SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER APPLICATIONS

23CAT607- CROSS-PLATFORM APP DEVELOPMENT

I YEAR II SEM

UNIT 3 – INTRODUCTION TO LAYOUTS

TOPIC 3 – Advanced Layout Application

Advanced Flutter Layout – Flow

“flow” widget.

In flutter, this is done using the Wrap widget leaving flow to serve a different purpose when it comes to layout in Flutter.

this is not the flow layout in flutter

Simple Flow Example

Let’s first start with the simplest example of the Flow widget. Once we have a basic understanding, we will look at it in greater detail and play a little more with it:

```
appBar: AppBar(title: Text('Flow Layout Flutter')),
body: Center(
child: Container(
color: Colors.teal,
child: Flow(
delegate: SampleFlowDelegate(),
children: <Widget>[
buildItem(0),
buildItem(1),
buildItem(2),
buildItem(3),
buildItem(4),
buildItem(5),
buildItem(6),
],
),
),
),
);
```

The buildItem method return a simple circle with the number passed as an argument


```
return Container(
width:
MediaQuery.of(context).size.width,
height: 100,
decoration: BoxDecoration(
color: Colors.deepOrangeAccent,
border: Border.all(color: Colors.black,
width: 1),
boxShadow: [BoxShadow(blurRadius:
2)],
),
child: FittedBox(
child: Center(
child: Text('$i'),
),
),
);
}
```

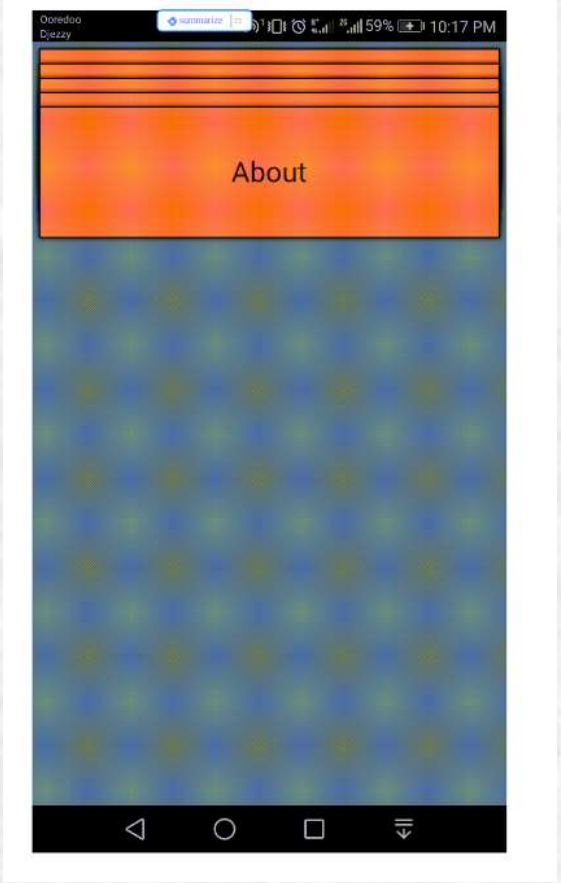
SampleFlowDelegate is where all the fun happens

```
class SampleFlowDelegate extends
FlowDelegate {

@override
void paintChildren(FlowPaintingContext context) {
for (int i = 0; i < context.childCount; ++i) {
context.paintChild(i);
}
}

@override
bool shouldRepaint(SampleFlowDelegate oldDelegate) {
// false just for now
return false;
}
}
```

At this point, you might be thinking that the Flow widget operates similar to Stack. In someways, yes, they do have a few similar aspects such as the order in which the widgets are drawn. This is the reason we are only able to see the last item in our example.

This is the key difference from the Stack widget. Unlike Stack which does all positioning and sizing during the layout phase, with Flow,we can avoid the build phase.

Practical example



Step 1: Stack children gradually

Using context.getChildSize(i). Now, we multiply the height by 0.1 to get that nice effect. You can change the value to 0.5 or 0.9 to see how they layout will look.

```
void paintChildren(FlowPaintingContext
context) {
                                        double dy = 0.0;
                                        for (int i = 0; i < context.childCount; ++i) {
                                        dy = context.getChildSize(i).height * i;
                                        context.paintChild(
                                        i,
                                        transform: Matrix4.translationValues(
                                        0,
                                        dy * 0.1,
                                        0,
                                        ),
                                        );
                                        }
                                        }
```

Step 2: Animate the childr

The power of the Flow layout shines because of the FlowDelegate accepts an optional repaint arguments of type Listenable. This listenable will reprint the Flow whenever the listeners are notified.

```
SampleFlowDelegate({this.openAnimation
}) : super(repaint: openAnimation);
```

```
final Animation<double> openAnimation;

@override
bool shouldRepaint(SampleFlowDelegate oldDelegate) {
return openAnimation != oldDelegate.openAnimation;
}
```

Step 3: Create the animation

```
AnimationController openAnimation;
```

```
@override
void initState() {
super.initState();
openAnimation = AnimationController(
lowerBound: 1,
upperBound: 10,
duration: Duration(seconds: 2),
vsync: this,
);
}
```

Step 4: Animate on Tap / DoubleTap

```
GestureDetector(
onTap: () {
openAnimation.reverse();
},
onDoubleTap: () {
openAnimation.forward();
},
```

And the final step is to pass the animation to the delegate delegate:

```
SampleFlowDelegate(openAnimation: openAnimation)
```

Flow doesn't give you anything that you can't get with "normal" widgets however, when it comes to animations and re-positioning widgets using translations, it does provide optimal performance.


thank
you