

MineSweeper

Start Game

Difficulty Adjustment

Mines Number

10

Board Size (n×n)

8



Practice Homework #4

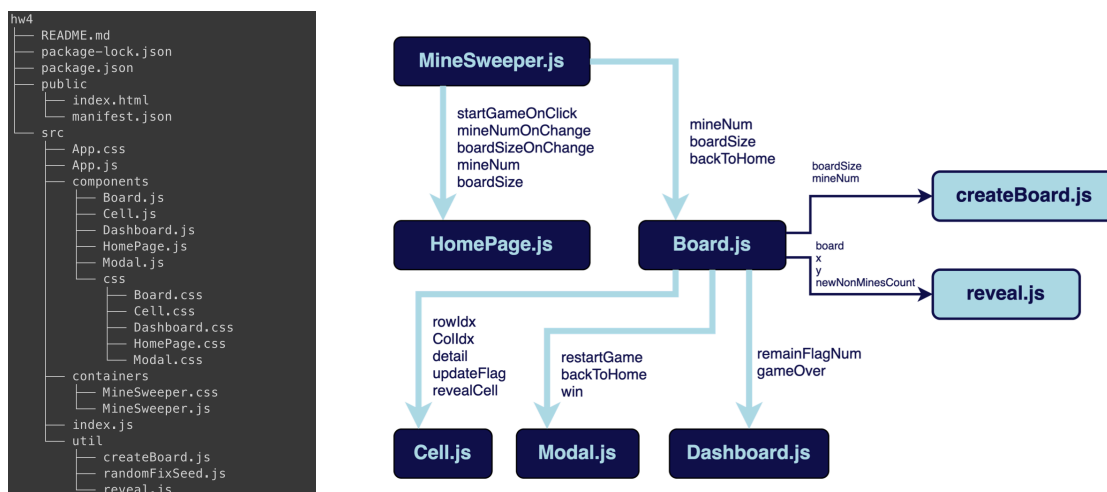
React App Development — MineSweeper

Due : 9pm, Monday, 2022-10-17

Prepared by TA: Cheng-Hua Lu

作業規定與建議步驟：

1. 這個作業的宗旨是訓練大家看懂一個稍大一點架構的React App，並使用 React.js 及 Hook 的技巧實作一個如封面所展現的踩地雷遊戲，使用者可以在開始畫面調整地雷數量、地圖大小來調整遊戲難度，並在按下Start Game後開始遊戲。請先參考這個 [MineSweeper demo 影片](#)，詳細規格如下所述。
2. 這個作業規定一定要用 **ReactJS** 來寫，**強烈建議練習使用 Hooks**，但不能使用參考程式碼沒有使用到的第三方 **JS frameworks/libraries/modules**。沒有按照此規定者，請 **reviewer** 把它判定為屍體。
3. 你可以先把“hw4”這個目錄移除，然後直接下載 [hw4.tgz](#) 並解壓縮，再將hw4整個 folder 搬到 wp1111 底下。檔案架構如下圖。

