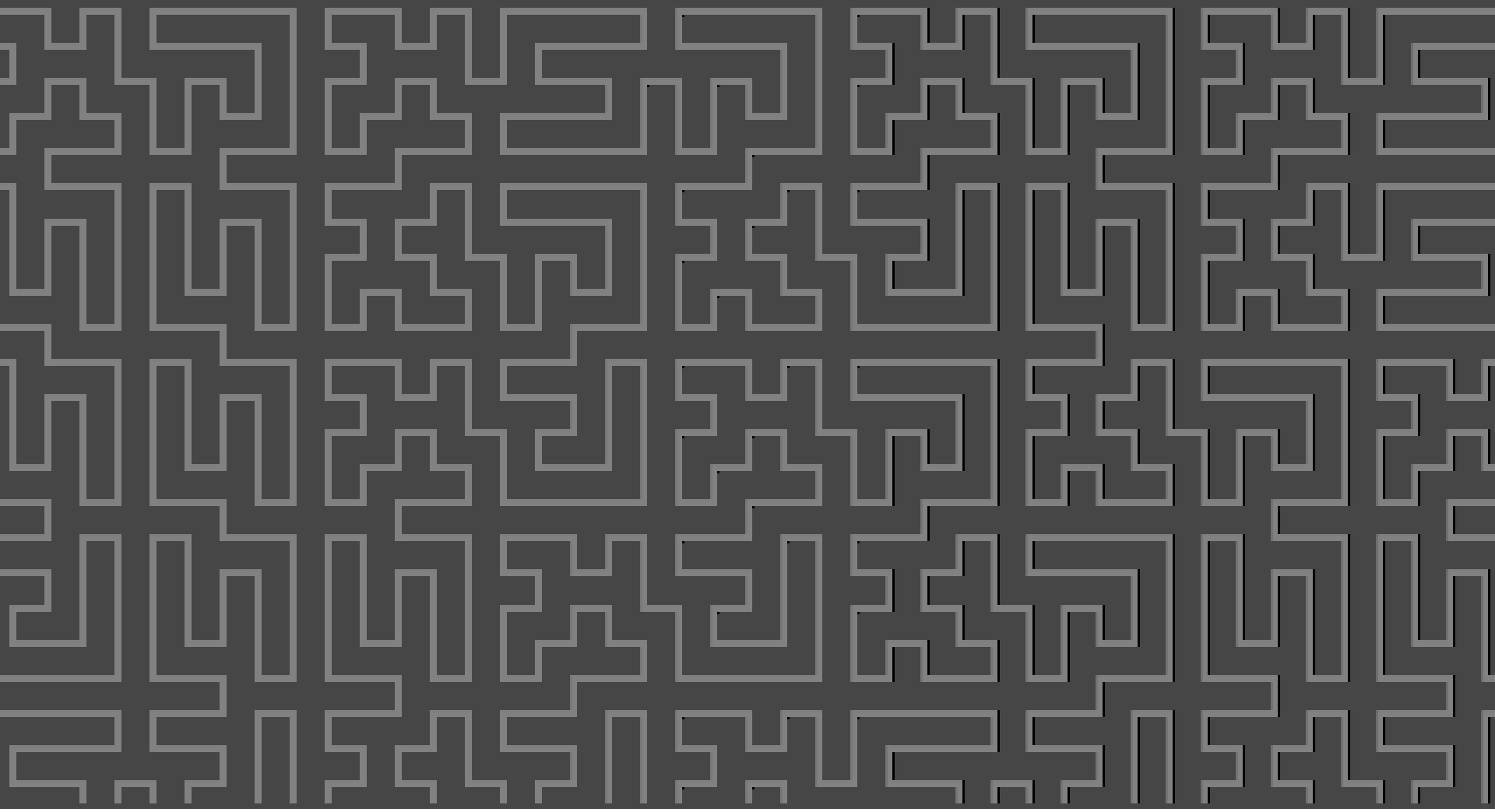
MOBILE APP DEVELOPMENT



Lecture 2 – Events

Content

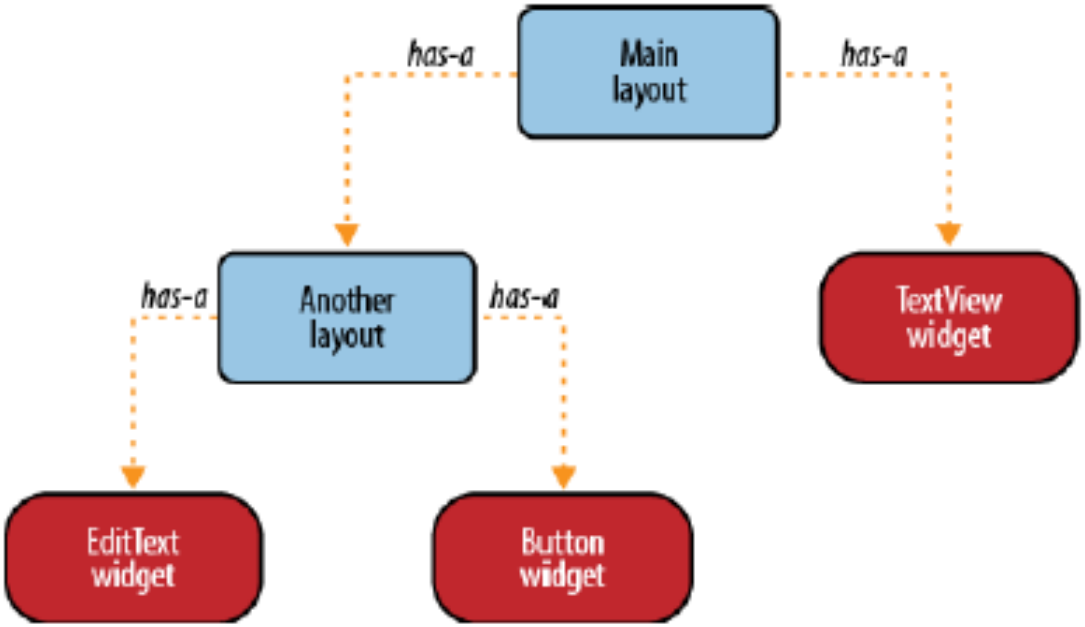
- User Interfaces
- Layouts
- Events
- Simple Media player

User Interfaces

- Two ways to develop an interface
  - Declaratively using XML
  - Dynamically using Java
- XML gets translated into Java code behind the scenes
- https://developer.android.com/training/constraint-layout/index.html#constraints-overview

Layouts

- Are special View objects which can contain children
- Can include child Layout objects to define a complex user interface



Layouts

- Different Layout objects have different layout / drawing strategies
- Most Layouts have horizontal or vertical stacking options – and layout re-distribute components for different screen sizes
- LinearLayout: widgets stacked with a simple left to right or top to bottom layout strategy
- AbsoluteLayout: pixel perfect WYSIWYG placement of widgets – but useless with different screen sizes / orientations

Layouts

- Other layouts to check out:
- GridLayout – divides screen into a grid
- RelativeLayout – widgets laid out relative to each other and other layouts
- ConstraintLayout – default layout and designed to be used with the editor

The default ConstraintLayout appears to be good enough in most cases
We'll be using LinearLayout Today

Layout Parameters

- Define size of a widget using layout_width and layout_height
  - wrap_content — size to match content
  - match_parent — size to match the parent
- These options great for re-sizing to match different screen sizes
- Exact Dimensions – define in screen indepenedent dp units

