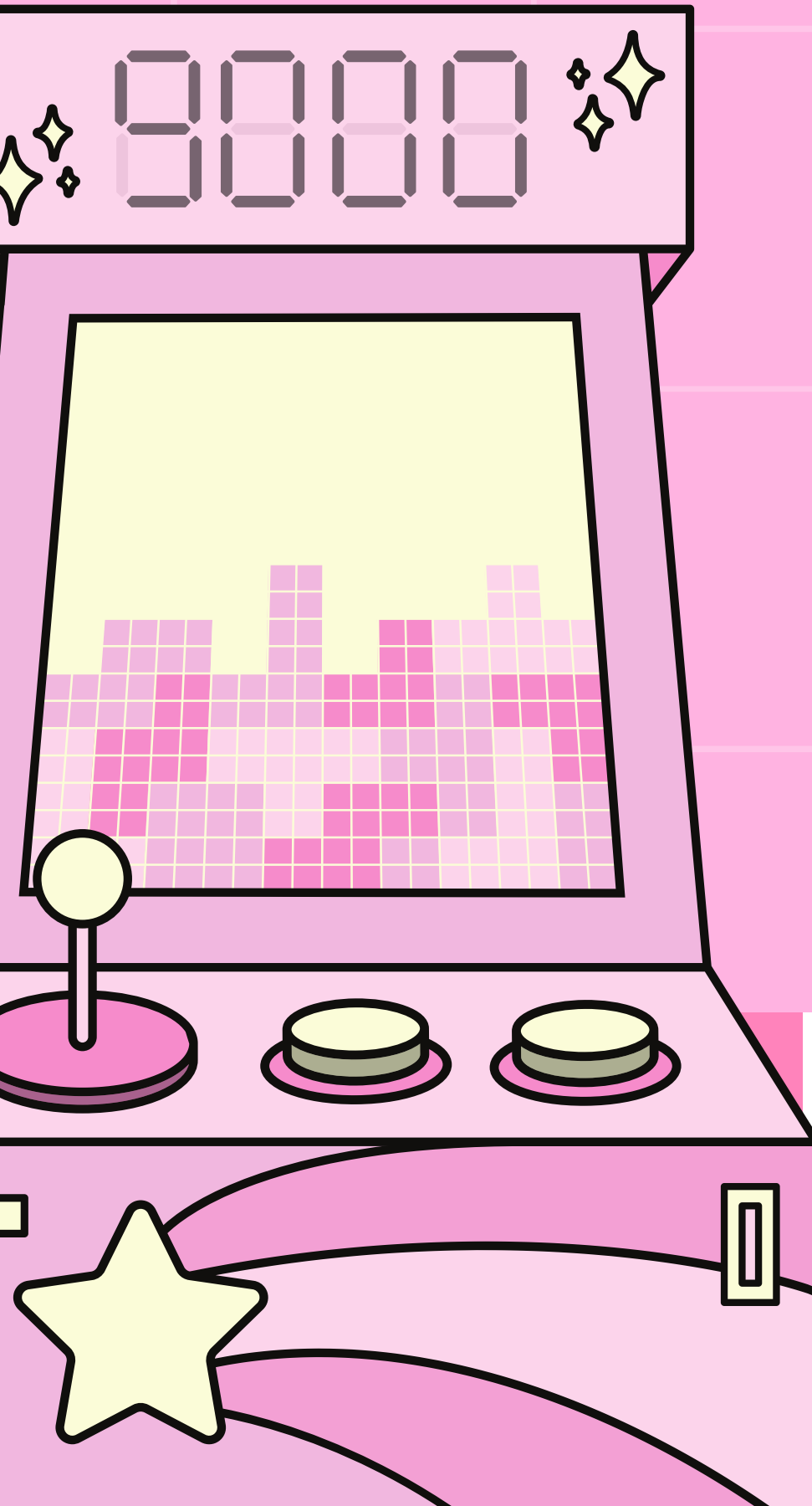


LET'S PLAY

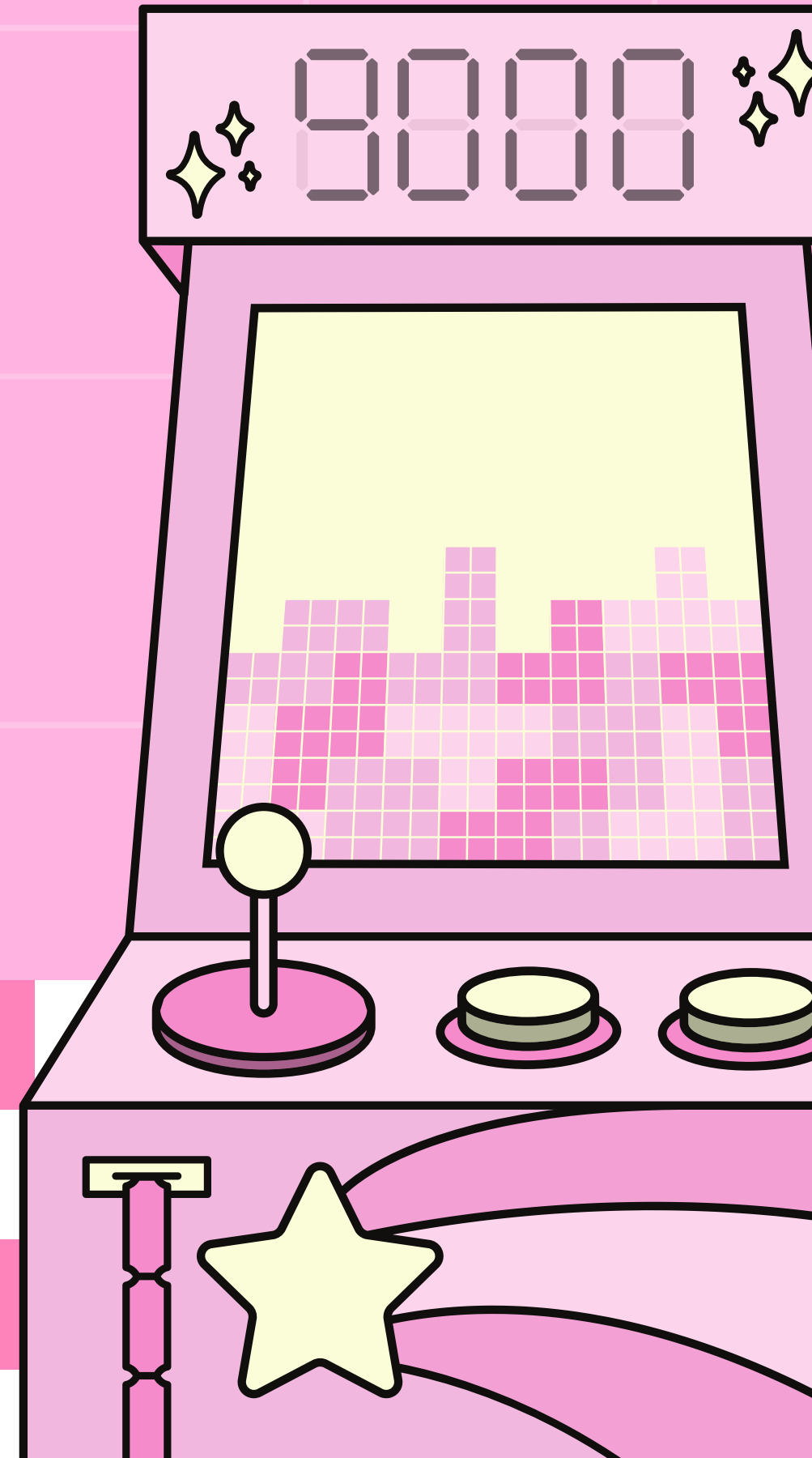
GAME

