CPE 121 Mobile Application Developments

บทที่ 13

การพัฒนาแอพพลิเคชั่นบนมือถือ
(Mobile Application Development)

Faculty of Engineering
SRINAKHARINWIROT UNIVERSITY

Pramual Choorat, Ph.D.
Department of Computer Engineering

Mobile Application Development

แอนดรอยด์ (Android) เป็นซอฟต์แวร์ที่มีโครงสร้างแบบสแต็ก (Stack) หรือแบบเรียงทับซ้อนกัน ซึ่งเป็นการรวมเอาระบบปฏิบัติการมิดเดิลแวร์และแอพพลิเคชั่นที่สำคัญเข้าไว้ด้วยกัน สำหรับทำงานบนอุปกรณ์เคลื่อนที่แบบพกพา โดยเฉพาะโทรศัพท์มือถือ หรือ Tablet เป็นต้น

โดยการทำงานของแอนดรอยด์นั้นจะมีพื้นฐานอยู่บนระบบลีนุกซ์ เคอร์เนล (Linux Kernel) ซึ่งใช้ Android SDK (Software Development Kit) เป็นเครื่องมือสำหรับการพัฒนาแอพพลิเคชั่นบนระบบปฏิบัติการแอนดรอยด์ โดยใช้ภาษา Java ในการพัฒนา

Application Component

ส่วนประกอบของแอพพลิเคชั่น (Application Component) เป็นแอพพลิเคชั่นที่มีคุณลักษณะการทำงานที่สามารถใช่เป็นส่วนประกอบของแอพพลิเคชั่นอื่นๆได้ โดยสามารถนำแอพพลิเคชั่นส่วนอื่นๆที่ได้มีผู้พัฒนาไว้แล้วมาเรียกใช้และนำมาพัฒนาต่อได้โดยไม่ต้องทำการพัฒนาขึ้นมาเอง เพื่อทำให้เกิดความสะดวกและรวดเร็วในการพัฒนาแอพพลิเคชั่น ส่วนประกอบของแอพพลิเคชั่นของแอนดรอยด์สามารถแบ่งออกได้เป็น 4 ส่วนคือ

1. Activity คือ หน้าจอที่ติดต่อกับผู้ใช้ โดยในแต่ละแอพพลิเคชั่นอาจจะมีมากกว่า 1 หน้าจอ หรือ 1 Activity ซึ่งแต่ละ Activity จะมีหน้าที่ในการเก็บสถานะการใช้งานในส่วนต่างๆ

2. Service คือ งานหรือบริการต่างๆ ที่ทำงานอยู่เบื้องหลัง เช่น Service ที่เปิดไฟล์รูปภาพอยู่ ขณะที่ผู้ใช้งานทำงานอื่นๆหรือใช้แอพพลิเคชั่นอื่นๆ ไปพร้อมกันในเวลาเดียวกัน

Application Component

3. Broadcast and Intent Receiver คือ การตอบสนองซึ่งโดยปกติแล้ว Broadcast Receiver จะเป็นการตอบสนองต่อการเกิดอีเวนต์ของระบบในวงกว้าง เช่น การแจ้งเตือนระดับแบตเตอร์รี่ในเครื่อง เป็นต้น นอกจากนี้ Intent Receiver ยังเป็นส่วนทำให้แอพพลิเคชั่นอื่นๆ เข้าถึงการทำงานของ Activity และ Service ซึ่งในการปฏิบัติงานแต่ละอย่างเป็นการตอบสนองการร้องขอข้อมูลหรือบริการของ Activity อื่นๆ

4. Content Provider คือ ส่วนของการให้บริการส่งข้อมูลสำหรับแต่ละแอพพลิเคชั่น โดยที่ข้อมูลสามารถเก็บอยู่ในรูปแบบของไฟล์หรือฐานข้อมูลก็ได้

Application Life Cycle

โดยทั่วไปแอพพลิเคชั่นจะทำงานแยกกันในแต่ละโพรเซสและในแต่ละโพรเซสก็อาจจะมี Activity/Service ที่ทำงานมากว่า 1 Activity/Service ดังนั้น ในแต่ละแอพพลิเคชั่นอาจจะมีมากกว่า 1 Activity

การทำงานของ Activity นั้น จะเริ่มด้วย startActivity() สำหรับแบบซิงโครนัส (Synchronous) และจะเริ่มด้วย starSubtActivity() สำหรับแบบอะซิงโครนัส (Asynchronous) โดยในแต่ละ Activity นั้น จะมีการทำงานหรือวงรอบชีวิต (Life Cycle) ที่แยกออกจากกันโดยชัดเจน โดยจะมีสถานะการทำงานหลักดังนี้

- **onCreate** (Bundle savedInstanceState) ส่วนนี้จะถูกเรียกใช้งานเมื่อเริ่มการทำงานของโปรแกรม ในกรณีที่มีการเรียกใช้งานเมธอด (Method) นี้ Android Framework จะนำ Bundle object ไปบันทึกไว้ที่ Activity ก่อนที่ Activity จะทำงาน หลังจากนั้นจึงจะตามด้วยเมธอด onStart()

- **onStart()** ส่วนนี้จะเป็นการระบุว่า Activity นั้นๆ จะถูกแสดงขึ้นมาหลังจากนั้นสถานะจะถูกย้ายไปยังสถานะ onResume() แต่ถ้า Activity นั้นไม่สามารถทำงานได้ด้วยเหตุผลใดๆก็ตาม สถานะจูกย้ายไปเป็นสถานะ onStop()

- **onRestart()** ส่วนนี้จะเป็นการระบุว่า Activity นั้น จะถูกแสดงขึ้นมาอีกครั้ง ซึ่งจะตามด้วยสถานะ onStart()

Application Life Cycle

- **onResume()** ส่วนนี้จะเป็นการระบุว่า Activity นั้นๆ กำลังมีการติดต่อปฏิสัมพันธ์กับผู้ใช้งาน เช่น นักพัฒนาต้องการ Activity นั้นให้ขึ้นมาทำงานอีกรอบหนึ่ง หลังจาก Activity นั้นอยู่ในสถานะ onPause()

- **onPause()** ส่วนนี้จะถูกเรียกใช้เมื่อ Activity นั้นๆ จะถูกเปลี่ยนไปเป็นการทำงานทางเบื้องหลัง (Background)

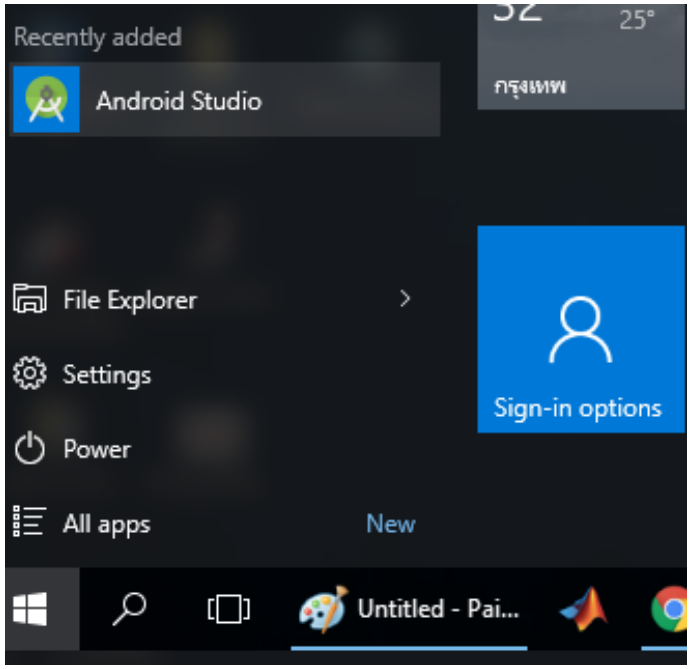
- **onStop()** ส่วนนี้จะถูกเรียกใช้เมื่อผู้ใช้ไม่ต้องใช้งาน Activity นั้นๆ ในช่วงระยะเวลาหนึ่งๆ ซึ่งจะตามด้วยสถานะ onRestart() เมื่อต้องการกลับมาทำ Activity นั้นอีกครั้งหนึ่งหรือตามด้วยสถานะ onDetroy() เมื่อต้องการปิด Activity นั้นๆ

- **onDetroy()** ส่วนนี้จะถูกเรียกเมื่อมีการปิดการทำงานของแต่ละ Activity

การเริ่มสร้าง Project

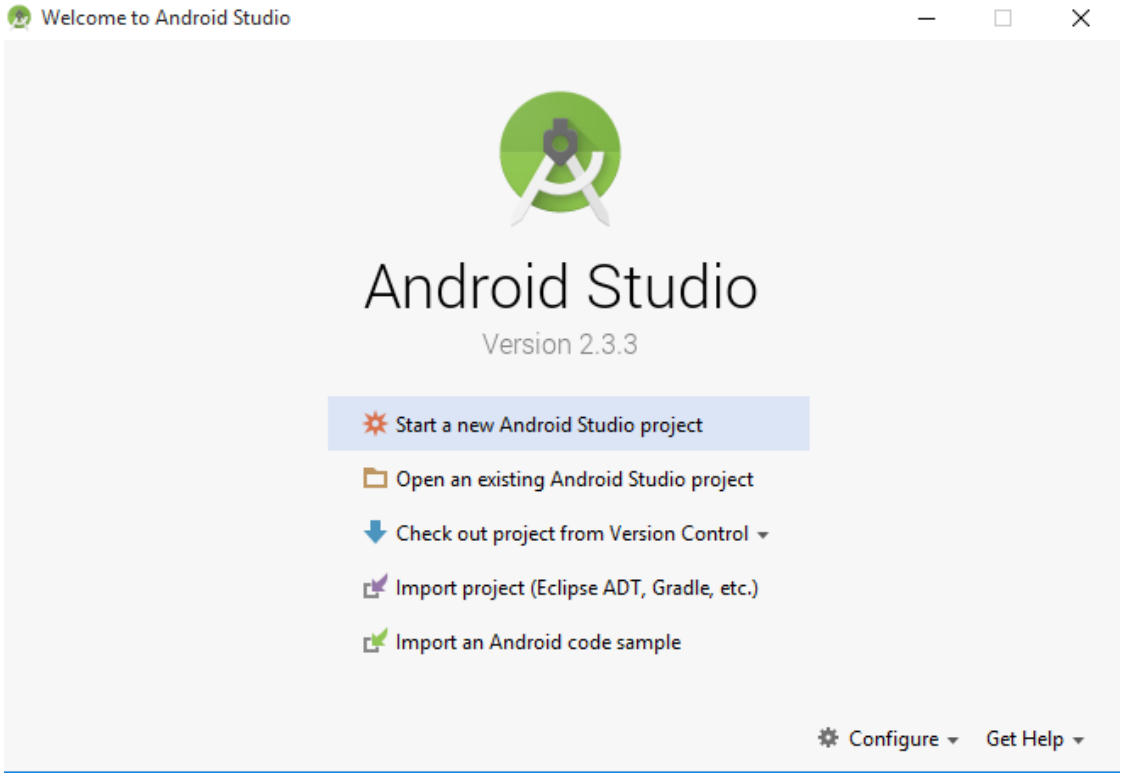
เราจะเริ่มทำการสร้าง Project อย่างง่ายเพื่อใช้ในการทดสอบการทำงานของโปรแกรม Android Studio หลังจากทำการติดตั้งโปรแกรมเรียบร้อยแล้ว โดยขั้นตอนมีดังนี้

1. คลิกปุ่มไอคอน Start ของ Windows แล้วทำการดับเบิ้ลคลิกที่ไอคอนของโปรแกรม Android Studio



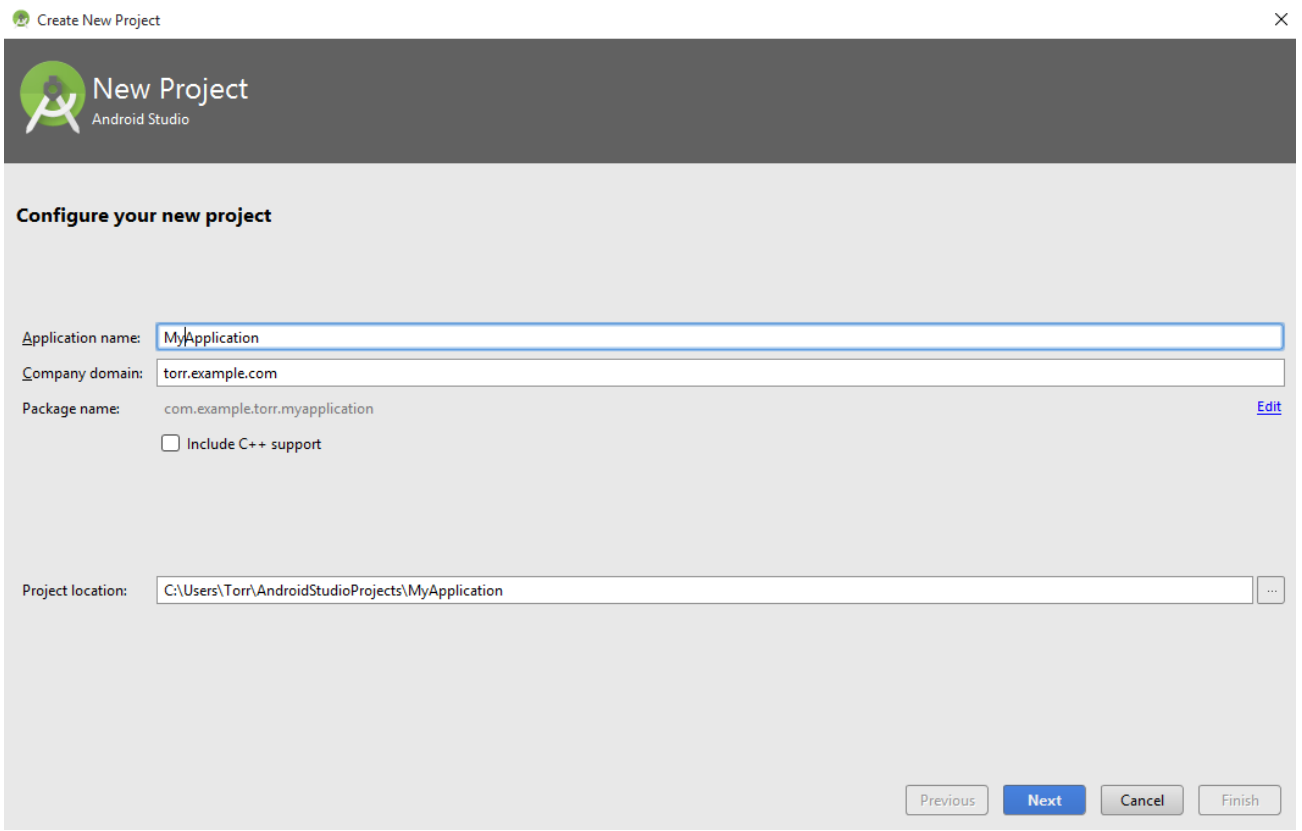
การเริ่มสร้าง Project

2. ทำการคลิกเลือก Start a new Android Studio Project ที่หน้าต่างหน้าแรกของโปรแกรม Android Studio



