droidcon academy

Building the UI

Simone Alessandria

in linkedin.com/in/simone-alessandria

Section Overview

- Building the User Interface of the app
- Main life cycle events of a Stateful Widget
- Using a TextEditingController
- Updating the UI with setState()

Speed Calculator UI

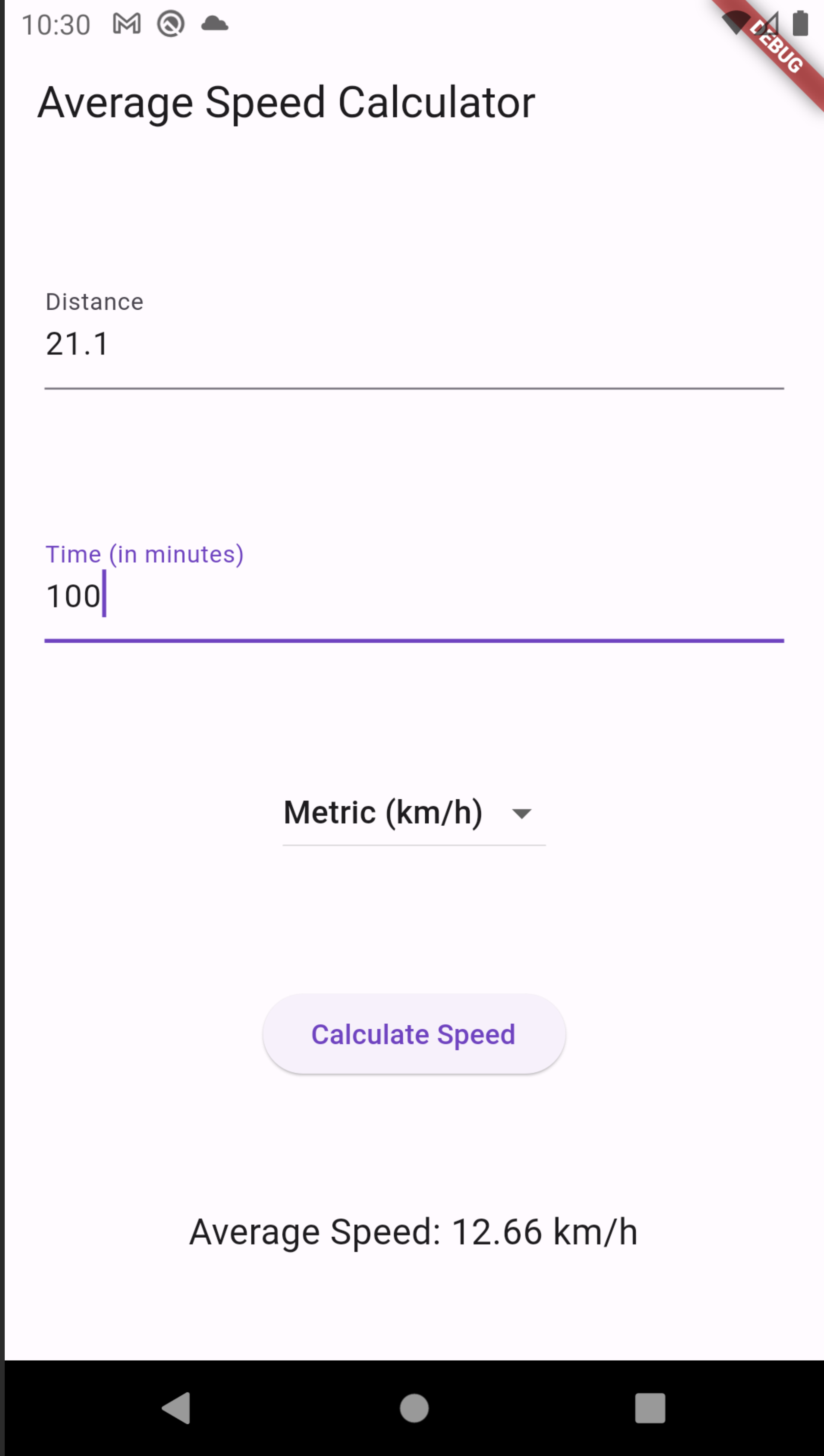
TextField →

TextField →

DropdownButton →

ElevatedButton →

Text →



Average Speed Calculator

Distance
21.1

Time (in minutes)
100

Metric (km/h)

Average Speed: 12.66 km/h

TextEditingController

A widget to control the text of a TextField (or a TextFormField).

Use to:

- Read and write text
- Listen for changes

```
TextEditingController controller =
TextEditingController();


// Get text
String currentText = controller.text;


// Set text
controller.text = "New text";


// Listen for changes
controller.addListener(() {
  print("New text: ${controller.text}");
});
```

Disposing TextEditingController

A TextEditingController takes resources and retains a small piece of memory.

