Tic Tac Toe Plan

GameContext

- gameBoard- Array of 9 tiles ['X' , 'X' , 'O', …]
- currentPlayer ('X' or 'O')
- gameIsActive (true or false)
- gameMessage (says who's turn it is, who won or if its a cats game)

Components

- App.js
  - Calls gameBoard component
  - Checks for endGame() state on every render
    - Tie: All spaces are taken
    - Win/Lose: Player has 3 in a row
      - If endGame → set gameIsActive to false and update gameMessage
- GameBoard.js
  - Renders 9 tiles
- Tile.js
  - onClick → HandleClick()
  - HandleClick()
    - if space contains 'X' or 'O' → Return
    - if gameIsActive === false → Return
    - set tile's content to currentPlayer

        - switch currentPlayer
- Reset.js
    - Button for resetting state
        - Reset currentPlayer to ‘X’
        - update gameMessage

Steps

1. Declare state variables in GameContext.js
2. Initialize state values
    a. Check → **import state in App.js and console.log it to make sure its all wired up correctly**
3. Create GameBoard and Tile components. GameBoard component reads gameBoard state from context and renders a Tile component for each value. Style a little.
    a. Board displays on page and values of tiles change with gameBoard array values
4. Add onClick event for tile component that calls HandleClick() function from context.
    a. consoleLog in HandleClick() function to make sure it’s triggering
5. Define HandleClick() function.
    a. Clicking on tiles should render X or O. Does nothing if gameIsActive state is false.
6. Create function to check for endGame conditions and trigger it on every render.
    a. Game Works!
7. Reset Button
    a. Reset works

Win Conditions:

3 in a row combinations:

1's: 1, 2, 3

2's: 4, 5, 6

3's: 7, 8, 9

a's: 1, 4, 7

b's: 2, 5, 8

c's: 3, 6, 9

diagonal 1: 1, 5, 9

diagonal 2: 3, 5, 7

turns before check needed: 5