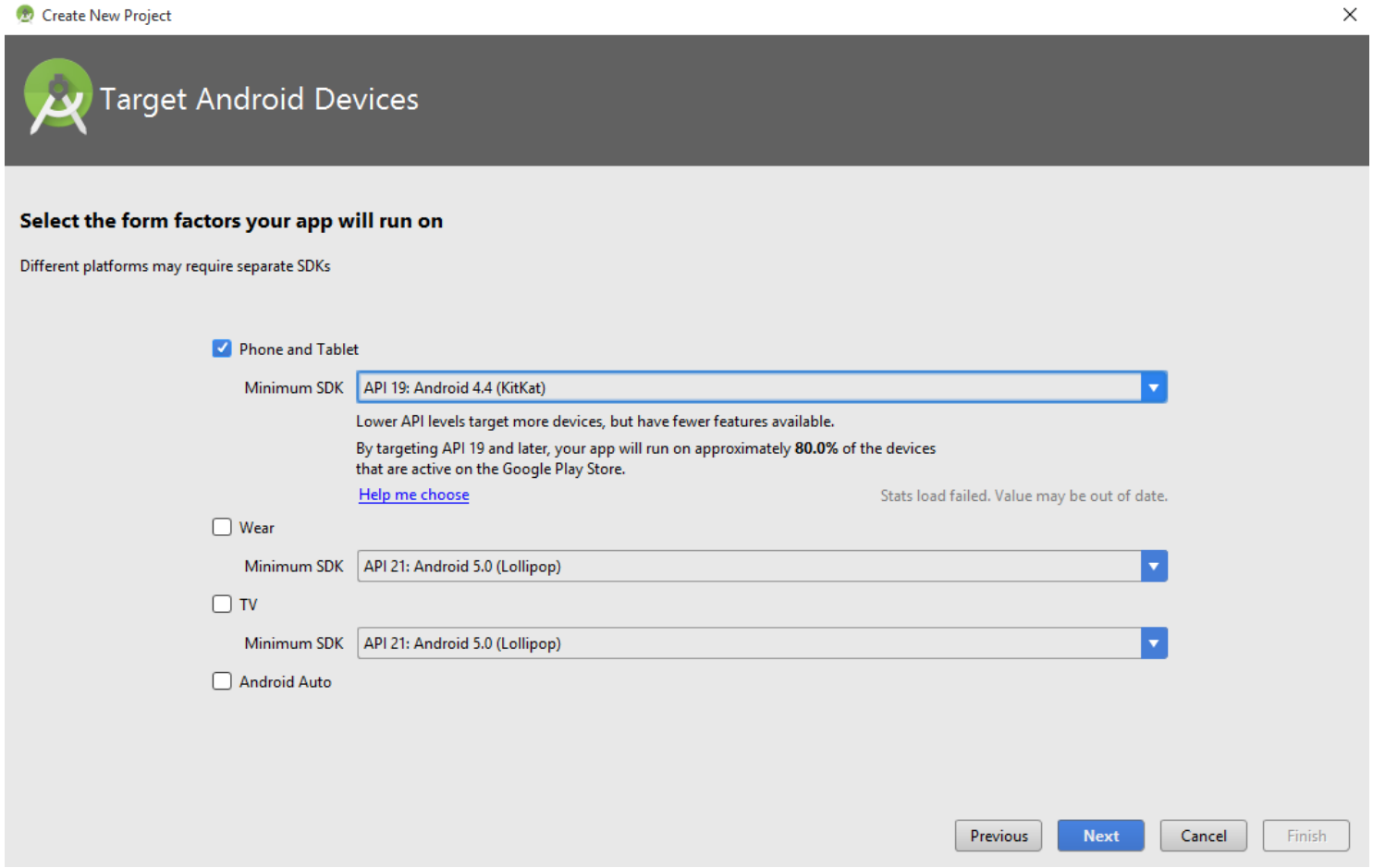
การเริ่มสร้าง Project

3. หลังจากนั้นหน้าต่างสำหรับทำการกำหนดรายละเอียดของโปรเจ็คต์จะปรากฏขึ้น ทำการตั้งชื่อและทำการกำหนดตำแหน่งในการเก็บไฟล์โปรเจ็คต์ กรณีที่เราใช้ตำแหน่งในการเก็บไฟล์ที่โปรแกรมกำหนดมาให้ ให้ทำคลิกปุ่ม Next ได้เลย



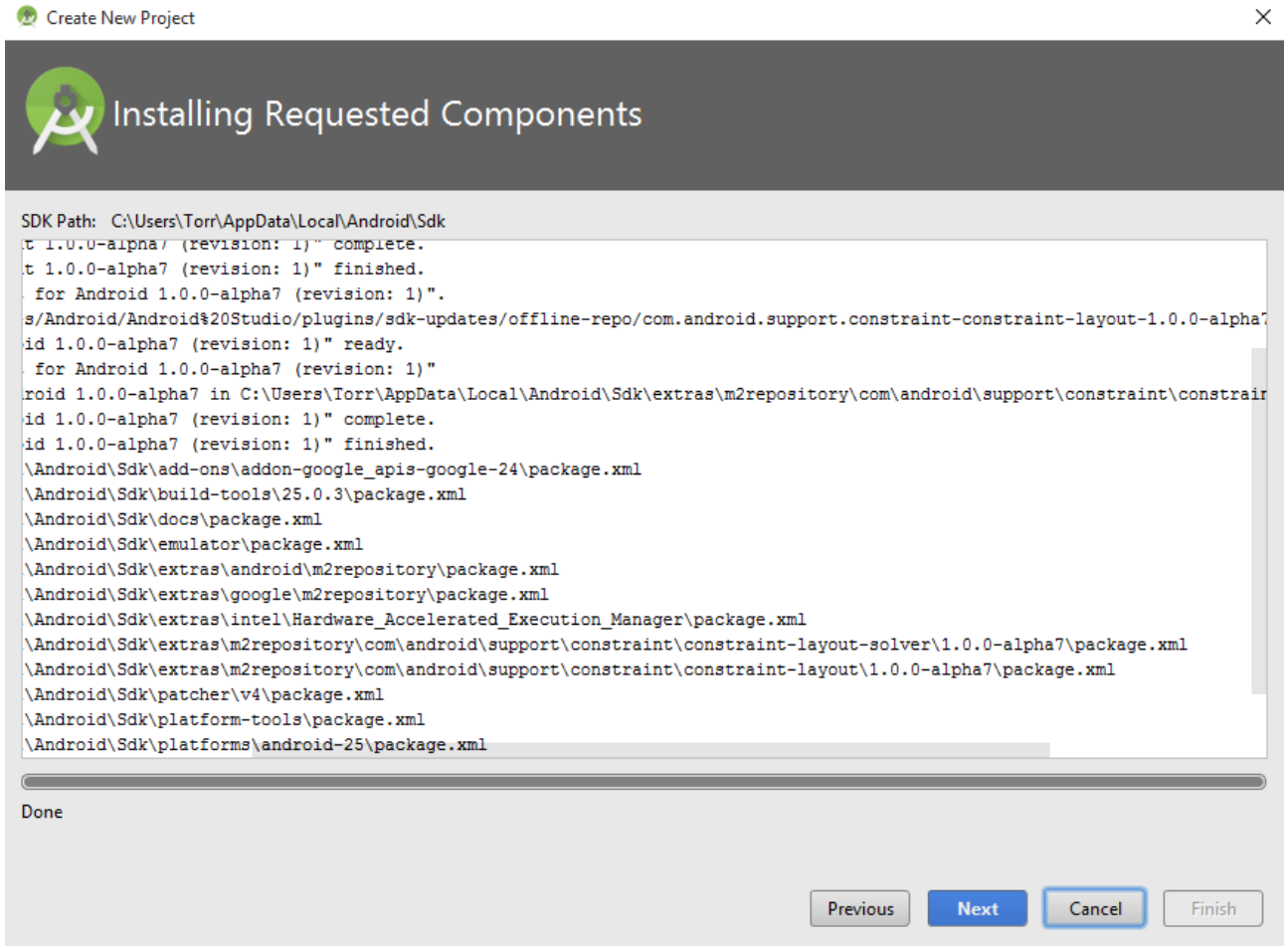
การเริ่มสร้าง Project

4. ทำการตั้งค่า minimum SDK สำหรับจะใช้ในการเขียนโปรแกรมในโปรเจ็คต์ กรณีนี้เราจะทำการเลือก API 19: Android 4.4 (Kitkat) แล้วคลิกปุ่ม Next



