# Android Studio 导入Android Studio步骤

## 1、导入

Android Studio中点击左上角File >> New >> import module

选择unity export文件夹中的UnityLibrary文件夹

点击OK

## 修改gradle.properties文件

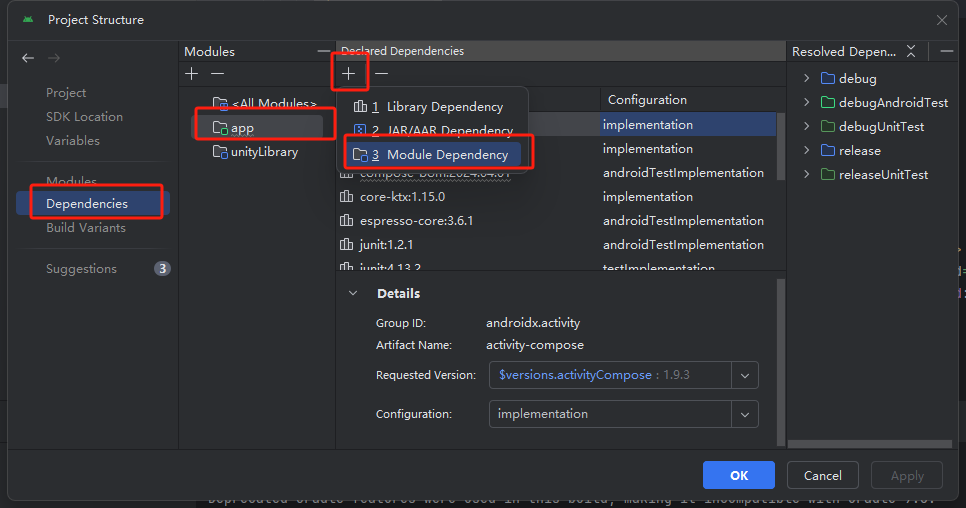
在最后一行添加

unityStreamingAssets=.unity3d

之后点击大象

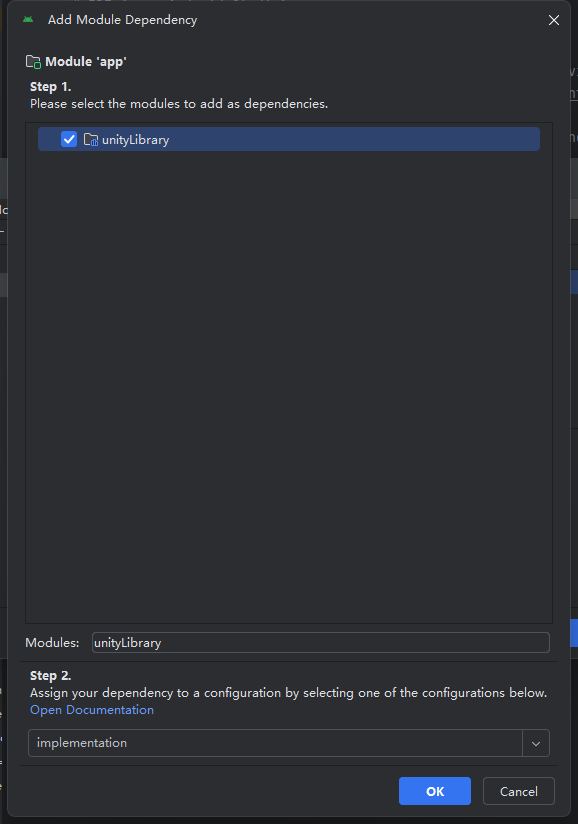
## 点击File >> Project Structural

选择Dependencies >> app >> 左上角的+号 >> 3、Module Deperdency



## 选择unityLibrary

并点击ok



## 修改unityLibrary下的AndroidManifest.xml

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

<category android:name="android.intent.category.LEANBACK\_LAUNCHER" />

</intent-filter>

<meta-data android:name="unityplayer.UnityActivity" android:value="true" />

将上述内容注释或删除

在 <activity> 添加 android:process="e.unity3d"

完成修改的文件如下所示：

<?xml version="1.0" encoding="utf-8"?>

<!--存在以下报错信息：Incorrect package="com.unity3d.player" found in source AndroidManifest.xml: 故需要删除 package="com.unity3d.playe-->

<!--<manifest xmlns:android="http://schemas.android.com/apk/res/android" -->

<!-- package="com.unity3d.player"-->

<!-- xmlns:tools="http://schemas.android.com/tools">-->

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools">

<application android:extractNativeLibs="true" android:isGame="true">

<activity android:name="com.unity3d.player.UnityPlayerActivity"

android:theme="@style/UnityThemeSelector"

android:screenOrientation="fullUser"

android:launchMode="singleTask"

android:configChanges="mcc|mnc|locale|touchscreen|keyboard|keyboardHidden|navigation|orientation|screenLayout|uiMode|screenSize|smallestScreenSize|fontScale|layoutDirection|density"

android:resizeableActivity="false"

android:hardwareAccelerated="false"

android:exported="true"

android:process="e.unity3d"

