CPSC 475/575 - Mobile Computing

Administrative

- First Proj Assigned
- See Readings (are testable)
- See Examples (use git to open)

Topics

- Communications between Android Studio and Device and Emulator
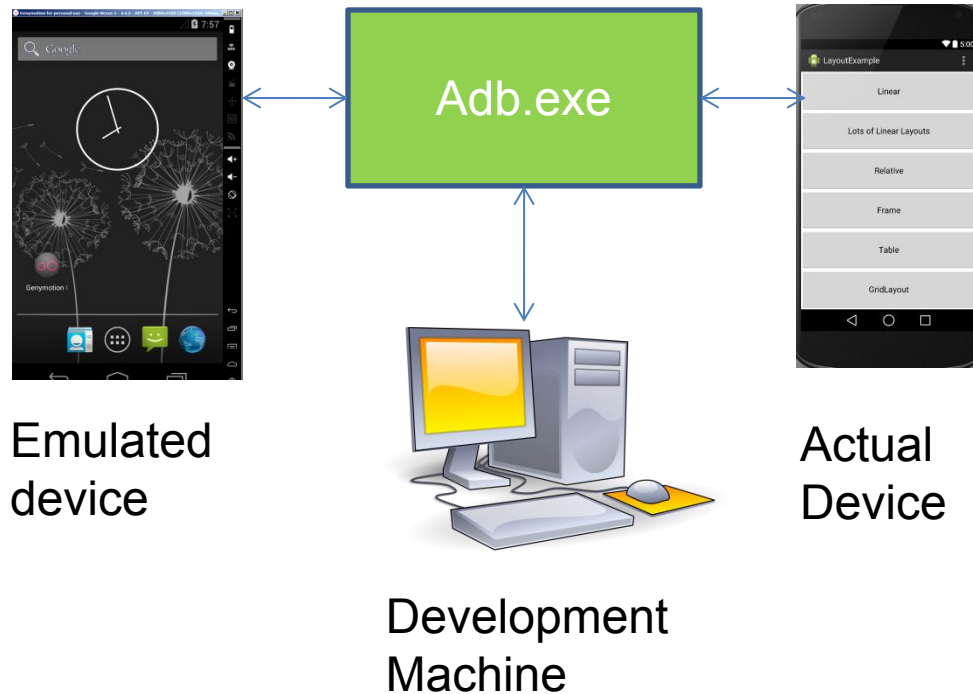- Compilation
- Sizing
- Layouts

Topics

- Communications between Android Studio and Device and Emulator
- Compilation
- A word about Interfaces
- Animations
- Layouts

Communications
Adb.exe

- How does Android Studio communicate with emulator or device?

Emulated device

Development Machine

Actual Device

Compilation – generate apk

What is an apk file?
What your application compiles to

Where is it?
Depends, can be a release signed apk or a debug build
Search for *.apk in project

More Info:
https://developer.android.com/sdk/installing/studio-build.html

Install apk

- Dalvik verses ART
  - ART is the future (from 5.0 on)
  - Dalvik is dead
- See ART vs Dalvik for more info

https://software.intel.com/en-us/blogs/2014/06/18/art-vs-dalvik-introducing-the-new-android-x86-runtime

Interfaces and Animations

Interfaces
Used to define classes of behavior

Define a type of behavior
- Abstract (methods empty, derived classes fill in)
- *You cannot instantiate an interface.*
- *An interface does not contain any constructors.*
- *All of the methods in an interface are abstract.*
- *An interface cannot contain instance fields. The only fields that can appear in an interface must be declared both static and final.*
- *An interface is not extended by a class; it is implemented by a class.*

```
interface animal {
    public void eat();
    public void travel();
}


public class Mammal
implements animal{
}
```

You are forced to override methods or make class abstract

Animations
(easy way to make UI move around)

Steps:
1. Create XML animation (in res/anim)
2. Load it
3. Set listeners if wanted (you want it usually). Easiest is to implement Animation.AnimationListener interface in activity.
4. Start it (button click, onResume()…)

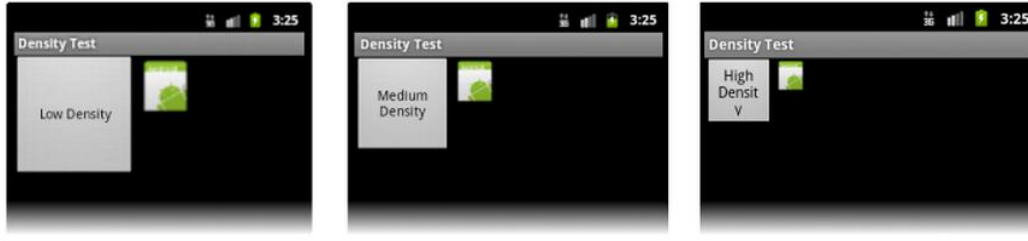Demo- create empty project and open Animation_StepByStep.txt to follow along