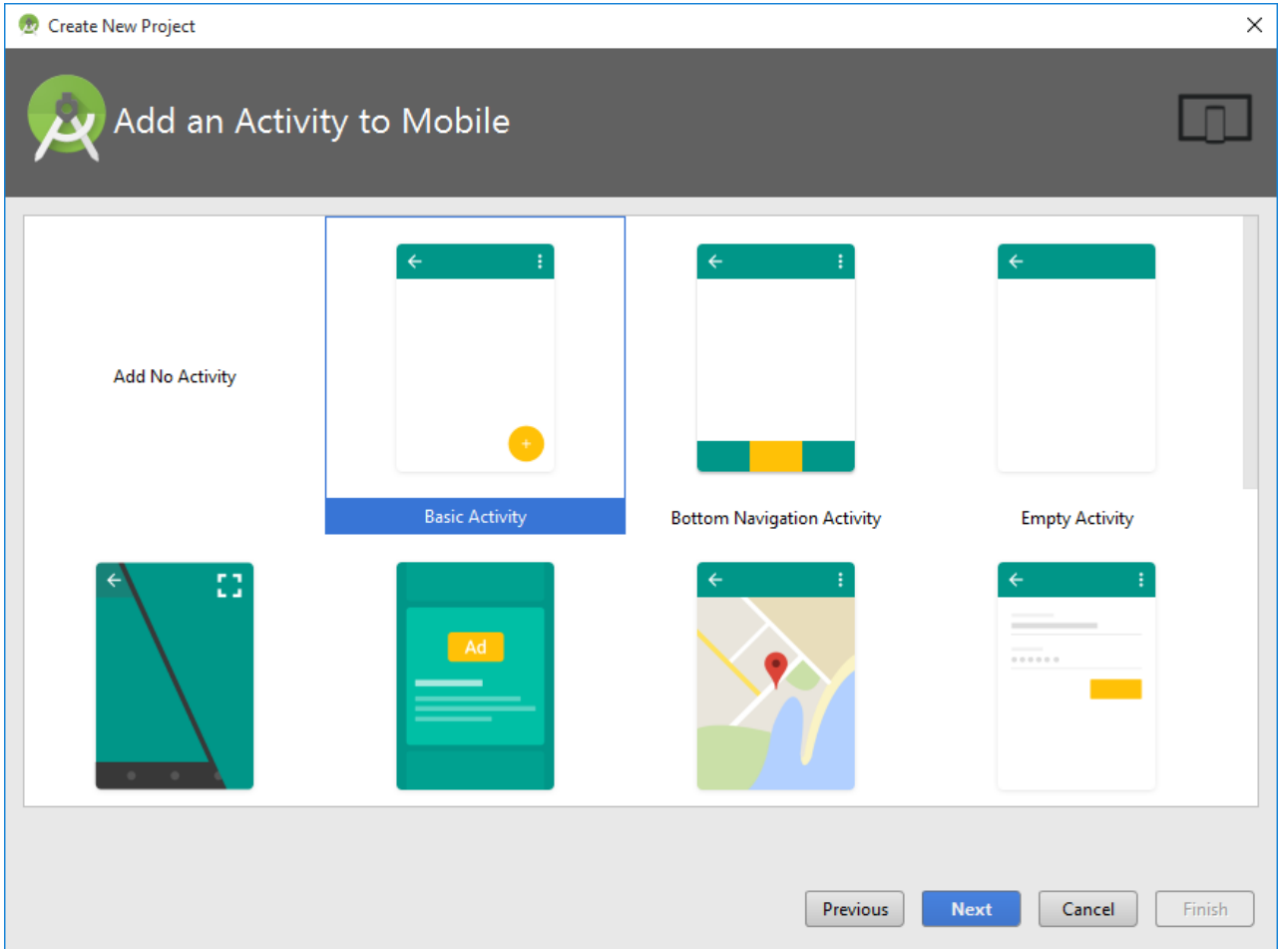
การเริ่มสร้าง Project

5. โปรแกรมจะทำการติดตั้ง SDK จาก API ที่เราได้ทำการเลือกไว้ ให้คลิกปุ่ม Next



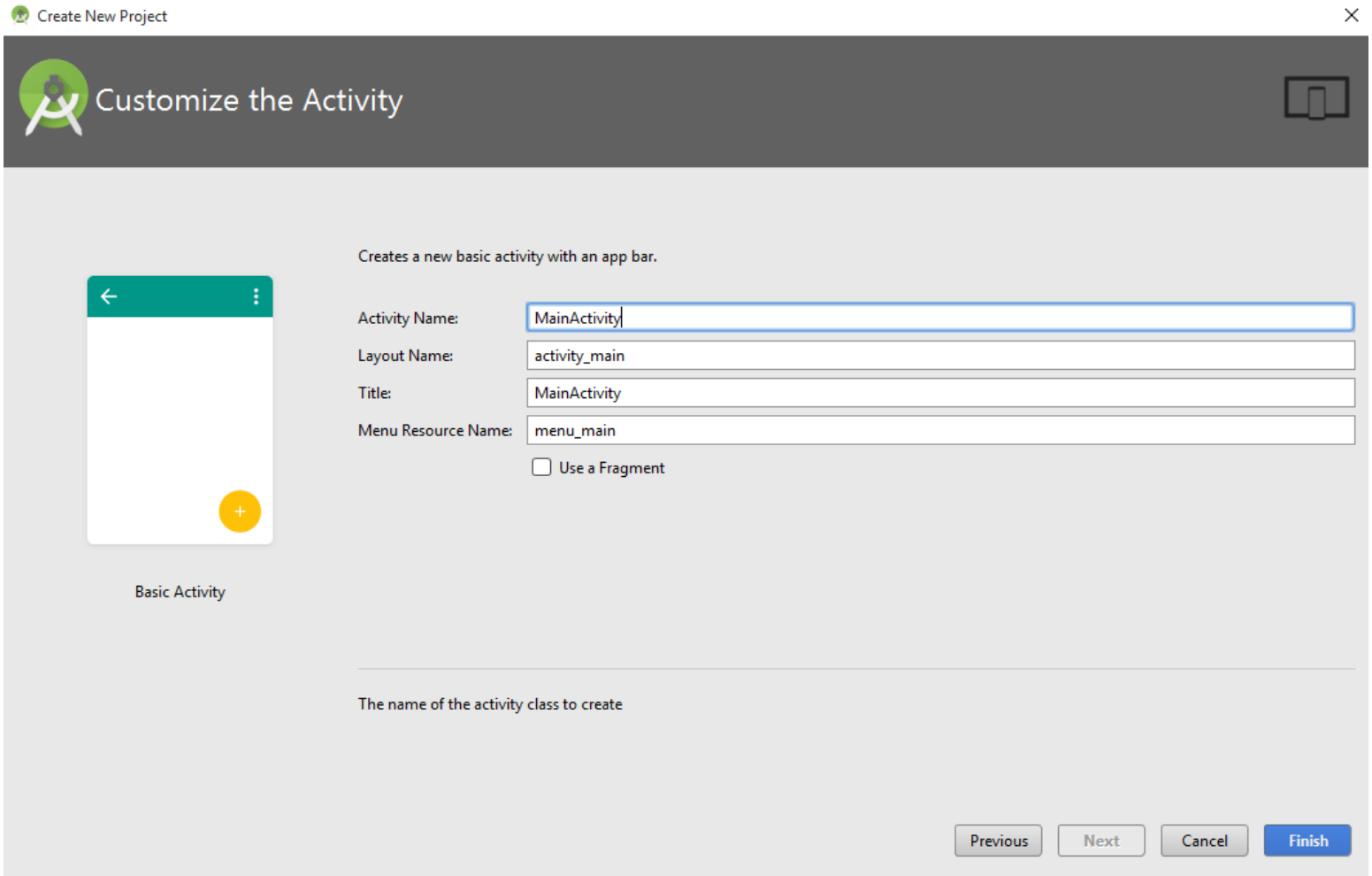
การเริ่มสร้าง Project

6. ทำการเลือก Basic Activity จากหน้าต่างที่ปรากฏ แล้วคลิกปุ่ม Next



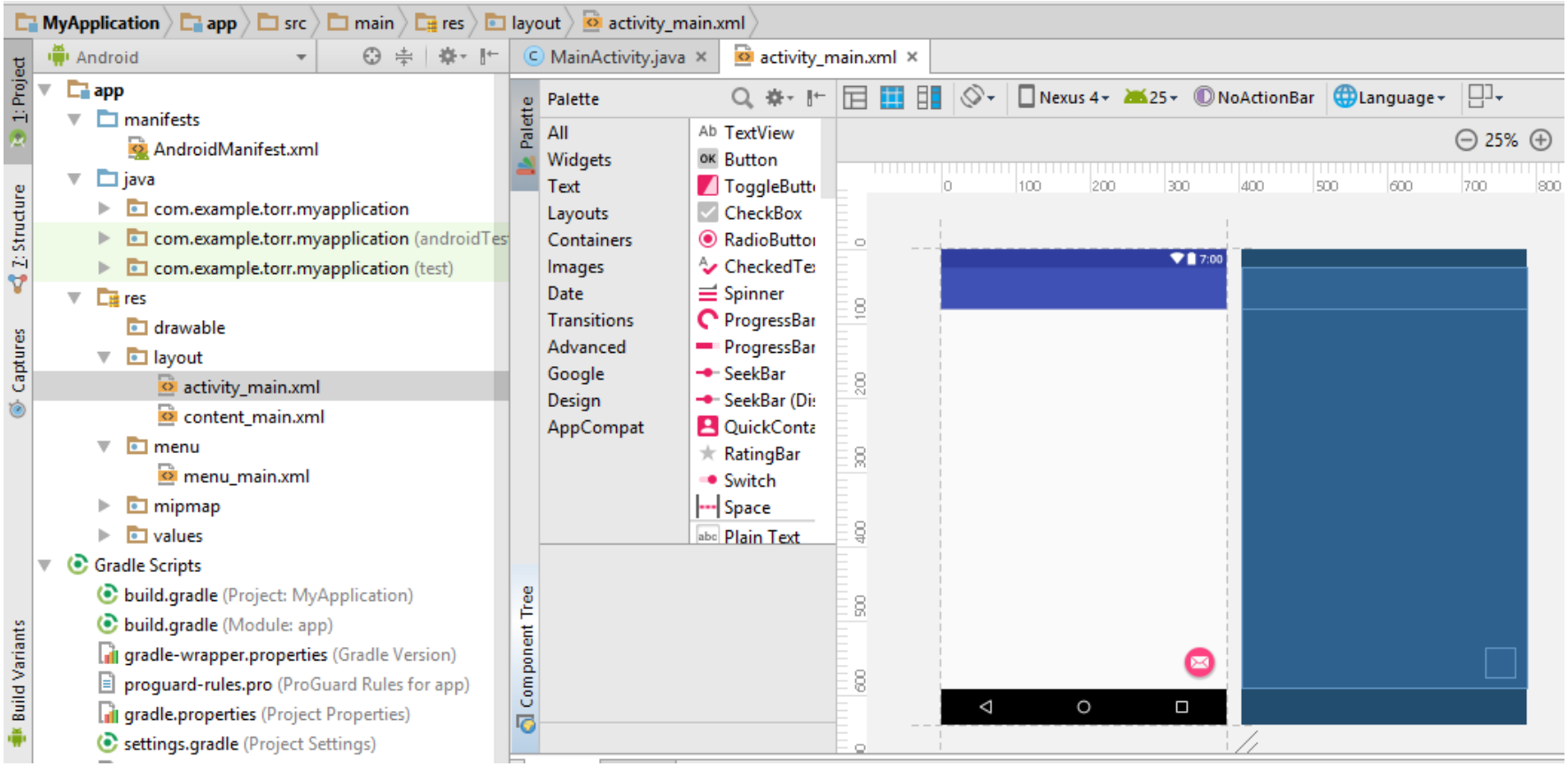
การเริ่มสร้าง Project

7. ทำการกำหนดค่า Option ต่างๆของโปรเจ็กต์ ค่าชื่อ Activity Name, Layout Name, Title และ Menu Resource Name เมื่อกำหนดค่าเรียบร้อยแล้วทำการคลิกปุ่ม Finish



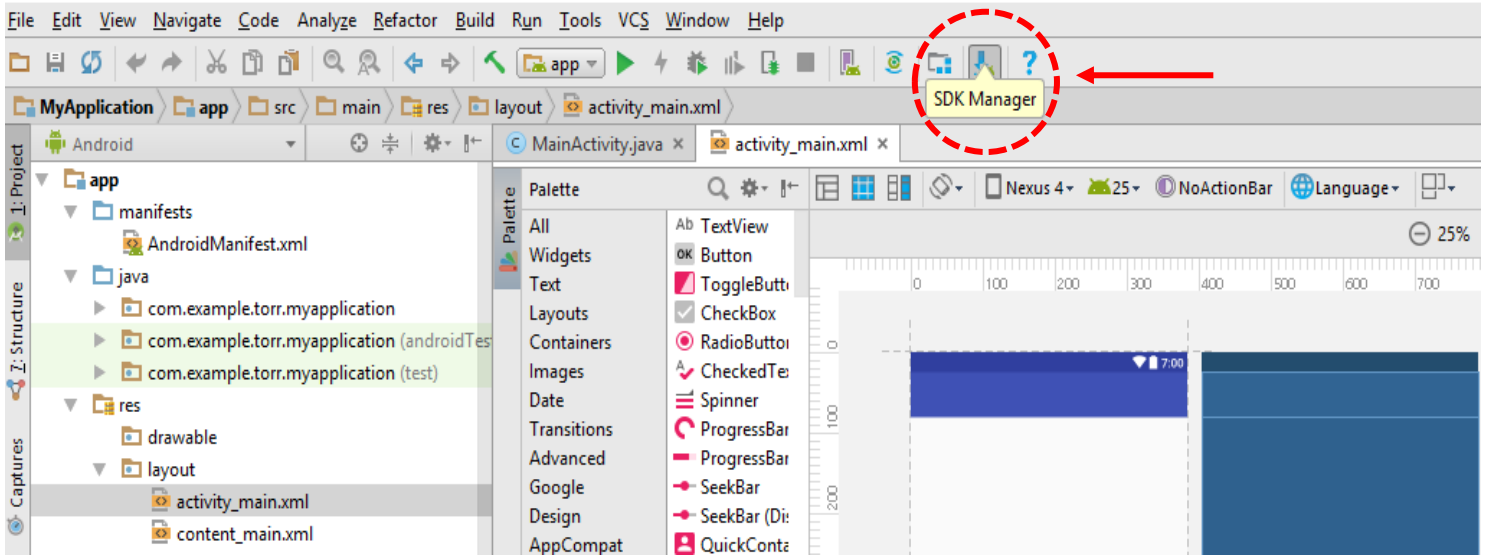
การเริ่มสร้าง Project

8. หน้าโปรแกรมสำหรับแสดงหน้าจอที่ใช้สำหรับการเขียนโปรแกรมในโปรเจ็คต์และหน้าตาเริ่มต้นของแอพพลิเคชั่นก็จะแสดงขึ้นมา



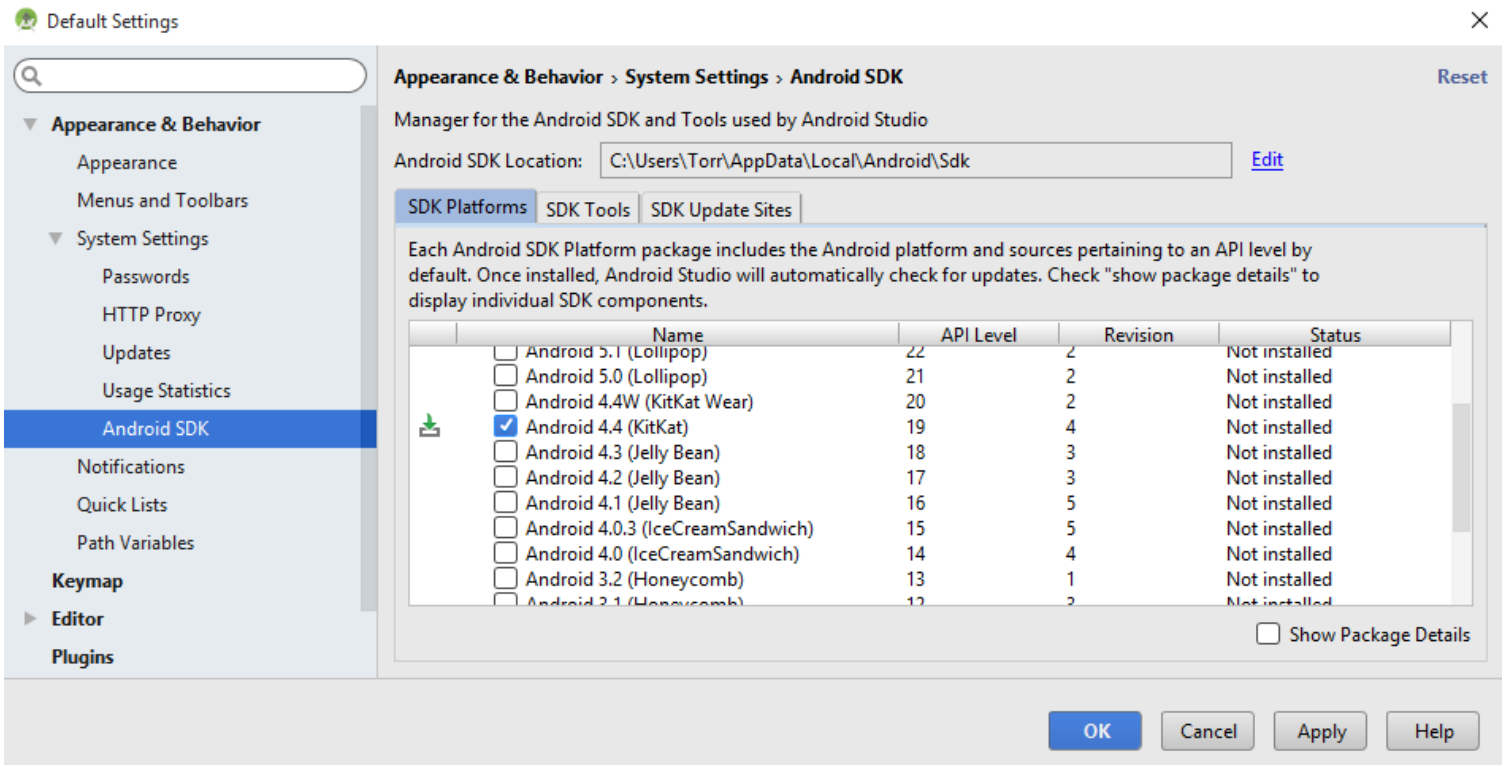
การเริ่มสร้าง Project

9. ทำการติดตั้ง SDK Platform ให้กับตัวโปรแกรม Android Studio ด้วยการคลิกปุ่ม SDK Manager เพื่อทำการดาวน์โหลด Android SDK



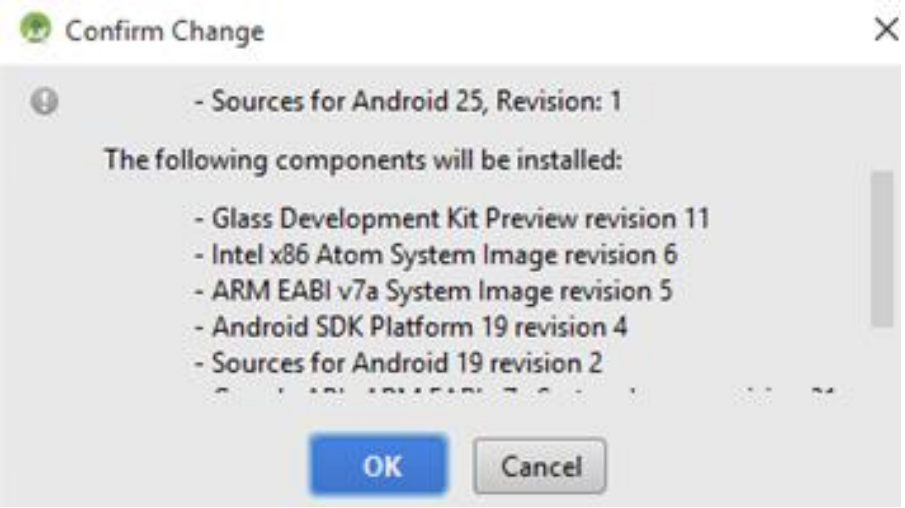
การเริ่มสร้าง Project

10. หน้าต่างแสดง SDK Platform รุ่นต่างๆก็จะปรากฏขึ้น ให้ทำการเลือกรุ่นที่จะใช้ในการเขียนโปรแกรมเพื่อรองรับกับแอพพลิชั่นที่จะนำไปใช้กับโทรศัพท์จริง เนื้อหาในบทนี้จะทำการเลือก Android 4.4 (Kitkat) จากนั้นทำการคลิกที่ปุ่ม OK



