

Typical hardware capabilities

- Mobile telephony
- Sensors and actuators
 - Magnetometer, accelerometer/gyro, temperature (and humidity) sensors, proximity/light sensors, vibrator and haptic feedback, camera (front and back) and flash, vibrate motor
- Location detector
 - GPS (fine – to a few meters), cell grid triangulation (to 100 meters), IP network location (suburb / city)
- CPU (multi-core)/ GPU (optimised to run on battery)
 - CPU + GPU on single die; Quad core CPUs now common; GPU (16-core or 32-core); ~ 75% of the graphics power of a PS3; entire phone feature set on single chip
- Touch screen
 - Responds to conductive surface (like a human finger) – capacitance change – electrical properties of the human body; ideal for gesture / multi-touch UI / can glide
 - Not suitable in some environments (e.g. manufacturing)

Typical hardware capabilities

- Networking:
 - WiFi, Bluetooth, 3G, 4G
- High resolution battery-efficient display
 - Bright / full colour - AMOLED, OLED, LCD, different resolutions are used, e.g., eXtra High density (640 x 960 @ 320 dpi), physical width often ~ 2 inches, height often between 3 - 5 inches
- Battery
 - Lithium-Ion, generally around 1500 - 3200 mAh battery pack, will last 1.5 days for light use, hours for heavy use (e.g., internet/game play/video)
- Memory (onboard + SD Card)
 - Persistent Solid State Storage (flash, SD card, special memory for OS and drivers, volatile (RAM up to 4 Gb, shared with GPU), memory read/write speed is a constraint

What are the constraints?

- As a developer you need to:
 - Know typical usage (minutes or hours)
 - Understand sensors (efficiency / accuracy / reliability / battery drain)
 - Know network is unreliable and offers erratic speeds
 - Be aware that the phone may/will/can ring
 - Display size, DPI & processor speed are variable
- You must develop apps that:
 - Use only a few MB of memory
 - Are small in size
 - Store data as efficiently as possible
 - Can work with sensors where precision is not identical across devices (even iPhone has 3 generations in wide-spread use)
 - Can work with a range of memory, CPU, and GPU capabilities

The biggest constraint of all?

- What happens to your app when the phone rings ?
 - Your app is put in the background by O/S
 - What happens to *the state* of your app ?
 - e.g. data entry ?
- What happens when the phone call ends?
 - Does your app reset?
 - Does it remember state?
 - It depends on you.

Mobile Operating System

- Operating System == Resource Manager
- Different to PC operating systems
- The life cycle of an application is very tightly controlled by OS
 - If an app is a resource hog – it will get killed (or suspended) by the OS
 - OS allocates memory to apps, but aggressively deallocates it
 - If a view/image is not in the foreground/running – then it will be destroyed.
- Tight resource constraints on mobile
 - Low powered (Batt. 1200 - 1500 mAh)
 - Limited RAM (256 MB - 2 GB)
 - Relatively small disk (8GB - 16GB - 32GB)
 - Small (yet power draining) display
 - Network access - expensive yet needed!
 - If phone rings – call gets highest priority

Mobile OSs are cranky

- If you request a network connection -- but have not used it for a while,
 - OS will power down network hardware
 - OS may close the connection / port
- If the device goes to sleep (default is few minutes) then the memory allocated to app is deallocated.

Mobile File Systems

- Designed to work with Flash memory
 - Flash memory is designed for max. 10k writes
 - File system is optimised to ensure that the same block does not receive too many writes (known as wear levelling)
 - Write operation -> (erase first), then write data
 - Read speed ~ 6 MB/sec, Write ~ 3 MB/sec
- I/O speed optimisation critical in games

When building mobile apps...

- **Recall: phones ring**
 - Save state regularly (in fact, be paranoid)
- Assume resources are allocated **reluctantly**
- Assume **erratic** network connectivity
- Assume **slow** read/write speed to Flash card
- Use RAM carefully, **minimise** sensor use
- Inform user if any operation takes **over** 3 sec.
- Keep core use case as **short** as possible



Question 1

2 / 2 pts

Which of the following are capabilities of a mobile device?

☐ External keyboard and mouse

☒ Location detection

☐
High resolution display with no touch recognition

☒ Memory and storage

☒ Telephony

☒
Sensors and actuators, including temperature sensor, accelerometer and camera

Question 2

2 / 2 pts

Match the following tools to their definition/purpose.

The IDE is

the editor whe ▼

The SDK is

the tools used ▼

The emulator is

a stand-alone i ▼

Gradle is

the build syste ▼

Well done! You seem to have a good grasp of the tools we will be using this semester.

Question 3

2 / 2 pts

When creating a new project in Android Studio, which of the following is **not** a default activity that can be selected?

- ☐ Empty activity
- ☐ Basic views activity
- ☒ List view activity
- ☐ Bottom navigation views activity

Well done! If only a list activity was included it would make life a lot easier, however we will come back to this in module 3.