

Mobile Development :

7 : Flutter for Mobile Development : Part 2

More Interactivity, Navigation and Nav Widgets



Professor Imed Bouchrika

National School of Artificial Intelligence
imed.bouchrika@ensia.edu.dz

Outline :

- More on Constructing UIs
- More Interactivity
 - Tic Tac Toe Game
 - Sliding Number Puzzle
- Navigation & Routings
 - Multiple Screens/Pages
 - Passing Data.
 - Expenses App
- Navigation Widgets
 - Sidebar Drawer
 - Bottom Navigation
 - Top Bar



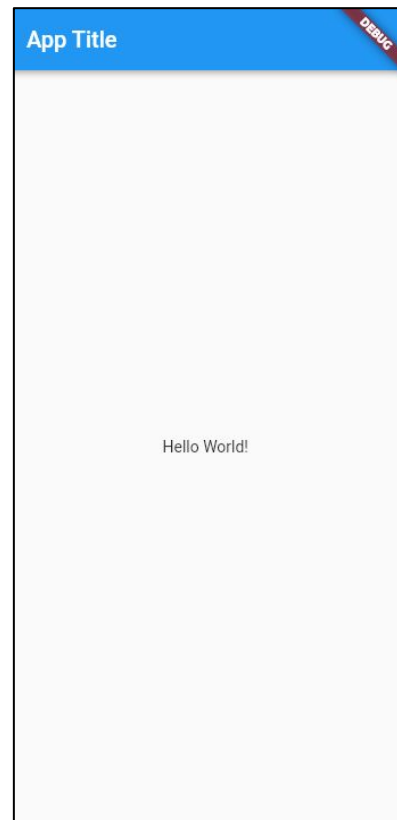
Section 1

More Interactivity on a single screen



Constructing UIs for Flutter Apps

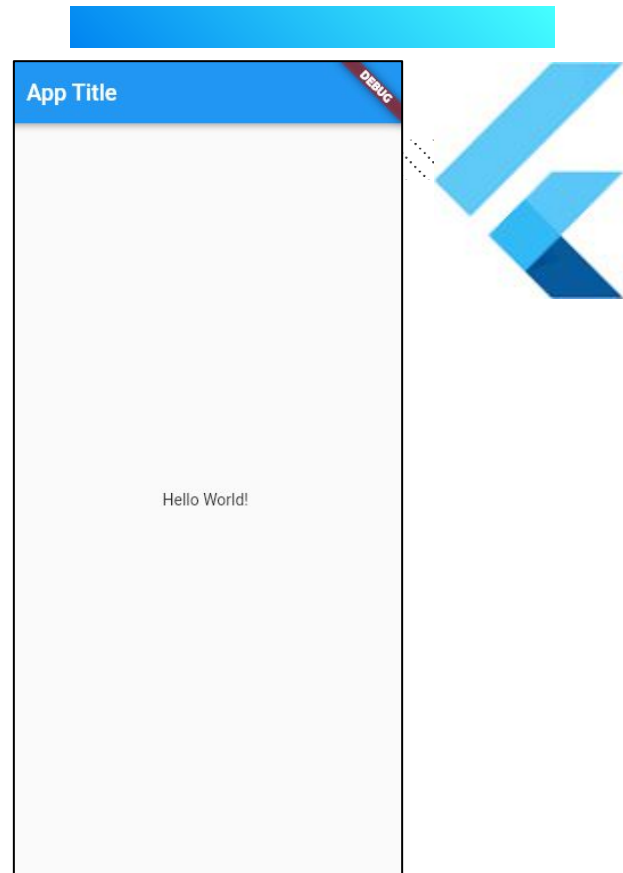
- Skeleton for Building an App



Constructing UIs for Flutter Apps

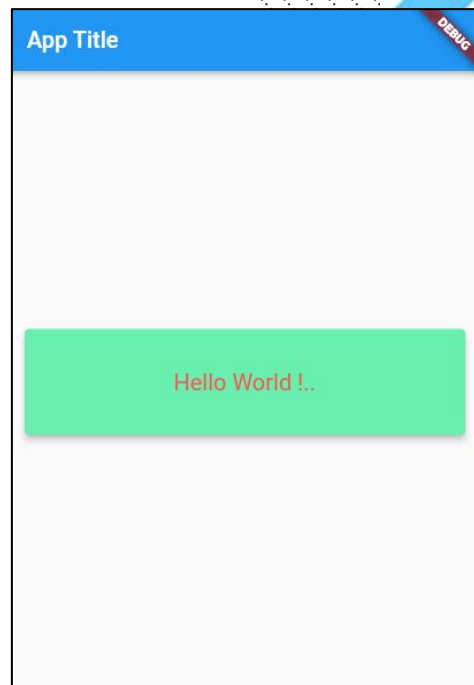
- Skeleton for Building an App

```
import 'package:flutter/material.dart';  
void main() {  
  runApp(MaterialApp(  
    home: Scaffold(  
      appBar: AppBar(title: const Text('App Title')),  
      body: const Center(  
        child: Text('Hello World!'),  
      ),  
    ),  
  ));  
}
```



Constructing UIs for Flutter Apps

- Skeleton for Building an App : Styling



Constructing UIs for

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

void main() {
  runApp(MaterialApp(
    home: Scaffold(
      appBar: AppBar(title: const Text('App Title')),
      body: Center(
        child: Container(
          margin: EdgeInsets.all(10),
          width: double.infinity,
          height: 100,
          child: Card(
            color: Colors.greenAccent,
            elevation: 5,
            child: Center(
              child: Text(
                'Hello World !..',
                style: TextStyle(
                  fontSize: 20,
                  color: Colors.redAccent),
              ),
            ),
          ),
        ),
      ),
    ),
  ));
}
```



Constructing UIs for Flutter Apps

- **Either Encapsulate Widgets via:**
 - **Widget Class with a Build Method**
 - A Function that returns a Widget

```
import 'package:flutter/material.dart';
import
'package:reminder_flutter_hello_world_two/widgets/myWidget.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: const Text('App Title')),
        body: const Center(
          child: Column(children: [
            Text('Hello World!'),
            SizedBox(height: 30),
            myWidget(),
          ]),
        ),
      ),
    );
  }
}
```


Constructing UIs for Flutter Apps

- Either Encapsulate Widgets via:

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

class myWidget extends StatelessWidget {
  const myWidget({super.key});

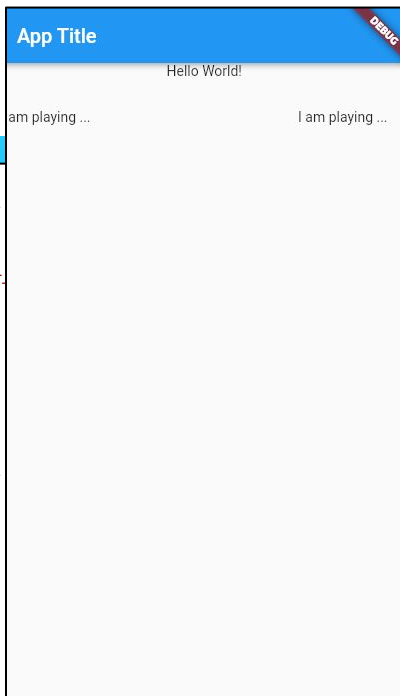
  @override
  Widget build(BuildContext context) {
    return const Row(
      children: [
        Text('I am playing ...'),
        Spacer(),
        Text('I am playing ...'),
        SizedBox(width: 20),
      ],
    );
  }
}
```

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

void main() {
  runApp(const MainApp());
}
```

```
class MainApp extends StatelessWidget {
  const MainApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: const Text('App Title')),
        body: const Center(
          child: Column(children: [
            Text('Hello World!'),
            SizedBox(height: 30),
            myWidget(),
          ]),
        ),
      ),
    );
  }
}
```



Constructor is defined to take the super.key

Constructing UIs for Flutter Apps

- Either Encapsulate Widgets via:
 - A Function that returns a Widget

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

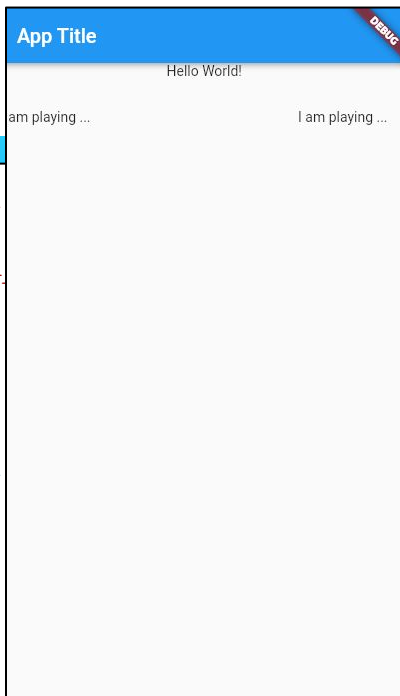
Widget getMyWidget() {
  return const Row(
    children: [
      Text('I am playing ...'),
      Spacer(),
      Text('I am playing ...'),
      SizedBox(width: 20),
    ],
  );
}
```

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

void main() {
  runApp(const MainApp());
}

class MainApp extends StatelessWidget {
  const MainApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: const Text('App Title')),
        body: const Center(
          child: Column(children: [
            Text('Hello World!'),
            SizedBox(height: 30),
            getMyWidget(),
          ]),
        ),
      ),
    );
  }
}
```



Constructing UIs for Flutter Apps

- Stateful Widget

The stateful widget Constructor takes another **real** argument to be used.

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

class MyStatefulWidget extends StatefulWidget {
  var myStart = 0;

  MyStatefulWidget({super.key, this.myStart = 0});

  @override
  _MyStatefulWidgetState createState() => _MyStatefulWidgetState();
}

class _MyStatefulWidgetState extends State<MyStatefulWidget> {
  @override
  Widget build(BuildContext context) {
    return Text('Value is ${widget.myStart}');
  }
}
```

Constructing UIs for Flutter Apps

- Stateful Widget

To access data of the Widget from the state instance, use :
widget.varName

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

class MyStatefulWidget extends StatefulWidget {
  var myStart = 0;

  MyStatefulWidget({super.key, this.myStart = 0});

  @override
  _MyStatefulWidgetState createState() => _MyStatefulWidgetState();
}

class _MyStatefulWidgetState extends State<MyStatefulWidget> {
  @override
  Widget build(BuildContext context) {
    return Text('Value is ${widget.myStart}');
  }
}
```

Constructing Flutter A

- **Stateful Widget**

Stateless and Stateful Widgets are integrated into the UI by calling the constructor.

No argument is given for the stateful Widget

```
import 'package:flutter/material.dart';
import
'package:reminder_flutter_hello_world_two/widgets/myStatefulWidget.d
art';
import
'package:reminder_flutter_hello_world_two/widgets/myWidget.dart';

void main() {
  runApp(const MainApp());
}

class MainApp extends StatelessWidget {
  const MainApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: const Text('App Title')),
        body: Center(
          child: Column(children: [
            const Text('Hello World!'),
            const SizedBox(height: 30),
            const myWidget(),
            const SizedBox(height: 30),
            const Text('Below is a stateful Widget'),
            const SizedBox(height: 30),
            MyStatefulWidget(),
          ])),
        ),
      );
```

Constructing Flutter A

- Stateful Widget

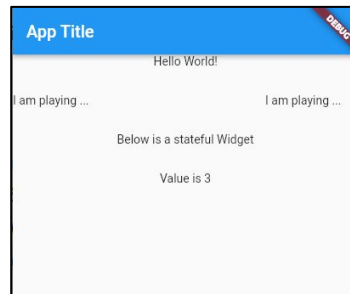
To pass an argument into
the constructor of the
Stateful Widget