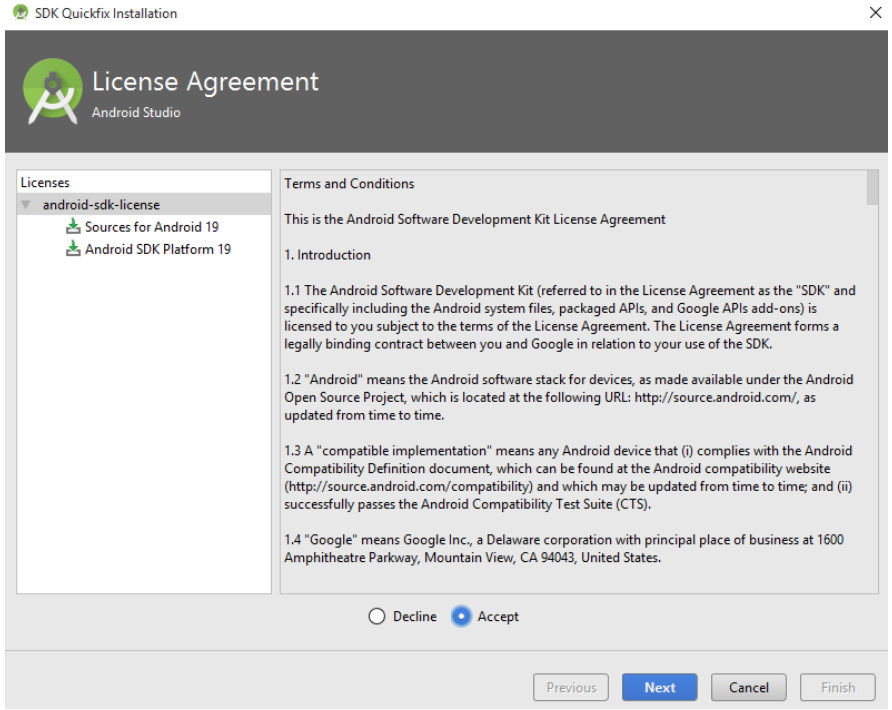
การเริ่มสร้าง Project

11. หน้าต่าง Confirm Change สำหรับแสดงรายละเอียดของส่วนประกอบที่จะทำการติดตั้งลงในโปรแกรมจะปรากฏขึ้นเพื่อให้เราทำการยืนยันการติดตั้ง ทำการคลิกที่ปุ่ม OK



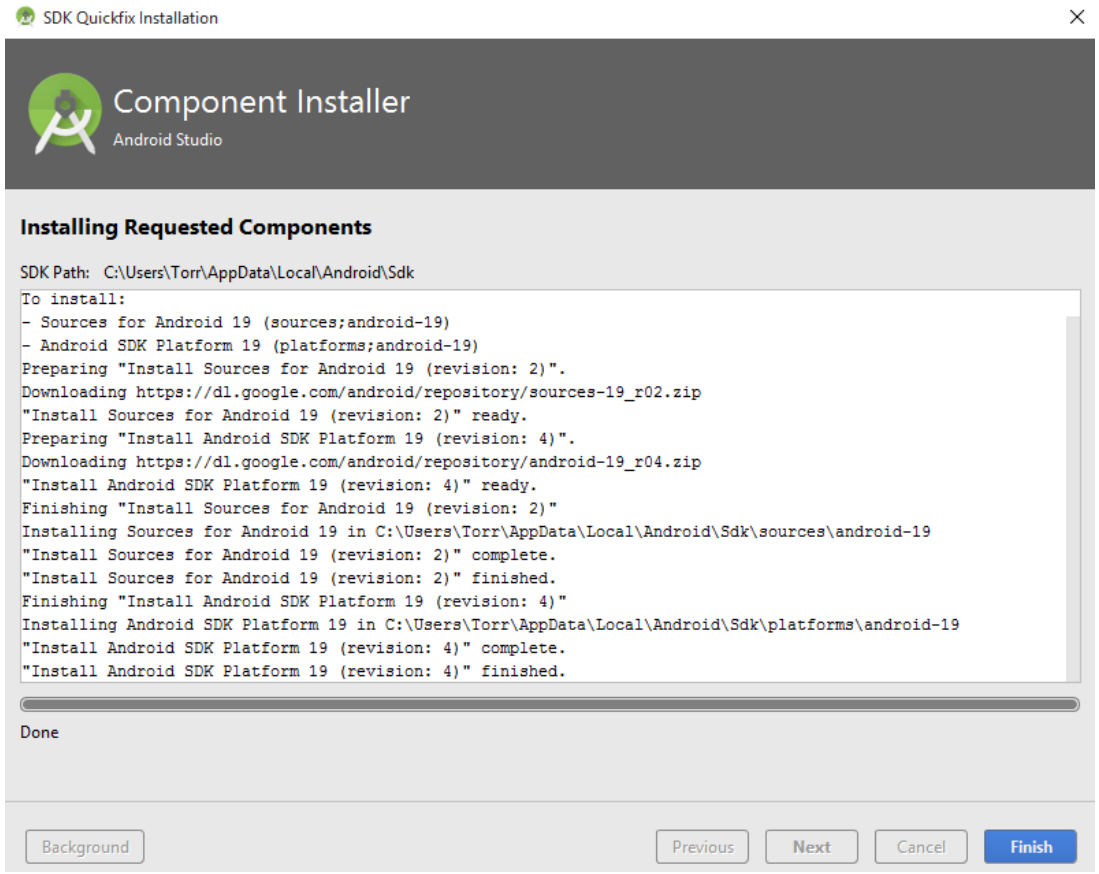
การเริ่มสร้าง Project

12. หลังจากนั้นทำการกดเลือก Accept เพื่อยอมรับเงื่อนไขในการติดตั้งส่วนประกอบของโปรแกรม ทำการคลิกที่ปุ่ม Next



การเริ่มสร้าง Project

13. โปรแกรมจะทำการติดตั้ง SDK จาก Platform ที่เราเลือก เมื่อทำการติดตั้งเสร็จแล้วทำการคลิกที่ปุ่ม Finish



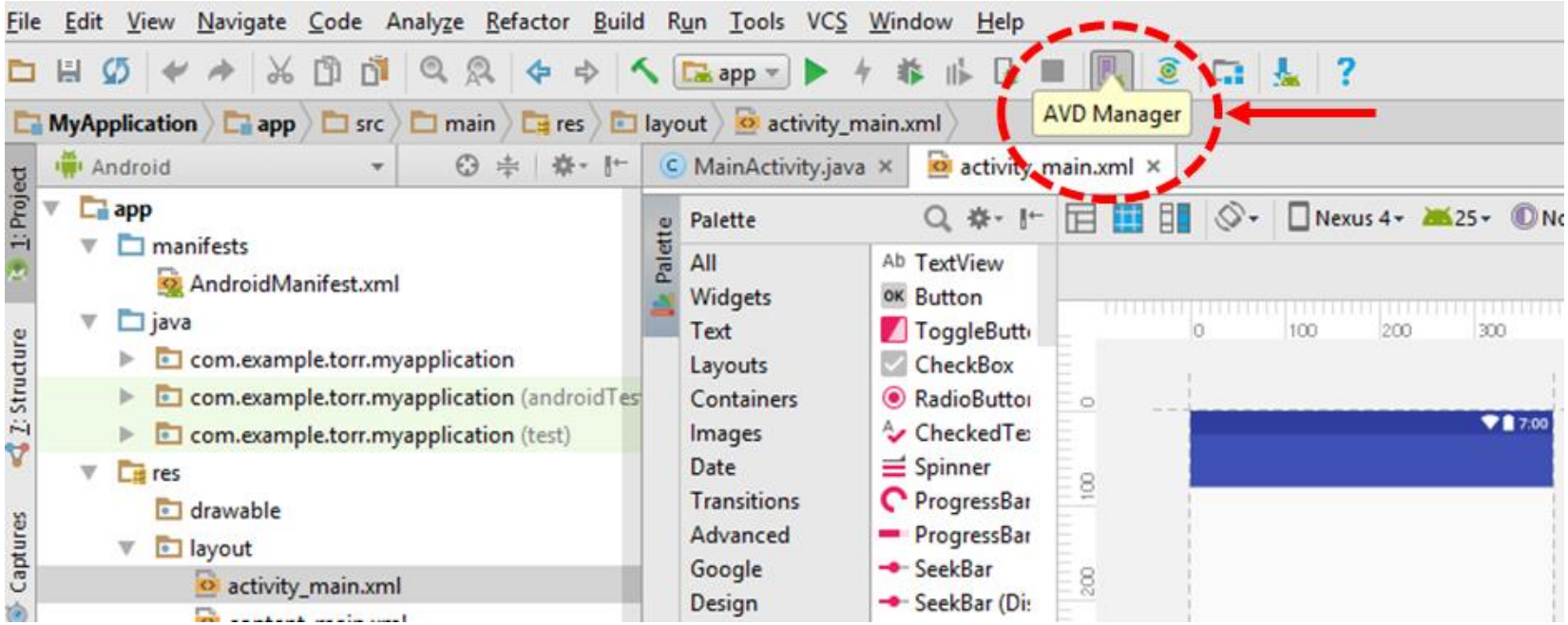
การสร้างโทรศัพท์จำลอง (Emulator)

AVD (Android Virtual Device) คือ การจำลองสภาวะแวดล้อมเสมือนจริงของโทรศัพท์เคลื่อนที่บนระบบปฏิบัติการแอนดรอยด์ขึ้นมา โดยมีจุดประสงค์เพื่อนำมาใช้สำหรับทำการทดสอบการทำงานของแอพพลิเคชั่นที่พัฒนาขึ้นมาเอง แอพพลิชั่นที่พัฒนาขึ้นมาและได้ทำการทดสอบบนโทรศัพท์เสมือนจริงเรียบร้อยแล้วนั้น เมื่อทำการนำแอพพลิเคชั่นไปติดตั้งบนโทรศัพท์จริงก็จะยังสามารถทำงานได้จริงด้วยเช่นกัน สำหรับการทดสอบการทำงานของแอพพลิเคชั่นที่พัฒนาขึ้นมานั้นเราจะต้องทำการสร้าง AVD ที่มีสภาพแวดล้อมเหมือนกับโทรศัพท์จริงที่เราจะนำแอพพลิเคชั่นที่ได้พัฒนาไปใช้งานขึ้นมาก่อน ซึ่งจะเรียกว่า Emulator

การใช้ Emulator ทำให้สามารถทำการพัฒนาแอพพลิเคชั่นได้อย่างสะดวกและรวดเร็วมากยิ่งขึ้น โดยผู้พัฒนาแอพพลิเคชั่นสามารถทำการทดสอบความถูกต้องและความผิดพลาดในการทำงานของแอพพลิเคชั่นใน Emulator ได้เลย เพื่อลดความเสียหายกับตัวโทรศัพท์จริง ซึ่งขั้นตอนในการสร้าง AVD จะมีขั้นตอนด้วยกันดังนี้

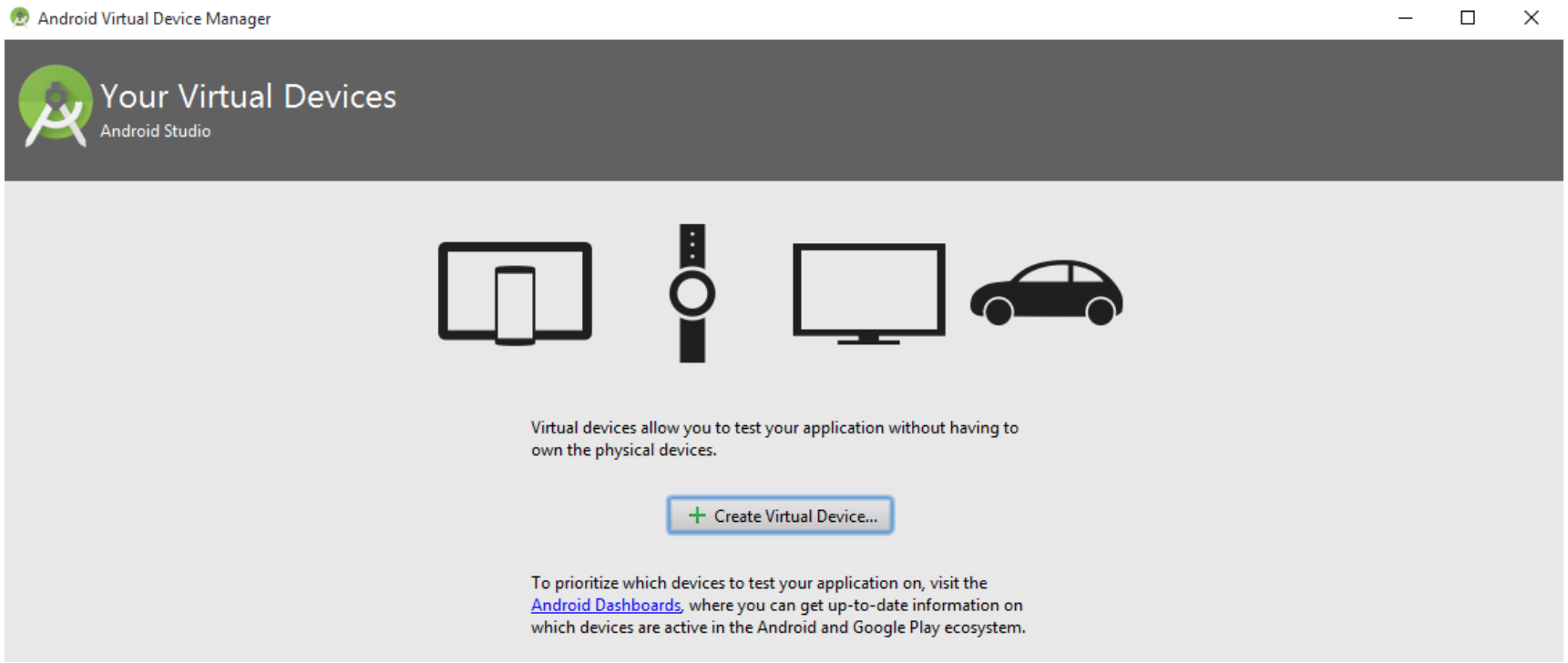
การสร้างโทรศัพท์จำลอง (Emulator)