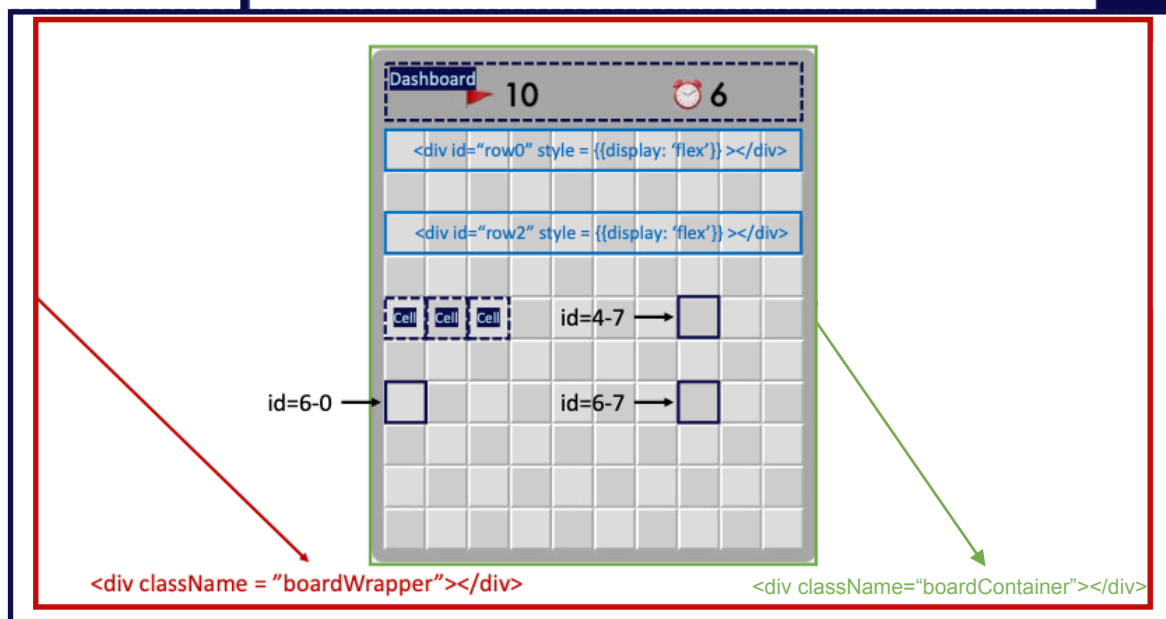


4. 我們有提供參考的 CSS 檔案們 (***.css)，建議大家使用這裡頭定義好的 class names 以及 ids, 以確保大家 Minesweeper 的 look-and-feel 與題目所示相同。當然，在此之上你如果想要增加功能或者是嘗試更多的 JavaScript or CSS 互動效果，請自行增加。
5. 根據上述 CSS files 所定義出來的 class hierarchy 如下圖所示：

HomePage.js & HomePage.css



Board



- 一開始會出現全空白的畫面，要在 `MineSweeper.js` 裡面呼叫 `<HomePage/>`，讓開始畫面 (`HomePage`) 能夠顯示出來。
- `HomePage` 的初始畫面只有 `MineSweeper` 這個標題，要在 `HomePage.js` 實作一個 `content` 為 “Start Game” 的 `button`。同時，這個 `button` 要可以在按下時，開始遊戲。請比對 `HomePage.js` 和 `MineSweeper.js` 裡面相對應的部分，完成 `start button` 的功能。
- 在按下 `start button` 後，應該要切換到遊戲頁面 (`Board`)，因此要在 `MineSweeper.js` 裡面加上顯示 `HomePage` 和 `Board` 的條件 (`startGame`)，並注意將 `Board` 需要的參數傳進去 (ex: `boardSize`, ...)。
- `Board` 的初始畫面有一行 “This is the board Page!”，請在實作 **Board** 時把它刪掉，此行僅供測試使用，成品內不應出現。
- 請完成 `freshBoard`。這個 `function` 的功能是在設定一個 `Board` 的各種參數，`createBoard` 已經幫大家寫好了，請閱讀理解 `createBoard.js` 隨機產生一個 `Board` 的方法，並根據 `createBoard` 回傳的 `Objects` 設定 `Board` 的基本參數，請參考 `Board.js` 內第19行到24行的 `Hook useState functions`。
- 完成好 `freshBoard`，請根據上圖的架構建構 `Board`，強烈建議依照上圖的架構以及提供給大家的 `class names`。提示：這個項目請大家練習使用 `map` 這個 `function`，若對 `map` 不熟悉，請 google “nested map for ReactJS”。
- 在 `Board` 上加上可以插旗子的功能 (`updateFlag`)，在遊戲中，可以透過按右鍵插旗子，`x`, `y` 為要插上旗子的 `board` 座標，請判斷以下幾種情況：