```
import 'package:flutter/material.dart';
import
'package:reminder_flutter_hello_world_two/widgets/myStatefulWidget.d
art';
import
'package:reminder_flutter_hello_world_two/widgets/myWidget.dart';

void main() {
  runApp(const MainApp());
}

class MainApp extends StatelessWidget {
  const MainApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: const Text('App Title')),
        body: Center(
          child: Column(children: [
            const Text('Hello World!'),
            const SizedBox(height: 30),
            const myWidget(),
            const SizedBox(height: 30),
            const Text('Below is a stateful Widget'),
            const SizedBox(height: 30),
            MyStatefulWidget(myStart:3),
          ]),
        ),
      ),
    );
```



Constructing Flutter A

- Stateful Widget

initState is invoked only **one time** once the widget is initialized.

Vs

The method build is called **often whenever a refresh** is required.

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

class MyStatefulWidget extends StatefulWidget {
  var myStart = 0;
  MyStatefulWidget({super.key, this.myStart = 0});

  @override
  _MyStatefulWidgetState createState() => _MyStatefulWidgetState();
}

class _MyStatefulWidgetState extends State<MyStatefulWidget> {
  int increment = 0;
  @override
  void initState() {
    super.initState();
    increment = widget.myStart;
  }
  @override
  Widget build(BuildContext context) {
    return Column(children: [
      Text('Value is $increment'),
      SizedBox(height: 10),
      ElevatedButton(
        onPressed: () {
          increment = increment + 1;
          setState(() {});
        },
        child: const Text('Increment'))
    ]);
  }
}
```

Constructing Flutter A

- **Stateful Widget**

setState({});
Is called to invoke the
build method of the
stateful widget state.
=
**Refresh the UI of the
widget**

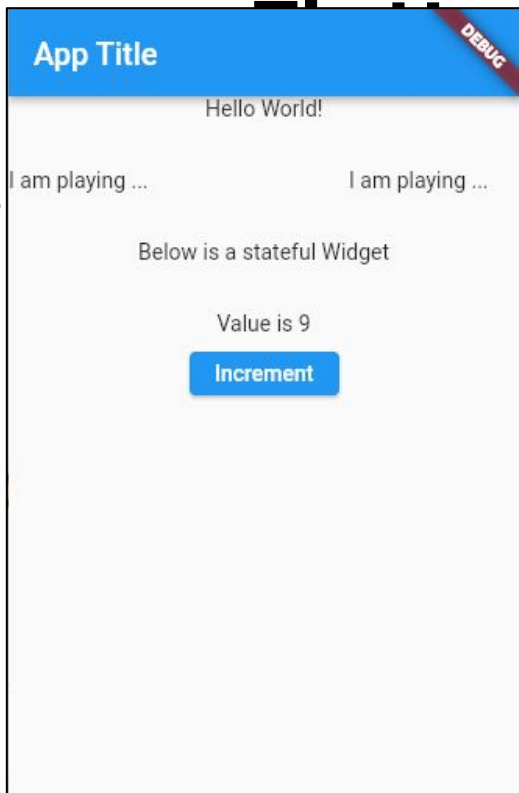
```
import 'package:flutter/material.dart';

class MyStatefulWidget extends StatefulWidget {
  var myStart = 0;
  MyStatefulWidget({super.key, this.myStart = 0});

  @override
  _MyStatefulWidgetState createState() => _MyStatefulWidgetState();
}

class _MyStatefulWidgetState extends State<MyStatefulWidget> {
  int increment = 0;
  @override
  void initState() {
    super.initState();
    increment = widget.myStart;
  }
  @override
  Widget build(BuildContext context) {
    return Column(children: [
      Text('Value is $increment'),
      SizedBox(height: 10),
      ElevatedButton(
        onPressed: () {
          increment = increment + 1;
          setState(() {});
        },
        child: const Text('Increment'))
    ]);
  }
}
```


Constructing

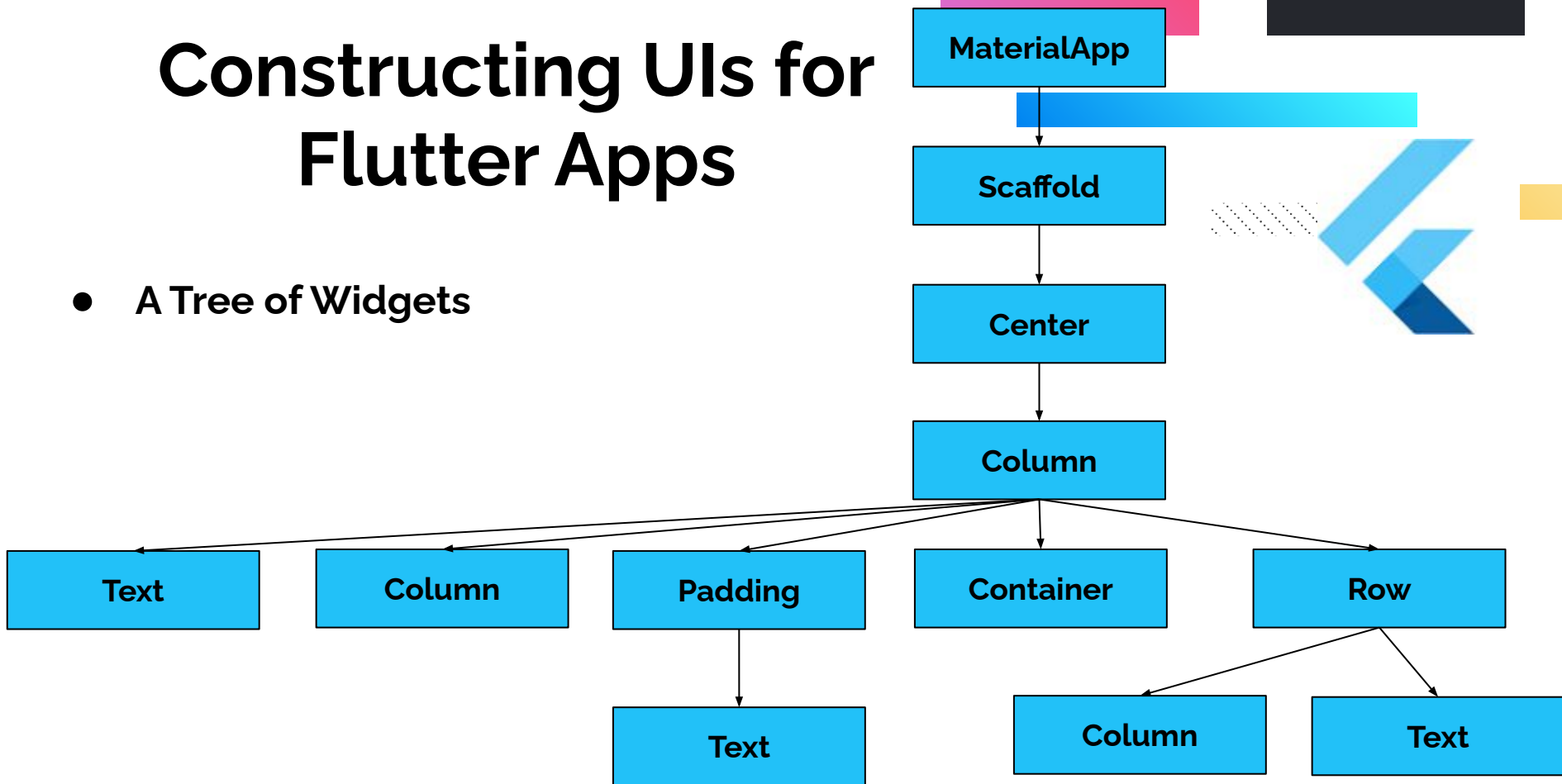


```
import 'package:flutter/material.dart';
import
'package:reminder_flutter_hello_world_two/widgets/myStatefulWidget.d
art';
import
'package:reminder_flutter_hello_world_two/widgets/myWidget.dart';
void main() {
  runApp(const MainApp());
}
class MainApp extends StatelessWidget {
  const MainApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: const Text('App Title')),
        body: Center(
          child: Column(children: [
            const Text('Hello World!'),
            const SizedBox(height: 30),
            const myWidget(),
            const SizedBox(height: 30),
            const Text('Below is a stateful Widget'),
            const SizedBox(height: 30),
            MyStatefulWidget(myStart: 3),
          ])),
        ),
      );
  }
}
```

Constructing UIs for Flutter Apps

- A Tree of Widgets



Constructing UIs for Flutter Apps

- **Scaffold** : appBar, body, floatingActionButton, drawer, bottomNavigationBar
- **AppBar** : title, leading, elevation
- **Container** : child, padding, margin, height, width, decoration alignment
- **Column** : children, mainAxisAlignment, crossAxisAlignment
- **Row** : children, mainAxisAlignment, crossAxisAlignment
- **Stack** : alignment, children,
- **SizedBox** : child, height, width,
- **Card** : child, color, elevation, margin, clipBehavior, shape
- **Expanded** : child, flex
- **Spacer** : flex
- **Padding** : padding --> EdgeInsets.all|only|fromLTRB|symmetric
- **Center** : child
- **Align** : child, alignment
- **ListView.builder** : scrollDirection, physics, itemCount, itemBuilder(context,index)
- **GridView.builder** : padding, physics, itemCount, itemBuilder(context,index)

- **Text** : style, textAlign
- **TextFormField** : onChanged, controller, decoration, validator
- **ElevatedButton** : style, onPressed, child
- **IconButton** : icon, onPressed
- **Icon** : color, size,
- **Slider** : value, max, divisions, onChanged, label
- **Switch** : value, onChanged, thumbIcon
- **ListTile** : leading, title, subtitle, trailing, tileColor, onTap
- **Image.asset** : imagePath,
- **TextStyle** : fontSize, color, fontWeighth
- **BoxDecoration** : image, shape, border, borderRadius, color
- **Opacity** : opacity, child
- **RoundedRectangleBorder** : borderRadius, side
- **Position** : left, bottom, right, top
- **BoxShadow** : color, offset, blurStyle

Constructing UIs for Flutter Apps

- **Getting Help ! Speeding up the learning curve**
 - To get the list of attributes for a Widget, hover over the Widget
 - Use VS Code refactoring Tools to wrap widgets by Container Widgets (Container, Row, Column...etc)



Constructing UIs for Flutter Apps



- **Questions**

- Final vs Const :
 - final : The Value must be initialized during the declaration and will never change
 - const : The value will be computed during compile time and will never change later during execution.
 - Why use const : To optimize the performance of flutter via the reuse of the same constant widgets.

Constructing UIs for Flutter Apps



- Questions

- ? VS ?. VS ?? VS !!

- ? : To say a variable can take the value Null during the declaration
 - String? name ;
 - ?. : to avoid null pointer exception and return null :
 - **return person?.name**
 - For the case when person is null, it returns null directly without accessing **name**
 - ?? : If the null, return the value on the right side:
 - **return person??Person()**
 - If person is null, instantiate a new Person and return it.
 - ! : To cast away nullability (or simply get rid of the warnings)
 - **return person!.name** (Compiler will complain that person is nullable, we cast it to non-nullable.
Warning : you use this only when you are completely certain that the object is non-nullable

Constructing UIs for Flutter Apps

- **Questions**

- Set vs Map vs List
 - Set is {'one', 'two', 'three'} not like Python



More Interactivity for Flutter Apps



- **Tic Tac Toe Game**

- Reminder how we did it in Native Android using Kotlin
 - Do the XML Widget
 - Putting the business logic inside Kotlin
 - Registering EventListener
 - Whenever they click on a TextView, we check, if empty, we set its text to the current player symbol
 - We check if there is a win

