twitch.tv/dancojocar

youtube.com/dancojocar

Lecture #11
Jetpack Compose & SwiftUI

Mobile Applications 2020-2021

twitch.tv/dancojocar

youtube.com/dancojocar

Happy New Year!

2021

twitch.tv/dancojocar

youtube.com/dancojocar

twitch.tv/dancojocar

youtube.com/dancojocar

A brief history of Android

HTC Sooner

twitch.tv/dancojocar
youtube.com/dancojocar

HTC Sooner

OMAP 850 processor

64 MB of RAM

64 MB of on-board memory

A “generous”
320 x 240 pixel resolution

NO touch input

twitch.tv/dancojocar

youtube.com/dancojocar

T-Mobil G1

A “generous”
320 x 240 pixel resolution

twitch.tv/dancojocar

youtube.com/dancojocar

T-Mobil G1

A “generous”
320 x 240 pixel resolution

Qualcomm MSM7201A

192MB RAM, 256MB

TFT capacitive touchscreen

twitch.tv/dancojocar

youtube.com/dancojocar

T-Mobil G1

A “generous”
320 x 240 pixel resolution

Qualcomm MSM7201A

192MB RAM, 256MB

TFT capacitive touchscreen

320 x 480 pixels

Pixel 5

twitch.tv/dancojocar

youtube.com/dancojocar

twitch.tv/dancojocar

youtube.com/dancojocar

Pixel 5

Qualcomm SM7250 Snapdragon 765G (7 nm)

8GB RAM, 128GB

1080 x 2340 pixels

twitch.tv/dancojocar

youtube.com/dancojocar

Pixel 5

twitch.tv/dancojocar

youtube.com/dancojocar

Pixel 5

Architectural Components approach to UI

- What are developers asking for?
- What should the framework team do to help developers?
- How to make things easier?

Among Top Answers

twitch.tv/dancojocar
youtube.com/dancojocar

Among Top Answers

- Unbundle the UI toolkit!

Among Top Answers

twitch.tv/dancojocar

youtube.com/dancojocar

- Unbundle the UI toolkit!
- aka move android.widget to Jetpack.

Among Top Answers

twitch.tv/dancojocar

youtube.com/dancojocar

- Unbundle the UI toolkit!
- aka move android.widget to Jetpack.
- … what else can be fixed along the way?

Room for Improvements

twitch.tv/dancojocar
youtube.com/dancojocar

How many lines should a class have to be considered big?

CanvasExample – /usr/local/Caskroom/android-sdk/4333796/sources/android-30/android/view/View.java

app | Pixel 3a API 30 | Git:

View.java

Code | Split | Design

```
30393         */
30394         public void removeOnUnhandledKeyEventListener(OnUnhandledKe
30395             if (mListenerInfo != null) {
30396                 if (mListenerInfo.mUnhandledKeyListeners != null
30397                         && !mListenerInfo.mUnhandledKeyListeners.is
30398                     mListenerInfo.mUnhandledKeyListeners.remove(lis
30399                     if (mListenerInfo.mUnhandledKeyListeners.isEmpt
30400                         mListenerInfo.mUnhandledKeyListeners = null
30401                         if (mParent instanceof ViewGroup) {
30402                             ((ViewGroup) mParent).decrementChildUnh
30403                         }
30404                     }
30405                 }
30406             }
30407         }
30408     }
30409
```

Database Inspector | TODO | 9: Git | Profiler | 6: Logcat | Terminal | Layout Inspector | Event Log

Android Studio 4.2 Beta 2 available: // Update... (2 minutes ago) 30404:18 LF UTF-8 2 spaces master

itch.tv/dancojocar
be.com/dancojocar

Room for Improvements

twitch.tv/dancojocar
youtube.com/dancojocar

```
public class Spinner extends AbsSpinner implements OnClickListener
```

Room for Improvements

twitch.tv/dancojocar
youtube.com/dancojocar

```
public class Spinner extends AbsSpinner implements OnClickListener
```

Room for Improvements

twitch.tv/dancojocar
youtube.com/dancojocar

```
public class Button extends TextView
```

Room for Improvements

twitch.tv/dancojocar
youtube.com/dancojocar

```
public class Button extends TextView
```

Too Much Code

Too Much Code

- My{Activity|Fragment}.{kt|java}

Too Much Code

- My{Activity|Fragment}.{kt|java}
- main_{activity|fragment}.xml

Too Much Code

- My{Activity|Fragment}.{kt|java}
- main_{activity|fragment}.xml
- … stuff in res folder.

