```
<!-- URL -->
<!-- https://austinshigh.github.io/worldle-react-2/wordle-react-2.html
-->

<!-- Which seems better for creating the elements of the app: JS alone
or JS with React?
    JSX is much easier to use than JS alone or JS with React. Because
the syntax mimics HTML, it is far more intuitive than plain JS or JS
with React.
-->
<!DOCTYPE html>
<head>
    <title>Guess the Word!</title>
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <style type="text/css">
        .backgroundContainer{height: 100%;}
                .innerContainer{display: flex; flex-direction: column;
align-items: center; height: 100%;}
                .outterContainer{background-color: white;
display:flex; height: 100%; justify-content: center; width: 100%;}
                .grid{display: grid; grid-template-columns: 1fr 1fr
1fr 1fr 1fr; grid-template-rows: 1fr 1fr 1fr 1fr 1fr 1fr;}
                .square {text-align: center; height: 50px; width:
50px; line-height: 50px; margin: 5px; border: 1px; border-color:
rgb(0, 0, 0); border-style: solid; font-size: 30px;}
        .gameWon{animation-name: example; animation-duration: 4s;
background-color: black;}
       @keyframes example { from {background-color: white;} to
{background-color: black;}}
        .turn{border: 1px; border-color: black; border-style: solid;
width: 100px;}
        .winner{border: 1px; border-color: green; border-style: solid;
width: 100px;}
        .keyboard{display: flex; flex-direction: column; align-items:
center;}
        .row{display: flex; flex-direction: row;}
        .key{text-align: center; width: 35px; height: 35px; margin:
5px; border: 1px; border-color: #333333; border-style: solid; font-
size: 28px; border-radius: 5px; box-shadow: 1px 1px 1px #333333;}
        .key:hover{background-color: rgb(0, 132, 255);}
                .key:active{background-color: rgb(0, 54, 104); color:
white;}
        .wideKey{text-align: center; height: 35px; line-height: 35px;
margin: 5px; border: 1px; border-color: #333333; border-style: solid;
font-size: 20px; border-radius: 5px; width: 80px; box-shadow: 1px 1px
1px #333333;}
        .wideKey:hover{background-color: rgb(0, 132, 255);}
                .wideKey:active{background-color: rgb(0, 54, 104);
color: white;}
```

```
        .disable{pointer-events: none; background-color: grey;}
        .correct{background-color: lightgreen;}
        .close{background-color: rgb(238, 190, 20);}
        .gridContainer{display: flex; justify-content: center;}
        .keyboard{margin-top: 30px;}
        #newGame{width: 100px;}
        #debugMode{width: 100px;}
        .active{background-color: rgb(0, 132, 255); color: white;}
        #correctAnswerContainer{color: green;}
        .show{display:block !important;}
                .statContainer{display: flex; gap: 10px; margin: 15px
0px;}
        .whiteText{color: white;}
        @media (max-width: 700px){
            h1{
                font-size: 25px;
                margin-top: 0;
            }
            .square{
                width: 30px;
                height: 30px;
                line-height: 30px;
                font-size: 25px;
                        }
            .keyboard{
                margin-top: 10px;
            }
            .key{
                width: 25px;
                height: 30px;
                font-size: 22px;
                line-height: 30px;
                margin: 4px;
            }
            .wideKey{
                height: 30px;
                line-height: 30px;
                font-size: 18px;
                width: 50px;
            }
            .wideKey#newGame{
                width: 100px;
            }
            .statContainer{
                font-size: 18px;
            }
            .correctAnswerContainer{
                font-size: 18px;
            }
        }
```

```
        </style>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/
jquery.min.js"></script>
    <script crossorigin src="https://unpkg.com/react@17/umd/
react.development.js"></script>
    <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-
dom.development.js"></script>
    <script src="https://unpkg.com/@babel/standalone/babel.min.js"></
script>
</head>
<body>
    <div id="app"></div>
    <script type = 'text/babel'>
    function Letter(letter, key){
        return <div className="key" id={letter} key={key} tabIndex={0}
>{letter}</div>;
    }

    function WideKey(id, innerText){
        return <div className="wideKey" id={id}>{innerText}</div>
    }

    function Row(keyArray){
        let arrayElements = keyArray.map((item, i) => <div key={i}
>{item}</div>);
        return <div className="row">{arrayElements}</div>
    }

    const NUM_SQUARES = 5;
    const NUM_ROWS = 6;

    let grid = [];
    for (let i=0; i<NUM_ROWS; i++){
        for (let j=0; j<NUM_SQUARES; j++)
        {
            grid.push(<div className={`square row${i} col${j}`}
key={`${i}${j}`}/>)
        }
    }

    let gridComponent = <div className="gridContainer"><div
className="grid">{grid}</div></div>;

    var buttonRow = Row([WideKey("newGame", "New Game"),
WideKey("debugMode", "Debug")]);
    var letRowOne = Row([Letter("Q"),
Letter("W"),Letter("E"),Letter("R"),Letter("T"),Letter("Y"),Letter("U"
),Letter("I"),Letter("O"),Letter("P")]);
    var letRowTwo = Row([Letter("A"),
Letter("S"),Letter("D"),Letter("F"),Letter("G"),Letter("H"),Letter("J"
```

```
),Letter("K"),Letter("L")]);
    var letRowThree = Row([WideKey("backspace", "Delete"),Letter("Z"),
