

# 15

## Advanced Jetpack Compose

### Activity 15.01: Adding Animations to the TV Guide App Solution

You can use the *TV Guide* app you developed in *Chapter 14, Architecture Patterns* or make a copy of it. Here is one way you can add animations to the TV Guide app:

1. Open the *TV Guide* project in Android Studio.
2. Open the `DetailsScreen` file. The first animation will be an animation on the height change of the overview text using `animateContentSize`. Add the following at the top of the content block of the outer `Column`:

```
var largeOverview by remember { mutableStateOf(true) }
```

This will be used for changing the size of the overview text.

3. Add the following code to change the number of lines of the overview, depending on the value of `largeOverview`, which will be toggled when the overview text is clicked:

```
Text(
    text = stringResource(
        id = R.string.tv_show_overview,
        overview
    ),
    overflow = TextOverflow.Ellipsis,
    modifier = Modifier
        .animateContentSize()
```

```
        .fillMaxWidth()  
        .height(if (largeOverview) 240.dp else 40.dp)  
        .clickable {  
            largeOverview = largeOverview.not()  
        },  
    ),  
)
```

This will set the height of the overview text to 240.dp when the `largeOverview` is true and to 40.dp when it is false. The `animateContentSize` will add an animation with the change in the height of the overview text.

4. Run the application and click on the overview text multiple times to show change the height of the overview text. Your details screen should look like the following:

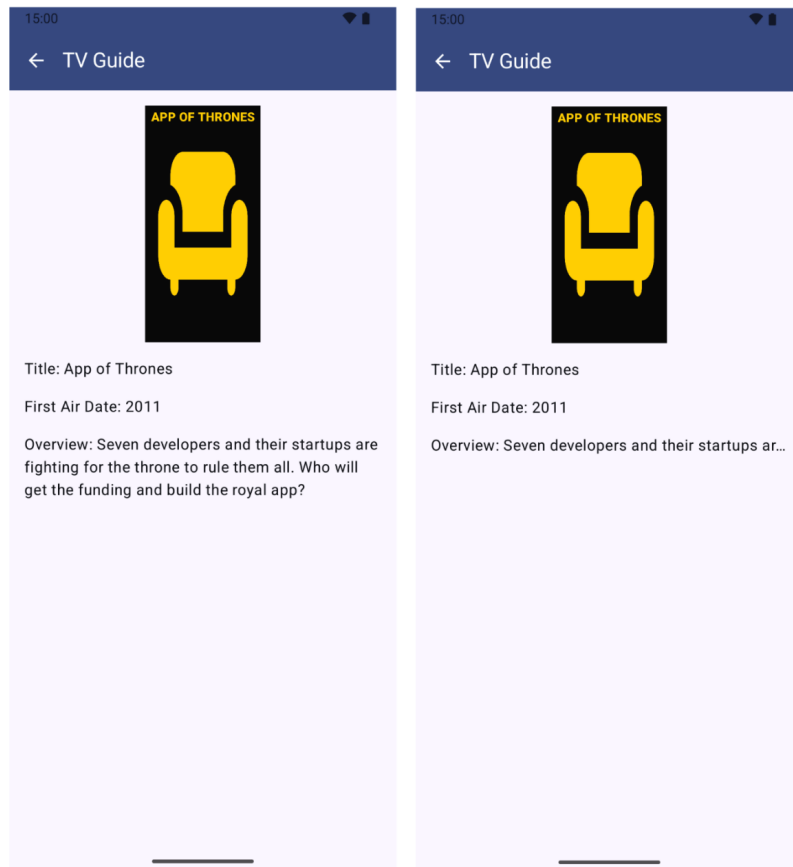


Figure 15.5 – The details screen with the overview text height changes: when `largeOverview` is true (left) and when `largeOverview` is false (right)

5. The second animation will be the showing and hiding of the first air date text when the title text is clicked. Add the following after the initialization of the `largeOverview` in the `DetailsScreen`:

```
var showRelease by remember { mutableStateOf(true) }
```

This will control whether the first air date text is displayed or not.