Too Much Code

twitch.tv/dancojocar
youtube.com/dancojocar

- My{Activity|Fragment}.{kt|java}
- main_{activity|fragment}.xml
- … stuff in res folder.
- Some more stuff in styles.xml

Goals

twitch.tv/dancojocar
youtube.com/dancojocar

Goals

- Unbundle from platform releases.

Goals

- Unbundle from platform releases.
- Fewer choices when trying to build a UI element.

Goals

twitch.tv/dancojocar
youtube.com/dancojocar

- Unbundle from platform releases.
- Fewer choices when trying to build a UI element.
- Clarify state ownership and event handler.

Goals

twitch.tv/dancojocar
youtube.com/dancojocar

- Unbundle from platform releases.
- Fewer choices when trying to build a UI element.
- Clarify state ownership and event handler.
- Write less code.

Can we go back to a simpler time?

twitch.tv/dancojocar

youtube.com/dancojocar

```
fun main(){
  print("Hello World!")
}
```

Can we go back to a simpler time?

twitch.tv/dancojocar

youtube.com/dancojocar

```
fun Greeting(){
  print(“Hello World!")
}
```

Can we go back to a simpler time?

twitch.tv/dancojocar

youtube.com/dancojocar

```
fun Greeting(){
  Text(“Hello World!")
}
```

twitch.tv/dancojocar

youtube.com/dancojocar

developer.android.com/jetpack/compose

twitch.tv/dancojocar

youtube.com/dancojocar

developer.android.com/jetpack/compose

UI as a Function!

twitch.tv/dancojocar

youtube.com/dancojocar

- Take data as input.
- Emit UI hierarchy when invoked.

```
@Composable
fun Greeting(name: String){
  Text("Hello $name")
}
```

UI as a Function!

twitch.tv/dancojocar

youtube.com/dancojocar

```
@Composable
fun Greeting(name: String){
  Text("Hello $name")
}
```

What is Compose?

twitch.tv/dancojocar
youtube.com/dancojocar

What is Compose?

twitch.tv/dancojocar
youtube.com/dancojocar

- A new set of Jetpack UI widgets.
  - Not widgets/fragments
  - Are much smaller => Functions
- A Kotlin compiler plugin.
- Fully compatible with the existing view system.
- Experimental, still in development.

Use in Existing App

```
class MainActivity : AppCompatActivity() {
  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContent {
      Greeting("World!")
    }
  }
}
```

twitch.tv/dancojocar

youtube.com/dancojocar

Composable UI Widget

```
@Composable
fun Greeting(name: String){
  Text("Hello $name")
}
```

Intercept & Rewrite

twitch.tv/dancojocar
youtube.com/dancojocar

```
@Composable
fun Greeting(name: String){
  for (i in 1..10) {
    Text("Hello $name")
  }
}
```

```
@Composable
fun Greeting(names: List<String>){
  for (name in names) {
    Text("Hello $name")
  }
}
```

```
@Composable
fun Greeting(names: List<String>){
  if (names.isEmpty()){
    Text("No soup for you!")
  } else {
    for (name in names) {
      Text("Hello $name")
    }
  }
}
```

Composable Building Blocks

twitch.tv/dancojocar

youtube.com/dancojocar

- Composable functions are defined in terms of other composable functions.

```
@Composable
fun NewsFeed(stories: List<Story>) {
  for (story in stories) {
    StoryWidget(story)
  }
}

@Composable
fun StoryWidget(story: Story) {
  Padding(8.dp){
    Column{
      Title(story.title)
      Image(story.image)
      Text(story.content)
    }
  }
}
```

twitch.tv/dancojocar
youtube.com/dancojocar

RecyclerView Adapter

```
public class MyRecyclerViewAdapter extends RecyclerView.Adapter<MyRecyclerViewAdapter.ViewHolder> {

    private List<String> mData;
    private LayoutInflater mInflater;
    private ItemClickListener mClickListener;

    // data is passed into the constructor
    MyRecyclerViewAdapter(Context context, List<String> data) {
        this.mInflater = LayoutInflater.from(context);
        this.mData = data;
    }

    // inflates the row layout from xml when needed
    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View view = mInflater.inflate(R.layout.recyclerview_row, parent, false);
        return new ViewHolder(view);
    }

    // binds the data to the TextView in each row
    @Override
```

twitch.tv/dancojocar

youtube.com/dancojocar

```
}

// binds the data to the TextView in each row
@Override
public void onBindViewHolder(ViewHolder holder, int position) {
    String animal = mData.get(position);
    holder.myTextView.setText(animal);
}

// total number of rows
@Override
public int getItemCount() {
    return mData.size();
}


// stores and recycles views as they are scrolled off screen
public class ViewHolder extends RecyclerView.ViewHolder implements View.OnClickListener {
    TextView myTextView;

    ViewHolder(View itemView) {
        super(itemView);
        myTextView = itemView.findViewById(R.id.tvAnimalName);
        itemView.setOnClickListener(this);
    }

    @Override
```