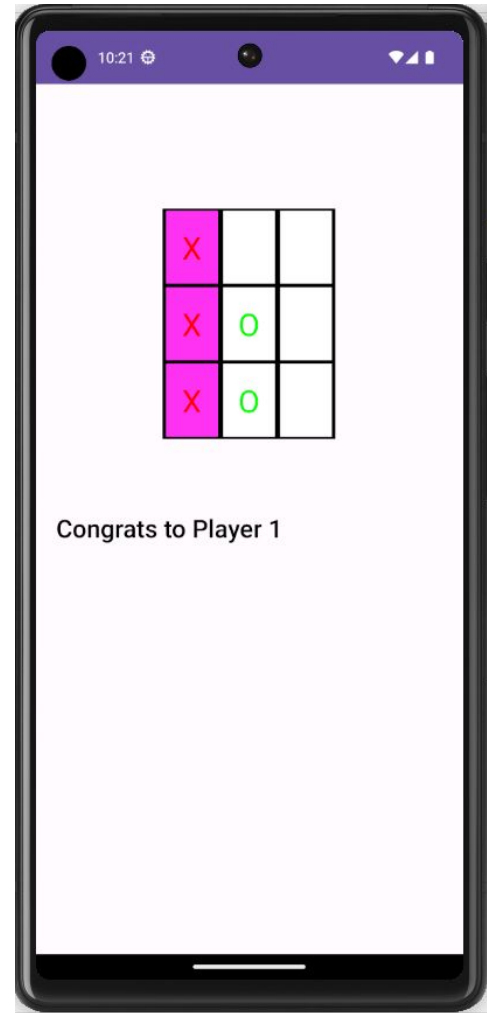

```

var win=checkWin()
if (win){
    findViewById<TextView>(R.id.tx_turn).setText("Congrats to Player "+turn)
    gameover=true
    return
}
turn=if (turn==1) 2 else 1
findViewById<TextView>(R.id.tx_turn).setText("Turn of Player: "+turn)
}
var possible_wins= arrayOf(
    arrayOf(R.id.tx_a1,R.id.tx_a2,R.id.tx_a3),
    arrayOf(R.id.tx_a4,R.id.tx_a5,R.id.tx_a6),
    arrayOf(R.id.tx_a7,R.id.tx_a8,R.id.tx_a9),

    arrayOf(R.id.tx_a1,R.id.tx_a5,R.id.tx_a9),
    arrayOf(R.id.tx_a7,R.id.tx_a5,R.id.tx_a3),

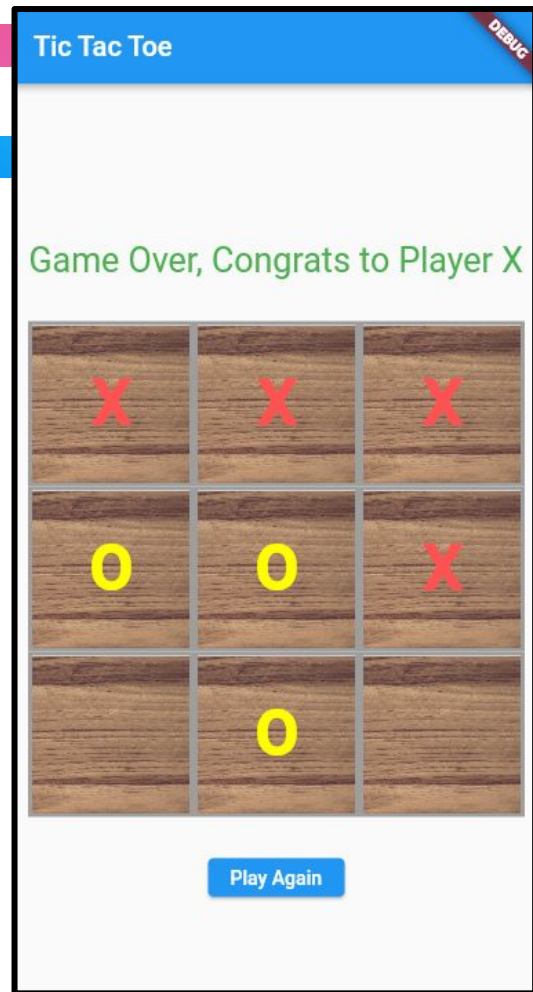
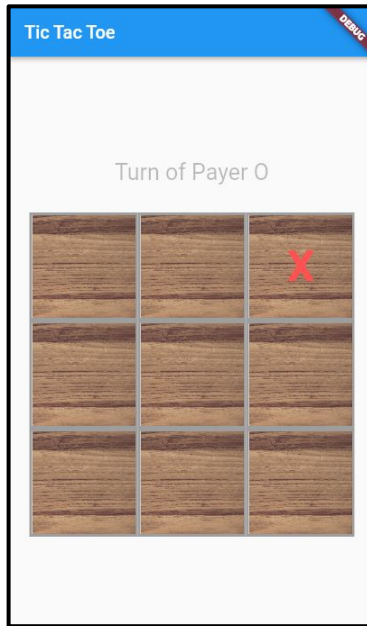
    arrayOf(R.id.tx_a1,R.id.tx_a4,R.id.tx_a7),
    arrayOf(R.id.tx_a2,R.id.tx_a5,R.id.tx_a8),
    arrayOf(R.id.tx_a3,R.id.tx_a6,R.id.tx_a9))
fun checkWin(): Boolean{
    for (possible in possible_wins){
        var seqStr=""
        for (cellId in possible){
            var existingValue:String=findViewById<TextView>(cellId).text.toString()
            if (existingValue.length==0)break
            seqStr=seqStr+existingValue
        }
        if (seqStr=="000" || seqStr=="XXX") {
            for (cellId in possible){
                findViewById<TextView>(cellId).setBackgroundColor(Color.parseColor("#eeeeee"))
            }
            return true
        }
    }
    return false
}
}

```



More Interactivity for Flutter Apps

- **Tic Tac Toe Game**
 - How to design using Flutter ?



MaterialApp

home : Scaffold

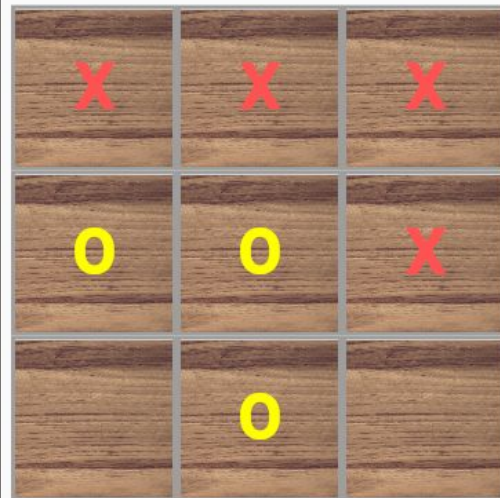
appBar: AppBar

body: Column

Tic Tac Toe

DEBUG

Game Over, Congrats to Player X



Play Again

MaterialApp

home : Scaffold

appBar: AppBar

title: Text

body: Column

Text

Row X 3

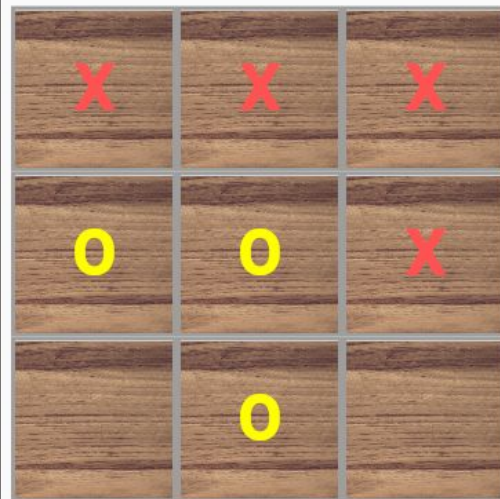
Cell X 3

Button

Tic Tac Toe

DEBUG

Game Over, Congrats to Player X



Play Again

MaterialApp

home : Scaffold

appBar: AppBar

body: Column

If (gameover)

title: Text

Text

Row X 3

Cell X 3

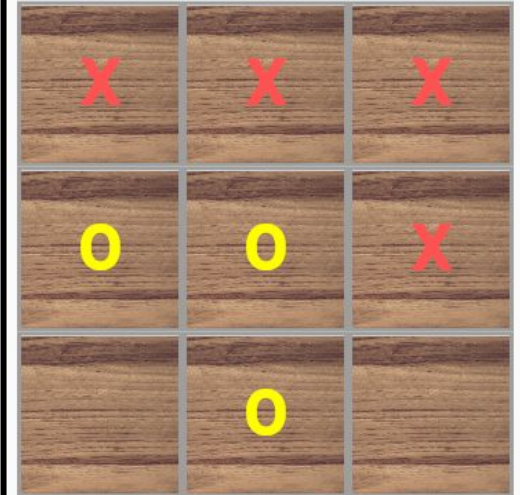
If (gameover)

Button

Tic Tac Toe

DEBUG

Game Over, Congrats to Player X



Play Again

MaterialApp

home : Scaffold

appBar: AppBar

body: Column

Visibility

Row X 3

Visibility

Text

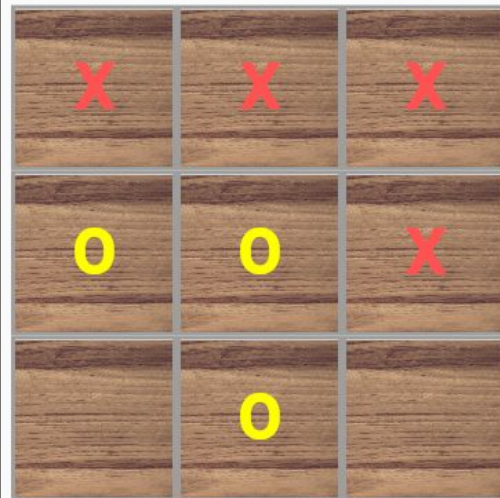
Cell X 3

Button

Tic Tac Toe

DEBUG

Game Over, Congrats to Player X



Play Again

```
import 'package:flutter/material.dart' ;
import 'package:tic_tac_toe_no_bl/screens/homescreen.dart' ;
```

```
void main() {
  runApp(const MainApp());
}

class MainApp extends StatelessWidget {
  const MainApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: HomeScreen(),
    );
  }
}
```

```
//homescreen.dart
```

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

```
class HomeScreen extends StatefulWidget {
  @override
  _HomeScreenState createState() => _HomeScreenState();
}
```

```
class HomeScreenState extends State<HomeScreen> {
  var boardSize = 3;
  var currentPlayer = 'X';

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text('Tic Tac Toe')),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          crossAxisAlignment: CrossAxisAlignment.center,
          children: [
```

```

children: [
  SizedBox(height: 40),
  Visibility(visible: false,
    child: Text(
      'Game Over, Congrats to Player X',
      style: TextStyle(color: Colors.green, fontSize: 25),
    ),
  ),
  Visibility(visible: true,
    child: Text(
      'Turn of Payer $currentPlayer',
      style: TextStyle(color: Colors.black26, fontSize: 25),
    ),
  ),
  SizedBox(height: 30),
  for (int row = 0; row < boardSize; row++)
    Row(
      mainAxisAlignment: MainAxisAlignment.center,
      crossAxisAlignment: CrossAxisAlignment.center,
      children: [
        for (int col = 0; col < boardSize; col++) getCell()
      ],
    ),
  SizedBox(height: 30),
  Visibility(
    visible: true,
    child: ElevatedButton(
      onPressed: () {
        setState(() {});
      },
      child: Text("Play Again"),
    ),
  ),
]

```



```

Widget getCell() {
  return GestureDetector(
    onTap: () {},
    child: Container(
      decoration: BoxDecoration(
        border: Border.all(color: Colors.grey, width: 3),
      ),
      width: 120,
      height: 120,
      child: Stack(
        children: [
          Positioned.fill(
            child: Opacity(
              opacity: 0.8,
              child: Image.asset('assets/images/wood.png', fit: BoxFit.cover),
            ),
          ),
          Center(
            child: Text('',
              style: TextStyle(
                fontSize: 45,
                fontWeight: FontWeight.bold,
              )),
          ),
        ],
      ),
    ),
  );
}

```

More Interactive Flutter App

- **Tic Tac Toe Game**
 - Create it using Flutter

```
//homescreen.dart

import 'package:flutter/material.dart';

class HomeScreen extends StatefulWidget {
  @override
  _HomeScreenState createState() => _HomeScreenState();
}

class HomeScreenState extends State<HomeScreen> {
  var boardSize = 3;
  var currentPlayer = 'X';

  Map<String, String> data = {};
  bool gameover = false;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text('Tic Tac Toe')),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          crossAxisAlignment: CrossAxisAlignment.center,
          children: [
```