6. In the title text, add a clickable modifier to toggle the value of `showRelease` when the text is clicked:

```
Text(
    text = stringResource(
        id = R.string.tv_show_title,
        title
    ),
    overflow = TextOverflow.Ellipsis,
    modifier = Modifier
        .fillMaxWidth()
        .clickable {
            showRelease = showRelease.not()
        }
)
```

7. Wrap the release text composable inside `AnimatedVisibility` to add animation when the `showRelease` value is changed:

```
AnimatedVisibility(visible = showRelease) {
    Text(
        text = stringResource(
            id = R.string.tv_show_release,
            release
        ),
        overflow = TextOverflow.Ellipsis,
        modifier = Modifier.fillMaxWidth()
    )
}
```

8. Run the application and click on the title text multiple times to show or hide the first air date text. The details screen should look like the following:

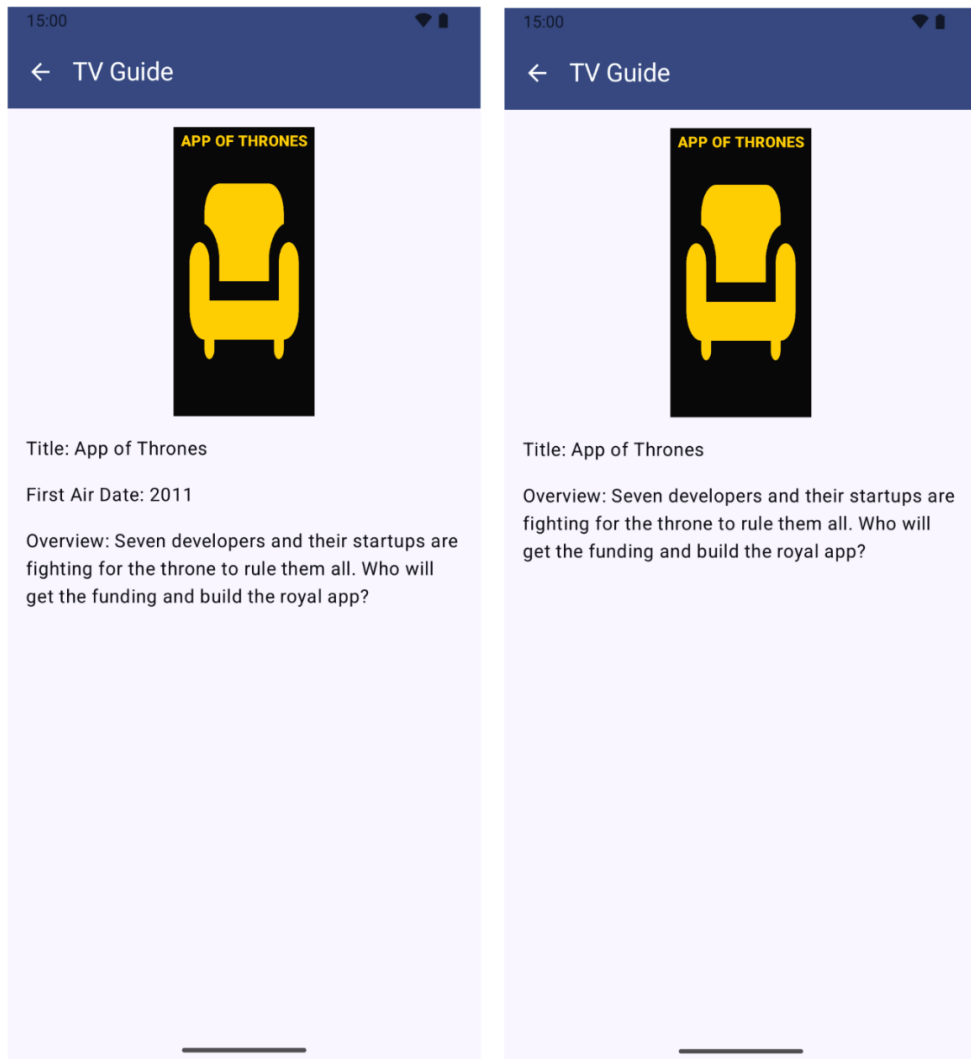


Figure 15.6 – The details screen when the first air date text is displayed (left) and hidden (right)

9. The last animation you will add is adding an animation on value change using `animateFloatAsState`. Add the following in the `DetailsScreen`, after the initialization of `showRelease`:

```
var fullAlpha by remember { mutableStateOf(true) }  
val alpha: Float by animateFloatAsState(  
    if (fullAlpha) 1f else 0.5f,  
    label = "Photo-Alpha"  
)
```

This will change the alpha of the TV show photo from 0.5f to 1f when the `fullAlpha` is changed.

