asks ‘Are we doing anything’",

"Someone asks for paper",

"Teacher says 'stop talking’",

"Someone says 'When is this due?'",

"Pencil sharpener used",

"Someone says 'I wasn't here'",

"Teacher points at board",

"Someone asks 'Can I go get...'",

"Someone says 'do you have a charger’",

"Teacher tells a great and very funny joke",

"Someone asks about ceiling tiles",

"Someone searches for chair",

"Someone says 'I'm confused'",

"Teacher says 'Moving on'",

"Someone asks 'Will this be on the test?'",

"Someone says 'I can't see'",

"Someone asks ‘can we go outside’",

"Someone asks 'What time is it?'"

];

let gameBoard = [];

let markedCells = new Set();

function shuffleArray(array) {

const newArray = [...array];

for (let i = newArray.length - 1; i > 0; i--) {

const j = Math.floor(Math.random() \* (i + 1));

[newArray[i], newArray[j]] = [newArray[j], newArray[i]];

}

return newArray;

}

function createBoard() {

const shuffled = shuffleArray(classroomEvents);

gameBoard = shuffled.slice(0, 24);

gameBoard.splice(12, 0, "FREE SPACE");

markedCells.clear();

markedCells.add(12); // Free space is automatically marked

renderBoard();

}

function renderBoard() {

const board = document.getElementById('bingoBoard');

board.innerHTML = '';

gameBoard.forEach((event, index) => {

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

cell.className = 'bingo-cell';

cell.textContent = event;

cell.onclick = () => toggleCell(index);

if (index === 12) {

cell.classList.add('free-space');

}

if (markedCells.has(index)) {

cell.classList.add('marked');

}

board.appendChild(cell);

});

updateStats();

}

function toggleCell(index) {

if (index === 12) return; // Can't toggle free space

if (markedCells.has(index)) {

markedCells.delete(index);

} else {

markedCells.add(index);

}

renderBoard();

checkForWin();

}

function checkForWin() {

const winPatterns = [

// Rows

[0, 1, 2, 3, 4], [5, 6, 7, 8, 9], [10, 11, 12, 13, 14], [15, 16, 17, 18, 19], [20, 21, 22, 23, 24],

// Columns

[0, 5, 10, 15, 20], [1, 6, 11, 16, 21], [2, 7, 12, 17, 22], [3, 8, 13, 18, 23], [4, 9, 14, 19, 24],

// Diagonals

[0, 6, 12, 18, 24], [4, 8, 12, 16, 20]

];

for (let pattern of winPatterns) {

if (pattern.every(index => markedCells.has(index))) {

showWinner();

return;

}

}

}

function showWinner() {

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

overlay.className = 'overlay';

overlay.onclick = () => document.body.removeChild(overlay);

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

message.className = 'winner-message';

message.innerHTML = 'ð BINGO! ð<br>Great job observing!<br><small>Click anywhere to continue</small>';

document.body.appendChild(overlay);

overlay.appendChild(message);

}

function updateStats() {

const marked = markedCells.size - 1; // Subtract 1 for free space

document.getElementById('markedCount').textContent = `Marked: ${marked}/24`;

}

function newGame() {

createBoard();

}

function clearBoard() {

markedCells.clear();

markedCells.add(12); // Keep free space marked

renderBoard();

}

// Initialize the game

createBoard();

</script>

</body>

</html>