

TASK GUIDE (B3.01)

A. Objectives.

Student will start an Android project to create an Android application that contains several activities entitled Soccer Match. First with project configuration and resource configuration.

B. Requirements.

Hardware:

- 2 GB RAM minimum, 8 GB RAM recommended
- 2 GB of available disk space minimum, 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution
- Intel processor with support for Intel VT-x, Intel EM64T (Intel 64), and Execute Disable (XD) Bit functionality

Software:

- Microsoft Windows 7/8/10 (32-bit or 64-bit)
- JDK 8
- Android Studio IDE (Minimum 3.2) with AndroidX library.

C. Resources.

Documents:

- Guide

Supplement files:

- ResourceTest.java
- ViewTest.java
- icon_add_button.png
- icon_goal.png
- icon_next.png
- icon_player.png
- icon_red_card.png
- icon_reload.png
- icon_yellow_card.png
- socceranim.gif
- teamlogo.png

Test code:

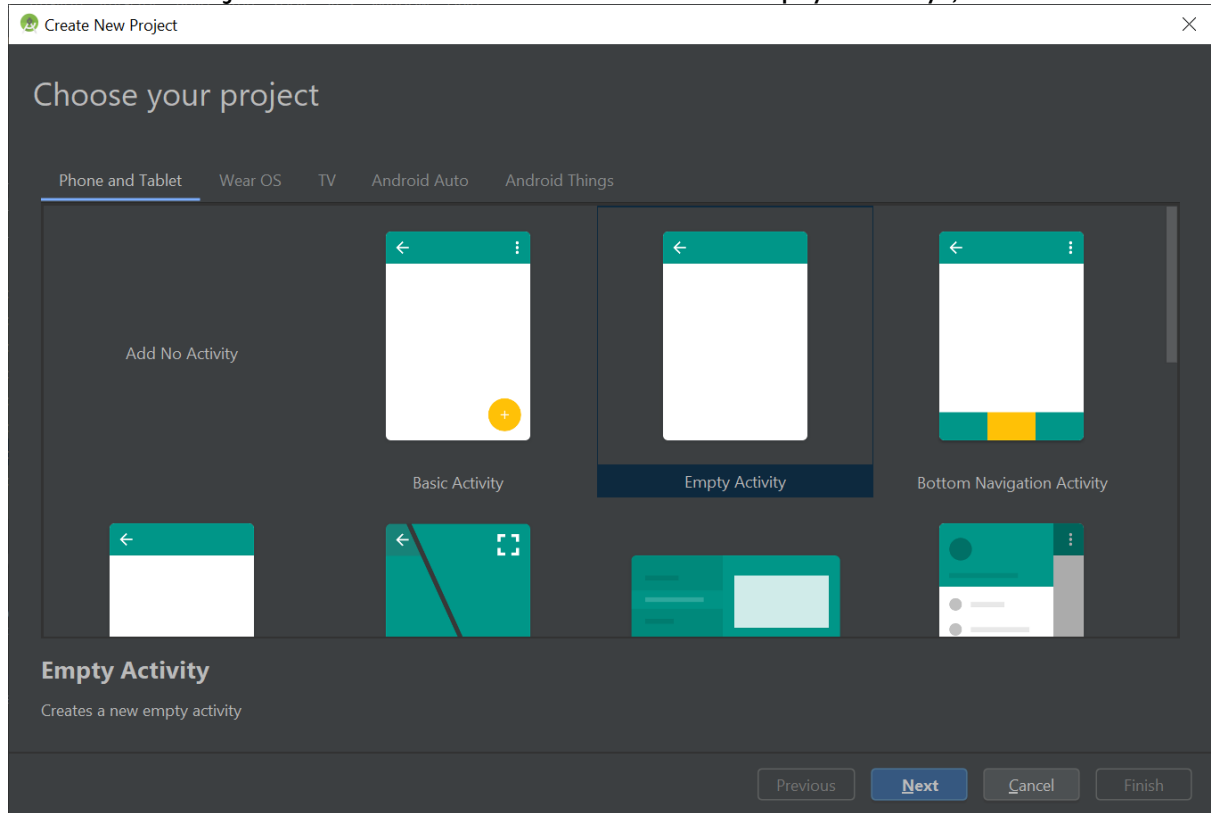
- TestB3MultiActivities011.java

D. Task Description.

Student start to create a project configuration and resource configuration.