See animation tutorial in Readings
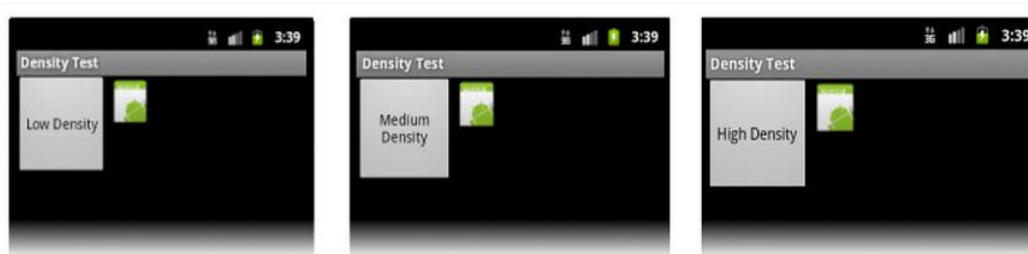See animation demo project

Sizing

Sizing Note
Use dp not px

Using px – buttons get smaller as density goes up

Using dp - scales as density changes

Layouts

Layouts

- Location


```
▼  📁 res
   ▶  📁 drawable
   ▼  📁 layout
         📄 activity_main.xml
```
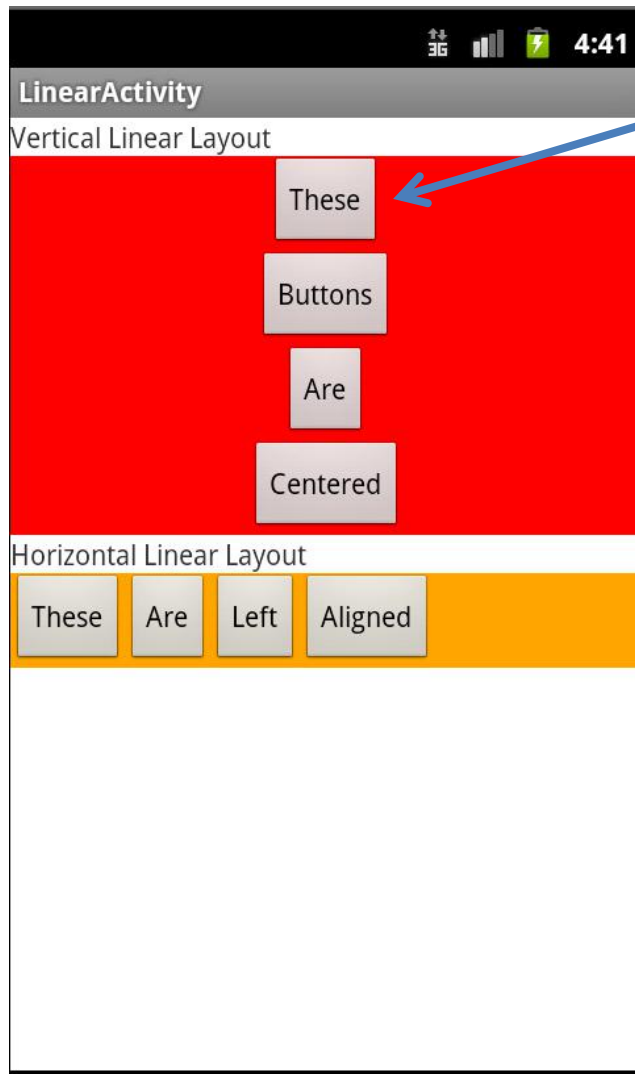
- 2 today
  - Linear (3_layoutExample project)
  - Framelayout   "

- See the online doc on Constraint Layout

LinearLayout
(see 3_layoutExample project)


```xml
<Button android:text="These"
    android:layout_gravity="center"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
/>
```

```xml
<LinearLayout
    android:layout_height="wrap_content"
    android:layout_width="match_parent"
    android:orientation="vertical"
    android:background="#ffff180a">
```

```xml
<LinearLayout
    android:orientation ="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="left"
    android:background="#ffff980c">
```

LinearLayout

LinearLayout
gravity layout_gravity

gravity – align text inside widget

```
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="left"
    android:text="Left" />
```

layout-gravity – align widget in parent

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="right"
    android:text="right" />
```

LinearLayout
android:layout_weight (There is no android:weight!)

Proportions widget spacing


