

คู่มือพัฒนาแอปพลิเคชันด้วย

android

STUDIO

ฉบับสมบูรณ์



- ครอบคลุมเนื้อหาตั้งแต่เริ่มต้น จนอัพโหลดแอปฯ ขึ้น Google Play Store ได้จริง
- อ่านเข้าใจง่าย มีตัวอย่างประกอบทุกหัวข้อ เกมโค้ดโปรแกรมทุกตัวอย่าง
- เหมาะสำหรับนักเรียน นักศึกษา และผู้ที่สนใจพัฒนา Android App

Free : Source Code <https://serazu.com/9786164870109>

■ ศุภชัย สมพานิช
■ บรรณาธิการ กิตติพันธ์ พลสวัสดิ์



มีเพียง “ความรู้” เท่านั้นที่มนุษย์ใช้พลิก “โลก” และเปลี่ยน “ชีวิต”
เราจึงสร้างสรรค์ และส่งมอบ “ความรู้” ในรูปแบบที่ดีกว่า
เพื่อให้คนไทย “เรียนรู้” ได้ตลอดชีวิต



คู่มือพัฒนาแอปพลิเคชันด้วย Android Studio ฉบับสมบูรณ์

ผู้แต่ง	ศุภชัย สมพานิช http://www.facebook.com/thaivb.net
บรรณาธิการ	กิตินันท์ พลสวัสดิ์ kitinan_p@idcpremier.com
ออกแบบปก	วสันต์ พึ่งพลผล, ยุทธนา เกิดประดิษฐ์
ออกแบบและจัดรูปเล่ม	สิริลักษณ์ วาระเลิศ
พิสูจน์อักษร	มนฤดี ศรีอุทโยภาส
ประสานงานการผลิต	วรพล ณีธิกุล, สุพัตรา อางปู้, พัทธน์ อ้อสถิตย์

Android Studio เป็นเครื่องหมายการค้าของบริษัท Google และเครื่องหมายการค้าอื่นๆ ที่อ้างถึงเป็นของบริษัทนั้นๆ

สงวนลิขสิทธิ์ตามพระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537 โดยบริษัท ไอดีซี พรีเมียร์ จำกัด ห้ามลอกเลียนไม่ว่าส่วนใดส่วนหนึ่งของหนังสือเล่มนี้ ไม่ว่าในรูปแบบใด ๆ นอกจากจะได้รับอนุญาตเป็นลายลักษณ์อักษรจากผู้จัดพิมพ์เท่านั้น

บริษัท ไอดีซี พรีเมียร์ จำกัด จัดตั้งขึ้นเพื่อเผยแพร่ความรู้ที่มีคุณภาพสู่ผู้อ่านชาวไทย เรายินดีรับงานเขียนของนักวิชาการและนักเขียนทุกท่าน ท่านผู้สนใจกรุณาติดต่อผ่านทางอีเมลที่ infopress@idcpremier.com หรือทางโทรศัพท์หมายเลข 0-2962-1081 (อัตรา 10 คู่สาย) โทรสาร 0-2962-1084

ข้อมูลทางบรรณานุกรม



ศุภชัย สมพานิช
คู่มือพัฒนาแอปพลิเคชันด้วย Android Studio ฉบับสมบูรณ์
นนทบุรี : ไอดีซีฯ, 2562
368 หน้า

1. แอนดรอยด์

สมาชิกสัมพันธ์

2. การเขียนโปรแกรม

โทรศัพท์ 0-2962-1081-3 ต่อ 121

1 ชื่อเรื่อง

โทรสาร 0-2962-1084

006.76

ร้านค้าและตัวแทนจำหน่าย

885-916-100-675-6

โทรศัพท์ 0-2962-1081-3 ต่อ 112-114

โทรสาร 0-2962-1084

ISBN

พิมพ์ครั้งที่ 1

กุมภาพันธ์ 2562

2 4 6 8 1 0 9 7 5 3

จัดพิมพ์และจัดจำหน่ายโดย

บริษัท ไอดีซี พรีเมียร์ จำกัด



200 หมู่ 4 ชั้น 19 ห้อง 1901 อาคารจัสตินอินเตอร์เนชั่นแนลทาวเวอร์
ถ.แจ้งวัฒนะ อ.ปากเกร็ด จ.นนทบุรี 11120

โทรศัพท์ 0-2962-1081 (อัตรา 10 คู่สาย) โทรสาร 0-2962-1084

ราคา 349 บาท

บรรณาธิการ

ในปัจจุบันสมาร์ทโฟน มือถือ เหมือนจะเป็นปัจจัยที่ 5 ในการดำเนินชีวิตไปแล้ว ผู้อ่านลองมองไปรอบๆ ก็จะเห็นว่า แทบทุกคนมีสมาร์ทโฟนใช้งาน บางคนอาจจะมียากกว่า 1 เครื่อง ซึ่งนั่นเป็นตัวบ่งบอกว่า ปัจจุบันสมาร์ทโฟนเป็นสิ่งที่ทุกคนต้องมี

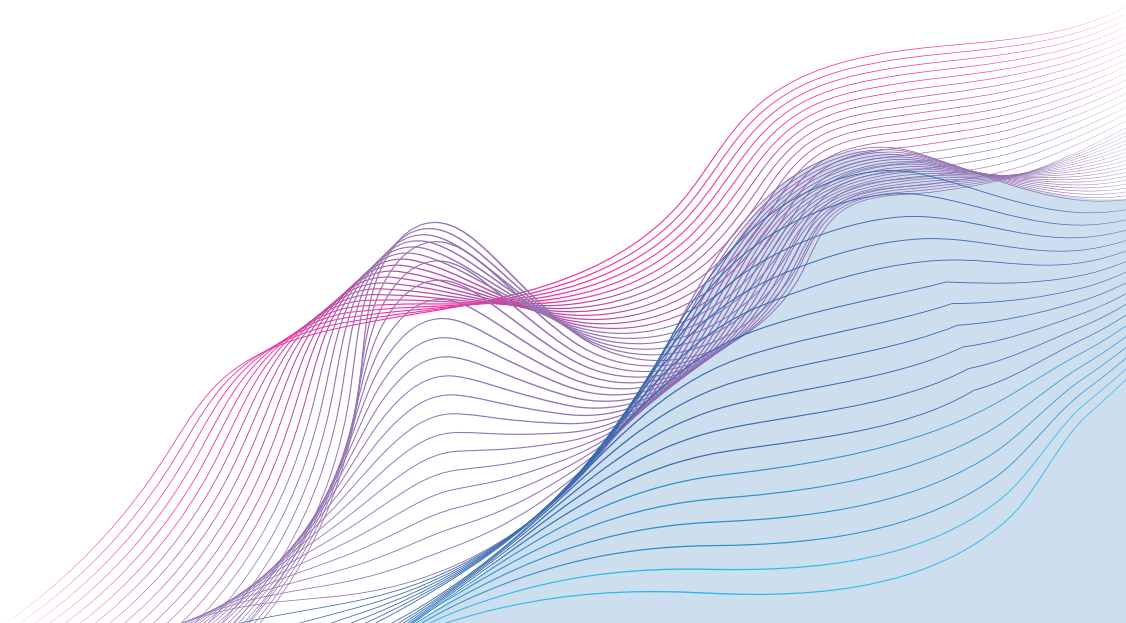
ในความจริงแล้ว สิ่งที่สำคัญไม่ใช่สมาร์ทโฟนแต่อย่างใด แต่เป็นแอปพลิเคชันที่ใช้งานในสมาร์ตโฟน นั้นเสียอีกที่มีความสำคัญ ไม่ว่าจะเป็นโน้ต, สมุดโทรศัพท์, แผนที่, Google Chrome, อีเมล, Facebook หรือ Line ซึ่งสมาร์ทโฟนเกือบทุกเครื่องจะมีแอปพลิเคชันเหล่านี้ใช้งาน และนั่นคือที่มาของหนังสือเล่มนี้

“หนังสือคู่มือพัฒนาแอปพลิเคชันด้วย Android Studio ฉบับสมบูรณ์” เล่มนี้ จะแนะนำให้รู้จักการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการ Android ด้วย Android Studio และอธิบายขั้นตอนการพัฒนาต่างๆ ตั้งแต่เริ่มต้นจนผู้อ่านสามารถพัฒนาแอปพลิเคชันเหล่านั้นให้สามารถใช้งานได้จริงบนสมาร์ตโฟน รวมไปถึงการพัฒนาแอปพลิเคชันที่มีโฆษณาเพื่อหารายได้ด้วย

สุดท้ายนี้ ทางสำนักพิมพ์หวังเป็นอย่างยิ่งว่า หนังสือเล่มนี้จะช่วยให้ผู้อ่านสามารถพัฒนาแอปพลิเคชัน Android – ขึ้นมาใช้งานได้อย่างที่ต้องการ

กิตินันท์ พลสวัสดิ์

บรรณาธิการ





คำนำ

“หนังสือคู่มือพัฒนาแอปพลิเคชันด้วย Android Studio ฉบับสมบูรณ์” เล่มนี้ นำเสนอการพัฒนา Android Apps ด้วยภาษา JAVA โดยปูพื้นฐานตั้งแต่ระดับง่ายที่สุด ไปจนกระทั่งถึงการสร้างแอปฯ แบบง่าย เพื่อแสดงให้เห็นวงจรการพัฒนาตั้งแต่เริ่มต้นจนได้ Android Apps แรกของคุณ

ในปัจจุบันรูปแบบการพัฒนา Android Apps มีหลายสิ่งหลายอย่างเปลี่ยนแปลงไปจากเดิมมากพอสมควร Google มีการปรับปรุงและกำหนดมาตรฐานใหม่ หลายสิ่งหลายอย่างถูกเพิ่มเติมเข้ามา ก็เป็นไปตามยุคสมัยที่เราคุ้นเคยกันเป็นอย่างดี

หากคุณผู้อ่านท่านใดที่ต้องการสอบถาม ดิฉัน เนื้อหาที่นำเสนอในหนังสือเล่มนี้ รวมไปถึงข้อผิดพลาดต่าง ๆ ที่อาจจะเกิดขึ้นได้ สามารถโพสต์ได้ 2 ช่องทางของผู้เขียน คือ

แฟนเพจ <https://www.facebook.com/thaivb.net/>

กลุ่มพูดคุยปัญหาเขียนโปรแกรม <https://www.facebook.com/groups/Thaivb.NET/>

ศุภชัย สมพานิช

กุมภาพันธ์ 2562

บทที่ 1 ก้าวแรกก่อนเข้าสู่การพัฒนา Android Apps ด้วยภาษา JAVA

รูปแบบการนำเสนอของหนังสือเล่มนี้.....	1
การดาวน์โหลดและติดตั้ง JAVA SDK.....	2
การดาวน์โหลด Android Studio.....	3
การติดตั้ง Android SDK เพิ่มเติมตามนโยบายใหม่ของ Google.....	4
การใช้งาน Android Studio เวอร์ชัน Preview	11
การอัปเดต Android Studio ให้เป็นเวอร์ชันใหม่ล่าสุด	12
การสร้างโปรเจกต์ใน Android Studio ตามนโยบายใหม่ของ Google	15
ทำความรู้จักกับสภาพแวดล้อมของ Android Studio ในขั้นต้น.....	18
การสร้าง Android Emulator สำหรับทดสอบโปรเจกต์.....	20
ขั้นตอนการสร้าง Android Emulator.....	21
การทดสอบโปรเจกต์ด้วย Android Emulator.....	26
การเพิ่มภาษาไทยใน Android Emulator	27
ปัญหารัน Android Emulator ไม่ได้เนื่องจาก Hyper-V.....	29
การทดสอบโปรเจกต์ Android Apps บนเครื่องจริง (ใช้กับ Android 6.0 ขึ้นไปเท่านั้น).....	30
การเปิดและปิดโปรเจกต์.....	33
วิธีการ Export โปรเจกต์เป็นไฟล์ ZIP.....	34
สรุปท้ายบท.....	34

บทที่ 2 พื้นฐานการพัฒนา Android Apps ด้วย Android Studio

ทำความรู้จักกับโครงสร้างโปรเจกต์ที่ได้จาก Android Studio	35
ทำความรู้จักกับ Gradle ใน Android Studio	39
การแก้ไขปัญหาจากคลาส R ในขั้นต้น.....	41
พื้นฐานการพัฒนา Android Apps	42
ทำความรู้จักกับระบบ dependencies.....	49

การกำหนดขนาด widget	56
การเปิดโปรเจกต์ Android Studio เก่าที่มีเวอร์ชันไม่ตรงกัน.....	58
การทดสอบโปรเจกต์แบบหน้าจอแหง (DisplayCutout).....	61
สรุปท้ายบท.....	62

บทที่ 3 ทำความรู้จักกับระบบ Layout ที่น่าสนใจ

พื้นฐานการระบุตำแหน่งใน ConstraintLayout	63
การระบุตำแหน่งด้วย Guideline.....	71
ส่วนแสดงผลแบบ LinearLayout.....	75
การสร้างช่องรับข้อความแบบใหม่ด้วย TextInputLayout.....	78
การแปลง Layout เดิมเป็น ConstraintLayout.....	83
การสร้างกรอบแจ้งข้อมูลแบบ Bottom Sheet Dialog	84
สรุปท้ายบท.....	91

บทที่ 4 การสร้างส่วนแสดงผลแบบรายการ (list) ด้วย RecyclerView

การใช้งาน RecyclerView ร่วมกับ CardView.....	94
การสร้างรายการใน RecyclerView ตามแนวนอน.....	107
การสร้างรายการใน RecyclerView แบบตาราง Grid.....	112
การแสดงผลข้อมูลแบบ Master Detail ด้วย RecyclerView	114
สรุปท้ายบท.....	123