1. ที่หน้า Project ทำการสร้าง AVD ให้กับตัวโปรแกรม โดยทำการคลิกปุ่ม AVD Manager



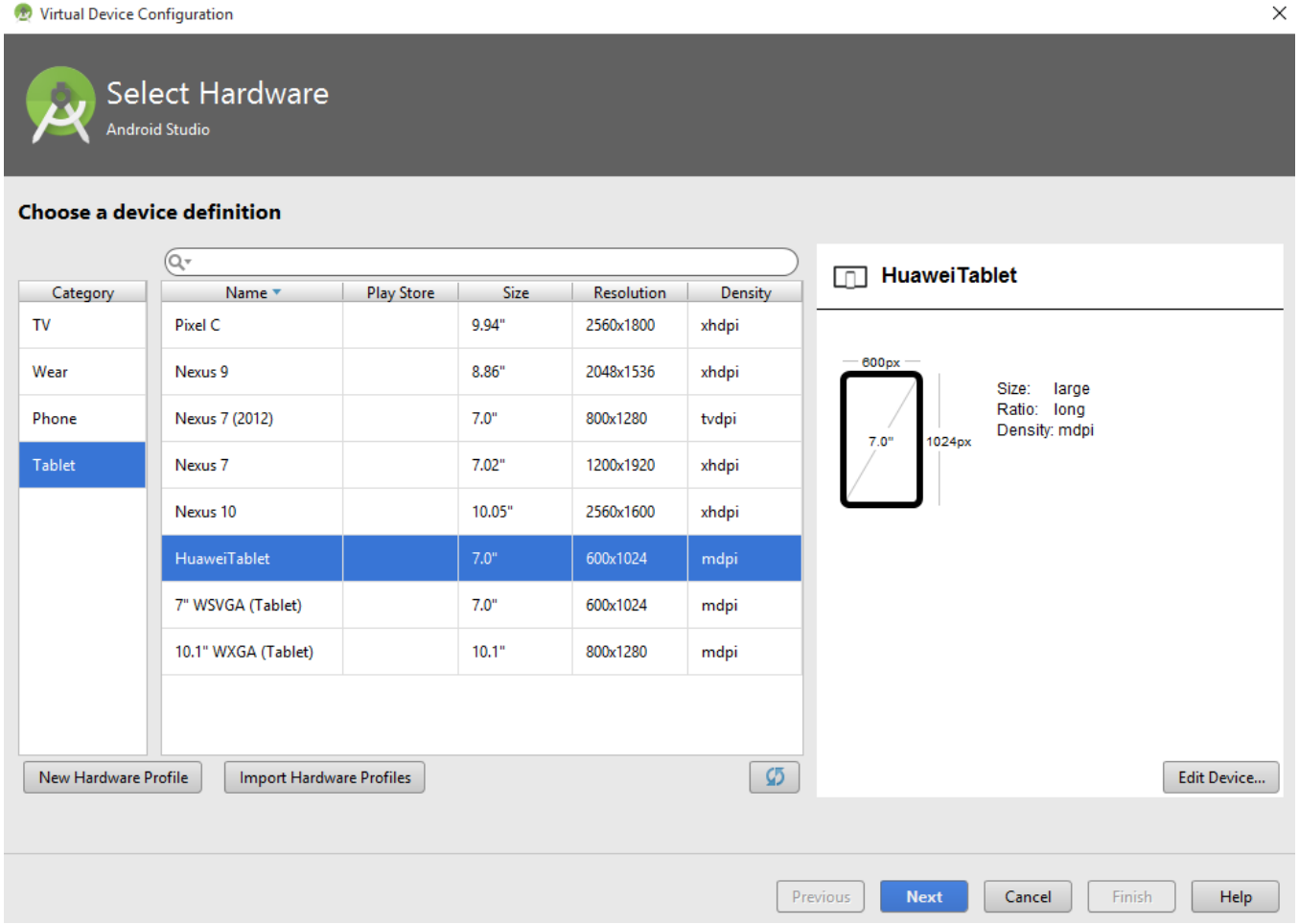
การสร้างโทรศัพท์จำลอง (Emulator)

2. หน้าต่าง Android Virtual Device Manager จะปรากฏขึ้นมา ให้ทำการคลิกที่ปุ่ม Create Virtual Device... เพื่อทำการสร้าง AVD



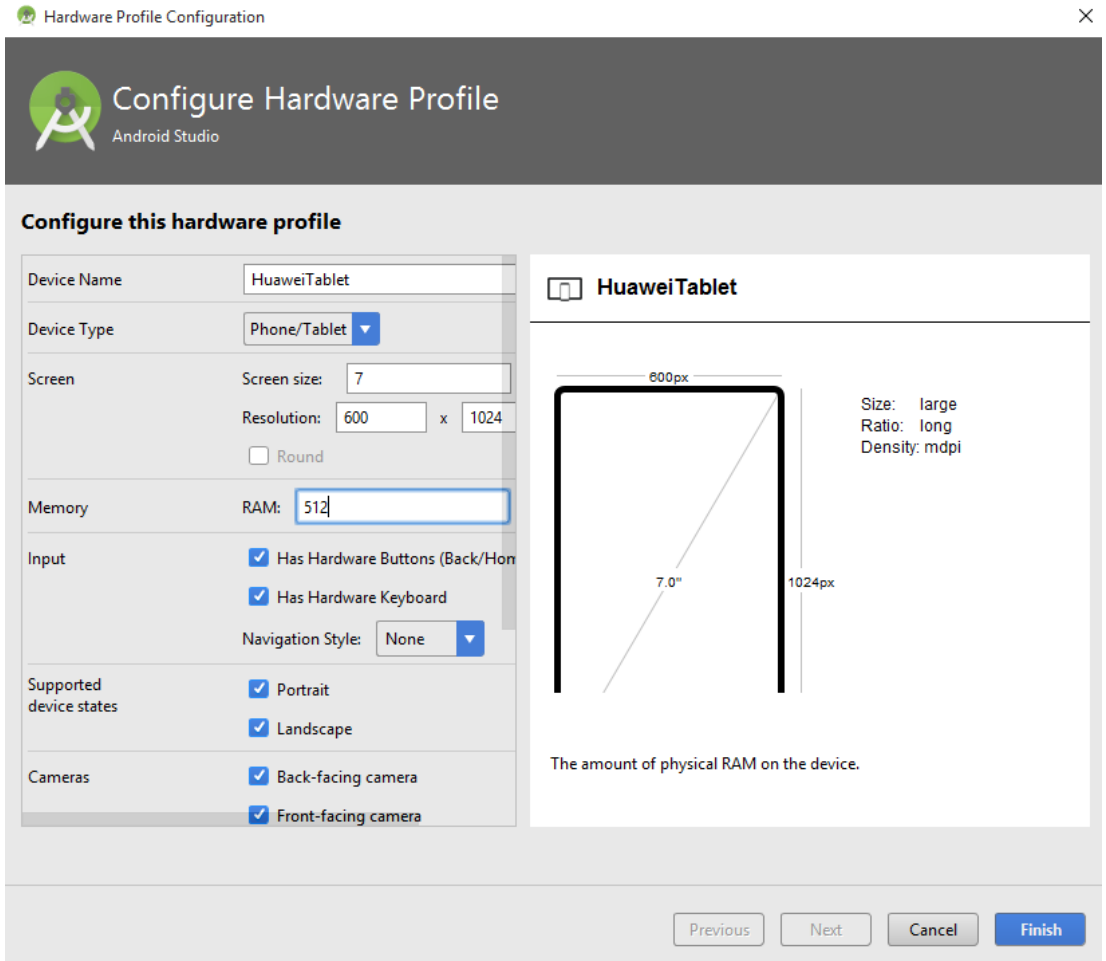
การสร้างโทรศัพท์จำลอง (Emulator)

3. หน้าต่าง Virtual Device Configuration จะปรากฏขึ้นมาเพื่อให้เราทำการเลือกรุ่นของอุปกรณ์ที่จะทำการสร้าง AVD ในกรณีนี้รุ่นและขนาดของอุปกรณ์ที่เราจะใช้ในการสร้าง AVD ไม่มีในลิสต์จึงต้องทำการสร้างขึ้นมาใหม่ โดยให้ทำการคลิกที่ปุ่ม New Hardware Profile



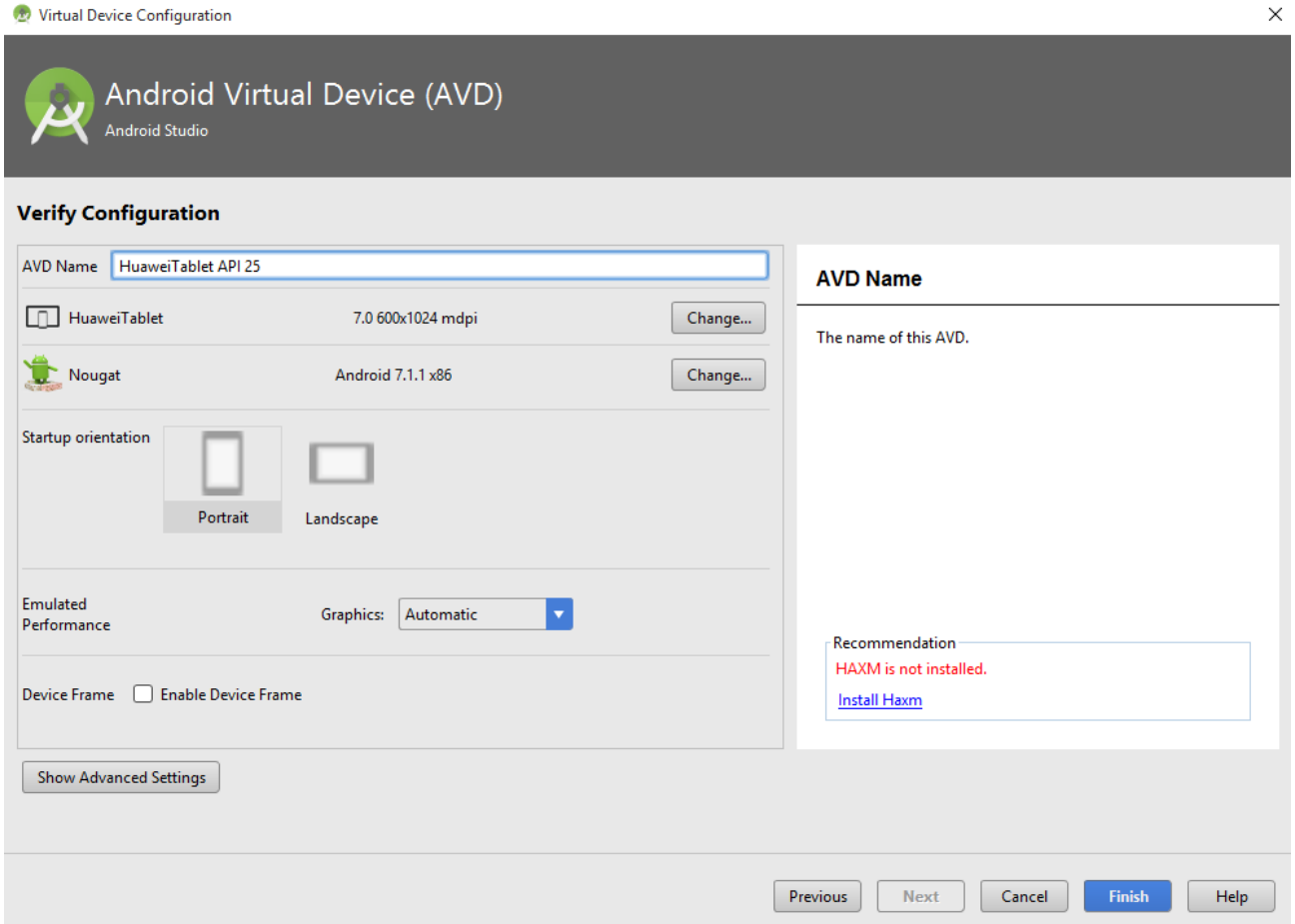
การสร้างโทรศัพท์จำลอง (Emulator)

4. หน้าต่าง Hardware Profile Configuration จะปรากฏขึ้นมาเพื่อให้ทำการกรอกรายละเอียดของอุปกรณ์ที่จะทำการสร้าง AVD ขึ้นมาใหม่ ทำการกรอกรายละเอียดและให้ทำการคลิกที่ปุ่ม Finish



