

Component, Styling & Layout

1. Creating a customer header component

<https://github.com/arkarhtetmyint/rn-complete-guide2>

```
// Header.js
import React from "react";
import { View, Text, StyleSheet } from "react-native";

const Header = (props) => {
  return (
    <View style={styles.header}>
      <Text style={styles.headerTitle}>{props.title}</Text>
    </View>
  );
};

const styles = StyleSheet.create({
  header: {
    width: '100%',
    height: 90,
    paddingTop: 36,
    backgroundColor: '#f7287b',
    alignItems: 'center',
    justifyContent: 'center'
  },
  headerTitle: {
    color: 'black',
    fontSize: 18,
  },
});

export default Header;
```

```
// App.js

import { StatusBar } from "expo-status-bar";
import React from "react";
import { StyleSheet, Text, View } from "react-native";
import Header from "../components/Header";

export default function App() {
  return (
    <View style={styles.screen}>
      <Header title="Guess a Number"/>
    </View>
  );
}

const styles = StyleSheet.create({
  screen: {
    flex: 1
  }
});
```

2. Adding Game Screen

```
// StartGameScreen.js'

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

const StartGameScreen = (props) => {
  return <View style={styles.screen}>
    <Text>The Game Screen</Text>
  </View>;
};

const styles = StyleSheet.create({
  screen: {
    flex: 1,
    padding: 10,
    alignItems: "center",
  },
});

export default StartGameScreen;
```

```
// App.js

import { StatusBar } from "expo-status-bar";
import React from "react";
import { StyleSheet, Text, View } from "react-native";
import Header from "../components/Header";
import StartGameScreen from "../screens/StartGameScreen";

export default function App() {
  return (
    <View style={styles.screen}>
      <Header title="Guess a Number"/>
      <StartGameScreen />
    </View>
  );
}

const styles = StyleSheet.create({
  screen: {
    flex: 1
  }
});
```



3. Modification StartGameScreen.js

```
import React from "react";
import { View, Text, StyleSheet, TextInput, Button } from "react-native";

const StartGameScreen = (props) => {
  return (
    <View style={styles.screen}>
      <Text style={styles.title}>Start a New Game!</Text>
      <View style={styles.inputContainer}>
        <Text>Select a Number</Text>
        <TextInput />
        <View style={styles.buttonContainer}>
          <Button title="Reset" onPress={() => {}}/>
          <Button title="Confrim" onPress={() => {}}/>
        </View>
      </View>
    </View>
  );
};

const styles = StyleSheet.create({
  screen: {
    flex: 1,
    padding: 10,
    alignItems: "center",
  },
  buttonContainer: {
    flexDirection: 'row',
    width: '100%',
    justifyContent: 'space-between',
    paddingHorizontal: 15
  },
  inputContainer: {
    width: 300,
    maxWidth: '80%',
    alignItems: 'center',
    shadowColor: 'black',
    shadowOffset: {width: 0, height: 2},
    shadowRadius: 6,
```

```

        shadowOpacity:0.26,
        backgroundColor:'white',
        elevation:8,
        padding:10,
        borderRadius:10
    },

    title:{
        fontSize:20,
        marginVertical:10,
    },
  },
});

export default StartGameScreen;

```

4. Add Card componet

```

// Card.js
import React from "react";
import {View, StyleSheet} from 'react-native';

const Card = props => {
  return <View style={{...styles.card,...props.style}}>{props.children}</View>
};

const styles = StyleSheet({
  card:{
    shadowColor:'black',
    shadowOffset:{width:0,height:2},
    shadowRadius:6,
    shadowOpacity:0.26,
    backgroundColor:'white',
    elevation:8,
    padding:10,
    borderRadius:10
  },
});

export default Card;

// modify the StartGameScreen.js
import React from "react";
import { View, Text, StyleSheet, TextInput,Button } from "react-native";
import Card from "../components/Card";

const StartGameScreen = (props) => {
  return (
    <View style={styles.screen}>
      <Text style={styles.title}>Start a New Game!</Text>
      <Card style={styles.inputContainer}>
        <Text>Select a Number</Text>
        <TextInput />
        <View style={styles.buttonContainer}>
          <Button title="Reset" onPress={() => {}}/>
          <Button title="Confrim" onPress={() => {}}/>
        </View>
      </Card>
    </View>
  );
};

```

```
const styles = StyleSheet.create({
  screen: {
    flex: 1,
    padding: 10,
    alignItems: "center",
  },

  buttonContainer:{
    flexDirection:'row',
    width:'100%',
    justifyContent:'space-between',
    paddingHorizontal:15
  },

  inputContainer:{
    width:300,
    maxWidth:'80%',
    alignItems:'center',
  },

  title:{
    fontSize:20,
    marginVertical:10,
  },
});

export default StartGameScreen;
```

