Recording in Progress

This class is being recorded

Please turn off your video and/or video if you do not wish to be recorded

CMSC436: Programming Handheld Systems

User Interface Classes

Today’s Topics

Views & View Events

View Groups, AdapterViews & Layouts

Menus & ActionBar

Dialogs

Android User Interfaces

Activities usually display a user interface

Android provides many classes for constructing user interfaces

View

Key building block for UI components

Occupies a rectangular space on screen

Responsible for drawing itself and for handling events

Common View Operations

Set visibility: Show or hide View

Set checked state: Checked or not checked

Set listeners: Code that will be executed when specific events occur

Set properties: Opacity, background, rotation

Manage input focus: Allow View to take focus, request focus, etc.

Some Predefined Views

Button

ToggleButton

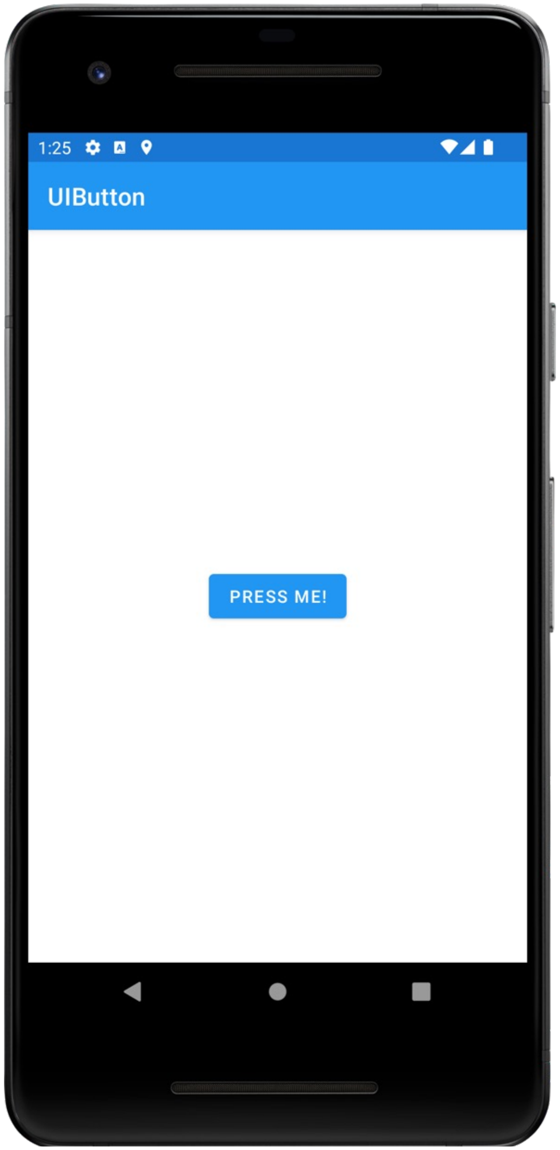
Checkbox

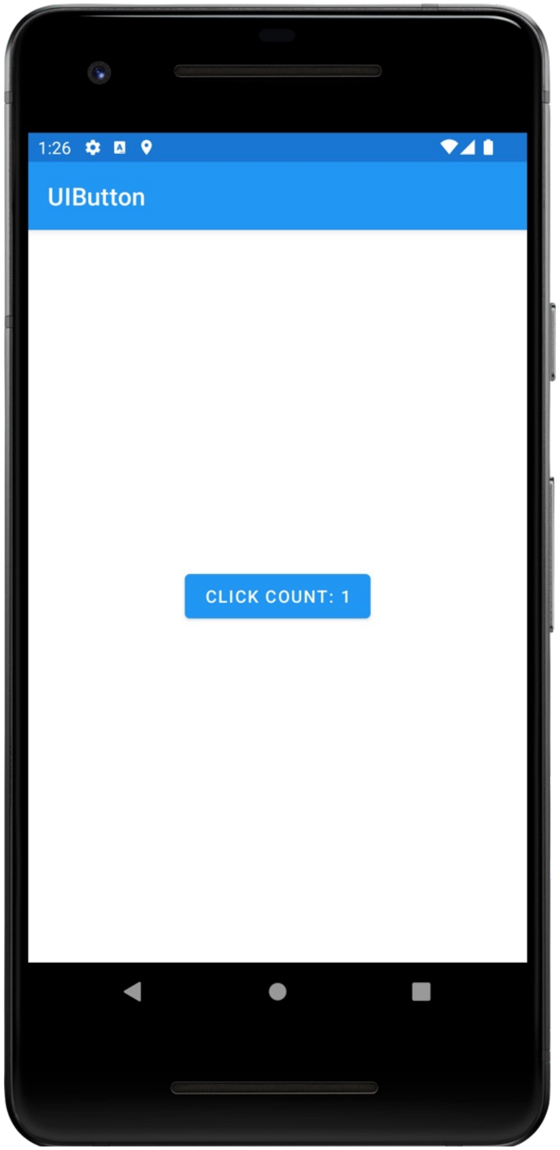
AutoCompleteTextView

Button

View that can be clicked on to perform an action

UIButton

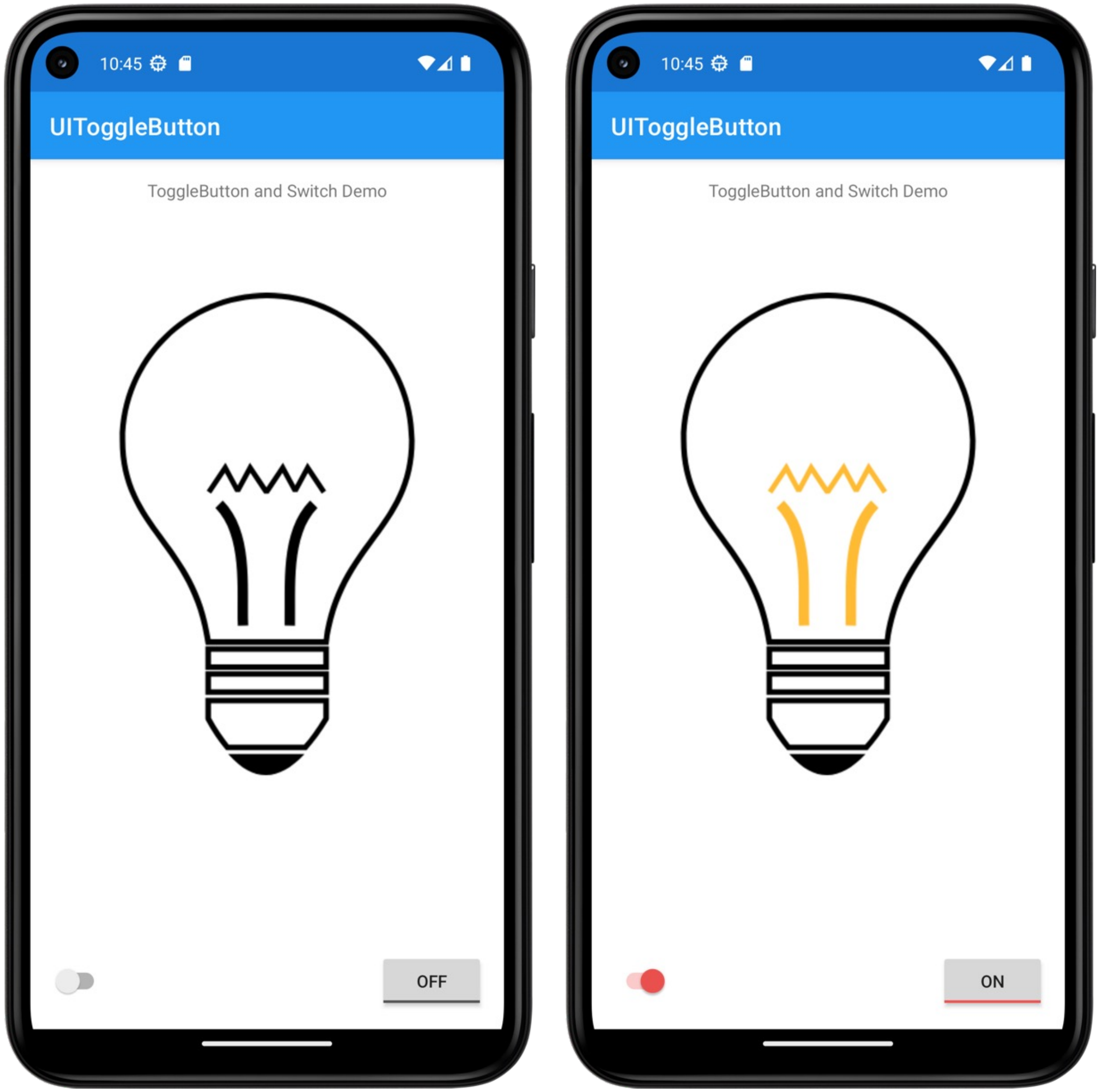




ToggleButton

- A 2-state Button
  - Checked/not checked state
  - Light indicator showing state

