Tyler Plihcik CS499 HW2

Sources used: LinkedIn Learning Video

Flutter: Part 4 - Building Apps with State

CS 499-007: Advanced Mobile Application Development Spring 2020 x

Homework-02 Due Friday February 14 30 Points Total

- 1) This is an individual assignment. It is okay to exchange ideas with other people or review external material on the assignment topics but the work you submit should be your work.

- 2) Include a header in your assignment sheet including your name, the section number, and the homework number.

- 3) Cite any external sources you use to help you with your assignment.

- 4) Submit your assignment on the assignment page on BBLearn. Submissions via email will
not be accepted.

Android Application Requirements:

Implement a Flutter mobile application modeled based on the *Dice App (Dicee Project)* application implemented in the *Flutter: Part4* LinkedIn training. Requirements for the application are listed below:

1. The application will have a single screen displaying five die images.

2. Use *Scaffold* widget for the basic material design structure of your application screen display.

3. Include an *AppBar* widget for the screen display and set its title to a name representative of the applications purpose/operations.

4. The application at the start shows five dice with the dice images the side one of the dice.

5. Clicking on any die image will roll all five dice.

6. After dice roll the applications screen displays the five dice after rolling.

7. Use unique choice of Material colors for the *Scaffold* and *AppBar* widgets *background* colors. You can select colors from collection of Material colors available here.

8. Use a custom *app icon* for your application for the iOS and Android phone homepage.

9. Be creative and aim for an elegant and visually appealing screen display.

Implementation:

1. Implement your application using Android Studio, Flutter platform and Dart programming language.

1

2. Start your Flutter application using the starting project provided for the *Dicee Project* in *Flutter: Part 4* LinkedIn training. Use the Git Hub link provided in this training to clone the project form the Git Hub. This project includes the necessary image files and the required project configuration.

3. Eliminate code redundancy using a modular design. Create functional abstractions for creation of dice and updating the state of the 5 dice after dice Roll.

Testing: Use *Nexus 6* virtual device with API level 28 to run your application. ***Deliverables:***

- 1) Complete Android project files in a zip folder. Name your folder using the naming convention like: *lastname_firstname_CS499_hw2_Dice.zip*.

- 2) Include snapshots of the following artifacts as displayed in the Android Studio. Paste the snapshots in a copy of this document.

- The Dart file under the *lib* directory. Expand all the code to show the complete Dart code including the *import* instructions.



```dart
import 'package:flutter/material.dart' ;

import 'dart:math' ;

void main() {
  return runApp(
    MaterialApp(
      home: Scaffold(
        backgroundColor: Colors.blue,
        appBar: AppBar(
          title: Text('Dice'),
          backgroundColor: Colors.orange,
        ), // AppBar
        body: DicePage(),
      ), // Scaffold
    ), // MaterialApp
  );
}

class DicePage extends StatefulWidget {
  @override
  _DicePageState createState() => _DicePageState();
}

class _DicePageState extends State<DicePage> {

  int oneDiceNumber = 1 ;
  int twoDiceNumber = 2 ;
  int threeDiceNumber = 3 ;
  int fourDiceNumber = 4 ;
  int fiveDiceNumber = 5 ;




  void changeDiceFace()
  {
    setState() {
      oneDiceNumber = Random().nextInt(6) + 1 ;
      twoDiceNumber = Random().nextInt(6) + 1 ;
      threeDiceNumber = Random().nextInt(6) + 1 ;
      fourDiceNumber = Random().nextInt(6) + 1 ;
      fiveDiceNumber = Random().nextInt(6) + 1 ;

    });
  }


  @override
  Widget build(BuildContext context) {

    return Center(
      child: Row(
        children: <Widget>[
          Expanded(
            child: FlatButton(
              onPressed: (){
                changeDiceFace() ;

              },
              child: Image.asset('images/dice$oneDiceNumber.png'),
            ), // FlatButton
          ), // Expanded

          Expanded(
            child: FlatButton(
              onPressed: (){
                changeDiceFace() ;

              },
              child: Image.asset('images/dice$twoDiceNumber.png'),
            ), // FlatButton
          ), // Expanded
          Expanded(
            child: FlatButton(
              onPressed: (){
                changeDiceFace() ;

              },
              child: Image.asset('images/dice$threeDiceNumber.png'),
            ), // FlatButton
          ), // Expanded

          Expanded(
            child: FlatButton(
              onPressed: (){
                changeDiceFace() ;

              },
              child: Image.asset('images/dice$fourDiceNumber.png'),
            ), // FlatButton
          ), // Expanded
          Expanded(
            child: FlatButton(
              onPressed: (){
                changeDiceFace() ;

              },
              child: Image.asset('images/dice$fiveDiceNumber.png'),
            ), // FlatButton
          ), // Expanded




        ], // <Widget>[]
      ), // Row
    ); // Center
  }
}
```

