ANKARA ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BLM4538 - IOS İle Mobil Uygulama Geliştirme II

Proje Raporu

Taş Kağıt Makas Oyunu

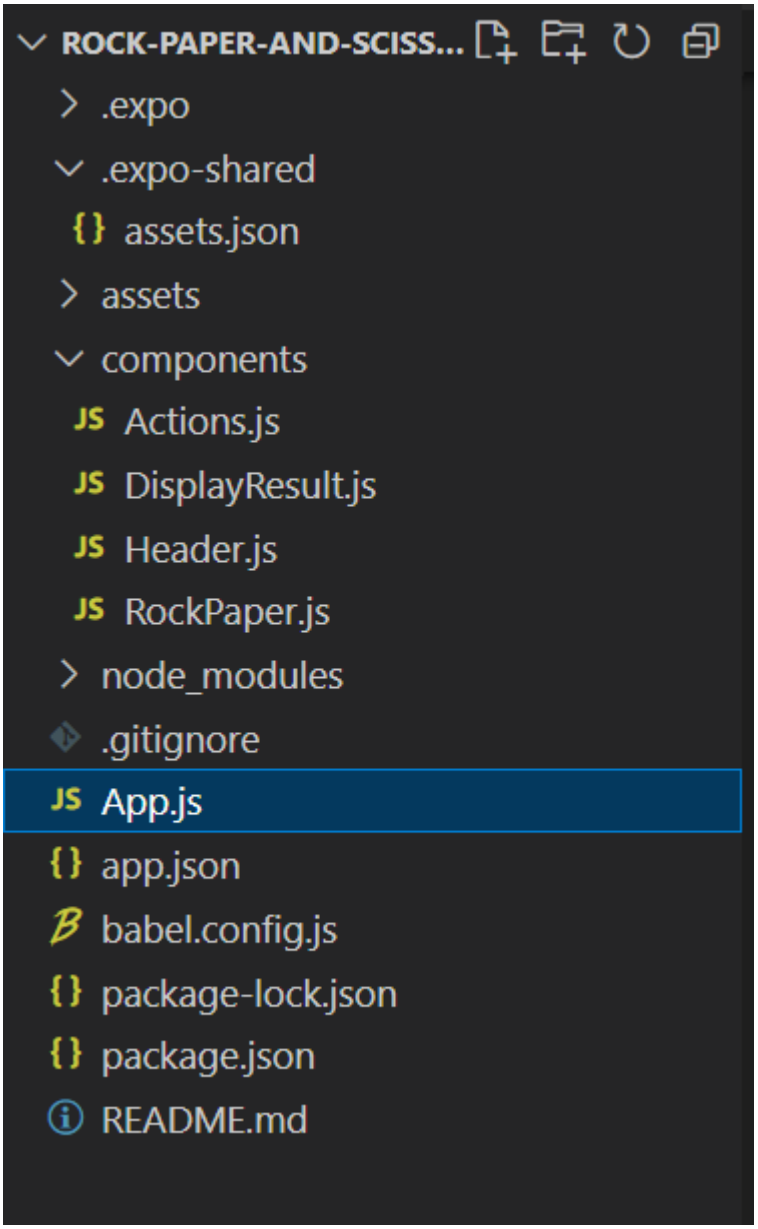
Refik Can ÖZTAŞ

19290266

5/6/2023

BLM4538 için React Native üzerinde Cross-Platform bir Taş Kağıt Makas oyunu yazılmıştır.

Kodların incelenmesi aşağıdadır, oyun içi görüntüler code review kısmından sonradır.
github linki: https://github.com/canoztas/RockPaperScissorsGame



Projede yandaki gibi bir yapı kurulmuştur.