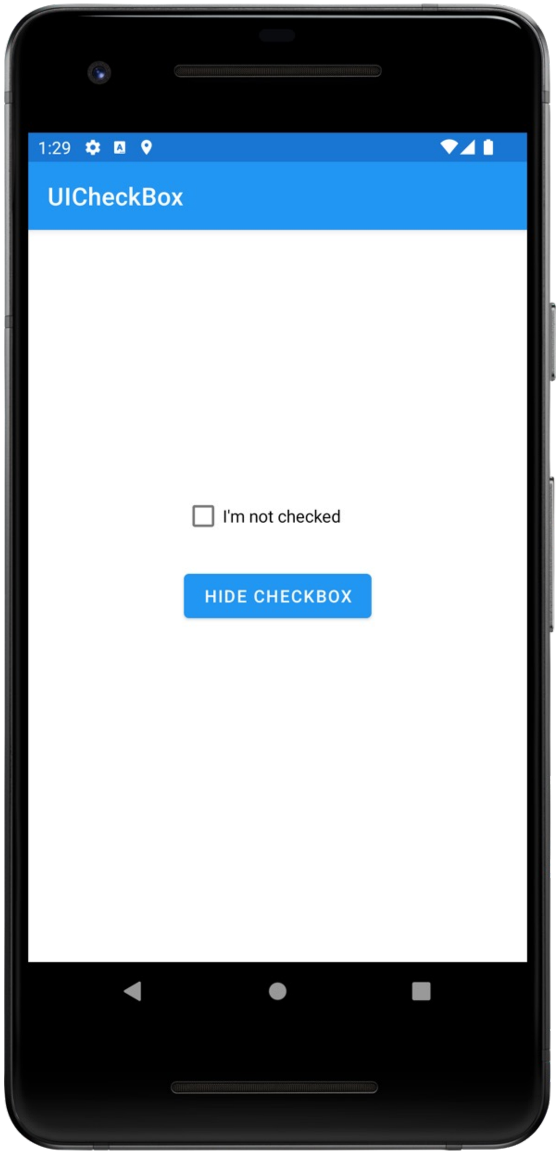
UIToggleButton

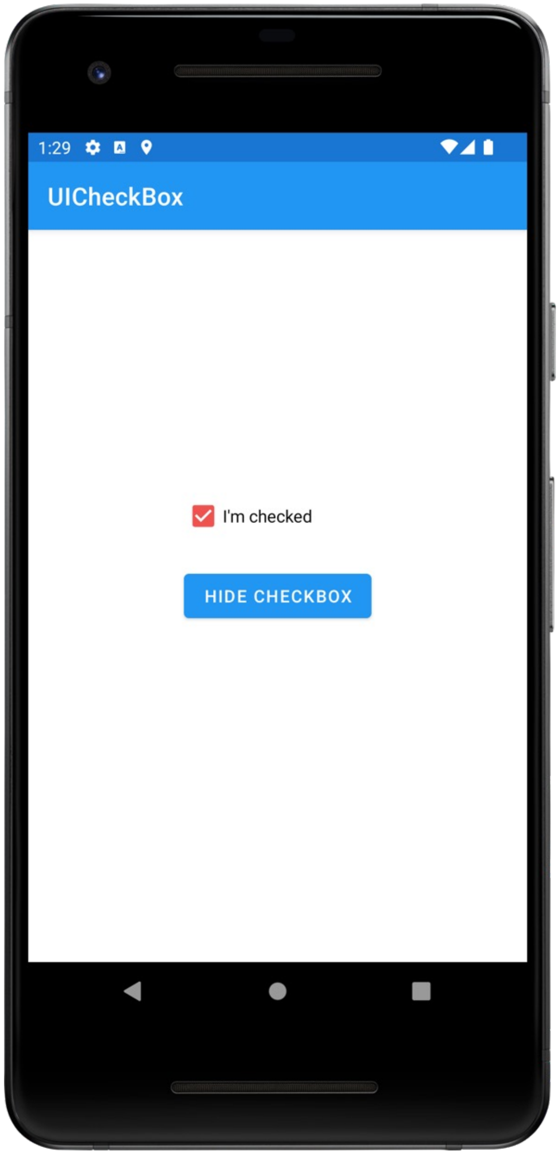


Checkbox

- Another kind of 2-state button
  - Checked/not checked

UICheckbox

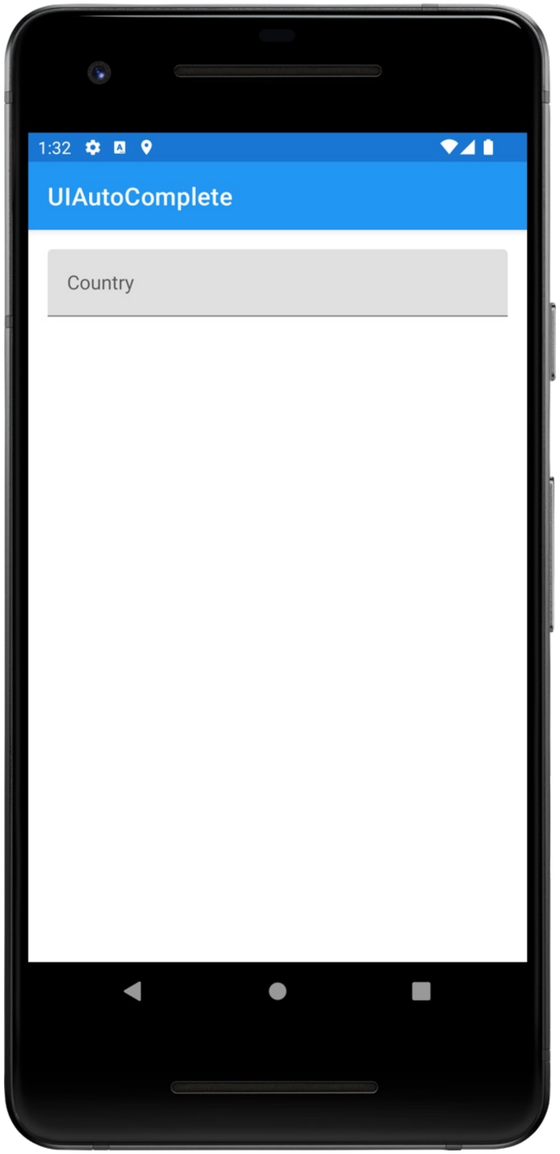

1:29
UICheckBox
☐ I'm not checked
HIDE CHECKBOX

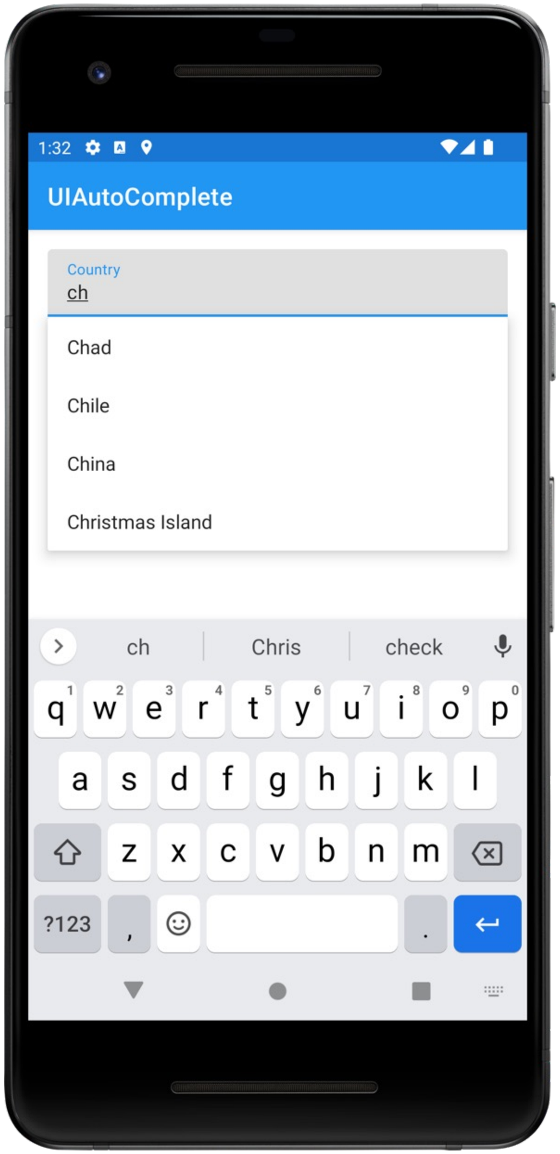

1:29
UICheckBox
☑ I'm checked
HIDE CHECKBOX

AutoCompleteTextView

An editable text field that provides completion suggestions as the user types in text

UIAutoComplete
TextView





View Event Sources

- User interaction
  - Touch
  - Keyboard/trackball/D-pad
- System control
  - Lifecycle changes

Handling View Events

You will often handle events using listeners

Many Listener interfaces defined by View class

View Listener interfaces

OnClickListener.onClick()

- View has been clicked

OnLongClickListener.onLongClick()

- View has been pressed & held

View Listener interfaces