บทที่ 5 ระบบเมนูและทูลบาร์

การสร้างโปรเจกต์ที่มีเมนูแบบ Navigation Drawer.....	126
พื้นฐานการปรับแต่งทูลบาร์ (Toolbar).....	138
การปรับแต่ง Layout ในแถบ Toolbar.....	138
การปรับปรุง Style ให้กับทูลบาร์.....	140
การสร้างแถบทูลบาร์แบบยืด-หดได้ (Collapse Toolbar).....	143
การใช้งานระบบเมนูพื้นฐาน.....	150
การนำทางแบบ Bottom Navigation.....	156
สรุปท้ายบท.....	161

บทที่ 6 ทำงานกับส่วนแสดงผลหลายหน้าจอด้วย Fragment

การปรับโครงสร้างโปรเจกต์ Navigation Drawer Activity ให้ใช้ส่วนแสดงผลแบบ Fragment	163
การเพิ่มส่วนแสดงผลแบบ Fragment เข้ามาในโปรเจกต์	167
การส่งข้อมูลจาก Activity ให้กับ Fragment.....	175
การใช้งาน Bottom Navigation ร่วมกับส่วนแสดงผลแบบ Fragment.....	180
สรุปท้ายบท.....	185

บทที่ 7 ระบบ Runtime Permission

Runtime Permission คืออะไร.....	187
สิทธิ์การขอใช้อินเทอร์เน็ต	189
สิทธิ์การโทร.....	191
สิทธิ์ขอเข้าถึงกล้องและพื้นที่จัดเก็บ	198
สรุปท้ายบท.....	204

บทที่ 8 การขอใช้บริการแผนที่ Google Maps

พื้นฐานการใช้งานแผนที่ Google Maps	205
การแก้ไขปัญหาเวอร์ชันของ Play services ไม่เข้ากัน	212
การใช้งานแผนที่ Google Maps กับโปรเจกต์ที่มีอยู่เดิม	215
ทำงานกับแผนที่ขั้นต้น.....	216
การปักหมุดตามตำแหน่งที่เรากำหนด.....	216
ปักหมุดบนแผนที่ด้วยวิธีสัมผัสและแสดงพิกัดละติจูดกับลองจิจูด	216
การล้างหมุดที่ปักอยู่บนแผนที่.....	217
แสดงตำแหน่งปัจจุบันของอุปกรณ์.....	217
สิทธิ์ในการขอใช้แผนที่ Google Maps.....	218
การกำหนดชนิดของแผนที่ Google Maps.....	224
การค้นหาที่อยู่และปักหมุดในแผนที่.....	227
การปรับแต่ง Style ให้กับแผนที่ Google Maps	235
การปรับแต่งรายละเอียดในแผนที่.....	239
สรุปท้ายบท.....	240

บทที่ 9 การผูกติดข้อมูล (Databinding)

การผูกติดข้อมูล (Databinding) ใน Android Apps	241
สรุปท้ายบท	248

บทที่ 10 ทำงานกับระบบฐานข้อมูล SQLite ด้วย Android Architecture Components

Android Architecture Components คืออะไร.....	249
การทำงานร่วมกับระบบฐานข้อมูล SQLite ด้วย Room.....	250
การนำฐานข้อมูล SQLite ออกจาก Android Emulator.....	271
การจัดการข้อมูลในฐานข้อมูล SQLite.....	272
ภาษา SQL คืออะไร.....	274
โครงสร้างพื้นฐานของภาษา SQL.....	274
การจัดการข้อมูลในฐานข้อมูลด้วยชุดคำสั่ง SQL.....	275
ตัวดำเนินการ (Operators) ในภาษา SQL.....	275
ลักษณะการใช้งานของกลุ่มคำสั่ง DML ของภาษา SQL.....	275
การลบข้อมูลด้วยคำสั่ง DELETE.....	275
การเพิ่มข้อมูลด้วยคำสั่ง INSERT.....	277
การอัปเดตข้อมูลด้วยคำสั่ง UPDATE.....	277
การเลือกดูข้อมูลด้วยคำสั่ง SELECT.....	278
การคิวรีข้อมูลแบบไม่มีเงื่อนไข.....	278
การคิวรีข้อมูลแบบมีเงื่อนไข.....	279
การใช้งานคำสั่ง SELECT ร่วมกับคำสั่ง LIKE.....	280
การจัดการข้อมูล CRUD ในฐานข้อมูล SQLite.....	281
สรุปท้ายบท	296

บทที่ 11 การจัดเก็บข้อมูลในระบบไฟล์ด้วย Shared Preferences

การใช้งาน Shared Preferences.....	298
การนำไฟล์ Shared Preferences ออกมาใช้งาน.....	305
สรุปท้ายบท	306

บทที่ 12 การสร้างไฟล์ apk สำหรับใช้งานจริง

ไฟล์ apk คืออะไร.....	307
วิธีการสร้างไฟล์ apk.....	308
การติดตั้งไฟล์ Signed APK ในเครื่องจริง	311
การสร้าง Signed APK กรณีใช้บริการแผนที่ Google Maps.....	313
สรุปท้ายบท.....	316

บทที่ 13 การอัปโหลดแอปฯ ขึ้น Google Play Store

การสมัครบัญชีนักพัฒนา Google.....	318
ขั้นตอนการอัปโหลดไฟล์ Signed APK ไปสู่ Google Play Store.....	319
สรุปท้ายบท.....	324

บทที่ 14 Workshop 1 : การสร้างแอปฯ ที่มีโฆษณาด้วย AdMob

การสมัครขอใช้บริการ AdMob.....	326
ทำความเข้าใจกับโครงสร้างโปรเจกต์แบบ Google AdMob Ads Activity	332
สรุปท้ายบท.....	333

บทที่ 15 Workshop 2 : การทำระบบส่ง E-mail ในแอปฯ

การทำระบบส่ง E-mail ในแอปฯ.....	336
สรุปท้ายบท.....	340

บทที่ 16 Workshop 3 : การอ่านตำแหน่ง (Location) ปัจจุบัน

การอ่านตำแหน่งปัจจุบัน.....	342
สรุปท้ายเล่ม.....	354



Android Studio

ก้าวแรกก่อนเข้าสู่การ พัฒนา Android Apps ด้วยภาษา JAVA

การพัฒนา Android Apps โดยอาศัยโปรแกรม Android Studio มีหลายสิ่งหลายอย่างเปลี่ยนแปลงไปมากพอสมควร ไม่ว่าจะเป็นตัวโปรแกรมที่ถูกเพิ่มเติมฟีเจอร์ต่างๆ มากมาย, ความก้าวหน้าทางตัวภาษา JAVA และยังรวมไปถึงรายการ widget มีทั้งที่ยังใช้งานต่อไป และบางส่วนถูกกำหนดให้อยู่ในสถานะล้าสมัย (Legacy หรือ Deprecated) ไปแล้ว เพื่อให้การพัฒนา Android Apps ในยุคใหม่มีประสิทธิภาพและความปลอดภัยสูงขึ้น

ในปัจจุบัน Google กำหนดให้ใช้ภาษา JAVA หรือภาษา Kotlin เป็น 2 ภาษาหลักของการพัฒนา Android Apps ด้วยโปรแกรม Android Studio และเนื้อหาการนำเสนอในหนังสือเล่มนี้ เป็นการใช้งานภาษา JAVA เพียงภาษาเดียว

รูปแบบการนำเสนอของหนังสือเล่มนี้

ขอบเขตการนำเสนอเนื้อหาในหนังสือเล่มนี้ เป็นการพัฒนา Android Apps โดยอาศัย Android Studio ด้วยภาษา JAVA ยึดถือตามนโยบายใหม่ของ Google บนฐานของโครงสร้างโปรเจกต์แบบใหม่ทั้งหมดที่มากับโปรแกรม Android Studio เริ่มต้นตั้งแต่เนื้อหาลำดับแรก จนถึงสร้างแอปฯ ขึ้นมาใช้งานจริงในลำดับสุดท้าย

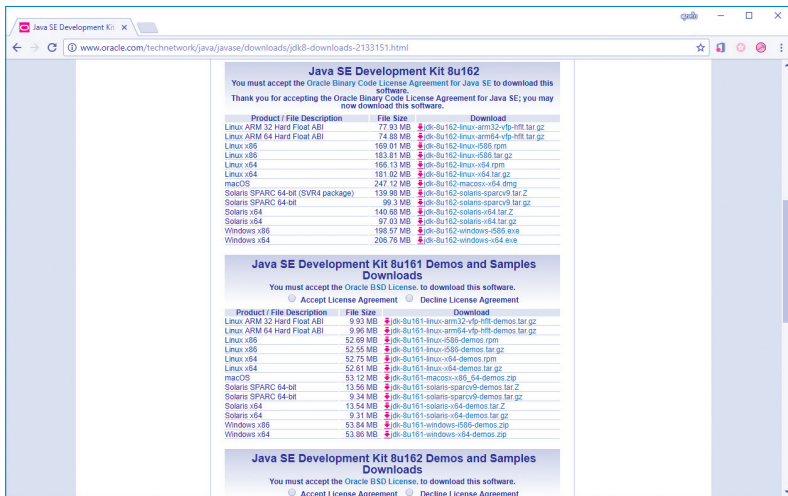
โค้ดของภาษา JAVA และ widget แบบเดิมหลายส่วน ถูกกำหนดให้อยู่ในสถานะล้าสมัย (Deprecated หรือ Legacy) ไปแล้ว กล่าวคือ โปรเจกต์เดิมที่คุณมียังคงใช้งานได้ตามปกติ แต่ไม่แนะนำให้ใช้งานแล้ว และยิ่งรวมไปถึงการลดการกระจายเวอร์ชันของระบบปฏิบัติการ Android เพื่อให้แอปฯ ที่ได้มามีประสิทธิภาพและความปลอดภัยมากขึ้น โดยลดความสำคัญของ Android เวอร์ชันเก่าลง ผู้เขียนจะกล่าวถึงอีกครั้งในลำดับต่อไป

การดาวน์โหลดและติดตั้ง JAVA SDK

คุณต้องติดตั้ง JAVA SDK ในเครื่องของคุณก่อน โดยสามารถดาวน์โหลดมาใช้งานฟรีได้ที่ <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

รูปที่ 1-1

แสดงการ
ดาวน์โหลด JAVA
SDK

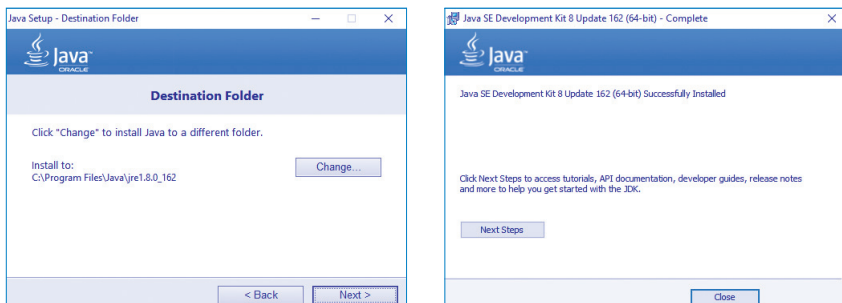


จากรูปที่ 1-1 ตามคำแนะนำของ Android Studio คุณควรติดตั้ง JAVA SDK เวอร์ชัน 8 ขอให้คุณตอบรับข้อตกลงด้านบน แล้วดาวน์โหลด JAVA SDK ให้ตรงกับระบบปฏิบัติการของคุณ ในกรณีที่ผู้เขียนใช้ Windows แบบ 64 บิต จึงเลือกดาวน์โหลดแบบ x64

หลังจากที่คุณดาวน์โหลด JAVA SDK เสร็จแล้ว ให้คุณดับเบิลคลิกไฟล์ที่ดาวน์โหลดมา เพื่อเริ่มต้นติดตั้ง JAVA SDK โดยใช้ค่าเริ่มต้นที่มักกับตัวติดตั้งทั้งหมด ดังรูปที่ 1-2

รูปที่ 1-2

แสดงการติดตั้ง
JAVA SDK เสร็จ
สมบูรณ์แล้ว



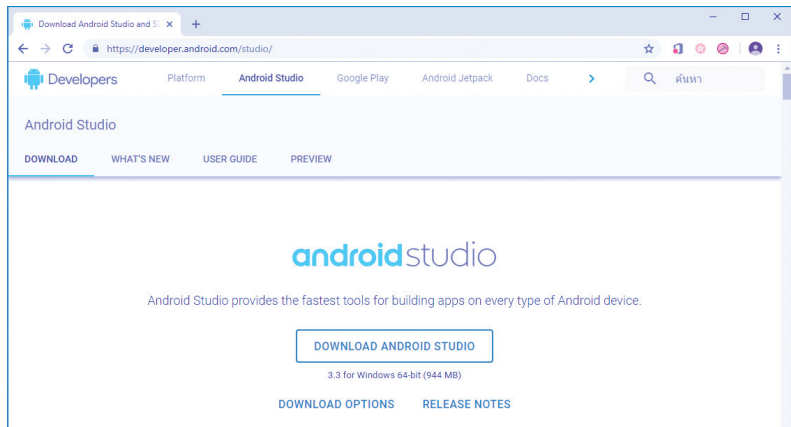
การดาวน์โหลด Android Studio

ในปัจจุบัน Google แยกตัวโปรแกรม Android Studio ออกเป็น 2 แบบ คือ

1. **Android Studio เวอร์ชันหลักหรือ Stable** สามารถดาวน์โหลดได้ที่ <https://developer.android.com/studio/index.html> เป็นเวอร์ชันที่คุณสามารถพัฒนา Android Apps เพื่อนำไปใช้งานจริง