Proje expo üzerinde geliştirilip test edilmiştir. Bu sayede cross-platform olmuştur.

```
e eustache on ■ D:/projects/Rock-Paper-And-Scissors-main
# $env:NODE_OPTIONS="--openssl-legacy-provider"
e eustache on ■ D:/projects/Rock-Paper-And-Scissors-main
# npm start

> rock@1.0.0 start
> cross-env NODE_OPTIONS=--openssl-legacy-provider && expo start

WARNING: The legacy expo-cli does not support Node +17. Migrate to the new local Expo CLI: https://blog.expo.dev/the-new-expo-cli-f4250d8e3421.
› Port 19000 is
√ Use port 19001 instead? ... yes
Starting project at D:\projects\Rock-Paper-And-Scissors-main
Starting Metro Bundler
```

```
› Metro waiting on exp://192.168.1.4:19001
› Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

› Press a │ open Android
› Press w │ open web

› Press r │ reload app
› Press m │ toggle menu

› Press ? │ show all commands

Logs for your project will appear below. Press Ctrl+C to exit.
› Opening on Android ...
```

Kodlar:

```jsx
import * as React from 'react';
import { Appbar } from 'react-native-paper';


const Header = () => {

  return (
    <Appbar.Header style={{backgroundColor:'#a60bff'}}>
      <Appbar.Content title="Rock - Paper - Scissors" color="black" style={{ alignItems: 'center', transform:[{s
    </Appbar.Header>
  );
};

export default Header;
```