OnFocusChangeListener.onFocusChange()

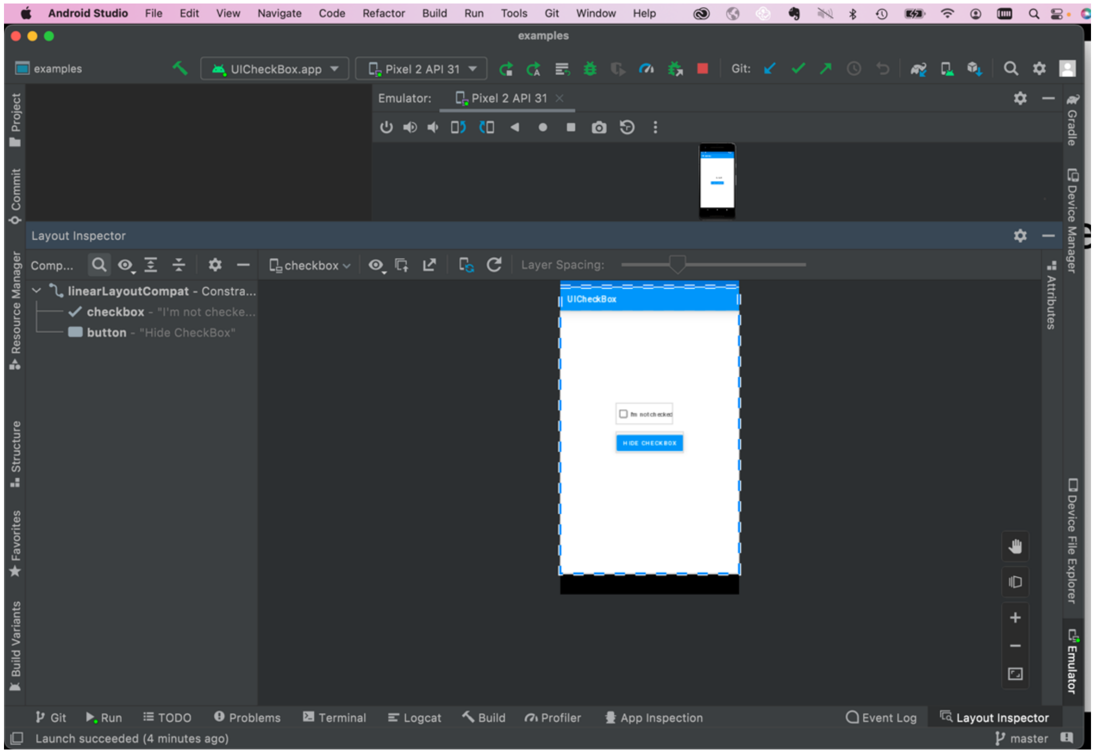
View has received or lost focus

OnKeyListener.onKey()

View is about to receive a hardware key press

Displaying Views

Views within a UI are logically organized as a tree



```
linearLayoutCompat - Constra...
  checkbox - "I'm not checke...
  button - "Hide CheckBox"
```

Launch succeeded (4 minutes ago)

Displaying Views

Displaying/refreshing the UI has multiple steps

- Measure – get dimensions of each View
- Layout – Position each View
- Draw – Draw each view

Handling View Events

Create View subclasses

Override View methods

Handling View Events

onMeasure()

Determine the size of this View and its children

onLayout()

Assign a size and position to all View’s children

onDraw()

Render View content

Handling View Events

onFocusChanged()

Called when View's focus state has changed

onKeyUp(), onKeyDown()