TOGETHER



TIC TAC TOE

PRESENTATION



SIOMARA ANGELINA
VALLE SANCHEZ

LET'S PLAY

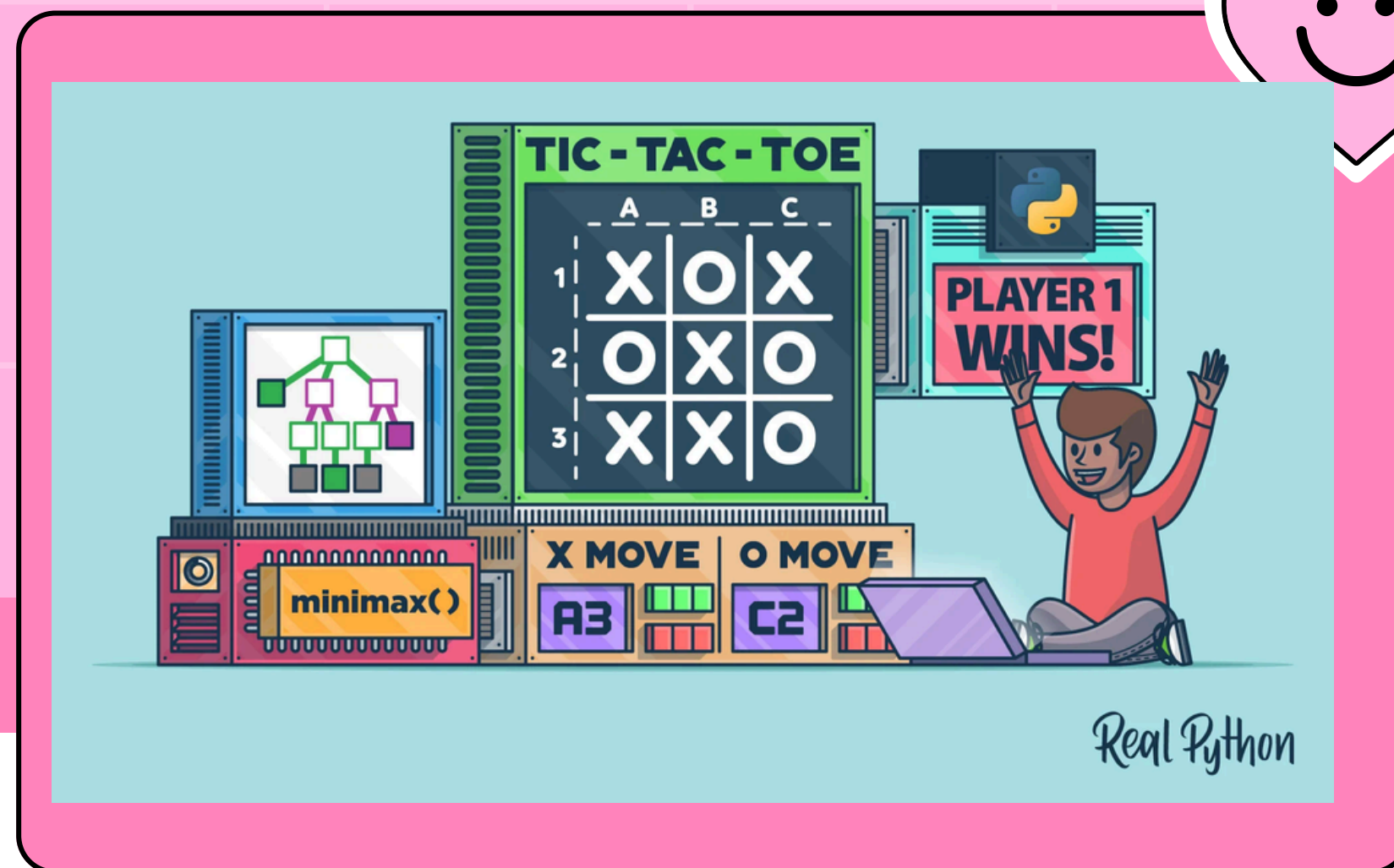
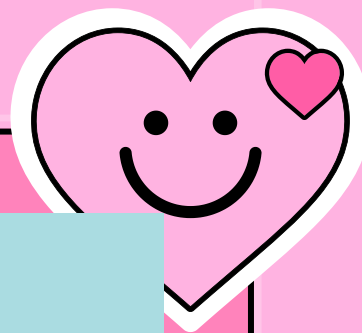
GAME

TOGETHER

WELCOME TO
THE GAME!

