Day 6:

Topic:

Widget tree, element tree, render tree, Think declaratively, Designing layouts,

Widget

Widget tree?

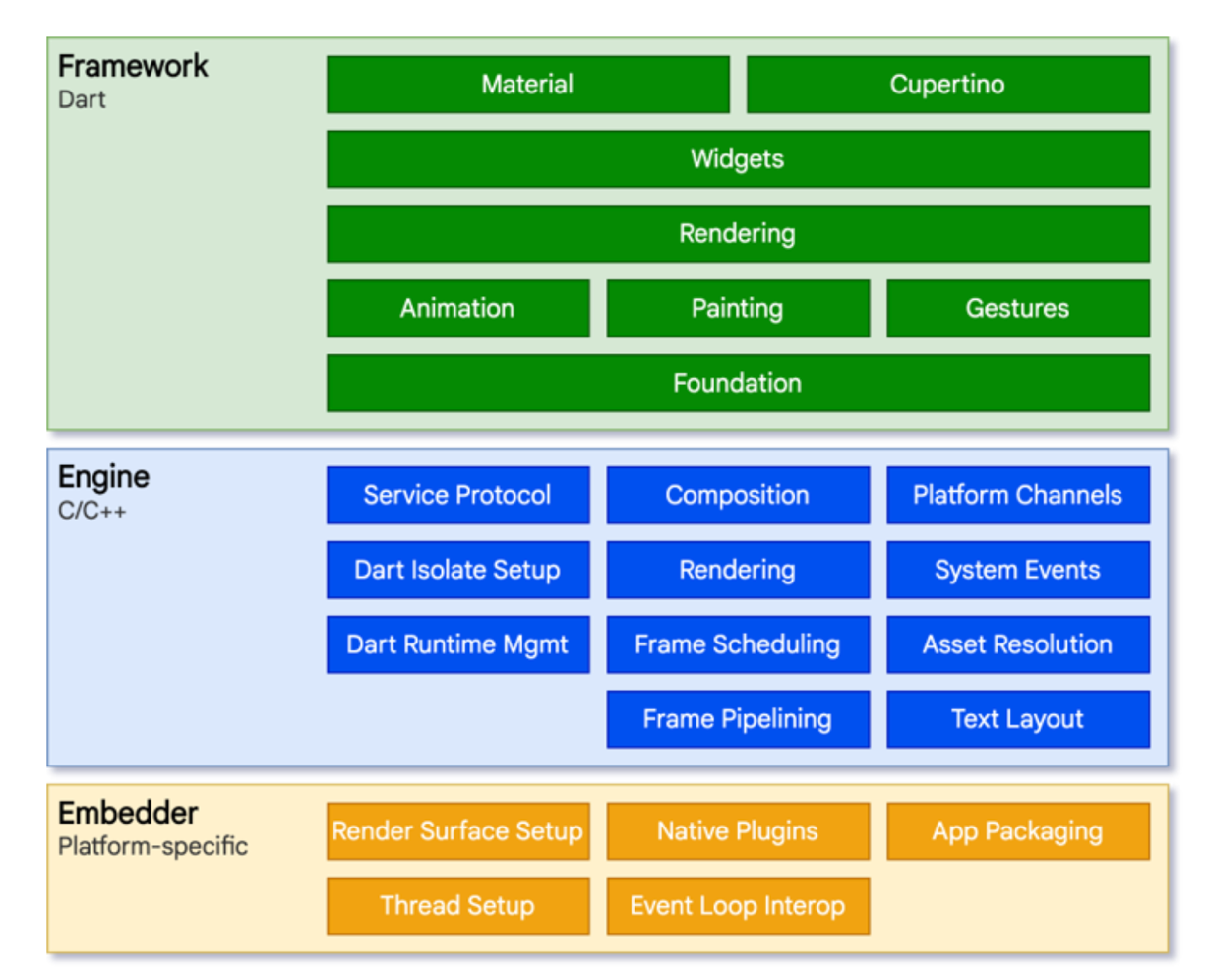
Element

Element tree?

Render Object

What is render object?

Overview:



Widget – hold configuration properties you added into UI, offer a public api

Element – hold a location or spot in hierarchy, actual piece of UI, manage parent and child relationship or reference

