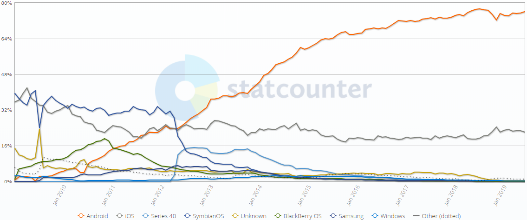
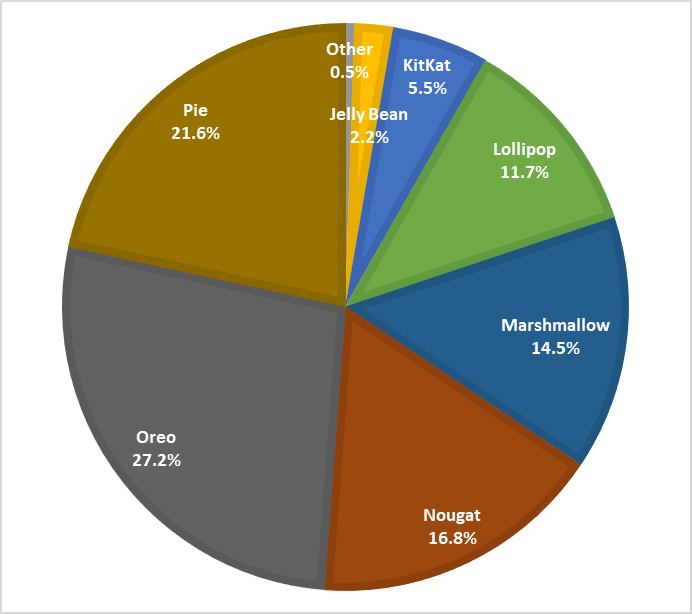
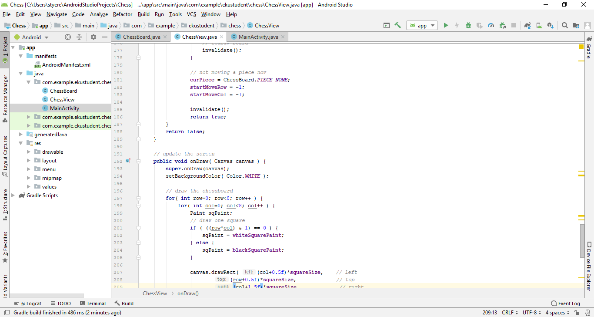
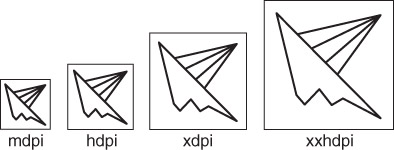
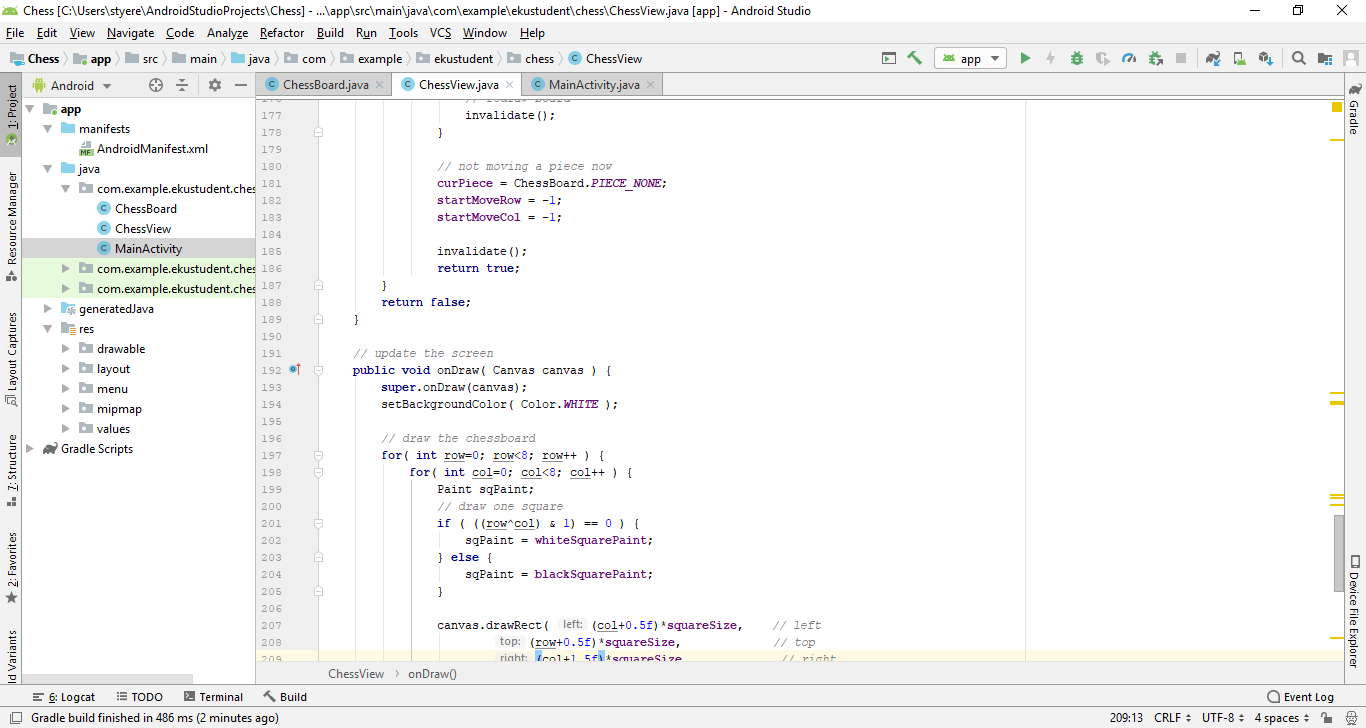
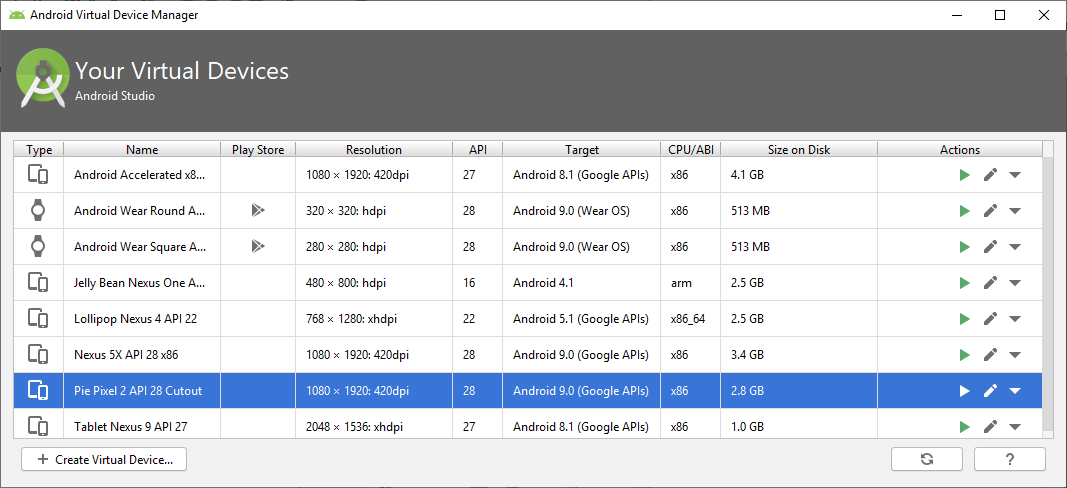
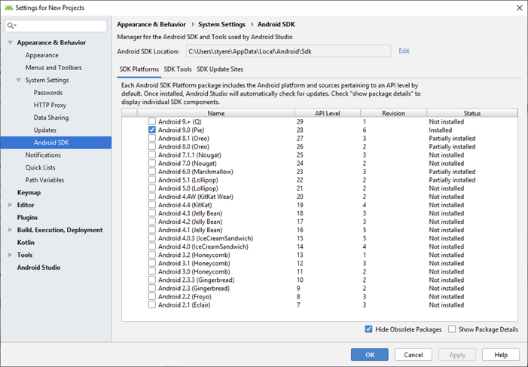
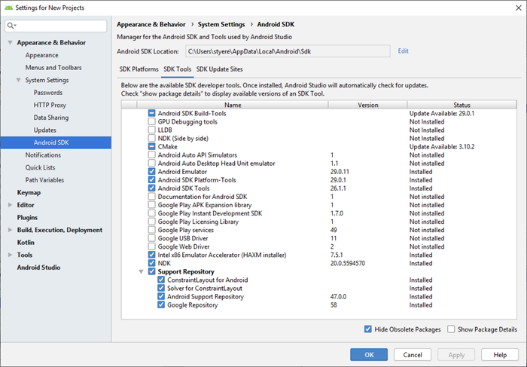