รูปที่ 1-3

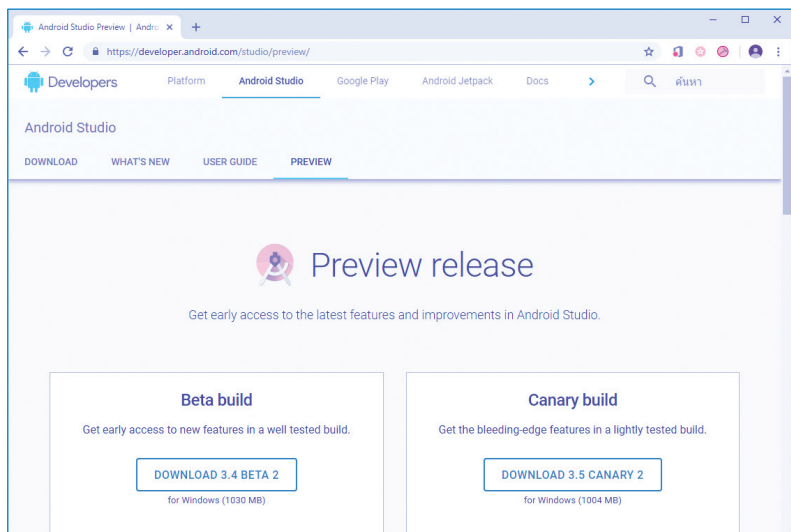
แสดงการ
ดาวน์โหลด
Android Studio
เวอร์ชันหลัก



2. **Android Studio เวอร์ชันพรีวิหรือ Preview** สามารถดาวน์โหลดได้ที่ <https://developer.android.com/studio/preview/index.html> ฟีเจอร์ใหม่ๆ ของ Android จะถูกนำมาใส่ในแบบนี้ก่อน อาจมีความเสี่ยงในแง่ที่ว่า ฟีเจอรันั้นๆ อาจจะไม่คงอยู่หรือถูกตัดออกก็ได้ เป็นเวอร์ชันที่เหมาะสมสำหรับคนที่ต้องการทดสอบฟีเจอร์ใหม่ๆ ก่อนเวอร์ชันหลัก ดังรูปที่ 1-4

รูปที่ 1-4

แสดงการ
ดาวน์โหลด
Android Studio
เวอร์ชันพรีวิ



โปรเจกต์ที่ถูกสร้างจาก Android Studio ทั้ง 2 แบบ สามารถเปิดไปมาระหว่างกันได้ โดยมีข้อแม้ว่า Android Studio เวอร์ชันนั้นๆ จะต้องมียูเอชเอชที่เท่าเทียมกัน

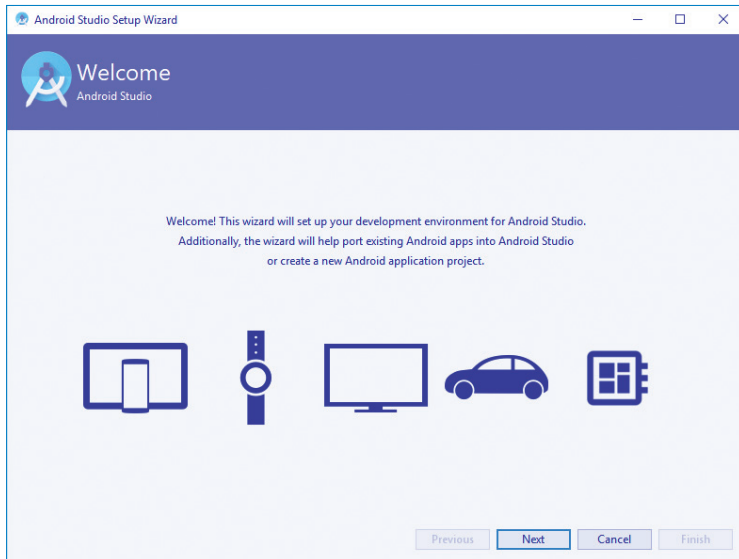
การติดตั้ง Android SDK เพิ่มเติมตามนโยบายใหม่ของ Google

นโยบายใหม่ของ Google ต้องการที่จะลดการกระจายเวอร์ชันของ Android โดยใช้ Android เวอร์ชัน 8 (API 26) ขึ้นไป เป็นเวอร์ชันพื้นฐานสำหรับการพัฒนาแอปฯ ในยุคใหม่

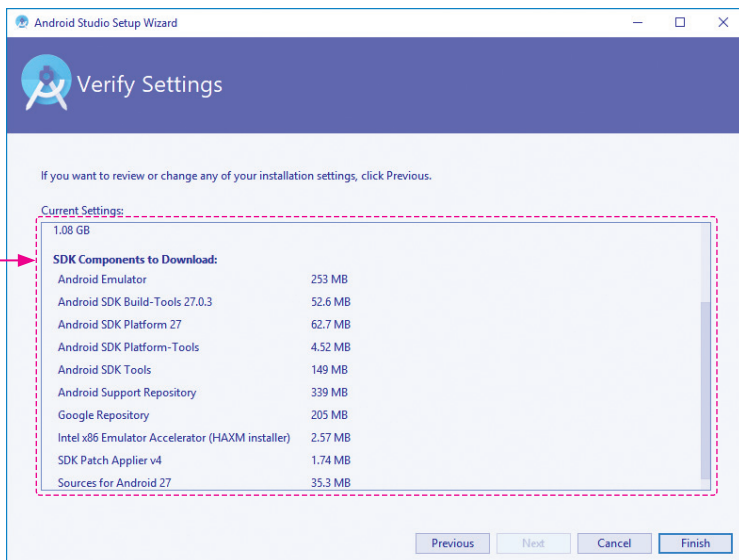
ในขั้นตอนติดตั้งโปรแกรม Android Studio จะมีการติดตั้ง Android SDK ตั้งแต่เวอร์ชัน 8 ขึ้นไปให้โดยอัตโนมัติอยู่แล้ว แต่ยังไม่สามารถนำมาใช้พัฒนา Android Apps ได้ในโลกของความเป็นจริง ดังรูปที่ 1-5

รูปที่ 1-5

การติดตั้ง
รายการต่างๆ
ครั้งแรกของ
ตัวโปรแกรม
Android Studio

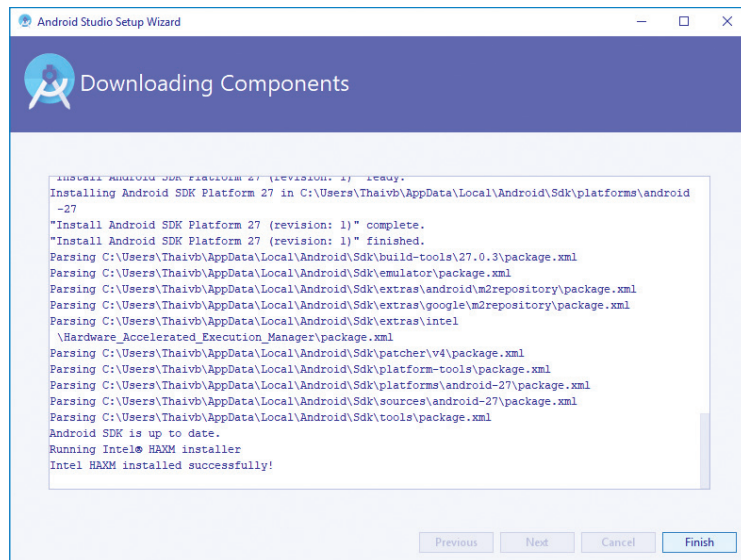
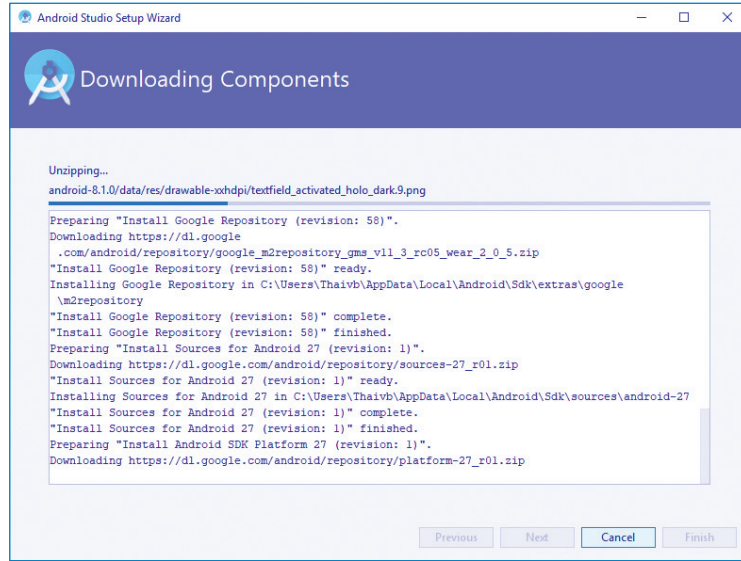


เวอร์ชันที่ติดตั้งเป็น
ไปตามช่วงเวลาของ
คุณเป็นหลัก



รูปที่ 1-6

หลังจากติดตั้ง
รายการต่างๆ
เสร็จแล้ว



เหตุผลคือ เราต้องการให้แอปฯ ที่สร้างขึ้นมา สามารถใช้งานบนระบบปฏิบัติการ Android หลายเวอร์ชันมากที่สุดเท่าที่เป็นไปได้ ส่งผลให้ Android Studio ต้องมีการติดตั้ง Android SDK เพิ่มเติม

ในช่วงรอยต่อระหว่างนี้ การกำหนดให้แอปฯ ที่สร้างขึ้นมามีรันได้ตั้งแต่เวอร์ชัน 8 ขึ้นไป อาจจะทำให้แอปฯ ของเรามีพื้นที่แคบเกินไป มือถือและแท็บเล็ตที่ติดตั้ง Android เวอร์ชัน 8 ยังมีจำนวนในตลาดไม่มากนัก จึงต้องใช้ระยะเวลาหนึ่ง (ขยับเวอร์ชันใหม่ตามช่วงเวลา)

ผู้เขียนคิดว่า ยังคงสามารถเลือกติดตั้ง Android SDK เวอร์ชัน 6 (API 23) เป็นเวอร์ชันต่ำสุดเท่าที่จะเป็นไปได้ เพราะว่า Android เวอร์ชัน 6 ผู้ใช้งานสามารถเลือกยอมรับหรือปฏิเสธการขอสิทธิ์ต่างๆ จากแอปฯ ได้ตามที่ต้องการ ยังเกาะกลุ่มของ Android 8 หรือใหม่กว่าได้ในระดับหนึ่ง

หลักการพื้นฐานในการเลือกติดตั้ง Android SDK ก็คือ ติดตั้งเฉพาะเวอร์ชันหัวกับท้ายก่อน ในกรณีนี้ คือ

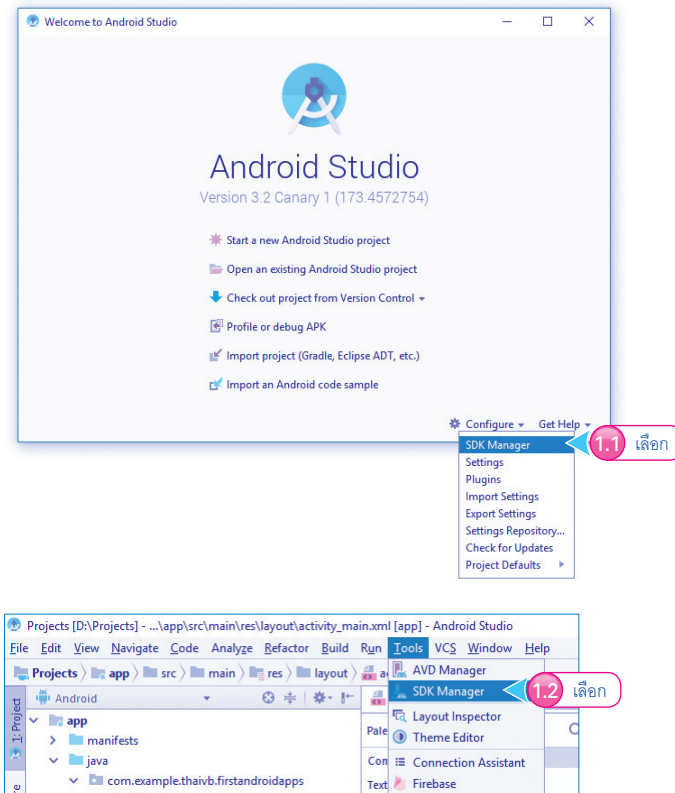
- ส่วนหัว คือ Android SDK เวอร์ชัน 8.1 (API 27) ติดตั้งมากับ Android Studio แล้ว
- ส่วนท้าย คือ Android SDK เวอร์ชัน 6.0 (API 23) ต้องติดตั้งเพิ่มเติมใหม่

ส่งผลให้แอปฯ ที่เราจะสร้างขึ้นมา สามารถรันได้ตั้งแต่ Android 6.0 เป็นต้นไปนั่นเอง ส่วน Android SDK ระหว่างทางคุณจะต้องติดตั้งหรือไม่ก็ได้ ขึ้นอยู่กับการทำงานของคุณ การติดตั้ง Android SDK เวอร์ชันอื่นๆ เพิ่มเติม มีขั้นตอนดังนี้

1. ที่หน้าจอแรกของ Android Studio ให้คุณเลือกรายการ Configure > SDK Manager หรือคลิกเมนู Tools > SDK Manager ดังรูปที่ 1-7