- The Android Studio Project Pane showing any directories and components that you have added/changed for your application. Expand android, images, ios and lib directories to show all application components.
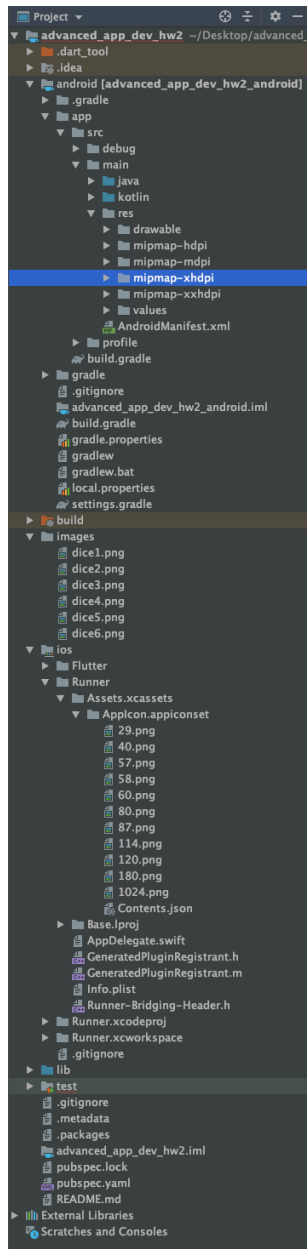
- Show at least one android *mipmap* image and one iOS *icon-App* image displayed in the Android Studio Editor pane.
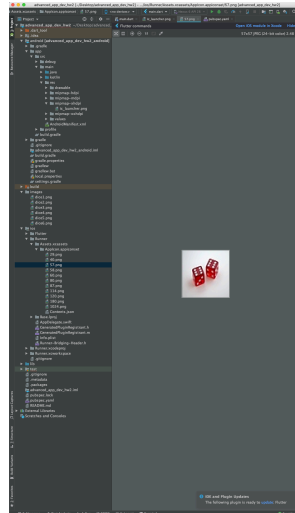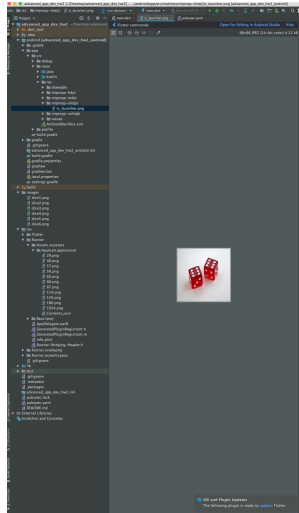
- The *pubspec.yaml* file after removing all comments.


```yaml
1    name: dice
2    description: A new Flutter application.
3
4    version: 1.0.0+1
5
6    environment:
7      sdk: ">=2.1.0 <3.0.0"
8
9    dependencies:
10     flutter:
11       sdk: flutter
12
13     cupertino_icons: ^0.1.2
14
15   dev_dependencies:
16     flutter_test:
17       sdk: flutter
18
19   flutter:
20
21     uses-material-design: true
22
23     assets:
24       - images/
25
```

-
- 3) View of your application when it is run on the *Nexus 6* virtual device.
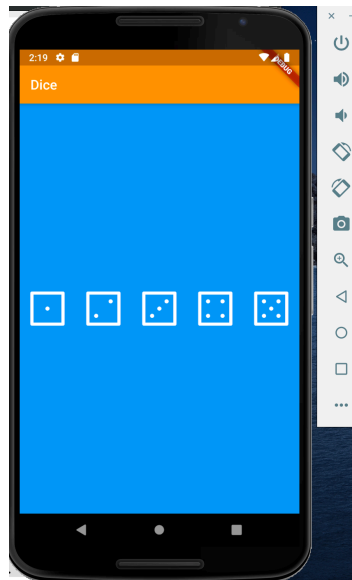
- 4) View of Android Nexus 6 home page including display of your *app ico*n.

-