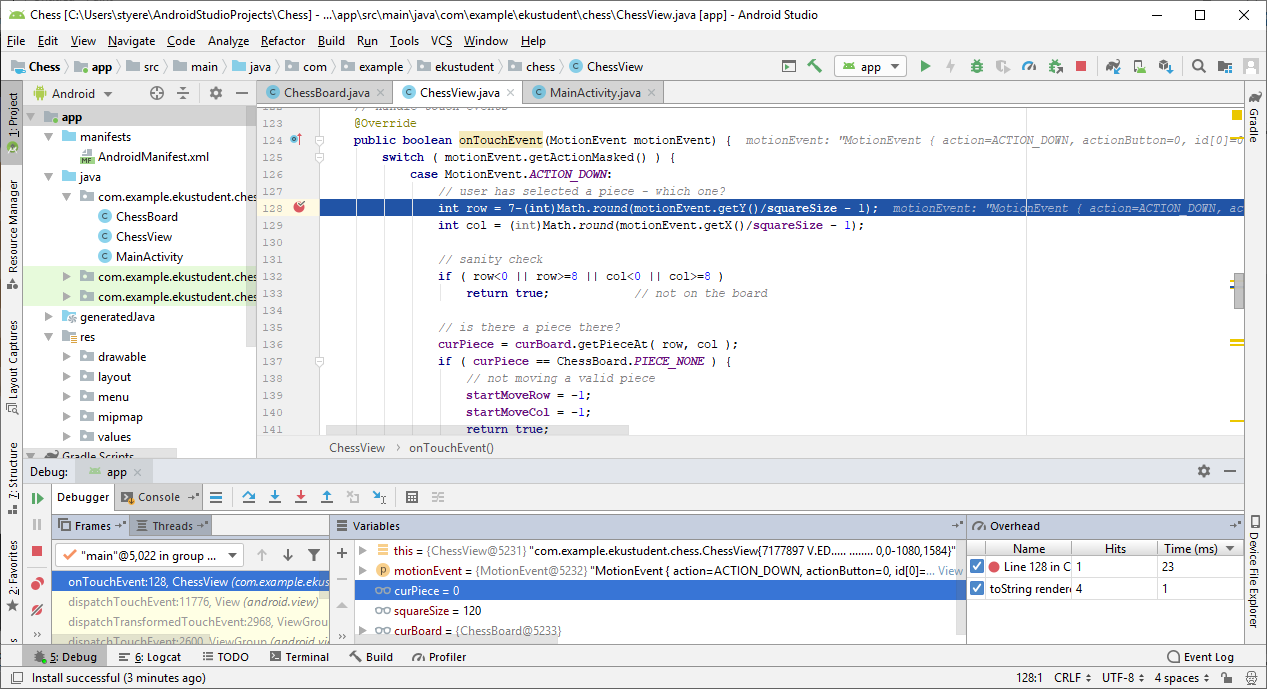
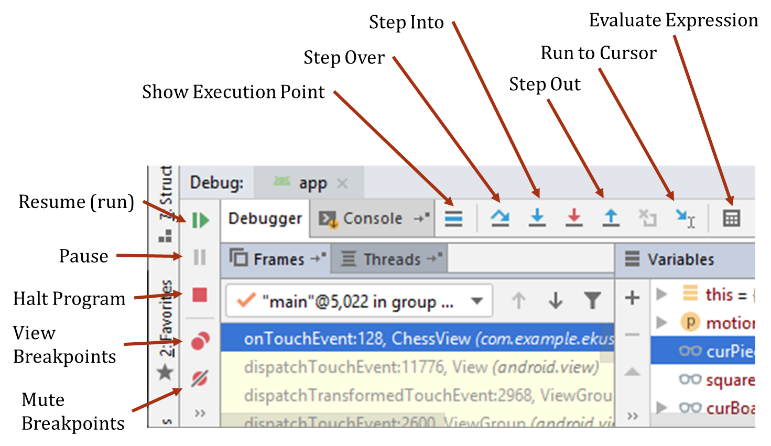
* Android smartphone share
  + Roughly 76% of smartphones are based on Android
  + <http://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-200901-201906>
  + Has Smartphone, tablet, TV, auto, and wearable versions
* Android History
  + First android phone released in 2008
  + 2003 – Android, Inc. founded by Andy Rubin, Rich Miner, Nick Sears, and Chris White
    - Wanted to have smarter mobile devices
  + 2005 – Acquired by Google
    - Goal of abstracting hardware features, yet maintain flexibility to tune devices
    - Supports a wide variety of displays, etc.
