

==============================================================

import React from 'react';

import logo from './logo.svg';

import './App.css';

import Stavki from './components/stavki'

function App() {

return (

<div className="App">

<Stavki numbersCount={10}/>

</div>

);

}

export default App;

=============================================================

import React, { Component } from 'react';

class Stavki extends Component {

constructor(props) {

super(props);

const { userNumbers, compNumbers } = this.getStartNumbersSets()

this.state = {

history: [],

gameState: {

prevSteps: [],

userNumbers,

compNumbers,

userScore: 0

}

}

}

getStartNumbersSets() {

const { numbersCount } = this.props

let userNumbers = []

let compNumbers = []

for (let i = 1; i <= numbersCount; i++) {

compNumbers.push(i)

}

for (let i = 0; i < numbersCount / 2; i++) {

let randIndex = Math.floor(Math.random() \* compNumbers.length)

userNumbers.push(compNumbers[randIndex])

compNumbers.splice(randIndex, 1)

}

return { userNumbers, compNumbers }

}

getCompNumIndex() {

const { compNumbers } = this.state.gameState

let randIndex = Math.floor(Math.random() \* compNumbers.length)

return randIndex

}

makeStep = () => {

const { compNumbers,userNumbers, userScore, prevSteps } = this.state.gameState

const userNum = parseInt(this.refs.userNum.value)

const compNum = compNumbers[this.getCompNumIndex()]

let userScore1=userScore

if (userNum > compNum)

userScore1++

//---------------

const { history } = this.state

const { gameState } = this.state

const copyGameState = { ...gameState, prevSteps: [...prevSteps] }

this.setState({

history: [...history, copyGameState],

gameState: {

prevSteps:[...prevSteps,{compNum,userNum}],

userNumbers: userNumbers.filter(el => el != userNum),

compNumbers: compNumbers.filter(el => el != compNum),

userScore:userScore1

}

})

}

backToStep = (stepNum) => {

const {history}=this.state

this.setState({

gameState:history[stepNum]

})

}

render() {

const { history } = this.state

const { userNumbers, prevSteps,userScore } = this.state.gameState

return (

<>

<div>

User numbers:

{userNumbers.map((el) => String(el)).join('-')}

</div>

<div>

Previous steps:

{prevSteps.map((step, i) => (

<p key={i}>{step.compNum} - {step.userNum}</p>

))}

</div>

<div>

Ваша ставка:

<select ref='userNum'>

{userNumbers.map((num, i) => (

<option key={i} value={num}>{num}</option>

))}

</select>

<button onClick={this.makeStep}>

Зробити хід

</button>

</div>

<hr />

<div>

{history.map((histStep, i) => (

<button key={i} onClick={() => this.backToStep(i)}>

Step {i + 1}

</button>

))}

</div>

<hr/>

Кількість балів: {userScore}

</>

);

}

}

export default Stavki;