Views and ViewGroups

CS 240 – Advanced Programming Concepts

Views and ViewGroups

- View = Widget = Control
  - Examples: Button, Switch, Spinner, TextView, EditText
- ViewGroup = Container
  - Views that contain other views
  - Examples: LinearLayout, TableLayout, ScrollView
- Common Attributes
  - id (i.e. **android:id="@+id/myViewId"**)
  - Layout_width, layout_height
    - Values: match_parent, wrap_content, 16dp

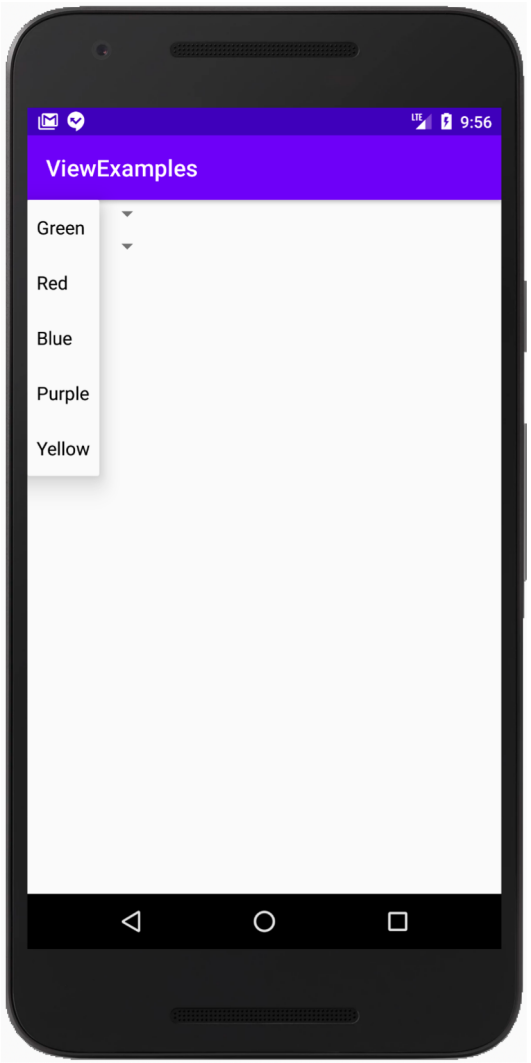
Views (Attributes and Listeners)

- TextView (text labels)
  - text, textAppearance
  - .setClickable(boolean) – make clickable
  - .setOnClickListener(View.OnClickListener)
- EditText (text fields)
  - inputType
  - .addTextChangedListener(TextWatcher)
- Button
  - text
  - .setOnClickListener(View.OnClickListener)
- ImageView (display image)
  - .setClickable(boolean) – make clickable
  - .setImageDrawable(IconDrawable) – set icon to display
  - .setOnClickListener(View.OnClickListener)

Views (Attributes and Listeners)

- Switch (on/off)
  - .setChecked(boolean) – set check state
  - .setOnCheckedChangeListener(CompoundButton.OnCheckedChangeListener)
- Spinner (dropdown list)
  - entries or .setAdapter(ArrayAdapter) – specify list values
  - .setSelection(int) – specify selected item
  - onItemSelectedListener(AdapterView.OnItemSelectedListener)
- SearchView
  - queryHint – Background text displayed when the field is empty
  - iconifiedByDefault – Display the field or just an icon until clicked
  - .setIconified(boolean) – make always visible
  - .setOnQueryTextListener(SearchView.OnQueryTextListener)
- Space (blank space)
  - set layout_width and layout_height to specific values (e.g., 30dp)