- ① 若此 Cell 已經被按開 (revealed)，則不能再插上旗子

② 若此 Cell 沒有被插上旗子，且沒有被按開，可以插上旗子

③ 若此 Cell 已經被插上旗子，必須拔掉旗子。

注意：請記得更新 remainFlagNum 的情況，並於 function 最後複寫 board 和 remainFlagNum 的資訊。

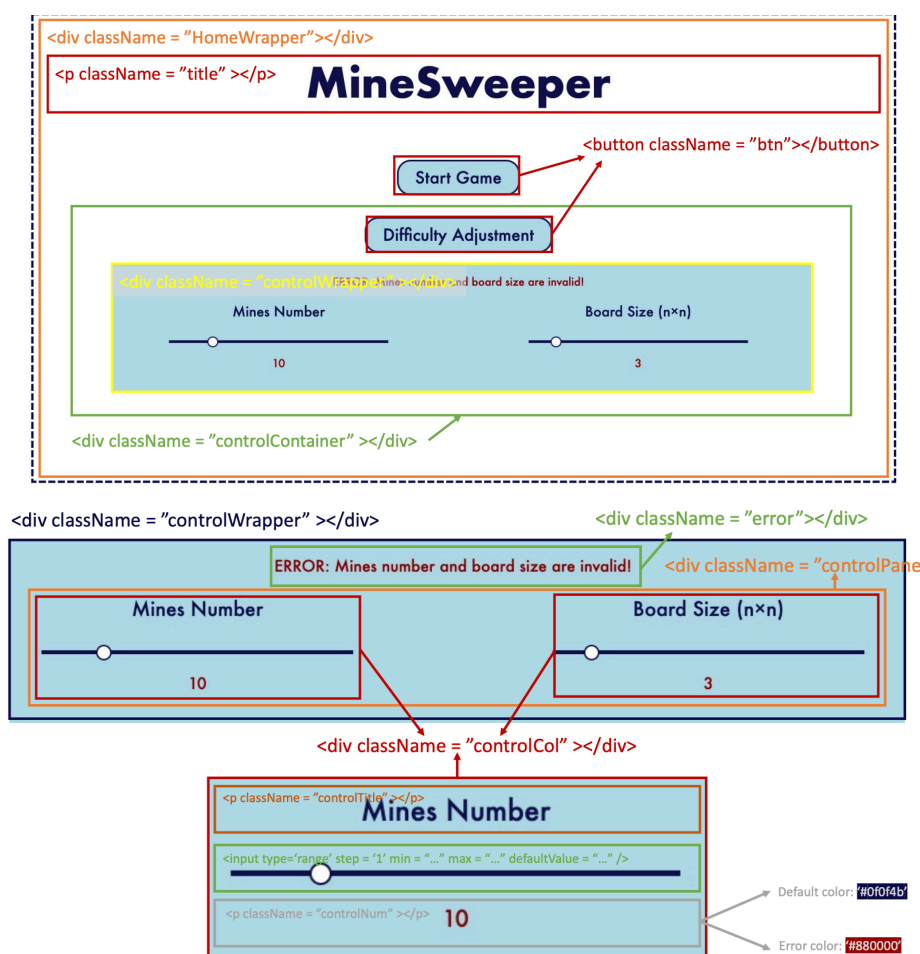
- 完成 Board.js 裡的 revealCell function。判斷兩種情況：

① 若正要被reveal 的這個 Cell 裡面是地雷，則...

② 若並不是地雷，則...