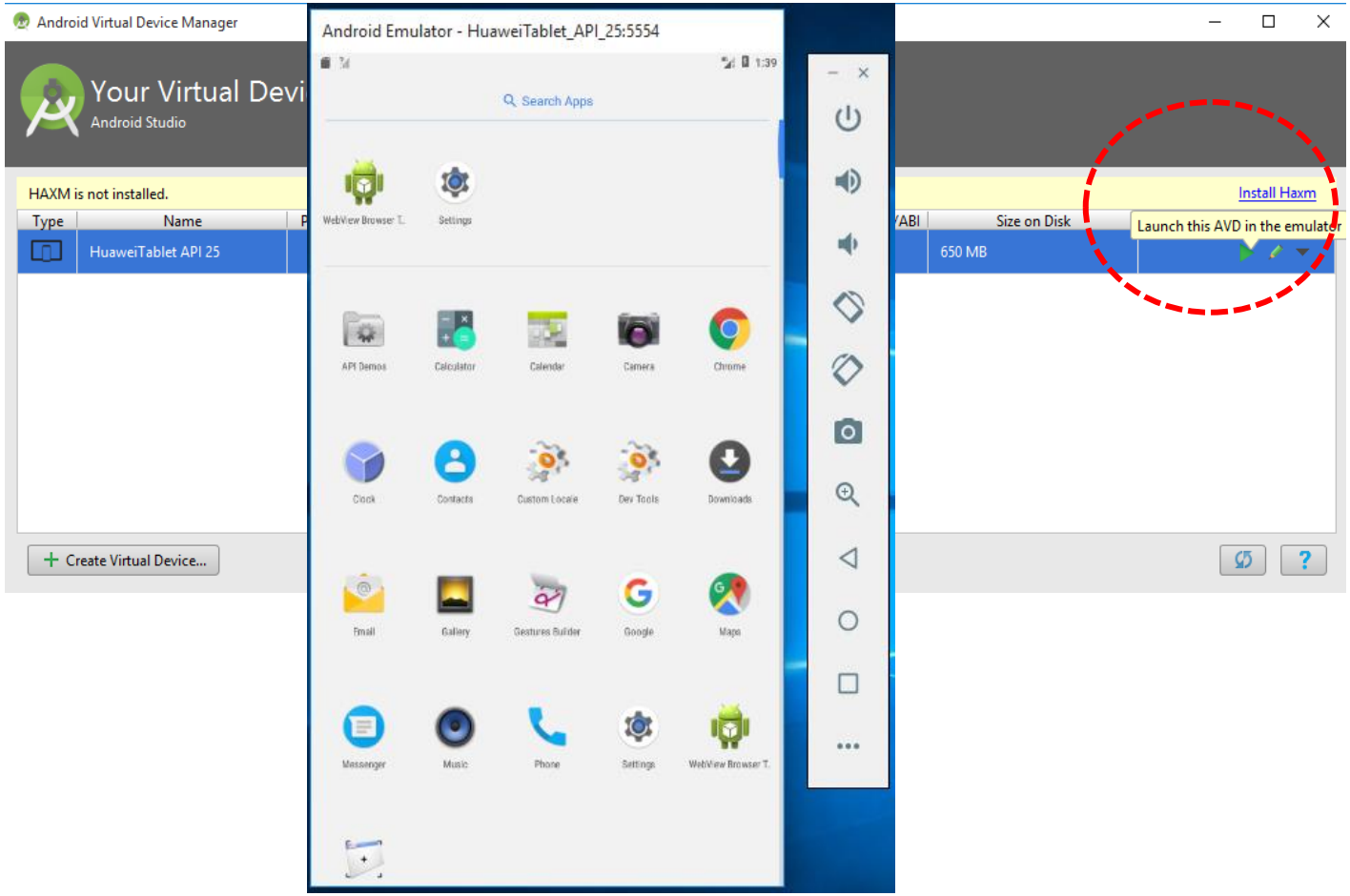
การสร้างโทรศัพท์จำลอง (Emulator)

5. หน้าต่าง Verify Configuration จะปรากฏขึ้นมาเพื่อให้เราทำการเช็ครายละเอียดของ AVD ที่สร้างขึ้นมาใหม่อีกครั้งก่อนทำการสร้างไว้ในตัวโปรแกรม ทำการคลิกที่ปุ่ม Finish



การสร้างโทรศัพท์จำลอง (Emulator)

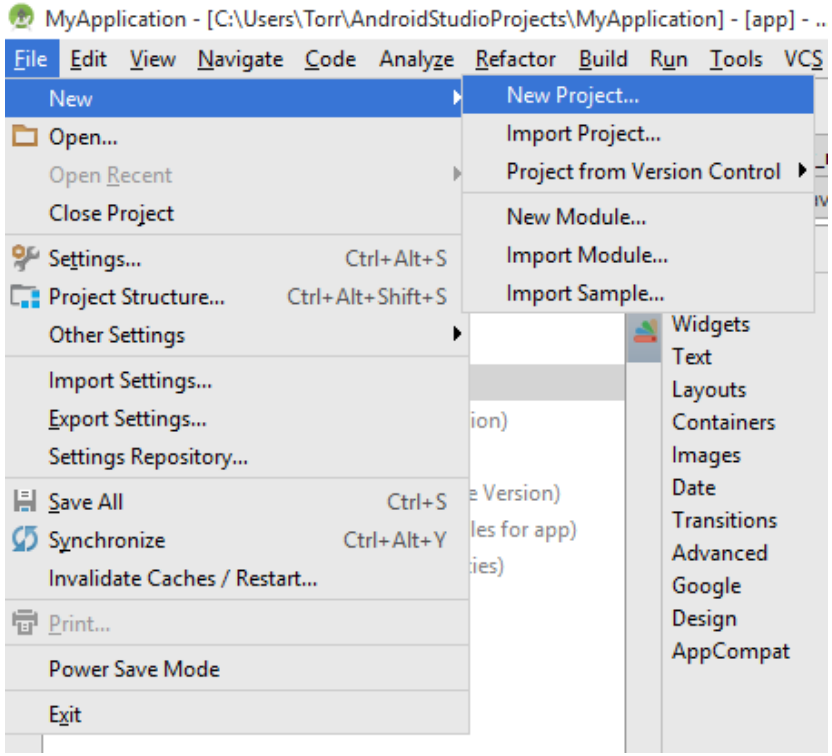
6. จากนั้น AVD ที่ได้ทำการสร้างขึ้นก็จะปรากฏขึ้นมาในแถบตัวเลือก ทำการคลิกที่ปุ่ม Run เพื่อทำการรัน Emulator สำหรับทำการทดสอบแอพพลิเคชั่นที่ได้สร้างไว้



การพัฒนาแอพพลิเคชั่นอย่างง่าย

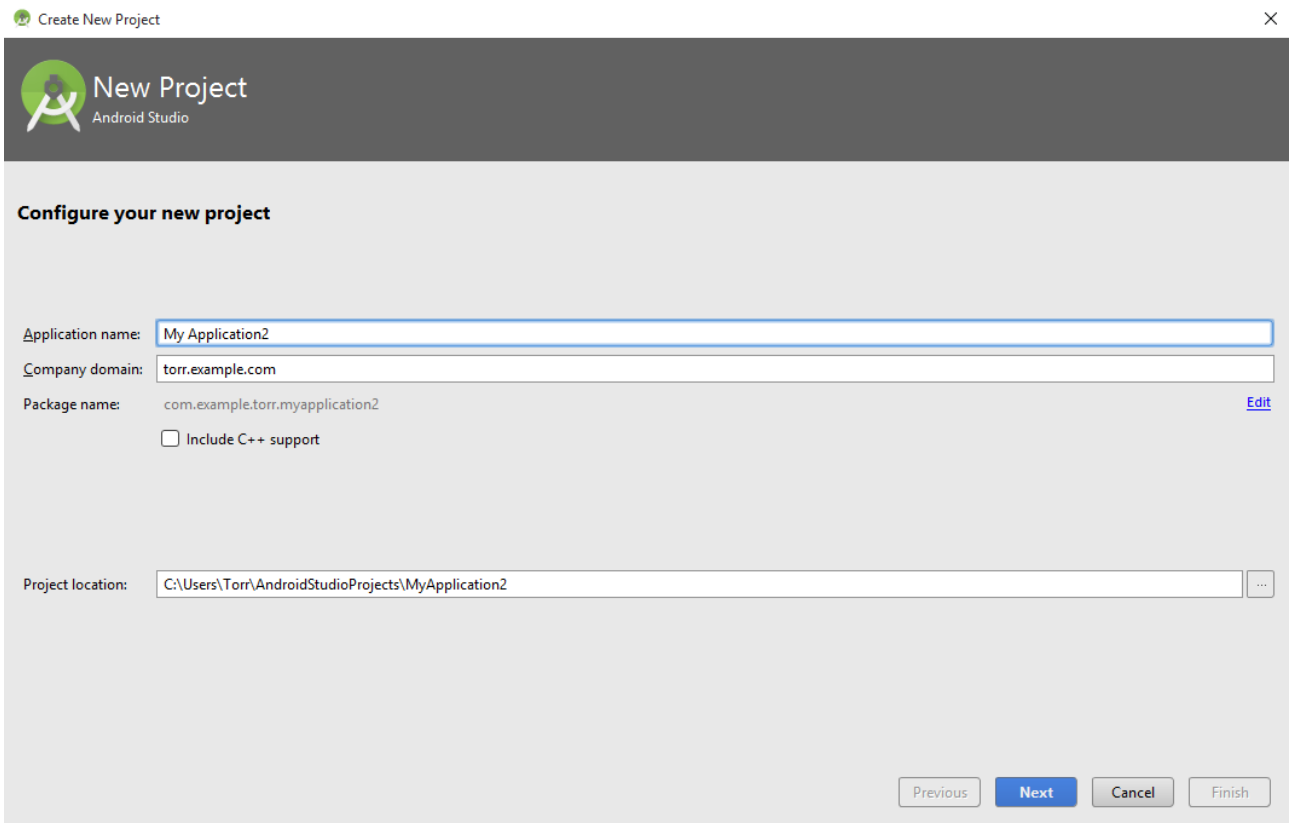
เนื้อหาในส่วนนี้จะเป็นการฝึกพัฒนาแอพพลิเคชั่นที่มีความซับซ้อนมากขึ้น โดยเราจะนำความรู้ด้านการเขียนโปรแกรมที่ได้เรียนมาแล้วในบทเรียนก่อนหน้ามาทำการพัฒนาโปรแกรมต่อยอดให้อยู่ในรูปแบบของแอพพลิเคชั่นเบื้องต้น ในบทนี้เราจะทำการนำโปรแกรมแลกเหรียญและโปรแกรมเปลี่ยนอุณหภูมิที่ได้เคยทำไว้แล้วมาทำการพัฒนาให้อยู่ในรูปแบบแอพพลิเคชั่น ซึ่งมีขั้นตอนดังนี้

1. เปิดโปรแกรม Android Studio คลิกไปที่เมนู File > New > New Project…



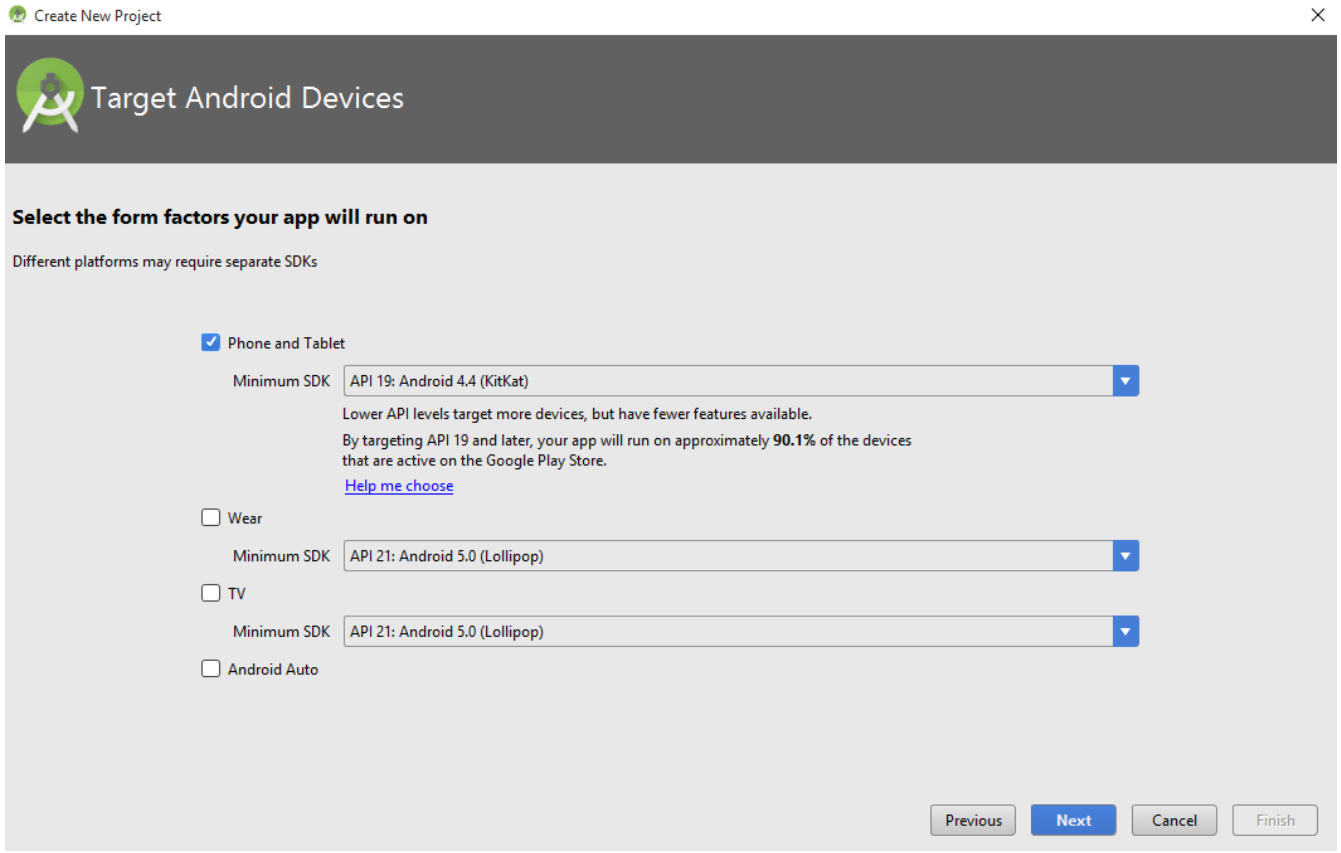
การพัฒนาแอพพลิเคชั่นอย่างง่าย

2. หน้าต่าง Create New Project จะปรากฏขึ้นมา เพื่อให้ทำการกรอกชื่อแอพพลิเคชั่นที่ช่อง Application name และกำหนดตำแหน่งในการเก็บไฟล์ เมื่อทำการตั้งชื่อเสร็จแล้วให้ทำการคลิกที่ปุ่ม Next >



