import React from "https://cdn.skypack.dev/react"

import ReactDOM from "https://cdn.skypack.dev/react-dom"

import { Provider, connect } from "https://cdn.skypack.dev/react-redux"

import { createStore, combineReducers, applyMiddleware } from "https://cdn.skypack.dev/redux"

import PropTypes from "https://cdn.skypack.dev/prop-types"

import redux from "https://cdn.skypack.dev/redux"

const keyClick = 'keyClick';

const buttons = [

{

id: 'seven',

value: 7

},

{

id: 'eight',

value: 8

},

{

id: 'nine',

value: 9

},

{

id: 'divide',

value: '/'

},

{

id: 'four',

value: 4

},

{

id: 'five',

value: 5

},

{

id: 'six',

value: 6

},

{

id: 'multiply',

value: '\*'

},

{

id: 'one',

value: 1

},

{

id: 'two',

value: 2

},

{

id: 'three',

value: 3

},

{

id: 'add',

value: '+'

},

{

id: 'zero',

value: 0

},

{

id: 'dot',

value: '.'

},

{

id: 'equals',

value: '='

},

{

id: 'substract',

value: '-'

},

{

id: 'clear',

value: 'CE'

}

];

const keyClicked = (key) => {

return {

type: keyClick,

key: key.value

}

};

function isOperator(key) {

if (key == '-' || key == '+' || key == '\*' || key == '/') return true;

return false;

}

function isNumber(key) {

if (key == '0' || key == '1' || key == '2' || key == '3' || key == '4' || key == '5' || key == '6' || key == '7' || key == '8' || key == '9') return true;

return false;

}

function isNegativePermitted(operations, key) {

if ((operations[operations.length - 1] == '/' || operations[operations.length - 1] == '\*') && key == '-') return true;

return false;

}

function twoOperators(operations, key) {

if (isOperator(operations[operations.length - 1]) && isOperator(key)) return true;

return false;

}

function twoDots(operations, key) {

let twodots = false;

if (key == '.' && operations.includes("."))

twodots = true;

return twodots;

}

function isOperatorAfterMinus(operations, key) {

return (operations[operations.length - 1] == '-' && isOperator(key));

}

function agregarParentesis(operations) {

let primeraParte = [], segundaParte = [], resultado = operations;

if (operations.length >=3 && !isOperator(operations[operations.length - 1]) && operations[operations.length - 2] == '-' && (operations[operations.length - 3] == '/' || operations[operations.length - 3] == '\*')) {

primeraParte = operations.splice(0, operations.length - 2);

segundaParte = operations.splice(operations.length - 2, 2);

resultado = primeraParte.concat('(').concat(segundaParte).concat(')');

}

return resultado;

}

function continuarParentesis(operations, key) {

let primeraParte = [], segundaParte = [], resultado = operations;

if (operations.length >= 2 && operations[operations.length - 1] == ')' && isNumber(key)) {

primeraParte = operations.splice(0, operations.length - 1);

resultado = primeraParte.concat(key).concat(')');

}

return resultado;

}

function processOperations(operations, key) {

let resultado = [...operations];

if (key == '=') {

let res = eval(resultado. join(""));

const customEvent = new CustomEvent('imprimir', { detail: { text: res }});

document.dispatchEvent(customEvent);

resultado = [res];

}

else {

resultado = continuarParentesis(resultado, key);

if (resultado.length == operations.length) { //sólo entra si no se agregaron elementos al paréntesis

if (key == 'CE')

resultado = [0];

else

if (resultado.length == 1 && (resultado.includes(0) || (resultado.includes('0'))))

resultado = [key];

else

if (twoDots(resultado, key))

new Audio('https://media.geeksforgeeks.org/wp-content/uploads/20190531135120/beep.mp3').play();

else

if (twoOperators(resultado, key) && (!isNegativePermitted(resultado, key)) && !isOperatorAfterMinus(operations, key))

resultado[resultado.length - 1] = key;

else

if (!isOperatorAfterMinus(resultado, key))

resultado = resultado.concat(key);

}

resultado = agregarParentesis(resultado); //transformo \*-5 en \*(-5)

const customEvent = new CustomEvent('imprimir', { detail: { text: resultado. join("") }});

document.dispatchEvent(customEvent);

}

return resultado;

}

const opsReducer = (state = ['0'], action) => {

switch (action.type) {

case keyClick:

//console.log(eval("2 + 6"));

return processOperations(state, action.key);

default:

return state;

}

};

class Tecla extends React.Component {

constructor(props) {

super(props);

this.state = {

tecla: props.tecla,

}

}

clickTecla() {

this.props.submitKeyClicked(this.props.tecla, this.props.operations);

}

render() {

return (

<div>

<button id={this.state.tecla.id} onClick={this.clickTecla.bind(this)}>{this.state.tecla.value}</button>

</div>

);

}

}

class Display extends React.Component {

constructor(props) {

super(props);

this.state = {

text: '0',

}

document.addEventListener('imprimir', (texto) => {this.imprimirPantalla(texto)});

}

imprimirPantalla(texto) {

this.state.text = texto.detail.text;

this.forceUpdate();

}

render() {

return (

<div>

<label id="display">{this.state.text}</label>

</div>

);

}

}

class Panel extends React.Component {

constructor(props) {

super(props);

this.state = {

//operations: [],

}

}

render() {

return (

<div id="container">

<h1> Gorozito's Calculator </h1>

<Display />

<div id="drum-machine">

{buttons.map( (tecla, idx) => {

return <Tecla submitKeyClicked = {this.props.submitKeyClicked} operations = {this.props.operations} tecla={tecla} onClick={() => {this.togglePlay()}}/>;})

}

</div>

</div>

);

}

};

const mapStateToProps = (state) => {

return {

operations: state,

}

};

const mapDispatchToProps = (dispatch) => {

return {

submitKeyClicked: (key, operations) => {

dispatch(keyClicked(key, operations))

}

}

};

const store = createStore(opsReducer);

const Container = connect(mapStateToProps, mapDispatchToProps)(Panel);

class AppWrapper extends React.Component {

render() {

return (

<Provider store={store}>

<Container/>

</Provider>

);

}

};

ReactDOM.render(<AppWrapper/>, document.getElementById('root'));