More Interactive Flutter App

- **Tic Tac Toe Game**
 - Create it using Flutter

```
//homescreen.dart

import 'package:flutter/material.dart';

class HomeScreen extends StatefulWidget {
  @override
  _HomeScreenState createState() => _HomeScreenState();
}

class HomeScreenState extends State<HomeScreen> {
  var boardSize = 3;
  var currentPlayer = 'X';

  Map<String, String> data = {};
  bool gameover = false;

  void init_game_data() {
    data = {};
    gameover = false;
    currentPlayer = 'X';
    for (int r = 0; r < boardSize; r++) {
      for (int c = 0; c < boardSize; c++) {
        data[r.toString() + "x" + c.toString()] = '';
      }
    }
  }
}
```

More Interactive Flutter App

- **Tic Tac Toe Game**
 - Create it using Flutter

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

class HomeScreen extends StatefulWidget {
  @override
  _HomeScreenState createState() => _HomeScreenState();
}

class HomeScreenState extends State<HomeScreen> {
  var boardSize = 3;
  var currentPlayer = 'X';
  Map<String, String> data = {};
  bool gameover = false;
  void init_game_data() {
    data = {};
    gameover = false;
    currentPlayer = 'X';
    for (int r = 0; r < boardSize; r++) {
      for (int c = 0; c < boardSize; c++) {
        data[r.toString() + "x" + c.toString()] = '';
      }
    }
  }

  @override
  void initState() {
    super.initState();
    init_game_data();
  }
}
```

More Interactive Flutter App

- **Tic Tac Toe Game**
 - Create it using Flutter

```
Widget getCell() {  
  return GestureDetector(  
    onTap: () {},  
    child: Container(  
      decoration: BoxDecoration(  
        border: Border.all(color: Colors.grey, width: 3),  
      ),  
      width: 120,  
      height: 120,  
      child: Stack(  
        children: [  
          Positioned.fill(  
            child: Opacity(  
              opacity: 0.8,  
              child: Image.asset('assets/images/wood.png',  
                                fit: BoxFit.cover),  
            ),  
          ),  
          Center(  
            child: Text('',  
              style: TextStyle(  
                fontSize: 45,  
                fontWeight: FontWeight.bold,  
              )),  
          ),  
        ],  
      ),  
    ),  
  );  
}
```

More Interactive Flutter App

- **Tic Tac Toe Game**
 - Create it using Flutter

```
Widget getCell(int row, int col) {  
  var dataKey = row.toString() + "x" + col.toString();  
  Color myColor = Colors.redAccent;  
  if (data[dataKey] == 'O') myColor = Colors.yellowAccent;  
  return GestureDetector(  
    onTap: () {},  
    child: Container(  
      decoration: BoxDecoration(  
        border: Border.all(color: Colors.grey, width: 3),  
      ),  
      width: 120,  
      height: 120,  
      child: Stack(  
        children: [  
          Positioned.fill(  
            child: Opacity(  
              opacity: 0.8,  
              child: Image.asset('assets/images/wood.png',  
                                fit: BoxFit.cover),  
            ),  
          ),  
          Center(  
            child: Text('',  
              style: TextStyle(  
                fontSize: 45,  
                fontWeight: FontWeight.bold,  
              )),  
          ),  
        ],  
      ),  
    ),  
  );  
}
```

More Interactive Flutter App

- Tic Tac Toe Game
 - Create it using Flutter

```
Widget getCell(int row, int col) {  
  var dataKey = row.toString() + "x" + col.toString();  
  Color myColor = Colors.redAccent;  
  if (data[dataKey] == 'O') myColor = Colors.yellowAccent;  
  return GestureDetector(  
    onTap: () {  
      if (gameover) return;  
      if (data[dataKey] == '') {  
        data[dataKey] = currentPlayer;  
        gameover = isWin();  
        if (!gameover) {  
          switchPlayerTurn();  
        }  
        setState(() {});  
      },  
    child: Container(  
      decoration: BoxDecoration(  
        border: Border.all(color: Colors.grey, width: 3),  
      ),  
      width: 120,  
      height: 120,  
      child: Stack(  
        children: [  
          Positioned.fill(  
            child: Opacity(  
              opacity: 0.8,  
              child: Image.asset('assets/images/wood.png',  
                                fit: BoxFit.cover),  
            ),  
          ),  
        ],  
      ),  
    ),  
  ),  
  Center(  
    child: Text('')  
  ),  
);
```

```

List<List<String>> possible_wins = [
  ['0x0', '0x1', '0x2'],
  ['1x0', '1x1', '1x2'],
  ['2x0', '2x1', '2x2'],
  ['0x0', '1x0', '2x0'],
  ['0x1', '1x1', '2x1'],
  ['0x2', '1x2', '2x2'],
  ['0x0', '1x1', '2x2'],
  ['0x2', '1x1', '2x0'],
];

bool isWin() {
  for (var line in possible_wins) {
    var result = '';
    for (var cell in line) {
      result = result + data[cell]!;
    }
    if (result == 'XXX' || result == 'OOO') {
      gameover = true;
      return true;
    }
  }
  return false;
}

void switchPlayerTurn() {
  if (currentPlayer == 'X') {
    currentPlayer = 'O';
  } else {
    currentPlayer = 'X';
  }
}

```

```

getCell(int row, int col) {
  dataKey = row.toString() + "x" + col.toString();
  for myColor = Colors.redAccent;
  (data[dataKey] == 'O') myColor = Colors.yellowAccent;
  return GestureDetector(
    onTap: () {
      if (gameover) return;
      if (data[dataKey] == '') {
        data[dataKey] = currentPlayer;
        gameover = isWin();
      }
      if (!gameover) {
        switchPlayerTurn();
      }
      setState(() {});
    },
    child: Container(
      decoration: BoxDecoration(
        border: Border.all(color: Colors.grey, width: 3),
      ),
      width: 120,
      height: 120,
      child: Stack(
        children: [
          Positioned.fill(
            child: Opacity(
              opacity: 0.8,
              child: Image.asset('assets/images/wood.png',
                fit: BoxFit.cover),
            ),
          ),
          Center(
            child: Text('')

```



```

List<List<String>> possible_wins = [
  ['0x0', '0x1', '0x2'],
  ['1x0', '1x1', '1x2'],
  ['2x0', '2x1', '2x2'],
  ['0x0', '1x0', '2x0'],
  ['0x1', '1x1', '2x1'],
  ['0x2', '1x2', '2x2'],
  ['0x0', '1x1', '2x2'],
  ['0x2', '1x1', '2x0'],
];

bool isWin() {
  for (var line in possible_wins) {
    var result = '';
    for (var cell in line) {
      result = result + data[cell]!;
    }
    if (result == 'XXX' || result == 'OOO') {
      gameover = true;
      return true;
    }
  }
  return false;
}

void switchPlayerTurn() {
  if (currentPlayer == 'X') {
    currentPlayer = 'O';
  } else {
    currentPlayer = 'X';
  }
}

```

```

getCell(int row, int col) {
  dataKey = row.toString() + "x" + col.toString();
  for myColor = Colors.redAccent;
  (data[dataKey] == 'O') myColor = Colors.yellowAccent;
  return GestureDetector(
    onTap: () {
      if (gameover) return;
      if (data[dataKey] == '') {
        data[dataKey] = currentPlayer;
        gameover = isWin();
      }
      if (!gameover) {
        switchPlayerTurn();
      }
      setState(() {});
    },
    child: Container(
      decoration: BoxDecoration(
        border: Border.all(color: Colors.grey, width: 3),
      ),
      width: 120,
      height: 120,
      child: Stack(
        children: [
          Positioned.fill(
            child: Opacity(
              opacity: 0.8,
              child: Image.asset('assets/images/wood.png',
                fit: BoxFit.cover),
            ),
          ),
          Center(
            child: Text('')

```

```

children: [
  SizedBox(height: 40),
  Visibility(visible: false,
    child: Text(
      'Game Over, Congrats to Player X',
      style: TextStyle(color: Colors.green, fontSize: 25),
    ),
  ),
  Visibility(visible: true,
    child: Text(
      'Turn of Payer $currentPlayer',
      style: TextStyle(color: Colors.black26, fontSize: 25),
    ),
  ),
  SizedBox(height: 30),
  for (int row = 0; row < boardSize; row++)
    Row(
      mainAxisAlignment: MainAxisAlignment.center,
      crossAxisAlignment: CrossAxisAlignment.center,
      children: [
        for (int col = 0; col < boardSize; col++) getCell()
      ],
    ),
  SizedBox(height: 30),
  Visibility(
    visible: true,
    child: ElevatedButton(
      onPressed: () {
        setState(() {});
      },
      child: Text("Play Again"),
    ),
  ),
]

```

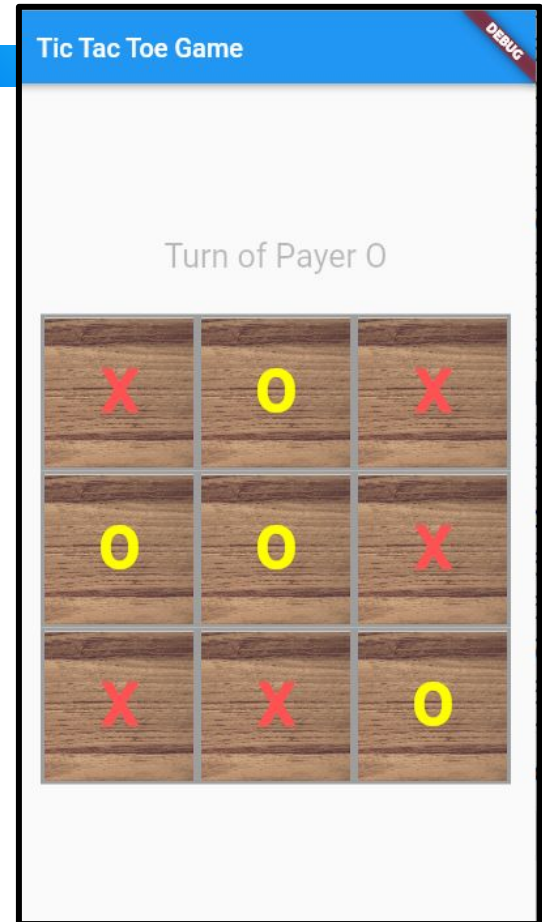
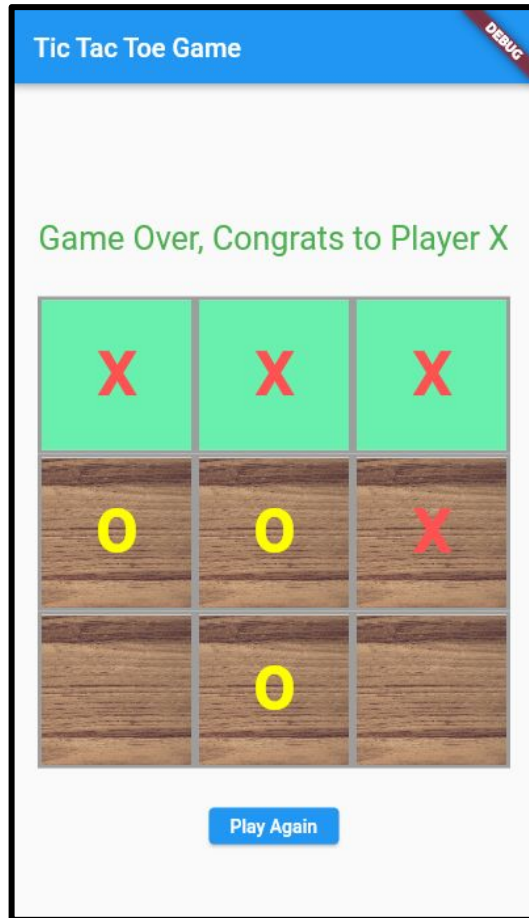
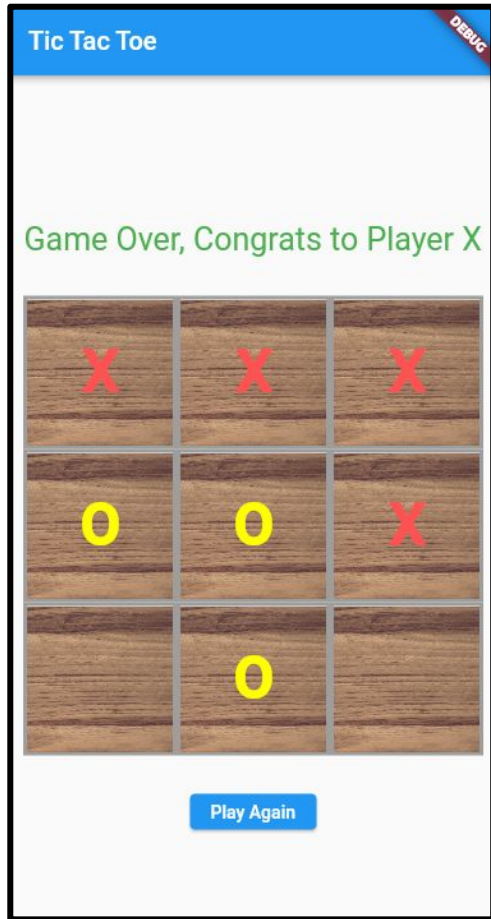
```