Setting Entries of a Spinner in the XML Layout File

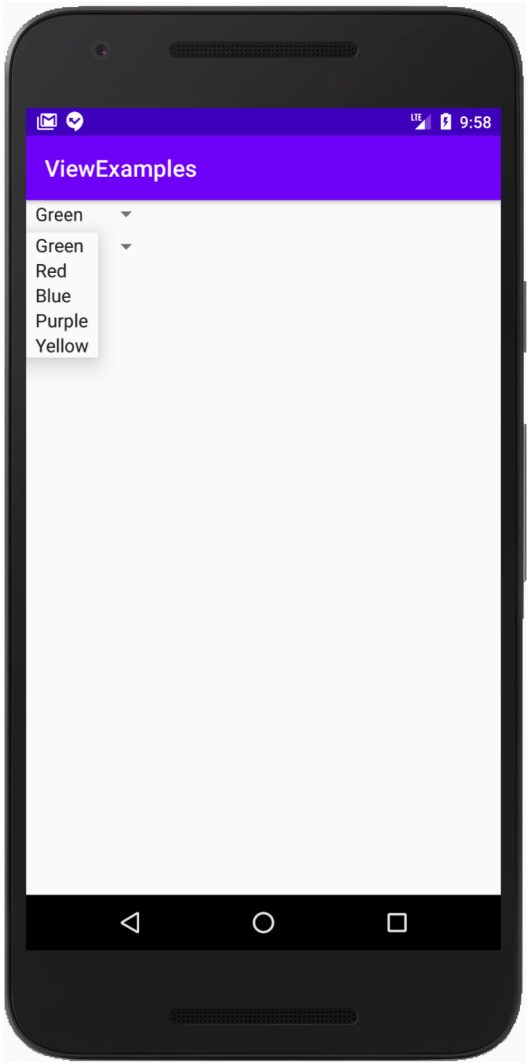



```
<Spinner
    android:id="@+id/colorSelector1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="32dp"
    android:layout_marginBottom="4dp"
    android:entries="@array/colorChoices"/>
```

strings.xml

```
1   <resources>
2       <string name="app_name">ViewExamples</string>
3
4       <string-array name="colorChoices">
5           <item>Green</item>
6           <item>Red</item>
7           <item>Blue</item>
8           <item>Purple</item>
9           <item>Yellow</item>
10      </string-array>
11
12  </resources>
```

Setting Entries of a Spinner in Code




```
<Spinner
    android:id="@+id/colorSelector2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="32dp"
    android:layout_marginBottom="4dp"/>
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    Spinner colorSelector2 = findViewById(R.id.colorSelector2);
    ArrayAdapter<CharSequence> adapter =
            ArrayAdapter.createFromResource( context: this,
            R.array.colorChoices, android.R.layout.simple_spinner_item);
    colorSelector2.setAdapter(adapter);
}
```

Advanced Views (covered later)

- RecyclerView
  - Dynamic list of items
- ExpandableListView
- MapView

View Groups

- ScrollView
  - Wraps around any one View to make it scrollable
  - The one View can be a ViewGroup/layout
  - Example: GridLayoutExampleWithScrollView
- LinearLayout
  - orientation: vertical or horizontal
  - Example: GeoQuiz/QuizActivity
- FrameLayout
  - Used as a placeholder for dynamically created fragments
  - Example: FragmentExample

View Groups: TableLayout

- Contains multiple TableRow objects and/or other layouts
- Columns determined by row with most views
- Useful Attributes:
  - gravity
    - Attribute of any widget
    - Use to position the widget in it's column
    - Example: **android:gravity="end"** will right justify the widget in it's column
  - stretchColumns
    - Attribute of the TableLayout
    - Specify which if any column should take up empty space
    - Example: **android:stretchColumns="3"** (4^{th} column will take space)
  - layout_span
    - Attribute of a widget inside a TableRow
    - Specify that the widget should span multiple columns
    - Example: **android:layout_span="3"** (declaring item will span 3 columns)
- Example: TableLayoutExample

View Groups: GridLayout

- Very similar to TableLayout
  - No need to use TableRow or any other indicator of rows
  - Specify columns with columnCount attribute
  - Grid filled row-by-row from left to right
- Useful Attributes:
  - columnCount
    - Attribute of GridLayout
  - layout_gravity (use layout_gravity, instead of gravity)
    - Attribute of any widget
    - Use to position the widget in it's column(s) or row(s)
    - Example: **android:layout_gravity="end"** will right justify the widget in it's column or columns
    - Example: **android:layout_gravity="fill-horizontal"** will fill the entire column or columns if using layout_columnSpan
  - layout_columnSpan
    - Attribute of any widget
    - Use to span multiple columns
    - Example: **android:layout_columnSpan="3"** (declaring item will span 3 columns)
  - layout_rowSpan
    - Attribute of and widget
    - Use to span multiple rows
    - Example: **android:layout_rowSpan="2"** (declaring item will span 2 rows)
- Example: GridLayoutExample