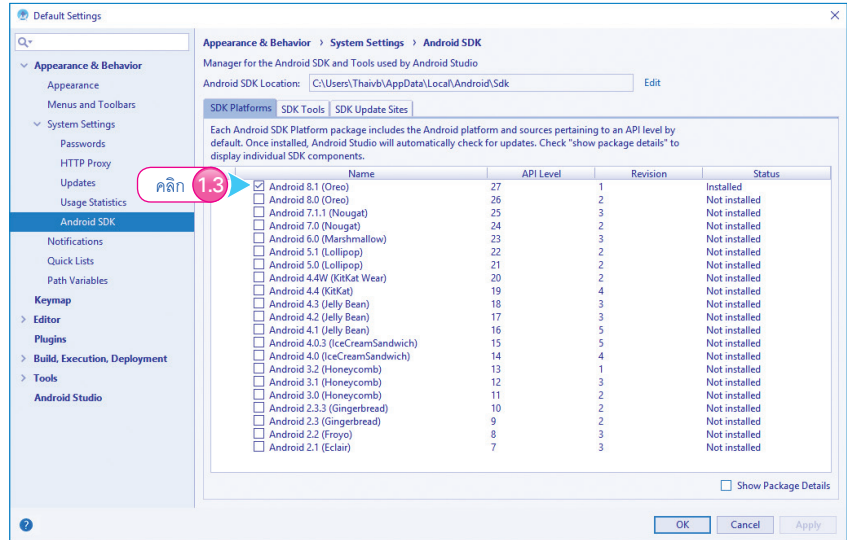
รูปที่ 1-7

แสดงวิธีการเปิด
SDK Manager



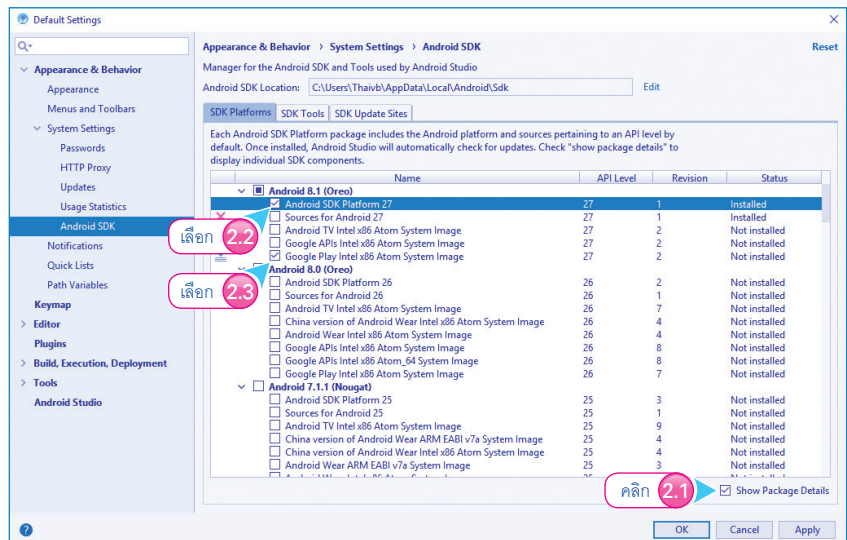
ในขณะที่แต่งหนังสือเล่มนี้อยู่ มีการติดตั้ง Android SDK เวอร์ชัน 8.1 (API 27) ให้ผู้เขียนโดยอัตโนมัติ (มีคำว่า Installed ต่อท้ายรายการ) ส่วนของผู้อ่านจะเป็นเวอร์ชันใหม่กว่าของผู้เขียน ดังรูปที่ 1-8

รูปที่ 1-8
แสดงหน้าจอเลือก
Android SDK

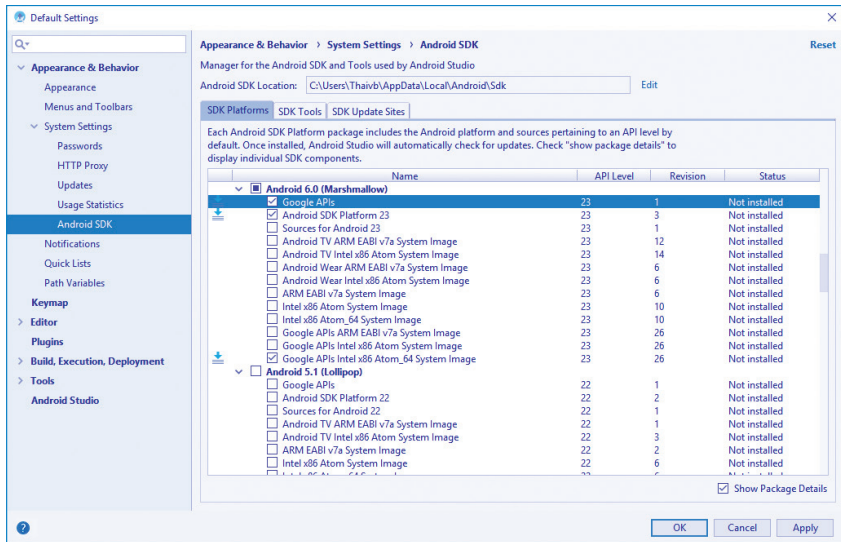


- คลิกตัวเลือก Show Package Details เพื่อแสดงรายละเอียดของ Android SDK แต่ละเวอร์ชันก่อน โดยคุณต้องติดตั้งอย่างน้อย 2 รายการหลัก คือ
 - Android SDK Platform คือ รายการหลักของแต่ละเวอร์ชัน
 - รายการที่ลงท้ายด้วยคำว่า System Image คือ Android Emulator สำหรับทดสอบโปรเจกต์ที่เราสร้างขึ้นมา มีข้อเสนอแนะว่าควรเลือกรายการที่มีคำว่า Google Play, x86 หรือ x64

รูปที่ 1-9
แสดงการเลือก
รายการให้กับ
เวอร์ชันส่วนหัว



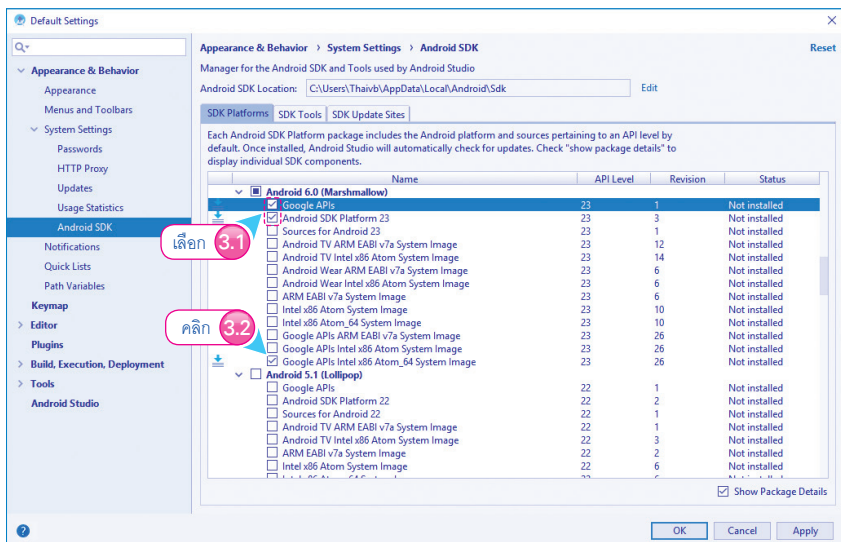
รูปที่ 1-9 (ต่อ)
แสดงการเลือก
รายการให้กับ
เวอร์ชันส่วนหัว



- ไอคอน ✖ หมายถึง รายการที่คุณต้องการถอดออกจากเครื่องของคุณ
- ไอคอน ⇩ หมายถึง รายการที่คุณต้องการดาวน์โหลดและติดตั้งเพิ่มเติม

3. ที่เวอร์ชันส่วนท้าย Android 6.0 ใช้หลักการเช่นเดียวกัน คือ เลือกรายการ Platform กับ System Image ดังรูปที่ 1-10

รูปที่ 1-10
แสดงการเลือก
รายการให้กับ
Android 6
(API 23)



4. คลิกแท็บ SDK Tools และรายการบังคับที่คุณต้องติดตั้ง ประกอบด้วย 6 รายการ คือ

- Android SDK Build-Tools
- Android SDK Platform-tools
- Android SDK Build-tools
- Android Emulator
- Google Play Services
- Support Repository เลือกติดตั้งทั้งหมด

ทั้ง 6 รายการข้างต้น คุณควรที่จะอัปเดตเป็นเวอร์ชันใหม่ล่าสุดเสมอ

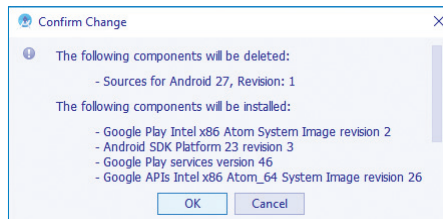
รูปที่ 1-11
แสดงการเลือก
รายการติดตั้ง
ในหมวด SDK
Tools



จากรูปที่ 1-11 หลังจากที่คุณเลือกรายการที่ต้องการดาวน์โหลดติดตั้งและถอดออกแล้ว ให้คุณคลิกปุ่ม **OK** เพื่อเข้าสู่ขั้นตอนต่อไป

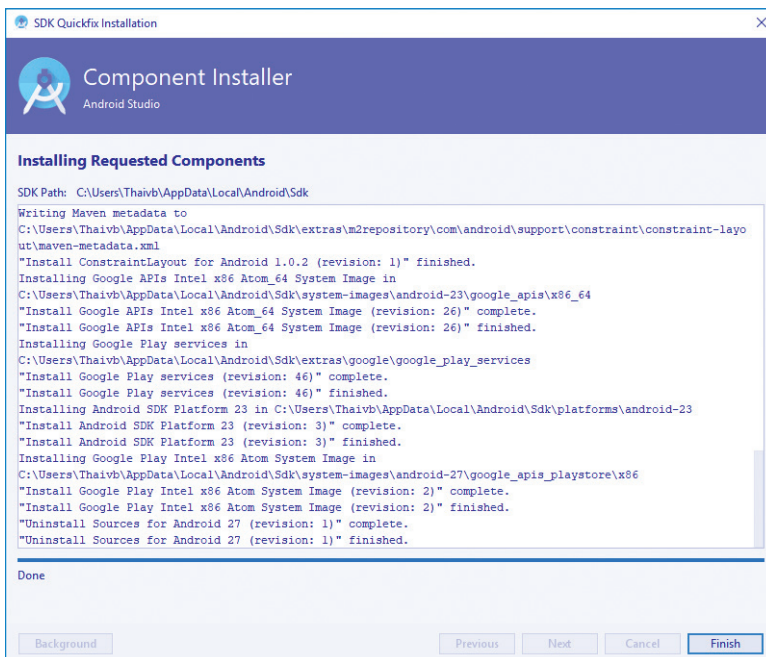
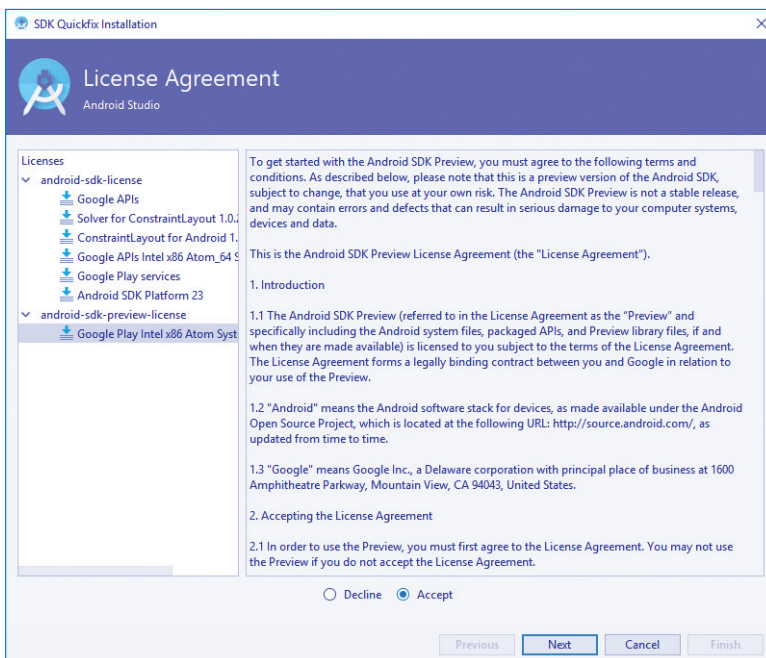
5. ท้ายที่สุดให้คุณตอบรับเงื่อนไขการใช้งาน รวดานโหลดและติดตั้งจนครบทุกรายการที่เลือกโดยอัตโนมัติ ทั้งหมด ดังรูปที่ 1-12

รูปที่ 1-12
แสดงการ
ดาวน์โหลดและ
ติดตั้งรายการ
ต่างๆ เสร็จ
สมบูรณ์แล้ว



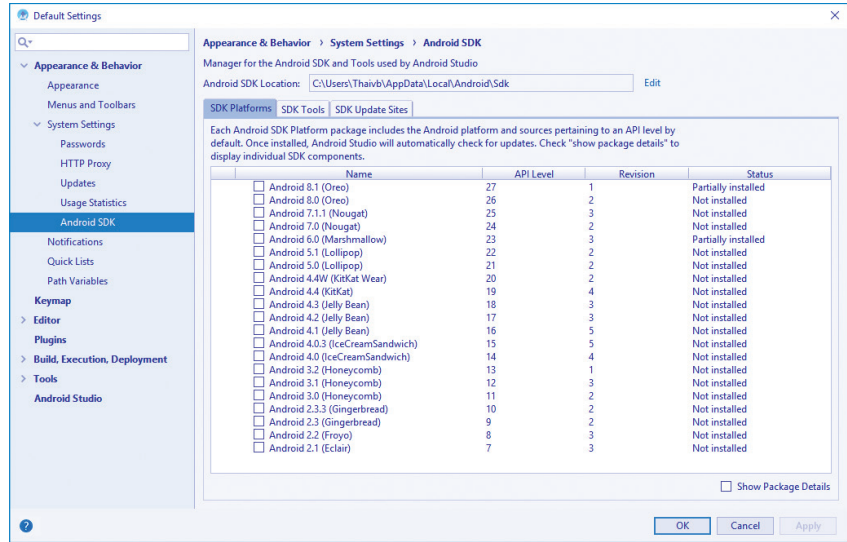
รูปที่ 1-12 (ต่อ)