children: [
  SizedBox(height: 40),
  Visibility(visible: gameOver,
    child: Text(
      'Game Over, Congrats to Player X!',
      style: TextStyle(color: Colors.green, fontSize: 25),
    ),
  ),
  Visibility(visible: !gameOver,
    child: Text(
      'Turn of Payer $currentPlayer',
      style: TextStyle(color: Colors.black26, fontSize: 25),
    ),
  ),
  SizedBox(height: 30),
  for (int row = 0; row < boardSize; row++)
    Row(
      mainAxisAlignment: MainAxisAlignment.center,
      crossAxisAlignment: CrossAxisAlignment.center,
      children: [
        for (int col = 0; col < boardSize; col++) getCell(row , col)
      ],
    ),
  SizedBox(height: 30),
  Visibility(
    visible: true,
    child: ElevatedButton(
      onPressed: () {
        init_game_data();
        setState(() {});
      },
      child: Text("Play Again")),
  ),
)

```

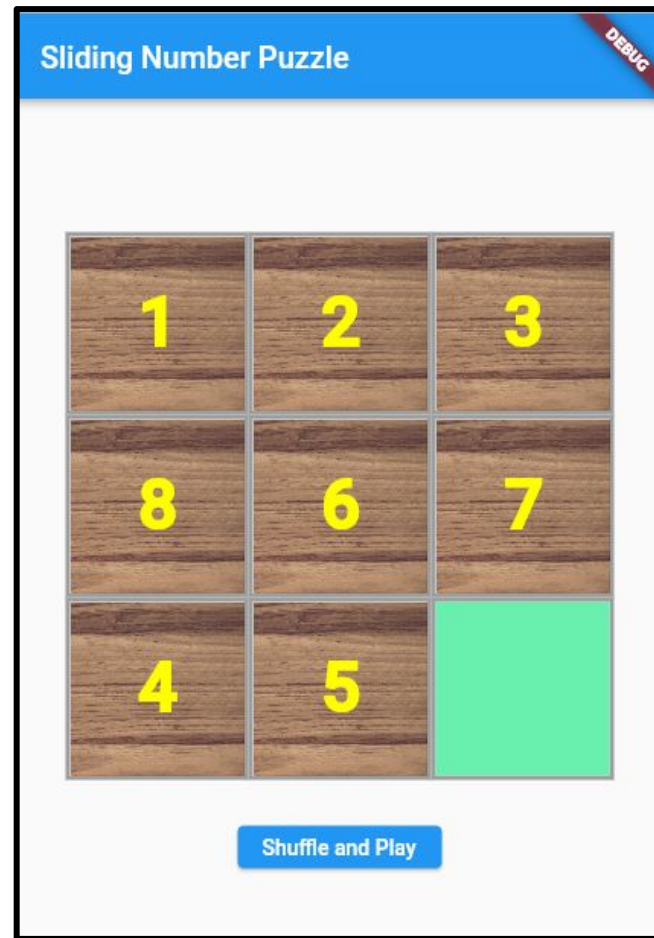
Coloring the winning Cells ?

When it is a draw ?



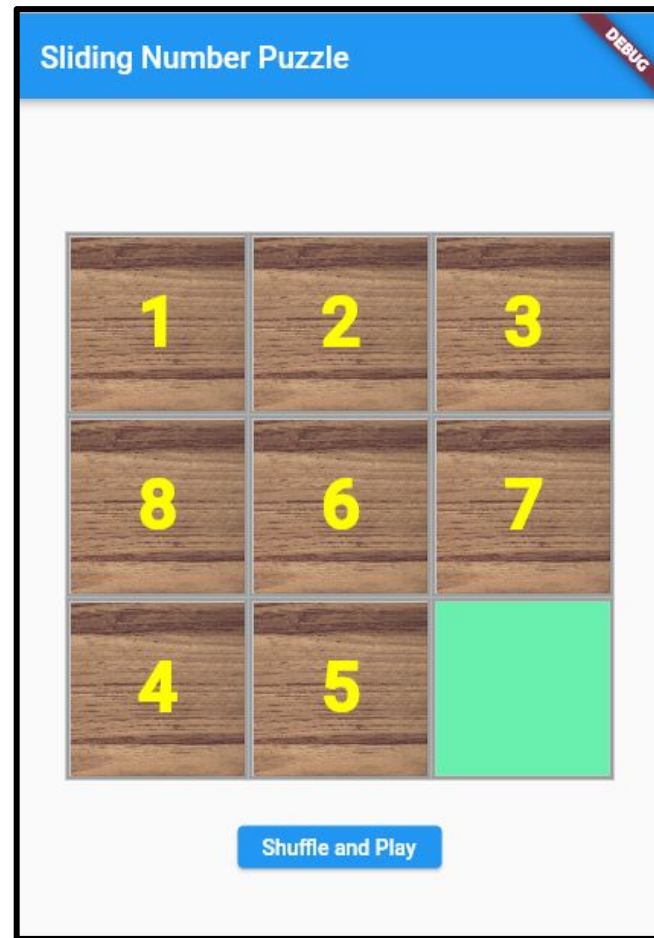
More Interactivity for Flutter Apps

- Sliding Number Puzzle
 - Flutter UI



More Interactivity for Flutter Apps

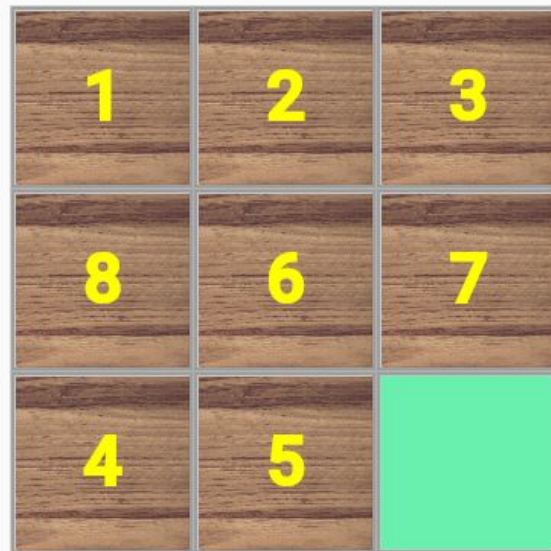
- Sliding Number Puzzle
 - Data Structures



```
List<List<String>> data = [];  
int gridSize = 3;  
int empty_row = 2;  
int empty_col = 2;  
  
void init_game_data() {  
    data = [];  
    empty_row = gridSize - 1;  
    empty_col = gridSize - 1;  
  
    int inc = 1;  
    for (int r = 0; r < gridSize; r++) {  
        List<String> row = [];  
        for (int c = 0; c < gridSize; c++) {  
            row.add(inc.toString());  
            inc = inc + 1;  
        }  
        data.add(row);  
    }  
    data[gridSize - 1][gridSize - 1] = '';  
}
```

Sliding Number Puzzle

DEBUG

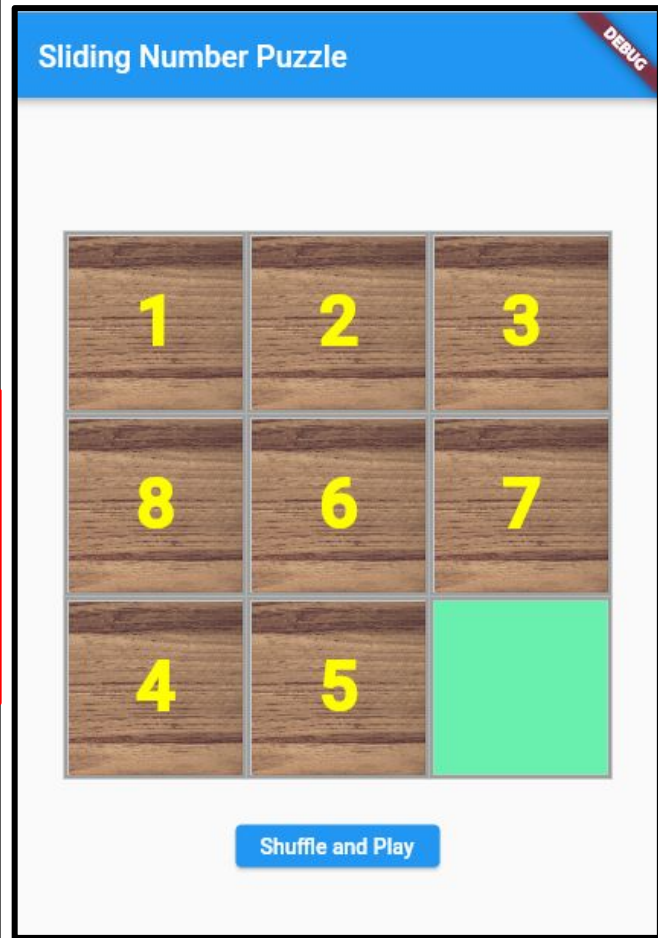


Shuffle and Play