GAME PRESENTATION - GAME PRESENTATION - GAME PRESENTATION

RULES



LET'S PLAY

HTML

TOGETHER

```

1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4    <meta charset="utf-8" />
5    <meta name="viewport" content="width=device-width,initial-scale=1" />
6    <title>Tic Tac Toe • Angelina</title>
7    <link rel="stylesheet" href="tictactoe.css" />
8  </head>
9  <body>
10   <main class="game" role="main" aria-label="Tic Tac Toe">
11     <h1>Tic Tac Toe</h1>
12     <p class="sub">Jugador: <strong id="playerTurn">X</strong></p>
13
14     <section class="board" role="grid" aria-label="Tablero Tic Tac Toe">
15       <!-- 9 celdas -->
16       <button class="cell" data-index="0" aria-label="Celda 1" role="gridcell"></button>
17       <button class="cell" data-index="1" aria-label="Celda 2" role="gridcell"></button>
18       <button class="cell" data-index="2" aria-label="Celda 3" role="gridcell"></button>
19       <button class="cell" data-index="3" aria-label="Celda 4" role="gridcell"></button>
20       <button class="cell" data-index="4" aria-label="Celda 5" role="gridcell"></button>
21       <button class="cell" data-index="5" aria-label="Celda 6" role="gridcell"></button>
22       <button class="cell" data-index="6" aria-label="Celda 7" role="gridcell"></button>
23       <button class="cell" data-index="7" aria-label="Celda 8" role="gridcell"></button>
24       <button class="cell" data-index="8" aria-label="Celda 9" role="gridcell"></button>
25     </section>
26
27     <div class="controls">
28       <button id="restartBtn" class="btn">Reiniciar</button>
29       <button id="aiBtn" class="btn">Jugar vs CPU</button>
30     </div>
31
32     <div id="message" class="message" aria-live="polite"></div>
33   </main>
34
35   <script src="tictactoe.js"></script>
36 </body>
37 </html>
38
```

LET'S PLAY

CSS

TOGETHER

```
:root{
  --bg: #0b1220;
  --card: #0f172a;
  --accent: #06b6d4;
  --text: #e6eef8;
  --muted: #93a3b8;
}
*{ box-sizing:border-box; }
body{
  margin:0; min-height:100vh; display:flex; align-items:center; justify-content:center;
  font-family:Inter, system-ui, Arial, sans-serif; background: linear-gradient(180deg, #071024, #
  color:var(--text);
}
.game{ width: min(420px, 92vw); padding:2rem; background: linear-gradient(180deg, rgba(255,255,25
  border-radius:12px; text-align:center; box-shadow: 0 10px 30px rgba(2,6,23,.6);
}
h1{ margin:.1rem 0 .4rem; font-size:1.6rem; letter-spacing:.4px; }
.sub{ color:var(--muted); margin:0 0 1rem; }

.board{
  display:grid; grid-template-columns: repeat(3, 1fr); gap:10px;
  margin: 0 auto 1rem; max-width:360px;
}
.cell{
  aspect-ratio:1/1; font-size:2.4rem; font-weight:700; background:#071426;
  color:var(--text); border-radius:10px; border:1px solid rgba(255,255,255,.06);
  display:flex; align-items:center; justify-content:center; cursor:pointer;
  transition: transform .12s ease, background .12s ease;
}
.cell:active{ transform: scale(.98); }
.cell[disabled]{ cursor:default; opacity:.9; }
```

