Lights Out Solution

Download our solution

src/Board.js

```
import React, { useState } from "react";
import Cell from "./Cell";
import "./Board.css";

/** Game board of Lights out.
 *
 * Properties:
 *
 * - nrows: number of rows of board
 * - ncols: number of cols of board
 * - chanceLightStartsOn: float, chance any cell is lit at start of game
 *
 * State:
 *
 * - board: array-of-arrays of true/false
 *
 *    For this board:
 *       .  .  .
 *       0  0  .     (where . is off, and 0 is on)
 *       .  .  .
 *
 *    This would be: [[f, f, f], [t, t, f], [f, f, f]]
 *
 *  This should render an HTML table of individual <Cell /> components.
 *
 *  This doesn't handle any clicks --- clicks are on individual cells
 *
 **/

function Board({ nrows = 5, ncols = 5, chanceLightStartsOn = 0.25 }) {
  const [board, setBoard] = useState(createBoard());

  /** create a board nrows high/ncols wide, each cell randomly lit or unlit */
  function createBoard() {
    let initialBoard = [];
    for (let y = 0; y < nrows; y++) {
      let row = [];
      for (let x = 0; x < ncols; x++) {
        row.push(Math.random() < chanceLightStartsOn);
      }
      initialBoard.push(row);
    }
    return initialBoard;
  }

  /* Check if the player has won */
  function hasWon() {
    return board.every(row => row.every(cell => !cell));
  }

  /* Flip cells around a given cell */
  function flipCellsAround(coord) {
    setBoard(oldBoard => {
      const [y, x] = coord.split("-").map(Number);

      const flipCell = (y, x, boardCopy) => {
        // if this coord is actually on board, flip it

        if (x >= 0 && x < ncols && y >= 0 && y < nrows) {
          boardCopy[y][x] = !boardCopy[y][x];
        }
      };

      const boardCopy = oldBoard.map(row => [...row]);

      flipCell(y, x, boardCopy);
      flipCell(y, x - 1, boardCopy);
      flipCell(y, x + 1, boardCopy);
      flipCell(y - 1, x, boardCopy);
      flipCell(y + 1, x, boardCopy);

      return boardCopy;
    });
  }

  // if the game is won, just show a winning msg & render nothing else
  if (hasWon()) {
    return <div>You Win!</div>;
  }

  // make table board: rows of Cell components

  let tblBoard = [];

  for (let y = 0; y < nrows; y++) {
    let row = [];
    for (let x = 0; x < ncols; x++) {
      let coord = `${y}-${x}`;
      row.push(
        <Cell
          key={coord}
          isLit={board[y][x]}
          flipCellsAroundMe={() => flipCellsAround(coord)}
        />
      );
    }
    tblBoard.push(<tr key={y}>{row}</tr>);
  }

  return (
    <table className="Board">
      <tbody>{tblBoard}</tbody>
    </table>
  );
}

export default Board;
```

src/Cell.js

```
import React from "react";
import "./Cell.css";

/** A single cell on the board.
 *
 * This has no state --- just two props:
 *
 * - flipCellsAroundMe: a function rec'd from the board which flips this
 *      cell and the cells around of it
 *
 * - isLit: boolean, is this cell lit?
 *
 * This handles clicks --- by calling flipCellsAroundMe
 *
 **/

function Cell({ flipCellsAroundMe, isLit=false }) {
  const classes = `Cell ${isLit ? "Cell-lit" : ""}`;
  return <td className={classes} onClick={flipCellsAroundMe} role="button" />;
}

export default Cell;
```