Zoha Ahmed

213901

BSCS7A

Code

App.js

import React, { Component } from 'react';

import { View, Text, StyleSheet } from 'react-native';

import Button from './Button';

const inputButtons = [

[0, 'clear', '=', '\*'],

[1, 2, 3, '+'],

[4, 5, 6, '-'],

[7, 8, 9, '/']

];

export default class App extends Component {

constructor(props) {

super(props);

this.initialState = {

prevState: 0,

inp: 0,

selected: null,

};

this.state = this.initialState;

}

render() {

return (

<View style={Style.rootContainer}>

<View style={Style.displayContainer}>

<Text style={Style.displayText}>{this.state.inp}</Text>

</View>

<View style={Style.inputContainer}>{this.\_renderInputButtons()}</View>

</View>

);

}

\_renderInputButtons() {

let views = inputButtons.map((row, idx) => {

let inputRow = row.map((buttonVal, columnIdx) => {

return (

<Button

value={buttonVal}

highlight={this.state.selected === buttonVal}

onPress={this.\_onInputButtonPressed.bind(this, buttonVal)}

key={'butt-' + columnIdx}

/>

);

});

return (

<View style={Style.inputRow} key={'row-' + idx}>

{inputRow}

</View>

);

});

return views;

}

\_onInputButtonPressed(input) {

switch (typeof input) {

case 'number':

return this.\_handleNumberInput(input);

default:

return this.\_handleStringInput(input);

}

}

\_handleNumberInput(num) {

let inp = this.state.inp \* 10 + num;

this.setState({

inp: inp,

});

}

\_handleStringInput(str) {

switch (str) {

case '/':

case '\*':

case '+':

case '-':

this.setState({

selected: str,

prevState: this.state.inp,

inp: 0,

});

break;

case '=':

let symbol = this.state.selected,

inp = this.state.inp,

prevState = this.state.prevState;

if (!symbol) {

return;

}

this.setState({

prevState: 0,

inp: eval(prevState + symbol + inp),

selected: null,

});

break;

case 'ce':

this.setState(this.initialState);

break;

case 'c':

this.setState({ inp: 0 });

break;

}

}

}

const Style = StyleSheet.create({

rootContainer: {

flex: 1,

},

displayContainer: {

flex: 2,

backgroundColor: '#BDEAF5',

justifyContent: 'center',

},

displayText: {

color: 'black',

fontSize: 38,

fontWeight: 'bold',

textAlign: 'right',

padding: 20,

},

inputContainer: {

flex: 8,

backgroundColor: '#D4BDF5',

},

inputRow: {

flex: 1,

flexDirection: 'row',

},

});

Button.js

import React, { Component } from 'react';

import {

TouchableHighlight,

Text,

StyleSheet

} from 'react-native';

export default class button extends Component {

render() {

return (

<TouchableHighlight style={[Style.inputButton, this.props.highlight ? Style.inputButtonHighlighted : null]}

underlayColor="#193441"

onPress={this.props.onPress}>

<Text style={Style.inputButtonText}>{this.props.value}</Text>

</TouchableHighlight>

)

}

}

var Style = StyleSheet.create({

inputButton: {

flex: 1,

alignItems: 'center',

justifyContent: 'center',

borderWidth: 0.5,

borderColor: '#91AA9D',

},

inputButtonHighlighted: {

backgroundColor: '#193441',

},

inputButtonText: {

fontSize: 22,

fontWeight: 'bold',

color: 'black',

},

});

Output

