TIC TOC TOE

**App.js**

import { useState } from 'react';

function Square({ value, onSquareClick }) {

 return (

 <button className="square" onClick={onSquareClick}>

  {value}

 </button>

 );

}

function Board({ xIsNext, squares, onPlay }) {

 function handleClick(i) {

 if (calculateWinner(squares) || squares[i]) {

  return;

 }

 const nextSquares = squares.slice();

 if (xIsNext) {

  nextSquares[i] = 'X';

 } else {

  nextSquares[i] = 'O';

 }

 onPlay(nextSquares);

 }

 const winner = calculateWinner(squares);

 let status;

 if (winner) {

 status = 'Winner: ' + winner;

 } else {

 status = 'Next player: ' + (xIsNext ? 'X' : 'O');

 }

 return (

 <>

  <div className="status">{status}</div>

  <div className="board-row">

  <Square value={squares[0]} onSquareClick={() => handleClick(0)} />

  <Square value={squares[1]} onSquareClick={() => handleClick(1)} />

  <Square value={squares[2]} onSquareClick={() => handleClick(2)} />

  </div>

  <div className="board-row">

  <Square value={squares[3]} onSquareClick={() => handleClick(3)} />

  <Square value={squares[4]} onSquareClick={() => handleClick(4)} />

  <Square value={squares[5]} onSquareClick={() => handleClick(5)} />

  </div>

  <div className="board-row">

  <Square value={squares[6]} onSquareClick={() => handleClick(6)} />

  <Square value={squares[7]} onSquareClick={() => handleClick(7)} />

  <Square value={squares[8]} onSquareClick={() => handleClick(8)} />

  </div>

 </>

 );

}

export default function Game() {

 const [history, setHistory] = useState([Array(9).fill(null)]);

 const [currentMove, setCurrentMove] = useState(0);

 const xIsNext = currentMove % 2 === 0;

 const currentSquares = history[currentMove];

 function handlePlay(nextSquares) {

 const nextHistory = [...history.slice(0, currentMove + 1), nextSquares];

 setHistory(nextHistory);

 setCurrentMove(nextHistory.length - 1);

 }

 function jumpTo(nextMove) {

 setCurrentMove(nextMove);

 }

 const moves = history.map((squares, move) => {

 let description;

 if (move > 0) {

  description = 'Go to move #' + move;

 } else {

  description = 'Go to game start';

 }

 return (

  <li key={move}>

  <button id="btnmove" onClick={() => jumpTo(move)}>{description}</button>

  </li>

 );

 });

 return (

 <div className="game">

  <div className="game-board">

  <Board xIsNext={xIsNext} squares={currentSquares} onPlay={handlePlay} />

  </div>

  <div className="game-info">

  <ol>{moves}</ol>

  </div>

 </div>

 );

}

function calculateWinner(squares) {

 const lines = [

 [0, 1, 2],

 [3, 4, 5],

 [6, 7, 8],

 [0, 3, 6],

 [1, 4, 7],

 [2, 5, 8],

 [0, 4, 8],

 [2, 4, 6],

 ];

 for (let i = 0; i < lines.length; i++) {

 const [a, b, c] = lines[i];

 if (squares[a] && squares[a] === squares[b] && squares[a] === squares[c]) {

  return squares[a];

 }

 }

 return null;

}

**Index.css:**

\* {

 box-sizing: border-box;

}

body {

 font-family: sans-serif;

 margin: 20px;

 padding: 0;

}

h1 {

 margin-top: 0;

 font-size: 22px;

}

h2 {

 margin-top: 0;

 font-size: 20px;

}

h3 {

 margin-top: 0;

 font-size: 18px;

}

h4 {

 margin-top: 0;

 font-size: 16px;

}

h5 {

 margin-top: 0;

 font-size: 14px;

}

h6 {

 margin-top: 0;

 font-size: 12px;

}

code {

 font-size: 1.2em;

}

ul {

 padding-inline-start: 20px;

}

li{

 color:#4c00ff;

}

#btnmove{

 color:blue;

 font-size: 1rem;

}

\* {

 box-sizing: border-box;

}

body {

 font-family: sans-serif;

 margin: 160px;

 padding: 0;

}

.square {

 background: #fff;

 border: 4px solid hsl(129, 97%, 50%);

 float: left;

 font-size: 4rem;

 font-weight: bold;

 line-height: 34px;

 /\* height: 44px;\*/

 margin-right: -1px;

 margin-top: -1px;

 padding: 0;

 text-align: center;

 width: 94px;

 height: 94px;

 color:blue

}

.board-row:after {

 clear: both;

 content: '';

 display: table;

}

.status {

 margin-bottom: 10px;

 font-size: 2rem;

 color: blue;

}

.game {

 display: flex;

 flex-direction: row;

}

.game-info {

 margin-left: 20px;

}