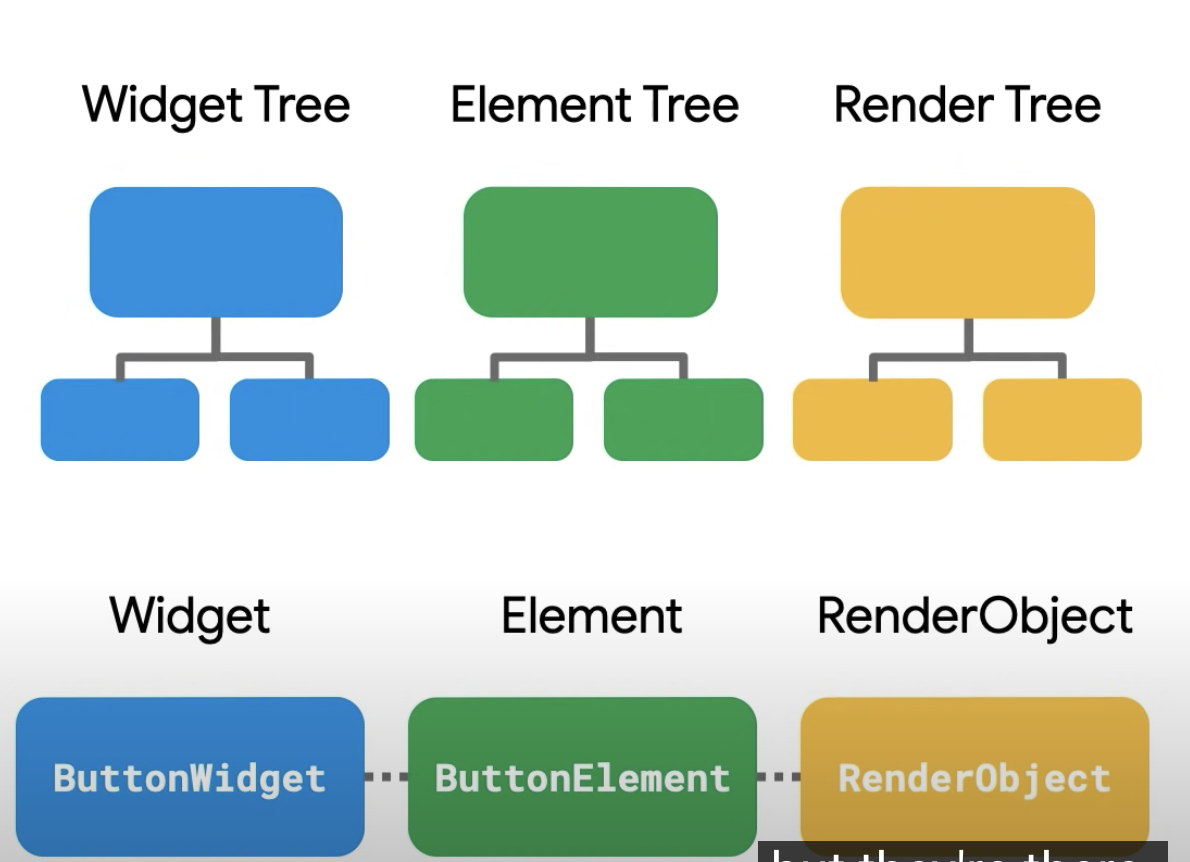
RenderObject – know about the UI size and layout painting, claim input events

Flutter has 3 tree.

Widget tree

Element tree

RenderTree



**What is element?**

An element represent the use of widget to configure specific location in the tree

In Flutter widget is the configuration of the UI, the main class which represents the widget or UI displayed on the screen is Element.

i.e Widget is actually the configuration data of Element and the Widget tree is actually a configuration tree,

for the Element tree which is automatically generated whenever we create a widget tree.

Life cycle of element:

Element(Widget)

Widget.createElement

Mount function

attachRenderObject

The framework creates an element by calling [Widget.createElement] on the  
 widget that will be used as the element's initial configuration.

The framework calls [mount] to add the newly created element to the tree  
at a given slot in a given parent.

The [mount] method is responsible for inflating any child widgets and calling [attachRenderObject] as necessary to attach any associated render objects to the render tree.

Stateless widget & element tree:

* Each stateless widget has corresponding stateless element
* Widget make request to create element
* Each element contains the reference of widget
* Element triggers the build method of widget
* Each child widget create its own element to be mounted to element tree.

Statefull widget & element tree:

* Each statefull widget has corresponding statefull element.
* Widget make request to create element
* Each element contains the reference of widget
* Element request widget to create state object
* It will call “**createState**” method
* Now element has reference to state object
* Element triggers the build method of widget
* Each child widget create its own element to be mounted to element tree.

**What is render object:**

RenderObjectElement:

An element that use a render object widget as its configuration

* Leaf render objects, with no children: The [LeafRenderObjectElement](https://api.flutter.dev/flutter/widgets/LeafRenderObjectElement-class.html) class handles this case.

Ex. ErrorWidget : This widget is used when build function is failed. <https://api.flutter.dev/flutter/widgets/ErrorWidget-class.html>

* A single child: The [SingleChildRenderObjectElement](https://api.flutter.dev/flutter/widgets/SingleChildRenderObjectElement-class.html) class handles this case.

Ex. Padding, SizeBox, Text

<https://api.flutter.dev/flutter/widgets/SingleChildRenderObjectElement/SingleChildRenderObjectElement.html>

* A linked list of children: The [MultiChildRenderObjectElement](https://api.flutter.dev/flutter/widgets/MultiChildRenderObjectElement-class.html) class handles this case.

Ex. Row, column, Stack

<https://api.flutter.dev/flutter/widgets/MultiChildRenderObjectElement/MultiChildRenderObjectElement.html>

Open flutter inspector & click render tree

Think Declarative:

* Older way is the designing the UI with
* Like in Android : XML way
* Like in iOS : UIKit
* Now Flutter is declarative

Video recording Link:

<https://deloitte.zoom.us/rec/share/VWYhIxOWJCE7LBLXec30RK2oEenFRez90yZj6U8tRnOv8_fSqPj8HylYkikQMwZQ.tFCr581NgBVn_AzT>

(Access Password: SzG.&0&^)

Reference Links:

1. <https://api.flutter.dev/flutter/widgets/Element-class.html>
2. Rendering objects: <https://www.youtube.com/watch?v=zmbmrw07qBc>
3. Flutter focus: <https://www.youtube.com/watch?v=AqCMFXEmf3w&list=PLjxrf2q8roU2HdJQDjJzOeO6J3FoFLWr2&index=5>
4. <https://www.youtube.com/watch?v=996ZgFRENMs&list=RDCMUCwXdFgeE9KYzlDdR7TG9cMw&index=3>
5. <https://docs.flutter.dev/development/data-and-backend/state-mgmt/declarative>