แสดงการ
ดาวน์โหลดและ
ติดตั้งรายการ
ต่างๆ เสร็จ
สมบูรณ์แล้ว



จากรูปที่ 1-12 ขอให้รอจนขั้นตอนการดาวน์โหลดและติดตั้งรายการต่างๆ เสร็จสมบูรณ์ก็จะทำให้ Android Studio ของคุณ พร้อมพัฒนา Android Apps ที่มีการอัปเดตล่าสุดเท่าที่มีอยู่แล้ว รายการ Android SDK ที่มีการติดตั้งไปบ้างแล้วในแต่ละเวอร์ชันจะแสดงด้วยคำว่า "Partially installed"

รูปที่ 1-13
แสดงสถานะหลังจากติดตั้งเสร็จแล้ว



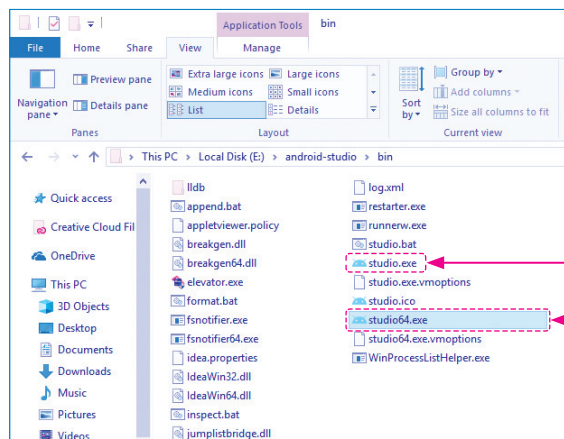
การใช้งาน Android Studio เวอร์ชัน Preview

ในกรณีที่คุณต้องการใช้งานโปรแกรม Android Studio เวอร์ชัน Preview มีข้อแนะนำเป็นแนวทางในขั้นต้นไว้ก่อนเลย ก็คือ

1. ไม่แนะนำให้ติดตั้งเวอร์ชัน Preview เพียงเวอร์ชันเดียว ควรที่จะติดตั้ง Android Studio เวอร์ชันที่เป็น Stable ไว้ด้วย เพื่อใช้สร้าง Android Apps ที่มีฟีเจอร์หลักและอยู่ในฐานะใช้งานได้จริงอย่างเป็นทางการ
2. โปรเจกต์ที่สร้างจากเวอร์ชัน Preview มีโอกาสเปิดในเวอร์ชัน Stable ไม่ได้ เพราะว่าฟีเจอร์ใหม่ๆ นั้นเอง คุณต้องรอเวอร์ชัน Stable ให้มีฟีเจอร์เท่าเทียมกัน

หลังจากที่คุณดาวน์โหลด Android Studio เวอร์ชัน Preview มาแล้ว ให้ขยายไฟล์แยกออกมาต่างหากไม่เกี่ยวกับ Android Studio เวอร์ชัน Stable แต่อย่างไร ในกรณีนี้ผู้เขียนแยกมาเก็บไว้ที่พาท E:\android-studio ดังรูปที่ 1-14

รูปที่ 1-14
แสดงรายการไฟล์ของ Android Studio เวอร์ชัน Preview



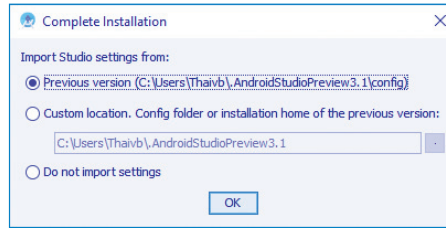
จากรูปที่ 1-14 วิธีการเปิดโปรแกรม Android Studio เวอร์ชัน Preview ทำได้โดยการเข้ามาที่โฟลเดอร์ bin เลือกเปิดได้ 2 รูปแบบ คือ

1. ดับเบิลคลิกที่ studio.exe เพื่อเปิดโปรแกรมแบบ 32 บิต
2. ดับเบิลคลิกที่ studio64.exe เพื่อเปิดโปรแกรมแบบ 64 บิต

การเปิดโปรแกรม Android Studio เวอร์ชัน Preview เฉพาะครั้งแรกจะมีการถามว่า ต้องการกำหนดค่า Config แบบใด ผู้เขียนแนะนำให้คลิกเลือก Previous version ใช้ค่าเดียวกับ Android Studio เวอร์ชัน Stable ที่เรามีอยู่เดิม

รูปที่ 1-15

แสดงการ
กำหนดค่า Config



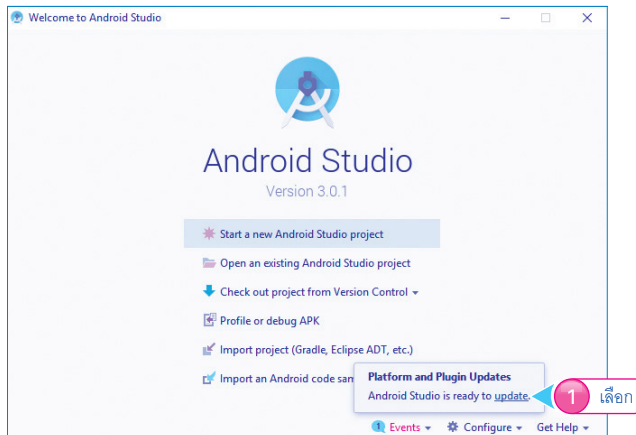
การอัปเดต Android Studio ให้เป็นเวอร์ชันใหม่ล่าสุด

ตัวโปรแกรม Android Studio จะมีการอัปเดตต่างๆ ออกมาอยู่เสมอ เพื่อปรับปรุงตัวโปรแกรมให้มีความสามารถใหม่ๆ อยู่ตลอดเวลา คุณสามารถตรวจสอบรายการอัปเดตได้ 2 ช่องทาง คือ

1. ที่หน้าแรกให้สังเกตที่ Events จะมีการแจ้งเตือนว่า มีรายการอัปเดตใหม่ ให้คลิกดูรายละเอียดที่แจ้งขึ้นมา อาจจะเป็นการเปลี่ยนเวอร์ชันใหม่, ปรับปรุงข้อผิดพลาดที่เกิดขึ้น, อัปเดตรายการเดิมที่เราติดตั้งไว้ ให้กลายเป็นเวอร์ชันใหม่ก็ได้

รูปที่ 1-16

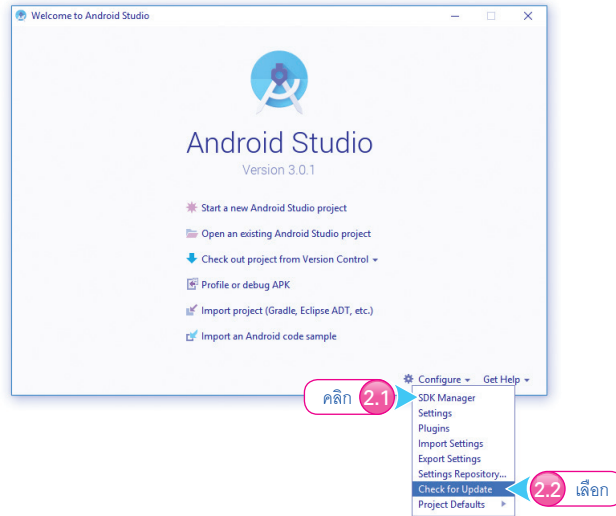
แสดงการแจ้ง
เตือนรายการ
อัปเดตปัจจุบัน



2. คลิกที่ **Configure > Check for Update** เพื่อตรวจสอบรายการอัปเดตด้วยตัวเอง อาจจะมีหรือไม่มีก็ได้ ดังรูปที่ 1-17

รูปที่ 1-17

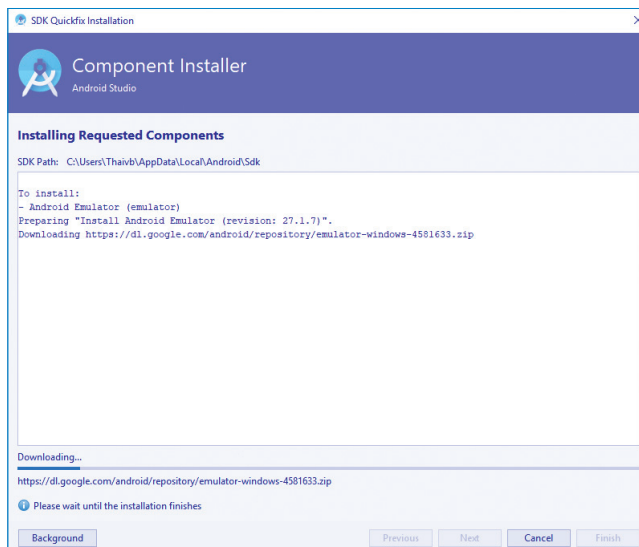
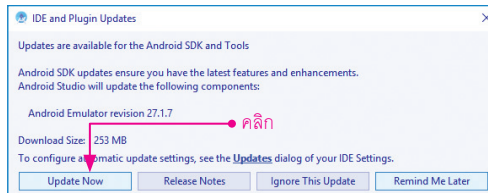
แสดงการ
ตรวจสอบรายการ
อัปเดต



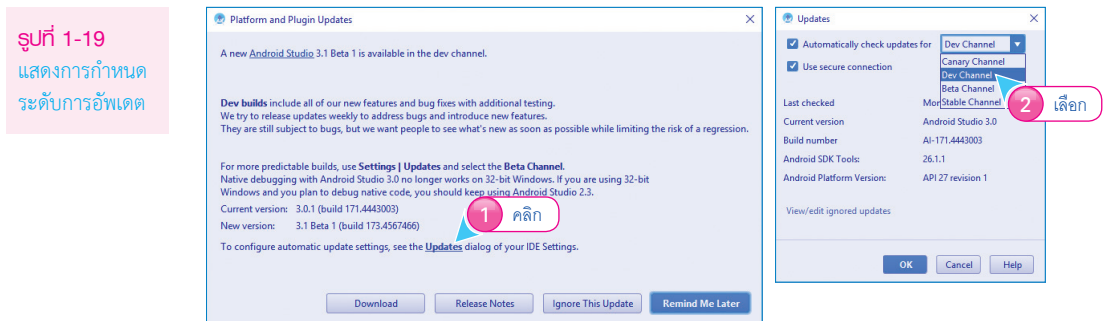
ในกรณีที่มียรายการอัปเดต ให้ทำตามหน้าจอที่ปรากฏขึ้นมา เพื่อดาวน์โหลดและติดตั้งโดยอัตโนมัติทั้งหมด ดังรูปที่ 1-18

รูปที่ 1-18

แสดงการ
ดาวน์โหลดและ
ติดตั้งอัปเดตเท่า
ที่มีอยู่



ในเมนู Check for Update ที่คลิกเลือก คุณสามารถคลิกที่ Updates เพื่อปรับปรุงระดับการแจ้งเตือนรายการอัปเดตให้เหมาะสมกับความต้องการของเรา ดังรูปที่ 1-19



1. **Canary Channel** ถ้าต้องการใช้งานหรือทดสอบฟีเจอร์ใหม่ๆ ก่อนใคร ต้องการความรวดเร็ว ตัวเลือกนี้เหมาะสมที่สุด แต่ก็มีความเสี่ยงสูงอย่างน้อย 2 ประการ คือ
 - ฟีเจอร์ที่ใช้อาจจะเกิดข้อผิดพลาดขึ้นก็ได้ เพราะเป็นฟีเจอร์ที่อยู่ในช่วงทดสอบ มีโอกาสที่จะเกิดการเปลี่ยนแปลงสูงมาก รวมไปถึงวิธีการเขียนโค้ด อาจจะต้องมีการแก้ไขอยู่เสมอ
 - ฟีเจอร์ที่ถูกเพิ่มเข้ามา อาจจะนำไปใช้งานจริงหรือไม่ใช้ในเวอร์ชันหลัก (Stable) ก็ได้
2. **Dev Channel** กับ **Beta Channel** ทั้ง 2 ระดับ เท่าที่ผู้เขียนลองใช้งานมีความใกล้เคียงกันมาก ฟีเจอร์ใหม่ๆ ที่เข้ามาในระดับนี้มีความแน่นอนมากกว่าระดับ Canary Channel มีความรวดเร็วและแน่นอนในระดับที่ผู้เขียนยอมรับได้
3. **Stable Channel** เป็นระดับการอัปเดตที่ถูกใช้ในเวอร์ชันหลัก สามารถนำไปใช้พัฒนา Android Apps ในงานจริงของคุณได้ เป็นการอัปเดตที่ช้าที่สุดแต่ก็มั่นคงที่สุดเช่นกัน

จากรูปที่ 1-25 เราได้โปรเจกต์วางๆ ใน Android Studio มาแล้ว ก็จะเข้าสู่ขั้นตอนการสร้างแอปฯ ของเรานั่นเอง

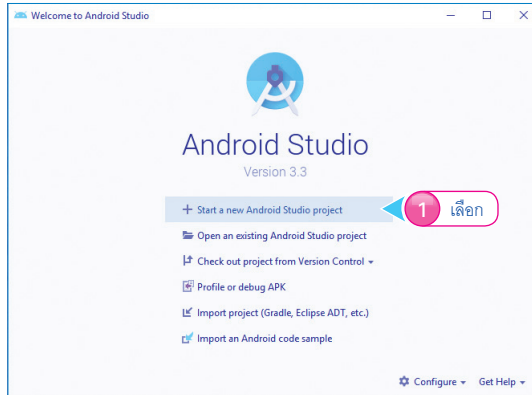
การสร้างโปรเจกต์ใน Android Studio ตามนโยบายใหม่ของ Google

การสร้างโปรเจกต์ใน Android Studio มีขั้นตอนดังนี้

1. ให้คุณคลิกเลือก Start a new Android Studio project เพื่อเริ่มต้นสร้างโปรเจกต์ Android ใหม่ ดังรูปที่ 1-20

