

A detailed white line-art pattern of a circuit board on a black background, featuring various electronic components like resistors, capacitors, and integrated circuits.

11

TEXTO BASE

DESENVOLVIMENTO PARA DISPOSITIVOS MÓVEIS

Desenvolvimento do app cara ou coroa 2

Victor Inacio de Oliveira

Resumo

Olá, nesta aula iremos iniciar o desenvolvimento de uma aplicação funcional. Será um jogo bem conhecido, o cara ou coroa. Nesta aplicação consolidaremos os conhecimentos adquiridos ao longo do curso e para isso vamos criar um aplicativo que tenha imagens, botões, lógica e interatividade com o usuário. Nessa segunda etapa iremos colocar as funcionalidades lógicas no aplicativo e executar em dispositivo virtual e físico.

11.1 App cara ou coroa 2.

Abra o projeto iniciado anteriormente e continue o seu desenvolvimento. Notem que o layout deve estar parecido como o visto na figura 1 abaixo.



Figura 1: Imagem do layout do aplicativo. fonte: próprio autor

A partir do código já desenvolvido, vamos inserir a condição de vitória. Foi criado uma função como a vista abaixo (figura 2).

```
16  class _HomeState extends State<Home> {  
17  
18      var _frases = [  
19          "DEU CARA",  
20          "DEU COROA"  
21      ];  
22  
23      var _fraseGerada = "QUAL SERÁ?";
```

Figura 2: Função para criar as frases de vitória. fonte: próprio autor

Outra função que temos que inserir é a de definir a função randômica para mostrar qual frase da vitória deve aparecer. Foi criada a função `_gerarFrase` que pode ser vista na figura 3 a seguir.

```

25 void _gerarFrase(){
26     // 0, 1
27     var numeroSorteado = Random().nextInt( _frases.length );
28     setState(() {
29         _fraseGerada = _frases[ numeroSorteado ];
30     });
31 }

```

Figura 3: Função random usada para gerar a frase. fonte: próprio autor

Logo após devemos inserir o trecho do código que irá escrever a frase na tela. A figura 4 mostra esse trecho.

```

Text(
  "ESCOLHA CARA OU COROA",
  textAlign: TextAlign.justify,
  style: TextStyle(
    fontSize: 30,
    fontStyle: FontStyle.italic,
    color: Colors.black
  ), // TextStyle
), // Text

Text(
  _fraseGerada,
  textAlign: TextAlign.justify,
  style: TextStyle(
    fontSize: 30,
    fontStyle: FontStyle.italic,
    color: Colors.black
  ), // TextStyle
), // Text

```

Figura 4: Função para mostrar a frase na tela. fonte: próprio autor

Após feito, o aplicativo pode ser executado tanto no emulador como no virtual.

A seguir o código completo é mostrado.

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

void main(){
  runApp(MaterialApp(
    home: Home(),
    debugShowCheckedModeBanner: false,
  ));
}

class Home extends StatefulWidget {
  @override
  _HomeState createState() => _HomeState();
}

class _HomeState extends State<Home> {

  var _frases = [
    "DEU CARA",
    "DEU COROA"
  ];
}
```



```
var _fraseGerada = "QUAL SERÁ?";
```

```
void _gerarFrase(){
```

```
    // 0, 1
```

```
    var numeroSorteado = Random().nextInt( _frases.length );
```

```
    setState() {
```

```
        _fraseGerada = _frases[ numeroSorteado ];
```

```
    });
```

```
}
```

```
@override
```

```
Widget build(BuildContext context) {
```

```
    return Scaffold(
```

```
        appBar: AppBar(
```

```
            title: Text("CARA OU COROA"),
```

```
            backgroundColor: Colors.blue,
```

```
        ),
```

```
        body: Center(
```

```
            child: Container(
```

```
                padding: EdgeInsets.all(18),
```

```
                child: Column(
```

```
                    mainAxisAlignment: MainAxisAlignment.spaceEvenly,
```

```
                    crossAxisAlignment: CrossAxisAlignment.center,
```

```
                    children: <Widget>[
```

```
                        Image.asset("imagens/caraoucoroa.jpg"),
```

```
                        Text(
```

```
                            "ESCOLHA CARA OU COROA",
```

```
textAlign: TextAlign.justify,

style: TextStyle(

    fontSize: 30,

    fontStyle: FontStyle.italic,

    color: Colors.black

),

),

Text(

    _fraseGerada,

    textAlign: TextAlign.justify,

    style: TextStyle(

        fontSize: 30,

        fontStyle: FontStyle.italic,

        color: Colors.black

    ),

),

RaisedButton(

    child: Text(

        "PRESSIONE",

        style: TextStyle(

            fontSize: 25,

            color: Colors.white,

            fontWeight: FontWeight.bold

        ),

    ),

    color: Colors.blue,

    onPressed: _gerarFrase,
```

```
)  
],  
,  
,  
,  
,  
);  
}  
}
```

Bom, é isso, dando tudo certo a imagem da figura 4 deve aparecer depois da execução do programa.

Referências

FLUTTER, “Tutorial Flutter”, <https://flutter.dev/> acesso em maio de 2020

FLUTTER, “Containers e widgets”, <https://flutter.dev/docs/development/ui/widgets> acesso em maio de 2020