Always dispose it to:

- Optimize resource usage
- Prevent memory leaks

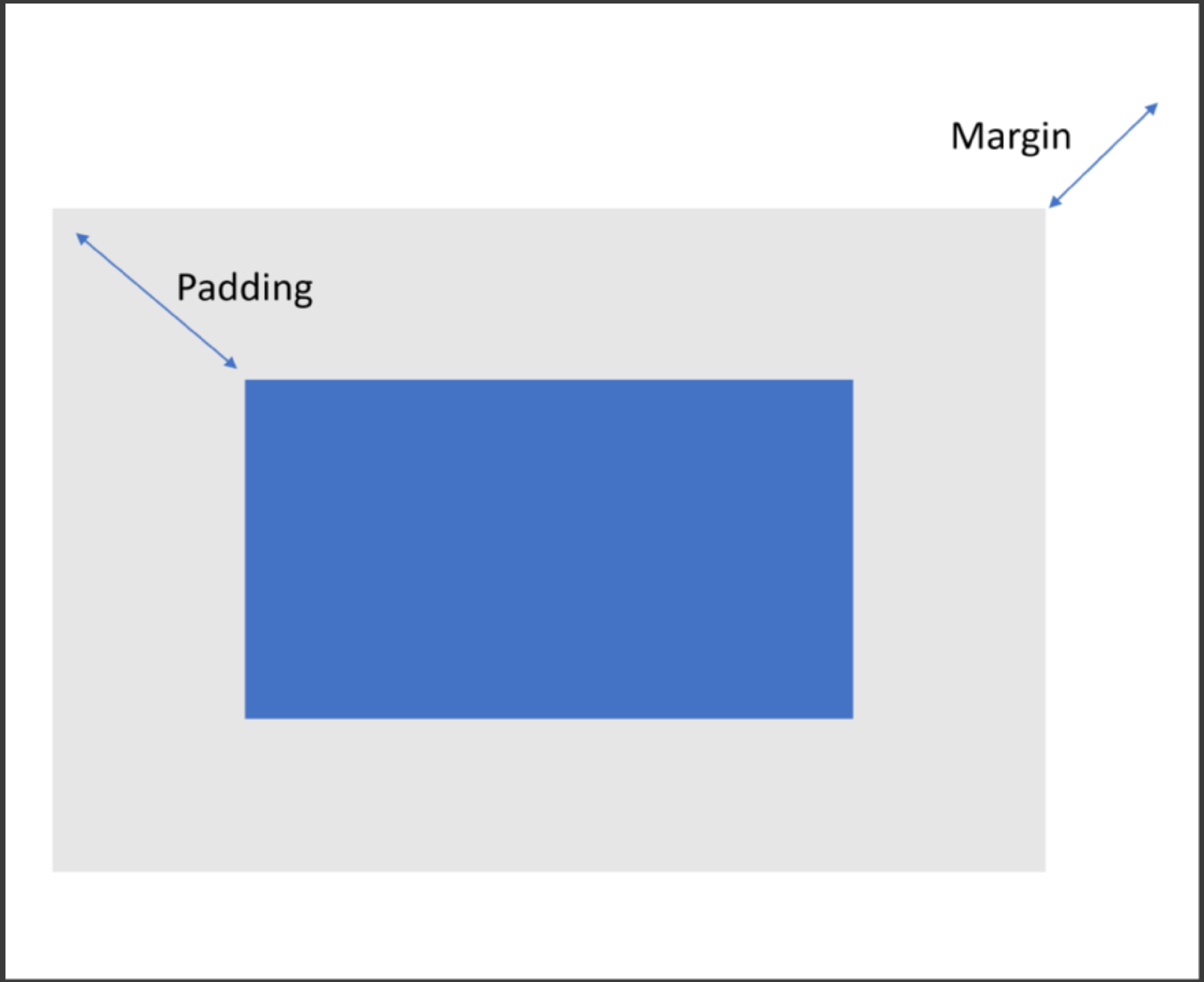
```
TextEditingController controller =
TextEditingController();


@override

void dispose() {

  controller.dispose();

  super.dispose();

}
```

TextField

- Create instance
- Set controller
- Set keyboardType
- Apply a decoration

```
TextField(

  controller: [A TextEditingController],

  keyboardType: [Choose your keyboard]

  decoration: InputDecoration(
     //actions
  )

)
```

Padding & Margin



Padding

Creates distance between a Widget and its container

Set padding

Set child

```
Padding(

  padding: EdgeInsets.all(20.0),

  child: [A Widget],

)
```

Section Summary

- Building the User Interface
- setState()
- TextEditingController
  - onChanged vs controller
- Life cycle events:
  - initState
  - build
  - dispose