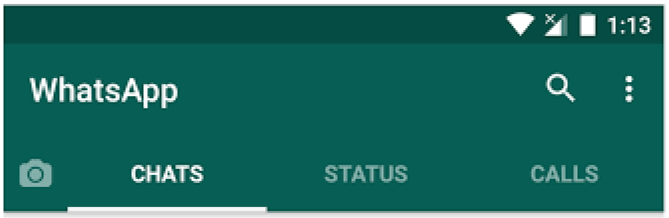
Android - Jetpack Compose Study Jam
Week -1

Welcome to week 1 of the Jetpack Compose Study Jam. This week we'll be learning how to create a Tab Layout and add a Floating Action Button to our project. If you haven't attended the introductory live stream, here's a link for that: *Live Session*. The live stream will introduce you to the basics of Android Studio, Jetpack Compose and Version Control with Git.

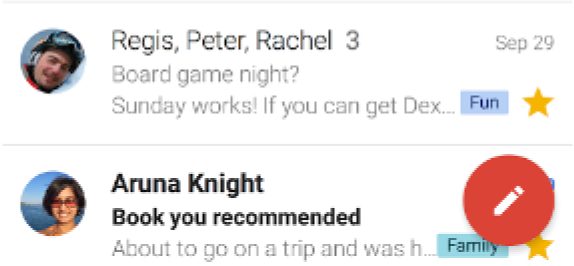
Tab Layout and Floating Action Button:



There are many types of layouts to facilitate app development in jetpack compose and we'll be dealing with one of them, i.e, Tab Layout. It provides a horizontal layout and helps display the tabs. The best example for tab layout is something you see almost everyday, Whatsapp!, where Chats, Status and Calls are called Tabs.

So, how exactly does a tab layout work?
Initially, you set up the tabs along with the information needed like title, icon etc. Then, you have to set up a pager that helps swipe through various pages of the tab. In simple words, if you have n tabs, you would have n pages associated with it and a pager helps you navigate through it.

After you have the tabs and the pager working, you need to add content to each screen in the tab like buttons, text etc. Let us try adding a floating action button to the first screen. Before that, what is a floating action button?