twitch.tv/dancojocar

youtube.com/dancojocar

```
        }

        @Override
        public void onClick(View view) {
            if (mClickListener != null) mClickListener.onItemClick(view, getAdapterPosition());
        }
    }

    // convenience method for getting data at click position
    String getItem(int id) {
        return mData.get(id);
    }

    // allows clicks events to be caught
    void setClickListener(ItemClickListener itemClickListener) {
        this.mClickListener = itemClickListener;
    }

    // parent activity will implement this method to respond to click events
    public interface ItemClickListener {
        void onItemClick(View view, int position);
    }
}
```

```
@Composable
fun NewsFeed(stories: List<Story>) {
  ScrollingList(stories) { story ->
    StoryWidget(story)
  }
}

@Composable
fun StoryWidget(story: Story) {
  Padding(8.dp){
    Column{
      Title(story.title)
      Image(story.image)
      Text(story.content)
    }
  }
}
```

twitch.tv/dancojocar
youtube.com/dancojocar

```
@Composable
fun NewsFeed(stories: LiveDate<List<Story>>) {
  ScrollingList(stories.observer()) { story ->
    StoryWidget(story)
  }
}
```

twitch.tv/dancojocar

youtube.com/dancojocar

```
@Composable
fun StoryWidget(story: Story) {
  val image = asyncLoad(defaultPlaceholder) {
    loadImage(story.imageUri)
  }
  Card(cornerRadius = 4.dp, elevation = 4.dp) {
    Column {
      Image(image)
      Padding(16.dp) {
        Text(story.headline)
      }
    }
  }
}
```

twitch.tv/dancojocar

youtube.com/dancojocar

```
@Composable
fun StoryWidget(story: Story) {
  val image = asyncLoad(defaultPlaceholder) {
    loadImage(story.imageUri)
  }
  Card(cornerRadius = 4.dp, elevation = 4.dp) {
    Column {
      Image(image)
      Padding(16.dp) {
        Text(story.headline)
      }
    }
  }
}
```

```
data class Story(
    var imageUri: Uri,
    var headling: String
)
```

twitch.tv/dancojocar

youtube.com/dancojocar

```
@Composable
fun StoryWidget(story: Story) {
  val image = asyncLoad(defaultPlaceholder) {
    loadImage(story.imageUri)
  }
  Card(cornerRadius = 4.dp, elevation = 4.dp) {
    Column {
      Image(image)
      Padding(16.dp) {
        Text(story.headline)
      }
    }
  }
}
```

```
@Model
data class Story(
    var imageUri: Uri,
    var headling: String
)
```

Top-down Data Flow

twitch.tv/dancojocar

youtube.com/dancojocar

```
@Composable
fun NewsFeed(stories: List<Story>) {
  for (story in stories) {
    StoryWidget(story)
  }
}

@Composable
fun StoryWidget(story: Story) {
  Padding(8.dp){
    Column{
      Title(story.title)
      Image(story.image)
      Text(story.content)
    }
  }
}
```

twitch.tv/dancojocar

youtube.com/dancojocar

```
@Composable
fun NewsFeed(stories: List<Story>,
             onSelected: (Story) -> Unit) {
  ScrolingList(stories) { story ->
    StoryWidget(story, onClick = { onSelected(story) })
  }
}

@Composable
fun StoryWidget(story: Story, onClick: () -> Unit) {
  Clickable(onClick) {
    Column {
      Padding(8.dp) {
        Title(story.headling)
        Image(story.image)
        Text(story.content)
      }
    }
  }
}
```

twitch.tv/dancojocar

youtube.com/dancojocar

twitch.tv/dancojocar

youtube.com/dancojocar

twitch.tv/dancojocar

youtube.com/dancojocar

twitch.tv/dancojocar

youtube.com/dancojocar

twitch.tv/dancojocar

youtube.com/dancojocar

twitch.tv/dancojocar

youtube.com/dancojocar

twitch.tv/dancojocar

youtube.com/dancojocar

twitch.tv/dancojocar

youtube.com/dancojocar

twitch.tv/dancojocar

youtube.com/dancojocar

Single Source of Truth

twitch.tv/dancojocar

youtube.com/dancojocar

Multiple Sources of Truth

twitch.tv/dancojocar
youtube.com/dancojocar

Multiple Sources of Truth

- The app has a state.
- The checkbox has a state.
- The checkbox doesn’t ask the app before letting the user change its value.