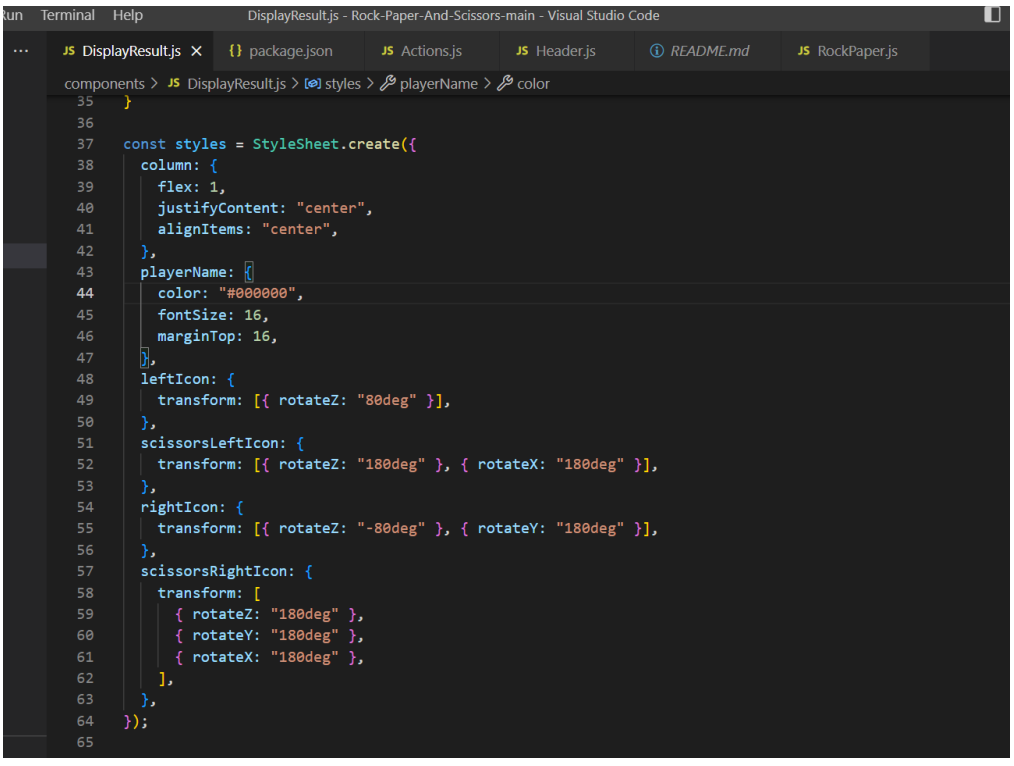
Sayfa headerı

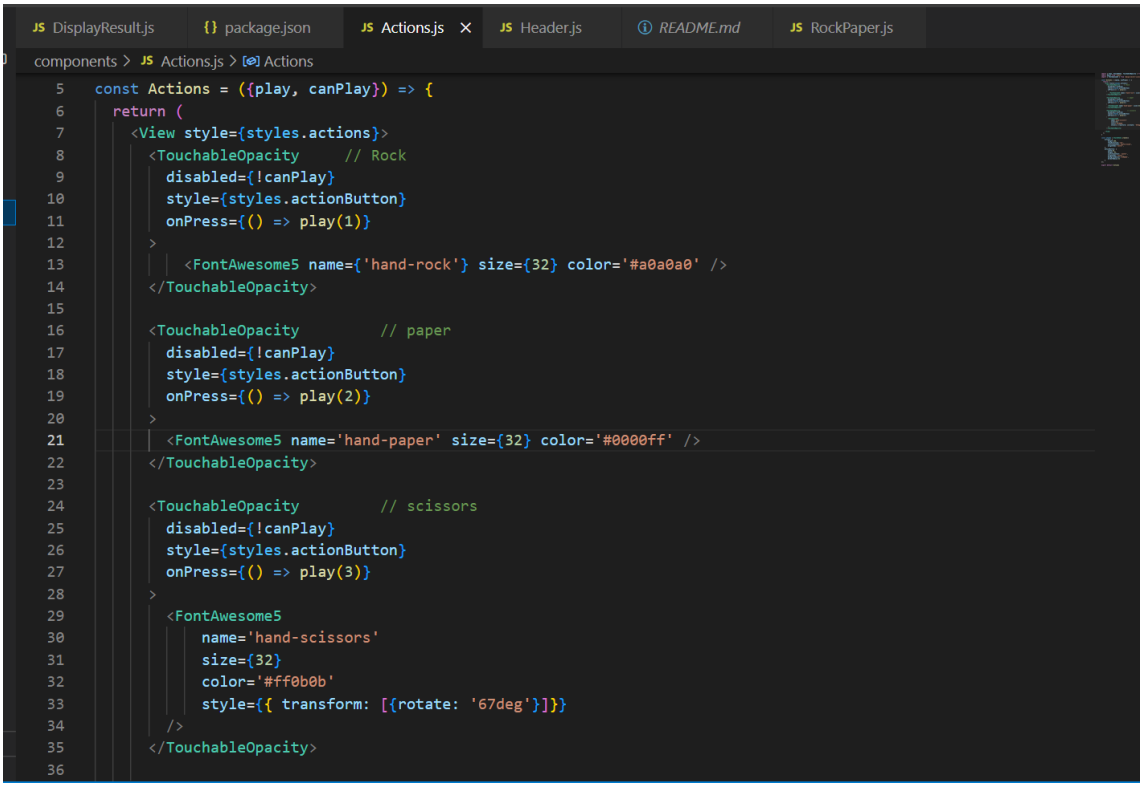
```jsx
import React from "react";
import { StyleSheet, Text, View } from "react-native";
import { FontAwesome5 } from "@expo/vector-icons";

const ICONS = ["hand-rock", "hand-paper", "hand-scissors"];

export default function DisplayResult({ userChoice, computerChoice }) {
  return (
    <>
      <View style={styles.column}>
        <FontAwesome5
          name={ICONS[userChoice - 1]}
          size={64}
          color="#0000ff"
          solid
          style={userChoice === 3 ? styles.scissorsLeftIcon : styles.leftIcon}
        />
        <Text style={styles.playerName}>You</Text>
      </View>

      <View style={styles.column}>
        <FontAwesome5
          name={ICONS[computerChoice - 1]}
          size={64}
          color="#ff0000"
          solid
          style={
            computerChoice === 3 ? styles.scissorsRightIcon : styles.rightIcon
          }
        />
        <Text style={styles.playerName}>Computer</Text>
      </View>
```