7. Minesweeper 的進階要求功能如下：

- 在 Minesweeper.js 與 HomePage.js 裡面完成難度調整的功能 `<div className = "controlContainer"></div>`，結構如下圖。在按下 "Difficulty Adjustment" 這個 `<button className="btn"></button>` 後，下面的 `<div className = "controlWrapper"></div>` 才會被顯示，若再按一下則會隱藏。在這個 controlWrapper 裡面會有兩個 `<div className = "controlCol"></div>`，而每個 controlCol 內會有 `<p className = "controlTitle"></p>`，`<input type="range" step = "1"/>` 和 `<p className=controlNum"></p>`；地雷數量不能超過 Cell 總數 ($n \times n$)，如果超過的話，`<div className="error"></div>` 才會被顯示，且 controlNum 也都會顯示成 #880000，與此同時 Start Button會沒辦法按 (難度程度不合法，因此遊戲無法開始)。



- 在 reveal.js 裡實作聰明的 reveal function，default 的版本是只會開啟按到的 Cell，但聰明的版本會根據 Board 內的數值有所不同。舉個例子：若要 reveal 的 Cell 數值為 0，則會同步開啟這個 Cell 四周未被插旗子且不是地雷的 Cell。（其他的 condition 就有待有興趣的同學自己發現了！）
- 在 Modal.js 中實作 Modal，並在 Board.js 中顯示它。Modal 出現的時機有兩個：(1) 所有非地雷的 Cell 都被 reveal (2) 任一個地雷的 Cell 被 reveal。在以上兩個時機時，遊戲結束，此時 Modal 會出現，如果前述情況 (1) 會顯示“WIN”，若是情況 (2) 則會顯示如下圖所示的“Game Over”。同時會出現兩個 `<div className = "modalBtn"></div>`，Back to Home 這個 button 在按下去後，畫面便回到 HomePage；如果是按下 Try again / New Game 這個 button，則會重新產生一個新的 Board，繼續遊戲。

Modal.js & Modal.css



- 在 Dashboard.js 中實作計時器。每個新遊戲皆從 0 秒開始計時，當遊戲結束時，計時器會停止在遊戲結束的秒數。
- 歡迎自行增加一些原本作業規定沒有、或者是不一樣的功能，但請在 README.md 中說明。
 - 請注意，你的 hw4 目錄底下應該會有一個叫做“.gitignore”的檔案，定義了「不會」被 push 到 GitHub 上面的檔案，請勿刪除，這樣你才不會把一些不必要的檔案，如 "node_modules" 等也 push 到 GitHub。此外，如果你 hw4 資料夾中還有其他的檔案是沒有需要 push 到 GitHub 上去的，請記得把他們的檔名或目錄也加到 .gitignore 去。
 - 承上，當你在 review 別人的作業的時候，由於你 pull 下來的檔案不會有那些定義在 .gitignore 的檔案，你直接 “yarn start” 是不會 work 的 (請不要因此把別人的作業判斷成屍體)！請先輸入 “yarn” or “yarn install”，它會根據 “package.json” 把一些檔案下載並安裝好，然後再來 “yarn start”。

11. 請注意，這只是個純前端的 App, 你不用把 MineSweeper 的結果存在資料庫或是檔案裡頭。換句話說，當你重新整理頁面後，MineSweeper 就會重新啟動。
12. 再次聲明，practice homework 不會有詳細的規格，「定義規格」也是 practice 的一部分，只要 make sure 你交上來的成品不要讓別人覺得是「屍體」就好，請不用再來問我們「這樣可不可以」、「這個要不要做」之類的問題了。

繳交

請在 deadline (9pm, Monday, 2022-10-17) 之前把你的作業 push 到你 GitHub 上 wp1111 這個 repo 上面 (Note: 我們不接受遲交哦！)。你應該會用到的指令有 `git add hw4`, `git commit -m "你想留的 comment"`, 以及 `git push`. 你也可以用 `git status` 來檢查一下檔案是否都已經成功 push 到 GitHub 上面，或者是登入 GitHub 去檢查。

如有任何的問題，請到 FB 社團的 "HW#4 Q&As" 貼文底下發問，或者是寫信到老師/助教信箱：eewebprogramming@googlegroups.com。
