Анимация в React Native

Код программы:

```
import Constants from 'expo-constants';
import {
  Animated,
  StyleSheet,
  Text,
  View,
  SafeAreaView,
  ScrollView,
  ImageBackground,
  Button, Dimensions
} from 'react-native';
import React, {useEffect, useRef} from "react";

const windowWidth = Dimensions.get('window').width;
const windowHeight = Dimensions.get('window').height;

const scrollX = new Animated.Value(0)
const springValue = new Animated.Value(1)
const imageRotateValue = new Animated.Value(0)

const interpolateScrollViewRotation = imageRotateValue.interpolate({
  inputRange: [0, 100],
  outputRange: ["0deg", "360deg"]
})

const images = new Array(6).fill('https://images.unsplash.com/photo-
1556740749-887f6717d7e4');

const flipAnimation = () => {

    Animated.sequence([
      Animated.timing(imageRotateValue, {
        toValue: 360,
        duration: 2000,
        useNativeDriver: true,
      }),
      Animated.timing(imageRotateValue, {
        toValue: 0,
        duration: 2000,
        useNativeDriver: true
      })
    ]).start()
}

const transitionAnimation = (index) => {
  return {
    transform: [
      { perspective: 800 },
      {
        scale: scrollX.interpolate({
          inputRange: [
            (index - 1) * windowWidth,
            index * windowWidth,
            (index + 1) * windowWidth
          ],
          outputRange: [0.5, 1, 0.5]
```