รูปที่ 1-20

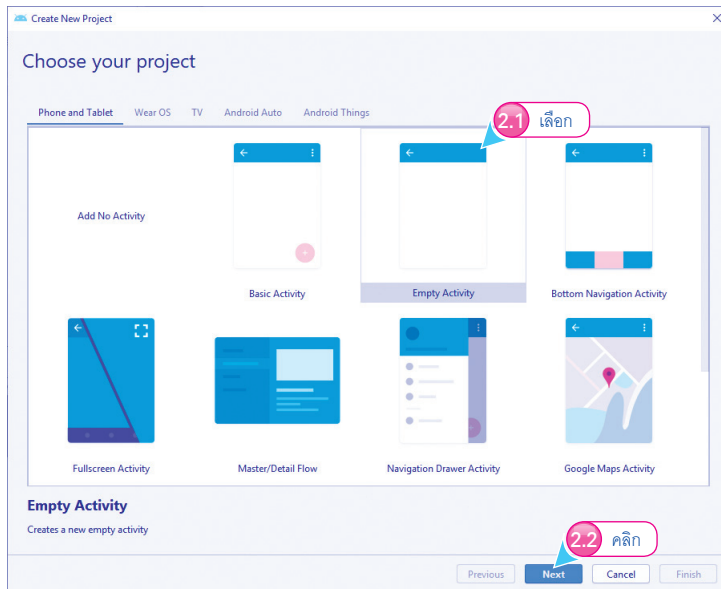
แสดงการเริ่มต้น
สร้างโปรเจกต์
Android



2. ในขั้นตอนนี้ผู้เขียนแนะนำให้เลือกสร้างโปรเจกต์แบบ Empty Activity เป็นโปรเจกต์ที่มีโครงสร้างพื้นฐานเหมาะสมกับผู้เริ่มต้นได้เป็นอย่างดี ดังรูปที่ 1-21

รูปที่ 1-21

แสดงการสร้าง
โปรเจกต์ว่างๆ



3. เป็นการกำหนดรายละเอียดให้กับโปรเจกต์ที่เราสร้างขึ้นมา โดยที่

- ช่อง **Name** หมายถึง ชื่อโปรเจกต์ Android Apps ของเรา
- ช่อง **Package name** หมายถึง ชื่อ Package ประจำโปรเจกต์ ใช้รูปแบบ URL กลับด้าน
- ช่อง **Language** หมายถึง ภาษาที่ต้องการใช้งาน ในกรณีนี้เลือกภาษา JAVA
- ช่อง **Minimum API Level** กำหนดให้รองรับ Android เวอร์ชันต่ำสุด คือ 6.0 (API 23) ตาม Android SDK ที่เราติดตั้งไว้ก่อนหน้านี้ ส่งผลให้แอปฯ ของเรา สามารถติดตั้งบนระบบปฏิบัติการ Android 6.0 (API 23) ขึ้นไป

ตั้งชื่อโปรเจกต์นี้ว่า FirstAndroidApps เก็บไว้ที่พาธ E:\Projects

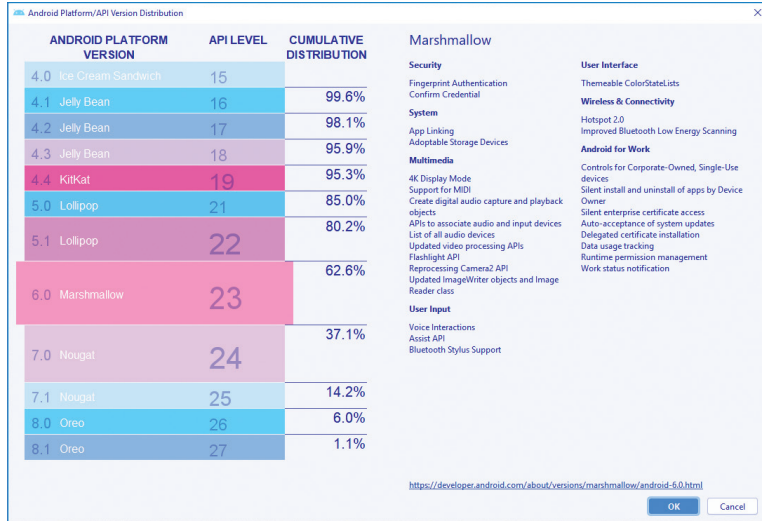
รูปที่ 1-22

แสดงการสร้าง
โปรเจกต์ขั้นต่ำ
Android 6
(API 23)

ความหมายก็คือ แอปฯ ของเราสามารถติดตั้งใน Android 6 (API 23) ขึ้นไป การเลือกเวอร์ชันต่ำสุดที่รองรับ เป็นเรื่องที่เราต้องให้ความสำคัญมากขึ้น เนื่องจากนโยบายใหม่ของ Google นั่นเอง ไม่มีใครทราบ ว่าวันเวลาไหนต้องเลือกอย่างไร หรือต้องเลือกเวอร์ชันไหน แต่ Google มีเครื่องมือช่วยตัดสินใจให้คุณ

ในกรณีที่ผู้อ่านต้องการข้อมูลเพิ่มเติม ให้คุณคลิกที่ Help me choose เพื่อแสดงสัดส่วนอุปกรณ์ Android แยกตามเวอร์ชัน ดังรูปที่ 1-23

รูปที่ 1-23
แสดงสัดส่วน
อุปกรณ์ Android
ในแต่ละเวอร์ชัน



จากรูปที่ 1-23 ในช่วงเวลาปัจจุบันของผู้เขียน สนใจ Android 6.0 (API 23) ขึ้นไป มีตัวเลขประมาณการหรือคาดการณ์หรือความน่าจะเป็นครอบคลุมอุปกรณ์ประมาณ 62.6% (ของคนที่ผู้อ่านจะแตกต่างไปจากรูปนี้)

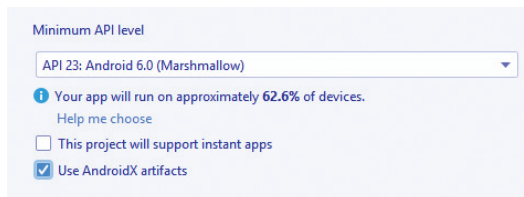
ในภายภาคหน้าผู้เขียนไม่ทราบว่าจะตัวเลขนี้เพิ่มขึ้นหรือลดลง ตัวเลขอาจจะกระโดดข้ามไปที่เวอร์ชัน 7 หรือ 8 เลยก็ได้ แต่ขอให้คุณยึดหลักไว้ว่า

อุปกรณ์ที่มีการติดตั้ง Android เวอร์ชันใหม่ๆ จะเข้าสู่ท้องตลาดเสมอ ทำให้สัดส่วนเวอร์ชันใหม่สูงขึ้นเรื่อยๆ ในขณะที่เวอร์ชันเก่าลดลงหรือหายไปจากท้องตลาดแน่นอน เพราะว่าอุปกรณ์เหล่านี้เสื่อมสภาพไปตามเวลา รวมถึงไม่รองรับฟีเจอร์ใหม่ๆ นั่นเอง

ผู้เขียนถือว่า นี่คือช่วงระยะเวลาที่เป็นรอยต่อระหว่างยุคเก่ากับยุคใหม่ ระบุไม่ได้ว่า เป็นวันเวลาใดที่ต้องปรับเปลี่ยน คุณอาจจะต้องขยับเวอร์ชันที่รองรับสูงขึ้นในปีหน้าเลยก็ได้ หรืออาจจะต้องรอคอยต่อไปก็ได้เช่นกัน

สำหรับตัวเลือก Use AndroidX artifacts ถ้าคุณเลือกตัวเลือกนี้ หมายความว่า คุณต้องการกำหนดให้โปรเจกต์ของคุณ ใช้ Package ตามโครงสร้างมาตรฐานใหม่ มีชื่อเรียกว่า "AndroidX"

รูปที่ 1-24
แสดง Package
ใหม่ที่ชื่อว่า
AndroidX



Minimum API level

API 23: Android 6.0 (Marshmallow)

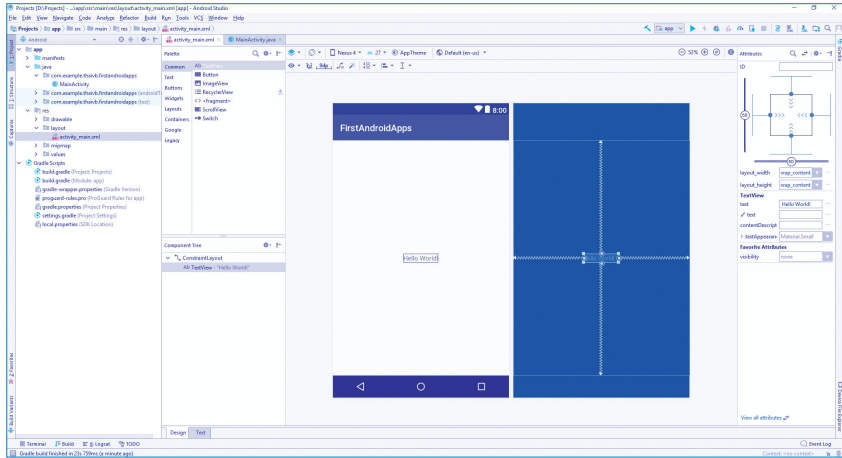
① Your app will run on approximately 62.6% of devices.
Help me choose

☐ This project will support instant apps

☒ Use AndroidX artifacts

4. จะได้โปรเจกต์ใหม่ ดังรูป

รูปที่ 1-25
แสดงโปรเจกต์
Android แบบ
ว่างใน Android
Studio

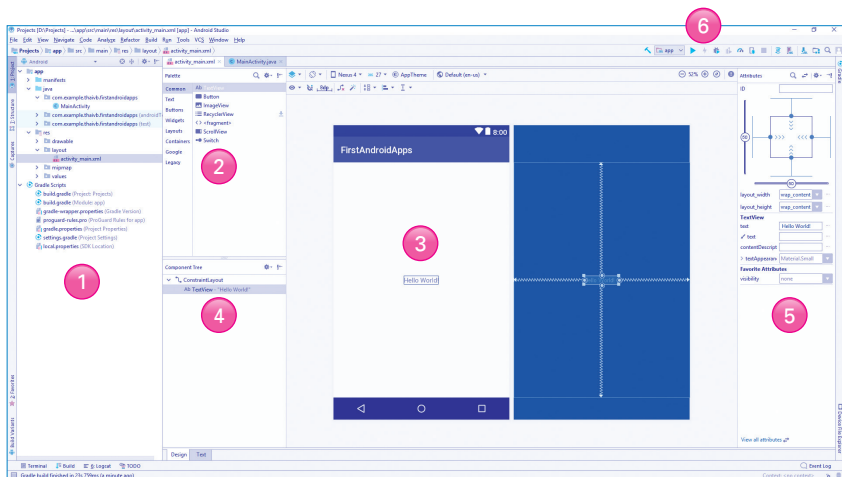


จากรูปที่ 1-25 เราได้โปรเจกต์ว่างๆ ใน Android Studio มาแล้ว ก็จะเข้าสู่ขั้นตอนการสร้างแอปฯ ของเรานั่นเอง

ทำความเข้าใจกับสภาพแวดล้อมของ Android Studio ในขั้นต้น

ให้คุณดูรูปต่อไปนี้ ประกอบคำอธิบาย

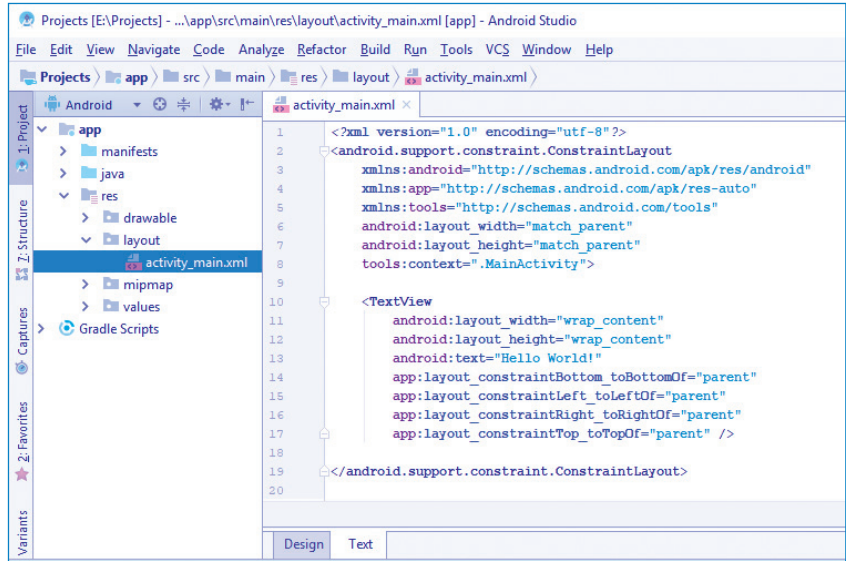
รูปที่ 1-26
แสดงสภาพ
แวดล้อมของ
Android Studio



- หมายเลข 1 แท็บ Project ทำหน้าที่แสดงโครงสร้างรายการไฟล์ทั้งหมดในโปรเจกต์ของคุณ
- หมายเลข 2 แถบ widget เก็บรายการ widget ประเภทต่าง ๆ ที่ใช้สำหรับออกแบบส่วนแสดงผลในแต่ละหน้าจอตามที่คุณต้องการ
- หมายเลข 3 ส่วนแสดงผลจำลอง ทำหน้าที่แสดงผลในแต่ละหน้าจอ ตามรายการไฟล์ที่คุณเลือกในโฟลเดอร์ \res\layout มี 2 โหมด คือ Design กับ Text