>

<!-- <intent-filter>-->

<!-- <action android:name="android.intent.action.MAIN" />-->

<!-- <category android:name="android.intent.category.LAUNCHER" />-->

<!-- <category android:name="android.intent.category.LEANBACK\_LAUNCHER" />-->

<!-- </intent-filter>-->

<!-- <meta-data android:name="unityplayer.UnityActivity" android:value="true" />-->

</activity>

<meta-data android:name="unity.splash-mode" android:value="0" />

<meta-data android:name="unity.splash-enable" android:value="True" />

<meta-data android:name="unity.launch-fullscreen" android:value="True" />

<meta-data android:name="unity.allow-resizable-window" android:value="False" />

</application>

<uses-feature android:glEsVersion="0x00020000" />

<uses-permission android:name="android.permission.INTERNET" />

<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />

<uses-feature android:name="android.hardware.touchscreen" android:required="false" />

<uses-feature android:name="android.hardware.touchscreen.multitouch" android:required="false" />

<uses-feature android:name="android.hardware.touchscreen.multitouch.distinct" android:required="false" />

</manifest>

## 修改settings.gradle.kts文件

在 dependencyResolutionManagement 中添加

flatDir {

dirs("${project(":unityLibrary").projectDir}/libs")

}

并在文件中添加

project(":unityLibrary").projectDir = File("path/to/unityLibrary")

将

include(":unityLibrary")

移动至 dependencyResolutionManagement 的上方

修改完成后的文件如下所示：

pluginManagement {

repositories {

google {

content {

includeGroupByRegex("com\\.android.\*")

includeGroupByRegex("com\\.google.\*")

includeGroupByRegex("androidx.\*")

}

}

mavenCentral()

gradlePluginPortal()

}

}

include(":app")

include(":unityLibrary")

dependencyResolutionManagement {

repositoriesMode.set(RepositoriesMode.FAIL\_ON\_PROJECT\_REPOS)

repositories {

google()

mavenCentral()

flatDir {

dirs("${project(":unityLibrary").projectDir}/libs")

}

}

}

// 确保路径正确，指向 unityLibrary 目录

//project(":unityLibrary").projectDir = File("path/to/unityLibrary")

rootProject.name = "UnityAndroidTest"

//include(":unityLibrary")

project(":unityLibrary").projectDir = file("unityLibrary")

## 创建UnityPlayer的子类

unityLibrary >> src >> main >> jave 右键com文件夹创建.com.example.unity

右键example.unity new >> java class

命名为MyUnityPlayer

并将以下内容覆盖MyUnityPlayer中的所有内容

package com.example.unity;

import android.app.Activity;

import android.content.Context;

import com.unity3d.player.UnityPlayer;

import com.unity3d.player.IUnityPlayerLifecycleEvents;

public class MyUnityPlayer extends UnityPlayer {

public MyUnityPlayer(Context context) {

super((Activity) context);

}

public MyUnityPlayer(Context context, IUnityPlayerLifecycleEvents iUnityPlayerLifecycleEvents) {

super((Activity) context, iUnityPlayerLifecycleEvents);

}

// 自定义 kill 方法

public void kill() {

// 自定义清理或关闭 UnityPlayer 的代码

}

}

## 修改UnityPlayerActivity

将以下内容

/\*

Android 退出App

\*/

public void ReturnAPP(){

finish();

}

复制至unityLibrary >> src >> main >> jave >> com >> unity3d.player 中的

Unity PlayerActivity中

## 添加一个按钮，用于启动unity场景

完整代码如下：

package com.example.unityandroidtest

import android.content.Intent

import android.os.Bundle

import androidx.activity.ComponentActivity

import androidx.activity.compose.setContent

import androidx.activity.enableEdgeToEdge

import androidx.compose.foundation.layout.fillMaxSize

import androidx.compose.foundation.layout.padding

import androidx.compose.material3.Scaffold

import androidx.compose.material3.Text

import androidx.compose.material3.TextButton

import androidx.compose.runtime.Composable

import androidx.compose.ui.Modifier

import androidx.compose.ui.platform.LocalContext

import androidx.compose.ui.tooling.preview.Preview

import androidx.compose.ui.unit.dp

import com.example.unityandroidtest.ui.theme.UnityAndroidTestTheme

import com.unity3d.player.UnityPlayerActivity

class MainActivity : ComponentActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

enableEdgeToEdge()

setContent {

UnityAndroidTestTheme {

Scaffold(modifier = Modifier.fillMaxSize()) { innerPadding ->

Greeting(

name = "Android",

modifier = Modifier.padding(innerPadding)

)

}

}

}

}

private fun navigateToUnity() {

val intent = Intent(this, UnityPlayerActivity::class.java)

startActivity(intent)

}

}

@Composable

fun Greeting(name: String, modifier: Modifier = Modifier) {

Text(

text = "Hello $name!",

modifier = modifier

)

val context = LocalContext.current

TextButton(

onClick = {

val intent = Intent(context, UnityPlayerActivity::class.java)

context.startActivity(intent)

},

modifier = modifier.padding(top = 16.dp)

) {

Text("Go to Unity")

}

}

@Preview(showBackground = true)

@Composable

fun GreetingPreview() {

UnityAndroidTestTheme {

Greeting("Android")

}

}