Copy code

```
.controls{ display:flex; gap:.6rem; justify-content:center; margin-bottom:.6rem; }
.btn{ padding:.6rem .9rem; border-radius:8px; border:0; background:var(--accent); color:#021024;
.btn:active{ transform:translateY(1px); }
.message{ min-height:1.2rem; color:var(--muted); margin-top:.35rem; }

/* estilo para celda ganadora */
.cell.win{ background: linear-gradient(90deg, rgba(6,182,212,.12), rgba(37,99,235,.12)); box-shad
```

LET'S PLAY

JS

TOGETHER

```
// Tic Tac Toe - Lógica simple (2 jugadores + opción CPU básica)
// Autor: Angelina (modificado para portafolio)
// Estructura: tablero como array de 9, X comienza.

const cells = Array.from(document.querySelectorAll('.cell'));
const playerTurnEl = document.getElementById('playerTurn');
const restartBtn = document.getElementById('restartBtn');
const aiBtn = document.getElementById('aiBtn');
const messageEl = document.getElementById('message');

let board;          // array 9: null, 'X' o 'O'
let currentPlayer;  // 'X' o 'O'
let playingVsAI = false;
let running = true;

// combinaciones ganadoras
const wins = [
  [0,1,2],[3,4,5],[6,7,8], // filas
  [0,3,6],[1,4,7],[2,5,8], // columnas
  [0,4,8],[2,4,6]          // diagonales
];
```

```
function init(){
  board = Array(9).fill(null);
  currentPlayer = 'X';
  running = true;
  playerTurnEl.textContent = currentPlayer;
  messageEl.textContent = '';
  cells.forEach((c, i) => {
    c.textContent = '';
    c.disabled = false;
    c.classList.remove('win');
    c.addEventListener('click', onCellClick);
  });
}

function onCellClick(e){
  const idx = Number(e.currentTarget.dataset.index);
  if (!running || board[idx]) return;
  makeMove(idx, currentPlayer);
  if (checkEnd()) return;
  if (playingVsAI && currentPlayer === 'O'){
    // espera un poco y juega la CPU (movimiento aleatorio sencillo)
    setTimeout(() => {
      cpuMove();
      checkEnd();
    }, 300);
  }
}
```


LET'S PLAY

```
function makeMove(index, player){
  board[index] = player;
  const el = cells[index];
  el.textContent = player;
  el.disabled = true;
  currentPlayer = player === 'X' ? 'O' : 'X';
  playerTurnEl.textContent = currentPlayer;
}

function cpuMove(){
  // CPU elige una casilla aleatoria vacía (simple)
  const empties = board.map((v,i)=> v ? null : i).filter(v=> v!==null);
  if (!empties.length) return;
  const choice = empties[Math.floor(Math.random()*empties.length)];
  makeMove(choice, 'O');
}

function checkEnd(){
  // comprobar victoria
  for (let combo of wins){
    const [a,b,c] = combo;
    if (board[a] && board[a] === board[b] && board[a] === board[c]){
      // hay ganador
      running = false;
      highlightWin(combo);
      messageEl.textContent = `¡Ganó ${board[a]}!`;
    }
  }
}
```

JS

```
messageEl.textContent = `¡Ganó ${board[a]}!`;
return true;
}
}
// empate
if (board.every(Boolean)){
  running = false;
  messageEl.textContent = "¡Empate!";
  return true;
}
return false;
}

function highlightWin(combo){
  combo.forEach(i => cells[i].classList.add('win'));
  cells.forEach(c => c.disabled = true);
}

restartBtn.addEventListener('click', () => {
  init();
});

aiBtn.addEventListener('click', () => {
  playingVsAI = !playingVsAI;
  aiBtn.textContent = playingVsAI ? 'Jugar vs Humano' : 'Jugar vs CPU';
  init();
});

// iniciar juego al cargar
init();
```

TOGETHER

SEE

YOU

NEXT!

THANK
YOU