```

Widget getCell(int row, int col) {
  return GestureDetector(
    onTap: () {
      bool ret = moveSquare(row, col);
      if (ret) setState(() {});
    },
    child: Container(
      decoration: BoxDecoration(
        border: Border.all(color: Colors.grey, width: 3),,
        width: 120,
        height: 120,
        child: Stack(
          children: [
            if (data[row][col] != '')
              Positioned.fill(
                child: Opacity(
                  opacity: 0.8,
                  child:
                    Image.asset('assets/images/wood.png',
                      fit: BoxFit.cover),),),),
            if (data[row][col] == '')
              Container(color: Colors.greenAccent),
          ],
          Center(
            child: Text('${data[row][col]}',
              style: const TextStyle(
                fontSize: 45,
                fontWeight: FontWeight.bold,
                color: Colors.yellowAccent))),
        ],
      ),
    ); }

```



More Interactivity for Flutter Apps

- **Sliding Number Puzzle**
 - Doing the shuffle ?



Section 2

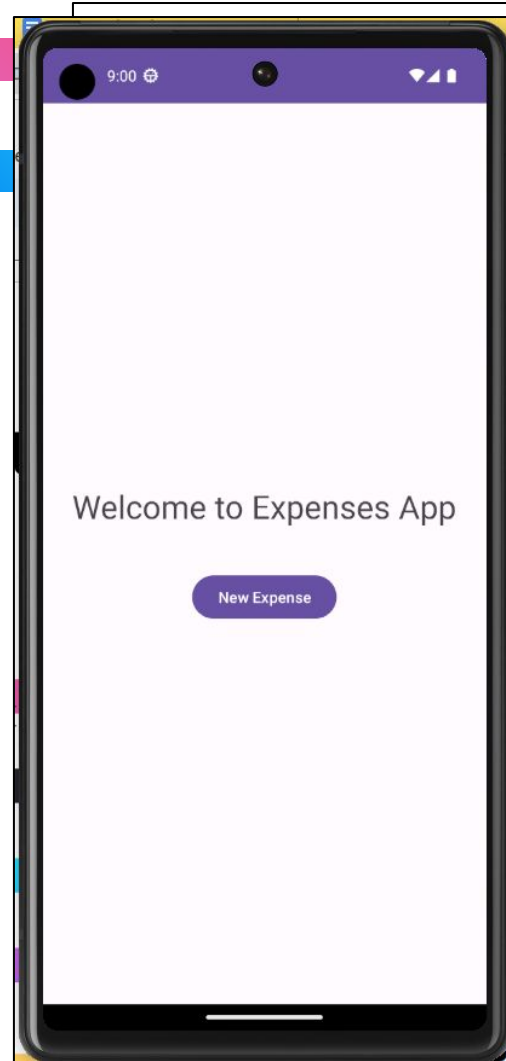
Navigation and Routes



Navigation across Multiple Screens in Flutter

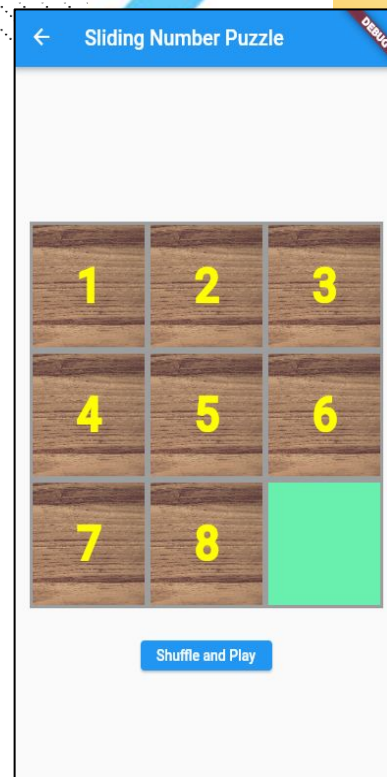
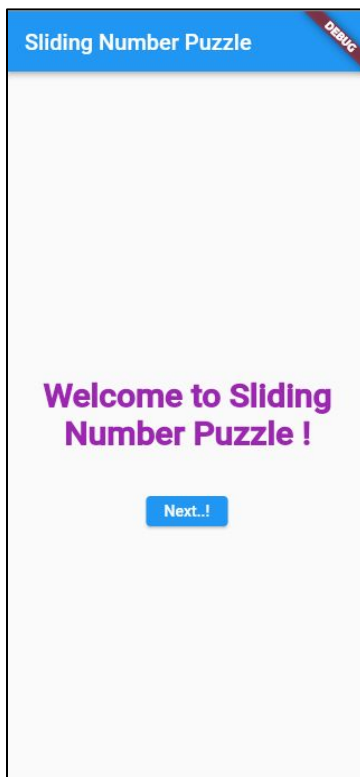
- Reminder for Android Native Development/Kotlin

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.activity_main)  
  
    var bt_new=findViewById<Button>(R.id.bt_new)  
    bt_new.setOnClickListener {  
        val intent = Intent(this, NewExpense::class.java)  
        intent.putExtra("some variable", "HelloWorld !")  
        intent.putExtra("another variable", 12)  
        this.startActivity(intent)  
    }  
}
```



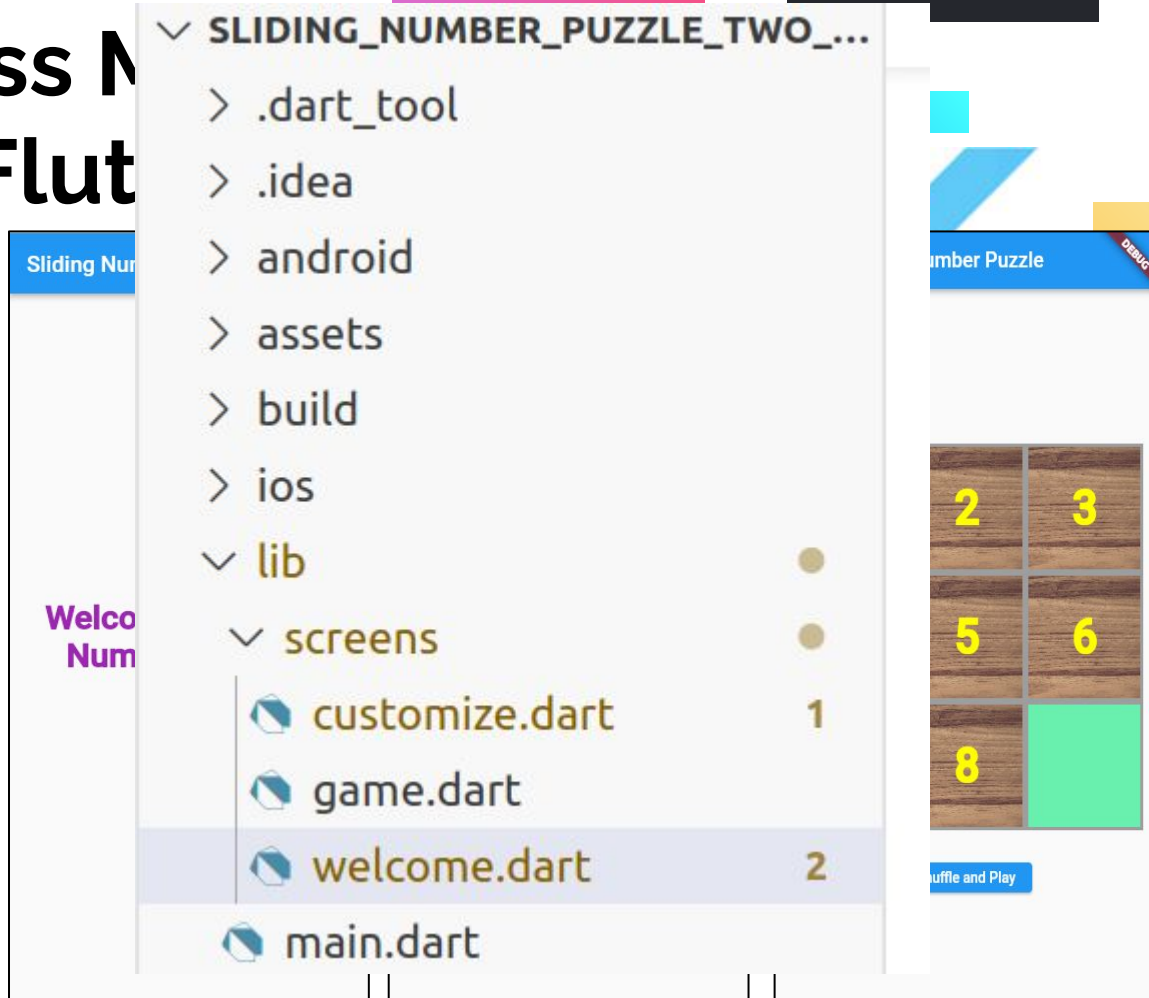
Navigation across Multiple Screens in Flutter

- **Sliding Number Puzzle :**
 - **Three Screens**
 - Welcome Screen
 - welcome.dart
 - Customize Screen
 - customize.dart
 - Game Screen
 - game.dart



Navigation across Multiple Screens in Flutter

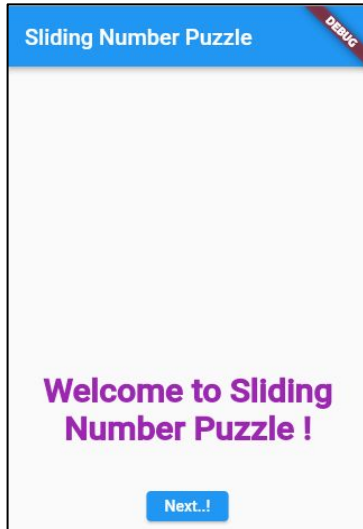
- **Sliding Number Puzzle :**
 - **Three Screens**
 - Welcome Screen
 - welcome.dart
 - Customize Screen
 - customize.dart
 - Game Screen
 - game.dart



Navigation across Screens in

- Sliding Number Puzzle :

- main.dart File :



```
import 'package:flutter/material.dart';
import
'package:sliding_number_puzzle_two_screens/screens/customize.dart';
import 'package:sliding_number_puzzle_two_screens/screens/game.dart';
import
'package:sliding_number_puzzle_two_screens/screens/welcome.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: WelcomeScreen(),
      routes: {
        HomeScreen.pageRoute: (ctx) => HomeScreen(),
        WelcomeScreen.pageRoute: (ctx) => WelcomeScreen(),
        CustomizeScreen.pageRoute: (ctx) => CustomizeScreen(),
      },
    );
  }
}
```

Navigation across Multiple Screens in Flutter

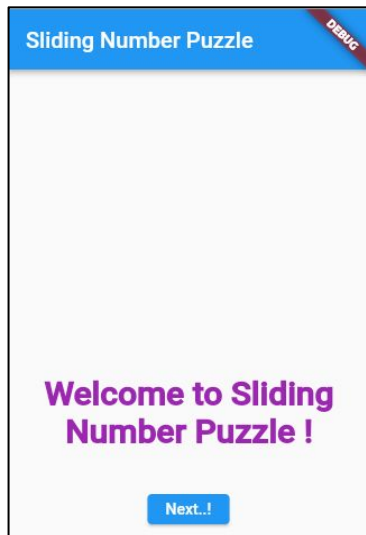


- **Routes and Navigation in Flutter**

- main.dart File :

Navigation across Screens in

- Sliding Number Puzzle :
 - Welcome Screen :



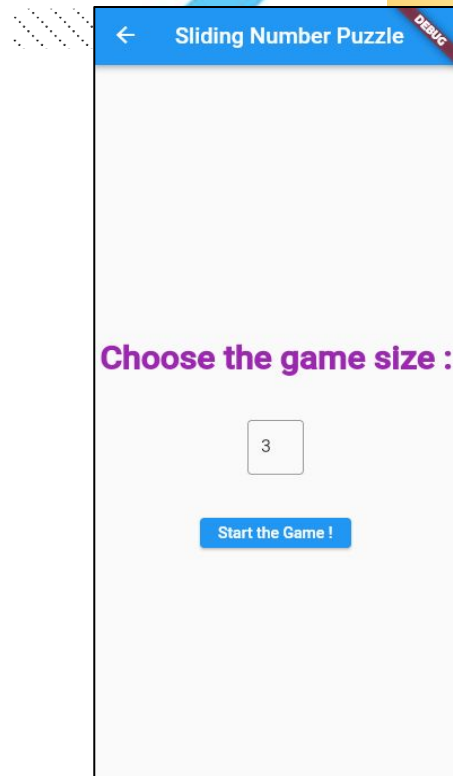
```
import 'package:flutter/material.dart';
import 'package:sliding_number_puzzle_two_screens/screens/customize.dart';

class WelcomeScreen extends StatelessWidget {
  static const String pageRoute = '/welcomescreen';

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text('Sliding Number Puzzle')),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          crossAxisAlignment: CrossAxisAlignment.center,
          children: [
            const Text(
              'Welcome to Sliding Number Puzzle !',
              textAlign: TextAlign.center,
              style: TextStyle(
                color: Colors.purple,
                fontWeight: FontWeight.bold,
                fontSize: 30),
            ),
            const SizedBox(height: 40),
            ElevatedButton(
              onPressed: () {
                Navigator.pushNamed(context, CustomizeScreen.pageRoute);
              },
              child: Text("Next..!"),
            ),
          ],
        ),
      ),
    ),
  ),
);
```


Navigation across Multiple Screens in Flutter

- **Second Screen : Customize.dart**
 - Stateless or Stateful ?



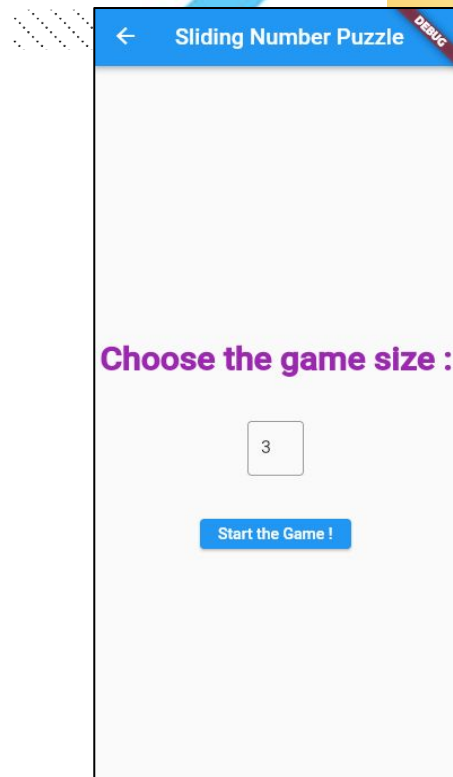
Navigation across Multiple Screens in Flutter

- **Second Screen : Customize.dart**

- Stateless or Stateful ?