Android Dimensions

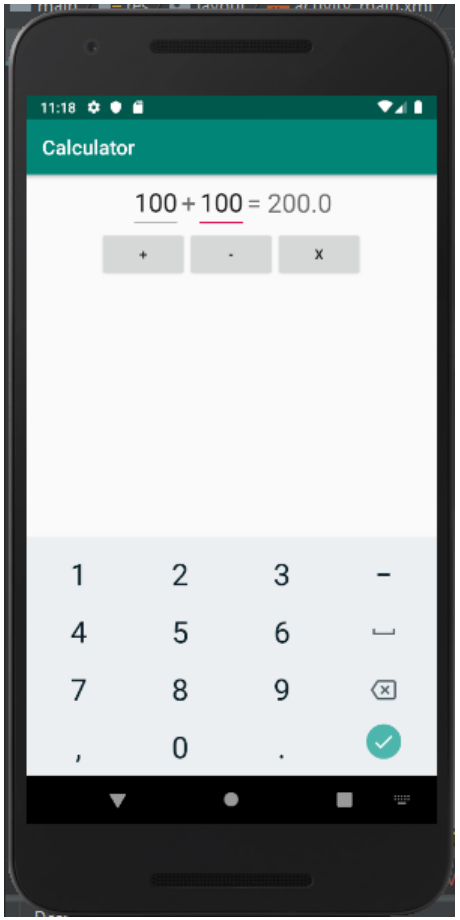
- dp — Density-independent Pixels
  - Consistent across different devices
- sp — Scale-independent Pixels
  - Same as dp, but also scaled by a user's font size prefs
- pt — Points 1/72 inch assuming a 72 dpi screen
- px — Pixels on screen not recommended
- mm — millimetres
- in — inches

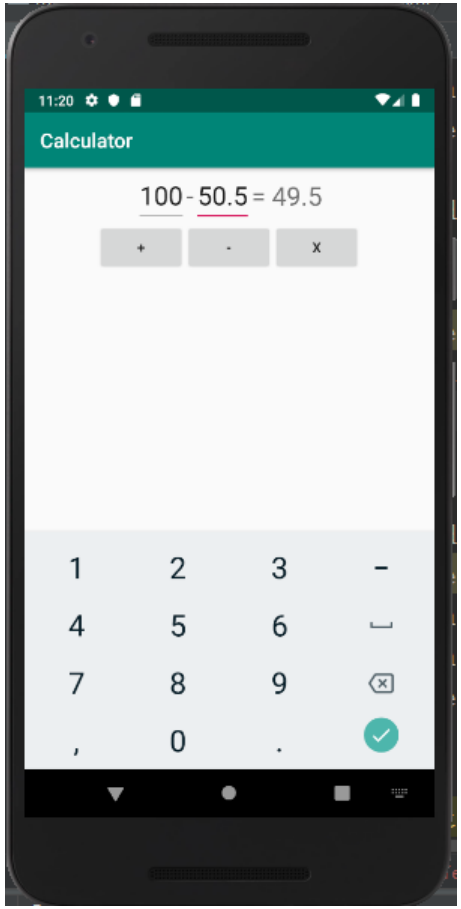
- https://developer.android.com/guide/topics/resources/more-resources.html#Dimension

Event Driven Programming

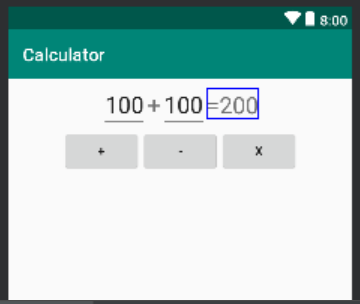
- Click
- Focus
  - Get
  - Leave
- Mouse Over
- Text Selected
- Item Selected