รูปที่ 1-27

ส่วนแสดงผล
จำลองในโหมด
Text ของส่วน
แสดงผลที่ชื่อว่า
activity_main.
xml



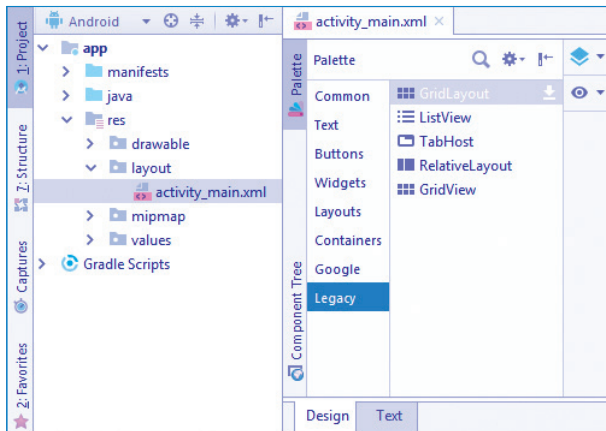
จากรูปที่ 1-27 ส่วนแสดงผลในแต่ละหน้าจอ ใช้รูปแบบของสคริปต์ XML เมื่อคุณออกแบบส่วนแสดงผลด้วย widget ต่างๆ คุณจะได้สคริปต์ XML โดยอัตโนมัติ ในทางกลับกันคุณสามารถแก้ไขสคริปต์ XML นี้โดยตรงได้เช่นกัน

- หมายเลข 4 แสดงโครงสร้างตามลำดับชั้น เป็นของส่วนแสดงผลปัจจุบันที่กำลังทำงานอยู่
- หมายเลข 5 หน้าต่าง Properties ทำหน้าที่แก้ไขแอตทริบิวต์ต่างๆ ของ widget ที่กำลังถูกโฟกัสในส่วนแสดงผลจำลอง (หมายเลข 3) หรือคุณสามารถแก้ไขที่สคริปต์ XML โดยตรงก็ได้เช่นกัน
- หมายเลข 6 แถบควบคุมทดสอบและหยุดโปรเจกต์ ทำหน้าที่ทดสอบโปรเจกต์ปัจจุบัน

ในปัจจุบัน Google กำหนดให้ widget ส่วนหนึ่งอยู่ในสถานะล้าสมัย ถูกจัดให้อยู่ในแท็บ Legacy ไม่ควรมานำใช้ออกแบบส่วนแสดงผล รายการที่มีอยู่เพื่อใช้ร่วมกับโปรเจกต์ Android ที่เกิดจากรุ่นเก่าเท่านั้น

รูปที่ 1-28

แสดง widget ที่
อยู่ในสถานะ
ล้าสมัย



การสร้าง Android Emulator สำหรับทดสอบโปรเจกต์

ขั้นตอนที่มีความสำคัญมากที่สุดในการพัฒนาแอปฯ ของคุณ ก็คือ การทดสอบโปรเจกต์ของคุณว่า มีการทำงานหรือมีผลลัพธ์เป็นไปตามที่คุณต้องการหรือไม่

ในปัจจุบันการทดสอบโปรเจกต์ Android Apps มีหลายวิธี ในขั้นตอนนี้ผู้เขียนนำเสนอการทดสอบโปรเจกต์ด้วย Android Emulator ซึ่งเป็นวิธีที่มักกับ Android Studio กล่าวคือ เป็นเพียงการทดสอบโปรเจกต์เบื้องต้นเท่านั้น ไม่ครอบคลุมทุกฟีเจอร์ของ Android SDK

เริ่มต้นให้คุณเปิด Android SDK ขึ้นมา เพื่อตรวจสอบก่อนว่า มีการติดตั้งตัว System Image ของ Android Emulator หรือไม่ ผู้เขียนให้หลักการในขั้นต้นที่ว่า คุณไม่จำเป็นต้องดาวน์โหลดและติดตั้งทุกเวอร์ชัน ให้คุณติดตั้งเพียงแค่เวอร์ชันสูงสุด และต่ำสุดก็พอ

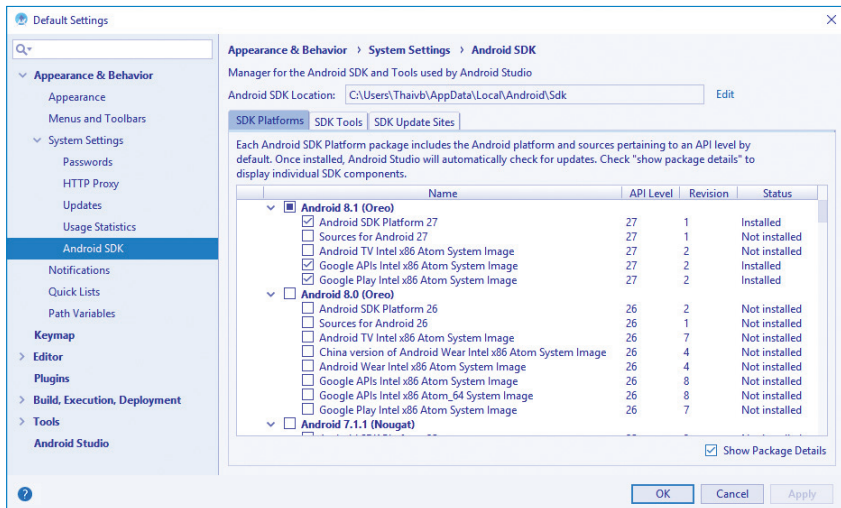
ในกรณีนี้ Android SDK เวอร์ชันสูงสุด คือ Android 8.1 (API 27) เวอร์ชันต่ำสุด คือ Android 6.0 (API 23)

- **System Image แบบ x86 หรือ x64** หมายถึง Android Emulator ที่อยู่ในรูปแบบของ CPU ตระกูล x86 กับ x64 (Intel หรือ AMD) ถ้าคุณเลือกติดตั้งแบบนี้ จะมีความเร็วมากกว่าแบบ CPU ตระกูล ARM
- **System Image แบบ ARM** หมายถึง Android Emulator ที่อยู่ในรูปแบบของ CPU ตระกูล ARM ซึ่งเป็น CPU เดียวกันกับที่อยู่ในมือถือหรือแท็บเล็ตบนเครื่องจริง

คุณควรที่จะติดตั้ง Android Emulator ที่มีคำว่า x86 หรือ x64 หรือคำว่า Google APIs เท่านั้น

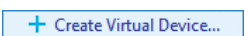
รูปที่ 1-29

แสดงการเลือกติดตั้ง System Image สำหรับ Android Emulator

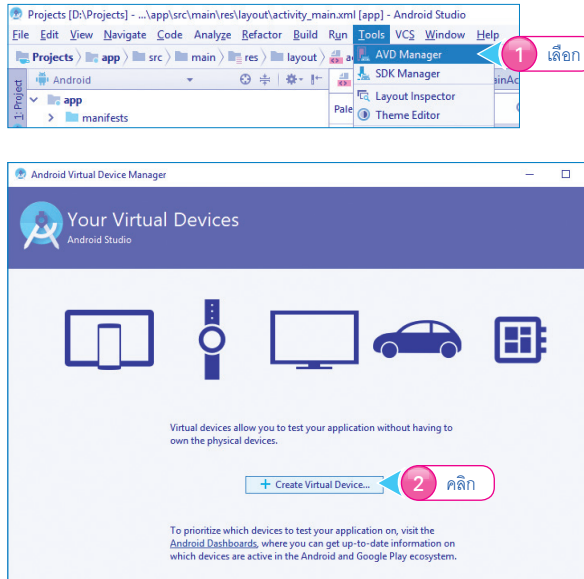


ขั้นตอนการสร้าง Android Emulator

หลังจากที่คุณดาวน์โหลดและติดตั้งตัว Android Emulator ให้กับ Android Studio แล้ว ก็จะเข้าสู่ขั้นตอนการสร้าง Android Emulator ขึ้นมาเพื่อทดสอบโปรเจกต์ของคุณ โดยมีขั้นตอนดังนี้

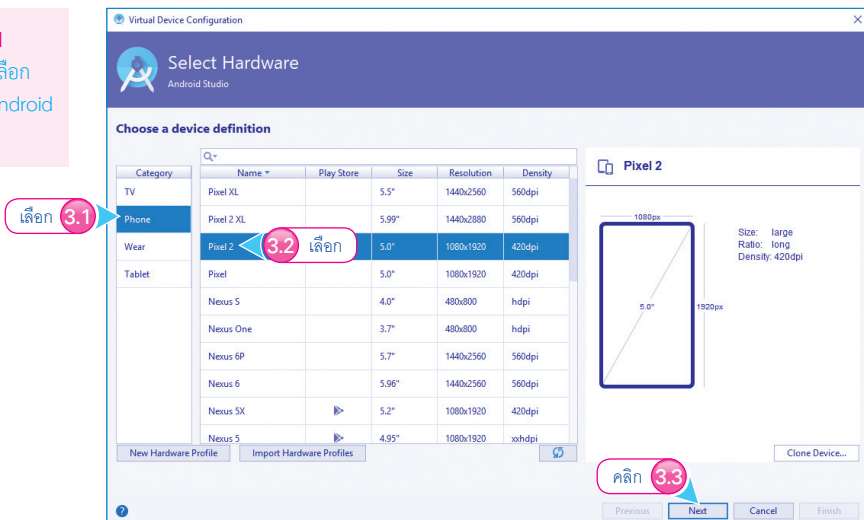
1. คลิกเมนู Tools > AVD Manager
2. คลิกปุ่ม  เพื่อเริ่มต้นสร้าง Android Emulator ดังรูปที่ 1-30

รูปที่ 1-30
แสดงการเริ่มต้น
สร้าง Android
Emulator

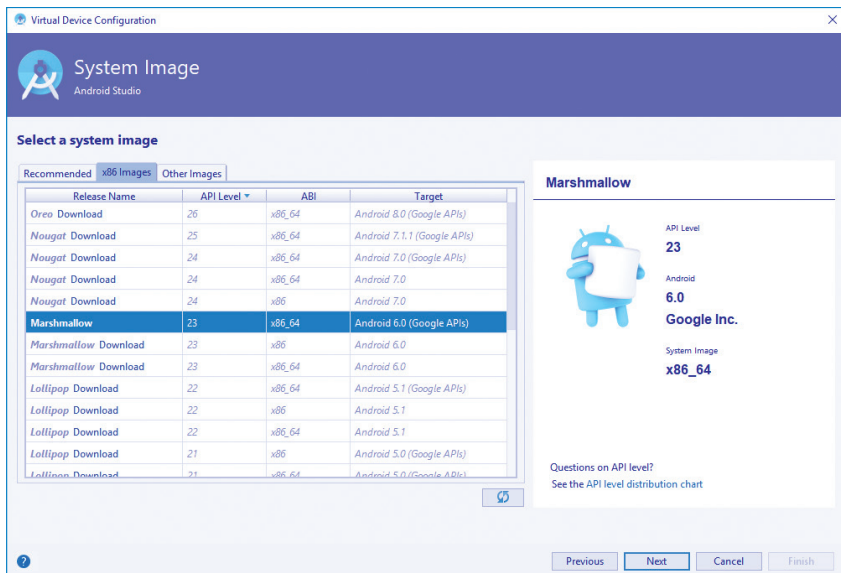


3. ในกรณีนี้เราต้องการสร้างตัว Android Emulator ในรูปแบบโทรศัพท์ ที่หัวข้อ Phone ให้เลือกความละเอียดของหน้าจอที่ต้องการสร้างขึ้นมา ผู้เขียนลองเลือกความละเอียด 1080*1920 (420 dpi) กับ Android 6.0 (API 23)

รูปที่ 1-31
แสดงการเลือก
ประเภท Android
Emulator



รูปที่ 1-31 (ต่อ)
แสดงการเลือก
ประเภท Android
Emulator

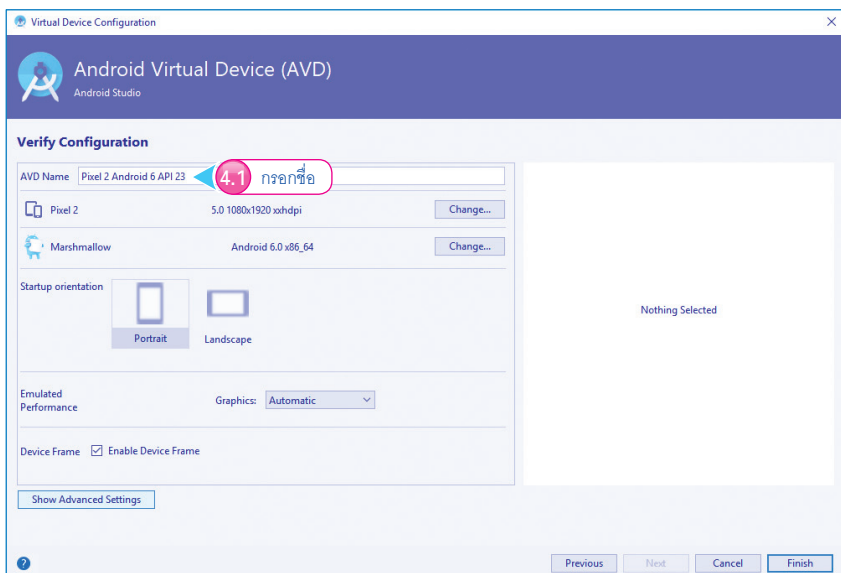


จากรูปที่ 1-31 คุณสามารถคลิกที่ download เพื่อดาวน์โหลด Android SDK ตามเวอร์ชันที่เห็นได้อีกด้วย

