

```

let currentPlayerSymbol = 'x';
let squareValues = ['', '', '', '', '', '', '', '', ''];
let gameStatus = '';
// Bonus: Computer Player
// let computerPlayerSymbol = '';
// assignComputerPlayer();

const gameStateKey = 'tic-tac-toe-game-state';

function saveGameState() {
  const state = {
    currentPlayerSymbol,
    squareValues,
    gameStatus,
    // computerPlayerSymbol // Bonus
  };

  window.localStorage.setItem(gameStateKey, JSON.stringify(state));
  // Bonus
  // if (currentPlayerSymbol === computerPlayerSymbol && gameStatus === '') takeComputerTurn();
}

function loadGameState() {
  const savedState = window.localStorage.getItem(gameStateKey);
  if (savedState === null) return;

  const state = JSON.parse(savedState);
  currentPlayerSymbol = state.currentPlayerSymbol;
  squareValues = state.squareValues;
  gameStatus = state.gameStatus;
  // computerPlayerSymbol = state.computerPlayerSymbol; // Bonus

  for (let i = 0; i < 9; i += 1) {
    if (squareValues[i] !== '') {
      const img = document.createElement('img');
      img.src = `https://assets.aaonline.io/Module-DOM-API/formative-project-tic-tac-toe/player-${squareValues[i]}.svg`;
      document.getElementById(`square-${i}`).appendChild(img);
    }
  }

  if (gameStatus !== '') {
    document.getElementById('game-status').innerHTML = `Winner: ${gameStatus}${getWinner()}`; // Bonus: getWinner()
  }
}

```

```

    document.getElementById('new-game').disabled = false;

    document.getElementById('give-up').disabled = true;
} else {
    document.getElementById('game-status').innerHTML = '';

    document.getElementById('new-game').disabled = true;

    document.getElementById('give-up').disabled = false;

    if (currentPlayerSymbol === computerPlayerSymbol && gameStatus === '') takeComputerTurn();
}
}

function checkGameStatus() {
    // Check rows
    for (let i = 0; i < 9; i += 3) {
        if (squareValues[i] !== '' && squareValues[i] === squareValues[i + 1] && squareValues[i] === squareValues[i + 2]) {
            gameStatus = squareValues[i].toUpperCase();
            break;
        }
    }

    // Check columns
    for (let i = 0; i < 3; i += 1) {
        if (squareValues[i] !== '' && squareValues[i] === squareValues[i + 3] && squareValues[i] === squareValues[i + 6]) {
            gameStatus = squareValues[i].toUpperCase();
            break;
        }
    }

    // Check the diagonals
    if (squareValues[0] !== '' && squareValues[0] === squareValues[4] && squareValues[0] === squareValues[8]) {
        gameStatus = squareValues[0].toUpperCase();
    }

    if (squareValues[2] !== '' && squareValues[2] === squareValues[4] && squareValues[2] === squareValues[6]) {
        gameStatus = squareValues[2].toUpperCase();
    }

    if (gameStatus === '') {
        let gridIsAllFilled = true;
        for (let i = 0; i < 9; i += 1) {

```

```

        if (squareValues[i] === '') {
            gridIsAllFilled = false;
            break;
        }
    }

    if (gridIsAllFilled) {
        gameStatus = 'None';
    }
}

if (gameStatus !== '') {
    document.getElementById('game-status').innerHTML = `Winner: ${gameStatus}${getWinner()}`; // Bonus: getWinner()

    document.getElementById('new-game').disabled = false;

    document.getElementById('give-up').disabled = true;
}

saveGameState();
}

window.addEventListener('DOMContentLoaded', () => {
    loadGameState();

    document.getElementById('tic-tac-toe-board').addEventListener('click', (e) => {
        if (gameStatus !== '') return;

        const targetId = e.target.id;

        if (!targetId.startsWith('square-')) return;

        const squareIndex = Number.parseInt(targetId[targetId.length - 1]);

        if (squareValues[squareIndex] !== '') return;

        // const img = document.createElement('img');
        // img.src = `https://assets.aonline.io/Module-DOM-API/formative-project-tic-tac-toe/player-${currentPlayerSymbol}.svg`;
        // e.target.appendChild(img);

        // squareValues[squareIndex] = currentPlayerSymbol;

        // if (currentPlayerSymbol === 'x') {

```

```

    // currentPlayerSymbol = 'o';
    // } else {
    // currentPlayerSymbol = 'x';
    // }

    // checkGameStatus();

    // Bonus: Refactor above into markSquare()
    // markSquare(e.target, squareIndex, currentPlayerSymbol);
  });

document.getElementById('new-game').addEventListener('click', () => {
  currentPlayerSymbol = 'x';
  squareValues = ['', '', '', '', '', '', '', '', ''];
  gameStatus = '';
  assignComputerPlayer();

  for (let i = 0; i < 9; i += 1) {
    document.getElementById(`square-${i}`).innerHTML = '';
  }

  document.getElementById('game-status').innerHTML = '';

  document.getElementById('new-game').disabled = true;

  document.getElementById('give-up').disabled = false;

  saveGameState();
});

document.getElementById('give-up').addEventListener('click', () => {
  if (currentPlayerSymbol === 'x') {
    gameStatus = 'O';
  } else {
    gameStatus = 'X';
  }

  document.getElementById('game-status').innerHTML = `Winner: ${gameStatus}${getWinner()}`; // Bonus: getWinner()

  document.getElementById('new-game').disabled = false;

  document.getElementById('give-up').disabled = true;
});

```

```

        saveGameState();
    });
});

// Bonus: Computer Player
// function assignComputerPlayer() {
//   computerPlayerSymbol = [ 'x', 'o' ][Math.floor(Math.random() * 2)];
//   console.log(`Computer player: ${computerPlayerSymbol}`);
// }

// function takeComputerTurn() {
//   const idx = Math.floor(Math.random() * 9);
//   if (squareValues[idx] === '') {
//     const square = document.getElementById(`square-${idx}`);
//     markSquare(square, idx, computerPlayerSymbol);
//   } else {
//     takeComputerTurn();
//   }
// }

// function getWinner() {
//   let winner = '';
//   switch (gameStatus.toLowerCase()) {
//     case computerPlayerSymbol:
//       winner = ' (Computer)';
//       break;
//     case 'none':
//       break;
//     default:
//       winner = ' (Human)';
//       break;
//   }
//   return winner;
// }

// We can find a square element from an idx or an idx from a square element, however,
// the computer is working from a random index and the human is working from clicking
// a square, so each player would have to find the opposite in order to pass in,
// so we just take in both parameters to leave the distinction to where it was called.
function markSquare(square, idx, symbol) {
  const img = document.createElement('img');
  img.src = `https://assets.aonline.io/Module-DOM-API/formative-project-tic-tac-toe/player-${symbol}.svg`;
  square.appendChild(img);

```

```
squareValues[idx] = symbol;

currentPlayerSymbol = currentPlayerSymbol === 'x' ? 'o' : 'x';

checkGameStatus();
}
```