  + Android 3.0 added support for tablets
  + API level – Android version (Table 1-1)

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Version** | **Release Date** | **API** |
|  | 1 | September 23, 2008 | 1 |
|  | 1.1 | February 9, 2009 | 2 |
| Cupcake | 1.5 | April 27, 2009 | 3 |
| Donut | 1.6 | September 15, 2009 | 4 |
| Eclair | 2.0–2.1 | October 26, 2009 | 5–7 |
| Froyo | 2.2–2.2.3 | May 20, 2010 | 8 |
| Gingerbread | 2.3–2.3.7 | December 6, 2010 | 9–10 |
| Honeycomb | 3.0–3.2.6 | February 22, 2011 | 11–13 |
| Ice Cream Sandwich | 4.0–4.0.4 | October 18, 2011 | 14–15 |
| Jelly Bean | 4.1–4.3.1 | July 9, 2012 | 16–18 |
| KitKat | 4.4–4.4.4 | October 31, 2013 | 19–20 |
| Lollipop | 5.0–5.1.1 | November 12, 2014 | 21–22 |
| Marshmallow | 6.0–6.0.1 | October 5, 2015 | 23 |
| Nougat | 7.0 – 7.1.2 | August 22, 2016 | 24-25 |
| Oreo | 8.0-8.1 | August 21, 2017 | 26-27 |
| Pie | 9.0 | August 6, 2018 | 28 |
| Q | 10.0 | ?? (In beta) | 29 |

* + Apps can specify the min, max, and preferred API levels
    - My examples are mostly min API 16 (4.1) or 22 (5.1)
    - Generally don’t specify max API
    - Android Q is expected to warn users if they install an app designed for Lollipop or earlier
    - KitKat is the oldest supported version
  + Versions in Current Use (July 2019)
    - <https://www.appbrain.com/stats/top-android-sdk-versions>
  + Based on the Linux kernel, but is a distinct project
  + Android is basically single-user, but uses User ID to provide protection between apps
* Java vs. Android Java
  + Java and Kotlin are the primary languages for Android Studio development
  + Java on Android:
    - .class file are then converted to Dalvik bytecode (.dex)
    - Android 5.0 uses ART (Android RunTime) and compiles apps when they are installed (7.0 added improvements to make code faster)
    - Incremental Compilation – Only redo what has changed
  + Many libraries (ArrayList, etc) are common, but others are not
    - Abstract Windows Toolkit and Swing are not supported by Android
    - Android has a number of dedicated classes (View, AsyncTask, etc.)
* Kotlin
  + Approved as an official Android development language May 2017
  + Some Differences
    - More concise
    - Compatible with Java (can run on the JVM)
    - Designed to avoid null pointer exceptions
    - No raw types (int)
* Android Studio
  + Current version is 3.4.2
  + <https://developer.android.com/studio/index.html>
  + Components
    - Java Editor
    - Layout Editor
      * Graphical layout editor, can also edit text XML layout files
    - Android SDK
      * Collection of libraries, etc.