Lets start from calculator



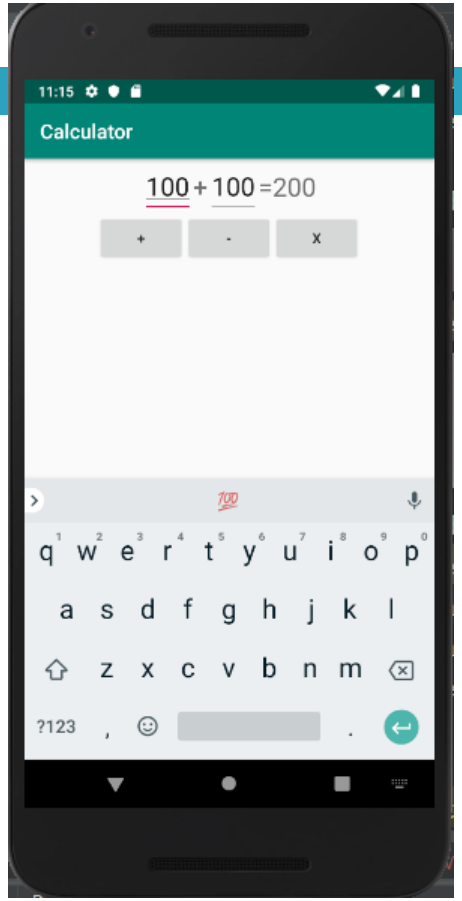


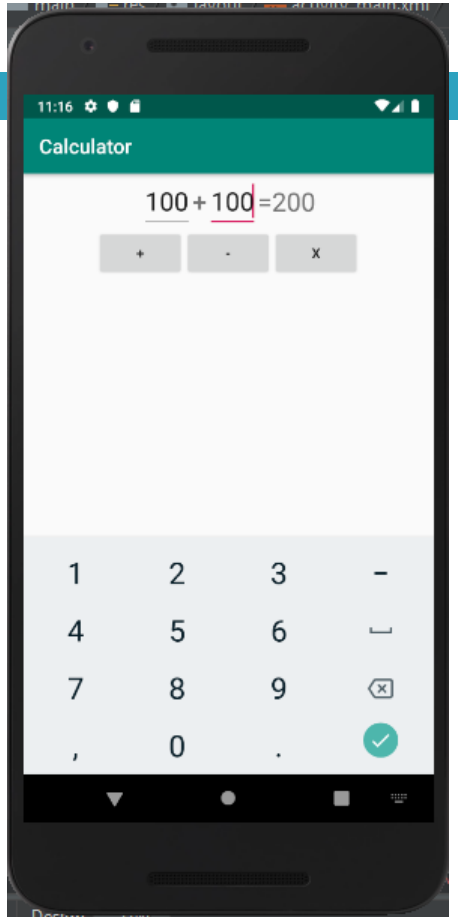
```xml
<EditText
    android:id="@+id/txt2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="100"
    android:inputType="numberSigned|numberDecimal"
    android:textSize="26sp"
     />
<TextView
    android:id="@+id/txtR"
    android:text="=200"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="26sp"
   />

</LinearLayout>
```

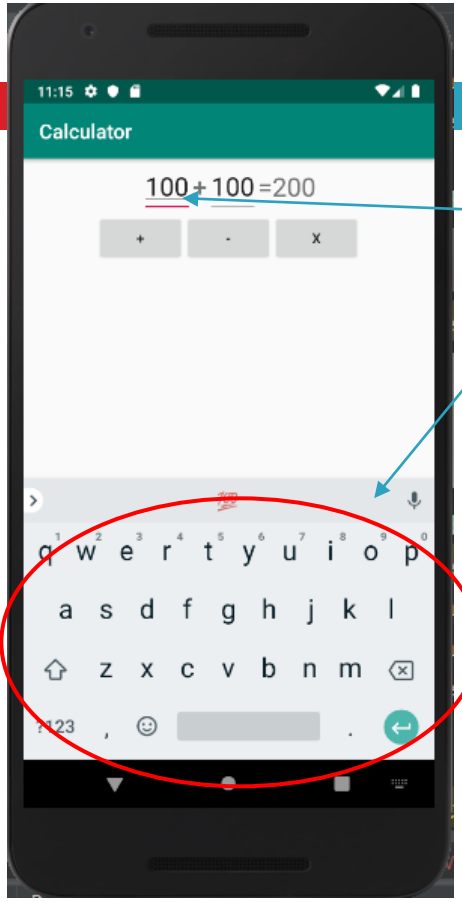


MainActivity.java

```java
    @Override
    protected void onCreate(Bundle savedInstanceState)
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button btnA =  findViewById(R.id.btnA);
        Button btnS =  findViewById(R.id.btnS);
        Button btnM =  findViewById(R.id.btnM);
        btnA.setOnClickListener(this);
        btnM.setOnClickListener(this);
        btnS.setOnClickListener(this);

    }

    @Override
    public void onClick(View v) {
        EditText txt1 = findViewById(R.id.txt1);
        EditText txt2 = findViewById(R.id.txt2);
        TextView txtO = findViewById(R.id.txtOp);
        TextView txtR = findViewById(R.id.txtR);