Ekrandaki taş kağıt makas sembollerinin ve scoreun ayarlanması

components > JS RockPaper.js > styles > content > backgroundColor

```
import React, { useState, useRef } from 'react';
import { StyleSheet, SafeAreaView, Text, View, Animated } from 'react-native';
import  Constants  from 'expo-constants';
import Actions from './Actions';
import DisplayResult from './DisplayResult';
import Header from './Header';

win=0;
lose=0;
export default function RockPaper(){
    const [userChoice, setUserChoice] = useState(0);
    const [computerChoice, setComputerChoice] = useState(0);
    const [result, setResult] = useState("");
    const [scoreboard, setScoreboard] = useState("");
    const [canPlay, setPlay] = useState(true);

    // For Animation
    const fadeAnimation = useRef(new Animated.Value(1)).current;

    function play(choice){
        // We have 3 choice
        // 1 = rock
        // 2 = paper
        // 3 = scissors

        const randomComputerChoice = Math.floor(Math.random() * 3) + 1;
        let resultString = "";

        if (choice === 1) {
            resultString = randomComputerChoice === 3 ? "WIN" : "LOSE";
        }
```

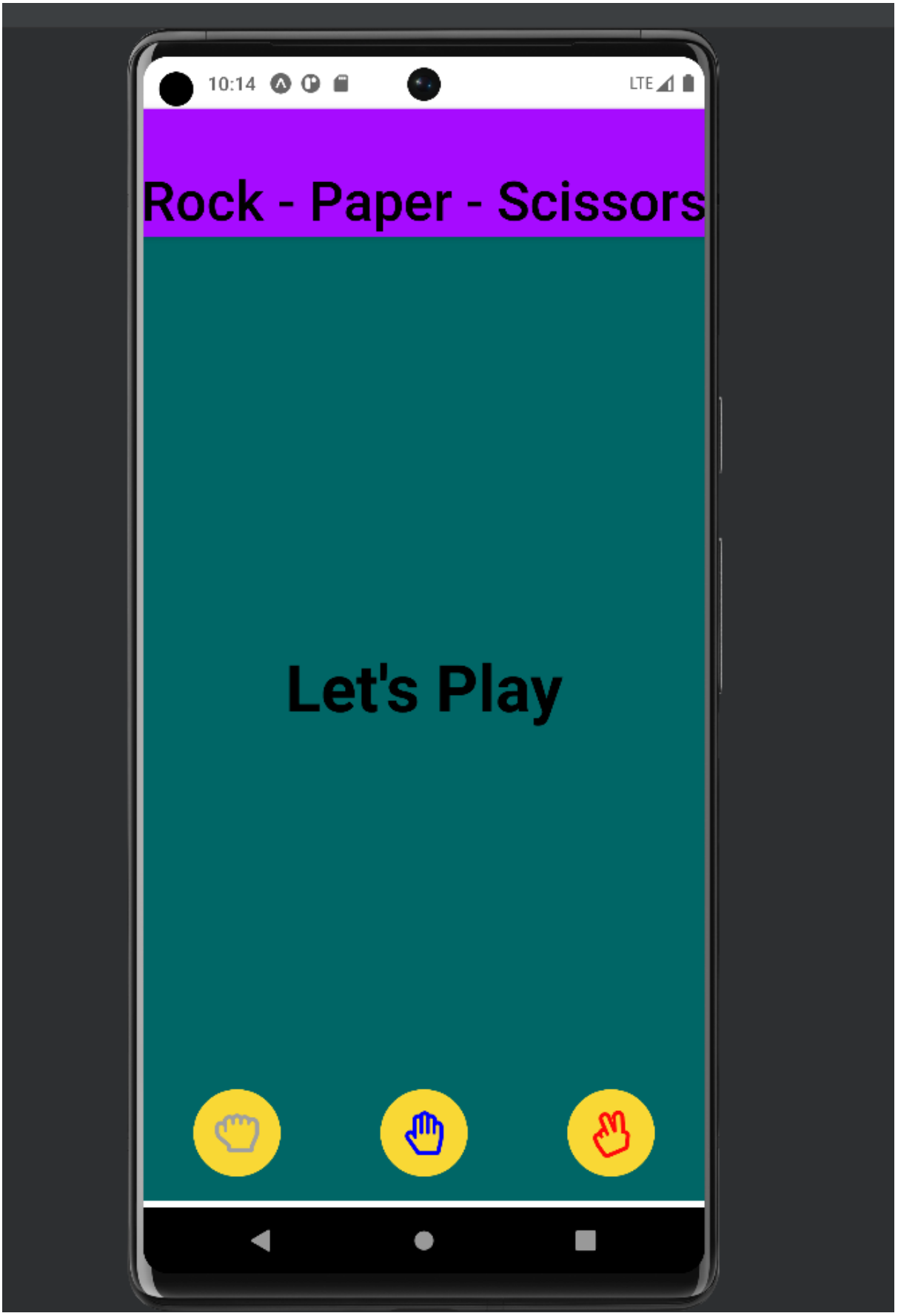
```
35          else {
36              resultString = randomComputerChoice === 2 ? "WIN" : "LOSE";
37          }
38
39          if (choice === randomComputerChoice){
40              resultString = "DRAW";
41          }
42
43          if (resultString=="WIN"){
44          win++;
45          }
46          else if(resultString=="LOSE"){
47          lose++;
48          }
49
50          setUserChoice(choice);
51          setComputerChoice(randomComputerChoice);
52
53          // Wait animation hide old result
54          setTimeout(() => {
55              setResult(resultString);
56              setScoreboard(win + " win / " + lose + " lose");
57          }, 300);
58
59          // Animation hide old result and show new result
60          Animated.sequence([
61              Animated.timing(fadeAnimation, {
62                  toValue:0,
63                  duration:300,
64                  useNativeDriver: true,
65              }),
66              Animated.timing(fadeAnimation, {
```