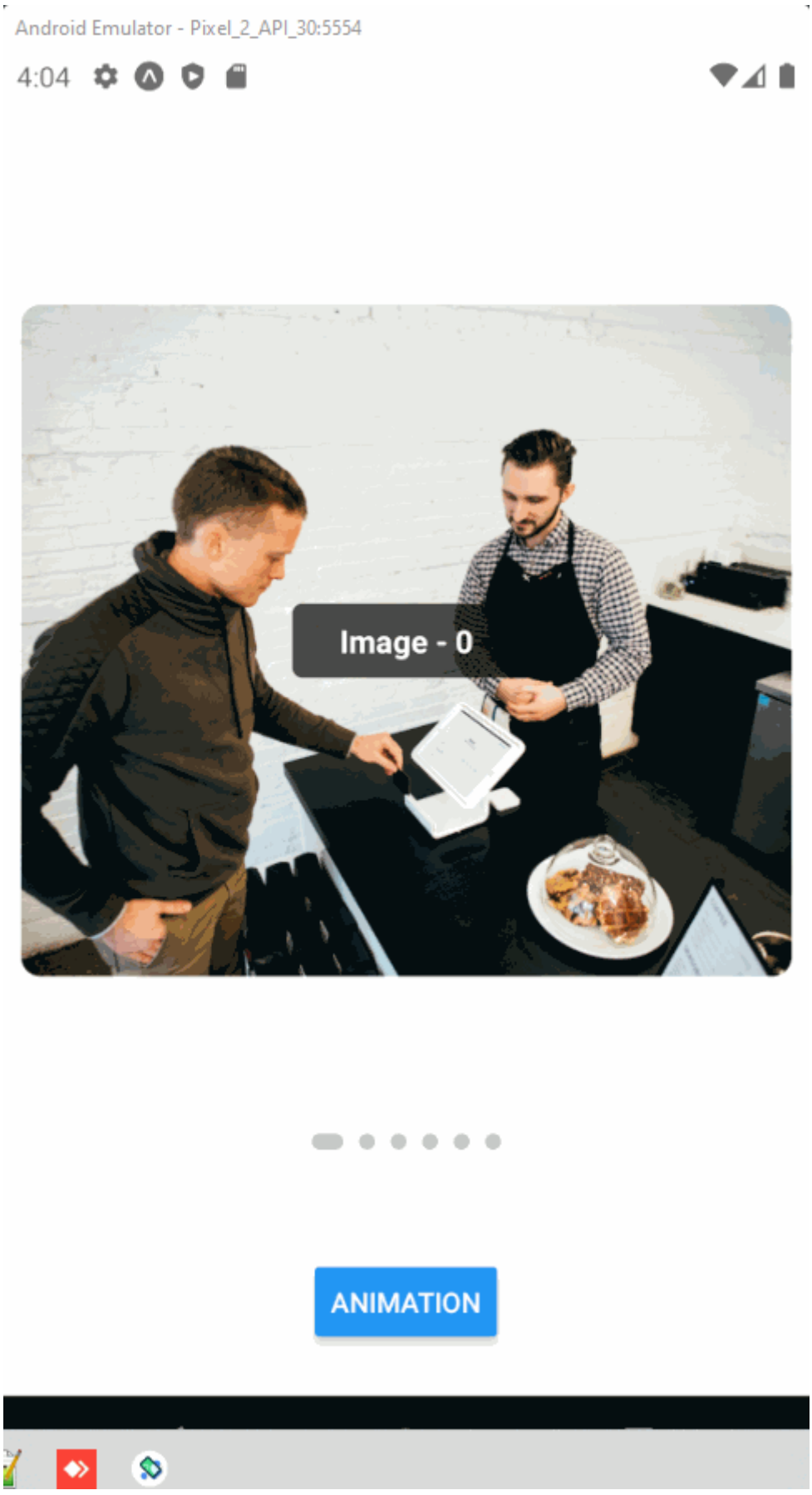
```
        })
      },
      {
        rotateX: scrollX.interpolate({
          inputRange: [
            (index - 1) * windowWidth,
            index * windowWidth,
            (index + 1) * windowWidth
          ],
          outputRange: ["15deg", "0deg", "15deg"]
        })
      },
      {
        rotateY: scrollX.interpolate({
          inputRange: [
            (index - 1) * windowWidth,
            index * windowWidth,
            (index + 1) * windowWidth
          ],
          outputRange: ["15deg", "0deg", "15deg"]
        })
      }
    ]
  };
};

const AnimatedImageCardView = (props) => {
  return (
      <Animated.View style={[styles.scrollPage,
        {transform: [{scale: springValue}, {rotate:
interpolateScrollViewRotation}]}]}>
        <Animated.View style={[styles.screen,
transitionAnimation(props.index)]}>
          {props.children}
        </Animated.View>
      </Animated.View>
  );
};

const FadeInView = (props) => {
  const fadeAnim = useRef(new Animated.Value(0)).current  // Initial value
for opacity: 0

  useEffect( () => {
    Animated.timing(
        fadeAnim,{
          toValue: 1,
          duration: 1000,
          useNativeDriver: true,
        }
    ).start()
  }, [fadeAnim])

  return (
      <Animated.View                 // Special animatable View
          style={{
            ...props.style,
            opacity: fadeAnim,         // Bind opacity to animated value
          }}
      >
        {props.children}
      </Animated.View>
  );
}
```

```
const ScrollViewScreen = () => {
  return (
      <SafeAreaView style={styles.container}>
        <FadeInView style={styles.scrollContainer}>

          <ScrollView
              horizontal={true}
              style={styles.scrollViewStyle}
              pagingEnabled
              showsHorizontalScrollIndicator={false}
              onScroll={Animated.event([
                {
                  nativeEvent: {
                    contentOffset: {
                      x: scrollX,
                    },
                  },
                },
              ], {useNativeDriver: false})}
              scrollEventThrottle={1}

          >
            {images.map((image, imageIndex) => {
              return (
                  <AnimatedImageCardView
                      key={imageIndex}
                      index={imageIndex}
                  >
                    <ImageBackground source={{ uri: image }}
style={styles.card}>
                      <View style={styles.textContainer}>
                        <Text style={styles.infoText}>
                          {"Image - " + imageIndex}
                        </Text>
                      </View>
                    </ImageBackground>
                  </AnimatedImageCardView>
              );
            })}
          </ScrollView>
          <View style={styles.indicatorContainer}>
            {images.map((image, imageIndex) => {
              const width = scrollX.interpolate({
                inputRange: [
                  windowWidth * (imageIndex - 1),
                  windowWidth * imageIndex,
                  windowWidth * (imageIndex + 1)
                ],
                outputRange: [8, 16, 8],
                extrapolate: "clamp"
              });
              return (
                  <Animated.View
                      key={imageIndex}
                      style={[styles.normalDot, { width }]}
                  />
              );
            })}
          </View>
        </FadeInView>
        <View style={{ margin: 30}}>
          <Button title="Animation" onPress={flipAnimation}/>
        </View>
```

```
      </SafeAreaView>
  );
}



export default ScrollViewScreen

const styles = StyleSheet.create({
  container: {
    flex: 1,
    paddingTop: Constants.statusBarHeight,
    alignItems: "center",
    justifyContent: "center"
  },
  fadingContainer: {
    backgroundColor: "powderblue"
  },
  scrollContainer: {
    flex: 1,
    alignItems: "center",
    justifyContent: "center"
  },
  scrollViewStyle: {
    flex: 1,
    backgroundColor:'white',
  },
  screen: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
  },
  button: {
    backgroundColor: "teal",
    height: 50,
    marginTop: 80,
    borderRadius: 15,
    alignItems: "center",
  },
  button_text: {
    color: "white",
    padding: 12,
    paddingHorizontal: 20,
    fontWeight: "bold",
    fontSize: 18,
  },
  card: {
    width: windowWidth - 20,
    height: windowHeight / 2,
    borderRadius: 10,
    overflow: "hidden",
    alignItems: "center",
    justifyContent: "center",
  },
  textContainer: {
    backgroundColor: "rgba(0,0,0, 0.7)",
    paddingHorizontal: 24,
    paddingVertical: 8,
    borderRadius: 5
  },
  infoText: {
    color: "white",
    fontSize: 16,
    fontWeight: "bold"
```

```
  },
  normalDot: {
    height: 8,
    width: 8,
    borderRadius: 4,
    backgroundColor: "silver",
    marginHorizontal: 4
  },
  scrollPage: {
    width: windowWidth,
    padding: 20,
  },
  indicatorContainer: {
    flexDirection: "row",
    alignItems: "center",
    justifyContent: "center",
    marginBottom: 30,
  },
  outerScroll: {
    flex: 1,
    flexDirection: "column",
  },
  row: {
    flex: 1,
  },
})
```

Пример работы:


Android Emulator - Pixel_2_API_30:5554
4:04
Image - 0
ANIMATION