```

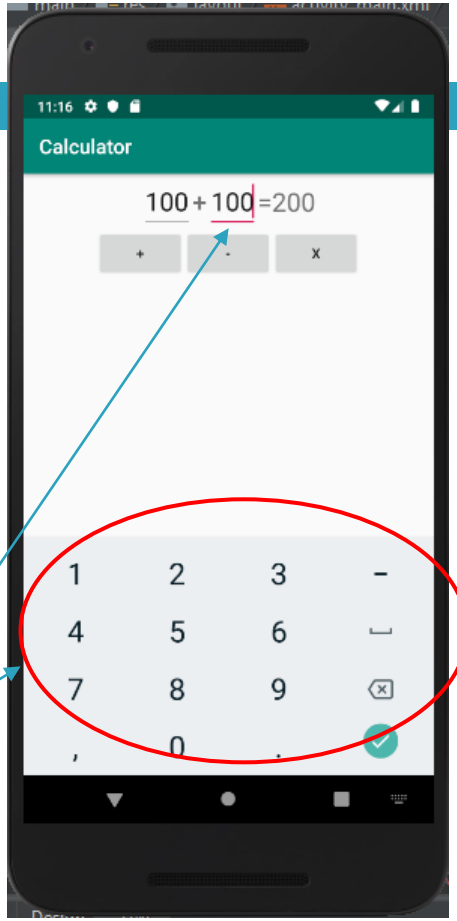

11:15
Calculator
100 + 100 =200
+
-
x
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
?123


11:16
Calculator
100 + 100 =200
+
-
x
1 2 3 -
4 5 6
7 8 9
, 0 .




```
<EditText
    android:id="@+id/txt1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="100"/>
```

```
<EditText
    android:id="@+id/txt2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="100"
    android:inputType="numberSigned|numberDecimal"
 />
```



Bird Song App

```
    <ImageView
        android:id="@+id/img"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/bird"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Bird Song"
        android:textColor="@color/colorAccent"
        android:textSize="26sp"
        app:layout_constraintBottom_toTopOf="@+id/img"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</
```

Image file

bird.jpg

Sound/mp3 file

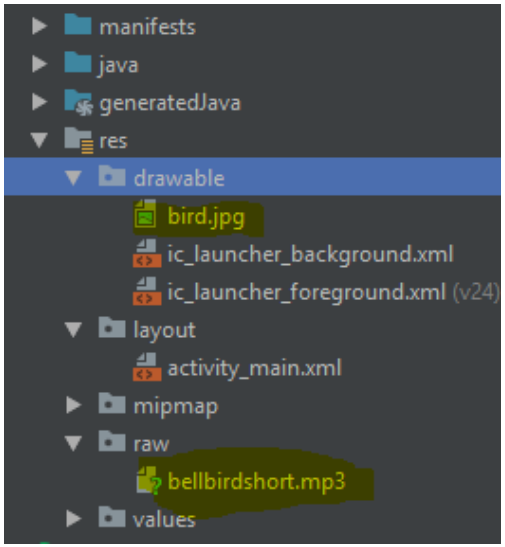
Bellbirdshort.mp3

MediaPlayer

- Android provides a MediaPlayer class to access built-in mediaplayer services like playing audio and video.

- To use the MediaPlayer , call its static Method **create()**.
- This method returns an instance of MediaPlayer class.

- Create a raw folder in the res folder, my song was called song.mp3




```
public void onClick(View v) {
    MediaPlayer mp = MediaPlayer.create(this, R.raw.bellbirdshort);
    mp.start();
}
```

Common methods

- mediaPlayer.start();
- mediaPlayer.pause();
- mediaPlayer.reset();

Method & description

isPlaying()

This method just returns true/false indicating the song is playing or not

seekTo(position)

This method takes an integer, and moves song to that particular second

getCurrentDuration()

This method returns the current position of song in milliseconds

getDuration()

This method returns the total time duration of song in milliseconds

reset()

This method resets the media player

release()

This method releases any resource attached with MediaPlayer object

setVolume(float leftVolume, float rightVolume)

This method sets the up down volume for this player

setDataSource(FileDescriptor fd)

This method sets the data source of audio/video file

selectTrack(int index)

This method takes an integer, and select the track from the list on that particular index

getTrackInfo()

This method returns an array of track information

References

- https://developer.android.com/training/constraint-layout/index.html#constraints-overview
- https://developer.android.com/guide/topics/ui/controls/button
- https://developer.android.com/guide/topics/ui/notifiers/toasts