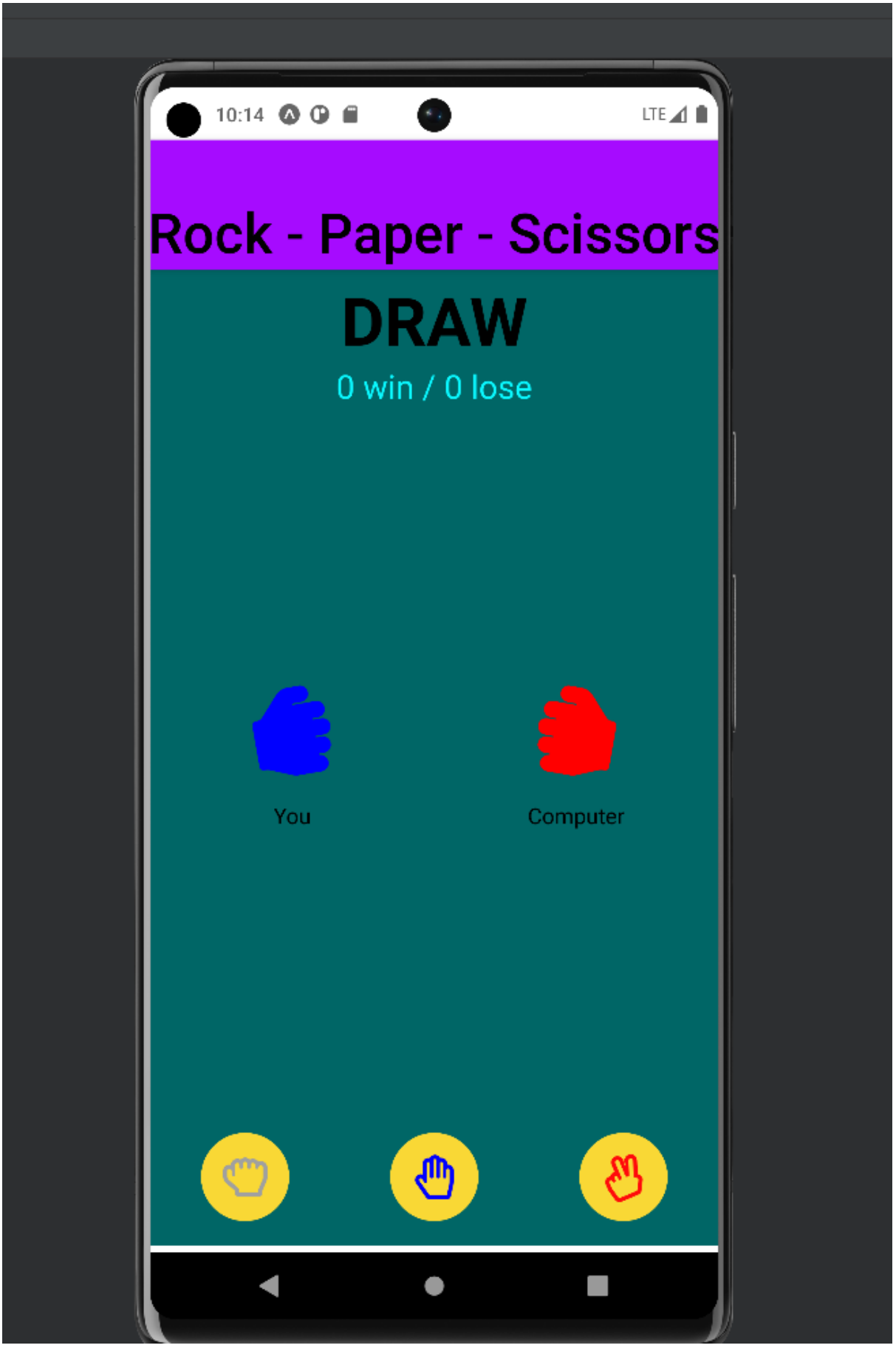
```
    return(
        <SafeAreaView style={styles.container}>
            <Header/>
            <View style={styles.content}>
                <View style={styles.result}>
                    <Animated.Text
                        style={[styles.resultText, {opacity: fadeAnimation}]}
                    >
                        {result}
                    </Animated.Text>
                    <Animated.Text
                        style={[styles.scoreboardText, {opacity: fadeAnimation}]}
                    >
                        {scoreboard}
                    </Animated.Text>
                </View>
                <View style={styles.screen}>
                    {!result ? (
                        <Text style={styles.readyText}>Let's Play</Text>
                    ) : (
                        <DisplayResult
                            userChoice={userChoice}
                            computerChoice={computerChoice}
                        />
                    )}
                </View>
                <Actions play={play} canPlay={canPlay} />
            </View>
        </SafeAreaView>
    );
```