    - AVD – Android Virtual Devices
      * Emulated devices that can be used to test apps
    - Gradle – Build system
  + Can also develop apps in Visual Studio, other software
    - May use C++ or other languages
* Creating a new app
  + File -> New -> New App
  + This will also be an option on the start screen if you don’t have an active project
  + Note: In the lab, create projects on C: and copy them elsewhere when done
    - USB or network drives will be noticeably slower
* AndroidX
  + New support library stuff will be here, old versions of the library are deprecated
* Basics of App Design
  + Start with the idea
  + Useful to create prototypes (including drawings on paper) to plan the user interface
  + Rules for forward compatibility
    - Don’t use internal or unsupported APIs
    - Don’t directly manipulate settings without asking
    - Future versions may constrain settings
    - Limit # of layouts
    - Don’t assume given hardware is necessarily supported
  + Robustness
    - Ensure device orientation don’t disrupt the application
    - Use Android libraries instead of Java libraries
    - Take care of memory allocation
    - Try to reuse objects instead of reallocating
* App structure
  + Much of this is created automatically
  + Android Manifest
    - Application-wide settings
      * Name, start icon, start activity, required permissions, required hardware
  + Java source code
    - Activity classes and other classes
  + Drawable resources
    - Typically measured in device-independent pixels (dp)
      * Equivalent to 160 pixels/inch
    - Common to have different sizes of images depending on the screen size
      * mdpi: 1.0x (Example: 100x100)
      * hdpi: 1.5x (Example: 150x150)
      * xhdpi: 2.0x (Example: 200x200)
      * xxhdpi: 3.0x (Example: 300x300)
      * nodpi: Not resized when loaded
  + Mipmap
    - Holds the application icon
    - Creating an icon
      * Right click on “app” -> “New” -> “Image Asset”
      * Select an image under “Source Asset”
  + Layout XML files
    - Describes the location of items on the screen
    - Activities will handle buttons, etc.
  + Value Resources
    - Strings, colors, dimensions, styles
    - Can have multiple versions for different layouts and languages
  + Gradle Scripts
    - Rules for building an application
    - Includes Min/Target SDK versions
* Running apps on an Emulator
  + Create and start AVDs from the AVD Manager
    - Can choose screen size, Android version
  + HAXM
    - Hardware Accelerated Execution Manager
    - Speeds up x86-based emulators
      * ARM emulators will be ~10x slower
    - Can be installed through the SDK Manager
    - You must enable VT-x in BIOS before installing HAXM
    - HAXM is not compatible with other virtual machine software
      * Note: Windows Phone Emulator uses Hyper-V
      * Avast antivirus also appears to create conflicts
    - Recommend setting memory to 2 GB or more
      * Can be changed using the installer
      * Some AVDs expect 1.5+ GB of memory, so don’t skimp
      * HAXM doesn’t actually allocate memory until the emulator starts
* SDK Manager
  + Can install various platforms and tools
* Submitting projects
  + Make sure it will run on Android 5.1 (Lollipop, API 22) device or earlier
  + Clean the project (eliminates a lot of temporary files)
  + Zip the entire project directory
  + Submit the zip file
* Debugging
  + Start with run debug
  + Side of Debugging Window
    - Resume (run)
    - Pause
    - Halt program
    - View Breakpoints
    - Mute Breakpoints
  + Top of debugging window
    - Show Execution Point
    - Step Over
    - Step Into
    - Force Step Into
    - Step Out
    - Run to Cursor
    - Evaluate Expression
  + Breakpoints
    - Tell the app to stop at the given location
    - Conditional Breakpoint
      * Right-click on a breakpoint to add a condition such as if a variable has a particular value
* Profiling
  + View > Tool Windows > Android Profiler
  + Allows you to see where your app is slow or using resources
* Log – Collect debug output while running the app
  + static final String TAG = "MyActivity";
    - Tag is usually per class or activity
    - Not required, but recommended
  + Log.v( TAG, “Verbose Message” )
  + Log.d( TAG, “Debug Message” )
  + Log.i( TAG, “Information Message” )
  + Log.w( TAG, “Warning Message” )
  + Log.e( TAG, “Error Message” )
  + Good to remove them before releasing your app
  + Use “//” or an if statement so they can be restored later
* Publishing Apps
  + <https://developer.android.com/studio/publish/>
  + Preparation
    - Remove Log and other debug entries
    - Test it
    - Prepare any resources or servers needed
  + Releasing through Google Play
    - Create developer account ($25 one time fee)
    - Prepare screenshots and other promotional materials
    - Must have target API of 28 or higher
    - build.gradle (Module: app) – Also contains the version number (updates)
  + Can release through email or a website