

PROMPT:

Juego interactivo de memoria sobre 10 frutas.

```
<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<style>

  /* Contenedor Principal */

  .memory-game-wrapper {

    font-family: 'Segoe UI', Roboto, Helvetica, Arial, sans-serif;

    max-width: 600px;

    margin: 20px auto;

    padding: 20px;

    background-color: #f0fdf4; /* Verde muy claro */

    border-radius: 20px;

    box-shadow: 0 10px 25px rgba(0,0,0,0.1);

    text-align: center;

  }


  .game-header h2 { color: #166534; margin-bottom: 10px; }

  .game-stats { margin-bottom: 15px; font-weight: bold; color: #15803d; }


  /* Rejilla del Juego */

  .memory-game {

    display: grid;

    grid-template-columns: repeat(4, 1fr);

    gap: 10px;

    perspective: 1000px;

    margin: 0 auto;

  }


  /* Estilo de las Cartas */

  .memory-card {

    height: 100px;
```

```
position: relative;

transform: scale(1);

transform-style: preserve-3d;

transition: transform .5s;

cursor: pointer;

background-color: #ffffff;

border-radius: 8px;

border: 2px solid #bdf1c5;

}
```

```
.memory-card:active {

    transform: scale(0.97);

    transition: transform .2s;

}
```

```
.memory-card.flip {

    transform: rotateY(180deg);

}
```

```
.front-face, .back-face {

    width: 100%;

    height: 100%;

    padding: 10px;

    position: absolute;

    border-radius: 5px;

    backface-visibility: hidden;

    display: flex;

    align-items: center;

    justify-content: center;

    font-size: 2.5em;

    box-sizing: border-box;

}
```

```
.front-face {
```

```
transform: rotateY(180deg);  
background-color: #ffffff;  
}
```

```
.back-face {  
  background-color: #22c55e;  
  color: white;  
}
```

```
.back-face::after {  
  content: "?";  
  font-size: 1.5em;  
}
```

```
/* Botón de Reinicio */
```

```
#restart-btn {  
  margin-top: 20px;  
  padding: 10px 25px;  
  background-color: #16a34a;  
  color: white;  
  border: none;  
  border-radius: 50px;  
  cursor: pointer;  
  font-weight: bold;  
  transition: background 0.3s;  
}  
  
#restart-btn:hover { background-color: #15803d; }
```

```
/* Responsivo */
```

```
@media (max-width: 480px) {  
  .memory-game { grid-template-columns: repeat(4, 1fr); gap: 5px; }  
  .memory-card { height: 75px; }  
  .front-face { font-size: 1.8em; }  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="memory-game-wrapper">
```

```
  <div class="game-header">
```

```
    <h2>🍎 Juego de Memoria: Frutas</h2>
```

```
    <div class="game-stats">Movimientos: <span id="moves">0</span></div>
```

```
  </div>
```

```
<div class="memory-game" id="game-board">
```

```
</div>
```

```
<button id="restart-btn" onclick="resetGame()">Reiniciar Juego</button>
```

```
</div>
```

```
<script>
```

```
const fruits = ['🍎', '🍌', '🍇', '🍓', '🍍', '🍉', '🥥', '🍐', '🍑', '🍒', '🍊'];
```

```
let cards = [...fruits, ...fruits];
```

```
let hasFlippedCard = false;
```

```
let lockBoard = false;
```

```
let firstCard, secondCard;
```

```
let moves = 0;
```

```
function createBoard() {
```

```
  const gameBoard = document.getElementById('game-board');
```

```
  gameBoard.innerHTML = '';
```

```
  shuffle(cards);
```

```
  cards.forEach((fruit, index) => {
```

```
    const card = document.createElement('div');
```

```
    card.classList.add('memory-card');
```

```
    card.dataset.fruit = fruit;
```

```
    card.innerHTML = `
```

```

        <div class="front-face">${fruit}</div>

        <div class="back-face"></div>

    `;

    card.addEventListener('click', flipCard);

    gameBoard.appendChild(card);

    });
}

```

```

function flipCard() {
    if (lockBoard) return;
    if (this === firstCard) return;

```

```

    this.classList.add('flip');

```

```

    if (!hasFlippedCard) {
        hasFlippedCard = true;
        firstCard = this;
        return;
    }

```

```

    secondCard = this;
    checkForMatch();
}

```

```

function checkForMatch() {
    moves++;
    document.getElementById('moves').innerText = moves;
    let isMatch = firstCard.dataset.fruit === secondCard.dataset.fruit;
    isMatch ? disableCards() : unflipCards();
}

```

```

function disableCards() {
    firstCard.removeEventListener('click', flipCard);
    secondCard.removeEventListener('click', flipCard);

```

```
    resetBoard();  
  }
```

```
function unflipCards() {  
  lockBoard = true;  
  setTimeout(() => {  
    firstCard.classList.remove('flip');  
    secondCard.classList.remove('flip');  
    resetBoard();  
  }, 1000);  
}
```

```
function resetBoard() {  
  [hasFlippedCard, lockBoard] = [false, false];  
  [firstCard, secondCard] = [null, null];  
}
```

```
function shuffle(array) {  
  array.sort(() => Math.random() - 0.5);  
}
```

```
function resetGame() {  
  moves = 0;  
  document.getElementById('moves').innerText = moves;  
  createBoard();  
}
```

```
// Iniciar al cargar  
createBoard();  
</script>
```

```
</body>  
</html>
```