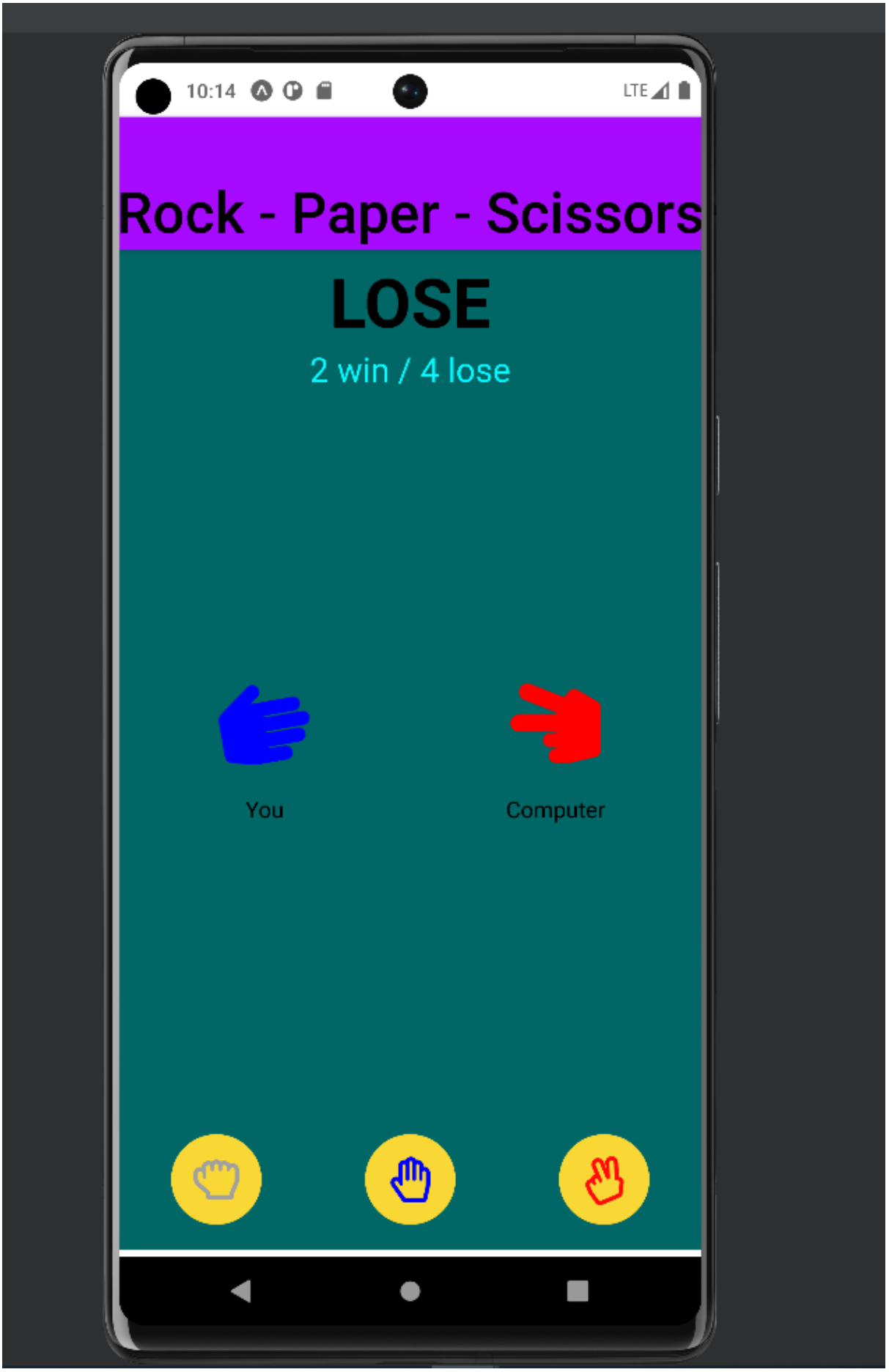
```
12
13  const styles = StyleSheet.create({
14      container: {
15          flex:1,
16          paddingTop: Constants.statusBarHeight,
17      },
18      content: {
19          flex:1,
20          marginBottom:5,
21          backgroundColor:'#006666'
22      },
23      result : {
24          height:100,
25          justifyContent:'flex-end',
26          alignItems: "center",
27      },
28      resultText:{
29          fontSize: 48,
30          fontWeight:"bold",
31      },
32      scoreboard : {
33          height:100,
34          justifyContent:'flex-end',
35          alignItems: "center",
36      },
37      scoreboardText :{
38          fontSize: 24,
39          fontWeight:"normal",
40          color:"#0bf8ff"
```