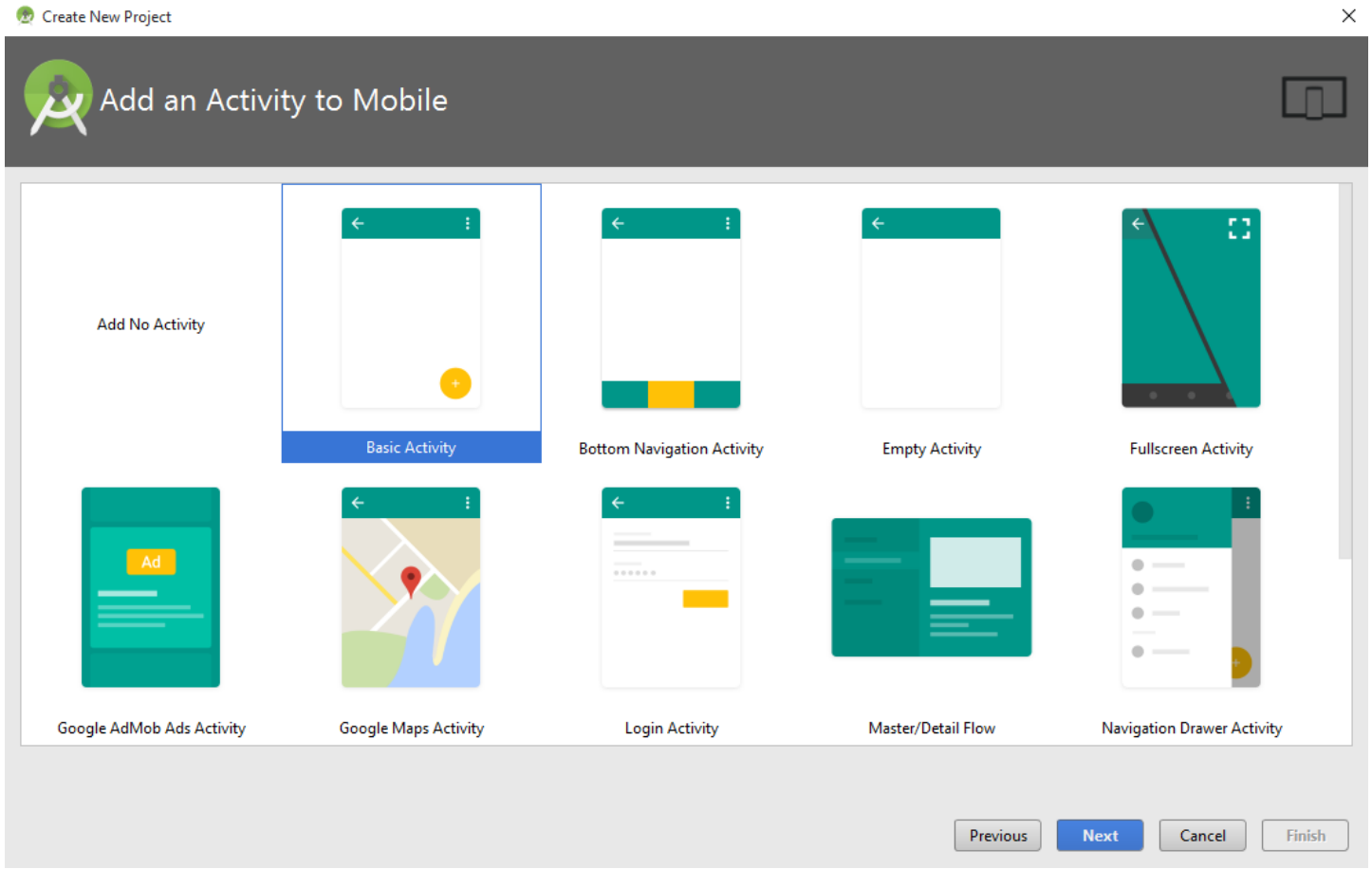
การพัฒนาแอพพลิเคชั่นอย่างง่าย

3. หน้าต่าง Target Android Devices จะปรากฏขึ้นเพื่อให้เราทำการกำหนดรุ่นของ SDK ที่จะใช้ในแอพพลิเคชั่น เมื่อทำการเลือกเสร็จแล้วให้ทำการคลิกที่ปุ่ม Next >



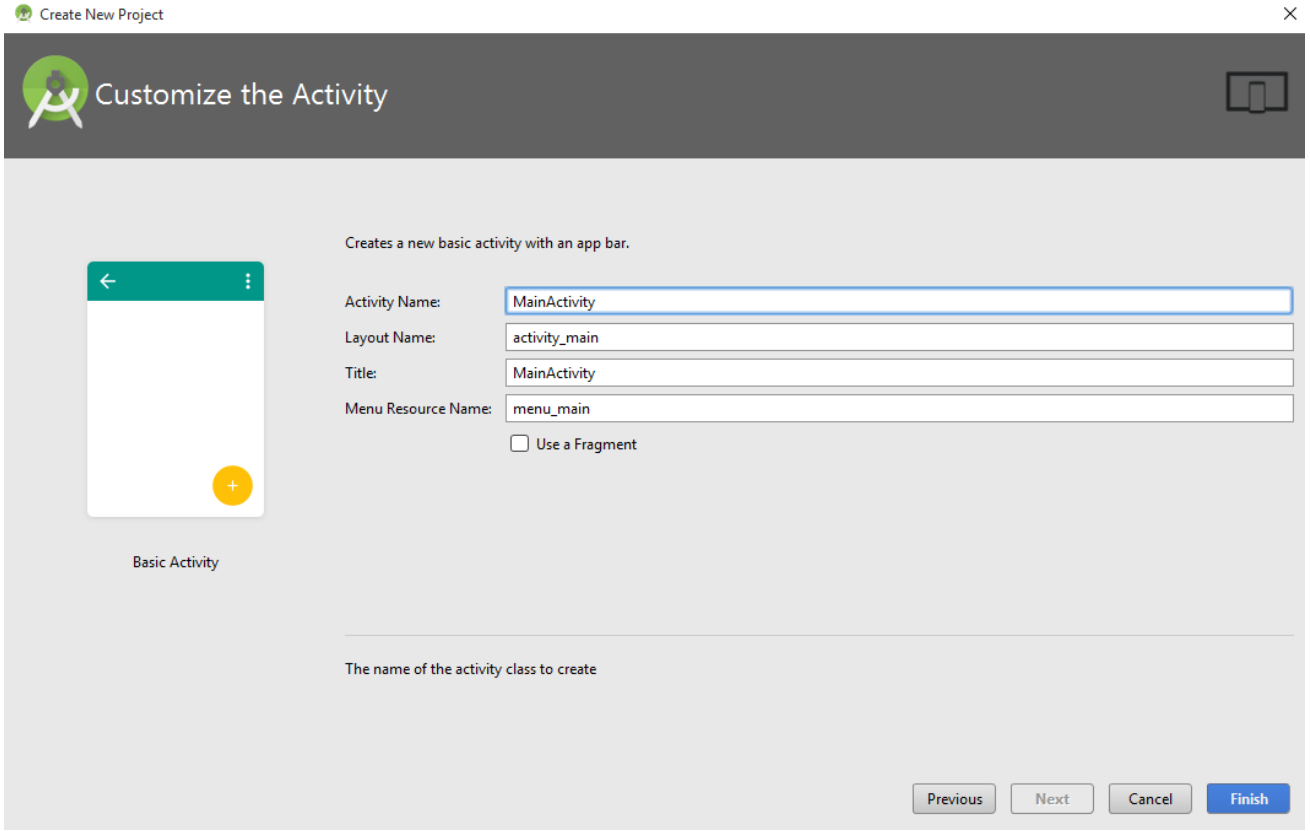
การพัฒนาแอพพลิเคชั่นอย่างง่าย

4. ทำการเลือก Activity เริ่มต้นสำหรับแอพพลิเคชั่นที่จะพัฒนา ในตัวอย่างนี้เราจะทำการเลือก Basic Activity ที่จะใช้ในโปรเจ็คต์ แล้วคลิกปุ่ม Next



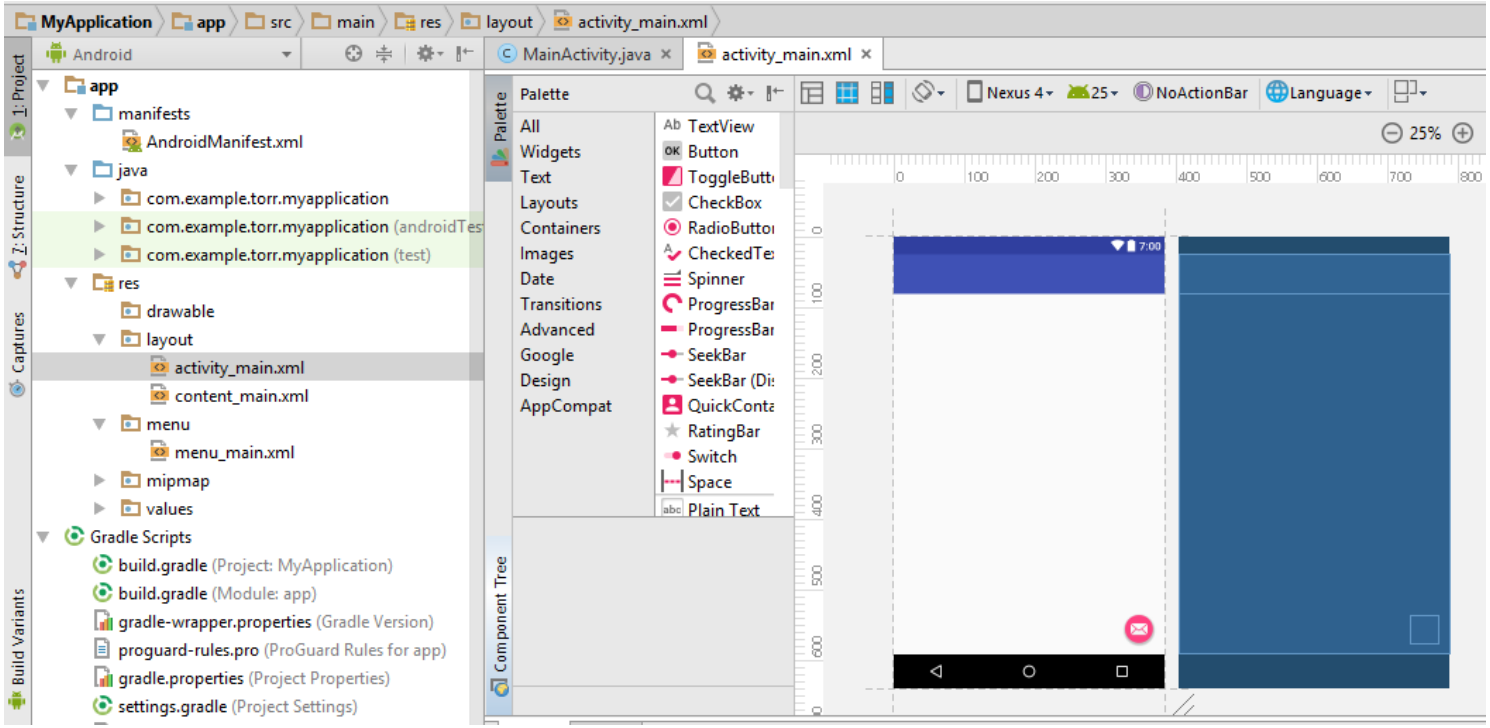
การพัฒนาแอพพลิเคชั่นอย่างง่าย

5. ทำการตั้งชื่อ Activity และLayout พร้อมทั้งชื่อของแถบ Title ที่ใช้ในโปรเจ็คต์ เมื่อทำแต่ละส่วนเสร็จแล้วให้ทำการคลิกที่ปุ่ม Finish



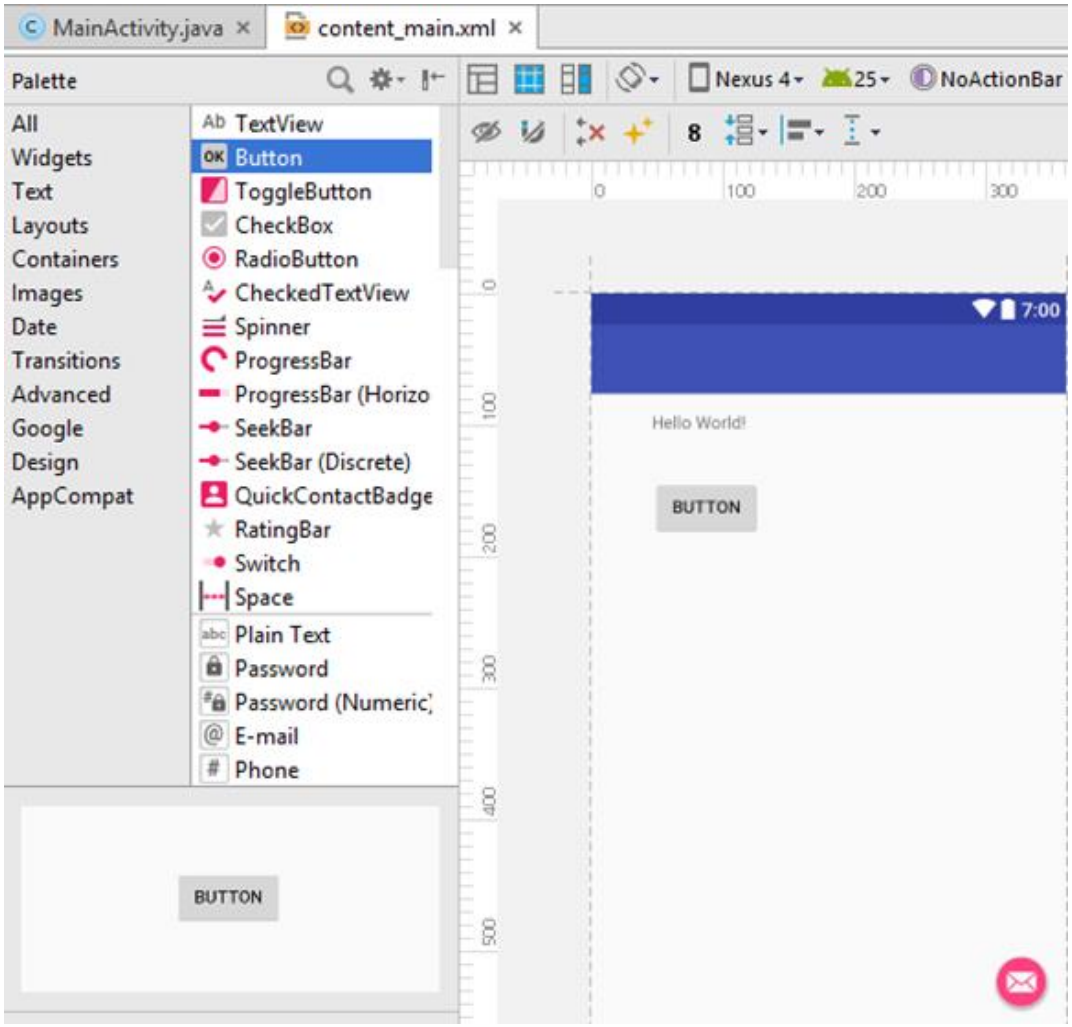
การพัฒนาแอพพลิเคชั่นอย่างง่าย

6. หน้าโปรแกรมสำหรับแสดงหน้าจอที่ใช้สำหรับการเขียนโปรแกรมในโปรเจ็คต์ และหน้าตาเริ่มต้นของแอพพลิเคชั่นก็จะแสดงขึ้นมา