E. Specification.

1. Create New Project on Android Studio and choose “Empty Activity”, then Next.



2. Project Configuration.

- Project Name: SoccerMatch
- Package Name: org.aplas.soccermatch
- Project Location: any
- Language: Java
- Target API Level: Android 6.0 (Marshmallow)
- Support instant apps: uncheck
- AndroidX: check

Then Click Finish.

RESOURCE CONFIGURATION

3. Configure the necessary string resource(s)

template	name	value
<code><string name="name">value</string></code>	app_name	Soccer Match

Note: you can add other string resource definitions by yourself if necessary.

4. Configure the necessary drawable resource(s)

Copy all pictures in supplement folder to drawable resource folder in Android Studio:

- icon_add_button.png
- icon_goal.png
- icon_next.png
- icon_player.png
- icon_red_card.png
- icon_reload.png
- icon_yellow_card.png
- socceranim.gif
- teamlogo.png

Note: you can add other image/drawable resource definitions by yourself if necessary.

5. Configure the necessary color and style resource(s)

Note: In this topic, you can modify the color and style of your application freely. You can apply your own taste of style/color combination for your own created application. Show your taste and be yourself !!

6. Change the content of “build.gradle (Module: app)” file like below, then **Sync it**.

```
apply plugin: 'com.android.application'

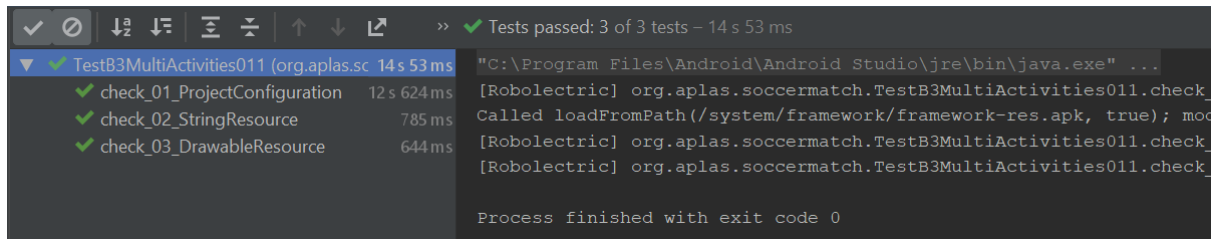
android {
    compileSdkVersion 28
    buildToolsVersion "28.0.3"
    testOptions.unitTests.includeAndroidResources = true

    defaultConfig {
        applicationId "org.aplas.soccermatch"
        minSdkVersion 23
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
        buildConfigField "int", "MIN_SDK_VERSION", "$minSdkVersion.apiLevel"
        buildConfigField "int", "TARGET_SDK_VERSION", "$targetSdkVersion.apiLevel"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'),
'proguard-rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'androidx.appcompat:appcompat:1.1.0'
    implementation 'androidx.recyclerview:recyclerview:1.1.0'
    implementation 'androidx.cardview:cardview:1.0.0'
    def fragment_version = "1.2.3"
    implementation "androidx.fragment:fragment:$fragment_version"
    implementation "androidx.fragment:fragment-testing:$fragment_version"
    testImplementation "androidx.fragment:fragment-testing:$fragment_version"
    testImplementation "org.robolectric:robolectric:4.2.1"
    testImplementation 'junit:junit:4.12'
    debugImplementation 'androidx.test:core:1.2.0'
}
```

F. Testing.

1. Copy “ResourceTest.java”, “ViewTest.java”, “TestB3MultiActivities011.java” file to “org.aplas.soccermatch (test)” folder.
2. Right click on the “TestB3MultiActivities011.java” file then choose Run. It may take long time to execute.
3. Get the result of your task. If passed you will get green check like below. If the test failed, you will get orange check get the messages and you must check your work again.



You have to try until get all green checks and continue to the next task.