Letter("X"),Letter("C"),Letter("V"),Letter("B"),Letter("N"),Letter("M"
),WideKey("enterKey", "Enter")]);
    var header =<h1>Guess the word!</h1>;
    var correctAnswer = <div id="correctAnswerContainer"></div>;

    var winCaption = <div id="winCaption">Wins:</div>;
    var wins = <div id="wins">0</div>;
    var lossesCaption = <div id="lossCaption">Losses:</div>;
    var losses = <div id="losses">0</div>;
    var guessesCaption = <div id="guessesCaption">0</div>;
    var guesses = <div id="guesses">0</div>;

    var statContent = [winCaption, wins, lossesCaption, losses,
guessesCaption, guesses].map((item, i) => <div key={i}>{item}</div>);
    var stats = <div className="statContainer">{statContent}</div>;

    var domContent = [header, correctAnswer, stats, buttonRow,
gridComponent, letRowOne, letRowTwo, letRowThree].map((item, i) =>
<div key={i}>{item}</div>);
    var backgroundContainer = <div className="backgroundContainer">
        <div className="outerContainer">
            <div className="innerContainer">{domContent}</div>
        </div>
    </div>;

    ReactDOM.render(backgroundContainer, app);

    </script>

    <script>

    window.addEventListener('DOMContentLoaded', function() {

        (async function($) {
            const NUM_SQUARES = 5;
            const NUM_ROWS = 6;
            let currentRow = 0;
            let currentCol = 0;
            let guess = [];

            let guessCount = 0;
            let wins = 0;
            let losses = 0;

            let winDiv = $("#wins");
            let lossDiv = $("#losses");
            let guessDiv = $("#guesses");
```

```
            const correctAnswerContainer = $
("#correctAnswerContainer");
            const backspaceKey = $("#backspace");
            const enterKey = $("#enterKey");
            const restartGameKey = $("#newGame");
            const debugKey = $("#debugMode");

            let letterCount = [];
            let letterCountTemp = [];

            let lettersGuessed = [];
            let answer;

            generateNewWord();

            updateGuess("m", "m");
            updateGuess("i", "i");
            updateGuess("g", "g");
            updateGuess("h", "h");
            updateGuess("t", "t");

            checkAnswer();

            updateGuess("f", "f");
            updateGuess("l", "l");
            updateGuess("o", "o");
            updateGuess("o", "o");
            updateGuess("d", "d");

            checkAnswer();

            updateGuess("s", "s");
            updateGuess("t", "t");
            updateGuess("r", "r");
            updateGuess("a", "a");
            updateGuess("y", "y");

            checkAnswer();

            function updateGuess(letter, key){
                if (currentCol < 5){
                    guess.push(letter.toLowerCase());
                    lettersGuessed.push(key);
                    let square = $(`.row${currentRow}.col${currentCol}
`);
                    $(square).text(letter);
                    currentCol += 1;
                }
            }
```

```
            function countLetters(word){
                letterCount = [];
                letterCountTemp = [];
                word.forEach((letter) => {
                    if (letterCount[letter] > 0){
                        letterCount[letter] = letterCount[letter] + 1;
                        letterCountTemp[letter] =
letterCountTemp[letter] + 1;
                    }else{
                        letterCount[letter] = 1;
                        letterCountTemp[letter] = 1;
                    }
                })
            }

            async function checkAnswer(){
                        guessCount += 1;
                        $(guessDiv).text(guessCount);
                        let correctLetters = 0;
                        let row = $(`.row${currentRow}`);
                        for(i = 0; i < 5; i++){
                            if (answer[i] === guess[i]){
                                $(row[i]).addClass("correct");
                                $(`#${guess[i]}`).addClass("correct");
                                correctLetters += 1;
                            }else if(answer.includes(guess[i])){
                                if (letterCount[guess[i]] > 0){
                                    $(row[i]).addClass("close");
                                    letterCount[guess[i]] -= 1;
                                }
                            }else{
                                $(`#${guess[i]}`).addClass("disable");
                            }
                        }
                        if(correctLetters === 5){
                            wins += 1;
                            displayWinningAnimation();
                            $(winDiv).text(wins);
                        }else if (currentRow === 5){
                            alert(`you lose :( the correct answer was
${$(correctAnswerContainer).text()}`);
                            losses += 1;
                            $(lossDiv).text(losses);
                        }
                        currentCol = 0;
                        currentRow += 1;
                        guess = [];
                    letterCount = letterCountTemp;
            }
```

```
            async function generateNewWord(){
                let word = "moody";
                $(correctAnswerContainer).text(word);
                countLetters(word.split(""));
                answer = word.split("");
            }

        })(jQuery);

    });
    </script>
</body>
</html>
```