|  |  |
| --- | --- |
|  |  |
|  | *//==============================================================* **const** Status={  **unknown**:0,  **empty**:1,  **mine**:2 };  //---------------- Презентаційний компонент Кнопка, що відображається у таблиці ----- **const** *GameButton*=(props)=> {  **const** marks=[**'-'**,**'0'**,**'X'**];  **if**(props.**status**==Status.**unknown**)  **return** (<**button onClick=**{()=>props.**onClick**(props.row,props.**col**)}> {marks[props.**status**]} </**button**>);  **else  return** <**button**>{(marks[props.**status**])}</**button**> }  //---------------- Презентаційний компонент Таблиця (призначення – відображення таблиці з кнопками) ----------- **const** *GameTable*=(props)=>{  **return** (  <**table**>  {[...Array(props.**rowCount**)].map((el,i)=>(  <**tr id=**{i}>  {[...Array(props.**colCount**)].map((el,j)=>(  <**td id=**{j}>  <**GameButton status=**{props.**gameTable**[i][j]} **onClick=**{props.**onClick**} **row=**{i} **col=**{j}/>  </**td**>  )   )}  </**tr**>  )  )}  </**table**>  ) }  //---------------- Компонент гра «Простий мінер» ------------------- **class** GameMine **extends** Component{  constructor(props)  {  **super**(props);  **this**.**rowCount**=**this**.**props**.**rowCount** || 5;  **this**.**colCount**=**this**.**props**.**colCount** || 5;  **this**.**gameField**=**this**.createTable(**this**.**rowCount**,**this**.**colCount**,**true**);  **this**.onClick=**this**.onClick.bind(**this**);  **if**(**localStorage**.getItem(**"GameMines"**)) {//🡨 Перевіряємо, чи є історія гри у **localStorage**  //------ Відновлюємо дані з **localStorage** ----------  **this**.**hystory** = ***JSON***.parse(**localStorage**.getItem(**"GameMines"**));  **this**.**state** = {  **gameTable**: **this**.**hystory**[**this**.**hystory**.**length**-1],  }  }  **else** {  //------- Створюємо нову гру ---------  **this**.newGame();  };  }  //------------------ Функція, для ініціалізації нової гри ------------  newGame(){  **var** table = **this**.createTable(**this**.**rowCount**, **this**.**colCount**);  **this**.**hystory**=[table]; //У **this**.**hystory** зберігаємо таблиці, що містять кроки користувача   **this**.**state** = {  **gameTable**: table,  };  **localStorage**.removeItem(**"GameMines"**);   }  //----------------- Метод, який викликається при натисканні на кнопку «New game»  onNewGame(){  **this**.newGame();  **this**.setState ( {  **gameTable**:**this**.**hystory**[**this**.**hystory**.**length**-1],  });   }  //------------------ Метод, який передається у *GameButton,* і викликається при натисканні на кнопку у таблиці *--*  onClick(row,col){  **var** table= **this**.**hystory**[**this**.**hystory**.**length**-1];  **let** copy=**this**.copyTable(table);   copy[row][col]=**this**.**gameField**[row][col];  **this**.**hystory**.push(copy);     **this**.setState(  {  **gameTable**:copy  }  )  **localStorage**.setItem(**"GameMines"**, ***JSON***.stringify(**this**.**hystory**)); //🡨 Зберігаємо історію у **localStorage**  }  //----------------- Метод для створення копії таблиці ----------------------  copyTable(table)  {  **return** table.map(  (row)=>(  row.map(  (el)=>el  )  )  )  }  // Метод для створення таблиці з нулями (якщо random==false), або з випадковими числами(1-немає міни, 2- є міна)  createTable(rowCount,colCount,random)  {  **return** [...Array(rowCount)].map(()=>[...Array(colCount)].map( ()=>(random ? 1+***Math***.floor( ***Math***.random()\*(Object.keys(Status).**length**-1)):0) ));  }  render(){  **return** (  <**React.Fragment**>  <**GameTable rowCount=**{**this**.**rowCount**} **colCount=**{**this**.**colCount**} **gameTable=**{**this**.**state**.**gameTable**} **onClick=**{**this**.onClick}/>  <**div**>  <**button onClick=**{**this**.onNewGame.bind(**this**)}>New game</**button**>  //---------------- Відображаємо кнопки з кроками --------------  {**this**.**hystory**.map(  (table,i)=>  //------ При натисненні на кнопку у **state** буде завантажено відповідну таблицю **gameTable**  <**button key=**{i} **onClick=**{()=>**this**.setState(**this**.setState({**gameTable**:table}))}>Step {i}</**button**>  )}  </**div**>  </**React.Fragment**>  )  } } |