



Prifysgol
Abertawe
Swansea
University

CS-230 Software Engineering

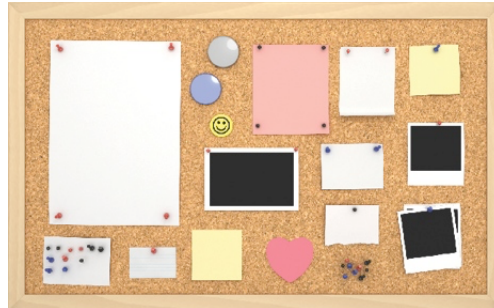
L14: JavaFX – Canvas

Dr. Liam O'Reilly

Semester 1 – 2020

Recap – Layout Management

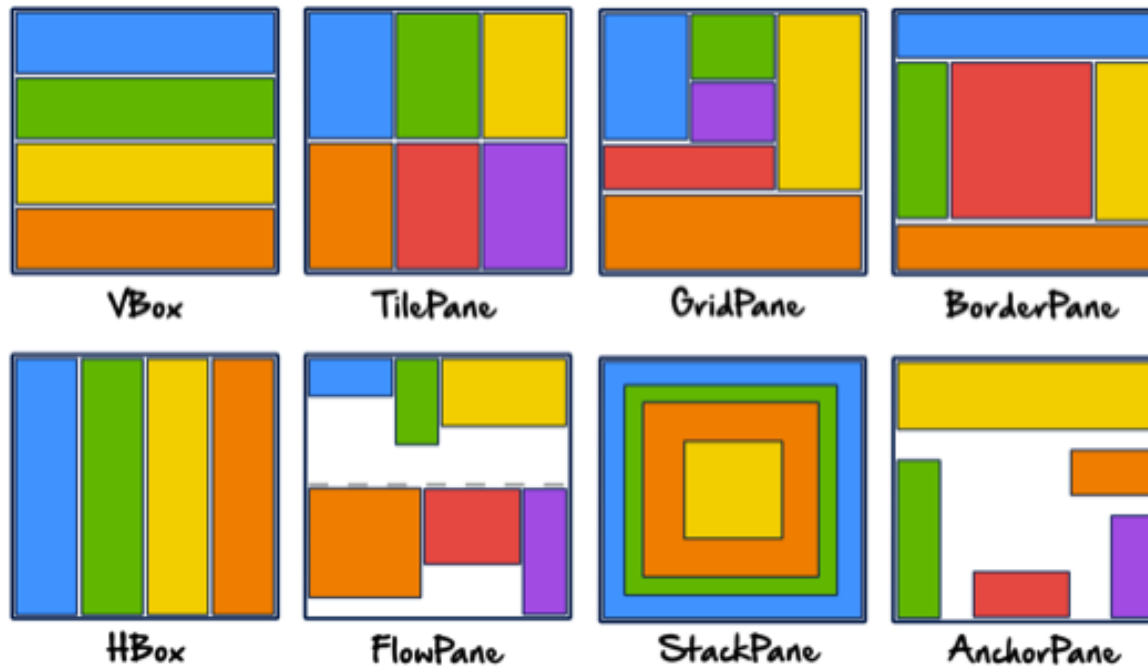
- Arranging components on the screen.
- User-interface components are arranged by placing them inside container nodes.
- There are a variety of “Pane” nodes that each layout their children in different ways.



- Useful Websites / Tutorials:
 - JavaFX:
<https://docs.oracle.com/javase/8/javase-clienttechnologies.htm>
 - JavaFX layouts:
<https://docs.oracle.com/javase/8/javafx/layout-tutorial/index.html>

Recap – Layout Nodes

- Main Idea:
 - There are various panes that layout the children in different ways.
 - You must **nest** the panes in order to produce desirable layouts.



Recap – FXML

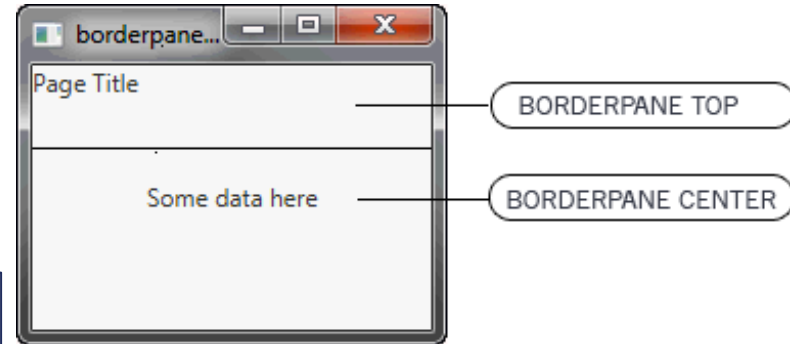
Separate graphical layout from logic of program

JavaFX without FXML

```
BorderPane border = new BorderPane();

Label topPaneText = new Label("Page Title");
border.setTop(topPaneText);

Label centerPaneText = new Label ("Some data here");
border.setCenter(centerPaneText);
```



FXML

```
*** Imports go here ***
<BorderPane>
  <top>
    <Label text="Page Title"/>
  </top>
  <center>
    <Label text="Some data here"/>
  </center>
</BorderPane>
```

In FXML we specify the scene graph that JavaFX will use in a separate xml file.

Today

- We will look at how to draw using a canvas.
- We will create a sample application.

JavaFX Node Types

- There are many node types in JavaFX.
- You place them in a **Scene Graph** to create the GUI that you want.
- Example nodes:
 - Button
 - BorderPane
 - **Canvas**

Canvas

- The **Canvas** class allows you to create images using simple drawing features like painting circles, lines and rectangles.
- It can also be used (with other classes) to create very complex images using effects like blending.
- The class Canvas is found in the package `javafx.scene.canvas`

- Import it by:

```
Import javafx.scene.canvas.Canvas;
```

- You should not need to worry about imports, your IDE should be doing it for you.
- You construct it by giving it by providing a width and height.
 - It can be a bit tricky to resize it after it is created.
- You need to add this to your scene somewhere.

Drawing on a Canvas

- A **Canvas** is just a visual node.
- The image data is stored in a buffer.
- This buffer is encapsulated in a **GraphicContext** object.
- From a Canvas object you can access its GraphicContext.
- You call **methods** on the **GraphicContext** to do things like:
 - Set the current stroke colour.
 - The colour used to draw lines and outlines from now on.
 - Set the current fill colour.
 - The colour used to fill shapes from now on.
 - Draw a shape.

Canvas Example

I will now code this from scratch.

You can learn:

- How to use Canvas and JavaFX.
- How to build this up in small steps.

Code is available on Blackboard.