(Of course, you can make everything stateful at the expense of performance)

- It has a TextFormField (It's value can change ?)



Navigation across Multiple Screens in Flutter

- **Second Screen : Customize.dart**
 - Stateless or Stateful ?

If you don't need to refresh the UI with a dynamic value, stateless would do. (Input entered by the user does not require UI refresh, whilst the slider/switch need a refresh)



```
import 'package:flutter/material.dart';
import 'package:sliding_number_puzzle_two_screens/screens/game.dart';

class CustomizeScreen extends StatelessWidget {
  static const String pageRoute = '/customizescreen';

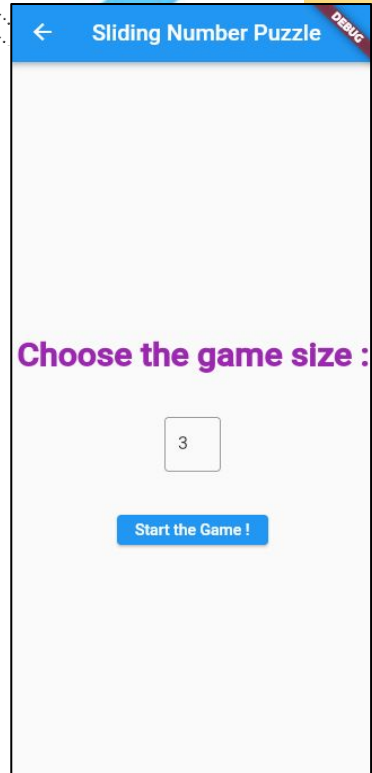
  String tx_size_value = '3';

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text('Sliding Number Puzzle')),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          crossAxisAlignment: CrossAxisAlignment.center,
          children: [

            ...

          ]),
    ),
  );
}
```

You can have instance variable in a stateless widget



```
ElevatedButton(  
  onPressed: () {  
    Navigator.pushNamed(context, WelcomeScreen.pageRoute);  
  },  
  child: Text('Back'),  
)
```

OR ?

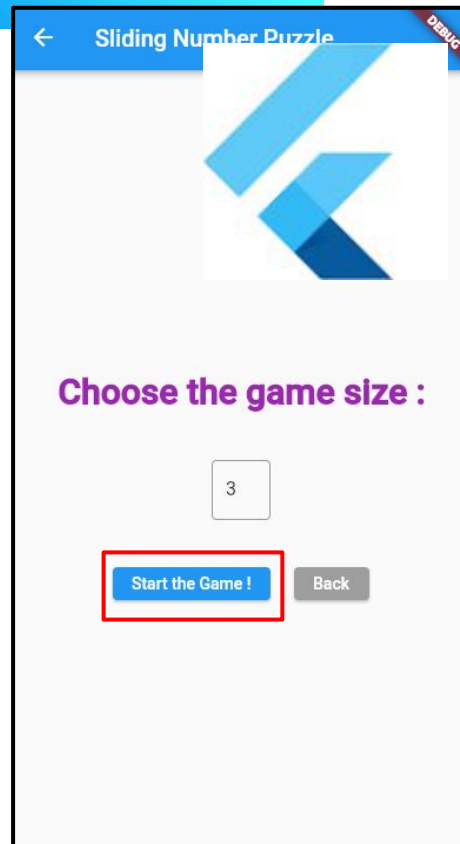
```
ElevatedButton(  
  onPressed: () {  
    Navigator.pop(context);  
  },  
  child: Text('Back'),  
)
```



The value 3 needs to be sent to the child screen (next screen)

Remember : Flutter is declarative

Not imperative programming : (`someRef.getText()` no longer works)



```

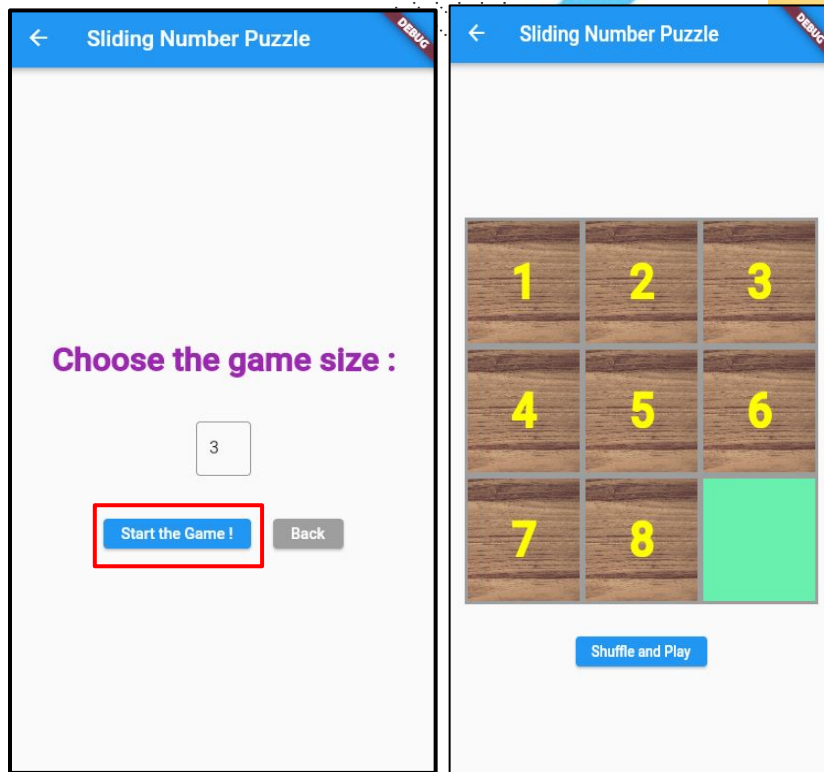
children: [
  const Text(
    'Choose the game size : ', textAlign: TextAlign.center,
    style: TextStyle(
      color: Colors.purple,
      fontWeight: FontWeight.bold, fontSize: 30),
  ),
  const SizedBox(height: 40),
  SizedBox(
    width: 50,
    child: TextFormField(
      decoration: const InputDecoration(
        border: OutlineInputBorder(
          borderSide: BorderSide(color: Colors.teal)),
      ),
      initialValue: '3',
      keyboardType: TextInputType.number,
      onChanged: (newValue) {
        debugPrint('> Printing new value $newValue');
        _tx_size_value = newValue;
      },
    ),
  ),
  const SizedBox(height: 40),
  ElevatedButton(
    onPressed: () {
      Navigator.of(context).push(MaterialPageRoute(
        builder: (context) => HomeScreen(myData: {
          'myBoardSize': _tx_size_value,
          'otherDataYouWish': 'hello',
        })));
    },
    child: Text("Start the Game !"),
  ),

```



Navigation across Multiple Screens in Flutter

- **Passing Data between Screens :**
 - The recommended way is to pass through the child constructor.
 - You may can use a Map Object to pass a variety of variables.



Navigation across Multiple Screens in Flutter



- **Passing Data between Screens :**

- Another way

Sending

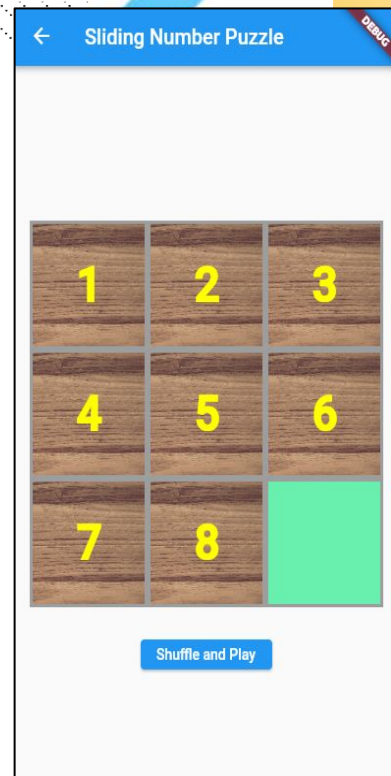
```
Navigator.pushNamed(context, someRoute, arguments: [  
    'Title of Screen',  
    ['data', 'another data'],  
    'some more'  
]),
```

Getting

```
@override  
Widget build(BuildContext context) {  
    final List args = (ModalRoute.of(context)!.settings.arguments) as List;  
    appBarTitle = args[0];  
    listOfItems = args[1];  
    infoOrState = args[2];  
}
```

Navigation across Multiple Screens in Flutter

- **Third Screen : game.dart**
 - Must be stateful
 - How to get the data passed from the parent screen ?



Navigation across Multiple Screens in Flutter

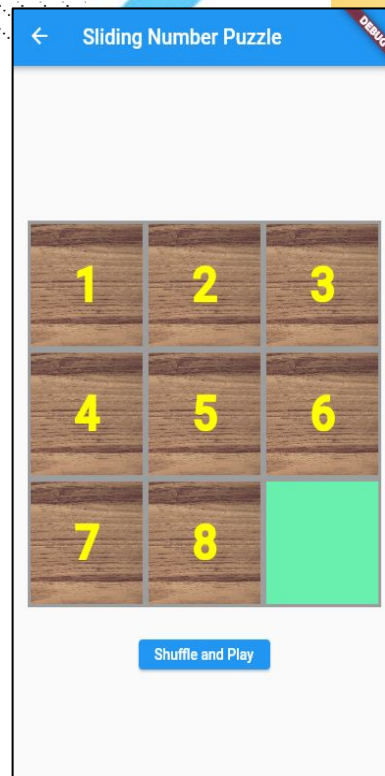
- Third Screen : game.dart

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

class HomeScreen extends StatefulWidget {
  static const String pageRoute = '/homescreen';
  final Map<String, String>? myData;

  const HomeScreen({super.key, this.myData});

  @override
  _HomeScreenState createState() => _HomeScreenState();
}
```



Navigation across Multiple Screens in Flutter

- Third Screen : game.dart

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

class HomeScreen extends StatefulWidget {
  static const String pageRoute = '/homescreen';
  final Map<String, String>? myData;
```

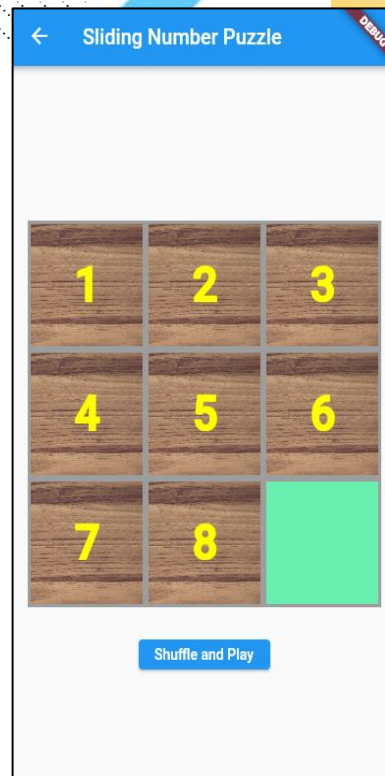
```
const
```

```
@over
```

```
_Home
```

```
}
```

How to pass the data to the Widget State Object ?



```

class _HomeScreenState extends State<HomeScreen> {
  List<List<String>> data = [];
  int gridSize = 3;
  int empty_row = 2;
  int empty_col = 2;

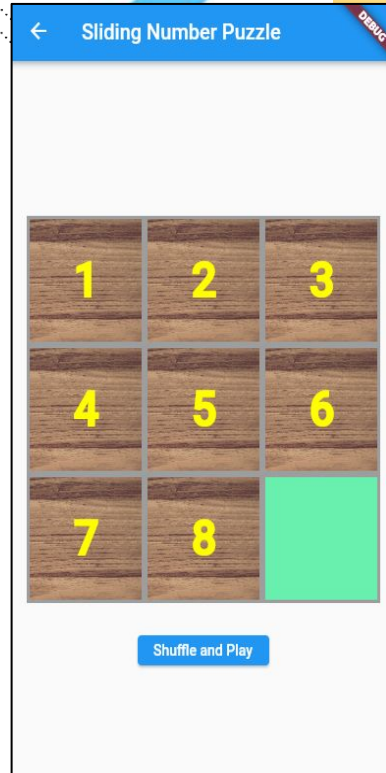
  void init_game_data() {
    try {
      gridSize = int.parse(widget.myData!['myBoardSize']!);
    } catch (e) { gridSize = 3; }

    data = [];
    empty_row = gridSize - 1;
    empty_col = gridSize - 1;

    int inc = 1;
    for (int r = 0; r < gridSize; r++) {
      List<String> row = [];
      for (int c = 0; c < gridSize; c++) {
        row.add(inc.toString());
        inc = inc + 1;
      }
      data.add(row);
    }
    data[gridSize - 1][gridSize - 1] = '';
  }

  @override
  void initState() {
    super.initState();
    init_game_data();
  }
}

```



Navigation across Multiple Screens in Flutter

- **Passing Data between Screens : Other ways ?**
 - The recommended way is to pass through the child constructor.
 - You may can use a Map Object to pass a variety of variables.