twitch.tv/dancojocar
youtube.com/dancojocar

~~Multiple~~ Sources of Truth

- Let the app be the source.

```
@Composable
fun FoodPreferences(data: UserPreferences) {
  val options = listOf(MILK, EGGS, BREAD)
  Checkbox(
      options = options,
      selected = data.favoriteFood,
      onChange = { selected -> data.favoriteFood = selected }
  )
}
```

twitch.tv/dancojocar
youtube.com/dancojocar

~~Multiple~~ Sources of Truth

- Let the app be the source.

```
@Composable
fun FoodPreferences(data: UserPreferences) {
  val options = listOf(MILK, EGGS, BREAD)
  Checkbox(
      options = options,
      selected = data.favoriteFood,
      onChange = { selected -> data.favoriteFood = selected }
  )
}
```

```
@Composable
fun NamePreferences(data: UserPreferrences) {
  EditableText(
      text = data.name,
      onChange = { newValue -> data.name = newValue }
  )
}
```

twitch.tv/dancojocar
youtube.com/dancojocar

```
@Composable
fun NamePreferences(data: UserPreferrences) {
  EditableText(
      text = data.name,
      onChange = { newValue -> data.name = newValue }
  )
}
```

twitch.tv/dancojocar

youtube.com/dancojocar

DEMO

```
@Composable
fun PhonePreferences(data: UserPreferrences) {
  EditableText(
      text = data.phoneNumber,
      onChange = { newValue ->
        val filtered = filterInvalidPhoneCharacters(newValue)
        val formatted = formatPhone(newValue)
        data.phoneNumber = formatted
      }
  )
}
```

developer.android.com/jetpack/compose

twitch.tv/dancojocar

youtube.com/dancojocar

Introducing SwiftUI

Better apps. Less code.

developer.apple.com/xcode/swiftui

Describe Layout Just Once

twitch.tv/dancojocar
youtube.com/dancojocar

- Declare the content and layout for any state of your view. SwiftUI knows when that state changes, and updates your view's rendering to match.

```
List(landmarks) { landmark in
  HStack {
    Image(landmark.thumbnail)
    Text(landmark.name)
    Spacer()

    if landmark.isFavorite {
      Image(systemName: "star.fill")
        .foregroundColor(.yellow)
    }
  }
}
```

twitch.tv/dancojocar
youtube.com/dancojocar

Build reusable components

- Combine small, single-responsibility views into larger, more complex interfaces. Share your custom views between apps designed for any Apple platform.

```
struct FeatureCard: View {
  var landmark: Landmark

  var body: some View {
    landmark.featureImage
      .resizable()
      .aspectRatio(3/2, contentMode: .fit)
      .overlay(TextOverlay(landmark))
  }
}
```

twitch.tv/dancojocar

youtube.com/dancojocar

Simplify Animations

- Creating smooth animations is as easy as adding a single method call. SwiftUI automatically calculates and animates transitions when needed.

```
VStack {
    Badge()
        .frame(width: 300, height: 300)
        .animation(.easeInOut())
    Text(name)
        .font(.title)
        .animation(.easeInOut())
}
```

twitch.tv/dancojocar

youtube.com/dancojocar

DEMO

Live in Xcode

Landmarks | Iphone Xs

Landmarks | Preview Landmarks: Succeeded | Today at 9:41 AM

Landmarks 〉 Landmarks 〉 Profile 〉 ProfileSummary.swift 〉 body

```
//
//  ProfileSummary.swift
//  Landmarks
//

import SwiftUI

struct ProfileSummary: View {
    var profile: Profile

    static var goalFormat: DateFormatter {
        let formatter = DateFormatter()
        formatter.dateFormat = "MMMM d, yyyy"
        return formatter
    }

    var body: some View {
        List {
            Text(profile.username)
                .bold()
                .font(.title)

            Text("Notifications: \(self.profile.prefersNotifications ? "On": "Off" )")

            Text("Seasonal Photos: \(self.profile.seasonalPhoto.rawValue)")

            Text("Goal Date: \(Self.goalFormat.string(from:self.profile.goalDate))")

            VStack(alignment: .leading) {
                Text("Completed Badges")
                    .font(.headline)
```

Font

Font: Headline

Weight: Inherited

Color: inherited

Alignment

Line Limit: inherited

Padding

Padding: Inherited

```
            HikeRow(hike: hike)
        }
    }
}

struct ProfileSummary_Previews: PreviewProvider {
    static var previews: some View {
```

developer.apple.com/xcode/swiftui

Lecture outcomes

- Learn about Jetpack Compose.
- Build composable components.
- Learn about SwiftUI.