5. Color theming

```
// wrap two buttons with view
<View style={styles.button}><Button title="Reset" onPress={() => {}}/></View>
<View style={styles.button}><Button title="Confrim" onPress={() => {}}/></View>

// Set the button width
button:{
  width:100,
}

// create color.js under constants folder
export default {
  primary : '#f7287b',
  accents: '#c717fc',
}

// import the color.js
import Colors from '../constants/color';

// change the button color
<View style={styles.button}><Button title="Reset" onPress={() => {}} color={Colors.accents}/></View>
<View style={styles.button}><Button title="Confrim" onPress={() => {}} color={Colors.primary}/></View>
```

6. Styling Text Input

```
// create input.js
import React from "react";
import { TextInput, StyleSheet } from "react-native";
```

```

const Input = (props) => {
  return <TextInput style={{ ...styles.input, ...props.style }}></TextInput>;
};

const styles = StyleSheet.create({
  input:{
    height:30,
    borderBottomColor:'gray',
    borderBottomWidth:1,
    marginVertical:10,
  },
});

export default Input;

// use <Input/> in StartGameScreen.js
const StartGameScreen = (props) => {
  return (
    <View style={styles.screen}>
      <Text style={styles.title}>Start a New Game!</Text>
      <Card style={styles.inputContainer}>
        <Text>Select a Number</Text>
        <Input/>
        <View style={styles.buttonContainer}>
          <View style={styles.button}><Button title="Reset" onPress={() => {}} color={Colors.accent}></View>
          <View style={styles.button}><Button title="Confrim" onPress={() => {}} color={Colors.primary}></View>
        </View>
      </Card>
    </View>
  );
};

// added the ...props at input.js
const Input = (props) => {
  return <TextInput {...props} style={{ ...styles.input, ...props.style }}></TextInput>;
};

// add textInput properties at StartGameScreen.js
//Reference
//https://reactnative.dev/docs/textinput
<Input
  style={styles.input}
  blurOnSubmit
  autoCapitalize="none"
  autoCorrect={false}
  keyboardType="number-pad"
  maxLength={2}
/>

```

7. Clearing user Input & controlling the soft keyboard

```

// declar the enter value
const [enteredValue, setEnteredValue] = useState("");

const numberInputHandler = (inputText) => {
  setEnteredValue(inputText.replace(/[^0-9]/g, ""));
};

```

```

    };

    <Input
      style={styles.input}
      blurOnSubmit
      autoCapitalize="none"
      autoCorrect={false}
      keyboardType="number-pad"
      maxLength={2}
      onChangeText={numberInputHandler}
      value={enteredValue}

    />

```

```

// add TouchableWithoutFeedback
// for Keyboard hide

return (
  <TouchableWithoutFeedback
    onPress={() => {
      Keyboard.dismiss();
    }}
  >
    <View style={styles.screen}>
      <Text style={styles.title}>Start a New Game!</Text>
      <Card style={styles.inputContainer}>
        <Text>Select a Number</Text>
        <Input
          style={styles.input}
          blurOnSubmit
          autoCapitalize="none"
          autoCorrect={false}
          keyboardType="number-pad"
          maxLength={2}
          onChangeText={numberInputHandler}
          value={enteredValue}
        />
        <View style={styles.buttonContainer}>
          <View style={styles.button}>
            <Button title="Reset" onPress={() => {}} color={Colors.accent} />
          </View>
          <View style={styles.button}>
            <Button
              title="Confirm"
              onPress={() => {}}
              color={Colors.primary}
            />
          </View>
        </View>
      </Card>
    </View>
  </TouchableWithoutFeedback>
);
};

```

8. Reset user Input

```

// set reset value
const resetInputHandler = () => {
  setEnteredValue("");
};

```

```

<View style={styles.buttonContainer}>
  <View style={styles.button}>
    <Button
      title="Reset"
      onPress={resetInputHandler}
      color={Colors.accent}
    />
  </View>
</View>

```

```

// set confirm stage
const [confirmed, setConfirmed] = useState(false);

// when reset press, set the value false
const resetInputHandler = () => {
  setEnteredValue("");
  setConfirmed(false);
};

```

```

// declare select number
const [selectedNumber, setSelectedNumber] = useState();

const confirmInputHandler = () => {
  const chooseNumber = parseInt(enteredValue);
  if (chooseNumber === NaN || chooseNumber <= 0 || chooseNumber > 99) {
    return;
  }

  setConfirmed(true);
  setSelectedNumber(chooseNumber);
  setEnteredValue("");
};

// display the selectNumber at Card view
let confirmedOutput;
if (confirmed) {
  confirmedOutput = <Text>Chosen Number : {selectedNumber}</Text>;
}

<Card>
  ...
  ...
</Card>
{confirmedOutput}