Called when a hardware key event has occurred

onWindowVisibilityChanged()

Window containing view has changed its visibility status

ViewGroup

An invisible View that contains other Views

Used for grouping & organizing a set of Views

Base class for View containers and Layouts

Some Predefined ViewGroups

RadioGroup

TimePickerFragment

DatePickerFragment

WebView

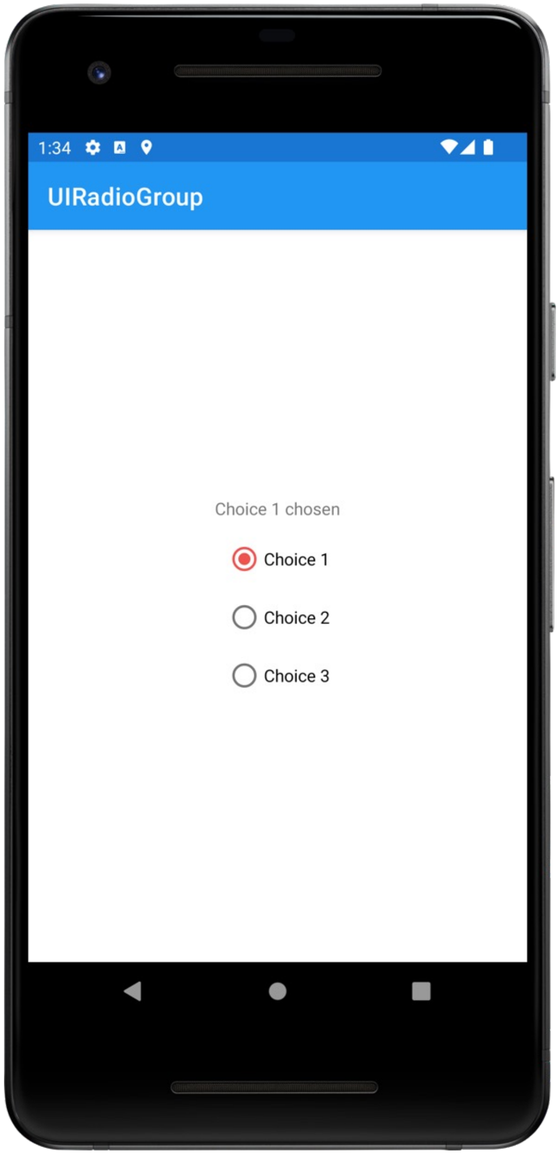
MapView

RadioGroup

A ViewGroup containing a set of Radio Buttons

Only one RadioButton can be selected at any one time

UIRadioGroup



1:34

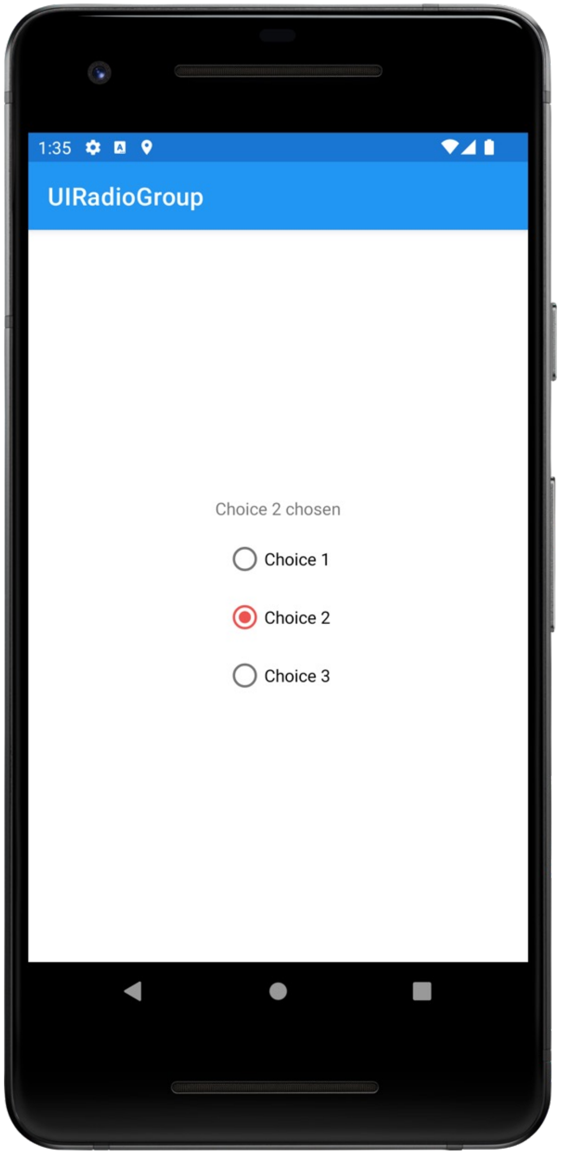
UIRadioGroup

Choice 1 chosen

● Choice 1

○ Choice 2

○ Choice 3



1:35

UIRadioGroup

Choice 2 chosen

○ Choice 1

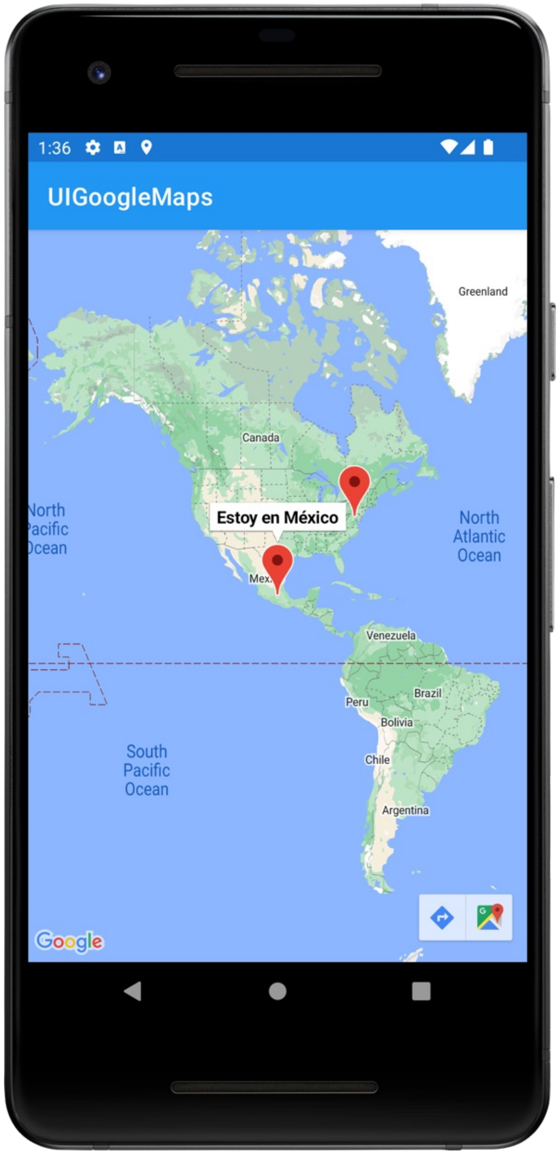
● Choice 2

○ Choice 3

MapView

A ViewGroup that displays a Map

UIGoogleMaps



1:36

UIGoogleMaps

Estoy en México

Adapters & AdapterViews

AdapterViews are Views whose children and data are managed by an Adapter

Interaction pattern

- Adapter manages the data and provides data Views to AdapterView
- AdapterView displays the data Views

