1. Triggering A Render With Nested Components

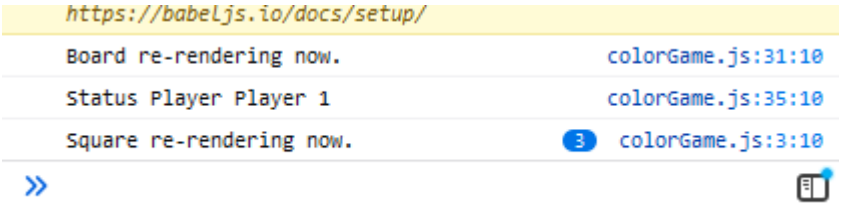
Through this week's videos, you've seen a Board component which comprises multiple Square components. `React.useState()` manages the state of all components. In this implementation of the Tic-tac-toe board, you'll notice that the state change (and subsequent re-render) of a parent also re-renders a child component.

In this activity, you'll create a slightly modified version of the Tic-tac-toe board from the videos. Here, you'll learn that it's possible to create components that update independently of each-other.

Through the course of this activity, you'll observe some key concepts in action as well as make a small change to restore the original functionality where a click in the child (Square) triggers a state change in both parent and child and results in both of them being re-rendered.

Before starting this activity, there are a few things to observe.

1. **Track re-render using the console:** You can see whether the Board (parent component) or the Square (child component) re-renders on change by looking at the developer tools console in the browser. It will look like this:



2. **When a parent is re-rendered, so is the child:** There is a new button on the board labelled `Change Player`. Selecting this button changes the player in the Board component but does nothing visible in the child. In the console, you'll observe that the console.log for Square re-rendering is still triggered. This shows that when a parent is re-rendered, so is any child of that parent component.
3. **Child state can change without impacting the parents:** Selecting the Squares does change their color (the first few times it's clicked) but doesn't change the Player in the parent *or* trigger a re-render of the parent component. This shows that the state of the child component *can* update without affecting the parent.

4. `innerTakeTurn` **and** `takeTurn` **have different properties:** Note the difference between the `innerTakeTurn` function vs the `takeTurn` function. This'll be important later on. Take a look at the comments above each function for clues.

Now that these observations have been made, you may notice that this isn't desired functionality. The way that you change the Square color should **also** change the player. This is expected functionality for a Tic-tac-toe application. Now you'll go through a step to re-connect the updates in the child to affect the parent and achieve the desired functionality.

Your task in this activity will be to make a code change such that the correct takeTurn function is called to ensure that both child state and parent state are updated upon selecting the child.

To complete this activity, simply call the `takeTurn()` **function rather than the** `innerTakeTurn()` **function which is currently triggered in the** `onClick` **of the** `Square` **button.** The difference between the two functions is that `takeTurn()` updates the `player` in the `Board` state as well as returning the player with the equation `(player + 1) % 2`.
The `innerTakeTurn()` function, however, simply returns the player value wth the same equation **without** updating that player in state. You can see from your observations above that this partial functionality doesn't work in the context of a Tic-tac-toe game where you want to change the color *and* change the player as each player only gets one turn.

Note the differences between those two functions.
In the `takeTurn` function, it sets a player based off of the formula ((player + 1) % 2) *and* returns that new player value to be used in `setColor`.
However, the `innerTakeTurn` function only calculates the value of the new player and returns it to be used in setColor.

You can further experiment with different combinations of button selections and see if you can find any other unexpected functionality in this version of code. Just remember that to submit this activity, the `takeTurn` function must change the color *and* update the current player. No other change is required.

Hint:

- You are only changing one instance of `innerTakeTurn` to `takeTurn`

- Go to this link if you are not sure how to view the logs produced by `console.log` in developer tools