```

9. Showing Alert

```

// import the alert
import {
  ...
  Alert,
} from "react-native";

const confirmInputHandler = () => {
  const chooseNumber = parseInt(enteredValue);
  if (isNaN(chooseNumber) || chooseNumber <= 0 || chooseNumber > 99) {
    Alert.alert(
      "Invalid number!",
      "Number has to be a number between 1 and 99. ",

```

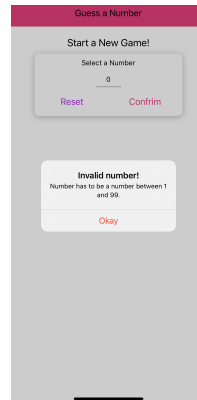


```

        [{text:'Okay',style:'destructive',onPress:resetInputHandler}]
    );
    return;
}

setConfirmed(true);
setSelectedNumber(chooseNumber);
setEnteredValue("");
};

```



10. Confirmation box

```

// create number input NumberContainer.js
import React from "react";
import { View, Text, StyleSheet } from "react-native";

import Colors from "../constants/color";

const NumberContainer = (props) => {
    return (
        <View style={styles.container}>
            <Text style={styles.number}>{props.children}</Text>
        </View>
    );
};

const styles = StyleSheet.create({
    container:{
        borderWidth:2,
        borderColor:Colors.accents,
        padding:10,
        borderRadius:10,
        marginVertical:10,
        alignItems:'center',
        justifyContent:'center',
    },
    number:{
        color:Colors.accents,
        fontSize:22,
    }
});

```

```

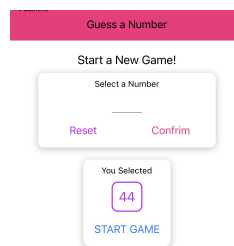
export default NumberContainer;

// implement at StartGameScreen.js
if (confirmed) {
  confirmedOutput = (
    <Card style={styles.summaryContainer}>
      <Text>You Selected</Text>
      <NumberContainer>{selectedNumber}</NumberContainer>
      <Button title="START GAME"/>
    </Card>
  );
}

// align center
summaryContainer: {
  marginTop: 20,
  alignItems: 'center'
},

// dismiss the keyboard when confirm complete
Keyboard.dismiss();

```



11. Adding a random number

```

// create the GameScreen.js

import React, { useState } from "react";
import { View, Text, StyleSheet, Button } from "react-native";

import NumberContainer from "../components/NumberContainer";
import Card from "../components/Card";

const generateRandomBetween = (min, max, exclude) => {
  min = Math.ceil(min);
  max = Math.floor(max);
  const rndNum = Math.random() * (max - min) + min;
  if (rndNum === exclude) {
    return generateRandomBetween(min, max, exclude);
  } else {
    return rndNum;
  }
}

```

```

};

const GameScreen = (props) => {
  const [currentGuess, setCurrentGuess] = useState(
    generateRandomBetween(1, 100, props.userChoice)
  );

  return (
    <View style={styles.screen}>
      <Text>Opponent's Guess</Text>
      <NumberContainer>{currentGuess}</NumberContainer>
      <Card style={styles.buttonContainer}>
        <Button title="LOWER" onPress={() => {}}></Button>
        <Button title="GREATER" onPress={() => {}}></Button>
      </Card>
    </View>
  );
};

const styles = StyleSheet.create({
  screen: {
    flex: 1,
    padding: 10,
    alignItems: "center",
  },
  buttonContainer: {
    flexDirection: "row",
    justifyContent: "space-around",
    marginTop: 20,
    width: 300,
    maxWidth: "80%",
  },
});

export default GameScreen;

```

```

// import GameScreen to App.js

import { StatusBar } from "expo-status-bar";
import React, { useState } from "react";
import { StyleSheet, Text, View } from "react-native";
import Header from "../components/Header";
import StartGameScreen from "../screens/StartGameScreen";
import GameScreen from "../GameScreen";

export default function App() {
  const [userNumber, setUserNumber] = useState();

  const startGameHandler = (selectedNumber) => {
    setUserNumber(selectedNumber);
  };

  let content = <StartGameScreen onStartGame={startGameHandler} />;

  if (userNumber) {
    content = <GameScreen userChoice={userNumber} />;
  }

  return (
    <View style={styles.screen}>
      <Header title="Guess a Number" />
      <StartGameScreen />
      {content}
    </View>
  );
}

```

```
        </View>
      );
    }

    const styles = StyleSheet.create({
      screen: {
        flex: 1,
      },
    });

    // StartGameScreen.js
    <Button title="START GAME" onPress={() => props.onStartGame(selectedNumber)} />
```

12. d

13. d

14. d

15.