RecyclerView

An AdapterView that displays a scrollable list of selectable items

Data items managed by a RecyclerView.Adapter

Some RecyclerView Terminology

Adapter: Component responsible for providing views that represent items in a data set

Position: The position of a data item within an Adapter

Index: The index of an attached child view as used in a call to ViewGroup.getChildAt()

Binding: The process of preparing a child view to display data corresponding to a position within the adapter

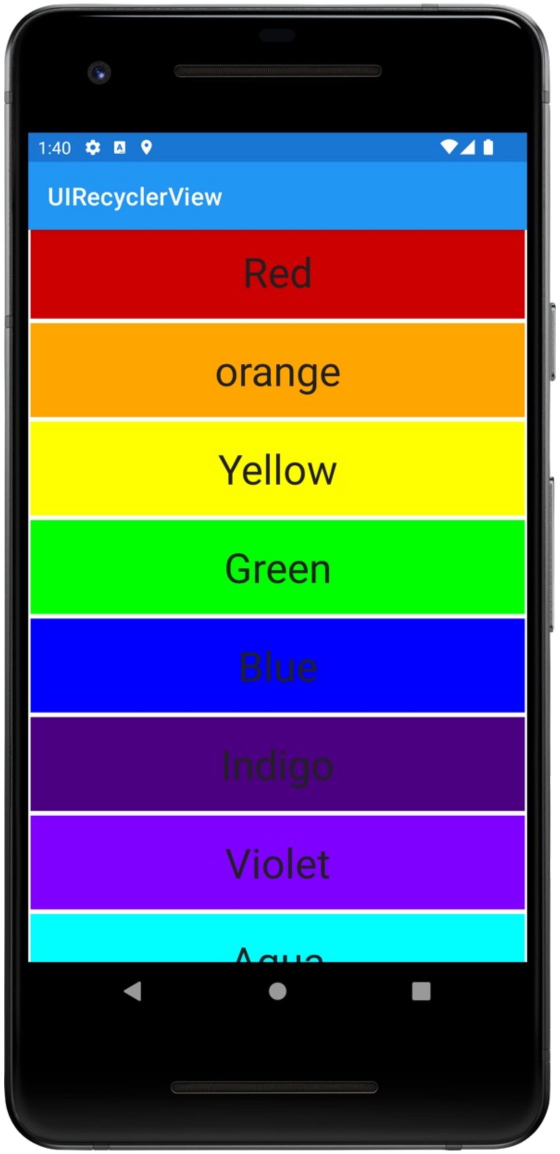
Recycle (view): A view previously used to display data for a specific adapter position may be placed in a cache for later reuse to display the same type of data again later

Scrap (view): A child view that has entered into a temporarily detached state during layout. Scrap views may be reused at a later time

Dirty (view): A child view that must be rebound by the adapter before being displayed

ViewHolder: A class that caches information about a view managed by the adapter

UIRecyclerView



1:40

UIRecyclerView

Red

orange

Yellow

Green

Blue

Indigo

Violet

Aqua

ViewPager

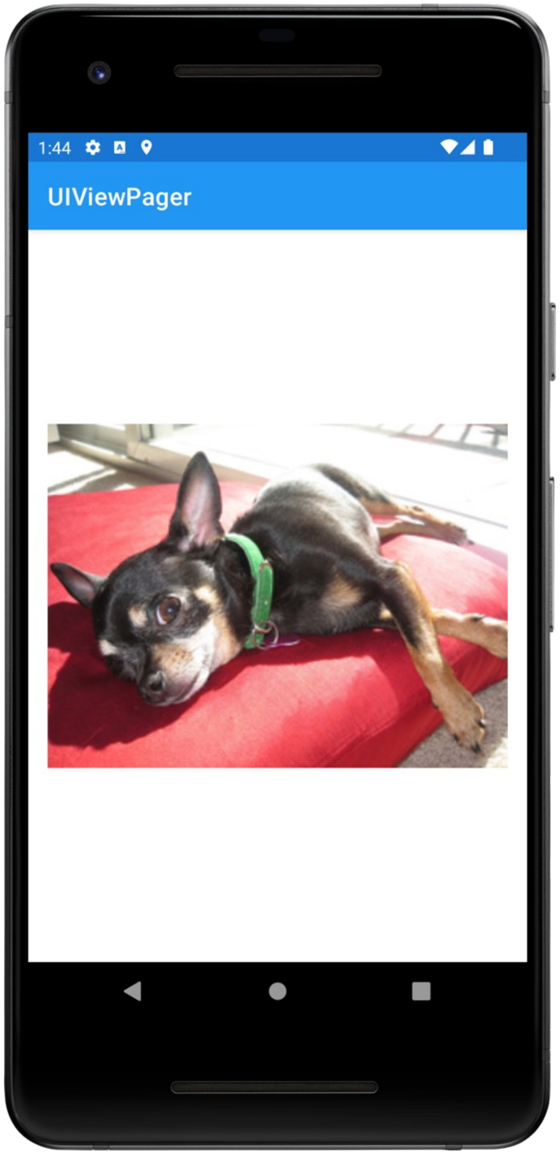
A ViewGroup showing a horizontally scrolling list

Items managed by a PagerAdapter

Builtin PagerAdapters using Fragments

- FragmentStatePagerAdapter

UIViewPager



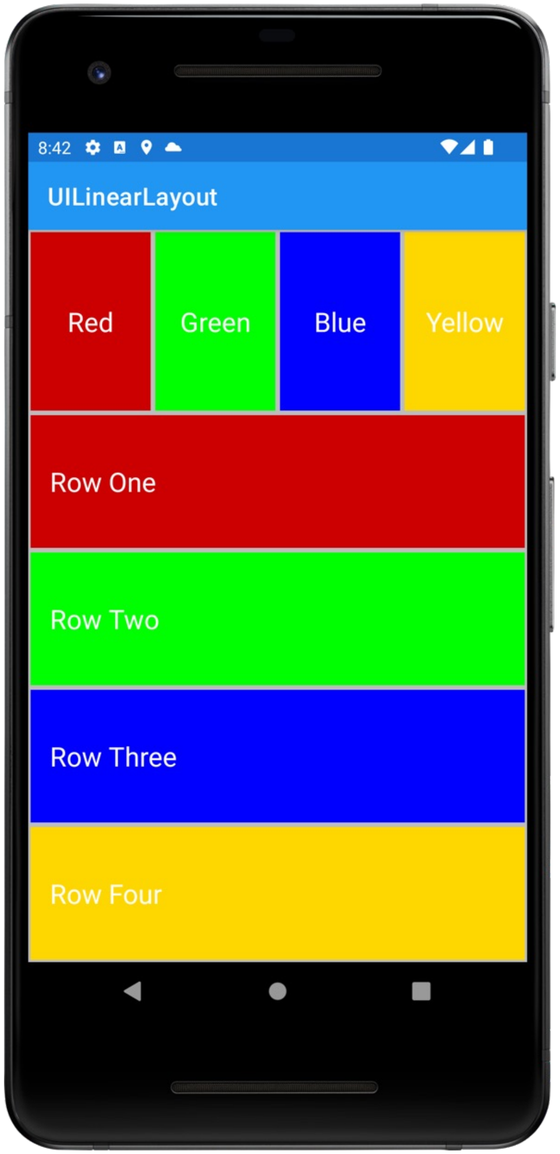
Layouts

A generic Viewgroup that defines a structure/rules for positioning the Views it contains