```
<Button android:text="Weight1"
        android:layout_weight="1"
        android:gravity="center"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>

<Button android:text="Weight2"
        android:layout_weight="2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>

<Button android:text="NO Weight"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>

<Button android:text="Weight1(all the rest)"
        android:layout_weight="1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
```

Note android:layout_width="wrap_content"
Necessary for proper function

RelativeLayout


```xml
<Button
    android:id="@+id/leftButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:text="Left" />

<Button
    android:id="@+id/MidButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Mid"
    android:layout_centerVertical="true"
    android:layout_centerHorizontal="true" />

<Button
    android:id="@+id/RightButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_toRightOf="@id/MidButton"
    android:text="Bot" />
```

FrameLayout

- Placed on screen
- On Top of each other
- In order declared

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/FL"
    android:layout_width="match_parent" android:layout_height="match_parent"
    android:background="@drawable/ic_launcher">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="96dp"
        android:text="New Text"
        android:id="@+id/textView"
        android:layout_gravity="center_horizontal|bottom"
        android:background="#44ee0bff" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="click me"
        android:onClick="doButton"
        android:id="@+id/button"
        android:layout_gravity="right|bottom" />
</FrameLayout>
```

Use dp not px

New Text

click me

FrameLayout


```java
public class FrameLayoutActivity extends Activity {

    private boolean isGreen = false;
    private int Id;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_framelayout);
        Resources resources = this.getApplicationContext().getResources();
        Id = resources.getIdentifier("ic_launcher", "drawable", getPackageName());
    }


    public void doButton(View view) {
        FrameLayout ml = (FrameLayout)findViewById(R.id.FL);


        if (isGreen)
            ml.setBackgroundResource(Id);
        else
            ml.setBackgroundColor(0xFF00FF00);


        isGreen = !isGreen;
    }
}
```
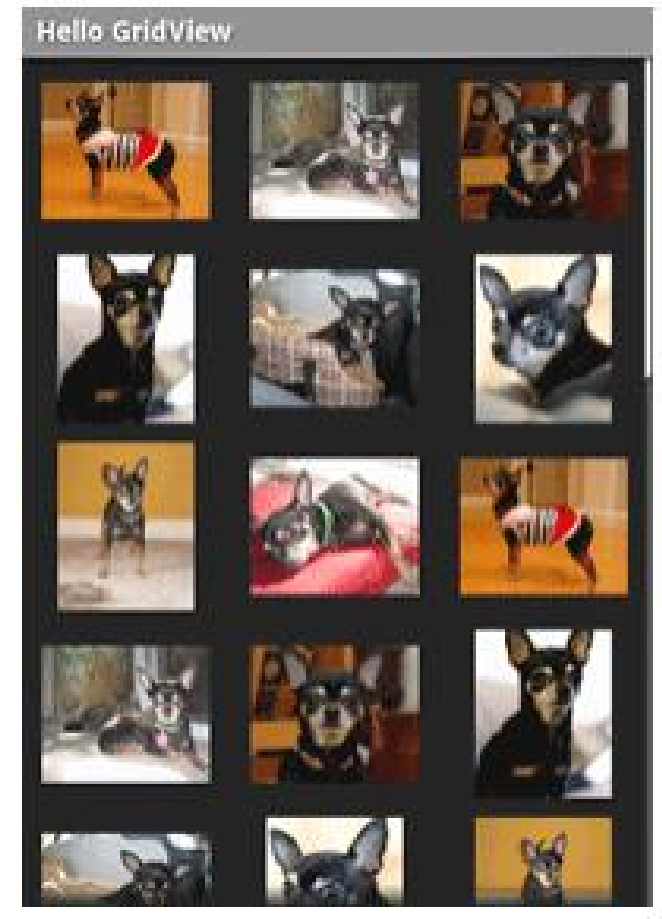
TableLayout

- rows and columns
- rows normally TableRows
- TableRows contain other elements such as buttons, text, etc.

TableActivity

Column1	Column2
Spans 2 columns	
Column1	Column2

Other Layouts - GridView

- Two Dimensional Scrollable Grid
- Items inserted into layout via a ListAdapter

Other Layouts - TabLayout

- Uses a TabHost and TabWidget
- Swap between views in same activity or switch between different activities

Other Layouts - ListView

- Creates a list of scrollable items
- Items added via a ListAdapter as in GridView
- May see this later

Other Views - Layouts

- ## Gallery
  - horizontal scrolling display of images from a list
- ## SurfaceView
  - provide access to a "drawing" surface. Intended to draw pixels, not display other views / widgets