Navigation across Multiple Screens in Flutter

- Ack from the Child to Parent when the Child is killed ?
 - In Kotlin

```
3 var bt_new=findViewById<Button>(R.id.bt_new)
  bt_new.setOnClickListener {
    val intent = Intent(this, NewExpense::class.java)
    intent.putExtra("some_variable", "HelloWorld !")
    intent.putExtra("another_variable", 12)
    launchActivityNewExpense.launch(intent)
  }
}

var launchActivityNewExpense = registerForActivityResult(
    ActivityResultContracts.StartActivityForResult()) { result ->
    if (result.resultCode == Activity.RESULT_OK) {
        println("Get data back<<<<<<")
        printExpenses()
    }
}
```

Navigation across Multiple Screens in Flutter



- **Validating User Input**

1. Define a global key variable for the form
2. Wrap the form elements with Form, with the key attribute variable.
3. Define the validator attribute for the TextFormfield to return error message in case
4. On submit the form, call the form validate() method.

Navigation across Multiple Screens in Flutter

- **Validating User Input**

1. **Define a global key variable for the form**
2. Wrap the form elements with Form, with the key attribute variable.
3. Define the validator attribute for the TextFormField to return error message in o
4. On submit th

```
class CustomizeScreen extends StatelessWidget {  
  static const String pageRoute = '/customizescreen';  
  
  String _tx_size_value = '3';  
  final _formKey = GlobalKey<FormState>();  
}
```

Navigation across Multiple Screens in Flutter

- **Validating User Input**

1. Define a global key variable for the form
2. **Wrap the form elements with Form, with the key attribute variable.**
3. Define the validator attribute for the TextFormField to return error message in case of invalid input
4. On submit the form, call the validator method

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: const Text('Sliding Number Puzzle')),
    body: Form(
      key: _formKey,
      child: Center(
        child: Column(
```

Navigation across Screens in

- **Validating User Input**

1. Define a global key
2. Wrap the form elements with Form, with the key attribute variable.
3. **Define the validator attribute for the TextFormField to return error message in case**
4. On submit the form, call the form validate() method.

```
child: TextFormField(  
  
  initialValue: '3',  
  keyboardType: TextInputType.number,  
  onChanged: (newValue) {  
    debugPrint('> Printing new value $newValue');  
    _tx_size_value = newValue;  
  },  
  validator: (value) {  
    if (value == null ||  
        int.tryParse(value.toString()) == null) {  
      return 'invalid number';  
    }  
    return null;  
  },  
)
```

Navigation across Multiple Screens in Flutter

- **Validating User Input**

1. Define a global key
2. Wrap the form elements
3. Define the validation logic and display an error message in case of failure

```
ElevatedButton(  
  onPressed: () {  
    if (!_formKey.currentState!.validate()) {  
      return;  
    }  
    Navigator.of(context).push(MaterialPageRoute(  
      builder: (context) => HomeScreen(myData: {  
        'myBoardSize': tx size value,  
        'otherDataYouWish': 'hello',  
      })));  
  },  
  child: Text("Start the Game !"),  
),
```

4. **On submitting the form, call the form validate() method.**

Section 3

Navigational Widgets



Navigational Widgets for Flutter Apps

- **AppBar**

```
import 'package:flutter/material.dart';
import 'package:navigation_hello_world/widgets/appbar.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        appBar: MyAppBar(myTitle: 'Homepage'),
        body: Center(
          child: Text('Hello World!'),
        ),
      ),
    );
  }
}
```

☰ Homepage



```

import 'package:flutter/material.dart';

class MyAppBar extends StatelessWidget implements PreferredSizeWidget {
  String myTitle;

  @override
  final Size preferredSize = const Size.fromHeight(60.0);

  MyAppBar({super.key, this.myTitle = ''});

  @override
  Widget build(BuildContext context) {
    return AppBar(
      leading: IconButton(
        icon: const Icon(Icons.menu),
        onPressed: () {},
      ),
      title: Text(myTitle),
      actions: [
        IconButton(icon: const Icon(Icons.favorite), onPressed: () {}),
        IconButton(icon: const Icon(Icons.search), onPressed: () {}),
        PopupMenuButton<Text>(
          itemBuilder: (context) {
            return [
              const PopupMenuItem(child: Text('Option 1'),),
              const PopupMenuItem(child: Text('Option 2'),),
            ];
          },
        ),
      ],
    );
  }
}

```

This is a stateless Widget, if you want to interact with the fav button, it needs to be converted to a stateful widget

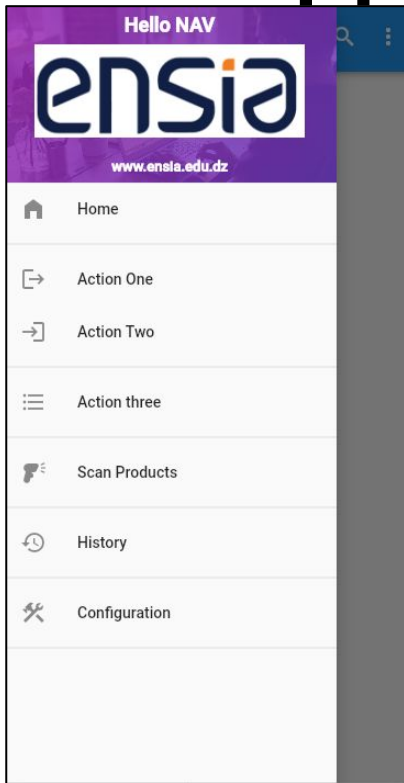
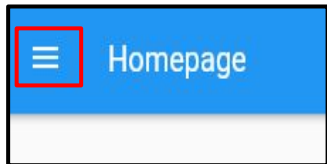
≡ Homepage



Hello World!

Navigational Widgets for Flutter Apps

- Drawer



```
import 'package:flutter/material.dart';
import '../widgets/appbar.dart';
import '../widgets/drawer.dart';

class HomeScreen extends StatefulWidget {
  static const String pageRoute = '/homescreen';
  const HomeScreen({super.key});
  @override
  State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      drawer: const SideBarDrawer(),
      appBar: MyAppBar(
        myTitle: 'Homepage',
      ),
      body: const Center(
        child: Text('Hello World!'),
      ));
  }
}
```



```

import 'package:flutter/material.dart';

class MyAppBar extends StatelessWidget implements PreferredSizeWidget {
  String myTitle;

  @override
  final Size preferredSize = const Size.fromHeight(60.0);

  MyAppBar({super.key, this.myTitle = ''});

  @override
  Widget build(BuildContext context) {
    return AppBar(
      leading: IconButton(
        icon: const Icon(Icons.menu),
        onPressed: () {},
      ),
      title: Text(myTitle),
      actions: [
        IconButton(icon: const Icon(Icons.favorite), onPressed: () {}),
        IconButton(icon: const Icon(Icons.search), onPressed: () {}),
        PopupMenuButton<Text>(
          itemBuilder: (context) {
            return [
              const PopupMenuItem(child: Text('Option 1'),),
              const PopupMenuItem(child: Text('Option 2'),),
            ];
          },
        ),
      ],
    );
  }
}

```

Must be removed

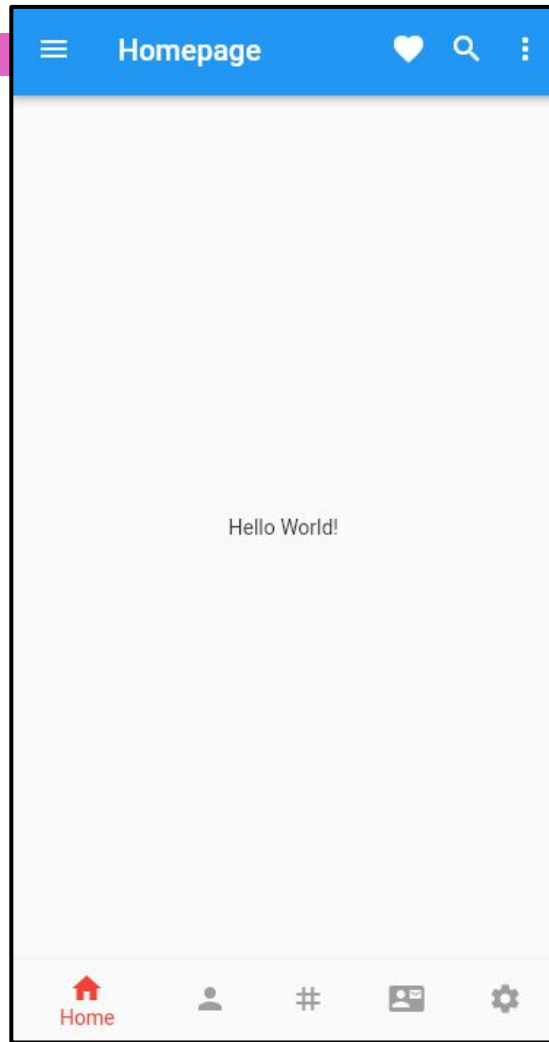
≡ Homepage



Hello World!

Navigational Widgets for Flutter Apps

- Other ways for navigation & interaction :
 - Bottom Bar
 - Floating Buttons
 - Tabs
 - ..



Lecture Demo Apps

- Reminder Examples

- <https://www.dropbox.com/scl/fo/ddnkks2s0k9qfwqlfx8ru/h?rlkey=c972kn3tsqjicthtgu4wekhp3&dl=0>
- <https://www.dropbox.com/scl/fo/hm3mkdvnhibf4macm78as/h?rlkey=zga7oxols3l72a82qt5kspui1&dl=0>

- Tic Tac Toe Game

- Without Business Logic:

- <https://www.dropbox.com/scl/fo/oskfiozepgyppc6z3lfmo/h?rlkey=0v2gwcgcw7w6nbstsqte58f78&dl=0>

- With Business Logic

- <https://www.dropbox.com/scl/fo/wv42hnx4r83g7huzhuc6e/h?rlkey=k4cx70oxnynv7cq5wclwcyndm&dl=0>

- Sliding Number Puzzle :

- <https://www.dropbox.com/scl/fo/tg5sgw9c72ekngs9u2zjm/h?rlkey=82a1jir7c66pdj7mjxob7kp6d&dl=0>

- Sliding Number Puzzle with Multiple Screens + Form validation :

- <https://www.dropbox.com/scl/fo/43h6f5uwgp98uywe2p4rq/h?rlkey=586k5xctnmi75nf1jwl0hugc3&dl=0>



Resources

- <https://docs.flutter.dev/cookbook/navigation/named-routes>
- <https://docs.flutter.dev/ui/navigation>
- <https://docs.flutter.dev/cookbook/design/tabs>

Title here

← Sport Activity



Big Title

Big description here ...Big
description here ...Big description
here ...Big description here ...Big
description here ...Big description
here ...Big description here ...Big
description here ...Big description
here ...Big description here ...Big
description here ...