LinearLayout

Child Views arranged in a single horizontal or vertical row

UILinearLayout



ConstraintLayout

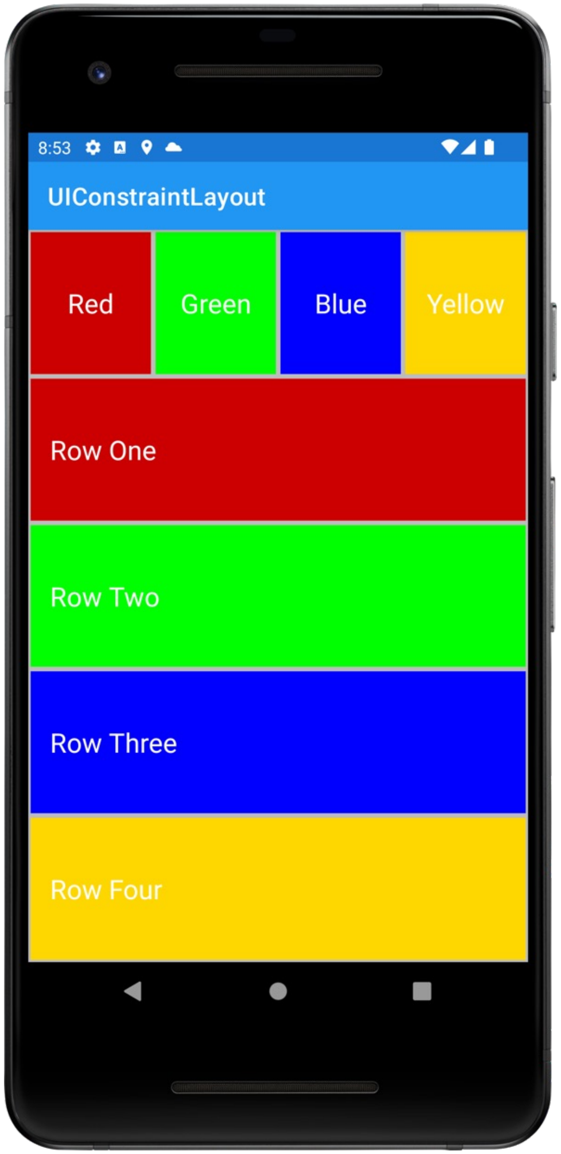
Combines features of LinearLayout and RelativeLayout (use now discouraged)

Avoids deeply nested layout structures with goal of improving drawing performance

Considered default UI layout for Android going forward

See: https://developer.android.com/reference/androidx/constraintlayout/widget/ConstraintLayout

UIConstraintLayout



8:53

UIConstraintLayout

Menus and ActionBar

Activities support menus

Activities can

- Add items to a menu
- Handle clicks on the menu items

Some Menu Types

Options

- Menu shown when user presses the menu button

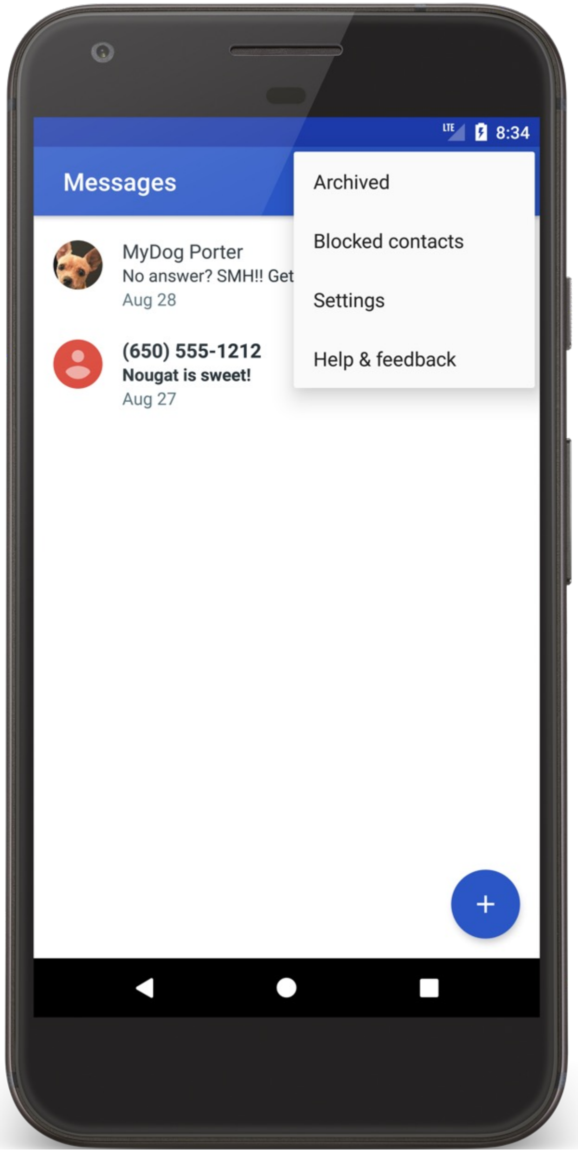
Context

- View-specific menu shown when user touches and holds the View

Submenu

- A menu activated when user touches a visible menu item

Options Menus



8:34

Messages

Archived

Blocked contacts

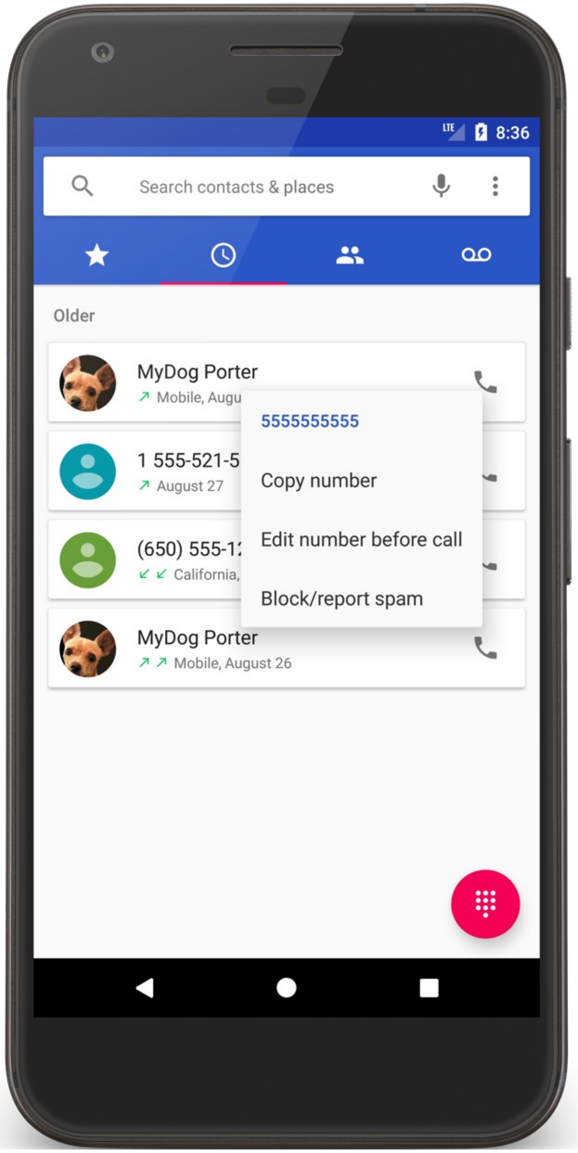
Settings

Help & feedback

MyDog Porter
No answer? SMH!! Get
Aug 28

(650) 555-1212
Nougat is sweet!
Aug 27

Context Menus



Creating Menus

- Define menu resource in XML file
  - Store in res/menu/filename.xml

Managing Options Menus

Implement MenuProvider Interface

Inflate menu resource using Menu Inflater in onCreateMenu()