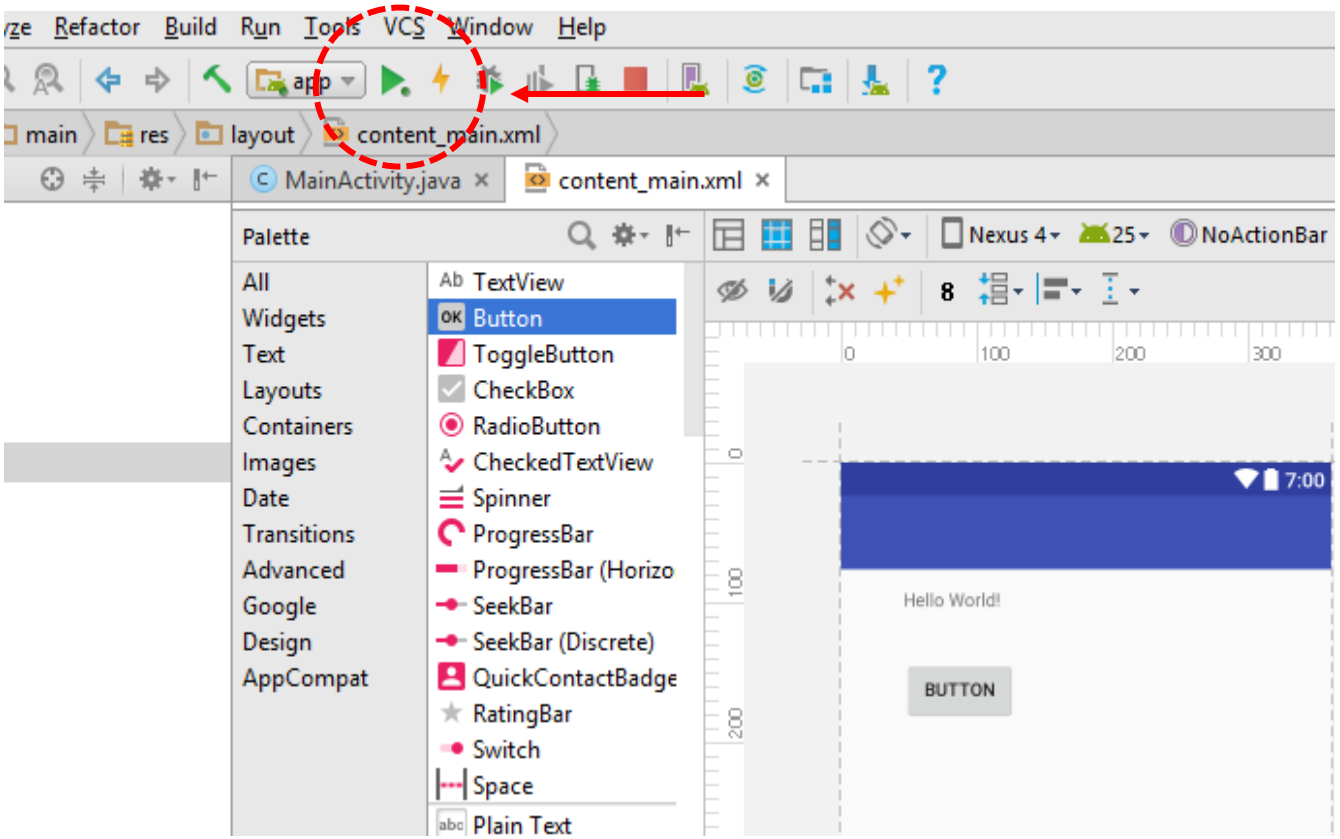
การพัฒนาแอพพลิเคชั่นอย่างง่าย

7. ทำการคลิกที่แถบ Palette ทำการออกแบบแอพพลิชั่นในเบื้องต้นแล้วทำการลาก Components ที่ต้องการไปวางยังตำแหน่งบนหน้าแอพพลิชั่นดังรูป



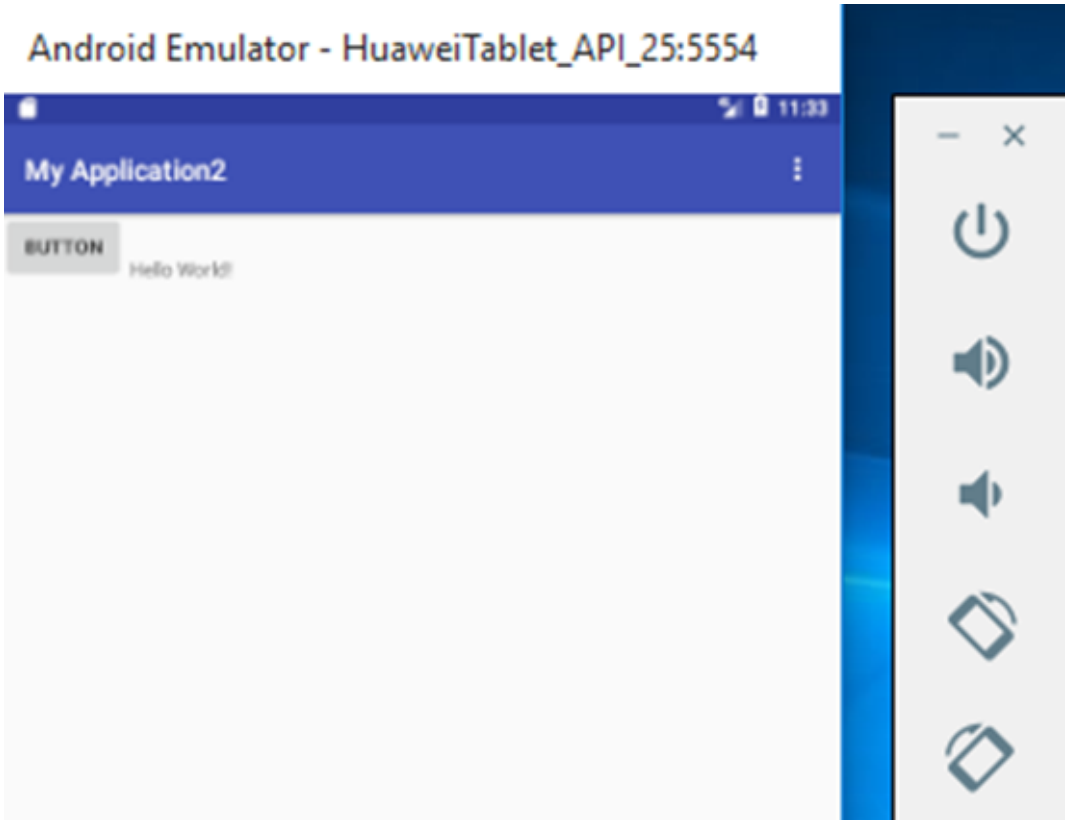
การพัฒนาแอพพลิเคชั่นอย่างง่าย

8. ทำการคลิกที่ปุ่ม Run เพื่อทำการทดสอบแอพพลิชั่นที่ได้ออกแบบไว้ในเบื้องต้นไปยังตัว Emulator



การพัฒนาแอพพลิเคชั่นอย่างง่าย

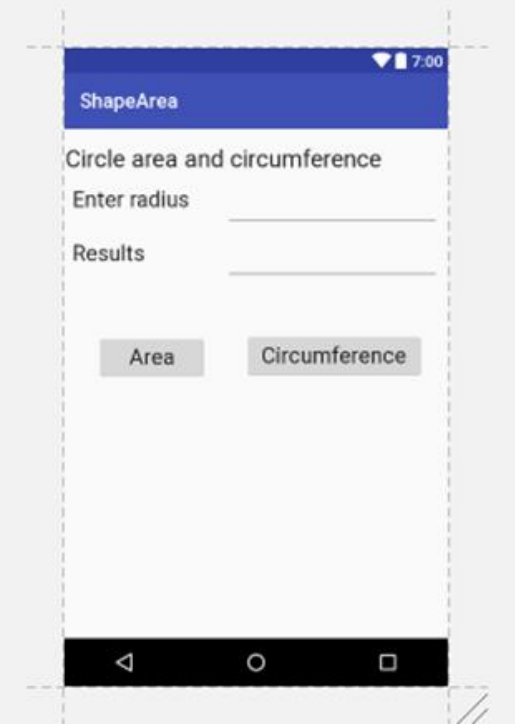
9. แอพพลิชั่นที่ได้ออกแบบไว้จะไปแสดงอยู่บนหน้า Emulator ที่ได้เคยสร้างไว้



การพัฒนาแอพพลิเคชั่นอย่างง่าย

ตัวอย่าง แอพพลิเคชั่นคำนวณพื้นที่วงกลมและคำนวณหาเส้นรอบรูปวงกลม

1.ทำการวางตำแหน่ง Components ต่างบนหน้าแอพพลิชั่นดังรูป



การพัฒนาแอพพลิเคชั่นอย่างง่าย

ตัวอย่าง แอพพลิเคชั่นคำนวณพื้นที่วงกลมและคำนวณหาเส้นรอบรูปวงกลม

2. ทำการแก้ไขโค้ดของโปรแกรมในไฟล์ Main.xml ดังนี้

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.torr.shapearea.MainActivity">

    <TextView
        android:id="@+id/showMessage01"
        android:layout_width="383dp"
        android:layout_height="33dp"
        android:layout_marginLeft="0dp"
        android:layout_marginRight="0dp"
        android:layout_marginTop="16dp"
        android:text="Circle area and circumference"
        android:textAppearance="@style/TextAppearance.AppCompat.Headline"
        android:visibility="visible"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

 <TextView
        android:id="@+id/inputNumber01"
        android:layout_width="135dp"
        android:layout_height="40dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="9dp"
        android:text="Enter radius"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/showMessage01" />
```

```
    <Button
        android:id="@+id/btn01"
        android:layout_width="111dp"
        android:layout_height="48dp"
        android:layout_marginLeft="32dp"
        android:layout_marginTop="50dp"
        android:text="Area"

android:textAppearance="@style/TextAppearance.AppCompat.Large"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/resultText01" />

    <Button
        android:id="@+id/btn02"
        android:layout_width="180dp"
        android:layout_height="49dp"
        android:layout_marginLeft="31dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="49dp"
        android:text="Circumference"

android:textAppearance="@style/TextAppearance.AppCompat.Large"
        app:layout_constraintHorizontal_bias="0.271"
        app:layout_constraintLeft_toRightOf="@+id/btn01"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editText02" />
```

การพัฒนาแอพพลิเคชั่นอย่างง่าย

ตัวอย่าง แอพพลิเคชั่นคำนวณพื้นที่วงกลมและคำนวณหาเส้นรอบรูปวงกลม

2. ทำการแก้ไขโค้ดของโปรแกรมในไฟล์ Main.xml ดังนี้

```
    <TextView
        android:id="@+id/resultText01"
        android:layout_width="135dp"
        android:layout_height="43dp"
        android:layout_marginLeft="8dp"
        android:layout_marginTop="14dp"
        android:text="Results"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/inputNumber01"
        android:layout_marginRight="8dp"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintHorizontal_bias="0.0" />

    <EditText
        android:id="@+id/editText01"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="number"
        app:layout_constraintLeft_toRightOf="@+id/inputNumber01"
        android:layout_marginLeft="0dp"
        android:layout_marginRight="8dp"
        app:layout_constraintRight_toRightOf="parent"
        android:layout_marginTop="9dp"
        app:layout_constraintTop_toBottomOf="@+id/showMessage01"
        app:layout_constraintHorizontal_bias="1.0" />
```

```
    <EditText
        android:id="@+id/editText02"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="number"
        android:layout_marginTop="11dp"
        app:layout_constraintTop_toBottomOf="@+id/editText01"
        app:layout_constraintLeft_toRightOf="@+id/resultText01"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintHorizontal_bias="1.0" />

</android.support.constraint.ConstraintLayout>
```

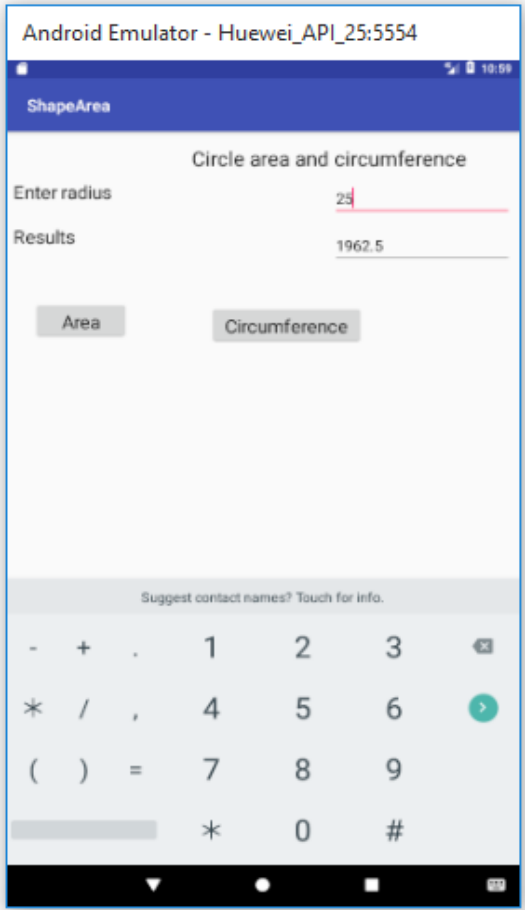
การพัฒนาแอพพลิเคชั่นอย่างง่าย

ตัวอย่าง แอพพลิเคชั่นคำนวณพื้นที่วงกลมและคำนวณหาเส้นรอบรูปวงกลม

3. แก้ไขโค้ด Java ในโปรแกรม shapearea.java ดังนี้

```
package com.example.torr.shapearea;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        final EditText et1 =(EditText) findViewById(R.id.editText01);
        final EditText et2 =(EditText) findViewById(R.id.editText02);

        Button btn1 = (Button) findViewById(R.id.btn01);        Button btn2 = (Button) findViewById(R.id.btn02);

        btn1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Float Area = (float)(3.14*(Float.parseFloat(et1.getText().toString()))*(Float.parseFloat((et1.getText().toString()))));
                et2.setText(Area.toString());
            }
        });

        btn2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Float Cir = (float)(3.14*(Float.parseFloat(et1.getText().toString()))*(Float.parseFloat((et1.getText().toString()))));
                et2.setText(Cir.toString());
            }
        });
    }
}
```

การพัฒนาแอพพลิเคชั่นอย่างง่าย

ตัวอย่าง แอพพลิเคชั่นคำนวณพื้นที่วงกลมและคำนวณหาเส้นรอบรูปวงกลม

4. ทำการคลิกที่ปุ่ม Run เพื่อทำการทดสอบแอพพลิชั่นที่ได้ออกแบบไว้ไปยังตัว Emulator