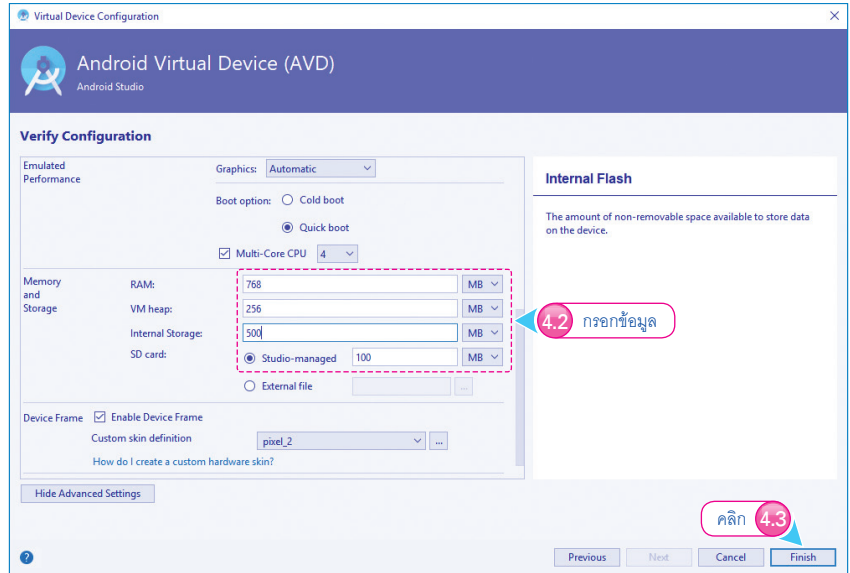
4. การปรับแต่งขั้นต้นที่คุณต้องกำหนด ได้แก่

- ชื่อ กำหนดได้ที่ช่อง AVD Name มีข้อเสนอแนะว่า ให้ตั้งชื่อตามเวอร์ชันของระบบปฏิบัติการ Android ในกรณีนี้ผู้เขียนต้องการสร้าง Android Emulator เวอร์ชัน 6 จึงตั้งชื่อว่า Android 6 (API 23)
- หน่วยความจำ กำหนดได้ที่ช่อง RAM เราไม่จำเป็นต้องกำหนดค่ามากเกินไป เท่าที่ผู้เขียนทดสอบ กำหนดระดับ 512-1024 MB ก็เพียงพอต่อการทดสอบโปรเจกต์ของคุณแล้ว

รูปที่ 1-32
แสดงการปรับแต่ง
Android
Emulator ที่
กำลังจะสร้าง
ขึ้นมา



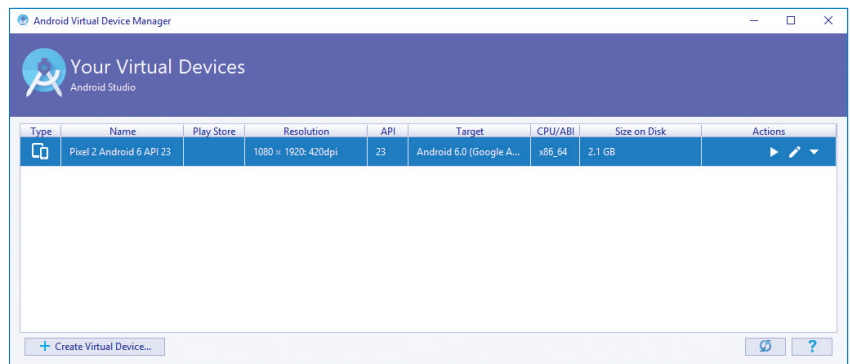
รูปที่ 1-32 (ต่อ)
แสดงการปรับแต่ง
Android
Emulator ที่
กำลังจะสร้าง
ขึ้นมา



จากรูปที่ 1-32 ให้คุณคลิกปุ่ม **Show Advanced Settings** เพื่อปรับแต่งตัว Android Emulator ที่กำลังจะสร้างขึ้นมา

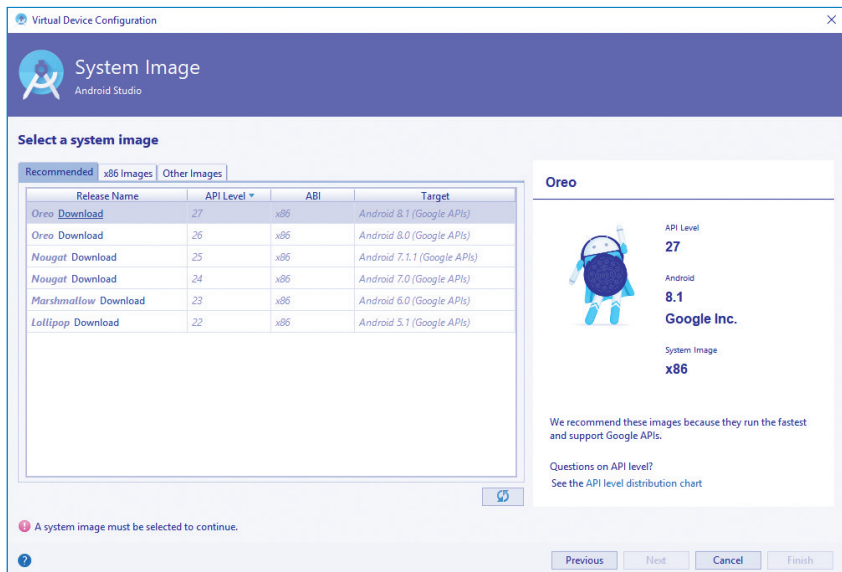
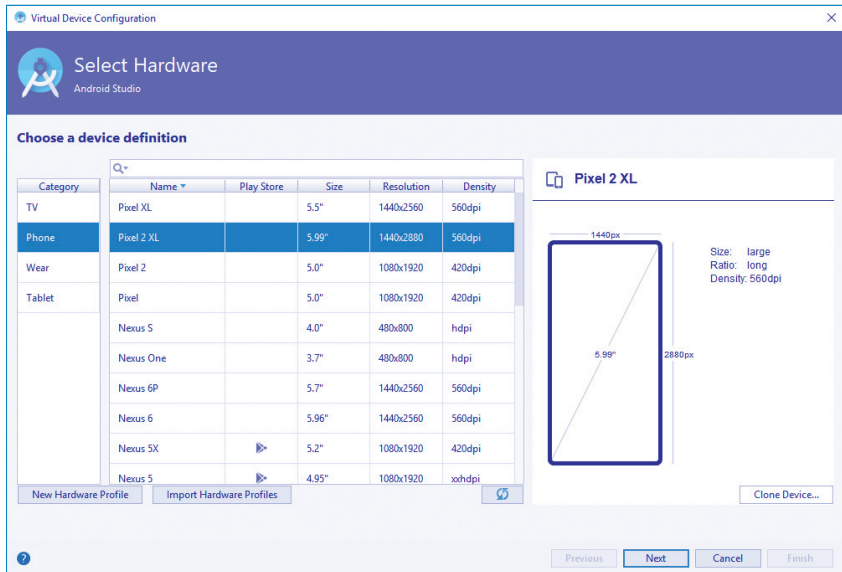
5. ท้ายที่สุด ตอนนี้ผู้เขียนได้ Android Emulator เวอร์ชัน 6 (API 23) สำหรับทดสอบโปรเจกต์แล้ว ดังรูปที่ 1-33

รูปที่ 1-33
แสดงการกำหนด
แพลตฟอร์มให้
กับ Android
Emulator

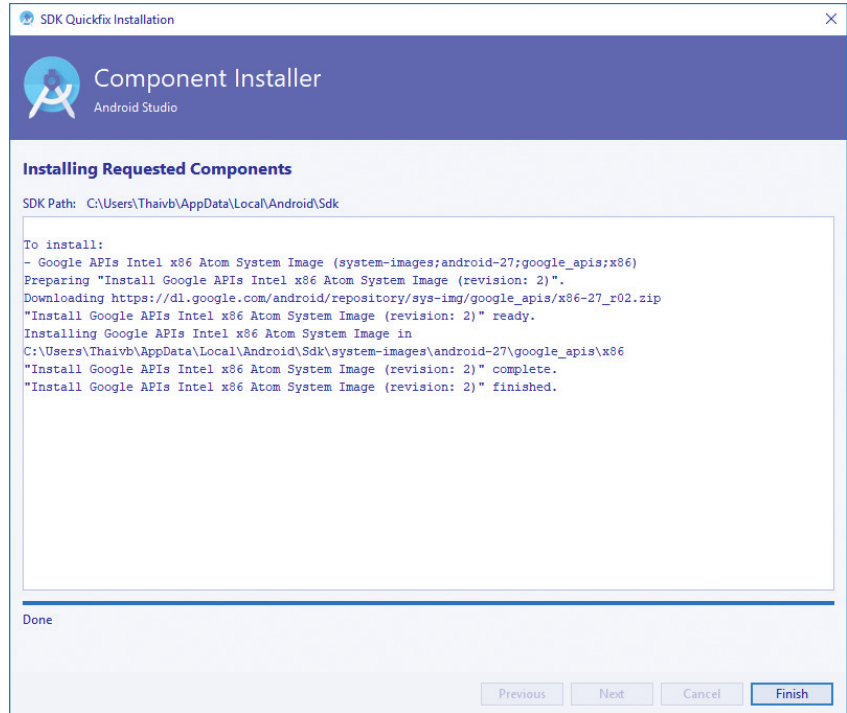


6. ผู้เขียนสร้าง Android Emulator เพิ่มขึ้นอีก 1 รายการ โดยกำหนดให้เป็นเวอร์ชัน 8.1 (API 27) เพื่อให้ครอบคลุมการทดสอบโปรเจกต์ของเรา กำหนดความละเอียดหน้าจอแตกต่างกัน คือ 1440*2880 (560 dpi) เพื่อให้เกิดข้อเปรียบเทียบในหน้าจอที่ไม่เหมือนกัน ดังรูปที่ 1-34

รูปที่ 1-34
แสดงการสร้าง
Android
Emulator
เวอร์ชัน 8.1

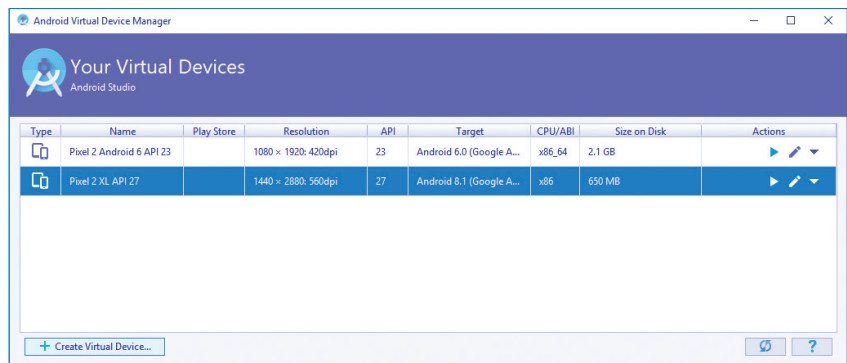


รูปที่ 1-34 (ต่อ)
แสดงการสร้าง
Android
Emulator
เวอร์ชัน 8.1



7. ท้ายที่สุด ผู้เขียนสร้าง Android Emulator ทั้งสิ้น 2 รายการ คือ เวอร์ชัน 6.0 (API 23) และเวอร์ชัน 8.1 (API 27) คุณจะสร้างครบทุกเวอร์ชันหรือไม่ก็ได้ ผู้เขียนเลือกเฉพาะเวอร์ชันหัว-ท้ายก็ครอบคลุมเช่นกัน

รูปที่ 1-35
แสดง Android
Emulator เท่า
ที่มีอยู่

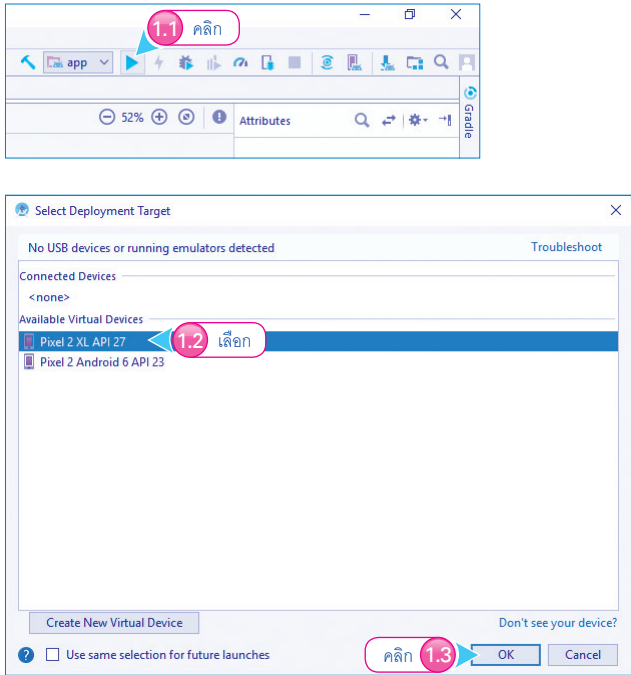


การทดสอบโปรเจกต์ด้วย Android Emulator

หลังจากที่คุณสร้าง Android Emulator ขึ้นมาแล้ว เราจะมาลองทดสอบโปรเจกต์ของเรา โดยมีคำแนะนำในขั้นต้นว่า ในการพัฒนาแอปฯ จริง ๆ Android Emulator "ไม่สามารถทดสอบทุกฟีเจอร์" เป็นเพียงการทดสอบโปรเจกต์ในขั้นต้นเท่านั้น ซึ่งมีขั้นตอนดังนี้

1. คลิกปุ่ม ▶ เพื่อเริ่มต้นทดสอบโปรเจกต์ปัจจุบัน ดังรูปที่ 1-36

รูปที่ 1-36
แสดงการเลือก
Android
Emulator ที่
ต้องการทดสอบ



จากรูปที่ 1-36 รายชื่อที่ปรากฏขึ้นมา เป็นไปตาม Android Emulator ที่คุณสร้างขึ้นมาในขั้นตอนที่แล้ว ผู้เขียนลองทดสอบโปรเจกต์ปัจจุบันด้วย Android 8.1 (API 27)

ในกรณีที่มีการเปิด Android Emulator ไว้อยู่แล้ว คุณสามารถคลิกเลือก Connected Devices เพื่อทดสอบโปรเจกต์ปัจจุบันกับ Android Emulator เดิมที่เปิดค้างไว้อยู่ได้เช่นกัน