Handle menu item selection in appropriate onMenuItemSelected()

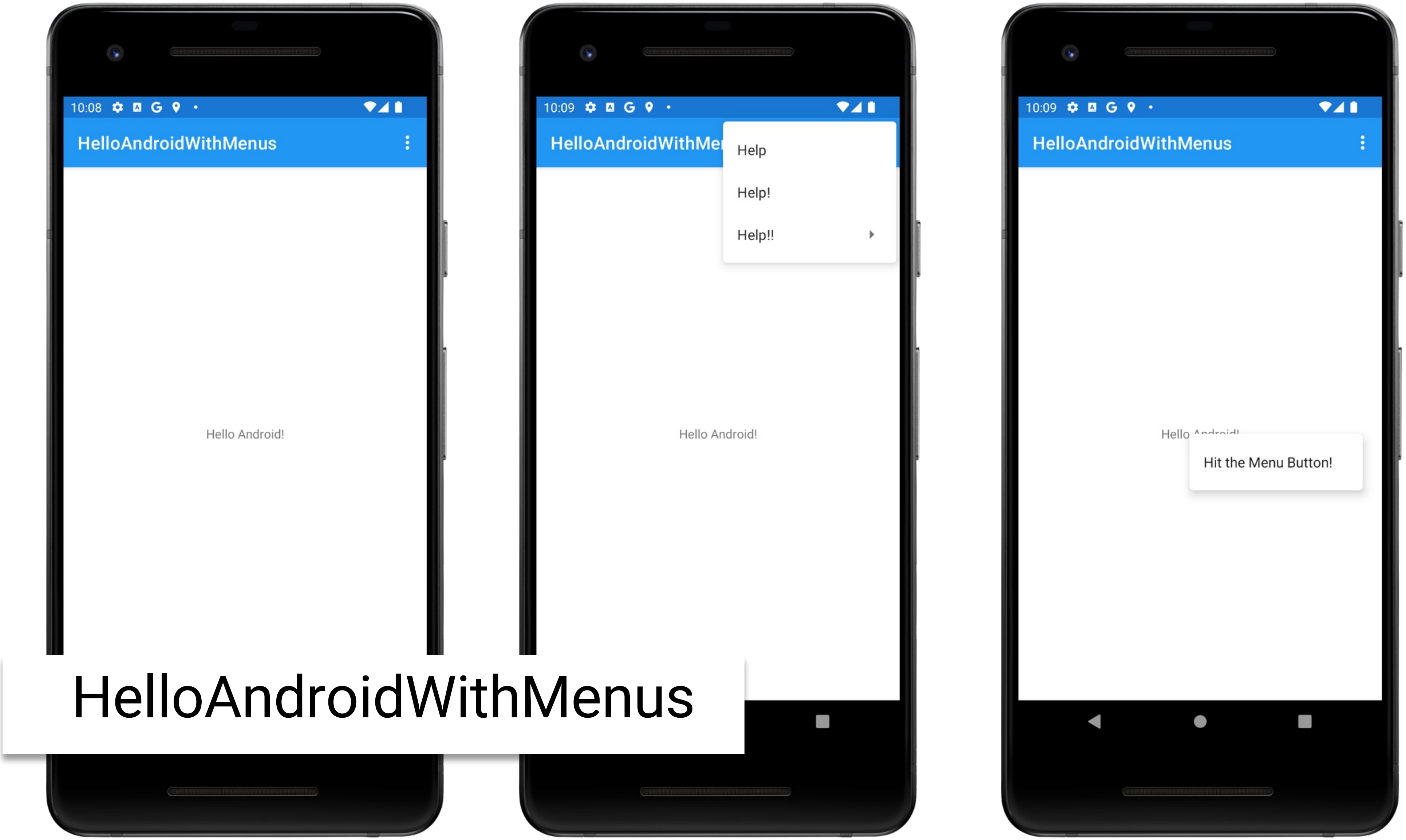
Managing Context Menus

Implement MenuProviderInterface

Inflate menu resource using Menu Inflater in onCreateContextMenu()

Handle item selection in appropriate onContextItemSelected()

HelloAndroidWithMenus



Hello Android!

Hello Android!

Hello Android!

Hit the Menu Button!

Menus

Many other features supported

- Grouping menu items
- Binding shortcut keys to menu items
- Binding Intents to menu items