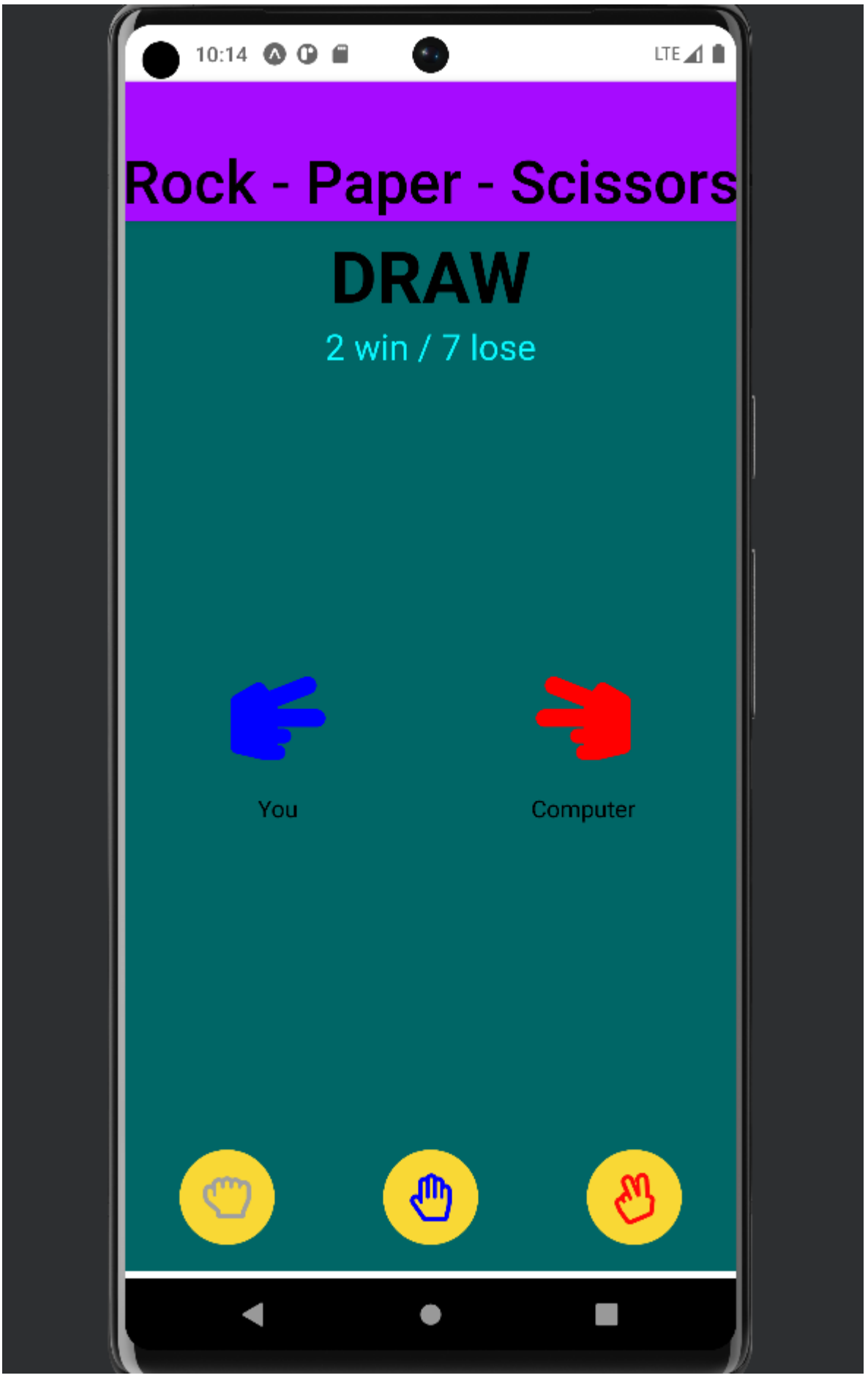
Asıl oyunun oynandığı, kazananın belirlendiği ve scoreun tutulduğu kısım ve bunların viewi.

Oyun içi görüntüler:


10:14
LTE
Rock - Paper - Scissors
Let's Play


10:14
LTE
Rock - Paper - Scissors
DRAW
0 win / 0 lose
You
Computer


10:14
LTE
Rock - Paper - Scissors
LOSE
2 win / 4 lose
You
Computer


10:14
LTE
Rock - Paper - Scissors
DRAW
2 win / 7 lose
You
Computer