

```

package com.example.arcgissetup

import android.content.Context
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import com.arcgismaps.mapping.ArcGISMap
import com.arcgismaps.mapping.layers.RasterLayer
import com.arcgismaps.raster.Raster
import com.example.arcgissetup.ui.theme.ArcgisSetupTheme

import com.arcgismaps.mapping.Basemap
import androidx.compose.runtime.*
import com.arcgismaps.toolkit.geoviewcompose.MapView
import java.io.File
import java.io.FileOutputStream
class MainActivity : ComponentActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        // Copy GeoTIFF once
        val tifPath = copyGeoTiffFromAssets(
            context = this,
            fileName = "INDIAN_HILL.tif"
        )

        setContent {
            ArcgisSetupTheme {
                GeoTiffMapScreen(tifPath)
            }
        }
    }
}

fun copyGeoTiffFromAssets(
    context: Context,

```

```

        fileName: String
    ): String {

        val dir = File(context.getExternalFilesDir(null), "INDIAN_HILL")
        if (!dir.exists()) dir.mkdirs()

        val outFile = File(dir, fileName)

        // Copy only once
        if (!outFile.exists()) {
            context.assets.open("maps/$fileName").use { input ->
                FileOutputStream(outFile).use { output ->
                    input.copyTo(output)
                }
            }
        }

        return outFile.absolutePath
    }

```

```

@Composable
fun GeoTiffMapScreen(tifPath: String) {

    // Remember map instance
    val map = remember {
        createOfflineGeoTiffMap(tifPath)
    }

    // ArcGIS Compose MapView
    MapView(
        arcGISMap = map,
        modifier = Modifier.fillMaxSize()
    )
}

fun createOfflineGeoTiffMap(tifPath: String): ArcGISMap {

    // Load GeoTIFF
    // val raster = Raster(tifPath)

```

```
val raster = Raster.createWithPath(tifPath)

// Create raster layer
val rasterLayer = RasterLayer(raster)

// Use raster as BASEMAP
val basemap = Basemap(rasterLayer)

// Create map with offline basemap
return ArcGISMap(basemap)
}
```