Dialogs

Independent subwindows used by Activities to communicate with user

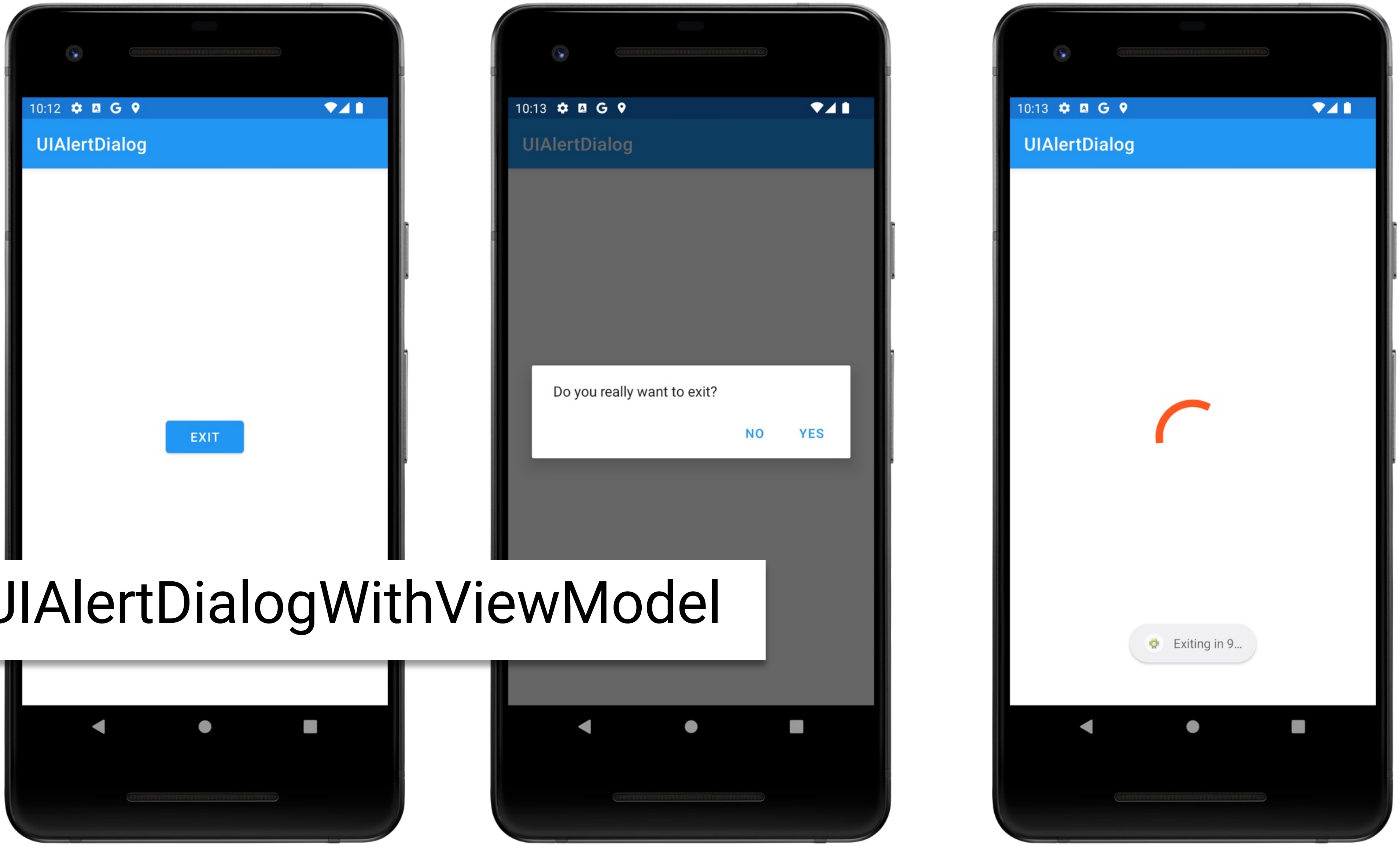
Some Dialog Subclasses

AlertDialog

DatePickerDialog

TimePickerDialog

UIAlertDialogWithViewModel



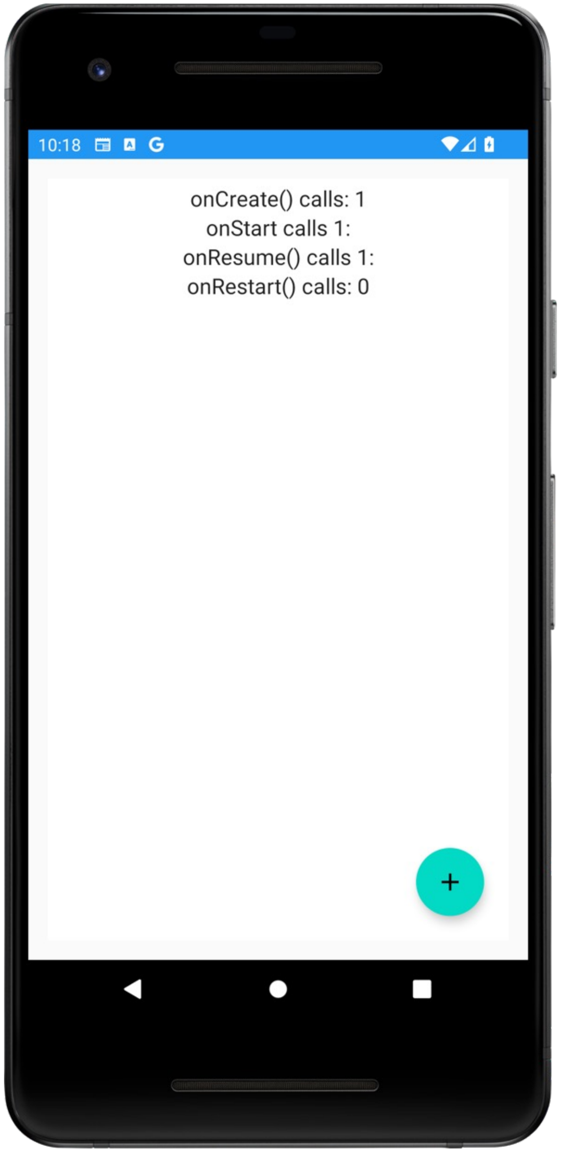
Do you really want to exit?

Exiting in 9...

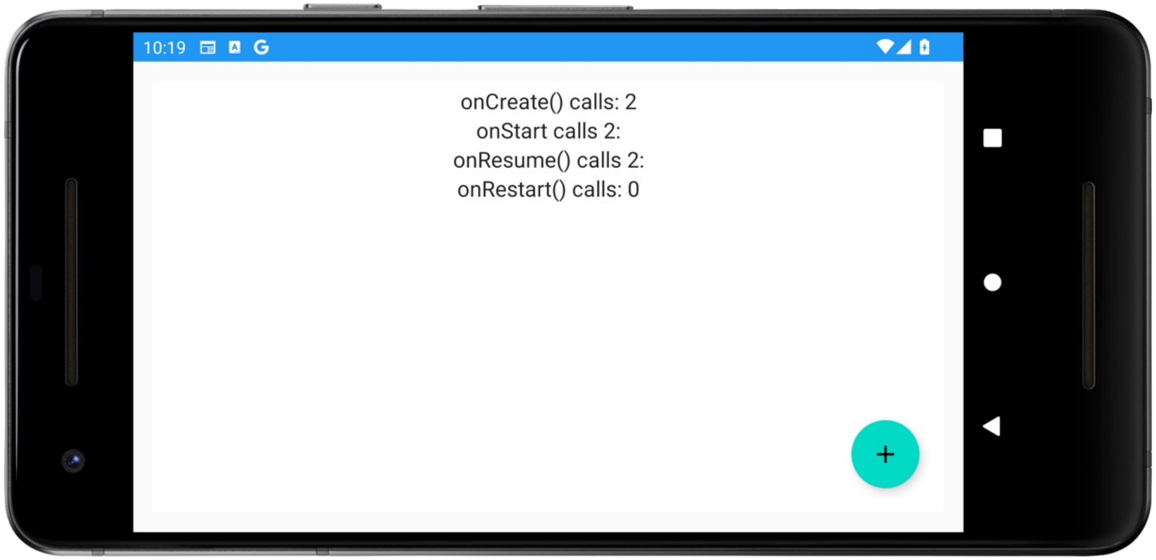
Jetpack Compose

- Describe individual UI elements within Composable functions
- Assemble UI hierarchy from composable functions
- Implementation concept
  - UI is immutable. UI presents current app state
  - When app state changes, Android recreates those parts of the UI hierarchy that have changed

See: https://developer.android.com/jetpack/compose



onCreate() calls: 1
onStart calls 1:
onResume() calls 1:
onRestart() calls: 0



onCreate() calls: 2
onStart calls 2:
onResume() calls 2:
onRestart() calls: 0

UICompose

Next

BroadcastReceivers

Example Applications

UIButton

UIToggleButton

UICheckbox

UIRatingBar

UIAutoCompleteTextView

UIRadioGroup

UIGoogleMaps

UIRecyclerView

UIViewPager

UILinearLayout

UIConstraintLayout

HelloAndroidWithMenus

UIAlertDialogWithViewModel