View Groups: ConstraintLayout

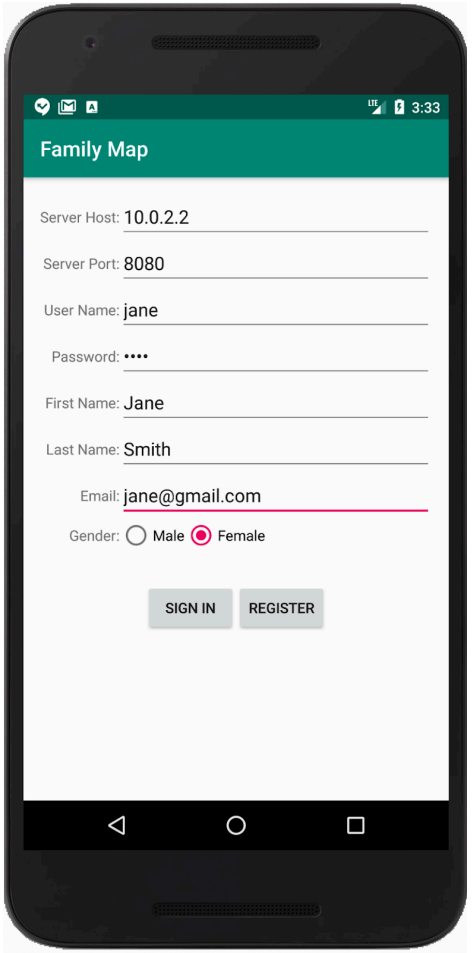
- Allows you to connect views to their parents and to each other in design or text view
- Designed to allow visual layout in the design view
- Useful Attributes:
  - layout_constraint*_to*Of (substitute Top, Bottom, Right or Left for *)
    - Attribute of any widget
    - Use to position the widget in it's parent or relative to another View
    - Example: **app:layout_constraintTop_toTopOf="parent"**
  - Layout_margin*
    - Attribute of any widget
    - Use to put space between views or between a view and it's parent
    - Example: **android:layout_marginBottom="8dp"**
- Example: ConstraintLayoutExample

View Groups: RelativeLayout

- Very similar to ConstraintLayout
  - Not as easily manipulated visually in the design view, but takes fewer attributes to position the views
- Useful Attributes:
  - layout_* (many substitutions for *)
    - Attribute of any widget
    - Use to position the widget in it's parent or relative to another View
    - Example: **android:layout_centerInParent="true"**
    - Example: **android:layout_above="@id/getVersionButton"**
    - Example: **android:layout_centerHorizontal="true"**
  - Layout_margin*
    - Attribute of any widget
    - Use to put space between views or between a view and it's parent
    - Example: **android:layout_marginBottom="8dp"**
- Example: RelativeLayoutExample

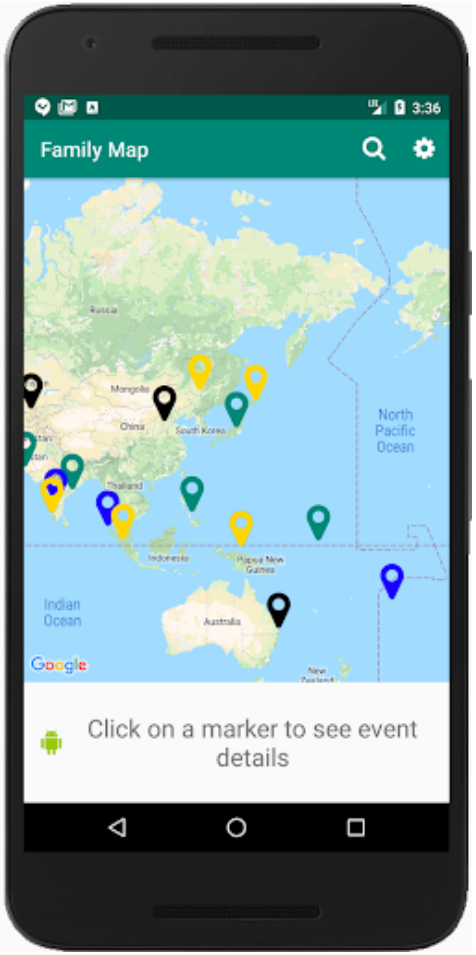
Family Map Client: Login Fragment

- How would you create this view?
- Does it need to scroll?