10. Add the following changes to the `AsyncImage` composable for the TV show image:

```
AsyncImage(  
    model = image,  
    contentDescription = stringResource(  
        id = R.string.tv_show_poster  
    ),  
    contentScale = ContentScale.Fit,  
    placeholder = painterResource(  
        id = R.drawable.ic_launcher_foreground  
    ),  
    modifier = Modifier  
        .alpha(alpha)  
        .clickable {  
            fullAlpha = fullAlpha.not()  
        }  
)  
contentDescription = stringResource(  
    id = R.string.tv_show_poster  
)  
,  
contentScale = ContentScale.Fit,  
placeholder = painterResource(  
    id = R.drawable.ic_launcher_foreground  
)  
,  
modifier = Modifier  
    .alpha(alpha)
```

```
.clickable {  
    fullAlpha = fullAlpha.not()  
}
```

This will update the alpha of the photo of the TV show when the photo is clicked.

11. Run the application and click on the photo a few times to change the photo's transparency (alpha). The details screen should look like the following:

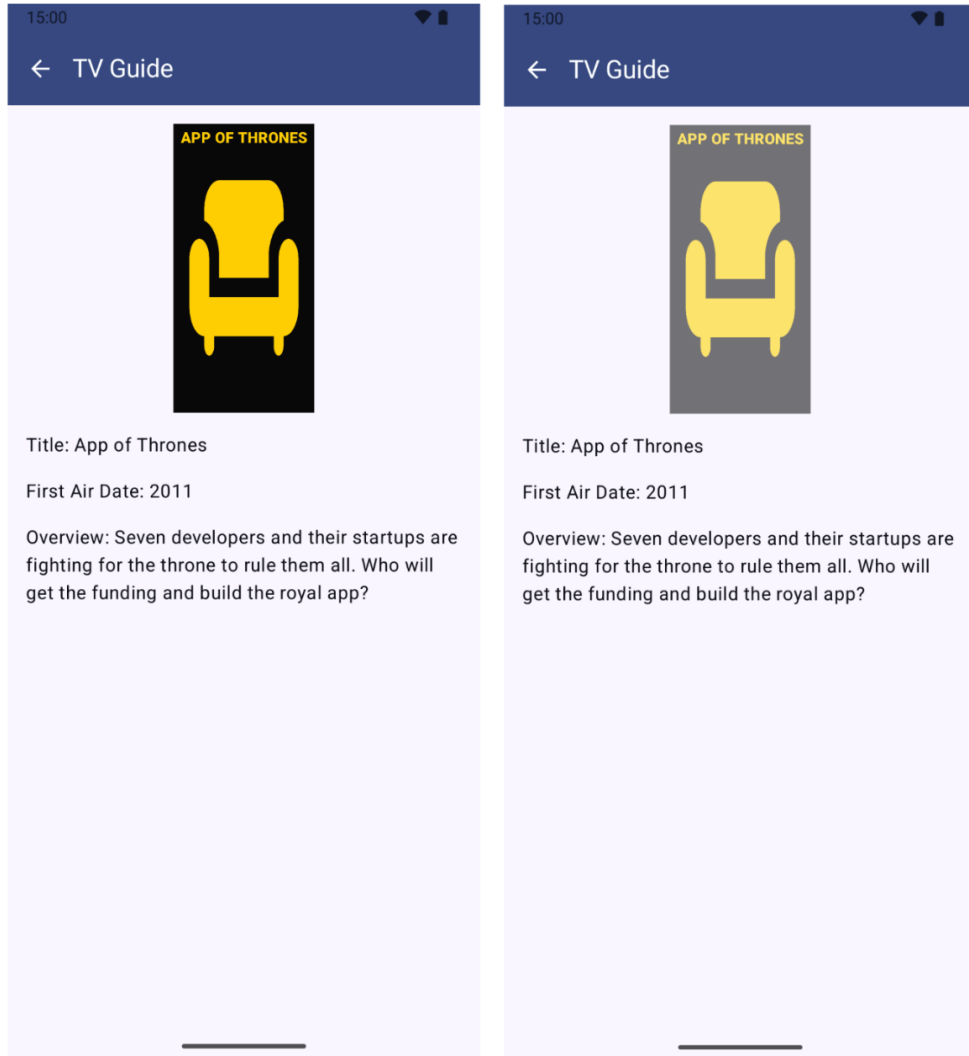


Figure 15.7 – The details screen when the photo alpha is 1f (left) and 0.5 (right)