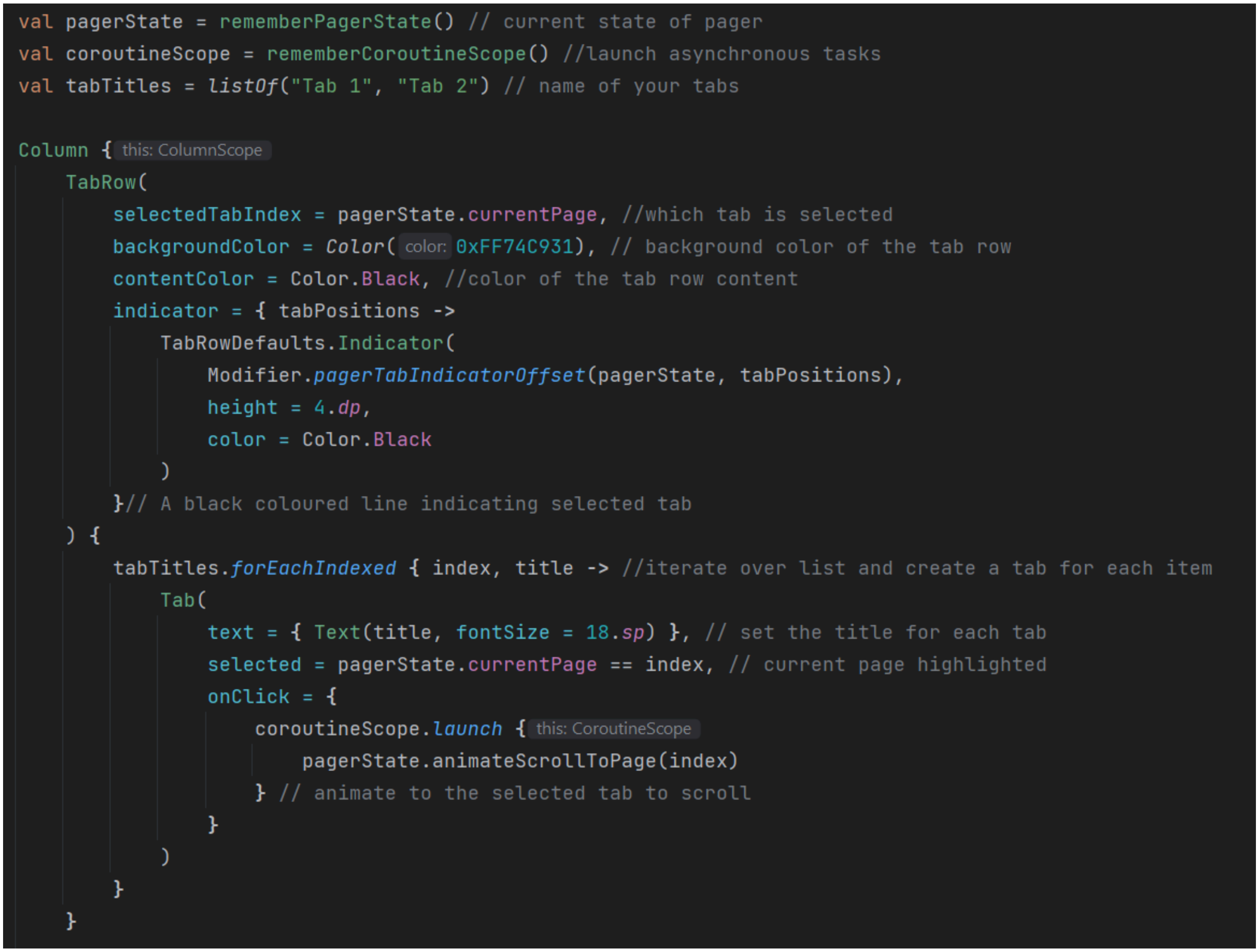


Regis, Peter, Rachel 3 Sep 29
Board game night?
Sunday works! If you can get Dex... Fun

Aruna Knight
Book you recommended
About to go on a trip and was h... Family

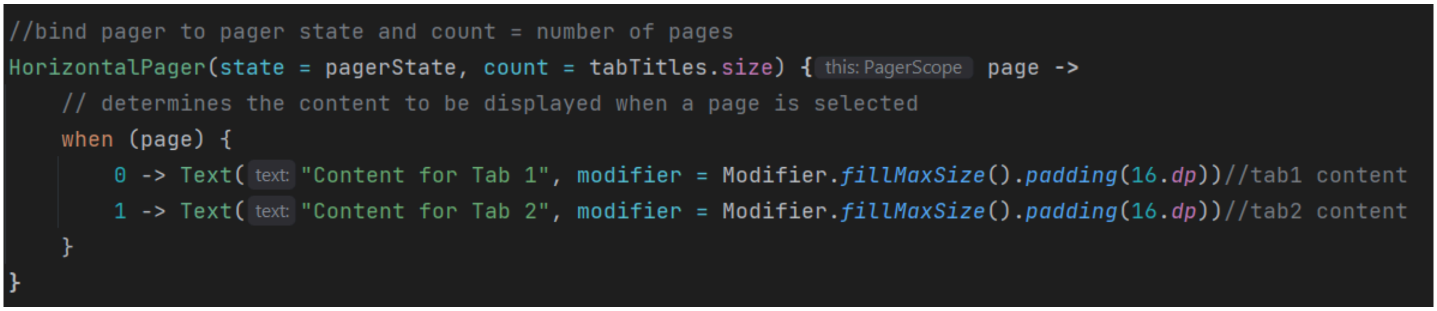
As the name suggests, a Floating Action Button(FAB) is a UI element that floats above the interface and performs the functionalities of a button. It is particularly preferred in many places because of its circular shape and shadow characteristics. An example of this is the compose button in gmail.