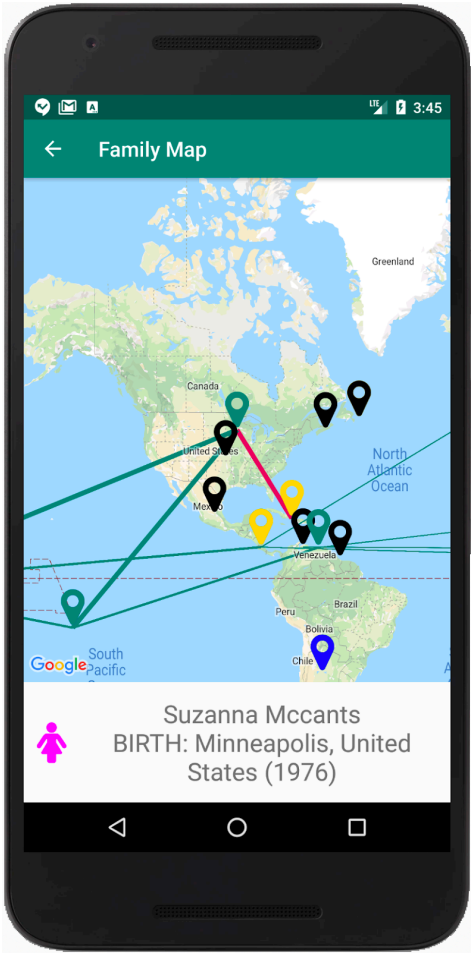
Family Map Client: Map Fragment

- How would you create this view?
- Does it need to scroll?



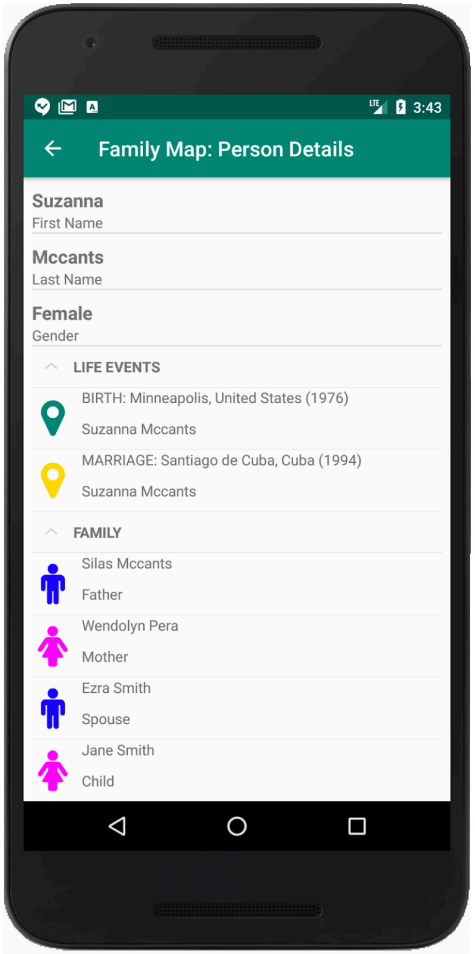
Family Map Client: Event Activity

- How would you create this view?
- Does it need to scroll?



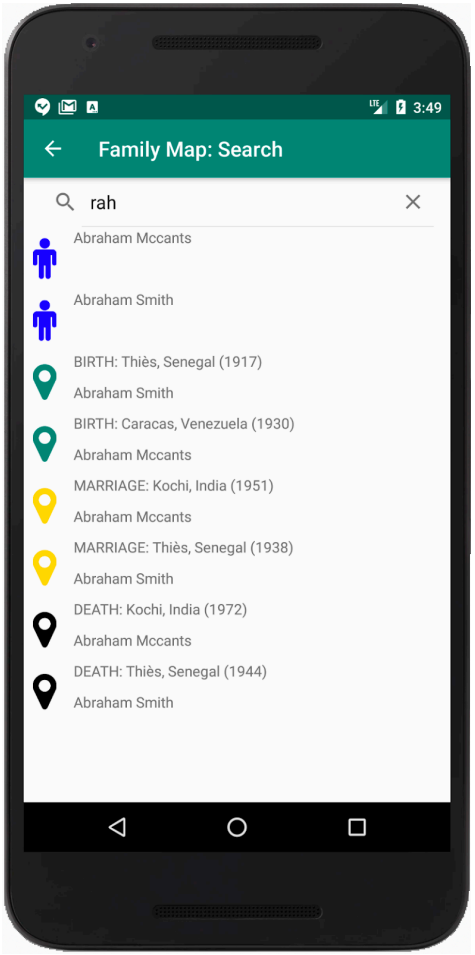
Family Map Client: Person Activity

- How would you create this view?
- Does it need to scroll?



Family Map Client: Search Activity

- How would you create this view?
- Does it need to scroll?



Family Map Client: Settings Activity

- How would you create this view?
- Does it need to scroll?