Tab Layout



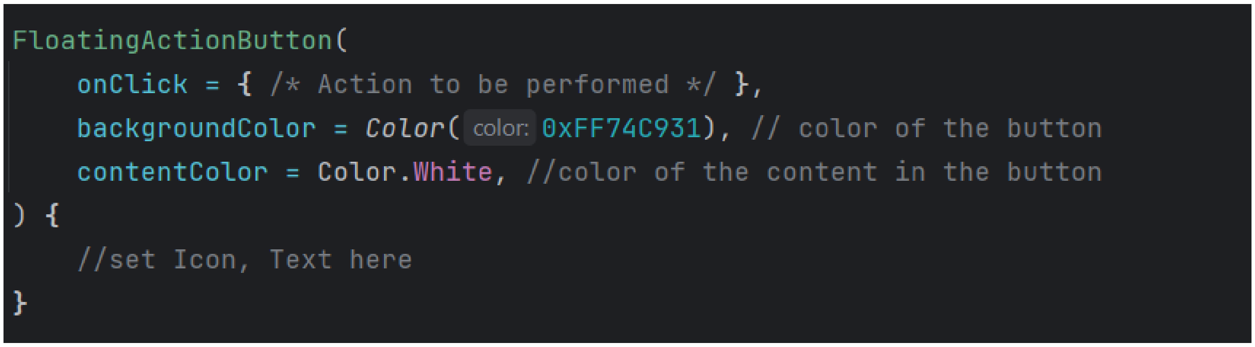
```kotlin
val pagerState = rememberPagerState() // current state of pager
val coroutineScope = rememberCoroutineScope() //launch asynchronous tasks
val tabTitles = listOf("Tab 1", "Tab 2") // name of your tabs

Column { this: ColumnScope
    TabRow(
        selectedTabIndex = pagerState.currentPage, //which tab is selected
        backgroundColor = Color( color: 0xFF74C931), // background color of the tab row
        contentColor = Color.Black, //color of the tab row content
        indicator = { tabPositions ->
            TabRowDefaults.Indicator(
                Modifier.pagerTabIndicatorOffset(pagerState, tabPositions),
                height = 4.dp,
                color = Color.Black
            )
        }// A black coloured line indicating selected tab
    ) {
        tabTitles.forEachIndexed { index, title -> //iterate over list and create a tab for each item
            Tab(
                text = { Text(title, fontSize = 18.sp) }, // set the title for each tab
                selected = pagerState.currentPage == index, // current page highlighted
                onClick = {
                    coroutineScope.launch { this: CoroutineScope
                        pagerState.animateScrollToPage(index)
                    } // animate to the selected tab to scroll
                }
            )
        }
    }
```

Pager



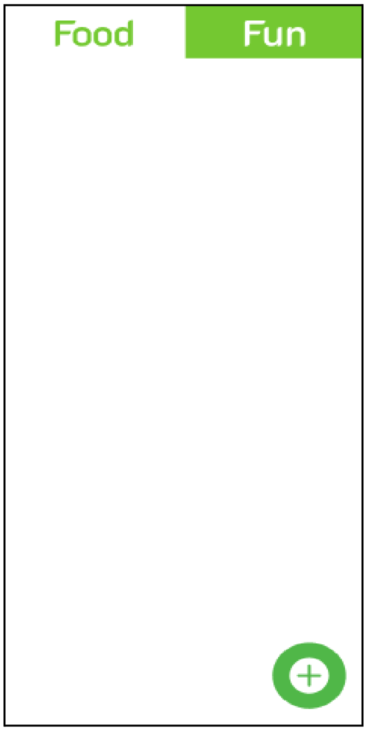
```kotlin
//bind pager to pager state and count = number of pages
HorizontalPager(state = pagerState, count = tabTitles.size) { this: PagerScope  page ->
    // determines the content to be displayed when a page is selected
    when (page) {
        0 -> Text( text: "Content for Tab 1", modifier = Modifier.fillMaxSize().padding(16.dp))//tab1 content
        1 -> Text( text: "Content for Tab 2", modifier = Modifier.fillMaxSize().padding(16.dp))//tab2 content
    }
}
```

Floating Action Button



```
FloatingActionButton(
    onClick = { /* Action to be performed */ },
    backgroundColor = Color( color: 0xFF74C931), // color of the button
    contentColor = Color.White, //color of the content in the button
) {
    //set Icon, Text here
}
```

You can make use of the above code snippets as a reference to build your own tab layout. This is not the exact code but just a snippet for your reference. Some websites are also provided in the end which you can make use of to build the app. The goal for this week is to create an app with the following design as a reference. PS: This is just for your reference and we would like to see your creativity as well. Good Luck!!



Links for reference:

1. https://medium.com/@ecemokan/introduction-to-jetpack-compose-c042e2544db6
2. https://itnext.io/floating-action-button-in-jetpack-compose-with-material-3-10ba8bff415a
3. https://www.freecodecamp.org/news/tabs-in-jetpack-compose/